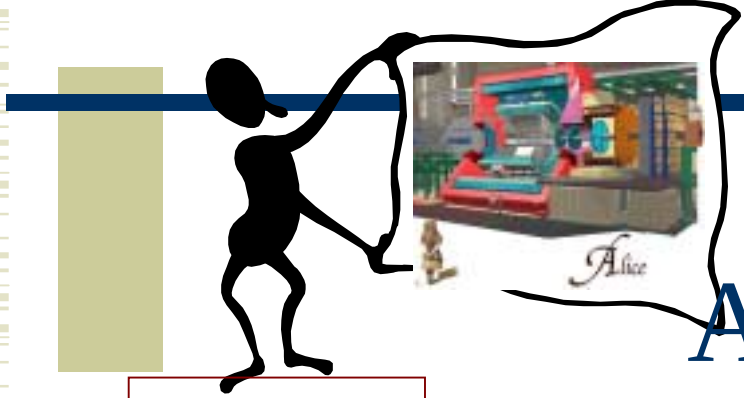
AliRoot: I/O organisation

ROOT in the ALICE experiment

I/O Structure

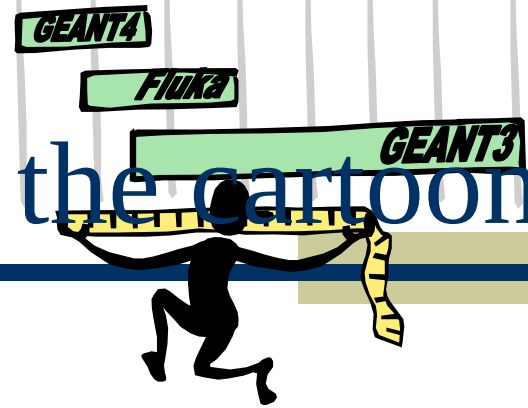
Run-Time Data-Exchange

AliRoot, the cartoon

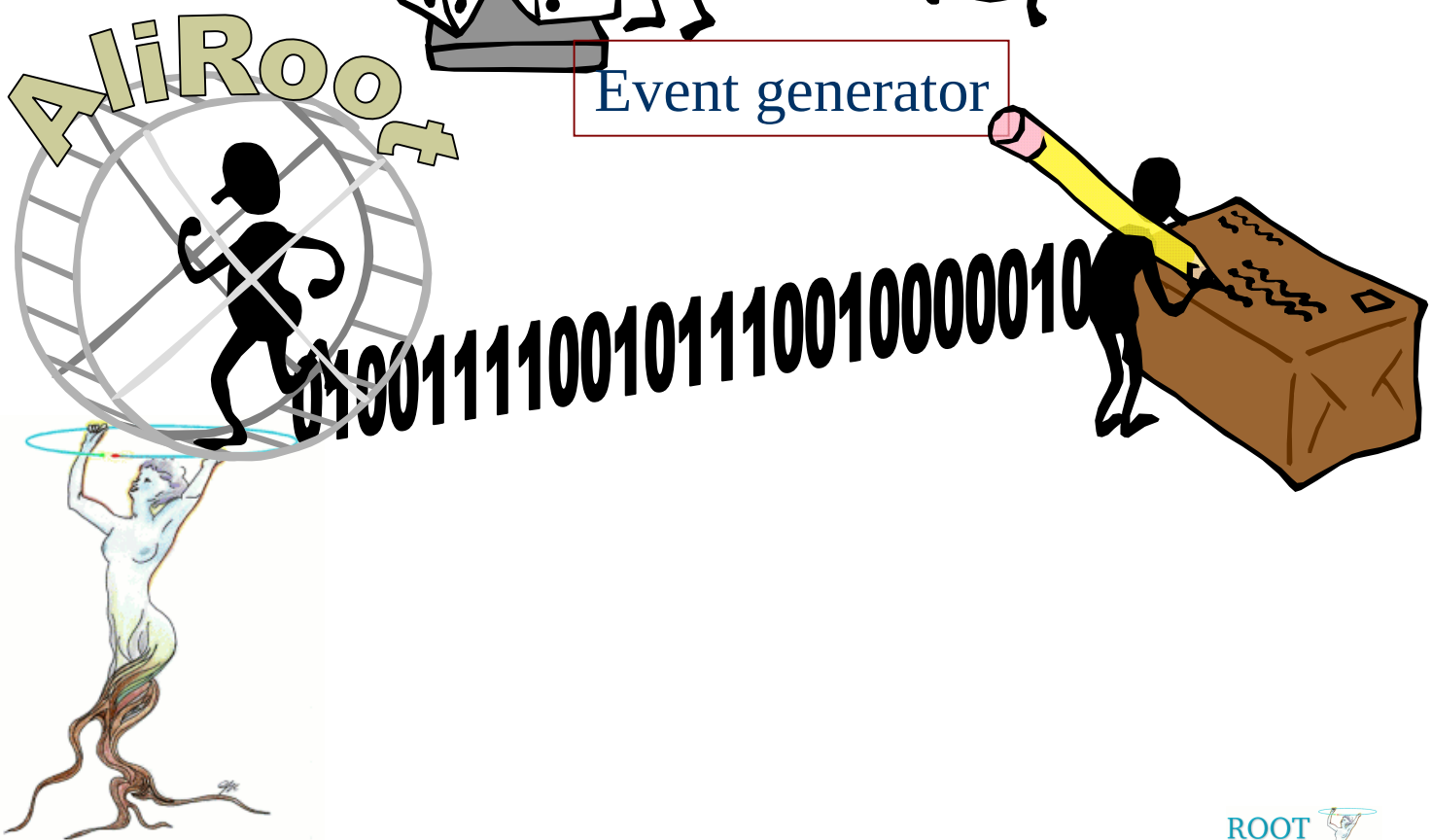


Geometry

Transport model



Event generator

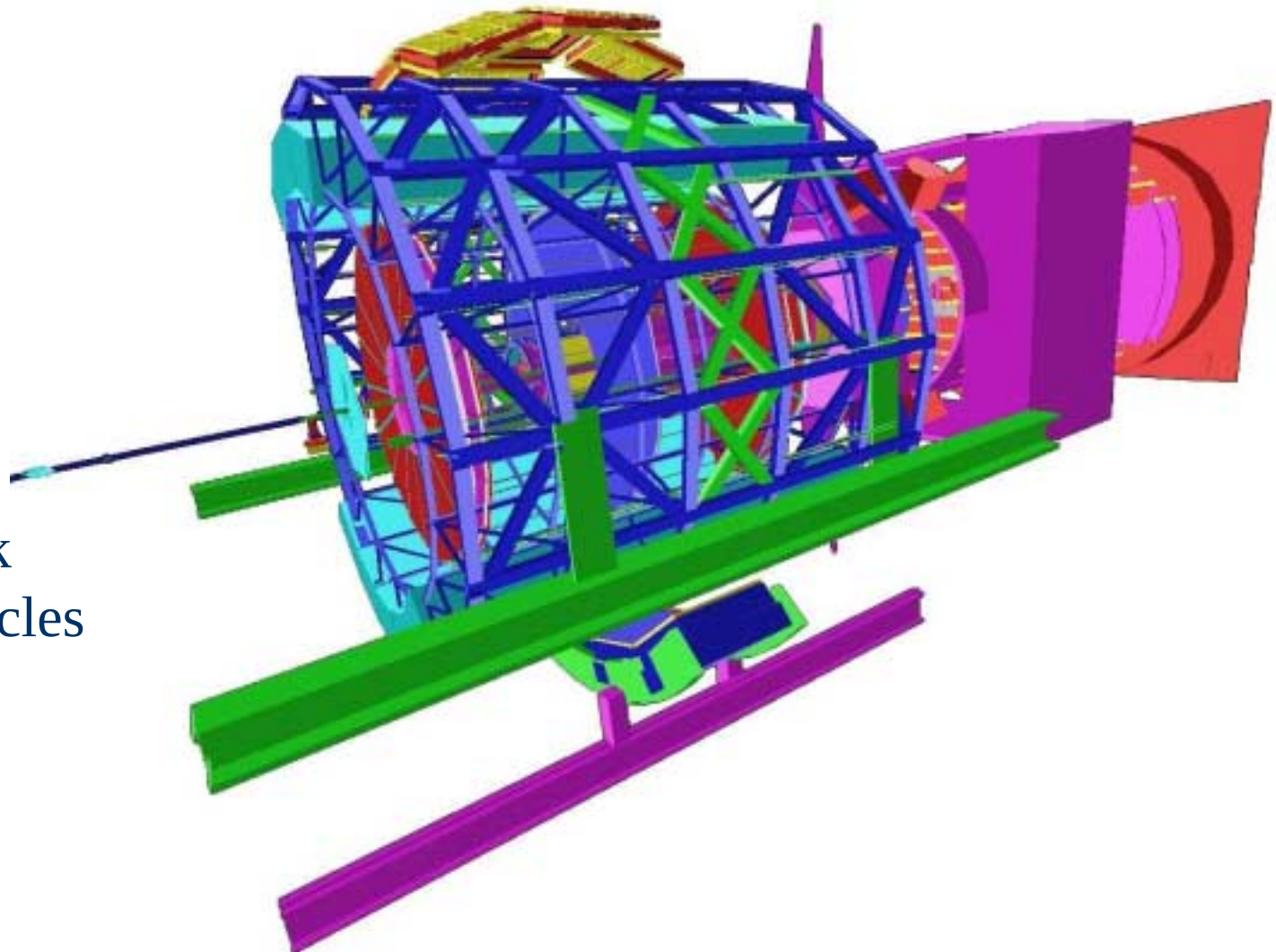


ROOT

An Object-Oriented
