Linear collider software from event simulation to reconstruction

Rosa Simoniello (CERN)

On behalf of the CLICdp software group

CERN-BINP workshop for young scientists in e+e- colliders

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement no. 654168.

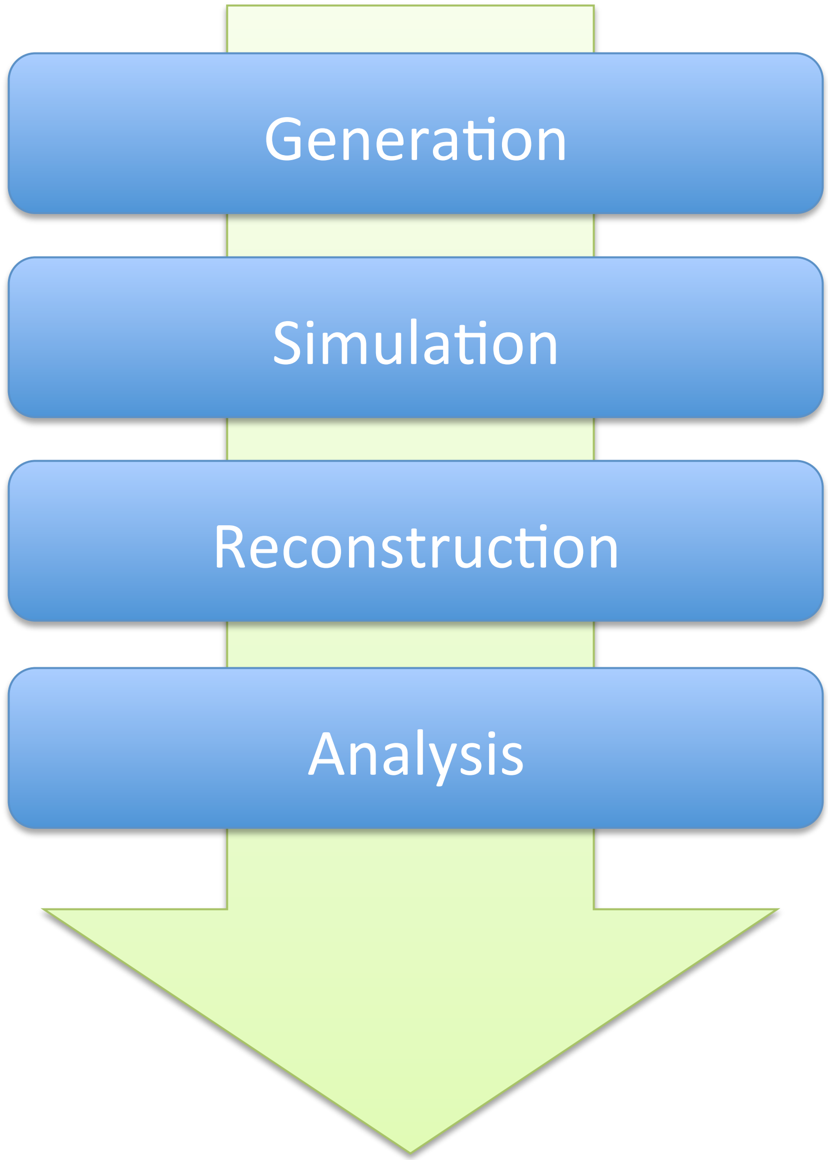
Introduction

- CLIC aims at a new frontier of precisions studies for a broad spectrum of physical analyses covering both SM and BSM
- Good performance is obtained by a combination of excellent hardware and software

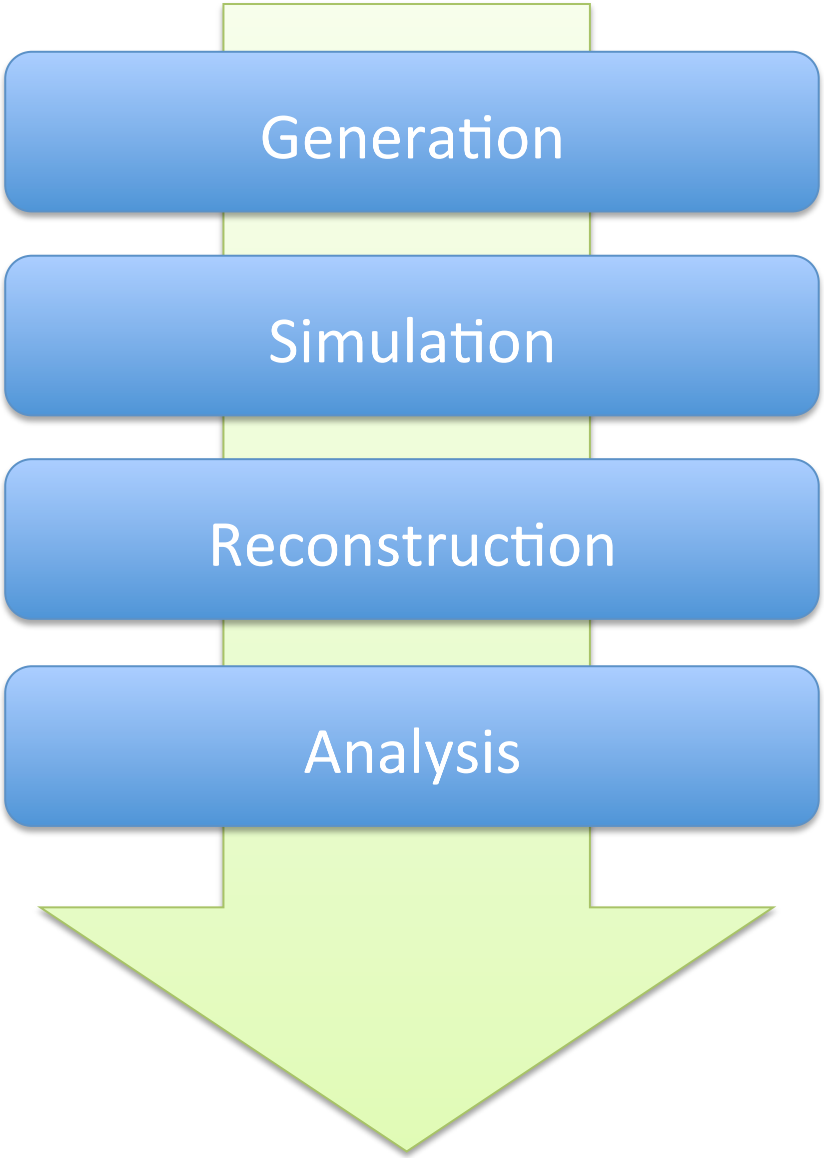
Requirement for software programs and tools:
- *Flexible, generic and robust tools*
  → common software development between CLIC and ILC
- *Efficient use of resources*
  → in particular be able to handle 60BX overlay

