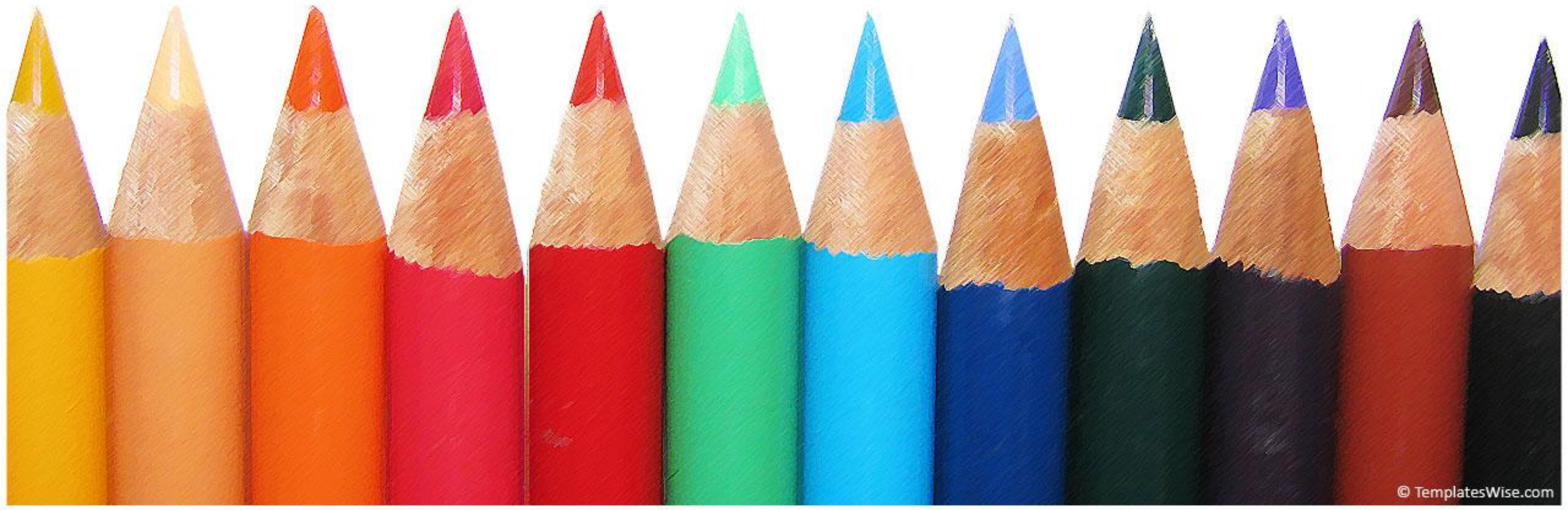
FastSim tutorial for beginners

Matteo Rama

Laboratori Nazionali di Frascati

1st SuperB Collaboration meeting

London, September 2011



Part I

FastSim overview

Schematic view of FastSim

1

Event Generators
- EvtGen
- KK2F
- …

Backgrounds
- Rad. Bhabha
- $e^+e^- \rightarrow e^+e^-e^+e^-$
- neutrons (Geant4)
- …

GTracks

TParticles

2

Interaction of particles with detector
- Simplified material x-sections
- Showering, decay, brems, …
- Voxel-based navigation

Configuration data (EDML)
- Detector geometry
- Material properties
- Measurement parameters
- …

SimTracks (+SimHits)

3

Reconstruction Simulation
- Parameterized detector response
- Kalman Filter track fit
- Hit merging and "pat. rec. "errors
- Energy merging and clustering

Reco Tracks, Clusters, PID

4

"BaBar" Analysis
- Composite reconstruction
- Event selection
- Physics reach evaluation

ROOT Tree Output

Event generation

Physics events

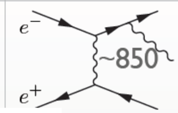
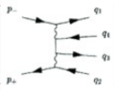
Generators available:

Event type	Generator
$e^+e^- \rightarrow Y(4S) \rightarrow$ BBbar $e^+e^- \rightarrow$ qqbar (q=u,d,s,c) $e^+e^- \rightarrow$ Psi(3770) $\rightarrow$ DDbar …	EvtGen + JETSET
$e^+e^- \rightarrow \tau^+\tau^-$ $e^+e^- \rightarrow \mu^+\mu^-$	Kk2f + tauola Kk2f
$e^+e^- \rightarrow e^+e^-$ (large polar angle)	Bhwide
single particles (for det. studies) …	SingleParticle

Event generation

Backgrounds

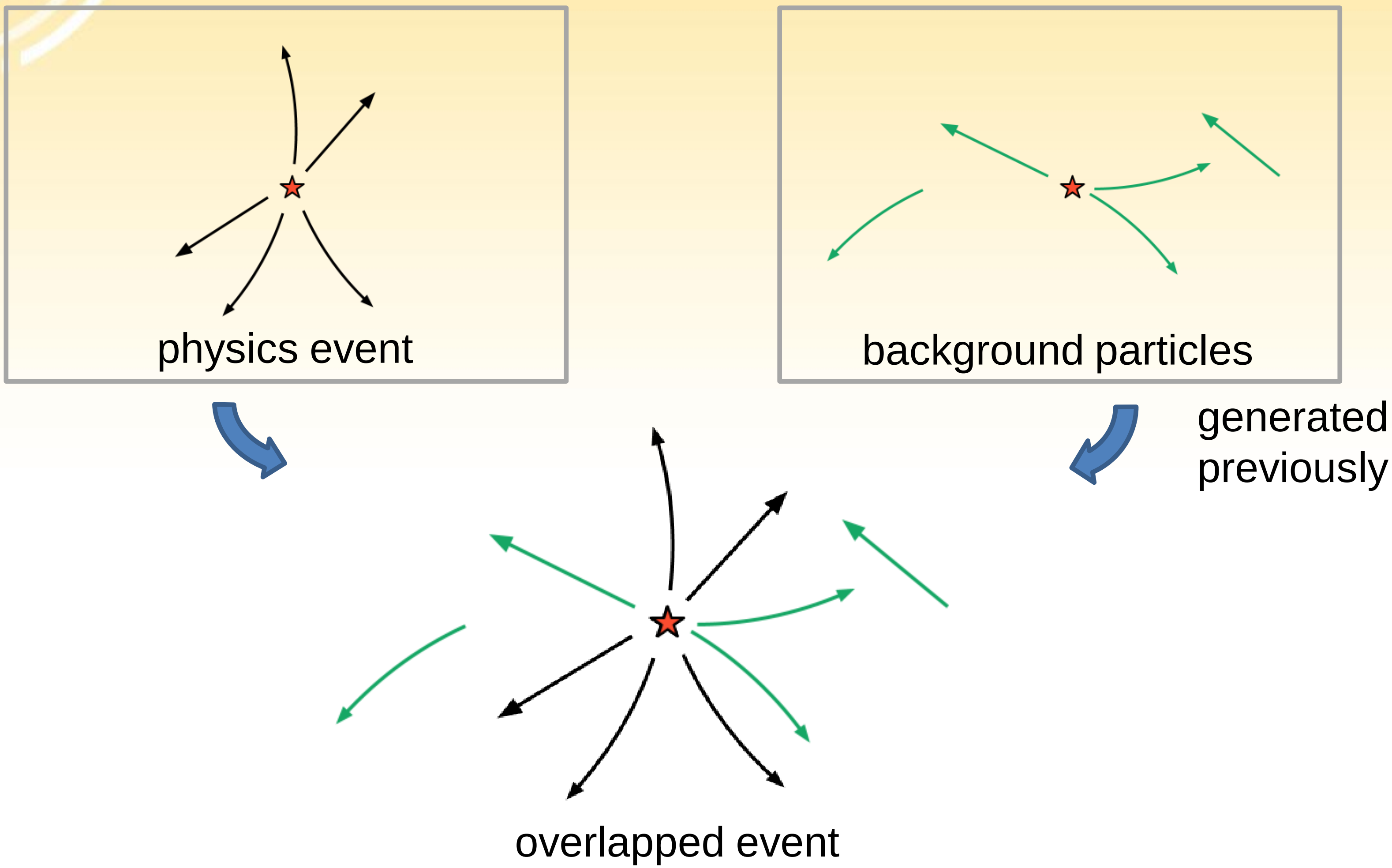
- Backgrounds can be optionally overlapped to the physics event
  - radiative Bhabhas
  - e+e- pairs
  - elastic Bhabhas