Data Analysis Framework

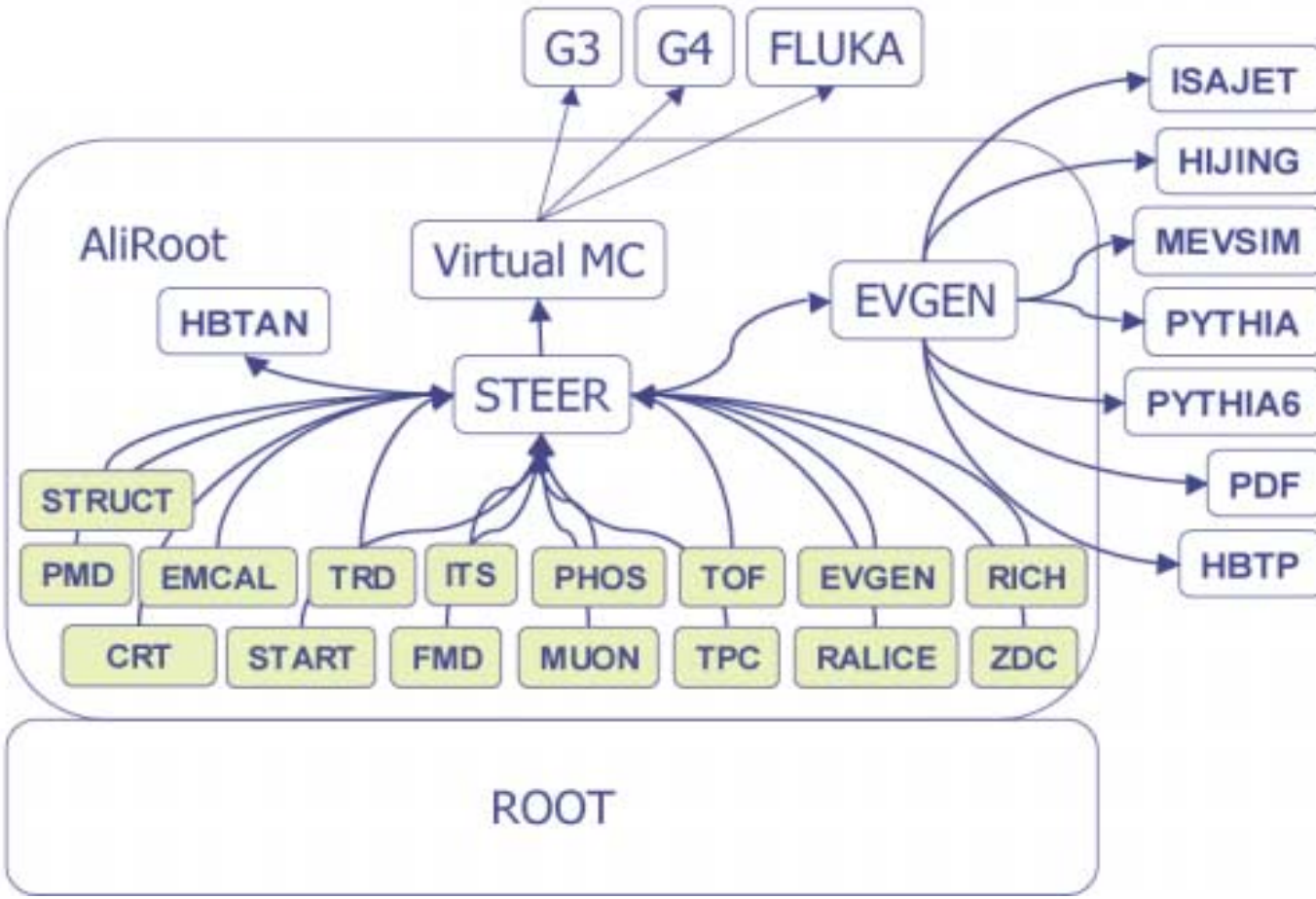
The Geometry

12 independent detectors which cooperate to track and identify particles



AliRoot Layout

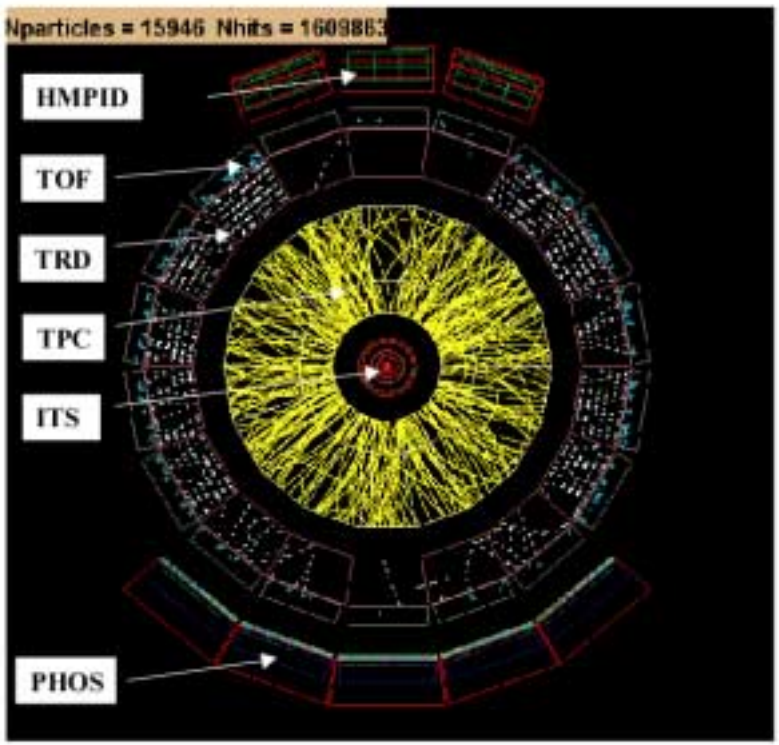
- Structure of the framework reflects the ALICE experiment sub-structure
- STEER coordinates the cooperation between detectors



