Dual-readout simulation in IDEA_o2

Lorenzo Pezzotti
INFN Bologna

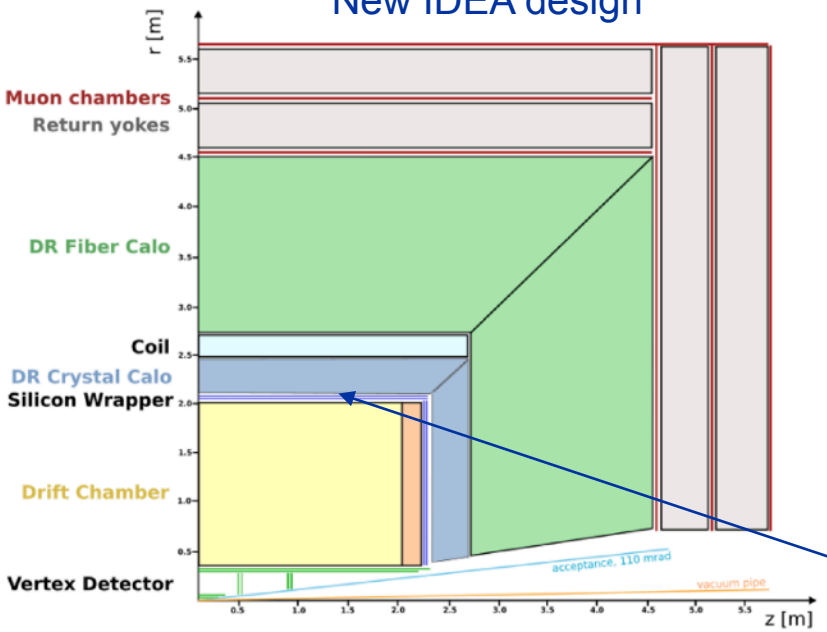
RD_FCC Simulation and Analysis Meeting
15/10/2025

Istituto Nazionale di Fisica Nucleare

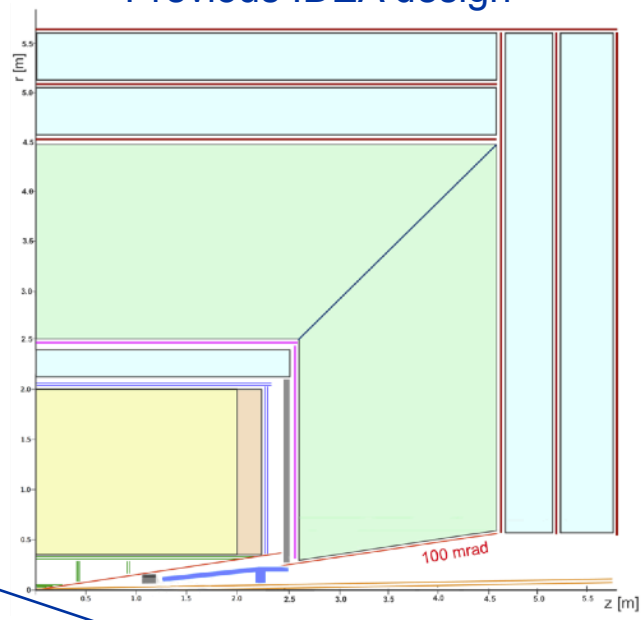
IDEA Detector Concept(s) simulations

- Currently there are two simulations for the IDEA Detector Concept in the FCC SW stack (Key4hep)

New IDEA design



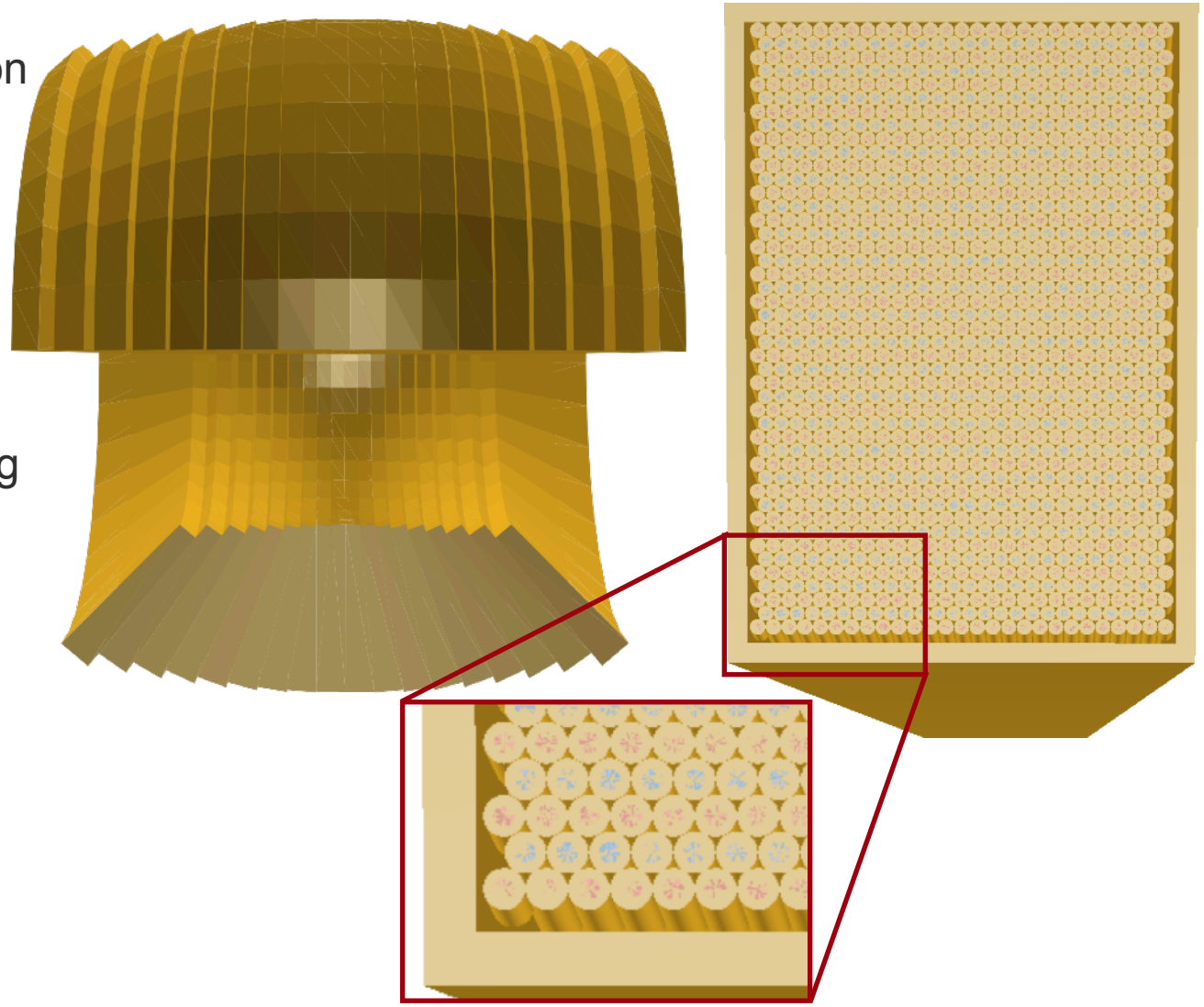
Previous IDEA design



- As recently the IDEA Study Group introduced a new design in which a crystal-em calorimeter is included (before the solenoid) together with a new Hidra-like calorimeter
- Both designs are available in full-simulation as IDEA_o(ption)1 (previous one) and IDEA_o(ption)2 (new one)

