

# Software Tools for the 4th Concept

Corrado Gatto  
INFN Napoli/Lecce

# Outlook

- Ilcroot framework: a tool for the HEP community
- Tools in ILCroot
  - Digitization for Si based Detectors
  - Track reconstruction in Ilcroot
  - Multiple Readout Calorimetry
  - PID in ILCroot
  - Jet reconstruction

# 4th Concept Software Strategy

- Use of public domain common tools
- Adopt the ROOT framework
  - all needed functionalities present (from data taking to final plots)
  - reconstruction & analysis are naturally developing in the same framework
  - Extensive CERN support
  - Unprecedented Large contributing HEP Community
  - Open Source project
  - Multiplatforms
  - Support multi-threading and asynchronous I/O
  - Optimised for different access granularity (Raw data, DST's, NTuple analysis)
- *Impose* a single framework
  - Provide central support, documentation and distribution
  - Train users in the framework

# 4<sup>th</sup> Concept Software Framework: ILCroot

- **CERN** architecture (based on **Alice's Aliroot**)
- Full support provided by Brun, Carminati, Ferrari, et al.
- Uses **ROOT** as infrastructure
  - All ROOT tools are available (I/O, graphics, PROOF, data structure, etc)
  - Extremely large community of users/developers
- TGenerator for events generation
- Virtual Geometry Modeler (VGM) for geometry
- **Virtual Montecarlo for particle transport**
- Growing number of experiments have adopted it: Alice (LHC), Opera (LNGS), (Meg), CMB (GSI), Panda(GSI), 4th Concept, **LHeC** **and the forthcoming International Dual Readout Collaboration**
- Six MDC have proven robustness, reliability and portability



April 15, 2009

Do not Reinvent the wheel  
Concentrate on Detector studies and Physics

# General Architecture: Guidelines

- Ensure high level of modularity (for easy of maintenance and development)
  - Absence of code dependencies between different detector modules (to C++ header problems)
  - Design the structure of every detector package so that static parameters (i.e. geometry and detector response parameters) are stored in distinct objects
- The data structure to be built up as ROOT TTree-objects
  - Access either the full set of correlated data (i.e., the event) or only one or more sub-sample (one or more detectors).

# The Virtual Montecarlo Concept

- Virtual MC provides a virtual interface to Monte Carlo
- It allows to run the same user application with all supported Monte Carlo programs
- The concrete Monte Carlo (Geant3, Geant4, Fluka) is selected and loaded at run time
  - Compare Montecarlo performance and possible flows
  - Choose the optimal Montecarlo for the study



April 19th, 2009

Perfect Tool for Designing/Optimizing new Detectors

# ILCroot Biggest Strenght: Modularity

- ILCroot Building Block: The Detector Class
- Detector-centric approach (vs Processor-centric)



- Easy to work for groups across many countries
- Allow for several versions of the same detector or several detector of the same kind (ex. TPC & DCH)

# Modularity in ILCroot

Two objects types: local and global modules

- **Local modules (aka detector modules)**

1. Local modules implement detector specific tasks
2. Each module executes a list of detector actions/tasks
3. Detector modules do not talk to each other (i.e. no cross-modules calls)
4. On demand data is forbidden
5. Local modules post their data to persistent objects (ex. Sdigits, Digits, Clusters, etc)

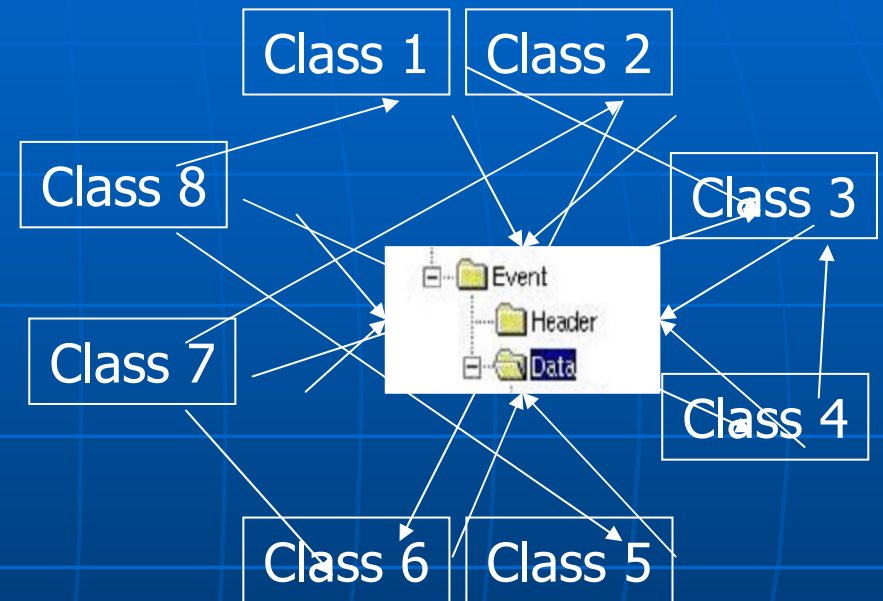
- **Global modules (global reconstruction, global PID, etc)**

1. One or more Global objects execute a list of actions involving objects from several detectors
2. They build informations from persistent objects infos filled by local modules
3. Tasks are still implements in the local modules
4. Detector modules do not talk to each other
5. Data exchange is implemented via posting through a whiteboard

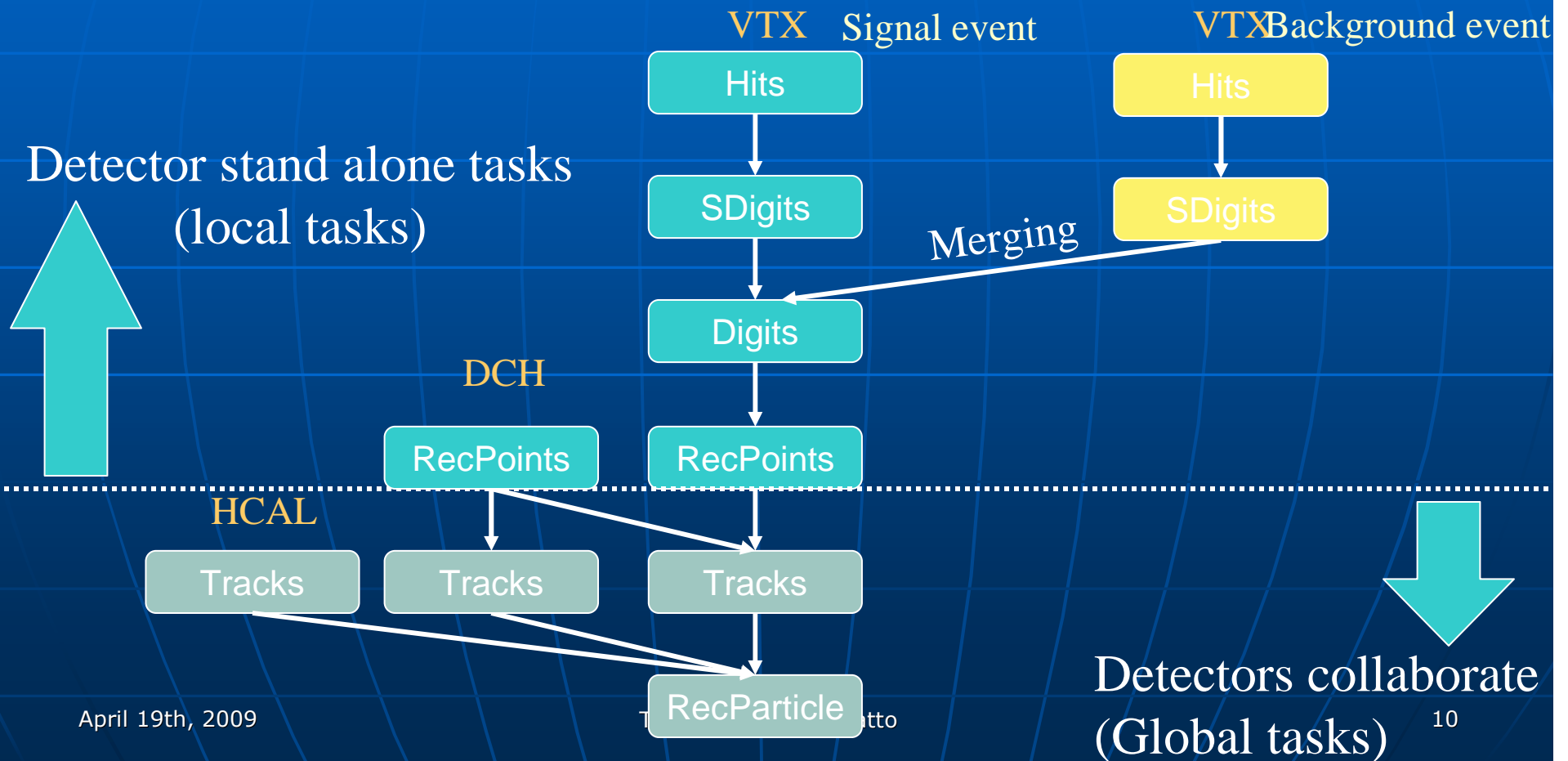


# Run-time Data-Exchange

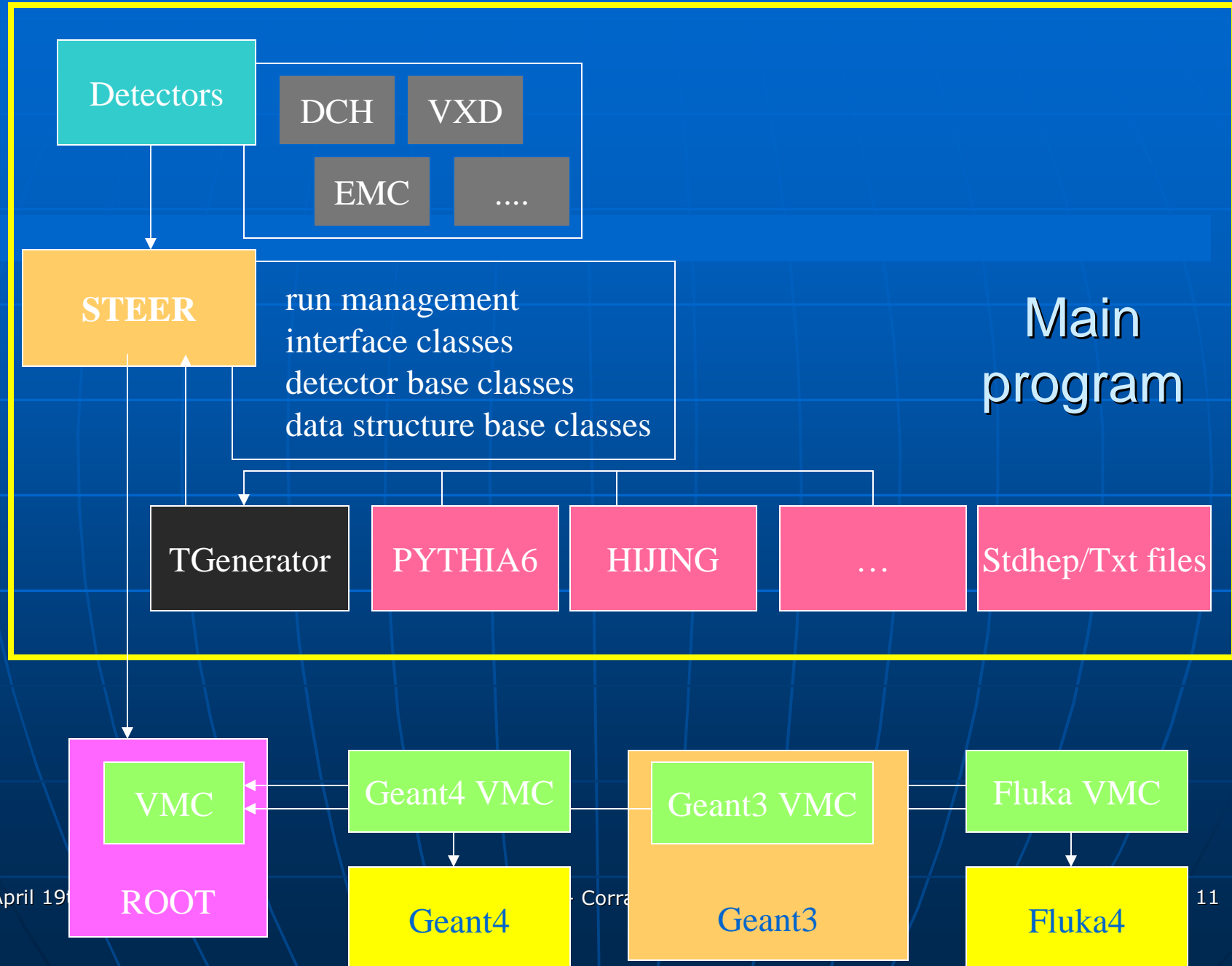
- Post transient and persistent data to a white board
- Structure the whiteboard according to detector sub-structure & tasks results
- Each detector is responsible for posting its data
- Tasks access data from the white board
- Detectors cooperate through the white board



