RAT-PAC

(is an Analysis Tool)

Javier Caravaca
jcaravaca@berkeley.edu

vPRISM workshop - 27 October 2015

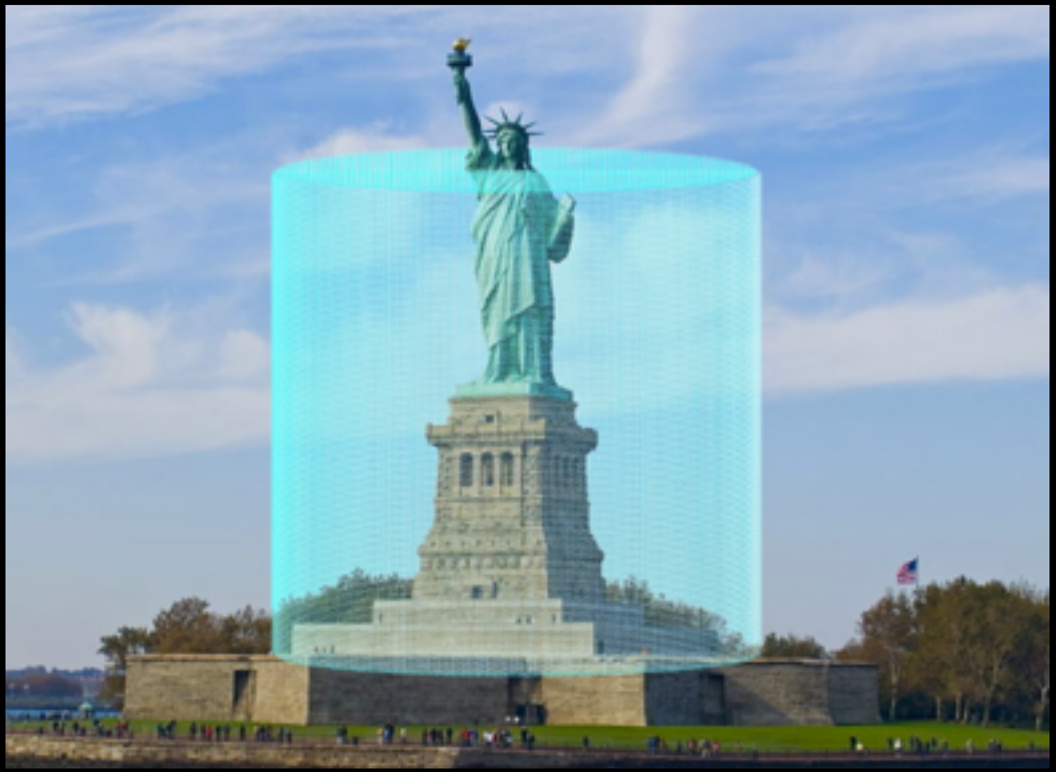
What is RAT?

- It is a toolkit for simulation and analysis in generic particle detectors - specially in optical detectors
- Based on GEANT4, ROOT, C++ and python
- Flexible and easy framework working out of the box, well documented and in continuous development

Used by several experiments

Originally developed for the Braidwood Reactor Neutrino experiment

Customized versions used currently by SNO+, MiniCLEAN and DEAP

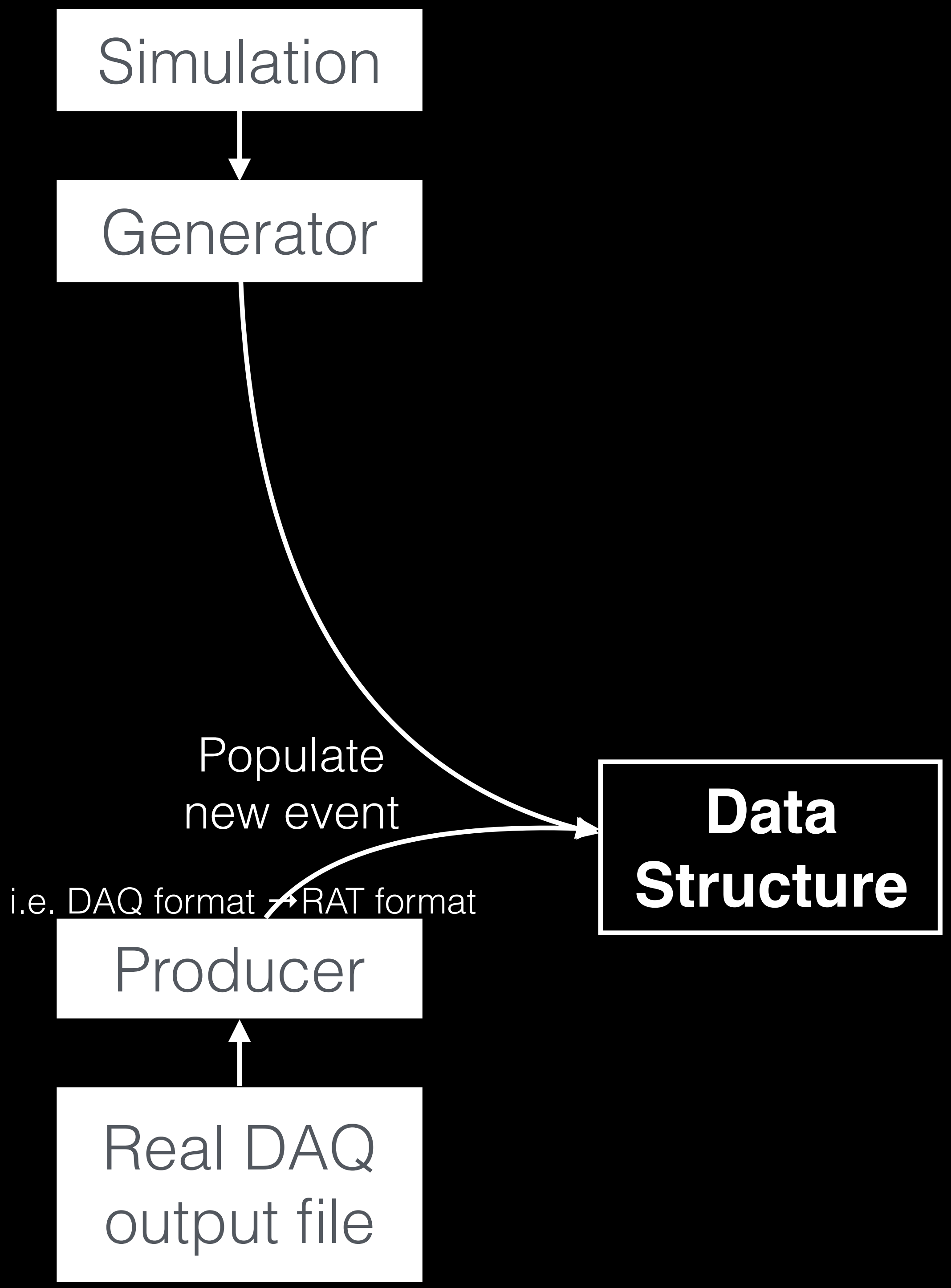


Open source version (RAT-PAC) detached from the MiniCLEAN version and further developed by THEIA and Watchman collaborations

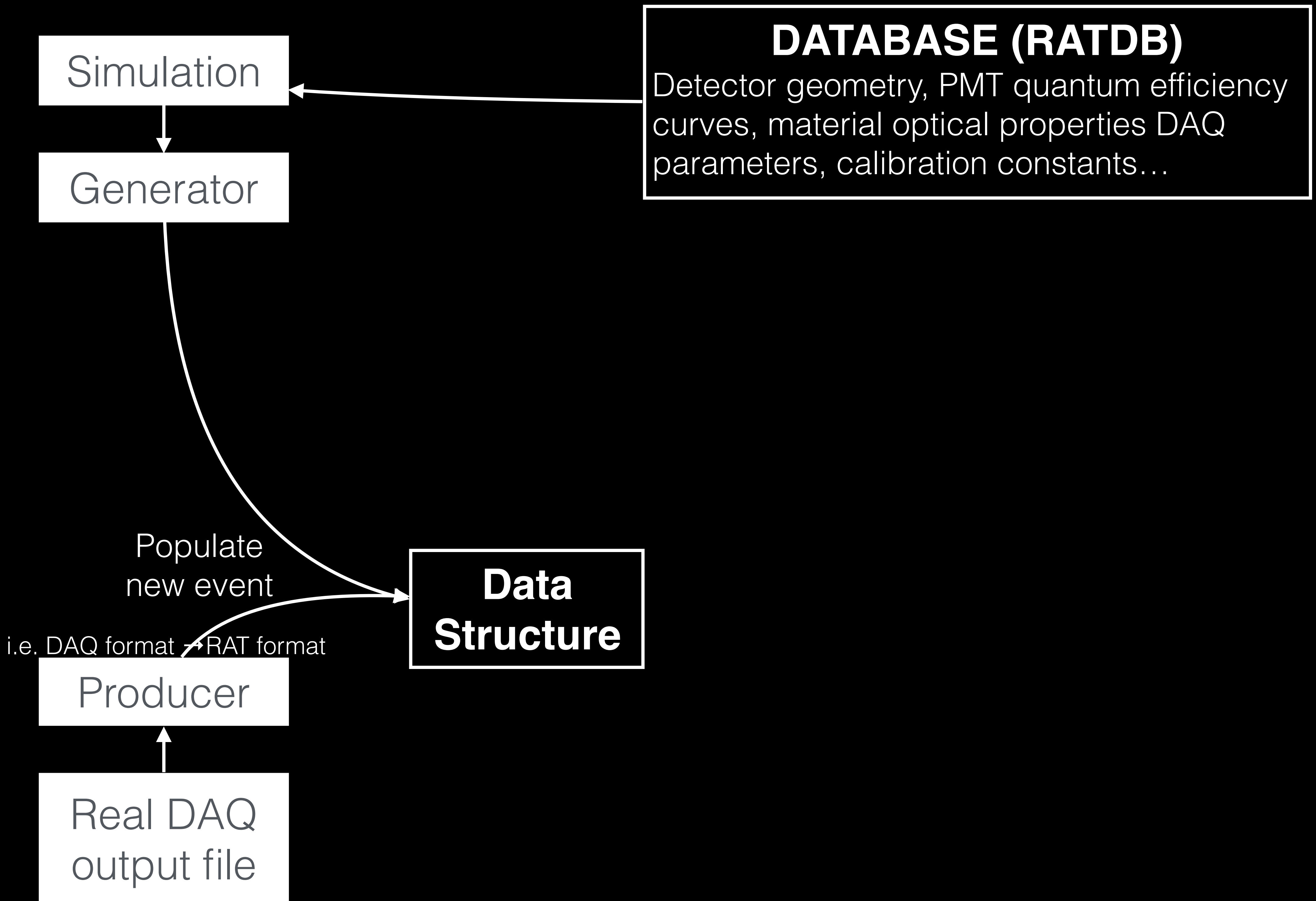
Out of the box toolkit

- **MC event generation**: particle guns, radio nuclei decays, input vertices from ν interactions (e.g. GENIE)…
- **Physics process**: cherenkov, scintillation, full optics, geant4 particle interactions, …
- Generates simulations with **individual photon tracking** and full PMT geometry model
- Provides a **extendible data structure (TTree)** that stores MC truth information as well as DAQ events, at run, event and hit level
- Convenient **database (JSON format)** that reduces hardcoding and allows to quickly change any feature of the simulation/analysis without recompiling
- **Flexible framework** to define custom features: DAQ simulation, reconstruction, calibration, …

