December 16th, 2022

Muon Collider
Physics and Detector workshop

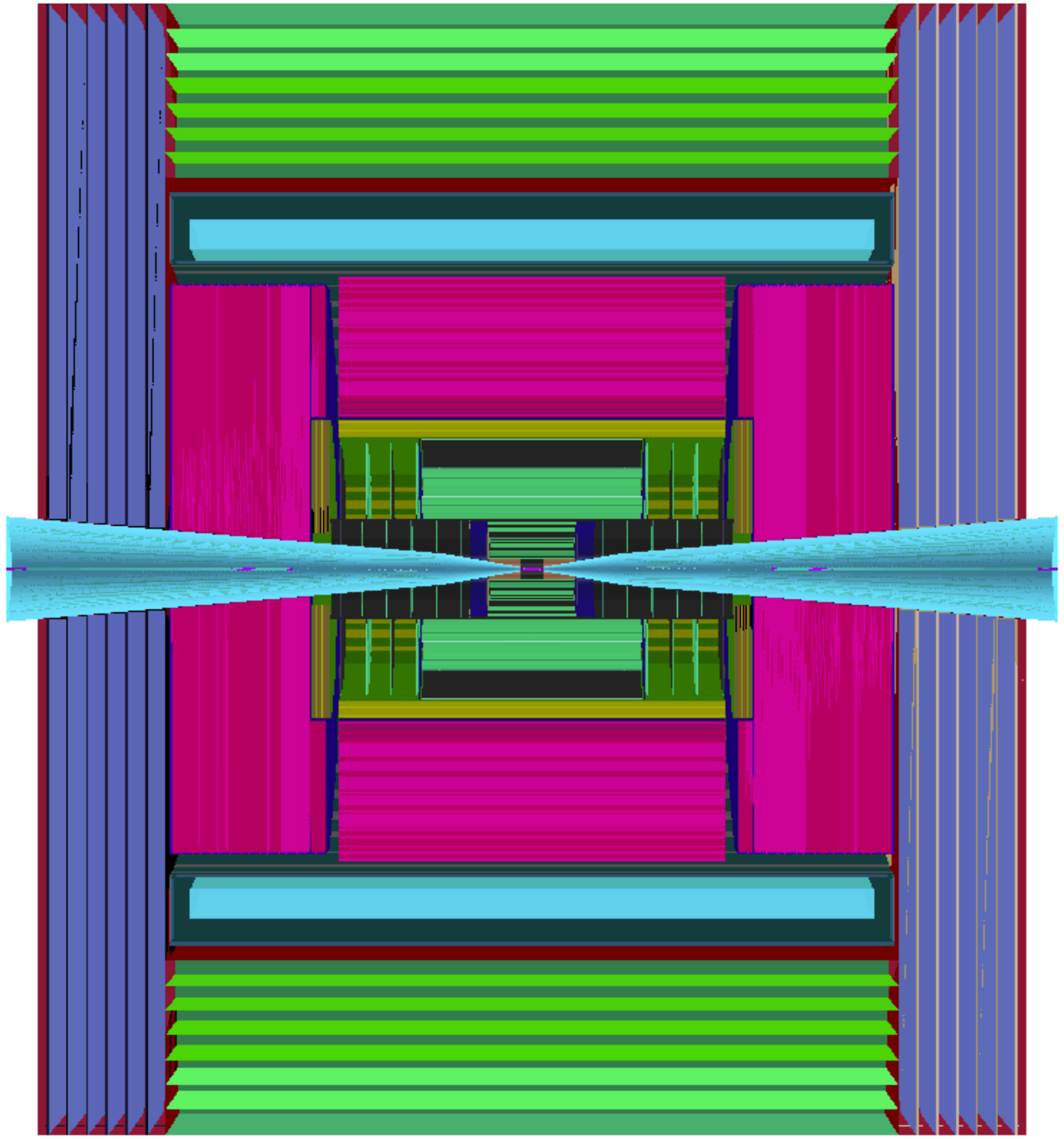
Overview of the full-simulation framework for Muon Collider

N. Bartosik [a]

on behalf of the
Muon Collider Detector and Physics Group

[a] INFN Torino *(Italy)*

Muon Collider detector: current baseline



Muon Collider detector follows the typical layout of general-purpose collider experiments:

- **low-material-budget tracking detector (TRK)**
  ↳ Vertex Detector (VXD) + Inner Tracker + Outer Tracker
- **electromagnetic calorimeter (ECAL)**
- **hadronic calorimeter (HCAL)**
- **superconducting solenoid**
- **muon spectrometer**
- **large tungsten nozzles (MDI)** → machine-detector interface
  ↳ essential for absorbing beam-induced background (BIB) induced by muon decays inside the beam

not so typical

Present model largely based on the CLIC design (e^+e^- at $\sqrt{s} \leq 3$ TeV) **works decently for μ+μ- at $\sqrt{s}$ = 1.5 TeV**

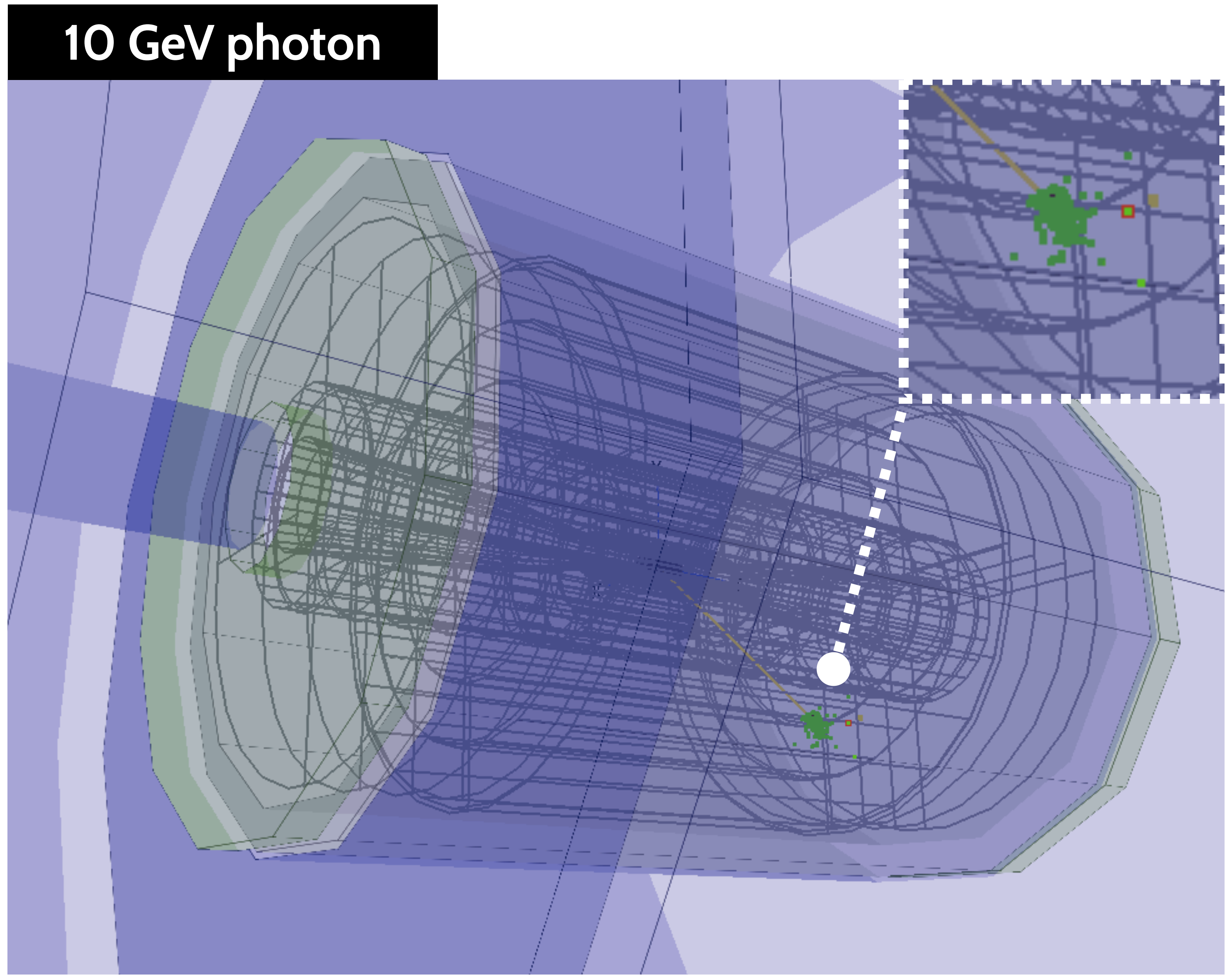
Need to redesign the detector from ground up for $\mu^+\mu^-$ collisions at $\sqrt{s}$ = 3, 10 or more TeV

↳ full simulation is essential for accurate evaluation of the detector performance