01001111001011100100001011

The Data

Central PbPb collisions at 5.5 TeV/nucleon pair

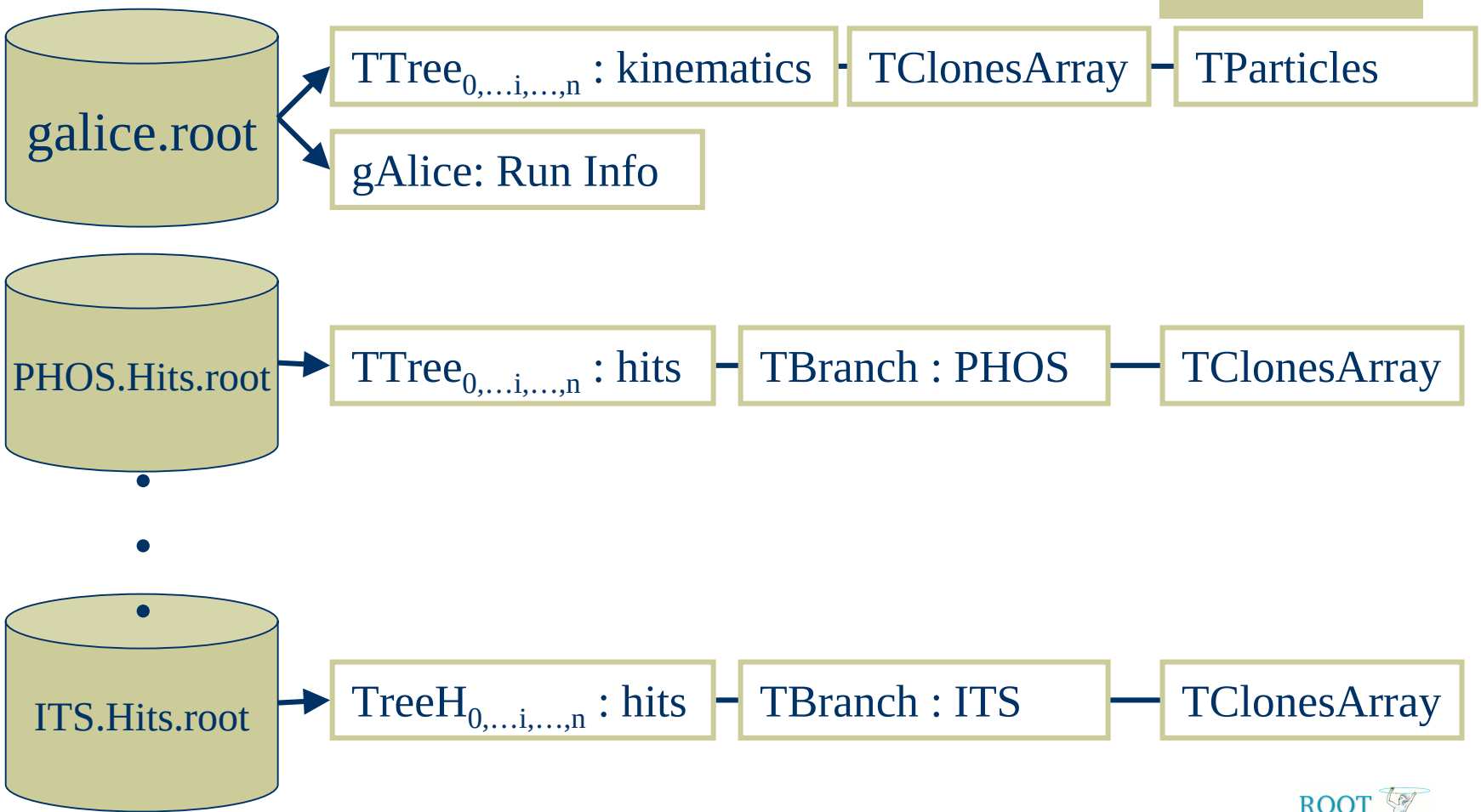


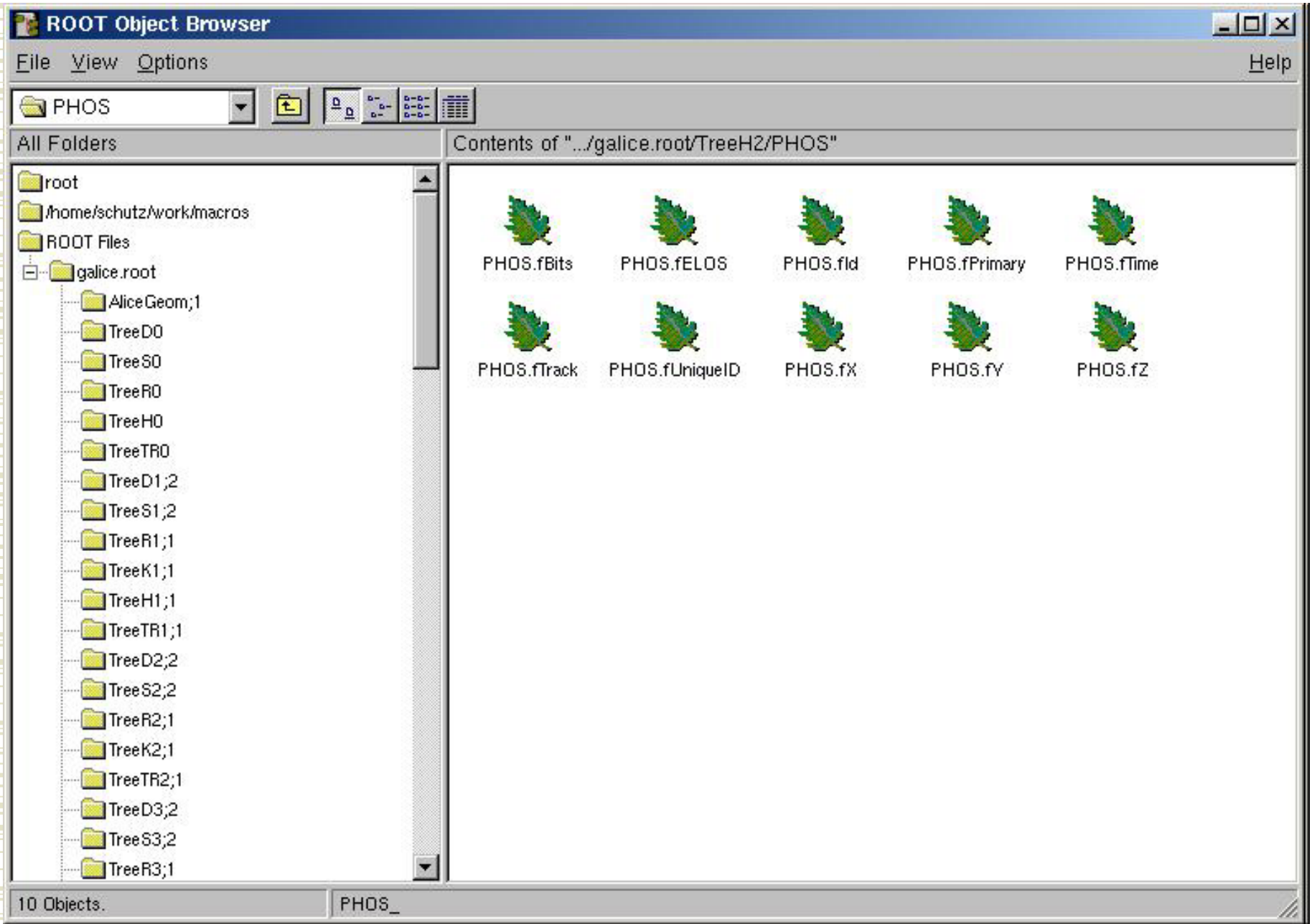
