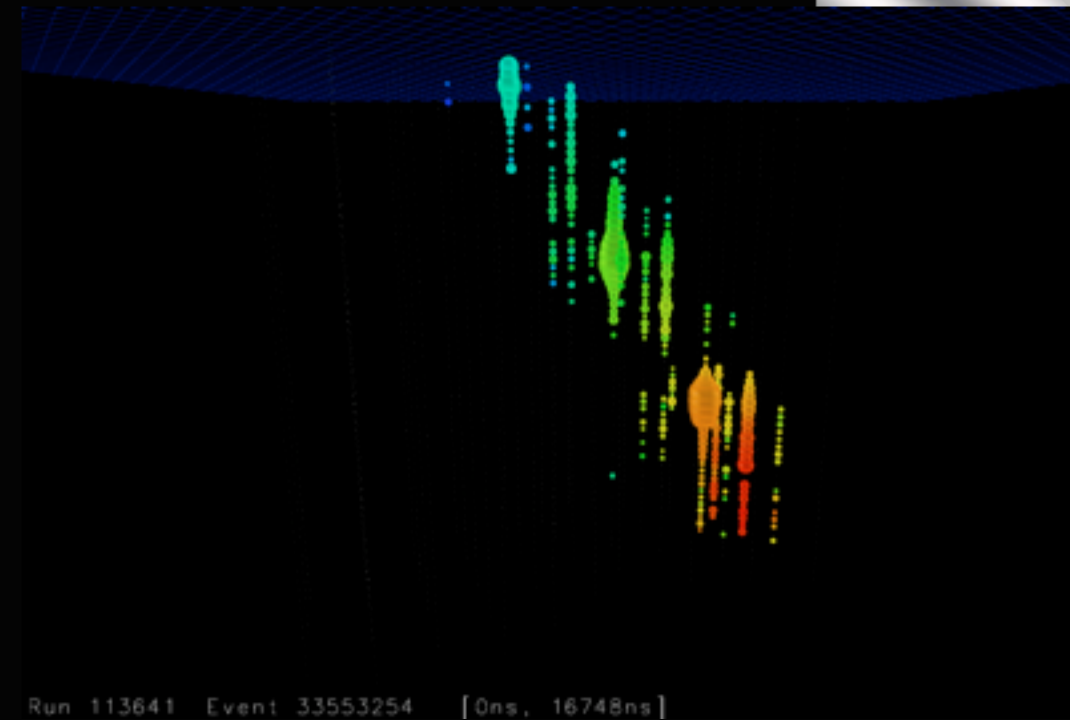
i3SIMulation

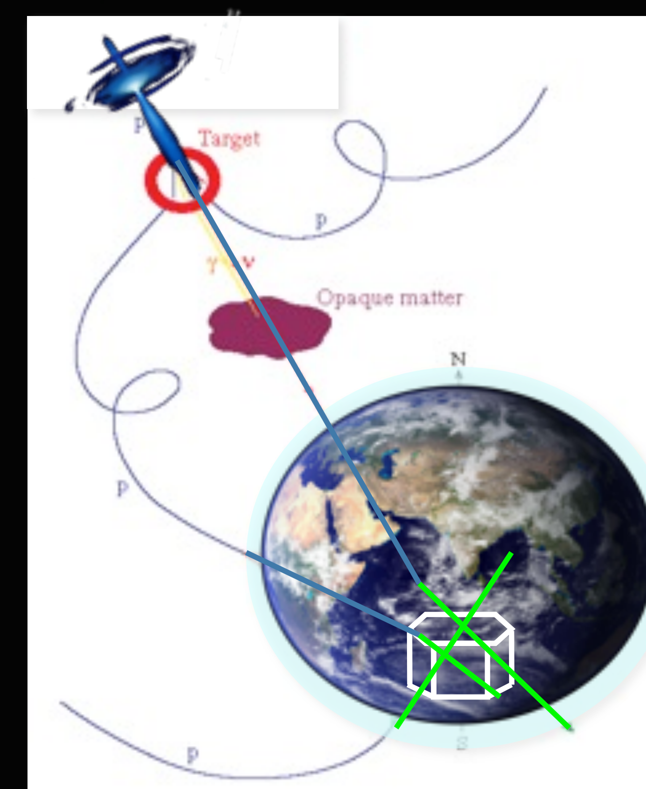
IceCube Bootcamp 2012

Juan Carlos Díaz Vélez

motivation

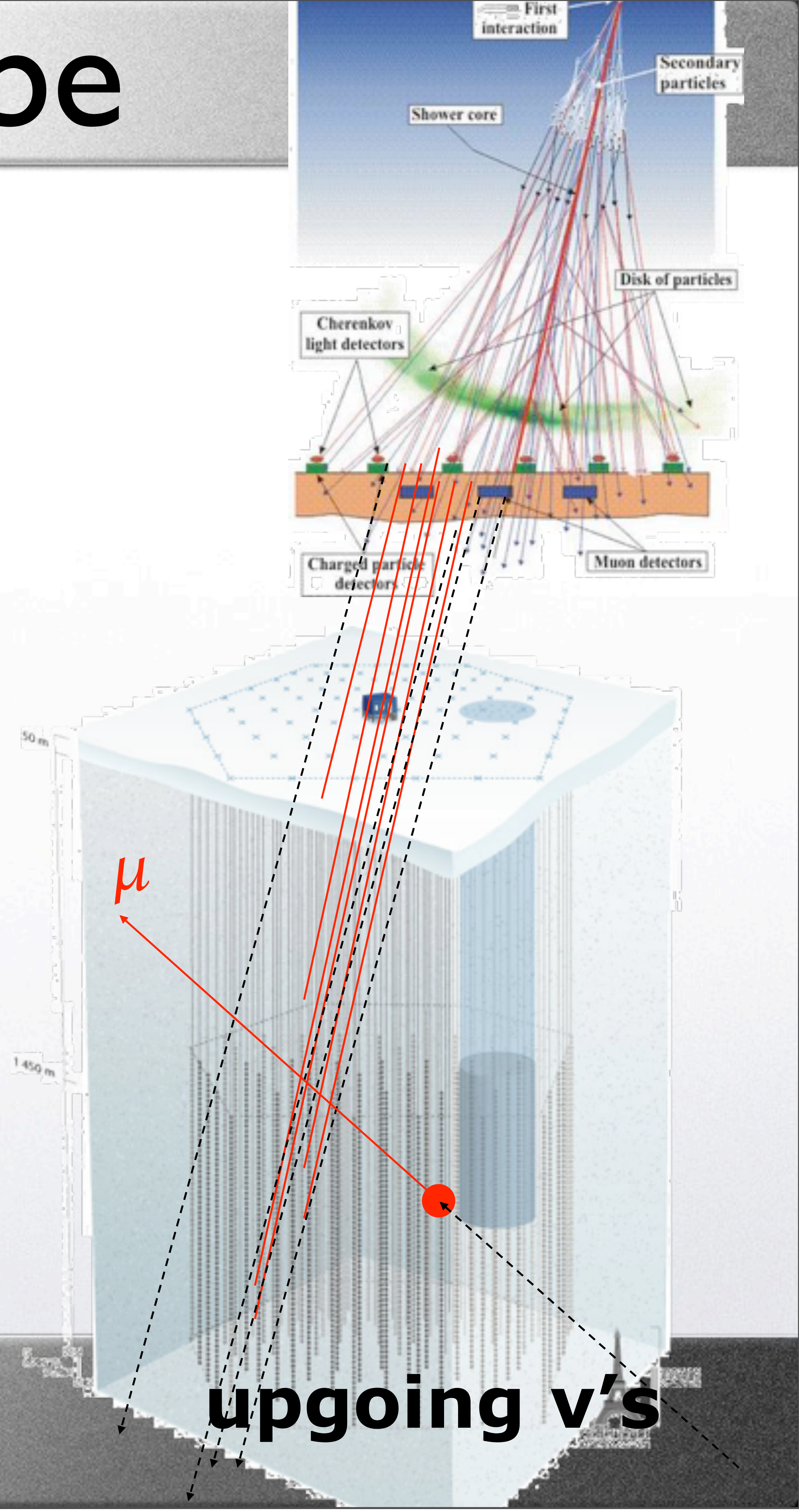


1. need to understand our detector
2. test reconstruction algorithms
3. understand the backgrounds in our analyses
1. atmospheric muons
2. atmospheric neutrinos

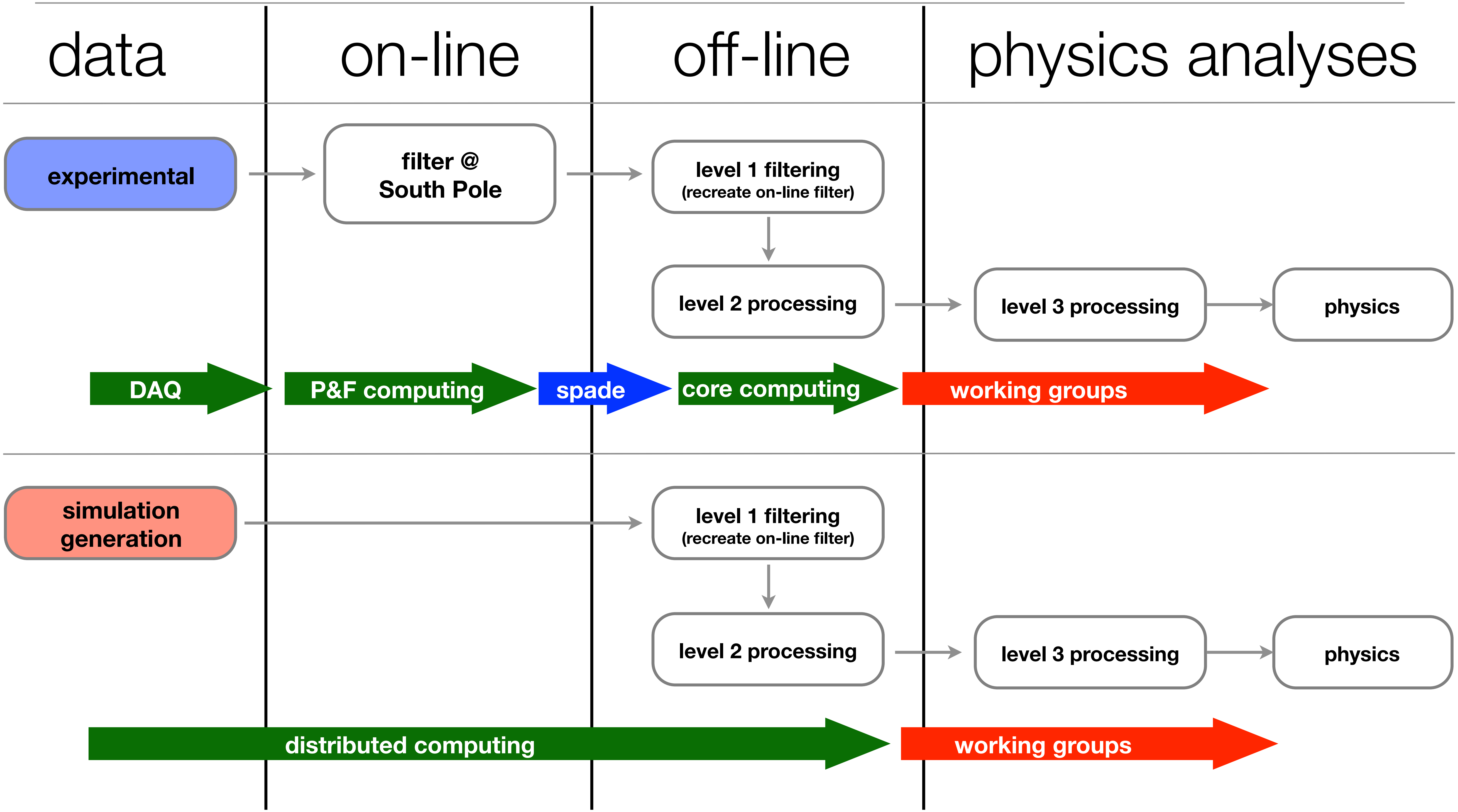


events in icecube

- Air shower detection @ surface
- Penetrating muon detection in deep ice
- Events dominated by cosmic ray muons : 10^6 μ for every ν that interacts in IceCube
- Upgoing ν's (atmospheric)



flow of experimental and simulation data



Simulation

tree<I3Particle>

(direction, position, energy, type)

OM, vector<I3MCHit>

(photoelectrons)

OM, vector<PMTWaveform>

(pe PMT response waveforms)

OM, vector<DOMLaunch>

(digitized PMT response waveforms)

