

ArgonCube Detector Geometry Simulation Status

**Jose Palomino, Guang Yang, Clark McGrew,
Chang Kee Yung.**

2017/10/16

ArgonCube Detector Geometry

- ❖ We have defined ArgonCube Detector Geometry using **DUNENDGGD**.
- ❖ **DUNENDGGD** is based on **GeGeDe*** (Brett Viren - BNL)
 - ★ GDML output that's compatible with both GEANT4 and ROOT
 - ★ Define geometries to evaluate the feasibilities of a wide range of possible detector configurations.
 - ★ Flexibly and quickly define geometry configurations.

<https://github.com/gyang9/dunendggd>

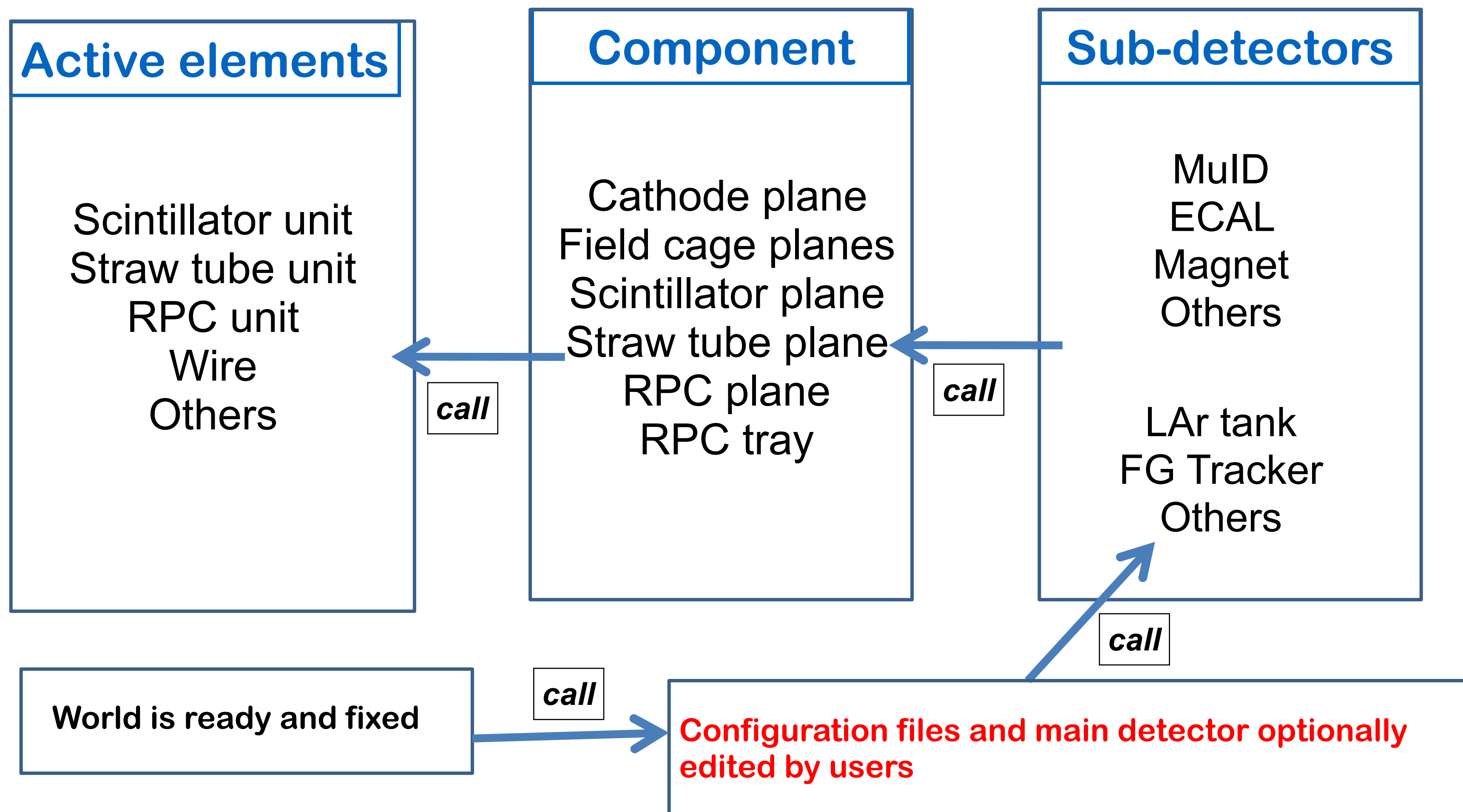
<https://github.com/brettviren/gegede>

General Geometry Description

- ❖ “GeGeDe is a software system to generate a description of a constructive solid geometry as represented in GDML files. Specifically as used by Geant4 or ROOT applications. It is implemented as a pure Python module”^{*}.
- ❖ The core of GeGeDe relies on Pint to enforce the consistent use of units.
- ❖ GeGeDe package was developed by Brett Viren, Brookhaven National Laboratory.

^{*}<https://github.com/brettviren/gegede>

DUNENDGGD Structure

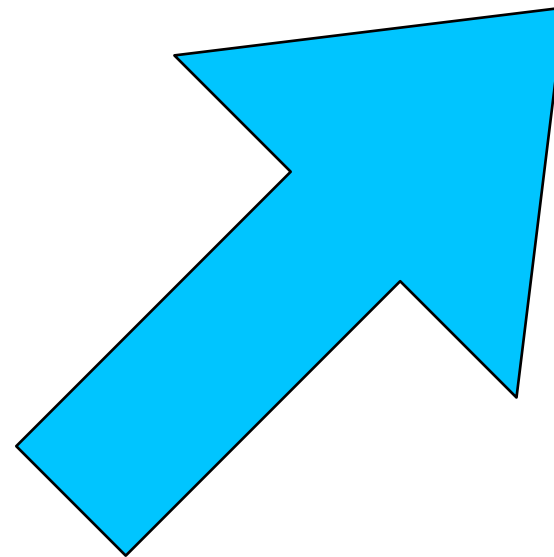
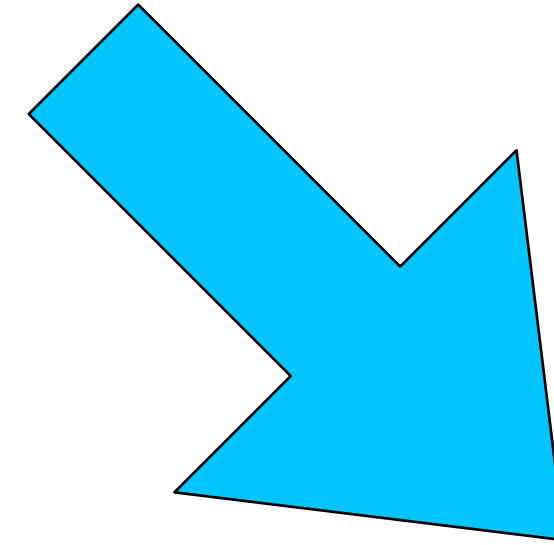


- ❖ All objects features can be defined in cfg files by users.
- ❖ It is generic structure, that each higher level just calls all sub-builders in lower level

DUNENDGGD can be used as input to..

