Experiment Simulation

CERN School of Computing 2006
Helsinki

Lecture 2

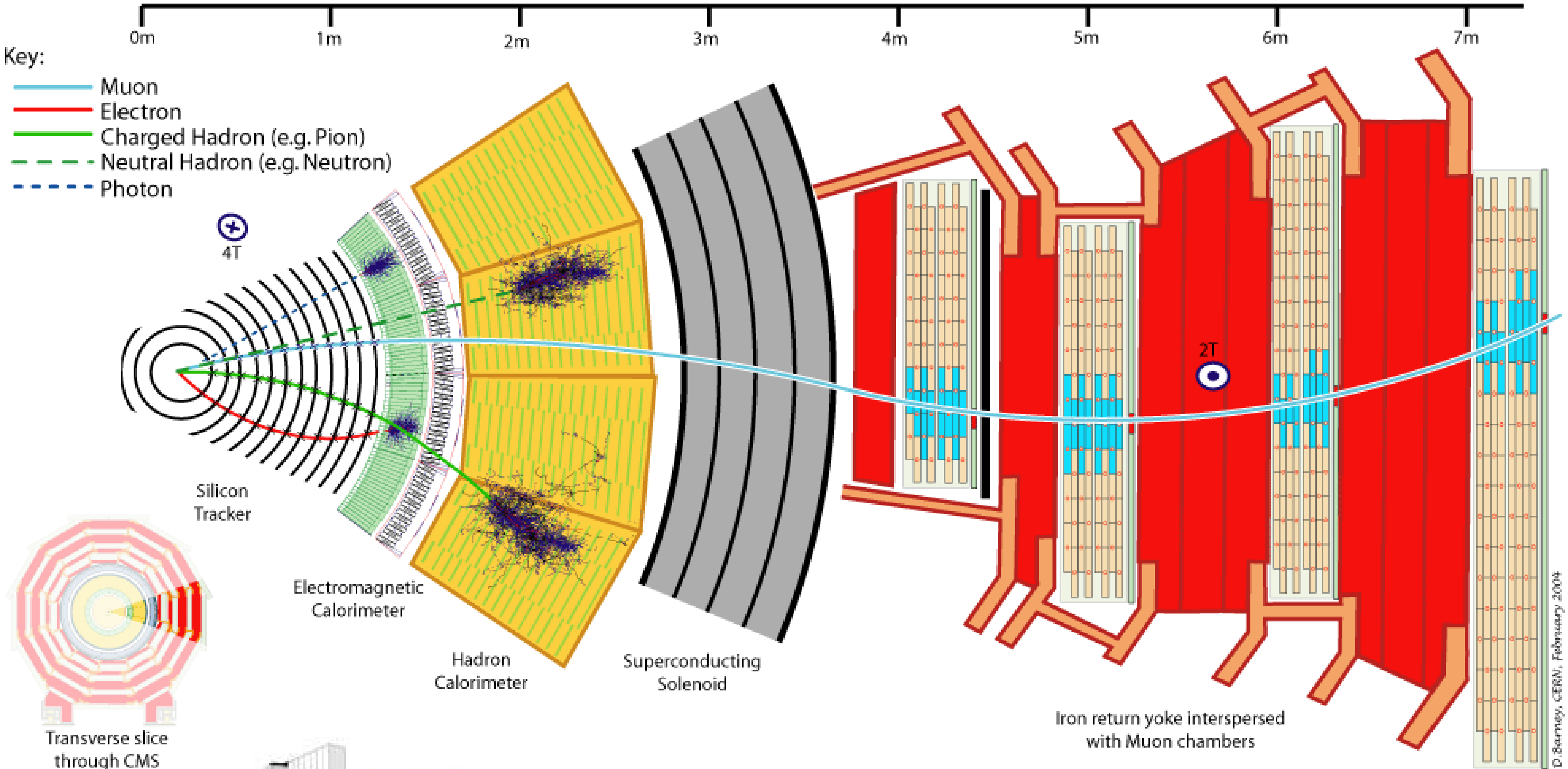
SWX SWISS EXCHANGE

Martin Liendl
SWX Swiss Exchange

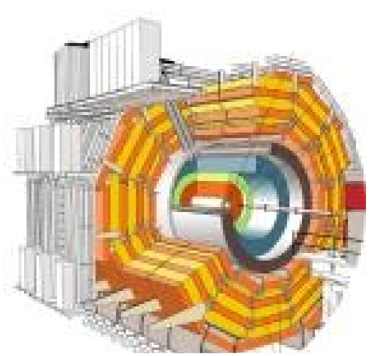
Overview Lecture ~2~

- Develop an understanding of the **Monte Carlo Method**
  - Passage of particles through matter
  - Random numbers, distributions
  - Basic tracking algorithm
- Introduction to **GEANT4**
  - Particles & primary particles
  - Materials

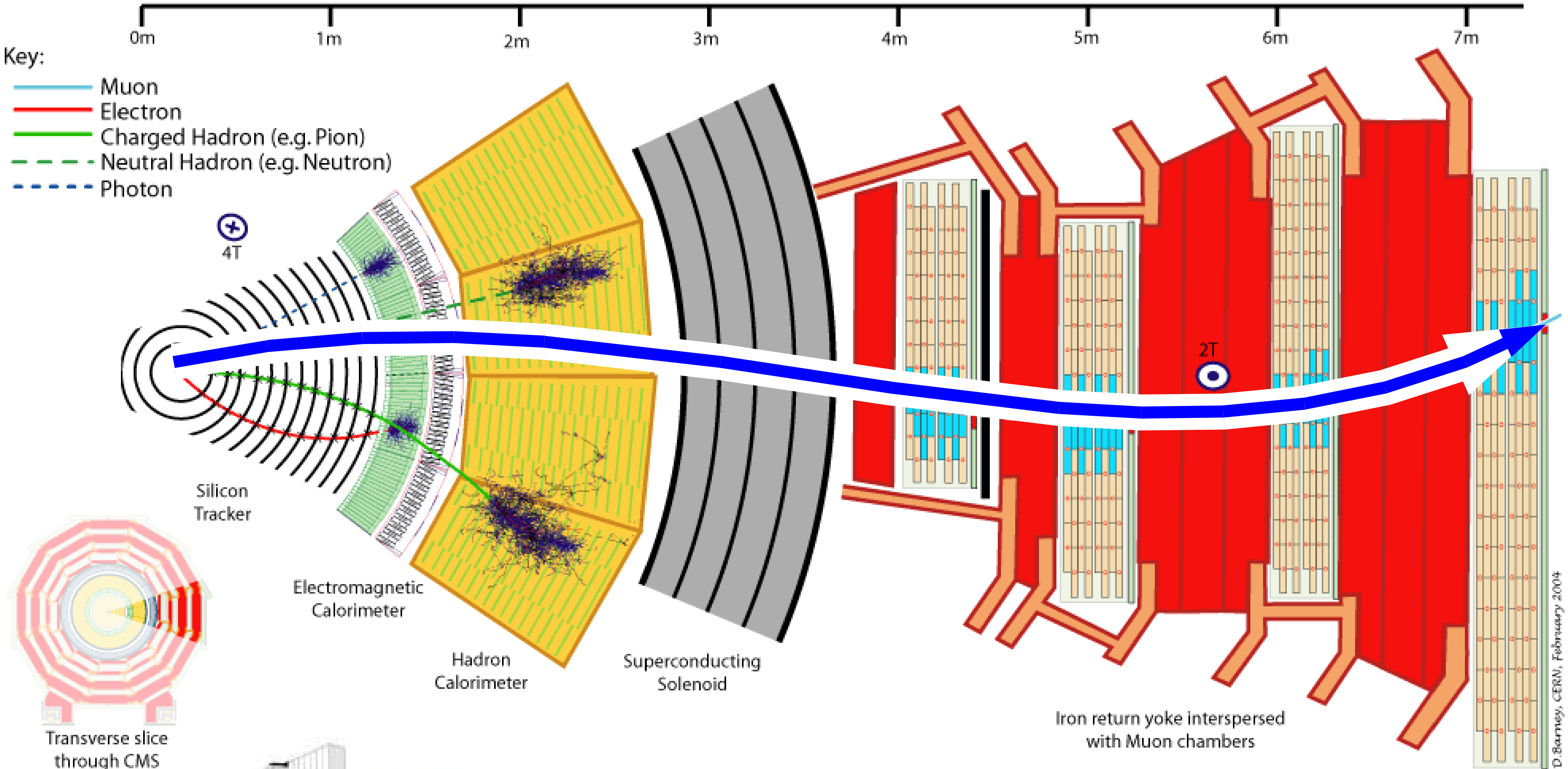
Our aim..

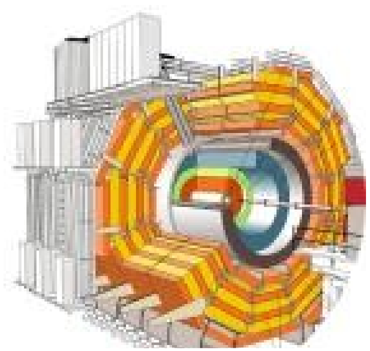


0m
1m
2m
3m
4m
5m
6m
7m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon
4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
Transverse slice through CMS
D.Barney, CERN, February 2004



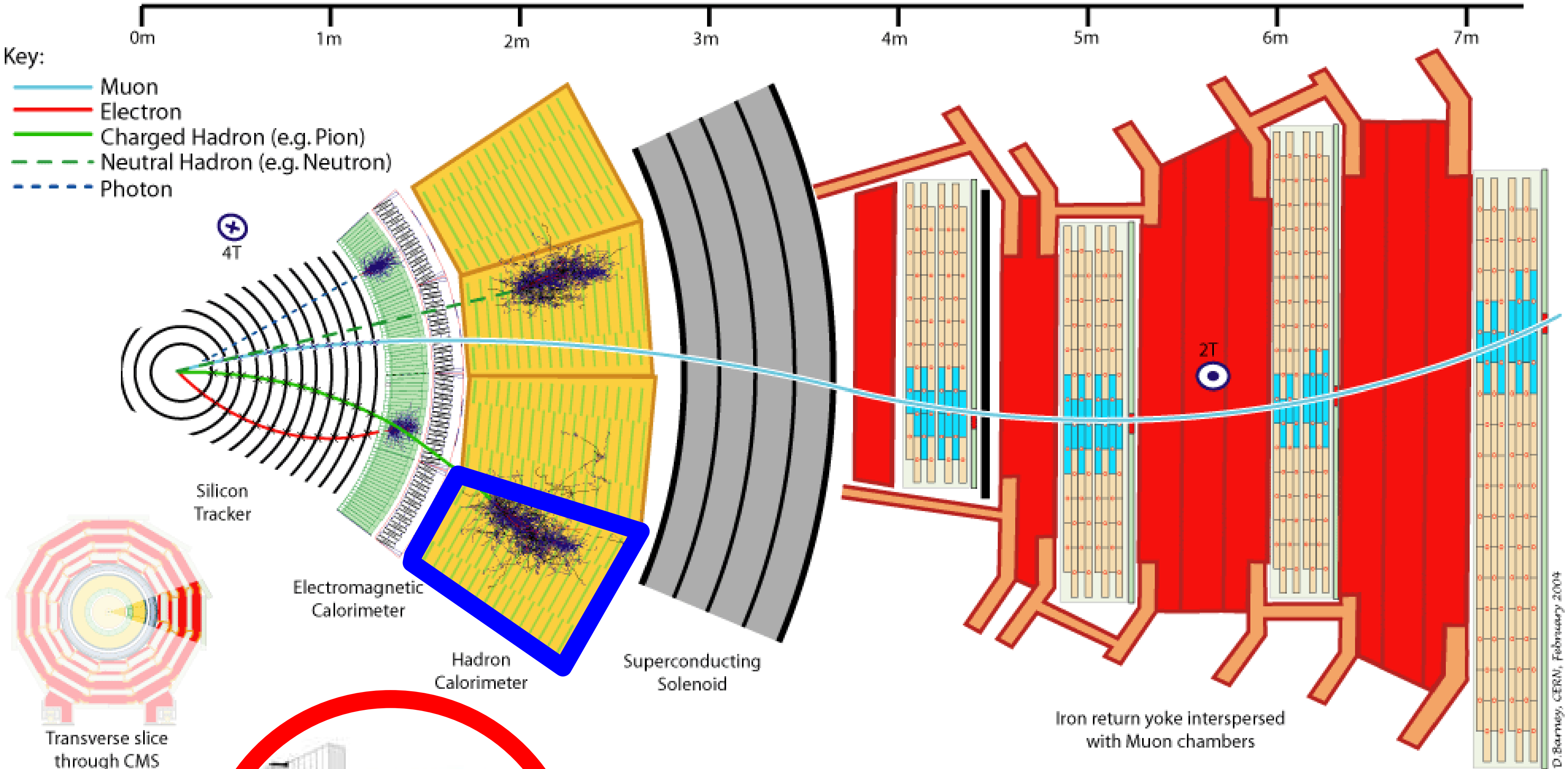
Understand & define procedures ..

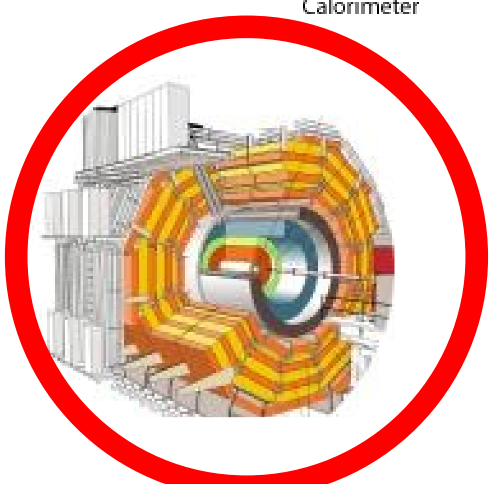




.. to describe the fate of **single particles** interacting with **macroscopic matter** based on the **microscopic description of particle interactions**

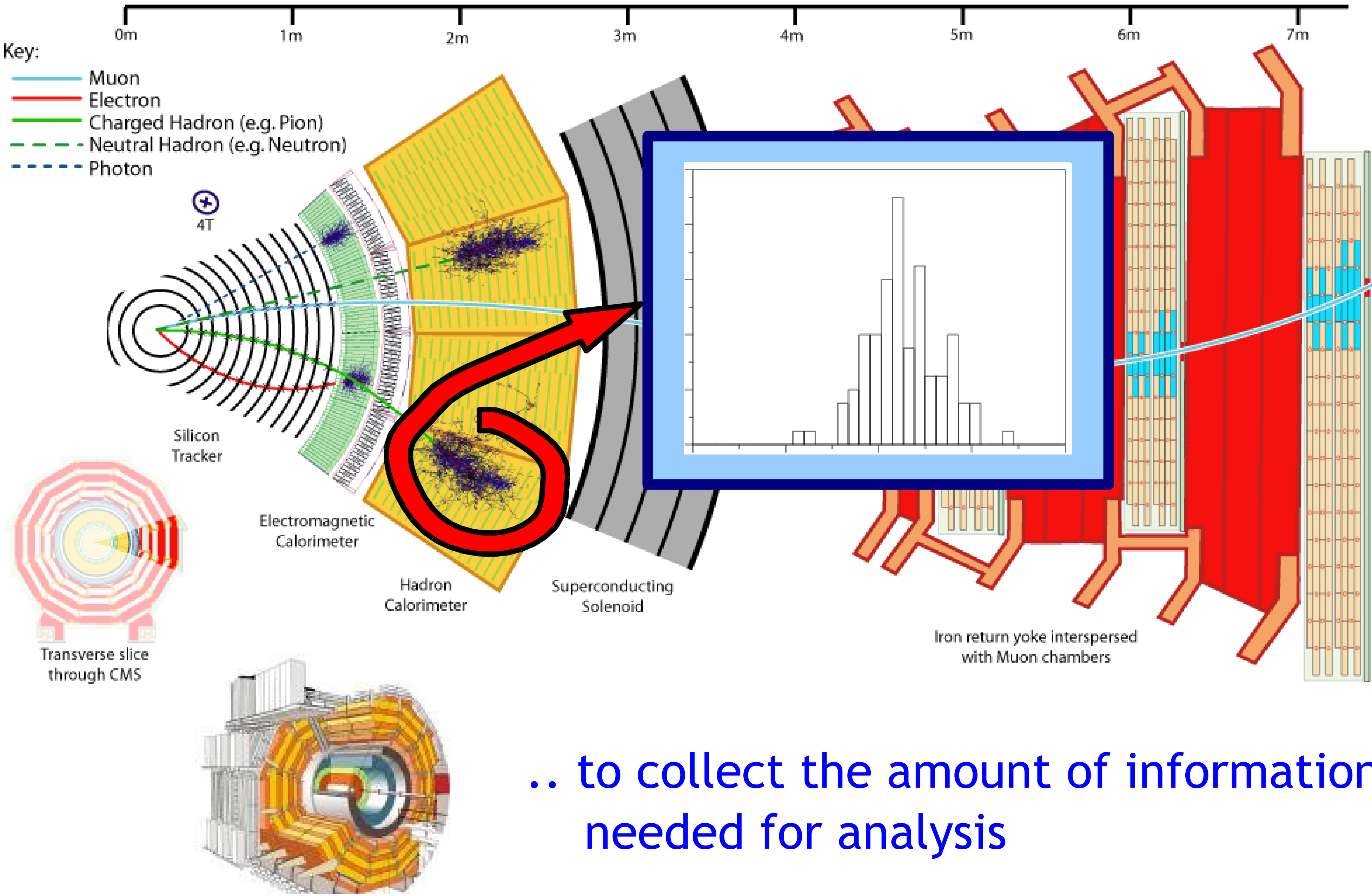
Understand & define procedures ..





.. to describe complex detectors in relevant detail

Understand & define procedures ..



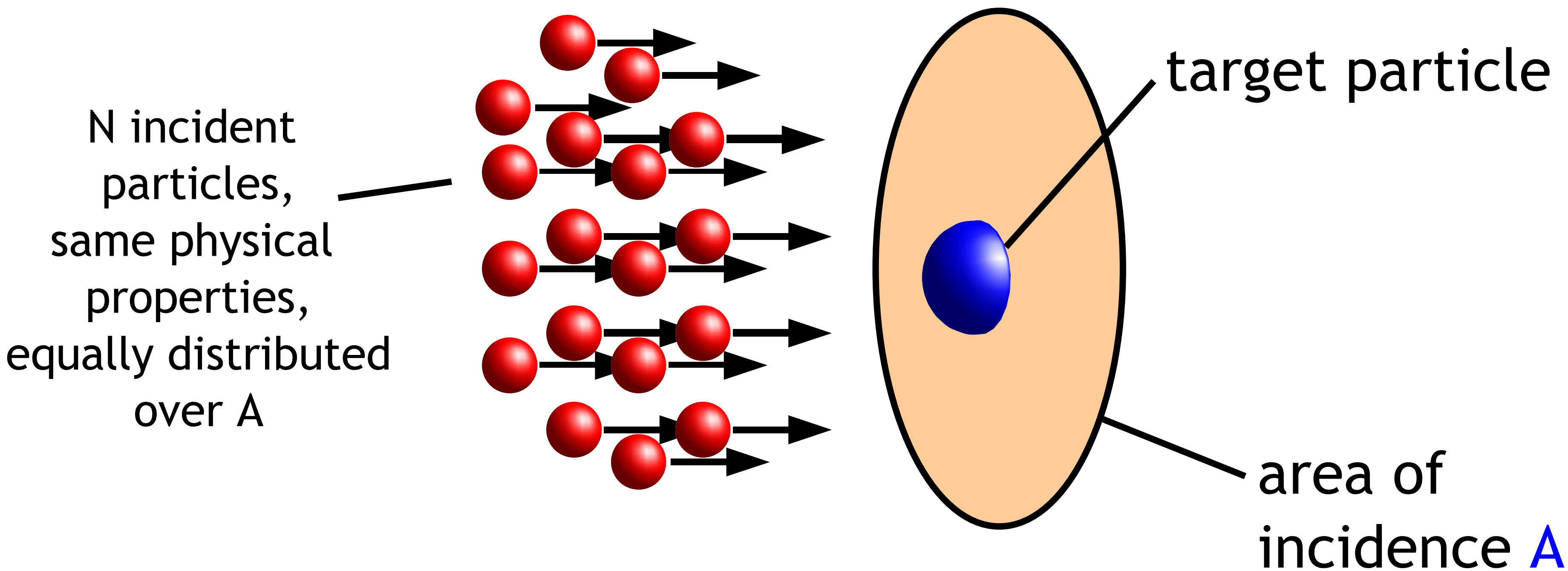
.. to collect the amount of information needed for analysis

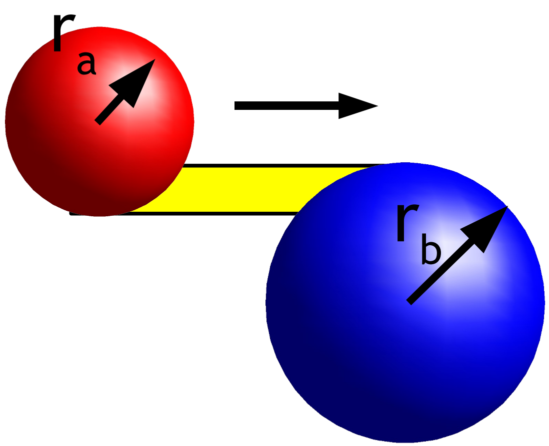
Starting point

- Single particle interaction 
  - particle type
    - electron, muon, atom (nucleus and/or shell), ...
  - dynamical properties:
    - momentum, energy, spin, ...
- Quantum mechanics of the interaction type (process):
  - well understood theories
    - classical QM, QED, phenomenological theories, ...
    - supported by lots of reference data
  - formulae for the total and the differential **cross-section**
    - quantitative description of the interaction probabilities

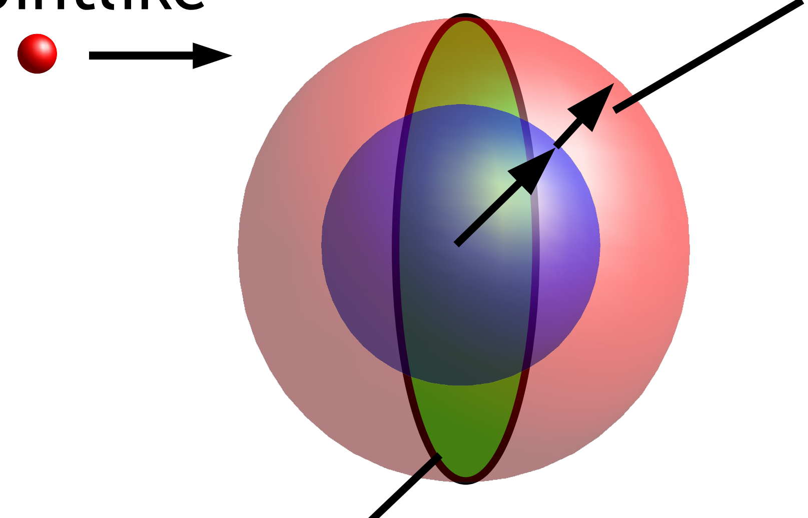
Total cross-section

- What is the **probability** of an interaction of a given physics process between two particles
  - at the crossing of the particles
  - within the **unit area** perpendicular to the flight path
- Geometrical interpretation:



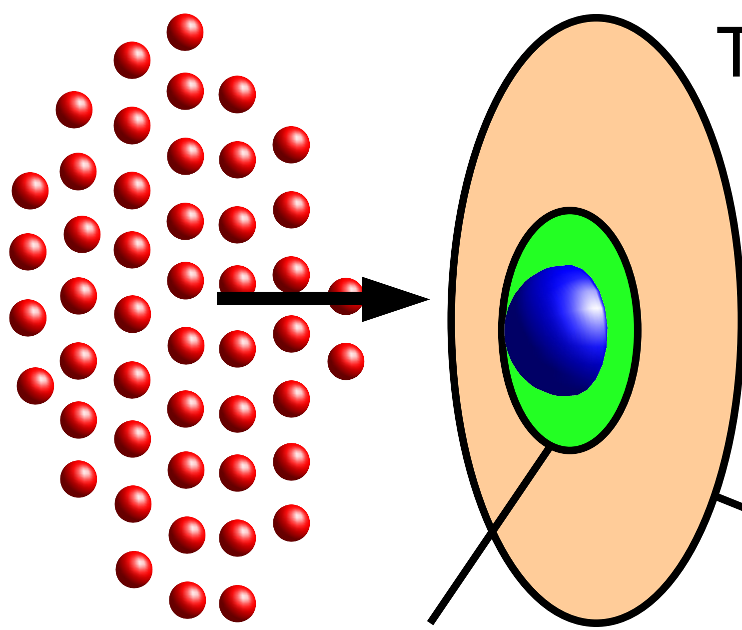


=



$r_a + r_b$ ~ max. distance of inter-action of both particles

$\sigma = (r_a + r_b)^2 \, \pi$

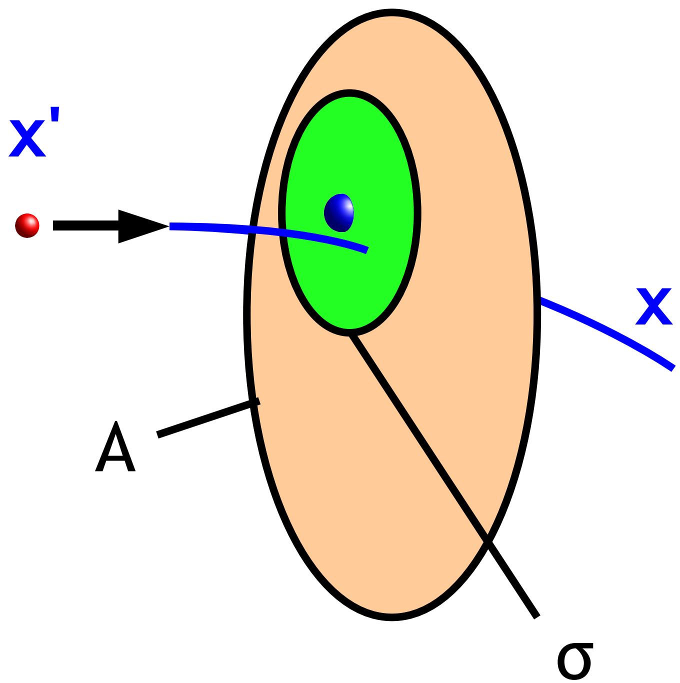


The fraction of $\frac{\sigma}{A}$ of the incident particles will interact with the target

σ is measured as an area (e.g. in barns: 1 barn = 10^{-28}m^2)

area of incidence A

σ .. cross section



Equivalent:
incident particle and target uniformly distributed at random over A

$\frac{\sigma}{A}$ .. probability of interaction

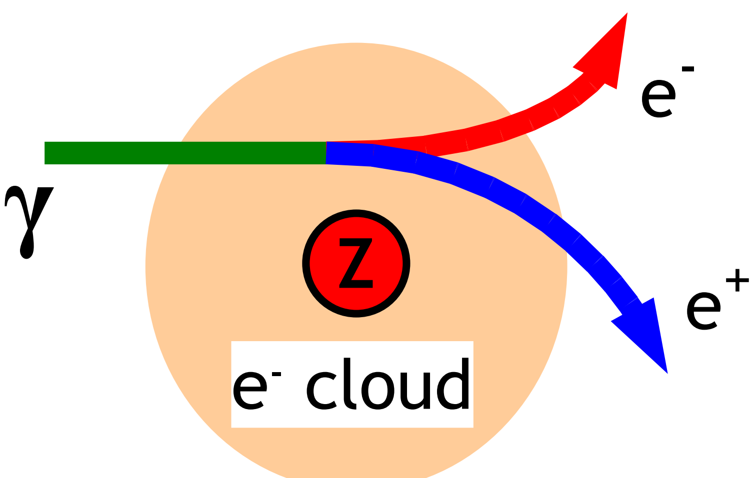
(assuming A > σ)

Quantum mechanical interpretation:
σ is proportional to the transition probabilities of the quantum states before (t=-∞) and after (t=+∞) the interaction.

