REST-for-Physics framework for GEANT4 simulations and data analysis

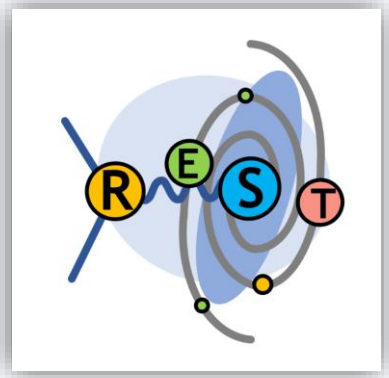
Álvaro Ezquerro Sastre

on behalf of REST-for-Physics developer group

aezquerro@unizar.es

Universidad Zaragoza
1542

CAPA Centro de Astropartículas y Física de Altas Energías
Universidad Zaragoza



What is REST-for-Physics ?

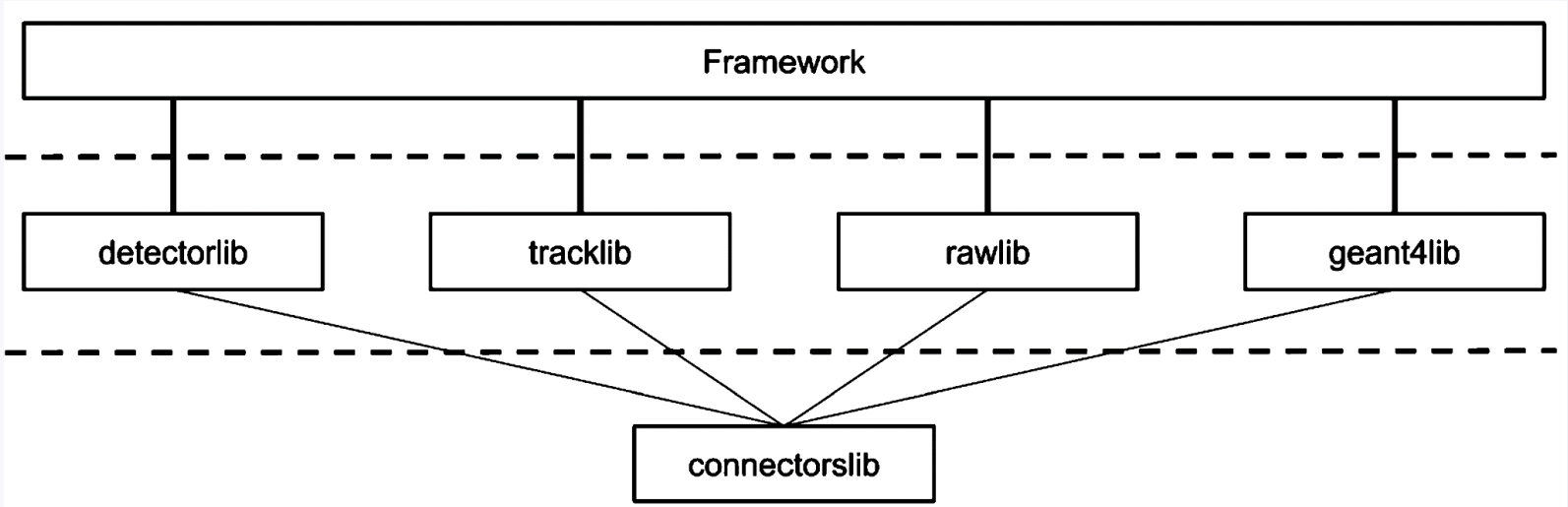
- **R**are **E**vent **S**earches **T**oolkit
- a ROOT-based open-source software toolkit to process the data from detector physics experiments
- Originated at University of Zaragoza as a common environment for experimental and analysis activities
- Used in several experiments:
  - TREX-DM
  - BabyIAXO
  - PandaX-III

Why REST?

- Traceability of code and reproducibility of data
- Modularity (libraries and packages)
- Common data structures and processes for both simulated (Geant4) and experimental events

How? REST Project structure

- Core framework: TRestMetadata, TRestEvent, TRestEventProcess classes
- Distributed in libraries:
  - Geant4Lib
  - RawLib
  - DetectorLib
  - TrackLib
- Packages:
  - restG4



REST-for-Physics, a ROOT-based framework for event oriented data analysis and combined Monte Carlo response, K. Altenmüller, S. Cebrián, T. Dafni et al., , Computer Physics Communications, 108281, April 2022

Core framework

- TRestEvent
  - Base class for any event
  - The framework is designed around them (event-oriented)

- TRestMetadata
  - Base class for every configurable object
    - Serialization via .rml (extended .xml) file
  - Allows traceability and reproducibility

