Introduction to Geanr4 Physics Component

Geant4 Training Event – Calorimetry in HEP
DESY Zeuthen
10-13 May 2011
V. Ivanchenko

Outline

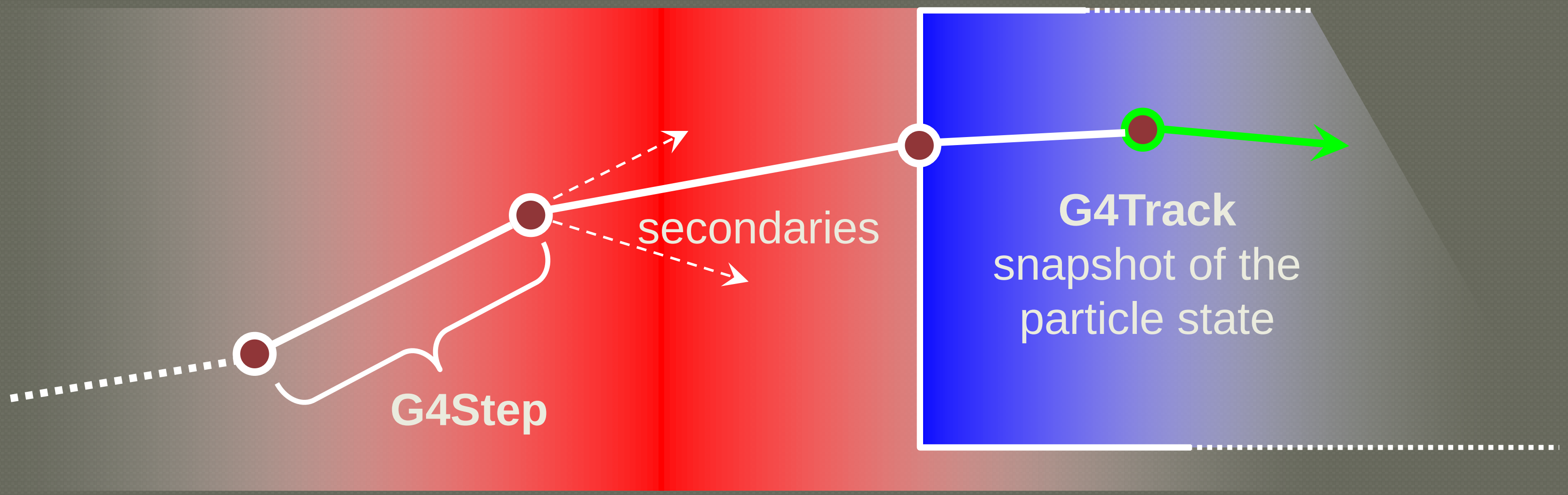
- General interface to Geant4 physics
  - Adaptation of Marc Verderi original lecture
- Geant4 cuts
- Cuts per G4Region

Geant4 interface to physics

- The **`G4ParticleDefinition`** interface
- The **`G4VProcess`** class process interface
- The **`G4ProcessManager`** class