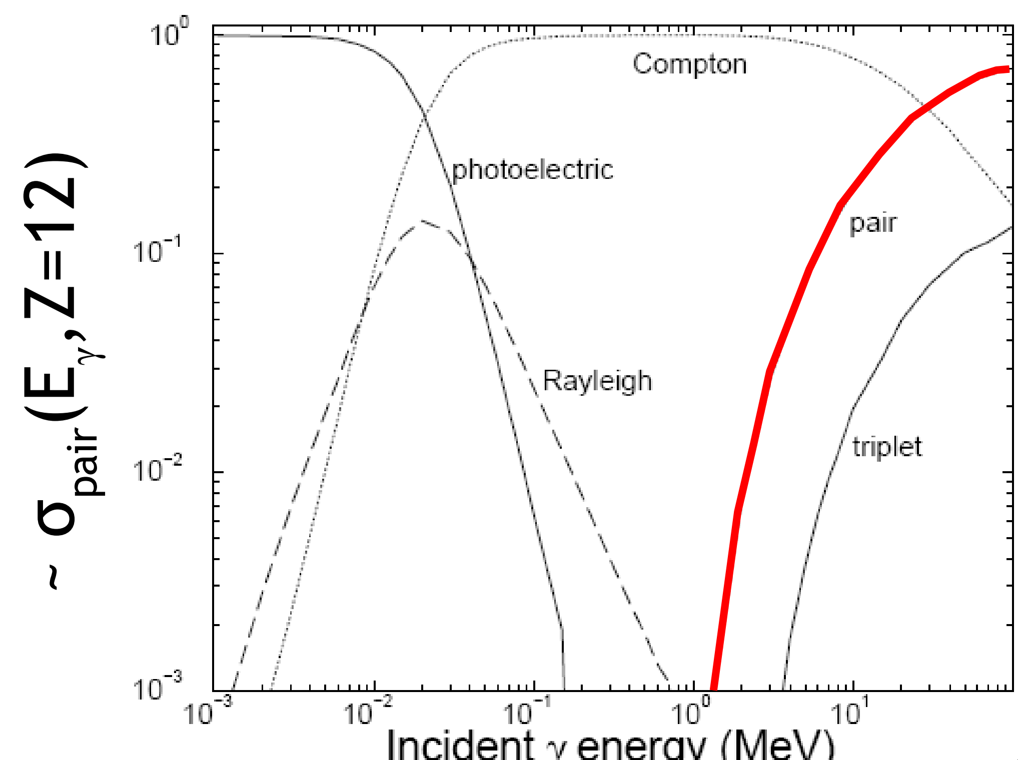
Example: pair creation

- Conversion of a photon γ into an electron-positron pair in the field of an atom
- The total cross-section is a function of
  - the γ energy (energy of the incident photon)
  - the charge Z of the nucleus

$$\sigma_{pair} = \sigma_{pair}(E_\gamma, Z)$$



Carbon, Z=12



Differential cross-section

- The total cross section is a measure of how likely a distinct interaction process will occur
  - varies with the energy of the interacting particles
- The differential cross-section describes the details of this interaction
  - e.g. in an scattering experiment
    - distribution of the scattering angles
    - distribution of energy distribution btw. the scattered particles
  - total and differential cross-section are related:

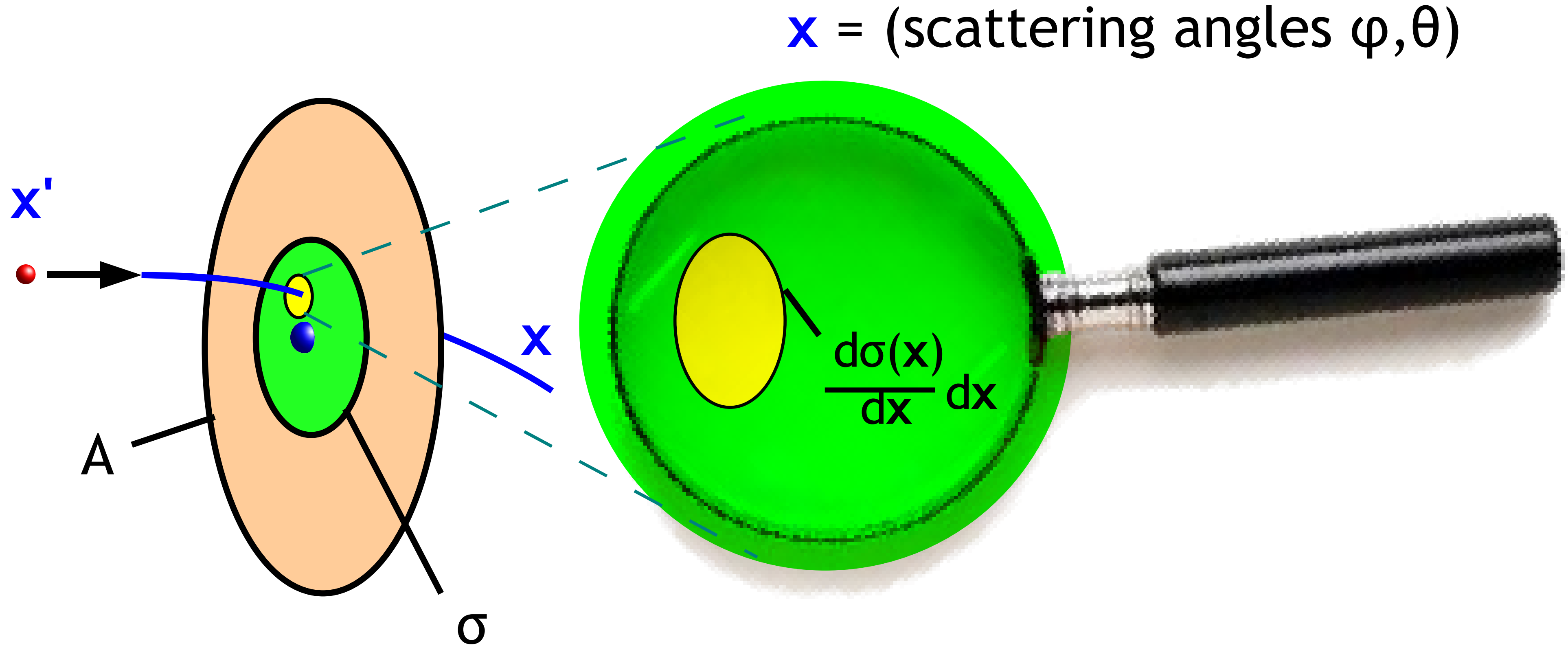
$$\sigma = \int \frac{d\sigma(x_1, x_2, ..)}{dx_1 dx_2 ..} dx_1 dx_2 .., $$

x_i .. state variables of the particles after the interaction

(Remark)

- Differential cross-section ~ probability distribution
  - we can never predict the outcome of one interaction deterministically
  - the “final states” are distributed according to the differential cross-section
- When we measure the interaction of one single particle, the result can be thought of drawing a sample of the probability distribution described by quantum mechanics
- The aim of HEP experiments is to measure estimates of distributions in order to conclude whether or not they are estimates of distributions described by the theory under test

Example: elastic scattering

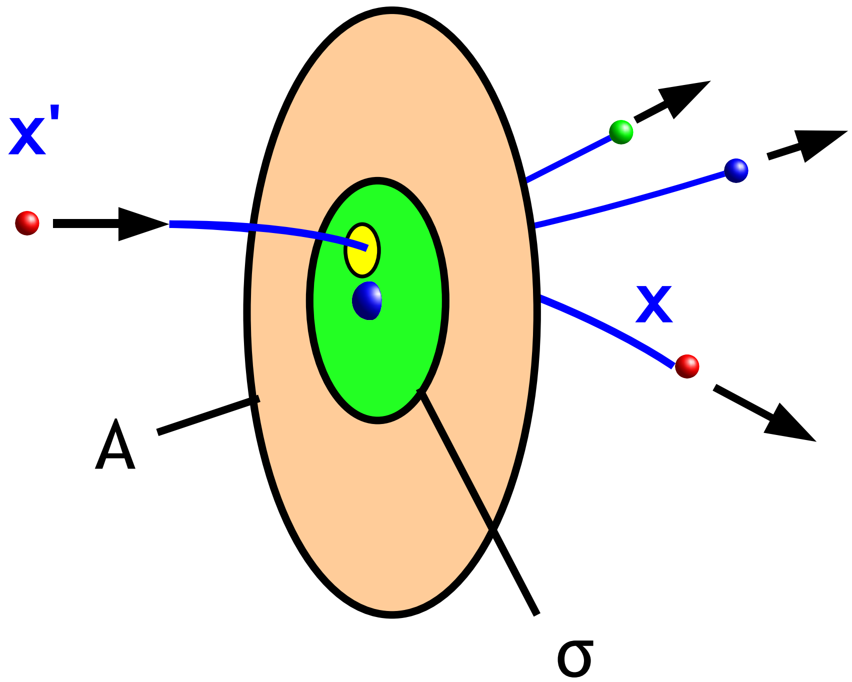


$$\sigma = \int \frac{d\sigma(\mathbf{x})}{d\mathbf{x}} d\mathbf{x}$$

$\frac{\sigma}{A}$ .. probability of interaction

$\frac{1}{A} \frac{d\sigma(\mathbf{x})}{d\mathbf{x}} d\mathbf{x}$ .. prob. of interaction into final state $\mathbf{x}$

More general:



x = (number and momenta of all particles after the interaction)

$$\sigma = \int \frac{d\sigma(\mathbf{x})}{d\mathbf{x}} \, d\mathbf{x}$$

$\frac{\sigma}{A}$ .. probability of interaction

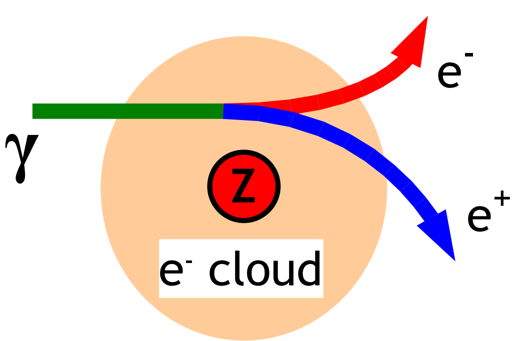
$\frac{1}{A} \frac{d\sigma(\mathbf{x})}{d\mathbf{x}} d\mathbf{x}$ .. prob. of interaction into final state **x**

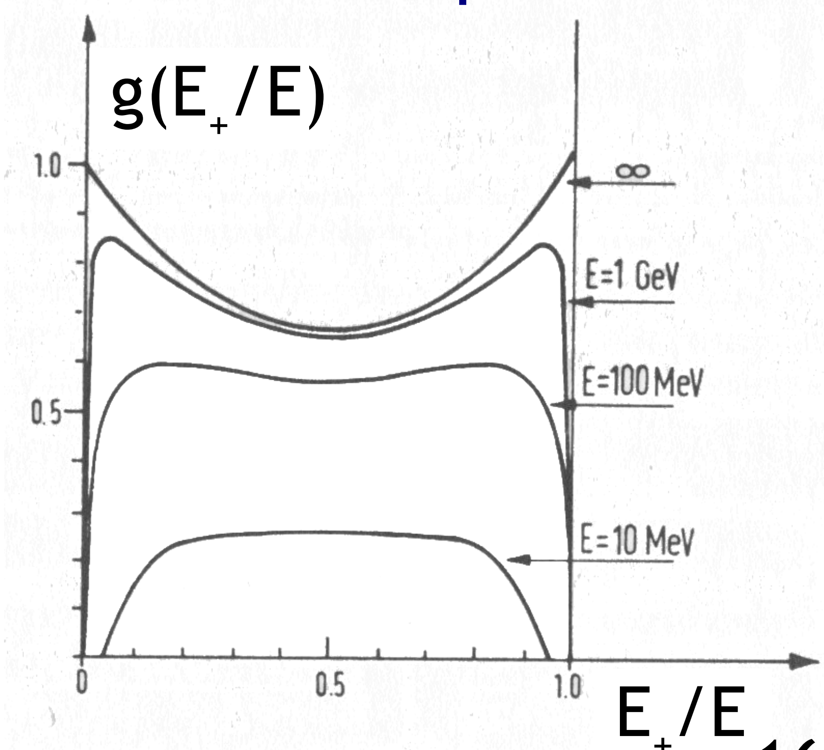
Example: pair creation

- polar angle distribution of the e^- or e^+ final state:
  - kinematic constraints into the "forward" direction
- azimuth angle distribution of the electron final state:
  - isotropic, $d\sigma/d\varphi = 1/360^0$
- energy distribution between the electron and photon:
  - asymmetric:

E .. energy of γ
E_+ .. energy of e^+

$1/E\, g(E_+/E)dE_+$ = probability of having an e^+ with energy E_+ provided a pair production has taken place from a photon with energy E





Martin Liendl

More than one process

- Usually we have more than one interaction type which can affect a particle-particle crossing
- We assume that we can treat different kind of interactions independently
  - luckily this is often possible and yields good results!
  - otherwise: use, for example, empirical physics models
- The total cross-section for any of the involved processes to occur is then simply the sum of the cross-sections of the individual processes:

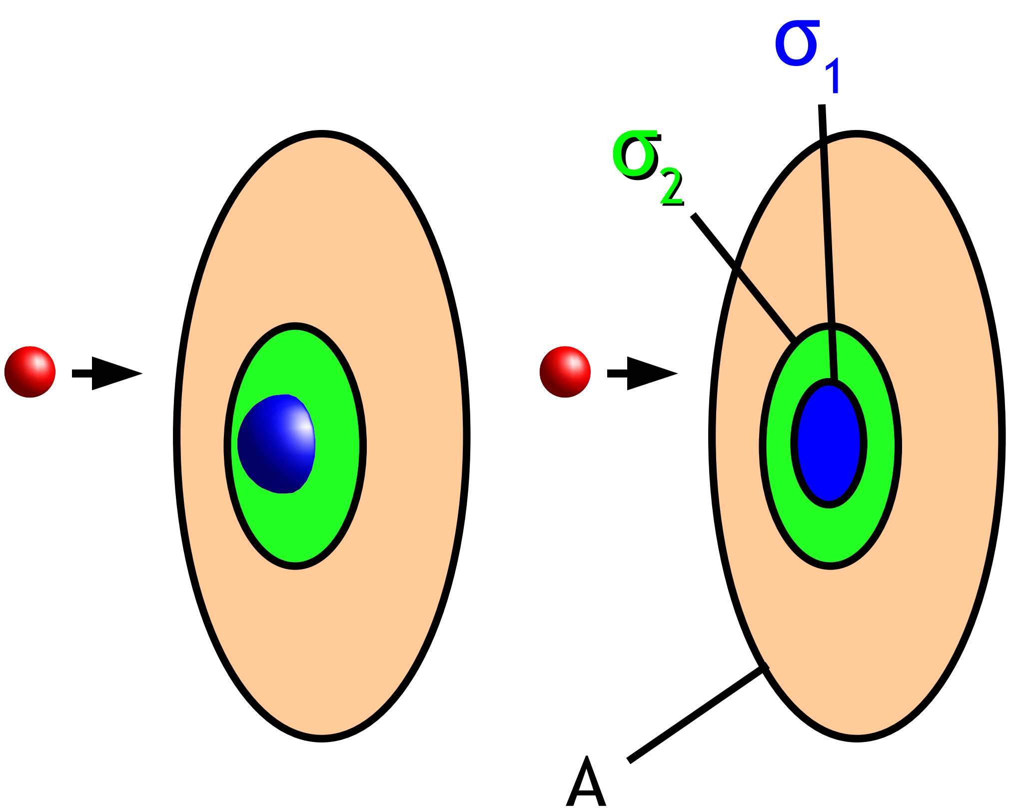
$$\sigma = \sigma_1 + \sigma_2 + \ldots + \sigma_n$$

More than one process

$\sigma = \sigma_1 + \sigma_2$

σ_1 .. particle gets stuck with target

σ_2 .. particle is scattered from target



σ_1/A .. probability of getting stuck

σ_2/A .. probability of getting scattered

σ/A ... probability of something to happen

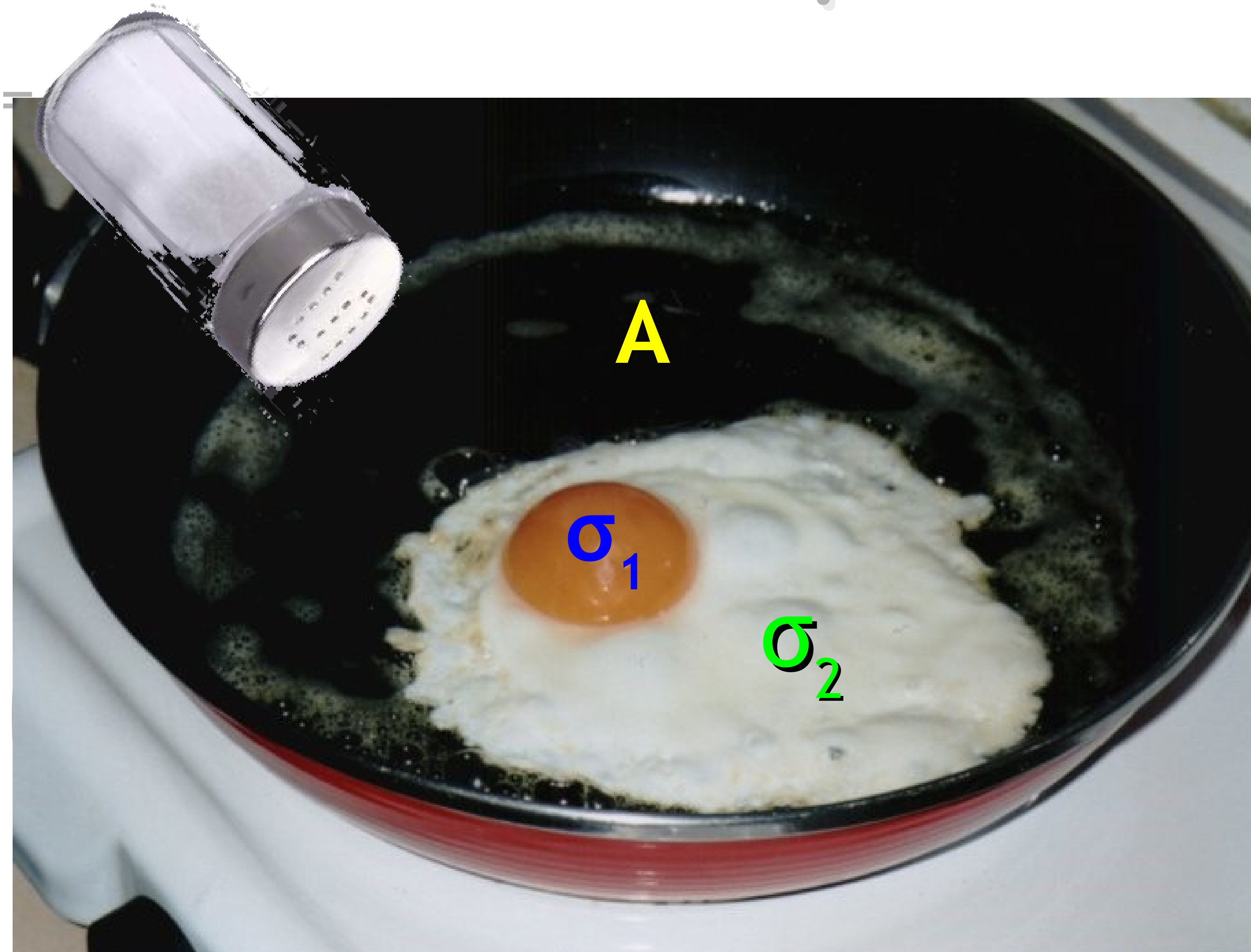
:-)