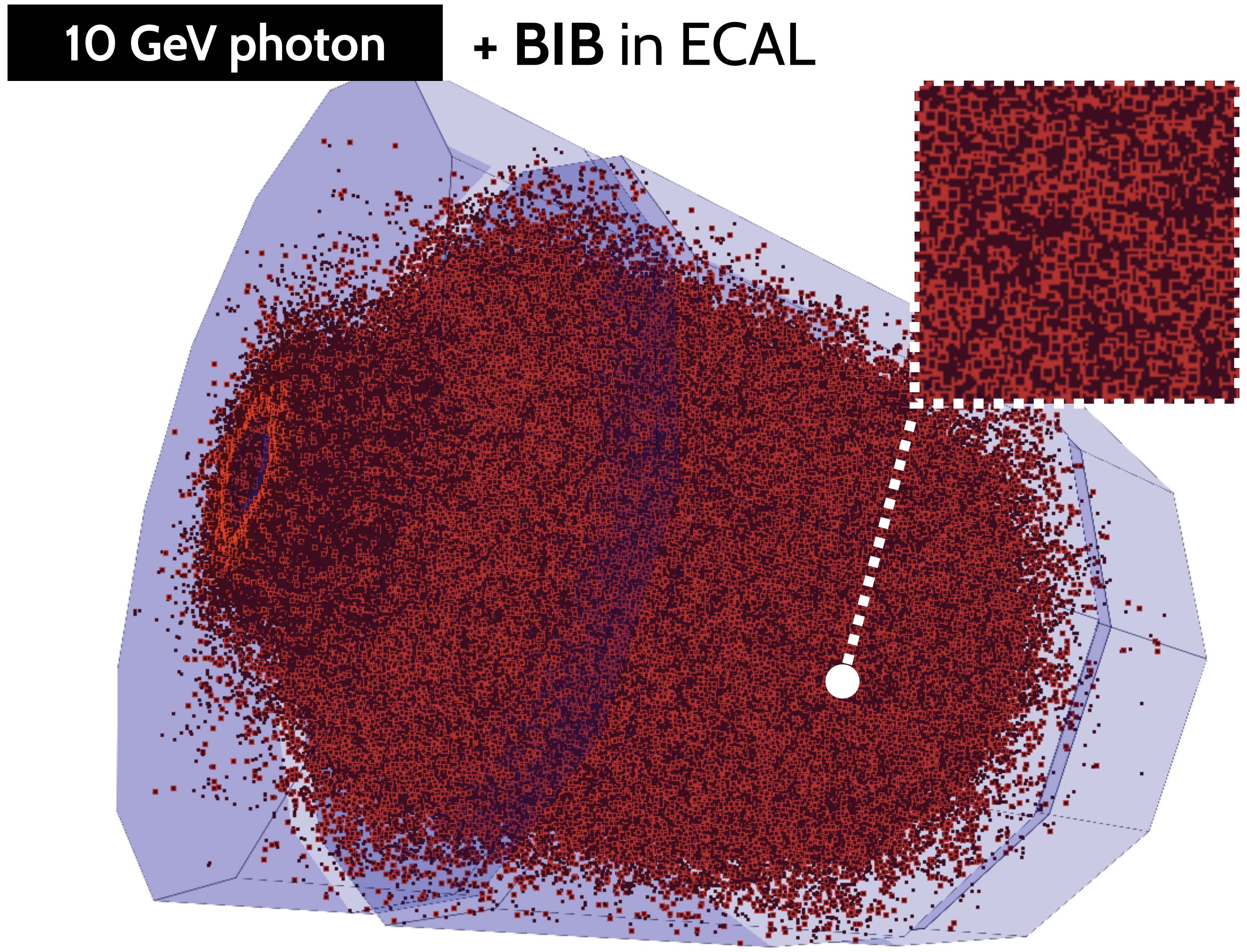
Beam Induced Background

We want our simulation studies to be representative of what it will look like in the actual experiment

↳ all BIB effects have to be included in the most realistic way possible



physics analysis will be done on this kind of events



BIB simulation is done in two separate stages:

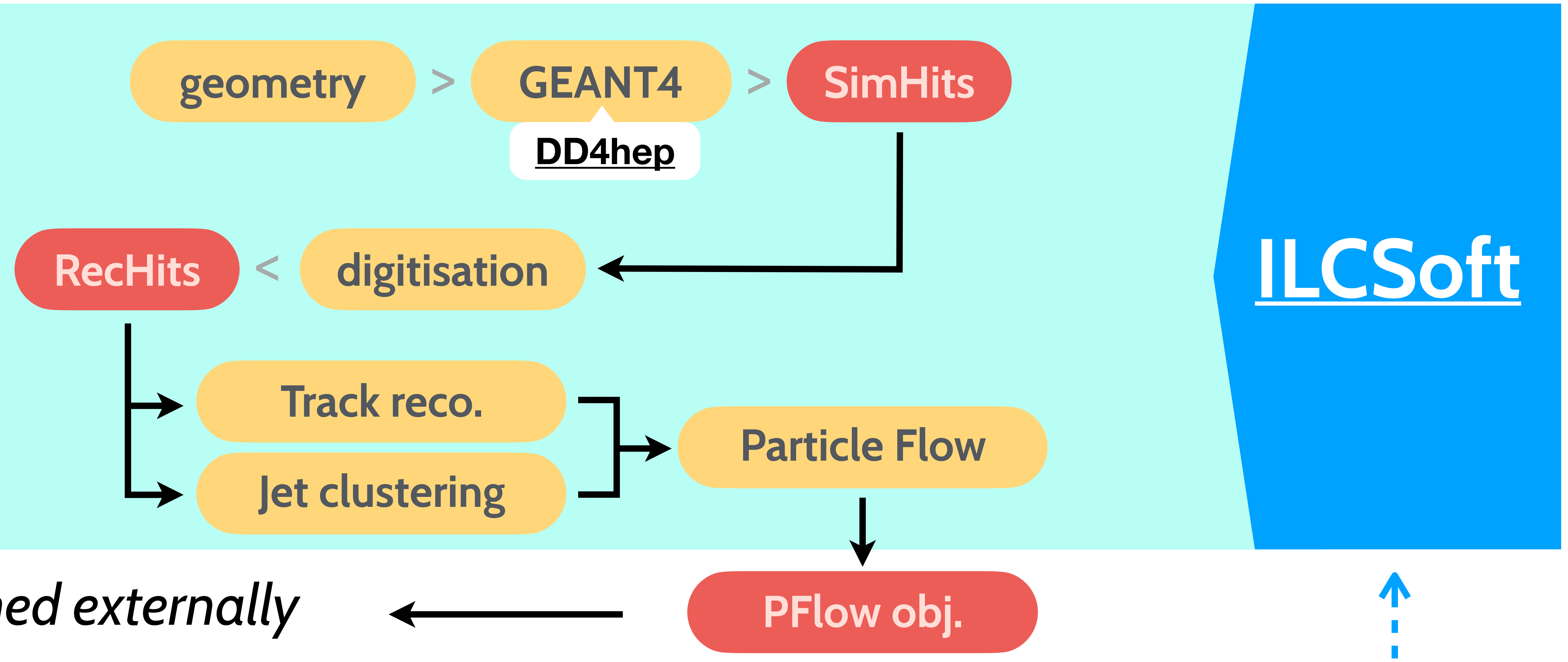
1. **Muons in the accelerator → FLUKA → BIB particles at the MDI surface**
2. **BIB particles in the detector → GEANT4 → detector signals for event reconstruction**

Simulation process

Main steps of a full-simulation study:

Matrix Element + Parton Shower — Beam Induced Background

1. **generation of stable input particles:** ME + PS **Pythia/Madgraph/Whizard** + BIB **FLUKA + FlukaLineBuilder**
2. **simulation of the detector response to the incoming particles**
3. **simulation of detector effects**
   efficiency, electronics noise + thresholds, ...
4. **reconstruction of higher-level objects**
   photons, tracks, jets, particle identification
5. **higher-level analysis** ⟵ *can be performed externally* ⟵



Detector simulation and event reconstruction handled within a single [framework](#)
↳ **inherited from the CLIC experiment:** comprehensive and modern workflow designed for e+e- colliders
Large overlap with the [Key4HEP](#) **software stack:** planning full transition in the future

Most of custom packages specific to the Muon Collider maintained in the public [Muon Collider Software](#) repository
[Installation instructions](#) available for CentOS 8 + [Docker image](#) for an easy and OS-independent local setup

best for SW/algorithnm development — best for data analysis with a fixed SW stack

ILCSoft framework

The main components of the ILCSoft framework:

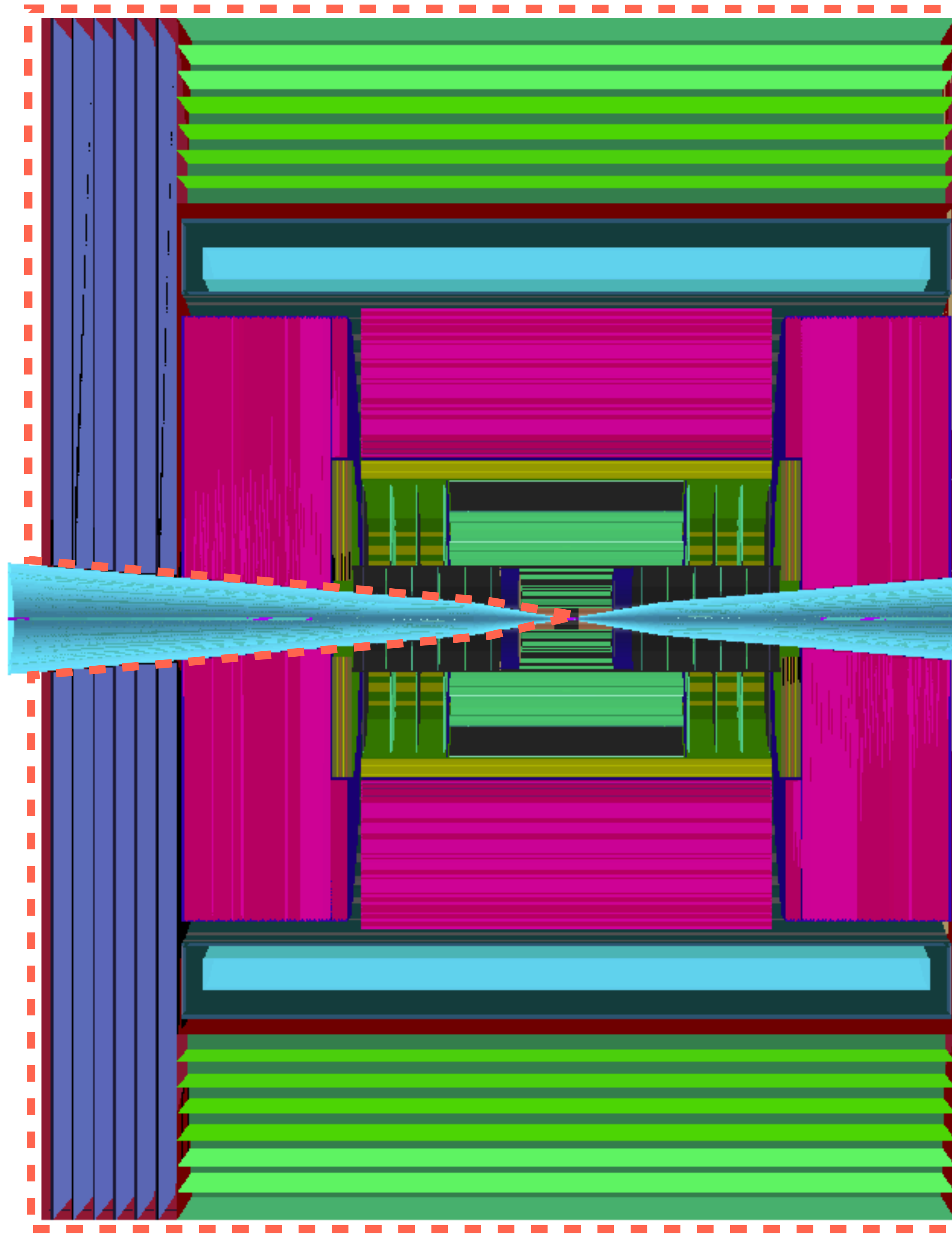
1. LCIO [**L**inear **C**ollider **I/O**]
   Provides consistent storage of event data (**MCParticles, SimHits/RecHits, higher-level and custom objects**) using the ***.slcio** file format

2. Marlin [**M**odular **A**nalysis & **R**econstruction for the **Lin**ear collider]
   Collection of processors for isolated tasks that can be chained into an arbitrarily complex sequence of processes using XML configuration files
   - everything after hits simulated by GEANT4 is handled by processors within the Marlin framework: *digitization, track/jet reconstruction, b-tagging, Particle Flow, ...*

3. DD4hep [**D**etector **D**escription **for H**igh **E**nergy **P**hysics]
   Efficient and flexible detector geometry description with the interface to GEANT4 and Marlin

The most recent tutorial on full simulation for Muon Collider available online

For 0.75 TeV beams at 2×10^{12} μ/bunch → **4×10^5 muon decays/m** in a single beam crossing
↳ muon decays and secondary interactions with the lattice are simulated externally



Result of a beam-decay simulation → **list of stable particles reaching the detector region**

- collected at the outer surface of the detector + MDI ▶
- **2 × 7.3M particles** → text file generated by MAP for $\sqrt{s}$ = 1.5 TeV using MARS15 software
  ↳ represents only a fraction of the full BX statistics [weight = 22.2]

Particle definitions converted to `LCIO::MCParticle` instances → ILCSoft data model

- every 1000 particles × 22.2 copies (randomised in ϕ angle) packed into **1 event**
  ↳ ~22K particles/event
- **2 × 160M particles** → saved to a few K events across 2 LCIO files (μ^+ and μ^- beams)
  ↳ splitting into smaller events reduces memory required for simulating 1 event
  \+ possibility to create fractional BIB samples, e.g. 10% of the full BX

GEANT4 simulation done in ILCSoft using `ddsim` executable

`LCIO::MCParticle` → `LCIO::Sim*Hit`

↳ final output structure depends on the processing strategy:

1 long job		several parallel jobs
2 beams → 1 file	OR	1 beam → many files

Detector simulation process

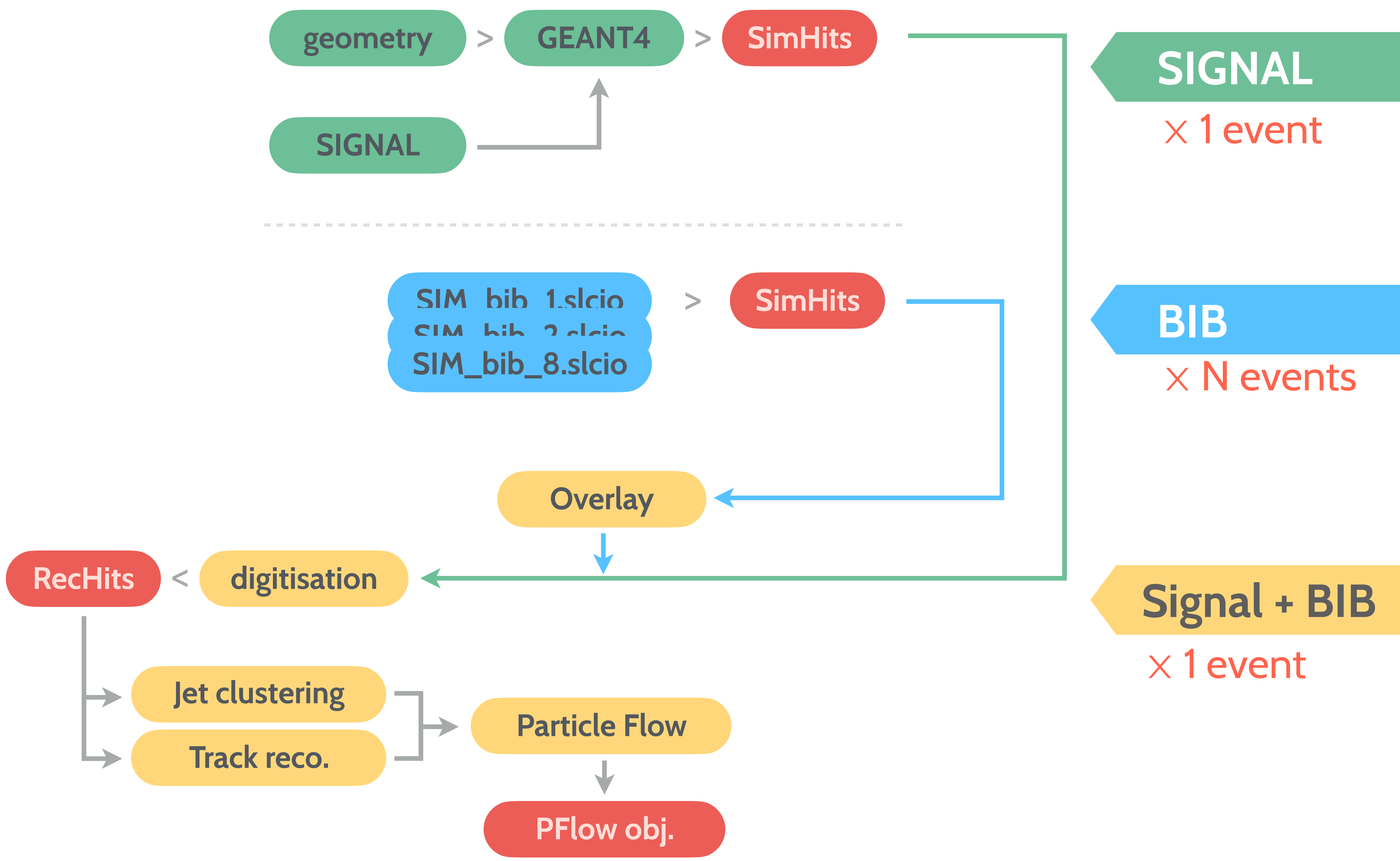
Full simulated event obtained via three distinct stages:

GEANT4 simulation of Signal: straightforward and fast

Overlay of BIB*: performed in each event before digitisation
↳ sensitive to the # of BIB SimHits and merging logics

Reconstruction speed of higher-level objects strongly depends on the amount of input RecHits from BIB

- especially relevant for track reconstruction *(combinatorics)*
- BIB contribution has to be suppressed as early as possible



*** Currently reading the same full simulated BX during the Overlay step**
↳ more flexible mixing of smaller batches of BIB particles will be possible with the new approach based on FLUKA