	Cross section	Evt/bunch xing	Rate
Radiative Bhabha	~340 mbarn (Eγ/Ebeam > 1%)	 ~850	0.3THz
e^+e^- pair production	~7.3 mbarn 	~18	7GHz
e^+e^- pair (seen by L0 @ 1.5 cm)	~0.3 mbarn	~0.8	0.3GHz
Elastic Bhabha	O(10^{-4}) mbarn (Det. acceptance)	~250/Million	100KHz
ϒ(4S)	O(10^{-6}) mbarn	~2.5/Million	1 KHz
	Loss rate	Loss/bunch pass	Rate
Touschek (LER)	14kHz / bunch (+/- 2 m from IP)	~7/100	14 MHz

- Backgrounds particles are generated previously in separate productions and stored in ROOT files
  - Radiative Bhabhas generated with Bruno
  - e+e- pairs and elastic Bhabhas generated with FastSim

- When FastSim generates the physics event, background particles are overlapped according to their expected rates

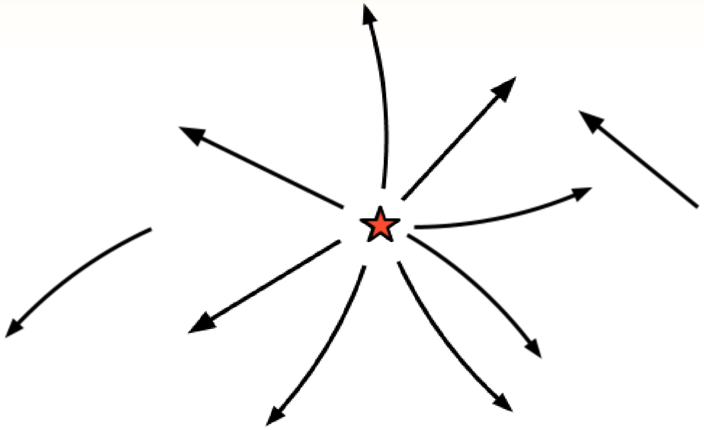
Event generation



Event generation

From this point on, the simulation makes no distinction between particles from backgrounds and particles from the physics event

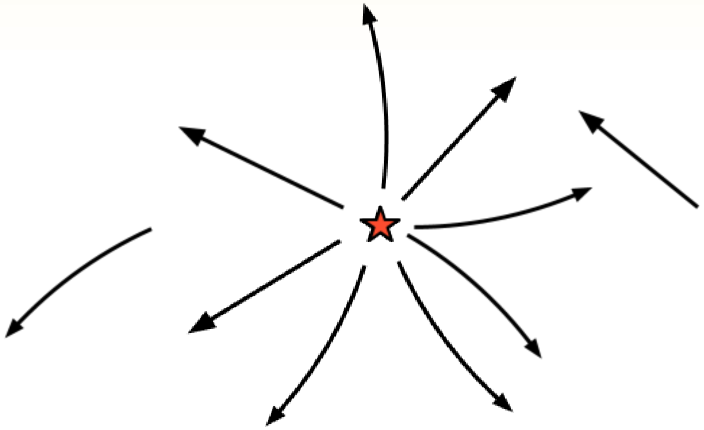
But NB: particles from the physics event and from background differ for the time distribution. The formers are peaked at t~0 (by definition).



overlapped event

Event generation

Examples to include backgrounds are in package PacProduction and later in this tutorial.
See also
http://mailman.fe.infn.it/superbwiki/index.php/FastSimDoc/Production



overlapped event

Detector simulation

- cylindrical symmetry
- detector elements modeled as surface sections
  - cylinders, rings, cones, rectangles
- configuration via XML (EDML) defines:
  - geometries
  - materials
  - measurement parameters

The SuperB detector master file is
PacDetector/pacrat_SuperB.xml

Detector simulation

from PacDetector/pacrat_SuperB.xml:

```
<!-- the following includes add volumes and elements to the detector, and define the specific measurements and
configurations of those detectors.  They need to be included in order from the IP outwards -->

<include file="PacDetector/IP_SuperB_shielded.xml"  />

<include file="PacTrk/Si_SuperB.xml"  />

<include file="PacTrk/Dch_SuperB.xml" />

<include file="PacEmc/PacEmcGeom_SuperB.xml" />

<include file="PacEmc/EmcBwd_SuperB.xml" />

<include file="PacDirc/Dirc_SuperB.xml" />

<include file="PacForwardPid/ForwardPid_SuperB.xml" />

<include file="PacEmc/Emc_SuperB.xml" />

<include file="PacIfr/Ifr_SuperB.xml" />

<include file="PacDetector/Machine_SuperB.xml"  />

<!-- generic configuration -->

<include file="PacTrk/Track_reconstruction.xml" />

<include file="PacSim/Material_simulation.xml" />

<include file="PacDetector/SuperBVolume.xml" />
```

Example: DCH

PacTrk/Dch_SuperB.xml
- PacTrk/Dch_SuperB_Geom_baseline.xml
- PacTrk/Dch_SuperB_Measures.xml

from PacTrk/ Dch_SuperB_Geom_baseline.xml:

```
<?xml version="1.0" encoding="UTF-8" ?>
<edml>
   <included>
                <define>
                <constant name="wirezmin" value="-101.5"></constant>
                <constant name="wirezmax" value="174.9"></constant>
                </define>
      <detector>
         <volume name="Drift_Tracking">

            <cyl name="DchInnerCyl" id="101" zmin="wirezmin" zmax="wirezmax" radius="23.60" thick="0.1000" mat="dch-CFiber" />
            <cyl name="dch-Wires" id="151" zmin="wirezmin" zmax="wirezmax" radius="26.05" thick="0.0120" mat="dch-Wires" gap="0.985" />
            <cyl name="dch-He-Ibu" id="152" zmin="wirezmin" zmax="wirezmax" radius="26.06" thick="1.4300" mat="dch-He-Ibu" meas="Axial,DchdEdx" />
            <cyl name="dch-Wires" id="153" zmin="wirezmin" zmax="wirezmax" radius="27.24" thick="0.0120" mat="dch-Wires" gap="0.986" />
            <cyl name="dch-He-Ibu" id="154" zmin="wirezmin" zmax="wirezmax" radius="27.25" thick="1.4300" mat="dch-He-Ibu" meas="Axial,DchdEdx" />
            <cyl name="dch-Wires" id="155" zmin="wirezmin" zmax="wirezmax" radius="28.43" thick="0.0120" mat="dch-Wires" gap="0.987" />
…
```