- *fVersion*: A string containing the human version number.
- *fCommit*: The latest commit hash value when the compilation took place.
- *fLibraryVersion*: The human version library. It is fixed by CMakeLists at the library submodules.
- *fOfficialRelease*: It will be true if the commit was tagged at the repository.
- *fCleanState*: It will be true if there are no local modifications (including submodules). To remove any local modifications and recover a clean state we may execute `source clean-state.sh` at the project root.

```
<TRestRun name="Process" title="${EXPERIMENT} Simulations. Version ${VERSION}." verboseLevel="info">
    <parameter name="experimentName" value="${EXPERIMENT}"/>
    <parameter name="readOnly" value="false" />
    <parameter name="runNumber" value="preserve"/>
    <parameter name="runTag" value="preserve"/>
    <parameter name="runType" value="G4analysis"/>
    <parameter name="runDescription" value="Analysis of Cd109 calibration"/>
    <parameter name="user" value="${USER}"/>
    <parameter name="overwrite" value="off" />
    <parameter name="outputFileName" value="R[fRunNumber]_[fExperimentName]_[fRunType]_V[fVersion].root"/>
</TRestRun>
```

Other important classes

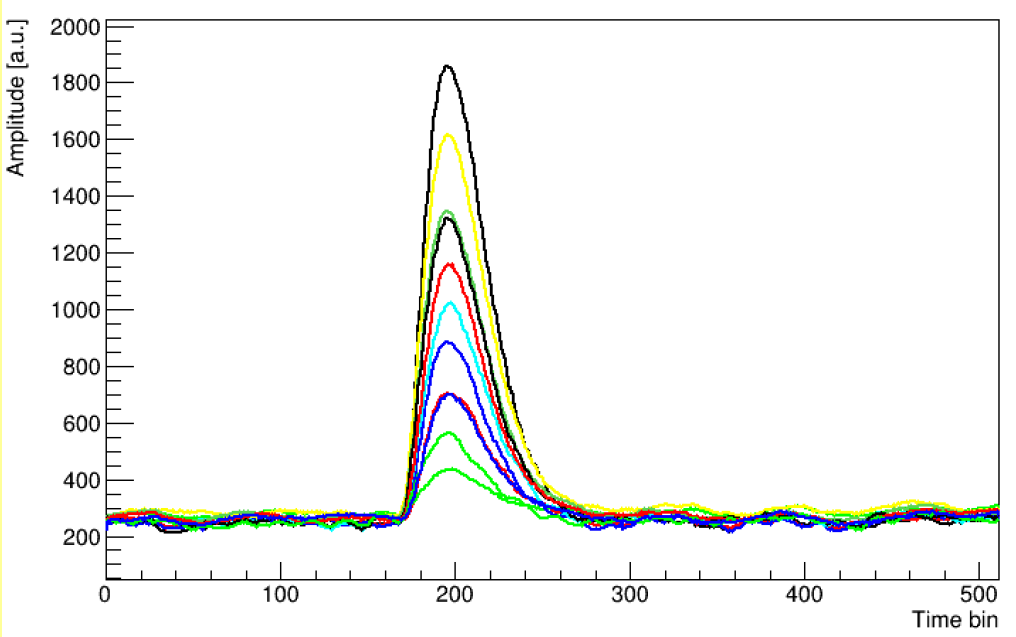
- TRestRun
  - Stores run information
  - Handles I/O (via ROOT)
  - Provides access to the events
- TRestEventProcess
  - Base class for every process with event as input and output
- TRestAnalysisTree
  - TTree derived class containing observables for each event

Libraries

RawLib

TRestRawSignalEvent

- Represents the raw pulse from the electronics
- Time bin sampled by the electromics