Every simulation step requires careful treatment of computing resources

DISK STORAGE | DISK I/O | CPU TIME | RAM USAGE | DISTRIBUTION

Properties of the BIB contribution

BIB has several characteristic features → crucial for its effective suppression

1. Predominantly very soft particles (p << 250 MeV) except for neutrons

fairly uniform distribution in the detector → no isolated signal-like deposits

↳ conceptually different from pile-up contributions at the LHC

2. Significant spread in time (few ns + long tails up to a few μs)

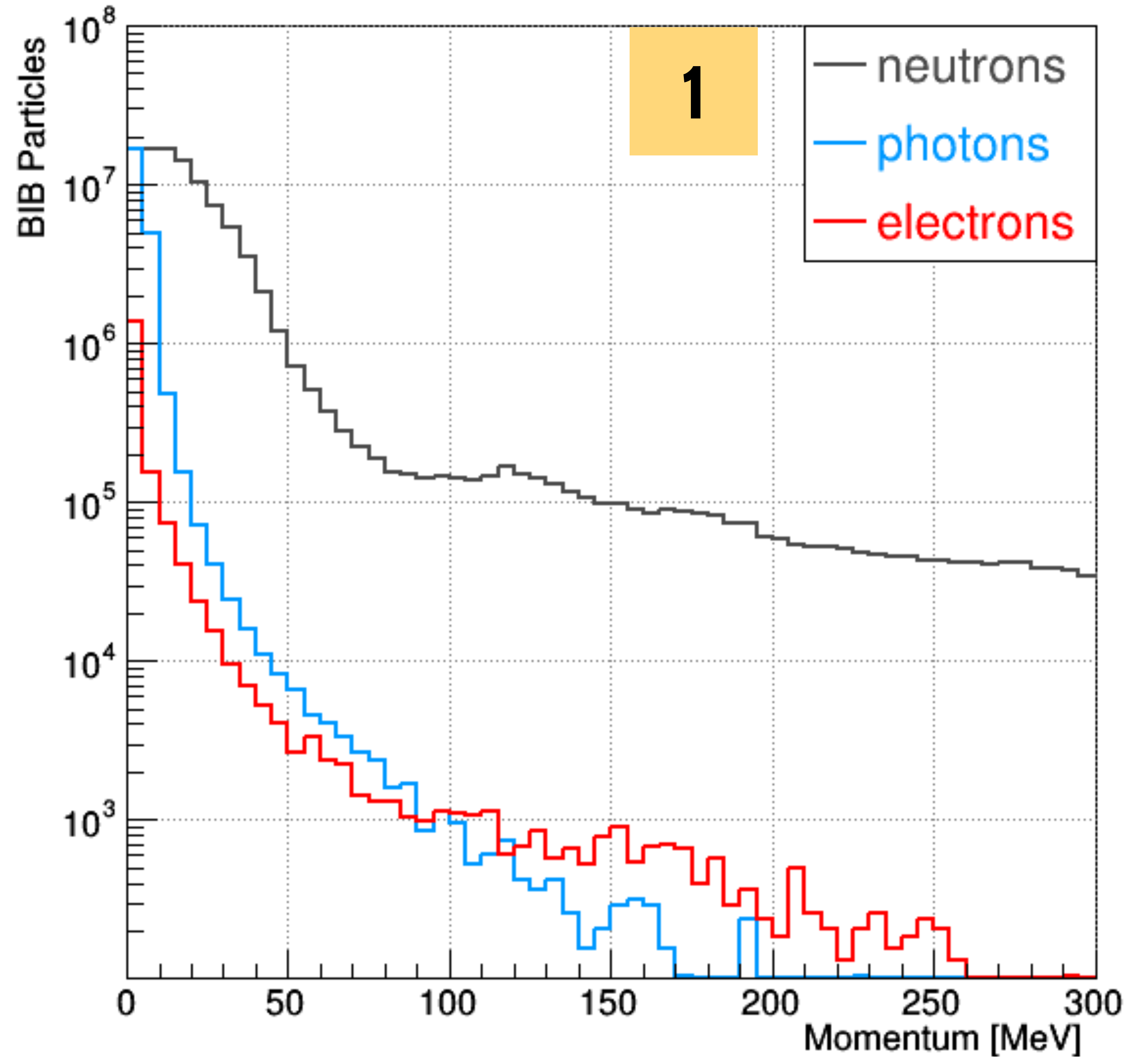
$\mu^+\mu^-$ collision time spread: 30ps (defined by the muon-beam properties)

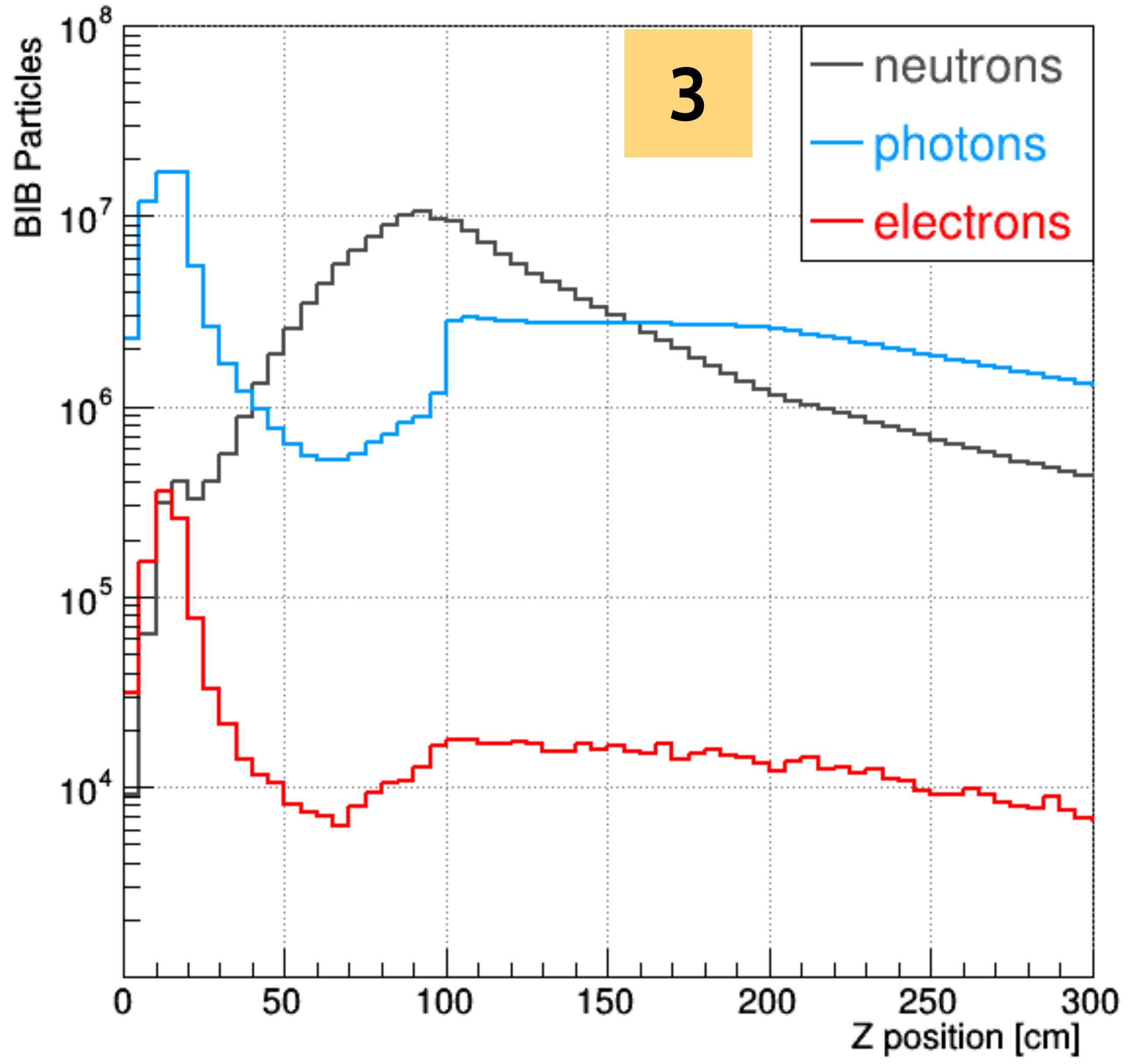
↳ strong handle on the BIB → requires state-of-the-art timing detectors

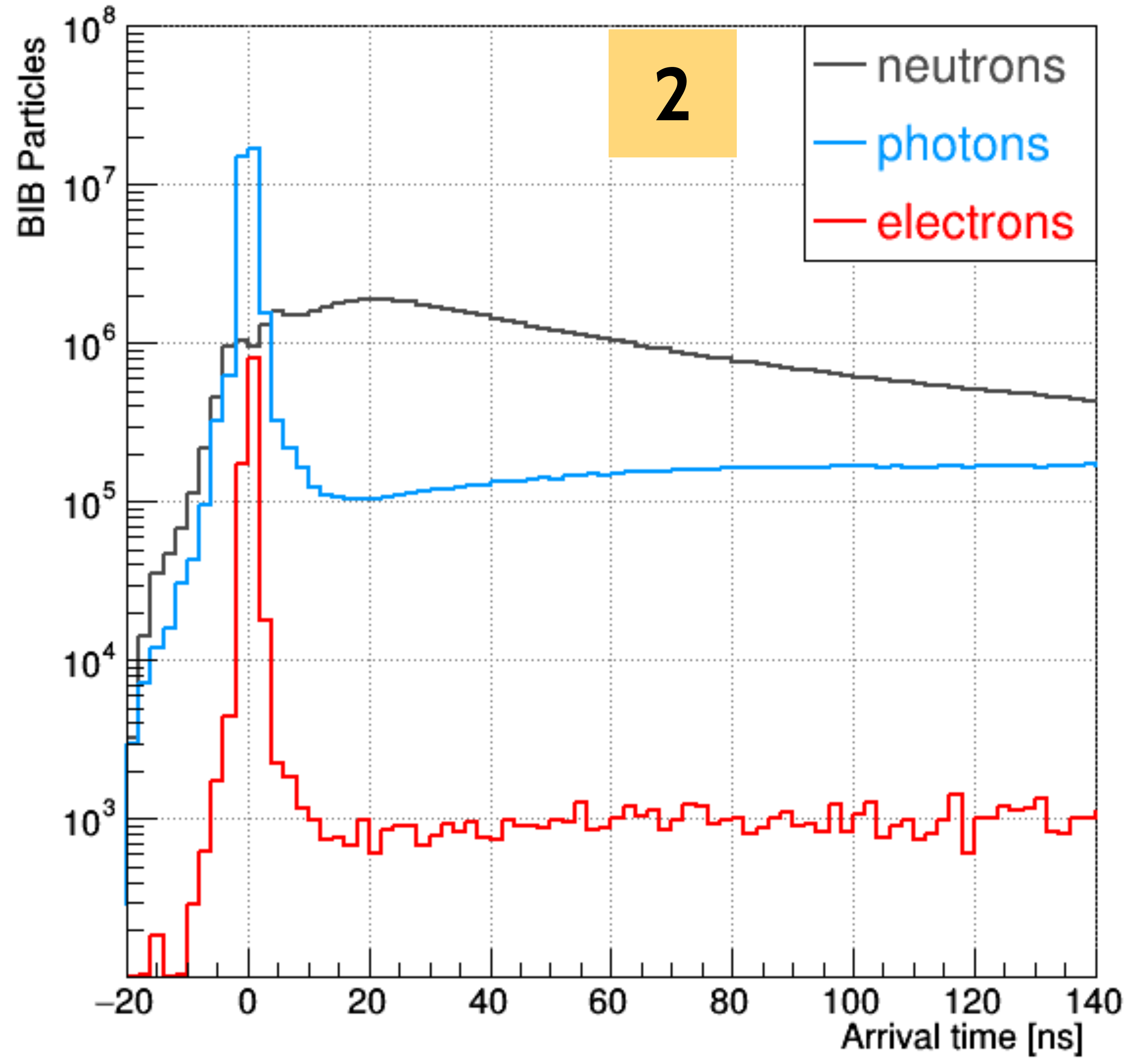
3. Large spread of the origin along the beam

