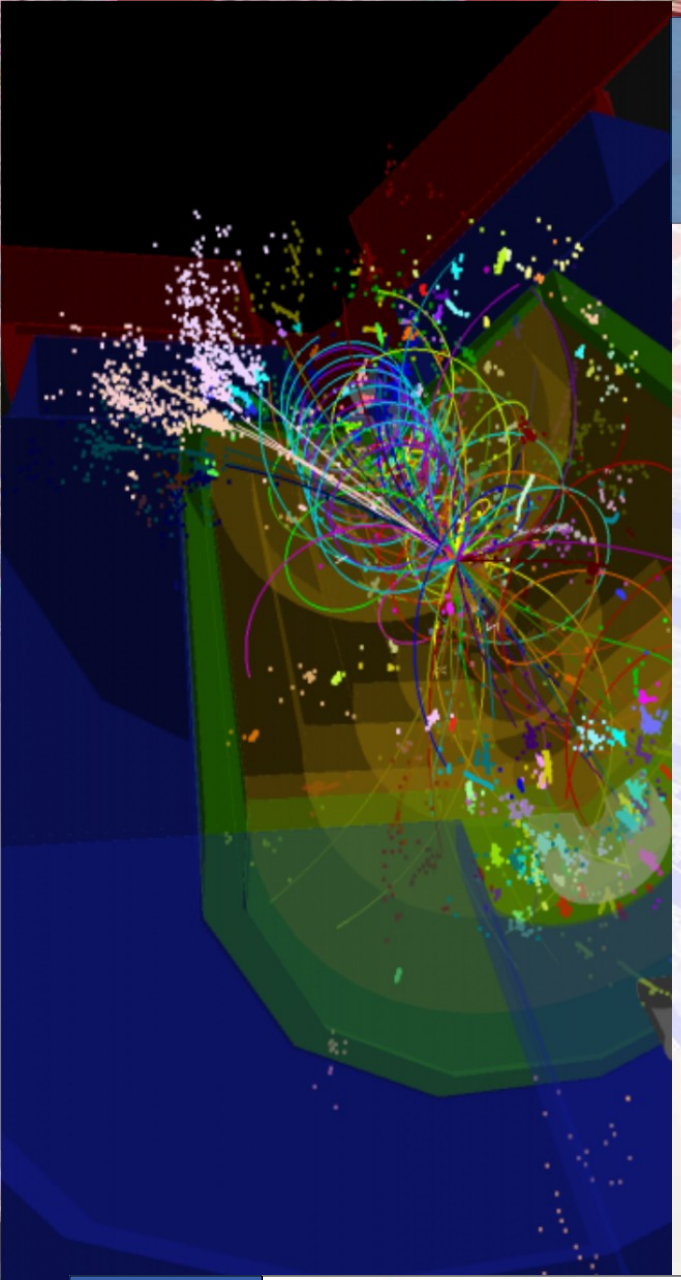


DD4hep

HEP detector description supporting the full experiment life cycle

Status Report

M.Frank, F.Gaede, M.Petric, A.Sailer

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

AIDA 2020

Motivation and Goal

DD4hep

Can I skip the intro ?
It is every time the same...

- **Develop a detector description**
  - **For the full experiment life cycle**
    - detector concept development, optimization
    - detector construction and operation
    - “Anticipate the unforeseen”
  - **Consistent description, single source, supporting**
    - simulation, reconstruction, analysis
  - **Full description, including**
    - Geometry, readout, alignment, calibration etc.

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

AIDA 2020

Philosophy of DD4hep & Co

DD4hep

- **Effort of very few people with a simple, humble and comprehensive vision**

  Detector description for the lazy
  Minimal effort, pragmatic, no technical restrictions,
  No religious wars

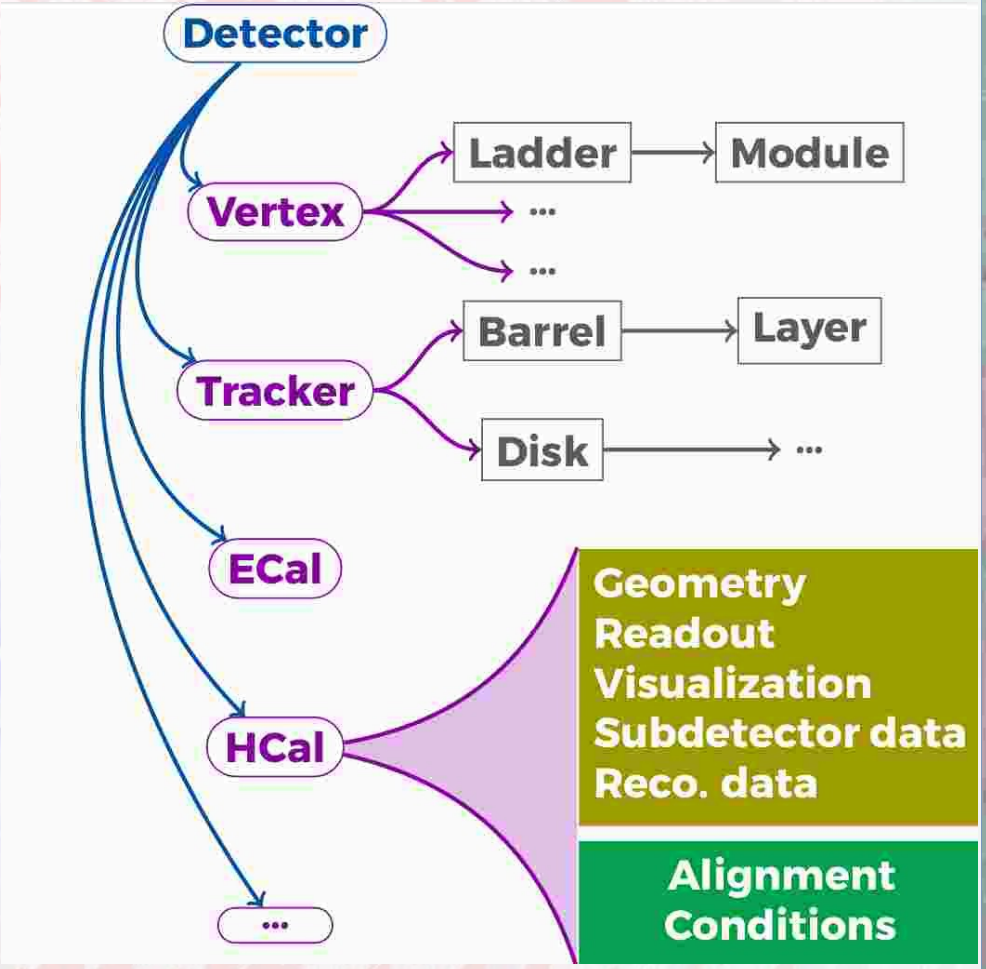
- **DD4hep is the "glue"**
  - **Bring together what belongs together:**

    Detector structure, geometry, simulation, conditions, etc
  - **Reuse existing modules: TGeo, Geant4, GitCondDB, etc**
- **'Responsible' users highly welcome**
- **Contributions even more !**