GENIE

DUNENDGGD



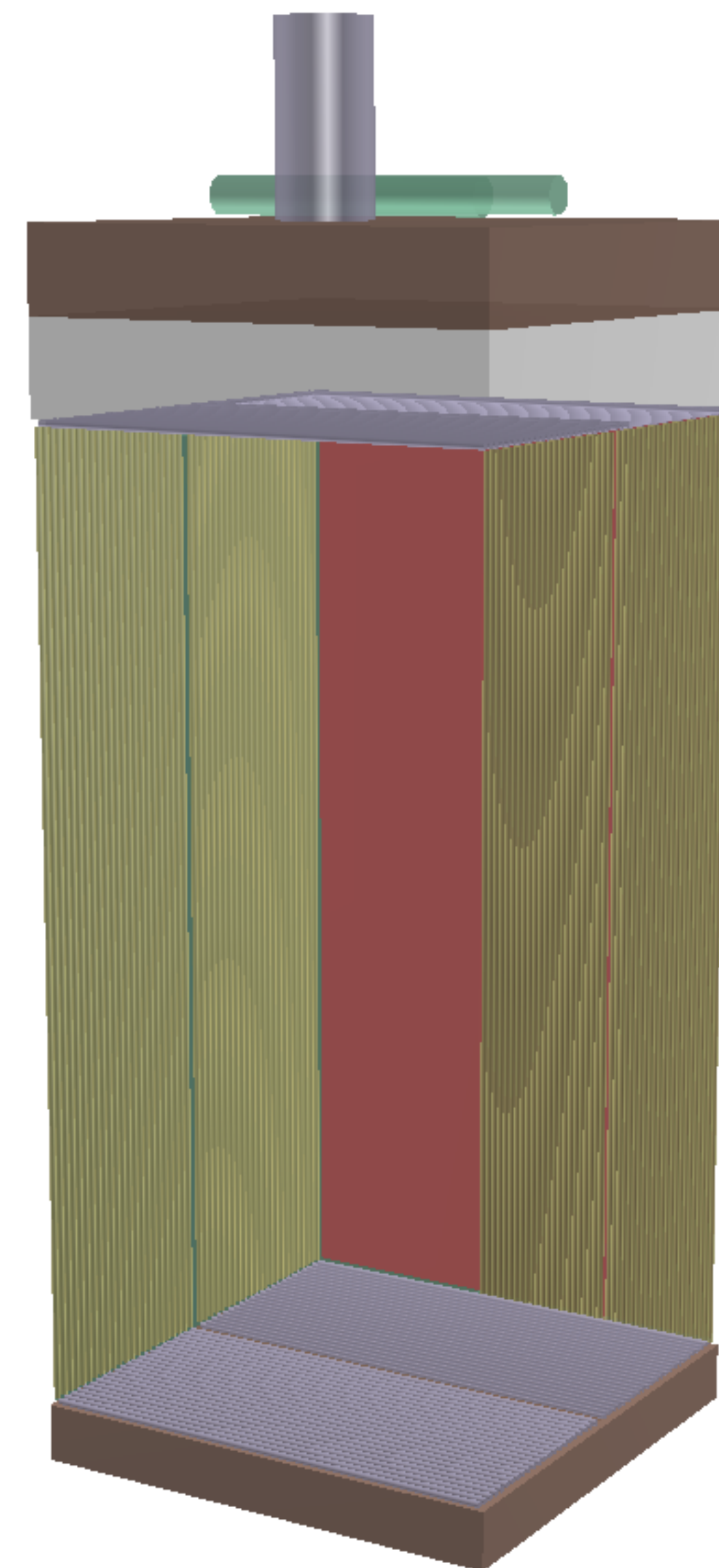
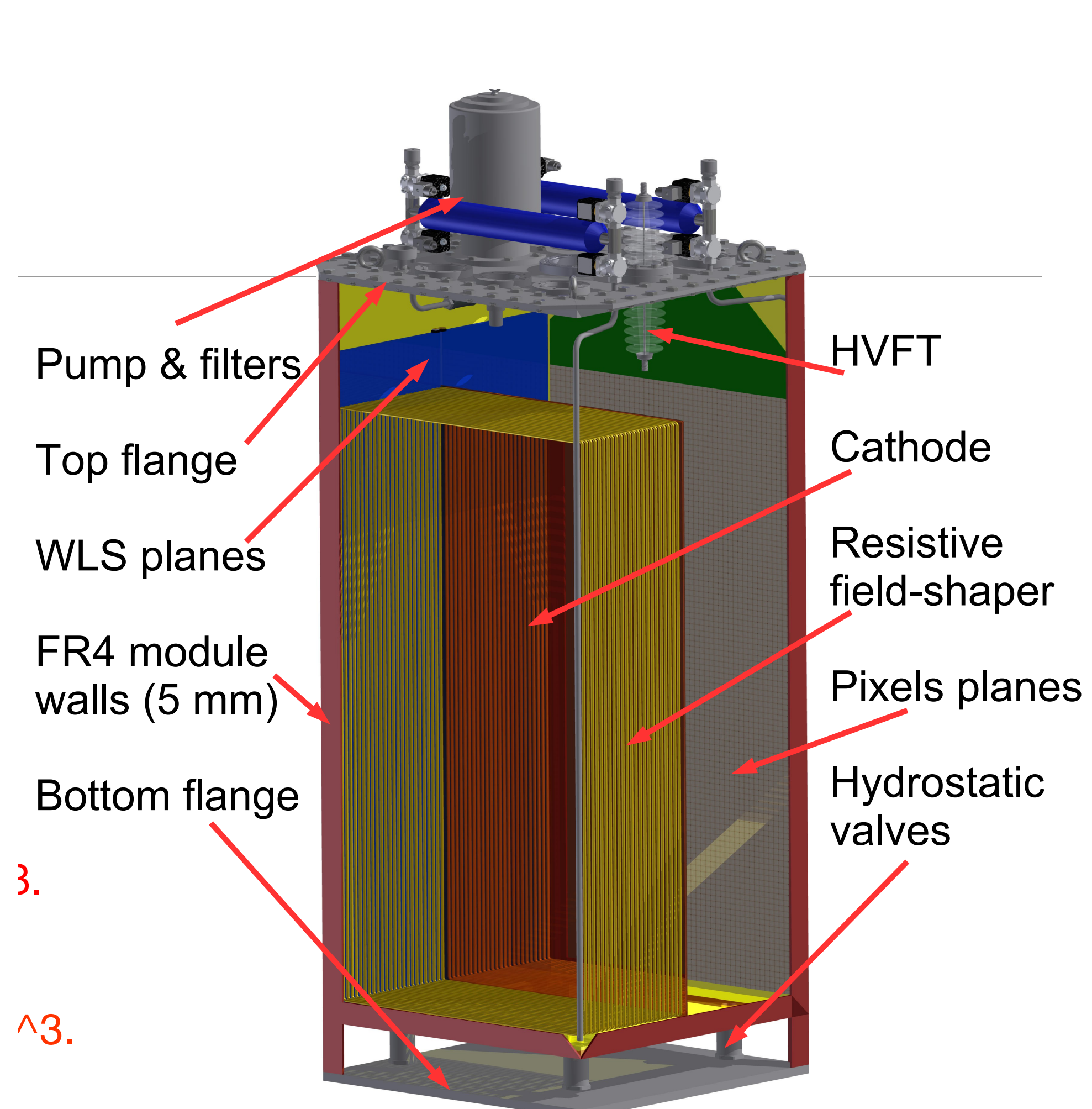
EDEPSIM

EDEPSIM dependencies:
ROOT, GEANT4

DUNENDGGD dependencies:
GeGeDe

DUNEND-X contains:
ROOT, GEANT4, GeGeDe

ArgonCube Module



DUNENDGGD:

Active LAr

Dead LAr

Dear GAr

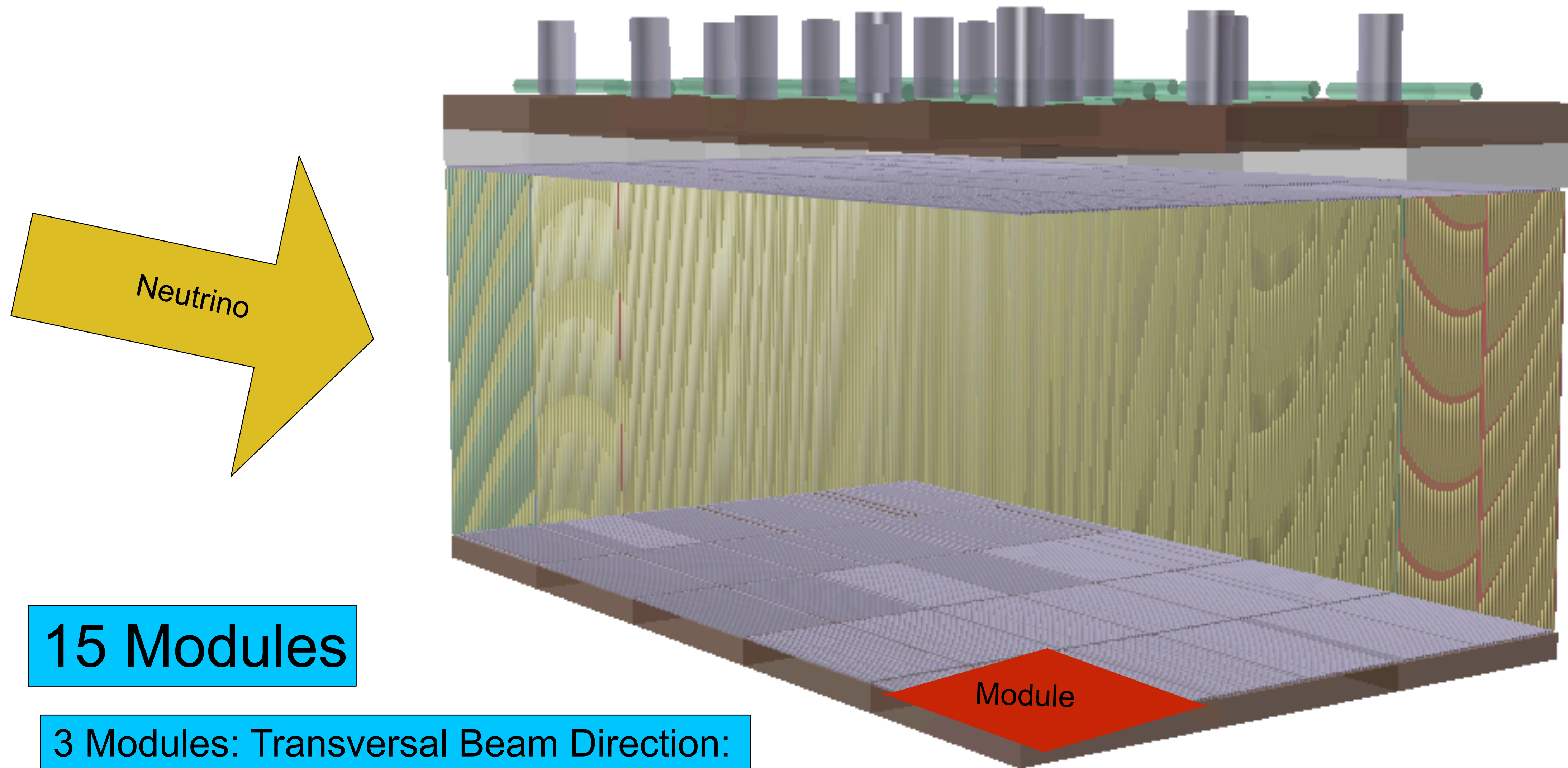
Field Cage (Copper)