# Processing Flow



# General Architecture

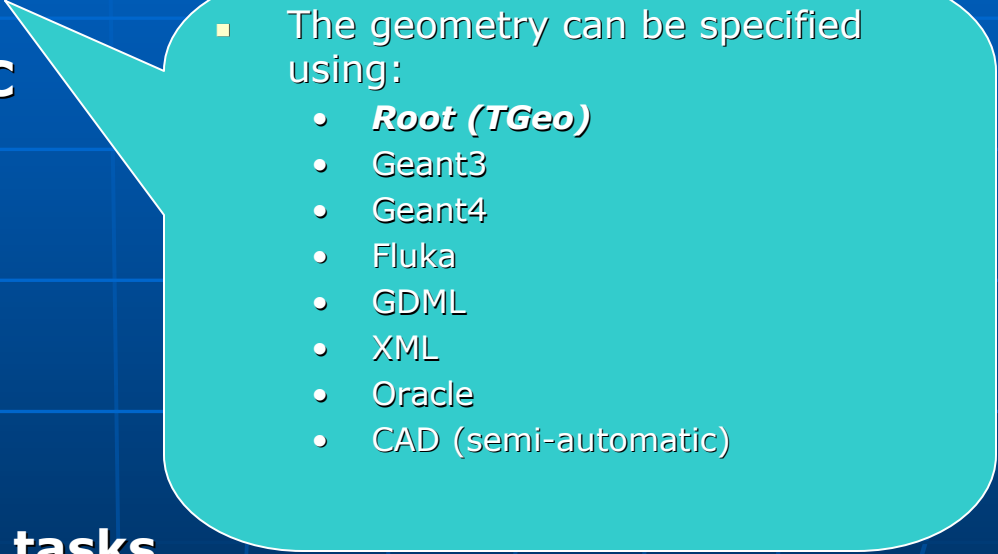


# Data Structure: LCIO vs MONARC

**This slide has been removed since it contains outdated informations not reflecting the current status of the Data structure**

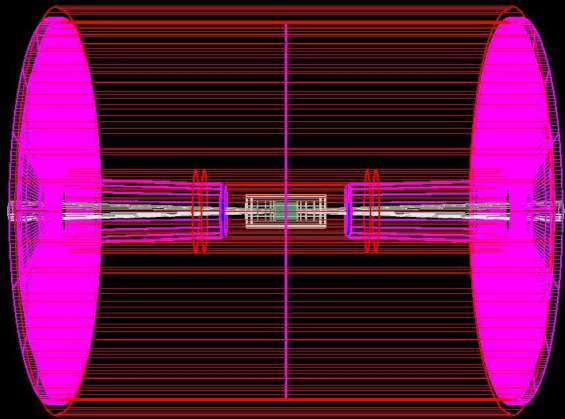
# The Detector Class

- Both sensitive modules (detectors) and non-sensitive ones are described by this base class.
- This class must support:
  - Geometry description
  - Event display
  - Simulation by the MC
  - Digitization
  - Pattern recognition
  - Local reconstruction
  - Local PiD
  - Calibration
  - QA
  - Data from the above tasks
- Several versions of the same detector are possible (choose at run time)

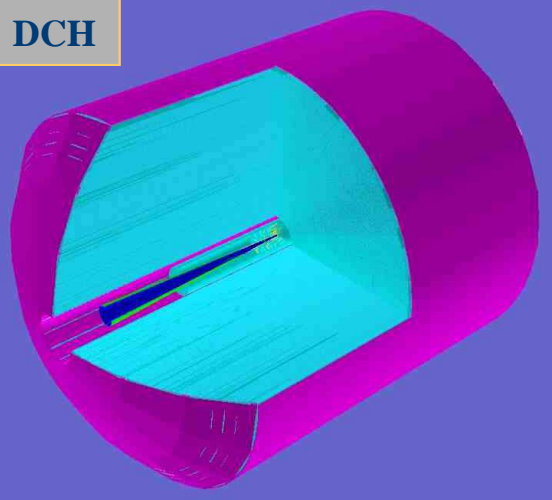
- 
- The geometry can be specified using:
    - *Root (TGeo)*
    - Geant3
    - Geant4
    - Fluka
    - GDML
    - XML
    - Oracle
    - CAD (semi-automatic)