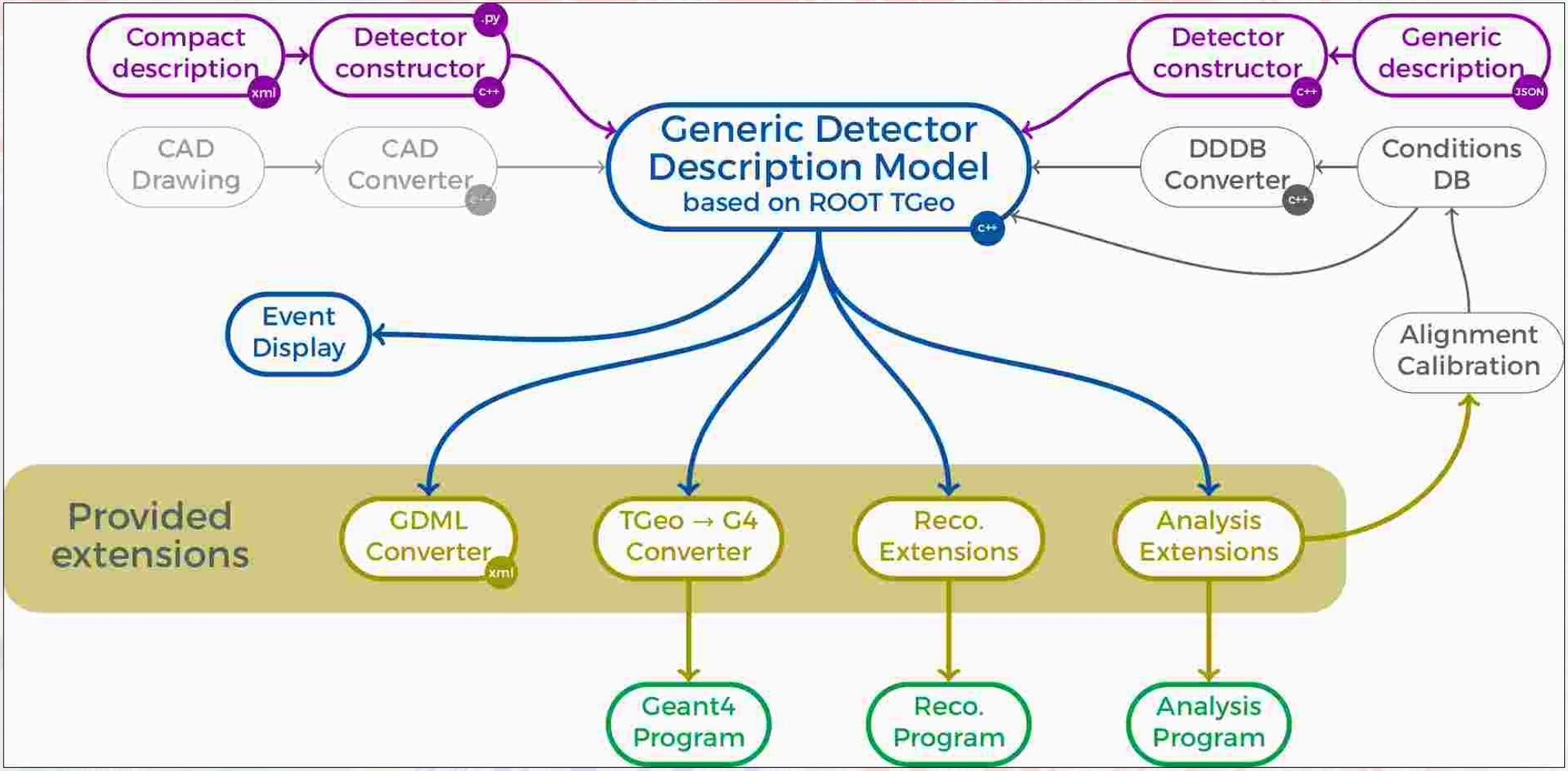
What is Detector Description ?

DD4hep

- Tree-like hierarchy of "detector elements"
  - Macroscopic (ie. not a strip)
  - Subdetectors or parts of subdetectors
- Detector Element
  - Geometry
  - Properties to process events
    - Environmental data
    - Alignments
    - Derivatives of these
  - Optionally experiment, sub-detector or activity specific data



DD4Hep - The Big Picture

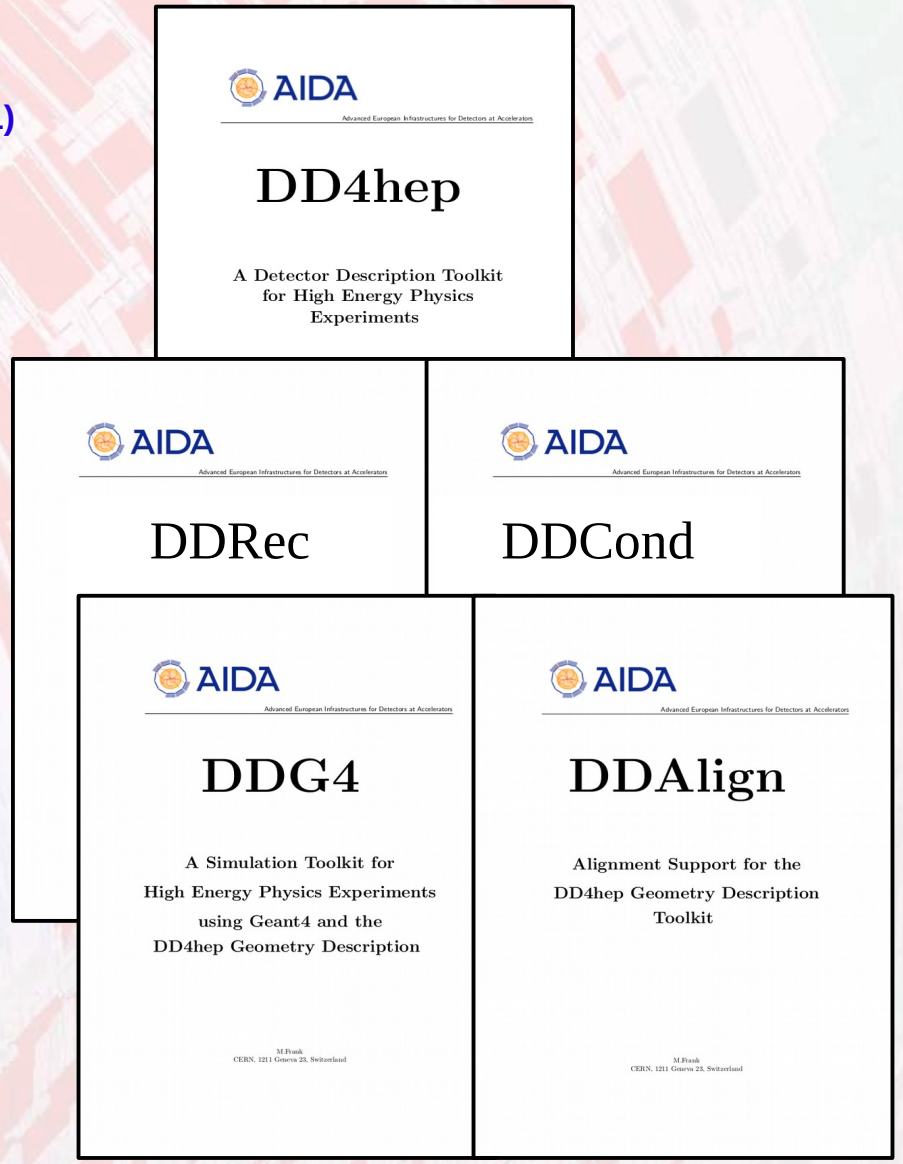


Saga in 6 Episodes

DD4hep

- DD4hep – basics/core (1)
- DDG4 – Simulation using Geant4 (1)
  - Fast simulation (4)
- DDRec – Reconstruction supp.(2)
- DDCond – Detector conditions (3)
- DDAlign – Alignment support (3)
- DDDigi – Generic Digitization (4)

