22.02.2020

Muon Collider detector simulation

Status. Challenges. Plans.

N. Bartosik
for the MuonCollider-Detector-Physics Group

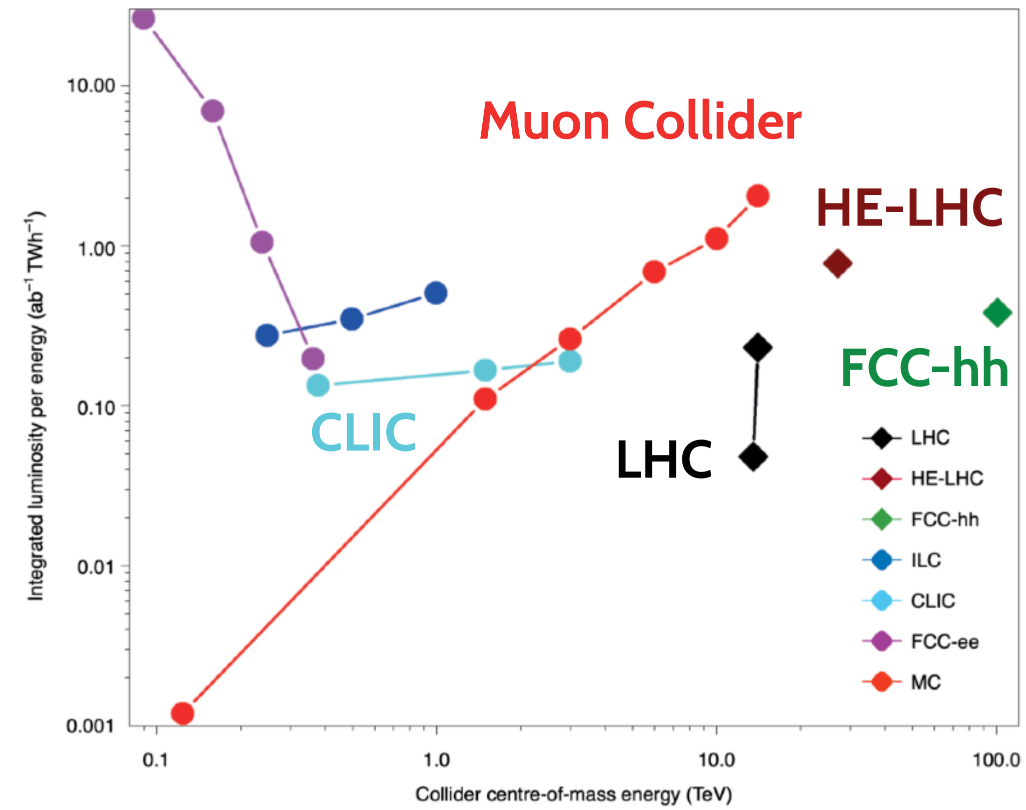
INFN Torino

Muon Collider: unique machine

Muon Collider allows to combine in a single machine high precision of e^+e^- colliders and high energy reach typical for *pp* colliders

- muons are elementary particles, like e^+/e^-
- much smaller synchrotron radiation losses

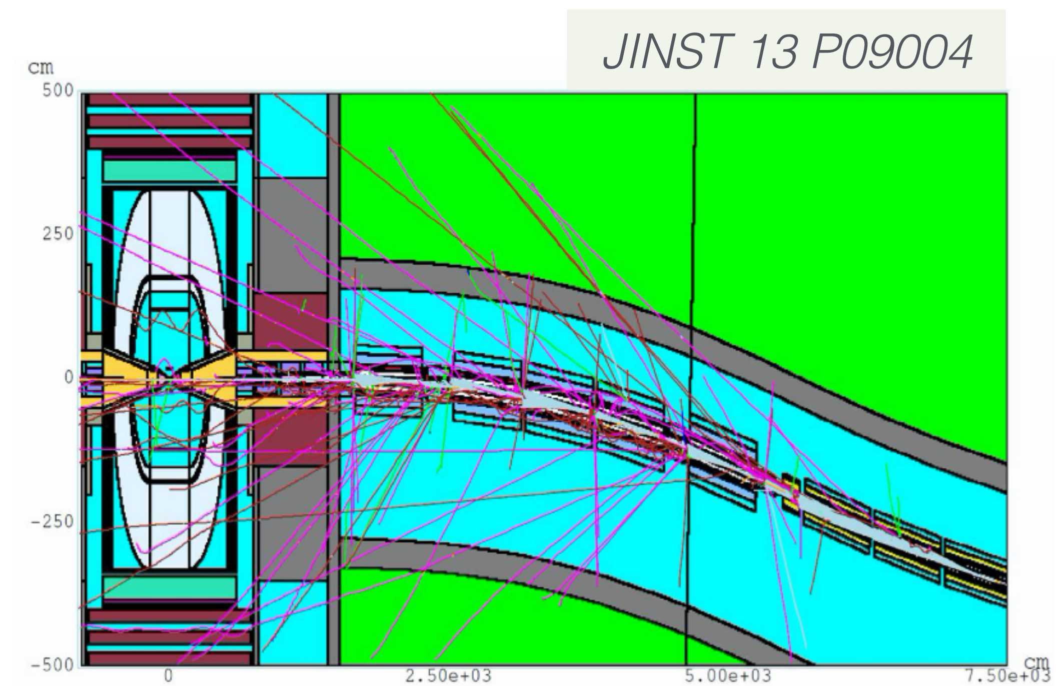
At $\sqrt{s}$ = 3 TeV and higher Muon Collider is the most energy efficient machine



Unique advantages come at a price of challenging machine design (short muon lifetime) **and harsh Beam Induced Background (BIB)**

↳ interaction of secondary/tertiary muon decay products with the accelerator complex and the Machine-Detector Interface (MDI)

We start with $\sqrt{s}$ = 1.5 TeV design that has been studied the most by the MAP program before →



Beam Induced Background

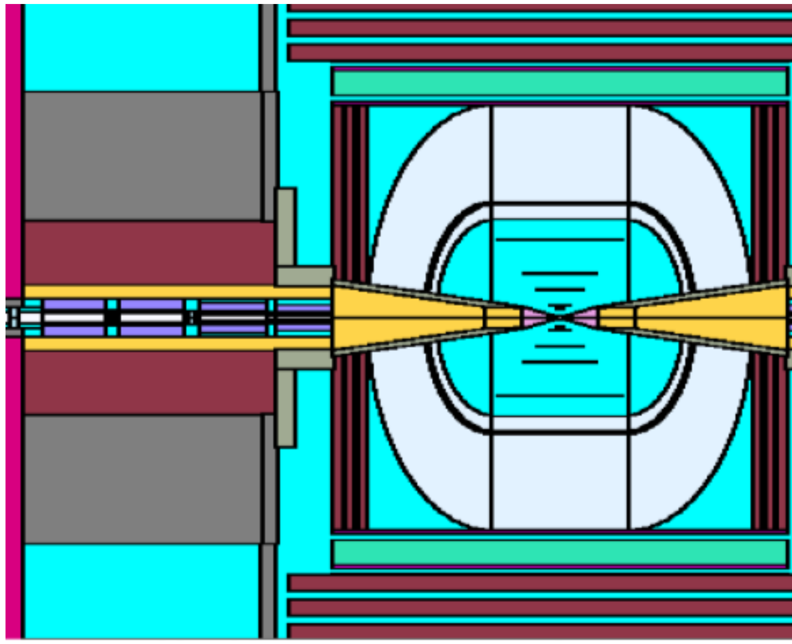
For 0.75 TeV beams at 2×10^{12} μ/bunch → 4×10^{5} muon decays/m in a single BX

Design of the accelerator complex directly affects the BIB characteristics

↳ interactions of the $\mu^{\pm}$ beams with ±200m of the machine lattice have to be simulated for every specific accelerator design at every specific $\sqrt{s}$

Essential component is the MDI – tungsten nozzles

- shape optimised by MAP for the specific $\sqrt{s}$
- reduce the BIB rate by a factor ~500



Currently we are using a BIB sample simulated with ***MARS15*** for $\sqrt{s}$ = 1.5 TeV within the MAP program

New independent workflow at the finalising stage based on ***FLUKA + FlukaLineBuilder***

Result of a simulation → list of stable particles reaching the detector region in a single bunch crossing (BX) *(mostly soft photons, neutrons, electrons)*

- collected at the outer surface of the detector and the MDI
- 2 × 180M particles → full simulation required to evaluate detector performance