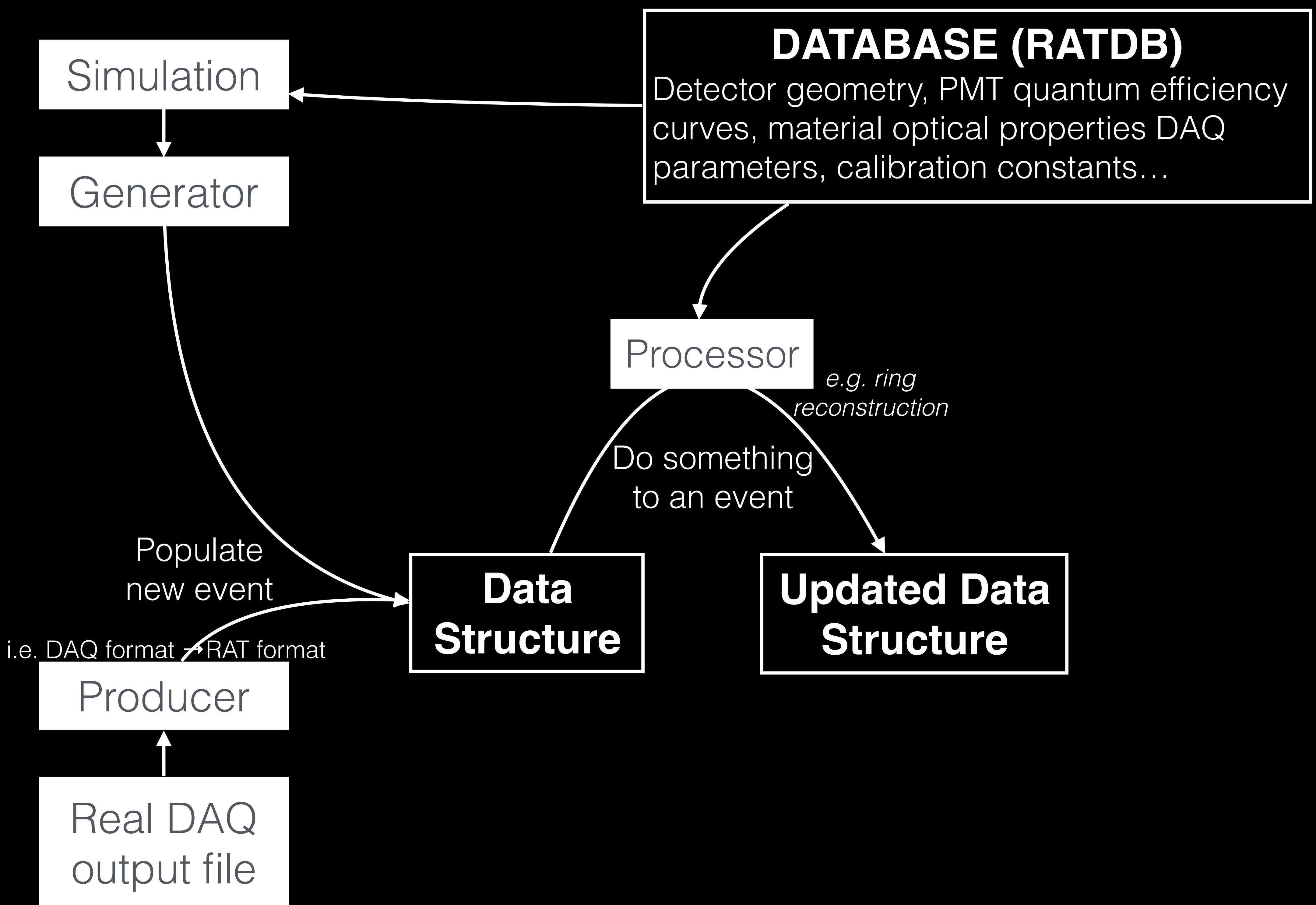
Work flow


Simulation
Generator
Populate
new event
Data
Structure
i.e. DAQ format → RAT format
Producer
Real DAQ
output file

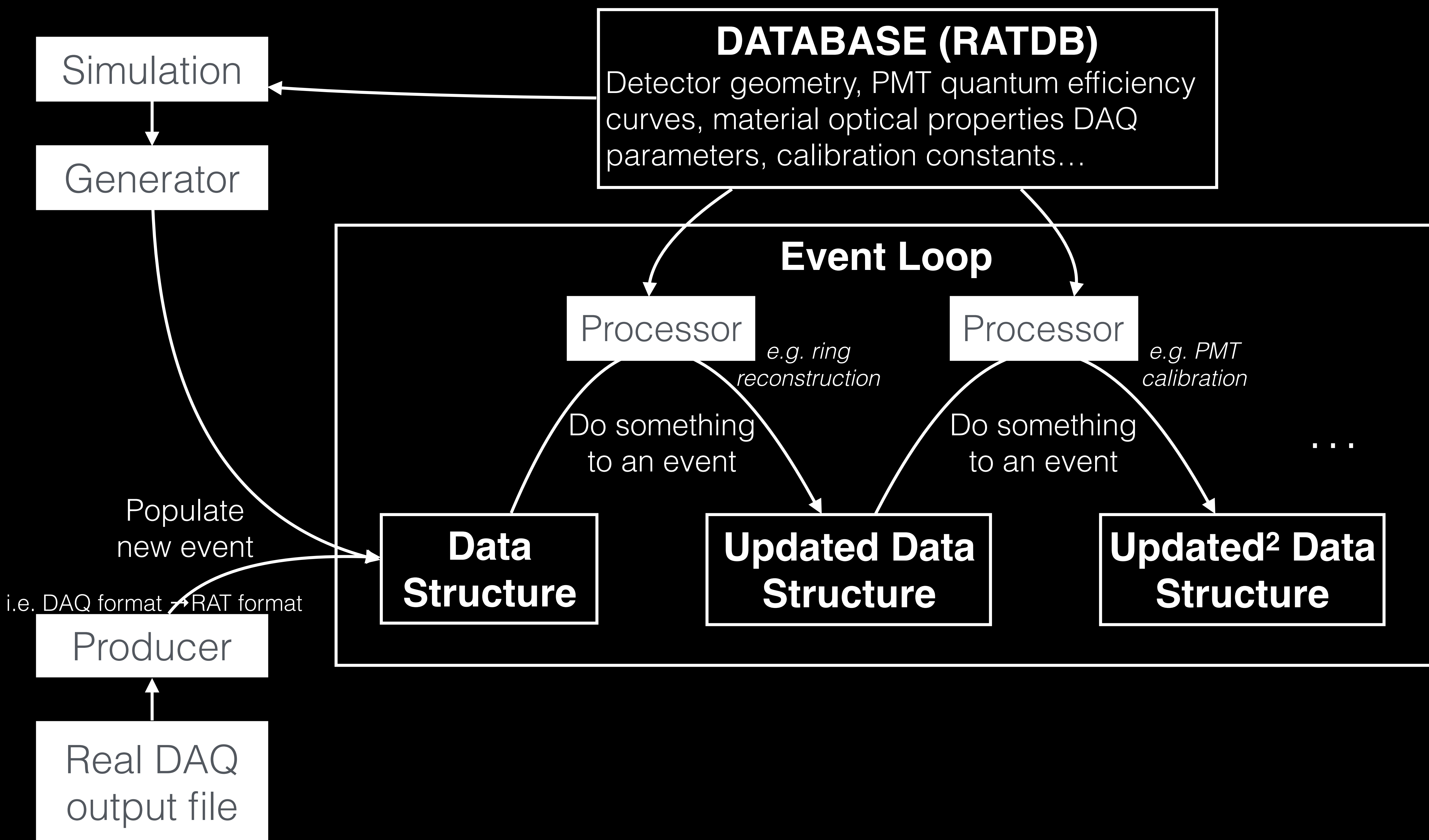
Work flow


DATABASE (RATDB)
Detector geometry, PMT quantum efficiency curves, material optical properties DAQ parameters, calibration constants…
Simulation
Generator
Populate new event
Data Structure
i.e. DAQ format → RAT format
Producer
Real DAQ output file