(1) Mature state: bug-fixes and maintenance
(2) F. Gaede (WP3, Task 3.6)
(3) Work since start of AIDA2020
(4) Planned extensions



Toolkit Users

DD4hep

Increasing interest in the HEP community

- **ILC** F. Gaede et al.
- **CLICdp** A. Sailer et al.
- **SiD** D. Protopopescu et al.
- **FCC-eh** P. Kostka et al.
- **FCC-hh** A. Salzburger et al.
- **FCC-ee** O. Viazlo (CLD design), N. Alipour, G. Voutsinas
- **SCTF** Super-Charm-Tau Factory designs (Novosibirsk, Bejing)
- **EIC** Evaluation considered/started (W. Armstrong et al.)
- **CEPC** Used for design studies (W. Li et al., IHEP)

- **LHCb** LHCb Upgrade for Run III (B.Couturier et al.)
- **CMS** Usage for upgrade - CHEP2020 (Y.Osborne et al.)
- **CALICE** Calorimeter R&D, started

Reading CAD files: Finally!

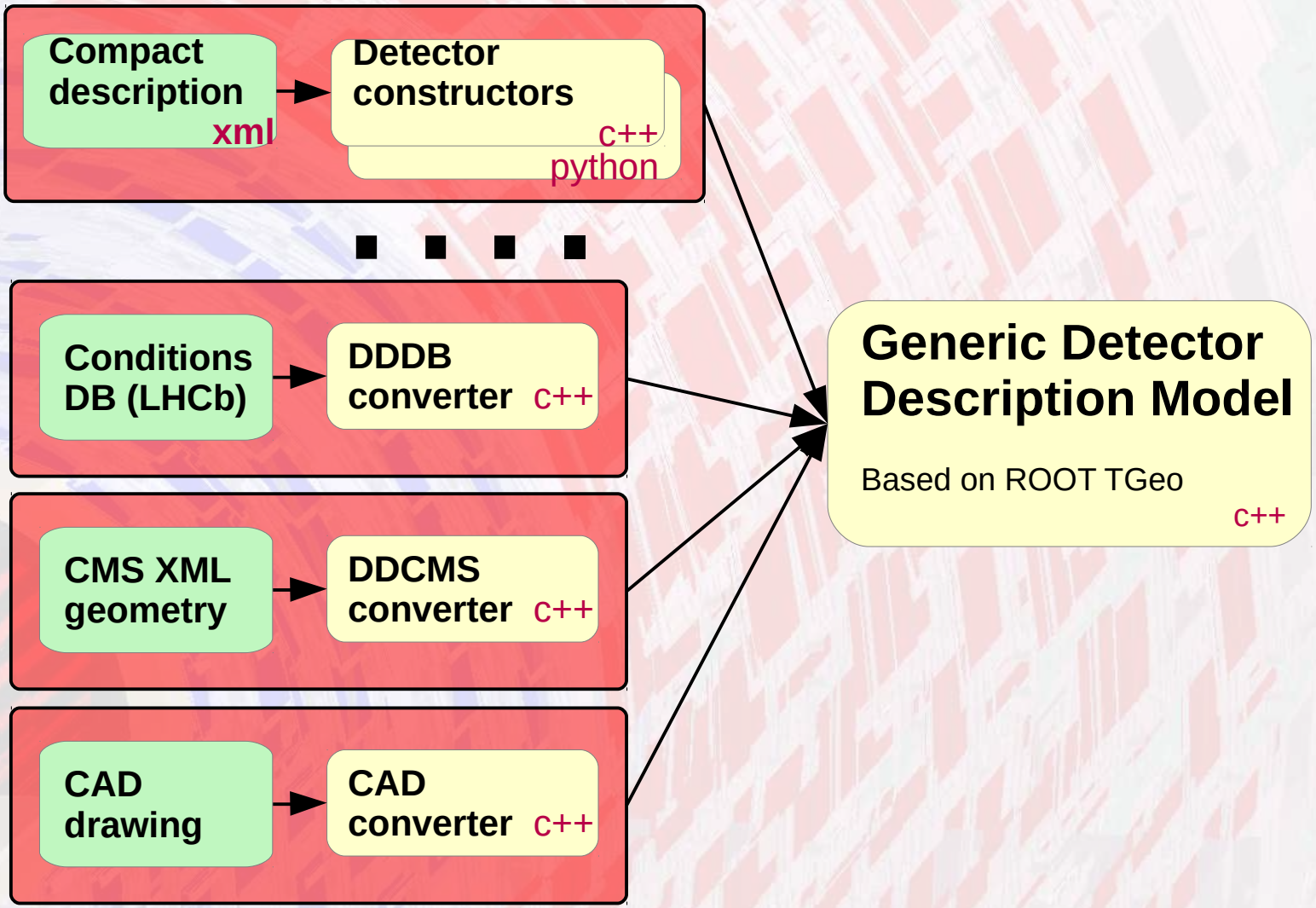
DD4hep

(Demonstrated)

(Demonstrated)

(Demonstrated)

Here we are!