Reconstruction

I3Particle

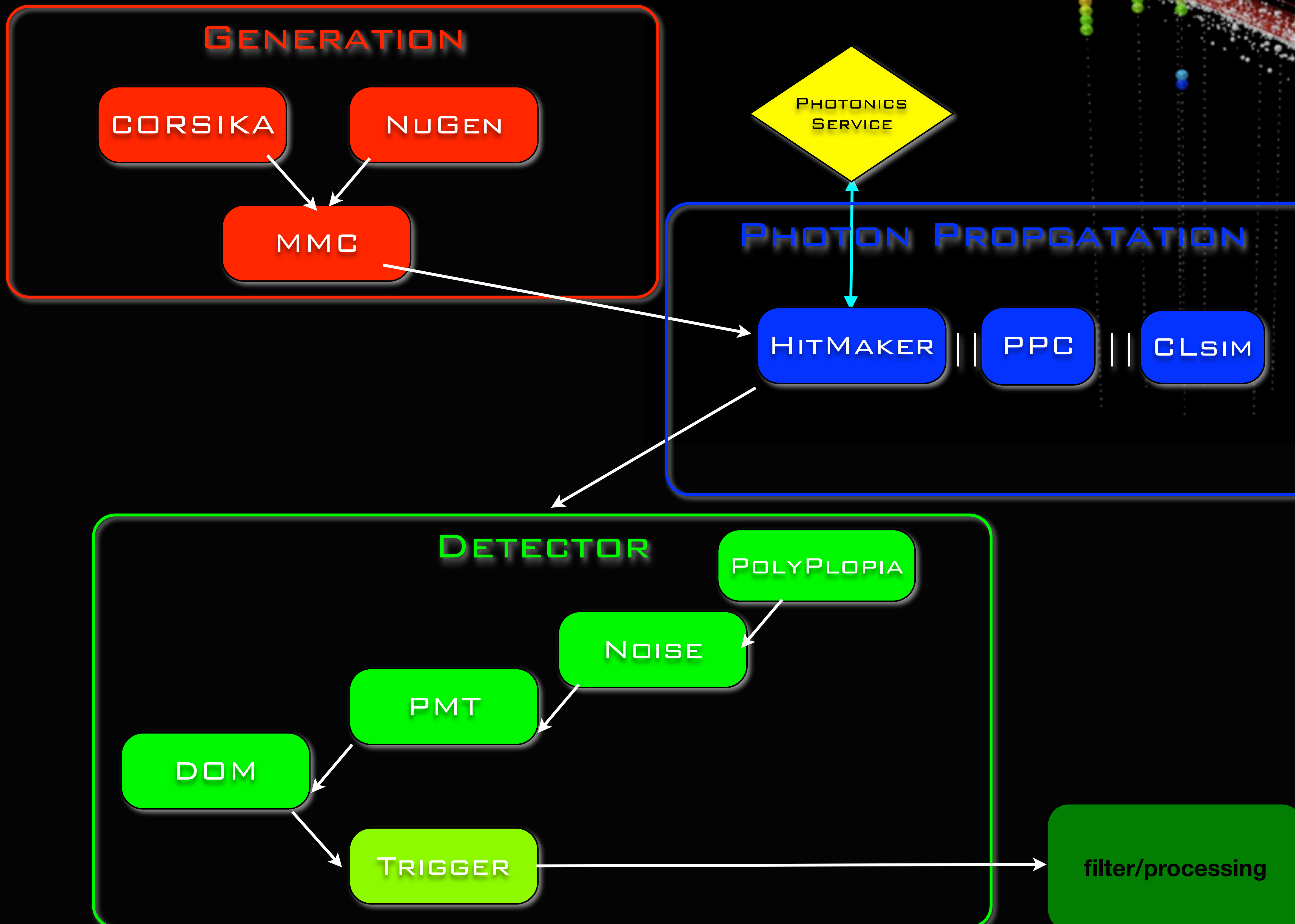
OM, vector<I3RecoPulse>

(pe pulses)

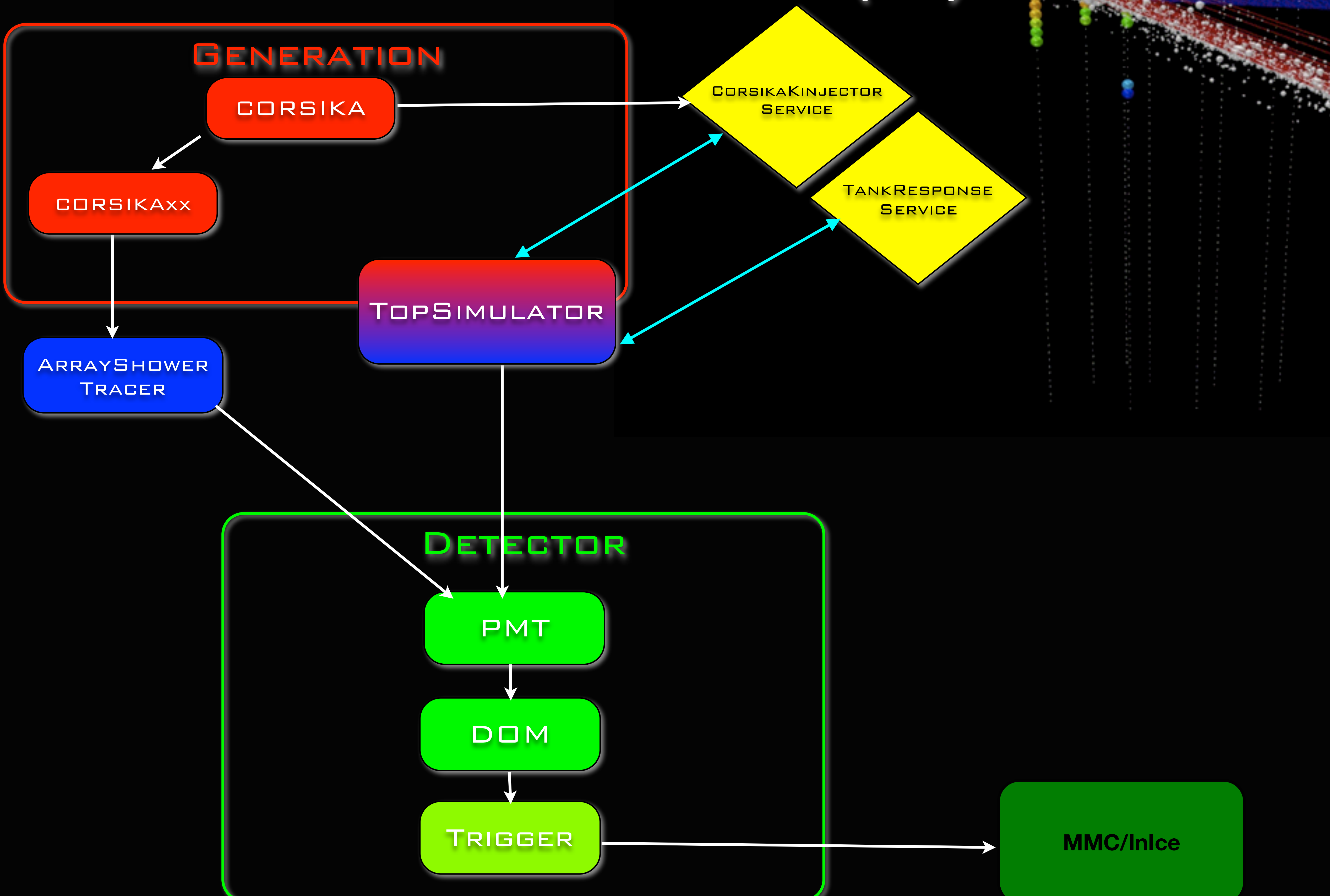
OM, vector<DOMLaunch>

(digitized PMT response waveforms)

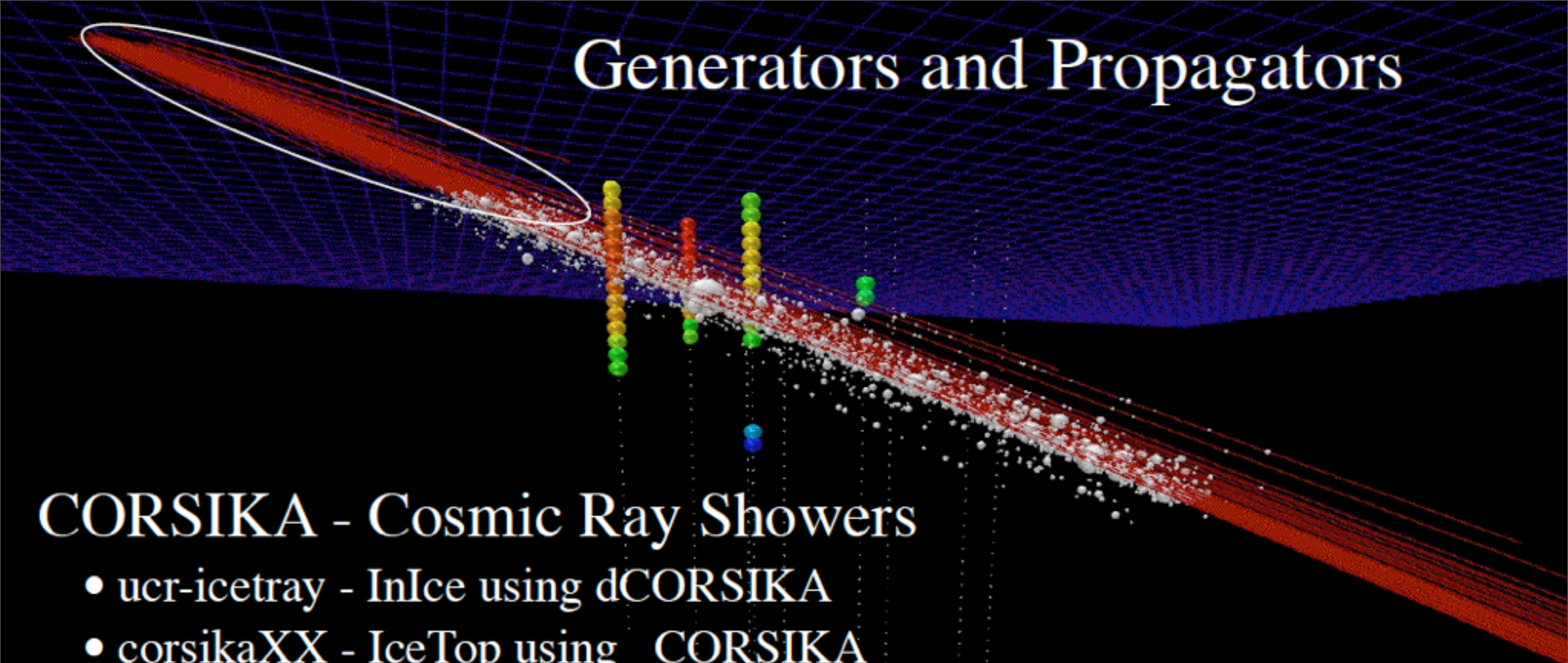
simulaton chain



simulaton chain (IT)



