A Geant4 framework for generic simulations of atmospheric muon detection experiments

Nicola Mori

University of Florence and INFN Florence

MUOGRAPHERS – Tokyo (JP), 12 Nov 2014

Atmospheric muon simulations

- Relevant for a large number of applications:
  - Geology/geophysics
  - Archaelogy
  - Nuclear waste monitoring
- Many possible simulation scenarios
  - Different detectors * different targets
- Some common features/needings:
  - Generation of atmospheric muon flux
  - Normalization to acquisition time
  - Overall detector layout (e.g. tracking layers)

Atmospheric muon simulations

- Goal: a fast and reliable deployment of Monte Carlo simulation code for different scenarios, allowing for as much code-reusage as possible
- Step 1: develop an experiment-independent simulation framework, tackling the common MC issues (geometry definition, simulation setup, output etc.)
  - Must be easily extensible in order to allow for experiment-specific bits
- Step 2: implement muon-specific, scenario-independent features
  - Muon generator, detector, normalization tools etc.
- Step 3: implement scenario-specific bits
  - Volcano, storage silo etc.

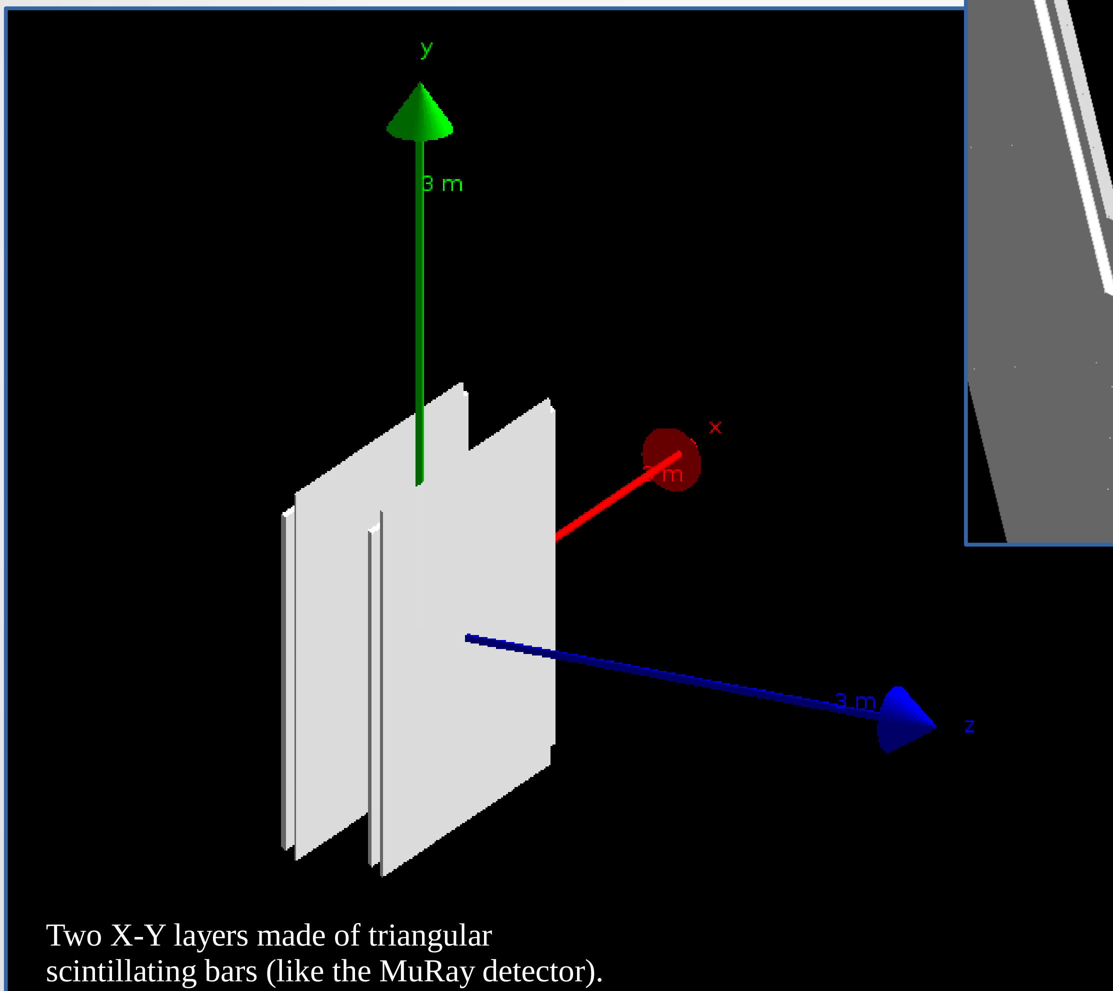