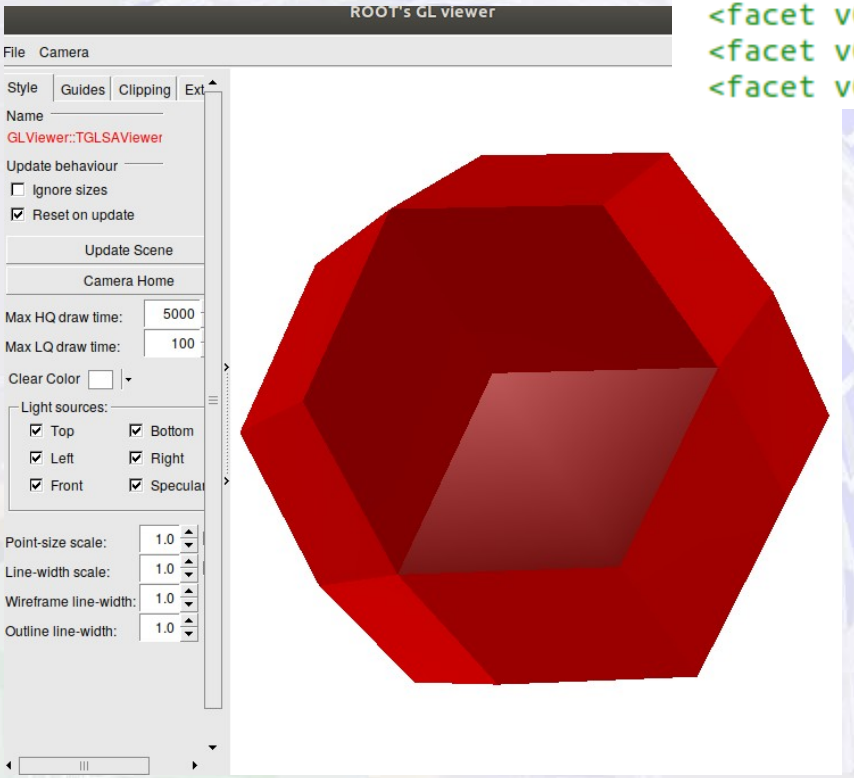
How It Begun: The Story

DD4hep

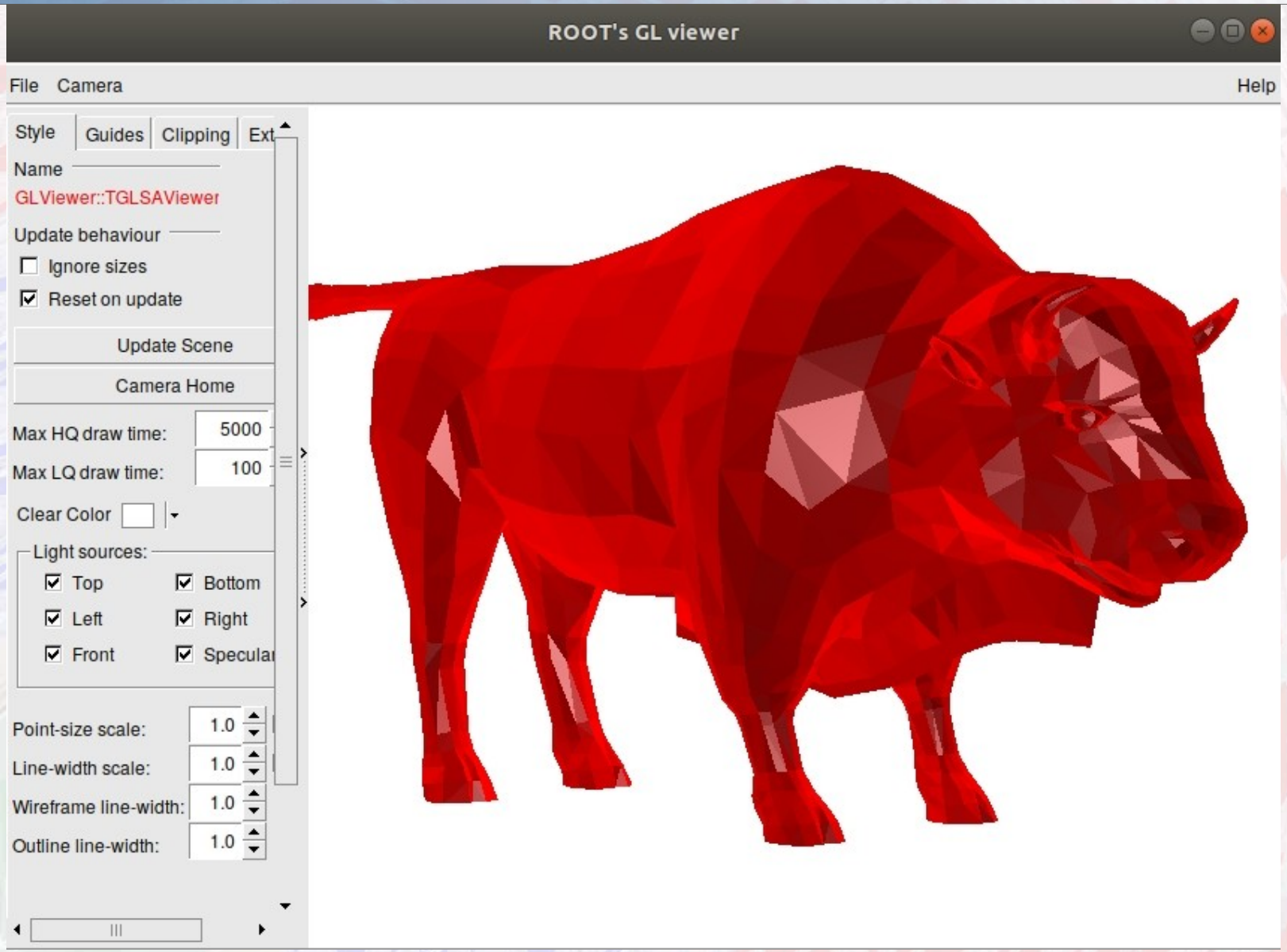
- We have now tessellated shapes in TGeo thanks to Andrei Gheata: TgeoTessellated
  - Shapes consist of sets of facets
  - CAD tools model shapes as tessellated shapes
  - Simply connect the dots...
- Libassimp: Open Asset Import Library
  - Creates standardized meshes from multiple CAD input formats
- Long outstanding dream of many physicists
  - Important to model complicated passive structures

Simple Tessellated Shape

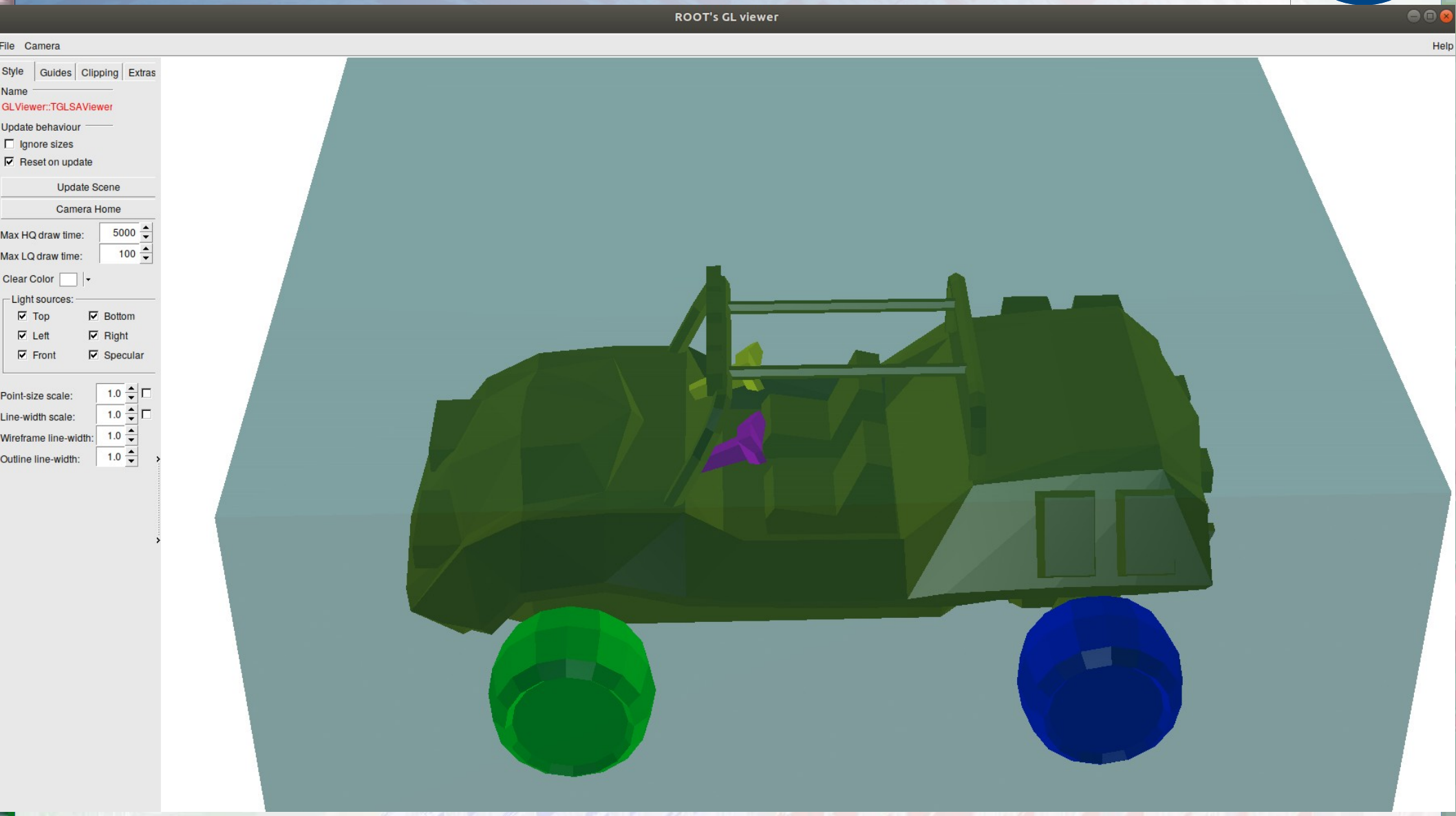
```
<vertex x="1" y="-1" z="-1"/>
<vertex x="0.5 * (1 + sqrt5)" y="-1" z="0"/>
<vertex x="0.5 * (-1 + sqrt5)" y="0.5 * (-1 - sqrt5)" z="0"/>
<vertex x="0" y="0.5 * (-1 - sqrt5)" z="-1"/>

<vertex x="0.5 * (1 + sqrt5)" y="-1" z="0"/>
<vertex x="1" y="-1" z="1"/>
<vertex x="0" y="0.5 * (-1 - sqrt5)" z="1"/>
<vertex x="0.5 * (-1 + sqrt5)" y="0.5 * (-1 - sqrt5)" z="0"/>

<facet v0="0"  v1="1"  v2="2"  v3="3"/>
<facet v0="4"  v1="7"  v2="6"  v3="5"/>
<facet v0="8"  v1="9"  v2="10" v3="11"/>
<facet v0="12" v1="15" v2="14" v3="13"/>
<facet v0="16" v1="17" v2="18" v3="19"/>
```