- In this talk: overview of the software used in the linear collider community, from detector description, to simulation, to reconstruction programs

Typical HEP software chain



Linear collider software chain

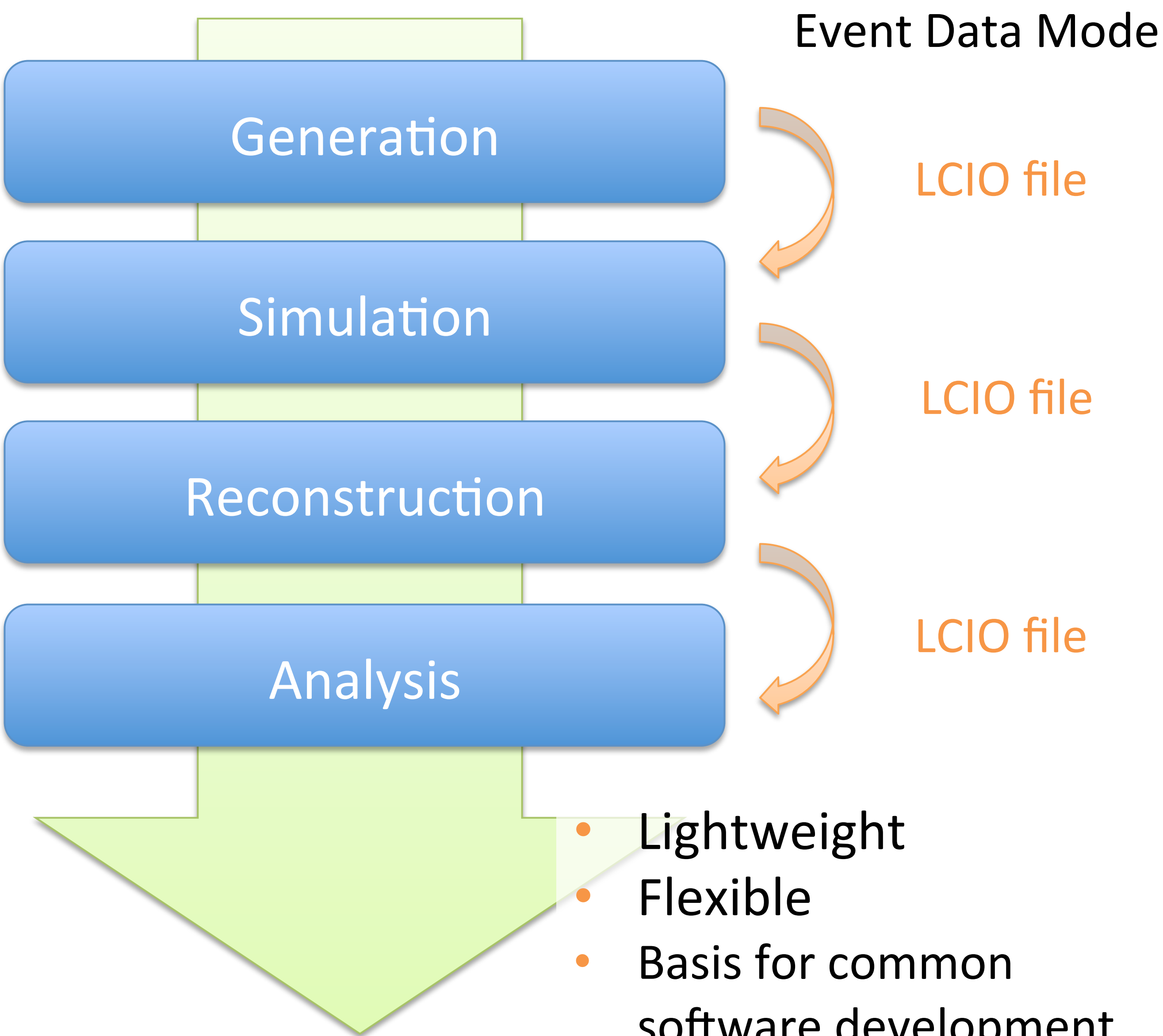


WHIZARD, PYTHIA

Interfaced with GEANT4

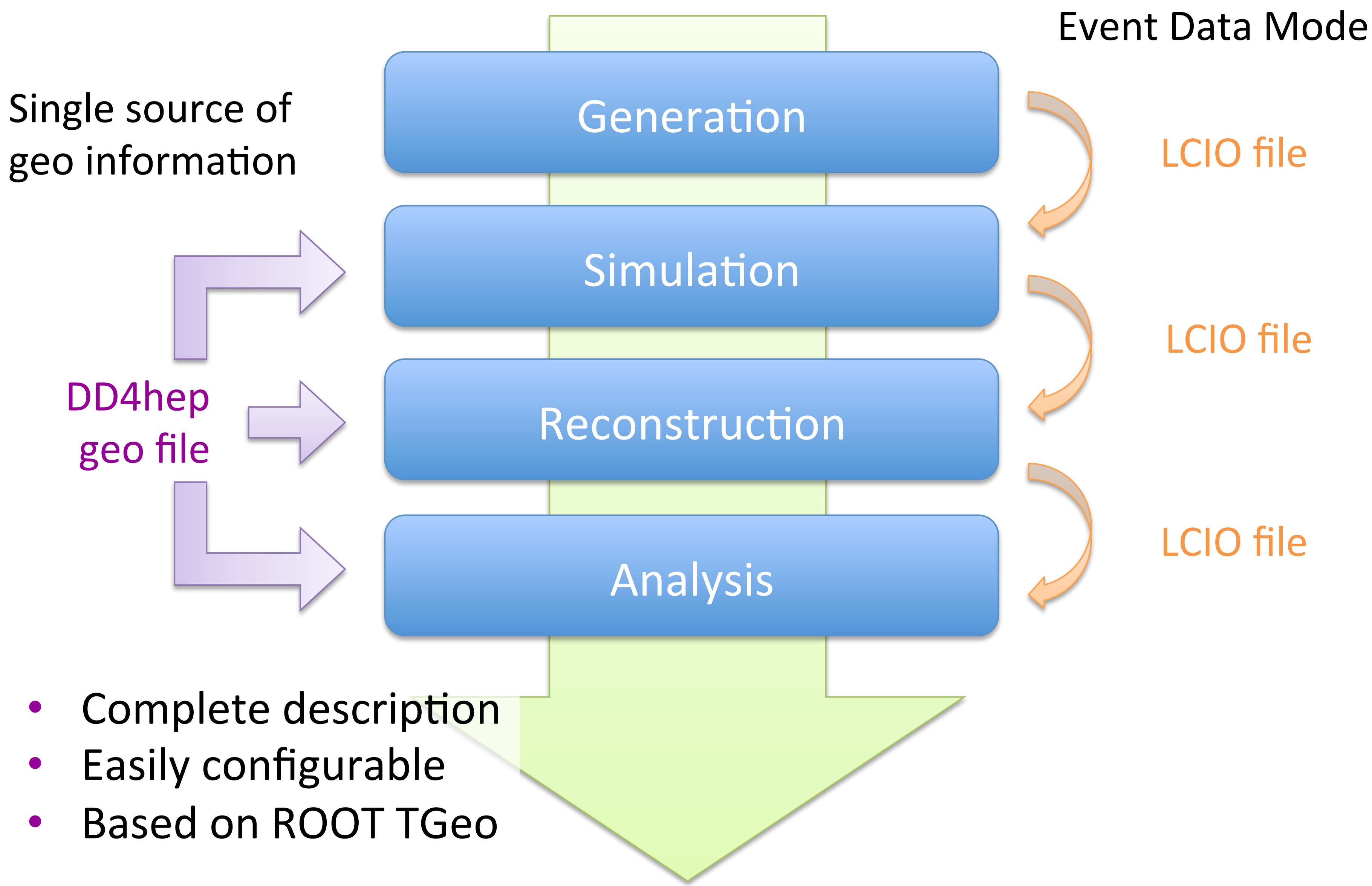
Tracks, PFOs, physics objects, …

Linear collider software chain



- Lightweight
- Flexible
- Basis for common software development

Linear collider software chain

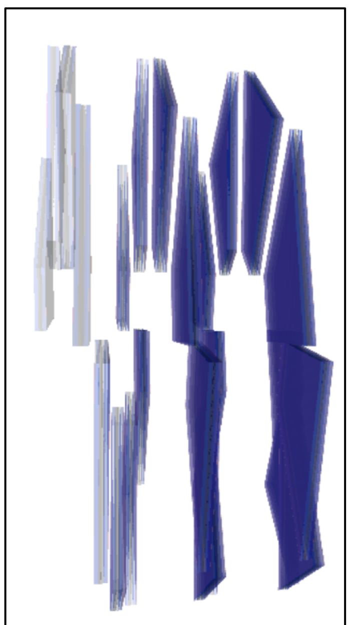


- Complete description
- Easily configurable
- Based on ROOT TGeo

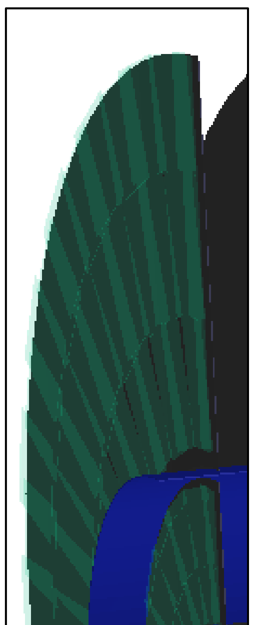
DD4hep geometry – Flexibility

- DD4hep is intended for different experiments
  - *Linear colliders (CLIC, ILC) and test beam prototypes, under investigation for FCC and CEPC*
- DD4hep aims to support the entire experimental lifecycle
  - *Detector design and optimization, construction, operation, data taking and analysis*