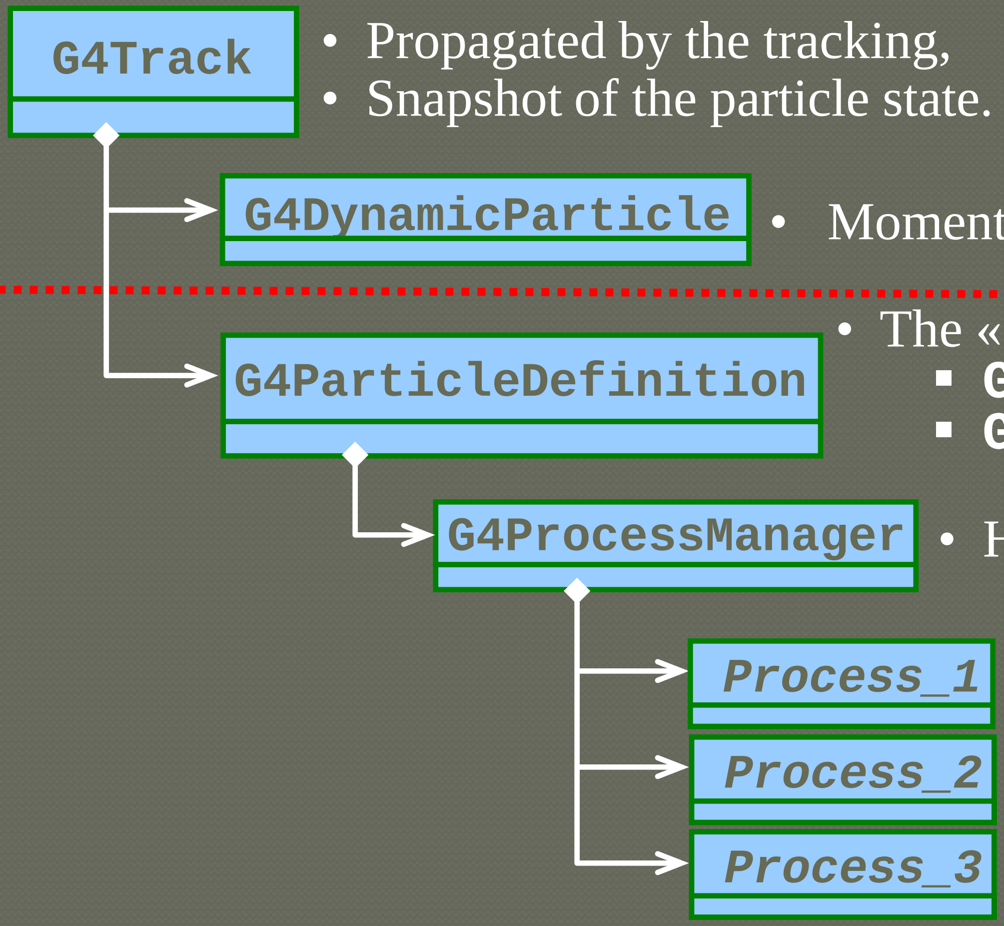
Geant4 tracking

- **`G4Track`** is the object "pushed" step by step by the tracking :



- Moving by one step is the responsibility of the "stepping"
  - Which is the core engine of the "tracking" machinery
- These moves/steps have to be physically meaningful
  - And the stepping invokes physics to realize them
- This physics is attached to the **`G4Track`**, let's see how.

From G4Track to processes



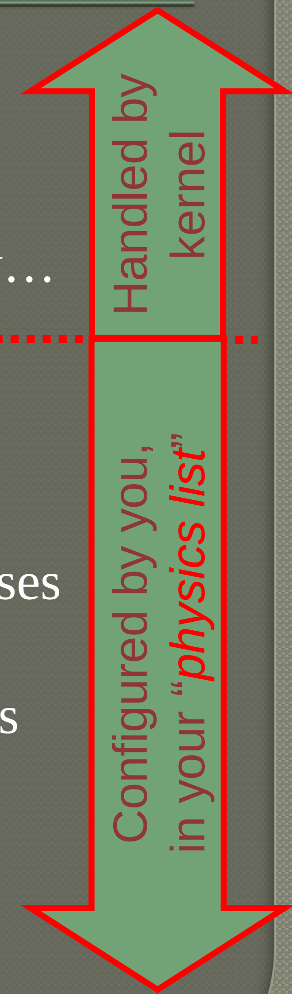
- Propagated by the tracking,
- Snapshot of the particle state.