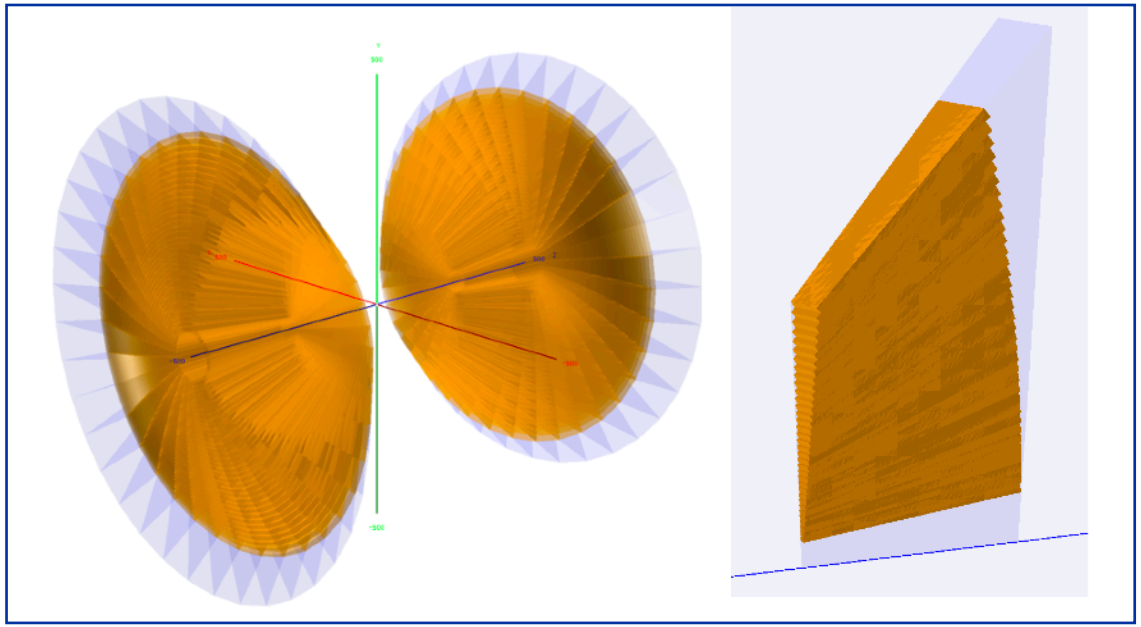
Tubes-based dual-readout calo @IDEA_o2

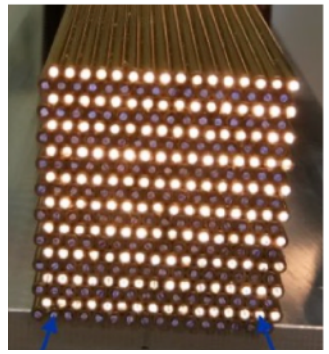
- The creation of the Hidra-like calorimeter simulation started in early 2024 and was completed at the beginning of 2025
- This geometry describes a novel dual-readout calorimeter design
  - now included in key4hep for IDEA_o2
  - exploiting a new construction technique housing optical-fibers into capillary-tubes
  - stands on previous test-beam simulations for which good Monte-Carlo to test-beam data agreement was found [article]

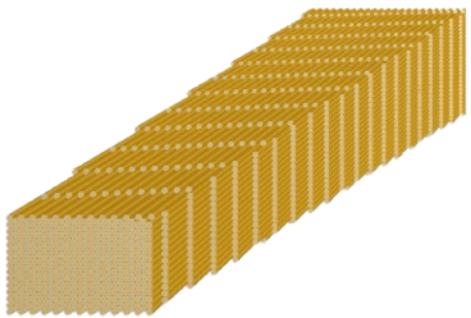


Tubes-based dual-readout calo @IDEA_o2

- The creation of the Hidra-like calorimeter simulation started in early 2024 and was completed at the beginning of 2025
- This geometry describes a novel dual-readout calorimeter design
  - now included in key4hep for IDEA_o2
  - exploiting a new construction technique housing optical-fibers into capillary-tubes
  - stands on previous test-beam simulations for which good Monte-Carlo to test-beam data agreement was found [article]
- The full calorimeter (barrel + endcap) simulation corresponds to ~80×10^6 tubs (touchable volumes)
  - Most of the IDEA_o2 volumes and hits are located in this subdetector!





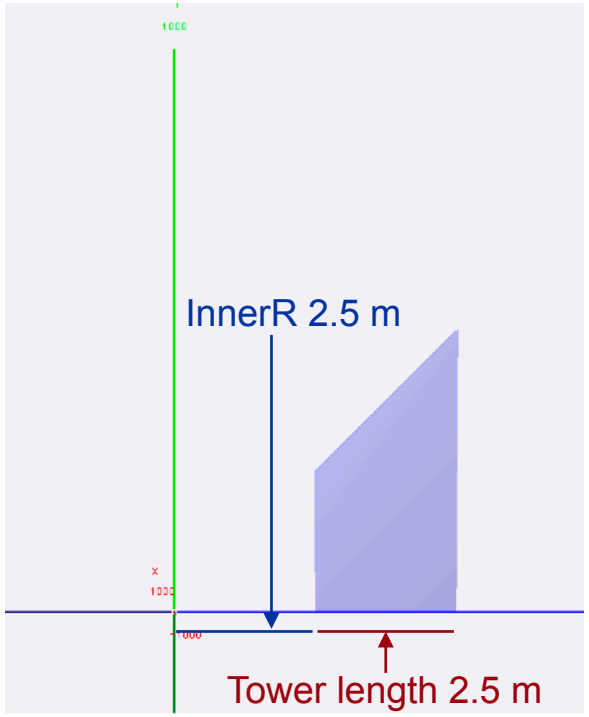


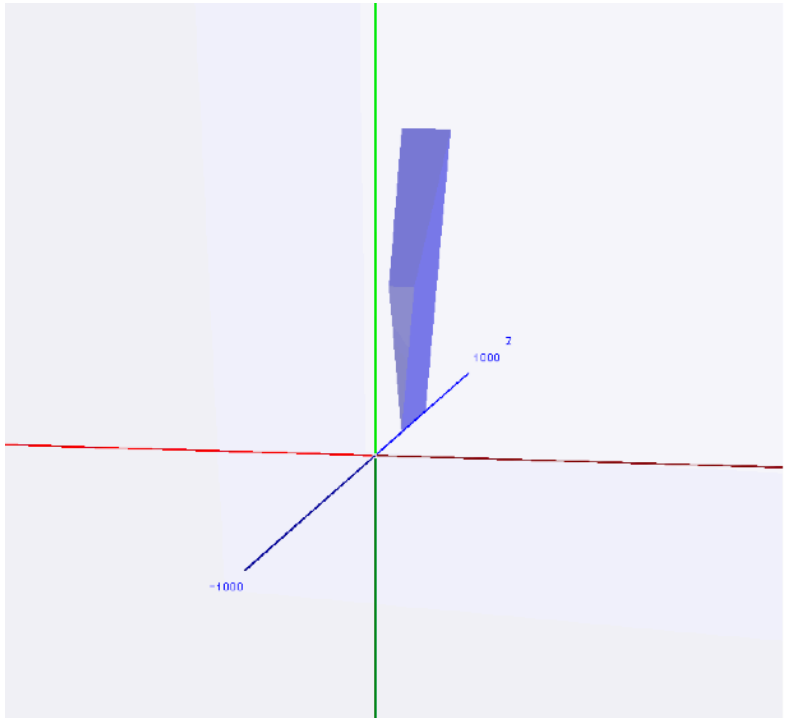
Geometry nesting

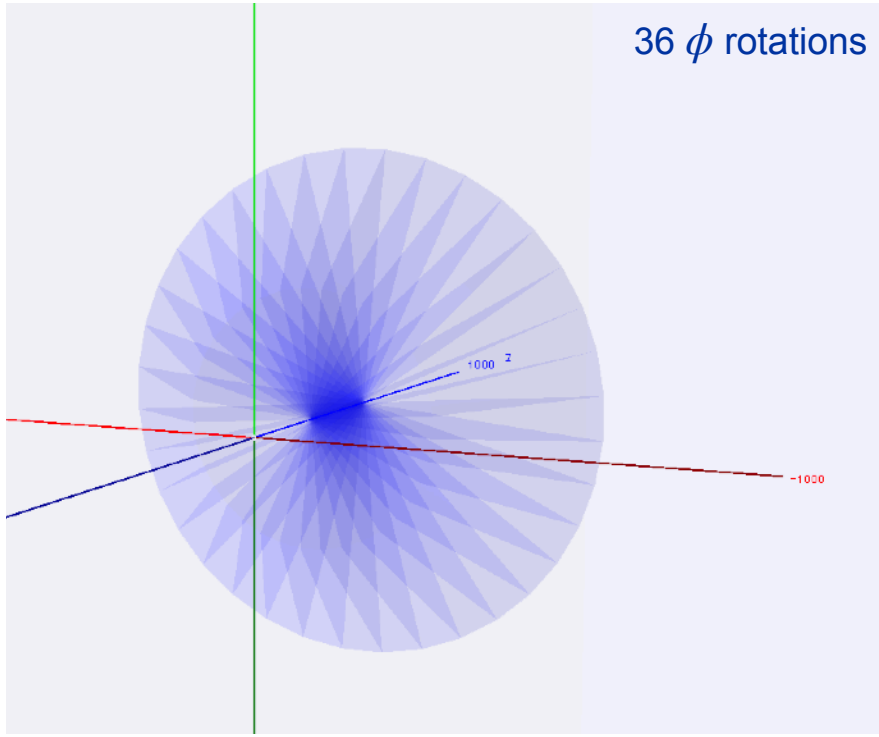
- The capillary tubes technology has no “tower” object, instead it glues together (many!) tubes to create a tower (trapezoid). However,
  - placing in the simulation individual tubes directly in the endcap/barrel region (or a ϕ-slice of it) is not the ideal solution as
  - navigating through many volumes is always a bad idea (even if we have voxelization),
  - also, having set of tubes housed in one “tower” will help the reconstruction, for instance for the calibration when each tower will be calibrated based on its signal and deposited energy from the MC truth
- Better to have “towers” of Air material and placing tubes inside each tower
  - Overall four levels of nesting are used

ϕ-slice

- A single ϕ-slice is a DD4hep EightPointSolid (equivalent to root Arb8, or Geant4 G4GenericTrap)
  - It is build according to the calorimeter inner radius, tower length, and ϕ-unit
  - Assumes that the barrel and endcap region will touch at $\pi/4$
  - Is replicated around the z-axis to full coverage