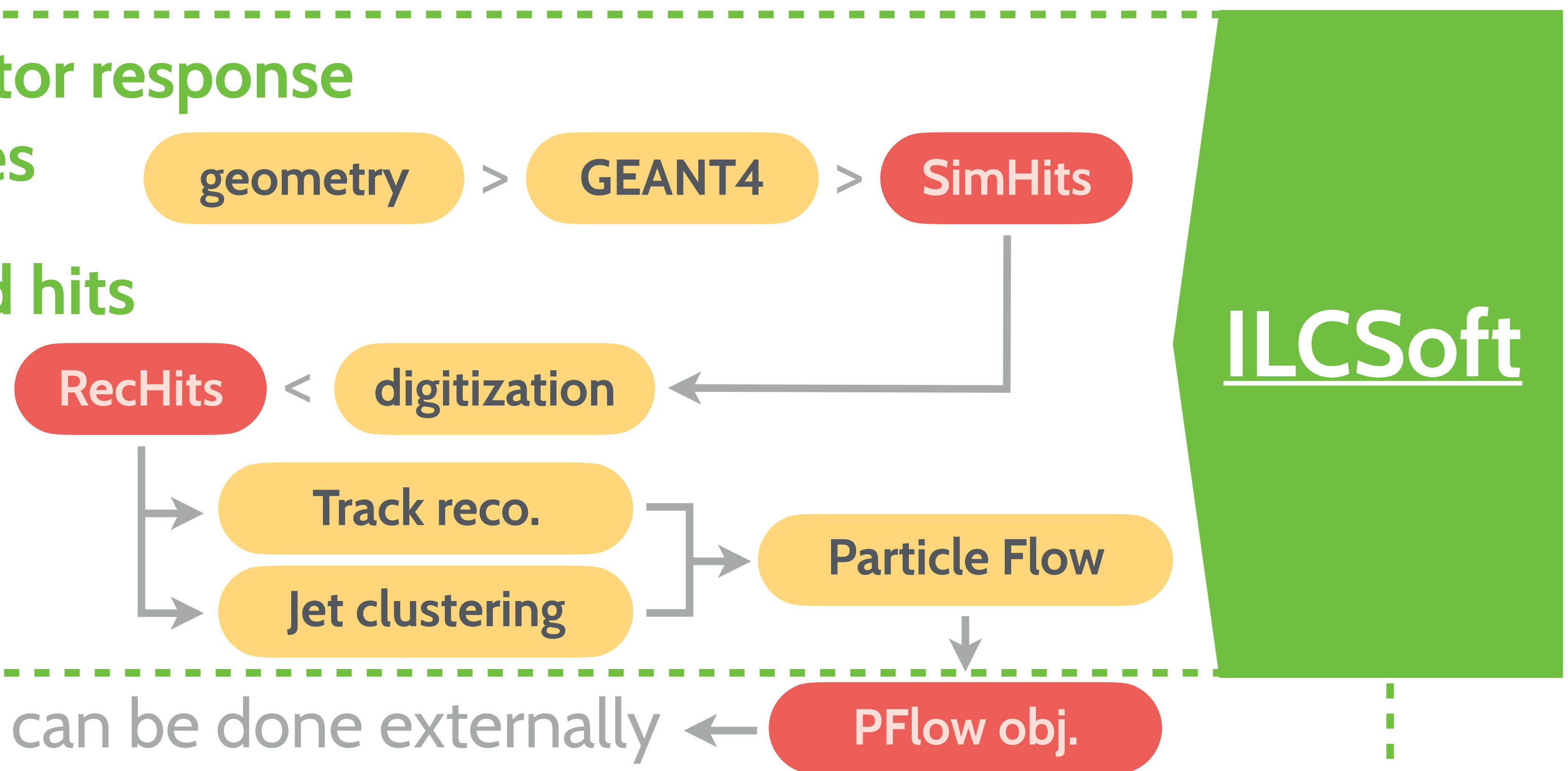
Software stack: from CLIC

Software framework of the CLIC experiment chosen as a starting point

- also a lepton-lepton collider, modern software, in active use and development

Key components of a physics analysis using full simulation:

1. **generation of the main process (ME + PS)** ← done externally (Whizard+Pythia)
2. **simulation of the detector response to the incoming particles**
3. **conversion of simulated hits to reconstructed hits**
4. **reconstruction of tracks/jets/particles**
5. **higher-level analysis** ← can be done externally



All the simulation and reconstruction done within a single framework

Most of package modifications/extensions specific to the Muon Collider are maintained in the separate public Muon Collider Software GitHub repository

Key software components

1. **LCIO** **[L**inear **C**ollider **I/O]**
   Consistent storage of event data (**MCParticles, SimHits/RecHits, higher-level and custom objects**) using the ***.slcio** file format

2. **DD4hep** **[D**etector **D**escription **for H**igh **E**nergy **P**hysics**]**
   Efficient and flexible detector geometry description with the interface to GEANT4 and simulation/reconstruction software

3. **Marlin** **[M**odular **A**nalysis & **R**econstruction for the **Lin**ear collider**]**
   Collection of processors for isolated tasks that can be chained into the necessary workflow by means of XML configuration files

Centralised software revisions distributed through Docker and Singularity containers + manual build instructions: documentation

Most of the framework overlaps with the Key4HEP stack

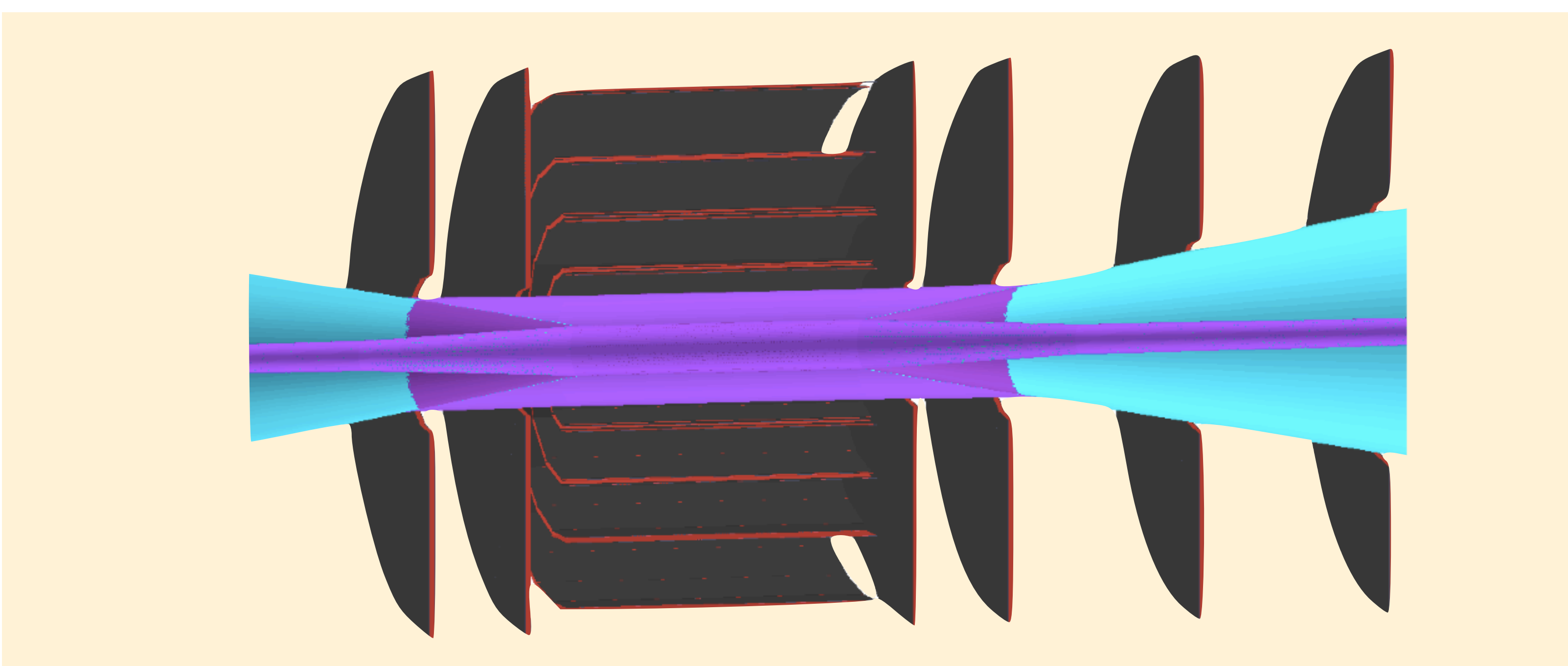
↳ we stay with the current CLIC framework for the near future, but will eventually transition to **Key4HEP**

