

# Recent Developments in the FullSimLight Simulation Tool from ATLAS

**Raees Khan** (on behalf of the ATLAS collaboration)



CHEP 2023



# FullSimLight

- FullSimLight brought to you by the ATLAS collaboration is an experiment independent tool used for lightweight full Geant4 simulation.
- FullSimLight reads in events **and geometries**.
- The goal of FullSimLight is to simplify studies of Geant4 tracking and physics processes, including tests on novel architectures as well as optimization studies.



# FullSimLight

Geometry →

Events →

Command Line  
or GUI

FullSimLight

GeoModel  
API

Geant 4 API



## Basic Scoring

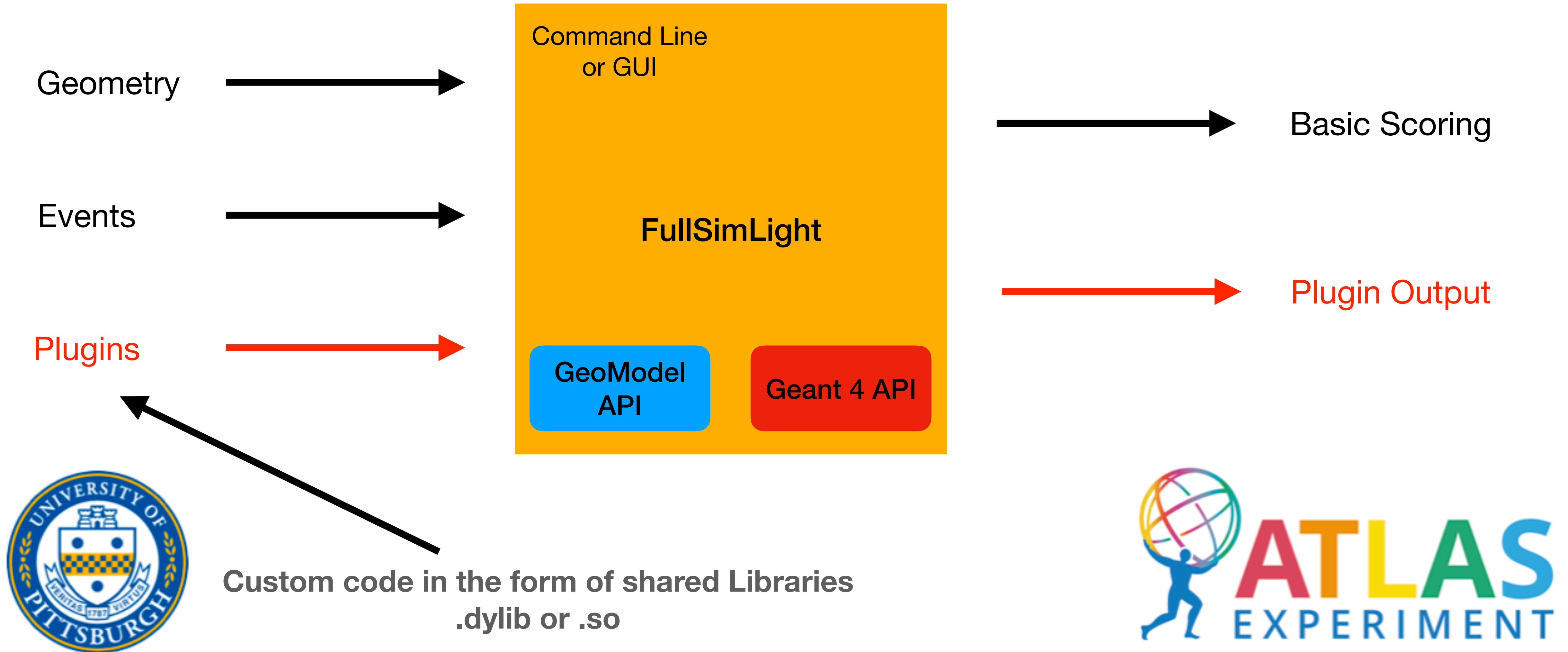
- (i) Mean energy deposition
- (ii) Mean track length
- (iii) CPU time taken



CHEP 2023



# Customized FullSimLight



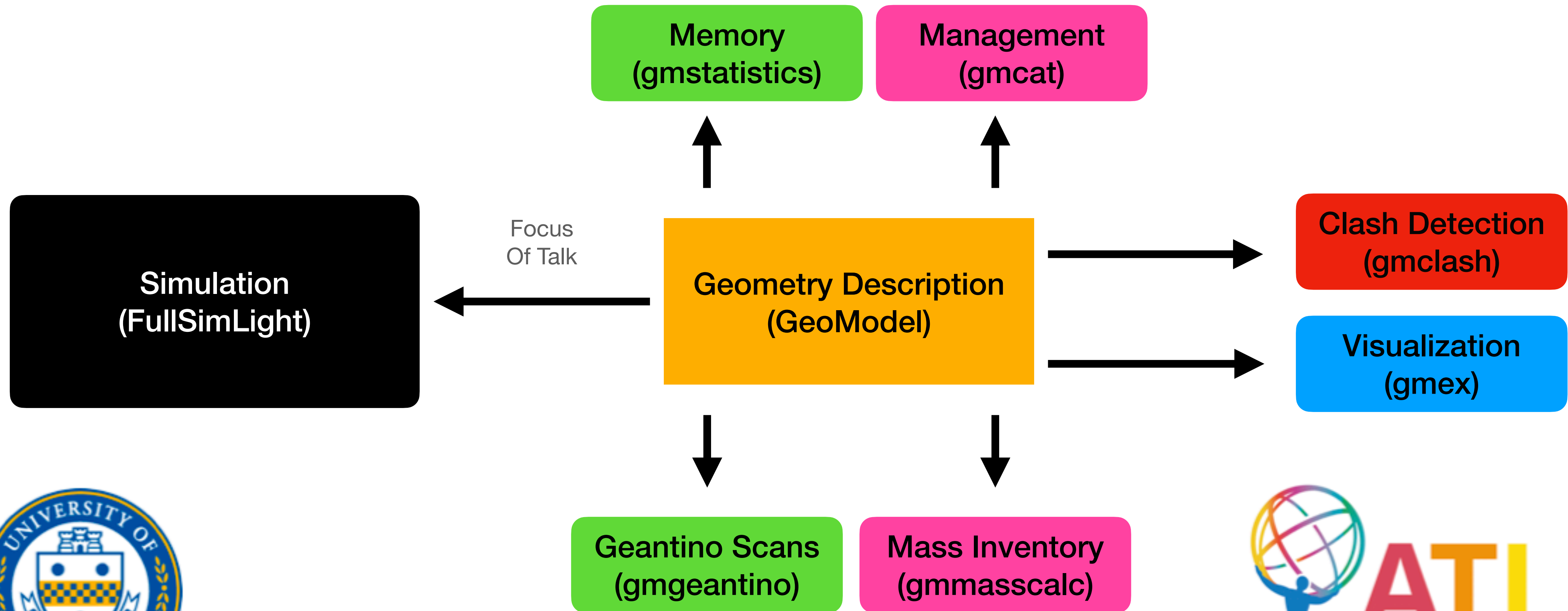


# Geant 4

- Geant4 is a toolkit to create simulations of the passage of particles or radiation through matter.
- Tracking of charged and neutral particles through geometry in a **magnetic field**.
- Decay of unstable particles and interaction with detector materials.
  - Customizable (**Physics Lists**).
- Customizable simulation of detector response (**Sensitive Detectors**).
- User hooks allowing one to intercept processing at various stages (**User Actions**).
  - **Run Actions**
  - **Event Actions**
  - **Tracking Actions**
  - **Stepping Actions**
  - **Stacking Actions**