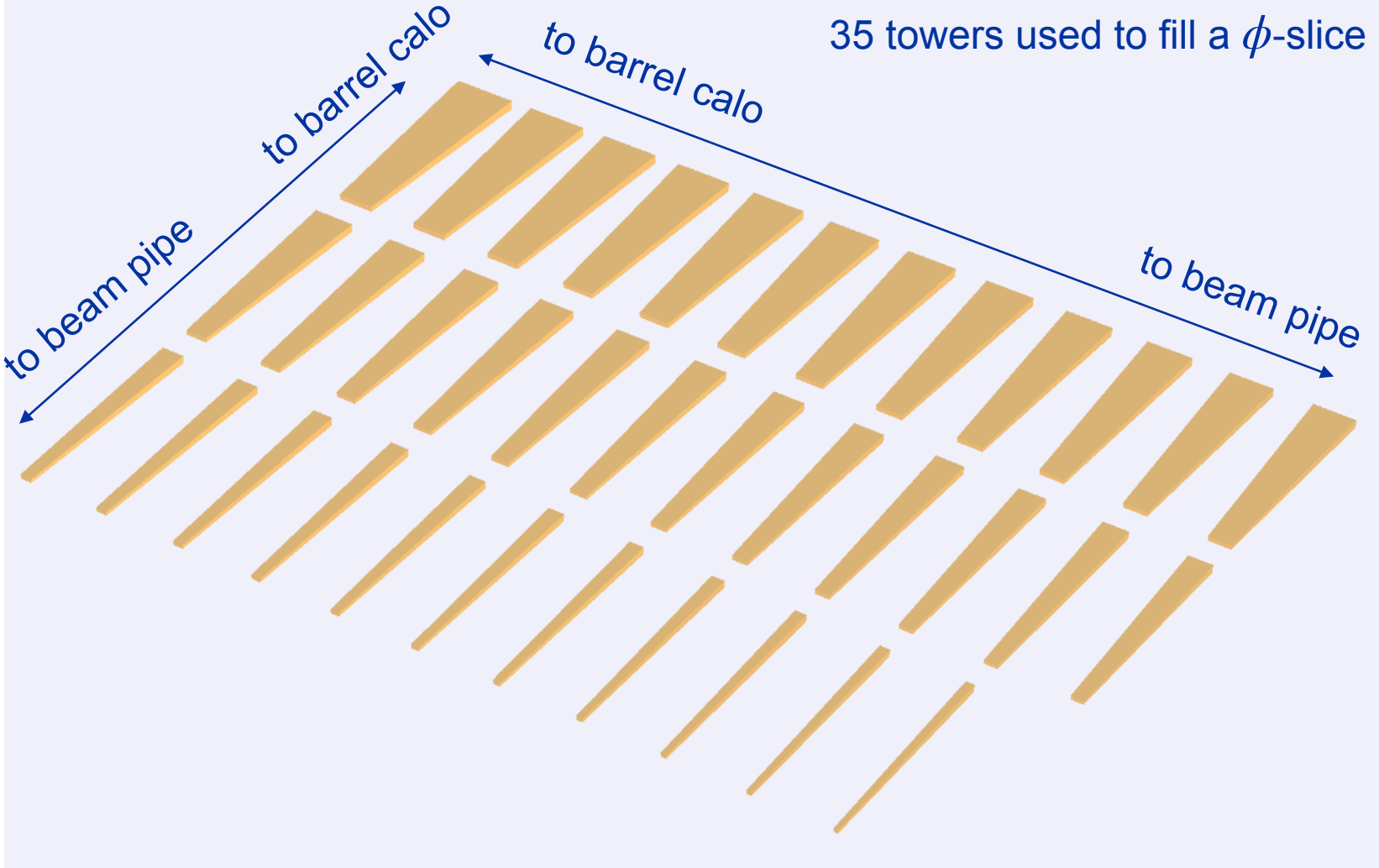






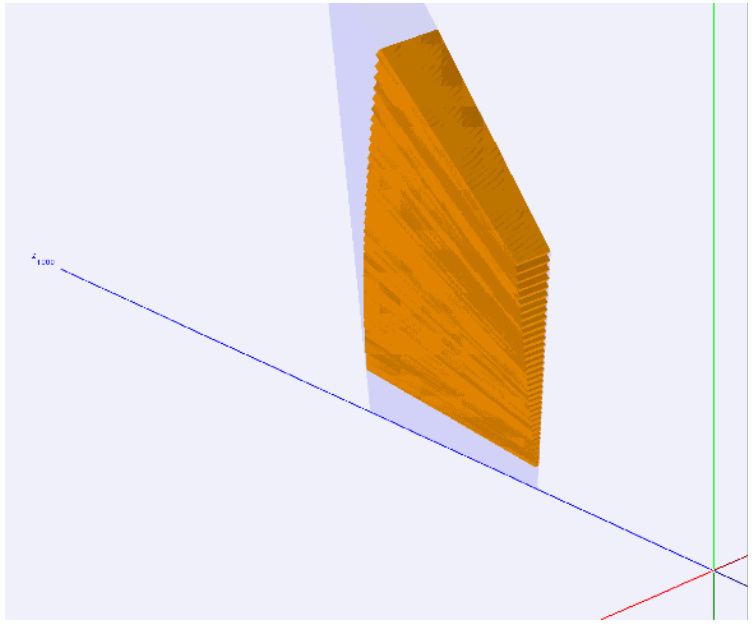
Towers

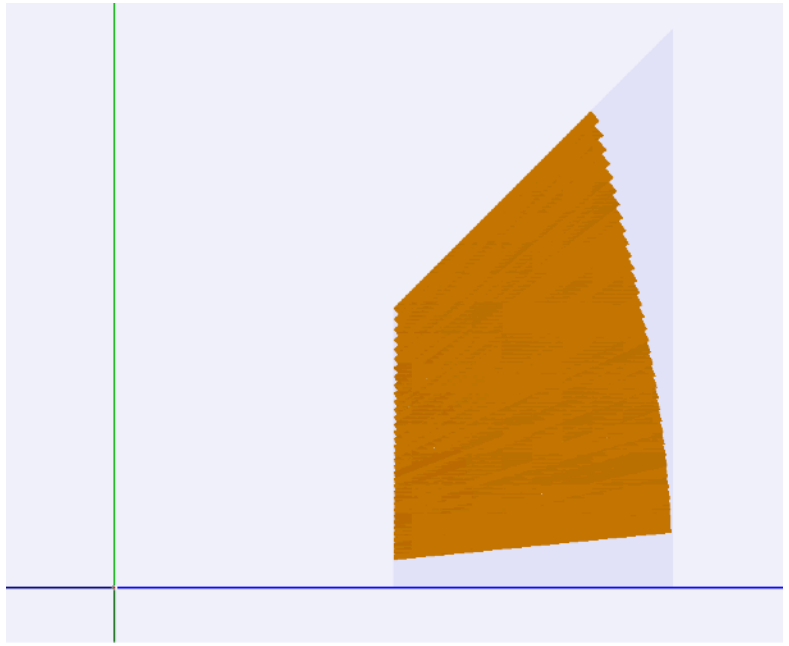
- Towers are Trap (equivalent to Geant4 G4Trap) that define a θ-region inside the ϕ-slice. The center of each tower points to the IP.
- As towers must fit inside the ϕ-slice their dimension changes with θ, this is clearly visible in the endcap regions

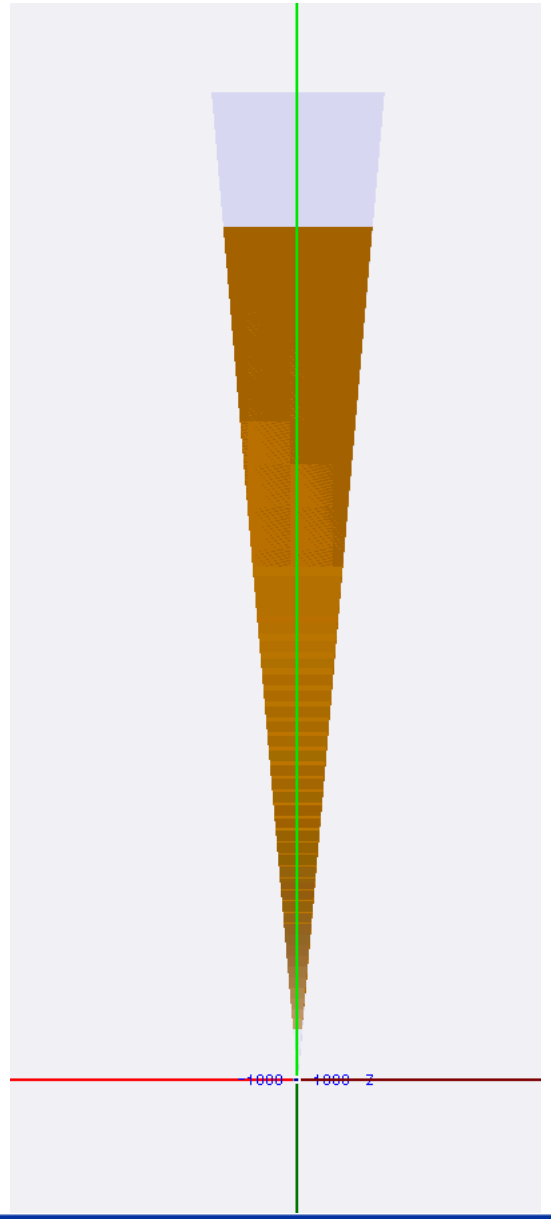


Towers

- Towers are Trap (equivalent to Geant4 G4Trap) that define a θ-region inside the ϕ-slice. The center of each tower points to the IP.
- Towers are placed in a projective manner inside the ϕ-slice
- The tower length is identical to every tower, *i.e*. the calo containment is independent of θ