# Detectors Galore

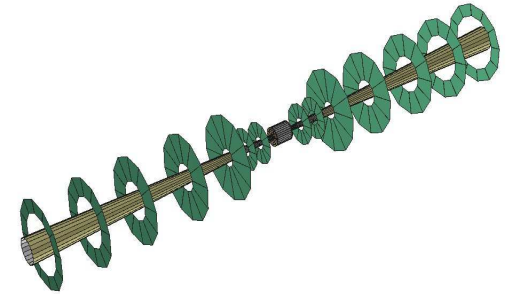
TPC



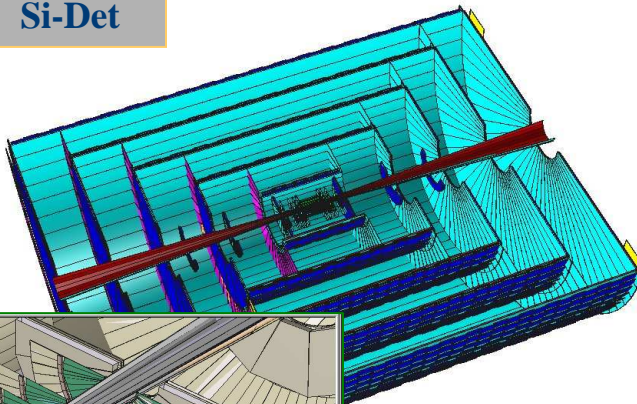
DCH



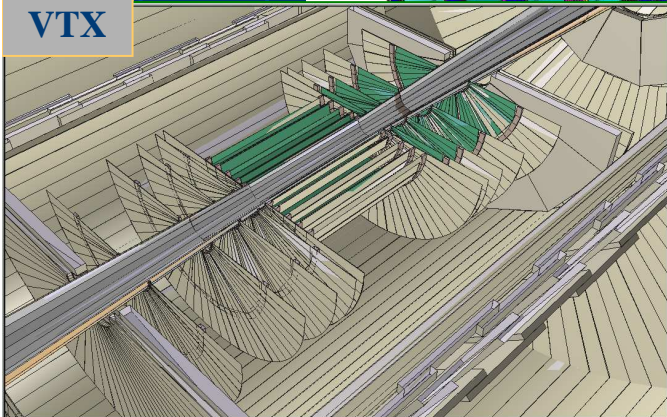
FTD



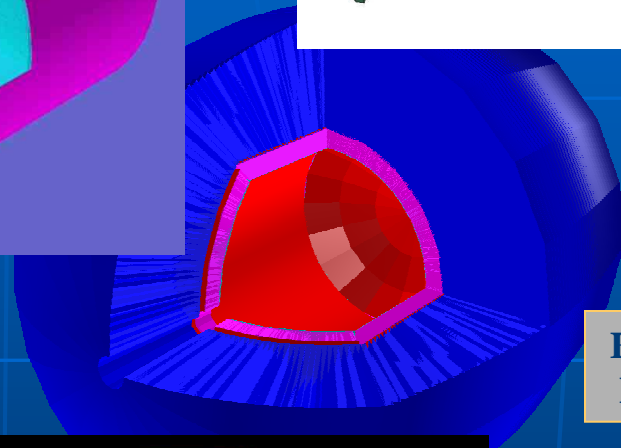
Si-Det



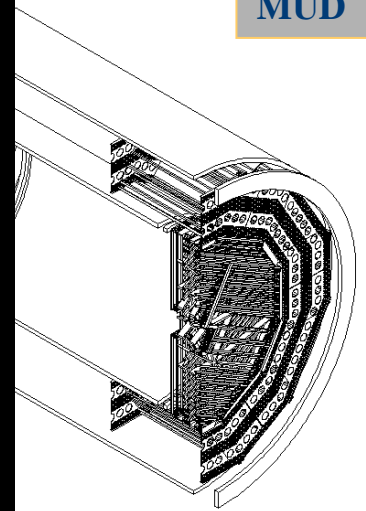
VTX



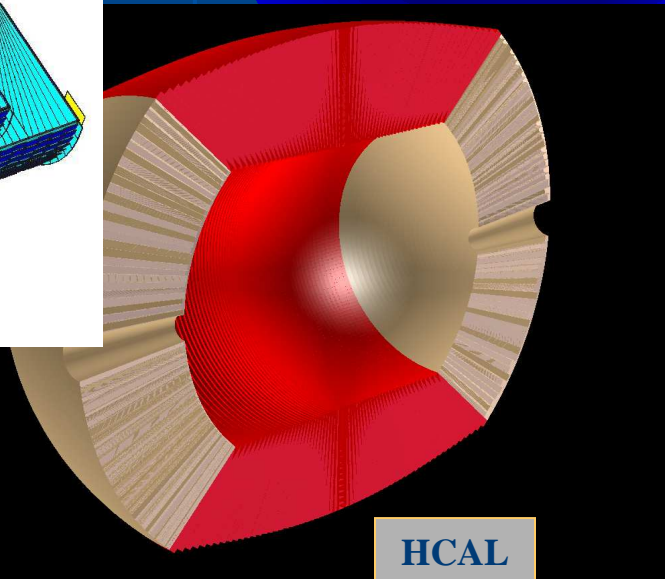
ECAL+  
HCAL



MUD



HCAL

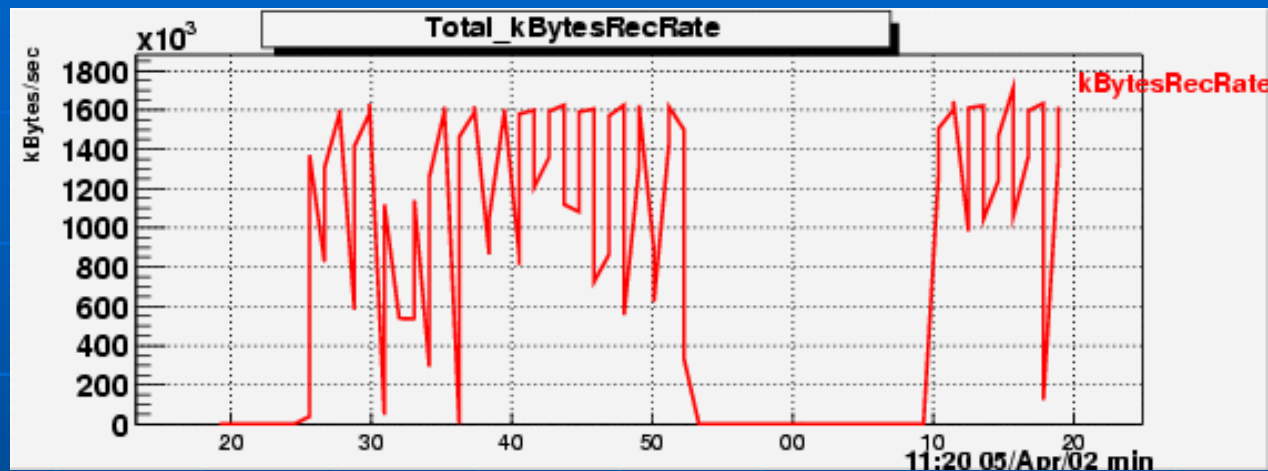


TI

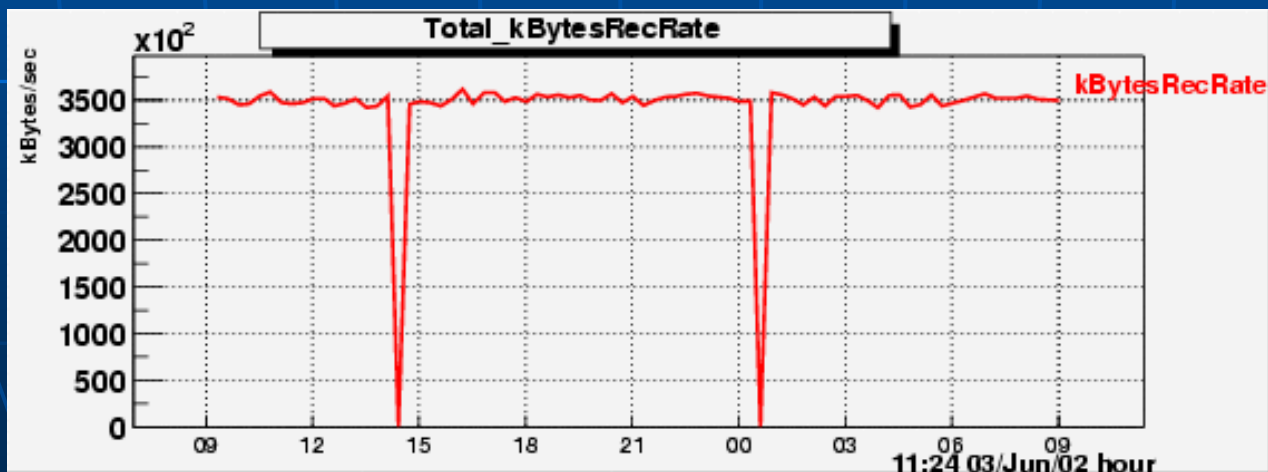


# Performance (Alice's IV MDC)

Data generation in LDC, event building, no data recording

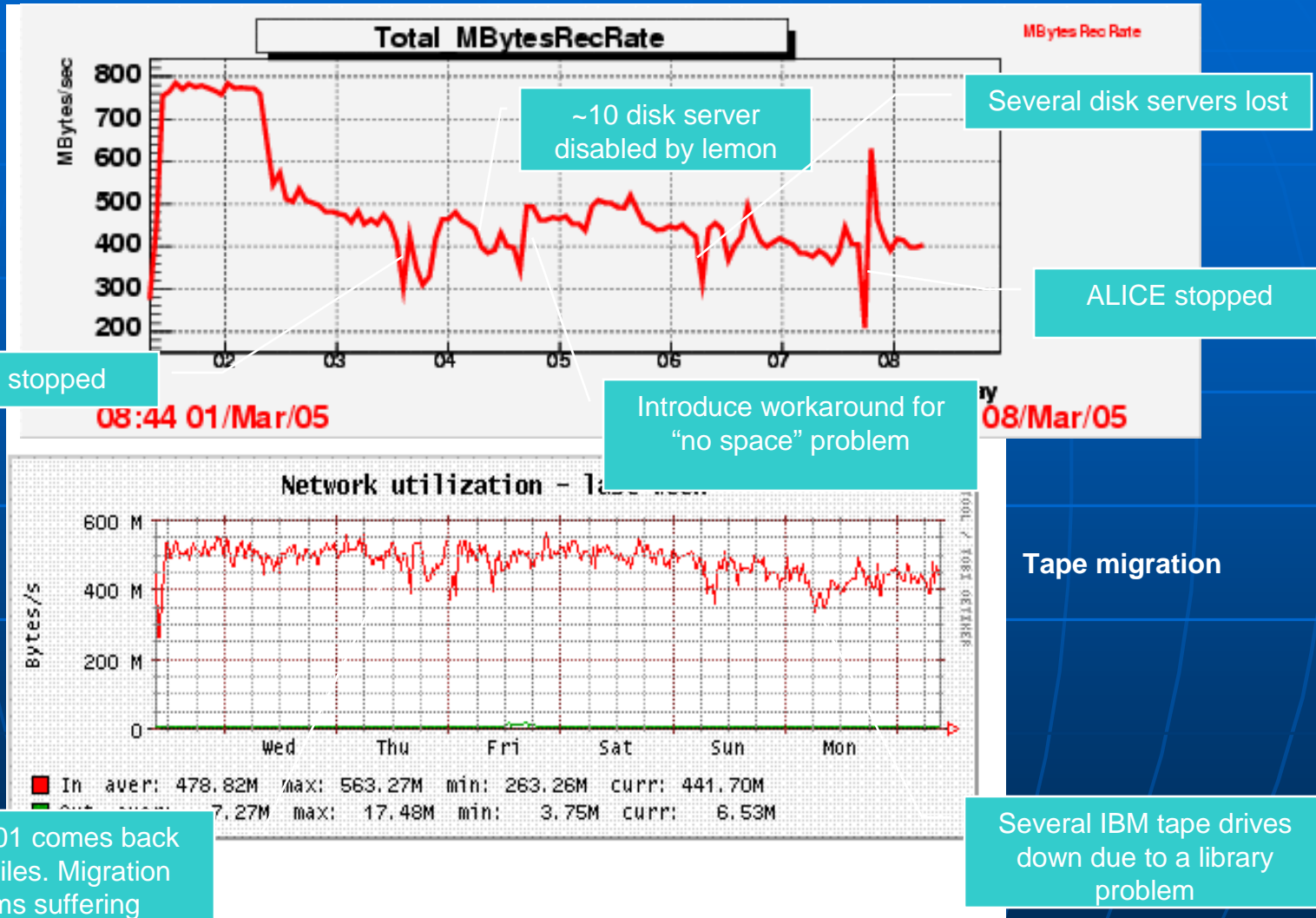


Data generation in LDC, event building, data recording to disk



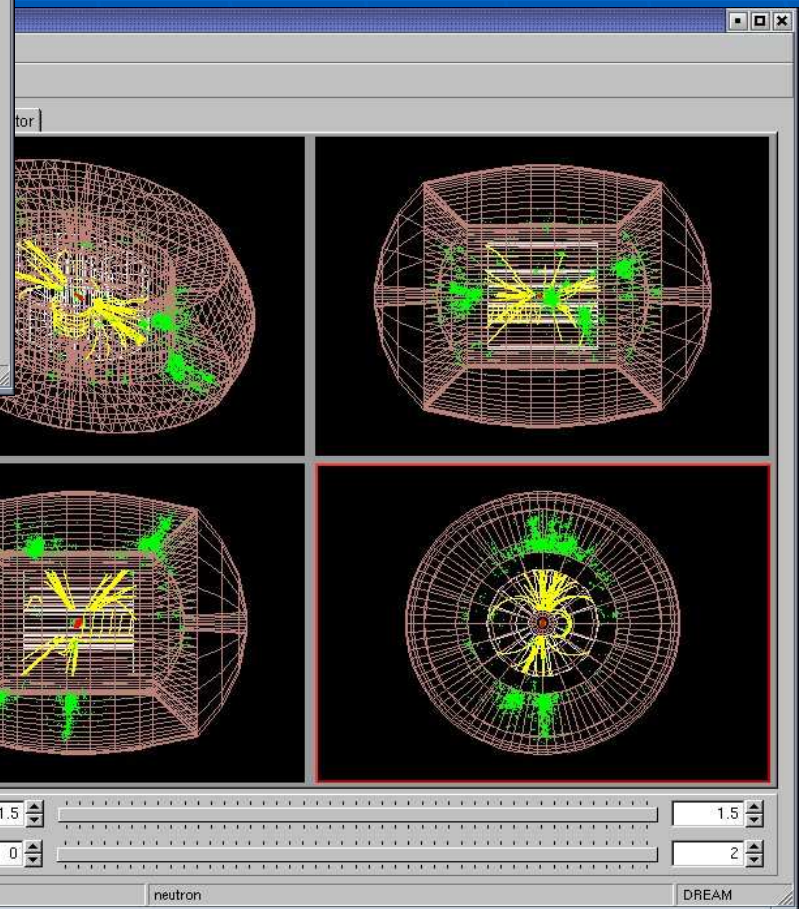
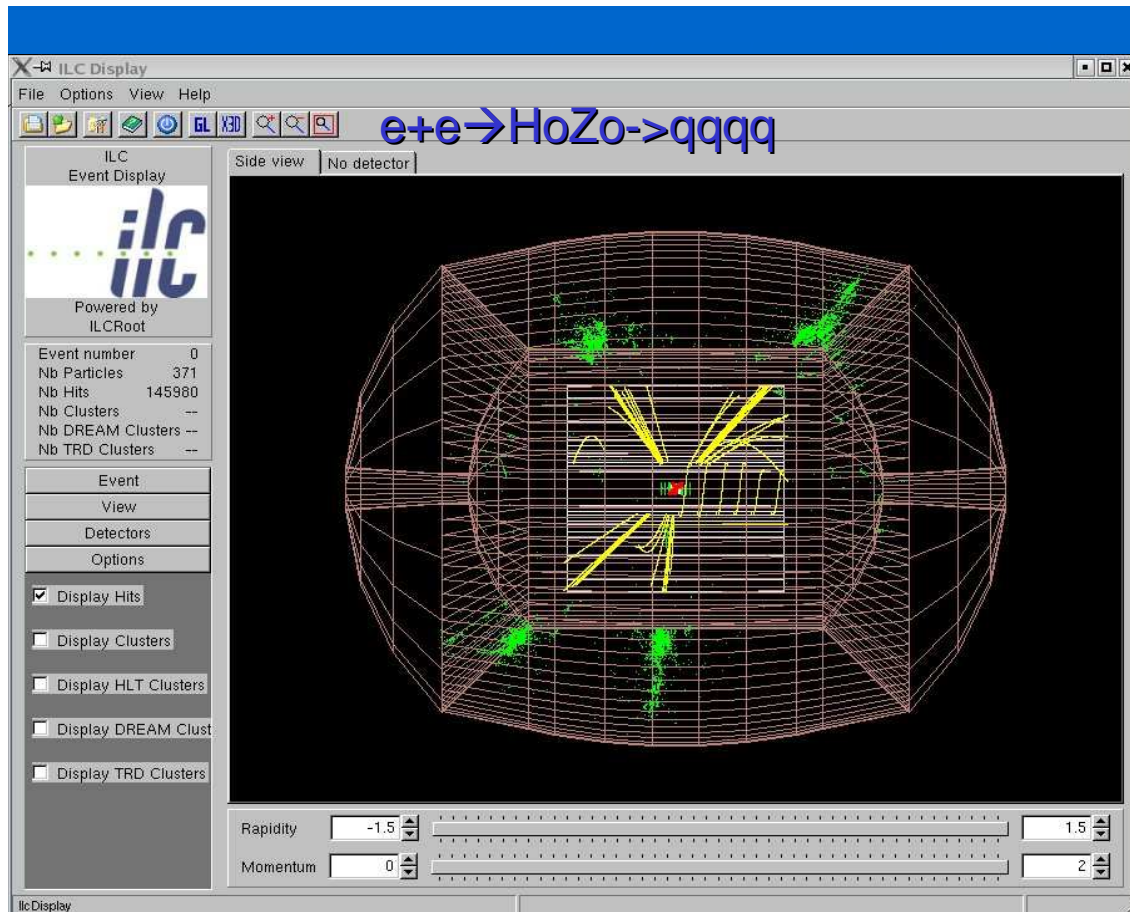
# Performance (Alice's VI MDC)

ALICE DAQ





# ILCroot Event Display



April 19th, 2009

# **Digitization and Clusterization of Si Detectors in Ilcroot**

# Technologies Implemented

- 3 detector species:
  - Silicon pixels (for VXD and FTD)
  - Silicon Strips (for Si Tracker and FTD)
  - Silicon Drift (not used at ILC)
- Pixel can have non constant size
- Strips can be stereo and on both sides
- Dead Regions are taken into account

# SDigitization in Pixel Detector

- Get the Segmentation Model for each detector ( from IlcVXDSegmentationSPD class)
- Get Calibration parameters ( from IlcVXDCalibrationSPD class)
- Load background hits from file (if any)
- Loop over the hits and create a segment in Si in 3D
  - Step (from MC) along the line  $>1 \mu\text{m}$  increments
    - Convert GeV to charge and get bias voltage:

$$el = dE*dt/3.6e-9 \quad dV = \text{thick}/\text{bias voltage}$$

- Compute charge spreading:

$$\sigma_x = \sqrt{2k/e * T^o * dV * L} , \quad \sigma_z = fda * \sigma_{xy}$$

- Spread charge across pixels using  $\text{Erfc}(x, z, \sigma_x, \sigma_{xy})$
- Charge pile-up is automatically taken into account

# SDigitization in Pixels (cont'd)

- Add couplig effect between nearby pixels row-wise and column-swise (constant probability)
- Remove dead pixels (use signal map)

# Digitization in Pixels

- Load SDigits from several files (signal or background)
- Merge signals belonging to the same channel
  - Non-linearity effects
  - Threshold
  - Saturation
- Add electronic noise
- Save Digits over threshold

# Clusterization in Pixel Detector

- Create a initial cluster from adjacent pixels (no for diagonal)
- Subdivide the previous cluster in smaller  $N \times N$  clusters
- Reconstruct Recpoints ant error matrix from coordinate average of the cluster
- Kalman filter picks up the best Recpoints



# The Parameters for the VXD

- Size Pixel X = 20  $\mu\text{m}$
- Size Pixel Z = 20  $\mu\text{m}$
- Eccentricity = 0.85 (fda)
- Bias voltage = 18 V volts
- cr = 0% (coupling probability for row)
- cc = 4.7% (coupling probability for column)
- threshold = 3000 Electrons
- electronics noise = 0
- $T^\circ = 300^\circ\text{K}$



# SDigitization in Strips Detector

- Get the Segmentation Model for each detector (from IlcVXDSegmentationSSD class)
- Get Calibration parameters (from IlcVXDCalibrationSSD class)
- Load background hits from file (if any)
- Loop on the hits and create a segment in Si in 3D
  - Step along the line in equal size increments
    - Compute Drift time to p-side and n-side:  
$$\text{tdrift}[0] = (y + (\text{seg} \rightarrow \text{Dy}() * 1.0\text{E-}4) / 2) / \text{GetDriftVelocity}(0);$$
$$\text{tdrift}[1] = ((\text{seg} \rightarrow \text{Dy}() * 1.0\text{E-}4) / 2 - y) / \text{GetDriftVelocity}(1);$$
    - Compute diffusion constant:  
$$\text{sigma}[k] = \text{TMath}::\text{Sqrt}(2 * \text{GetDiffConst}(k) * \text{tdrift}[k]);$$
    - integrate the diffusion gaussian from  $-3\sigma$  to  $3\sigma$
  - Charge pile-up is automatically taken into account

# SDigitization in Strips (cont'd)

- Add electronic noise per each side separately
  - `// noise is gaussian`
  - `noise = (Double_t) gRandom->Gaus(0,res->GetNoiseP().At(ix));`
  - 
  - `// need to calibrate noise`  
`noise *= (Double_t) res->GetGainP(ix);`
  - 
  - `// noise comes in ADC channels from the calibration database`
  - `// It needs to be converted back to electronVolts`
  - `noise /= res->GetDEvToADC(1.);`
- Add coupling effect between nearby strips
  - different contribution from left and right neighbours
  - Proportional to nearby signals
- Remove dead pixels (use signal map)
- Convert total charge into signal (ADC count)
  - `if(k==0) signal /= res->GetGainP(ix);`
  - `else signal /= res->GetGainN(ix);`
  - 
  - `// signal is converted in unit of ADC`
  - `signal = res->GetDEvToADC(fMapA2->GetSignal(k,ix));`

# Clusterization in Strip Detector

- Create a initial cluster from adjacent strips (no for diagonal)
- Separate into Overlapped Clusters
  - Look for through in the analog signal shape
  - Split signal of parent clusters among daughter clusters
- Intersect stereo strips to get Recpoints from CoG of signals (and error matrix)
- Kalman filter picks up the best Recpoints

# The Parameters for the Strips

- Strip size (p, n)
- Stereo angle (p-> 7.5 mrad, n->25.5 mrad)
- Ionization Energy in Si =  $3.62\text{E}-09$
- Hole diffusion constant (=  $11\text{ cm}^2/\text{sec}$ )
- Electron diffusion constant (=  $30\text{ cm}^2/\text{sec}$ )
- $v_{\text{drift}}^{\text{P}} (=0.86\text{E}+06\text{ cm/sec})$  ,  $v_{\text{drift}}^{\text{N}} (=2.28\text{E}+06\text{ cm/sec})$
- Calibration constants
  - Gain
  - ADC conversion (1 ADC unit = 2.16 KeV)
- Coupling probabilities between strips (p and n)
- $\sigma$  of gaussian noise (p AND n)
- threshold