- Momentum, pre-assigned decay…

- The « particle type »:
  - **G4Electron**,
  - **G4PionPlus**…

- Holds pointers to processes

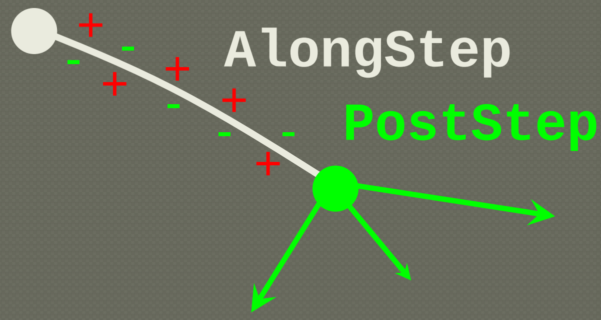
- Implementing physics processes



G4VProcess: 3 kind of actions

- Abstract class defining the common interface of all processes in Geant4:
  - Used by all processes
    - including transportation, etc…
  - Defined in `source/processes/management`
- Three kinds of actions:
  - AtRest actions:

    - Decay, e^+ annihilation …
  - AlongStep actions:
    - To describe continuous (inter)actions, occurring along the path of the particle, like ionisation;



  - PostStep actions:
    - For describing point-like (inter)actions, like decay in flight

G4VProcess : actions summary

- The virtual « action » methods are following:
  - **AtRestGetPhysicalInteractionLength(), AtRestDoIt();**
  - **AlongStepGetPhysicalInteractionLength(), AlongStepDoIt();**
  - **PostStepGetPhysicalInteractionLength(), PostStepDoIt();**

- Other important virtual method:
  - **G4bool IsApplicable(const G4ParticleDefinition &);**
    - Used to check if a process can handle the given particle type
    - It is called by the kernel when you set up your physics list

G4VProcess: extensions

- A process can implement any combination of the three **AtRest**, **AlongStep** and **PostStep** actions:
  - decay = **AtRest** + **PostStep**

- If you plan to implement your own process:
  - A set on intermediate classes exist implementing various combinations of actions:
    - For example:
      - **G4VDiscreteProcess**: only **PostStep** actions
      - **G4VContinuousDiscreteProcess**: **AlongStep + PostStep** actions

G4ProcessManager

- **`G4ProcessManager`** maintains three vectors of actions :
  - One for the **`AtRest`** methods of the particle;
  - One for the **`AlongStep`** ones;
  - And one for the **`PostStep`** actions.
- Components of these vectors you have to set up in your "Physics List"
  - These vectors are used by the tracking.
- Note that the ordering of processes provided by/to the **`G4ProcessManager`** vectors is relevant and used by the stepping
  - There are few critical points you should be aware of
    - Multiple scattering can shift end point of a step and step length
    - Scintillation, Cerenkov and some other processes assuming that step and energy deposition at the step are defined

Adding a process in physics list

- Get the process manager of the particle:

```
G4ProcessManager* pmanager = particle->GetProcessManager();
```

- Add the process:

```
pmanager->AddProcess(new G4eIonisation,        -1, 2, 2);
```

  - The indices provided are these of the ordering in the **DoIt()** vectors
  - Which is by default reverse of the ordering of the **GetPhysicalInteractionLength()** one ! ☹
    - Index in **AtRestDoIt()** vector
    - Index in **AlongStepDoIt()** vector
    - Index in **PostStepDoIt()** vector
  - There are more utility methods to add a process, but above one is probably the most clear

About process ordering

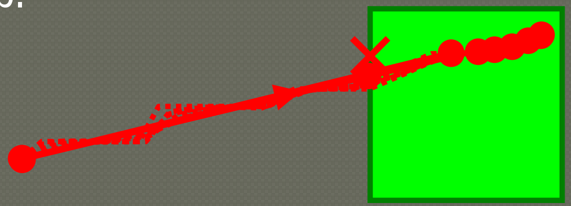
- **The most strong rule for multiple-scattering and transportation.**
- In your physics list, you should always have, for the ordering of the **`AlongGetPhysicalInteractionLength(…)`** methods:
  - Transportation last
    - For all particles
  - Multiple scattering second last
    - For charged particles only
      - assuming n processes

        [n-2] …

        [n-1] multiple scattering

        [n] transportation

- Why ?
  - Processes return a « true path length »;
  - The multiple scattering folds up this length into a *shorter* « geometrical » path length;
  - Based on this new length, the transportation can geometrically limits the step.