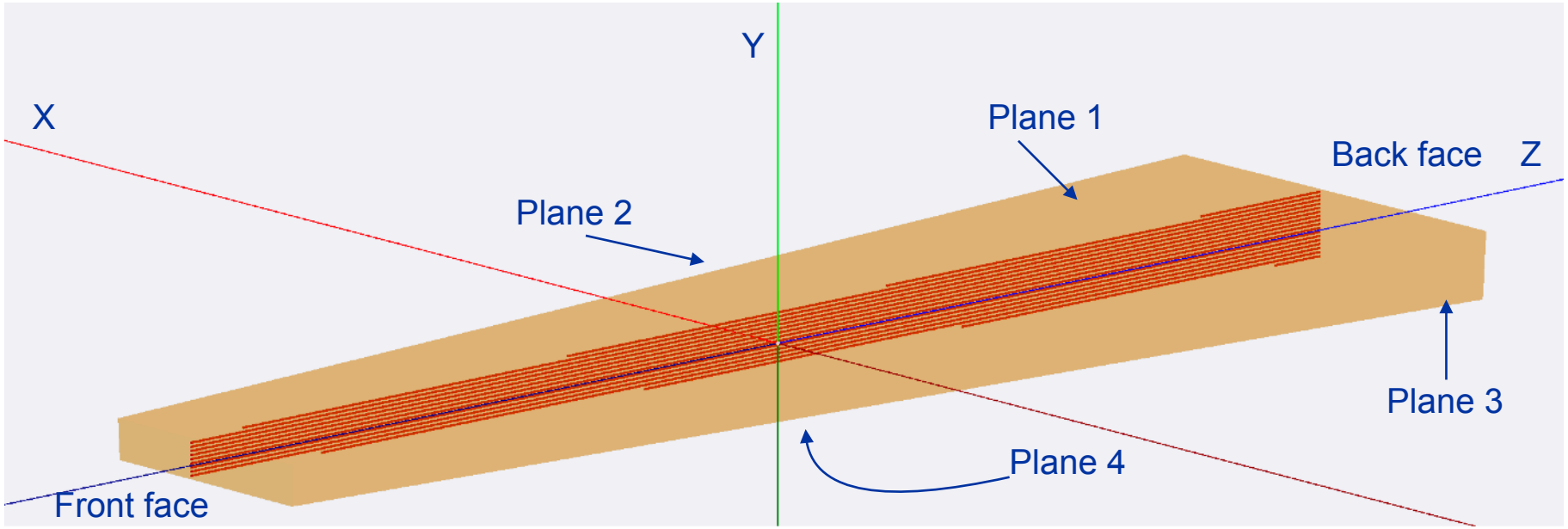
Tubes

- Tubes are 1-mm-thick radius Tubs (equivalent to G4Tubs) and house Scintillating or Cherenkov (clear) optical fibers. In the following optical fibers inside tubes are not displayed to aid visibility.
- Tubes z-axis is always parallel to the tower axis, i.e. they are projective pointing to the IP
- Tubes are placed starting from the back face of the tower (the biggest one) and,
- the tube length is changed towards the edges of the Trap to get the trapezoidal shape (examples are showed in the following)

Tubes placement (y-direction)

A tower back face

A tube column (of Scintillating fibers) whose x-y position is distributed over the the back face of a tower

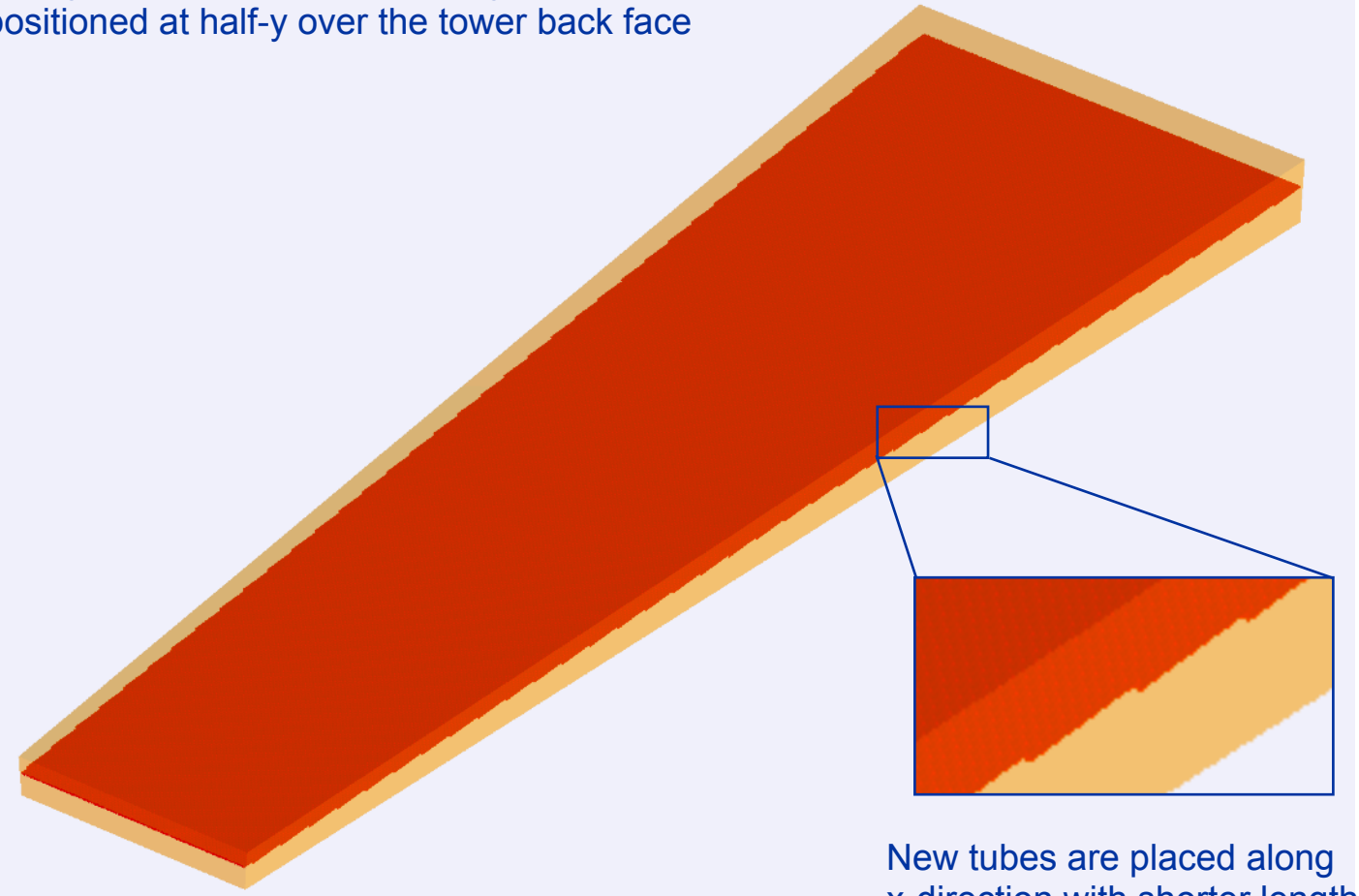


- For each tube starting at x-y from the back face, I calculate the intersection of a line parallel to the z-axis passing through x-y with each of the 4 planes in figure + the front face
- The shortest intersection defines the tube length
  - In reality some corrections are needed because the intersection happens between a plane and a cylinder (the tube)
- Tubes intersecting with the front face have the same length and are represented by the same Solid object in memory
- Tubes intersecting with any other plane are of specific length and are created on the fly

Tube radius was increased to 2 mm to help visualization

Tubes placement (x-direction)