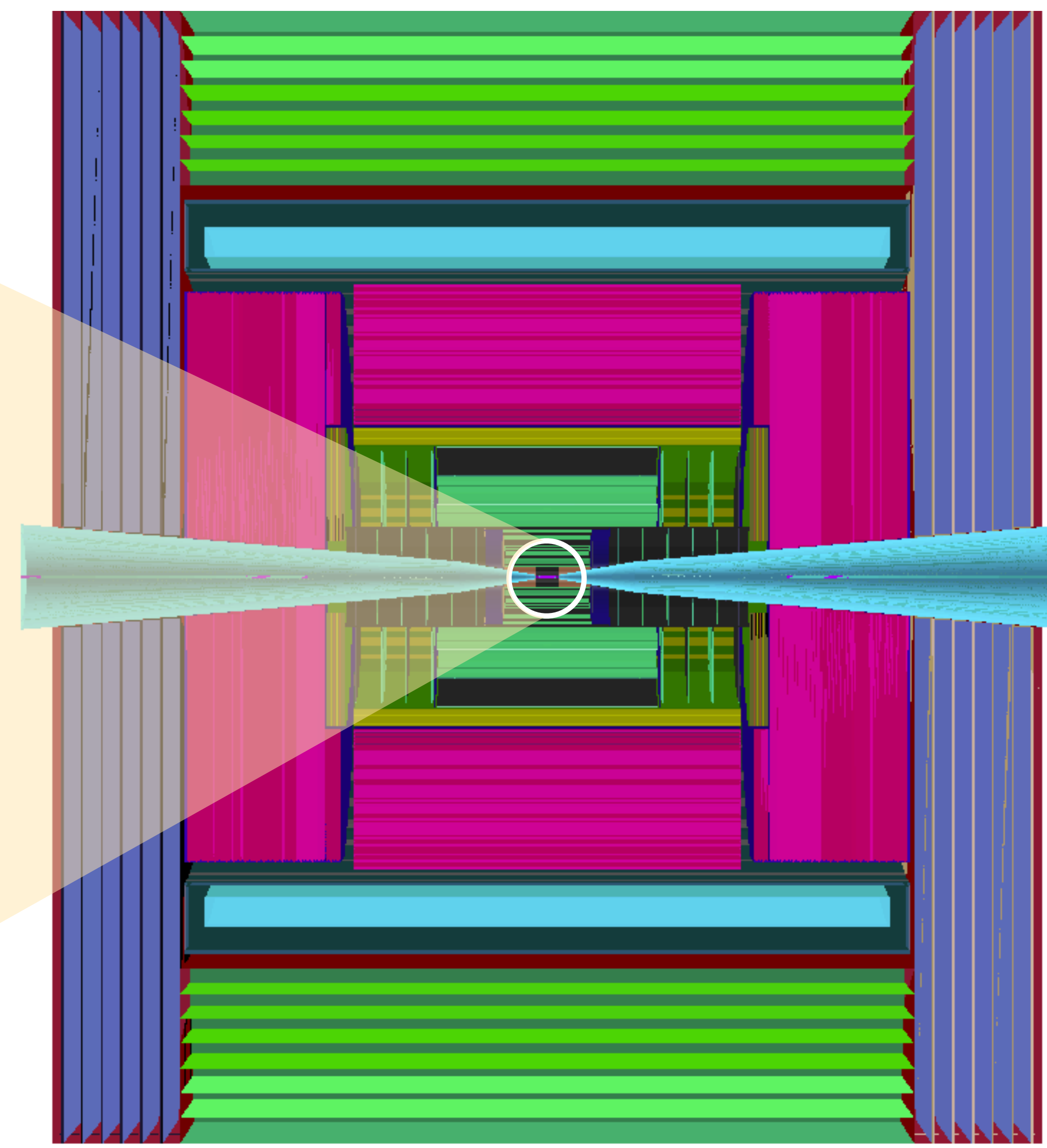
Detector: based on CLIC

Current geometry based on the CLIC detector design: implemented in DD4hep

Several modifications made for the Muon Collider environment:

- inserted BIB-absorbing **tungsten nozzles** developed by [MAP]
- enlarged inner openings of endcap detectors to fit the nozzles inside
- optimised layout of the Vertex detector to reduce occupancy near the tips of the nozzles



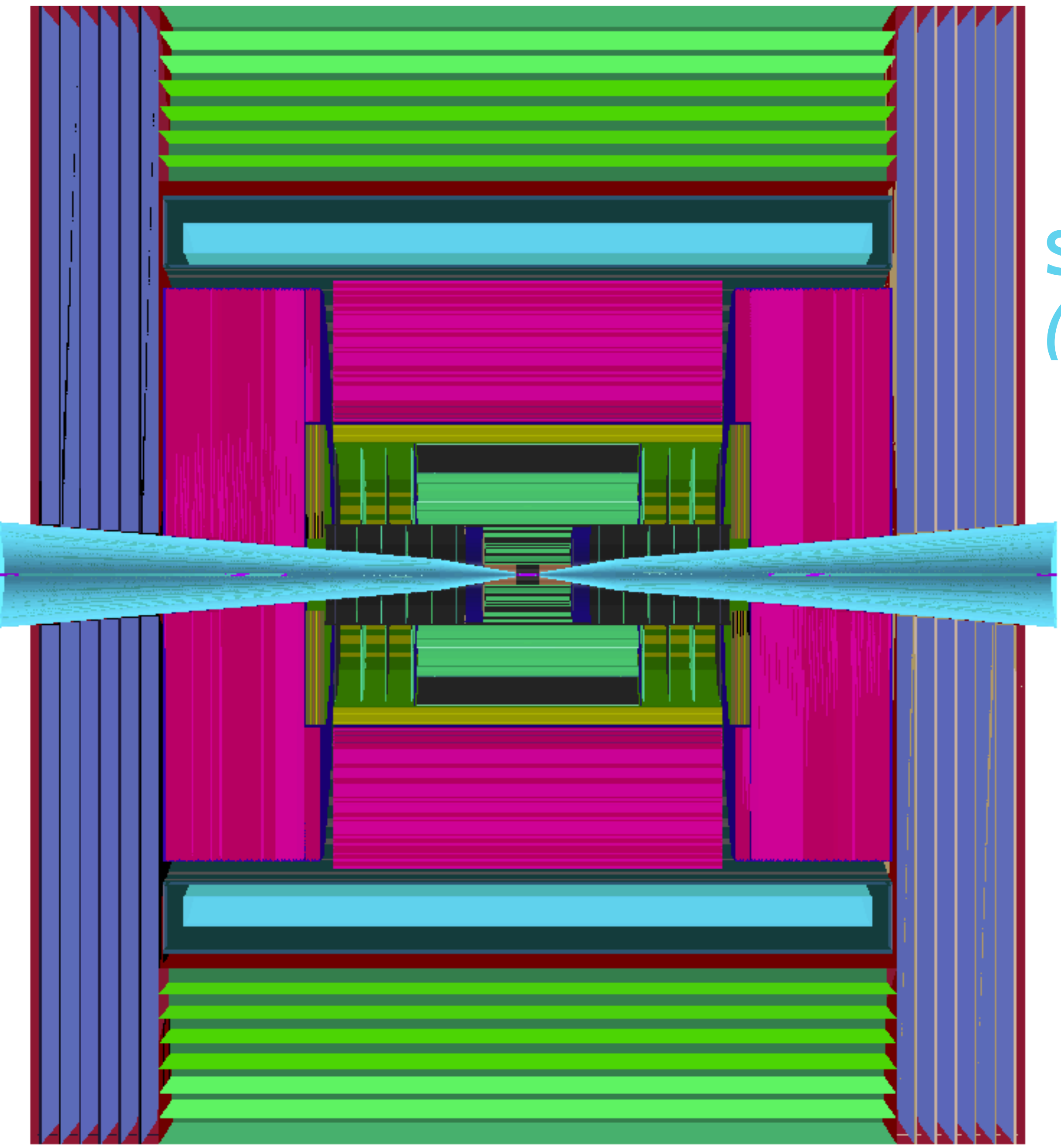


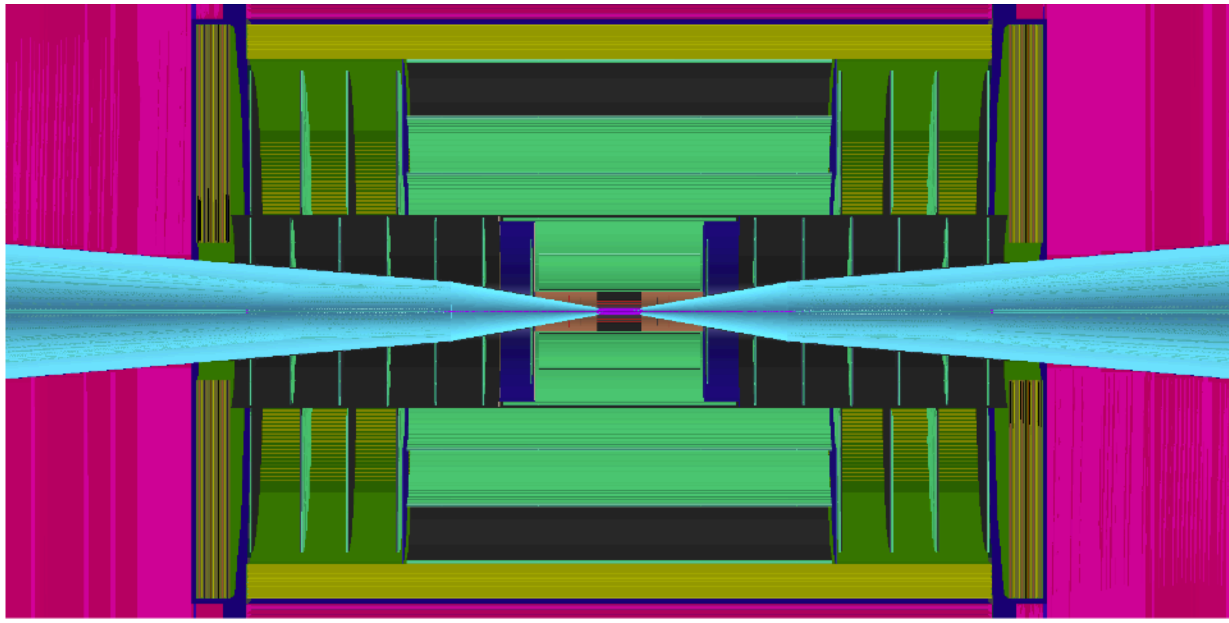
Frozen geometry for the SnowMass studies
available on [GitHub]