different azimuthal angle wrt the detector surface
+ affecting the time of flight to the detector

↳ relevant for position-sensitive detectors







GEANT4 simulation of BIB

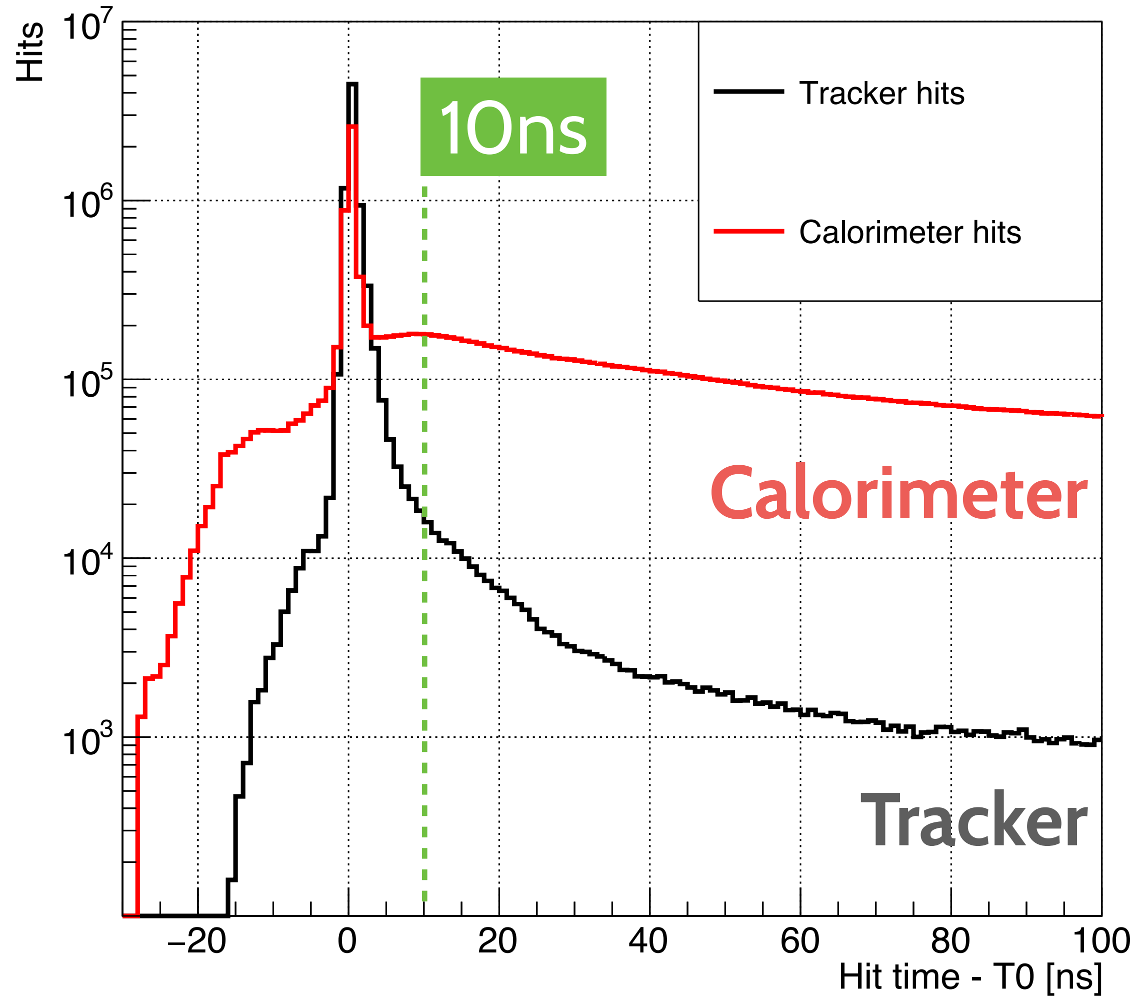
Not all of the ~10^8 BIB particles arriving to the detector are relevant for its performance in a real experiment
↳ **we want to exclude all BIB particles from the simulation chain as early as possible**

1. No GEANT4 simulation of particles arriving too late ×6 less CPU

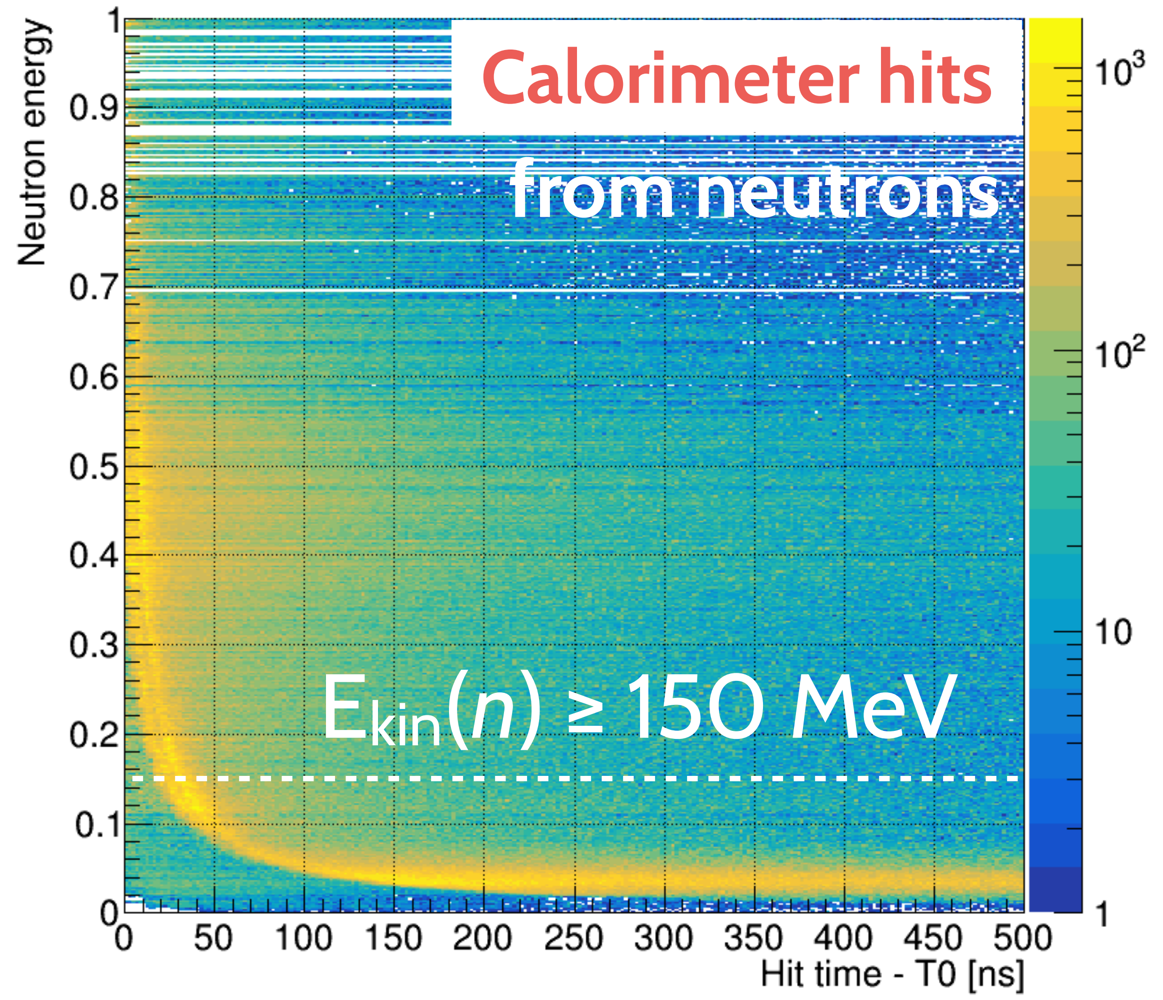
hits at t > 10ns will be outside of the realistic readout time windows
↳ all particles with t > 25ns at the MDI surface are discarded *(accounting for TOF)*

2. No GEANT4 simulation of low-energy neutrons ×20 less CPU

high-precision neutron model required for accurate simulation: `QGSP_BERT_HP`
but they are slow → arrive to the detector with a significant delay
↳ neutrons with E_{kin} < 150 MeV can be safely excluded + faster model: `QGSP_BERT`



1



2

Improved GEANT4 simulation of a single BIB event from 127 days → 1 day
↳ ~10-100 reusable events can be generated in several days

All these cutoffs might still introduce bias in certain edge cases (e.g. long-lived particles)
↳ exact cutoff values must be re-evaluated whenever something changes
- beam energy
- MDI design, etc.