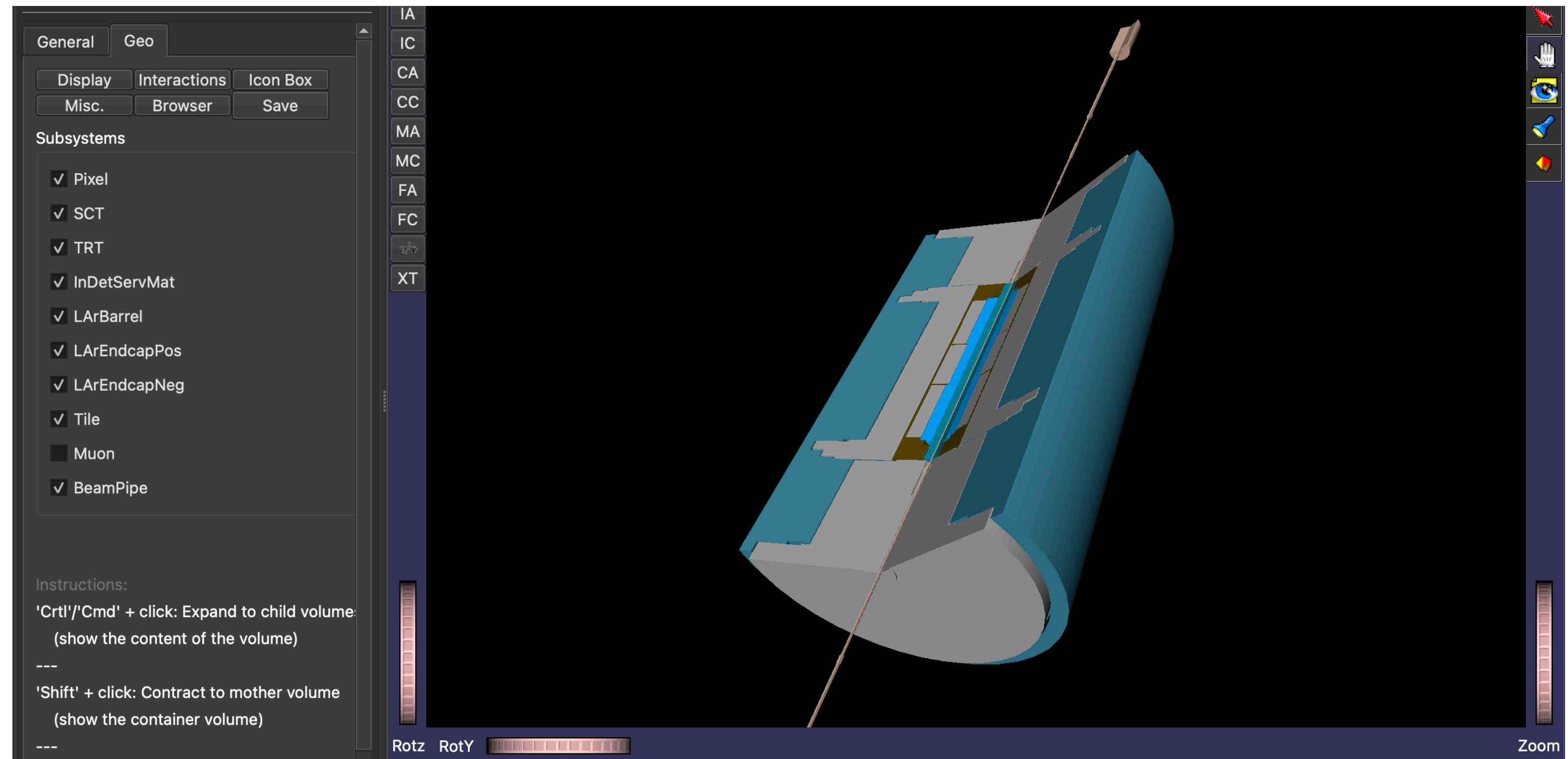


- The **GeoModel Toolkit** ([geomodel.web.cern.ch](http://geomodel.web.cern.ch)) offers classes that provide geometrical primitives for describing detectors, and a set of **command line tools** for accessing, handling, manipulating, dumping, restoring, visualizing, inspecting, and debugging the detector geometry.



# Gmex

- Gmex is an interactive 3d geometry visualization tool which is part of the GeoModel toolkit.
- Useful in developing a software description of a geometry.



ATLAS Detector visualized in gmex.