TRestRawSignalAnalysisProcess

DetectorLib

TrackLib

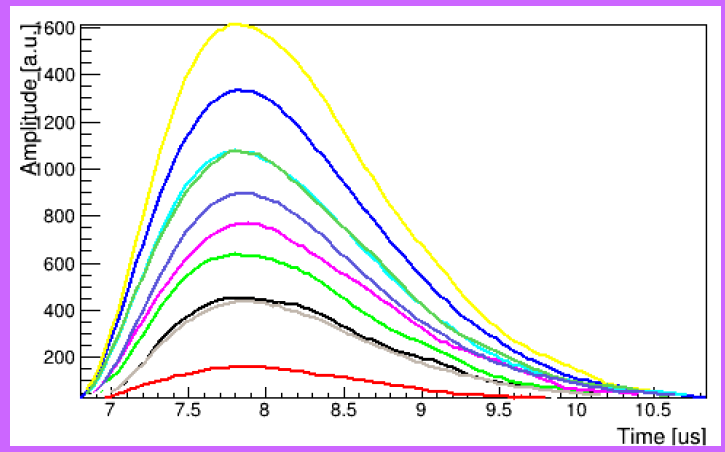
Geant4Lib

Libraries

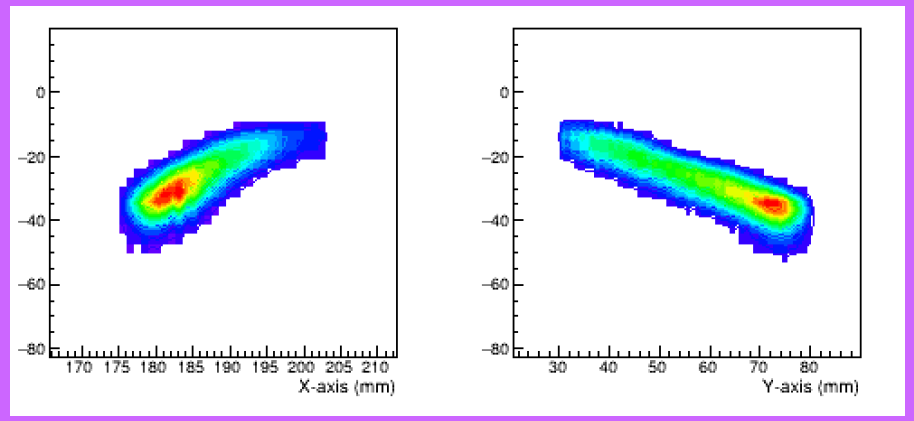
RawLib

DetectorLib

TRestDetectorSignalEvent
- Physical time
- Baseline substraction



TRestDetectorHitsEvent
- Signal ID translated to physical position
- Amplitude to “energy”