Step 1: the generic MC framework

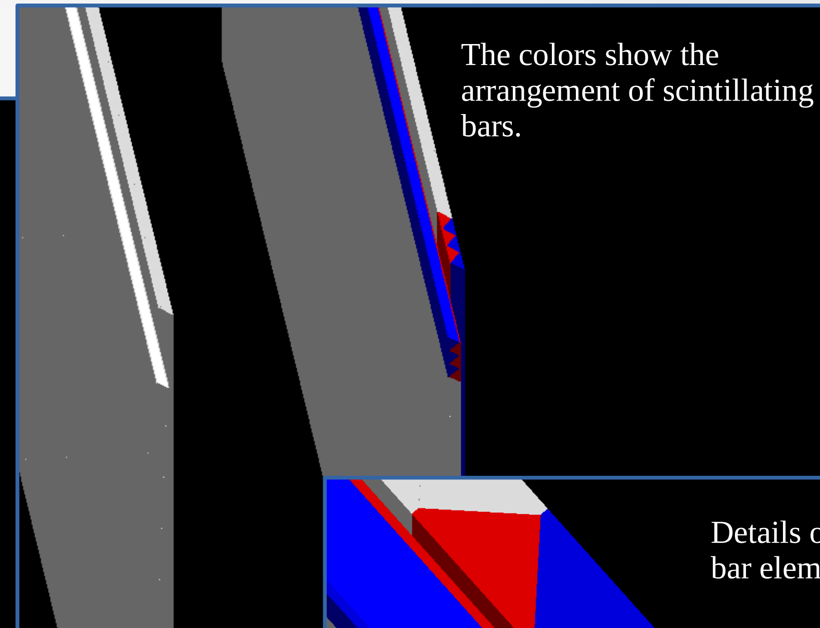
- GGS: Generic Geant4 Simulation
  - Developed at INFN Florence to serve as common MC base for many experiments (e.g. Calet, Gamma400, MuRay/MURAVES, …)
- Based on Geant4 (http://geant4.cern.ch/)
- Generic: works with any geometry
  - Generic implementations of all the Geant4 mandatory user classes
    - Except for experiment geometry, of course...
  - Generic energy-deposit and MC truth data objects
  - Adaptive output based on Root (http://root.cern.ch)
  - Readout and offline data analysis objects
- Extensible via runtime plugin libraries
  - Geometry, hits, particle generators, user-defined hooks

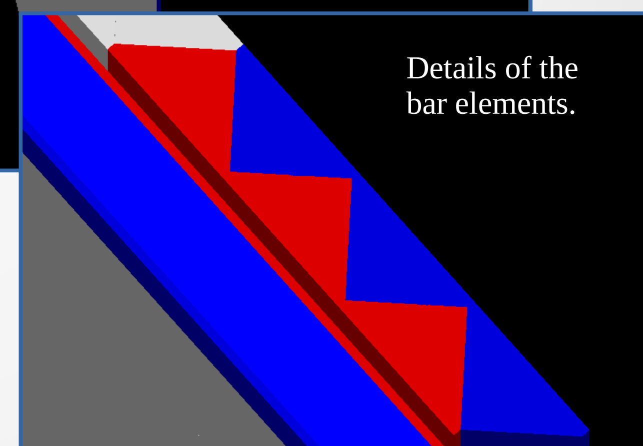
Step 2: the common muon stuff

- Detector geometry: a C++ Geant 4 geometry class implementing N pairs of X-Y detector layers made of PVT
  - Fully configurable: size, number of planes, shape of detector elements etc.
  - Runtime configuration via options file
    - no need to recompile the plugin for each combination of parametrs → less room for errors
  - Quick and dirty solution for rapid deployment
    - A more accurate, specific detector geometry can be defined, if needed