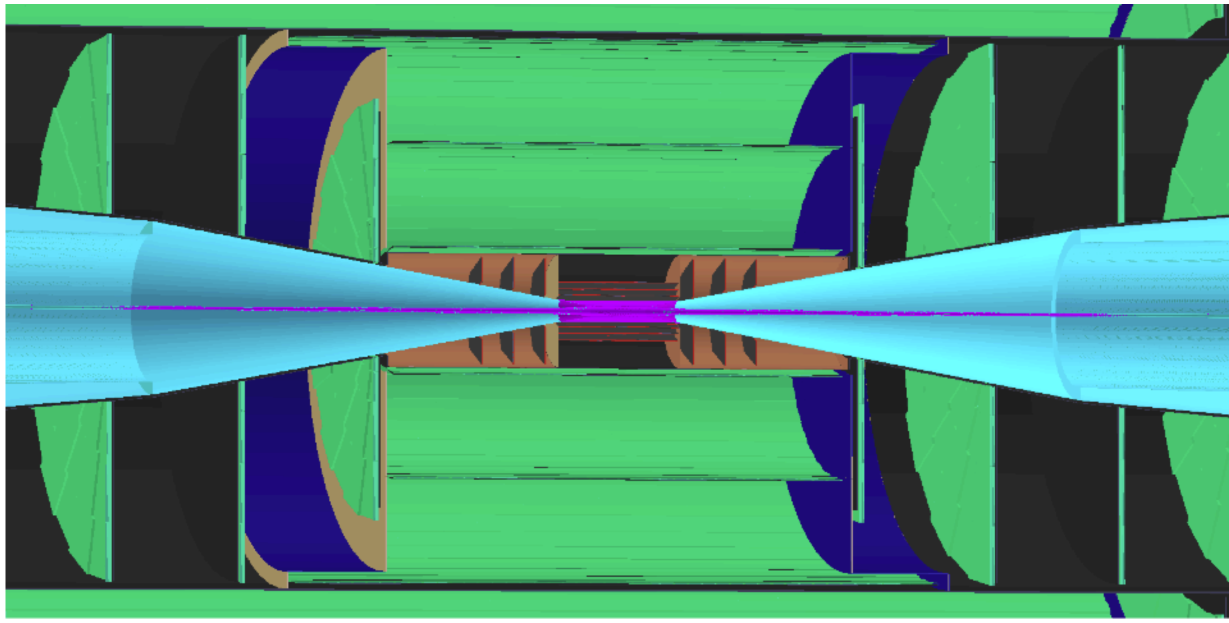
Detector: layout

Using the CLIC framework to perform GEANT4 simulation of detector hits:

Vertex Tracker,	Inner Tracker,	Outer Tracker,	ECAL,	HCAL,	Muon Det.
(4 + 4×2) × 2 *double layers*	3 + 7×2	3 + 4×2	*40 layers* W+Si	*60 layers* Fe+scint	7 + 6×2 Fe+RPC







Hit digitisation

SimHits from GEANT4 are converted to RecHits using Marlin processors

Calorimeter/Muon hits: SimCalorimeterHit → CalorimeterHit class

- use real detector granularity, 1 hit/cell
- SimCalorimeterHit keeps track of ≥1 MCParticle contributions: *E, time, MCP ref.*
- CalorimeterHit sums contributions in a fixed time window ← σ_t, $t_{showering}$

Tracker hits: SimTrackerHit → TrackerHitPlane class *(nearly identical)*

- no real pixel/strip granularity; hits assigned to a sensor
- position and time smeared according to σ_u, σ_v, σ_t
- only hits within a fixed time window after smearing by σ_t are kept
- minimal computations → very fast with BIB

A new Tracker digitiser with realistic granularity under development

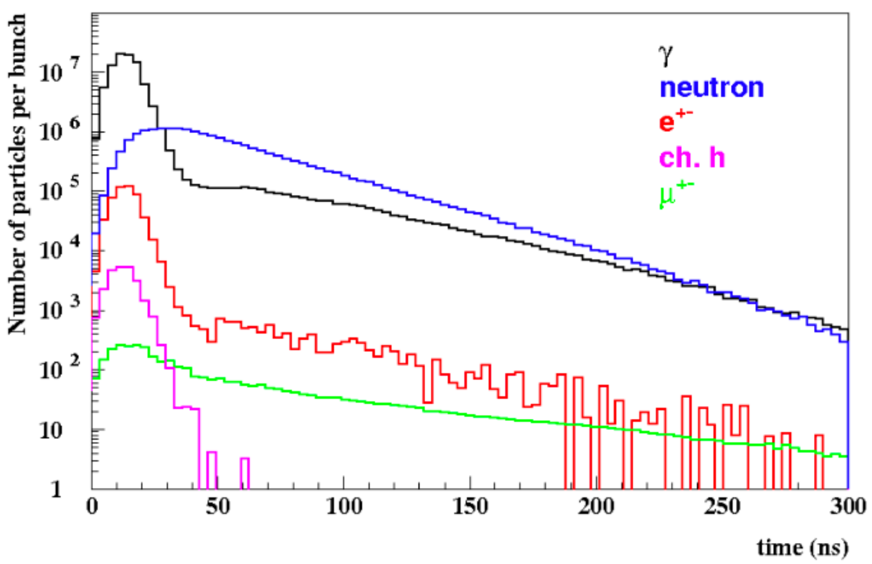
- simulating charge sharing between pixels, consistent treatment of time
- more computations → might become an issue with BIB hit multiplicities

BIB simulation: GEANT4

1 signal event corresponds to 1 BX including ~4×10⁸ BIB particles

Simulating BIB particles form 1 BX takes ~130 days (**QGSP_BERT_HP** at 1 thread)
but not all BIB particles are relevant for detector performance

Large fraction of BIB particles create hits too late: after the 10ns readout window