Example: drift chamber shaped as cylindrical layers of wires and gas

inner wall

gas

wires

Physical properties of materials stored in file database (PacEnv/MaterialsList.data)
- density, A, Z, radiation/int. lenghts…

Example: DCH

PacTrk/Dch_SuperB.xml
- PacTrk/Dch_SuperB_Geom_baseline.xml
- PacTrk/Dch_SuperB_Measures.xml

from PacTrk/ Dch_SuperB_Geom_baseline.xml:

Example: drift chamber shaped as cylindrical layers of wires and gas

inner wall   gas   wires

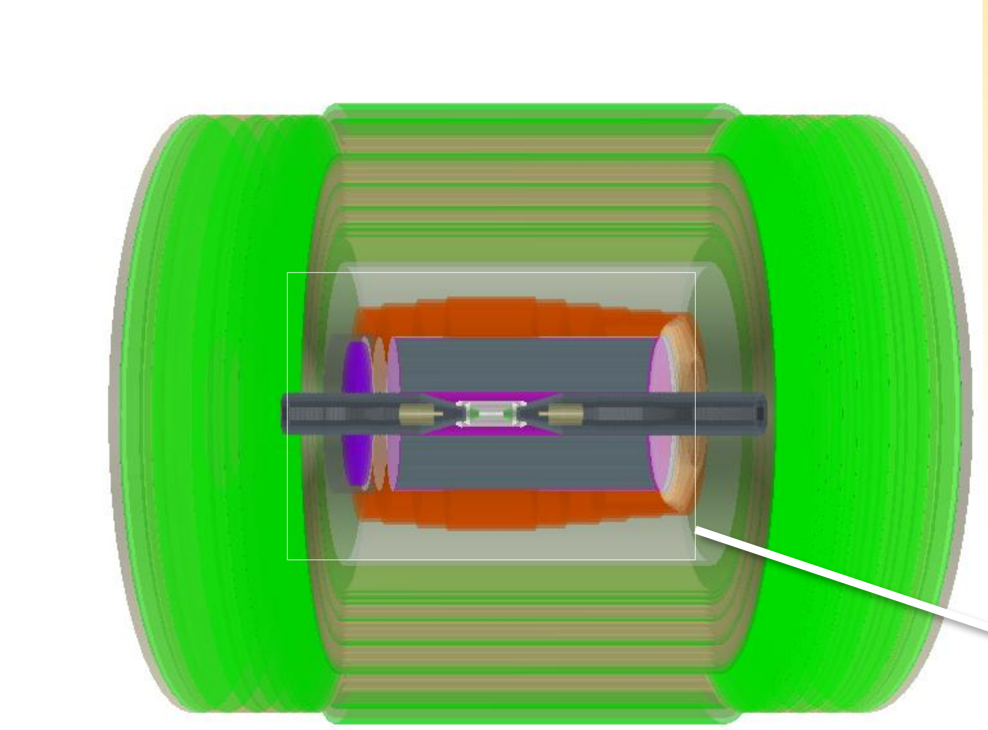
```
<?xml version="1.0" encoding="UTF-8" ?>
<edml>
  <included>
        <define>
        <constant name="wirezmin" value="-101.5"></constant>
        <constant name="wirezmax" value="174.9"></constant>
        </define>
    <detector>
      <volume name="Drift_Tracking">

        <cyl name="DchInnerCyl" id="101" zmin="wirezmin" zmax="wirezmax" radius="23.60" thick="0.1000" mat="dch-CFiber" />
        <cyl name="dch-Wires" id="151" zmin="wirezmin" zmax="wirezmax" radius="26.05" thick="0.0120" mat="dch-Wires" gap="0.985" />
        <cyl name="dch-He-Ibu" id="152" zmin="wirezmin" zmax="wirezmax" radius="26.06" thick="1.4300" mat="dch-He-Ibu" meas="Axial,DchdEdx" />
        <cyl name="dch-Wires" id="153" zmin="wirezmin" zmax="wirezmax" radius="27.24" thick="0.0120" mat="dch-Wires" gap="0.986" />
        <cyl name="dch-He-Ibu" id="154" zmin="wirezmin" zmax="wirezmax" radius="27.25" thick="1.4300" mat="dch-He-Ibu" meas="Axial,DchdEdx" />
        <cyl name="dch-Wires" id="155" zmin="wirezmin" zmax="wirezmax" radius="28.43" thick="0.0120" mat="dch-Wires" gap="0.987" />
…
```

from PacTrk/ Dch_SuperB_Measures.xml:

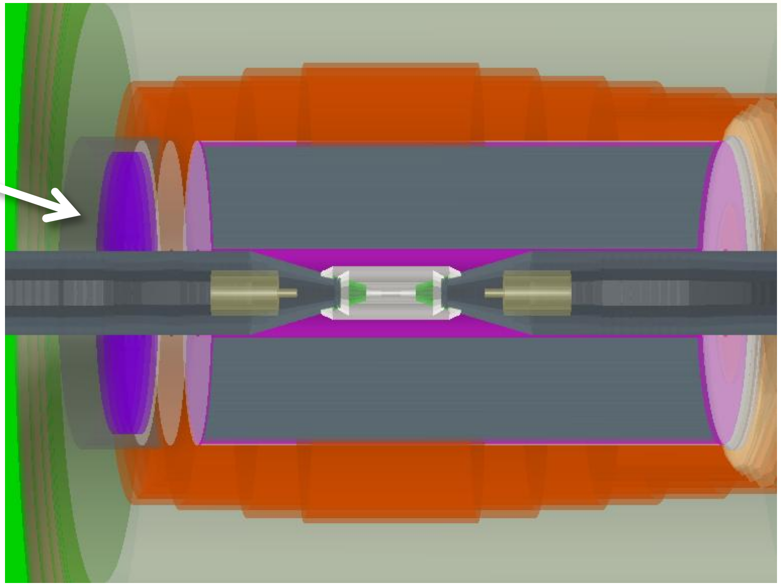
parameters defining the cells:
sensitive time window, spatial resolution, hit efficiency, cell width, stereo angle

```
<device name="Axial"
        type="DriftChamber"
         sensitiveTimeWindow="0.5e-6"
         rms_par0="0.0178977"
         rms_par1="0"
         rms_par2="-0.161932"
         rms_par3="0.357955"
         rms_par4="-0.238636"
         rms_par5="0.0409091"
         eff_par0="0.98"
         eff_par1="0.83"
         cell_size="1.8"
         angle="0"  />
```