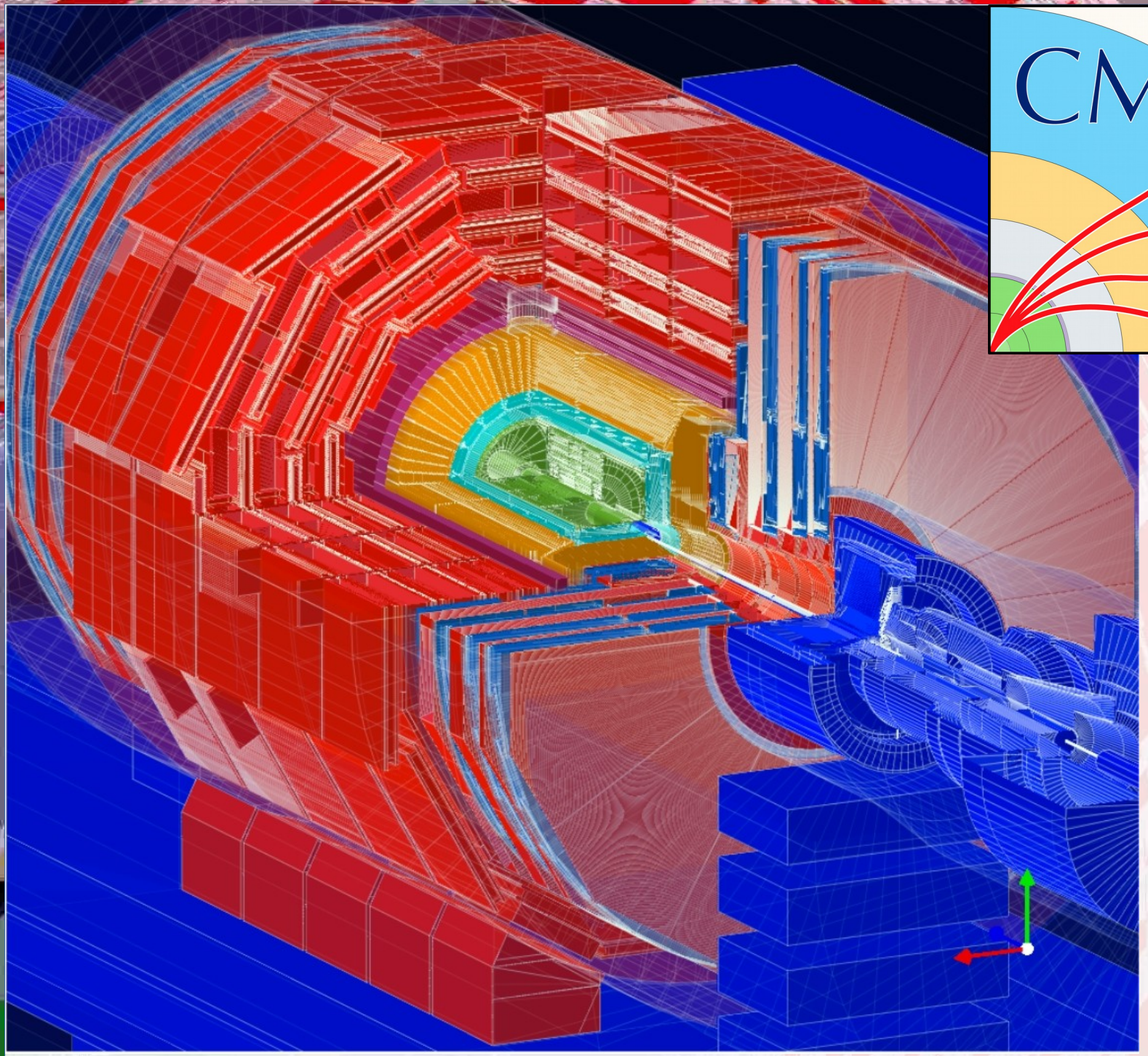


Single Mesh Example (PLY)



Multi Mesh Example (MS3D)