Displaying processes and particles

- When you application has started and when the run manager has been initialized, you can:

- Check the physics processes attached and their ordering:
  - `/particle/select e-`
  - `/particle/processes/dump`
- Check what particles exist:
  - `/particle/list`
- Check a particle property:
  - `/particle/select e-`
  - `/particle/property/dump`
- Please type "help" to get the full set of commands

Comment

- In 2011 Geant4 development plan there is a goal to extend user interface to process ordering:
  - Hidden numbers from users and force process ordering by Geant4 kernel at initialisation time

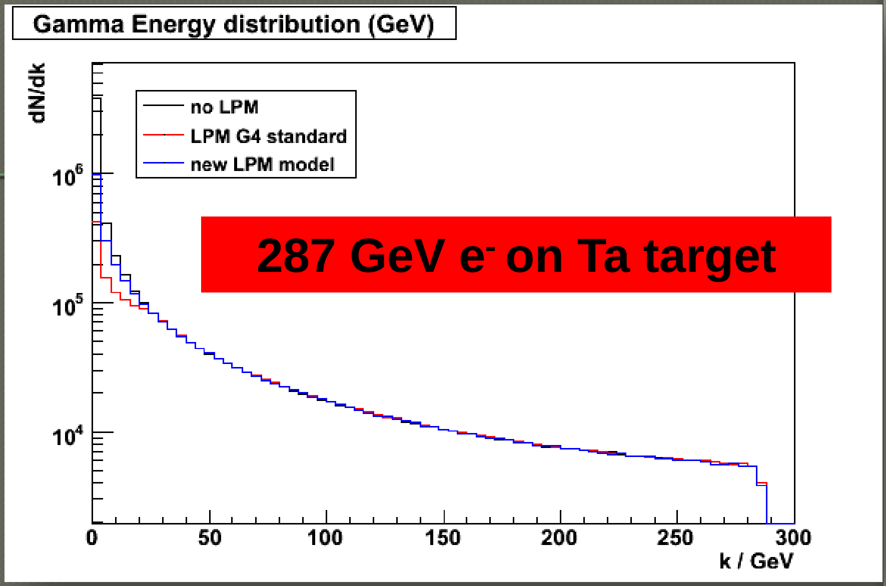
Geant4 cuts

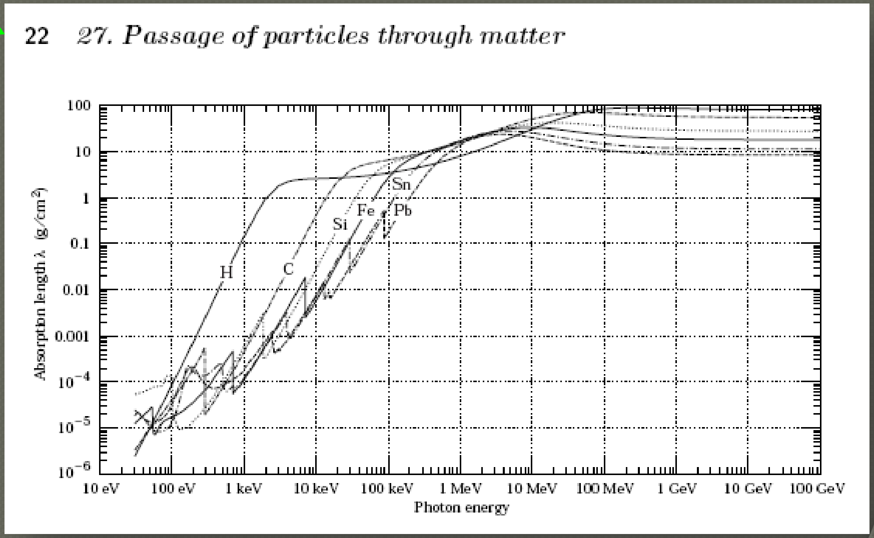
Geant4 approach for cuts

- All particles are tracked until it is killed by one of Geant4 process, for example:
  - Out of world volume
  - Inelastic interaction
  - Decay
- If kinetic energy is zero and there is no processes AtRest the particle is killed by stepping manager
- Geant4 by default has no tracking cut but only unique **cut in range**
  - Physically this means required spatial accuracy of simulation
  - This is the main difference between Geant4 and other simulation tools