Detector simulation

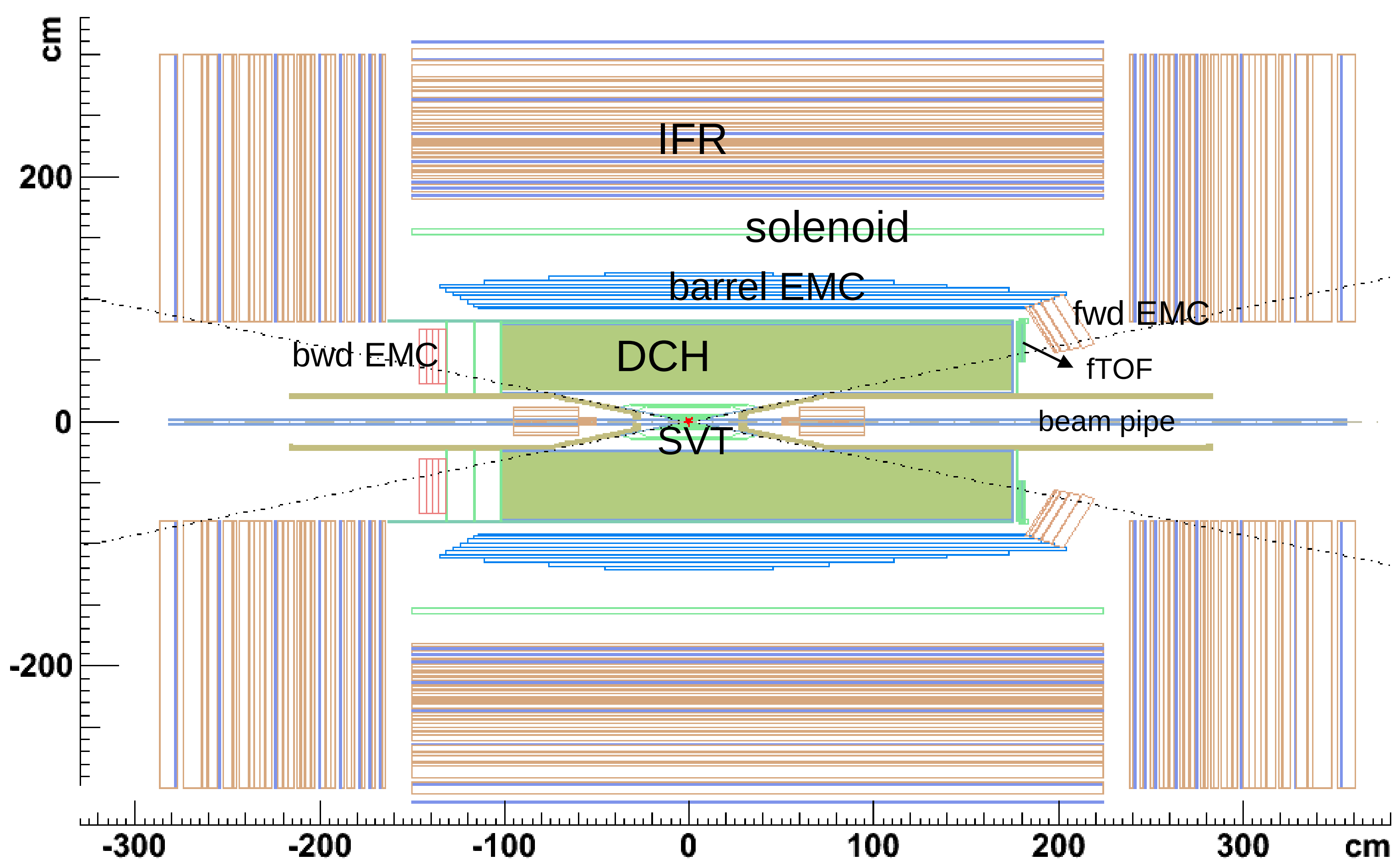


3-dim picture of PacDetector/pacrat_SuperB.xml produced using PacDisplay



Detector simulation

2-dim picture of
PacDetector/pacrat_SuperB.xml

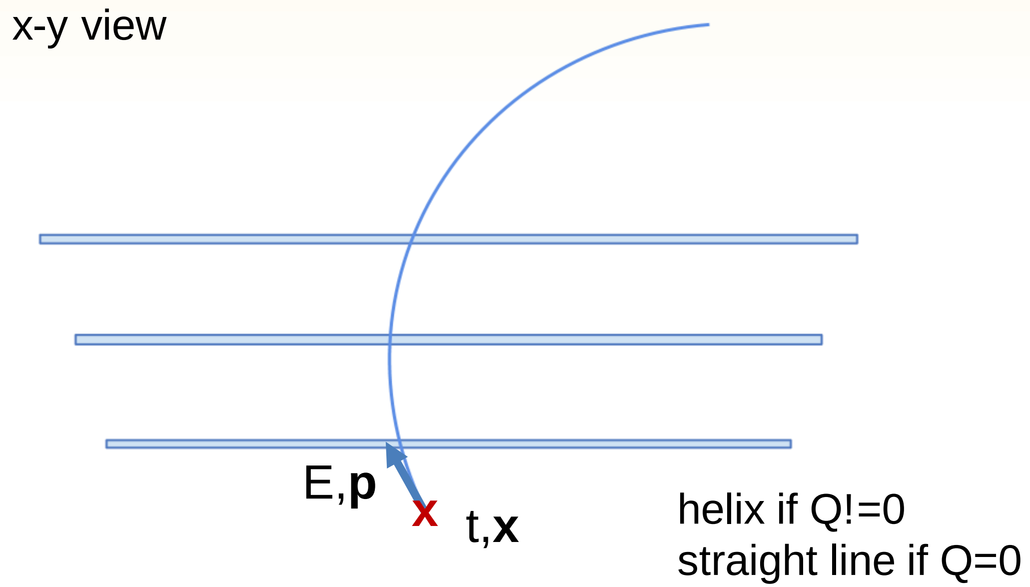


From the generated particle to the reconstructed particle step by step

Generated particle

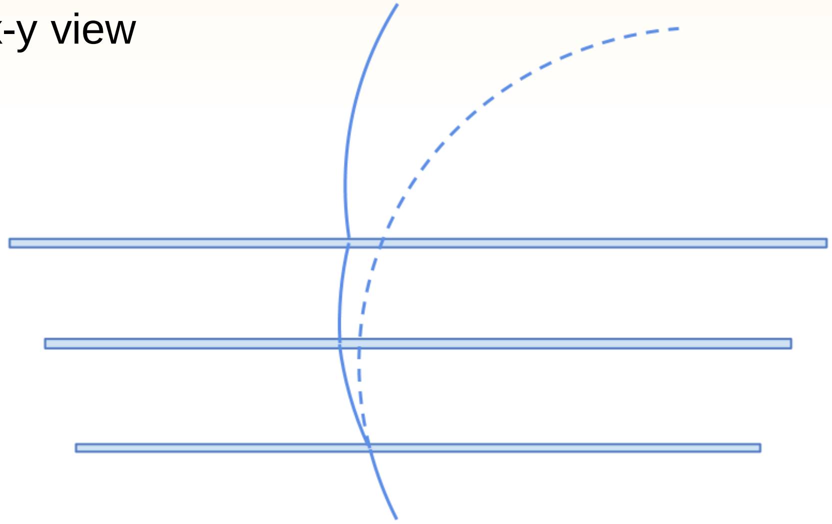
Generated particles are GTracks (or Tparticles)

A *GTrack* represents the particle at its production vertex:
- 4-momentum (E,**p**)
- vertex (t,**x**)
- charge Q