CHEP 2023

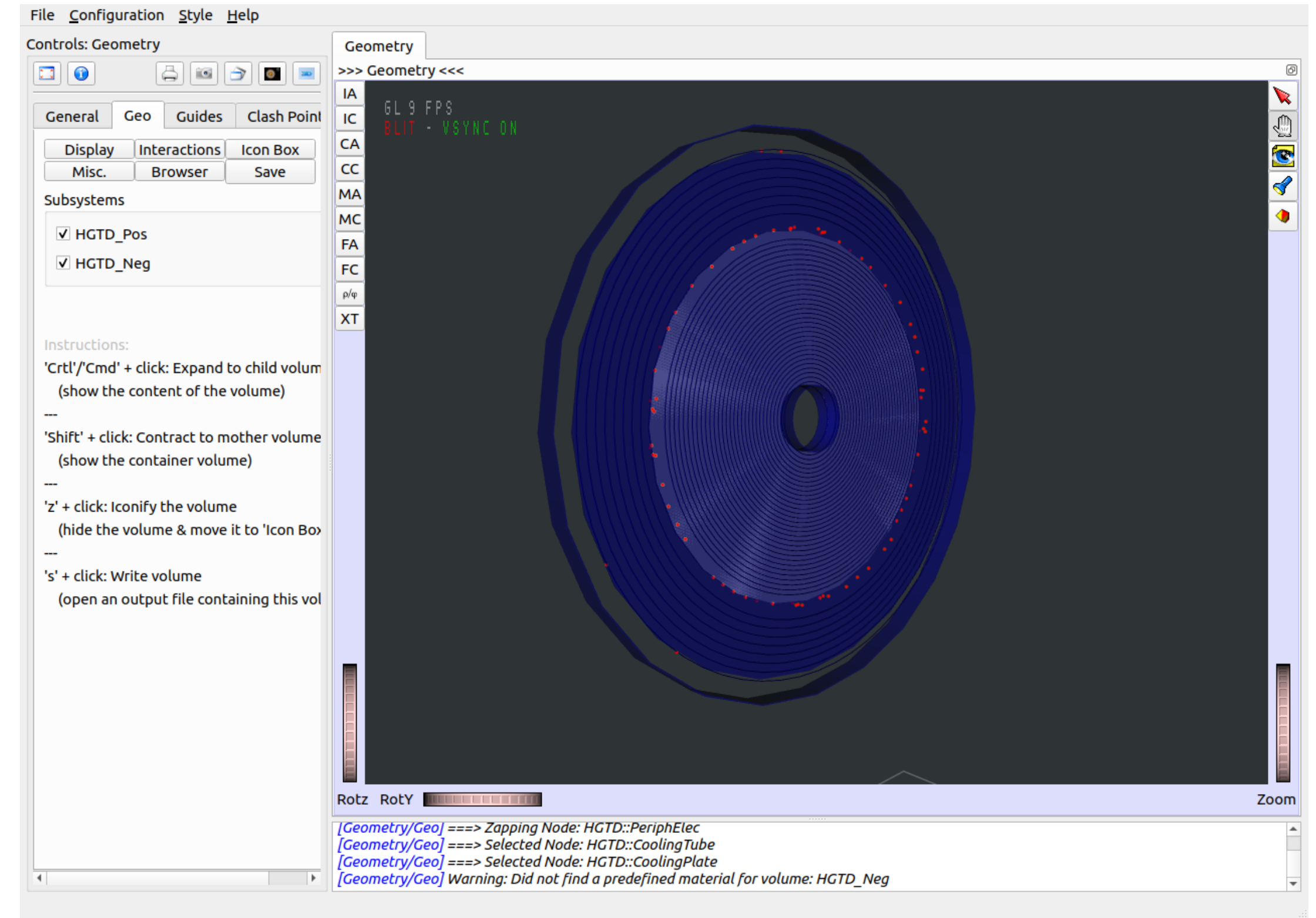




# Clash Detection

## Clash report

```
{
  "ClashesReport": [
    {
      "distance[mm]": 2.2229113814991024,
      "typeOfClash": 0,
      "volume1CopyNo": 16969,
      "volume1EntityType": "G4Box",
      "volume1Name":
"HGTDModule3_layer_8_8_12",
      "volume2CopyNo": -1897956176,
      "volume2EntityType": "G4Tubs",
      "volume2Name": "HGTD::ModuleLayer3",
      "x": 3.8780568483553015,
      "y": -20.659230748619382,
      "z": 3491.78
    },
    ....
  ]
}
```



Clash points visualized in gmex.



CHEP 2023





# Geometry Example (Kitchen Sink)

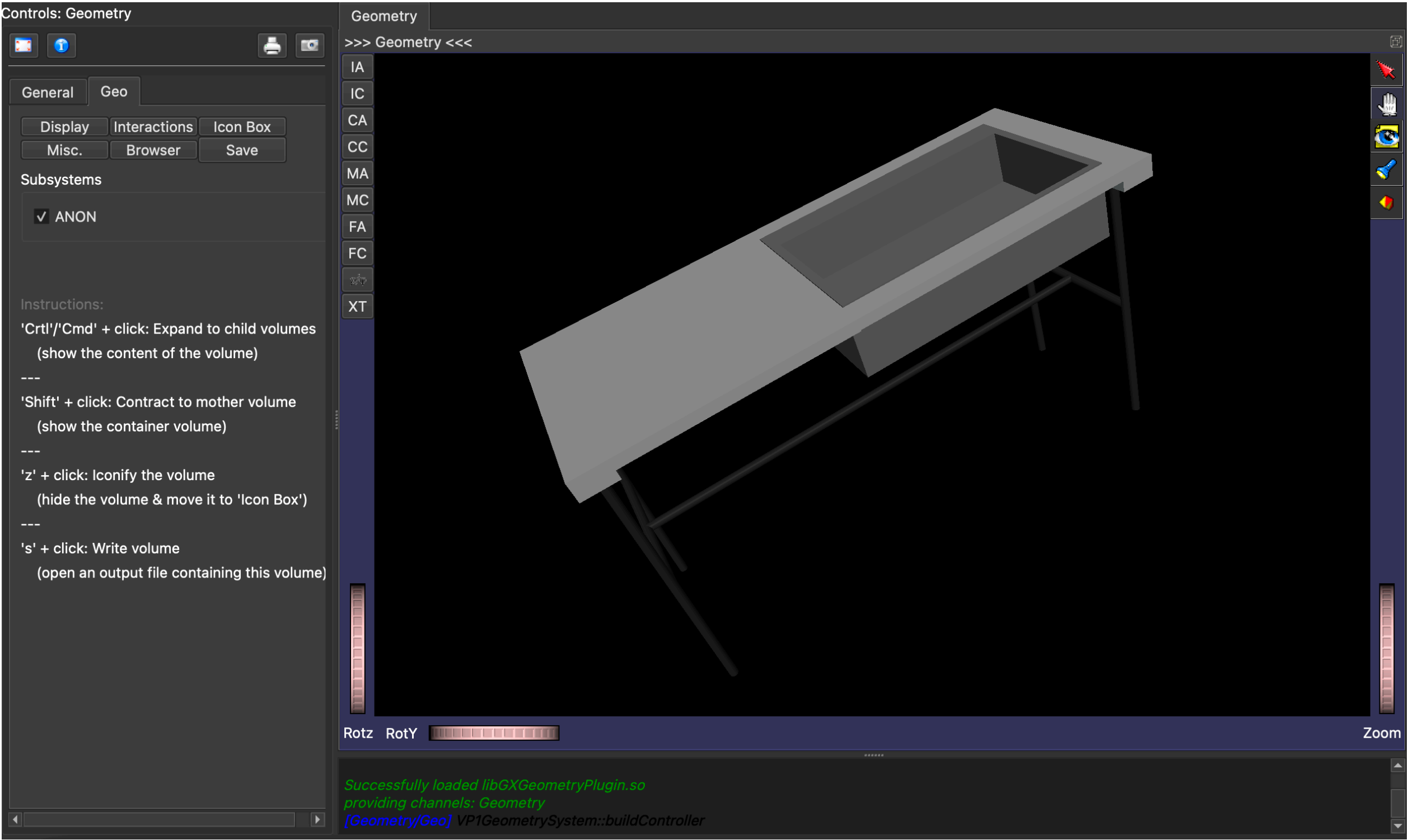
```
void KitchenSinkPlugin::create(GeoPhysVol *world, bool /*publish*/)
{
    const double degree=M_PI/180.0;

    // Define elements used in this example:
    GeoElement *oxygen      = new GeoElement("Oxygen",    "O",    19,  39*gram/mole);
    GeoElement *nitrogen    = new GeoElement("Nitrogen",  "N",    7,  14*gram/mole);
    GeoElement *argon       = new GeoElement("Argon",     "Ar",   18,  40*gram/mole);
    GeoElement *aluminium   = new GeoElement("Aluminium", "Al",   13,  26*gram/mole);
    GeoElement *iron        = new GeoElement("Iron",      "Fe",   26,  55.8*gram/mole);
    GeoElement *chromium    = new GeoElement("Chromium",  "Cr",   24,  52*gram/mole);

    // Some dimensions used below:
    double platformHeight=34.5;           // Height to the top of the flanges
    double flangeDiameter=3.375;         // Diameter of the flanges
    double flangeThickness=3.0/16.0;      // Thickness of the flanges
    double t1TubeLength = platformHeight-flangeThickness; // Overall length of tube t1;
    double innerRadius=0.75/2.0;          // 3/4 inch (inner diameter) pipe

    const GeoTube    *t1Tube    = new GeoTube(innerRadius,outerRadius, t1TubeLength/2.0);
    const GeoLogVol   *t1Log     = new GeoLogVol("T1Log", t1Tube, Iron);
    GeoPhysVol        *t1Phys    = new GeoPhysVol(t1Log);

    world->add(xform1);
    world->add(t1Phys);
    world->add(xform2);
    world->add(t1Phys);
}
```