Work flow

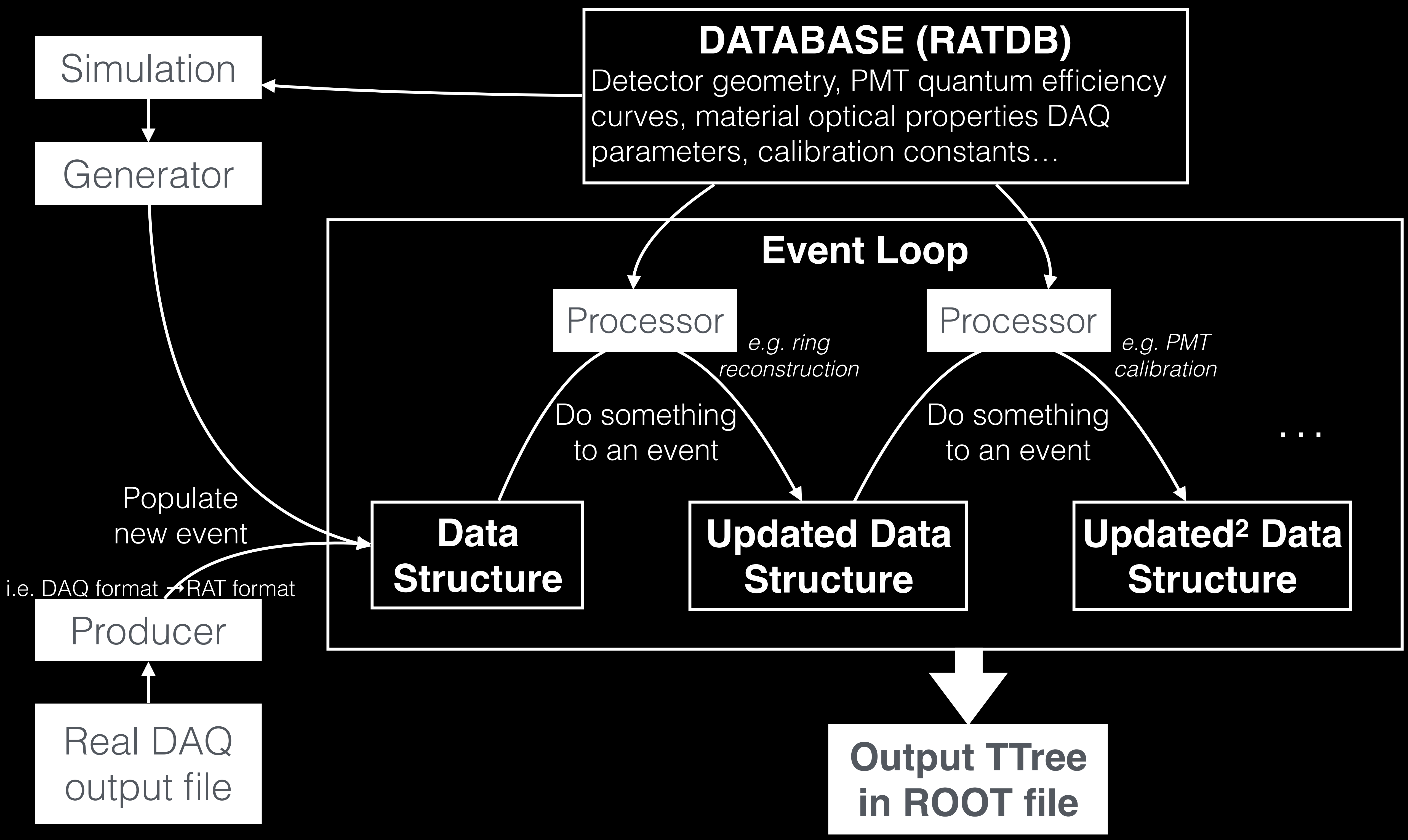

DATABASE (RATDB)
Detector geometry, PMT quantum efficiency curves, material optical properties DAQ parameters, calibration constants…
Simulation
Generator
Processor
e.g. ring reconstruction
Do something to an event
Populate new event
i.e. DAQ format → RAT format
Data Structure
Updated Data Structure
Producer
Real DAQ output file

Work flow



Simulation
Generator
DATABASE (RATDB)
Detector geometry, PMT quantum efficiency curves, material optical properties DAQ parameters, calibration constants…
Event Loop
Processor
e.g. ring reconstruction
Processor
e.g. PMT calibration
Do something to an event
Do something to an event
…
Populate new event
Data Structure
Updated Data Structure
Updated² Data Structure
i.e. DAQ format → RAT format
Producer
Real DAQ output file

Work flow

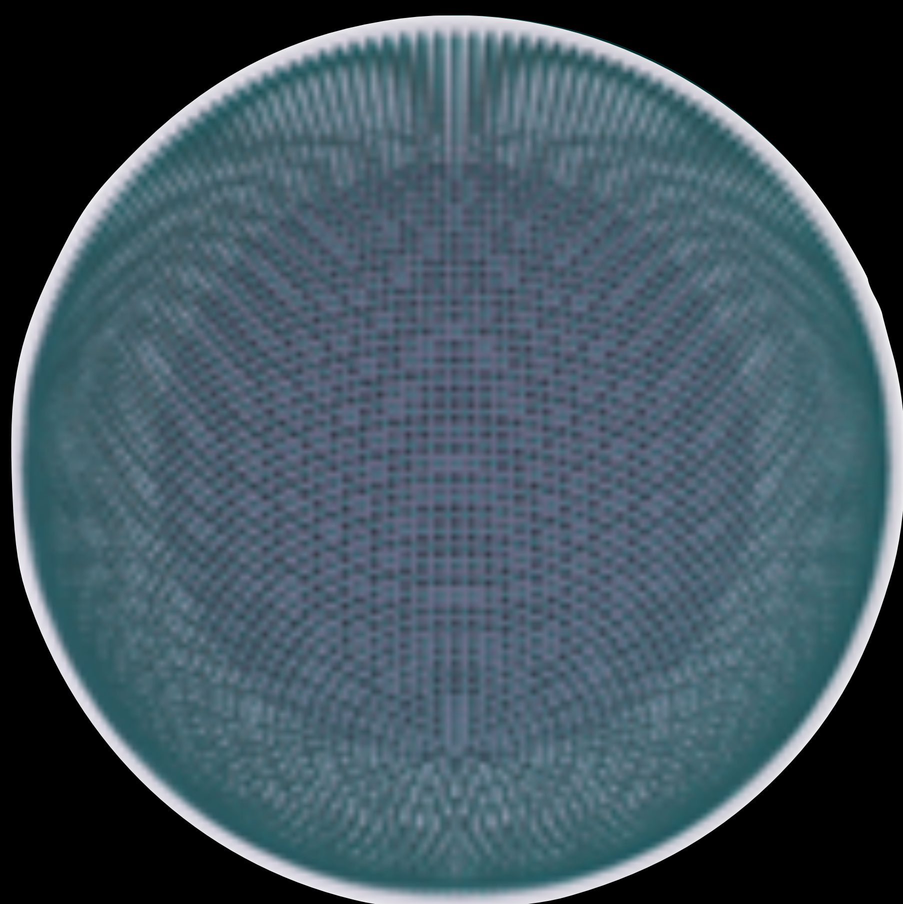


DATABASE (RATDB)
Detector geometry, PMT quantum efficiency curves, material optical properties DAQ parameters, calibration constants…
Simulation
Generator
Event Loop
Processor
e.g. ring reconstruction
Processor
e.g. PMT calibration
Do something to an event
Do something to an event
…
Populate new event
Data Structure
Updated Data Structure
Updated² Data Structure
i.e. DAQ format → RAT format
Producer
Real DAQ output file
Output TTree in ROOT file

Simulations

New geometries are defined quickly and easily

RAT geometry constructor is just a wrapper of the Geant4 tools, made flexible and user friendly



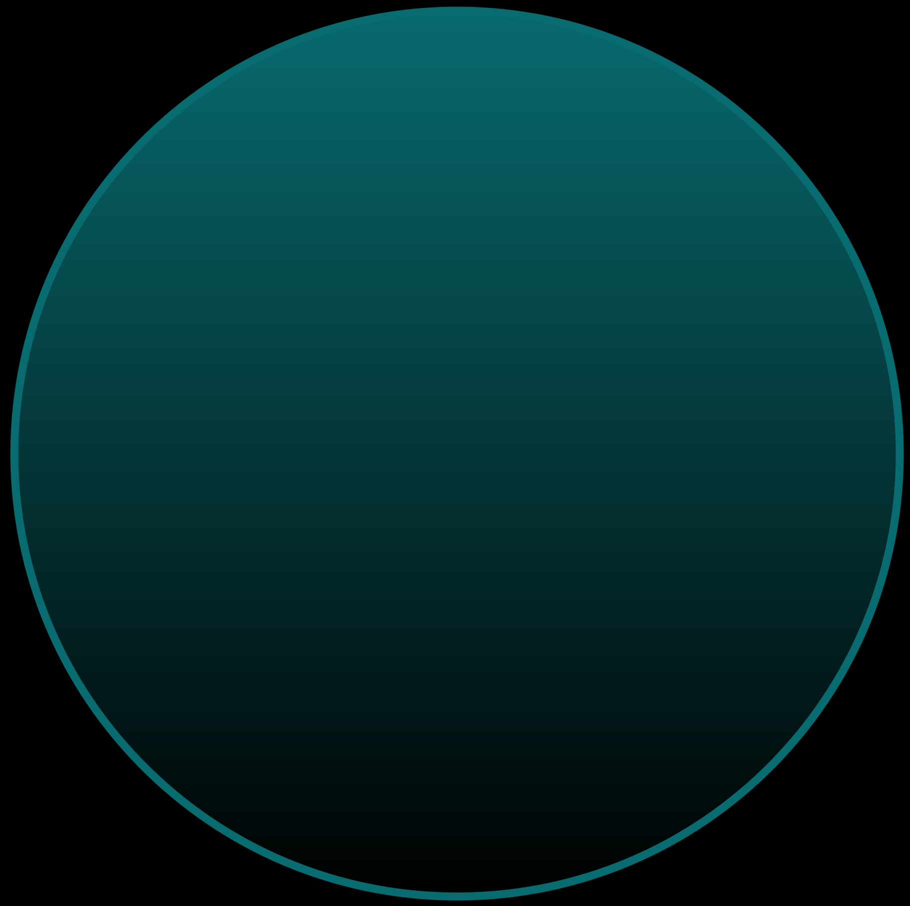
Spherical detector surrounded by PMTs

```
{
name: "GEO",
index: "world",
valid_begin: [0, 0],
valid_end: [0, 0],
mother: "", // world volume has no mother
type: "box",
size: [20000.0, 20000.0, 20000.0], // mm, half-length
material: "rock",
invisible: 1,
}

{
name: "GEO",
index: "water",
valid_begin: [0, 0],
valid_end: [0, 0],
mother: "world",
type: "sphere",
r_max: 9000.0,
position: [0.0, 0.0, 0.0],
material: "water",
color: [0.4, 0.4, 0.6, 0.05],
}

{
name: "GEO",
index: "target",
valid_begin: [0, 0],
valid_end: [0, 0],
mother: "water",
type: "sphere",
r_max: 6000.0,
position: [0.0, 0.0, 0.0],
material: "water",
color: [0.4, 0.4, 0.6, 0.05],
}

{
name: "GEO",
index: "pmts",
valid_begin: [0, 0],
valid_end: [0, 0],
mother: "water",
type: "pmtarray",
pmt_model: "r11780_hqe",
pmt_detector_type: "idpmt",
sensitive_detector: "/mydet/pmt/inner",
pos_table: "PMTINFO",
orientation: "point",
orient_point: [0.0, 0.0, 0.0],
}
```

FIN.