(only 4 % of particles in central barrel are displayed)

Structure the data to reflect the detector sub-structure

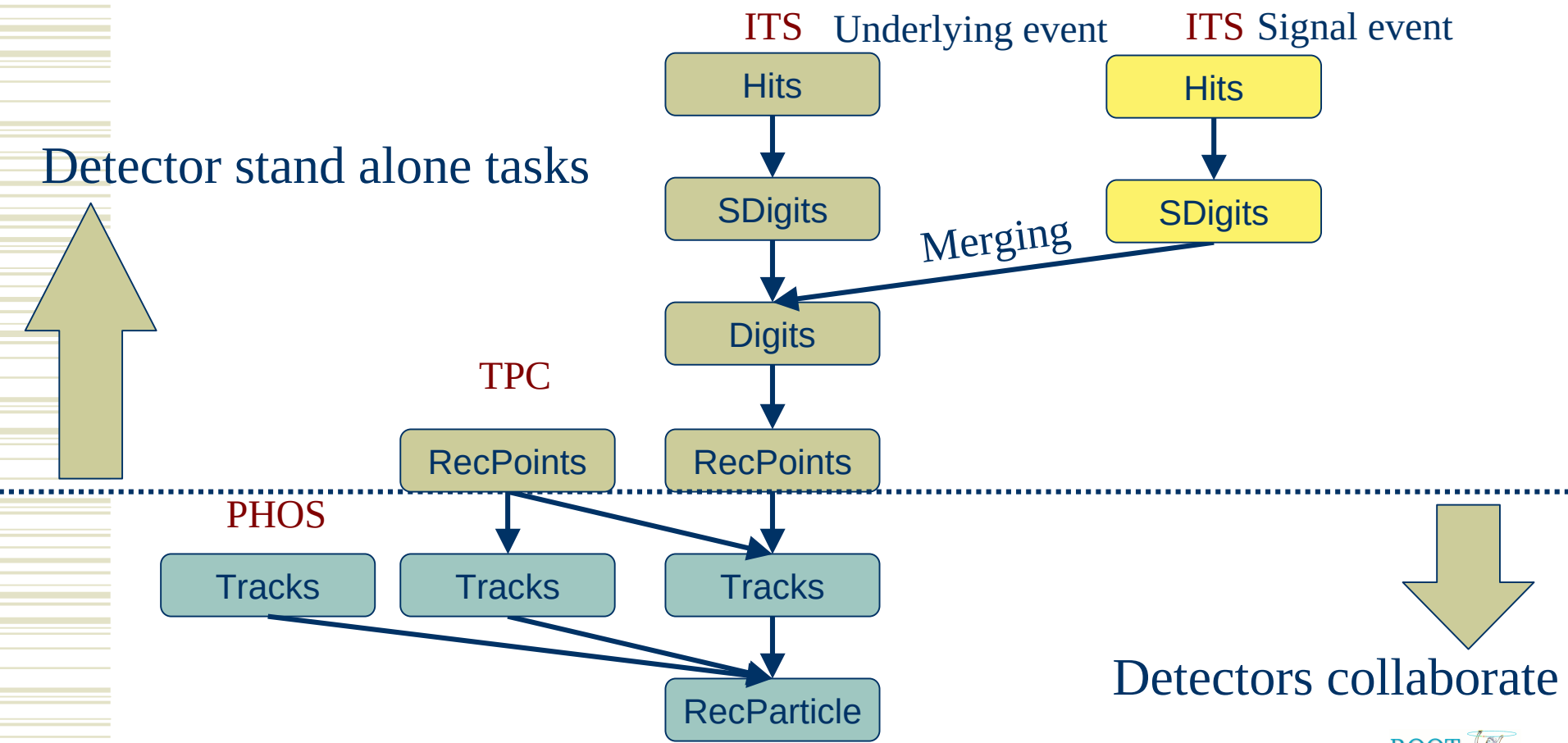
Structure of « raw » data:

1 common TFile + 1 TFile per detector +1 TTree per event

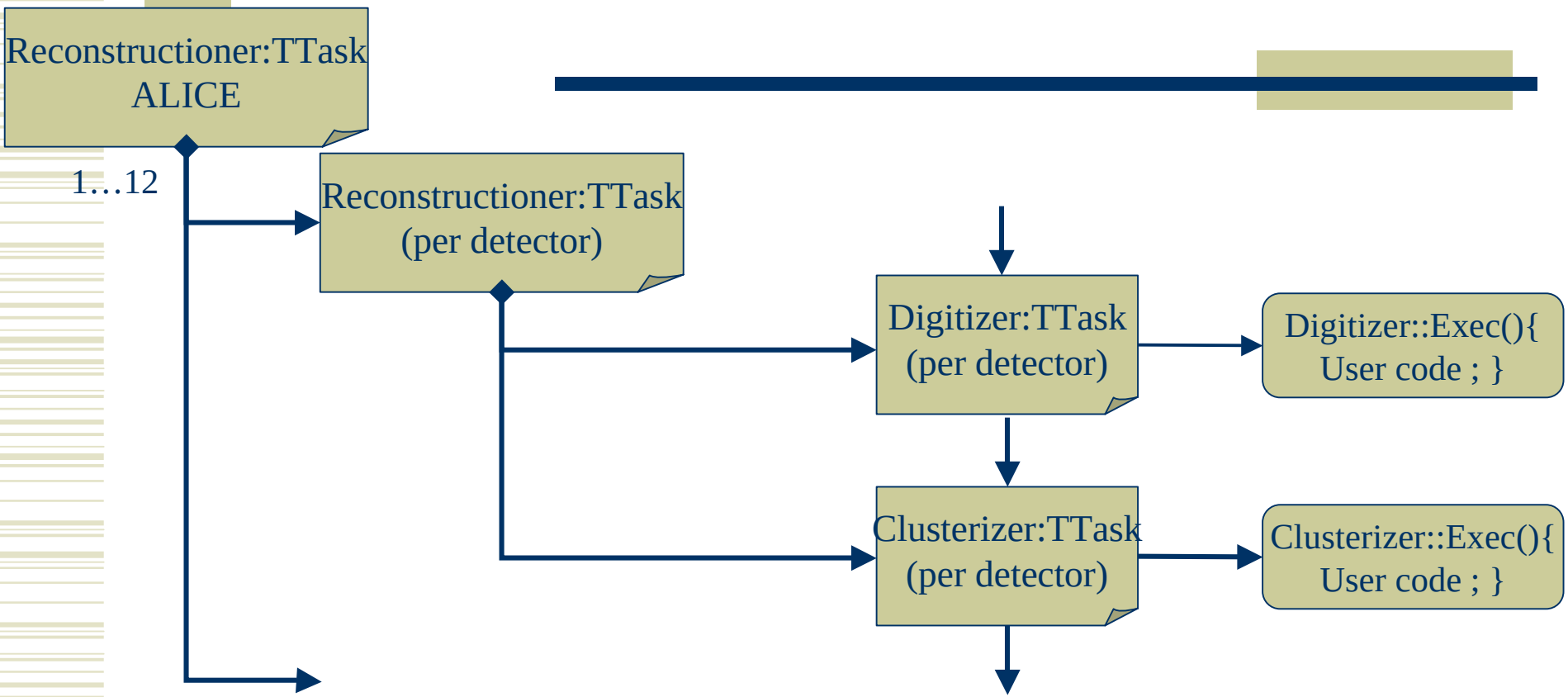



ROOT Object Browser
File View Options Help
PHOS
All Folders
Contents of ".../galice.root/TreeH2/PHOS"
root
/home/schutz/work/macros
ROOT Files
galice.root
AliceGeom;1
TreeD0
TreeS0
TreeR0
TreeH0
TreeTR0
TreeD1;2
TreeS1;2
TreeR1;1
TreeK1;1
TreeH1;1
TreeTR1;1
TreeD2;2
TreeS2;2
TreeR2;1
TreeK2;1
TreeTR2;1
TreeD3;2
TreeS3;2
TreeR3;1
PHOS.fBits
PHOS.fELOS
PHOS.fId
PHOS.fPrimary
PHOS.fTime
PHOS.fTrack
PHOS.fUniqueID
PHOS.fX
PHOS.fY
PHOS.fZ
10 Objects.
PHOS_