Spiral vertex geo

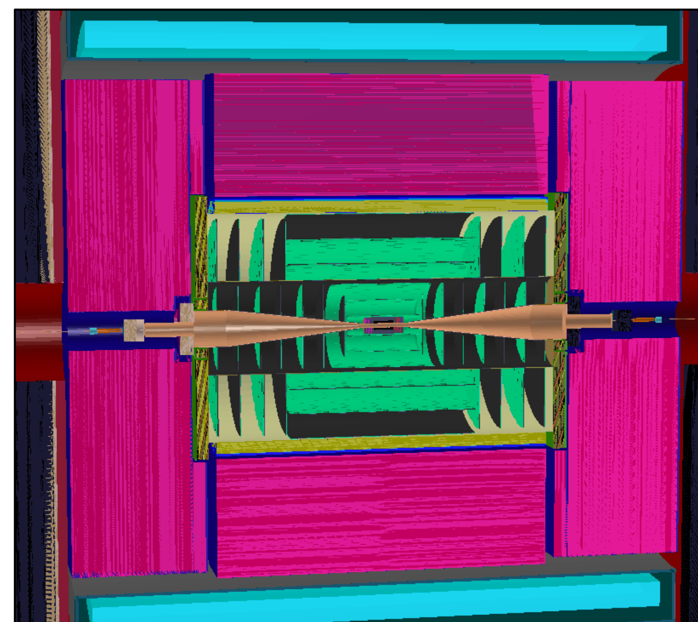


Tracker Disk



Sampling Hadronic Cal

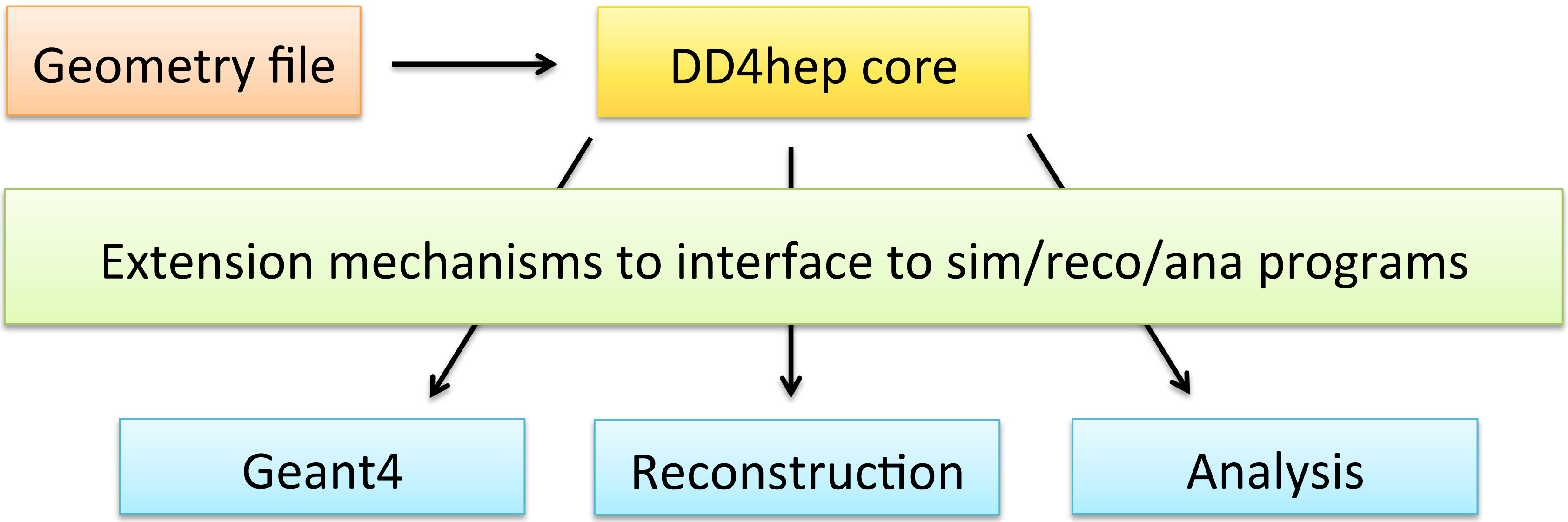
CLIC detector model



DD4hep structure

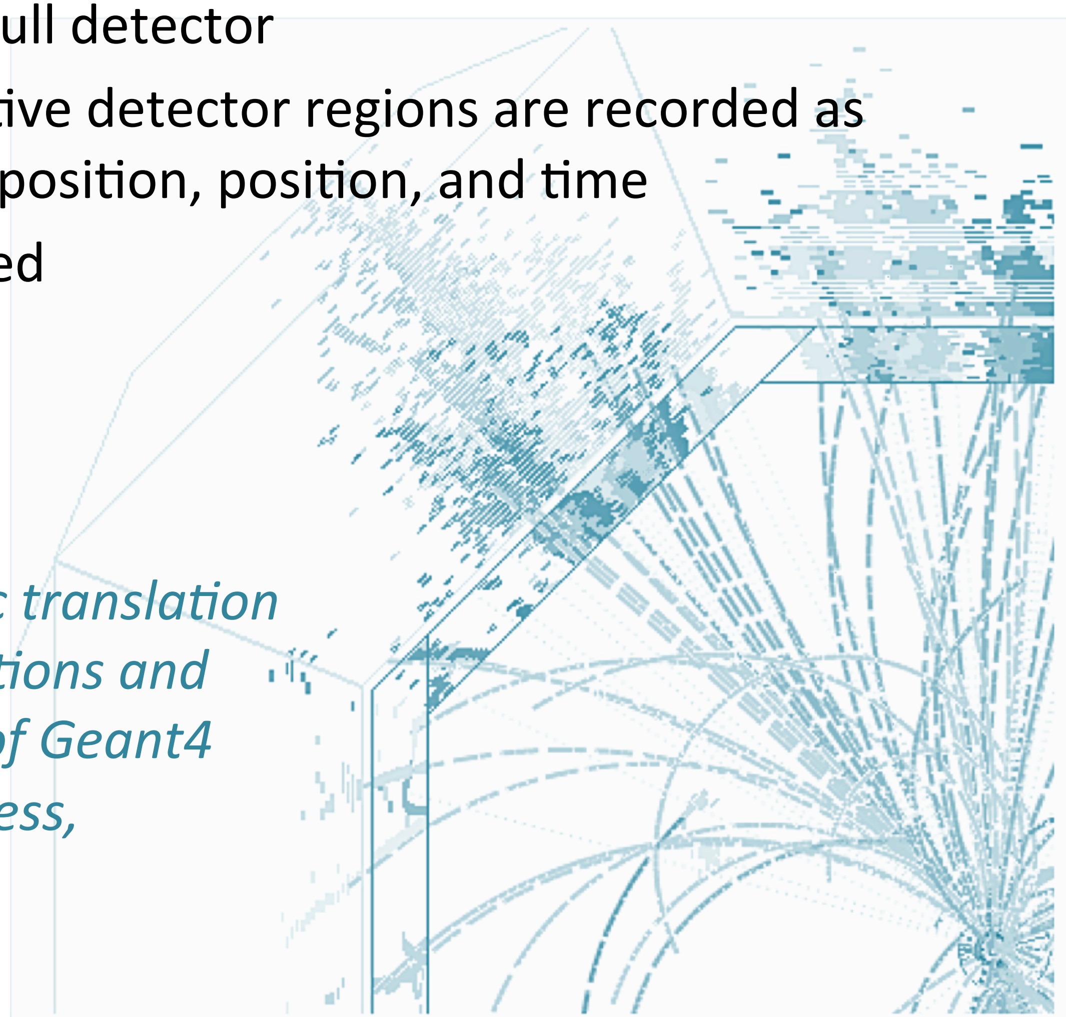
http://test-dd4hep.web.cern.ch/test-dd4hep/doxygen/html/index.html

- Simple and compact xml configuration file with geo information:
  - Dimensions, layer/module composition with materials and thickness