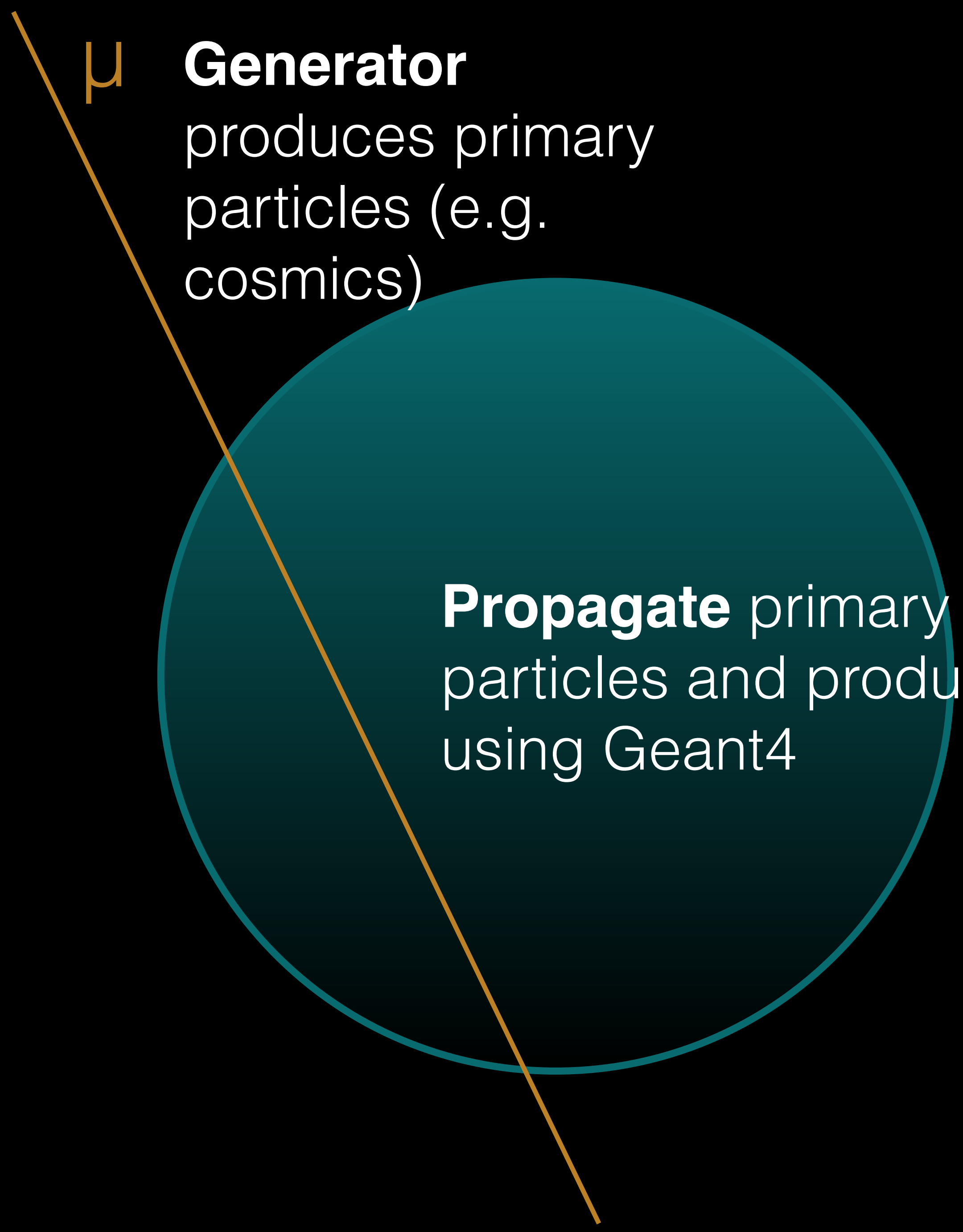
Simulations

AMARA: 'As Microphysical As Reasonably Achievable'



Simulations

AMARA: ‘As Microphysical As Reasonably Achievable’

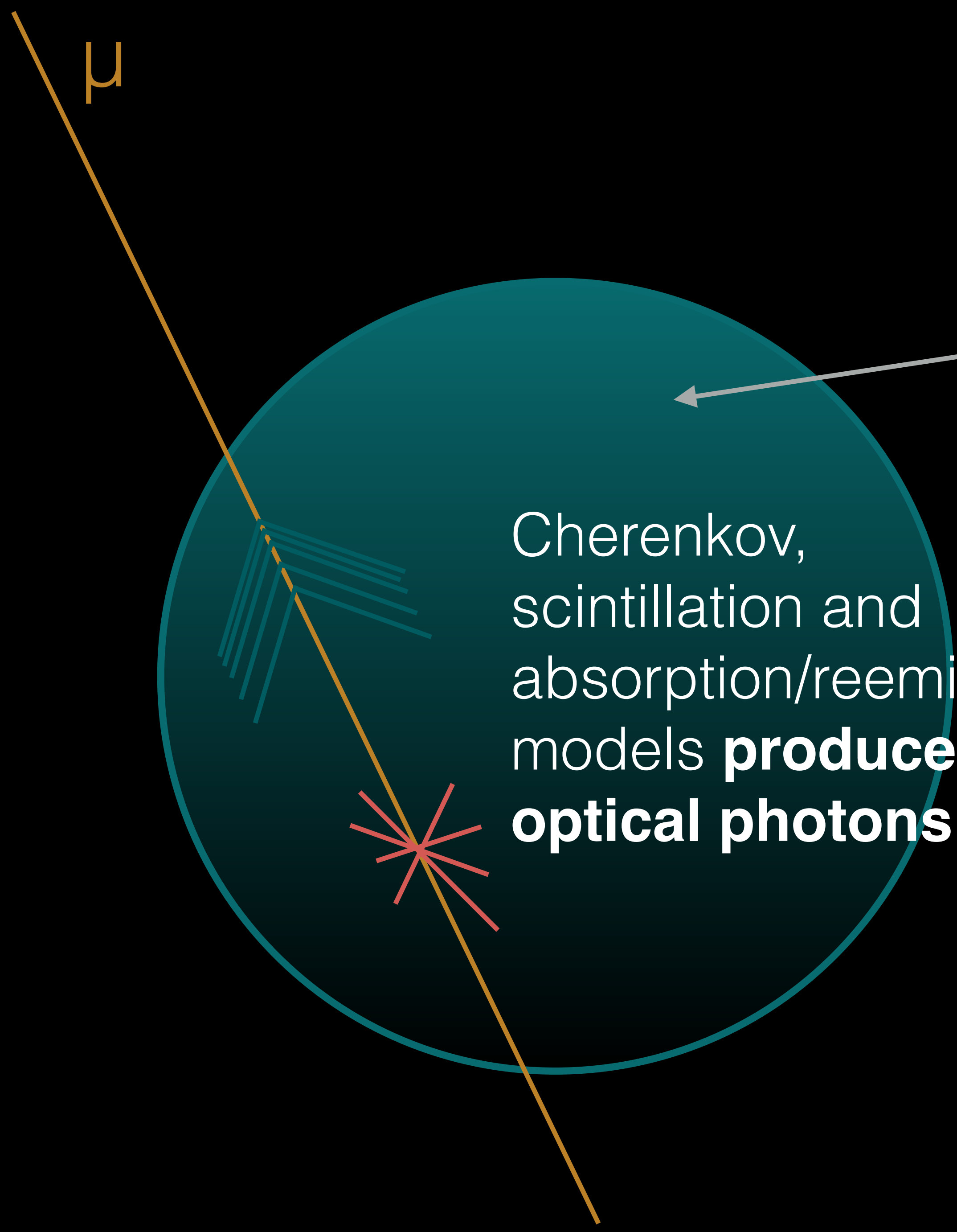


Generator produces primary particles (e.g. cosmics)

Propagate primary particles and products using Geant4

Simulations

AMARA: ‘As Microphysical As Reasonably Achievable’