More than one process

σ =

σ_1

σ_2



of

of

of

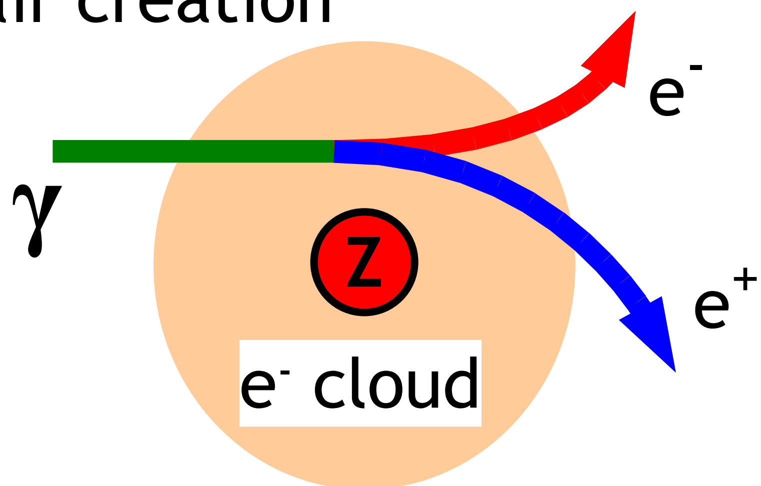
en

A

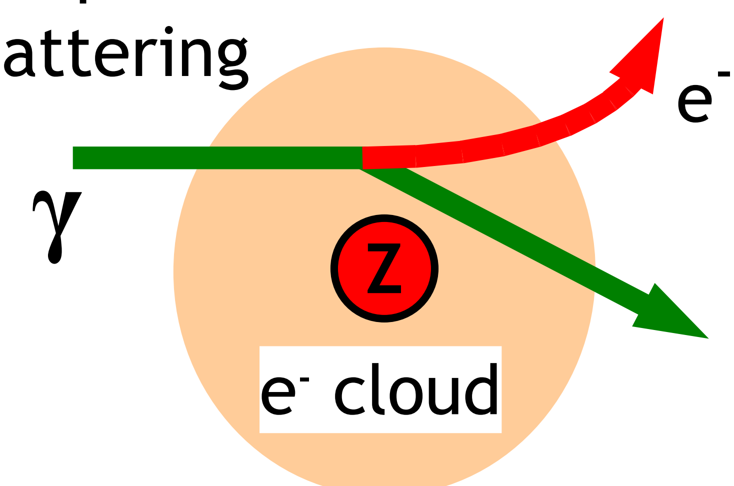
Example: photon passing atom

The pair creation process is just one of the possibilities

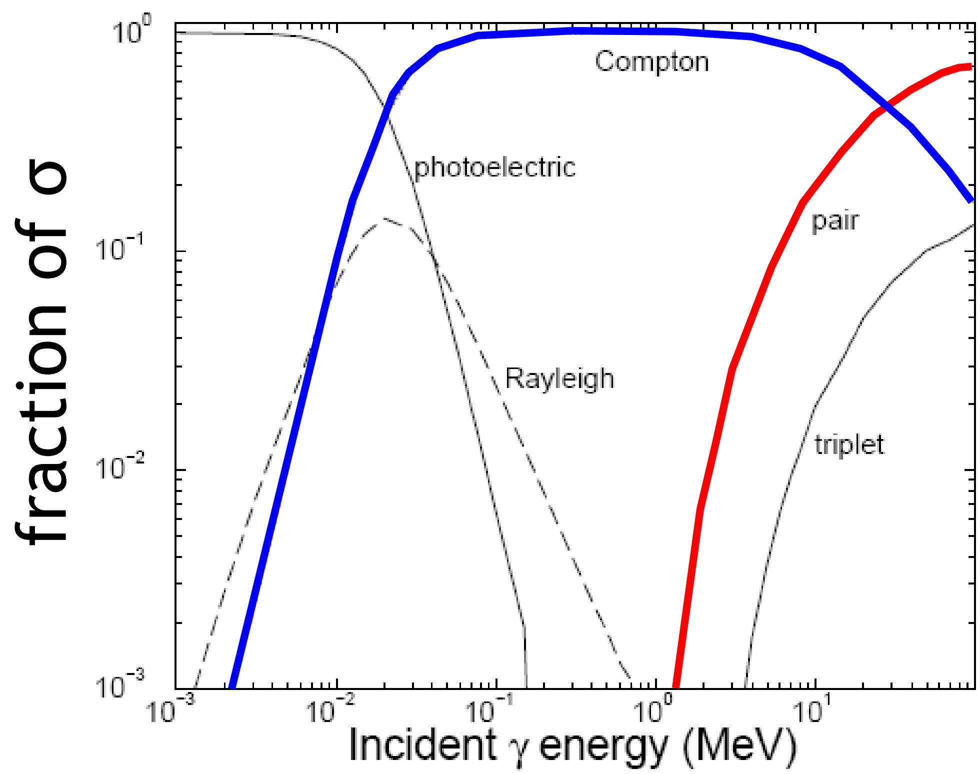
pair creation



Compton scattering

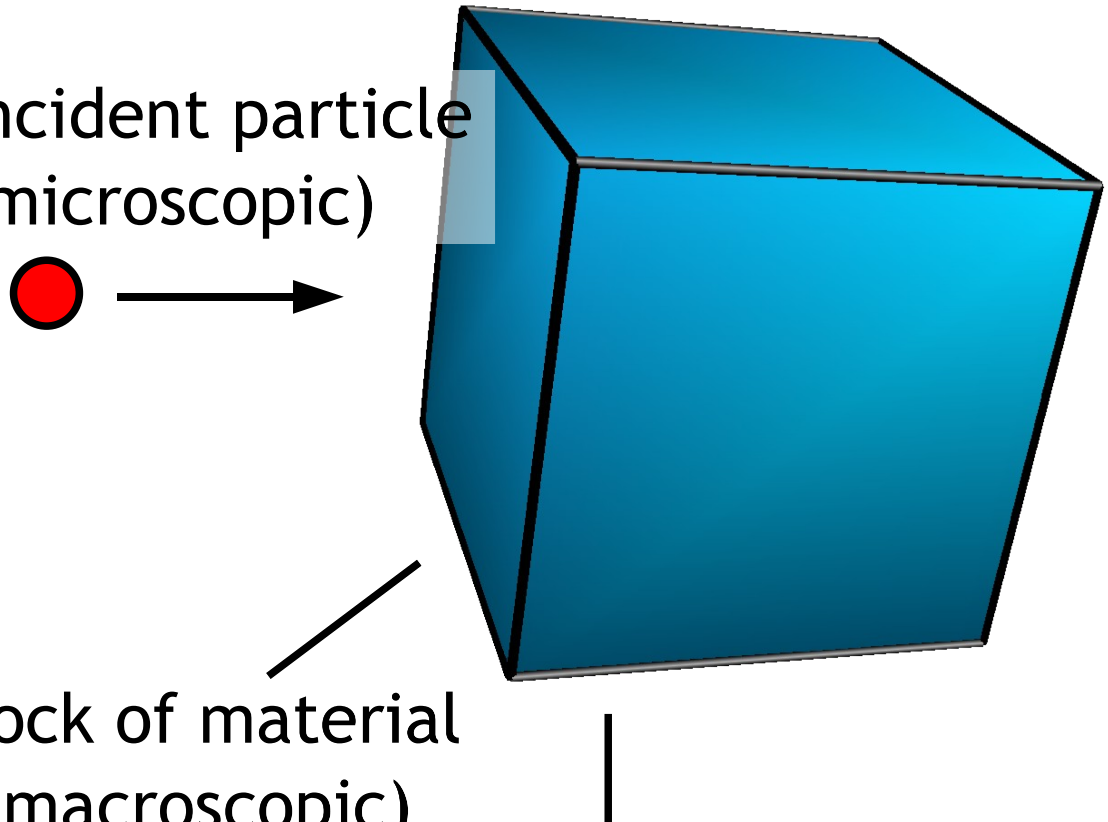


Carbon, Z=12



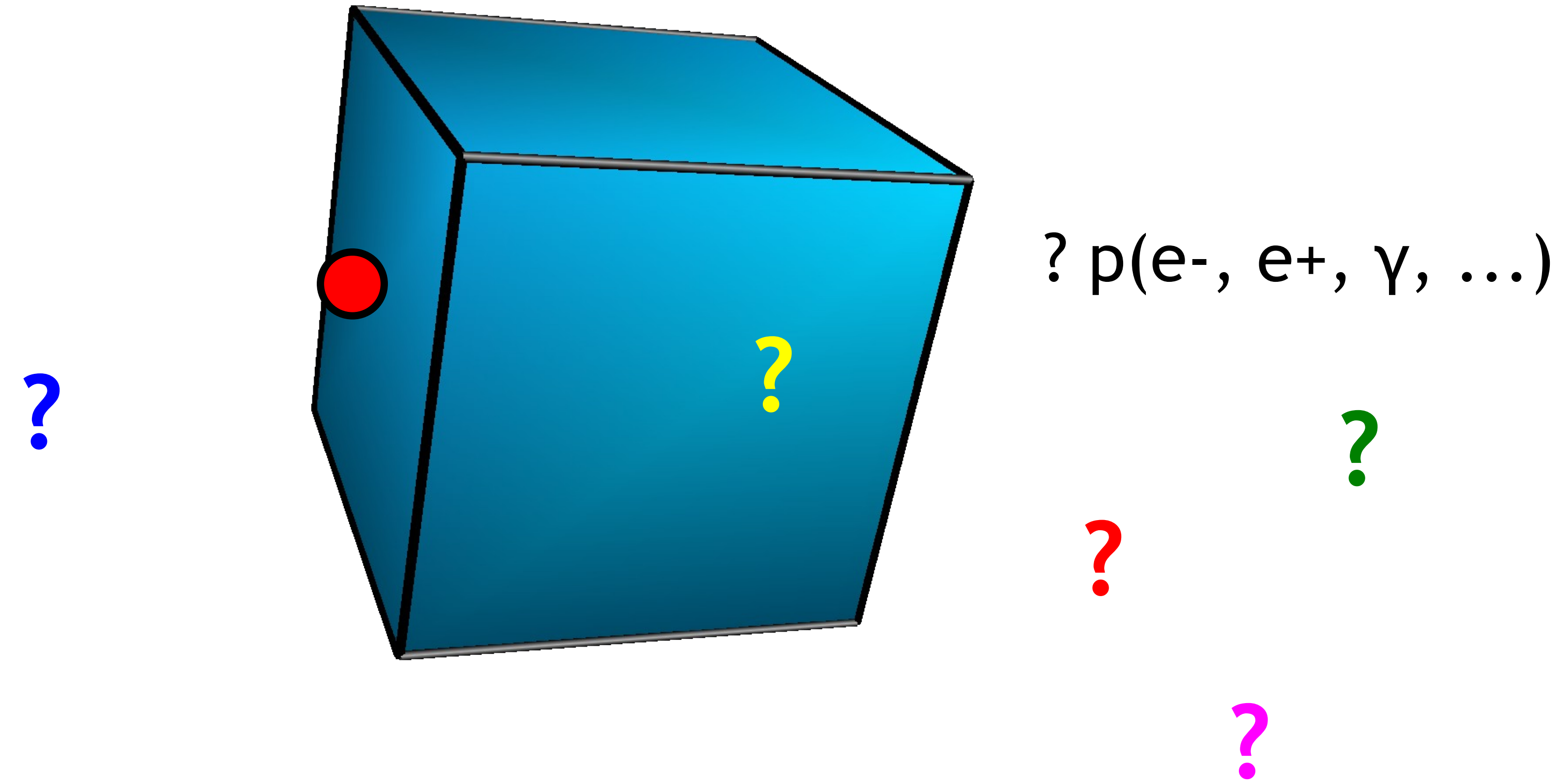
The problem: many more particles!!

We have not only one particle-particle interaction ...



The question:

After the incident particle has hit the block:
what is the probability distribution of finding it or any of its interaction products ANYWHERE in space???



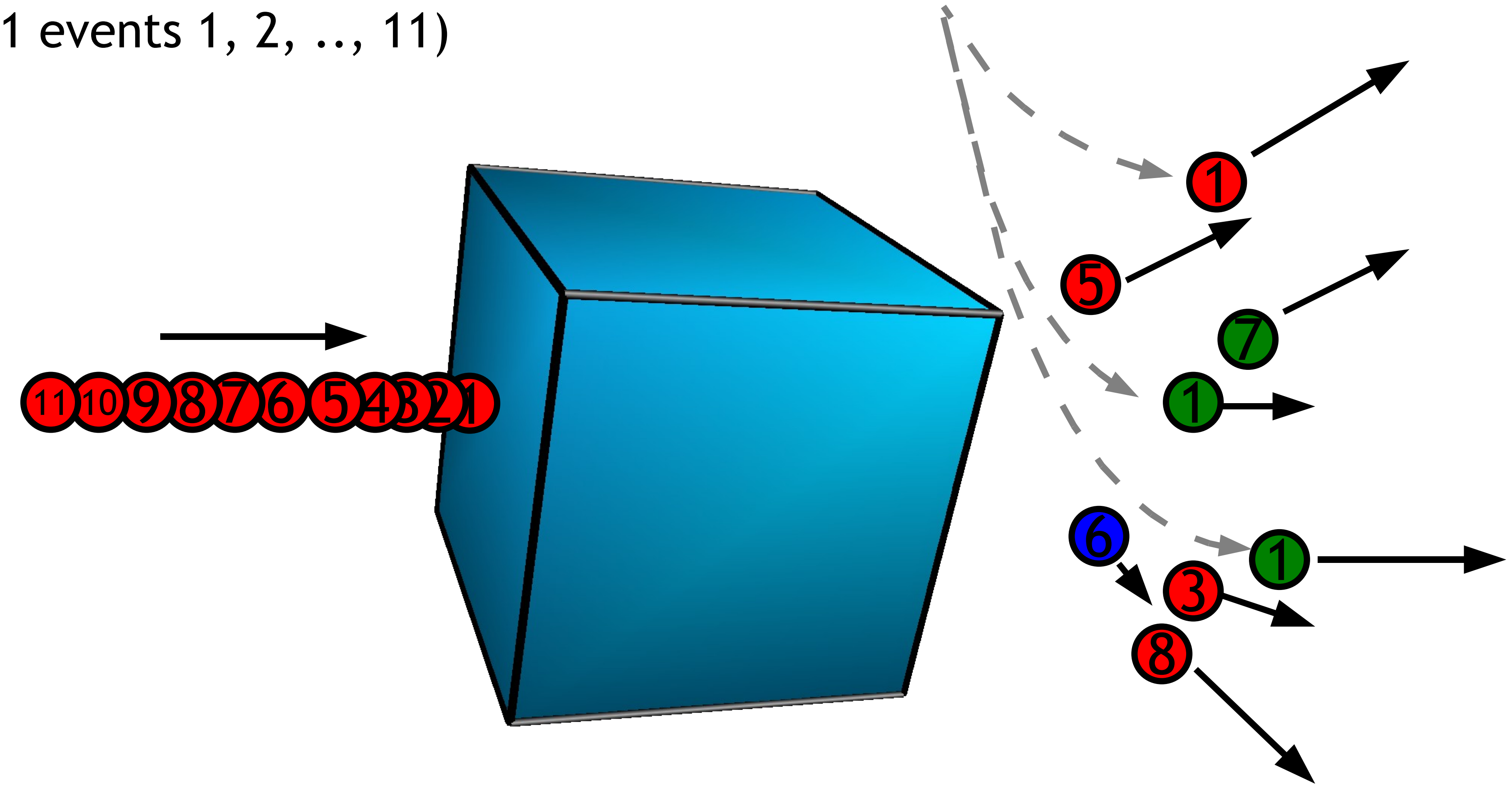
Monte Carlo Method

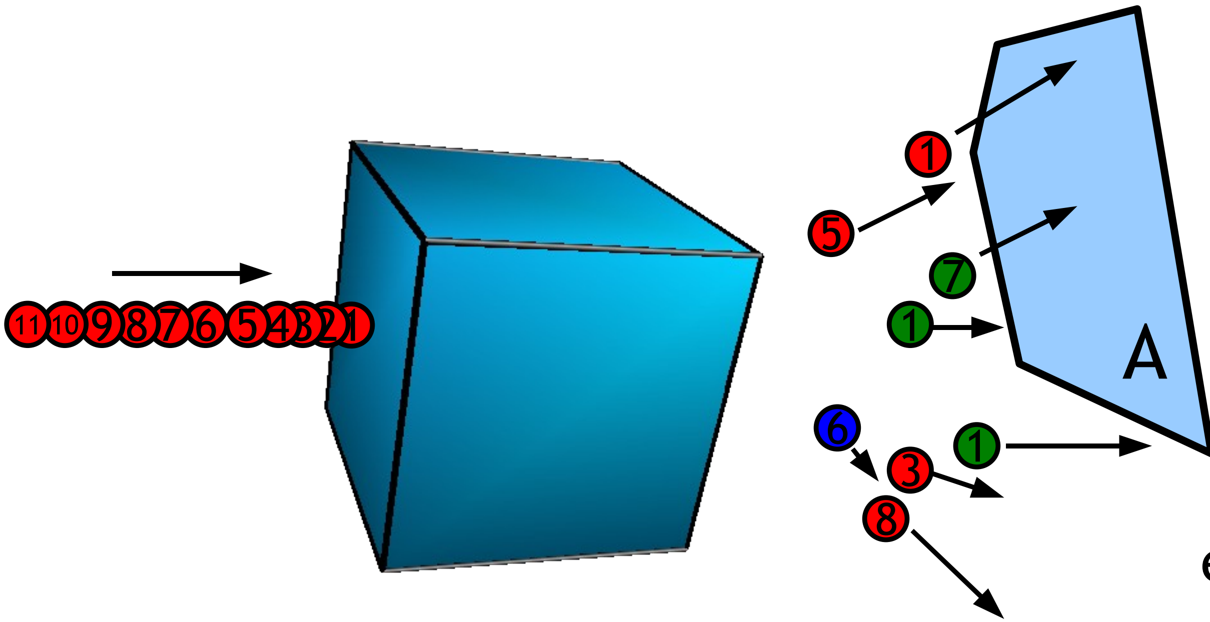
- Follow the (interaction) history of one individual incident particle
  - by calculating/sampling all its interactions according to the (quantum-mechanical) probability distributions
  - and follow the (interaction) history of all final state-particles in the same manner (i.e. recursively)
- Repeat the simulation many times under the same initial conditions
- Count the resulting particles and their states in the spatial regions of interest
  - **that will give us a statistical estimate of the distributions or any observables derived therefrom**

<u>Example</u>:

11 individual Monte Carlo experiments (11 events 1, 2, .., 11)

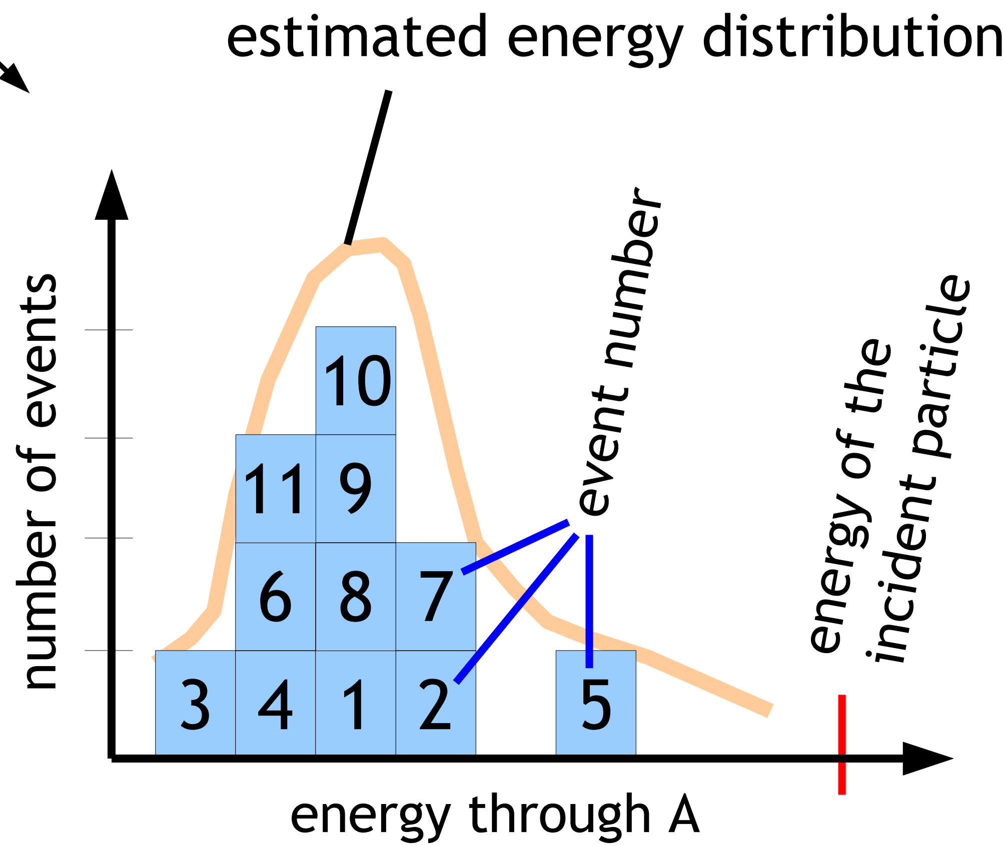
particles of event 1
not all particles shown ...



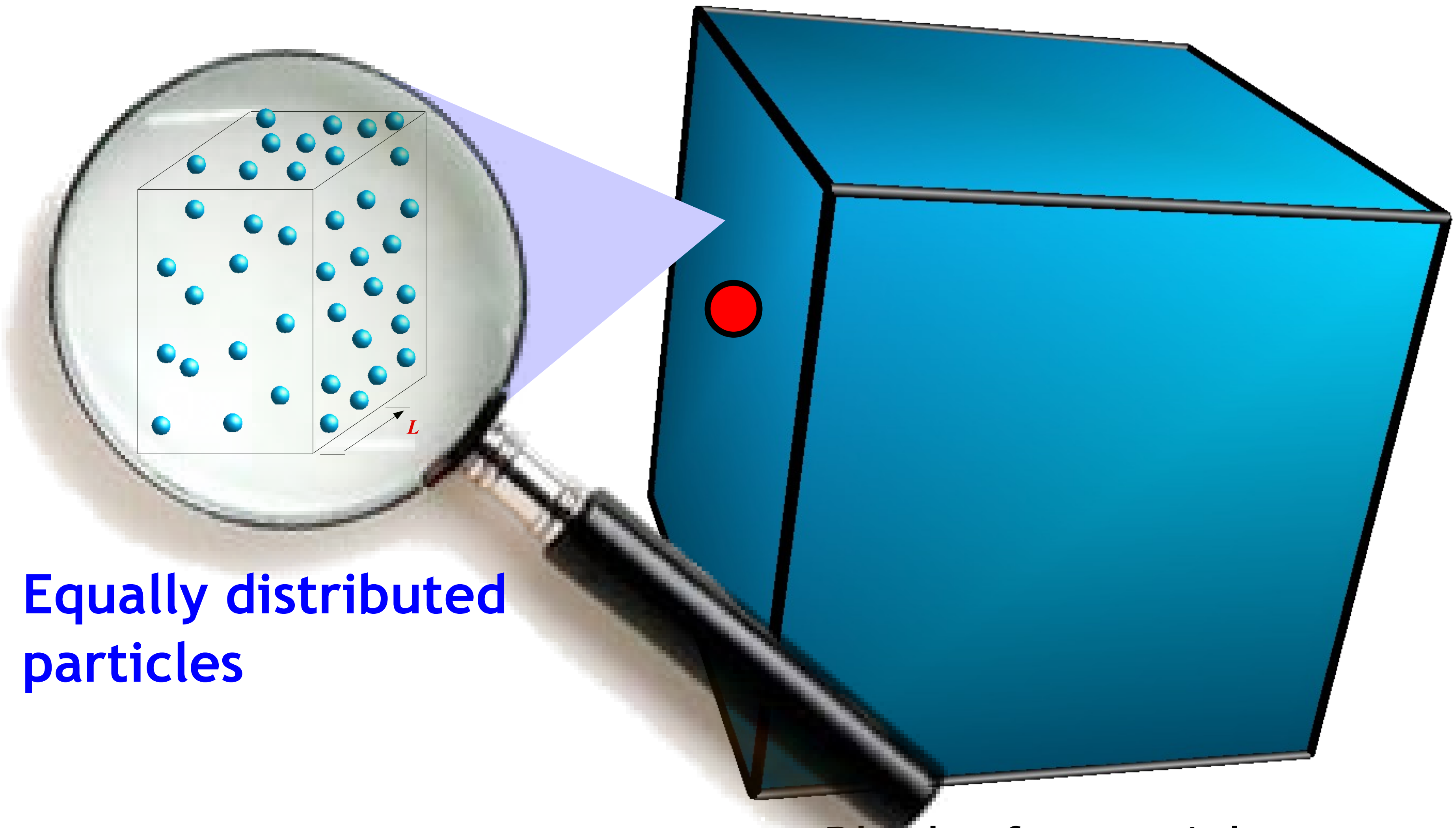


Assume, we are interested in the energy flowing through A

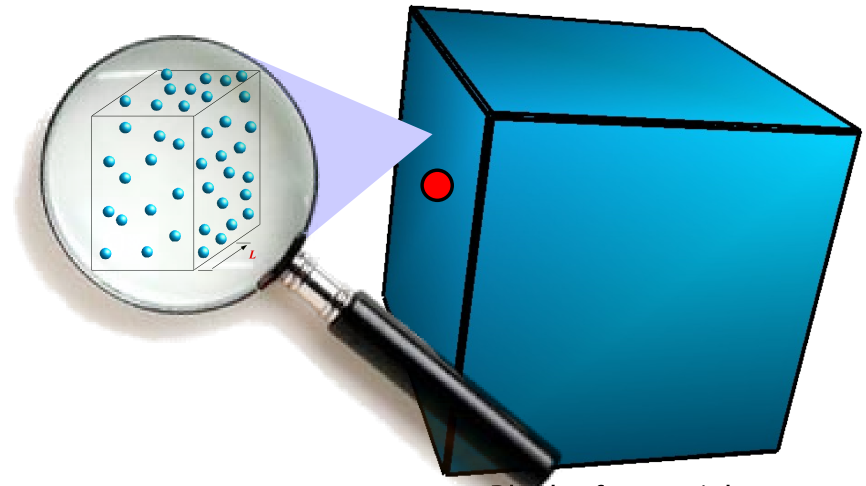
For each event 1,2,... we simply count the particles going through A, sum up there energies and histogram them:



An approach to an answer ..