TrackLib

Geant4Lib

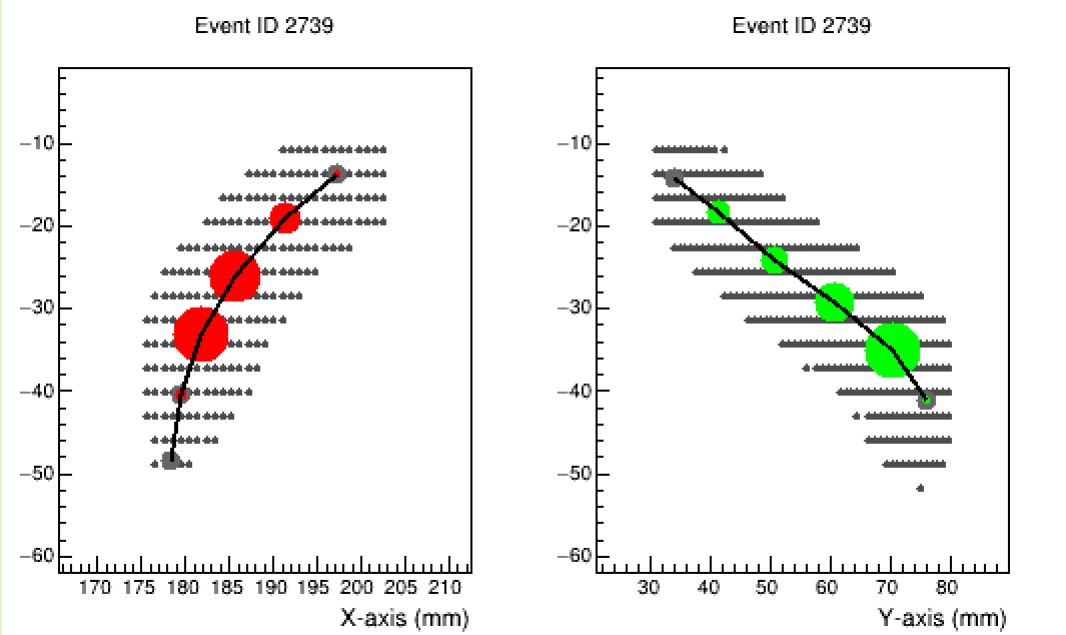
Libraries

RawLib

DetectorLib

TrackLib

TRestTrackEvent
- Aglutinate hits within a clusterDistance



TRestTrack(3D/2D)AnalysisProcess
TRestTrackLineAnalysisProcess
…

Geant4Lib

Libraries

RawLib

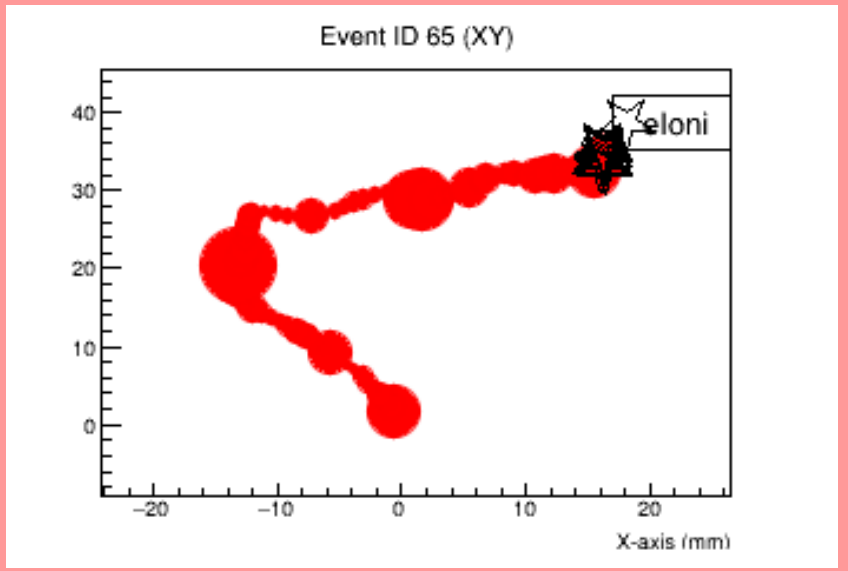
DetectorLib

TrackLib

Geant4Lib

TRestGeant4Event

- Particle
- Trajectory
- Energy depositions (hits)
- Primaries and processes



TRestGeant4AnalysisProcess

REST file content

REST-for-Physics, a ROOT-based framework for event oriented data analysis and combined Monte Carlo response, [*K. Altenmüller, S. Cebrián, T. Dafni et al., , Computer Physics Communications, 108281, April 2022*]

restG4 package

- Allows to fully configure the GEANT4 simulation **using RML files without writing a single line of C++/Geant4 code**

Command: `restG4 -c electrons.rml -e 1000`

```
3   <restG4>
4
5   >   <TRestRun name="schoolRun" title="REST-for-Physics school run">···
15      </TRestRun>
16
17  >   <TRestGeant4Metadata name="Electron" title="Electron">···
42      </TRestGeant4Metadata>
43
44  >   <TRestGeant4PhysicsLists name="default" title="First physics list implementation.">···
76      </TRestGeant4PhysicsLists>
77
78  </restG4>
```

REST-school23 (day3/session2)

restG4 package

- Allows to fully configure the GEANT4 simulation **using RML files without writing a single line of C++/Geant4 code**
  - Define the TRestRun (run information)
  - Define Geant4 parameters
    - Geometry (gdml)
    - Sensitive detectors
    - Generator
  - Define G4 physics

```
<TRestRun name="schoolRun" title="REST-for-Physics school run">
    <parameter name="experimentName" value="School"/>
    <parameter name="runType" value="simulation"/>
    <parameter name="runNumber" value="auto"/>
    <parameter name="runTag" value="Electron"/>
    <parameter name="outputFileName" value="Run[fRunNumber]_[fVersion].root"/>
    <parameter name="runDescription" value=""/>
    <parameter name="user" value="${USER}"/>
    <parameter name="overwrite" value="off"/>
    <parameter name="readOnly" value="false"/>
</TRestRun>
```

restG4 package

- Allows to fully configure the GEANT4 simulation **using RML files without writing a single line of C++/Geant4 code**
  - Define the TRestRun (run information)
  - Define Geant4 parameters
    - Geometry (gdml)
    - Sensitive detectors
    - Generator
  - Define G4 physics

```
<TRestGeant4Metadata name="Electron" title="Electron">
    <parameter name="gdmlFile" value="geometry.gdml"/>
    <parameter name="subEventTimeDelay" value="100" units="us"/>
    <parameter name="seed" value="17021"/>
    <parameter name="nEvents" value="10000"/>
    <parameter name="registerEmptyTracks" value="false"/>
    <parameter name="saveAllEvents" value="false"/>

    <generator type="point" position="(0,0,0)" units="mm">
        <source particle="e-">
            <angular type="isotropic"/>
            <energy type="mono" energy="250" units="keV"/>
        </source>
    </generator>

    <detector>
        <parameter name="energyRange" value="(0,2)" units="MeV"/>
        <parameter name="activateAllVolumes" value="false"/>

        <volume name="box" sensitive="true" maxStepSize="1mm"/>
    </detector>

</TRestGeant4Metadata>
```