# Track Fitting in ILCARoot

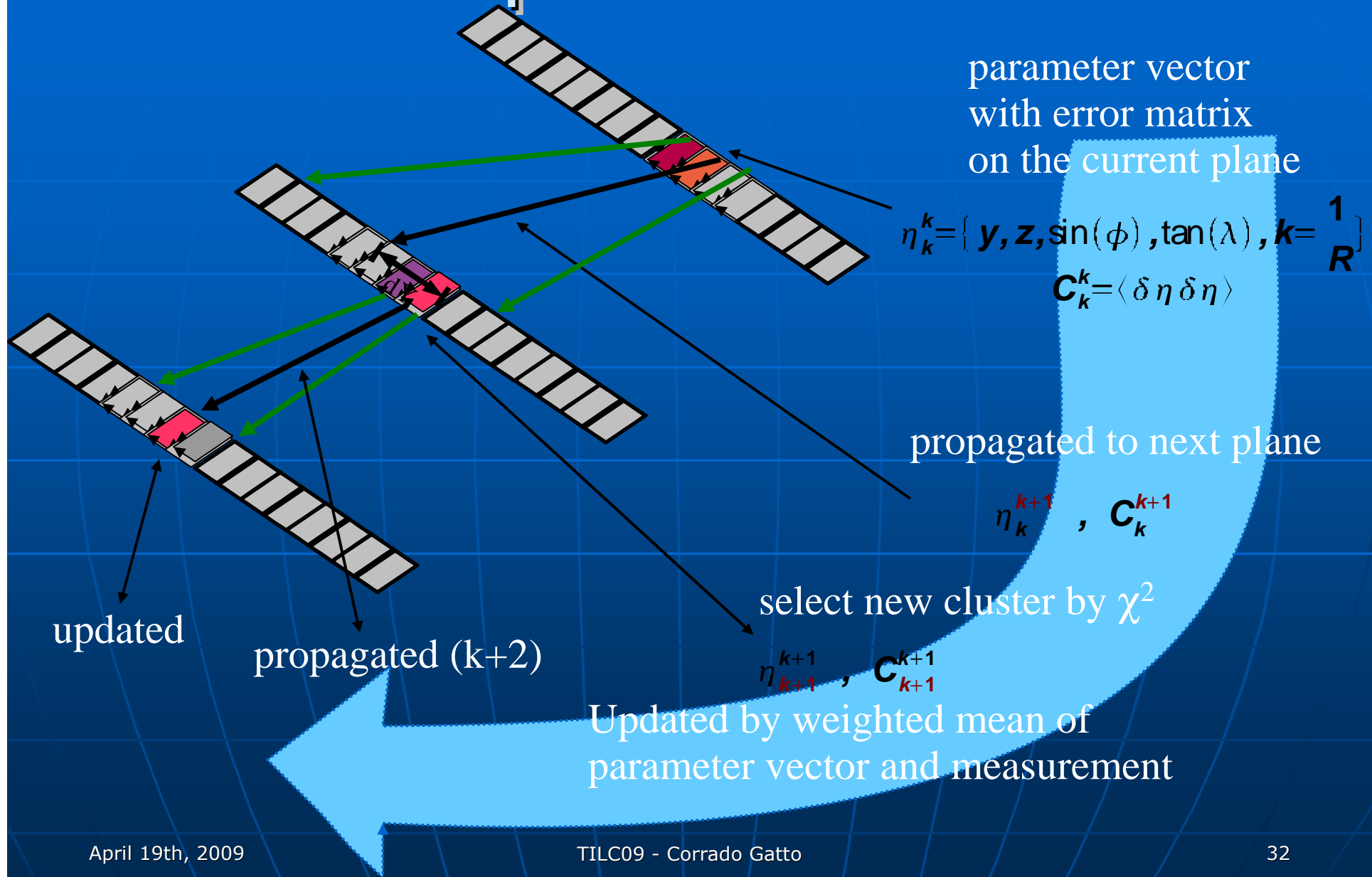
- Track finding and fitting is a global tasks: individual detector collaborate
- It is performed after each detector has completed its local tasks (simulation, digitization, clusterization, etc.)
- It occurs in three phases:
  - Seeding in CT and fitting in VXD+CT+MUD
  - Standalone seeding and fitting in VXD
  - Standalone seeding and fitting in MUD

Not  
implemented

# Kalman Filter (classic)

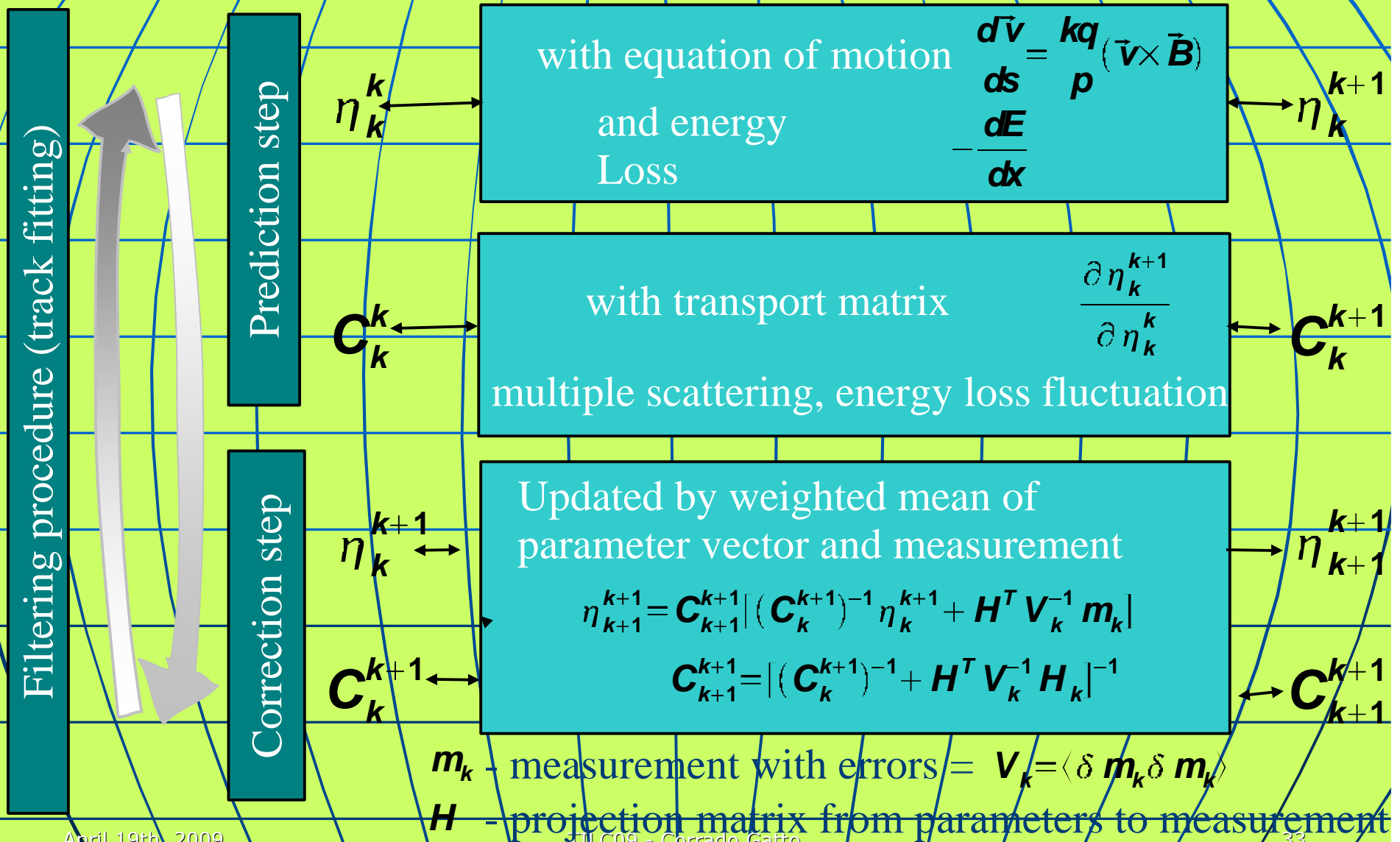
- Recursive least-squares estimation.
- Equivalent to global least-squares method including all correlations between measurements due to multiple scattering.
- Suitable for combined track finding and fitting
- Provides a natural way:
  - to take into account multiple scattering, magnetic field unhomogeneity
  - possibility to take into account mean energy losses
  - to extrapolate tracks from one sub-detector to another

# Basic Principle of Kalman Filter





# Basic Principle of Kalman Filter

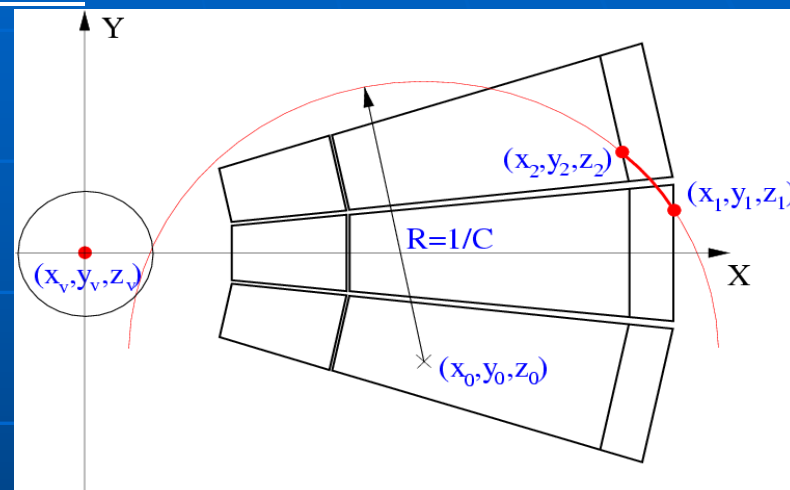


# Seeding

Track Efficiency limited by efficiency of seeding!

## Primary Seeding with vertex constrain

- Take 2 rows with gap 20 rows
- Check quality of track segment:
  - $\chi^2$
  - number of found clusters
  - number of shared clusters



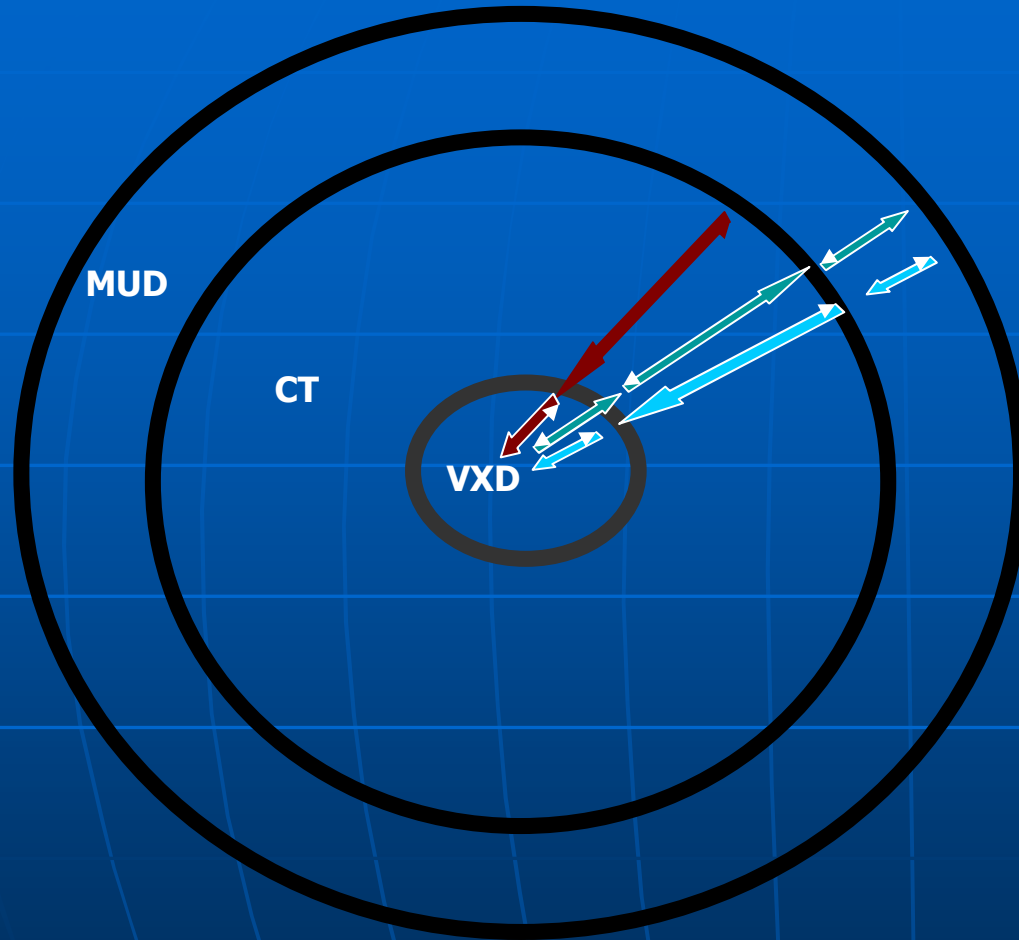
## Secondary Seeding without vertex constrain

- Simple track follower
- Algorithm
  - Seeding between 3 pad-rows (with gaps 2 rows)
  - Check that nearest clusters available at prolongation
  - Find prolongation to inner radius to make 20 rows segment
  - Check quality of track segment

# Parallel Tracking

- seedings with constraint + seedings without constraint at different radii (necessary for kinks) from outer to inner
- Tracking
  - Find for each track the prolongation to the next pad-row
  - Estimate the errors
  - Update track according current cluster parameters
  - (Possible refine clusters parameters with current track)
- Track several track-hypothesis in parallel
  - Allow cluster sharing between different track
- Remove-Overlap
- **Kinks and V0** fitted during the Kalman filtering

# Tracking strategy – Primary tracks



- Iterative process
  - Forward propagation towards to the vertex –CT-VXD
  - Back propagation –VXD-TPC-MUD
  - Refit inward MUD-CT-VXD
- Continuous seeding –track segment finding in all detectors