Pump

Pixel Panel

Cathode

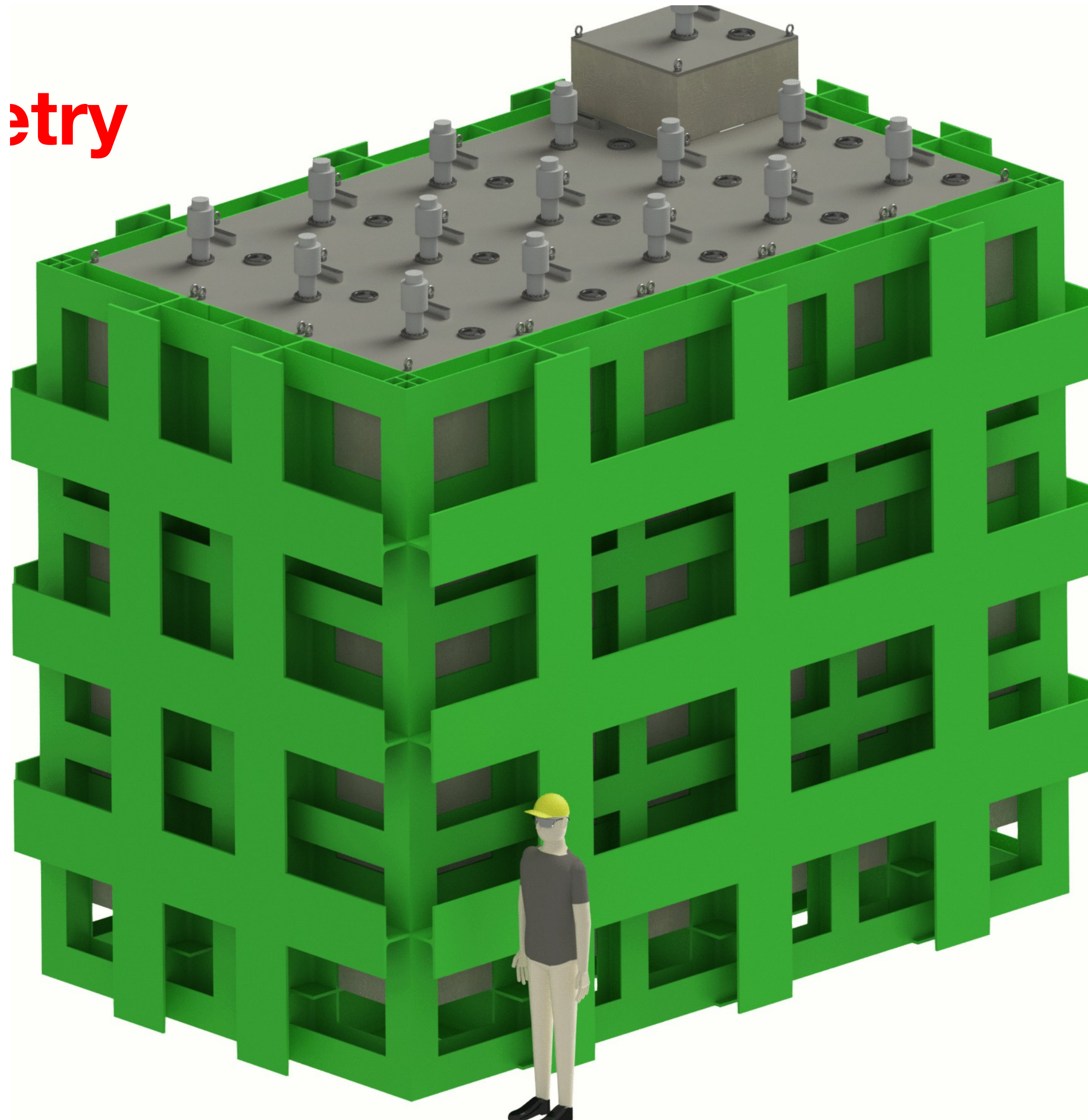
ArgonCube Detector



15 Modules

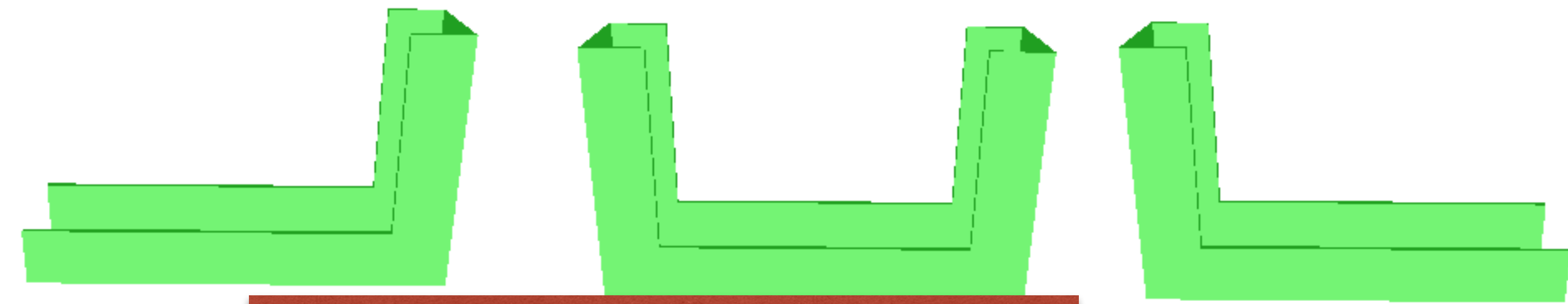
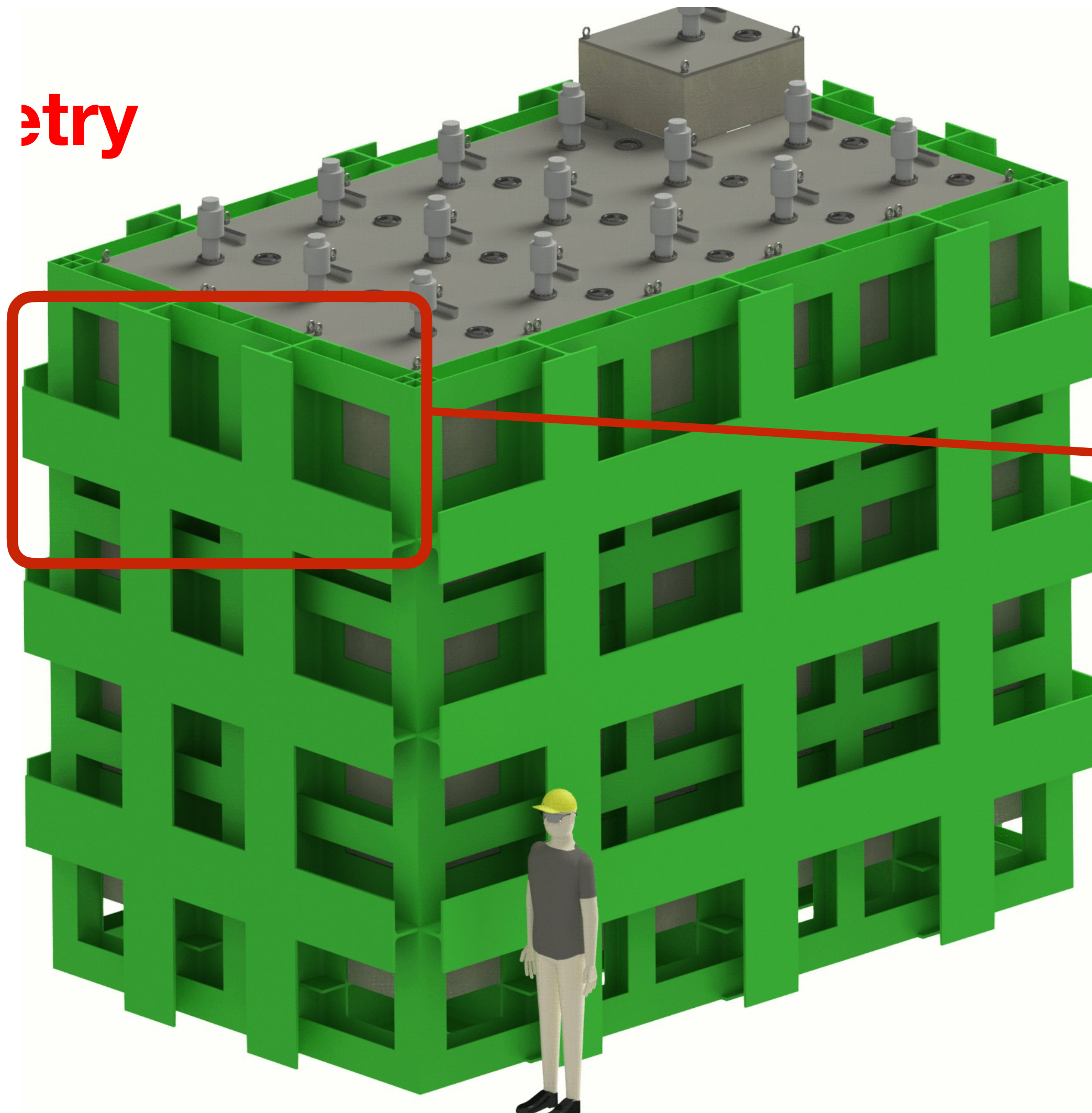
3 Modules: Transversal Beam Direction:
5 Modules: Beam Direction