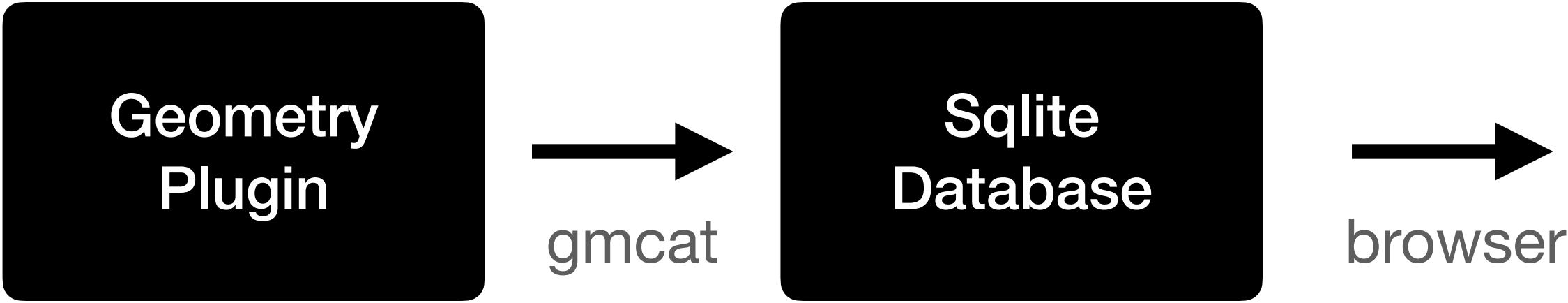


GeoModel Description

Gmex visualization

• [Check Out Kitchen Sink Plugin](#)

# Dump into Database file



The screenshot shows a database browser interface with a dark theme. The top toolbar includes buttons for "New Database", "Open Database", "Write Changes", "Revert Changes", "Open Project", and "Save Project". Below this is a tab bar with "Database Structure" (selected), "Browse Data", "Edit Pragmas", and "Execute SQL". Under the "Database Structure" tab, there are buttons for "Create Table", "Create Index", and "Print". The main area displays a tree view of the database structure. The "Tables (18)" folder is expanded, showing a list of tables with their names, types, and schemas. The tables listed are: AlignableTransforms, ChildrenPositions, Elements, FullPhysVols, Functions, GeoNodesTypes, IdentifierTags, LogVols, Materials, NameTags, PhysVols, RootVolume, SerialDenominators, SerialIdentifiers, SerialTransformers, Shapes, Transforms, and dbversion. Below the tables, there are sections for "Indices (0)", "Views (0)", and "Triggers (0)".

Name	Type	Schema
AlignableTransforms	Table	CREATE TABLE AlignableTransforms(id integer primary key,xx real,xy real,xz real,yx real,yy real,yz real)
ChildrenPositions	Table	CREATE TABLE ChildrenPositions(id integer primary key,parentId integer,parentTable integer)
Elements	Table	CREATE TABLE Elements(id integer primary key,name varchar,symbol varchar,Z varchar,A varchar)
FullPhysVols	Table	CREATE TABLE FullPhysVols(id integer primary key,logvol integer not null)
Functions	Table	CREATE TABLE Functions(id integer primary key,expression varchar)
GeoNodesTypes	Table	CREATE TABLE GeoNodesTypes(id integer primary key,nodeType varchar,tableName varchar)
IdentifierTags	Table	CREATE TABLE IdentifierTags(id integer primary key,identifier integer)
LogVols	Table	CREATE TABLE LogVols(id integer primary key,name varchar,shape integer not null,material integer)
Materials	Table	CREATE TABLE Materials(id integer primary key,name varchar,density varchar,elements varchar)
NameTags	Table	CREATE TABLE NameTags(id integer primary key,name varchar)
PhysVols	Table	CREATE TABLE PhysVols(id integer primary key,logvol integer not null)
RootVolume	Table	CREATE TABLE RootVolume(id integer primary key,volid integer not null,volTable integer not null)
SerialDenominators	Table	CREATE TABLE SerialDenominators(id integer primary key,baseName varchar)
SerialIdentifiers	Table	CREATE TABLE SerialIdentifiers(id integer primary key,baseId integer)
SerialTransformers	Table	CREATE TABLE SerialTransformers(id integer primary key,funcId integer not null REFERENCES Functions(id))
Shapes	Table	CREATE TABLE Shapes(id integer primary key,type varchar,parameters varchar)
Transforms	Table	CREATE TABLE Transforms(id integer primary key,xx real,xy real,xz real,yx real,yy real,yz real)
dbversion	Table	CREATE TABLE dbversion (id integer primary key,version integer)

Geometry Database

# Back to FullSimLight (Command Line)

```
./fullSimLight -g mygeometry.db
```

```
./fullSimLight -g libHGTDPPlugin.1.0.0.dylib
```

```
./fullSimLight -c myconfig.json
```