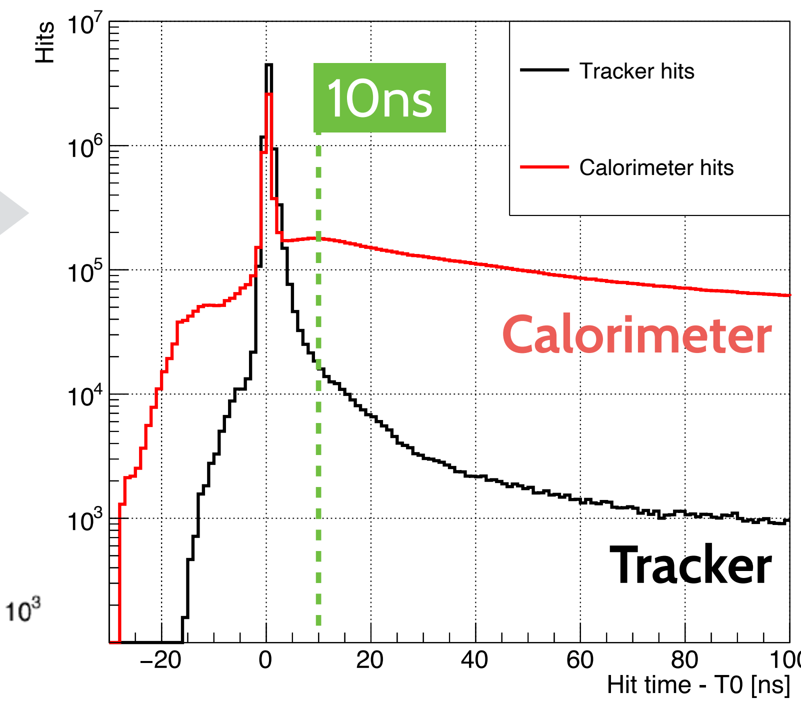


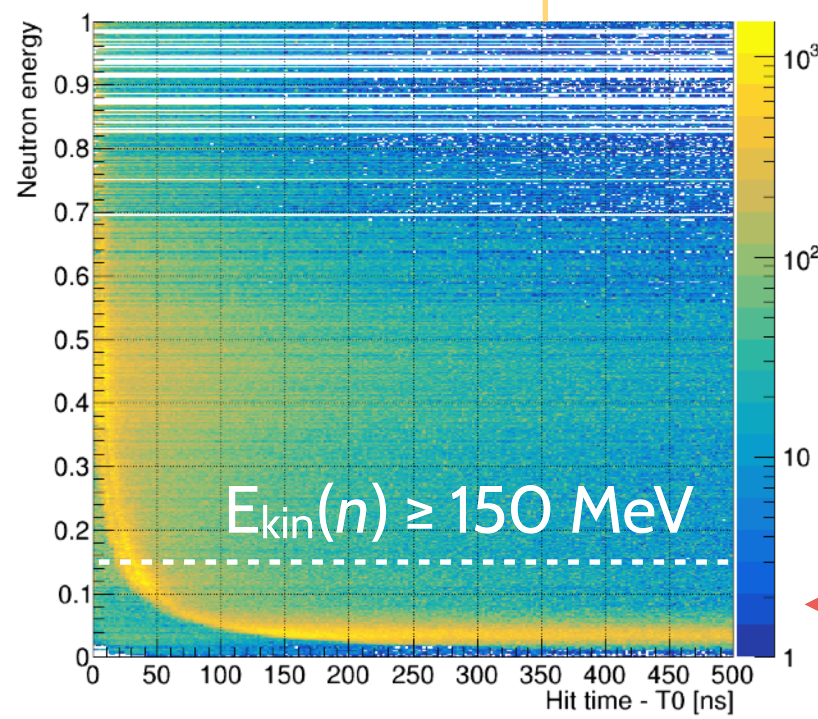
selection	*# particles*	*CPU time, h*
All BIB particles	380M	3040
+ t < 25ns	98M (26%)	480 (16%)
+ $E_{kin}(n)$ > 150 MeV	78M (20%)	200 (6.6%)
+ QGSP_BERT	78M (20%)	24 (0.8%)

→ sufficiently precise for the remaining neutrons

Hit time wrt TOF of a photon from the Interaction Point (IP)

t - T0 > 10ns → t < 25ns

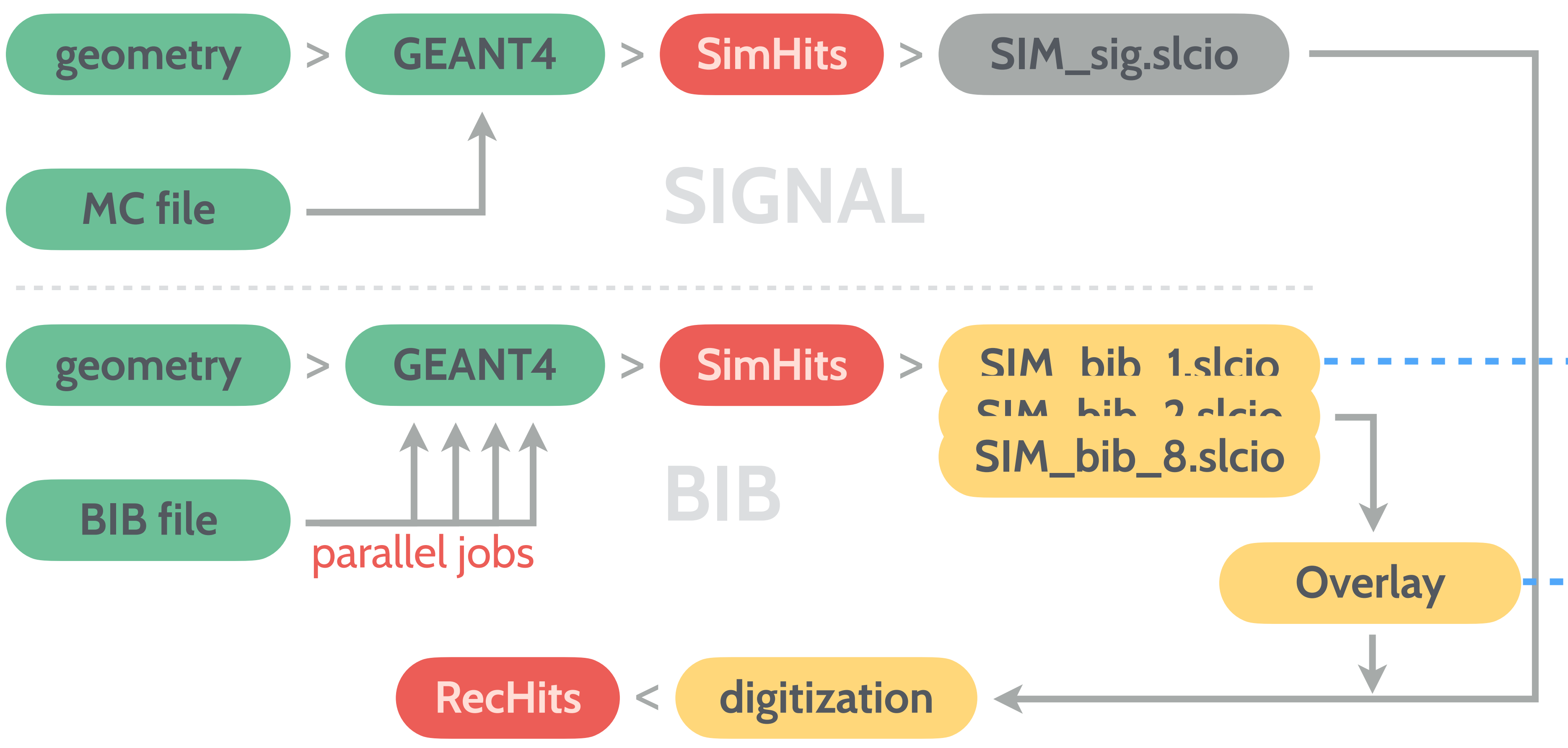




Slow neutrons with E < 150 MeV bounce around for too long before creating a hit

BIB overlay: Marlin

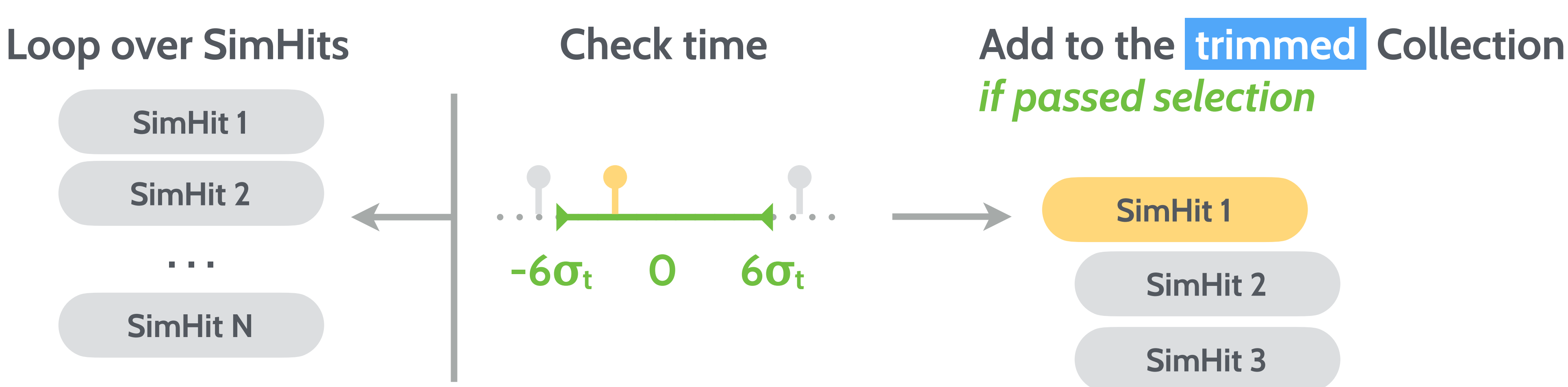
CLIC scheme for obtaining 1 digitised event using ddsim and Marlin



Very heavy part: ~9 GB
- MCParticles *heaviest
- SimCalorimeterHits
- SimTrackerHits

CPU/RAM intensive
- merging of all the MCParticles *RAM
- filtering of SimHits by their time *CPU