Generators and Propagators



CORSIKA - Cosmic Ray Showers

- ucr-icetray - InIce using dCORSIKA
- corsikaXX - IceTop using CORSIKA

Neutrinos - neutrino-generator, GENIE

Test Generators

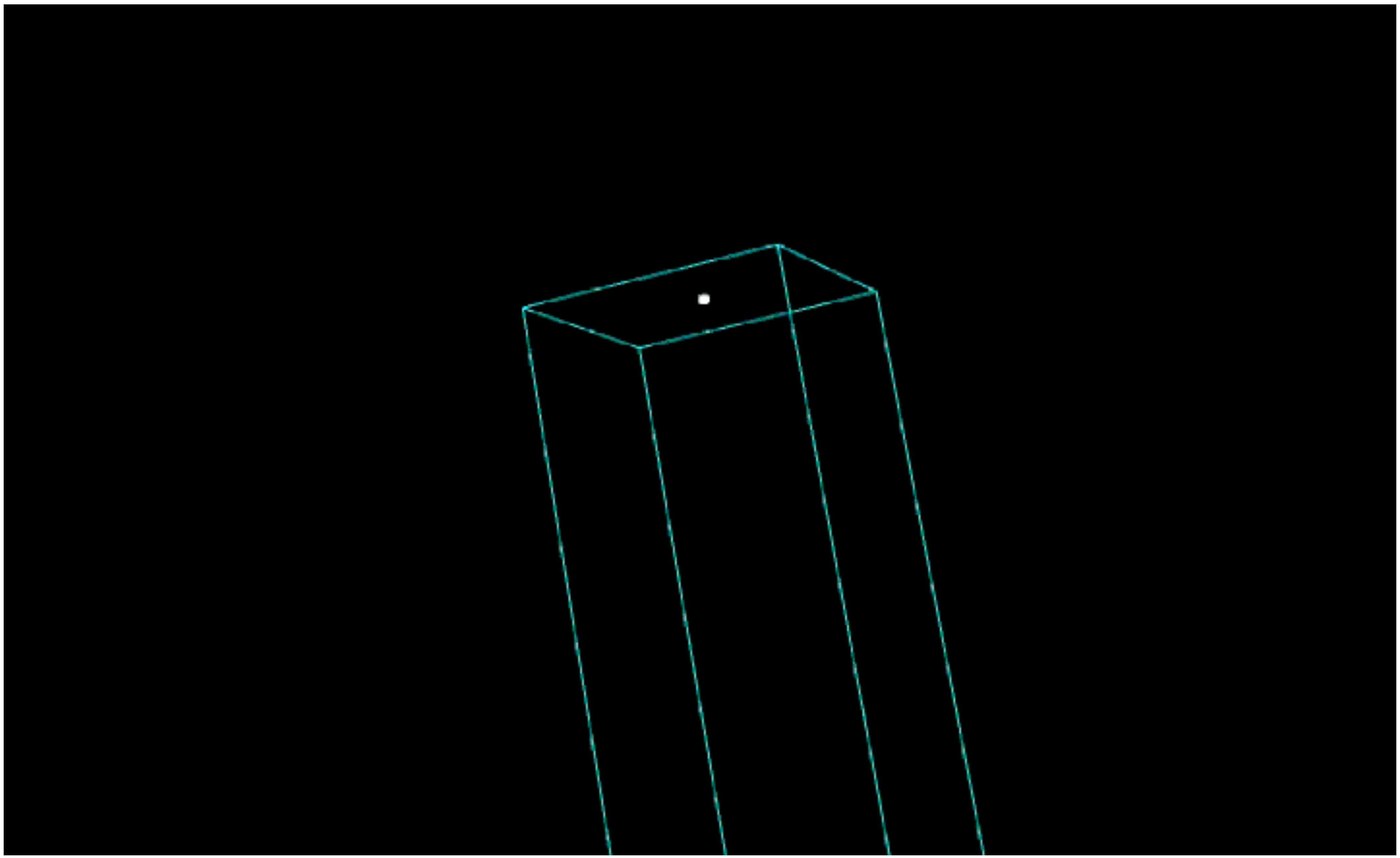
- simple-injector - injects **any** type of particle

Propagators

- MMC/mmc-icetray
- JULIeT/juliet-interface
- PROPOSAL (C++ implemetation of MMC)

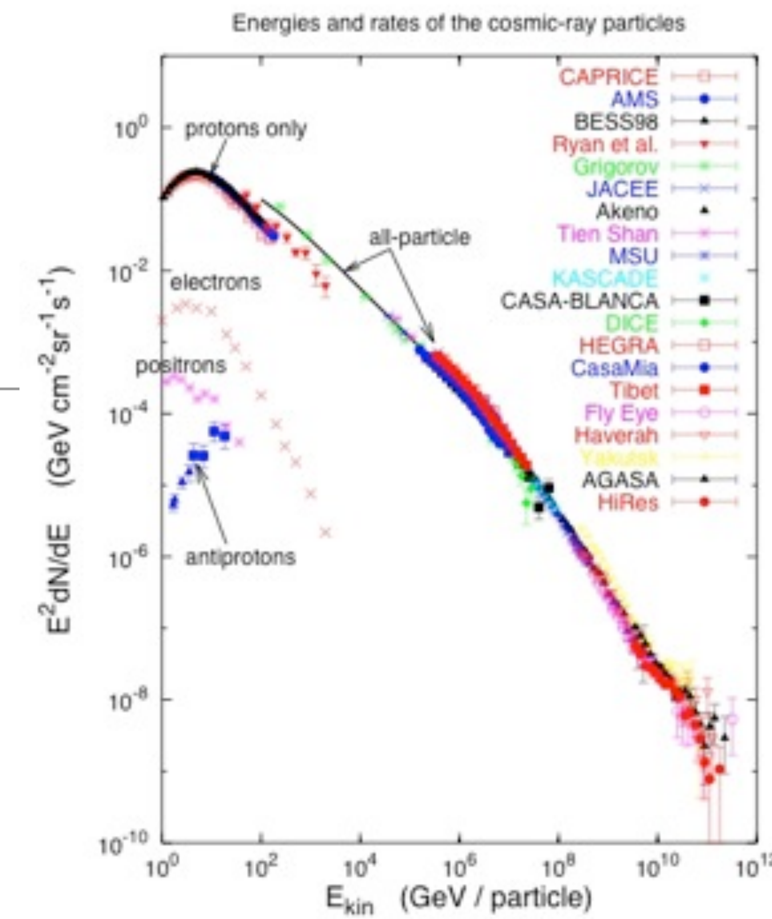
generators : CORSIKA

(**CO**smic **R**ay **SI**mulations for **KA**scade)



generators : CORSIKA

(**CO**smic **R**ay **SI**mulations for **KA**scade)



- FORTRAN code developed for KASCADE
  - dCorsika: IceCube-specific modifications
  - ucr-icetray: IceTray reader for dCorsika f2k
  - corsikaXX: IceTray reader for CORSIKA files
- poly-gonato model (fails > 10-100 PeV & @ horizon)

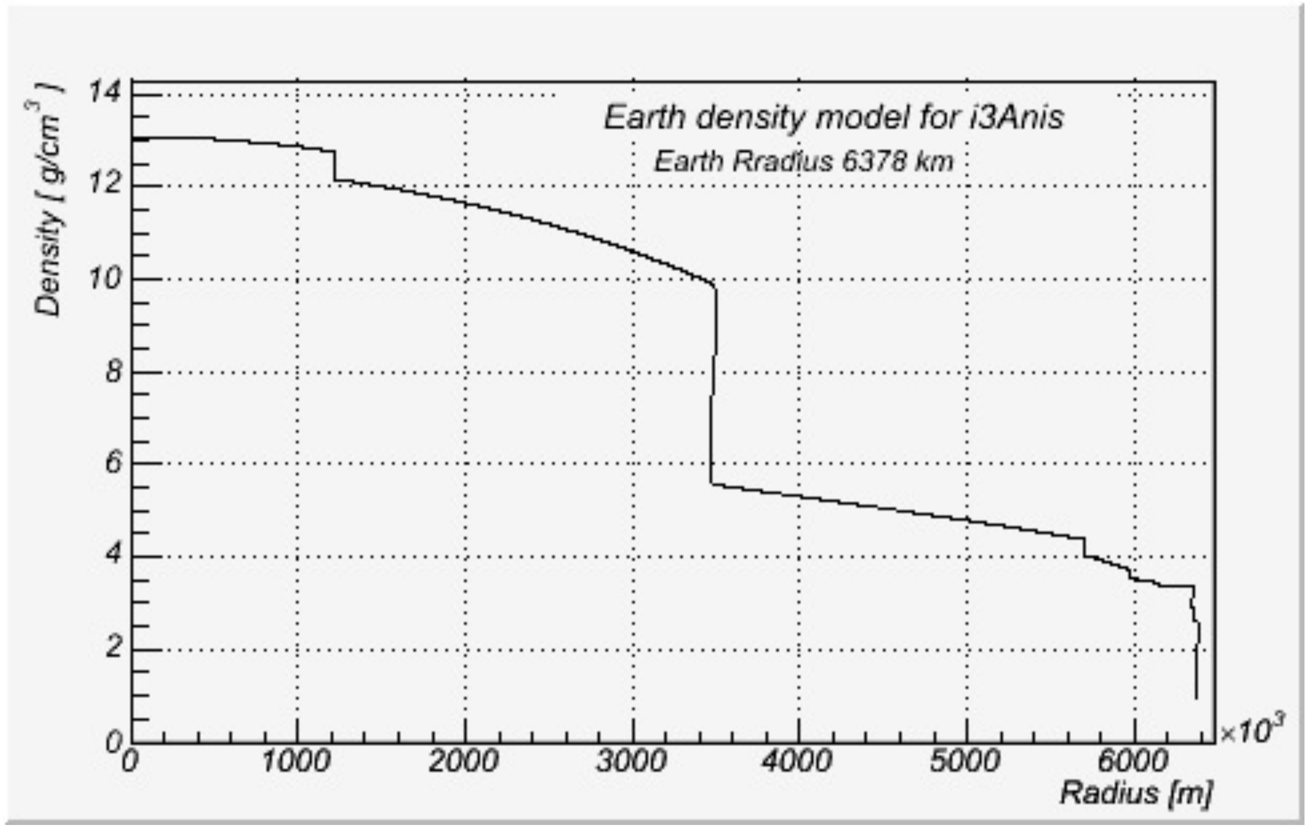
➡ weighted events : artificially flat spectrum

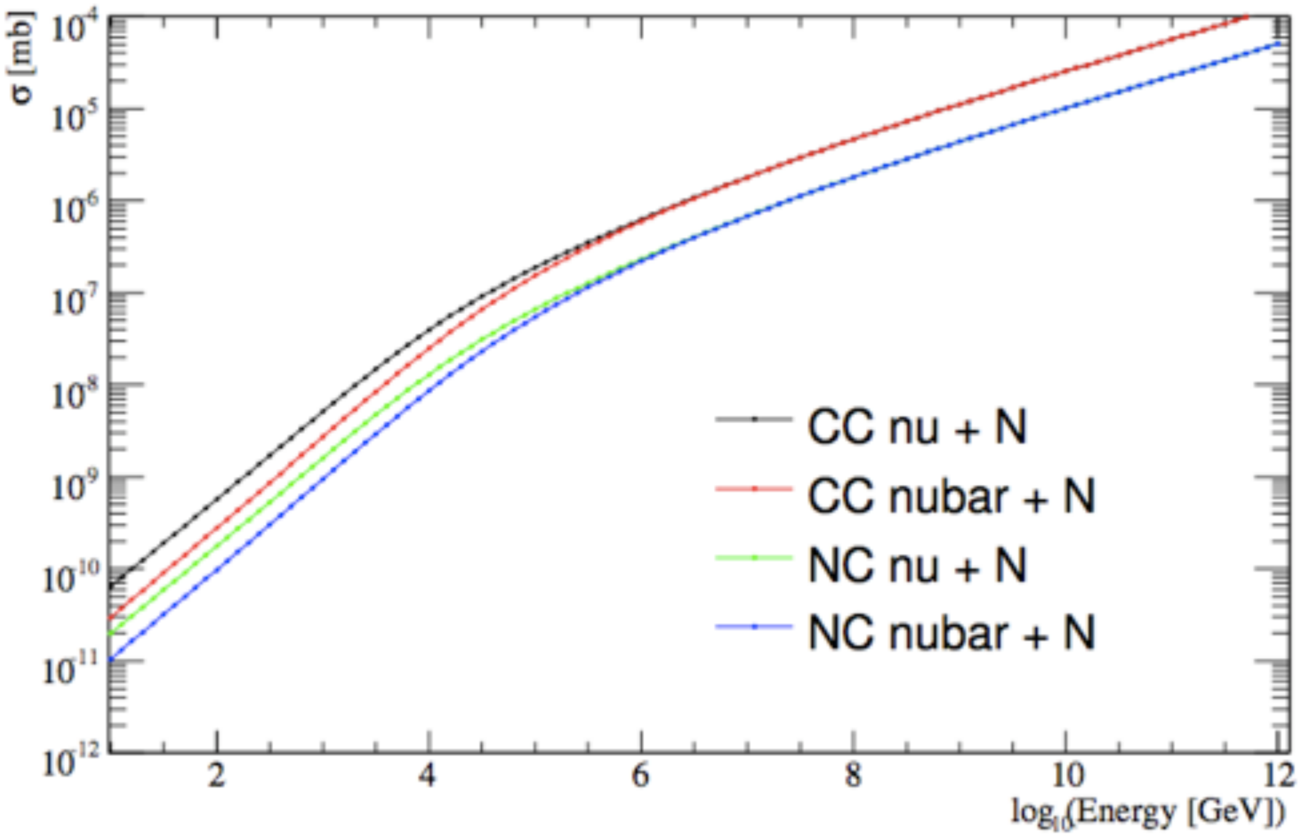
➡ better livetime efficiency @ 10 TeV but poor efficiency @ TeV