- Interfaces of DD4hep to the simulation and reconstruction programs are realised via extensions invisible to the general users



Simulation

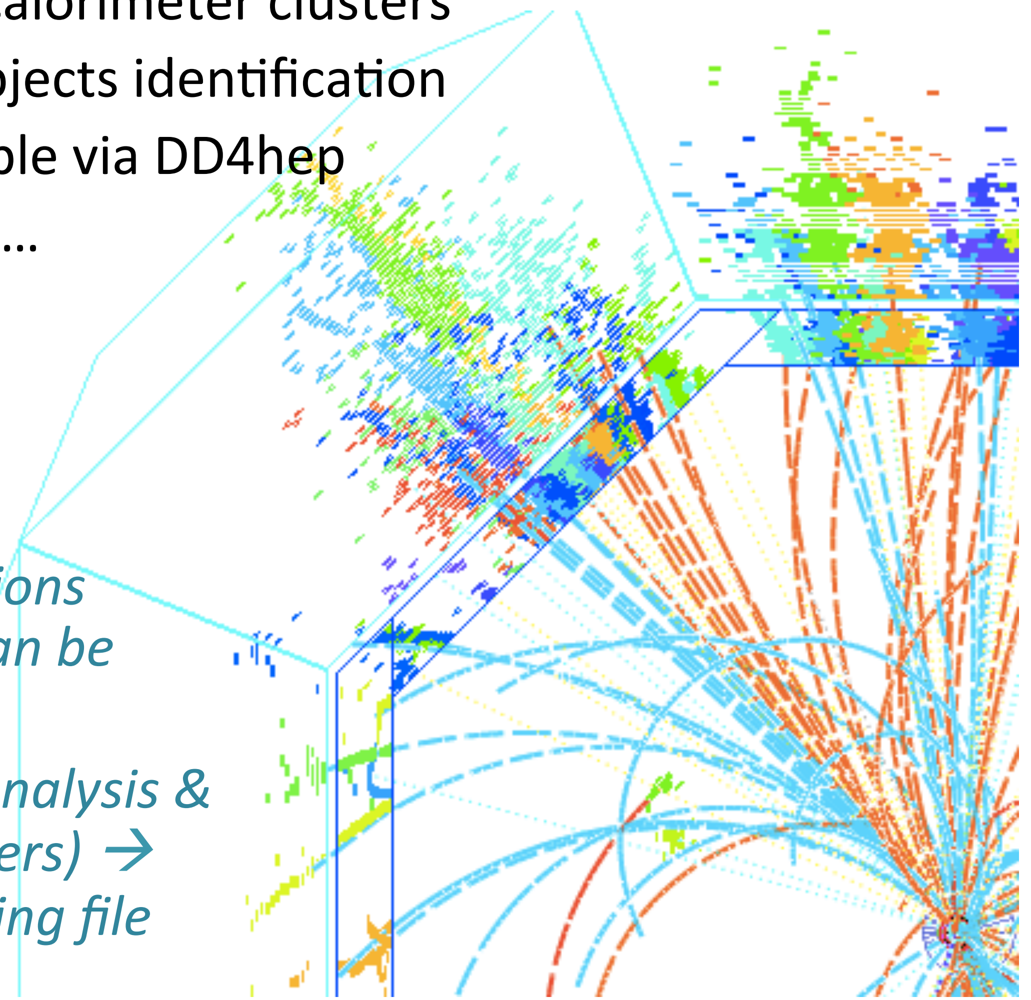
- Simulation applications for passage of particles through matter
  - Each particle provided by the event generator is propagated through a model of the full detector
  - Energy deposits in sensitive detector regions are recorded as *hits* with total energy deposition, position, and time
- In HEP, Geant4 is usually used