Using trimmed BIB samples at Muon Collider: no MCParticles + trimmed SimHits

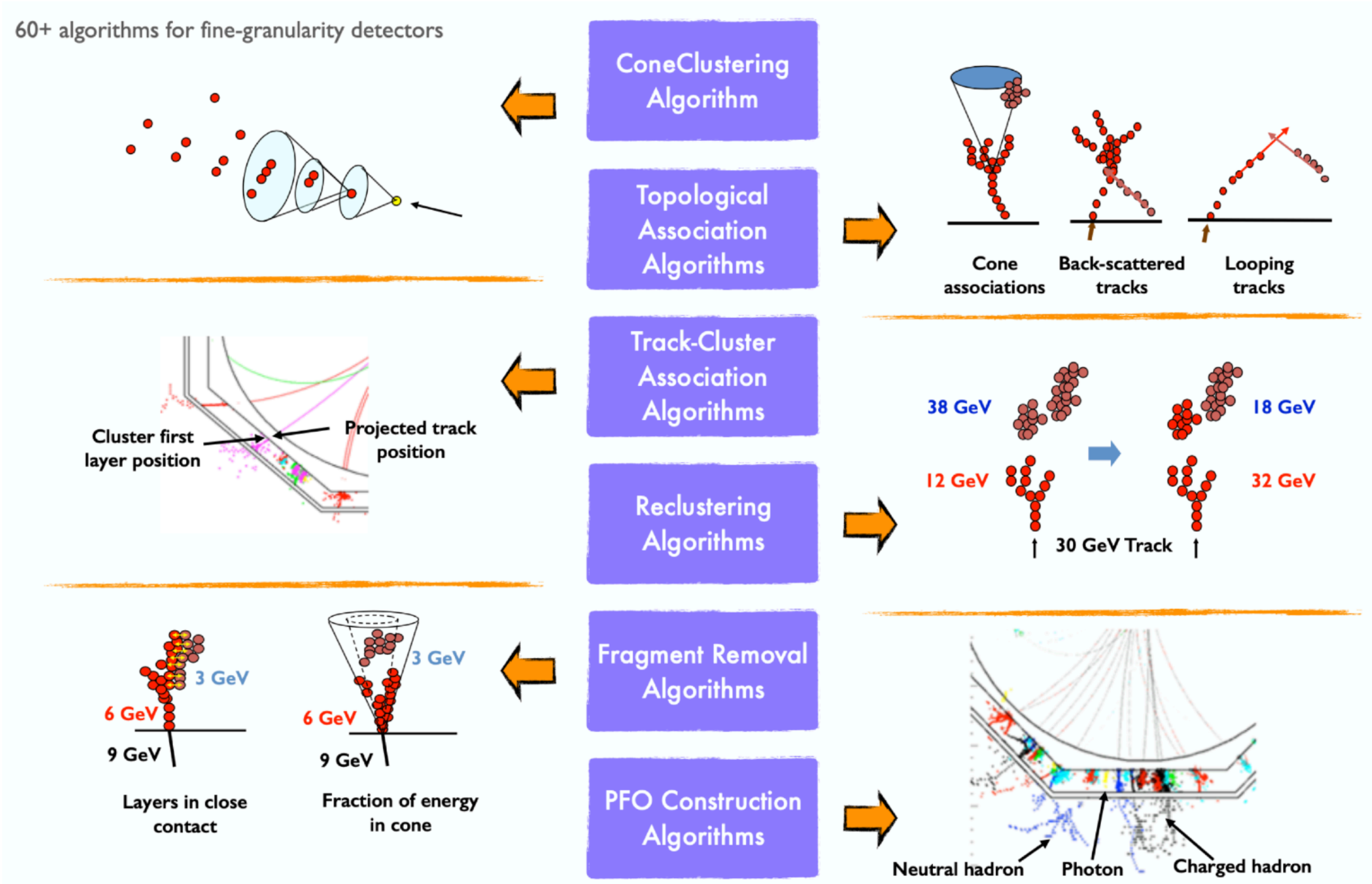


Much more efficient
- **HDD:** 9 GB → 0.4 GB
- **RAM:** 46 GB → 4 GB

Event reconstruction

Technically we can use PandoraPFA for Particle Flow reconstruction, like CLIC

- relies on two main building blocks: **tracks** + high-granularity **calorimeter data**