Bremsstrahlung

- Bremsstrahlung spectrum grows to low energy as 1/k
  - k is the gamma energy
- Low energy gammas have very small absorption length





- Simulation of all low-energy gammas is non-effective
- Cuts/production threshold are used in all Monte Carlo codes
- Gamma emission below production threshold is taken into account as a continuous energy loss
- Similar approach is used for the ionisation process where spectrum of δ-electrons is proportional to $1/T^2$

Cut and production thresholds for energy loss processes

- User defines cut in range expressed in units of length
- Using this range Geant4 kernel compute production threshold T_{cut} for each material during initialization
- For a typical process (G4hIonisation, G4eIonisation, …), the production threshold T_{cut} subdivides the continuous and discrete parts of energy loss:
  - Mean rate of energy lost due to soft energy transfers
  - Total XS for discrete δ-electron production above T_{cut}

$$\frac{dE(E,T_{cut})}{dx} = n_{at} \int_0^{T_{cut}} T \frac{d\sigma(Z,E,T)}{dT} dT$$

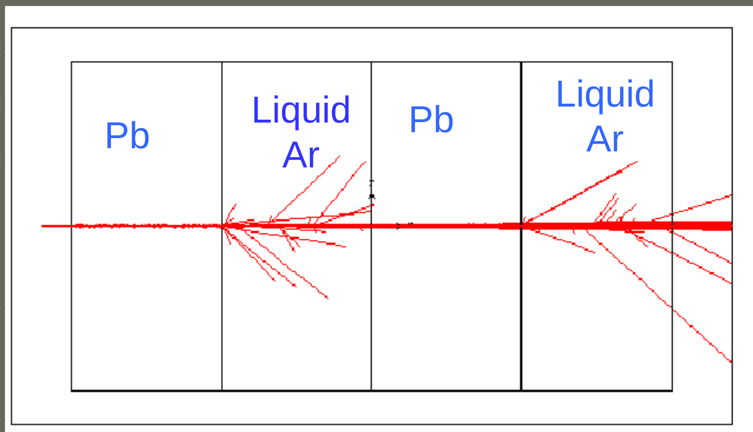
$$\sigma(Z,E,T_{cut}) = \int_{T_{cut}}^{T_{\max}} \frac{d\sigma(Z,E,T)}{dT} dT$$

- At each step energy deposition is sampled by a fluctuation model using the computed mean energy loss
- Optionally, energy loss may be modified :
  - for the generation of extra δ-electrons under the threshold when the track is in the vicinity of a geometrical boundary (sub-cutoff)
  - for the sampling of fluorescence and Auger–electrons emission
- 4-momentum balance is provided in all cases

Effect of production thresholds

500 MeV incident protons on EM Pb/LAr calorimeter

In Geant4



One sets the production threshold for delta rays as a unique range:

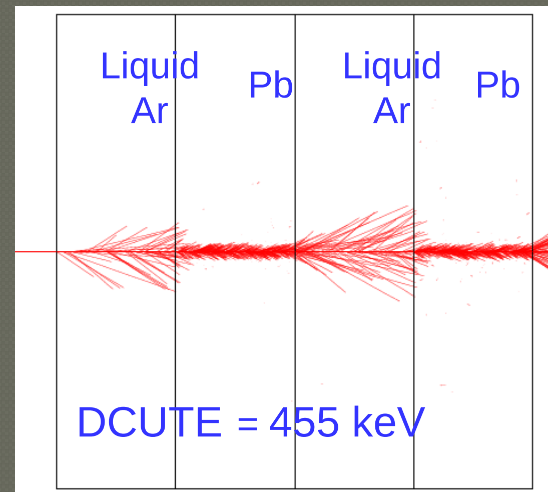
1.5 mm

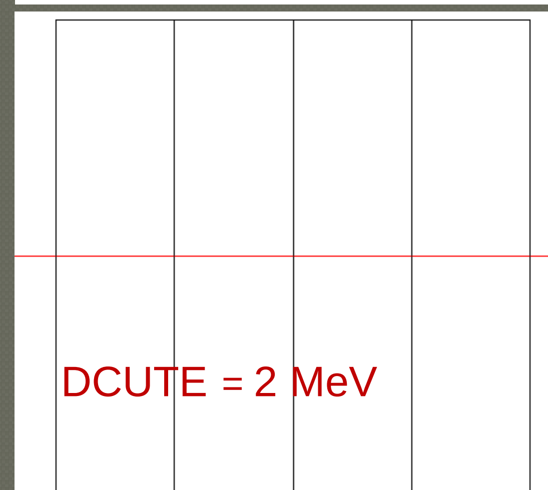
It is converted by Geant4 to energy:

T_c = 455 keV electron energy in liquid Ar

T_c = 2 MeV electron energy in Pb

In Geant3





one has to set the cut for delta-rays (DCUTE) as an energy threshold

either to the Liquid Argon value, thus producing many small unnecessary δ-rays in Pb,

or to the Pb value, thus killing the δ-rays production everywhere

What particles have cuts?

- Since Geant4 9.3 cuts are defined for
  - Gamma
  - Electron
  - Positron
  - Proton
- Cut for proton is used for all hadrons and ions by elastic scattering processes

Which processes use cuts ?

- It is not mandatory to use cuts
- Energy thresholds for gamma are used in Bremsstrahlung
- Energy thresholds for electrons are used in ionisation and e+e- pair production processes
- Energy threshold for positrons is used in the e+e- pair production process
- Energy thresholds for gamma and electrons are used optionally ("ApplyCuts" options) in all discrete processes
  - Photoelectric effect, Compton, gamma conversion
- Energy threshold for protons are used in processes of elastic scattering for hadrons and ions defining the threshold for kinetic energy of nuclear recoil

Comments

- Range cut approach was established for simulation of energy deposition inside solid or liquid media
  - Sampling and crystal calorimeters
  - Silicon tracking
- For specific user application, it may be revised, for example, by defining different cuts in range for electron and gamma
  - Gaseous detectors
  - Muon system
- Tracking cuts may be also used (saving some CPU) for simulation of penetration via shielding or for simulation in non-sensitive part of the apparatus
  - Astrophysics applications