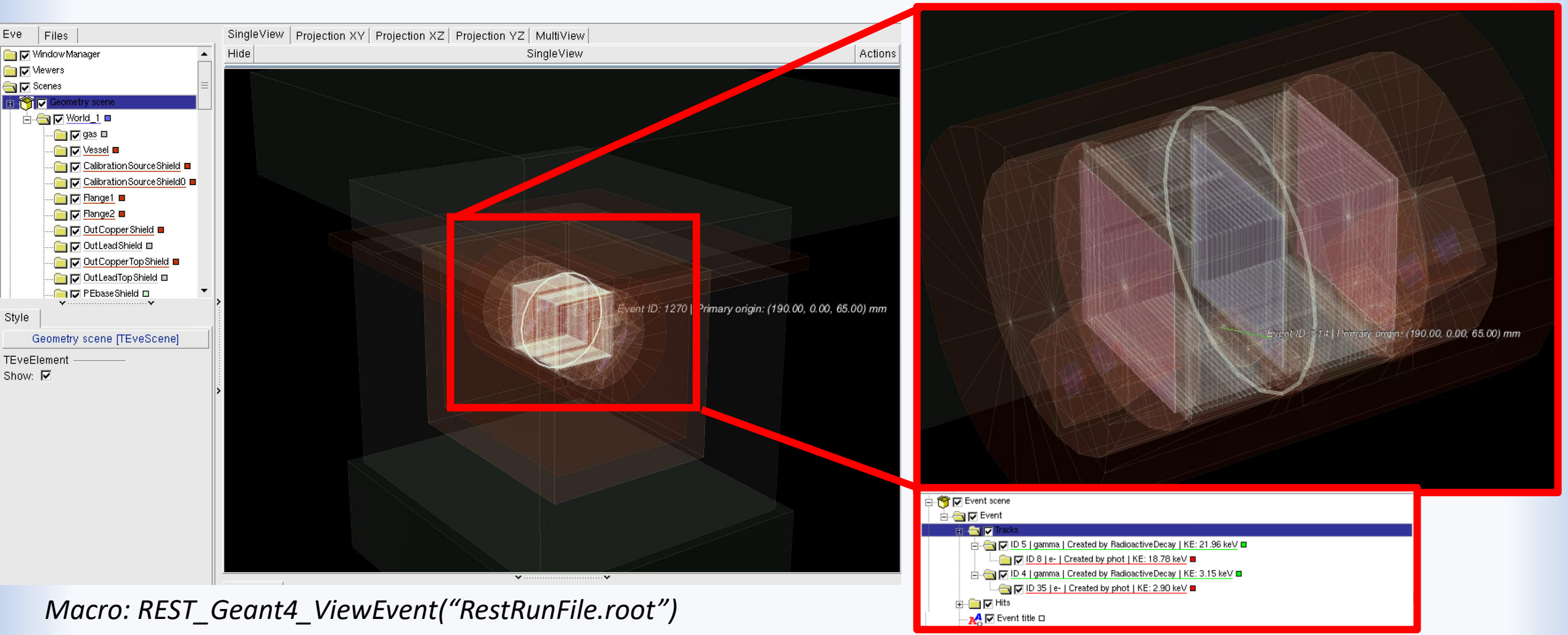
restG4 package

- Allows to fully configure the GEANT4 simulation **using RML files without writing a single line of C++/Geant4 code**
  - Define the TRestRun (run information)
  - Define Geant4 parameters
    - Geometry (gdml)
    - Sensitive detectors
    - Generator
  - Define G4 physics

```
<TRestGeant4PhysicsLists name="default" title="First physics list implementation.">
    <parameter name="cutForGamma" value="0.01" units="mm"/>
    <parameter name="cutForElectron" value="2" units="mm"/>
    <parameter name="cutForPositron" value="1" units="mm"/>
    <parameter name="cutForMuon" value="1" units="mm"/>
    <parameter name="cutForNeutron" value="1" units="mm"/>
    <parameter name="minEnergyRangeProductionCuts" value="1" units="keV"/>
    <parameter name="maxEnergyRangeProductionCuts" value="1" units="GeV"/>

    <!-- EM Physics lists -->
    <!--<physicsList name="G4EmLivermorePhysics"> </physicsList>-->
    <!-- <physicsList name="G4EmPenelopePhysics"> </physicsList> -->
    <physicsList name="G4EmStandardPhysics_option4"></physicsList>

    <!-- Decay physics lists -->
    <physicsList name="G4DecayPhysics"></physicsList>
    <physicsList name="G4RadioactiveDecayPhysics"></physicsList>

    <!-- Hadron physics lists -->
    <physicsList name="G4HadronElasticPhysicsHP"></physicsList>
    <physicsList name="G4IonBinaryCascadePhysics"></physicsList>
    <physicsList name="G4HadronPhysicsQGSP_BIC_HP"></physicsList>
    <physicsList name="G4NeutronTrackingCut"></physicsList>
    <physicsList name="G4EmExtraPhysics"></physicsList>

</TRestGeant4PhysicsLists>
```

restG4 output

- The result of the restG4 simulation is a REST file with TRestGeant4Event



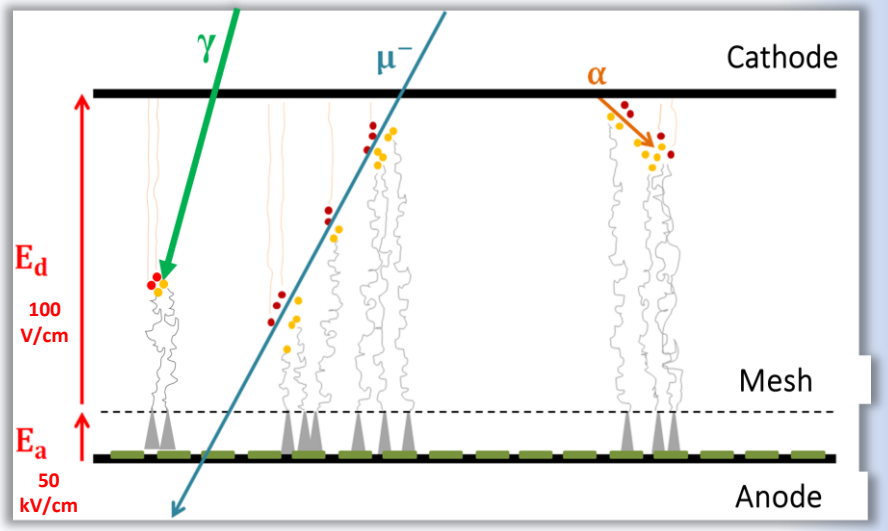
Macro: REST_Geant4_ViewEvent("RestRunFile.root")

restG4 output processing

- The result of the restG4 simulation is a REST file with TRestGeant4Event

- To obtain experimental-like events, they must be processed taking into account the specific physics of the detector.