Step 2: the common muon stuff

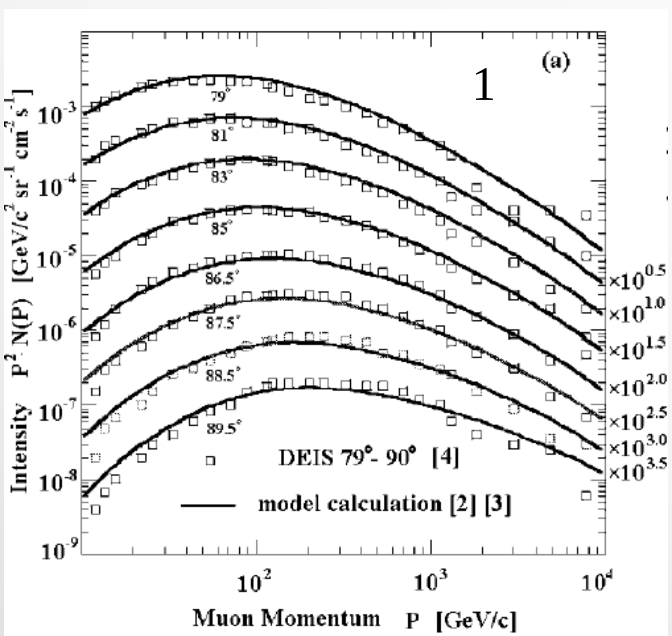


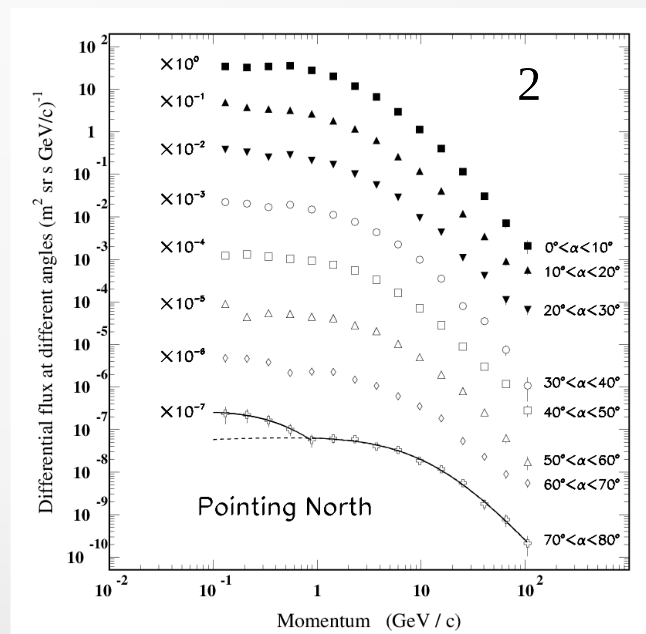




Step 2: the common muon stuff

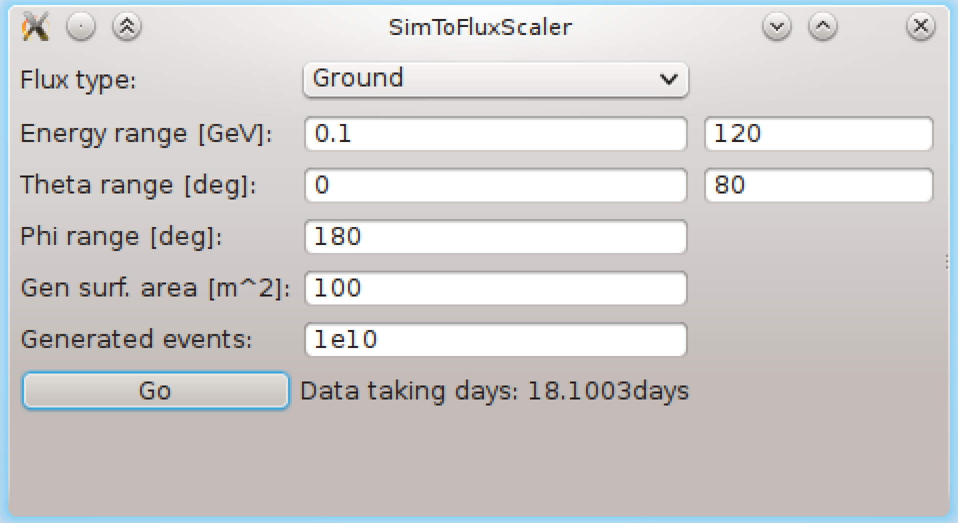
- Muon flux generator: two implementations (GGS plugin)
  - Implementation 1: based on a theoretical model (H. Tanaka *et al.*, Hyperfine Interactions 138 (2001))
  - Implementation 2: based on ground meas. with a magnetic spectrometer (L. Bonechi *et al.*, Intl. Cosmic Ray Conf. Proc. (2005), 283)