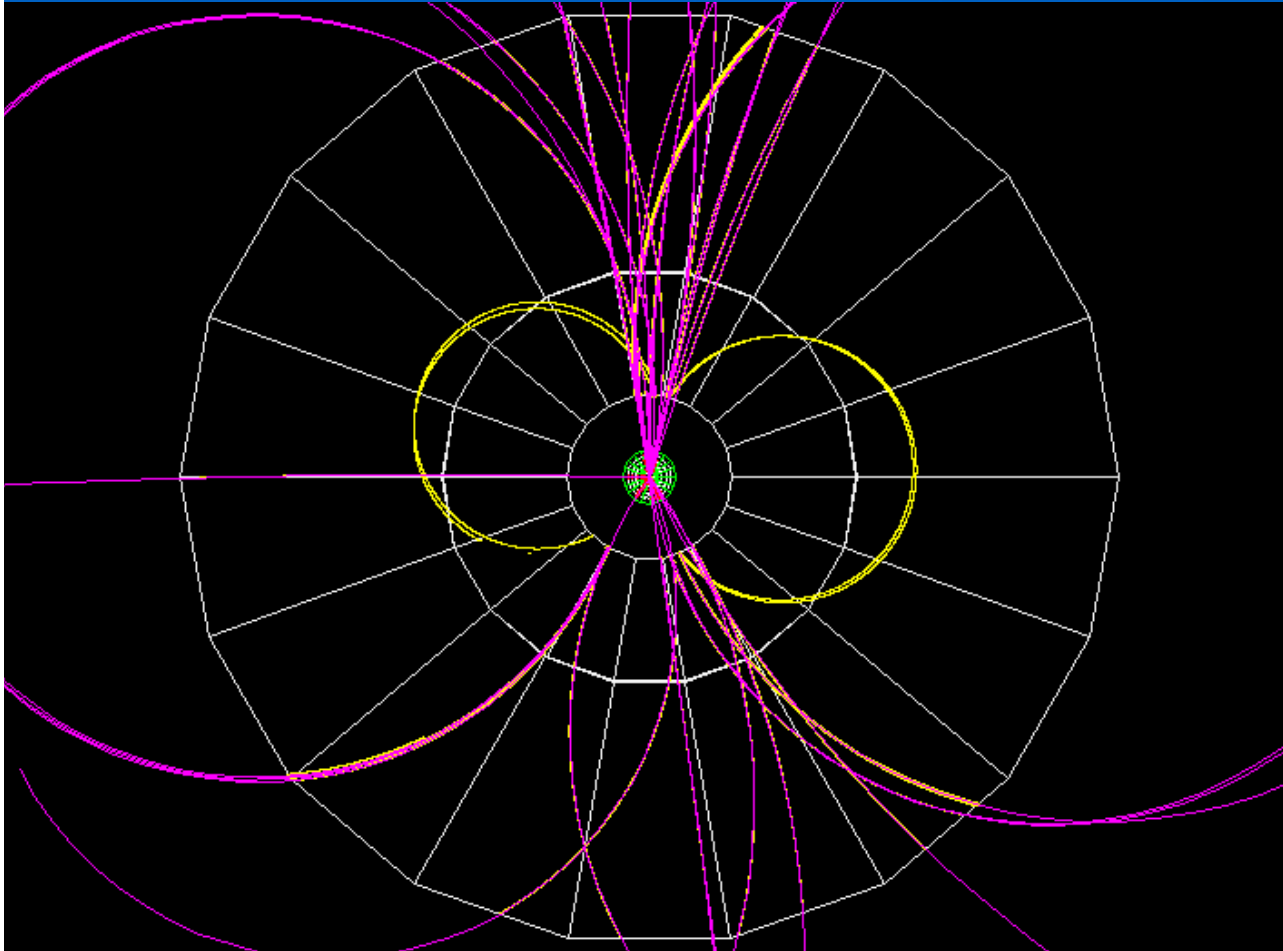
*currently is implemented CT+VXD+MUD propagation*

# VXD Standalone Tracker

- Uses Clusters leftover by Parallel Kalman Filter
- **Requires at least 4 hits to build a track**
- Seeding in VXD in two steps
  - Step 1: look for 3 RecPoints in a narrow row or 2 + the beampoint.
  - Step 2: prolongate to next layers each helix constructed from a seed.
- After finding clusters, all different combination of clusters are refitted with the Kalman Filter and the tracks with lowest  $\chi^2$  are selected.
- Finally, the process is repeated attempting to find tracks on an enlarged road constructed looping on the first point on different layers and all the subsequent layers.
- In 3.5 Tesla B-field ->  $P_t > 20$  MeV
- **Also used in VTX+Si Central Tracker simulations**

# Reconstructed Tracks

$$e^+ e^- \rightarrow Z H \rightarrow \mu^+ \mu^- H$$



yellow - hits

purple -reconstructed  
tracks

# The 4<sup>th</sup> Concept Dual/Triple Readout Calorimeter

# Dual Readout Calorimetry

Total calorimeter energy: use two measured signals and two, energy-independent, calibration constants

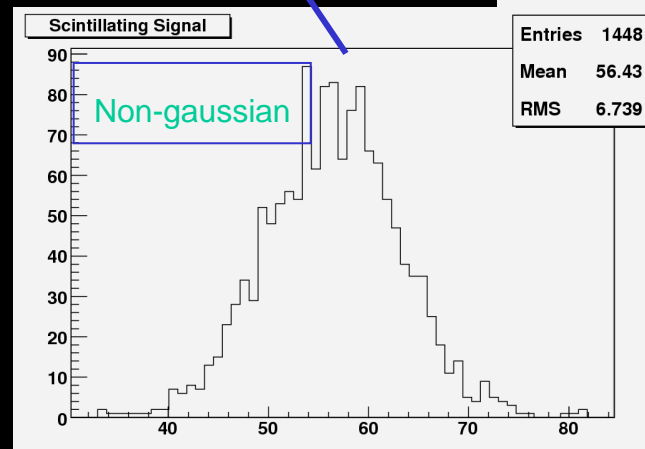
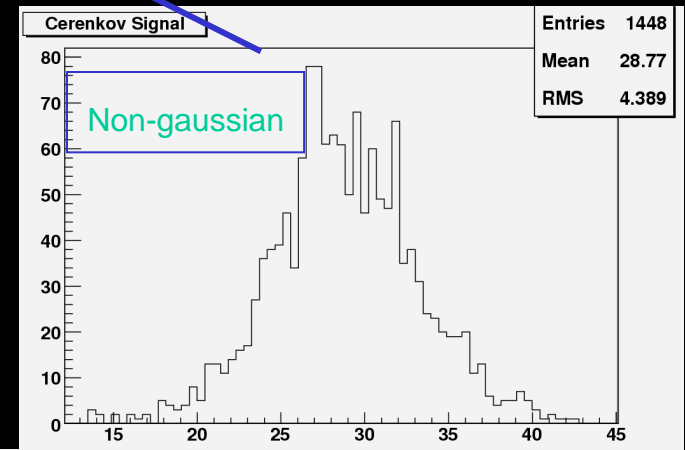
$$E_{HCAL} = \frac{\eta_s \cdot E_s \cdot (\eta_c - 1) - \eta_c \cdot E_c \cdot (\eta_s - 1)}{\eta_c - \eta_s}$$



# Dual Readout Calorimetry

Total calorimeter energy: use two measured signals and two, energy-independent, calibration constants

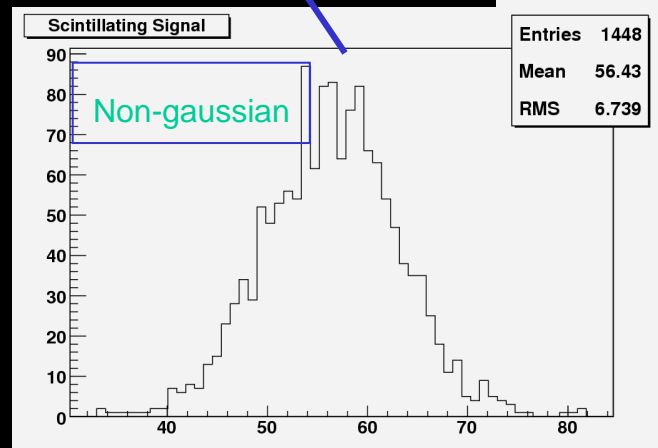
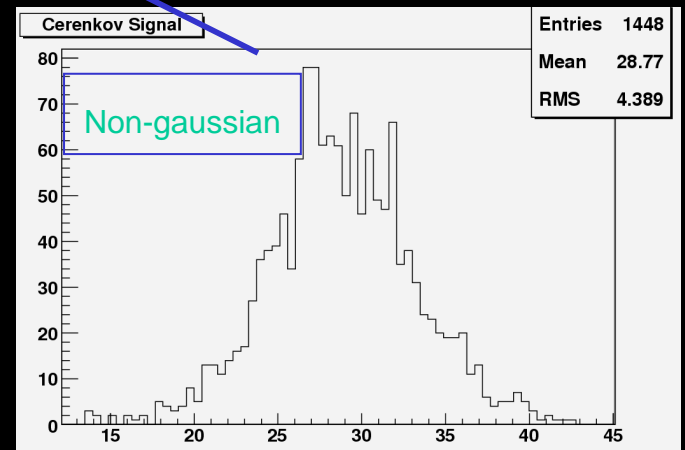
$$E_{HCAL} = \frac{\eta_s \cdot E_s \cdot (\eta_c - 1) - \eta_c \cdot E_c \cdot (\eta_s - 1)}{\eta_c - \eta_s}$$



# Dual Readout Calorimetry

Total calorimeter energy: use two measured signals and two, energy-independent, calibration constants

$$E_{HCAL} = \frac{\eta_s \cdot E_s \cdot (\eta_c - 1) - \eta_c \cdot E_c \cdot (\eta_s - 1)}{\eta_c - \eta_s}$$



$$\eta_c = \left( \frac{e}{h} \right)_c \quad \eta_s = \left( \frac{e}{h} \right)_s$$

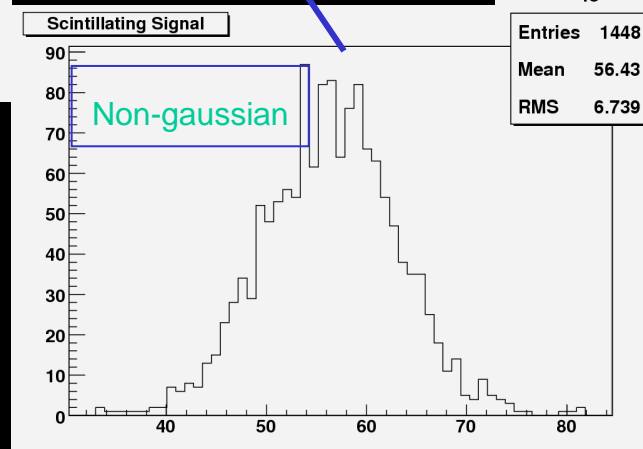
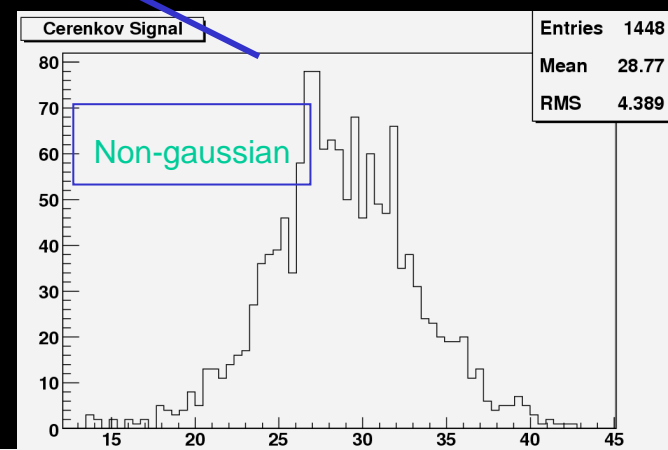
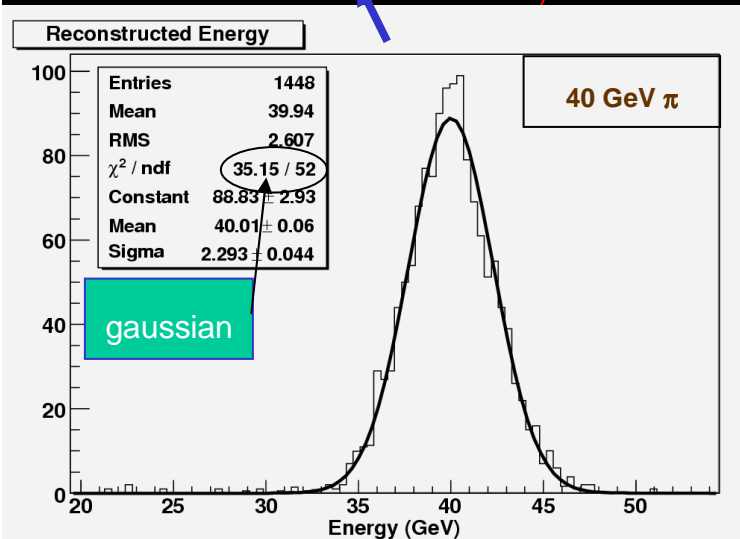
From calibration  
@ 1 Energy only

April 19th, 2009

# Dual Readout Calorimetry

Total calorimeter energy: use two measured signals and two, energy-independent, calibration constants

$$E_{HCAL} = \frac{\eta_s \cdot E_s \cdot (\eta_c - 1) - \eta_c \cdot E_c \cdot (\eta_s - 1)}{\eta_c - \eta_s}$$