Simulated particle

x-y view

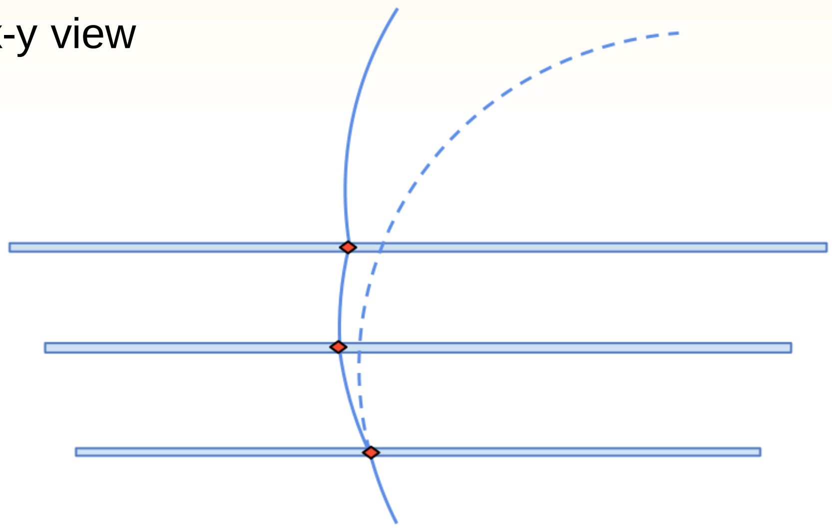


The generated particle is propagated through the detector elements and interacts with the material.

In general the interaction changes the particle momentum.
Thus the simulated trajectory is a set of helix pieces (Q!=0) or segment pieces (Q=0)

Simulated particle

x-y view



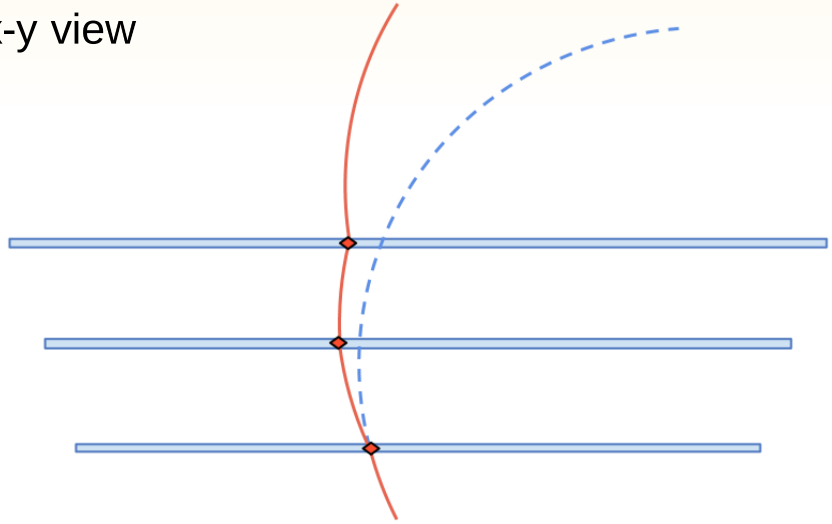
The intersection between the trajectory and a detector element is represented by a *PacSimHit* object

A PacSimHit has access to the relevant information related to the hit:

- (**x**,t) of the hit
- p at entrance and at exit
- type of interaction
- properties of the detector element
- etc

Simulated particle

x-y view

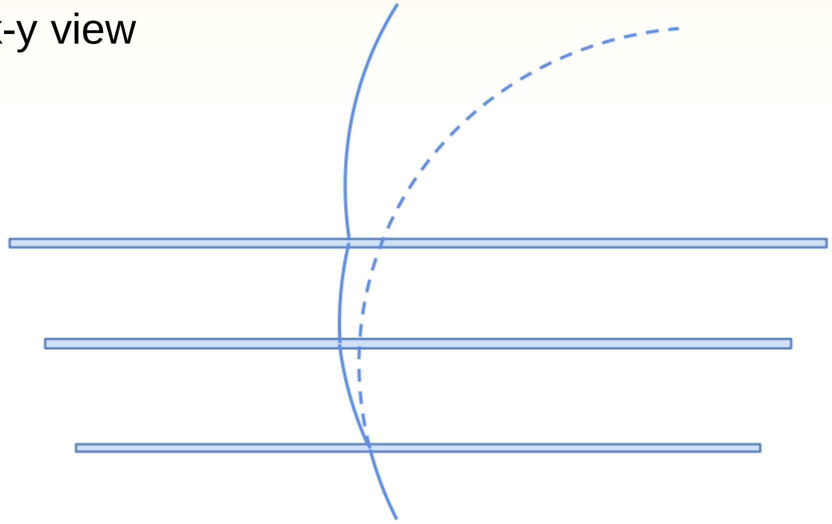


The set of PacSimHits and trajectories pieces of a particle is represented by a *PacSimTrack* object

A PacSimTrack thus represents the simulated particle. It has access to its
- PacSimHits
- trajectory pieces
- original GTrack

Simulated particle

x-y view

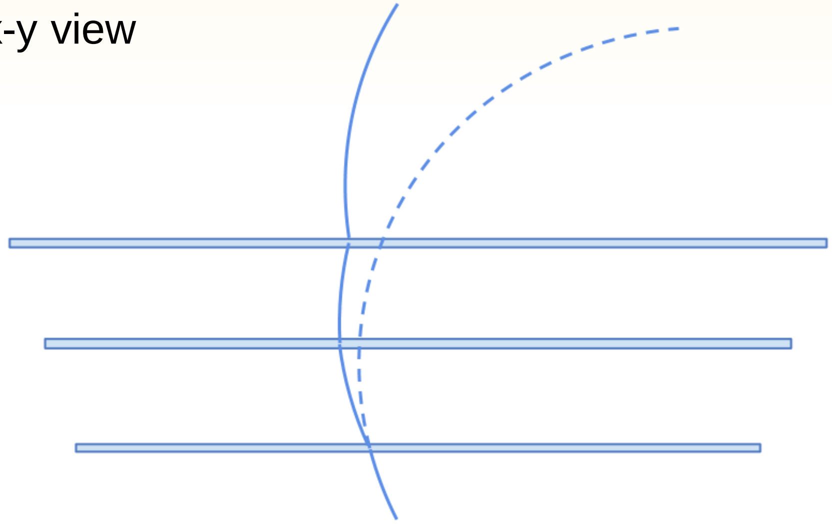


The following interaction processes are modeled:

- **brems**
photon radiation by electron
- **compt**
compton scattering, with electron emission - N/A yet -
- **normal**
adiabatic energy loss and scattering
- **interact**
hadronic (nuclear) interaction, with destruction of original particle and daughter creation
- **convert**
photon conversion into e+e- pair