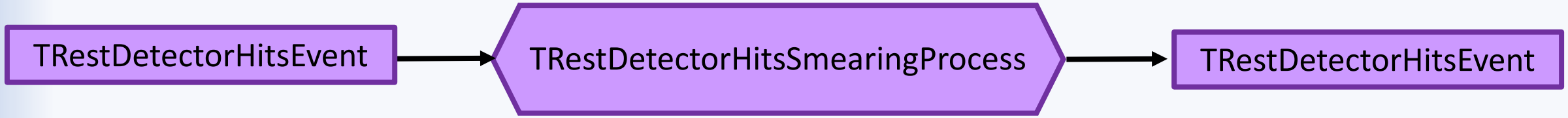
- For example: Gaseous **T**ime **P**rojection **C**hamber
  - Energy resolution
  - Electron diffusion

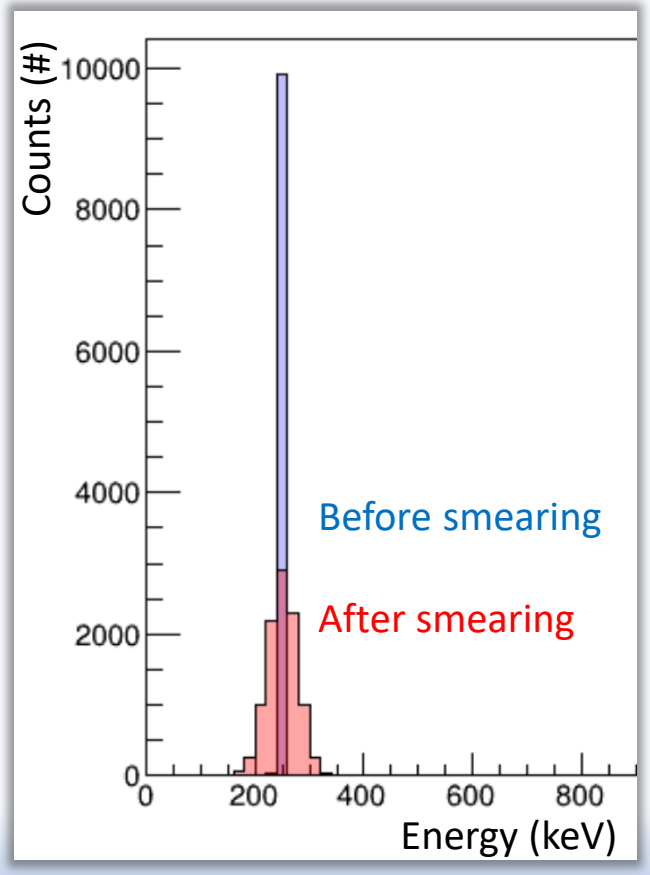


Geant4Event Workflow

- Stores a set of (energy, position) hits
- Removes Geant4 track information (step processes names, etc.)
- Ignore hits outside detector volume