What we know ..



Incident particle - microscopic properties:

type (e^+, μ^-,..), position, energy, momentum, spin, ..

Block of material - microscopic properties:

constituting element types & their properties (atomic charge, ..), mixture ratios (H_2O), molar masses

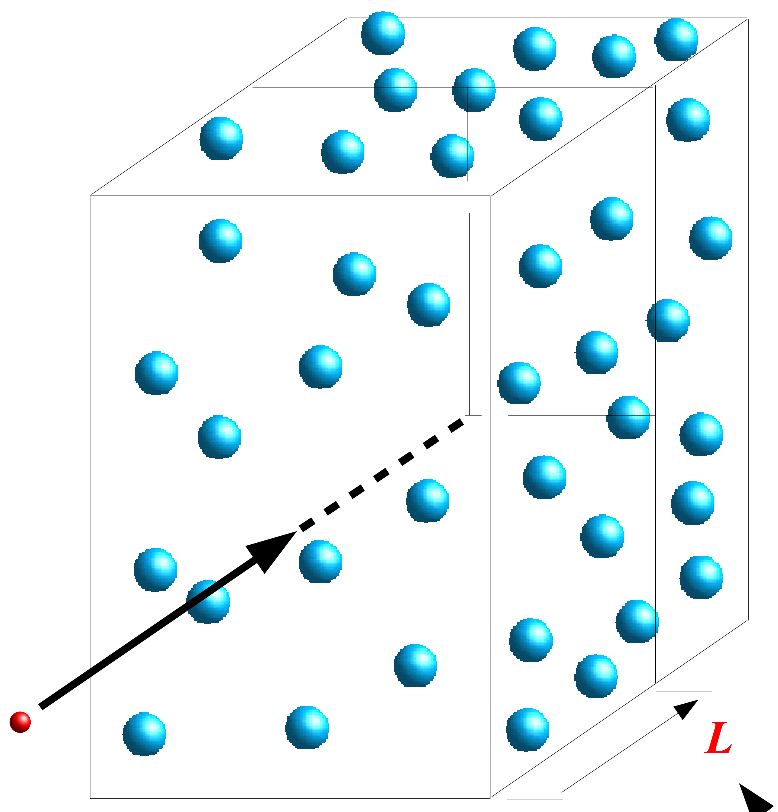
↕

macroscopic properties:

density (g/cm^3), size/shape, state (solid, gaseous, ..), ..

Physics of microscopic particle-particle interation

From microscopic to macroscopic distances



macroscopic distance L

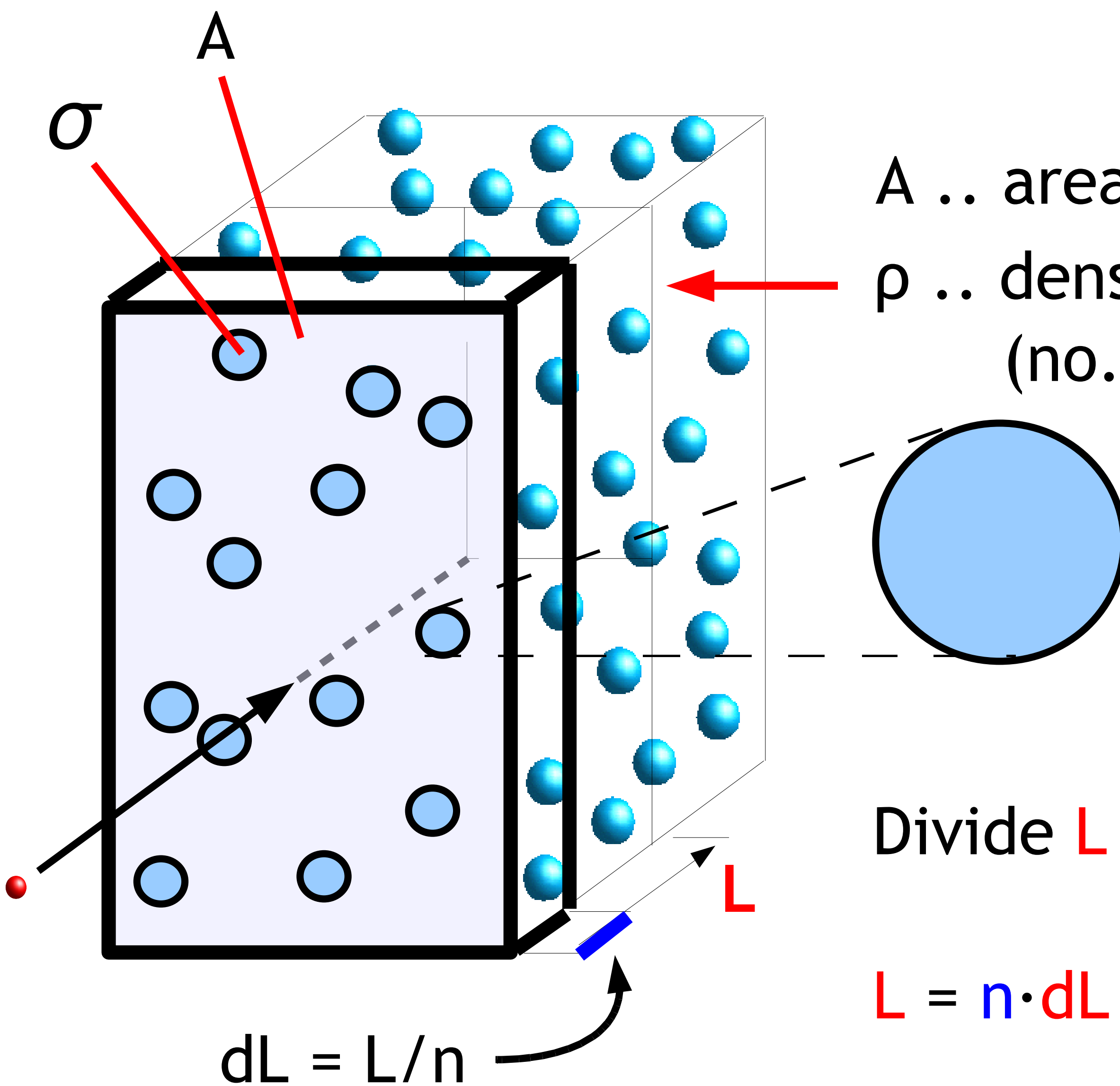
We assume that the particles making up the bulk material are **equally distributed at random!**

How far can an incident particle travel without interacting with the bulk material?

What is the **probability** for an incident particle to have its first interaction at distance **L**?

L is a **continuous random variable.** So we will have to look for a probability density **w(L)**:

w(L)dL … probability of the first interaction in (L, L + dL)



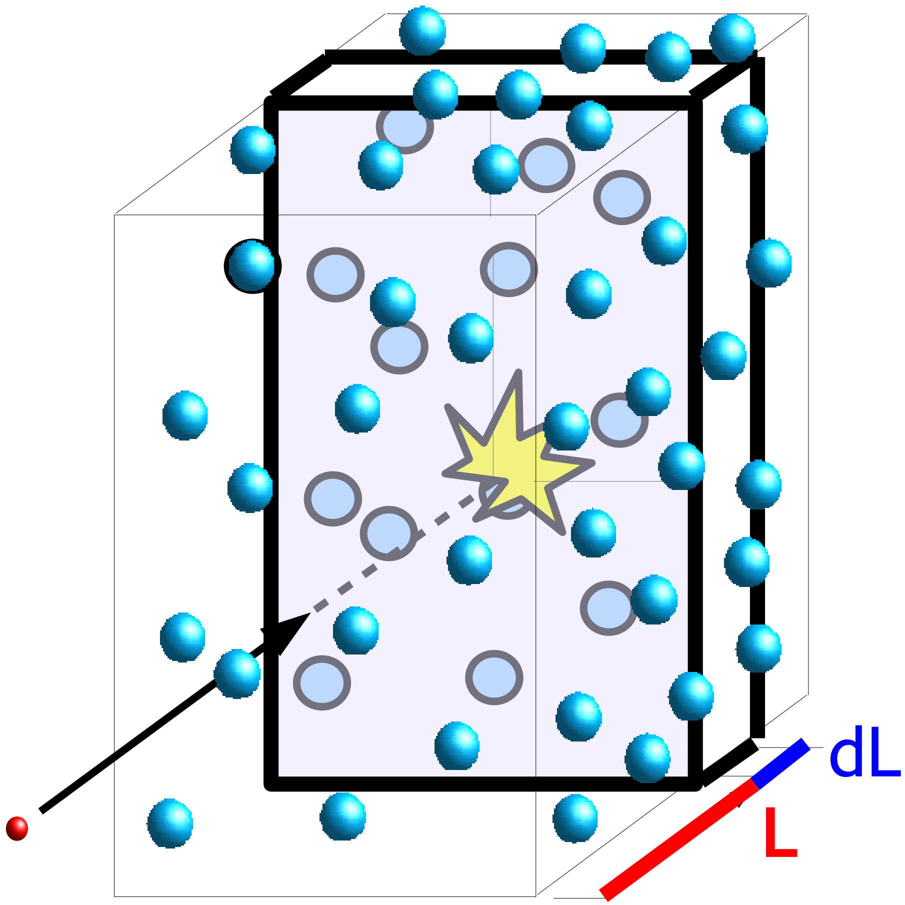
A .. area of incidence

ρ .. density of obstacles
(no. of targets per volume)

σ ... cross-section

Divide L into n thin slices dL:

$L = n \cdot dL$



Divide L into n thin slices dL

$L = n \cdot dL$

Probability to have **the first** interaction in (L, L + dL)

=

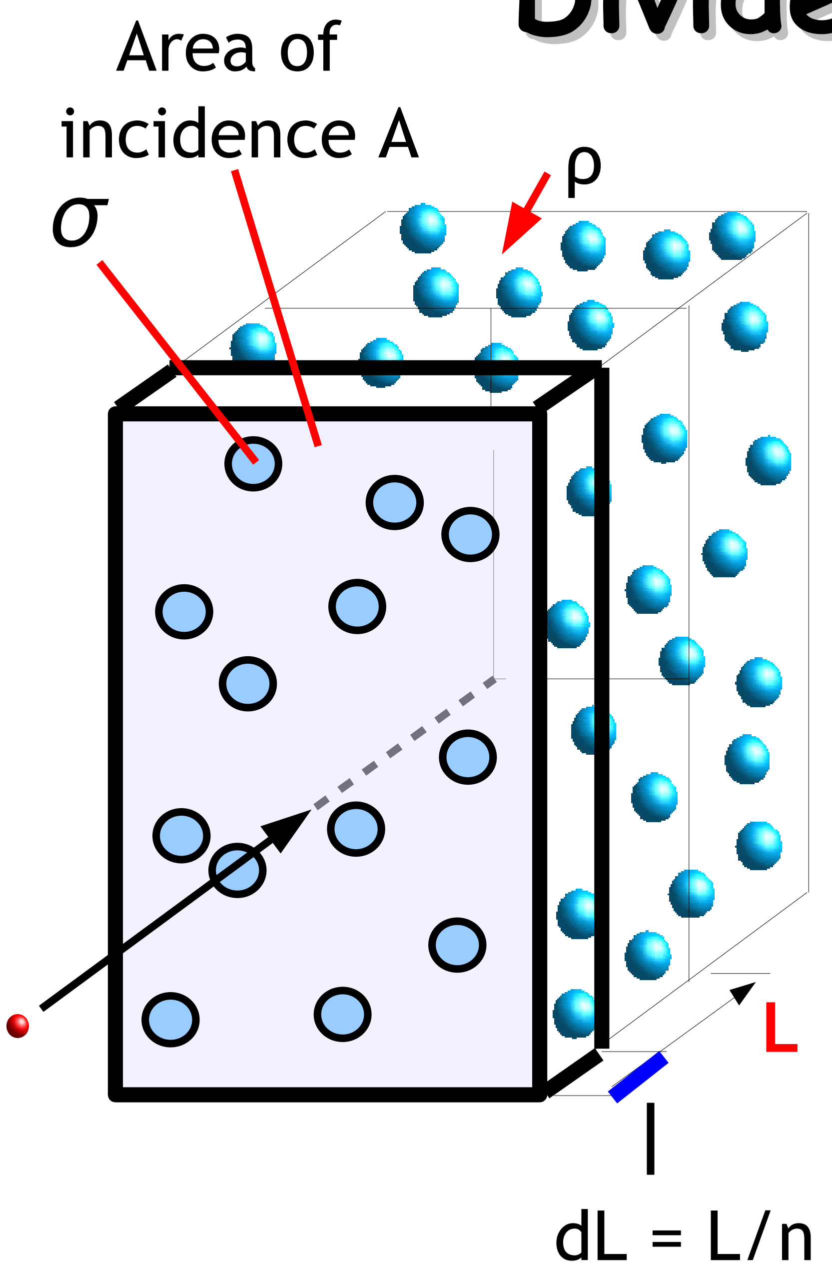
probability to travel undisturbed until L: **P(L)**

times

probability to have an interaction in slice (L, L + dL): **dw**

$$w(L)dL = P(L) \cdot dw$$

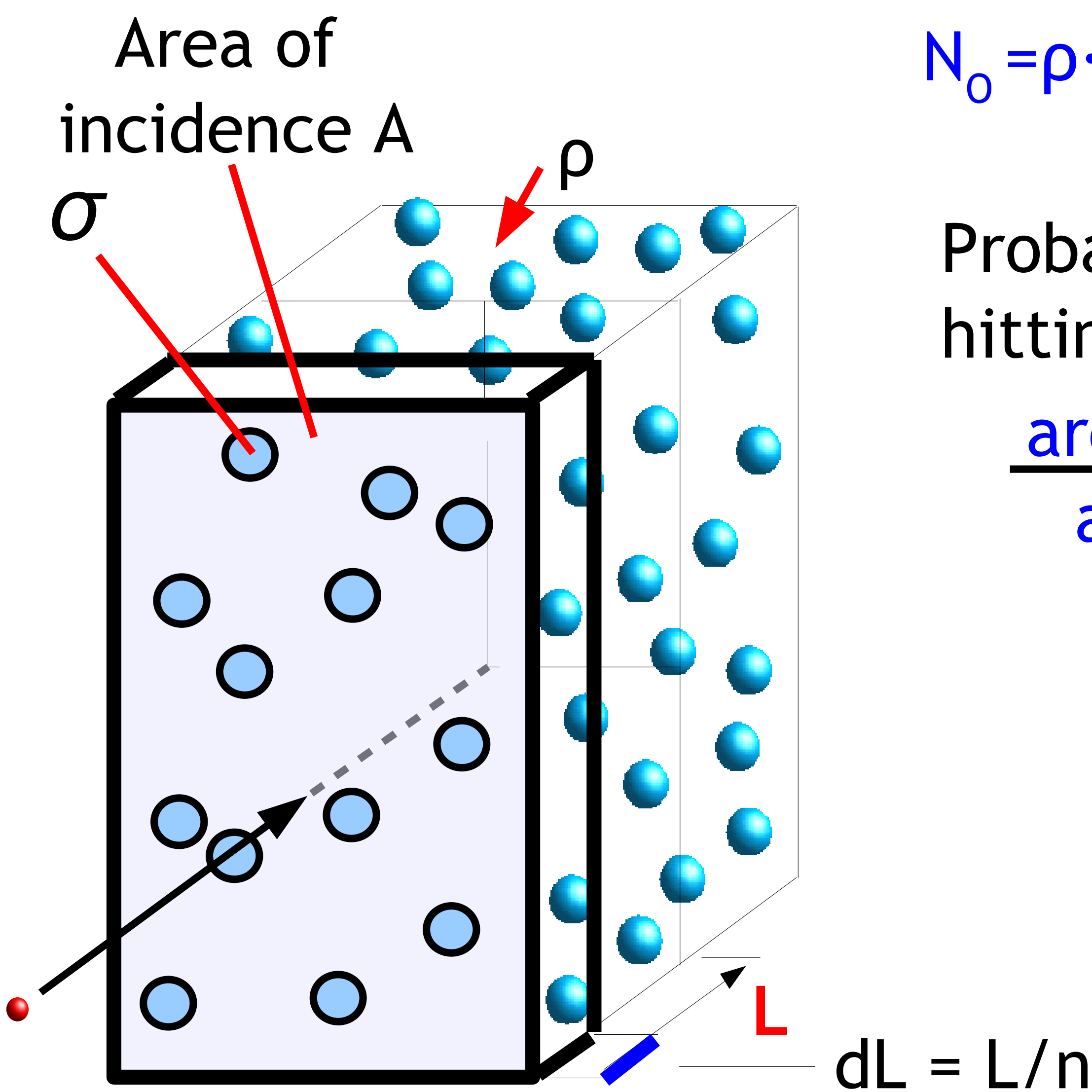
"Divide et Impera!"



Divide the block into n slices of width $dL = L/n$

Each block then contains
$N_o = \rho \cdot A \cdot dx = \rho \cdot A \cdot L/n$ targets

$$w(L)dL = P(L) \cdot dw$$



Each block contains

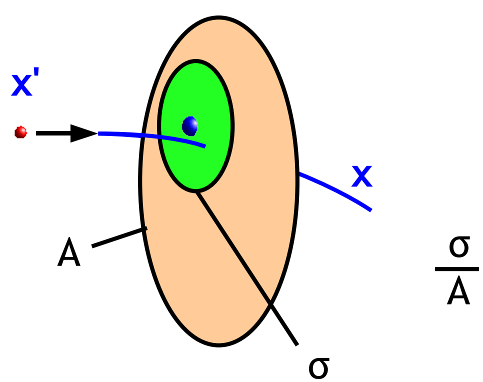
$N_o = \rho \cdot A \cdot dL = \rho \cdot A \cdot L/n$ targets

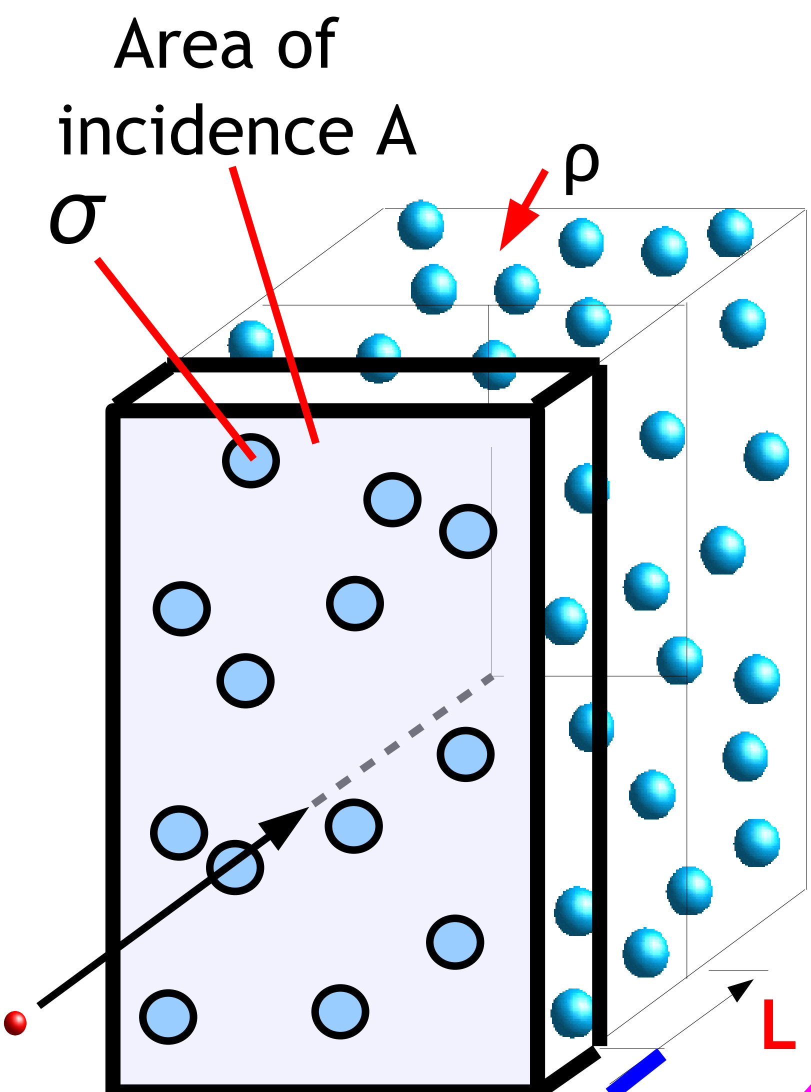
Probability to cross the slice hitting a target =

$$\frac{\text{area of targets in slice}}{\text{area of incidence A}} =$$

$$\frac{N_o \cdot \sigma}{A} = \frac{\rho \cdot L \cdot \sigma}{n}$$

Remember?





Probability to cross the slice hitting a target particle =

$$\frac{\text{area of targets in slice}}{\text{area of incidence A}} =$$

$$P_{hit} = \frac{N_0 \cdot \sigma}{A} = \frac{\rho \cdot L \cdot \sigma}{n} = \rho \cdot \sigma \cdot dL$$

$$= \mathbf{dw}$$

Note: **dw** is the same for every slice!

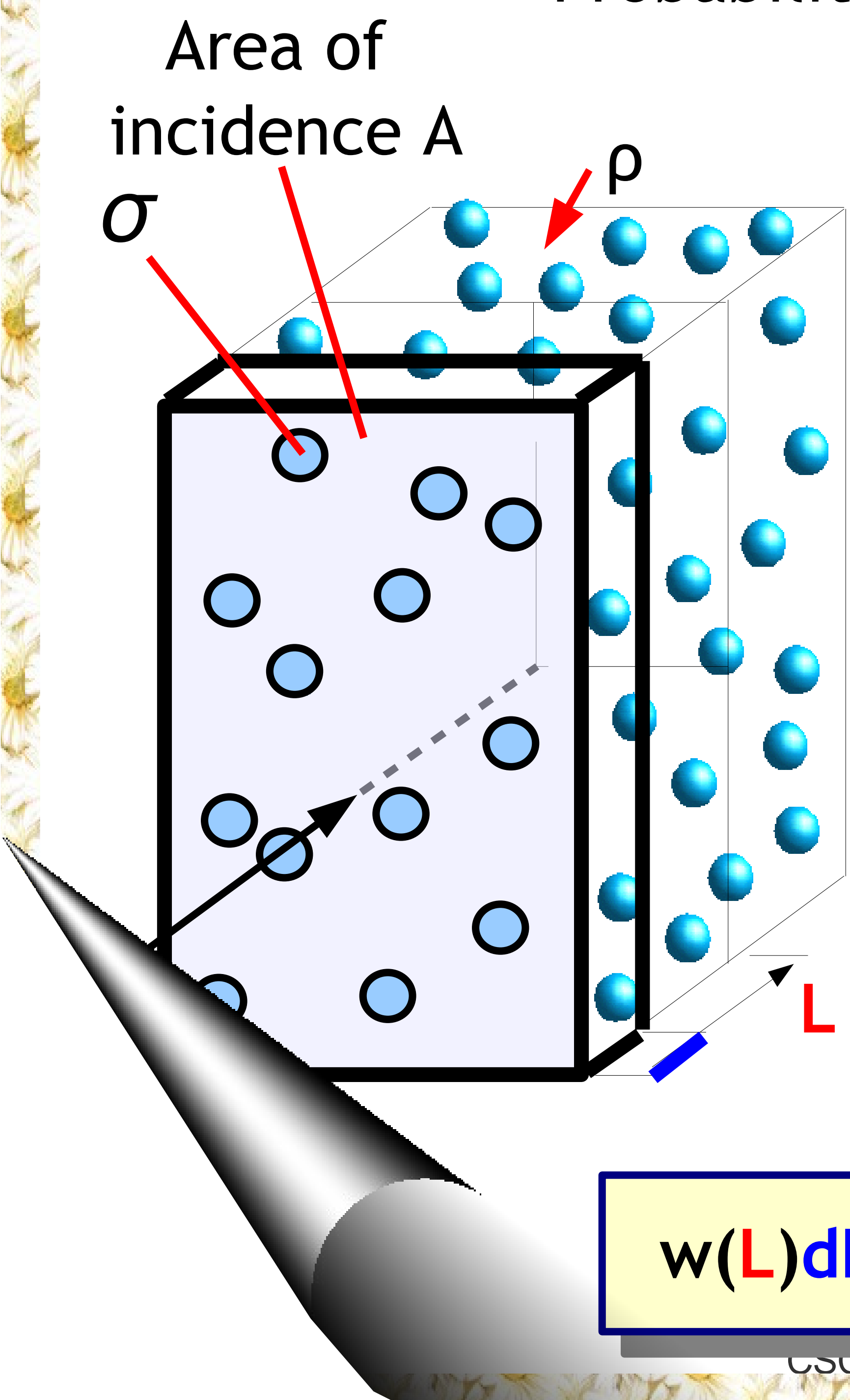
Probability to cross the slice **undisturbed**:

$$P_{no\text{-}hit} = 1 - P_{hit}$$

$$= 1 - \frac{\rho \cdot L \cdot \sigma}{n}$$

$$w(L)dL = P(L) \cdot dw$$

Probability to cross the slice dL **undisturbed**:



$$P_{no\text{-}hit} = 1 - \frac{\rho \cdot L \cdot \sigma}{n}$$

Recall, we have n slices: $L = n \cdot dx$

Multiply $P_{no\text{-}hit}$ n times:

$$P(L) = P_{no\text{-}hit} \cdot P_{no\text{-}hit} \cdot \ldots \cdot P_{no\text{-}hit}$$

$$= (P_{no\text{-}hit})^n$$

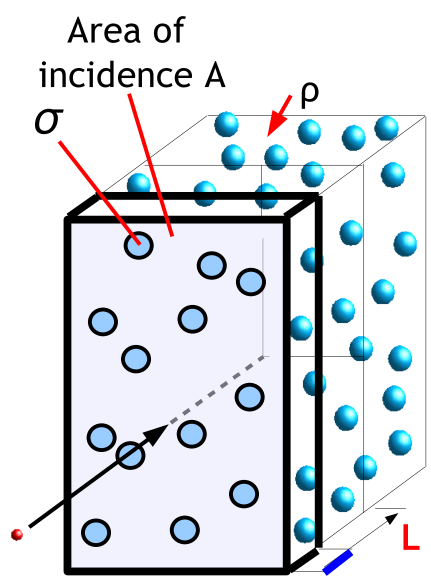
$$= \left(1 - \frac{\rho \cdot L \cdot \sigma}{n}\right)^n$$

$P(L) \sim \exp(-\rho \cdot L \cdot \sigma)$

$w(L)dL = P(L) \cdot dw$

$$w(L)dL = P(L)\cdot dw$$

Probabilities



$$dw = \rho\cdot\sigma\cdot dL$$

$$P(L) = \exp(-\rho\cdot L\cdot\sigma)$$

$$w(L)dL = \rho\cdot\sigma\cdot\exp(-\rho\cdot L\cdot\sigma)\cdot dL$$

Note: The probabilistic character of this result is due ONLY to the fact of our lack of knowledge concerning the exact distribution of interaction centres in the bulk material!