➡ energy-targeted generation of (H,He,CNO,Mg,Fe) with $E^{-1(2)}$

generators : neutrino-generator

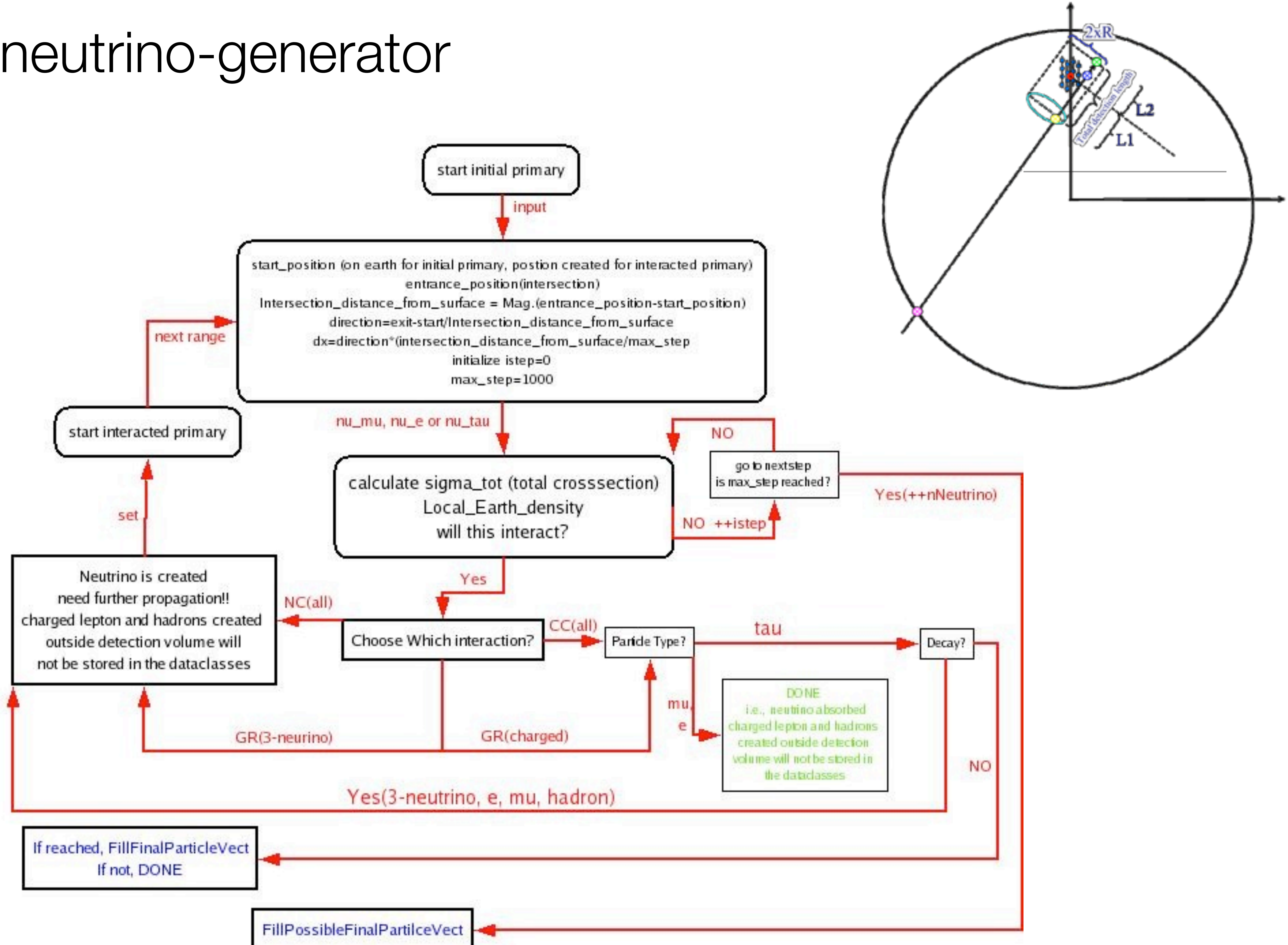
- produce a $E^{-\gamma}$ ν_μ, ν_e, ν_τ with
  - PRELIM Earth's density model



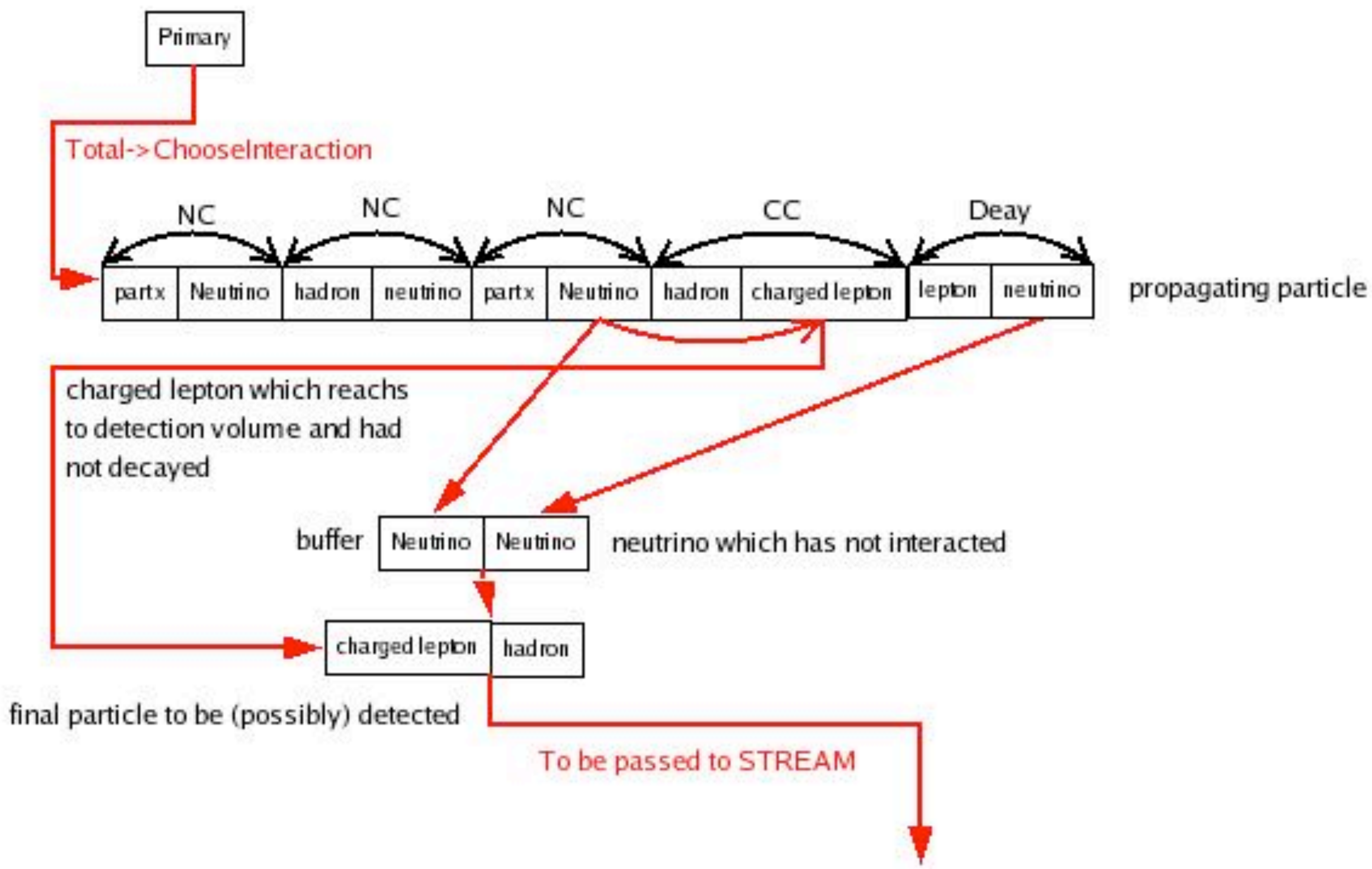


- CTEQ5 parton distribution functions
- prop & interaction of neutrinos into a weight : flexible spectral weight