CHEP 2023

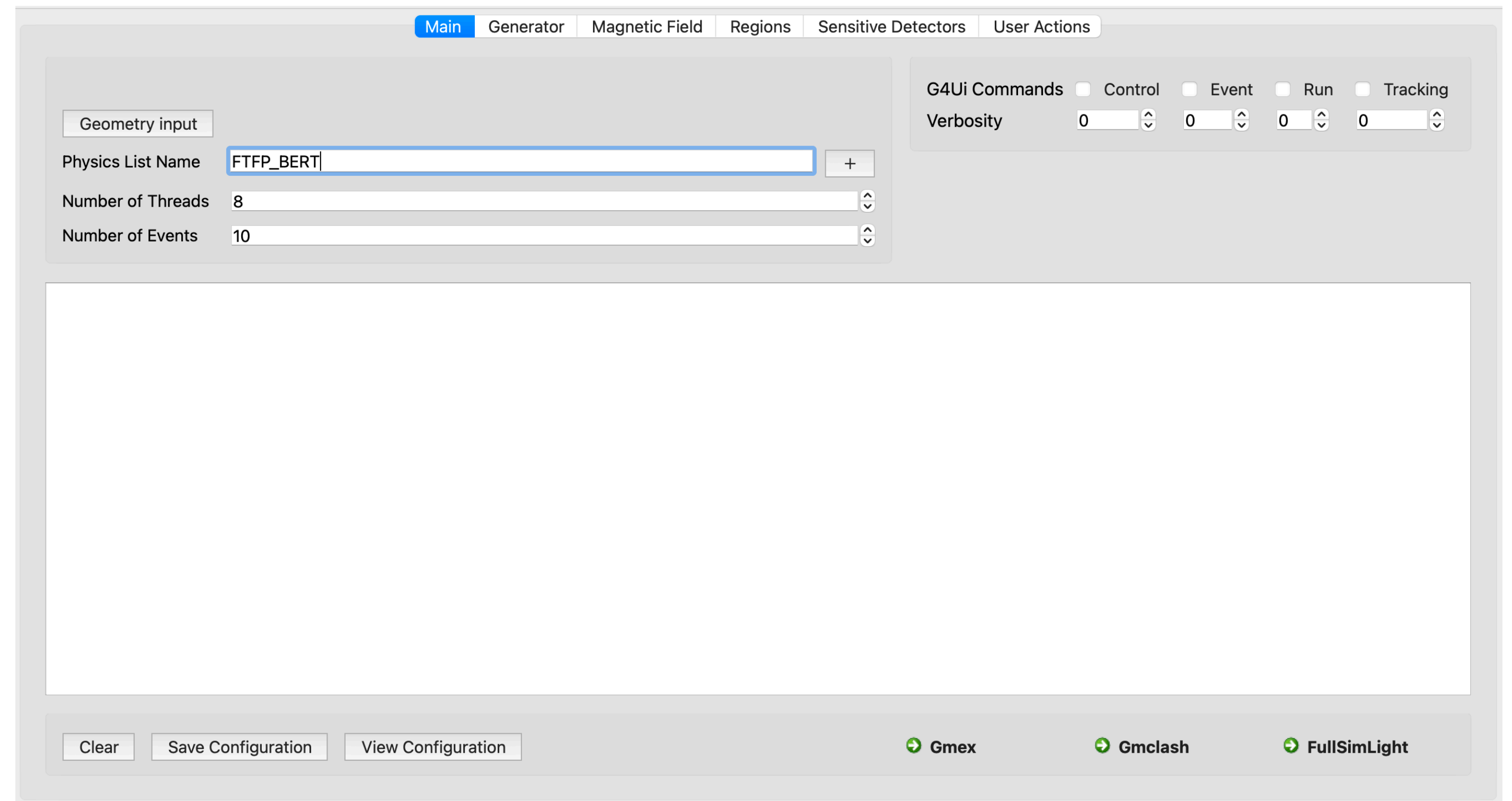
New way of running FullSimLight





# FSL

- FSL is the GUI to FullSimLight.
- Produces a fool-proof configuration file which can be ran with FullSimLight through the -c flag.
- FullSimLight and other GeoModel tools can also be run right within the FSL interface.



FSL GUI



CHEP 2023



# Event Generation

Particle gun.

Pythia

HepMC3 Files  
(New)

Event Generator  
Plugin (New)



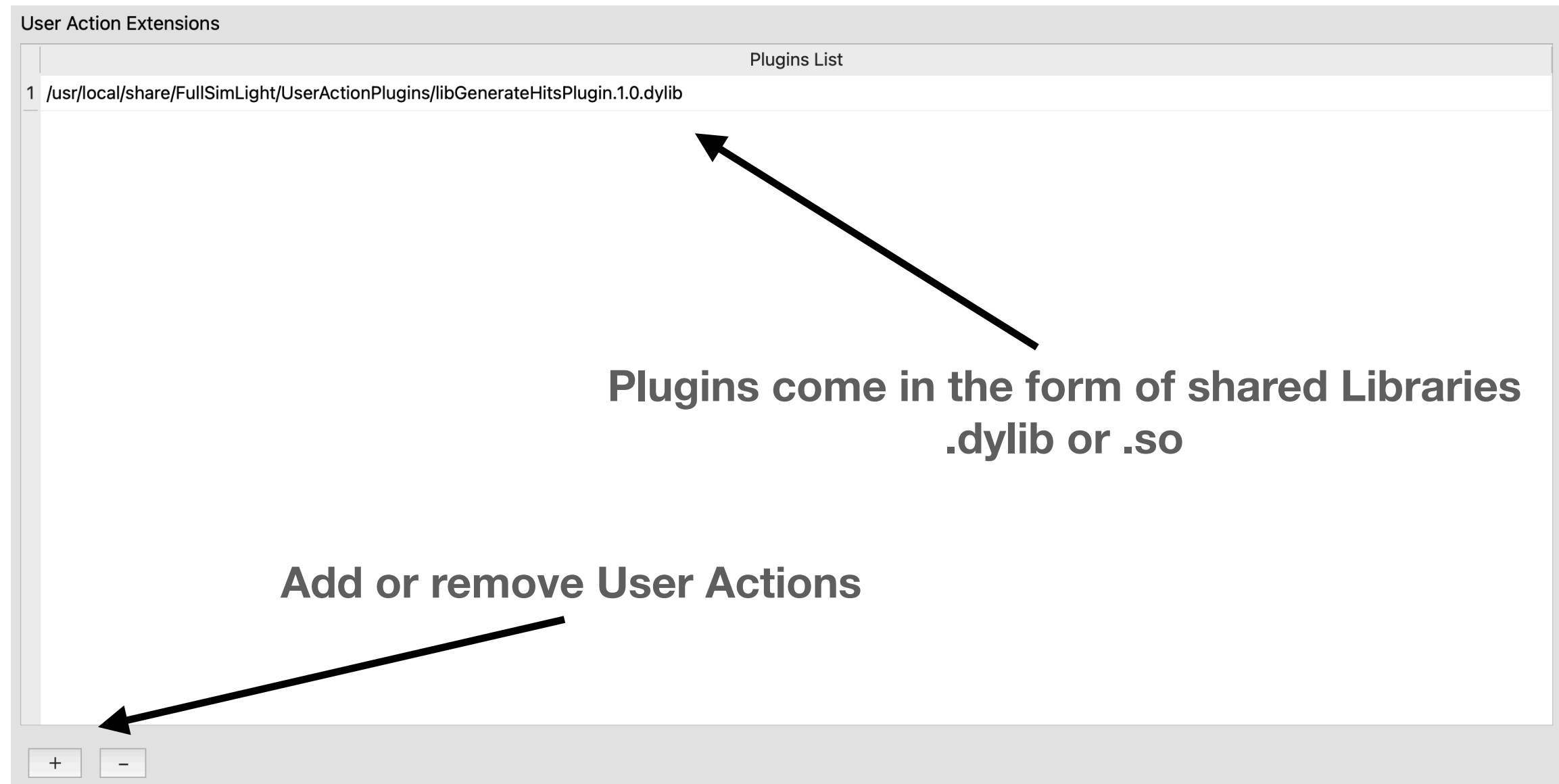
CHEP 2023

A screenshot of a web-based GUI for event generation. At the top, a dropdown menu titled "Select Generator" is open, showing four options: "Particle Gun" (selected with a checkmark), "Pythia", "HepMC3 File", and "Generator Plugin". Below this, the "Particle Gun" configuration panel is visible, with fields for "Particle name" (set to "e-"), "Px" (0 GeV), "Py" (10 GeV), and "Pz" (0 GeV). To the right, the "Pythia" panel shows "Pythia Config File" (with a "Browse Files" button) and "Type of Event" (set to "ttbar"). Below that, the "HepMC3" panel shows "HepMC3 Event File" (with a "Browse Files" button) and "File Format" (set to "Ascii"). At the bottom, the "Plugin" panel shows "Generator Plugin" (with a "Browse Files" button").

Generator Menu on GUI



# Plugin Architecture



User Action Plugin Menu



Magnetic Field Plugin Menu



CHEP 2023

- Plugins which come in the form of shared libraries are the mechanism for users to extend their simulations. Plugins can be used to add

- User Actions
- Sensitive Detectors
- Magnetic Field
- Physics Lists
- Event Generators

- FSL provides a simple interface to add the plugins through various menus, some of which are shown on the right.

- FullSimLight comes with a number of custom plugins to do various things such as record hits, generate the ATLAS magnetic field, etc.





# Writing Plugins

Geant4 provides User Actions to interrupt processing at specific points.



FullSimLight provides Abstract classes to interface with this mechanism via plugins.



The Plugin instantiates and uses user customized G4UserActions.