CMS described with DD4hep

CHEP 2019, Adelaide, AU

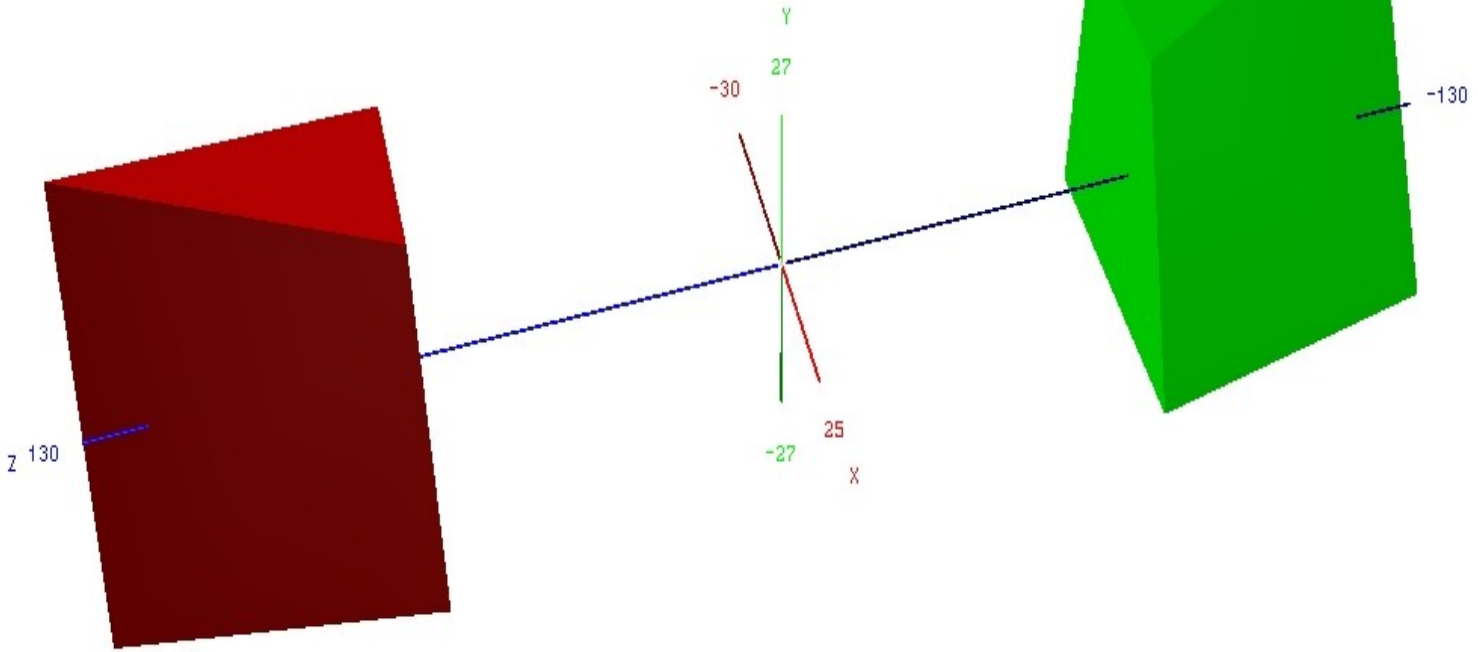
(C.Vuosalo / CMS)

DD4hep is CMS' Prime Choice

DD4hep

- Several developments were triggered by CMS
- Reflected volumes
  - Rather than rotate, reflect shapes/volumes
    => Left-handed coordinates
- CutTube
  - Tube segment cut with 2 planes.
- EightpointSolid
  - Arbitrary trapezoid with less than 8 vertices standing on two parallel planes perpendicular to Z axis
- One single units system: Geant4 units
  - DD4hep (and TGeo) can used Geant4 units rather than CGS
    One single unit system. Only for ROOT version >= 6.20

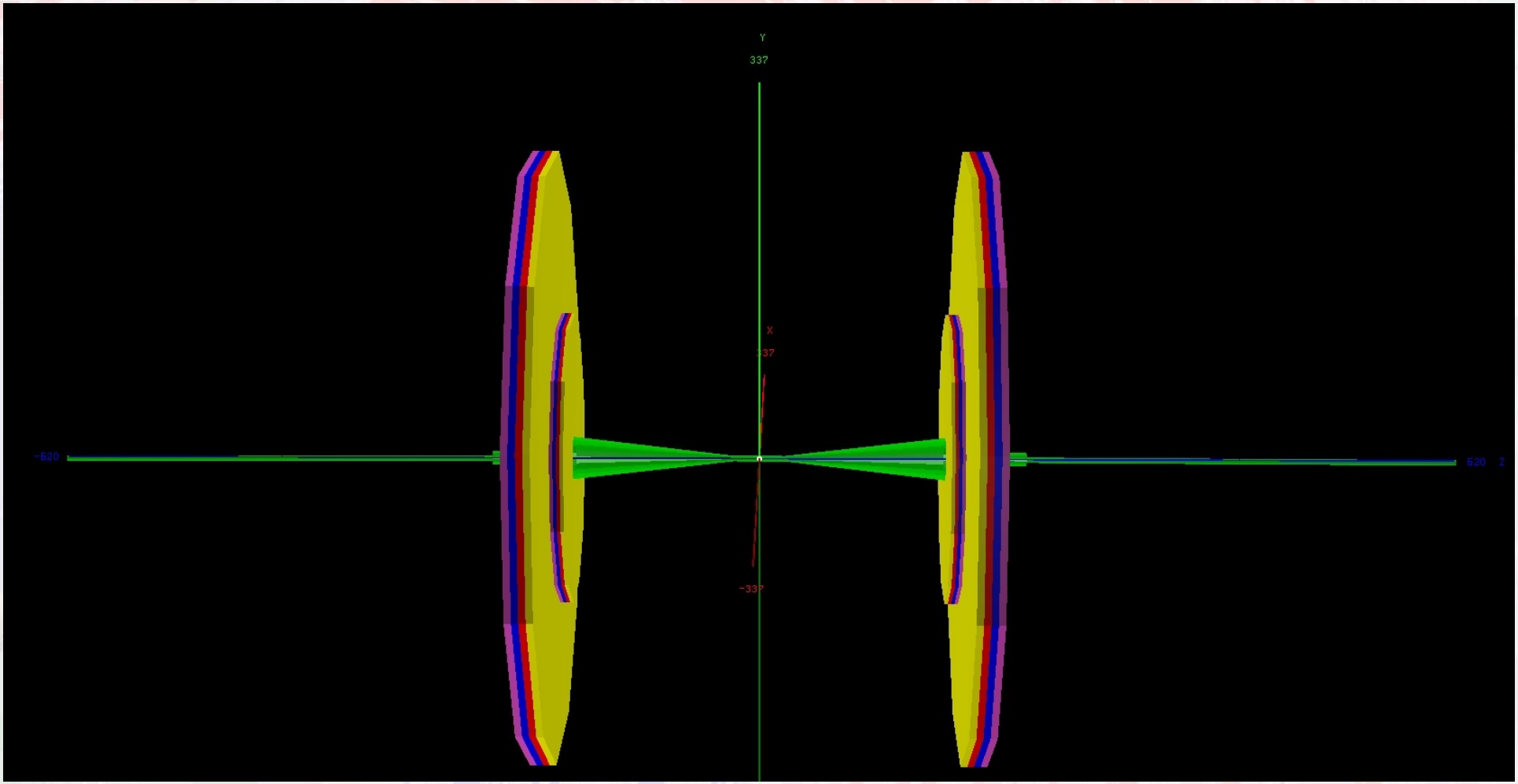
Reflected Volumes: TGeo



Original

Reflected

Reflected Detector Elements



Reflected Original

Requests from FCC

DD4hep

- Optical surfaces and material properties
- Energy dependent material properties important when interacting with light
  - True material properties (absorption, etc.)
  - Properties depending on material machining: surfaces
- Interface was implemented in ROOT
  - Design criteria following Geant4 design
- DD4hep interface was implemented
  - Extended compact xml model to accommodate data
  - Parameters are automatically applied to Geant4
- Client changed job: Still waiting for extensive tests

The Future: Digitization

DD4hep

- **Digitization:**
  Translation of simulated energy deposits into “signals”
- **Use the opportunity to peek a bit into future developments**
- **Some initial thoughts**

Digitization: Required Functionality

DD4hep

- Handle collision overlays
- Handle event spillover
  - Process detector response from collisions earlier or later to the central event
- Noise handling
  - Random noise hits
  - Hot/dead channel emulation
  - Add noise to all channels with cut-off
- List probably not exhaustive