- *DD4hep provides an automatic translation between geometry representations and an easy python configuration of Geant4*
- *Simulation is an intensive process, multithreading available*

Reconstruction

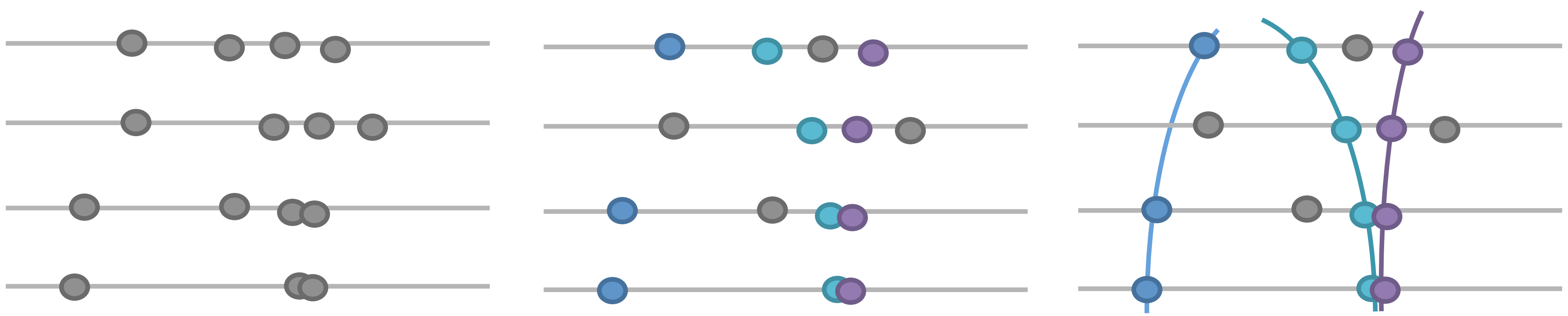
- Reconstruction chain starts with digitisation
  - Simulate the detector response including instrumental effects
- Then reconstruct tracks and calorimeter clusters
  - Basic inputs for physics objects identification
- Geometry information available via DD4hep
  - Number of layers, B field, …



- *Provides library of shared functions through a common interface, can be used by all experiments*
- *MARLIN framework (Modular Analysis & Reconstruction for LINear colliders) → C++, configurable via xml steering file*

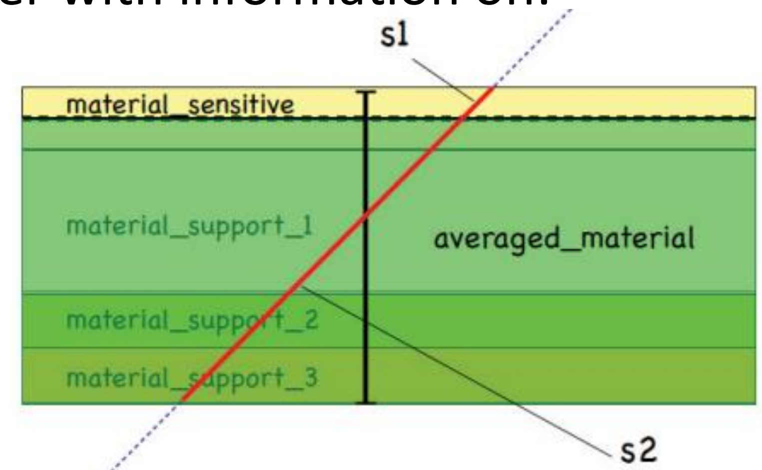
Track reconstruction

- Pattern recognition: associate hits that belong to one particle
  - *Several approaches available* to use also in combination
- Track fit: extract track parameters given a set of hits: *momentum, direction and impact parameters*
  - Methods available based on *Kalman Filter* or *General Broken Line*
- Mix-and-match different pattern recognition algorithms with different track fitters
- Interface with the DD4hep geometry

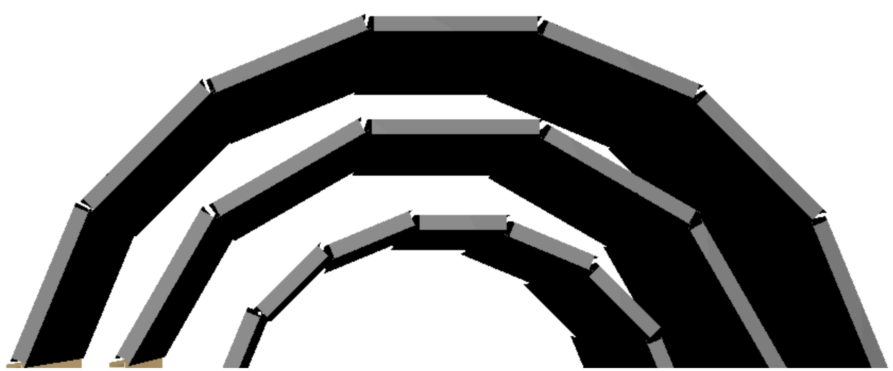


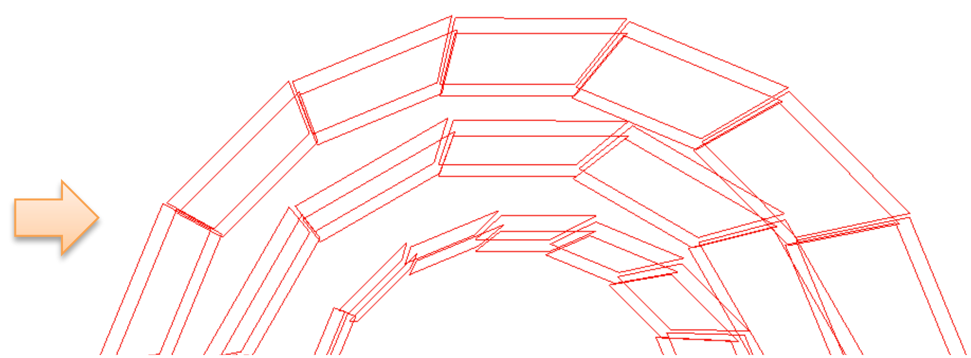
DD4hep interface with reconstruction

- Tracking needs a special interface to geometry for track intersection/ extrapolation and multiple scattering/energy loss effects
- Plugin mechanism → For each detector element volume attach a surface in the middle of the sensitive layer with information on:
  - *Local coordinate system*
  - *Volume thickness and material*
    → materials in a compound volume automatically averaged