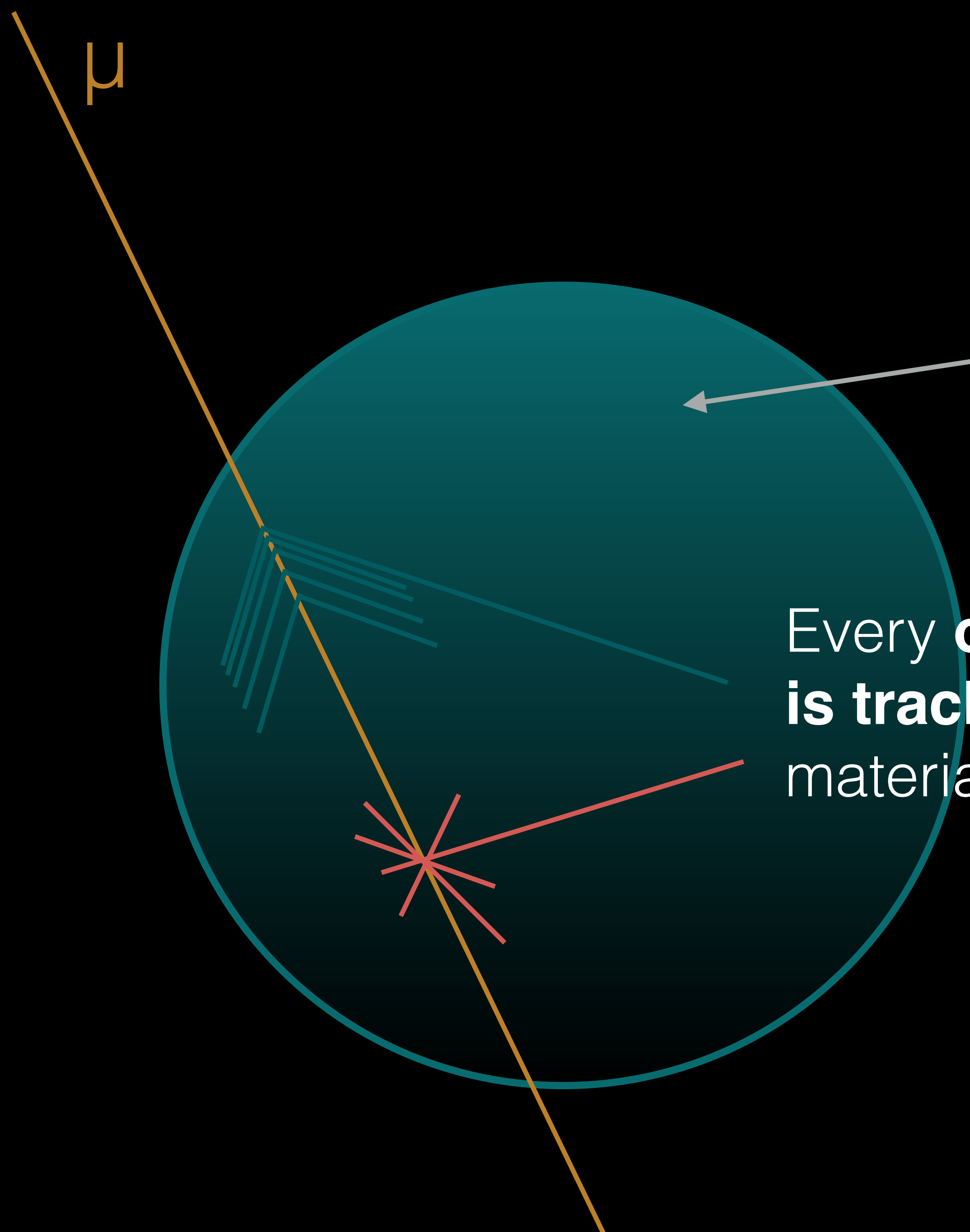


Models performances depend on the material properties defined in the detector geometry and **RATDB**

Cherenkov, scintillation and absorption/reemission models **produce optical photons**

Simulations

AMARA: ‘As Microphysical As Reasonably Achievable’

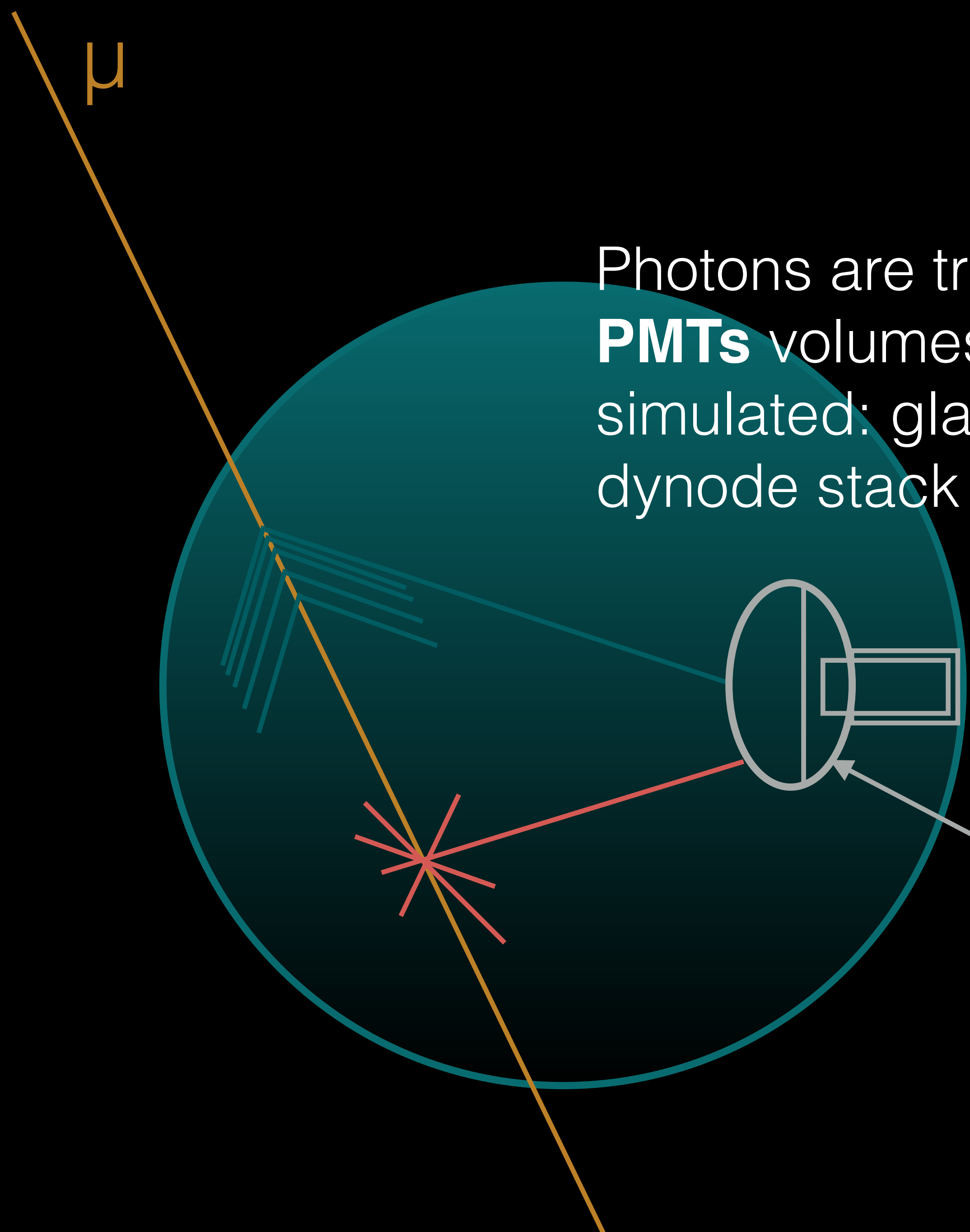


Optical properties of the detector materials are taken into account producing reflexion, refraction, scattering and absorption

Every **optical photon is tracked** in the material by Geant4

Simulations

AMARA: ‘As Microphysical As Reasonably Achievable’



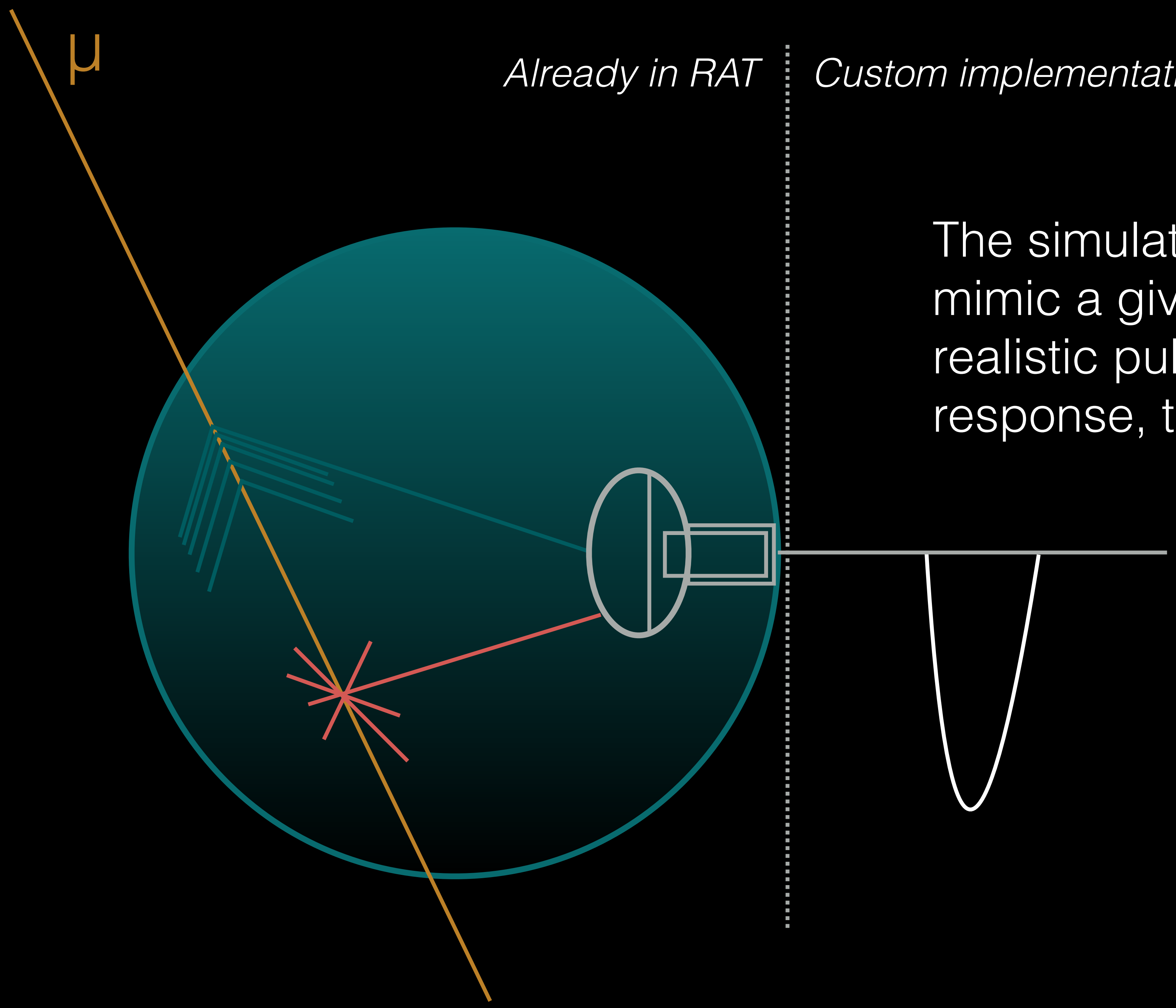
Photons are tracked **inside the PMTs** volumes, which are fully simulated: glass, PC, mirror and dynode stack

Eventually a photon will produce a PE with a given charge and time extracted from custom distributions

Charge and time distributions as well as quantum and collection efficiencies are taken into account and can be easily defined in **RATDB**