Simulated particle

x-y view



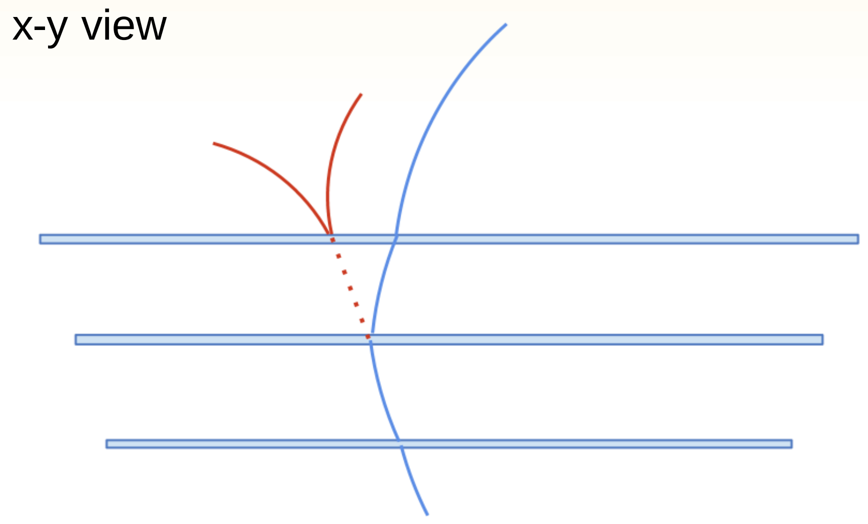
The following interaction processes are modeled

• **decay**
electroweak decay
• **shower**
EM cascade.
• **hadshower**
hadronic cascade

Simulated particle

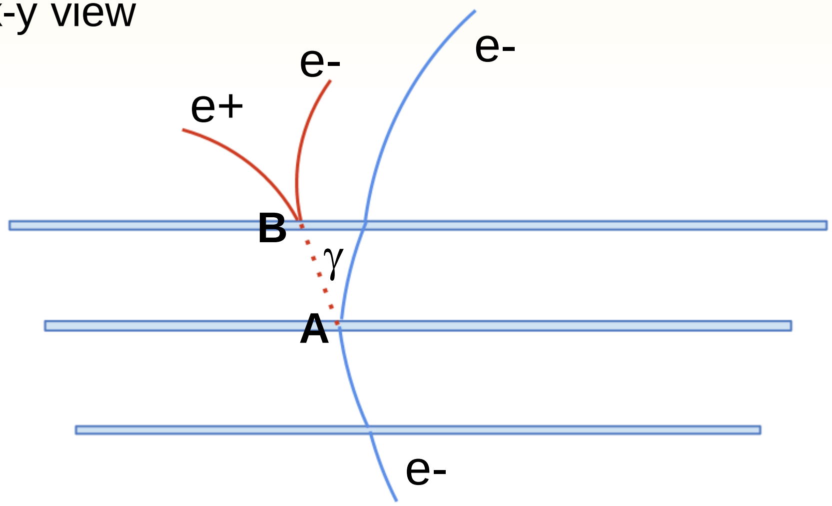
Depending on the interaction with the material layer a PacSimTrack will stop or continue, and one or more new particles may be created as GTracks.

The new GTracks are propagated through the space and simulated the same way as the original particle. (iterative procedure)



Simulated particle

x-y view



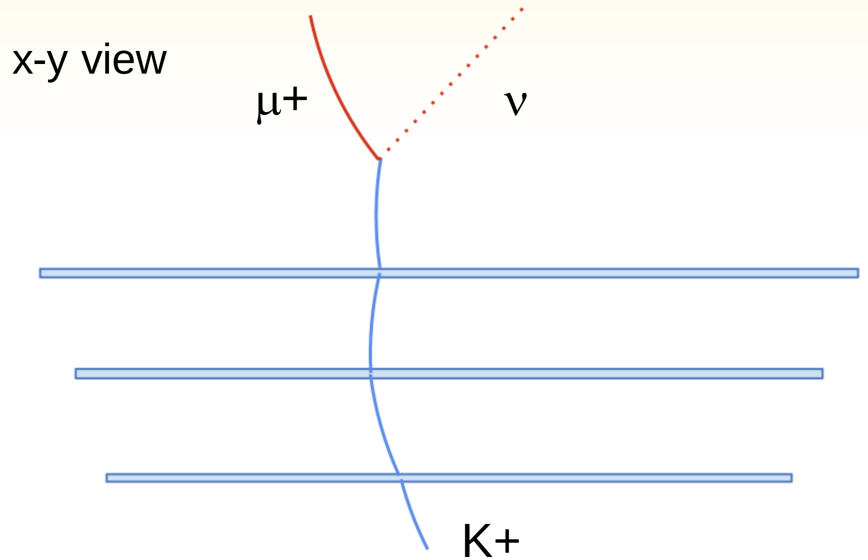
Example:

The electron emits a bremsstrahlung photon in **A** and continues with a new p.
The photon is generated in **A** and the corresponding GTrack and PacSimTrack are created.
The PacSimTrack stops in **B** following the conversion to e+e-.

Two GTracks for the e+e- pairs are created, and so on…

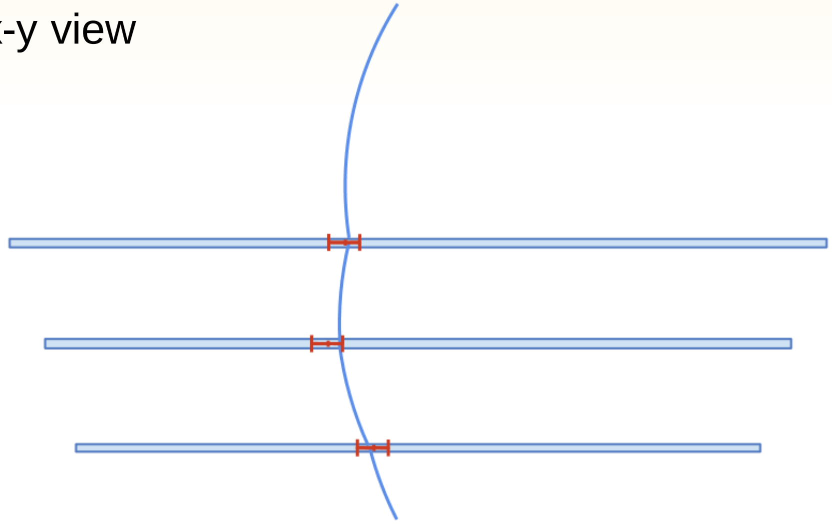
Simulated particle

Particles can decay.
The PacSimTrack stops and the GTracks of the particle daughters are created and propagated as usual.
(obviously, neutrinos are ignored…)



Reconstructed particle

x-y view



The simulation step (creation of the PacSimTracks) is followed by the **reconstruction**.

The first step is **tracking**.
Reco hits (*PacHitOnTrk*) are simulated for each PacSimHit associated to a tracker measurement layer.