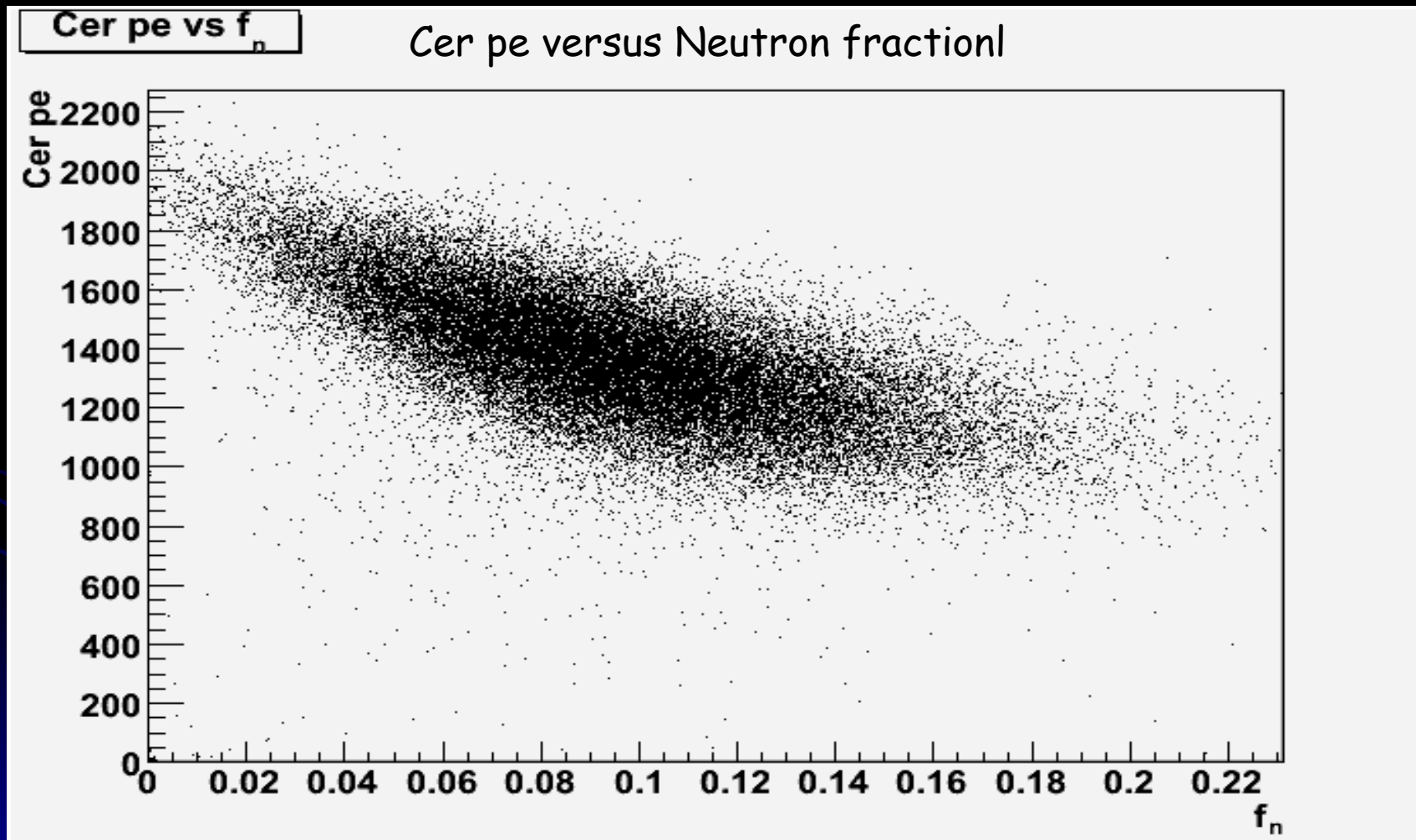
$$\eta_c = \left( \frac{e}{h} \right)_c \quad \eta_s = \left( \frac{e}{h} \right)_s$$

From calibration  
@ 1 Energy only

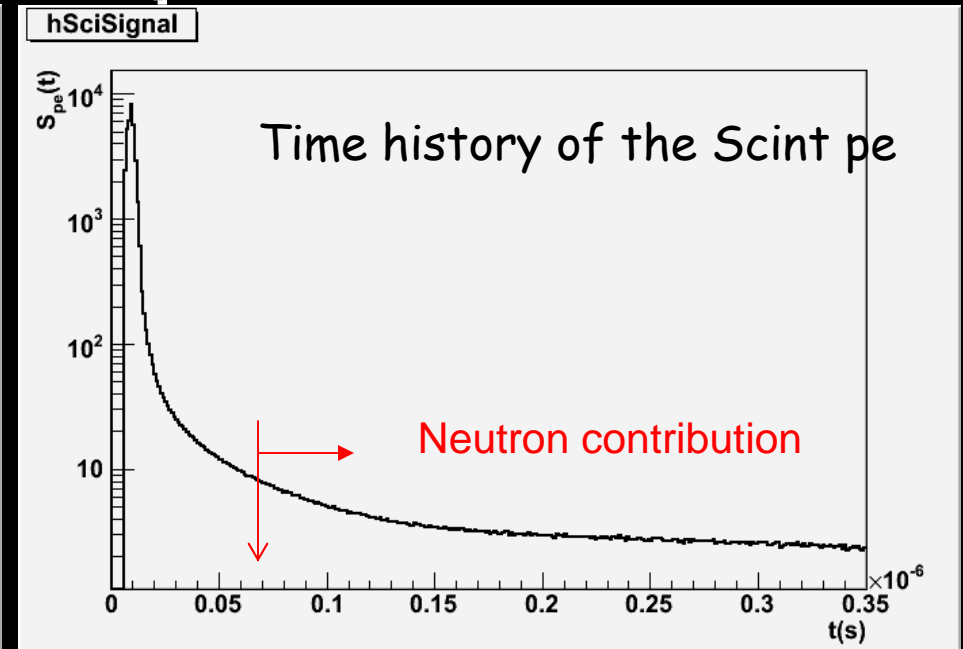
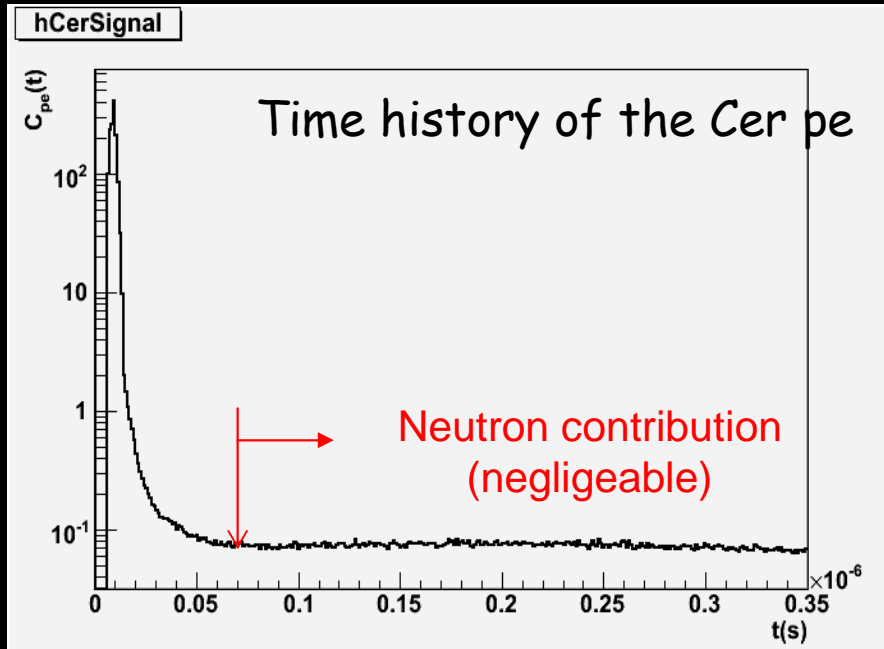
April 19th, 2009

# Improving the Energy Resolution: The Effect of Neutrons

45 GeV  $\pi^-$



# From Dual to Triple Readout



45 GeV  $\pi^-$

$$E_{HCAL} = \frac{\eta_s \cdot E_s \cdot (\eta_c - 1) - \eta_c \cdot E_c \cdot (\eta_s - 1)}{\eta_c - \eta_s} + \eta_n \cdot E_{neutrons}$$

Triple readout aka Dual Readout with time history readout

# PID in ILCroot

## ■ Local PID

- Performed at Local level (detector module)
- Compute the conditional probability density function  $r(s|i)$  ( $s$ =signal,  $i=e,\mu,K,\pi,p,\dots$ )
- Evaluate  $w_{local}(i|s)$  from  $r(s|i)$  and frequency  $C_i$

## ■ Global PID

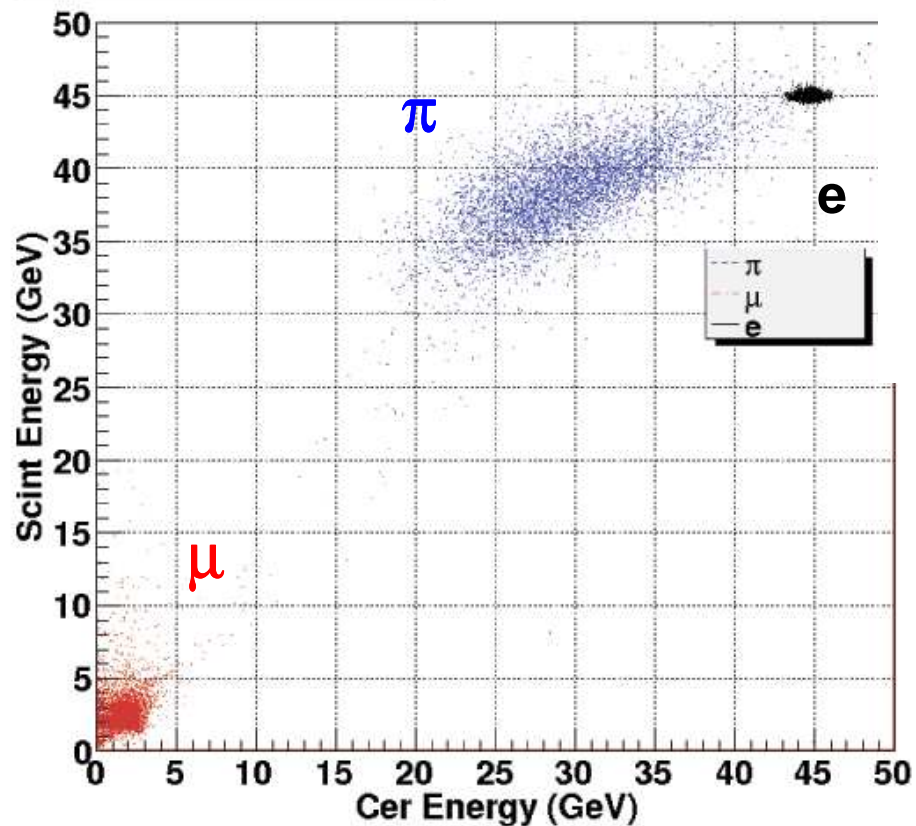
- Performed at Global level
- Combine the PID measurements from all the detectors in a Bayesian way