Example: Hits Plugin (produces record of Geant4 Stepping points)

Plugin

```
class GenerateHitsPlugin:public FSLUserActionPlugin {
public:
    GenerateHitsPlugin();
    virtual G4UserSteppingAction *getSteppingAction() const final override;
    virtual G4UserEventAction *getEventAction() const final override;

    GenerateHitsEvent* eventaction = new GenerateHitsEvent();
    GenerateHitsStep* stepaction = new GenerateHitsStep();
};
```

User Actions

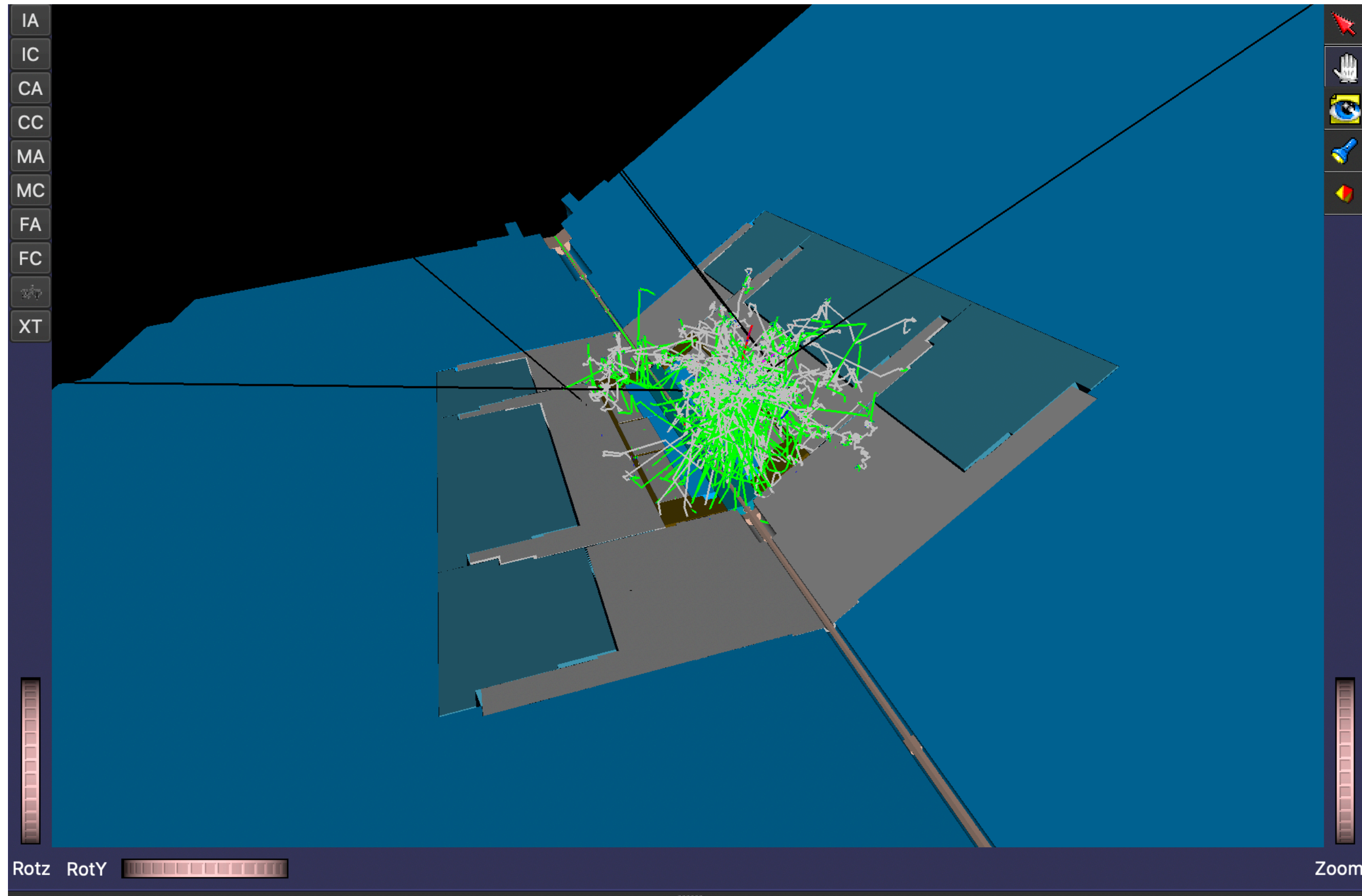
Abstract Class



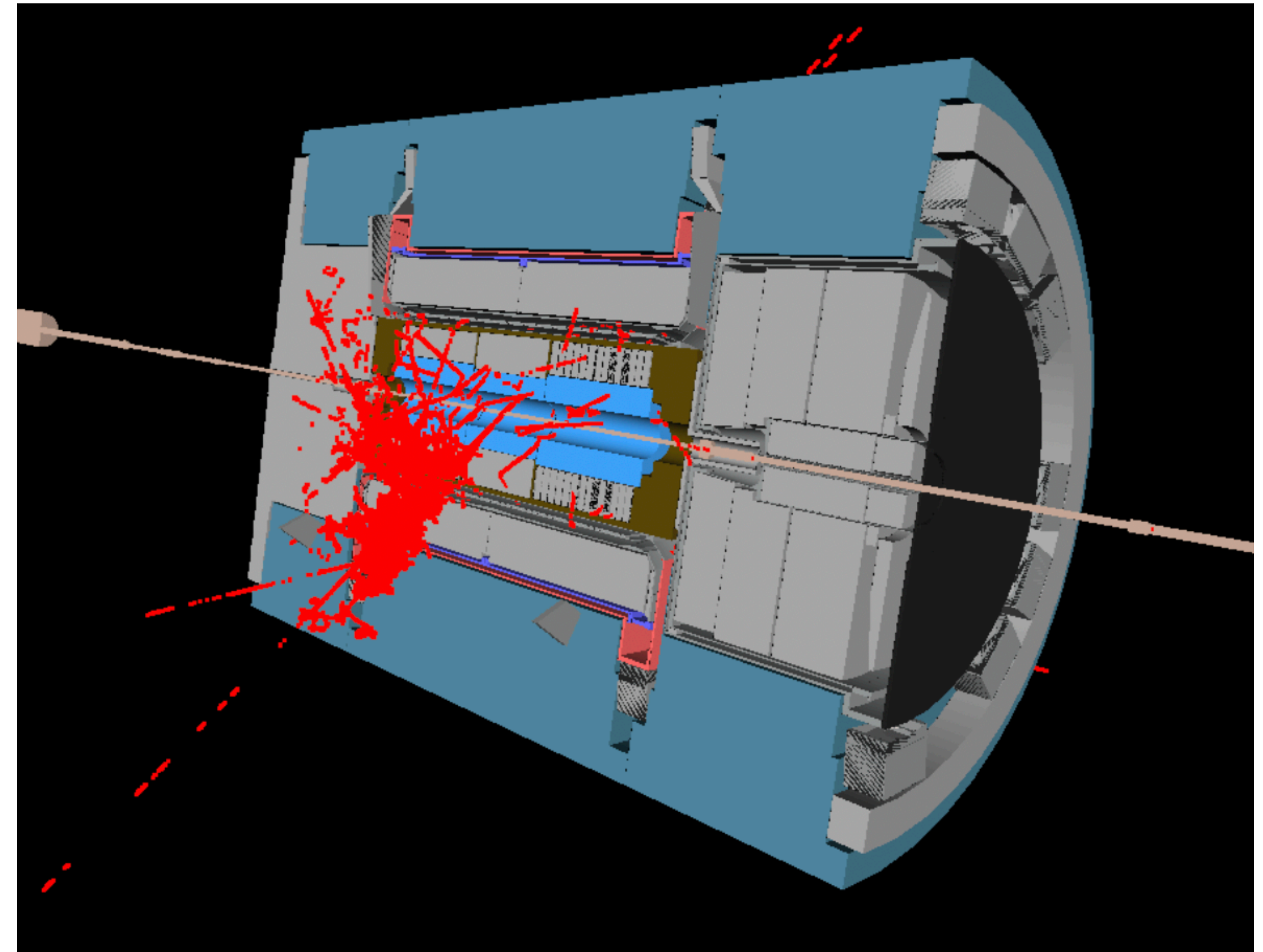
• [Check Out Hits Plugin](#)



# Step and Tracks Visualization in gmex



Tracks Visualization in gmex



Steps Visualization in gmex



# Regions

- The mechanism to configure regions is found on the Regions tab in FSL.
- Root Logical volumes and cuts can be specified as required.