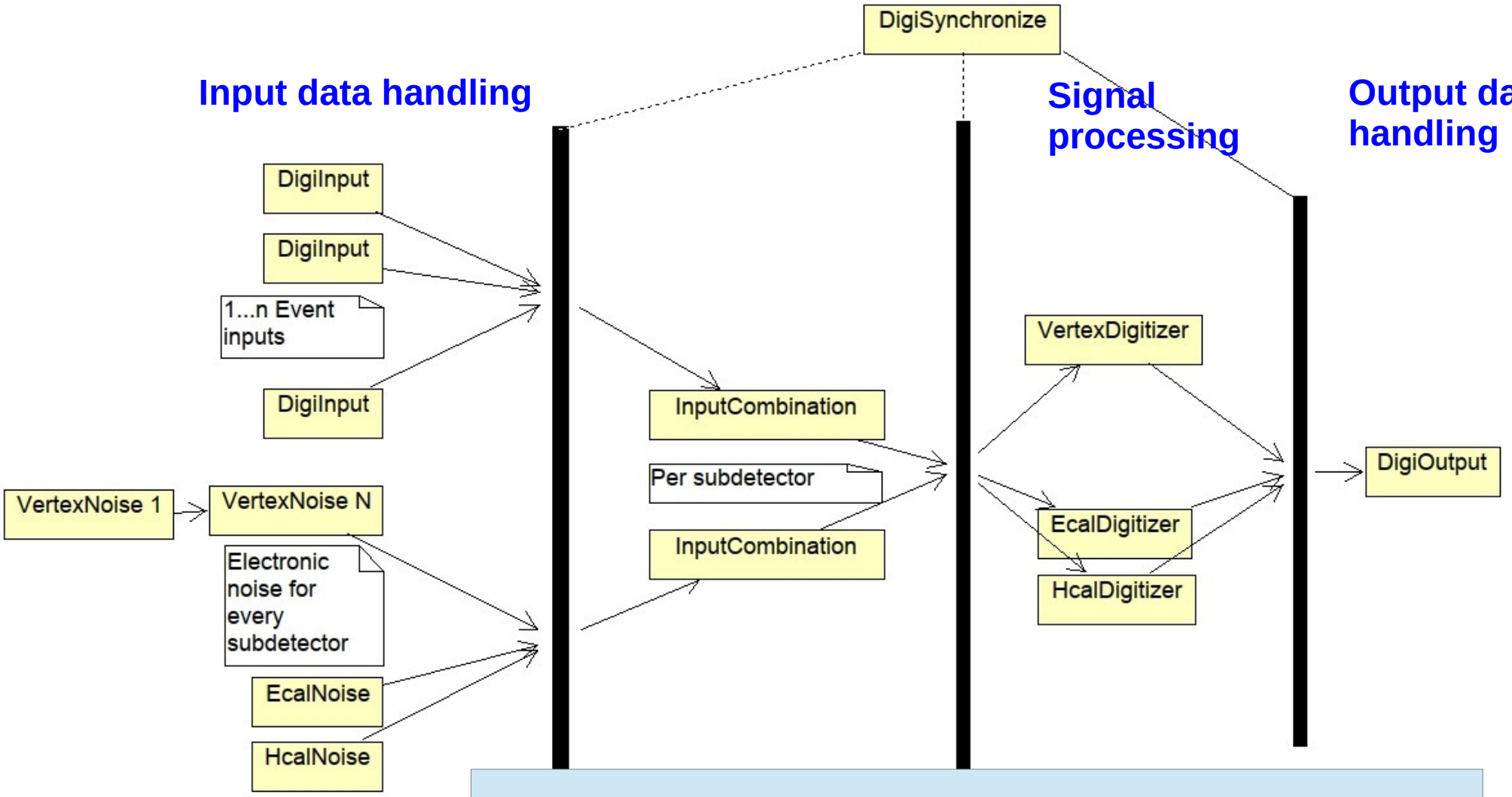
Digitization: Desirables

DD4hep

- **Efficiently use CPU**
  - **Multi-threading**
- **Turn key system:**
  - **Input: DD4hep detector description**
  - **Configure digitization using existing components (similar to DDG4)**
  - **Probably configure using python**

Basic Schematic

DD4hep



Pretty obvious, but ….

You Quickly Realize

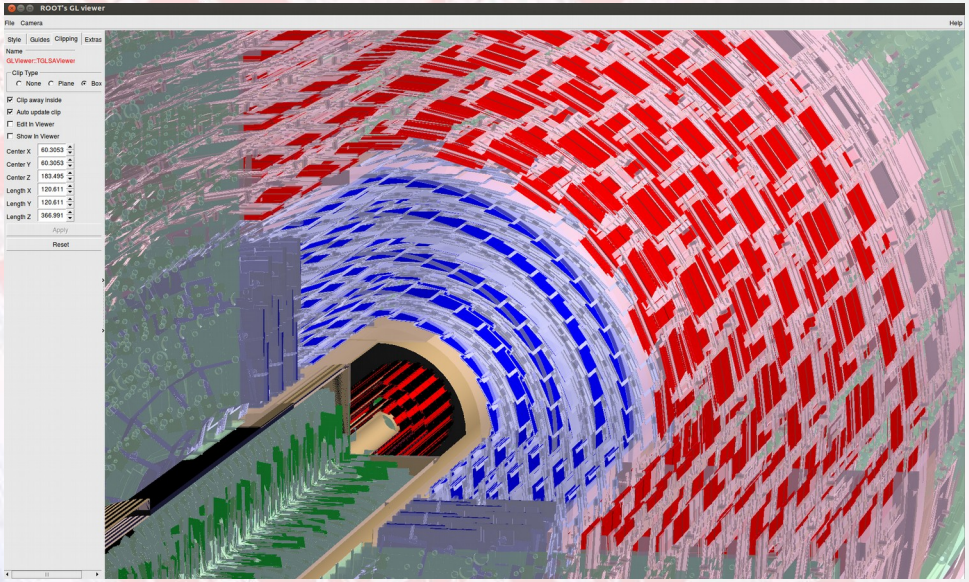
DD4hep

- Implementing this functionality in entirely independent modules would not work
  - Memory explodes
    Example: noise
    - Create noise per channel according to noise sources
    - Add up noise from various sources
    - Add signal (if present)
    - Apply cut with minimal signal strength
      (can only be done last!)
  - Would require multiple channel maps (per event !)
    to host all data: virtually impossible
- Brute force approach not feasible
- Need some more elaborate thinking

Handling Noise

DD4hep

- For scanning cells of sensitive volumes to add up noise
  - Handle one single cell at a time
    - Avoid anything using large channel maps
  - Can still execute in parallel
  - One thread deals with one layer, one ladder etc.