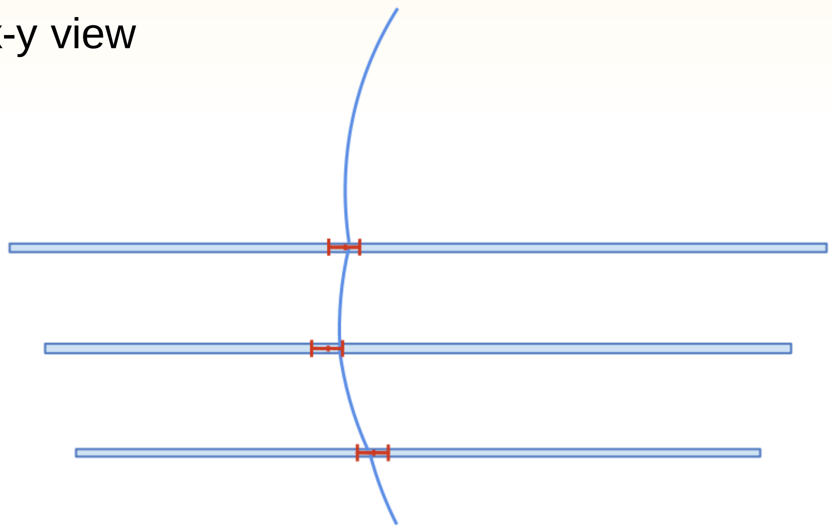
- Si striplets (SVT L0)
- MAPs (SVT L0)
- Si microstrips (SVT L1-5)
- wires axial/stereo (DCH)
- etc

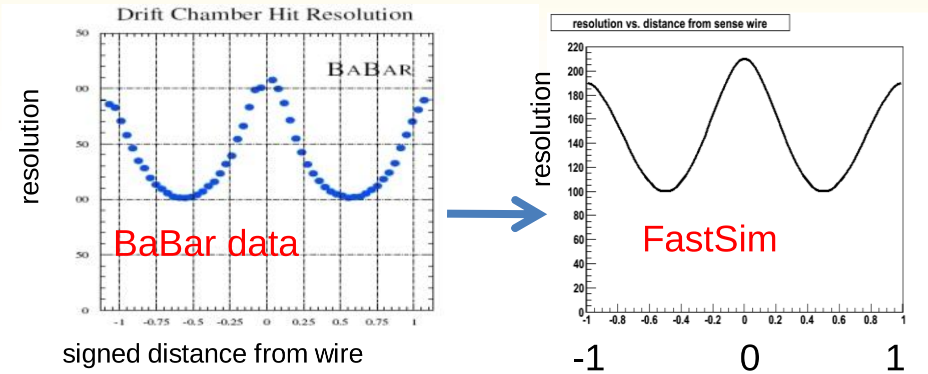
Reconstructed particle

The reco hit position is generated according to an efficiency and resolution function specific to each measurement layer and configurable via EDML.

Example: the DCH resolution function

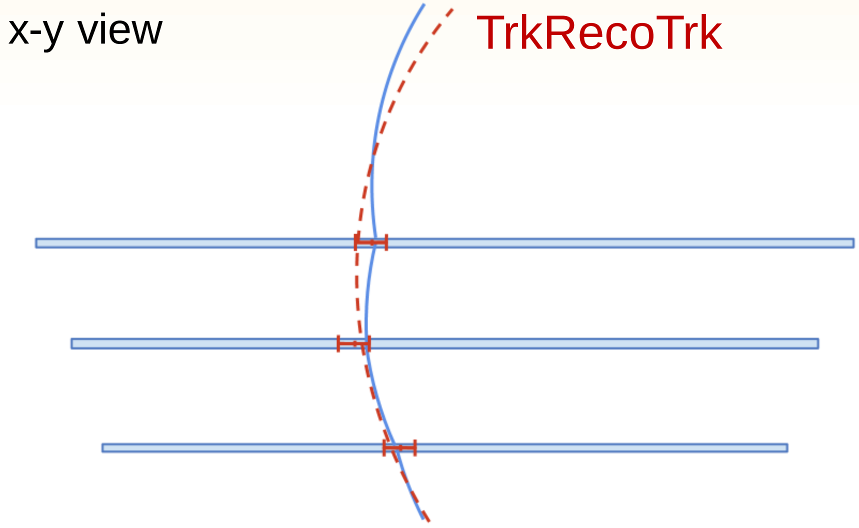
x-y view





The reco hit position w.r.t. the true value is determined from a Gaussian distribution whose width is determined from a random sample of the above function (tuned on Babar data).

Reconstructed particle



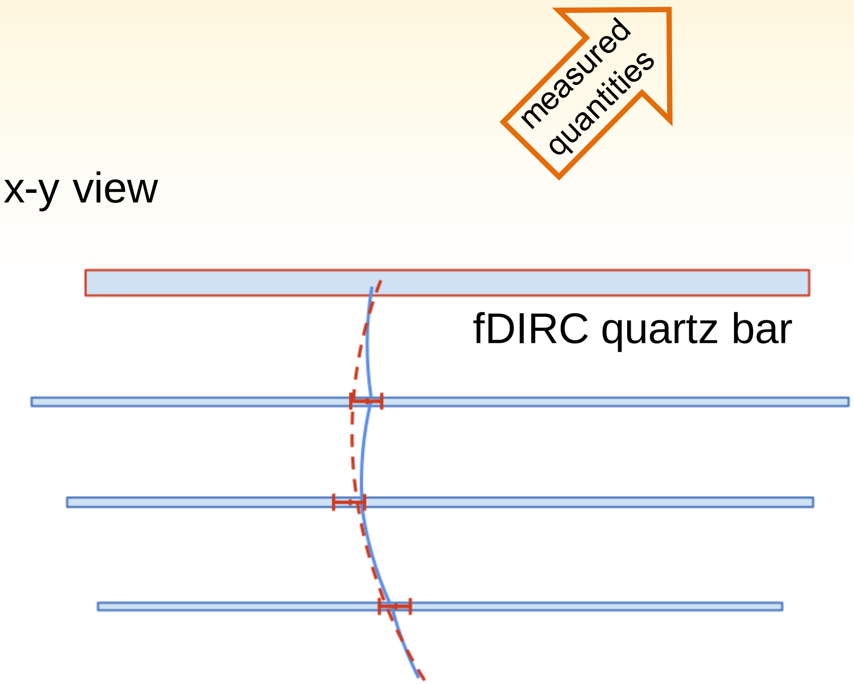
Once the track hits are simulated they are passed to the Babar Kalman filter track fit, which gives the measured track parameters.

The reconstructed track is represented by the *TrkRecoTrk* class.

Hit merging and pattern recognition confusion effects can be taken into account.

Reconstructed particle

Θ_C, err(Θ_C),
#γ(sig), #γ(bkg)



A PacSimHit associated to the measurement layer of a given detector triggers the corresponding signal reconstruction.