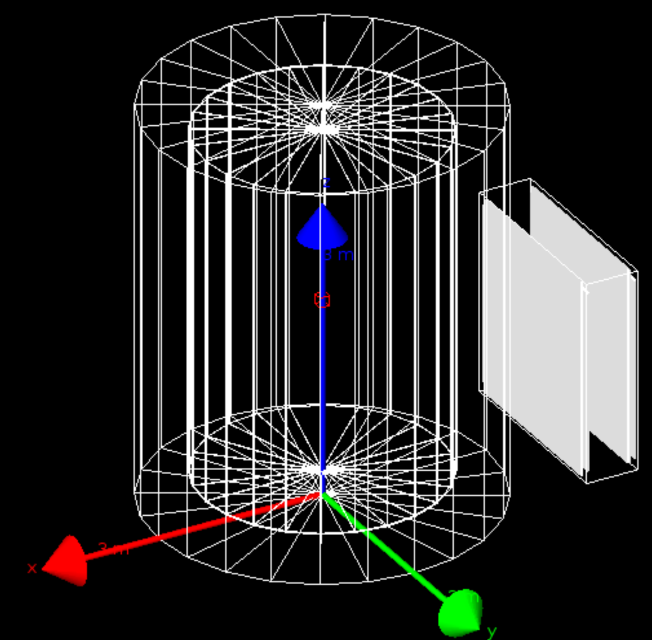
Step 2: the common muon stuff

- Normalization tool: graphical Qt interface
  - # of generated events → generation time
  - Numerical integration of $\Phi(E_k, \theta)$



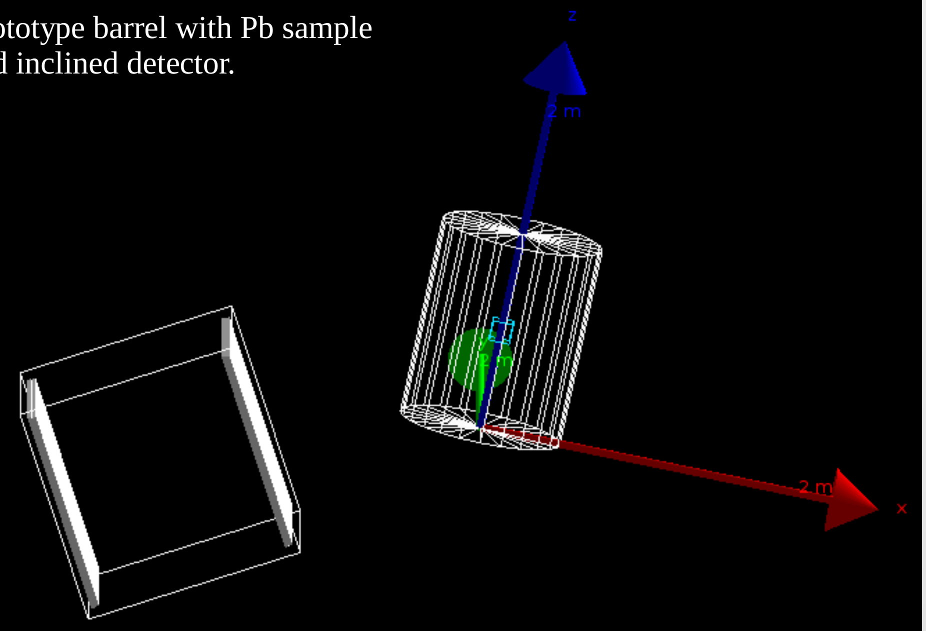
Step 3: the experiment-specific stuff

- Built on top of muon-specific stuff as GGS plugins
- Essentially: a GGS plugin geometry with target (storage silo, volcano, …) and the desired number and placements of detectors