ArgonCube Structure

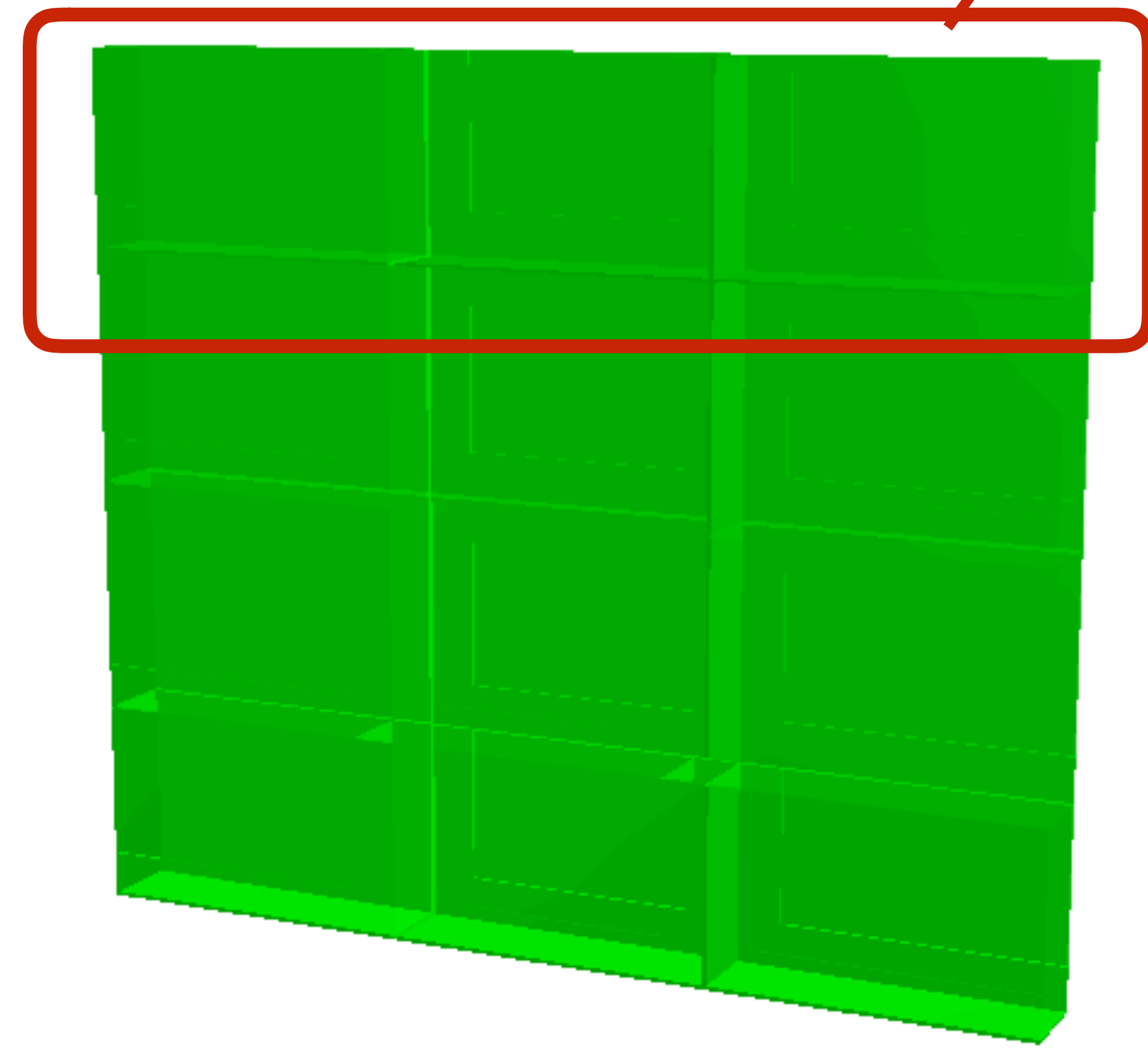


- ❖ Seems to be composed by two frames
- ❖ It is important to have realistic geometry for muon acceptance
- ❖ No trivial geometry shapes.
- ❖ The drawing is not scale