neutrino-generator



neutrino-generator

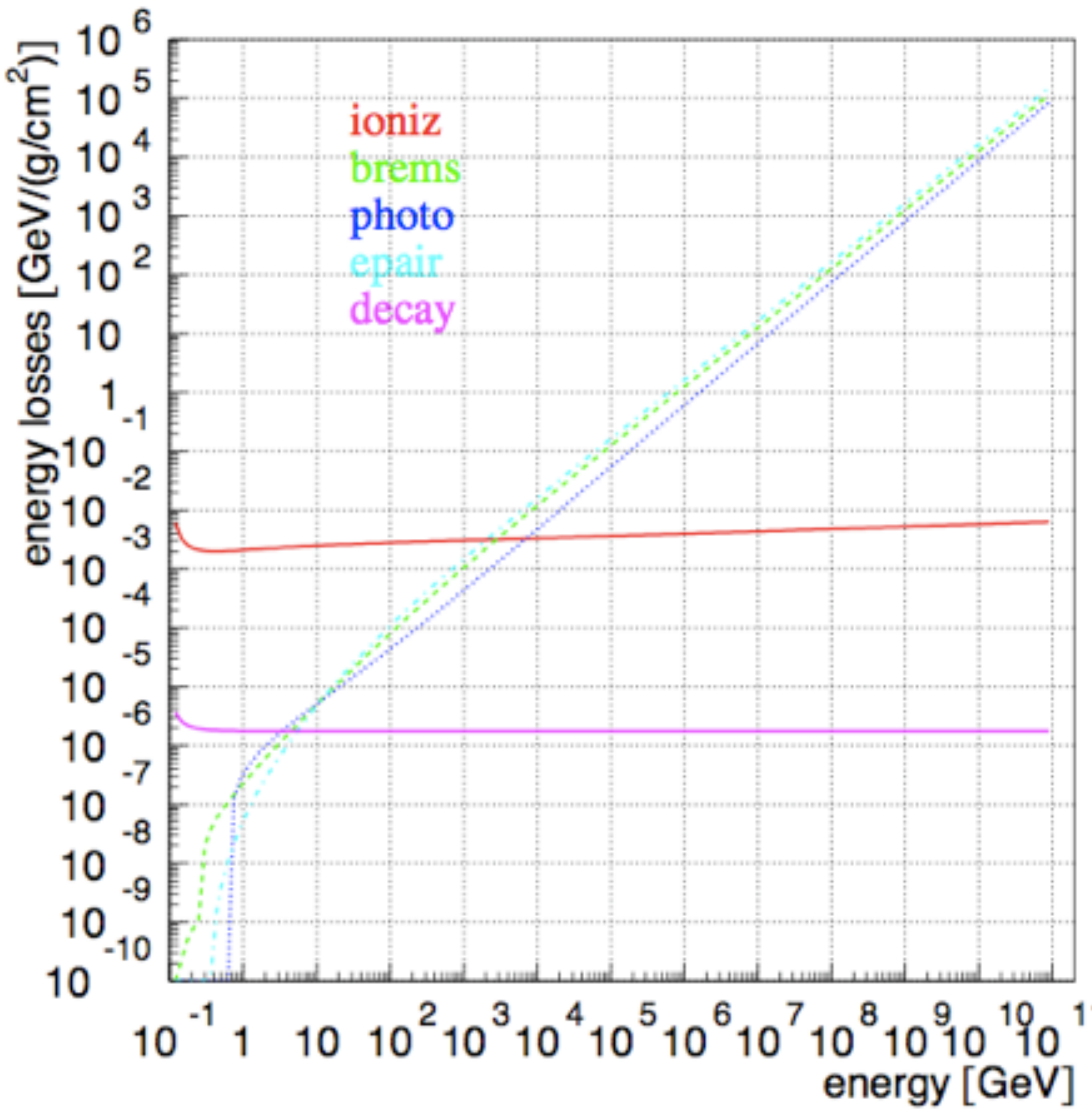


propagator : MMC, Juliet, PROPOSAL

propagates μ, e, τ & monopoles

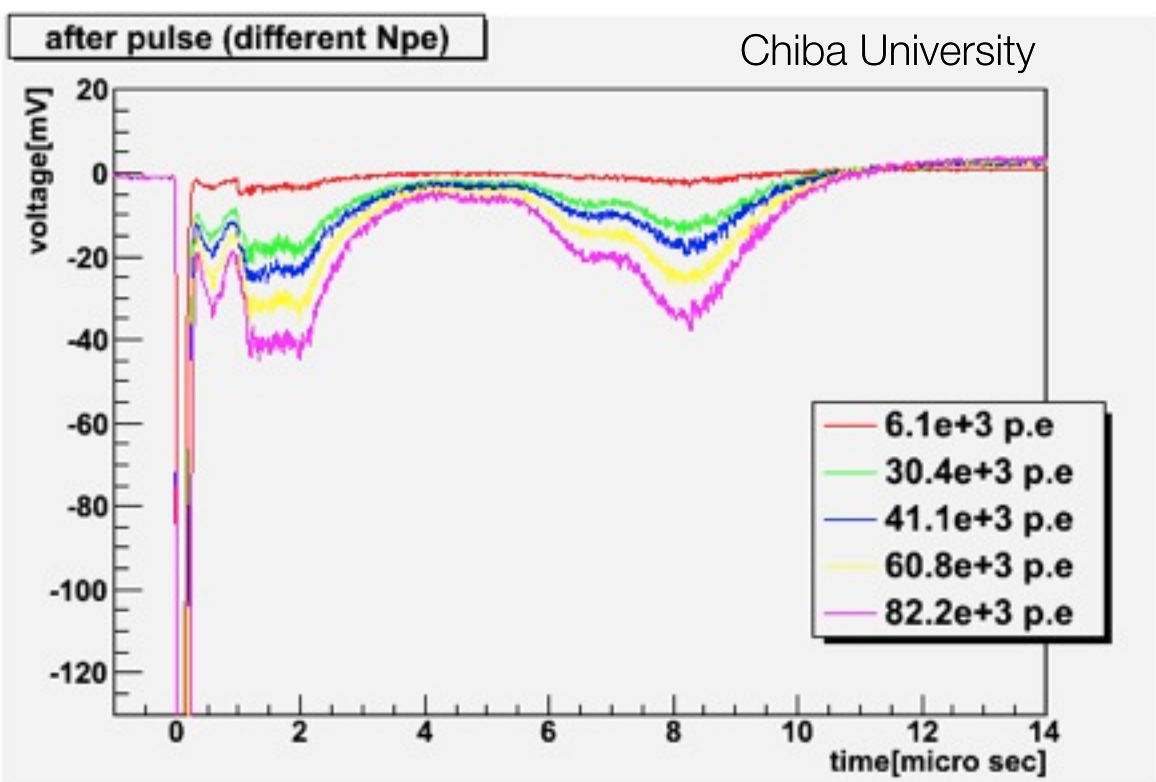
calculates continuous and stochastic energy losses

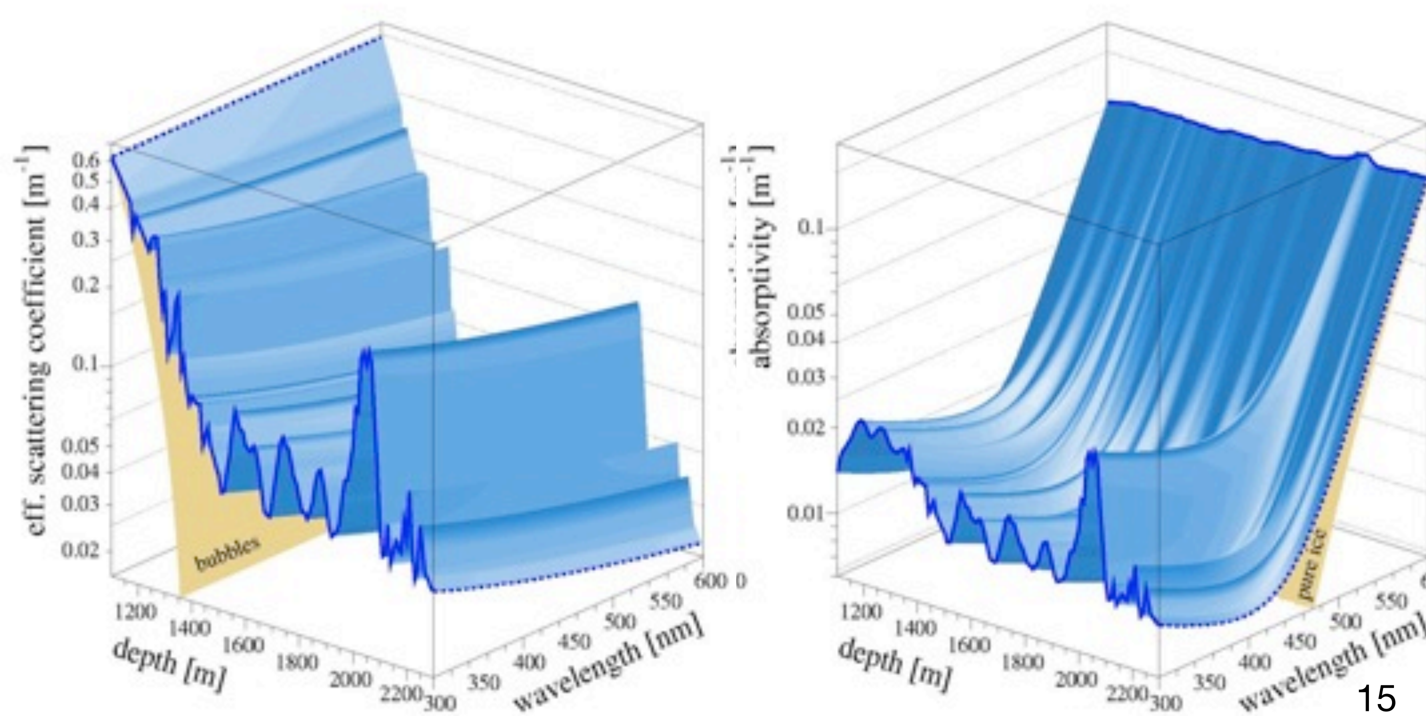
$$\frac{dE}{dx} = a(E) + b(E) \cdot E$$



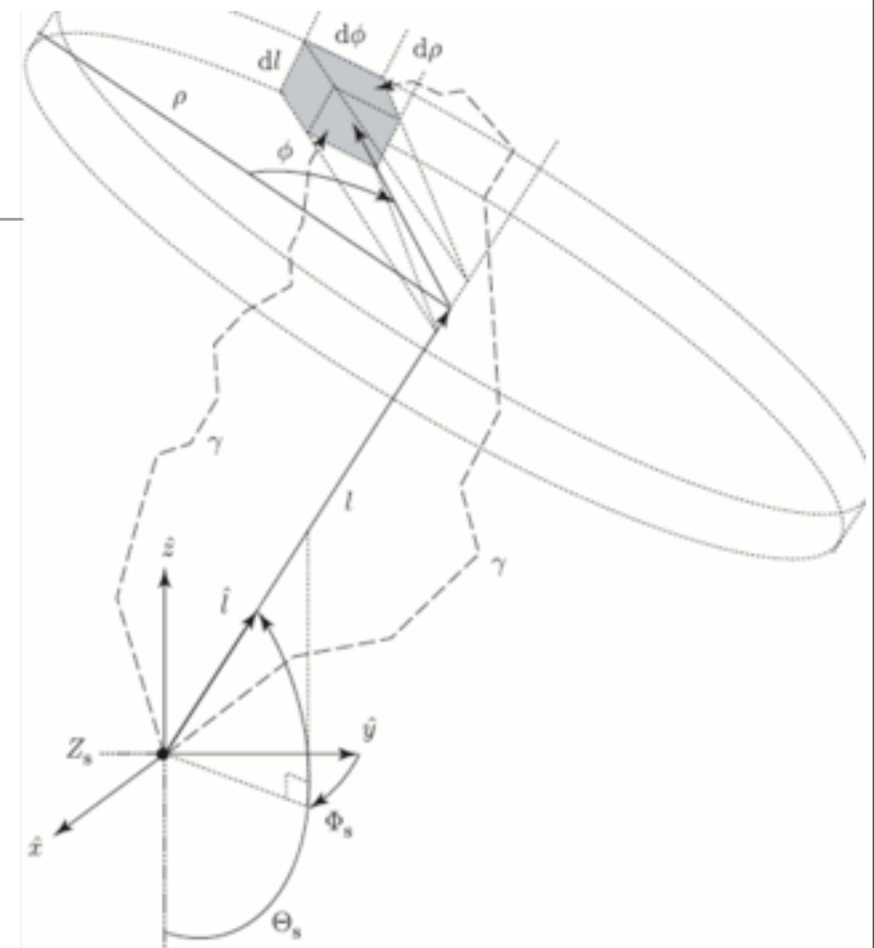
HitMaker

- μ energy lost + cascades ➙ photons ➙ p.e.
  - photon propagation : ice properties + PMT response + DOM glass/gel
  - pre-generated lookup table : amplitude ad time distribution