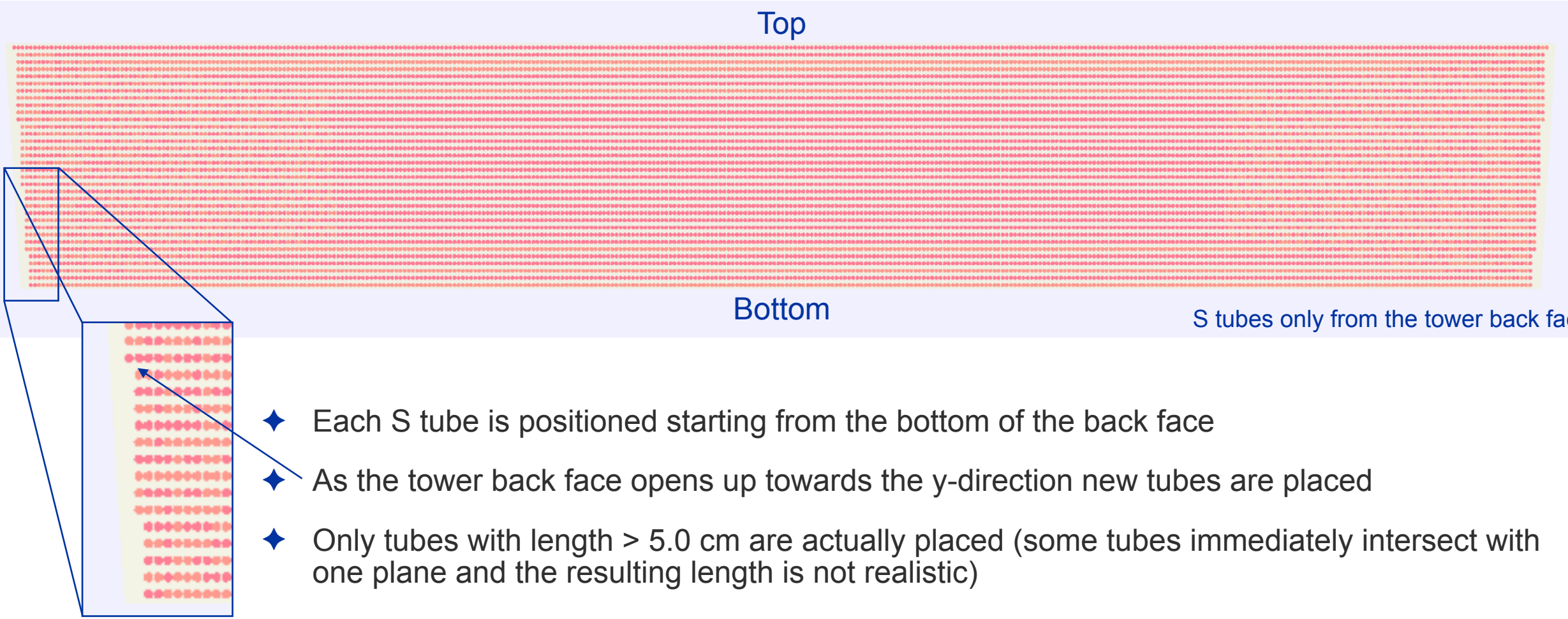
A single tube row (of Scintillating fibers)
positioned at half-y over the tower back face



New tubes are placed along
x-direction with shorter length

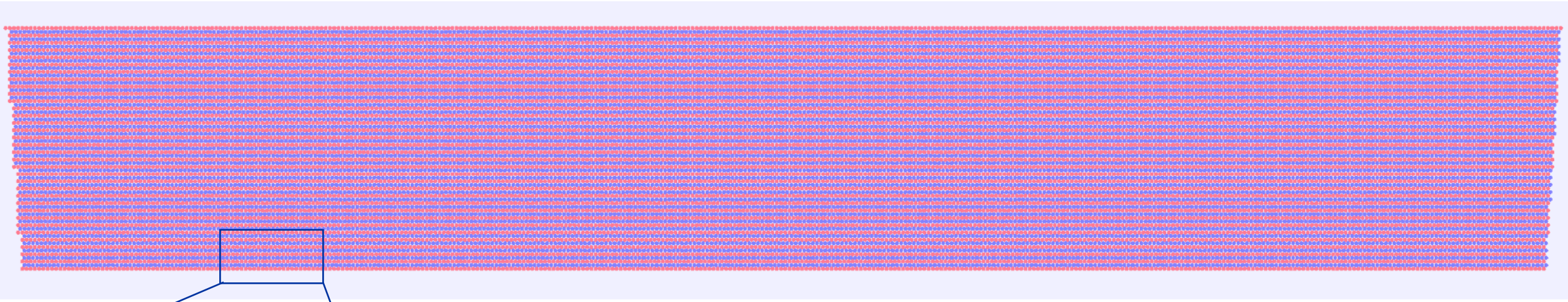
Tube radius was increased to 2 mm to help visualization

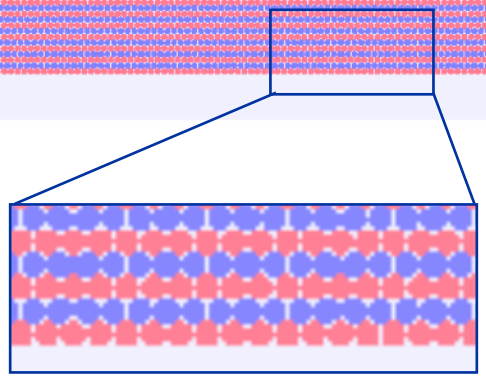
Filling the towers (Scintillating tubes only)

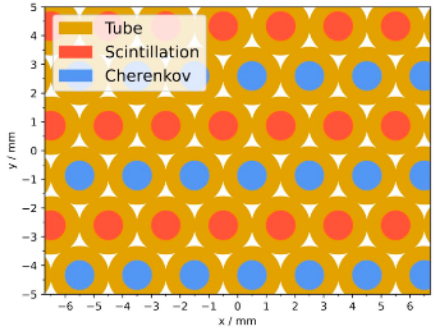


- Each S tube is positioned starting from the bottom of the back face
- As the tower back face opens up towards the y-direction new tubes are placed
- Only tubes with length > 5.0 cm are actually placed (some tubes immediately intersect with one plane and the resulting length is not realistic)

Filling the towers (all tubes)



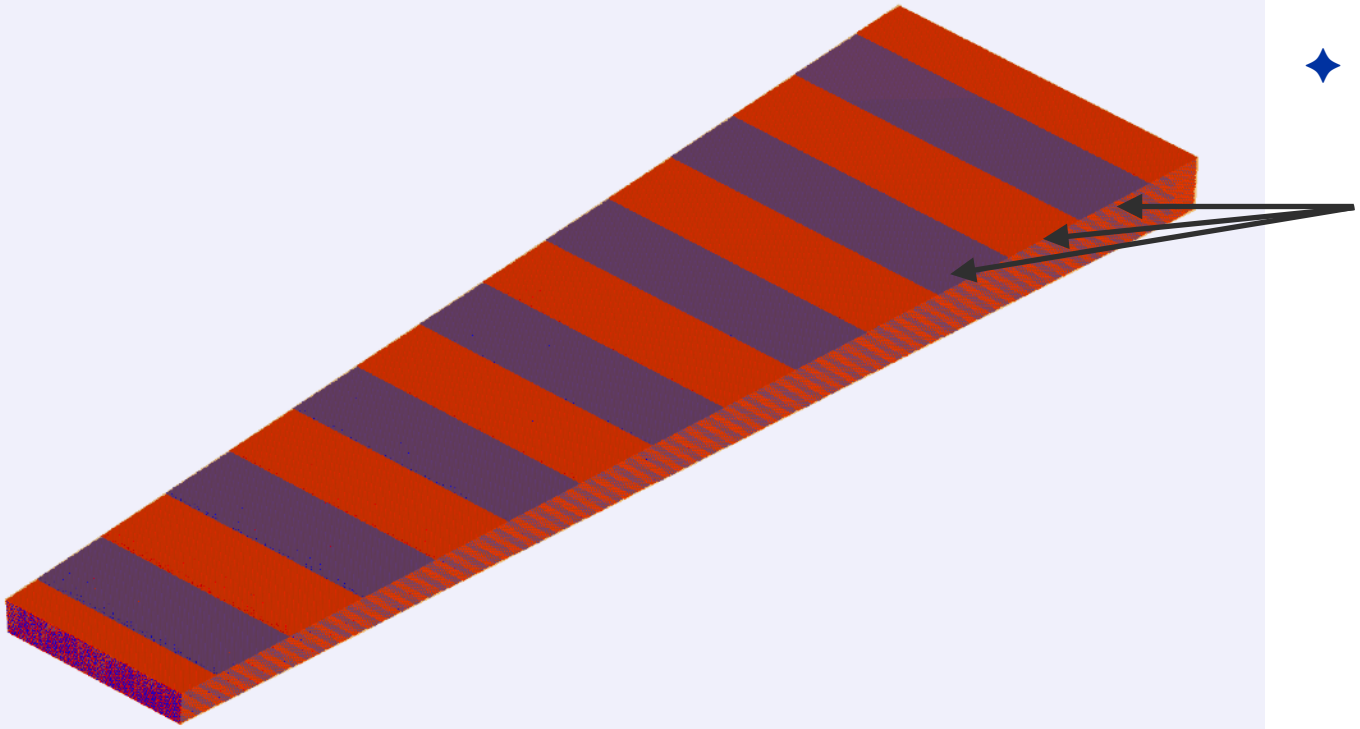




- The same technique applies to Cherenkov tubes housing clear plastic fibers
- C tubes (blue) are placed as rows in between S ones (red)