Reconstruction Tasks



Structure of a task:TTask

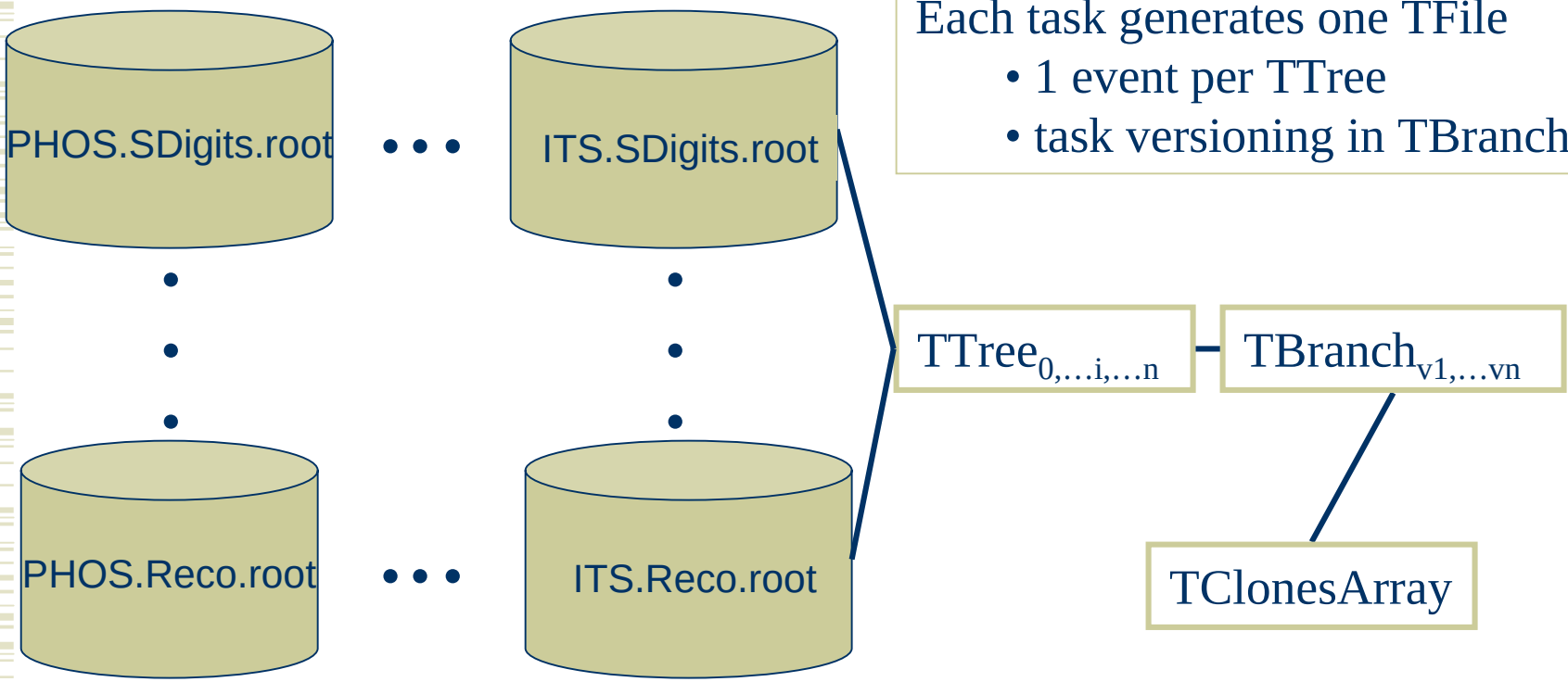


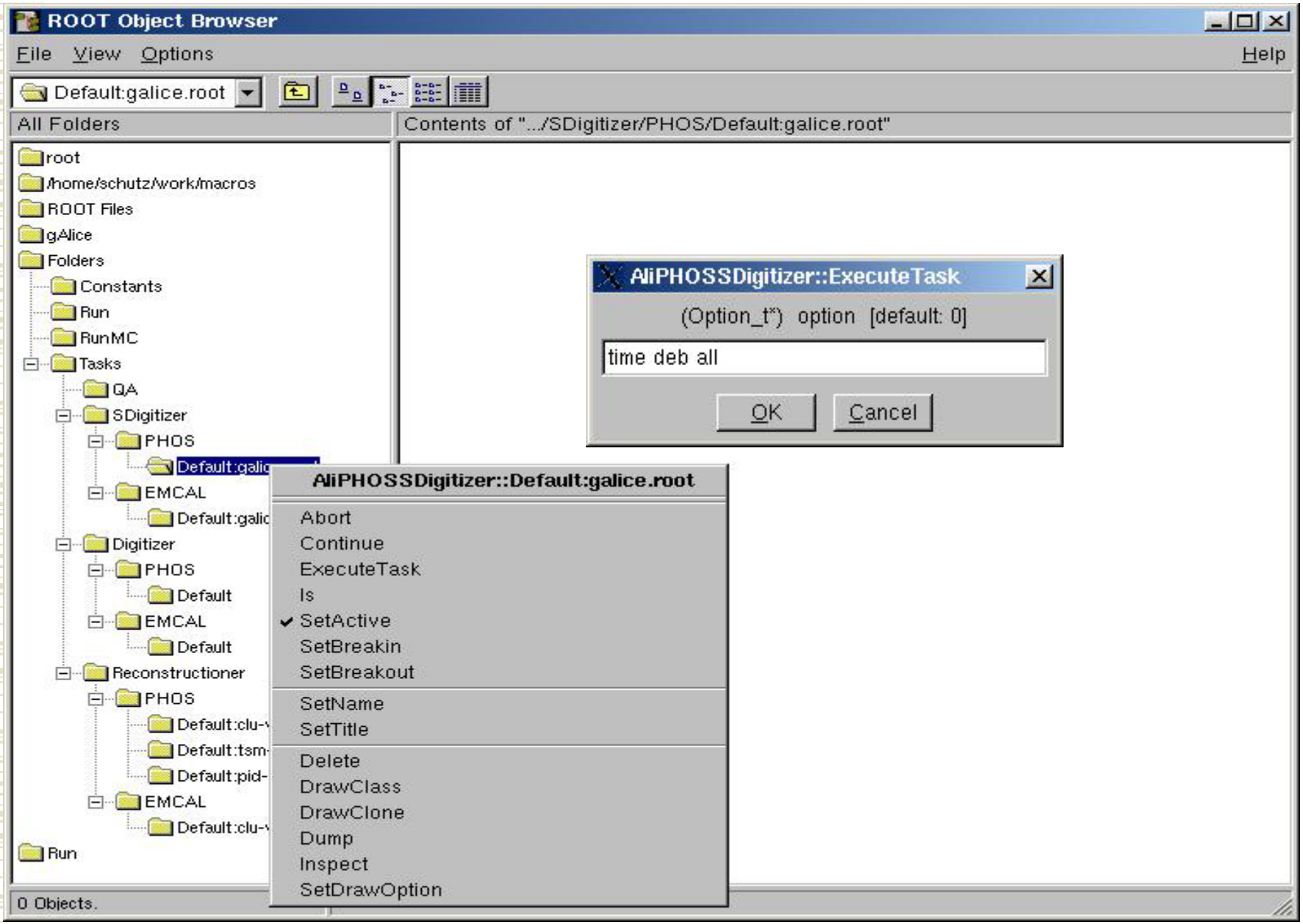
root> AliRMCAL Reconstructioner r;

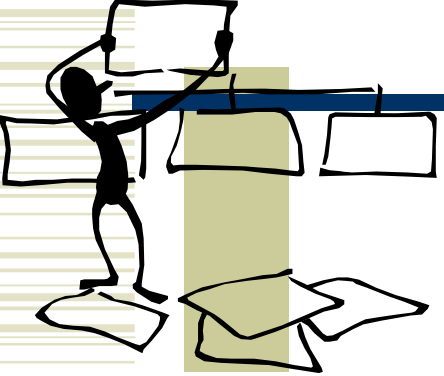
root> r.ExecuteTask()