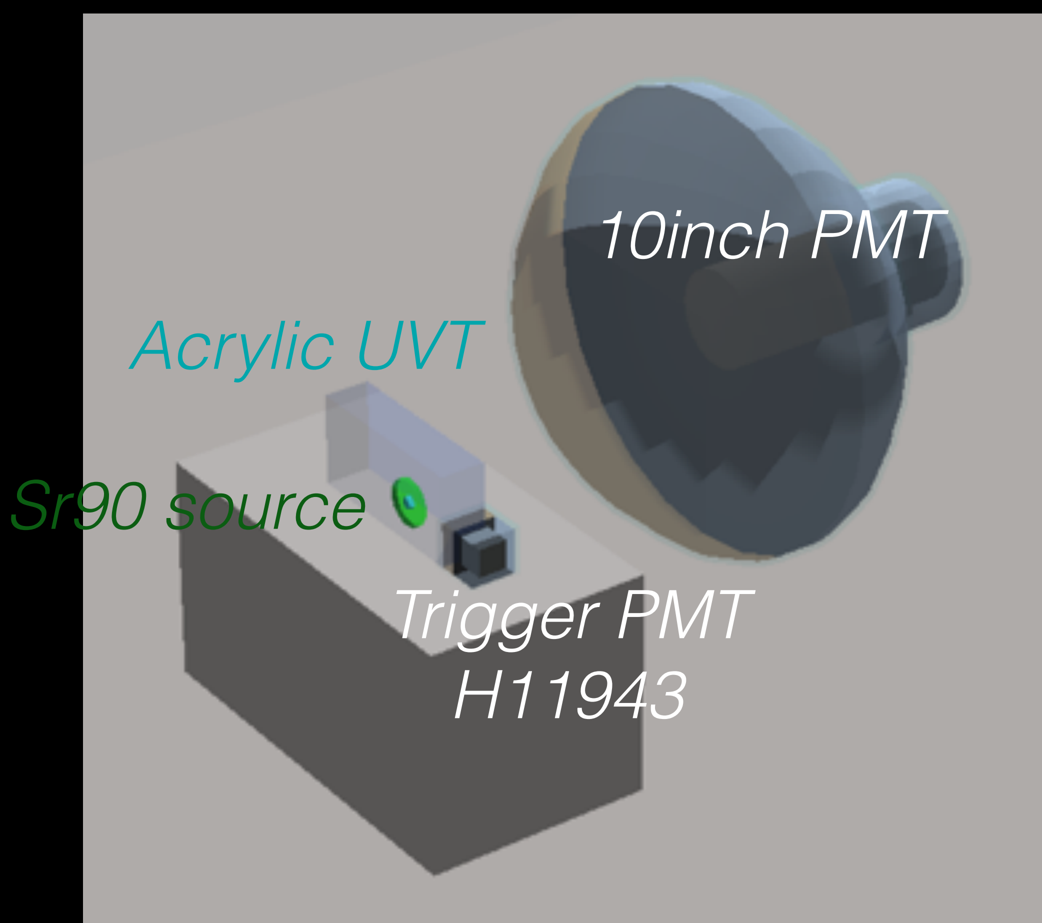
Simulations

AMARA: ‘As Microphysical As Reasonably Achievable’



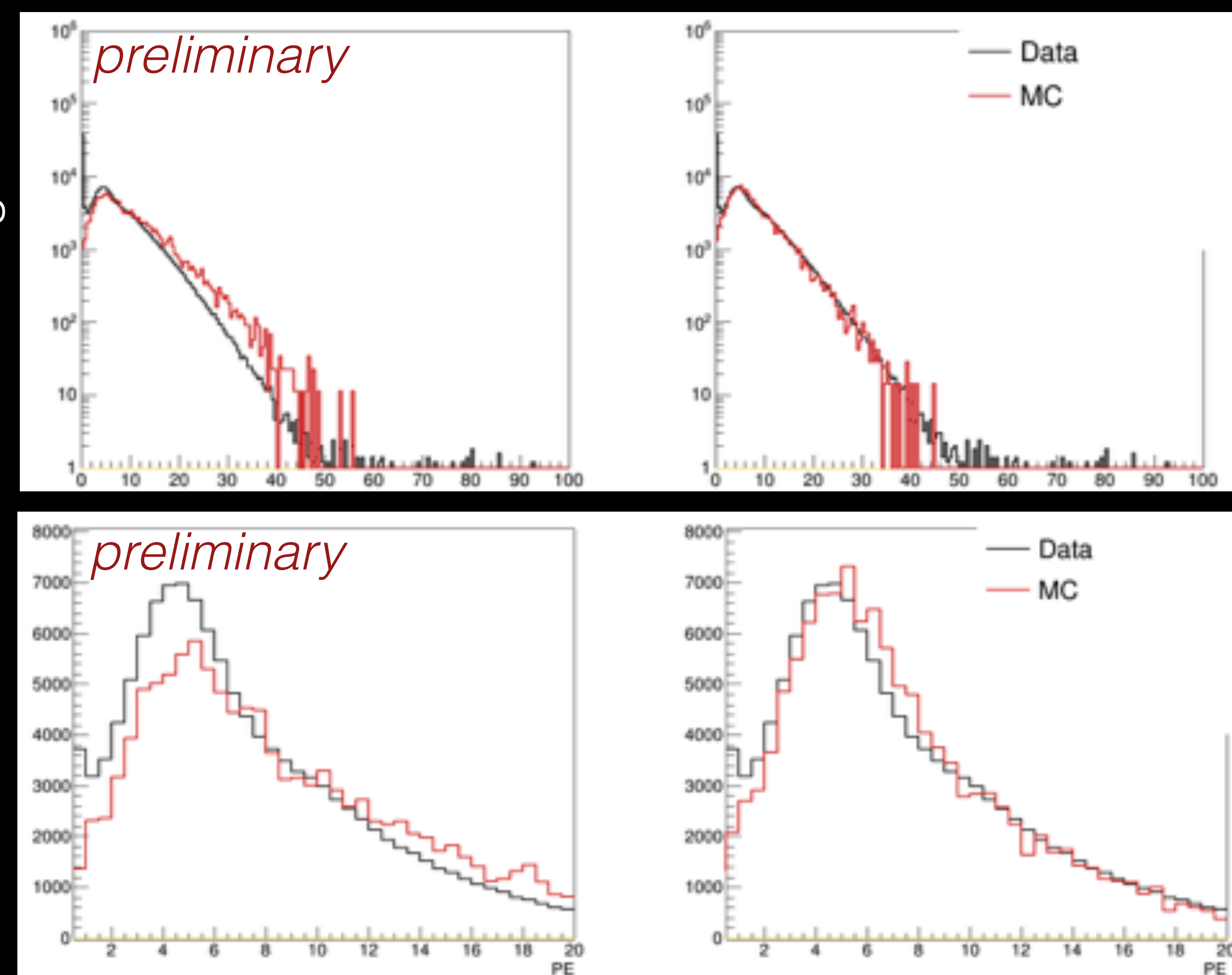
The simulation can be extended to mimic a given DAQ system: realistic pulses, electronic response, triggers, digitization, …

Simplest case:
THEIA RnD @Berkeley



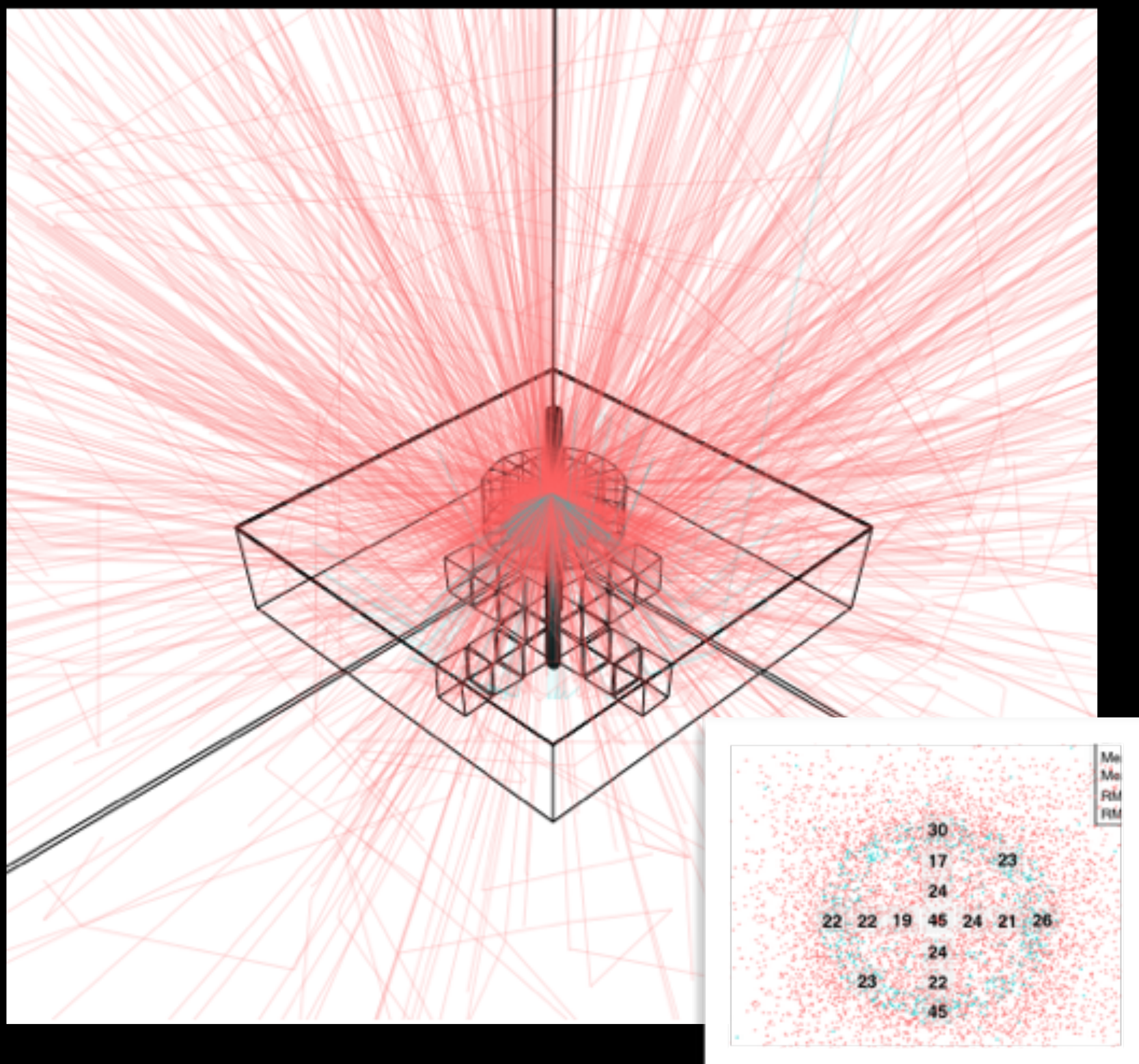
100% collection eff.   70% collection eff.