Digitisation logics

GEANT4 hits produced separately for Signal and BIB → merging + detector effects added during digitisation

↳ **two distinct classes of hits: CalorimeterHit** (ECAL, HCAL, Muon detector) + **TrackerHit** (Tracking detector)

1. Calorimeter hits: cell ID + E_{dep} + timestamp

large cells (0.5×0.5 - 3×3 cm) → manageable # of cells

↳ hits merged within a fixed readout time window *(0-10ns)*

2. TrackerHits: sensor ID + 2D position + time *and more*

small pixels (50×50 μm) to macro-pixels (0.05×10 mm)

↳ too many channels to treat them individually in GEANT4

2.1. Simple 3D smearing by σ_U | σ_V | σ_t (30-60ps)

simple and fast

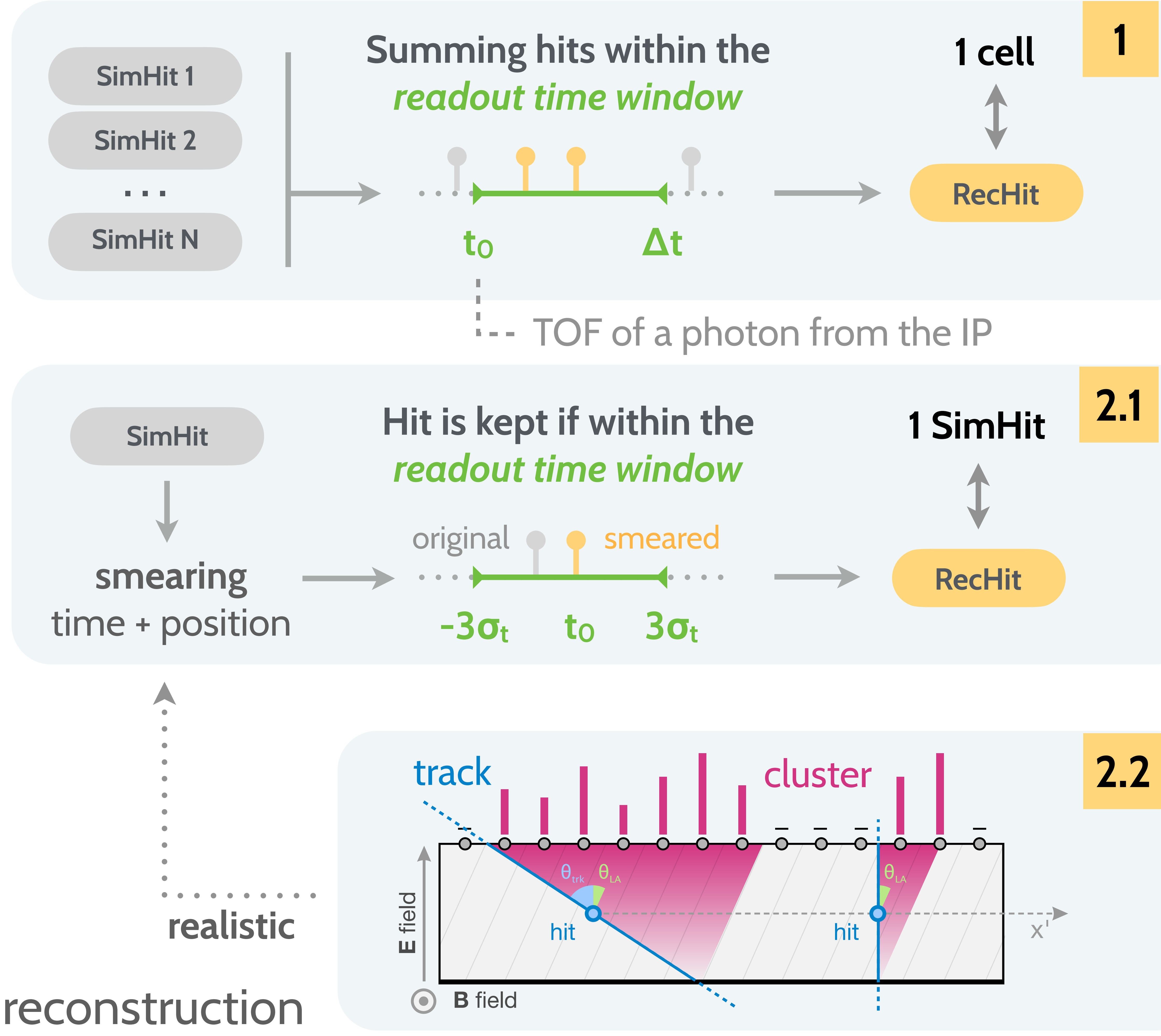
NO charge sharing, pile-up, electronics effects, etc.

2.2. Realistic simulation of sensor + readout-chip response

complex and slow

allows cluster-shape analysis for better BIB suppression

↳ more expensive digitisation → potential savings in track reconstruction



Optimised overlay input

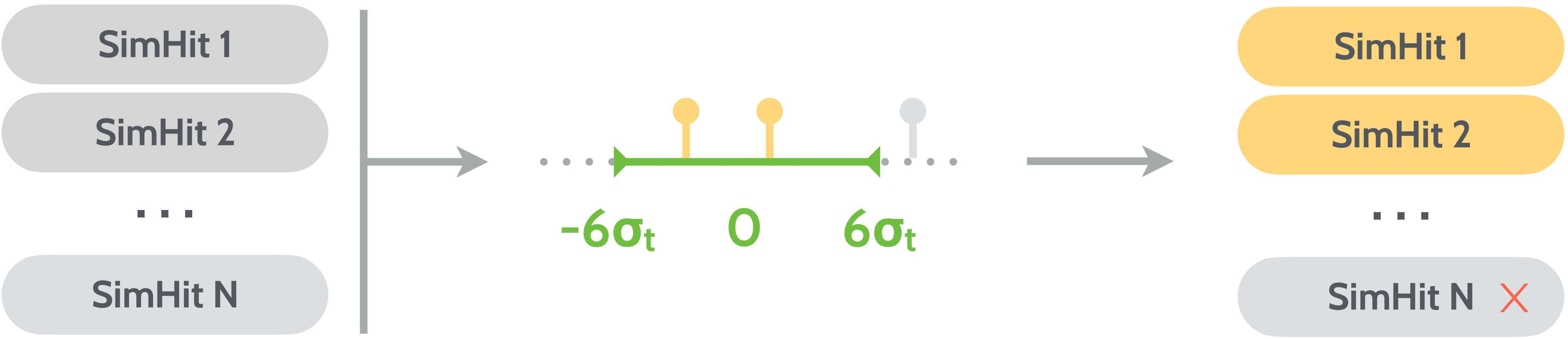
Tracker hits

Individual BIB particles have to be simulated

- **very small pixel size,** down to 50×50 μm
- **cluster-shape** → particle type/angle is important
- **precise timing** → pile-up effects are important

Hits created much later than the readout window will be discarded after digitisation anyway

Extend time windows by extra ±3σ for SimHits
to account for the time smearing during digitization