Photon tracking with tables

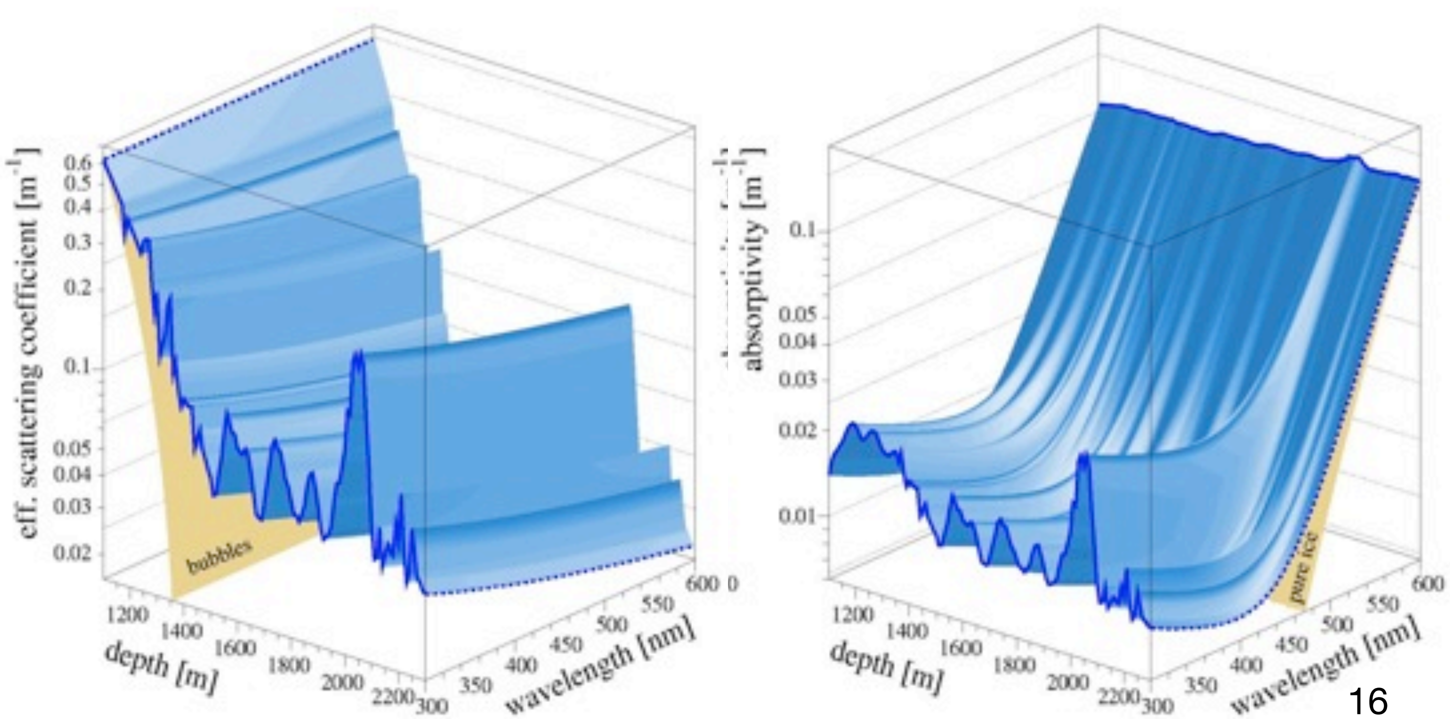


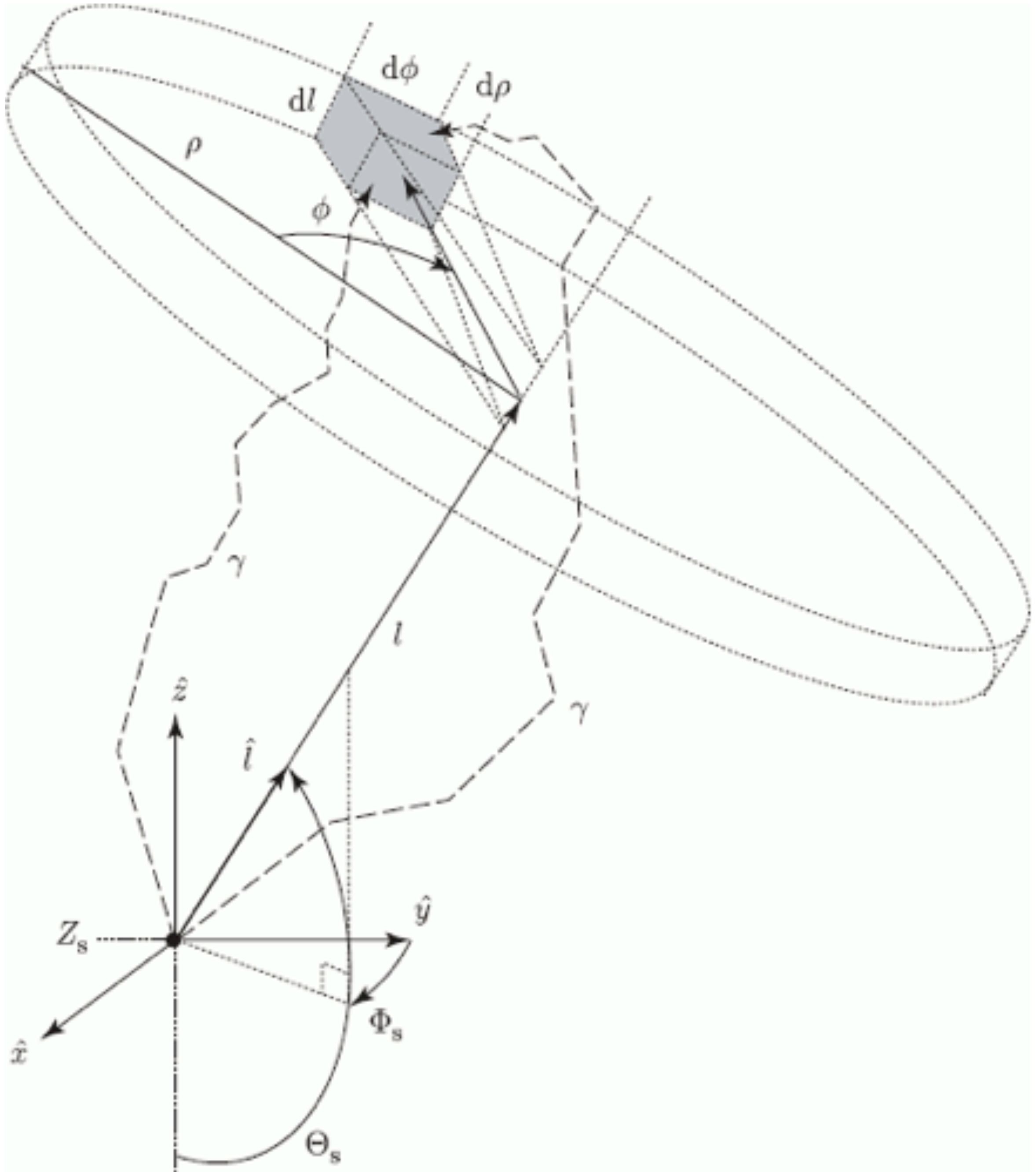
First, run photonics to fill space with photons, tabulate the result

Create such tables for nominal light sources: cascade and uniform half-muon

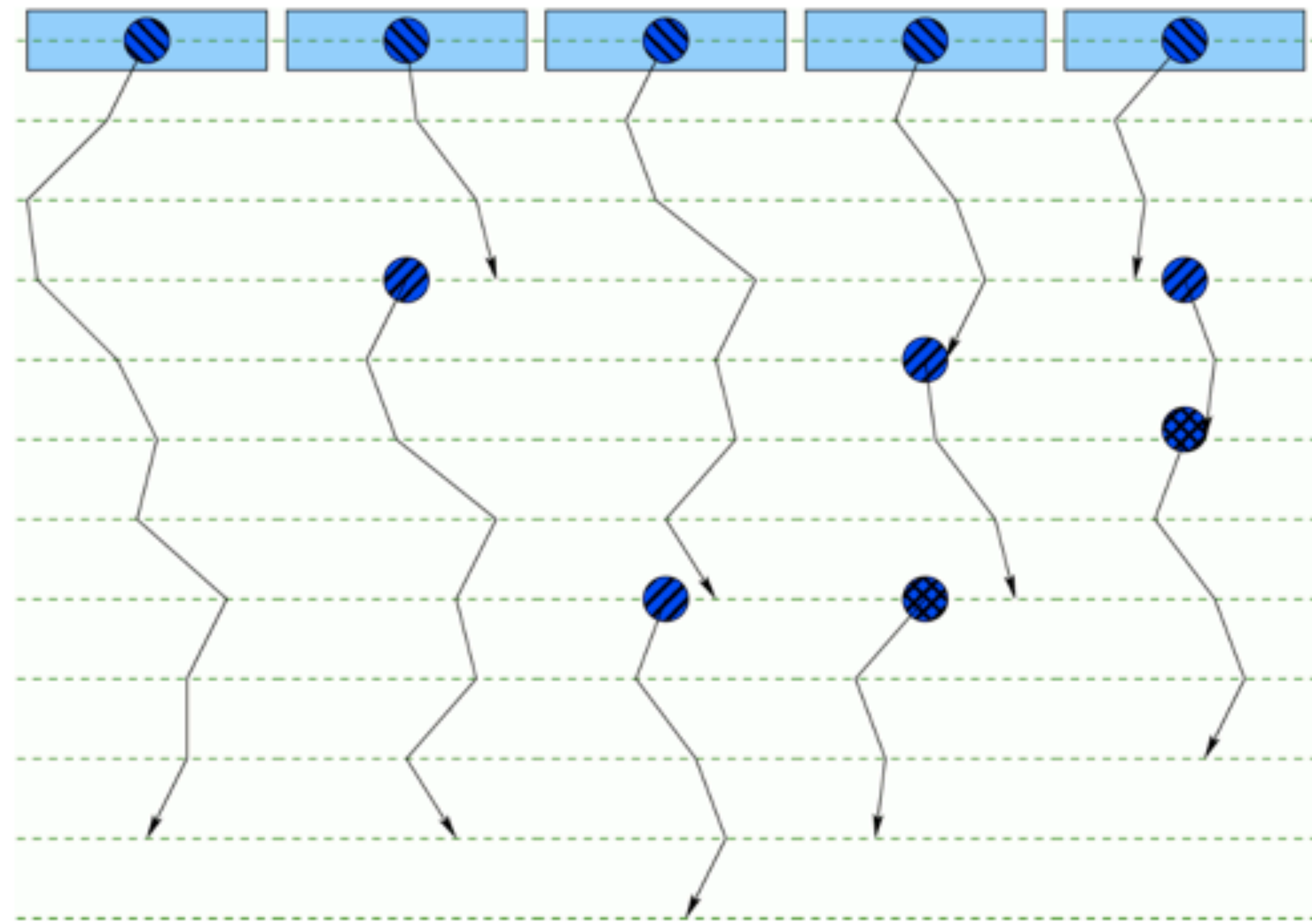
Simulate photon propagation by looking up photon density in tabulated distributions

Table generation is slow
Simulation suffers from a wide range of binning artifacts
Simulation is also slow! (most time is spent loading the tables)
Tables are large ~7GB





PPC simulation on GPU



Direct photon tracking with PPC

diplopia

(from gr. διπλόος, "double", and ὄψ, ὀπός, "vision")

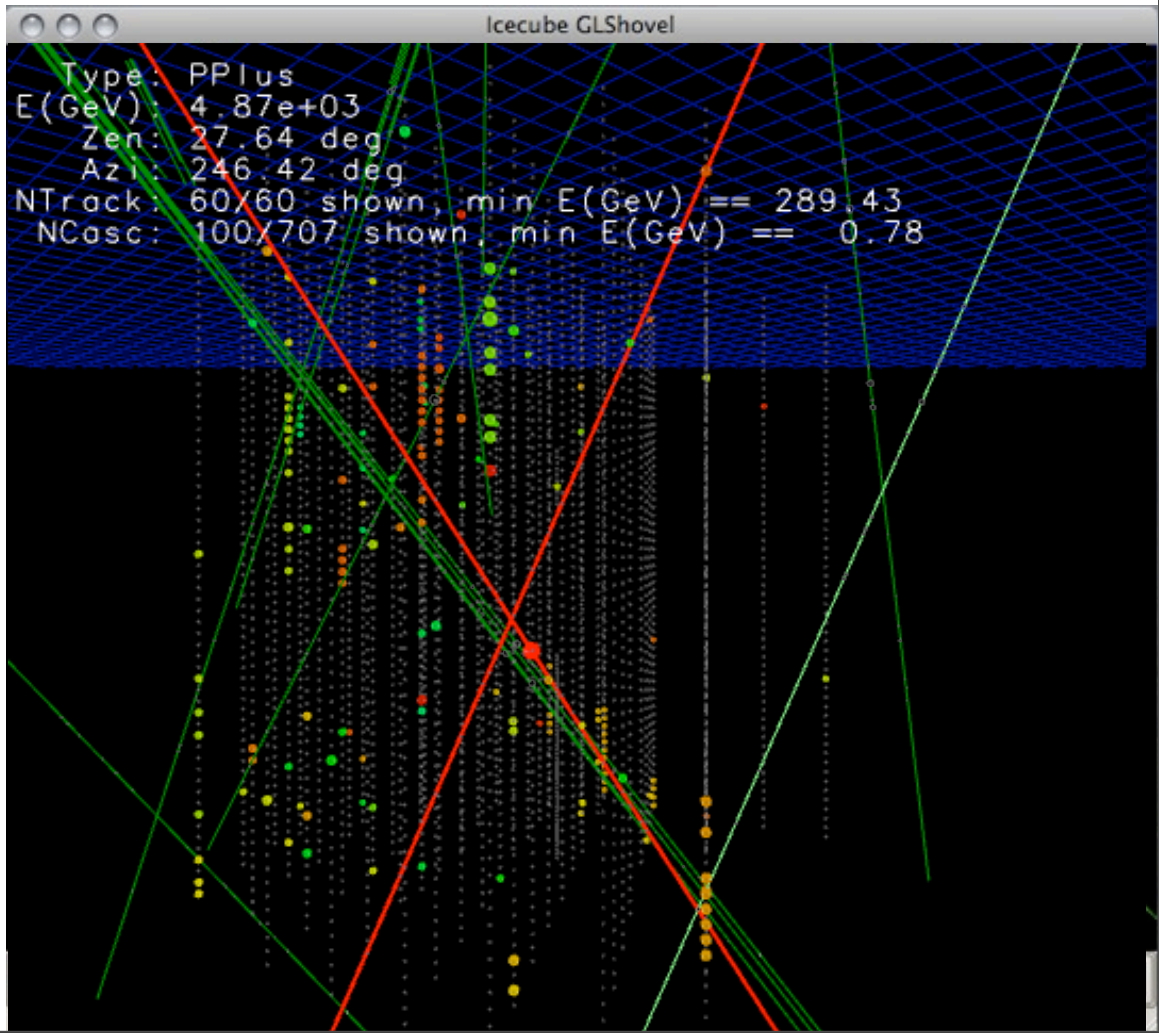
coincident atmospheric shower events in IceCube