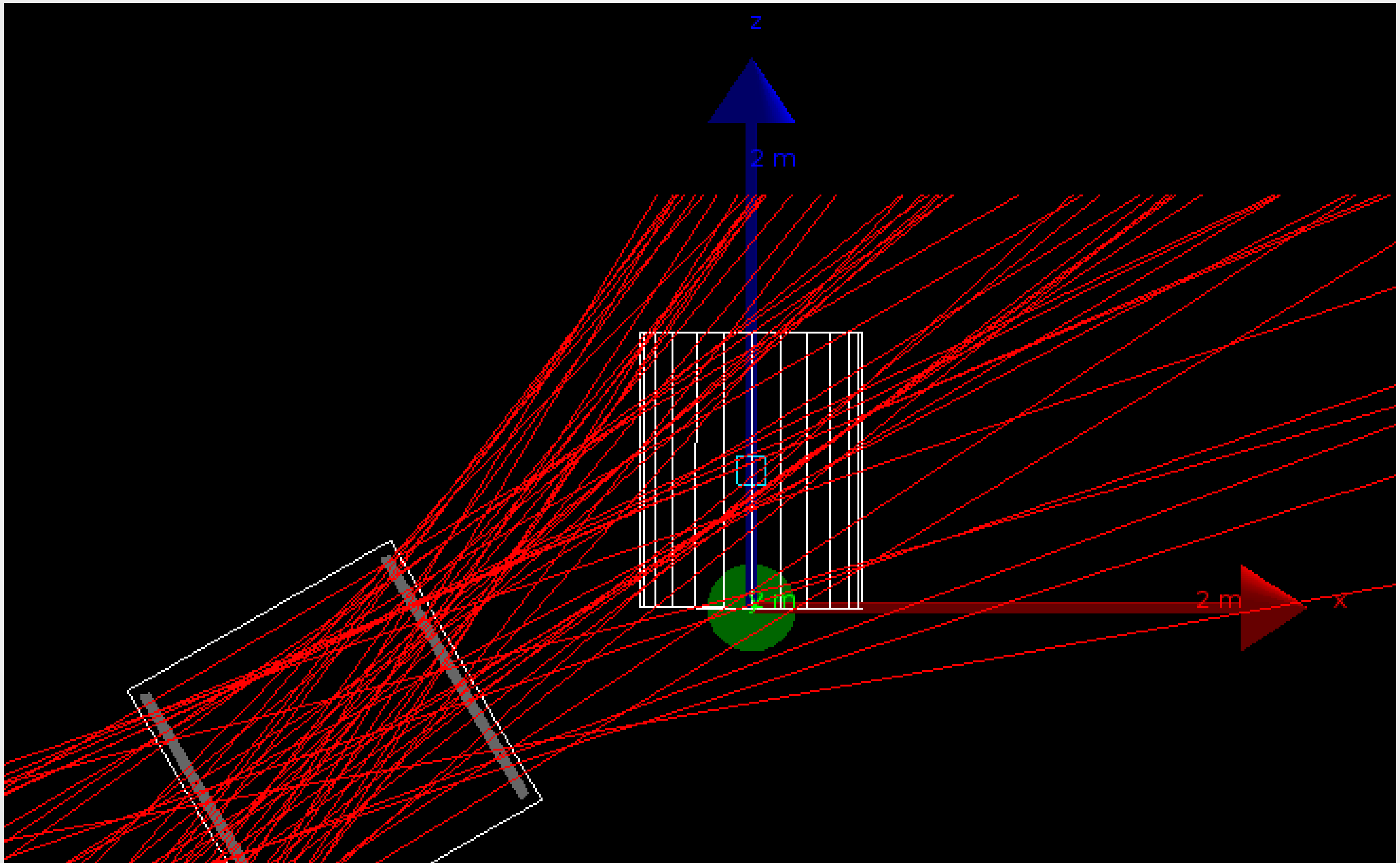


Nuclear waste storage silo with U sample and detector placed along the X axis.

Prototype barrel with Pb sample and inclined detector.