$$w(i | s) = \prod^{local} \alpha_{local} w_{local}(i | s)$$

# Particle

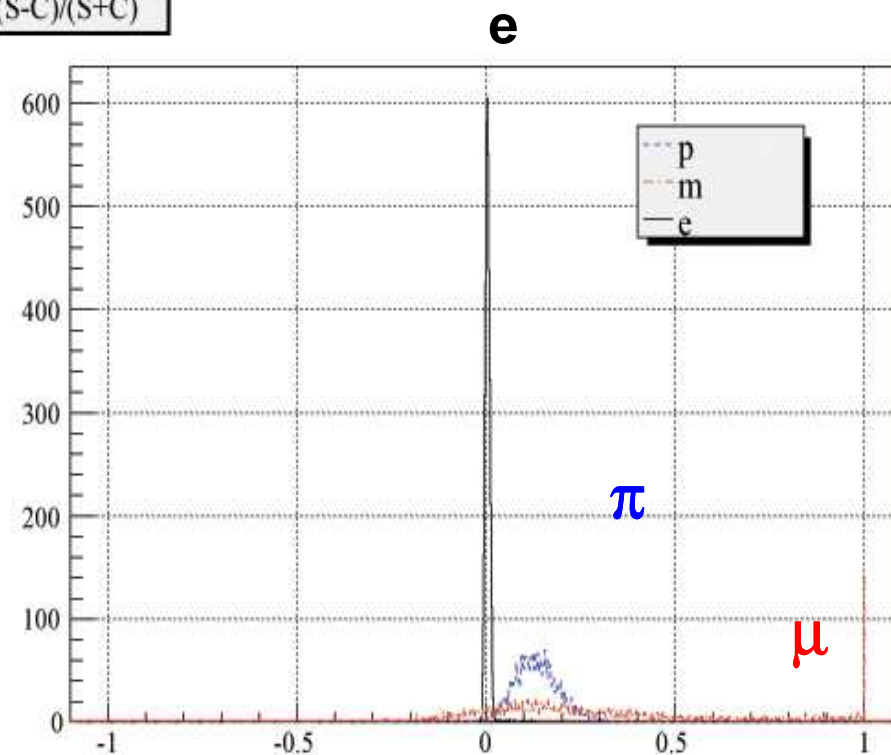
## Identification with Triple Readout

Cer Energy vs Scint Energy



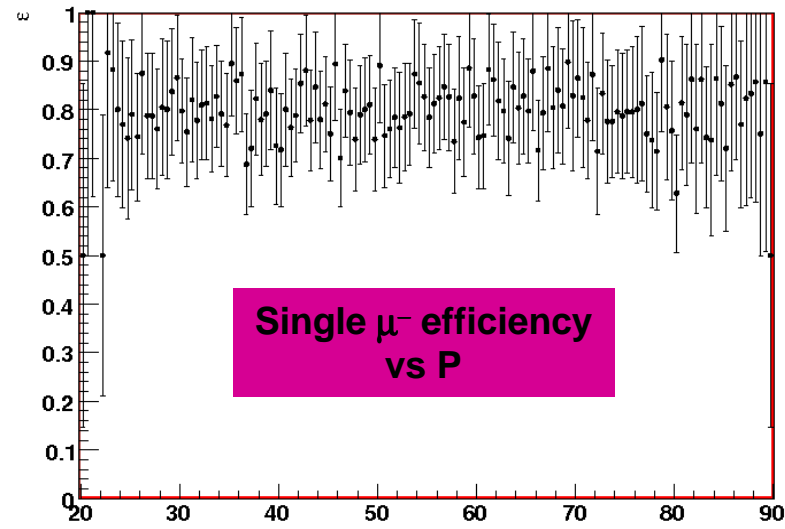
45 GeV particles

$(S-C)/(S+C)$

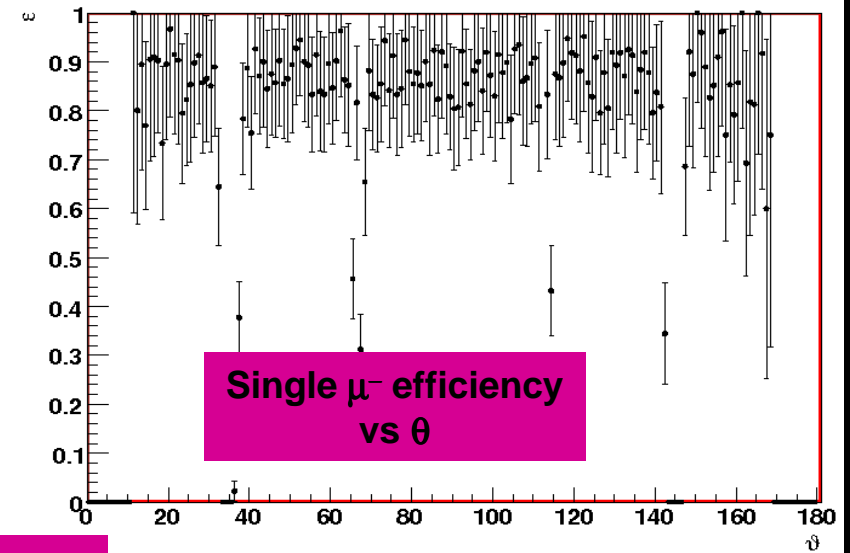


# Muon Identification with m spectrometer

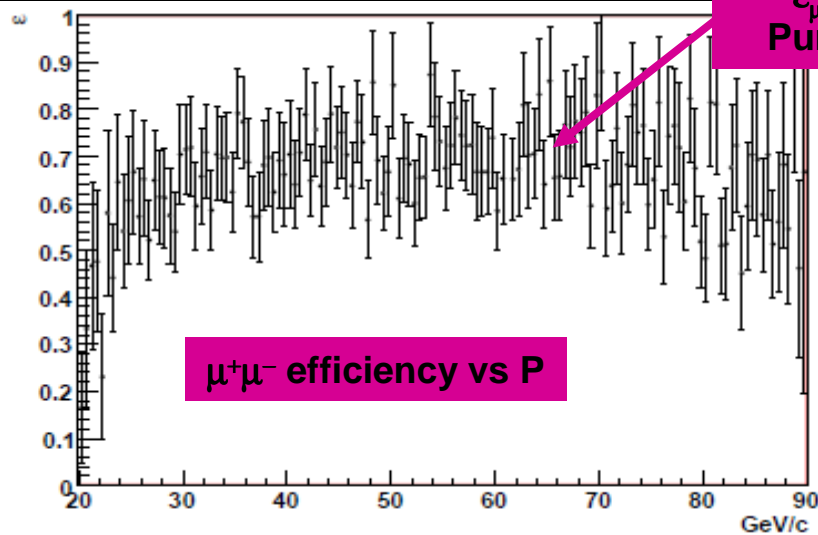
Efficiency of  $\mu^-$  reconstructed in the Muon Spectrometer



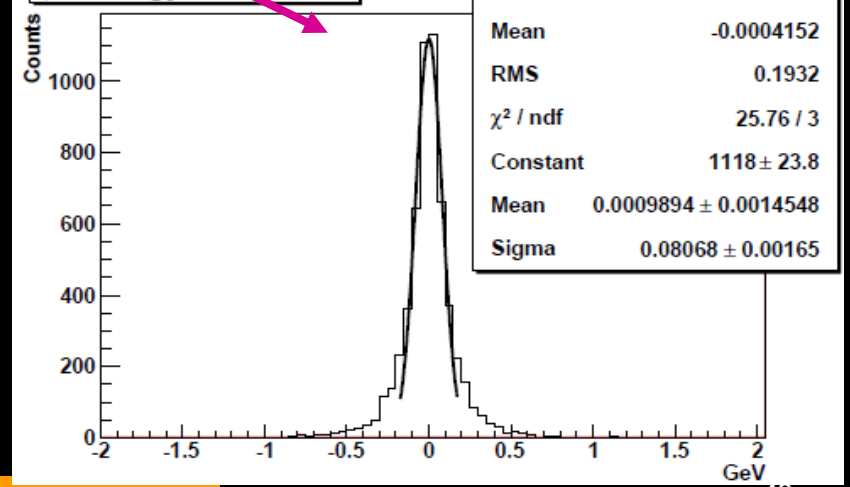
Efficiency of  $\mu^-$  reconstructed in the Muon Spectrometer



$\epsilon_{\mu\mu} = 80.5\%$   
Purity = 99.9%



Energy Resolution



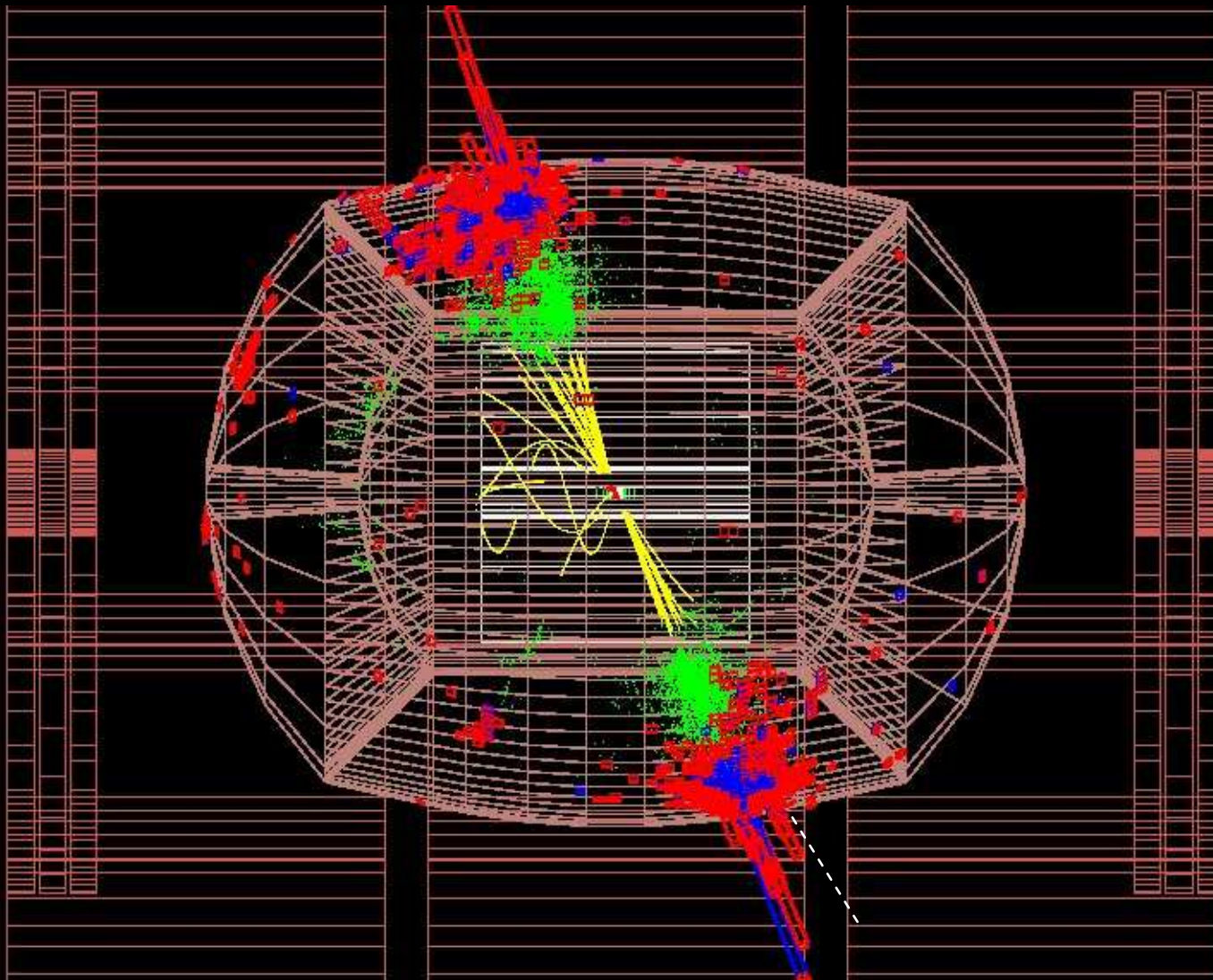
$$e^+e^- \rightarrow Z^0 H^0 \rightarrow \mu^+ \mu^- X \quad \sqrt{s} = 250 \text{ GeV}$$



# Jet reconstruction: combine calorimetric and tracking informations

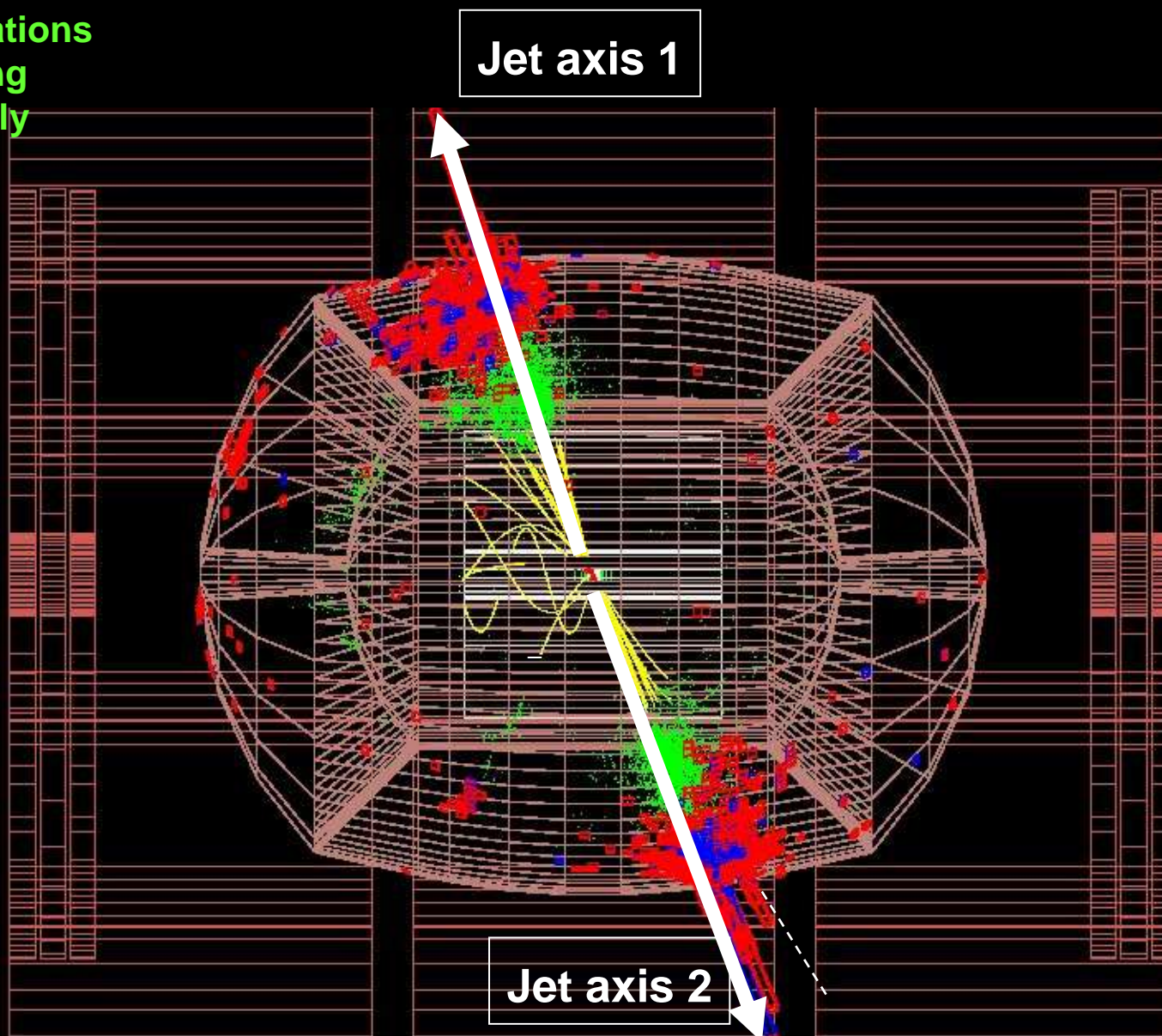
(work in progress)