Geant4Event Workflow

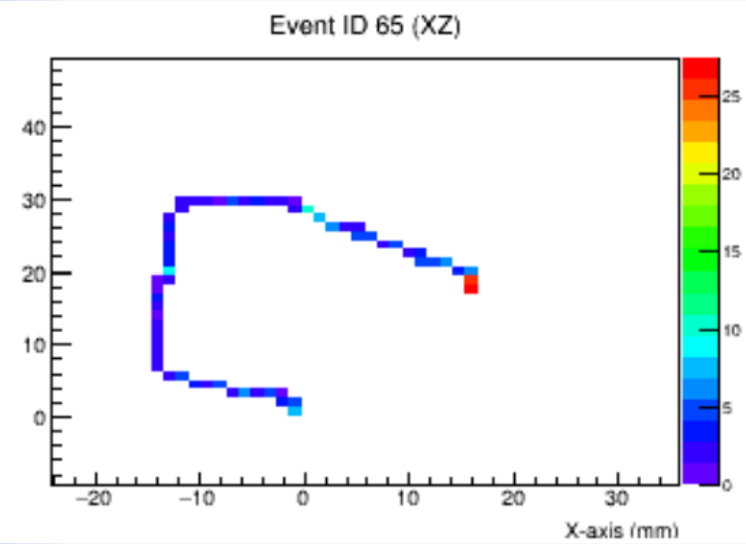




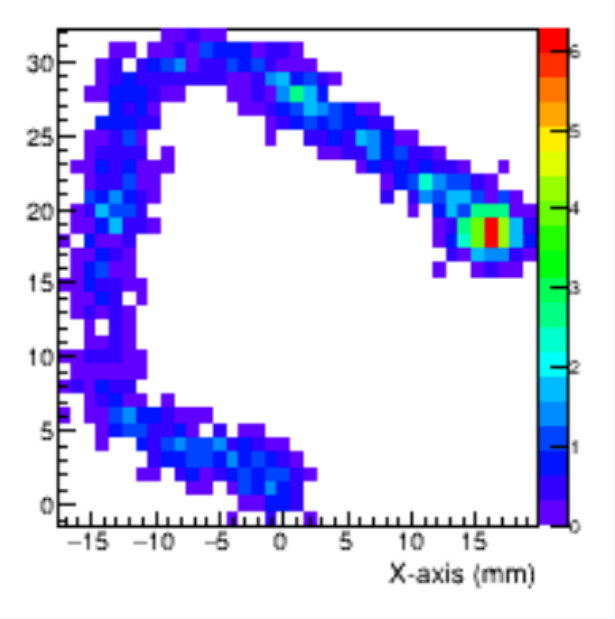
- Introduce an stochastic smearing on the energy of the hits for each event.
- The total energy of each event will be re-distributed following a gaussian shaped function.

Geant4Event Workflow

TRestDetectorHitsEvent → TRestDetectorElectronDiffusionProcess → TRestDetectorHitsEvent



- Particular physics of gaseous TPC detector
- Simulates the difusión of electrons while drifting
- Uses detector properties
  - Drieft field
  - Gas properties (TRestDetectorGas)



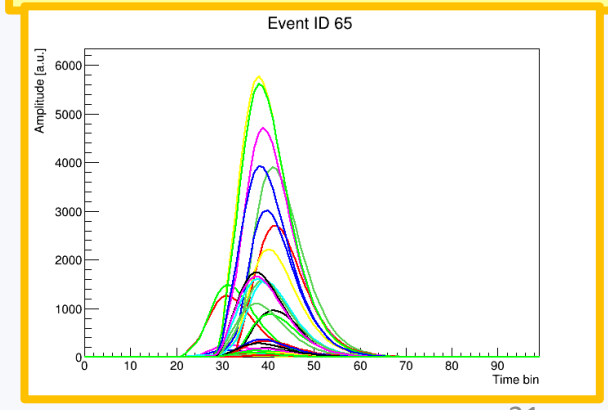
Geant4Event Workflow

TRestDetectorHitsEvent → TRestDetectorHitsToSignalProcess → TRestDetectorSignalEvent

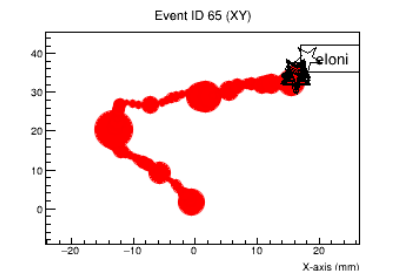
- Translates the Z-coordinate into a physical time using the drift velocity
- XY-coordinates into a electronics channel (TRestDetectorReadout)

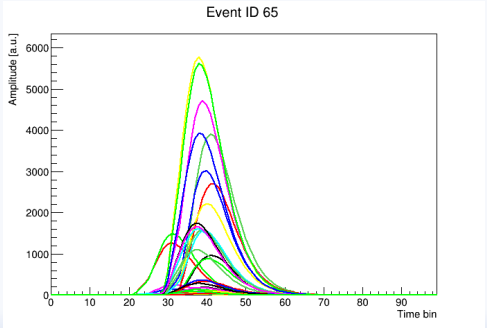
TRestDetectorSignalEvent → TRestDetectorSignalToRawProcess → TRestRawSignalEvent