Reminder: probabilities

x ... continuous random variable

$\underline{x}$ ... realisation of x (outcome of an experiment)

w(x) ... probability density of x

w(x)dx .. probability, that $\underline{x}$ is in the interval (x, x + dx)

$F(y) = \int_{-\infty}^{y} w(x)dx$ .. probability, that $\underline{x} \leq y$

(cumulative distribution)

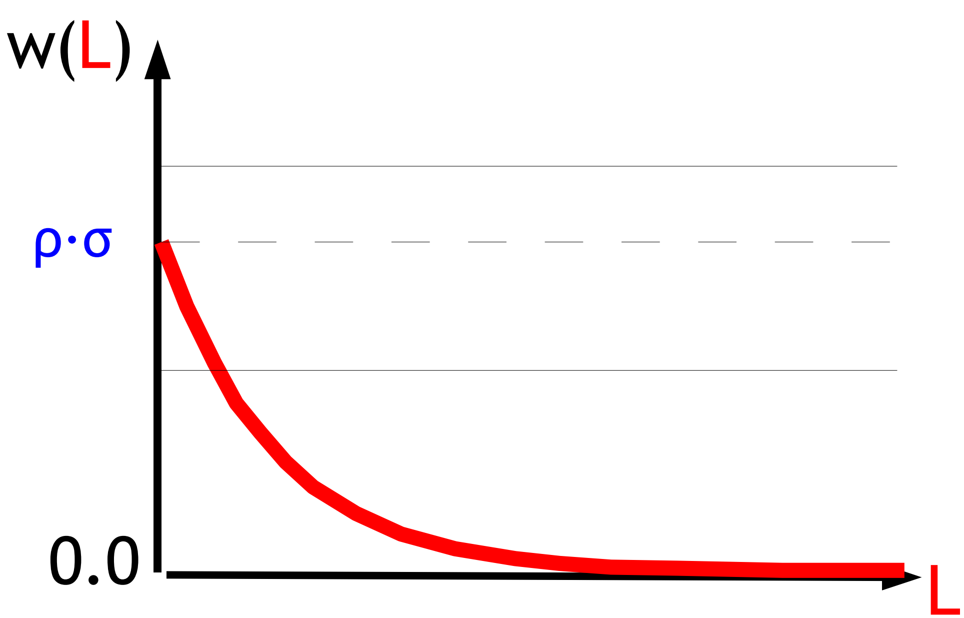
Sampling = carrying out repeatable experiments (real life)

= generating lots of $\underline{x}$'s ($\underline{x}_1$, $\underline{x}_2$, ..) (Monte Carlo)

Probability density:

Probability to have the first interaction ***at*** distance L is w(L)dL with

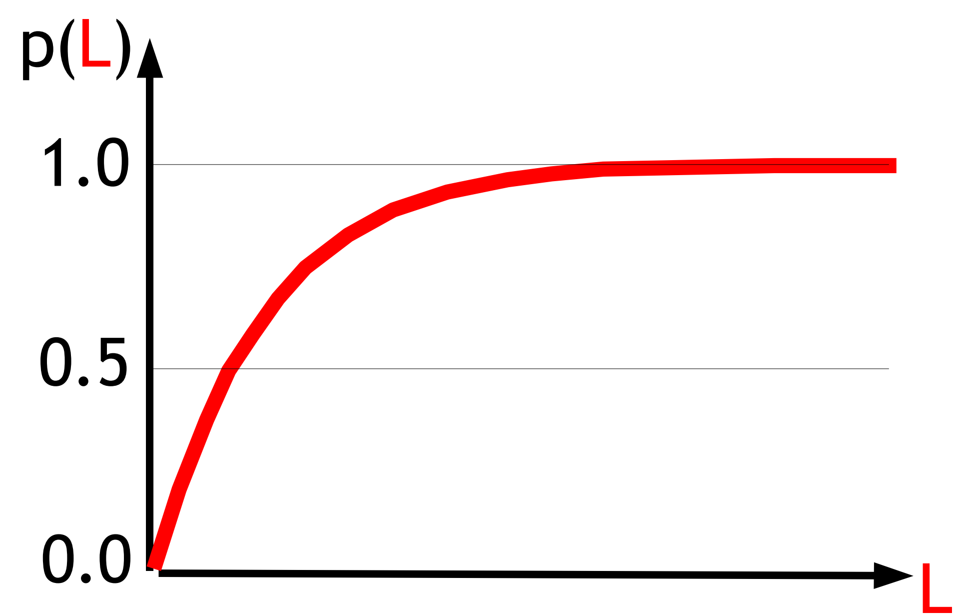
$$w(L) = \rho\cdot\sigma\cdot\exp(-\rho\cdot L\cdot\sigma)$$



Cumulative distribution:

Probability to have an interaction ***within*** distance L:

$$p(L) = \int_0^L w(x)dx = 1 - \exp(-\rho\cdot L\cdot\sigma)$$

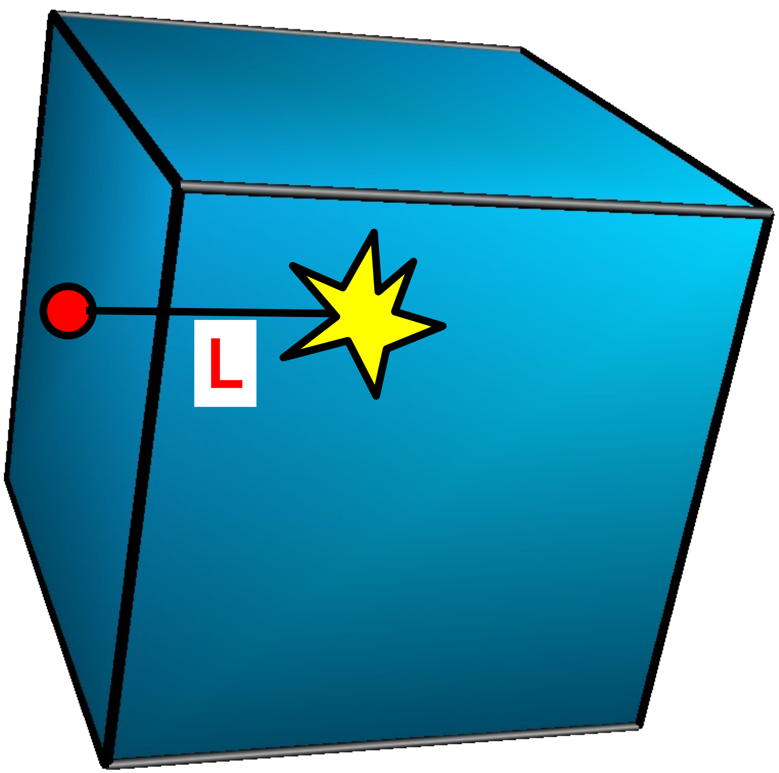


A “probable” answer!

The probability of having an interaction within distance L:

$$p(L) = 1 - \exp(-\rho \cdot L \cdot \sigma)$$

(the exponential distribution!)



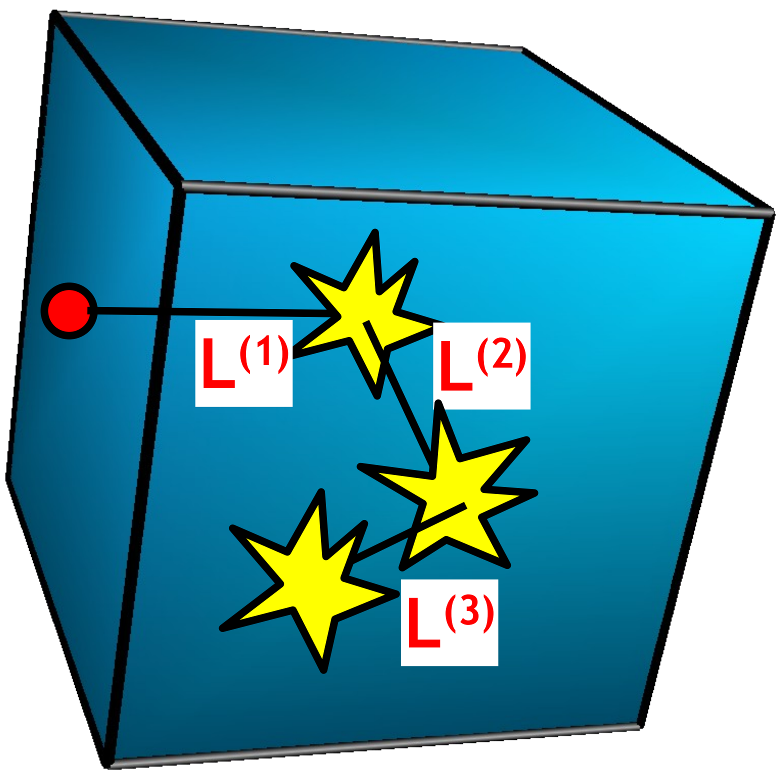
Block of material

Basic algorithm

The probability of having an interaction within distance L:

$$p(L) = 1 - \exp(-\rho \cdot L \cdot \sigma)$$

(the exponential distribution!)



Very basic algorithm:

(1) initial values for incident particle
(2) get values for ρ and σ
(3) sample $L^{(n)}$ from p(L)
(4) transport particle undisturbed by $L^{(n)}$
(5) simulate interaction
(6) if particle still exists: goto (1)

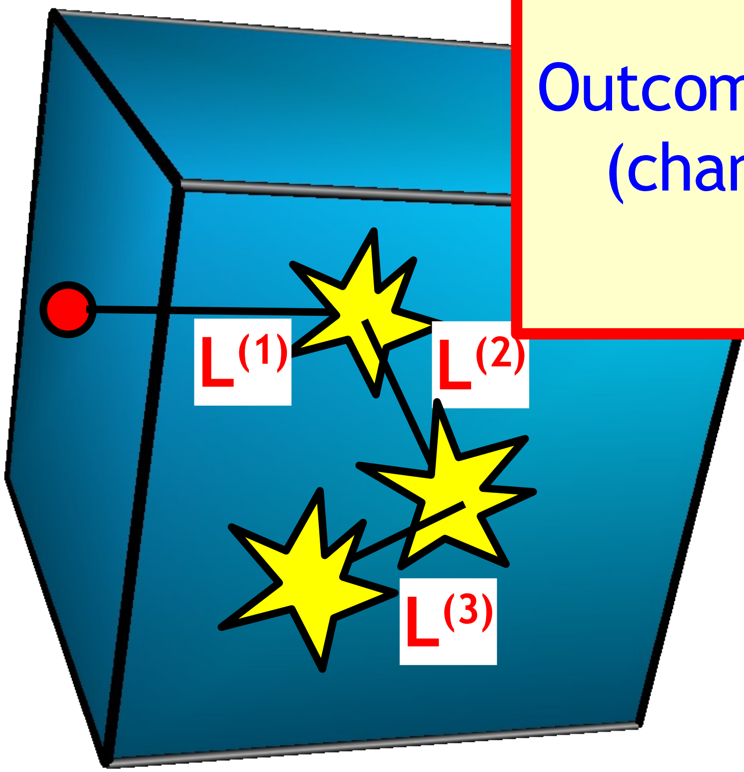
Core of the Monte Carlo Method: drawing of random numbers!

B

The probability c

interaction described by quantum mechanics - cross-section!!,
→distributions, random numbers,
Monte Carlo method again!!

Outcome: new state of the incident particle (change of energy, momentum, ...) AND other particles!!!



(2) get values for ρ and σ
(3) sample $L^{(n)}$ from p(L)
(4) transport particle undisturbed by $L^{(n)}$
(5) simulate interaction
(6) if particle still exists: goto (1)

Core of the Monte Carlo Method: drawing of random numbers!

Mean Free Path Length

$$p(L) = 1 - \exp(-\rho \cdot L \cdot \sigma)$$
$$= 1 - \exp(-L / \lambda)$$

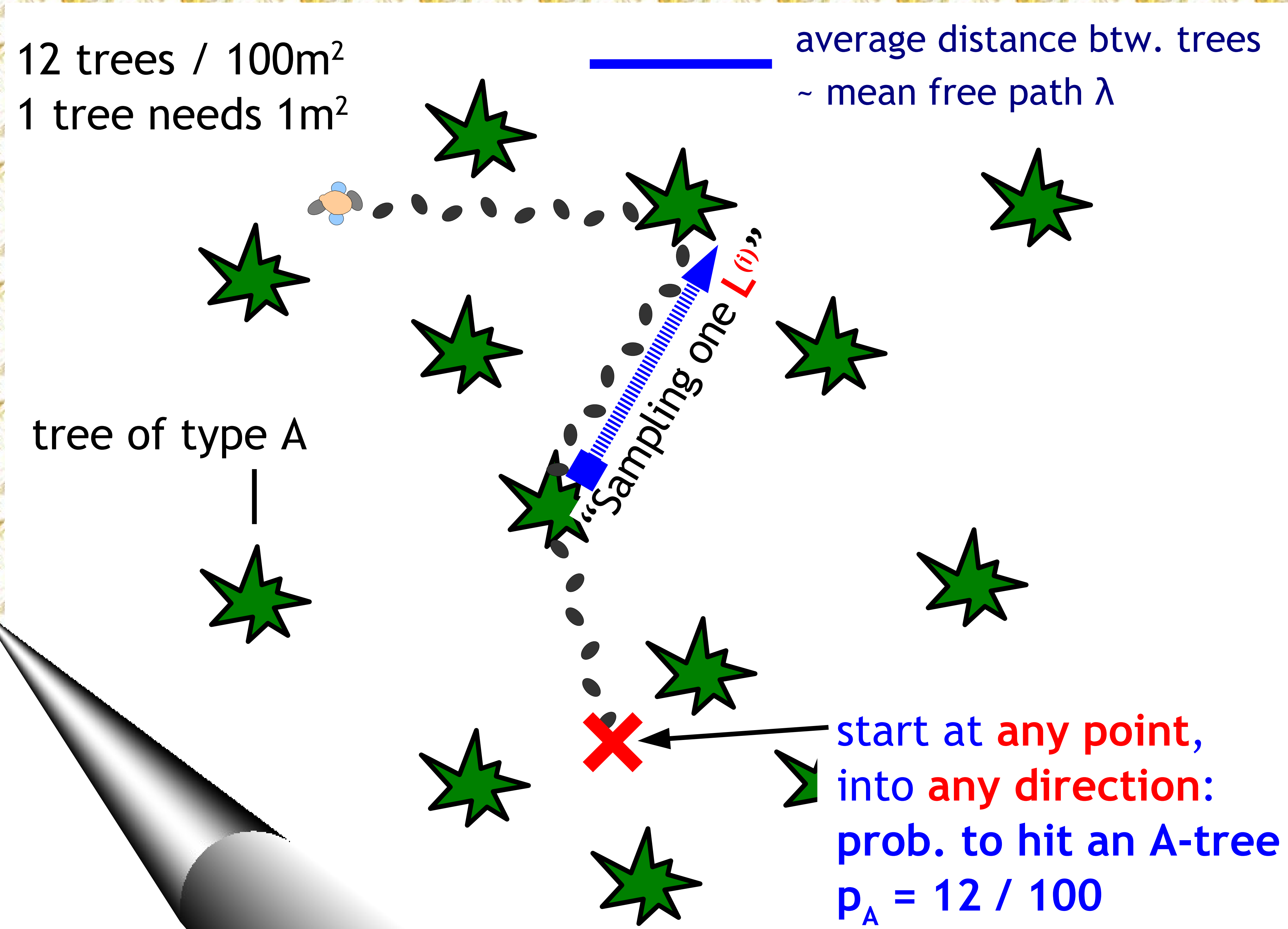
$\lambda := 1/(\rho \cdot \sigma)$ ... mean free path length,
average distance a particle moves undisturbed
with respect to process σ in material ρ

$\lambda = \lambda$(interaction type – particles, interaction energy,
density of targets, ..)

Microscopic and macroscopic properties united!

The Finnish For(r)est!

- Ingredients
  - the Finnish Forest
  - Forrest Gump
  - a Finnish night in winter or Forrest with closed eyes
- Forrest runs through the forest ..


12 trees / 100m²
1 tree needs 1m²
average distance btw. trees
~ mean free path λ
tree of type A
"Sampling one L(i)"
start at any point,
into any direction:
prob. to hit an A-tree
$p_A = 12 / 100$

Mean Free Path Length,

more than one interaction process

$$p_i(L) = 1 - \exp(-\rho_i \cdot L \cdot \sigma_i)$$

$$= 1 - \exp(-L / \lambda_i)$$

$\lambda_i := 1/(\rho_i \cdot \sigma_i)$ ...mean free path length,

average distance a particle moves undisturbed
with respect to process i

Process 1 has λ_1:

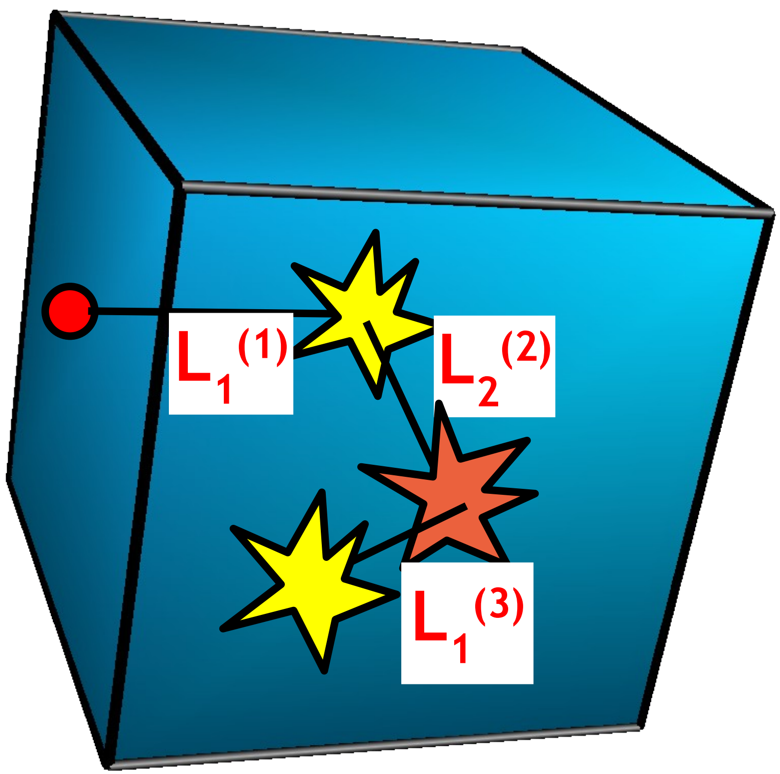
Process 2 has λ_2:

...

Process m has λ_m:

$\lambda_i = \lambda_i$(interaction type – particles, interaction energy, density of targets, ..)

Back to the "probable" Answer ..



Usually, there are more (i=1,2,..,m) processes responsible for interactions. For each of them we have the probability:

$$p_i(L) = 1 - \exp(-\rho_i \cdot L \cdot \sigma_i)$$

ρ_i .. density of interaction centers for physics process i

σ_i .. cross section of physics process i

Monte Carlo Algorithm

The free path lengths for an incident particle for each process are distributed according to the exponential distribution determined by the mean free path length of this process and material:

$$p_i(x) = 1 - \exp(-x / \lambda_i)$$

(1) set properties for incident particle (momentum, ..)

(2) get values for λ_i for all relevant processes i=1,2,..,m

(3) for each process i (i=1,2,..,m):

sample L_i from $p_i(x)$

(4) L_c = min(L_i) from all sampled L_i , c in (1,2,..,m)

(5) transport incident particle by L_c

(6) simulate interaction

(7) if particle still exists: goto (1)

A little Example

(1) set properties for incident particle (momentum, ..)

(2) get values for λ_i for all relevant processes i=1,2,..,m

(3) for each process i (i=1,2,..,m):

sample L_i from $p_i(x)$

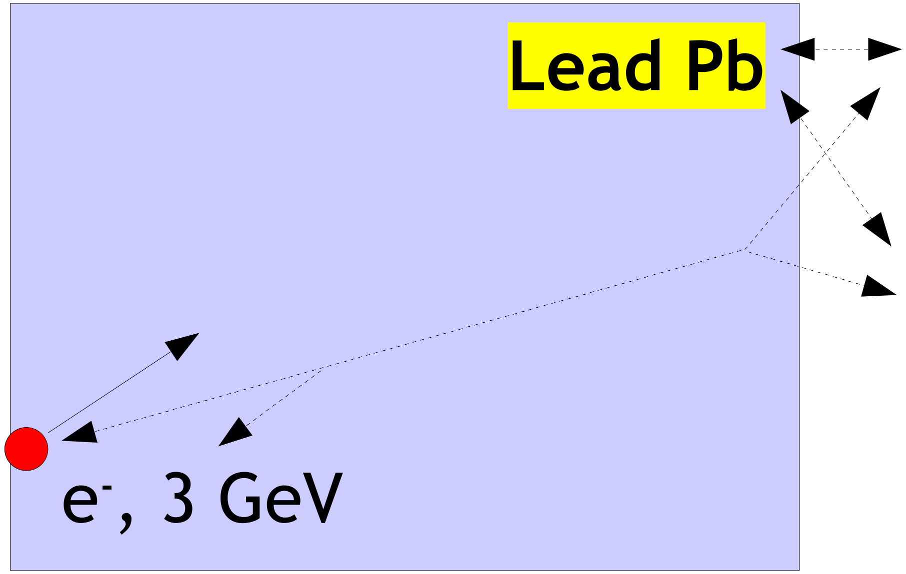
(4) L_c = min(L_i) from all sampled L_i , c in (1,2,..,m)

(5) transport incident particle by L_c

(6) simulate interaction

(7) if particle still exists: goto (1)

Let's take only 2 processes:



Proc.1 "e- Bremsstrahlung": λ_1 =

Proc.2 "e- Ionization": λ_2 =

(1) set properties for incident particle (momentum, ..)