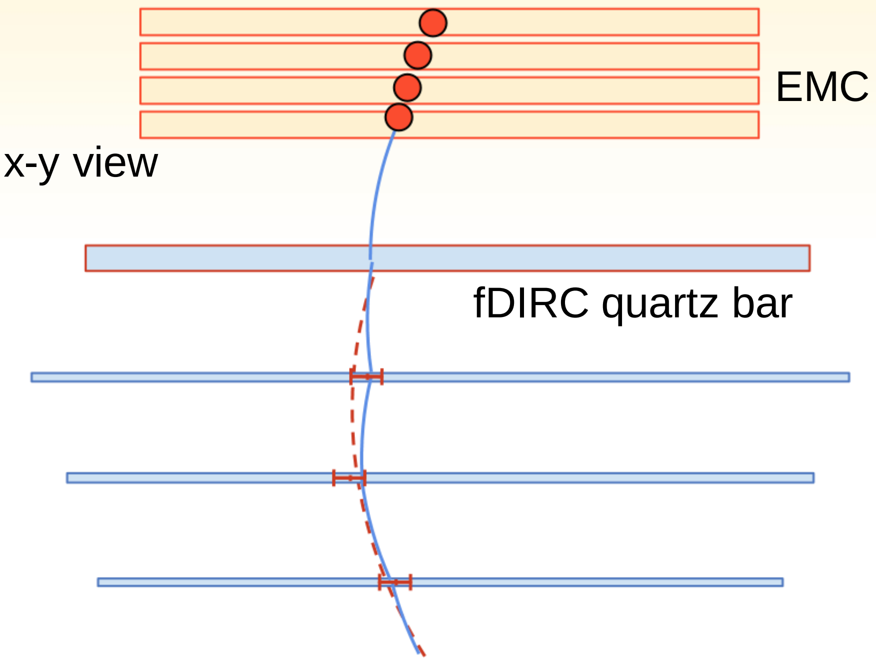
For example the simulation of the fDIRC includes the measurement of thetaC, σ(thetaC), #signal phot, #bkg photons

SuperB subsystems implemented in FastSim: **SVT, DCH, fDIRC, fTOF, barrel/fwd/bwd EMC, IFR**

Adding new ones is straighforward

Reconstructed particle

PacSimHits representing the EM shower



>1 discrete interaction within a material layer will start a shower (EM or had depending on the interaction).

The barrel EMC is simplified as set of cylindrical layers: crystals are not modeled individually.

Energy loss distributed according to the shower profile over a grid representing the crystal segmentation
Fluctuations included

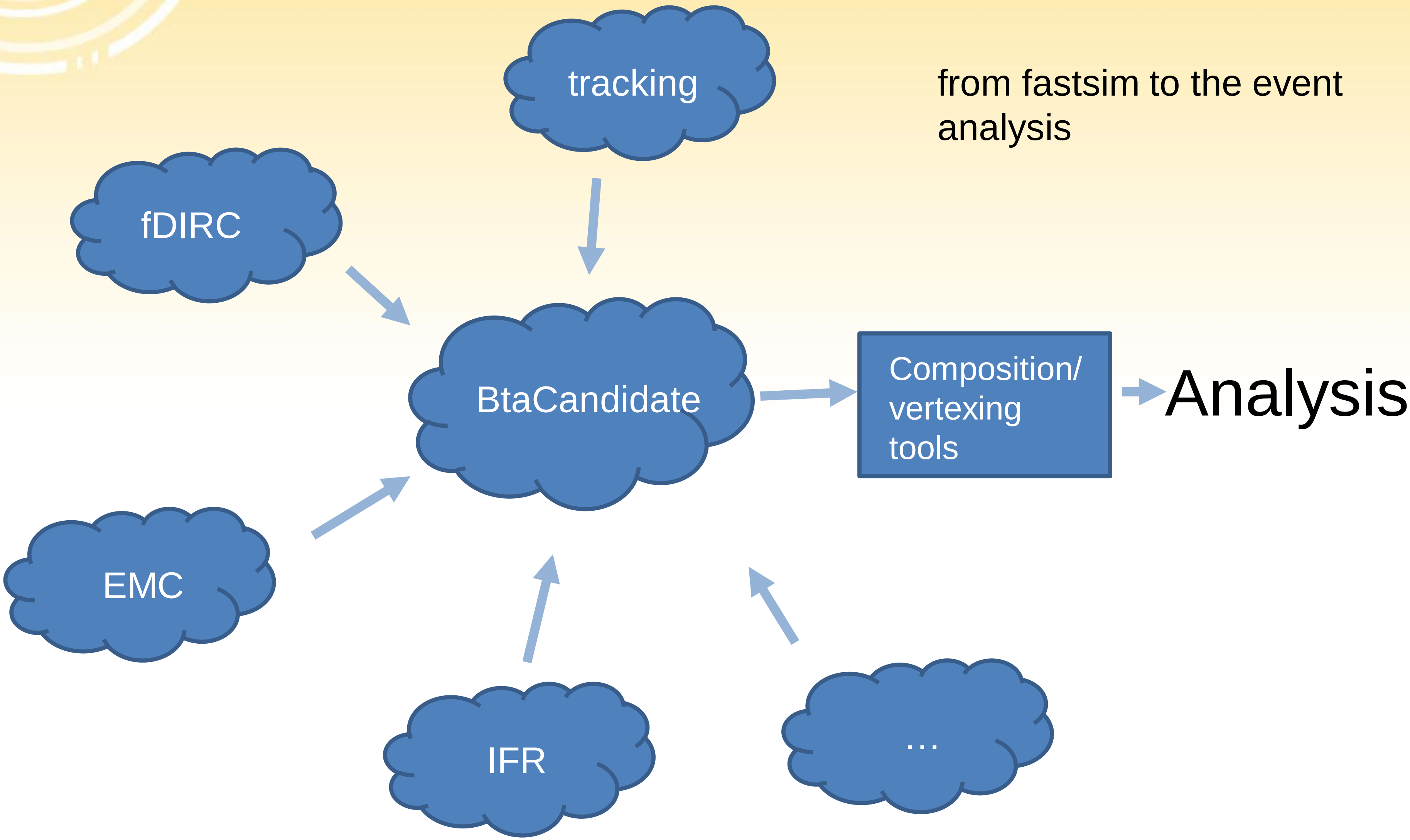
The EMC clusters reconstruction is not trivial

Reconstructed particle

PacMC/src/PmcReconstructionSequence.cc

```
//Simulate Reconstruction effects
forWhom->add(new PmcReconstruct("PmcReconstruct","Simulate Reconstruction Effects"));

//Split/merge Emc clusters
forWhom->add(new PacCaloSplitMerge("PacCaloSplitMerge","Split and Merge Emc Clusters"));

//EMC calibration
forWhom->add(new PacEmcCalibration("PacEmcCalibration","EMC energy calibration"));

//Build reco->gtrack map
forWhom->add(new PmcBuildCaloGTrkMap("PmcBuildCaloGTrkMap","Build RecoCalo-GTrack map"));

//Track intersection at the EMC
forWhom->add(new PacEmcMakeIntersections("PacEmcMakeIntersections","Make track intersections at the EMC"));

//Track-Calorimeter match
forWhom->add(new PacTrkClusterMatch("PacTrkClusterMatch","Track-Calorimeter match"));

//Split calorimeter list to neutral and charged
forWhom->add(new PacEmcListSplit("PacEmcListSplit","Split emc list to neutral/charged"));

// beta event initialisation module
forWhom->add( new BtaInitEvent("BtaInitEvent", "Beta- event initialisation") );

// Track filtering (anti-electron selector for now)
PacPidFilterTrackSequence( forWhom );
```

from fastsim to the event analysis



Analysis tools

- Set of tools inherited from Babar
  - Vertexing/composition tools, ntuple maker, …
- Others are being developed for SuperB
- A few examples in the practical tutorial
- SuperB documentation has holes
  - Important task. Volunteers needed.
  - Plan to use/adapt Babar documentation.

a personal special thank to David Brown

master of Babar code and core Fastsim

FastSim User Guide

http://mailman.fe.infn.it/superbwiki/index.php/SuperB_fast_simulation_User_Guide

Part II

Tutorial

http://mailman.fe.infn.it/superbwiki/index.php/FastSimDoc/Tutorial_London11