Filling the towers (all tubes)

A single tower filled with 24477 tubes

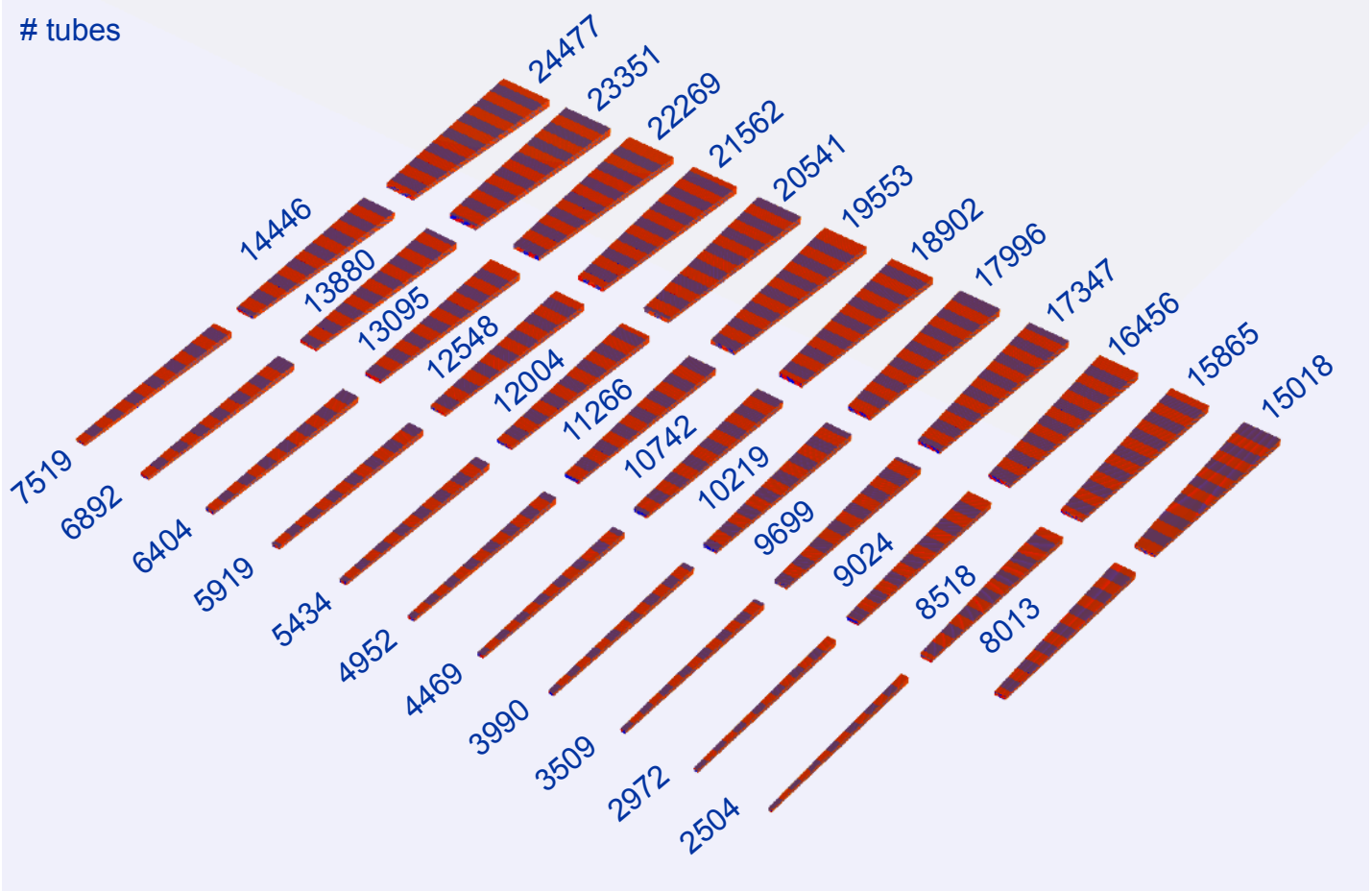


- A tower fully filled with tubes (S and C)
- S and C tubes alternates in y direction and the y-position determines the tube length
  - hence the colored bands visible from the top view

Some numbers

- Considering these parameters: 36 rotations around ϕ, 35 towers along θ, inner radius 2.5 m, 2.5 m long towers, we get
  - tubes per ϕ-slice: 421325
  - tubes per both endcaps: 30335400*
  - total tube length: 47128.5 km

- If we reduce the tower length to 2 m
  - tubes per ϕ-slice: 346877
  - tubes per both endcaps: 24975144
  - total tube length: 33072.5 km

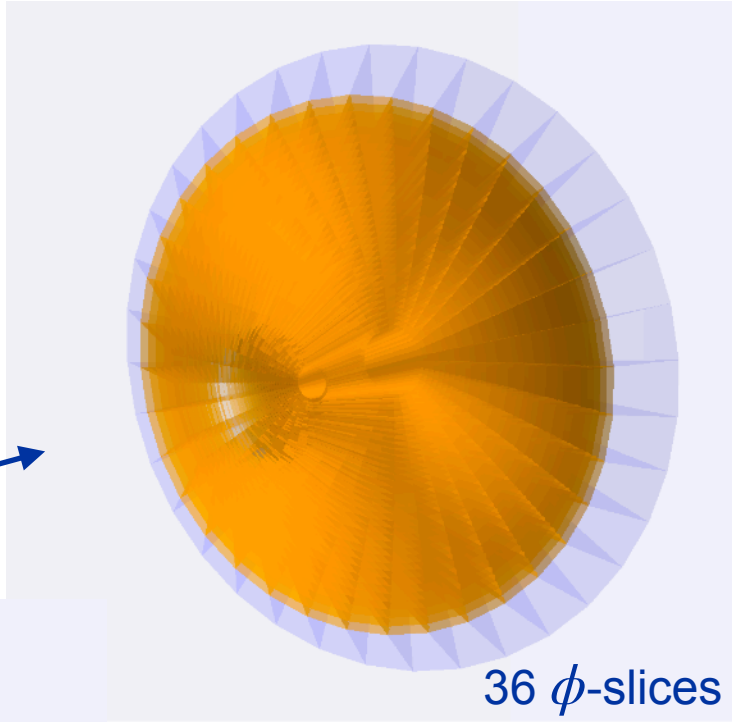


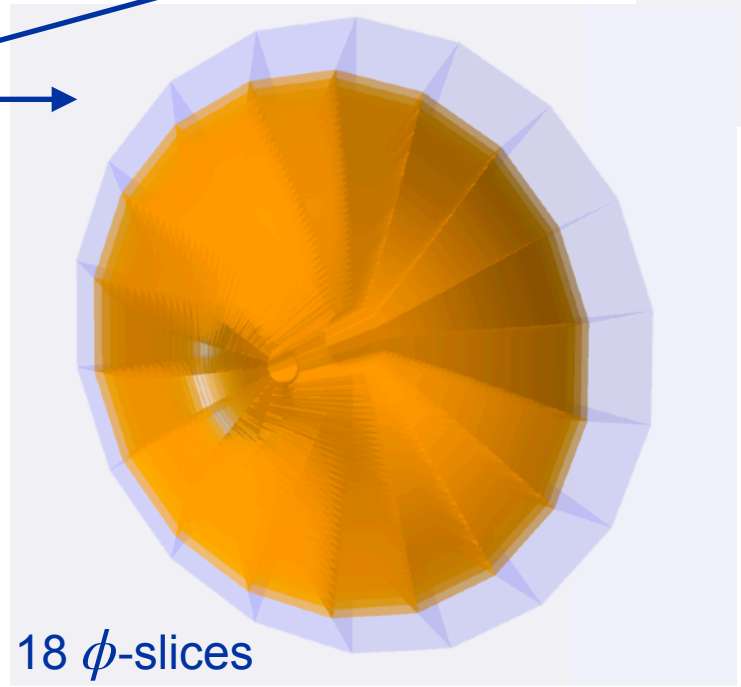
*the number of Tubs objects in memory is ~295648 (only counting for the absorber tubes)

Modularity

- All the geometry custom parameters are encapsulated in the XML description file

```
<define>
  <constant name="world_side" value="6*m"/>
  <constant name="world_x"    value="world_side/2"/>
  <constant name="world_y"    value="world_side/2"/>
  <constant name="world_z"    value="world_side/2"/>
  <constant name="innerRadius"value="2.5*m"/>
  <constant name="towerHeight"value="2.5*m"/>
  <constant name="NbOfZRot"   value="36"/>
  <constant name="TubeRadius" value="1.0*mm"/>
  <constant name="CladRadius" value="0.5*mm"/>
  <constant name="CoreRadius" value="0.45*mm"/>
</define>
```