Structure of reconstructed data

Detector wise splitting

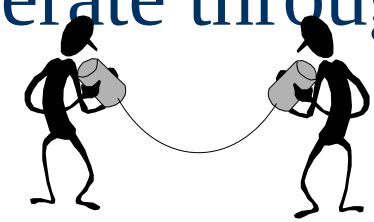


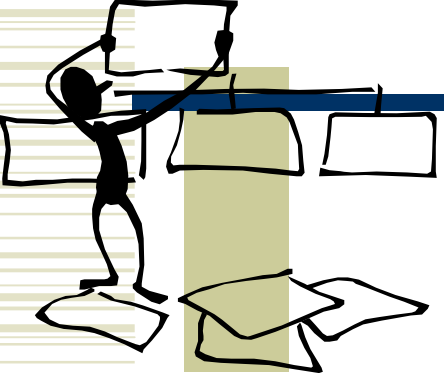

ROOT Object Browser
File View Options Help
Default:galice.root
All Folders
Contents of ".../SDigitizer/PHOS/Default:galice.root"
root
/home/schutz/work/macros
ROOT Files
gAlice
Folders
Constants
Run
RunMC
Tasks
QA
SDigitizer
PHOS
Default:galic
EMCAL
Default:galic
Digitizer
PHOS
Default
EMCAL
Default
Reconstructioner
PHOS
Default:clu-
Default:tsm-
Default:pid-
EMCAL
Default:clu-
Run
0 Objects.
AliPHOSSDigitizer::ExecuteTask
(Option_t*) option [default: 0]
time deb all
OK
Cancel
AliPHOSSDigitizer::Default:galice.root
Abort
Continue
ExecuteTask
ls
SetActive
SetBreakin
SetBreakout
SetName
SetTitle
Delete
DrawClass
DrawClone
Dump
Inspect
SetDrawOption



Run-time Data-Exchange

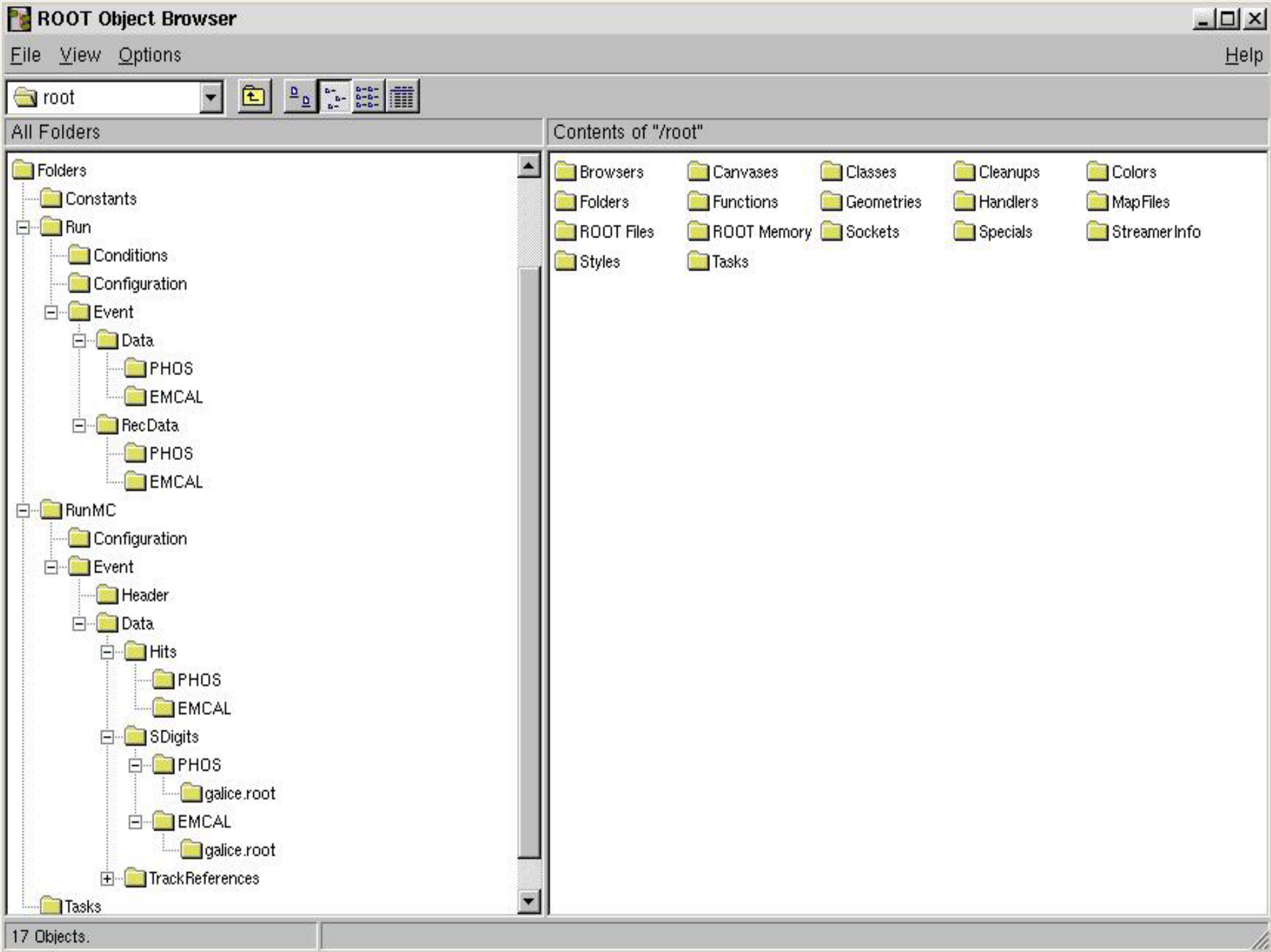
- Post transient data to a white board
- Structure the white board according to detector sub-structure & tasks results
- Each detector is responsible for posting its data
- Tasks access data from the white board
- Detectors cooperate through the white board

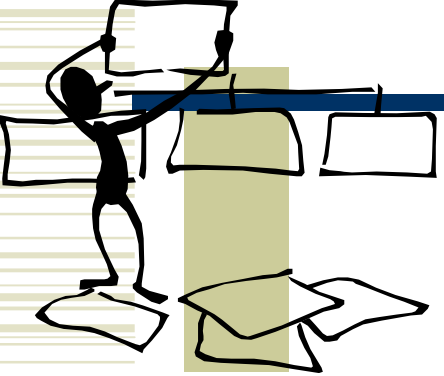




Run-time Data-Exchange

- Post transient data to a white board : make use of TFolder :
  - Unix-dir like structure
  - root dir //Folders
  - Direct search: TObject * TFolder::FindObjectAny(name) ;
  - Add a folder: TFolder::AddFolder(name) ;
  - Add any object: TFolder::Add(Tobject *) ;