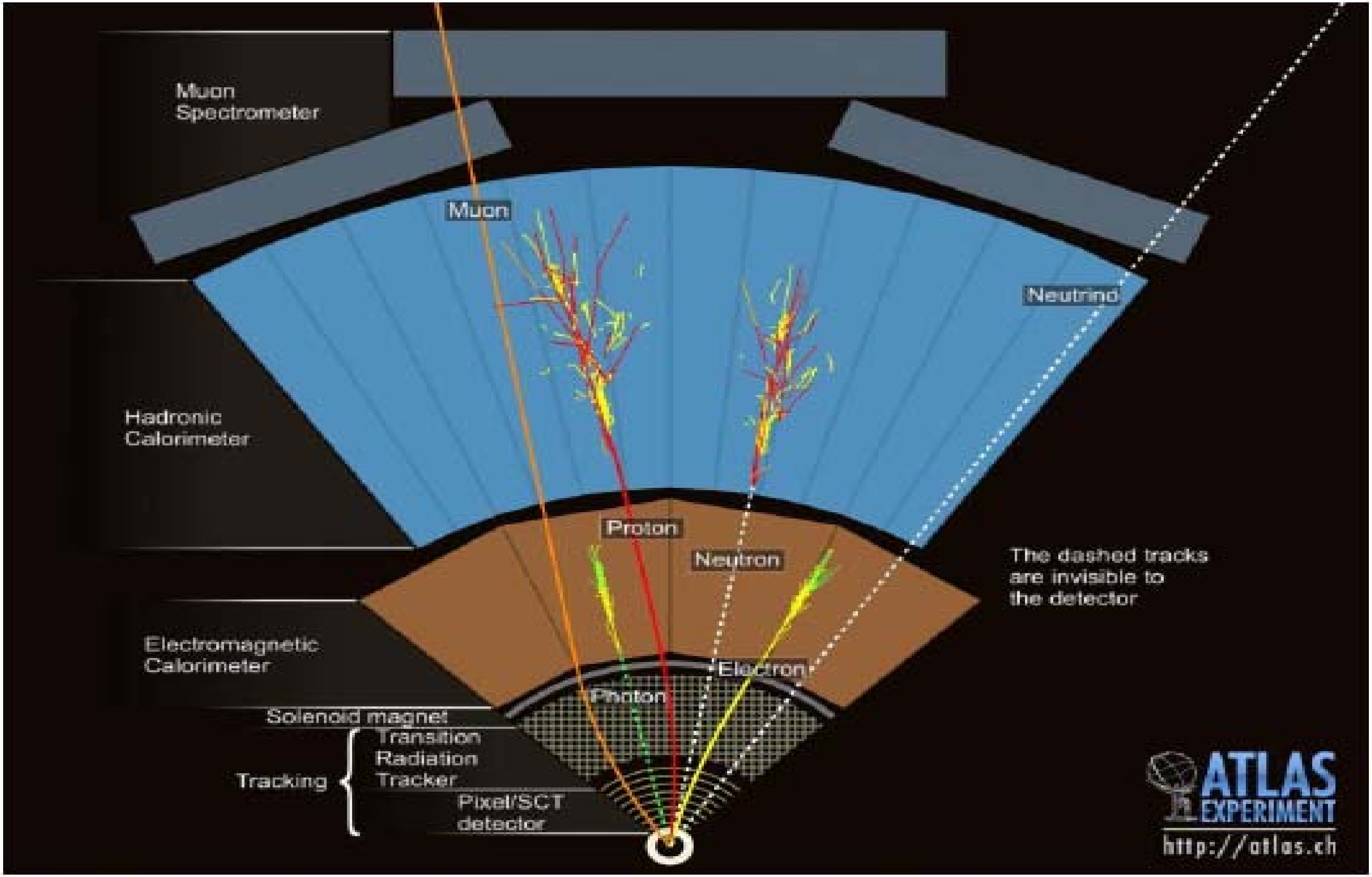
How to define cut?

- Using UI interface to geant4 kernel:
  - /run/setCut 0.1 mm
  - /run/setCutForAGivenParticle e- 10 um
- Implementing virtual method SetCuts() of G4VUserPhysicsList
- In Geant4 examples several different implementations of cut definition in user code are shown
  - Including user defined UI commands
  - $G4INSTALL/examples/extended/electromagnetic

Cuts per G4Region

- Uniform cut in range providing balanced simulation of particle transport in media with different density
- Requirements for precision in different part of complex geometry may be very different
  - Micron precision in tracking devices millimeter precision in calorimeters
  - Unique value of the cut in range may be not effective and not practical

Geant4 simulation of ATLAS experiment at LHC, CERN



Cuts per G4Region

- Geometrical volumes may be assigned to G4Region
- By default the only one G4Region is created associated with the World volume
- If more than one G4Region is created it is possible to have different cut values

How to define cut for G4Region?

- Using UI interface to geant4 kernel:
  - /run/setCutForRegion VertexDetector 1 um
- Implementing virtual method SetCuts() of G4VUserPhysicsList
- Examples are available:
  - $G4INSTALL/examples/extended/electromagnetic
  - TestEm8 – simple gaseous detector
  - TestEm9 – more complicate setup with tracker and muon detectors
- To printout cut values and production thresholds use UI command:
  - /run/particle/dumpCutValues

Thank you for your attention!