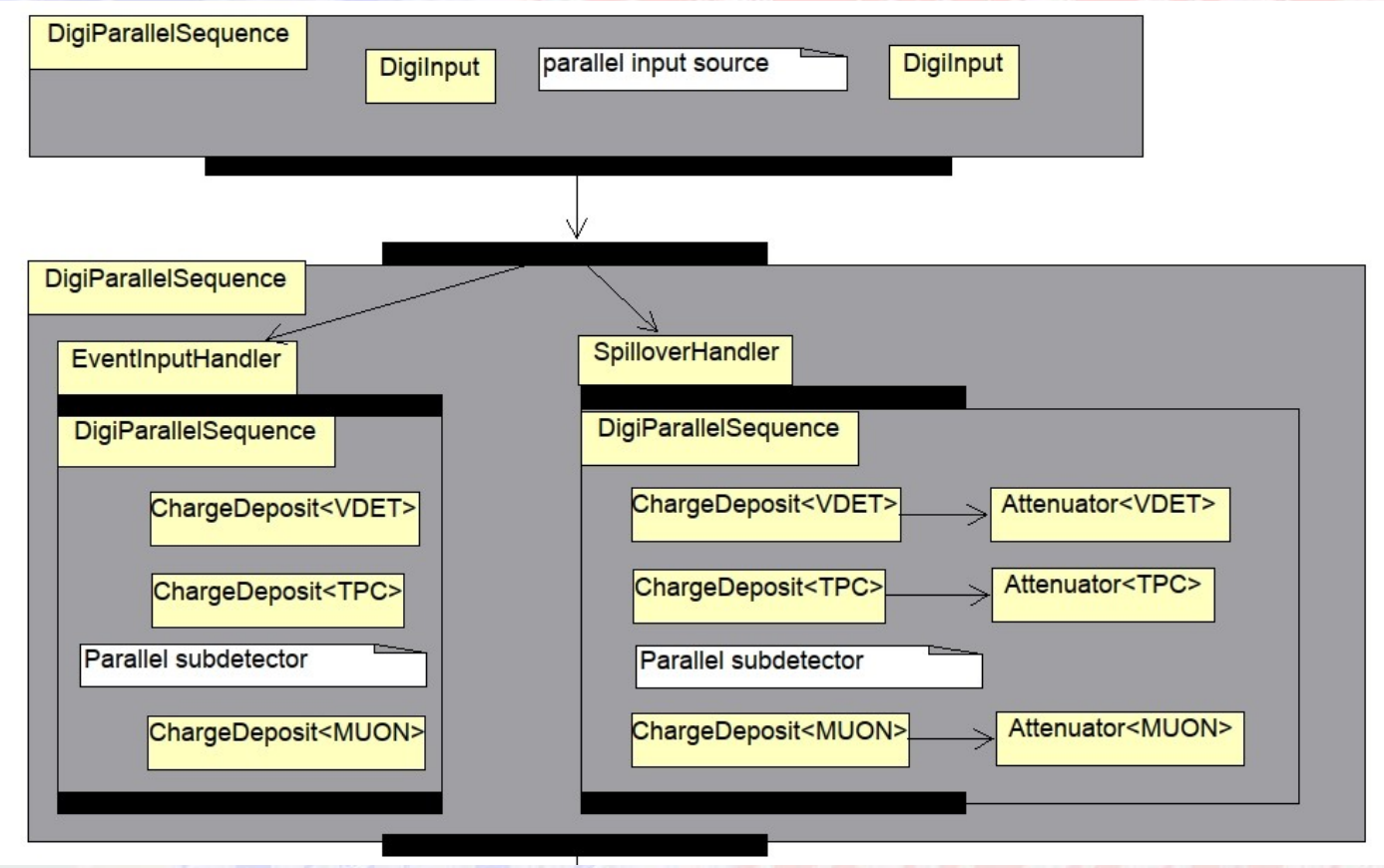


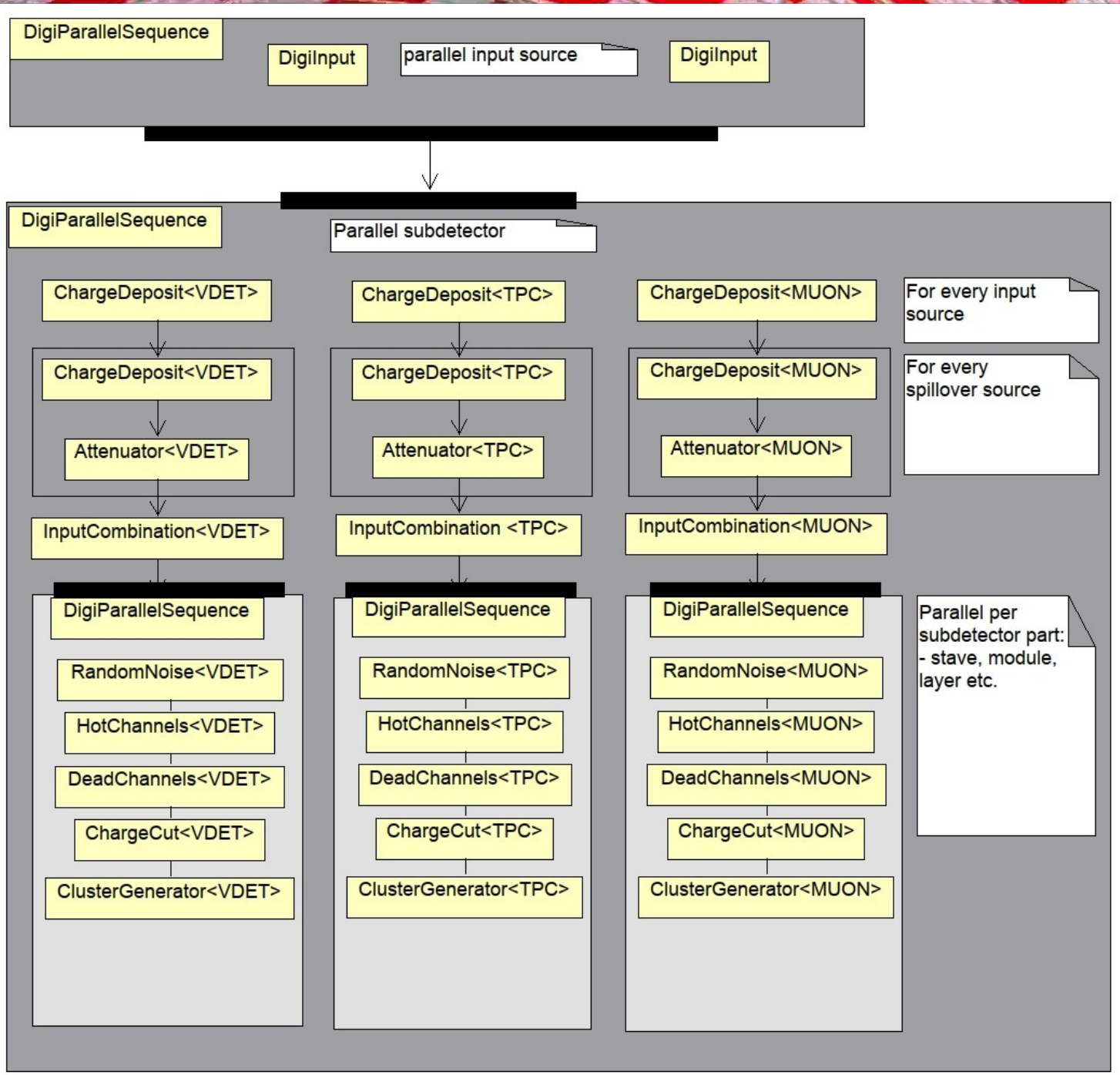
- Handling simulation input
  - No obvious reason to not handle in parallel on the file level
  - Both: event overlay and spill-over

Handling Event Input from Simulation

DD4hep

- **No obvious reason to not handle in parallel input on the file level**
  - **Both: event overlay and spill-over**





DD4hep

Black bar:

Synchronization Point

All threads in the block must finish before continuation

Conclusions

DD4hep

- Got numerous new clients
- CMS has demonstrated DD4hep to be able to describe complex existing detectors – with respect to planned experimental setups
- We have reacted to various client driven requests
- Tried to sketch up future developments