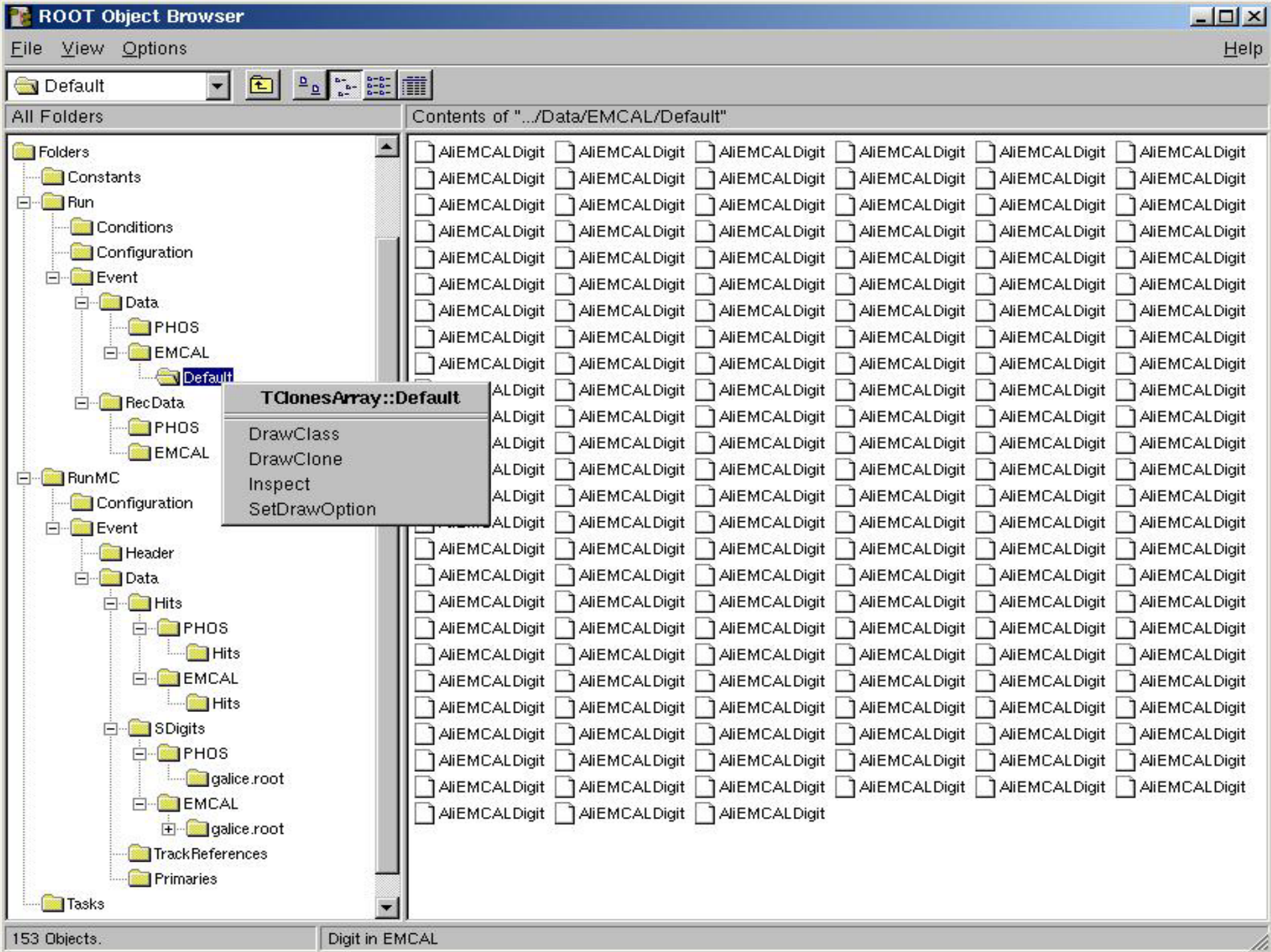

ROOT Object Browser
File View Options Help
root
All Folders
Contents of "/root"
Folders
Constants
Run
Conditions
Configuration
Event
Data
PHOS
EMCAL
RecData
PHOS
EMCAL
RunMC
Configuration
Event
Header
Data
Hits
PHOS
EMCAL
SDigits
PHOS
galice.root
EMCAL
galice.root
TrackReferences
Tasks
Browsers
Canvases
Classes
Cleanups
Colors
Folders
Functions
Geometries
Handlers
MapFiles
ROOT Files
ROOT Memory
Sockets
Specials
StreamerInfo
Styles
Tasks
17 Objects.

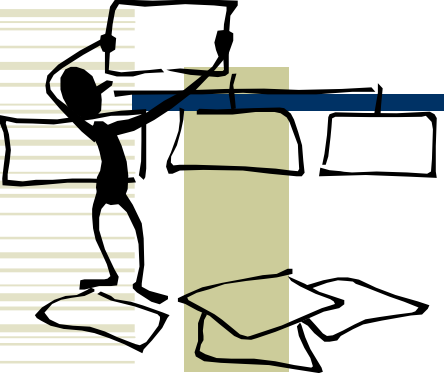


Run-time Data-Exchange

- Each detector is responsible for posting its data: delegate all the transactions to a singleton object, one per detector

```
root> AliPHOSGetter *gime = AliPHOSGetter::GetInstance(common_filename) ;
root> gime->Event(n) ;// one event at the time
root> TClonesArray digits = gime->Digits() ;
```


ROOT Object Browser
File View Options Help
Default
All Folders
Contents of ".../Data/EMCAL/Default"
Folders
Constants
Run
Conditions
Configuration
Event
Data
PHOS
EMCAL
Default
RecData
PHOS
EMCAL
RunMC
Configuration
Event
Header
Data
Hits
PHOS
Hits
EMCAL
Hits
SDigits
PHOS
galice.root
EMCAL
galice.root
TrackReferences
Primaries
Tasks
TClonesArray::Default
DrawClass
DrawClone
Inspect
SetDrawOption
AliEMCALDigit
153 Objects.
Digit in EMCAL



Run-time Data-Exchange

- Tasks access data from the white board
- Detectors cooperate through the white board
  - Have a general getter that has a list of detector and executes tasks recursively

```
AliGetter * gime = AliGetter::GetInstance() ;
AliPHOSGetter *gimeP = AliPHOSGetter::GetInstance() ;
AliEMCALGetter *gimeE = AliEMCALGetter::GetInstance() ;
gime->Add(gimeP, gimeE) ;
gime->Event(n) ;
for (i=0; i<gimeP->RecParticles()->GetEntries(); i++)
  for (j=0; j<gimeE->RecParticles()->GetEntries(); j++)
    if( AreBack2Back(gimeP->RecParticle(i), gimeE->RecParticle(j)) )
      njets++ ;
```