Calorimeter hits

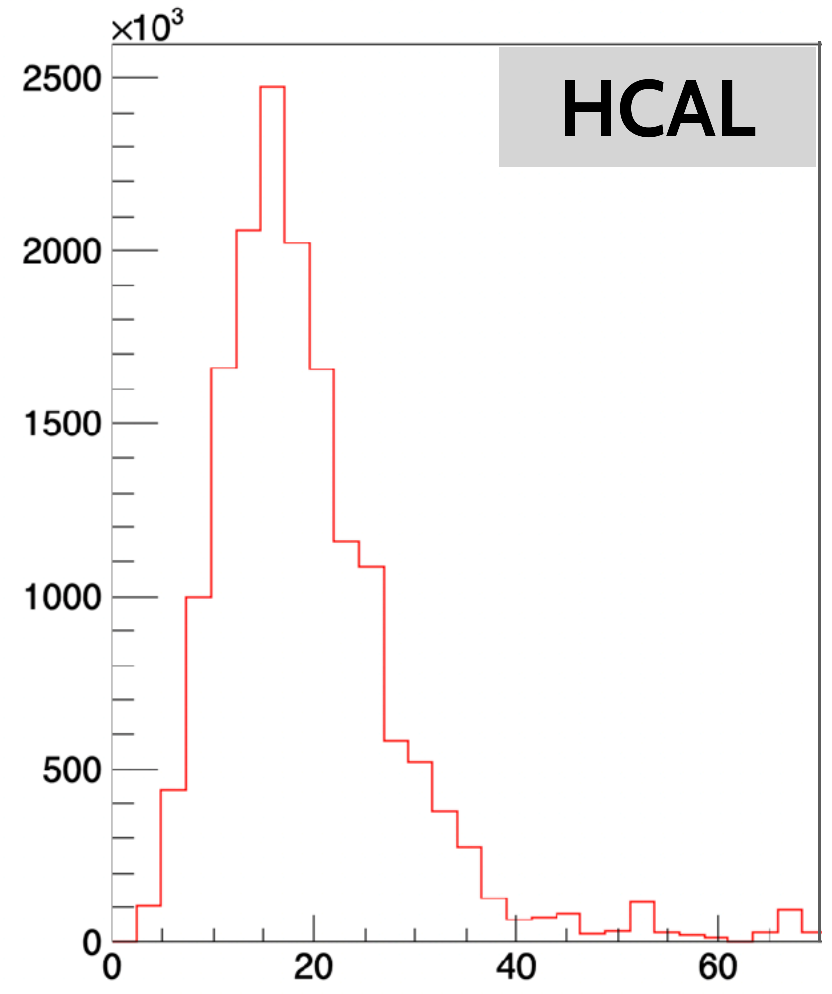
TO BE DONE

Only combined energy deposit in a single cell is relevant

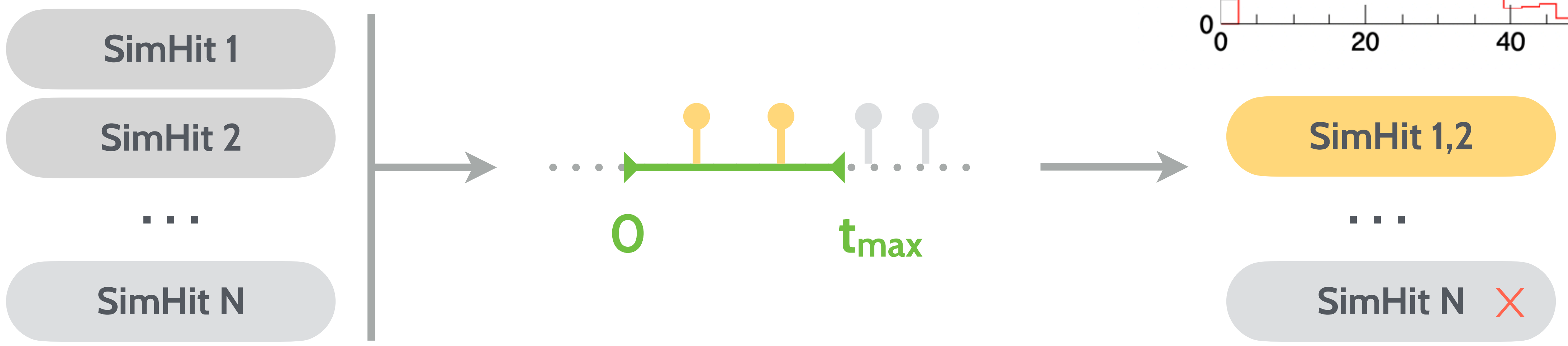
- **large cell size,** at least 5×5 mm
- **less precise timing,** order of 100ps - 1ns

Individual contributions in each cell occupy extra space
on average ×15 more than needed

\# of hits/cell



Merge SimHits within a cell into one
before storing it to disk



Significantly reduced number of individual hits used as input in digitization processors

×10 less DISK/RAM

Tracking optimisation

Reconstruction of tracks suffers from large combinatorial background

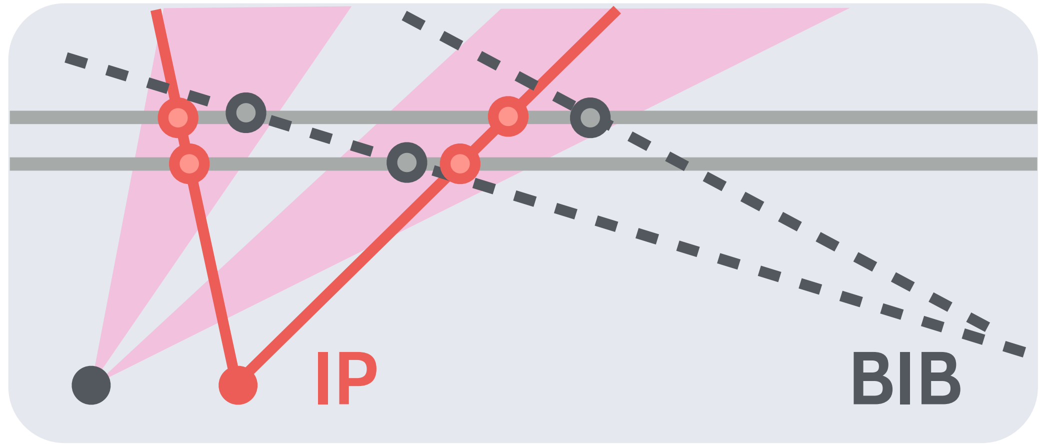
↳ **need suppression of BIB hits + efficient tracking strategies/algorithms**

1. Selection of hits in the narrow time window tailored to the sensor position

↳ limited by the time resolution + beamspot time spread + non-relativistic TOF

2. Selection of hit doublets aligned with the IP (double layers in the Vertex Detector)

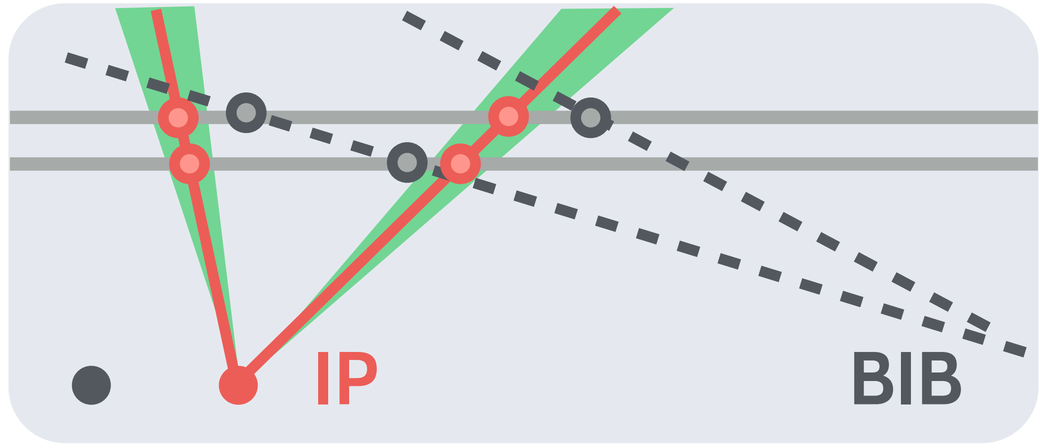
↳ limited by the IP position resolution → requires multi-stage tracking strategy