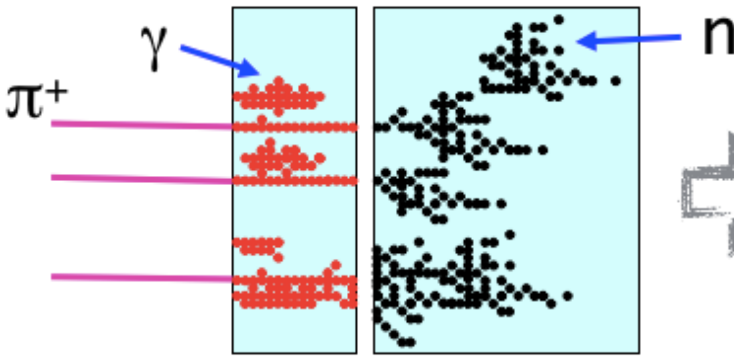
→ Track reconstruction works with all detectors described in DD4hep





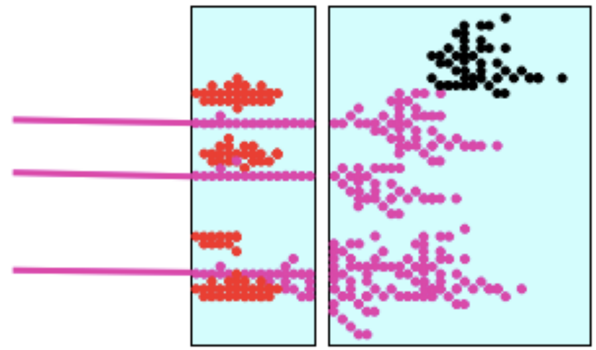
Particle Flow Analysis: concept

Classical approach



$E_{JET} = E_{ECAL} + E_{HCAL}$

Particle flow approach



$E_{JET} = E_{TRACK} + E_{\gamma} + E_{n}$

Typical jet composition:
- 60% charged particles
- 30% photons
- 10% neutrons

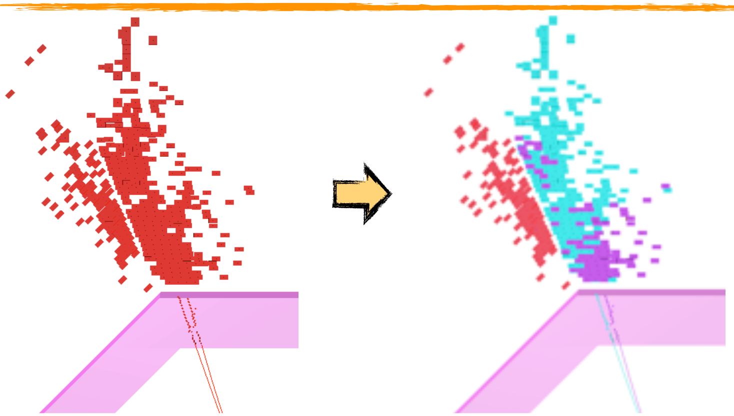
Always use the best info you have:
- 60% => tracker
- 30% => ECAL
- 10% => HCAL

- Requires highly granular calorimeters to resolve deposits from different particle and *sophisticated software to make correct associations*

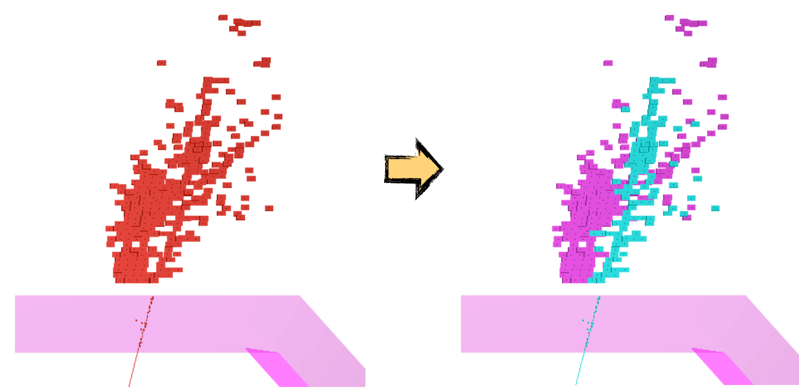
Particle Flow Analysis: Pandora

"The Pandora Software Development Kit for Pattern Recognition" → EPJC.75.439

- Pandora: approach to automated computer pattern recognition
- Exploit calorimeter granularity to gradually build-up picture of events
- More than 70 algorithms to address specific event topologies, with very few mistakes, and to avoid accidental merging of separate particles
  - *Cone Clustering, Topological Association, Track-Cluster Association, Reclustering Association, Fragment Removal, PFO Construction Algorithms*
- Interface with DD4hep available