(2) get values for λ_i for all relevant processes i=1,2,..,m

(3) for each process i (i=1,2,..,m):

sample L_i from $p_i(x)$

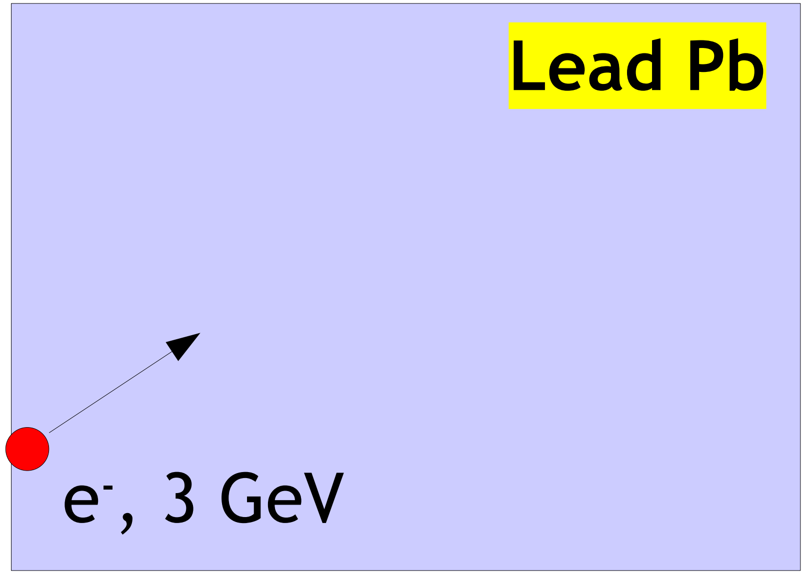
(4) L_c = min(L_i) from all sampled L_i , c in (1,2,..,m)

(5) transport incident particle by L_c

(6) simulate interaction

(7) if particle still exists: goto (1)

Roll the Dice!



Proc.1 "e⁻ Bremsstrahlung":
λ_1 =

Proc.2 "e⁻ Ionization":
λ_2 =

$$p_i(L) = 1 - \exp(-L/\lambda_i)$$

$L_1^{(1)}$ =

$L_2^{(1)}$ =

(1) set properties for incident particle (momentum, ..)

(2) get values for λ_i for all relevant processes i=1,2,..,m

(3) for each process i (i=1,2,..,m):
sample L_i from $p_i(x)$

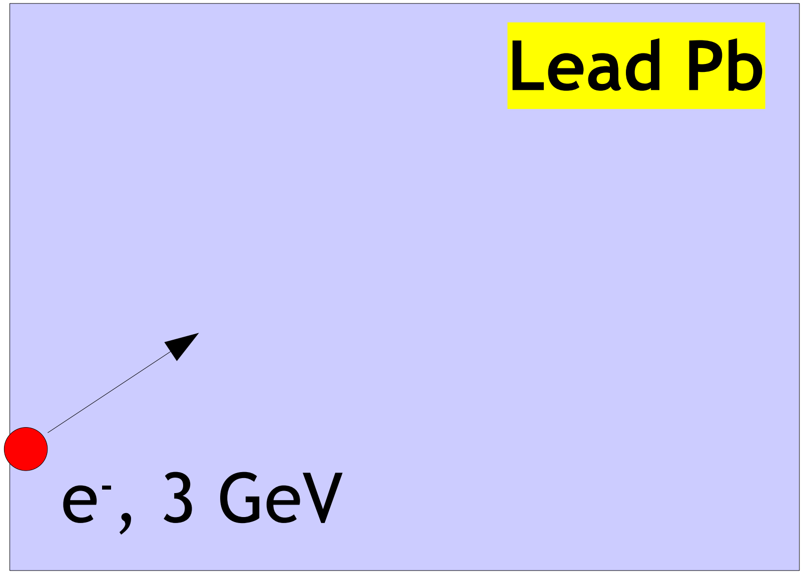
(4) L_c = min(L_i) from all sampled L_i , c in (1,2,..,m)

(5) transport incident particle by L_c

(6) simulate interaction

(7) if particle still exists: goto (1)

First come, first served ..



Proc.1 "e⁻ Bremsstrahlung":
λ_1 =

Proc.2 "e⁻ Ionization":
λ_2 =

$$p_i(L) = 1 - \exp(-L / \lambda_i)$$

$L_1^{(1)}$ =

$L_2^{(1)}$ =

(1) set properties for incident particle (momentum, ..)

(2) get values for λ_i for all relevant processes i=1,2,..,m

(3) for each process i (i=1,2,..,m):

sample L_i from $p_i(x)$

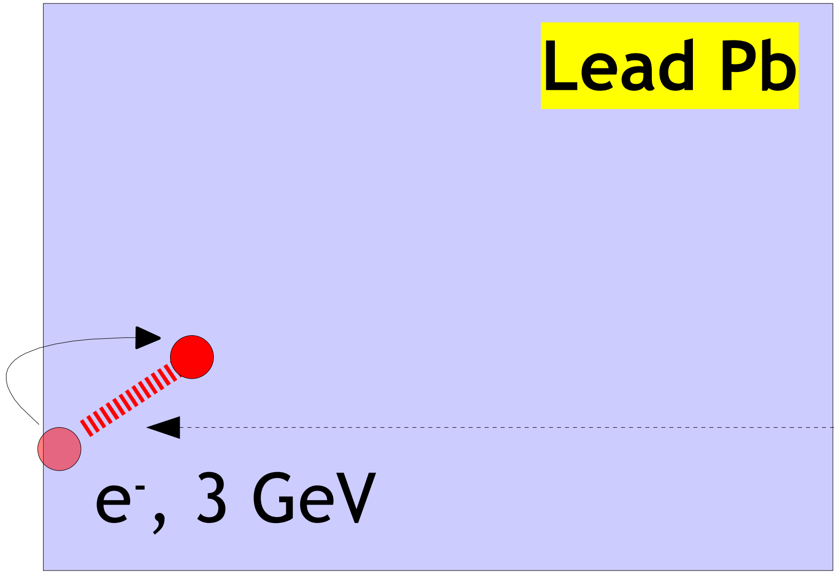
(4) L_c = min(L_i) from all sampled L_i , c in (1,2,..,m)

(5) transport incident particle by L_c

(6) simulate interaction

(7) if particle still exists: goto (1)

Move it!



Proc.1 “e⁻ Bremsstrahlung”:
λ_1 =

Proc.2 “e⁻ Ionization”:
λ_2 =

$$p_i(L) = 1 - \exp(-L / \lambda_i)$$

$L_1^{(1)}$ =

$L_2^{(1)}$ =

(1) set properties for incident particle (momentum, ..)

(2) get values for λ_i for all relevant processes i=1,2,..,m

(3) for each process i (i=1,2,..,m):

sample L_i from $p_i(x)$

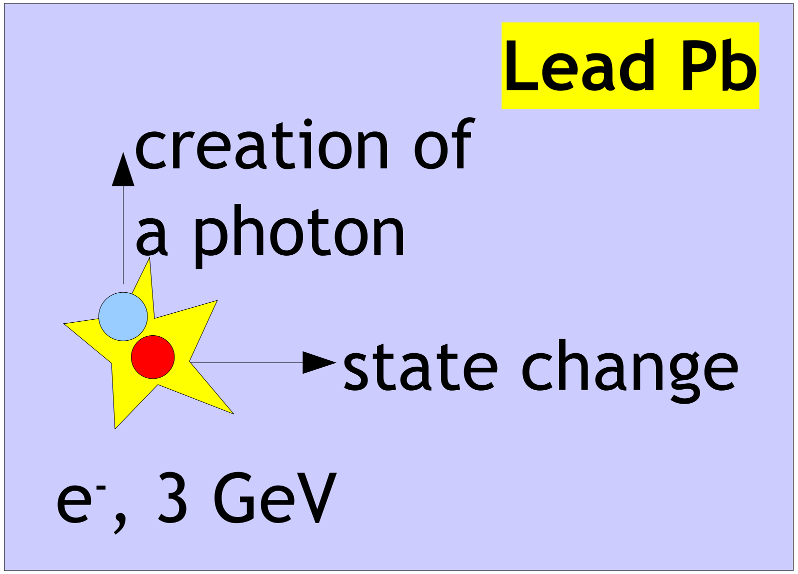
(4) L_c = min(L_i) from all sampled L_i , c in (1,2,..,m)

(5) transport incident particle by L_c

(6) simulate interaction

(7) if particle still exists: goto (1)

".. and (Inter)Action!"



Proc.1 "e- Bremsstrahlung":
λ_1 =

Proc.2 "e- Ionization":
λ_2 =

$$p_i(L) = 1 - \exp(-L / \lambda_i)$$

$L_1^{(1)}$ =

$L_2^{(1)}$ =

(1) set properties for incident particle (momentum, ..)

(2) get values for λ_i for all relevant processes i=1,2,..,m

(3) for each process i (i=1,2,..,m):

sample L_i from $p_i(x)$

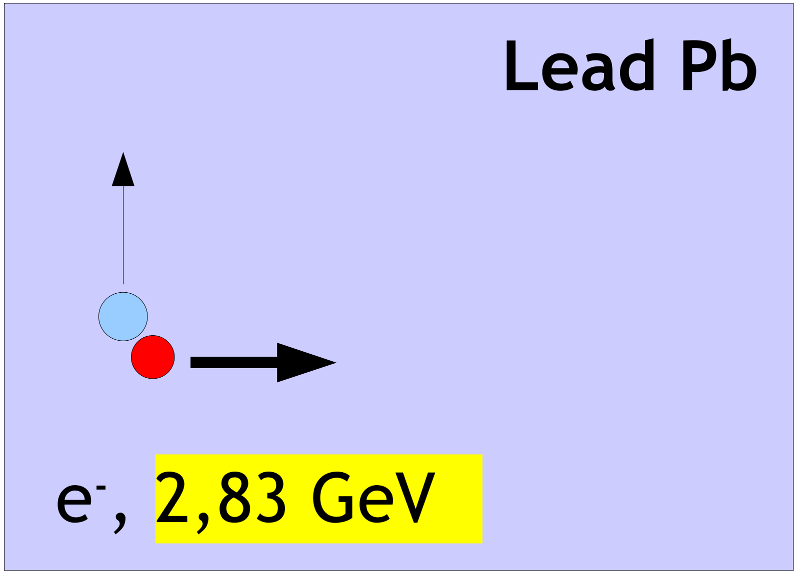
(4) L_c = min(L_i) from all sampled L_i , c in (1,2,..,m)

(5) transport incident particle by L_c

(6) simulate interaction

(7) if particle still exists: goto (1)

Da Capo!



Proc.1 "e^- Bremsstrahlung":
λ_1 =

Proc.2 "e^- Ionization":
λ_2 =

(1) set properties for incident particle (momentum, ..)

(2) get values for λ_i for all relevant processes i=1,2,..,m

(3) for each process i (i=1,2,..,m):

sample L_i from $p_i(x)$

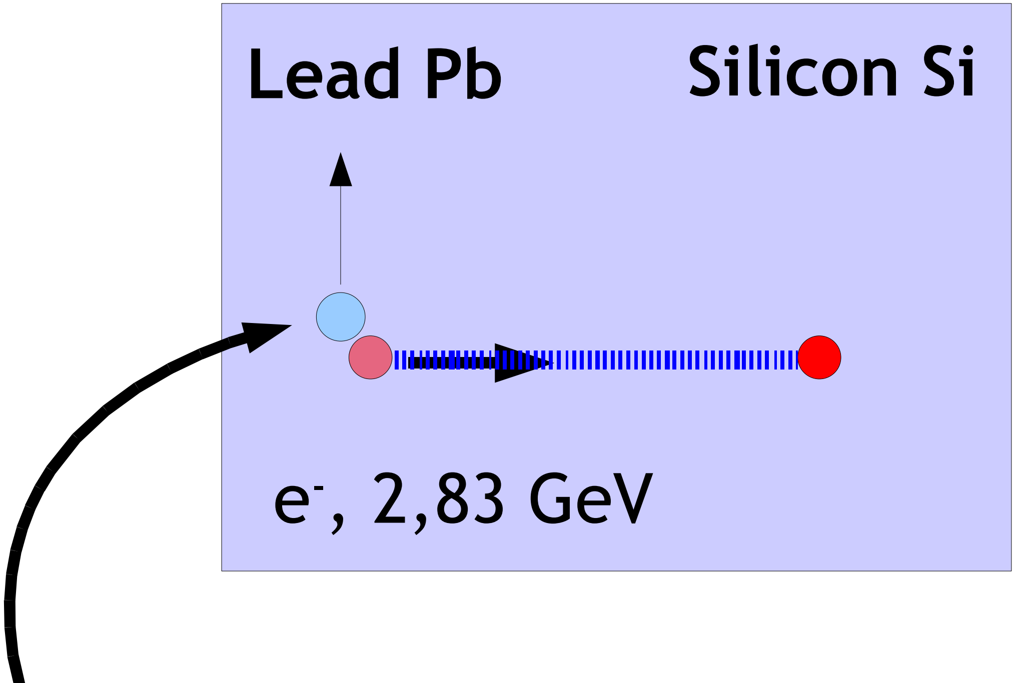
(4) L_c = min(L_i) from all sampled L_i , c in (1,2,..,m)

(5) transport incident particle by L_c

(6) simulate interaction

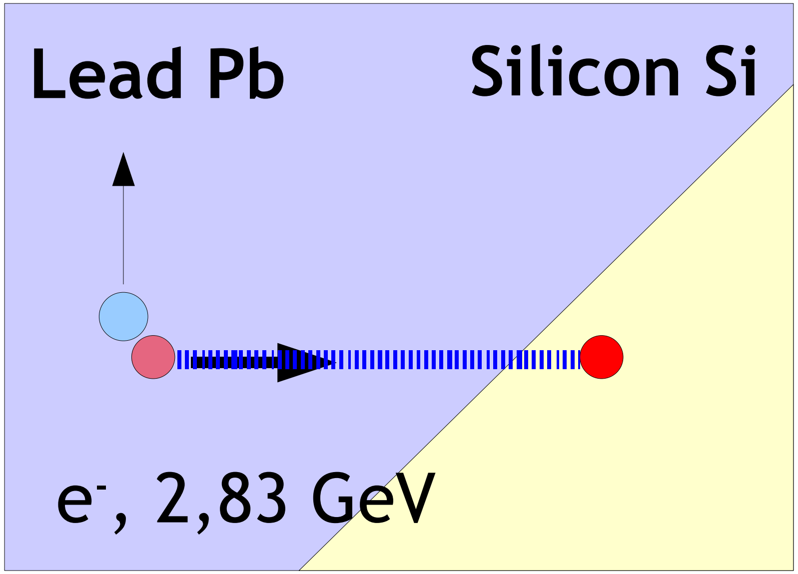
(7) if particle still exists: goto (1)

Secondaries



keep track of newly created particles ("secondaries")
- they will be simulated once the "primary" is done

Changing the material



re-scale the free path length once a particle enters another material during an undisturbed push forward

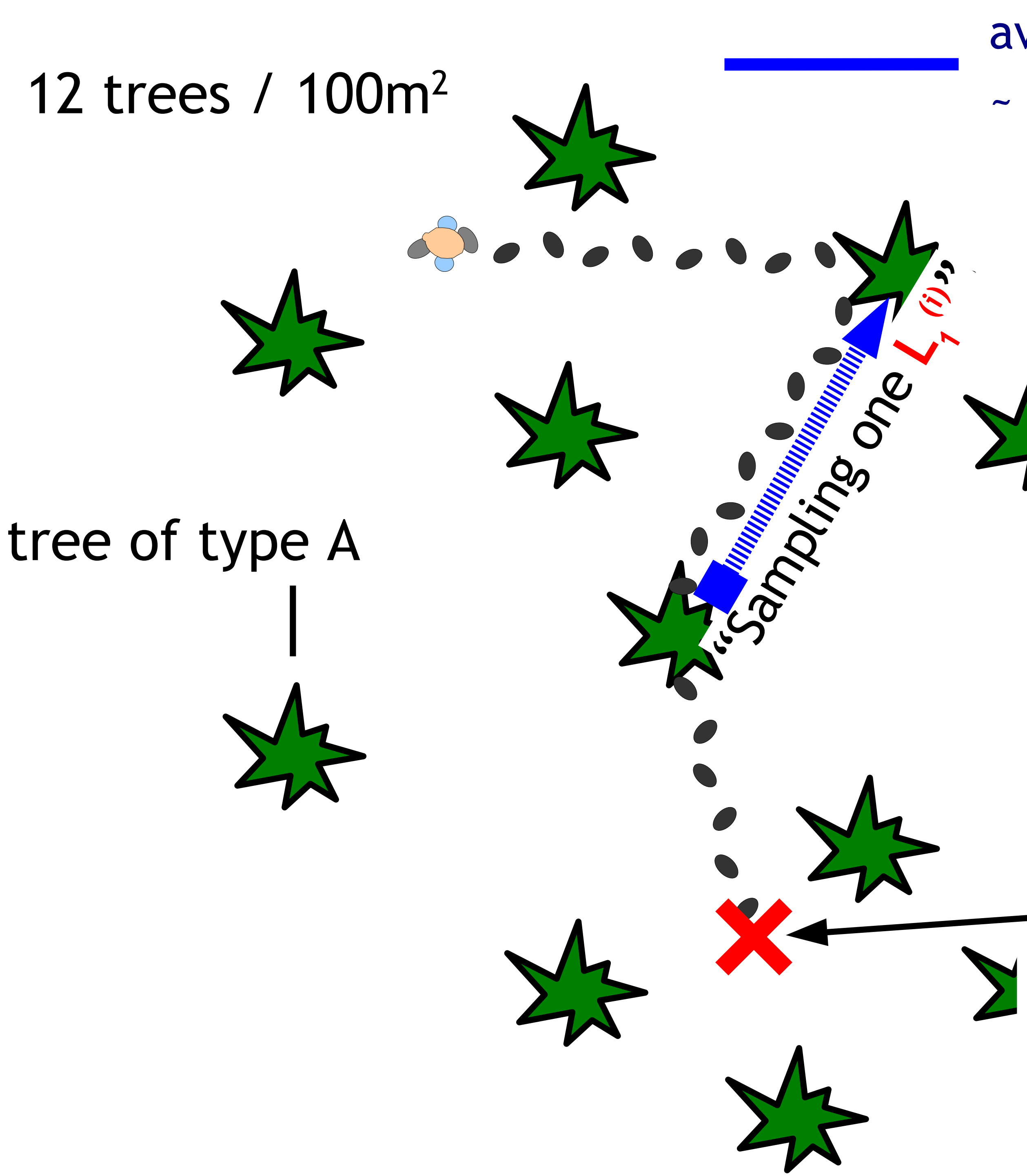
$\lambda_{ij} := 1/(\rho_{ij} \cdot \sigma_i)$ ... mean free path length,

average distance a particle moves undisturbed with respect to **process i** in **material j**

The Finnish For(r)est!

- Ingredients
  - the Finnish Forest – **mixted tree types**
  - Forrest Gump
  - a Finnish night in winter or Forrest with closed eyes
- Forrest runs through the forest ..

average distance btw. trees
~ mean free path λ_1



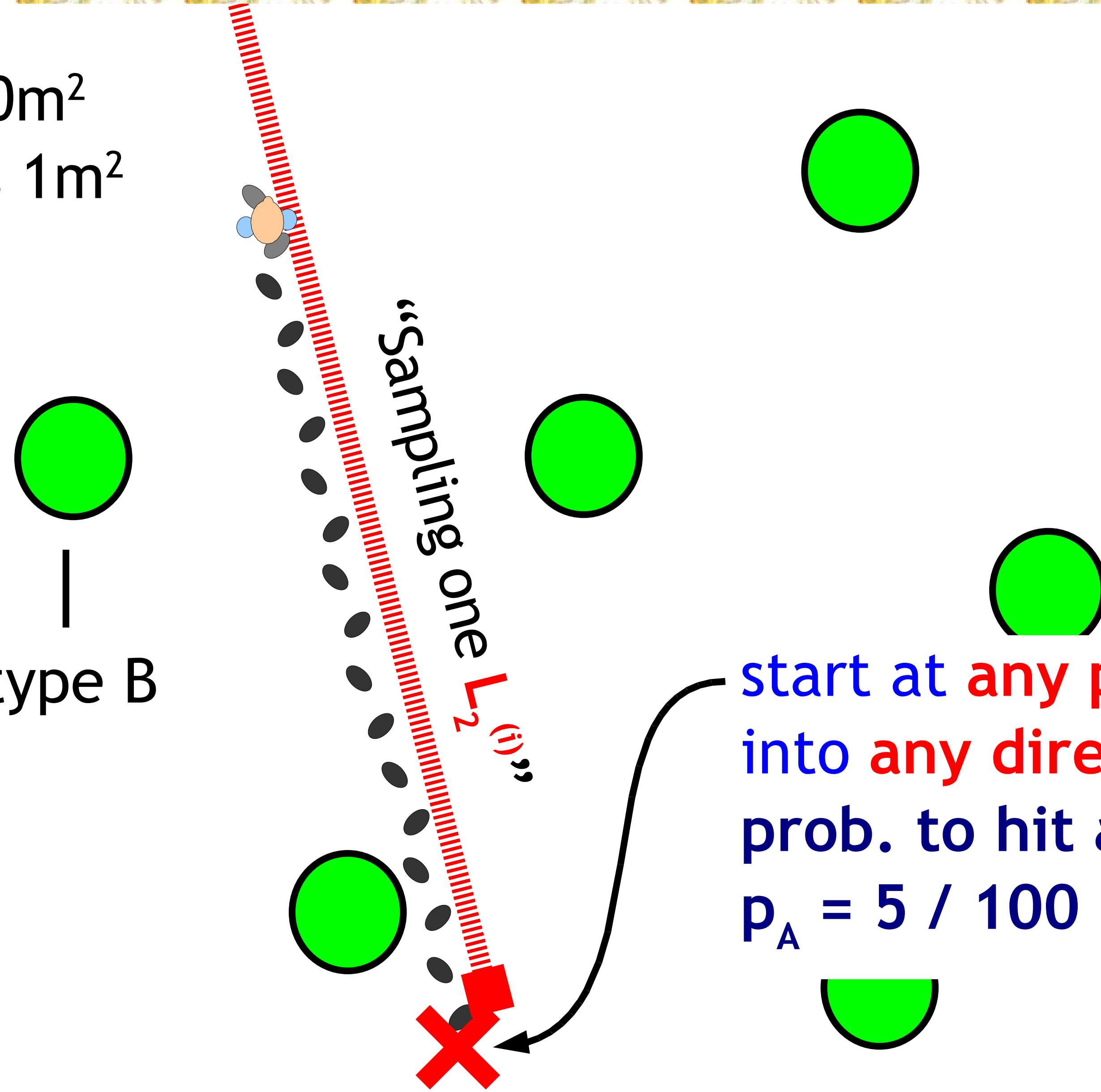
start at any point, into any direction:

prob. to hit an A-tree

p_A = 12 / 100

5 trees / 100m^2

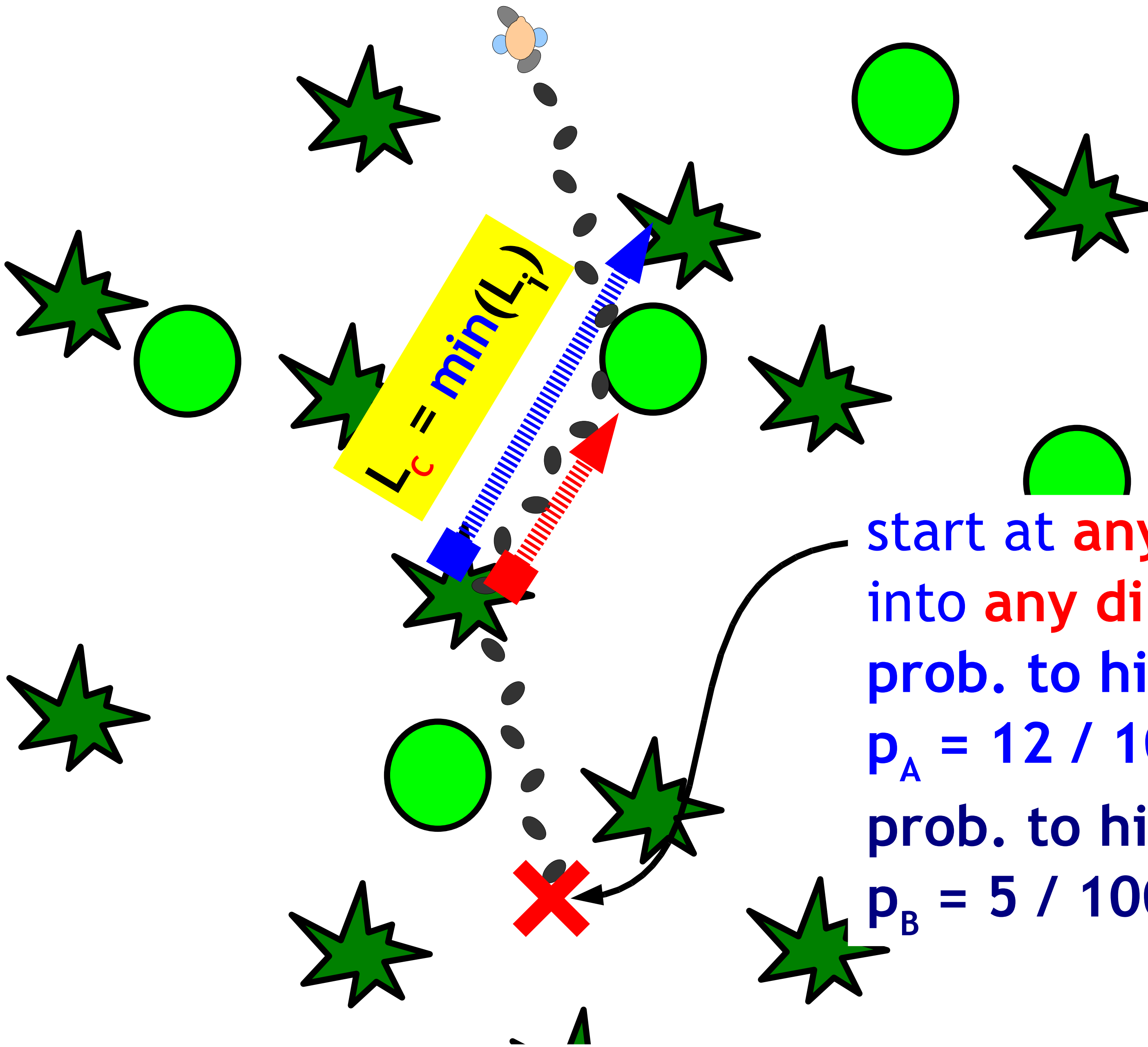
1 tree needs 1m^2



tree of type B

start at **any point**,
into **any direction**:
prob. to hit an B-tree
p_A = 5 / 100