	Region Name	RootLV Names	Electron Cut (GeV)	Proton Cut (GeV)	Positron Cut (GeV)	Gamma Cut (GeV)
1	TRT	Gas, GasMA	30	1	30	0.05
2	Pixel	siLog, siBLayLog, dbmDia...	0.05	0.05	1	0.05
3	DriftWall	MDTDriftWall	0.05	1	0.05	1
4	FCAL	LAr::FCAL::LiquidArgonC	0.03	1	0.03	0.03

List of Regions

Regions Configuration

Region Name

RootLV Names

Electron Cut (GeV)

Proton Cut (GeV)

Positron Cut (GeV)

Gamma Cut (GeV)

Add Region

+

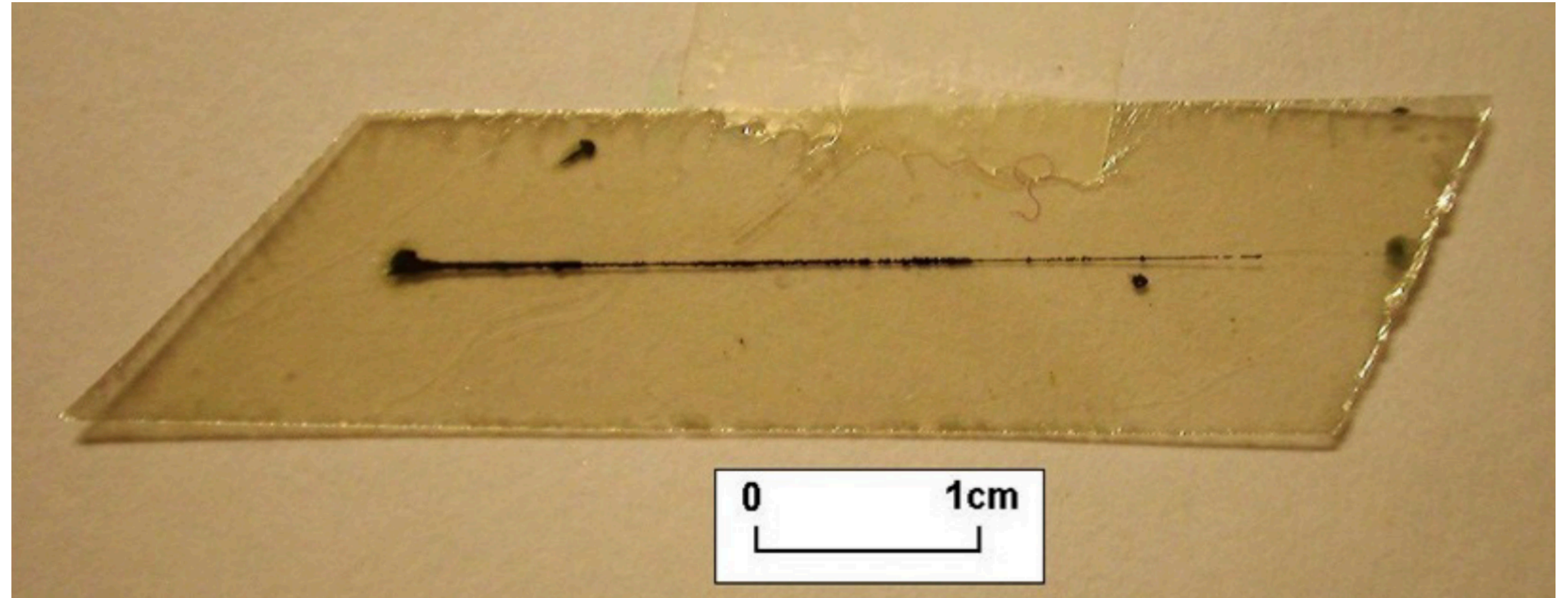
-





# Practical Example

- It is commonly believed that grand unified theories (GUTs) predict proton decay.
- Muscovite Mica is one possible place to look for evidence of such proton decay by analyzing positron tracks.
- Easy to simulate using FullSimLight.



**Positron track in muscovite mica** (F.M. Russell. *In Quodons in Mica*, J.F. R. Archilla et al, eds., Springer (2015) pp. 474–559.)



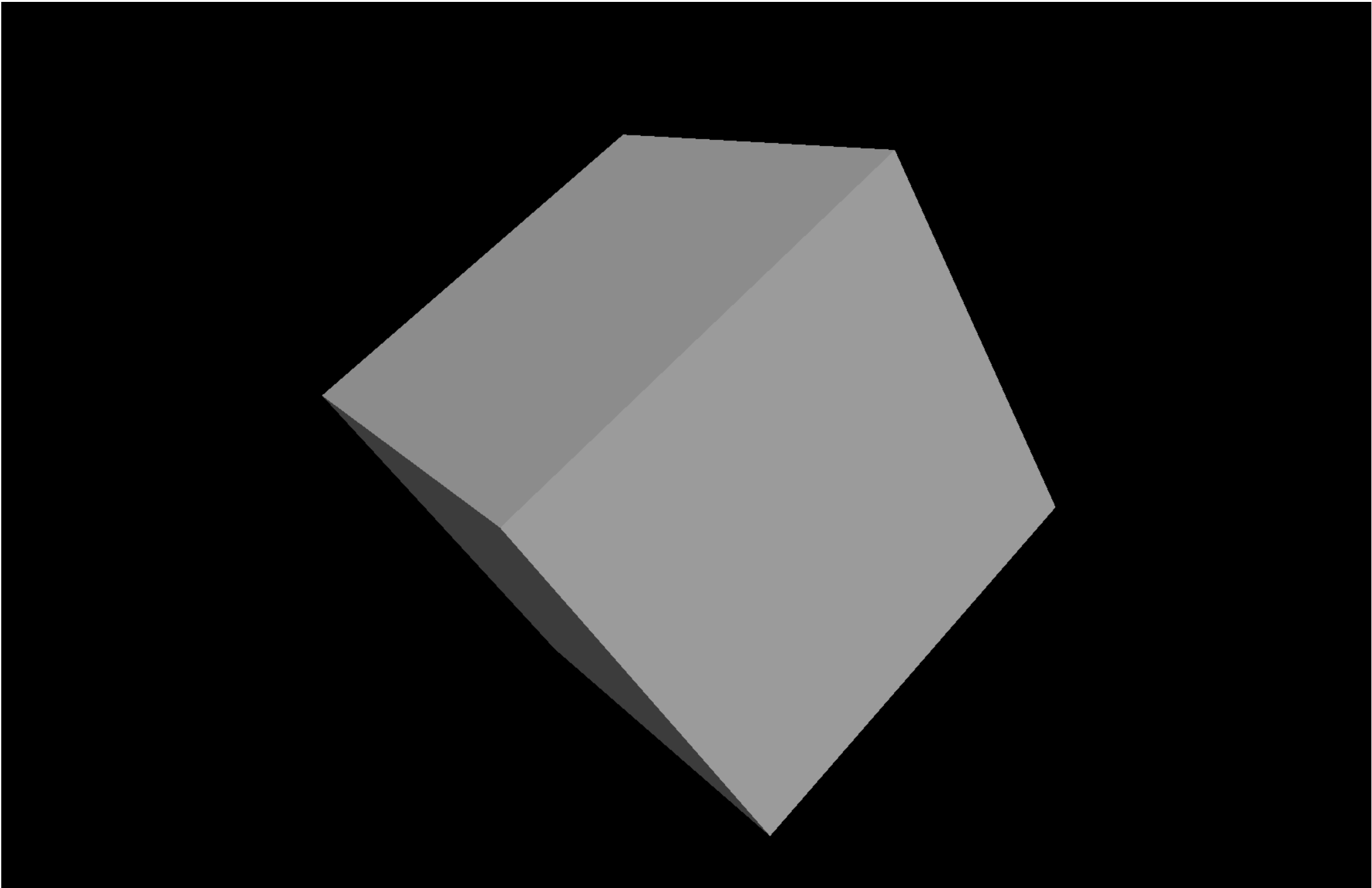
# Cube of Mica

```
void MicaPlugin::create(GeoPhysVol *world, bool /*publish*/)
{
    GeoElement *potassium = new GeoElement("Potassium", "K", 19, 39*gram/mole);
    GeoElement *oxygen     = new GeoElement("Oxygen", "O", 8, 16*gram/mole);
    GeoElement *aluminium  = new GeoElement("Aluminium", "Al", 13, 26*gram/mole);
    GeoElement *silicon    = new GeoElement("Silicon", "Si", 14, 28*gram/mole);
    GeoElement *hydrogen   = new GeoElement("Hydrogen", "H", 1, 1*gram/mole);
    GeoElement *fluorine   = new GeoElement("Fluorine", "F", 9, 19*gram/mole);

    //Defining Mica
    double densityOfMica = 2.82*gram/cm3;
    GeoMaterial *Mica = new GeoMaterial("Mica",densityOfMica);
    Mica->add(potassium,1);
    Mica->add(oxygen,11.8);
    Mica->add(aluminium,3);
    Mica->add(silicon,3);
    Mica->add(hydrogen,1.8);
    Mica->add(fluorine,0.2);
    Mica->lock();

    const double barWidth1=4.0;
    const double barWidth2=3.0;
    double barThickness    =1.25;
    double cutoutDepth     = 21.5;
    double cutoutWidth     = 32.375;

    const GeoBox *MicaBox  = new GeoBox(100*cm, 100*cm,100*cm);
    const GeoLogVol *MicaLog = new GeoLogVol("MicaLog", MicaBox, Mica);
    GeoPhysVol *MicaPhys = new GeoPhysVol(MicaLog);
    world->add(MicaPhys);
}
```