1. Multiple tracks associated to single cluster – split cluster.



2. Cluster energy much greater than track momentum – split cluster.

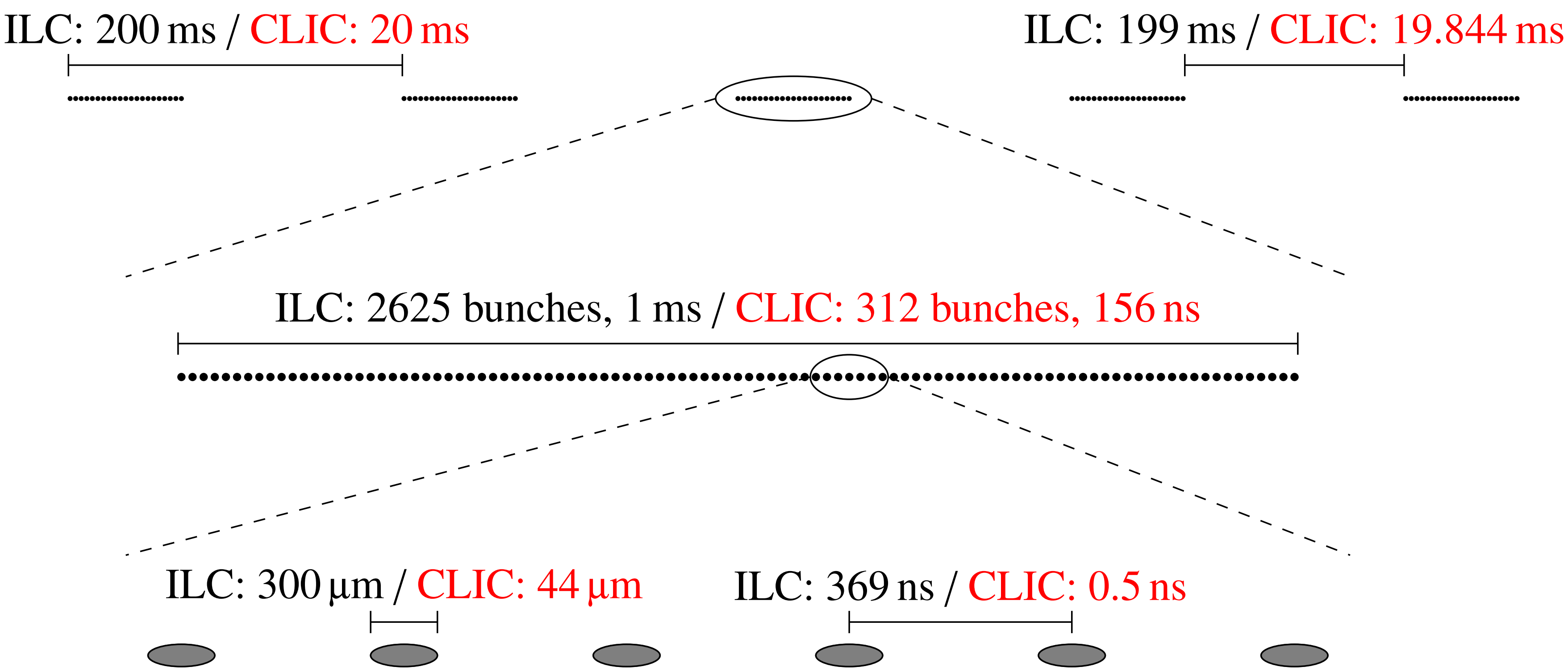
Conclusions

- Effort to have as much as possible possible common and flexible software in the linear collider community and outside
  → iLCSoft http://ilcsoft.desy.de/portal
  - DD4hep is a new concept for detector geometry description shared by different experiments (also outside the linear colliders)
  - Track reconstruction programs may rely on some specifics of the experiment but they share a set of common functions (discussion on-going for a possible inter-experimental shared library)
  - Pandora (particle flow analysis) is a standalone package used by linear colliders community, LAr neutrino experiments, CEPC, FCC

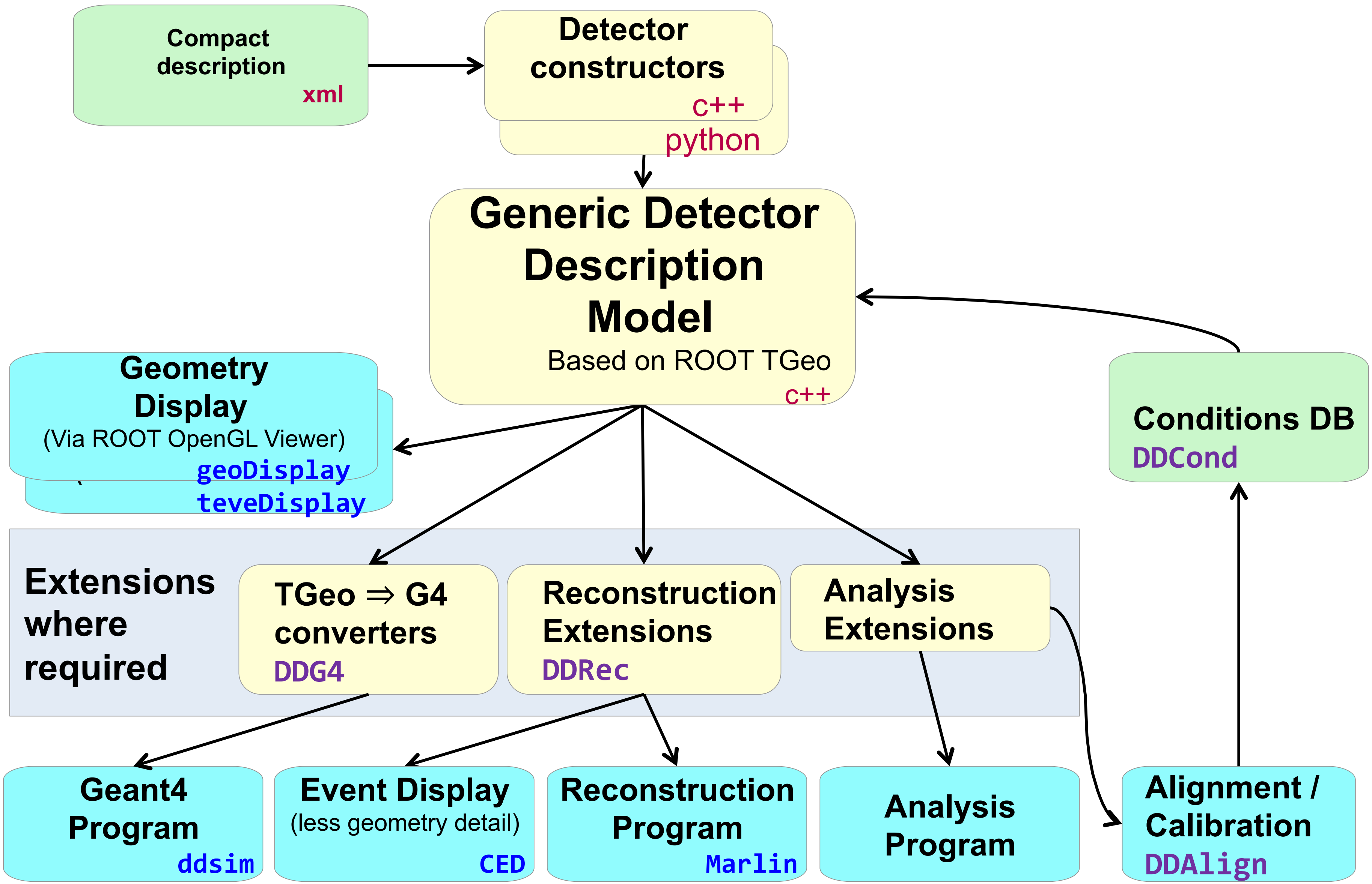
BACK-UP

Beam structure

For: **ILC at 500 GeV** and **CLIC at 3 TeV** centre-of-mass energy



DD4hep



Compact description
xml
Detector constructors
c++
python
Generic Detector Description Model
Based on ROOT TGeo
c++
Geometry Display
(Via ROOT OpenGL Viewer)
geoDisplay
teveDisplay
Conditions DB
DDCond
Extensions where required
TGeo ⇒ G4 converters
DDG4
Reconstruction Extensions
DDRec
Analysis Extensions
Geant4 Program
ddsim
Event Display
(less geometry detail)
CED
Reconstruction Program
Marlin
Analysis Program
Alignment / Calibration
DDAlign