→

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

→

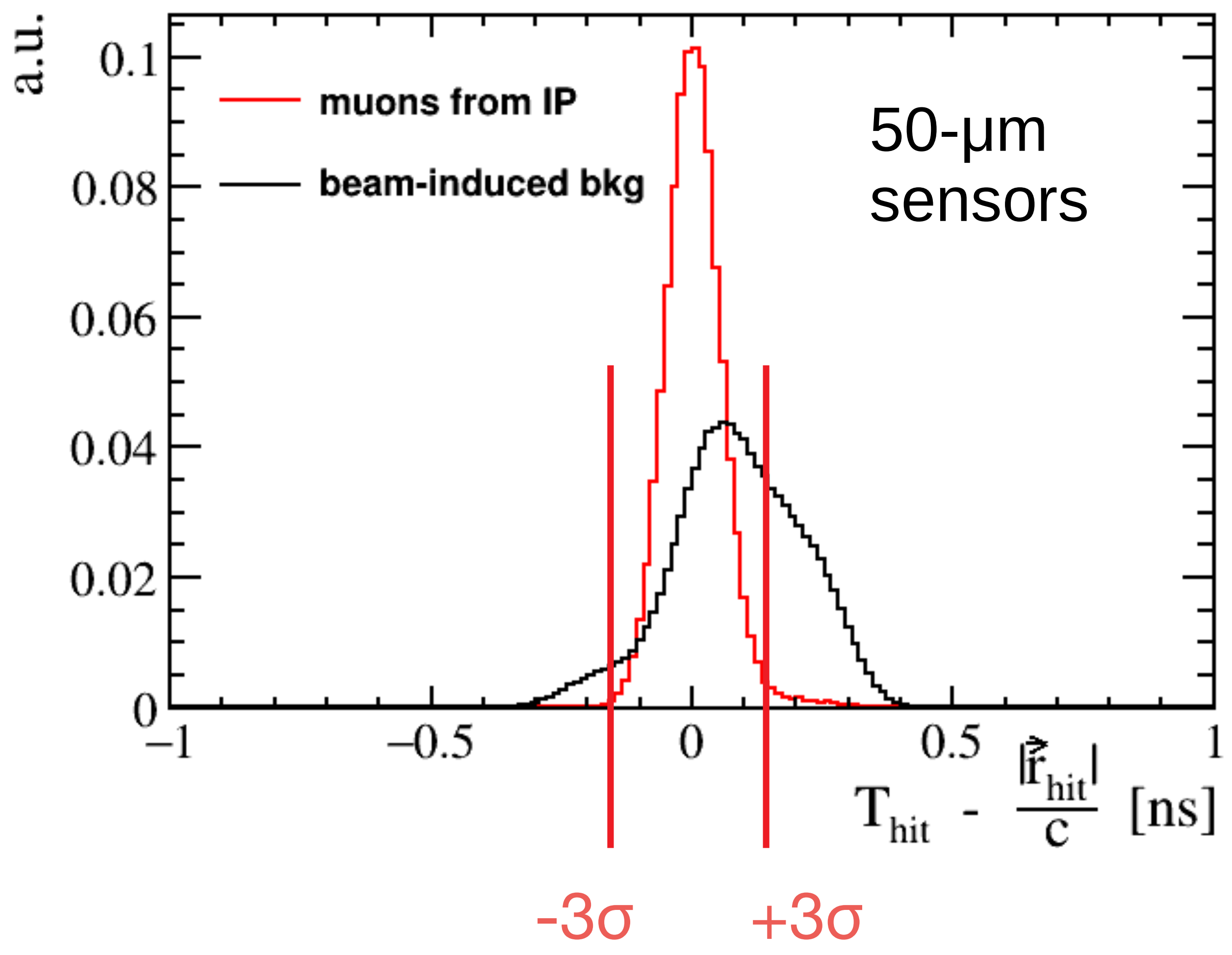


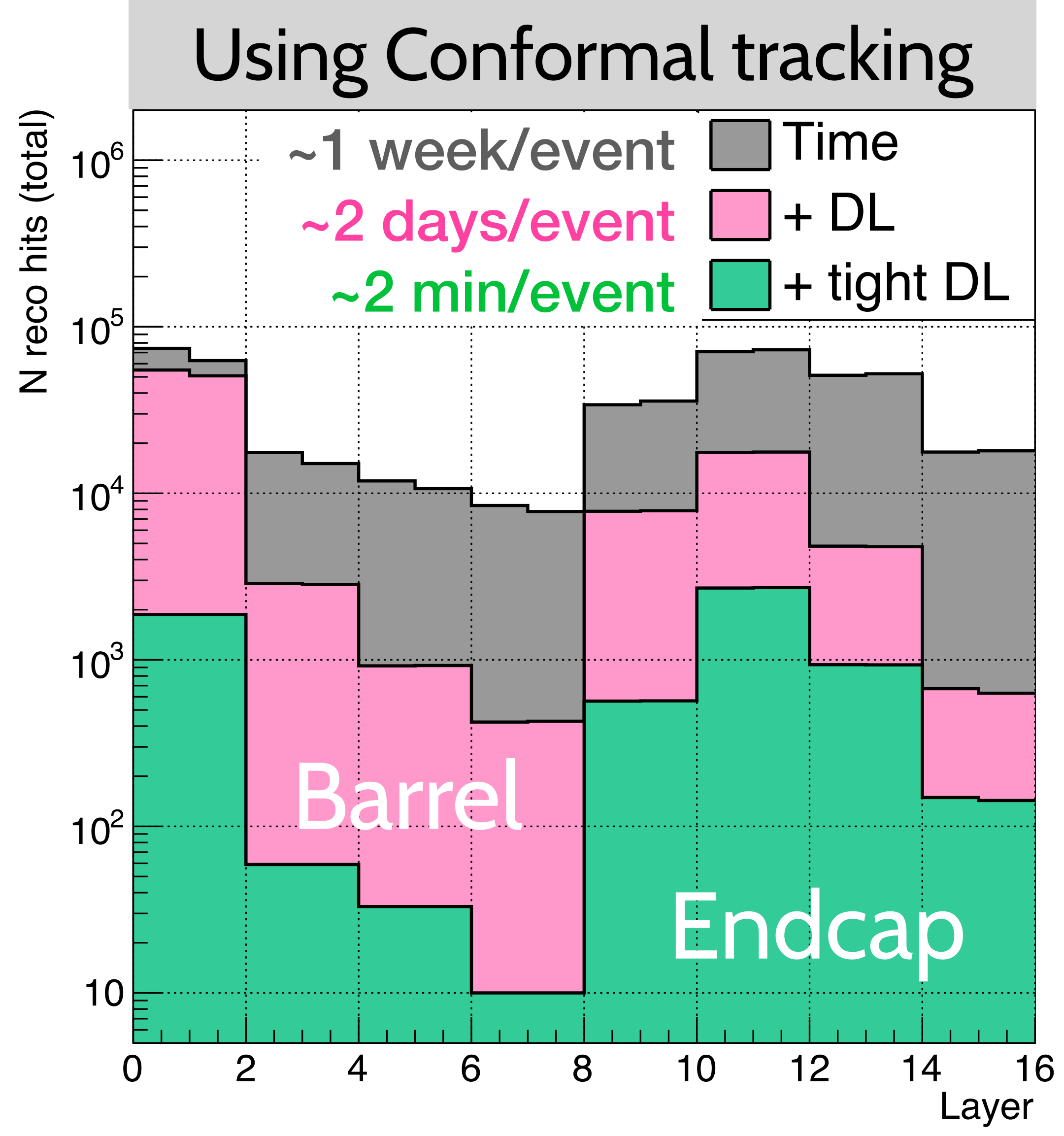
Impacts acceptance for displaced tracks affecting the b-tagging performance

↳ has to be used with care, keeping the track topologies in mind

Trying to maintain each filtering stage as a standalone configurable processor

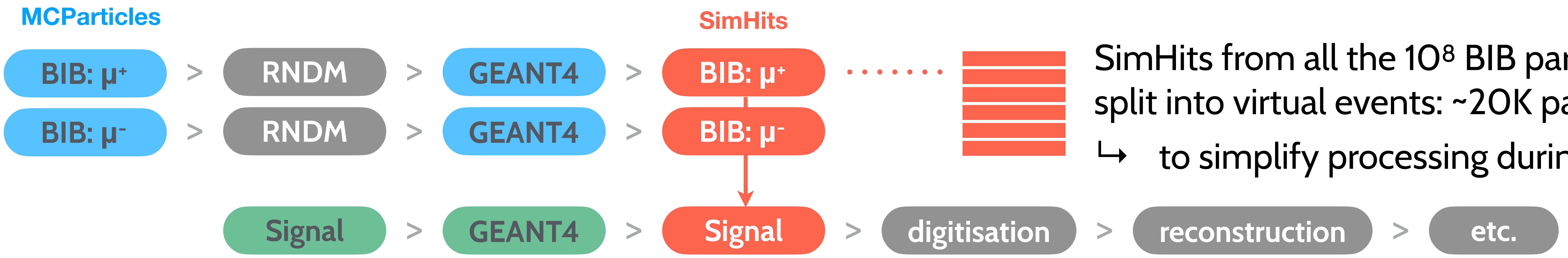
- **OverlayTimingGeneric:** selects SimHits within a wide time window for digitization
- **DDPlanarDigiProcessor:** drops out-of-time hits after time smearing during digitization
- **FilterDoubleLayerHits:** drops hits without a pair aligned with the IP





BIB organisation: event structuring

With the current approach we package BIB from the whole BX into just 2 files



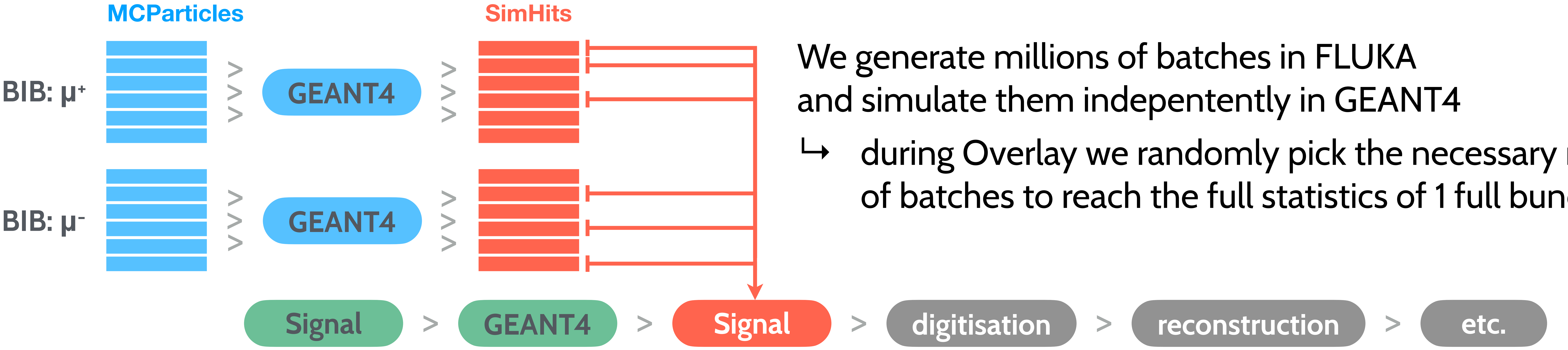
SimHits from all the 10^8 BIB particles are split into virtual events: ~20K particles/event

↳ to simplify processing during Overlay

BIB from the whole BX is treated as a single entity: 1K BIB simulations → 1K independent events

In FLUKA we force every muon to decay and record the resulting particles reaching the MDI surface

↳ muon decays are simulated in batches: e.g. 200 muons/batch → results written to small files (1 batch/file)



We generate millions of batches in FLUKA and simulate them indepentently in GEANT4

↳ during Overlay we randomly pick the necessary number of batches to reach the full statistics of 1 full bunch crossing

Summary

Muon Collider detector-simulation software is based on the solid ILCSoft framework
with centrally distributed releases: standalone + containers

We now have full control over the BIB generation thanks to the FLUKA-based workflow
providing new opportunities for detector + MDI optimisation

A lot of developments on top of the baseline CLIC version have been implemented
to provide sufficient computational performance in presence of BIB

Many ongoing developments are concentrated on improving the detector performance
at the level of digitisation and reconstruction algorithms

Considering a transition to Key4HEP framework
after several performance issues are resolved