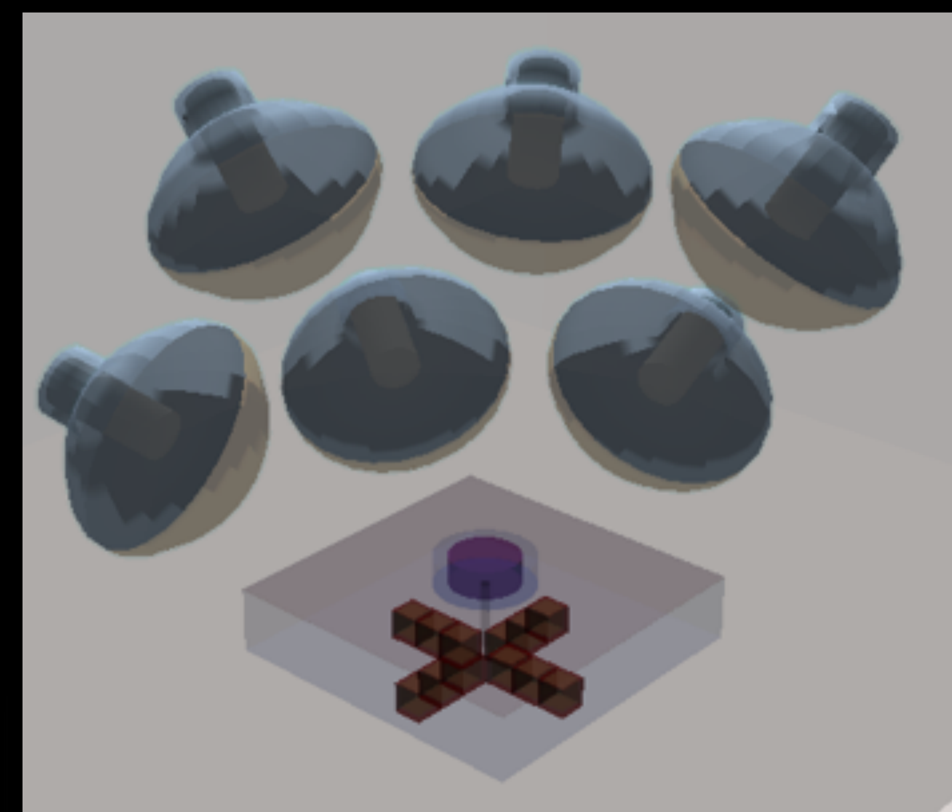
Log
Lineal



Charge distributions

A less simple case:
Theia RnD @Berkeley



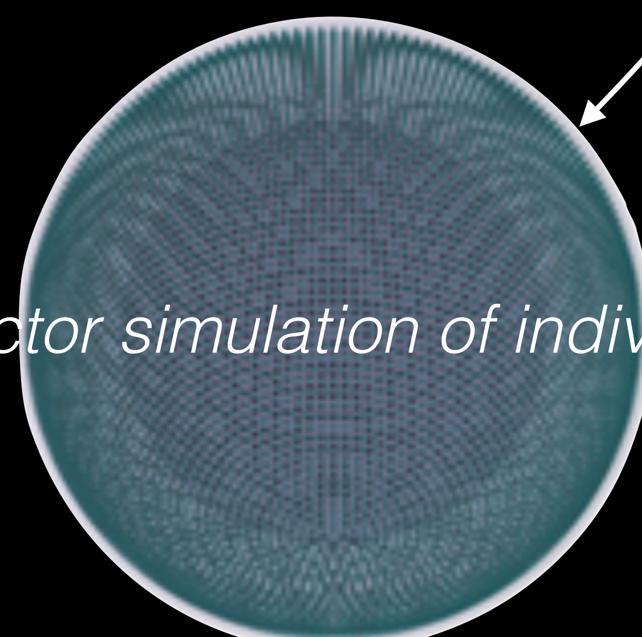


A complex case: SNO+

Bench-top studies determine material properties, test models, …

DATABASE (RATDB)
Detector geometry, PMT quantum efficiency curves, material optical properties DAQ parameters, calibration constants…

On site measurements provide calibration, detector state, …



Full detector simulation of individual runs

Simulation performances tested by SNO+ using SNO data and THEIA collaborators on RnD projects

RAT database: RATDB

- Tables in JSON format identified by names and indexes
- RAT provides support for interface with a CouchDB server fully compatible with the RATDB concept

Material optical properties

```
// -------------  General Environment ---------------

{
name: "OPTICS",
index: "cryostat_vacuum",
valid_begin : [0, 0],
valid_end : [0, 0],
RINDEX_option: "wavelength",
RINDEX_value1: [60.0, 800.0, ],
RINDEX_value2: [1.000001, 1.000001, ],
ABSLENGTH_option: "wavelength",
ABSLENGTH_value1: [60.0, 800.0, ],
ABSLENGTH_value2: [1.0e9, 1.0e9, ],
PROPERTY_LIST: ["RINDEX", "ABSLENGTH", ]
}

{
name: "OPTICS",
index: "air",
valid_begin : [0, 0],
valid_end : [0, 0],
RINDEX_option: "wavelength", //60 nm is bogus to prevent G4 from complaining when VUVs hit the air
RINDEX_value1: [60.0, 200.0, 300.0, 500.0, 700.0, 800.0, ], //with neck some vuvs reach the air
RINDEX_value2: [1.0003236, 1.0003236, 1.0002915, 1.0002792, 1.0002763, 1.0002756, ],
ABSLENGTH_option: "wavelength", //60 nm is bogus to prevent G4 from complaining when VUVs hit the air
ABSLENGTH_value1: [60.0, 200.0, 300.0, 330.0, 500.0, 600.0, 770.0, 800.0, ],
ABSLENGTH_value2: [1.0e6, 1.0e6, 1.0e6, 1.0e6, 1.0e6, 1.0e6, 1.0e6, 1.0e6, ],
PROPERTY_LIST: ["RINDEX", "ABSLENGTH", ]
}

{
name: "OPTICS",
index: "water",
valid_begin : [0, 0],
```

PMT geometry information

```
{
name: "PMT",
index: "r5912",
valid_begin: [0,0],
valid_end: [0,0],

construction: "toroidal",

dynode_material: "stainless_steel",
glass_material: "glass",
pmt_vacuum_material: "pmt_vacuum", // dilu
photocathode_surface: "photocathode",
mirror_surface: "mirror",
dynode_surface: "stainless_steel",

shape: "torus", //torus, cube
dynode_radius: 27.5, // mm
dynode_top:   -30.0, // mm
wall_thickness: 3.0, // mm

z_edge:   [ 75.00, 53.06,   0.00, -53.06, -73.86,  -85.00, -215.00],
rho_edge: [  0.00, 72.58, 101.00,  72.58,  44.32,   42.00, 42.00],
z_origin: [-56.00,  0.00,   0.00,  56.00, -85.00, -215.00],
}

{
name: "PMT",
index: "r1408",
valid_begin: [0,0],
valid_end: [0,0],

construction: "toroidal",

dynode_material: "stainless_steel",
glass_material: "glass",
pmt_vacuum_material: "pmt_vacuum", // dilute air
photocathode_surface: "photocathode_R1408",
mirror_surface: "mirror",
dynode_surface: "stainless_steel",

shape: "torus", //torus, cube
dynode_radius: 27.5, // mm
```

Quick start

- Install ROOT 5, Geant4.10 and SCons
- Get the code from gitHub: $ `git clone github@github.com/rat-pac rat-pac`
- `$./configure $source env.sh &scons`
- Run a rat session or invoke a rat script:

Electron gun on spherical detector

Geometry definition

Event loop

Processors

ROOT output

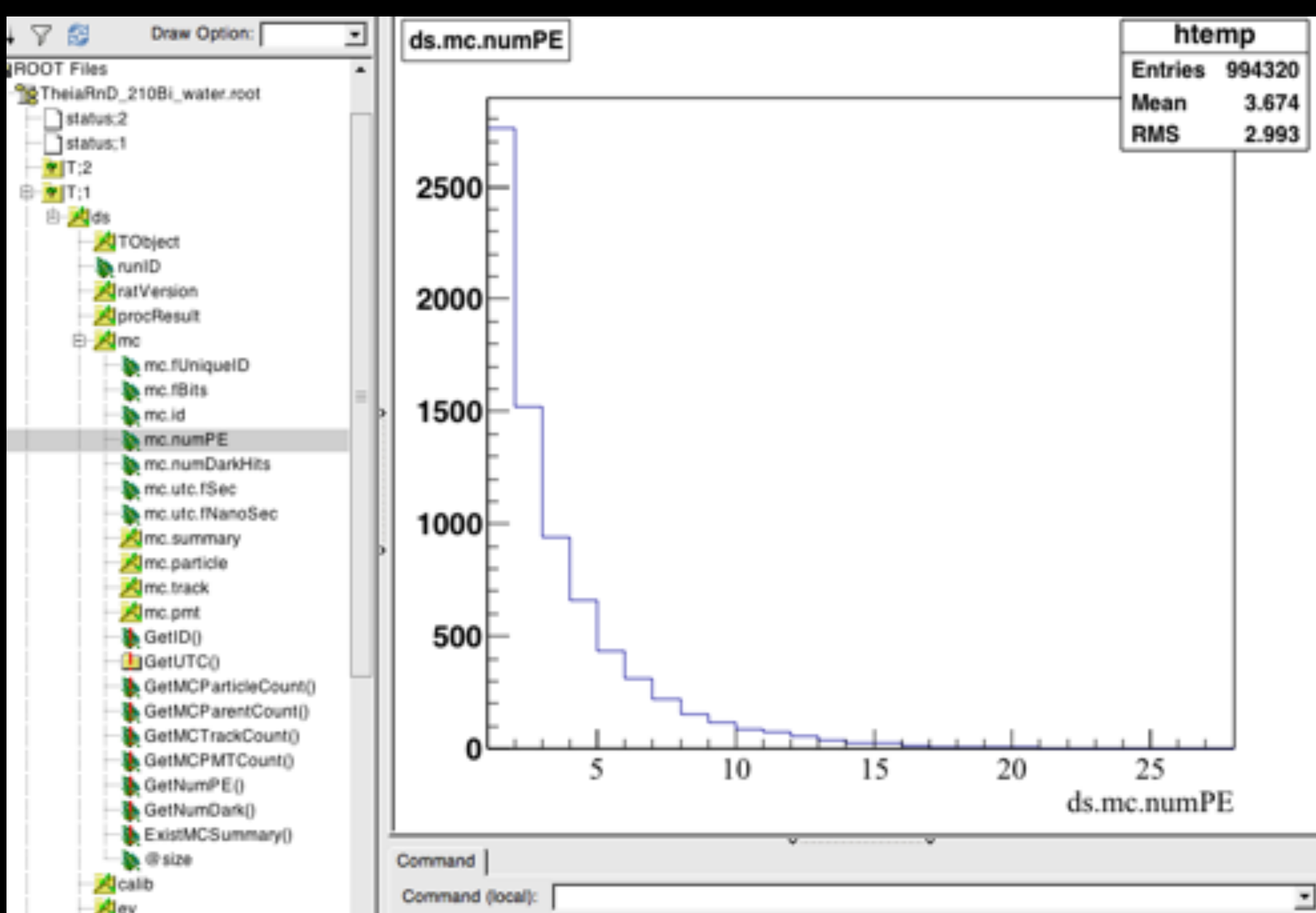
MC generation

```
/glg4debug/glg4param omit_muon_processes  1.0
/glg4debug/glg4param omit_hadronic_processes  1.0

/rat/db/set DETECTOR experiment "sphere"
/rat/db/set DETECTOR geo_file "sphere/sphere.geo"

/run/initialize

# BEGIN EVENT LOOP
/rat/proc simpledaq
/rat/proc count
/rat/procset update 10
/rat/proc fitcentroid

/rat/proclast outroot
/rat/procset file "sphere_betas.root"
#END EVENT LOOP

#/tracking/storeTrajectory 1

/generator/add combo gun:fill:poisson
/generator/vtx/set e- 0 0 0 10.0
/generator/pos/set 0 0 0

/run/beamOn 100
```