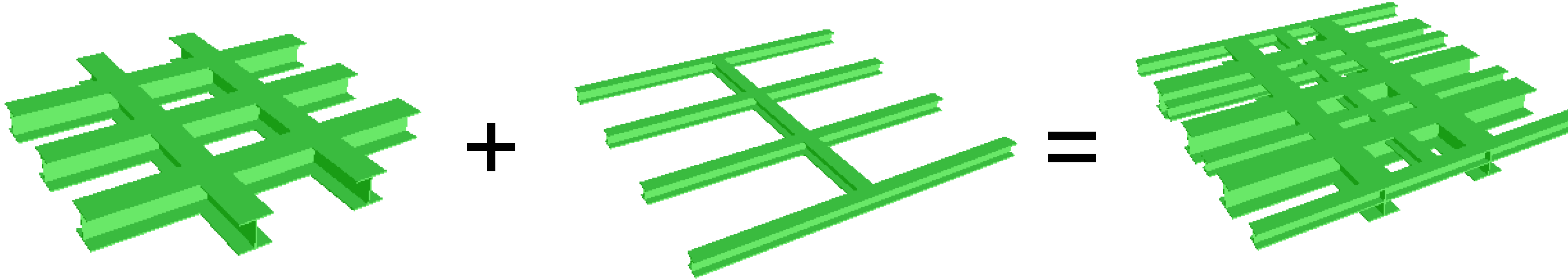
ArgonCube Structure



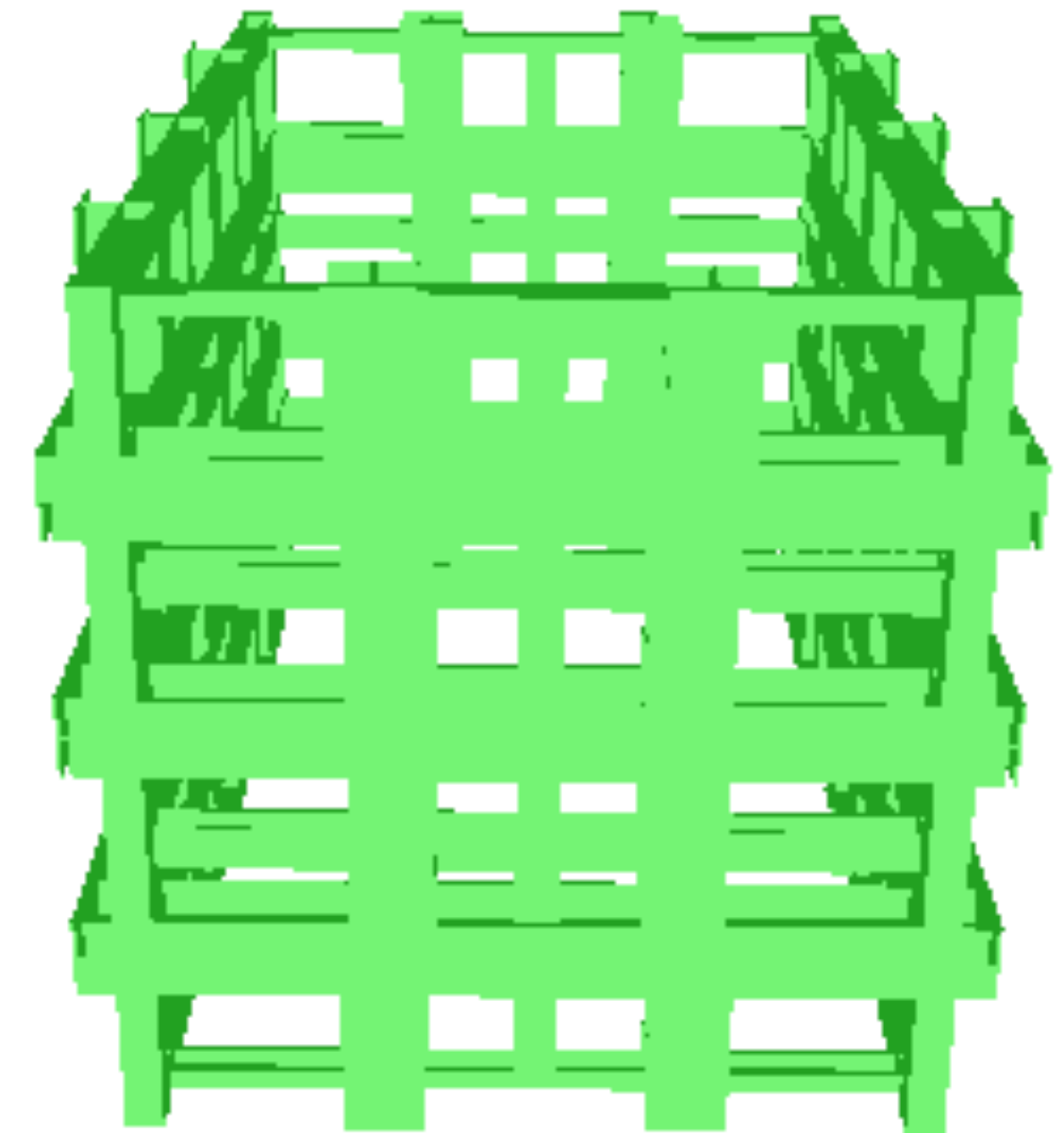
We have created Boolean shapes



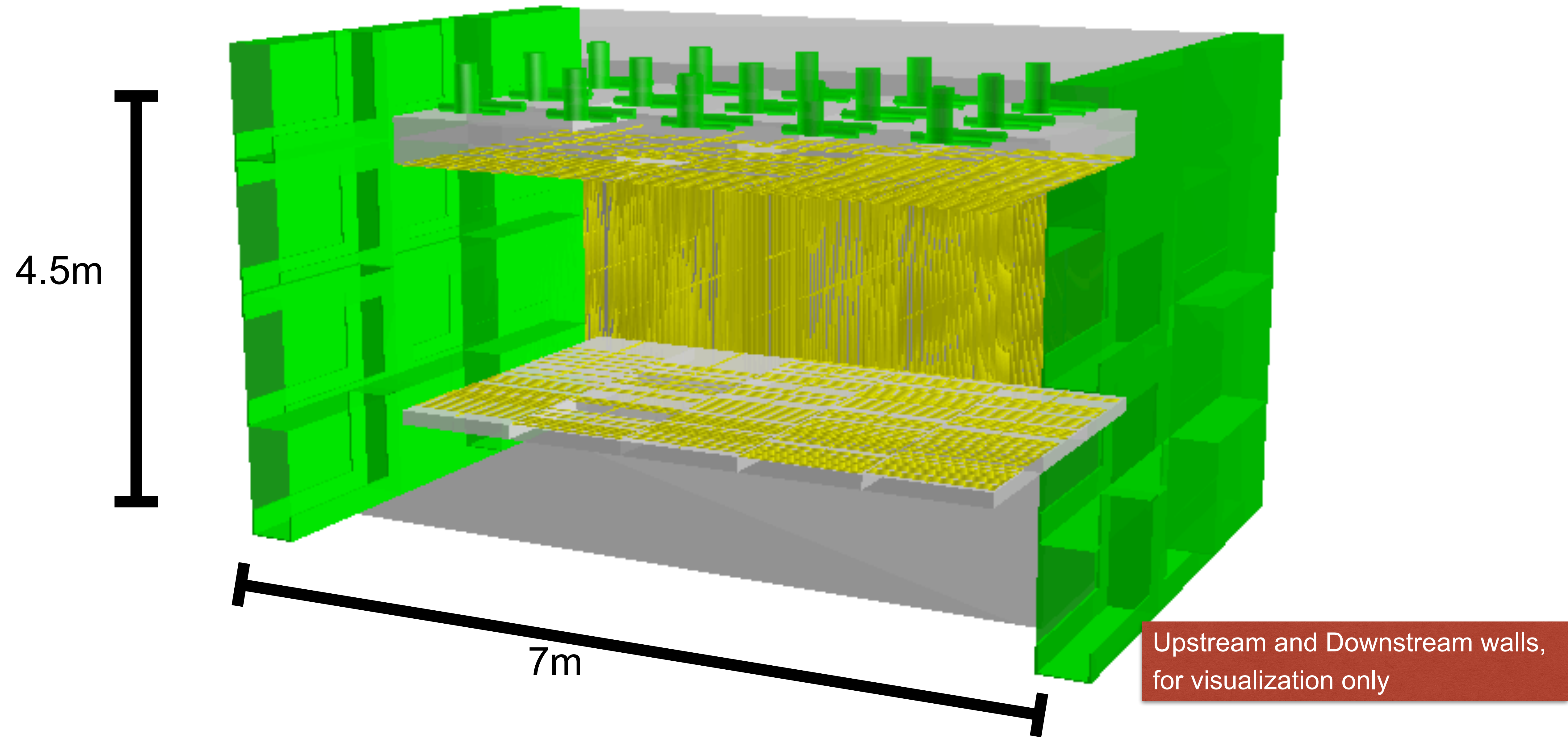
Two planes to create 1 wall



5 walls to create the Structure



ArgonCube: Status



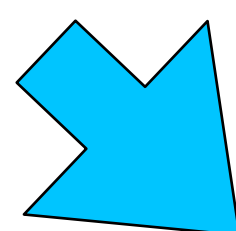
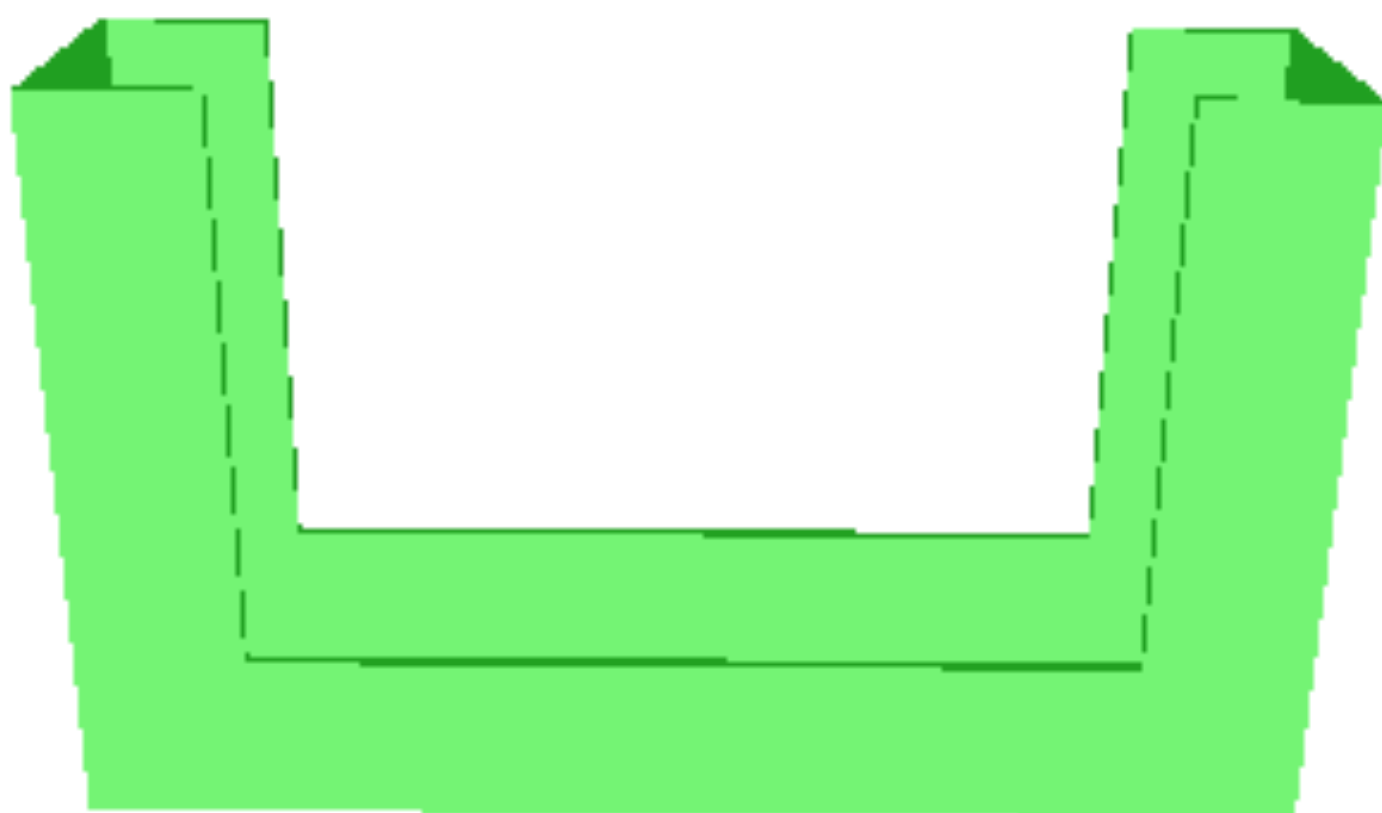
Backup Slides