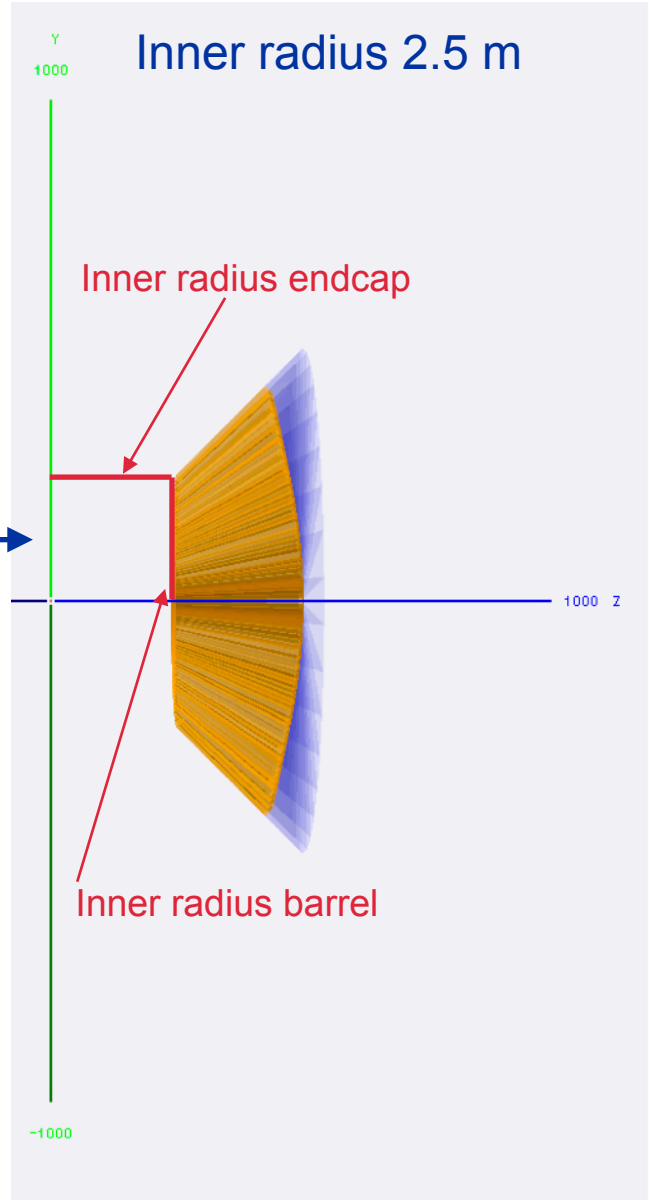


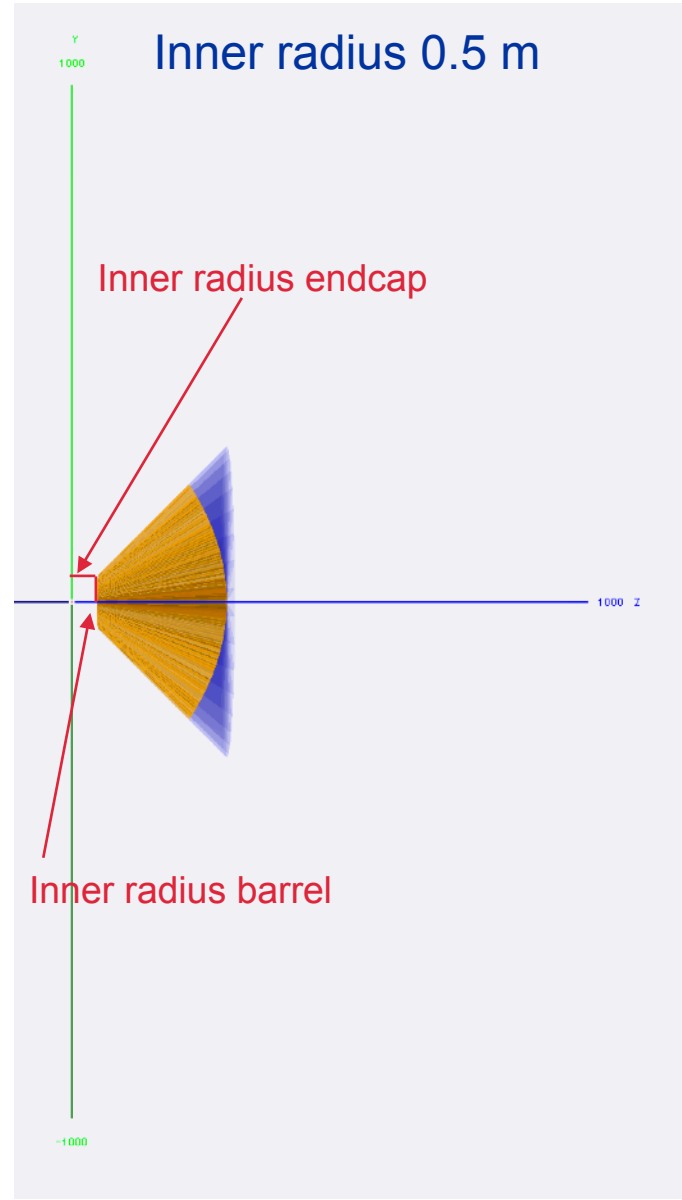
Modularity

- All the geometry custom parameters are encapsulated in the XML description file

```
<define>
  <constant name="world_side" value="6*m"/>
  <constant name="world_x"    value="world_side/2"/>
  <constant name="world_y"    value="world_side/2"/>
  <constant name="world_z"    value="world_side/2"/>
  <constant name="innerRadius"value="2.5*m"/>
  <constant name="towerHeight"value="2.5*m"/>
  <constant name="NbOfZRot"   value="36"/>
  <constant name="TubeRadius" value="1.0*mm"/>
  <constant name="CladRadius" value="0.5*mm"/>
  <constant name="CoreRadius" value="0.45*mm"/>
</define>
```

- The only geometry constrain: the barrel inner radius and the endcap inner radius are identical or, equivalently, the endcap starts at $\theta = \pi/4$



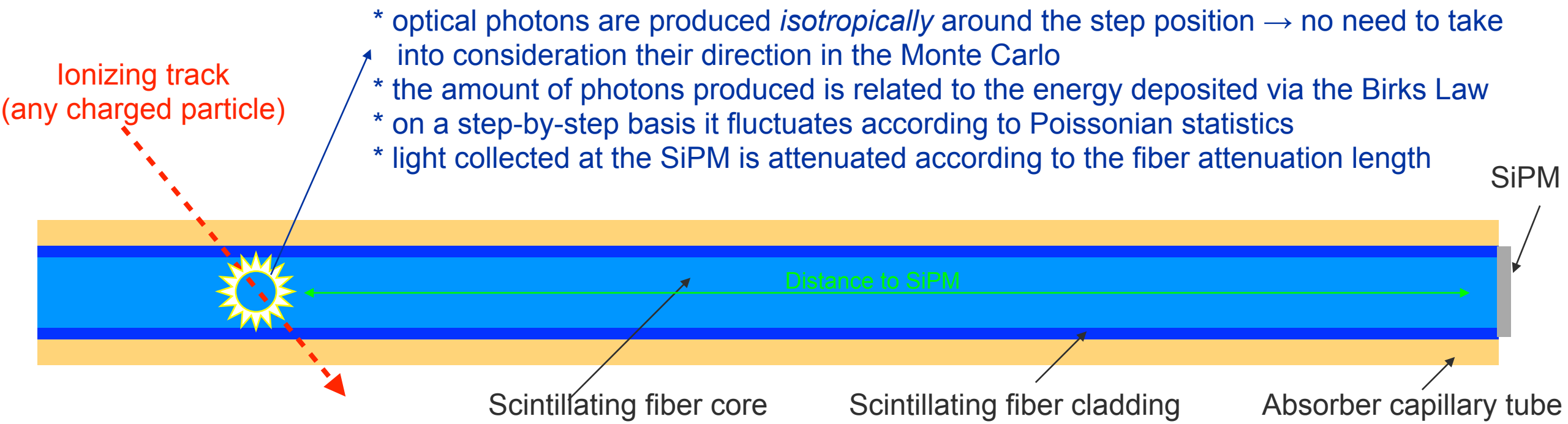


Signal treatment

- This calorimeter simulation must include two signals:
  Scintillation light in scintillator-doped optical fibers and *Cherenkov light* in clear optical fibers
- Simulating light in calorimeters has always been a problem:
  as of today no LHC Experiment simulation include light propagation in calorimeters, instead it is parametrized based on experimental inputs
- I believe the same should be adopted for the IDEA calorimeter simulation as a baseline to allow large MC productions
  - More advanced implementation are always possible but should be addressed in terms of physics improvements and CPU cost
- The following is an implementation as a DD4hep Geant4SensitiveAction<>