average distance btw. trees
~ mean free path λ_2



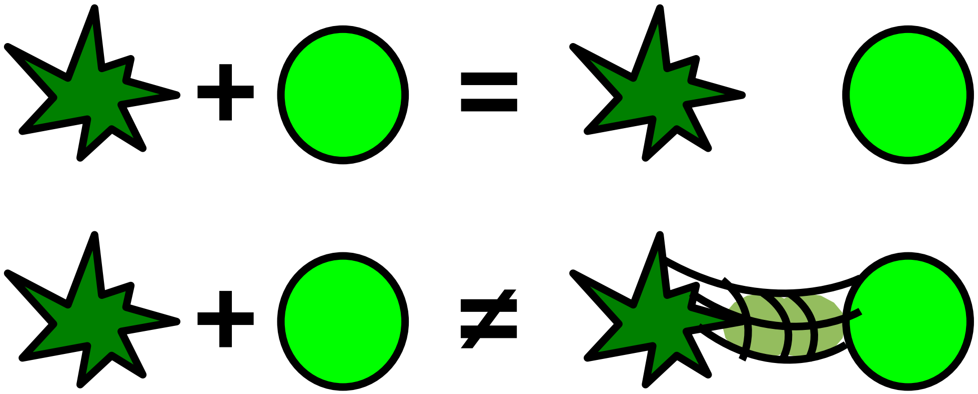
start at any point,
into any direction:
prob. to hit an A-tree
$p_A = 12 / 100$
prob. to hit a B-tree
$p_B = 5 / 100$

average distance btw. A and B trees
~ mean free paths λ_1, λ_2

start at any point,
into any direction:
prob. to hit an A-tree
$p_A = 12 / 100$
prob. to hit a B-tree
$p_B = 5 / 100$

prob. to hit any tree
$p = p_A + p_B$

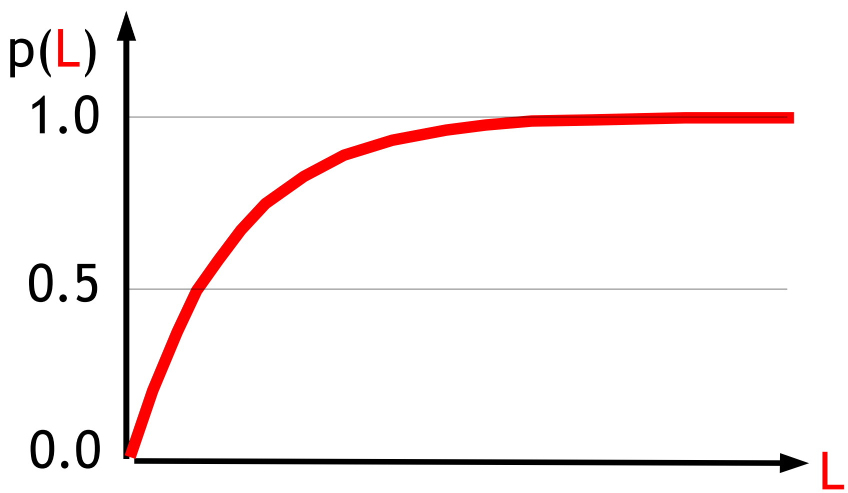
.. because mixing the tree types does not change the “interaction” properties of both types!

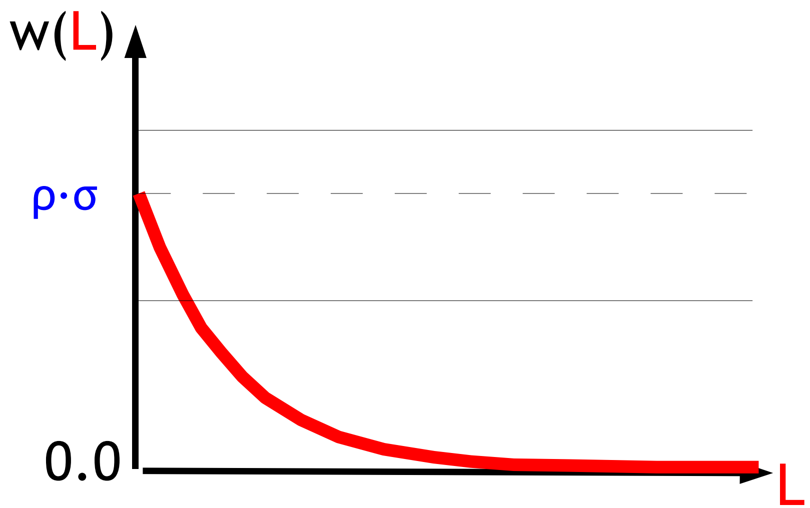


Most of the relevant physics processes in HEP simulation can be treated in the same manner!

How to "random" ...

- The Monte Carlo algorithm moves particles forward
  - by drawing a random number from the distribution $w(L) = 1/\lambda \exp(-L/\lambda)$, $p(L) = \int^{L} w(x)dx = 1-\exp(-L/\lambda)$
  - L being the random variable, L in $[0, \infty)$
  - in words: probability that a particle will have its first interaction in the infinitesimal interval [L, dL] is w(L)dL
- How can the computer compute L's accoring to w(L)??





The Problem

Given:

w(x) ... probability density function – our best knowledge of the system, or equivalently $F(a) = \int w(x)dx$

Task:

Find a statistical ensembles $\{x^{(m)}\}$ having F(a) as its underlying distribution with density w(x)!

Or, the other way round:

How can we produce a series $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ such that, given this series, we can conclude that it stems from the distribution w(x) for a sufficiently large amount of m.

One solution: the inverse method

- It's sufficient to be able to generate random numbers uniformly distributed in the interval [0,1]!
- All distributions are related to the uniform distribution in [0,1] through their cumulative distribution:
  - $y_{(i)}$ ... set of i=1,2,.., n uniformly distributed values in [0,1]
  - w(x) ... probability distribution of x
  - $F(y) = \int_{-\infty}^{y} w(x)dx$ ... cumulative distribution function
  - **$x_{(i)} = F^{-1}(y_{(i)})$ ... set of i=1,2,..,n random values distributed according to w(x)**
- This is the mathematical pillar relating all distributions with the uniform in [0,1]
- There are many other techniques not covered here ...

Random number generators

- **“Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin”**
  - John von Neumann (1951)
- Computer use arithmetical methods to produce random numbers, i.e. pseudo random numbers
  - typical scenario: one instance per application to generate [0,1] uniformly distributed doubles
  - reproducibility!!
- Difficult task to design good pseudo random number generators
  - all your results finally depend on them:
    generator must be unbiased, exhaustive, ..
  - various statistical tests exists to check their quality

Random number generators

- "Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin"
  - John von Neumann (1951)
- Computer use arithmetical methods to produce random nur
  - t … erate
  - r
- Dif … r gen
  - a … ot re-
  - various statistical tests exists to check their quality

```
double x = rand();
double z = norm(m, v);
....
```

We'll consider these problems as being solved by our simulation tool kit!

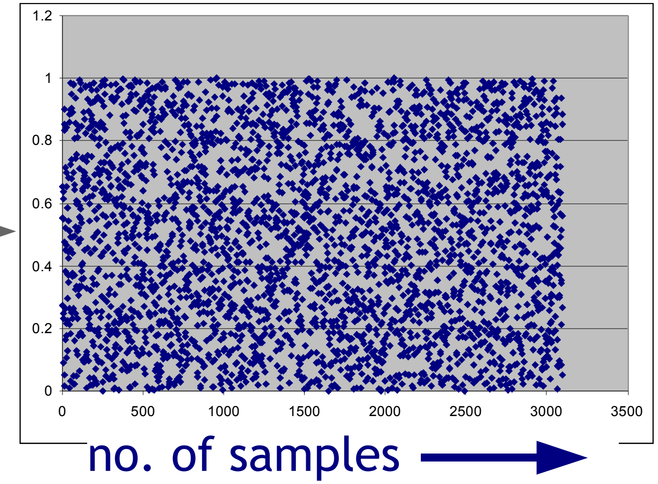
Our example

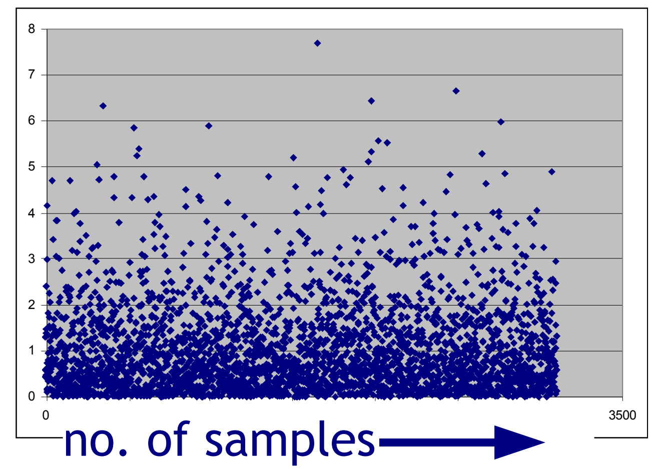
$w(L) = 1/\lambda \exp(-L/\lambda)$ ... distribution function

$p(L) = {}_0\int^L w(x)dx = 1-\exp(-L/\lambda)$ .. cummulative distr.

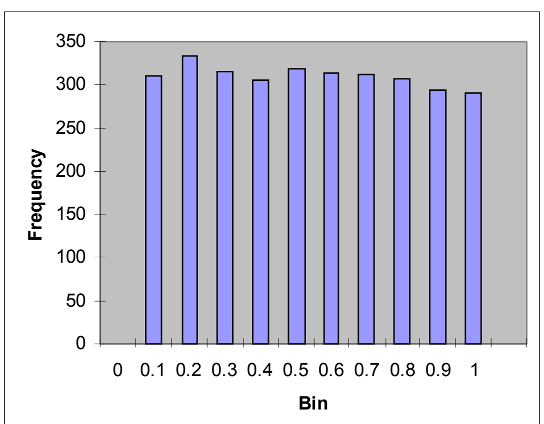
$p^{-1}(y) = -\lambda \ln(1-y)$ .. inverse commulative distr.

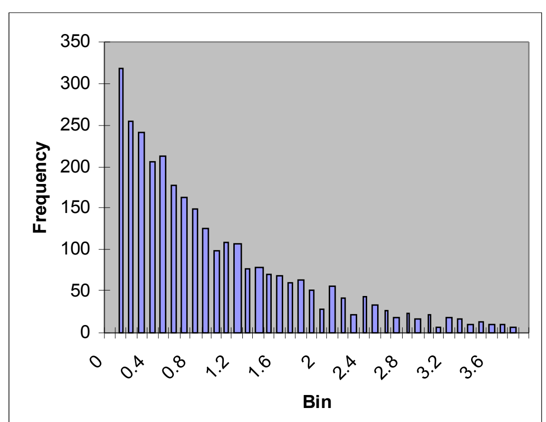
$L_{(i)} = -\lambda \ln(1-y_{(i)})$, $y_{(i)}$ .. uniformly in [0,1]

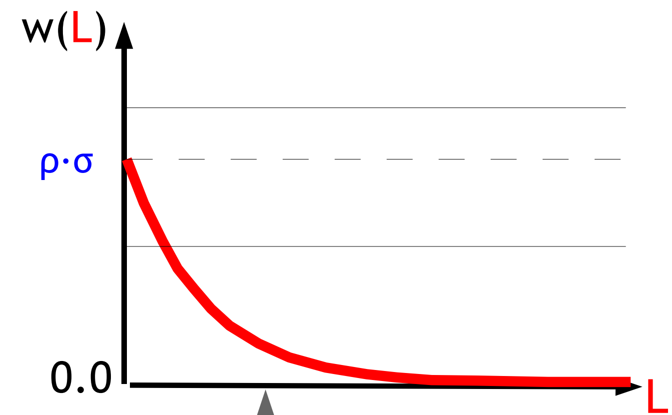




Histograms:







Quick flashback ...

- What we wanted to do:
  - follow the histories of single particles through the detector
  - keep track of these histories wherever we are interested in getting the resulting distributions
- Details of a physics process
  - interaction probabilities
  - cross sections: total and differential cross-section
- Interaction with bulk material
  - exponential distribution, mean free path length
- **Monte Carlo algorithm**
  - sampling from distributions
  - random numbers

Quick outlook ...

- What we wanted to do:
  - follo ... tector
  - keep ... sted in gett
- Details
  - inte
  - cross
- Interac
  - expo
- Monte
  - sampling from distributions
  - random numbers

In the following, we want to show how the introduced concepts and methods are mapped to the implementation of the GEANT4 tool kit.

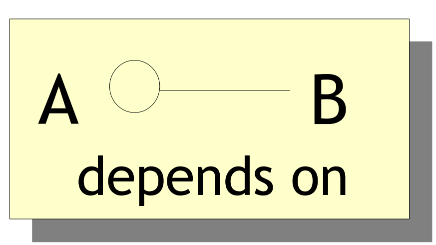
Then we want to see, how we can make use of GEANT4 in setting up our own simulation.

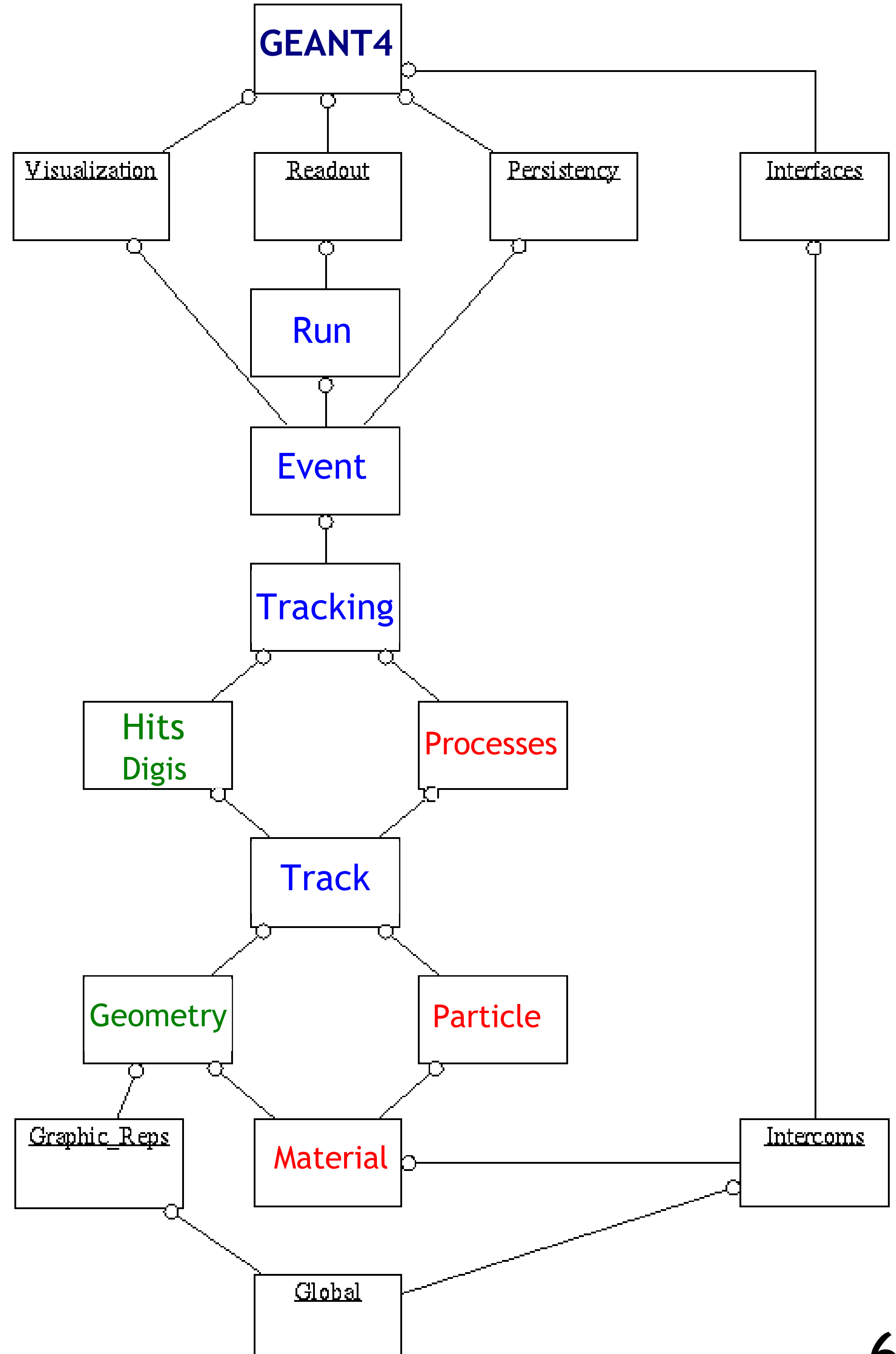
Introduction to GEANT4

Source code packages:

The coloured packages will be treated in the following.

They implement the simulation principles we have seen before.





Simulation flow in G4

- Initialization procedures
  - definition of physics processes and their particles to be used for the simulation
  - definition of the detector geometry including materials
  - definition of user-hooks to extract simulation data
- Event loop
  - generation of primary particles
  - tracking of primary particles and their decendents
  - extraction of simulation date during tracking

"Only" a tour of the basic G4 concepts

- Only the basic concepts of G4 will be discussed in the G4 section of these lectures
  - based on the topics we have seen up to now
  - new concepts will be introduced when required
- Will not discuss / explain in detail the G4 API, all the methods, signatures, ...
  - everybody can read the API-docs! ;-)
  - we have the exercises to get acquainted with the gory details
  - will discuss some essential classes and their interrelation in order to show how they implement the simulation requirements

Building blocks

- We will start to describe the basic building blocks for setting up a GEANT4 simulation
  - description of particles and their initial states
  - material definitions
  - detector description in terms of geometry & materials
  - physics processes
- Then we will analyse the dynamics of a simulation
  - starting the event-loop
  - following the particles
  - extracting simulation information

Quantities and units in GEANT4

"For those who want some proof
that physicists are human,
the proof is in the idiocy of **all
the different units** which they
use for measuring energy."

The Character of Physical Law (1967)
R. P. Feynman

Simple rules

- Every quantity such as mass, energy, .., is represented by the C++ **`double`** data type
- To express a quantity in a unit, the appropriate unit scaling factor must be applied
  - for creating or passing a quantity to G4, multiply the numerical value by the unit, e.g.
    ```
    double speed = 10*m/sec;
    ```
  - for reading returned quantites from G4, divide the numercial value by the unit you want the result to be in:
    ```
    double speed = getSpeed()/(km/h);
    ```
  - several unit constants are already predefined, e.g. **`cm, mm, kg, sec, ...`**

C++ example for units

Code:

```
double energy = 1.*GeV;
cout << "Energy of 1 GeV:" << endl;
cout << " in ev    : " << energy/eV << endl
     << " in Joule: " << energy/joule << endl;

double h = 60.*60.*s;
double velocity = 20.*km/h;
cout << "Velocity of 20 km/h:" << endl
     << "  internal : " << velocity << endl
     << "  km/h     : " << velocity/(km/h) << endl
     << "  m/sec    : " << velocity/(m/s) << endl;
```

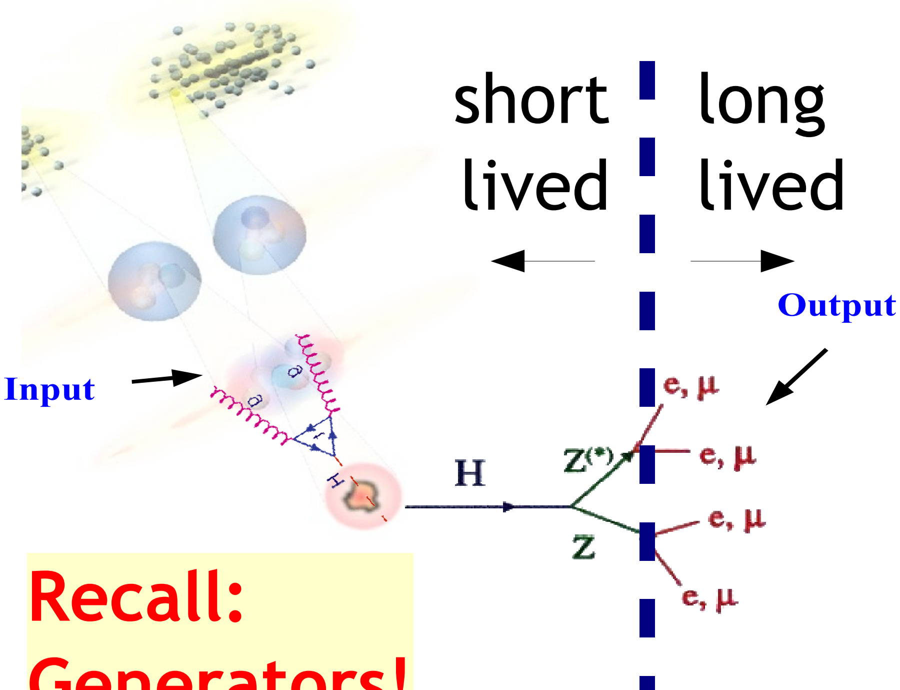
Result:

```
Energy of 1 GeV:
 in ev    : 1e+09
 in Joule: 1.60218e-10

Velocity of 20 km/h:
  internal : 5.55556e-06
  km/h     : 20
  m/sec    : 5.55556
```

Description of particles

- GEANT4 is designed to transport particles
  - along macroscopic distances in bulk matter
  - transported particles are "long lived"
- We have to tell G4 the initial conditions of the particles we want to have simulated



Recall: **Generators!**

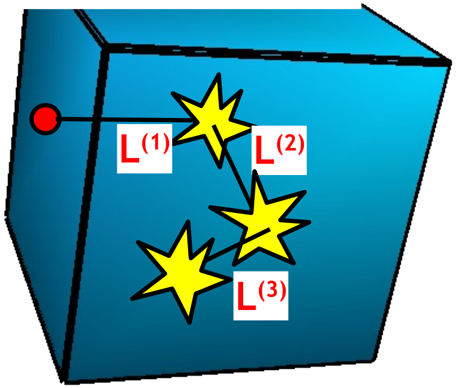
Example:

G4 will track the leptons ($e^{+,-}$, $\mu^{+,-}$), but not the H and the Z's.

However, we want to keep track of the information, from where the leptons were coming from -> analysis!

Primary Particles

Remember?



Very basic algorithm:

(1) initial values for incident particle
(2) get values for ρ and σ
(3) sample $L^{(n)}$ from p(L)
(4) transport particle
undisturbed by $L^{(n)}$
(5) simulate interaction
(6) if particle still exists:
goto (1)

We have to tell G4 the starting conditions of all the particles we want to track through the material!