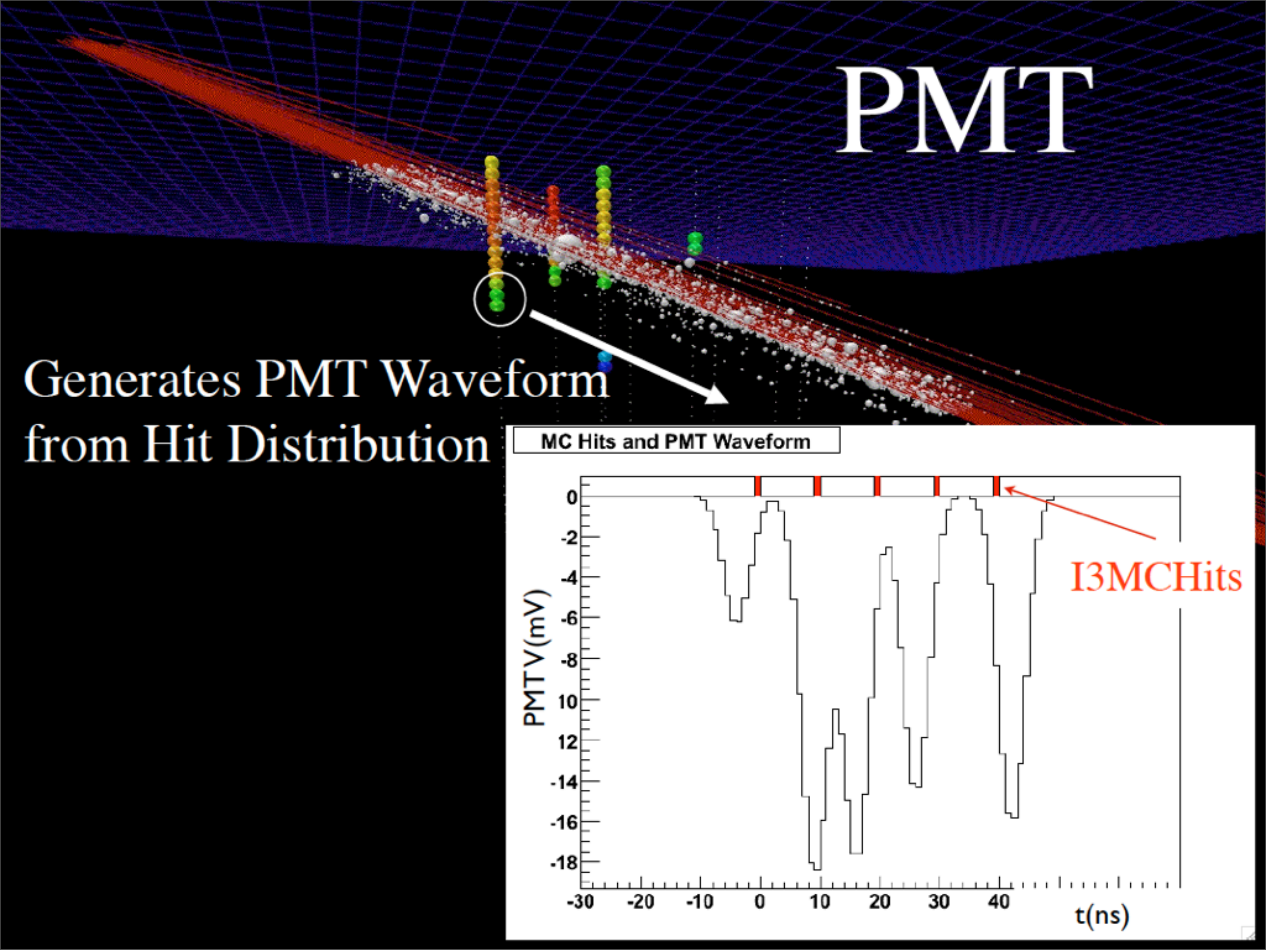
- I3Polyplopia:
  - Forces *n* coincidences in a time window *\Delta t*
  - calculates weight based on Poisson probability of events within time interval.

$$f(k;\lambda t) = \frac{e^{-\lambda t}(\lambda t)^k}{k!}$$

- I3PolyplopiaExp:
  - merges events with different multiplicities based on exp. distribution of time between events

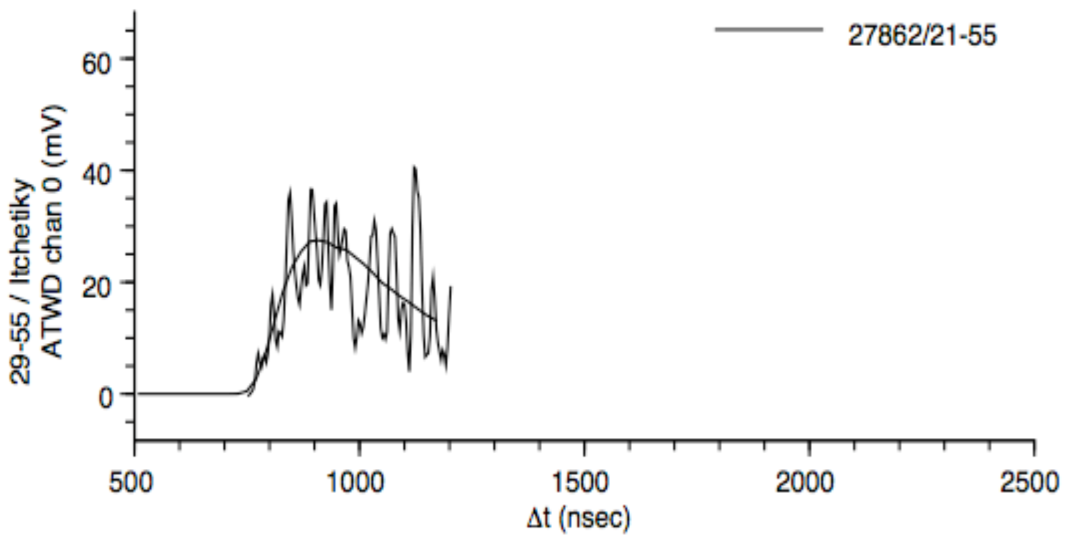
$$p(t) = \frac{1}{\tau}\exp(-t/\tau), \tau = \Delta t R$$

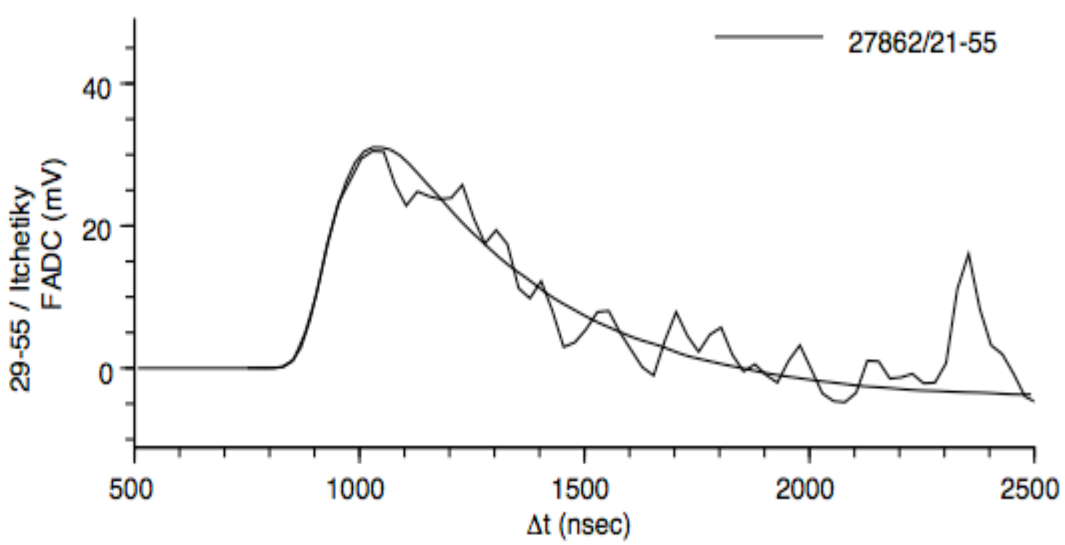


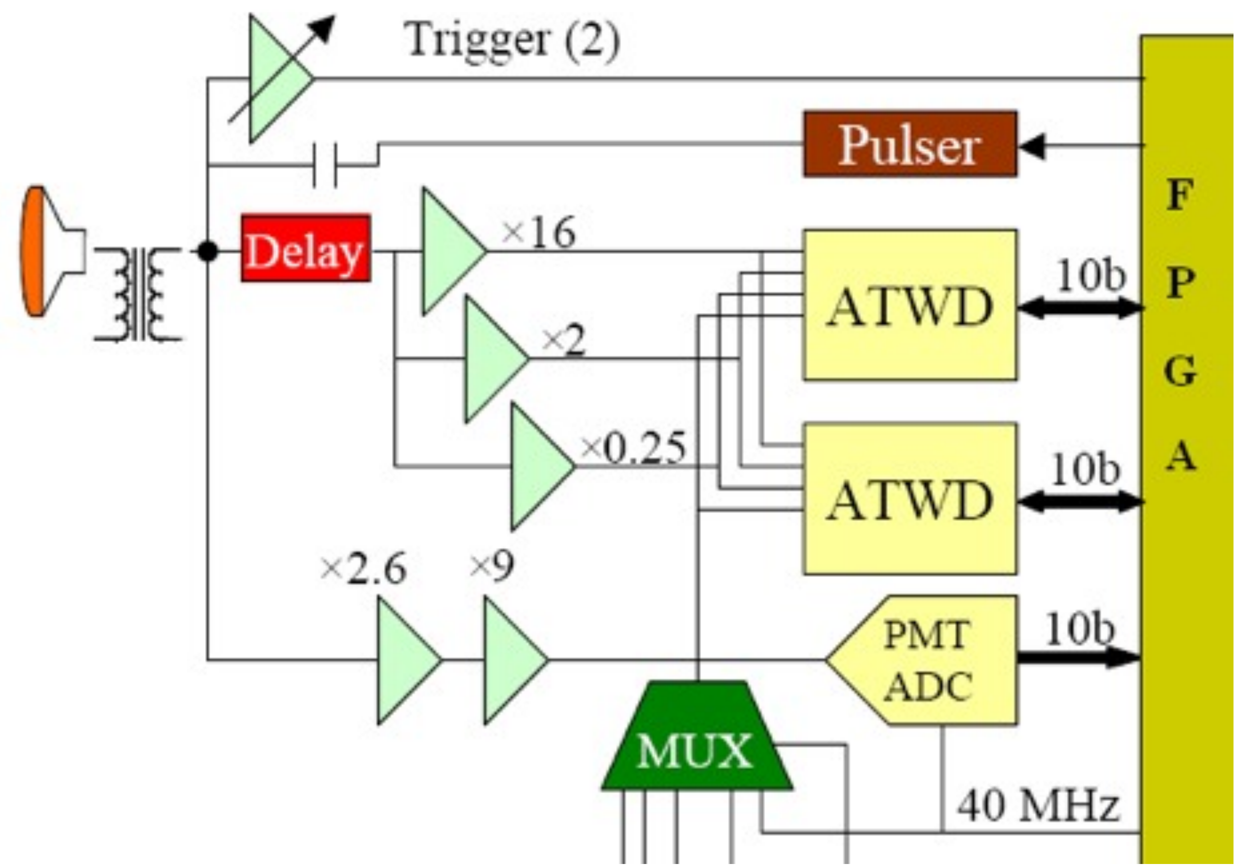

PMT
Generates PMT Waveform from Hit Distribution
MC Hits and PMT Waveform
PMT V(mV)
t(ns)
I3MCHits
0
-2
-4
-6
-8
10
12
-14
-16
-18
-30
-20
-10
0
10
20
30
40

DOM mother board simulation

- core of detector simulation
    - digitization & timing @ MB

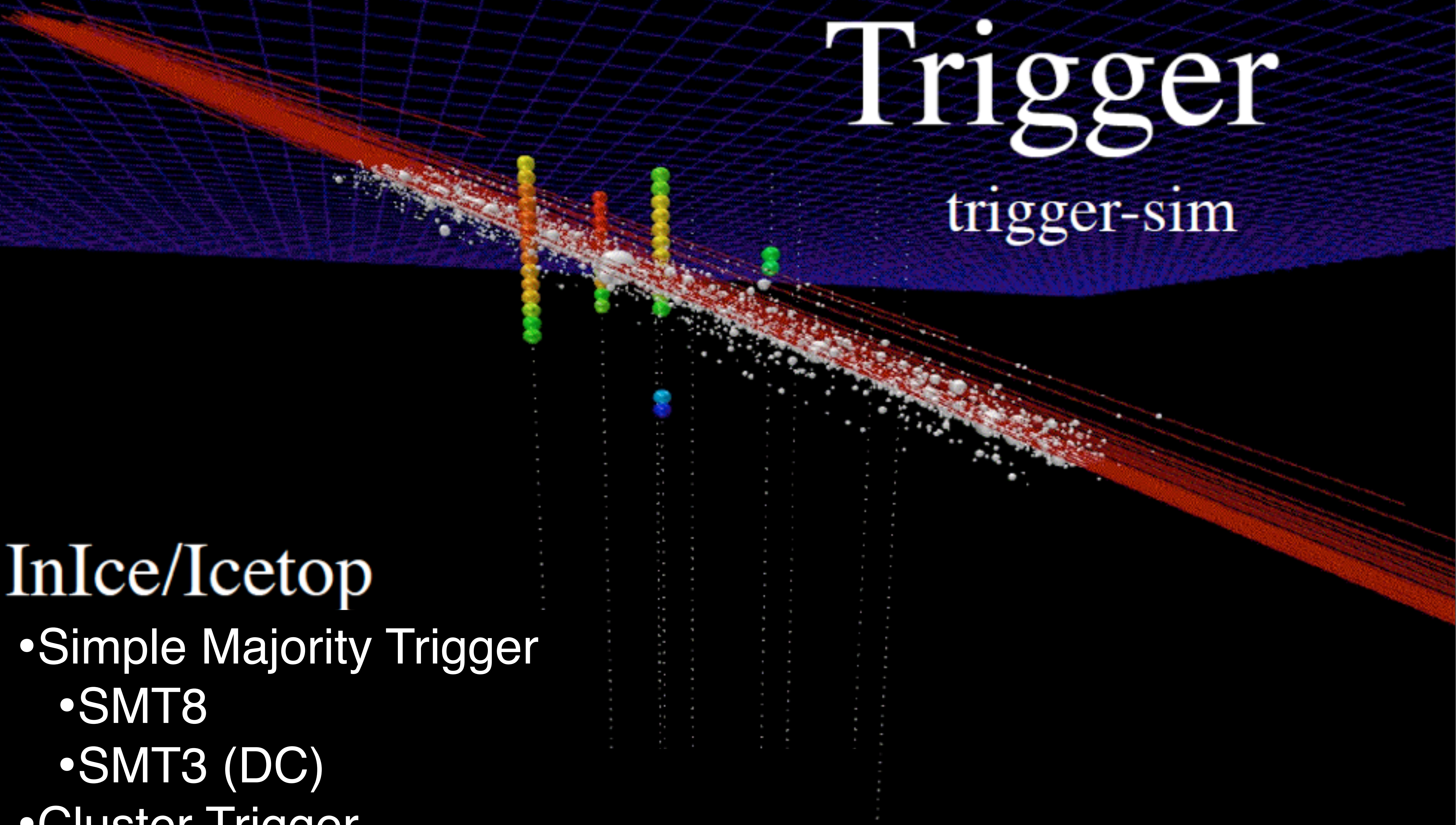






Trigger

trigger-sim



InIce/Icetop

- Simple Majority Trigger
  - SMT8
  - SMT3 (DC)
- Cluster Trigger
- Volume Trigger
- SlowMP