FIN.

Quick start

- Install ROOT 5, Geant4.10 and SCons
- Get the code from gitHub: $ `git clone github@github.com/rat-pac rat-pac`
- `$./configure $source env.sh &scons`
- Run a rat session or invoke a rat script:



Electron gun on spherical detector

```
/glg4debug/glg4param omit_muon_processes  1.0
/glg4debug/glg4param omit_hadronic_processes  1.0

/rat/db/set DETECTOR experiment "sphere"
/rat/db/set DETECTOR geo_file "sphere/sphere.geo"

/run/initialize

# BEGIN EVENT LOOP
/rat/proc simpledaq
/rat/proc count
/rat/procset update 10
/rat/proc fitcentroid

/rat/proclast outroot
/rat/procset file "sphere_betas.root"
#END EVENT LOOP

#/tracking/storeTrajectory 1

/generator/add combo gun:fill:poisson
/generator/vtx/set e- 0 0 0 10.0
/generator/pos/set 0 0 0

/run/beamOn 100
```

Geometry definition (lines 4–5)

Event loop (lines 7–17)

Processors (lines 10–13)

ROOT output (lines 15–16)

MC generation (lines 21–23)

FIN.

Development

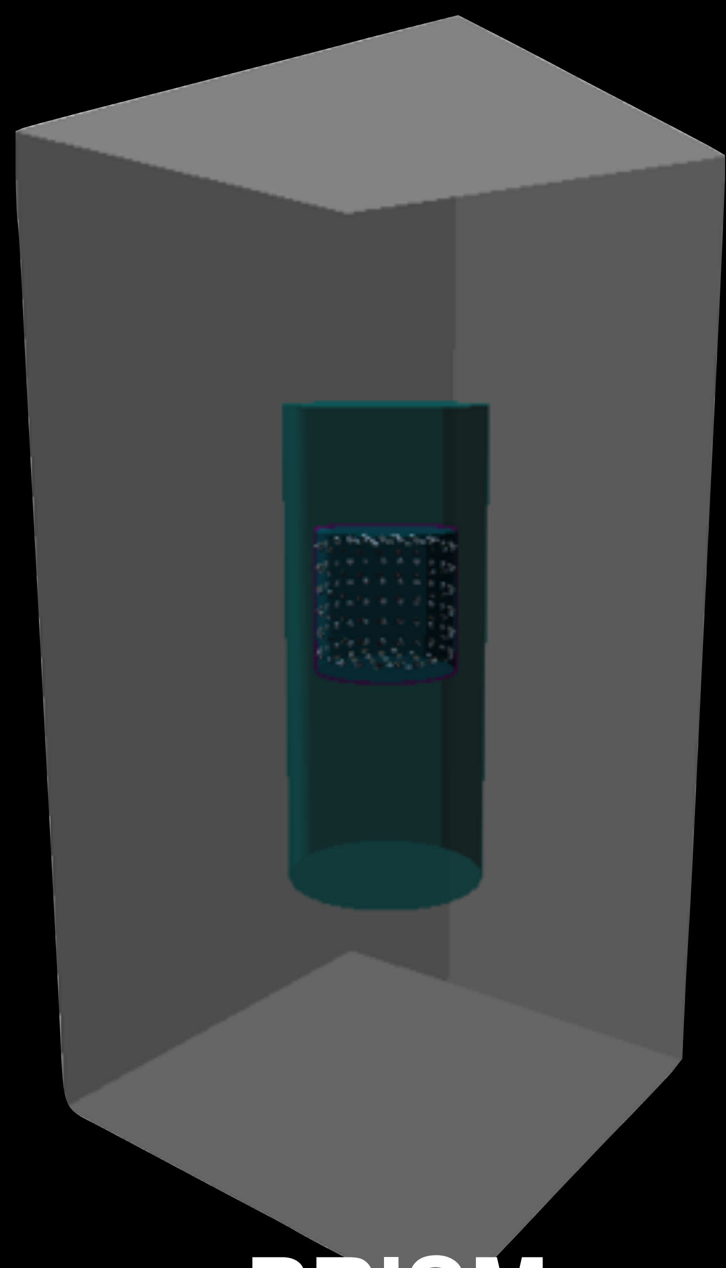
- Open source, version controlled on gitHub (*https://github.com/rat-pac/rat-pac*)
- “I don’t know how to use gitHub!” — *http://rat.readthedocs.org/en/latest/_downloads/ratpac_github.pdf*
- Continuous development by THEIA collaborators (and potentially ANNIE)
- Documented in http://rat.readthedocs.org/en/latest/
- Contact me (*jcaravaca@berkeley.edu*) or Nuno Barros (*nfbarros@hep.upenn.edu*) if you are interested in collaborating

Summary

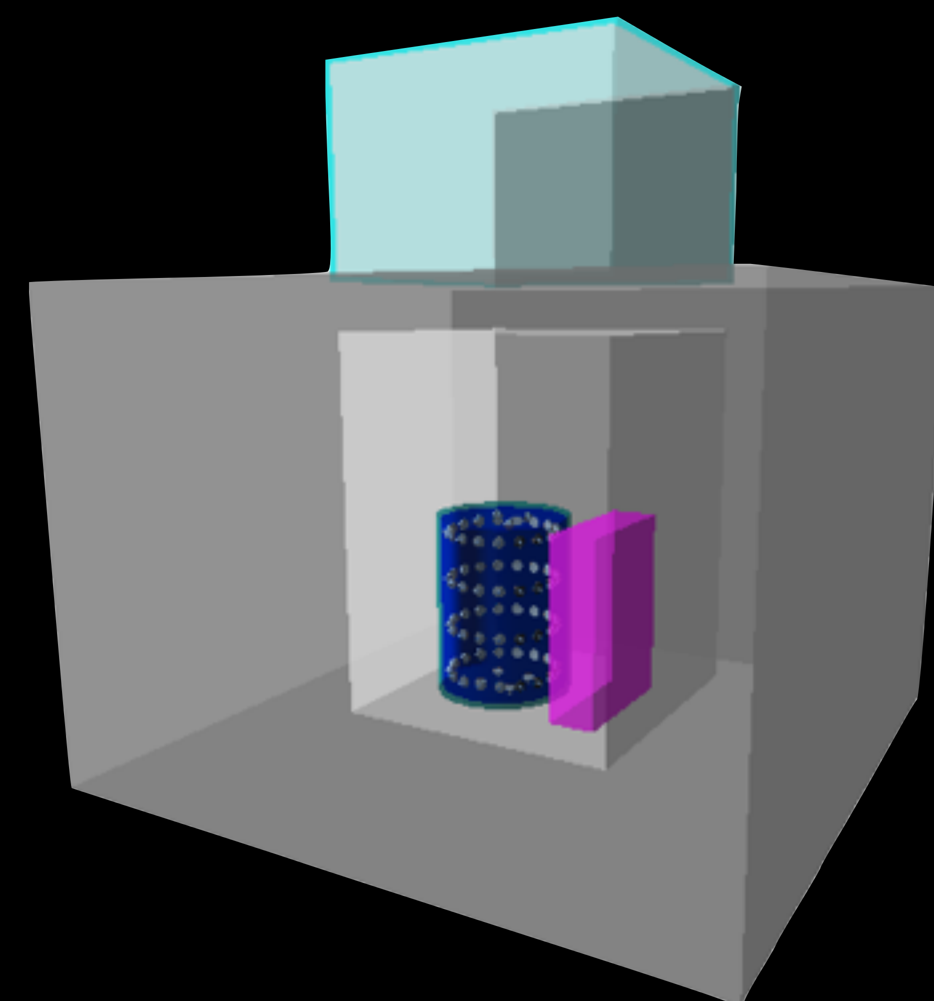
- RAT-PAC is a simulation/analysis toolkit that just works out of the box
- The main features that implements are a **data structure** in a TTree, an easily scalable **database** in JSON format and user friendly tools for **simulating geometry and physics**
- Main physics processed included: Cherenkov, scintillation and Geant4 processes
- Quick start and soft learning curve

Your favorite detector in RAT-PAC

(Made on the flight)



vPRISM



ANNIE

BACKUP

Scintillation model: GLG4Scint

- At each track step:
  - Get track energy loss dE/dx
  - Apply Birk's law to get deposited energy after quenching

$$dE_{\text{quench}} = \frac{dE}{1 + B \times dE/dx}$$

  - Times the light yield of the material gives you the total number of produced scintillation photons
  - The actual number of produced photons are extracted from a poisson distribution N
  - N photons are created isotropically