EDEP-SIM

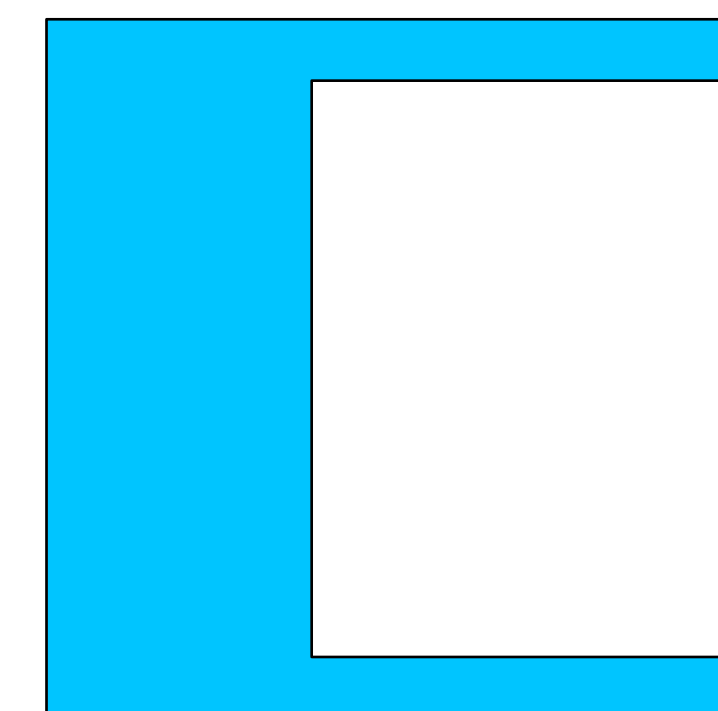
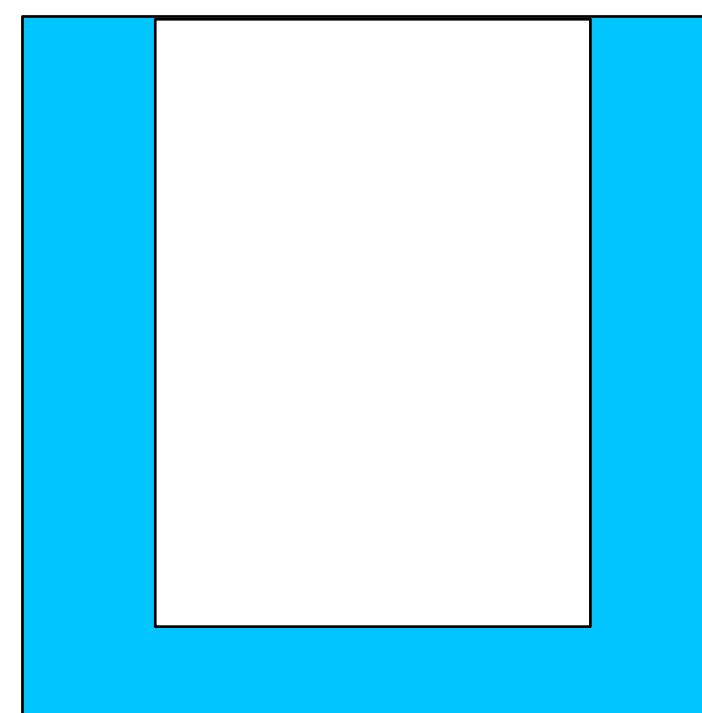
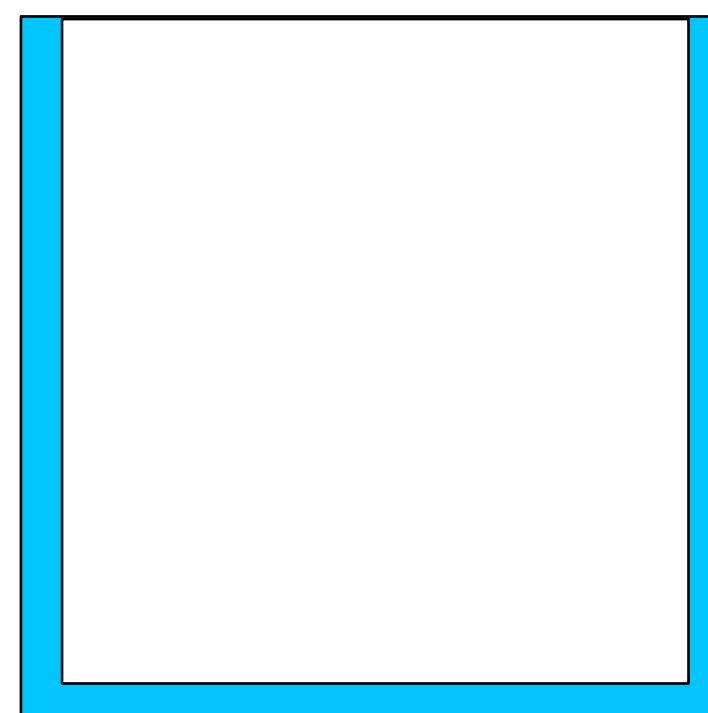
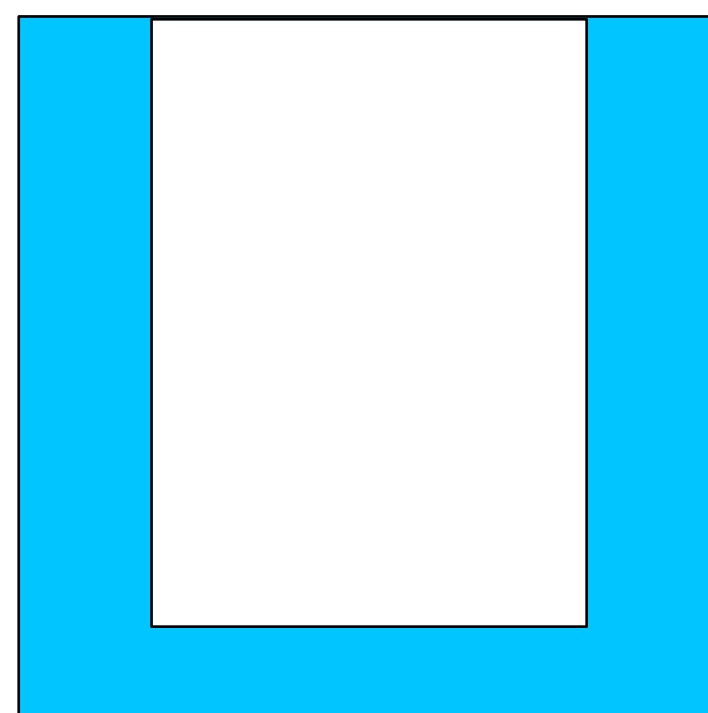
- Experiment independent Energy DEPosition SIMulation
 - ➔ Derived from the T2K near detector simulation
 - ➔ Provides the bookkeeping and infrastructure needed to track truth information and energy deposition.
 - In T2K, the output then drives a response/digitization simulation.
 - ➔ Can be called as a library, or to used to directly write a ROOT tree
 - ➔ Being used to simulate/debug the DUNE-ND-GGD geometries
- Detailed simulation
 - ➔ Electric and magnetic fields (from GDML)
 - ➔ Can simulate full beam structure, upstream and magnet interactions.
 - ➔ Detailed model for LAr recombination using NEST[†]
 - Handles both ionization and optical photon production
 - Validated by CAPTAIN collaboration against published ICARUS ionization measurements
- Major Features
 - ➔ Minimal dependencies (only ROOT and GEANT4 via cmake)
 - ➔ ROOT tree format designed to make analysis easy (more in some other meeting).
 - ➔ Provides a simple ROOT (Eve) based event display
 - ➔ Fast (can simulate 10's of GeV per second)
 - ➔ Reads interactions from GENIE, NEUT, NUANCE (easily expanded)
 - ➔ Scalable: Users can start with simple geometry, but edep-sim already handles the complexity needed for a running experiment.
 - ➔ Mature code. Except for cosmetic changes, it's been in used for a long time and has been thoroughly exercised.
 - ➔ Produces geometry that's ready for GENIE

[†]Enhancement of NEST capabilities for simulating low-energy recoils in liquid xenon, M Szydagis, A Fyhrie, D Thorngren and M Tripathi, Journal of Instrumentation, Volume 8, (2013)