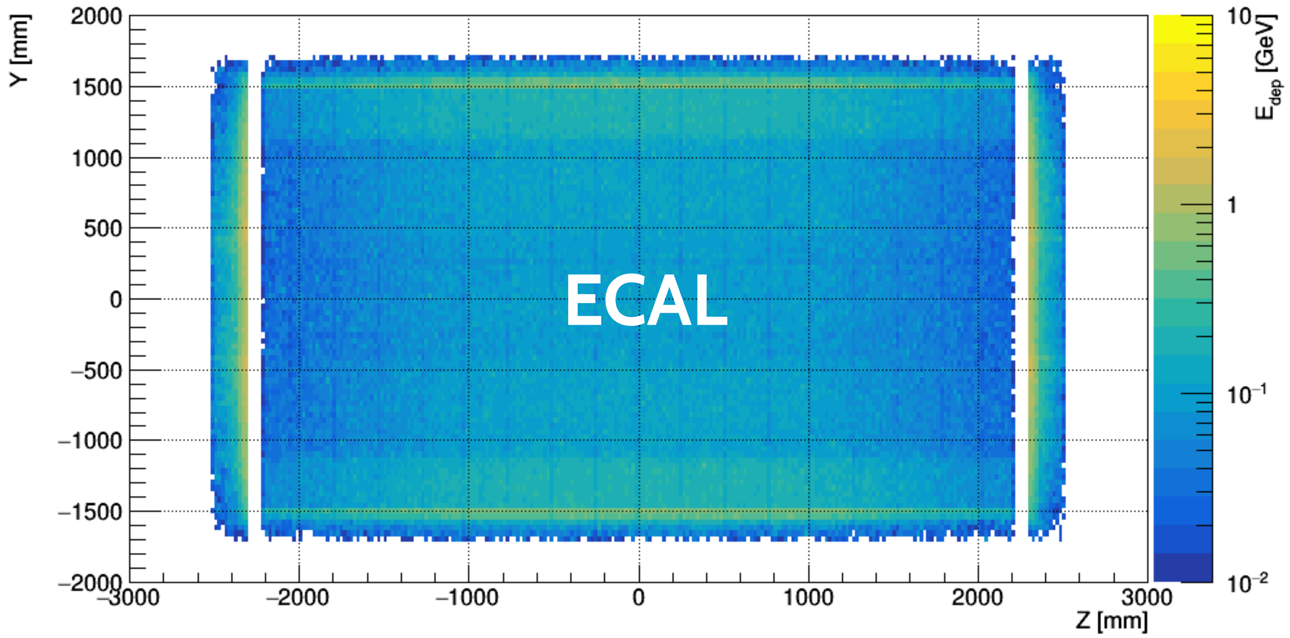


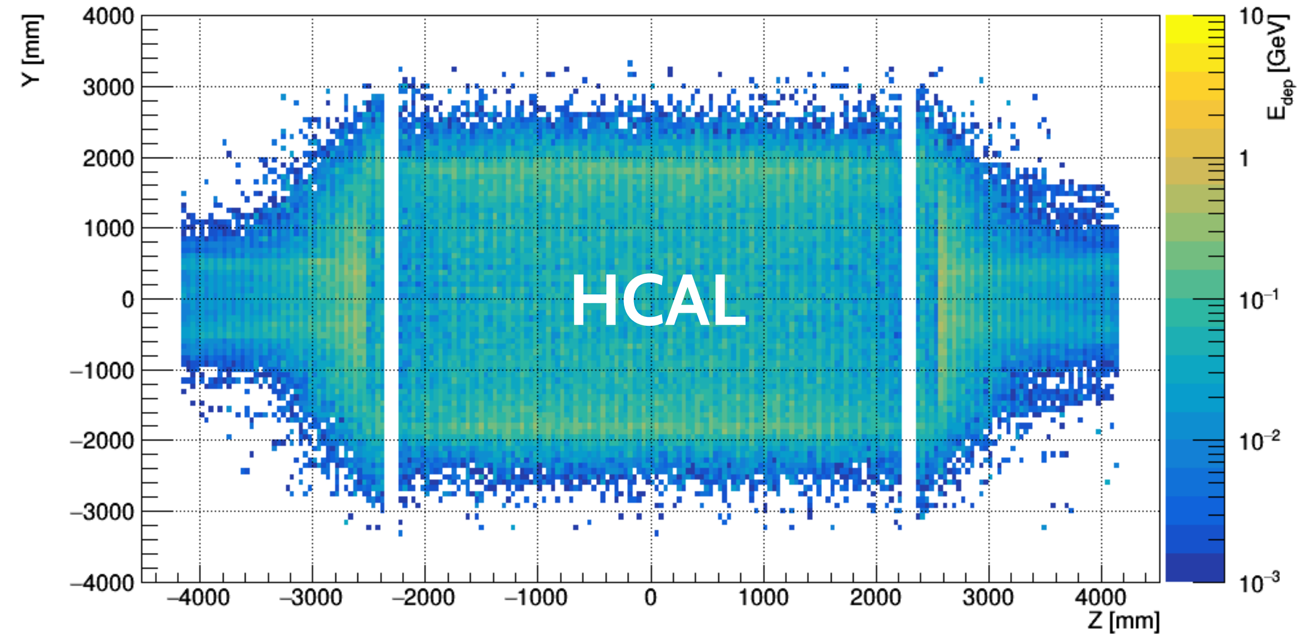
**need cleaning from BIB*

No issues with muon reconstruction: BIB contribution fairly low

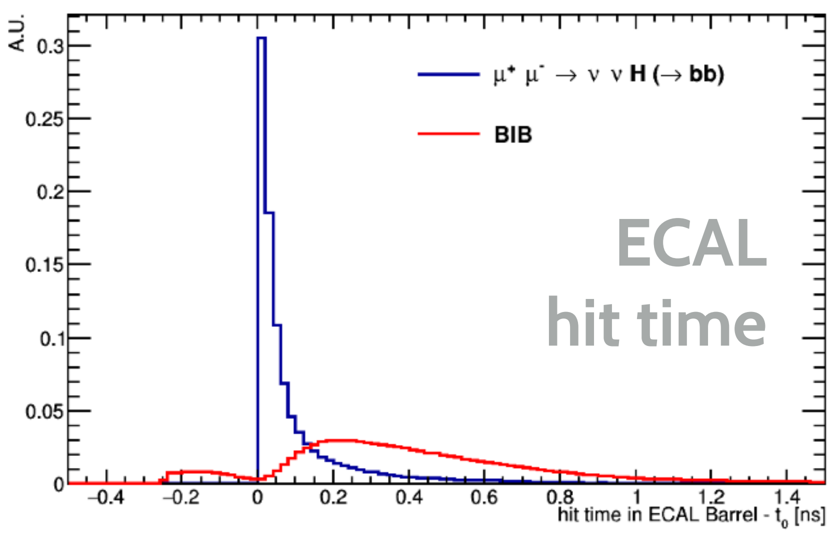
Calorimeter data

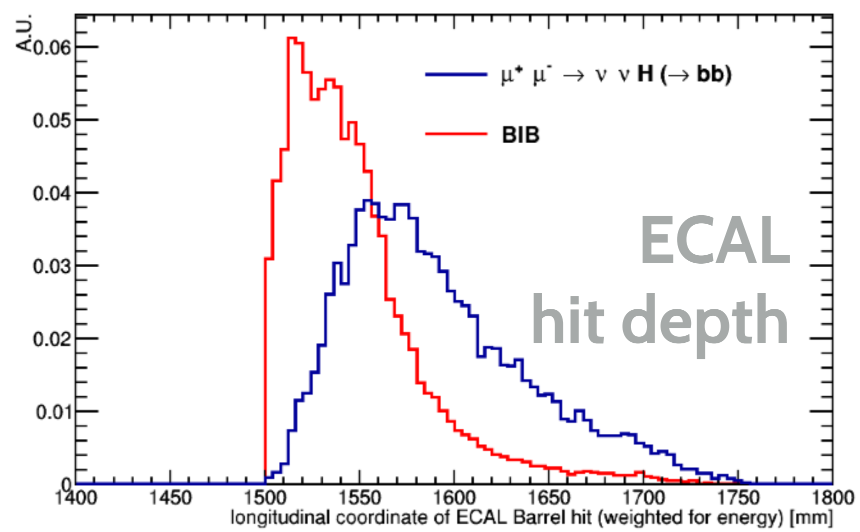
Keeping only hits in the short readout time window: 0 → 10ns





BIB showers have characteristic time and depth profiles

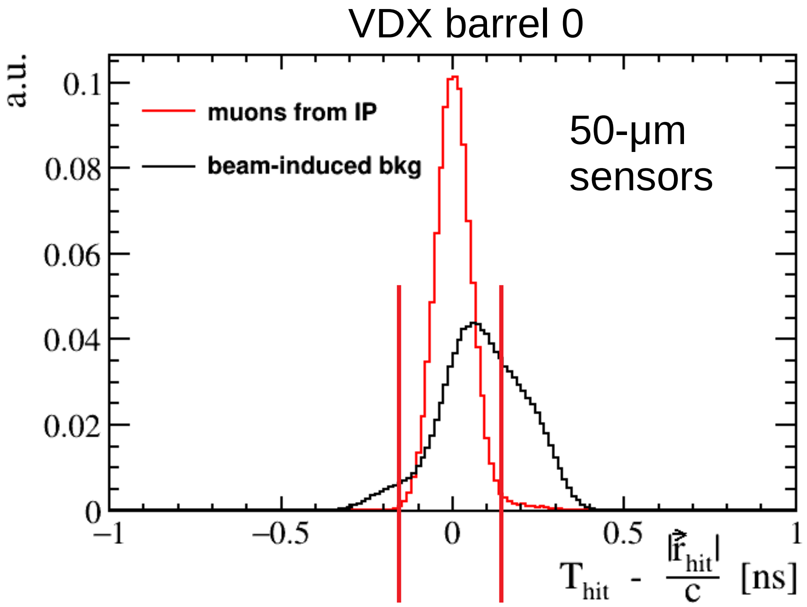




BIB subtraction based on polar-angle, depth and time is under study

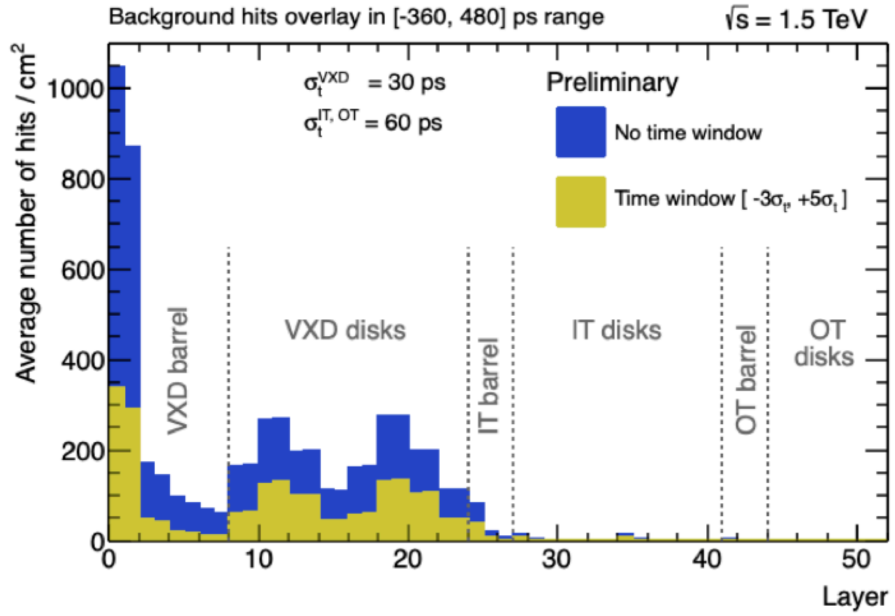
Tracker data

Timing provides a crucial handle on the tracker occupancy



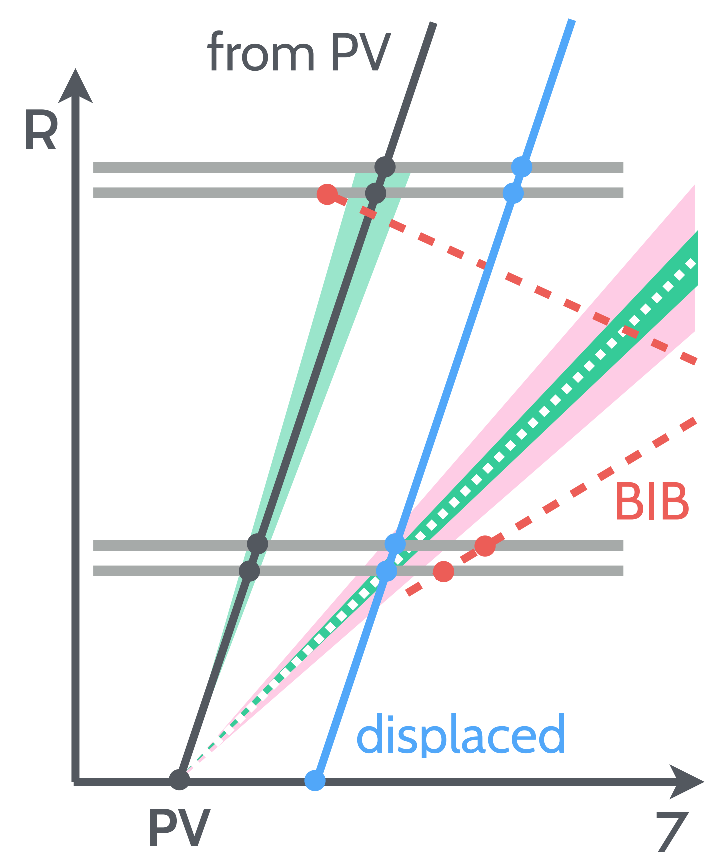
more than x2 reduction in the Vertex Detector

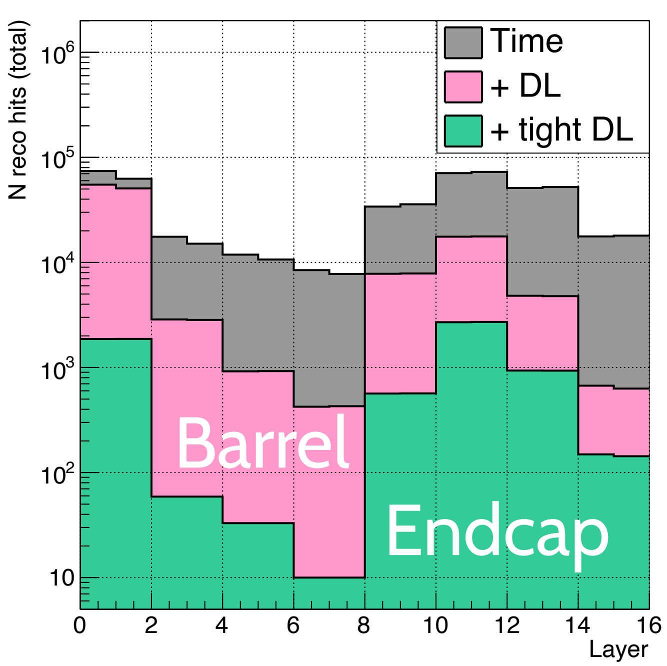
time filtering applied in the digitiser



BIB tracks are low-momentum + originate from the nozzles

- selecting doublets aligned with the Vertex
- implemented in a separate filter after digitisation