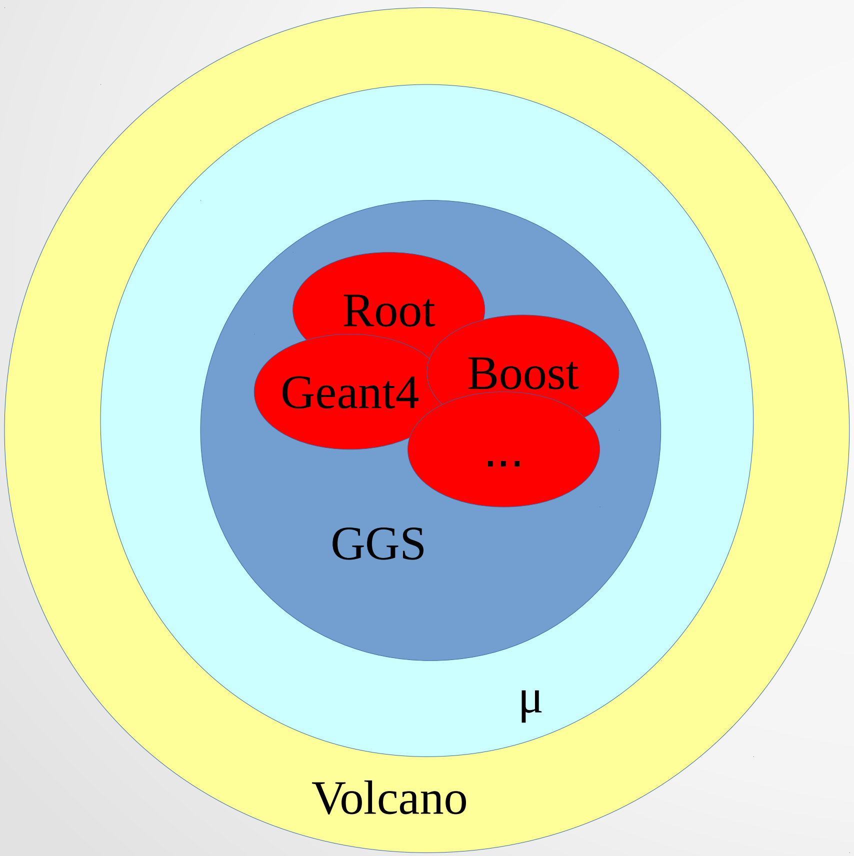


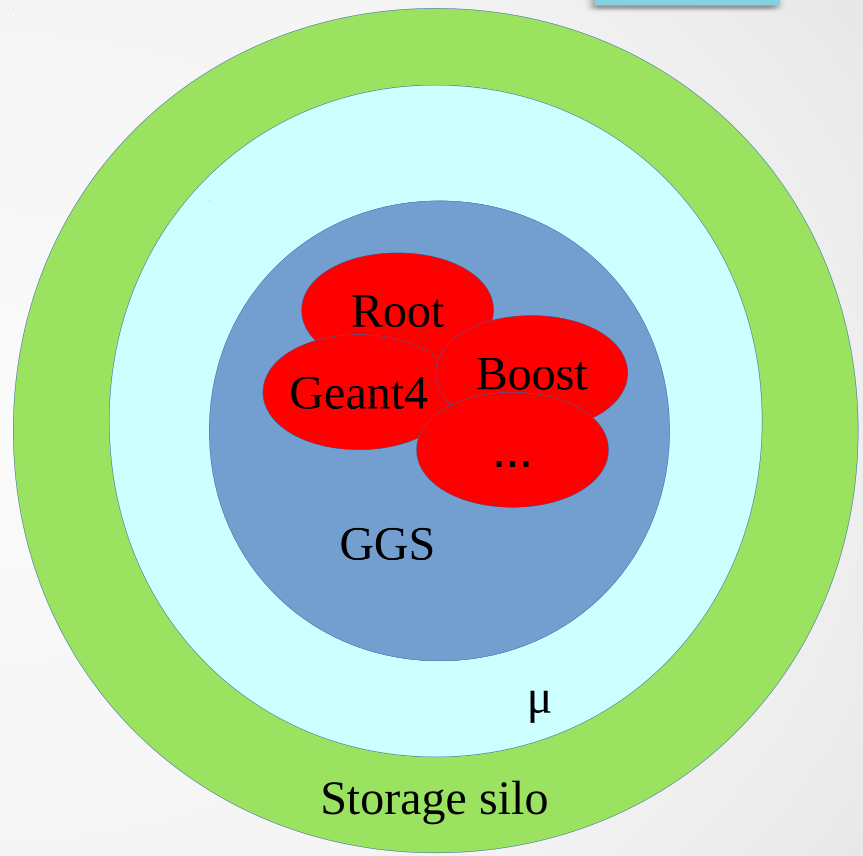
Step 3: the experiment-specific stuff



Simulation of the prototype barrel scenario. Muons are generated from a horizontal surface at Z = 2 m using generator 2.

A pictorial view





Summary

- A full-featured software stack for Monte Carlo simulations of atmospheric muon detection experiments has been designed and developed
- The main design goals (code reusage, fast deployment, customizability) have been achieved
- The software has already been used in production environments
- Scientific papers presenting results obtained with the MC software will appear soon