TRestRawSignalEvent → TRestRawSignalShapingProcess → TRestRawSignalEvent

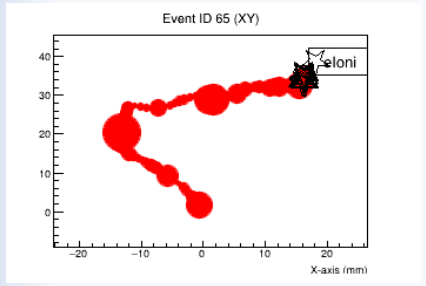


Geant4Event Workflow Summary

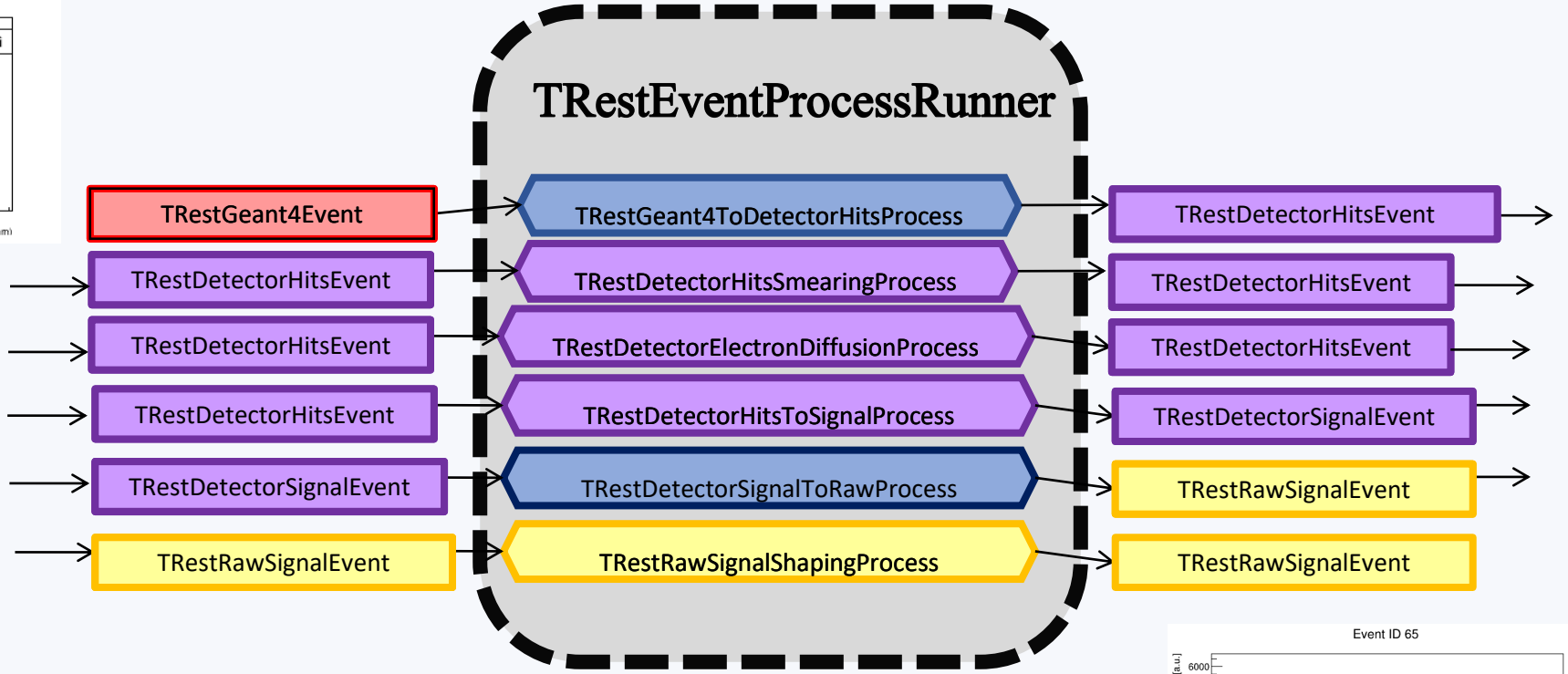


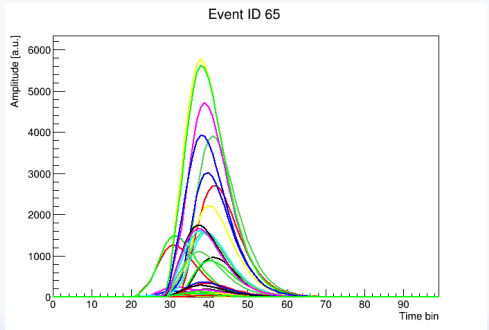


Geant4Event Workflow Summary



Command: `restManager --c response.rml --f data/Run00001_Electron_School.root`





Summary

- restG4 packages interfaces with Geant4 library so the user does not need to write a single line of C++ code to launch the simulation.

- The simulation output events can be processed to finally obtain experimental-like events.

- REST files where all metadata (simulation parameters, geometry, processes, etc.) is saved to ensure traceability and reproducibility of the physics results.

Where to find REST-for-Physics ?

- Github repository: https://github.com/rest-for-physics

- Downloading & installation guide
  - https://rest-for-physics.github.io/downloading.html
  - https://rest-for-physics.github.io/installation
- Doxygen documentation: https://rest-for-physics.github.io/framework/
- REST-school 2023:
  - https://indico.capa.unizar.es/event/26/ *includes installation on Windows from scratch (via WSL)*
  - https://github.com/rest-for-physics/rest-school/

Questions ?

```
**************************************************************************
W E L C O M E   to  R E S T

Commit  : 292f38e1 (2023-12-16 11:39:10 +0100)
Branch/Version : official/v2.4.1
Compilation date : 2023-12-16 12:58

Latest release: : v2.4.1 - Igor G. Irastorza - Sat Dec 16

Official release : Yes
Clean state : Yes

Installed at : /programas/rest/v2.4.1

REST-for-Physics site : rest-for-physics.github.io

Remember that REST is made by physicists for physicists,
who are supposed to toil and suffer till they become experts.
**************************************************************************
```