Gmex visualization

GeoModel Description



# Simulation

===== Run summary =====

Number of events = 10

-----

Mean energy deposit per event = 0.4981 +- 0.006169 [GeV]

Mean track length (charged) per event = 115.4 +- 49.83 [cm]

Mean track length (neutral) per event = 4568 +- 6626 [cm]

Number of steps (charged) per event = 1002 +- 106.6

Number of steps (neutral) per event = 719.1 +- 105.4

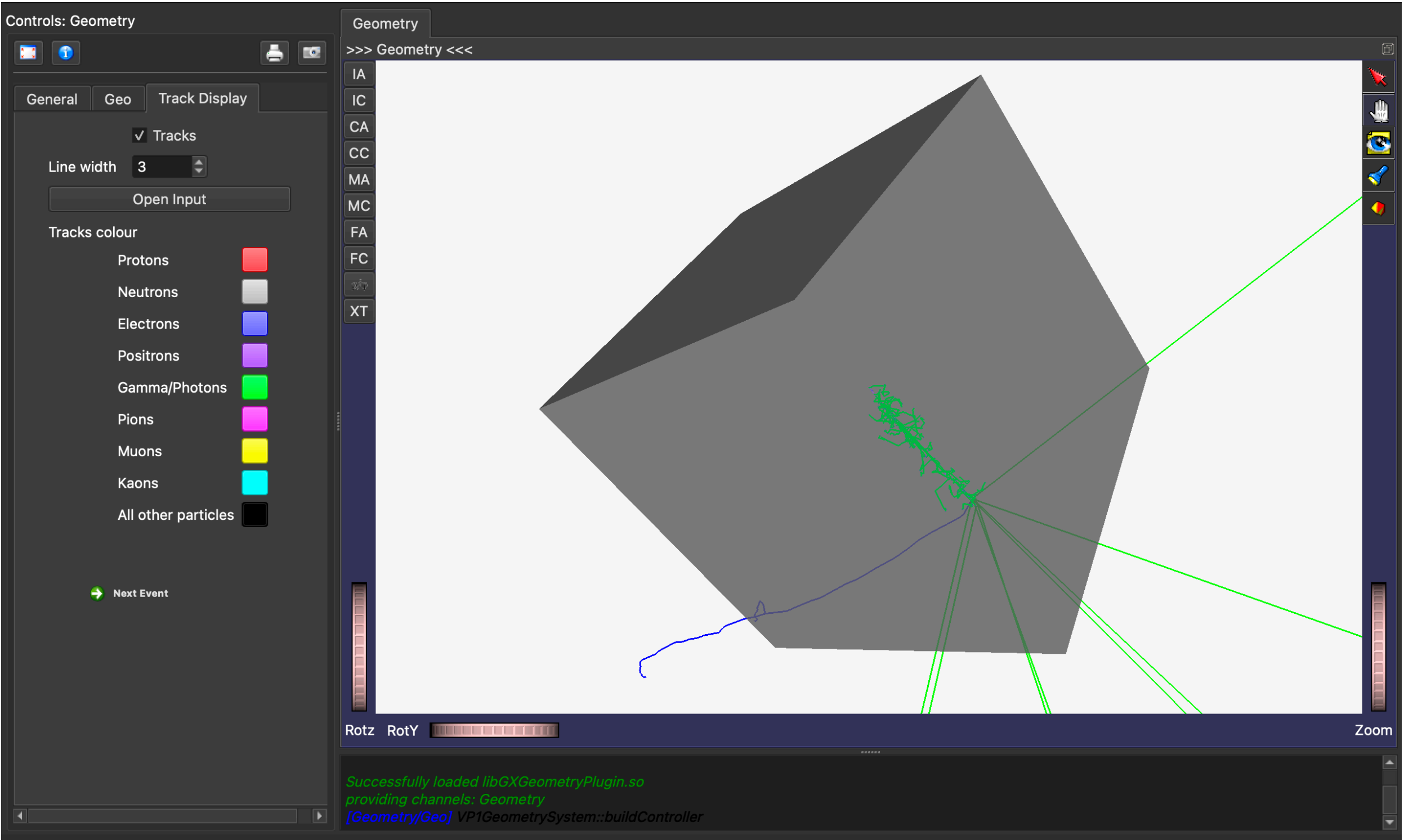
Number of secondaries per event :

Gammas = 108.5 +- 11.49

Electrons = 712.5 +- 96.47

Positrons = 7.1 +- 2.119

.....



FullSimLight Basic Scoring

Particle shower inside Mica Cube



CHEP 2023





# Summary of Recent Developments in FullSimLight

- Plugin Mechanism to allow users and developers with diverse goals to extend and customize the simulation.
- A GUI for fast, transparent, and foolproof configuration.
- Reading in events from HepMC3 files and region configuration.
- Built in visualization of steps and tracks in gmex.
- Source code can be found at [GeoModel](#) and documentation/installation instructions at [Documentation](#).



CHEP 2023