Procedure



Final Boolean shape contains
3 boolean shapes like

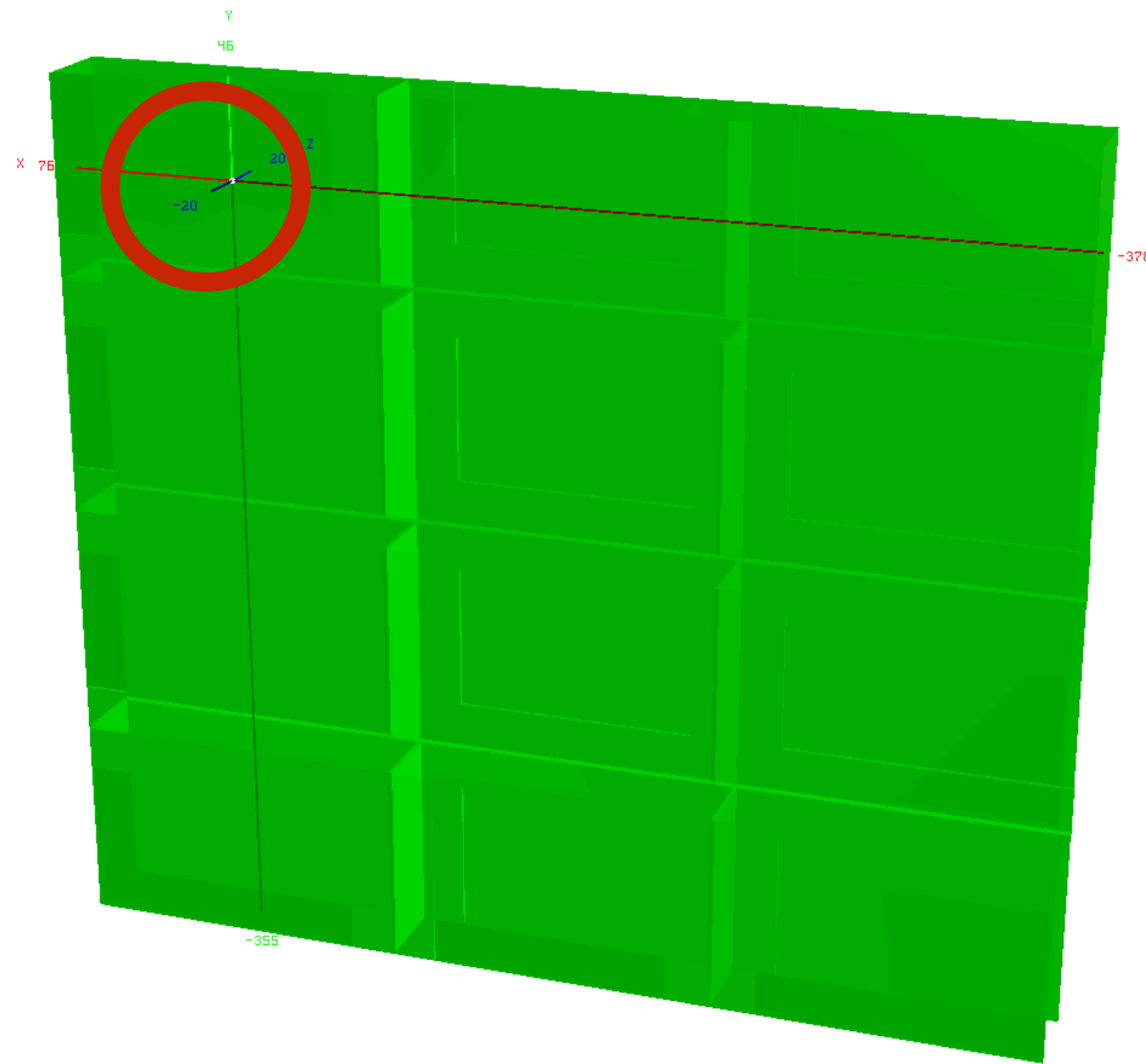


Every boolean contains 2 boxes

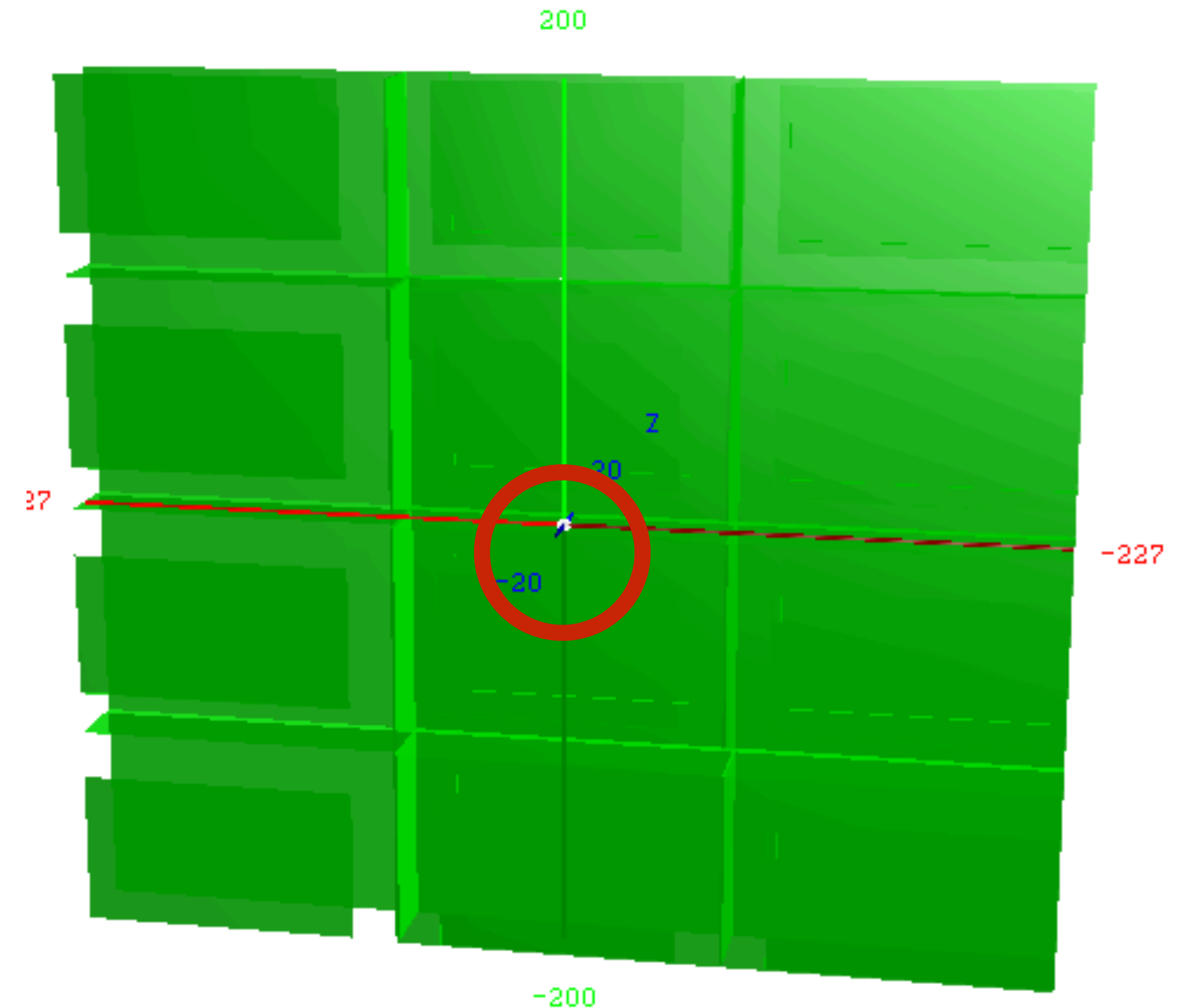
Boolean operations needed:

- ◆ union
- ◆ subtraction
- ◆ intersection

Operation with Booleans



The boolean shape operations,
always keep its center respect to
the first shape



I must create a empty bigger box
as first shape in order to apply
shape operations