# Jet Reconstruction Strategy



# Jet Reconstruction Strategy

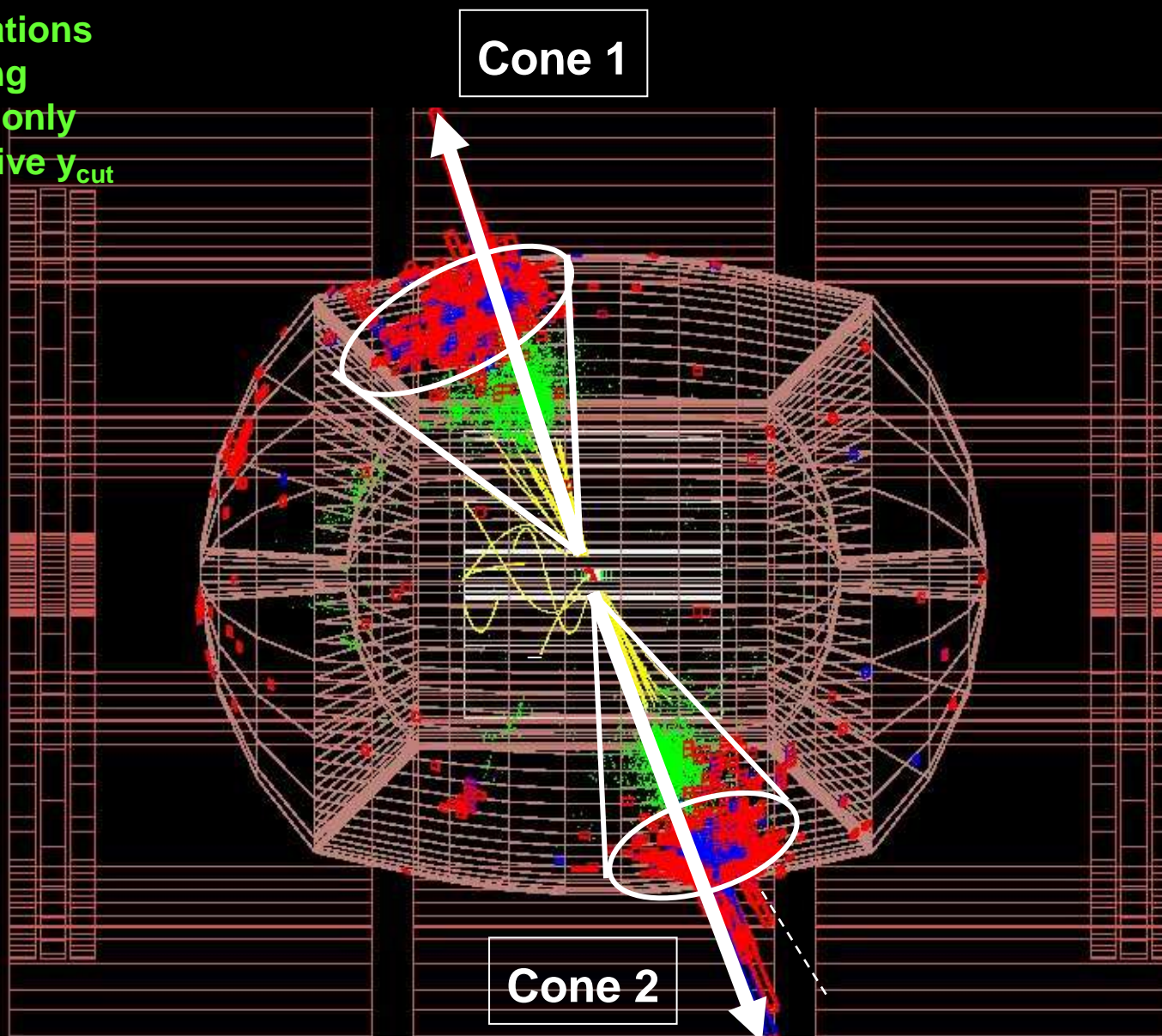
Use informations  
from tracking  
Systems only





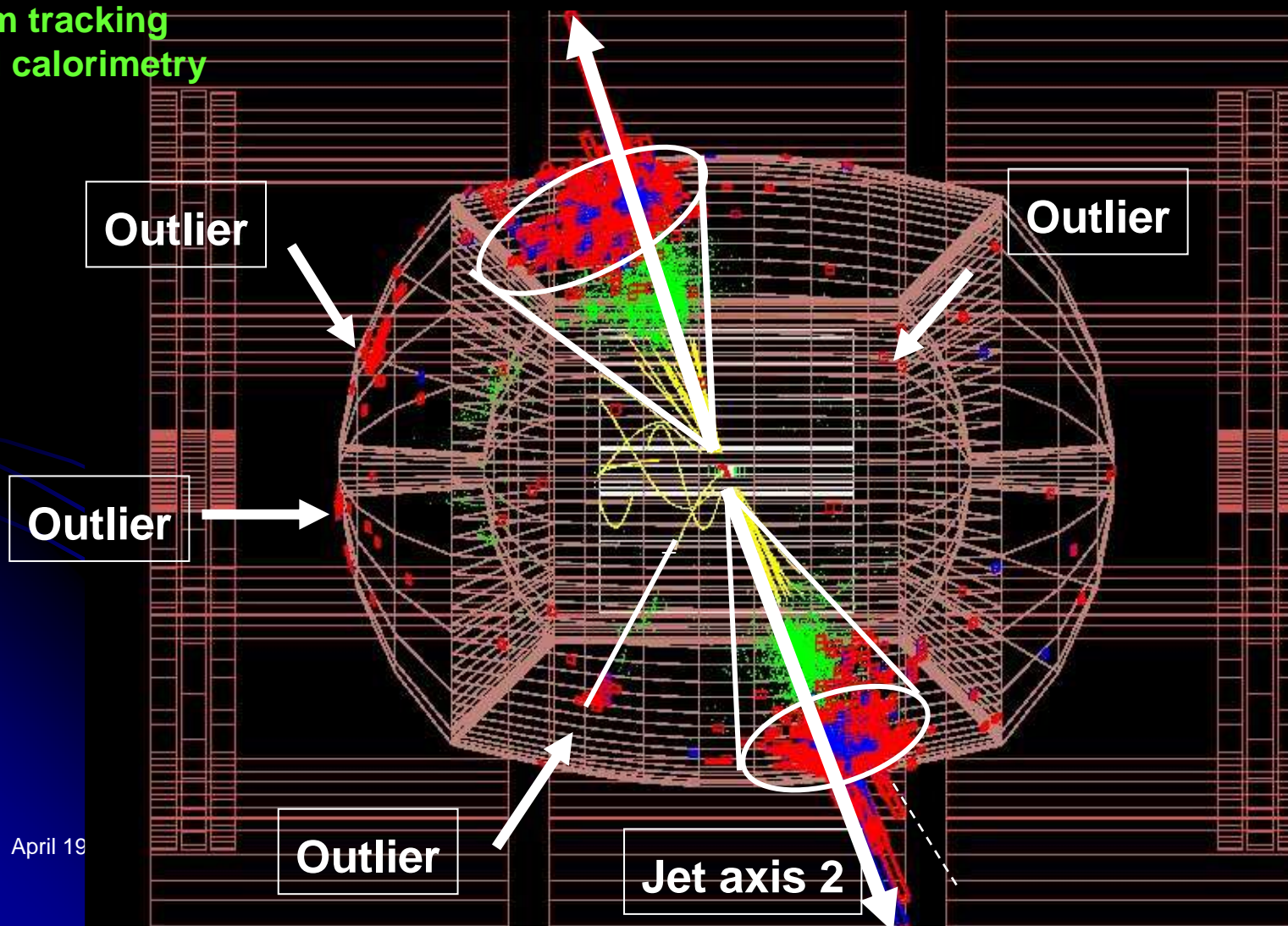
# Jet Reconstruction Strategy

Use informations  
from tracking  
calorimetry only  
With recursive  $y_{\text{cut}}$



# Jet Reconstruction Strategy

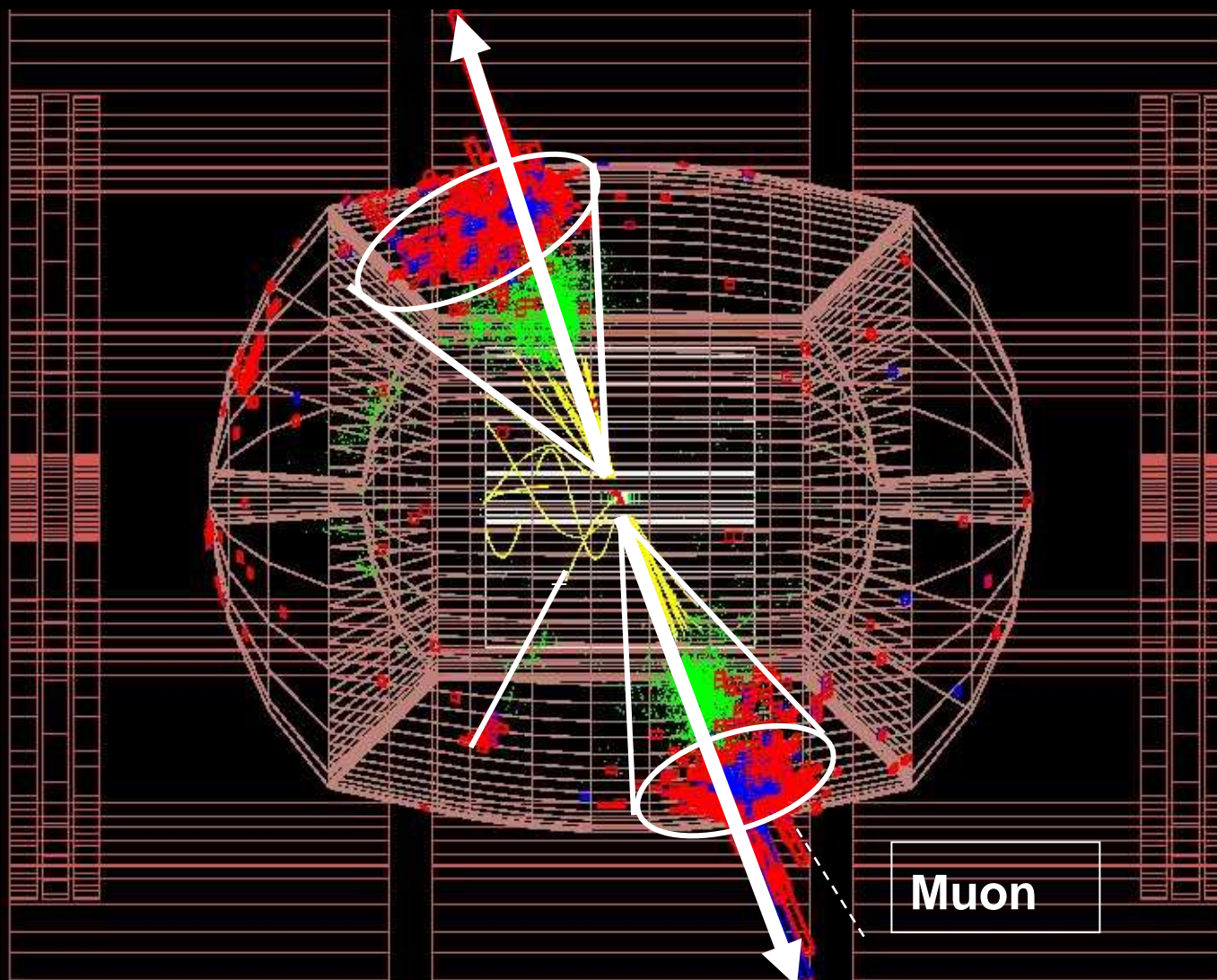
Use combined  
informations  
from tracking  
and calorimetry





# Jet Reconstruction Strategy

Include muon  
spectrometer



# Could it ever evolve into a general purpose tool for the HEP community (as ROOT)?

- Growing number of experiments have adopted it: Alice (LHC), Opera (LNGS), (Meg), CMB (GSI), Panda(GSI), 4th Concept, LHeC and the forthcoming International Dual Readout Collaboration Main advantage: exchange modules among experiments
  - Generators (signal/background)
  - Detector simulation (MC step, digitization, P.R,...)
  - Software tools (vertexing, flavor tagging, ....)
- Other advantages:
  - Development and support

# Is it feasible?

- Naked framework is already detector independent
  - Persistent object loaders (in highest efficient way)
  - Steering program
  - Generator interface (4th Concept made it even more general)
  - VMC interface
- Ilcroot adopted some of other experiment code:
  - TPC, ECAL and digitization in Si from Aliroot
  - DCH from Megroot
- Ilcroot is growing along Aliroot
  - About 1 code realignment/yr
  - Parametrize all the hard-coded stuff
- Preliminary discussion between Alice and 4th Concept for an *experiment-independent* framework (CERN – Dec. 2006)



# Ilcroot Status

- VXD, ECAL and HCAL: digitization + recpoints
- MUD: fastrecpoints (aka, gaussian smearing of hits)
- DCH: fast algorithm to guarantee correct occupancy with beam background
- Most of developments toward detector optimization
- Analysis and Physics studies in parallel

# Backup Slides

# ILCroot: a summary of features

- CERN architecture (based on Alice's Aliroot)
- Full support provided by Brun, Carminati, Ferrari, et al.
- Uses ROOT as infrastructure
  - All ROOT tools are available (I/O, graphics, PROOF, data structure, etc)
  - Extremely large community of users/developers
- Six MDC have proven robustness, reliability and portability

# 4<sup>th</sup> Concept Software Framework: ILCroot

- **Introduced at ACFA06 (Bangalore) after Aliroot and MEGroot**
- **Only three additions ever since:**
  1. Interface to external files in various format (STDHEP, text, etc.)
  2. Standalone VTX track fitter
  3. Pattern recognition from VTX (for si central trackers)
- Growing number of experiments have adopted it: Alice (LHC), Opera (LNGS), (Meg), CMB (GSI), Panda(GSI), 4th Concept, **LHeC and the forthcoming International Dual Readout Collaboration**
- Modularity allows to reuse subdetector modules developed by other collaborations
- Observation: it is a simulation framework and an Offline Systems:
  - It naturally evolves into the offline systems of your experiment
  - It is immediatly usable for test beams
  - Six MDC have proven robustness, reliability and portability



# From Bangalore to Now: Detectors in ILCroot

## ■ 4th Concept Baseline

- VTX: from SiD scaled to 3.5 Tesla (original version)
- Drift Chamber: 2° version
- Fiber Triple Readout Calorimeter: 3° version
- Muon Spectrometer: Original version

## ■ New additions:

- Crystal Triple Readout Calorimeter ← Included into the LoI
- FTD (from SiLC) ← Did not make it into the LoI

## ■ Also available

- VTX Detectors: Original SiD
- Central Trackers: TPC, Si-Strips (SID01), SPT (Pixel Tracker)

## ■ Total: 10 subdetectors (16 versions), most of them with full simulation

# Processing Flow

VMC

## Reconstruction

Steer