Need these classes:

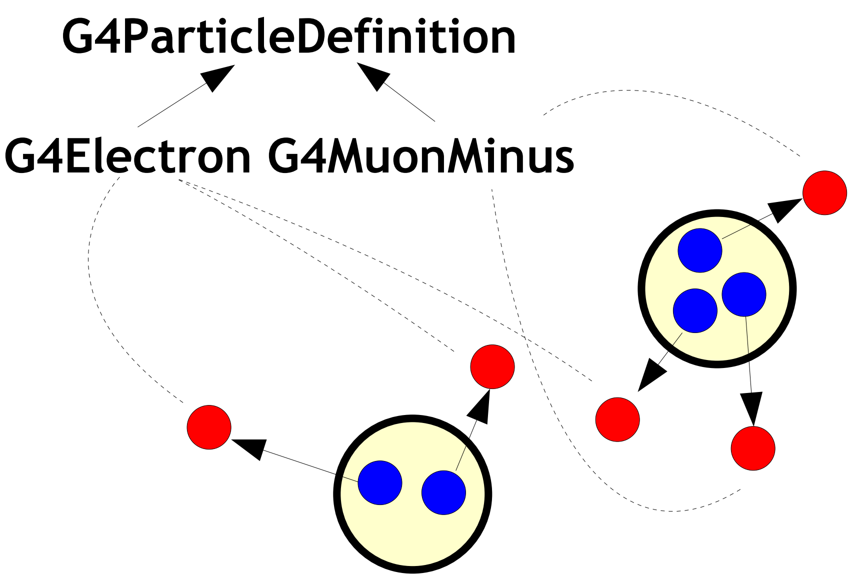
G4ParticleDefinition and subclasses: particle type such as e^-, μ^+ and its static data such as charge, mass, spin, ..

G4PrimaryParticle: momentum, kinetic energy of a particle described by [G4ParticleDefinition]

G4PrimaryVertex: position and time of a list of [G4PrimaryParticle]

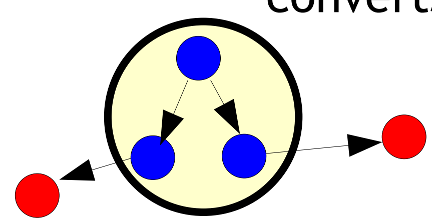
Primary Particles

When starting the simulation, G4 takes all the information of the instances of **G4PrimaryVertex**, **G4PrimaryParticle**, and **G4ParticleDefinition**, and creates **G4DynamicParticle** instances. These correspond to the particles being tracked.



G4PrimaryParticle
not tracked, for book-keeping
(x,z,z,t), momentum, energy, daughters, ptr. to particle type

Geant4: converts (blue) to (red)



G4DynamicParticle
tracked by G4
(x,z,z,t), momentum, energy, ptr. to type

G4PrimaryVertex
(x,y,z,t)

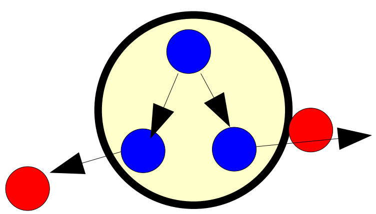
Primary Particles

G4ParticleDefinition

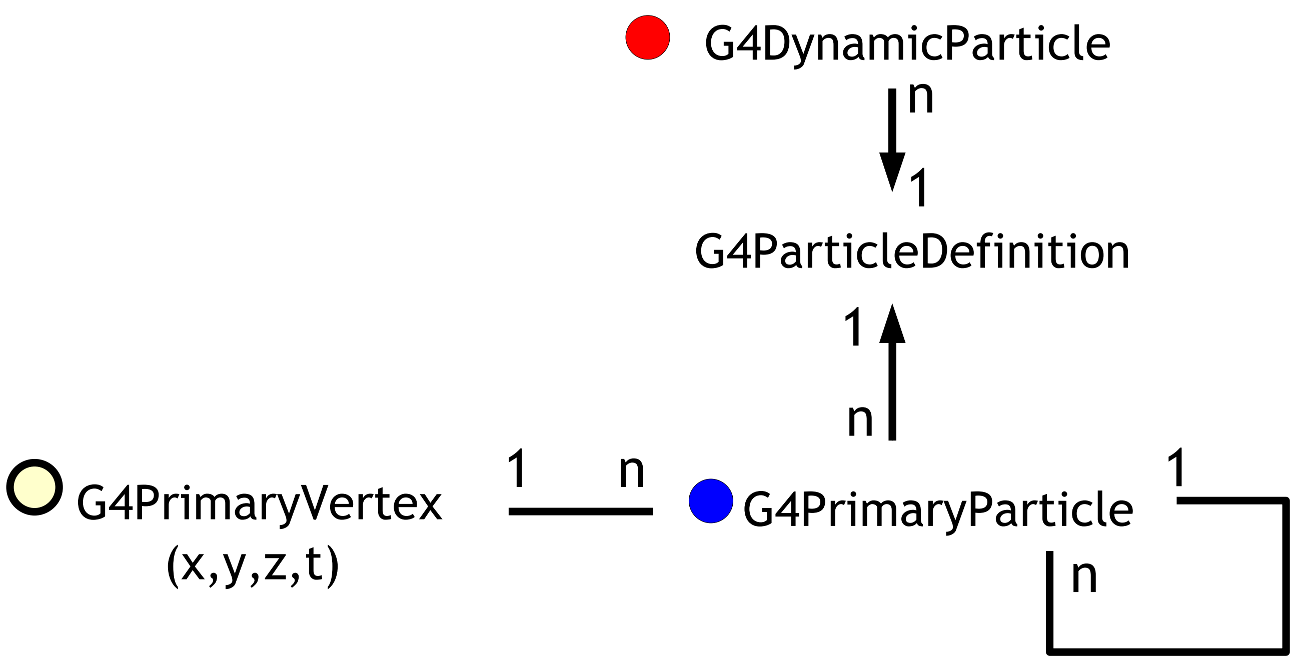
G4Electron G4MuonMinus
(particle type:
invariant properties such as charge, spin, ...)

G4PrimaryParticle
not tracked, for book-keeping
(x,z,z,t), momentum,
energy, particle type, daughters

G4DynamicParticle
tracked by G4
(x,z,z,t), momentum,
energy, ptr. to type



Geant4:
converts ● to ●

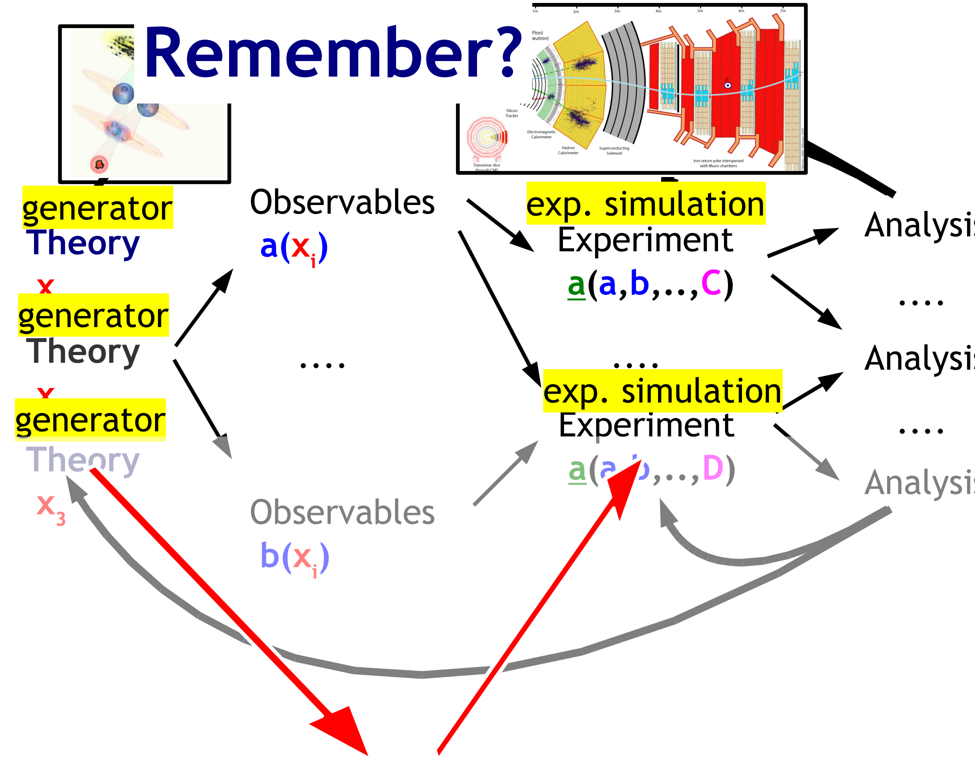


You have to instantiate these:

G4PrimaryVertex, G4PrimaryParticle

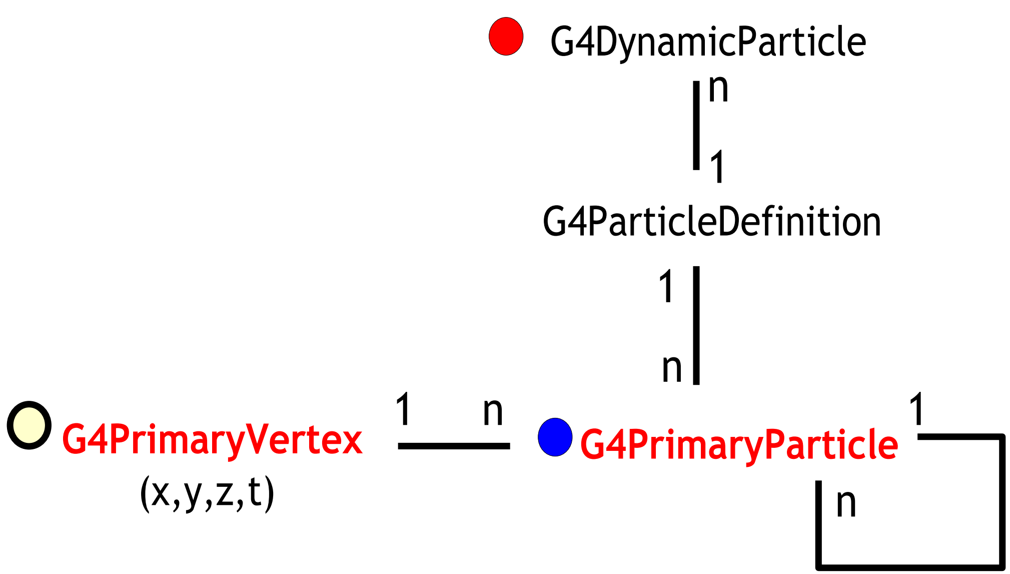
in a user-action.

(User-actions are discussed later.)



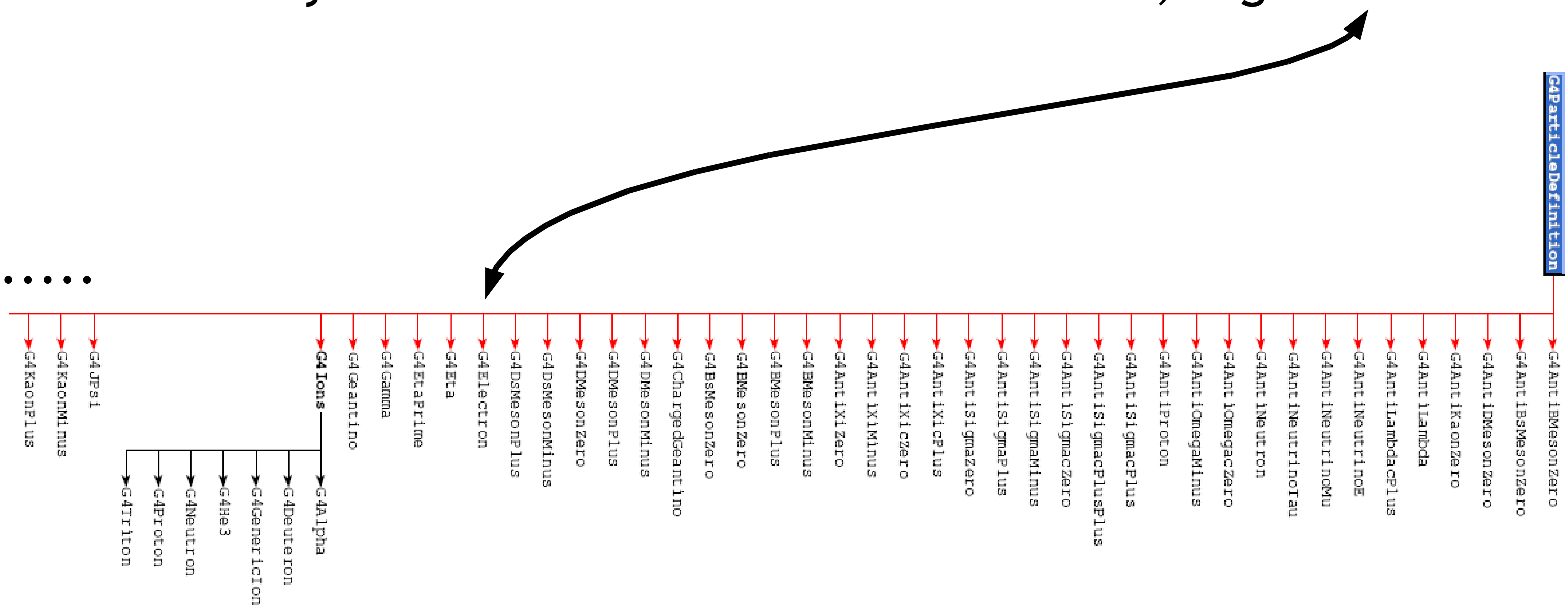
user-action

The implementation of the above mentioned user-action is the link between generators and experiment simulation!



G4DynamicParticle, -ParticleDefinition

A **G4DynamicParticle** contains the current, dynamic state of the particle during a simulation: position, time, momentum, energy; it also has a pointer to the particle type information modelled by a subclass of **G4ParticleDefintion**, e.g. **G4Electron**

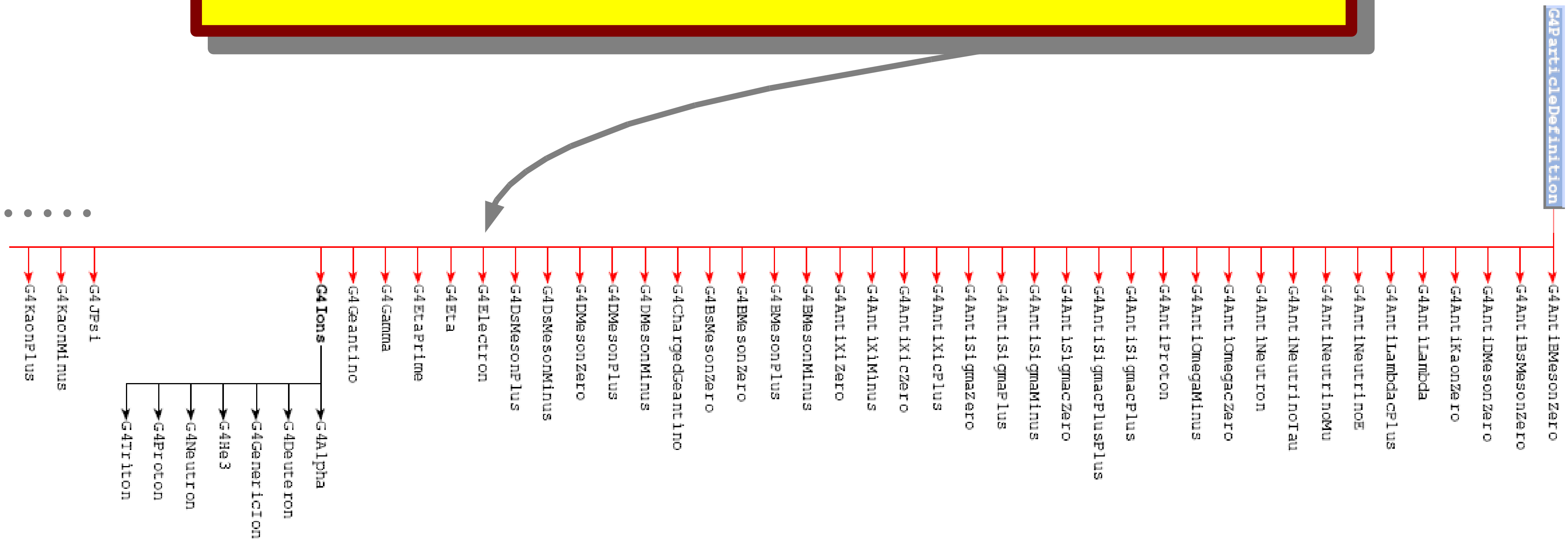


G4DynamicParticle, -ParticleDefinition

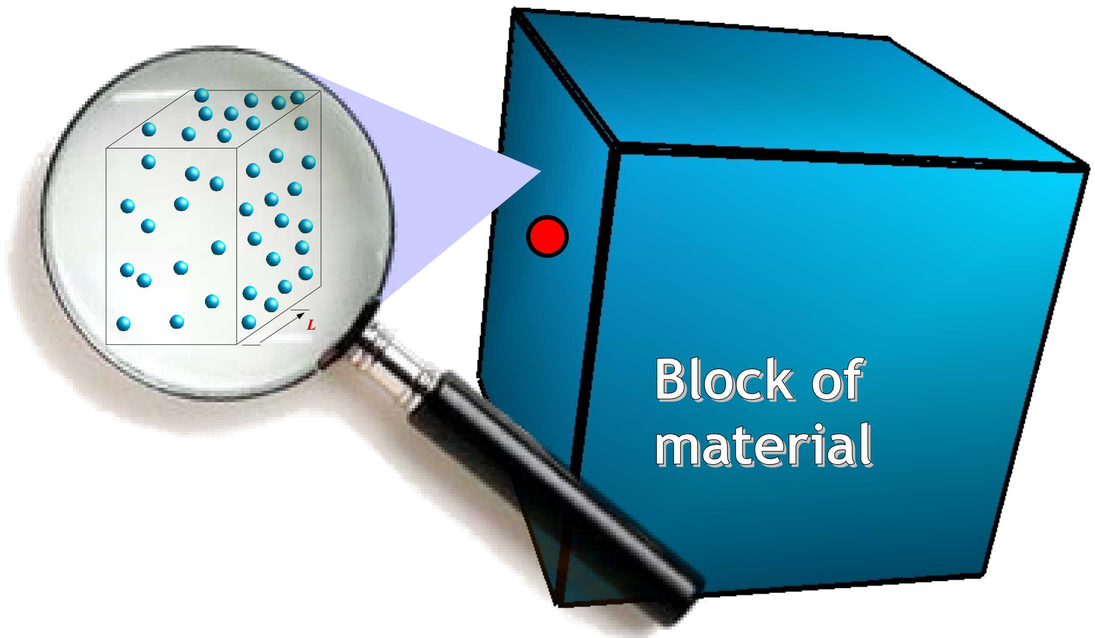
A G4Dy… …ate of
the pa… …tum,
energy… …ation
modell… …Electron

"Young man, if I could remember the names of these particles, I would have been a botanist!"

Enrico Fermi to his student Leon Lederman



Materials



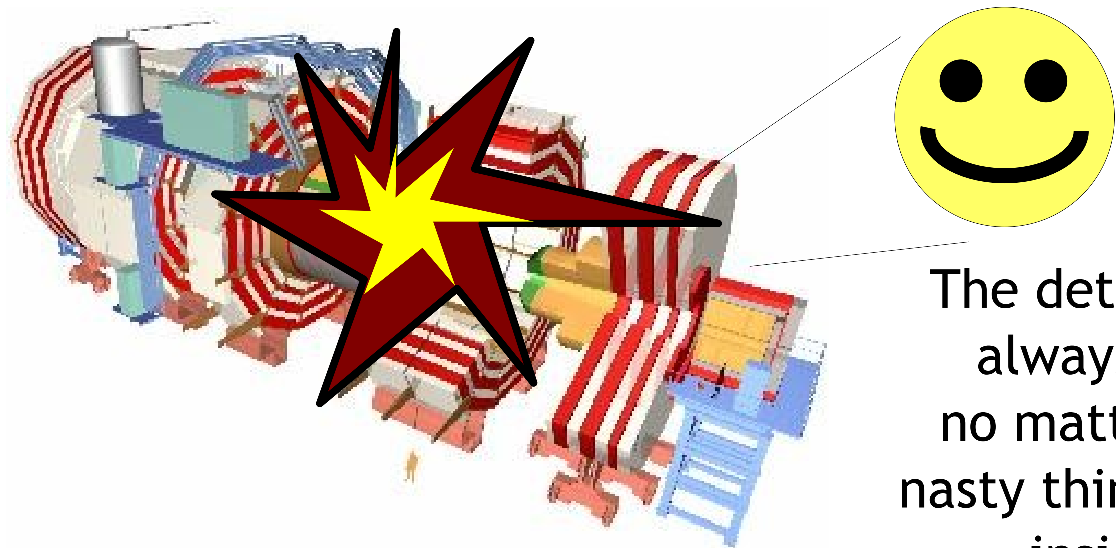
G4 supports the description of homogeneous materials, described by its elements and isotopes.

Properties of Materials in G4

- Materials are not affected by time
  - Gas does not diffuse, liquids don't flow
  - No thermodynamics: no heat transfer, no state changes (water <-> ice), ...

Properties of Materials in G4

- Materials are not affected by time
  - Gas does not diffuse, liquids don't flow
  - No thermodynamics: no heat transfer, no state changes (water <-> ice), ...
- Particle interaction does not harm!
  - No irradiation effects or radiation damages



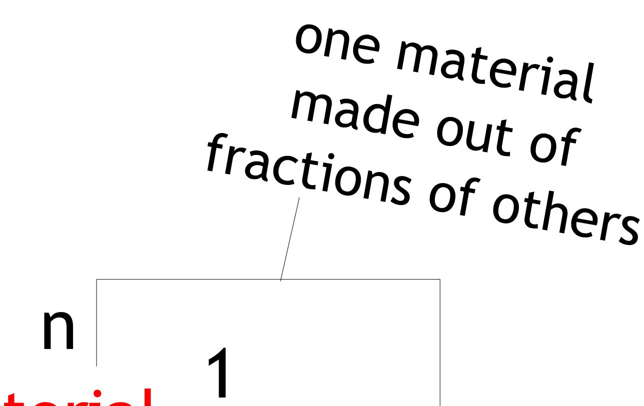
The detector will
always smile,
no matter which
nasty things happen
inside ;-)

Properties of Materials in G4

- Materials are not affected by time
  - Gas does not diffuse, liquids don't flow
  - No thermodynamics: no heat transfer, no state changes (water <-> ice), ...
- Particle interaction does not harm!
  - No irradiation effects or radiation damages
- Not affected by external influences. Materials don't change physical properties under
  - Mechanical influences (stresses, pressures)
  - External electromagnetic fields, i.e. materials don't get magnetized, electrostatically charged
    (-> see later: description of external fields)

So, what are G4 Materials?

- All materials are represented as objects of a single class: **G4Material**
- G4Material is a concrete class
  - only source of material information for G4
  - represents a homogeneous material made of elements, compounds, or mixtures of other materials
- There are several ways to define a material in a user-friendly way through helper classes;
  - isotopes <->G4Isotope
  - elements <->G4Element
  - molecules <->G4Material
  - compounds and mixtures <-> G4Material



class G4Material

excerpt from the public interface of G4Material:

G4Material | class | Filter

Edit | Hierarchy | Class | Xref | Include | Retriever | Grep

Name	Class	Type	Parameters
GetAtomicNumDensityVector(md)		const G4double *	()
GetFractionVector(md)		const G4double *	()
GetVecNbOfAtomsPerVolume(md)		const G4double *	()
GetElement(md)		const G4Element *	(G4int)
GetElementVector(md)		const G4ElementVector *	()
GetAtomsVector(md)		const G4int *	()
GetDensity(md)		G4double	()
GetElectronDensity(md)		G4double	()
GetNuclearInterLength(md)		G4double	()
GetPressure(md)		G4double	()
GetRadlen(md)		G4double	()
GetTemperature(md)		G4double	()
GetTotNbOfAtomsPerVolume(md)		G4double	()
GetTotNbOfElectPerVolume(md)		G4double	()

Remember:
GetDensity() / (kg/m*m*m)

Quick summary: G4

- Mapping of concepts ...
  - incident particles – primary particles
  - bulk material
- ... to GEANT4 class implementations
  - G4ParticleDefinition, G4PrimaryVertex, G4PrimaryParticle, G4DynamicParticle
  - G4Material & helpers
- Easy convention on physical units & quantities
  - different quantities not differently typed in C++, all is “double”
  - multiply / divide by predefined constants

Summary ~2~

- Passage of particles through bulk matter
  - cross section, free path length
  - exponential distribution
  - multiple physics processes
- Monte Carlo method
  - random numbers & sampling from distributions
  - basic tracking algorithm
    - for one active physical process
    - for multiple physical processes
- Gentle introduction to GEANT4
  - general flow
  - primary particles
  - materials
  - physical units & quantities