Loose doublet selection required to be consistent with the finite beamspot size: σ_z ~10 mm

Tight doublet selection reduces track reconstruction time in 1 event from 2 days → 3 minutes

Currently using Conformal Tracking algorithm: flexible and geometry agnostic

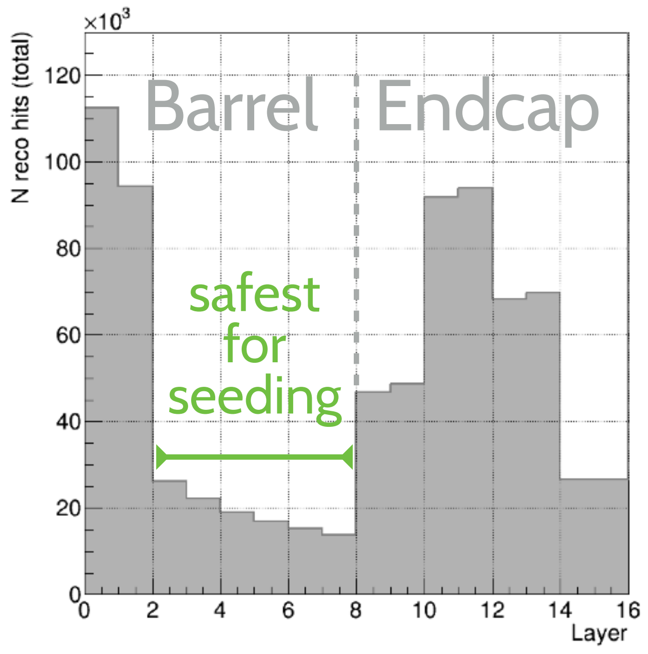
Simple sequence used by CLIC:

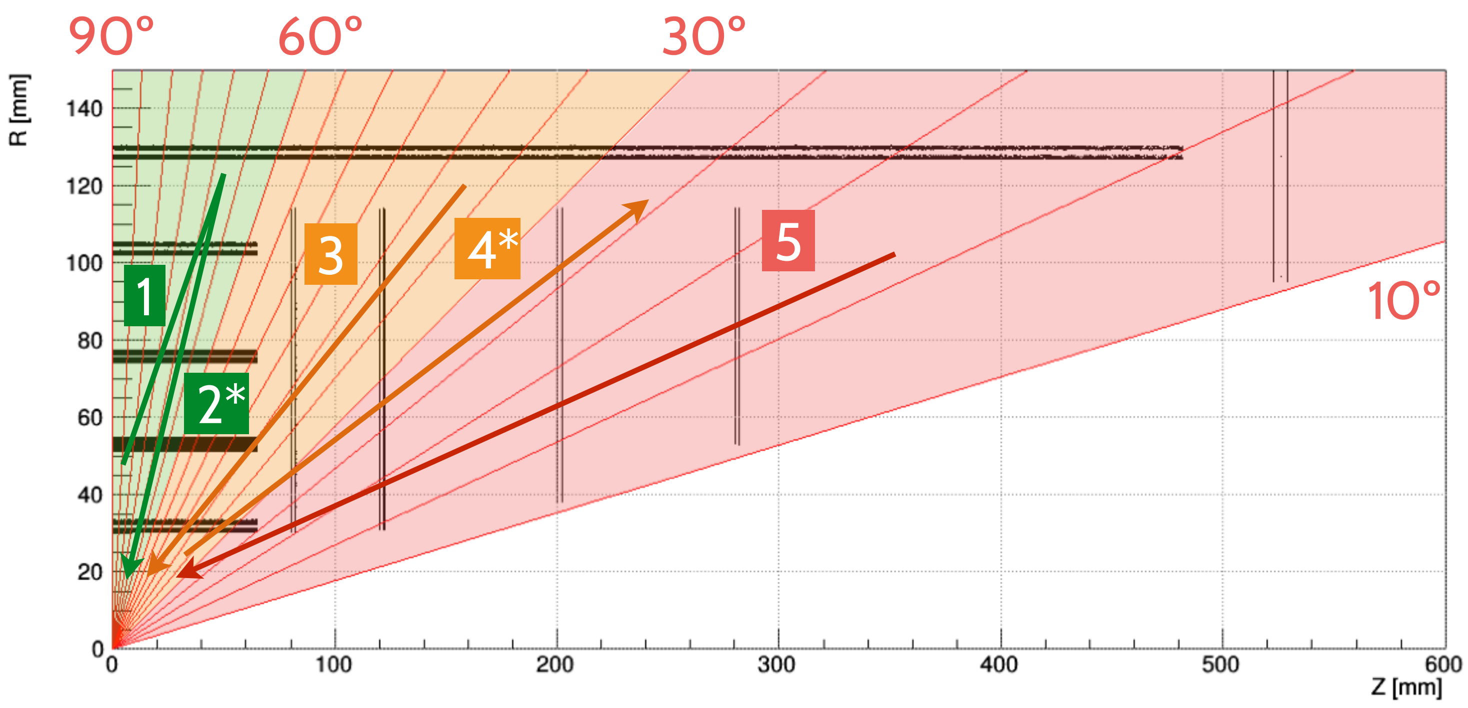
1. build tracks **in the Vertex Detector** ×3 iterations with relaxing search criteria
2. extend tracks **into the Inner/Outer Tracker** projecting the existing trajectories

We can't afford this with BIB at Muon Collider: huge combinatorics in step 1

We split tracker into three regions with individually optimised search criteria

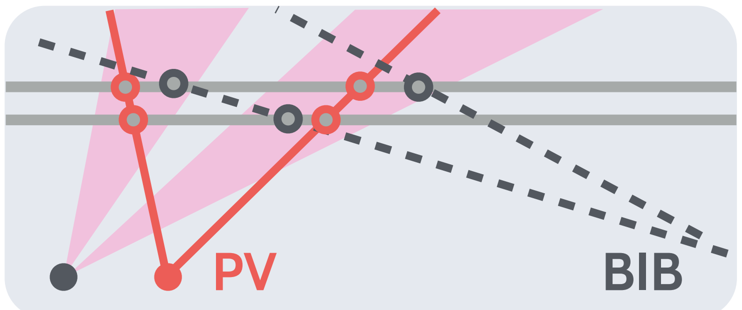
- no seeding from the innermost Barrel layers; only used for track extensions*
- no seeding from layers that are too far away from IP for creating a sufficiently long chain of hits





Track reconstruction: beyond

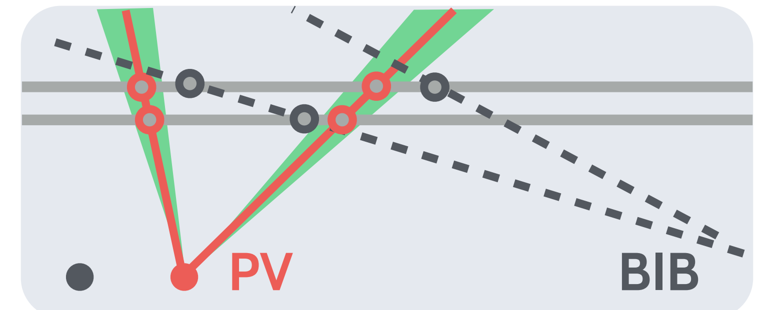
We've started investigating the possibility of fast pattern recognition for estimating the Primary/Secondary Vertex position



→

Determine IP position
with faster track reconstruction
- only central region
- inward seed from ROI

→



Ongoing effort on integrating ACTS in the ILCSoft framework

- general-purpose high-performance tracking package
- might perform faster than Conformal Tracking

We'll need to keep hits at t ≤10ns for reconstructing low-β or long-lived particles

- hit timing will have to be included in track-search logics
- at σ_t=60ps outer tracker layers can be used as TOF detectors

Single-hit BIB suppression might be possible based on cluster shapes

- realistic sensor granularity treatment is required to test this

Summary

Muon Collider studies are based on the iLCSoft framework

- planned transition to Key4HEP in the future

Beam Induced Background poses a serious computation challenge for detector simulations at a Muon Collider

Every step of the simulation process has to use only data relevant for the process, otherwise computation load explodes

Track reconstruction is by far the most CPU-intensive and time consuming part of the event reconstruction

- smart solutions for speeding it up are extremely valuable

The linear approach of reconstructing tracks → vertices → PFO might be not the optimal solution at a Muon Collider

Willing to contribute? Join in!

MUONCOLLIDER-DETECTOR-PHYSICS@cern.ch