Scintillation signal simulation

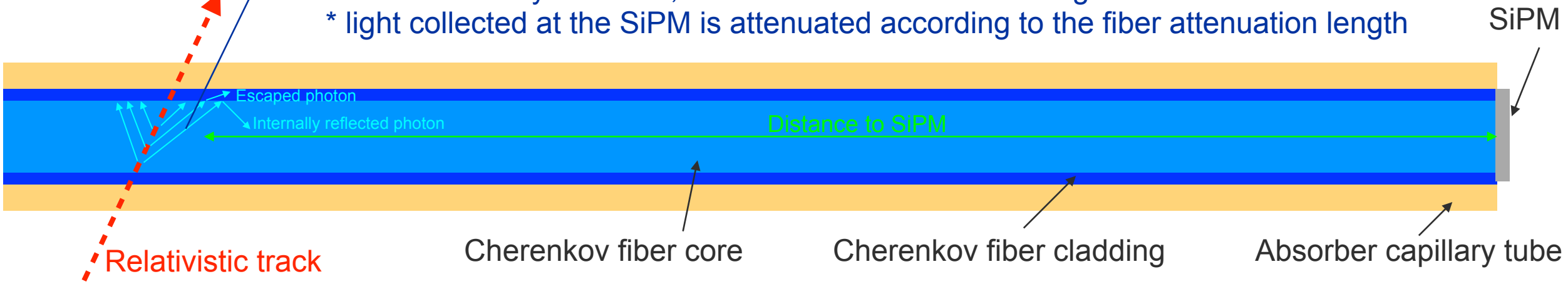
* optical photons are produced *isotropically* around the step position → no need to take into consideration their direction in the Monte Carlo
* the amount of photons produced is related to the energy deposited via the Birks Law
* on a step-by-step basis it fluctuates according to Poissonian statistics
* light collected at the SiPM is attenuated according to the fiber attenuation length



- This approach was validated against test-beam data from 2021 and 2023 finding a good MC-to-data agreement [Article]

Cherenkov signal simulation

* optical photons direction is calculated by the Geant4 Cherenkov process
  → the emission of the optical photon must be included in the Monte Carlo
* once reached the fiber cladding, photons can be internally reflected or may escape the fiber according to the material refractive index and the photon direction
  → we check with Geant4 if the photon undergoes an internal reflection
* if it is internally reflected, it will contribute to the fiber signal
* light collected at the SiPM is attenuated according to the fiber attenuation length



- This approach was validated against test-beam data from 2021 and 2023 finding a good MC-to-data agreement [Article]

Event rate

- Results from key4hep-nightlies on an lxplus9 machine with 2.9 GHz (single-threaded)
- Event time is taken using `G4Timer` between `BeginOfRunAction()` and `EndOfRunAction()`

```
ParticleHandler  INFO  +++ Event 9 Begin event action. Access event related information.
Geant4Output2EDM4hep INFO  +++ Saving EDM4hep event 9 run 0.
GenerationInit   WARN  +++ Finished run 0 after 10 events (10 events in total)
======================================================================
 --> DREndcapTubes: run terminated, 10 events transported
 --> DREndcapTubes: time: User=6.820000s Real=7.429838s Sys=0.090000s [Cpu=93.0%]
 --> DREndcapTubes: time per event: 0.682
======================================================================
```

- Calorimeter simulation output is an EDM4hep file containing one hit per fiber with the following information:
  - 64-bit identifier
  - Total signal in photo-electron
  - (x,y,z) of the fiber inner tip
  - Including calo hit contributions is yet to be done

electron	s/evt	s/evt/GeV
10 GeV	0,64-0,70	~0,065
90 GeV	5,7-6,0	~0,065
160 GeV	10,4-10,7	~0,065
240 GeV	16,2-16,7	~0,067
365 GeV	24,8-25,1	~0,068

Memory footprint (just a few words)

- A DD4hep simulation using this sub detector has a memory footprint of ~15.4 Gb
  - 2.1 Gb for the DD4hep/TGeo geometry representation
  - 2.2 Gb for the Geant4 geometry representation
  - 11.1 Gb of information cached during the geometry conversion

Default DD4hep

- DD4hep developers (many thanks to Markus and Andre) introduced `regexSensitiveDetector`, it allows to
  - Associate a sensitive (detector) action to every volume with name matching a substring pattern
  - No volume is marked as *sensitive* in geometry construction → no cache is created
  - Memory footprint is reduced to 4.3 Gb (de facto only geometry)

DD4hep + regexSensitiveDetector

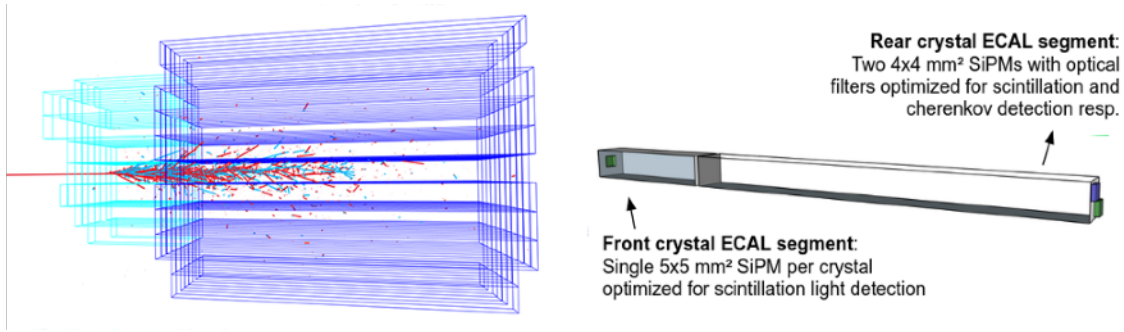
About `regexSensitiveDetector`

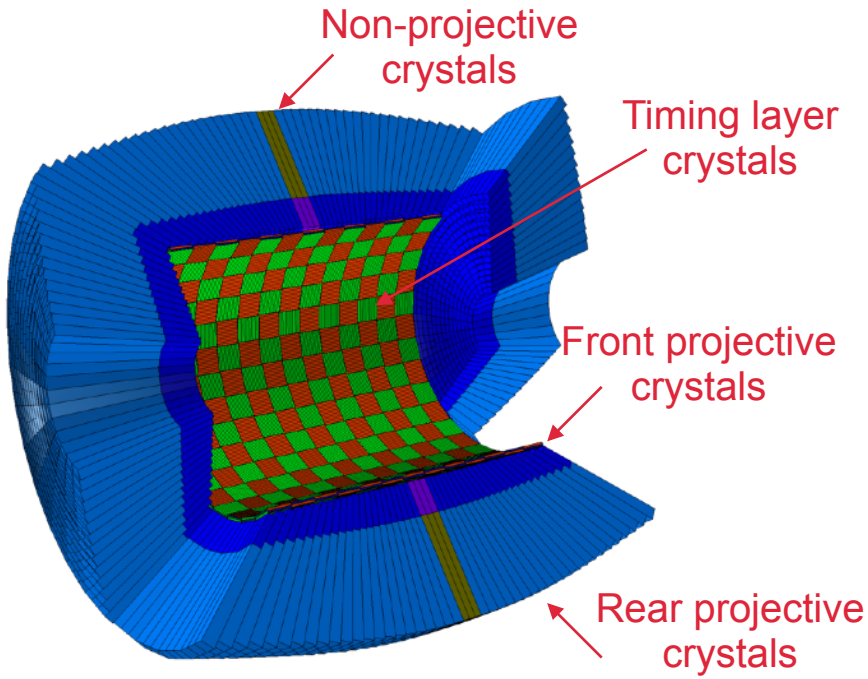
- `regexSensitiveDetector` unfortunately is not a free lunch. Some drawbacks:
  - Initialization time is slowed down heavily (a regex search over million of strings…)
  - Some DD4hep methods are not working any longer, for instance:
    - DD4hep 64 bit volume IDs stored in the TGeo geometry representation and only accessed in the Geant4 simulation via the cache. If the cache is not available there is not way we can retrieve the volume identifiers during the simulation step.
    - Our current solution is to recreate those identifiers on the fly during the simulation.
      Literally during the simulation with do bit shifting operations to recreate 64 bits from the simulation steps. Patchy and a bit slow but works.

- Recently DD4hep developers introduced a light volume manager which should create a much lighter cache for us → I had no time to test it yet but might be a long term solution for us!

Dual-readout crystals @IDEA_o2

- Recently the dual-readout crystal section simulation was developed in DD4hep
- The foal is to reach $\sigma/E \simeq 3\,\%/\sqrt{(E)}$ while not spoiling the dual-readout compensation technique when hadronic showers start showering in crystals. Solve the channelling effect for em-showers entering the fiber-calorimeter, help identification of γ's in jets
- Simulation: projective homogeneous ($PBWO_4$) crystal calorimeter
  - Each crystal is longitudinally segmented with front/rear section (6:16 ratio 22 X_0 (~20 cm))
  - Dual-readout capability ensured by two dedicated SiPMs instrumented on the rear section
  - Timing layer placed in front comprises two layers of fast-scintillating LYSO crystals with opposite orientation
- Some additional work is needed to properly simulate signals (S+C) in crystals





IDEA_o2 full simulation geometry