Geometry file

- Simple and compact xml configuration file with geo information:
  - Dimensions, layer/module composition with materials and thickness
- Actual volume building in C++: general and simple to allow maximum flexibility but users can easily add more level of details

```
<detector name="ECalBarrel" type="GenericCalBarrel_o1_v01" id="DetID_ECal_Barrel" readout="ECalBarrelCollection" >
  <dimensions numsides="ECalBarrel_symmetry" rmin="ECalBarrel_inner_radius" z="ECalBarrel_half_length*2" />
  <staves vis="ECalStaveVis" />
  <layer repeat="40" vis="ECalLayerVis">
    <slice material = "TungstenDens24" thickness = "1.90*mm" vis="ECalAbsorberVis" radiator="yes"/>
    <slice material = "G10"            thickness = "0.15*mm" vis="InvisibleNoDaughters"/>
    <slice material = "GroundOrHVMix"  thickness = "0.10*mm" vis="ECalAbsorberVis"/>
    <slice material = "Silicon"        thickness = "0.50*mm" sensitive="yes" limits="cal_limits" vis="ECalSensitiveVis"/>
    <slice material = "Air"            thickness = "0.10*mm" vis="InvisibleNoDaughters"/>
    <slice material = "siPCBMix"       thickness = "1.30*mm" vis="ECalAbsorberVis"/>
    <slice material = "Air"            thickness = "0.25*mm" vis="InvisibleNoDaughters"/>
    <slice material = "G10"            thickness = "0.75*mm" vis="InvisibleNoDaughters"/>
  </layer>
</detector>
```

Interface with simulation

- DD4hep supports simulation activities with Geant4 providing an automatic in memory translation mechanism between geometry representations
- Easy configuration of Geant4 applications provided via python executable with command-line/steering-file arguments

```
ddsim --compactFile myGeo.xml --enableGun
--gun.particle mu- --gun.energy 10*GeV --numberOfEvents 10
```

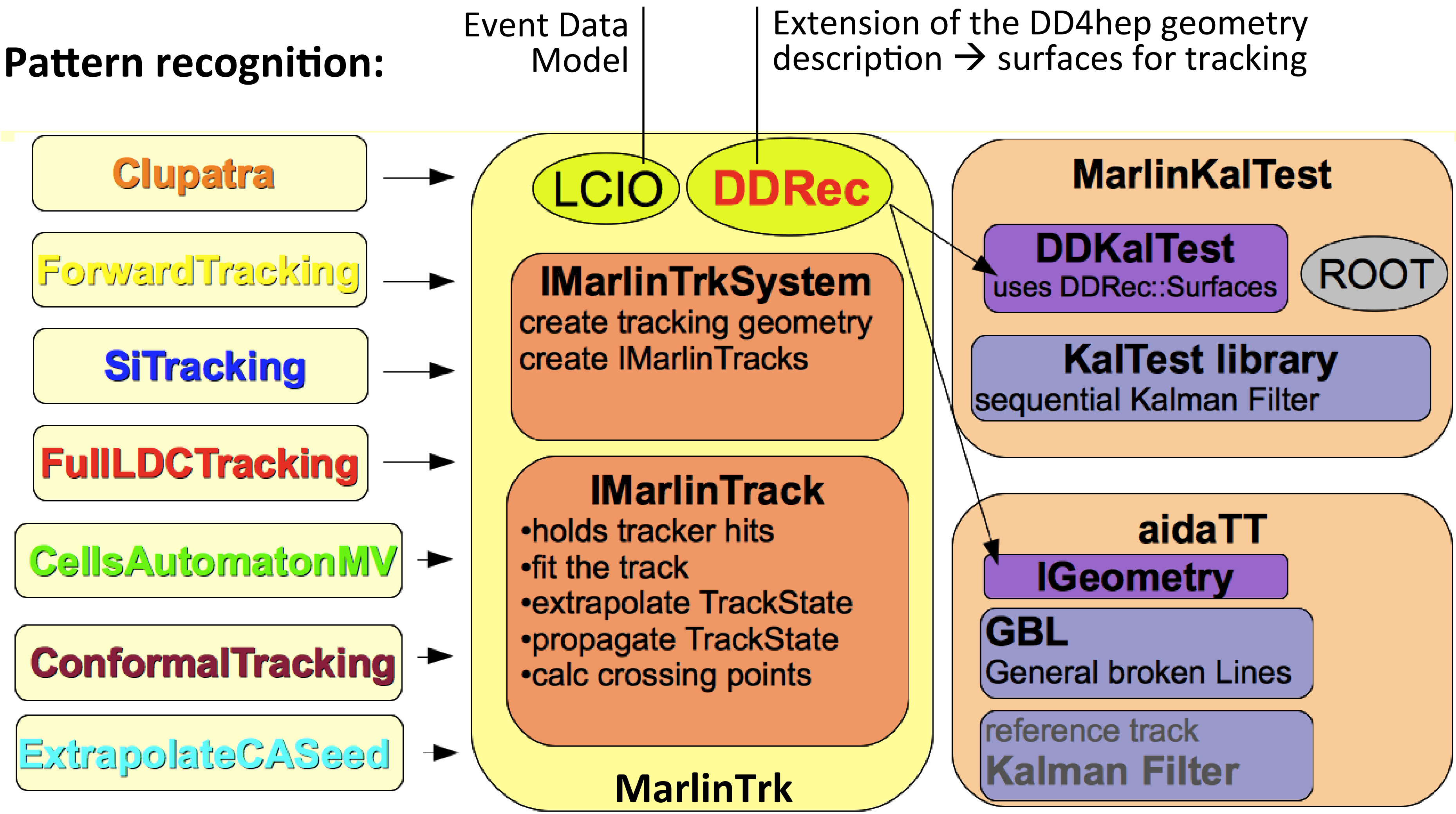
- Many more options available!
    - *Physics lists, generator file, filters, range cuts, etc…*
- Automatic completion available → Just tab to see all the options!

> - *Validation OK, ready for large scale production*
> - *Optimised for time performance, multithreading available*

Tracking software in ILCSOFT

- Requirements: *modularity* and *flexibility* → used by different experiments
- C++ implementation

Pattern recognition:



Cellular Automaton (CA)

- Based on the creation of cells:
  → a cell is a connection between two hits
  → every cell has a weight
- Local method → used local criteria (like angle between two cells) to connect cells
- Update cell weight: increase cell weight by 1 for each previous cell that passed criteria
- Track candidates: start from high weights and follow the connections
- Choose best tracks according to the χ^2 probability, number of tracker hits