Global Trigger
I3TimeShifter

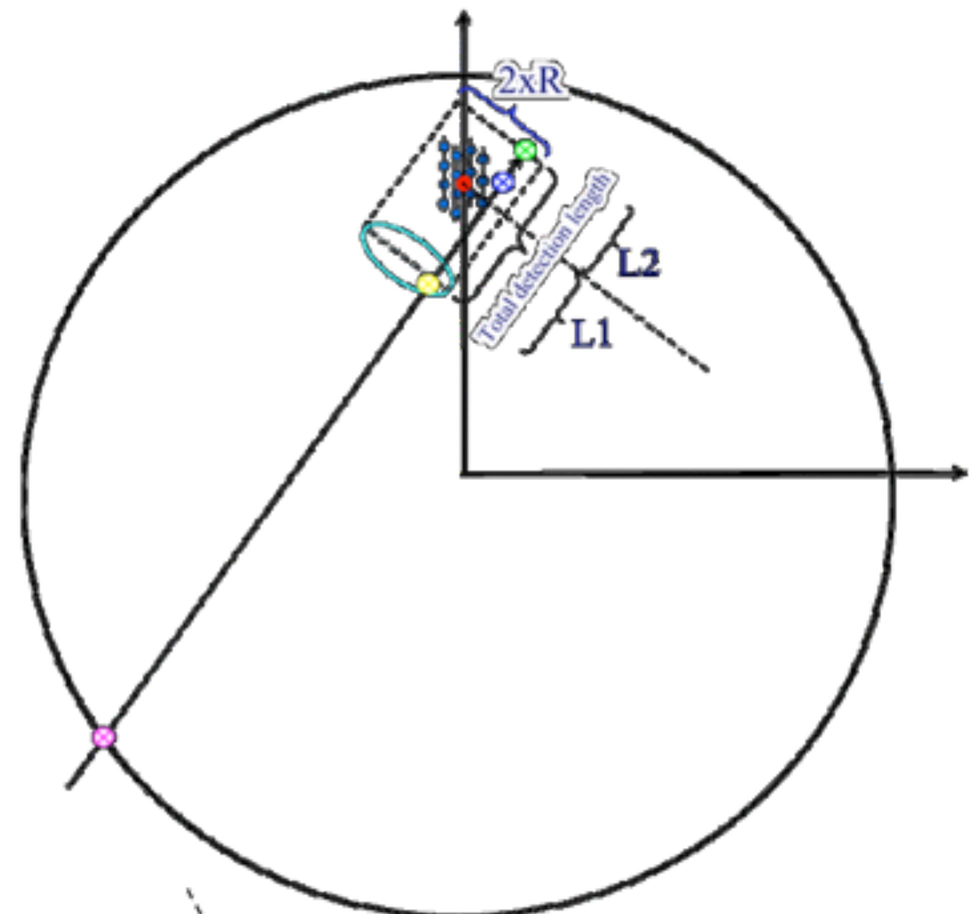
http://wiki.icecube.wisc.edu/index.php/Simulation_Time_Issues

Weights

neutrino-generator

- Calculates the propagation probability (i.e. that the neutrino will reach the detector)

- It forces an interaction within a volume around the dector and computes the probability of this interaction



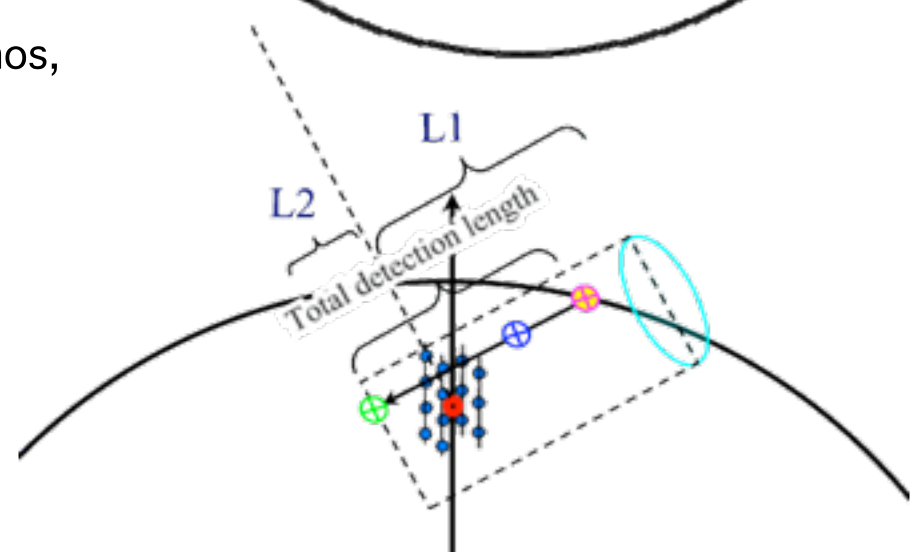
$$OneWeight = \left(\frac{P_{int}}{E^{-\gamma}}\right) \cdot \int_{E_{min}}^{E_{max}} E^{-\gamma} dE \cdot Area \cdot \Omega \cdot T [GeV \cdot cm^2 \cdot sec \cdot sr]$$

where

- P_{int} = TotalInteractionProbabilityWeight,
- $E^{-\gamma}$ is the neutrino generation energy spectrum shape,
- E_{min} and E_{max} are the minimum and maximum generation energy of neutrinos,
- Area is the generation surface,
- Ω the generation solid angle
- and T = 1sec is the timescale.

- The weight corresponding to a given theoretically motivated neutrino flux is

$$\boxed{w_i = \frac{OneWeight_i}{NEvents} \times \frac{d\Phi_\nu(E_\nu)}{dE_\nu}}$$



https://docushare.icecube.wisc.edu/dsweb/Get/Document-44937/OneWeight.pdf

CORSIKA weights

- In addition, we calculate a weight "DiplopiaWeight" which accounts for the probability of a given event to fall within the trigger window of other events.

- When combining events from different files *j* then the weight is

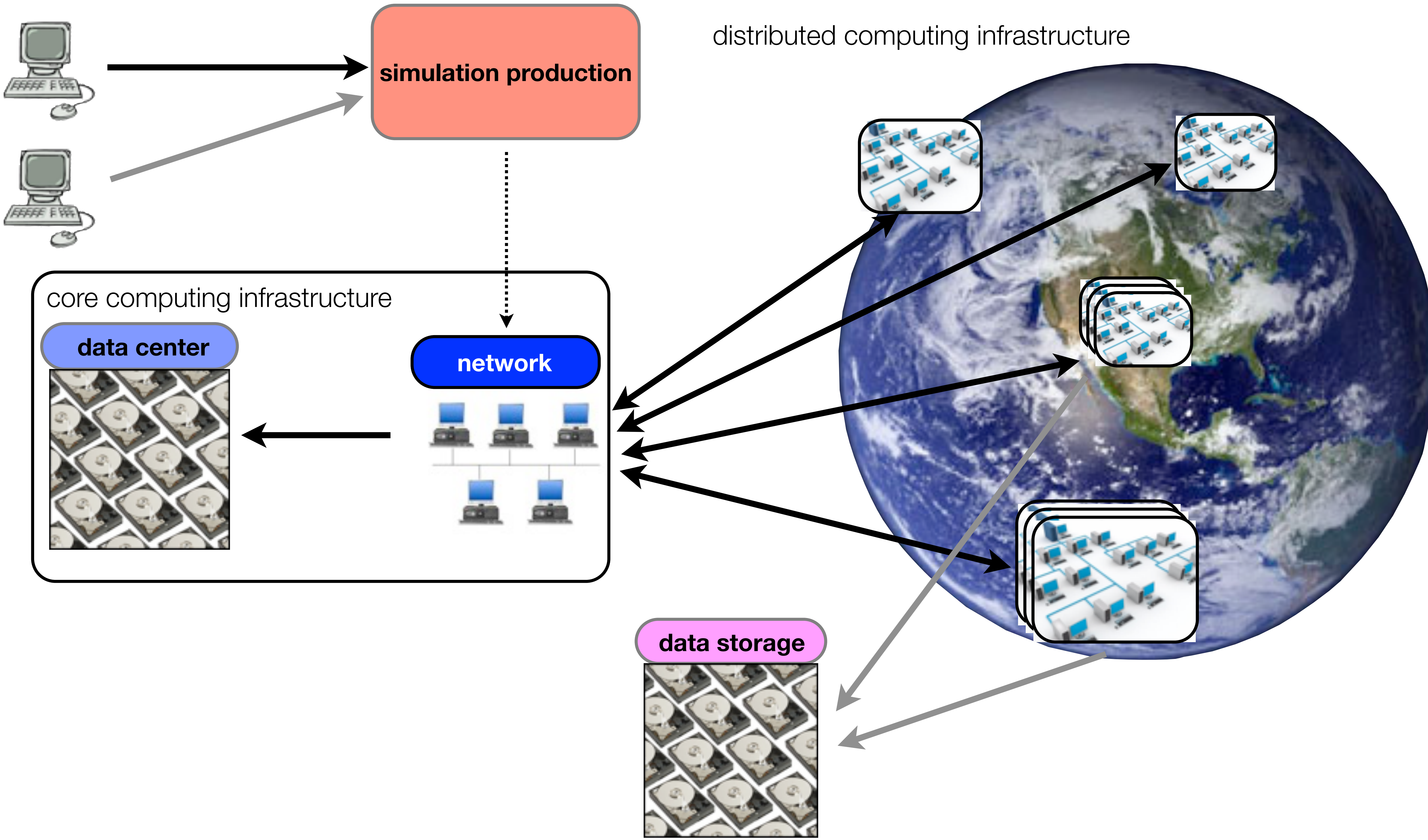
$$\boxed{w_i = \frac{W_i}{\sum_j TimeScale_j} = \frac{Weight_i \cdot DiplopiaWeight_i}{\sum_j TimeScale_j}}$$

- in general *TimeScale* might be different for different files. When using this weight we get, for each bin, the number of events per second.

CORSIKA weights Re-weighing 5-component CORSIKA

- http://wiki.icecube.wisc.edu/index.php/Weights_in_CORSIKA_Simulation_Data#5-component_CORSIKA

Simulation Production



More on simulation

1. http://wiki.icecube.wisc.edu/index.php/Simulation_Documentation_Wiki
2. http://wiki.icecube.wisc.edu/index.php/IceCuber's_Guide_to_IceSim_2.6
3. http://wiki.icecube.wisc.edu/index.php/Simulation_Production
4. http://internal.icecube.wisc.edu/simulation/datasets/dataset_category/PHYSICS
5. IRC @ irc.efnet.net: /join #icesim

On cobalt01 or cobalt02

Get IceSim:

cp /data/sim/sim-new/simulation.candidates.V03-02-04-RC.Linux-x86_64.gcc-4.4.5.tar.gz .

tar xzf simulation.candidates.V03-02-04-RC.Linux-x86_64.gcc-4.4.5.tar.gz

Replace env-she.sh:

cp /net/user/juancarlos/bootcamp/env-shell.sh simulation.candidates.V03-02-04-RC.Linux-x86_64.gcc-4.4.5/

Get scripts:

cp /net/user/juancarlos/bootcamp/*.py .