

# ROOT 2001

## ROOT in Nuclear Physics

Martin Schulte-Wissermann, TU-Dresden, Germany for the COSY-TOF Collaboration  
email: [schulte@pkitw09.phy.tu-dresden.de](mailto:schulte@pkitw09.phy.tu-dresden.de) www: <http://pkitw09.phy.tu-dresden.de/~schulte/>

# COSY-TOF

- a typical (**medium energy**) **nuclear physics** experiment -
  - 40 collaborators from 6 universities
  - 1500 channels (ADC, TDC, scaler, clock)
  - 100 GB per year
  - 2 times 3 weeks of beam time / year
  - **variable detector setup**

Many parameters change frequently!! Flexibility needed !!

# Our Work

- Detector development +++
- Detector setup +
- Electronics +
- Trigger -
- EventBuilder -
- Online Monitor +
- RAW-Data to tape +
- Data Calibration +++
- Data Reduction +++
- Data Analysis ++
- PhysRev/Conferences +

## Nuclear Physics :

- many different skills required
- `nowhere perfect`

Where I have contributed so far

# Software development : specific problems

- different languages (Fortran, C, C++ )
- many different data samples of varying structure
- mini-teams (1-3 persons) working on different beam times
- coordination?
- how to use, maintain and improve existing code?
- how to avoid parallel development?
- how to combine the work of all efficiently?
- documentation!!
- ...

# TofRoot

Summer 1999, after a major detector upgrade:

For data calibration and analysis we need more efficient programming, more team work and more complex algorithms than in the past.

and how can we achieve this ? ...

# TofRoot : Aim

## **1. Main decision :**

- ROOT as the fundamental framework

## **2. Create a „TOF-framework“ with :**

- one data format (a definition) for all beam times
- divide data analysis into (natural) substeps (programs/functions)
- modular code : container classes for each subdetector
- ONE data-base for calibration data (for each beam time)

## **3. And this should be :**

- supporting teamwork
- easy to use
- robust
- well documented!!

... who is in the TofRoot team? ...

# TofRoot : Our Team

**Developer** and user :

Martin Schulte-Wissermann

Christoph Plettner

**User** and developer :

Leo Karsch

all new diploma and PhD students

... what have we done so far? ...

# Our Classes

## Index

---

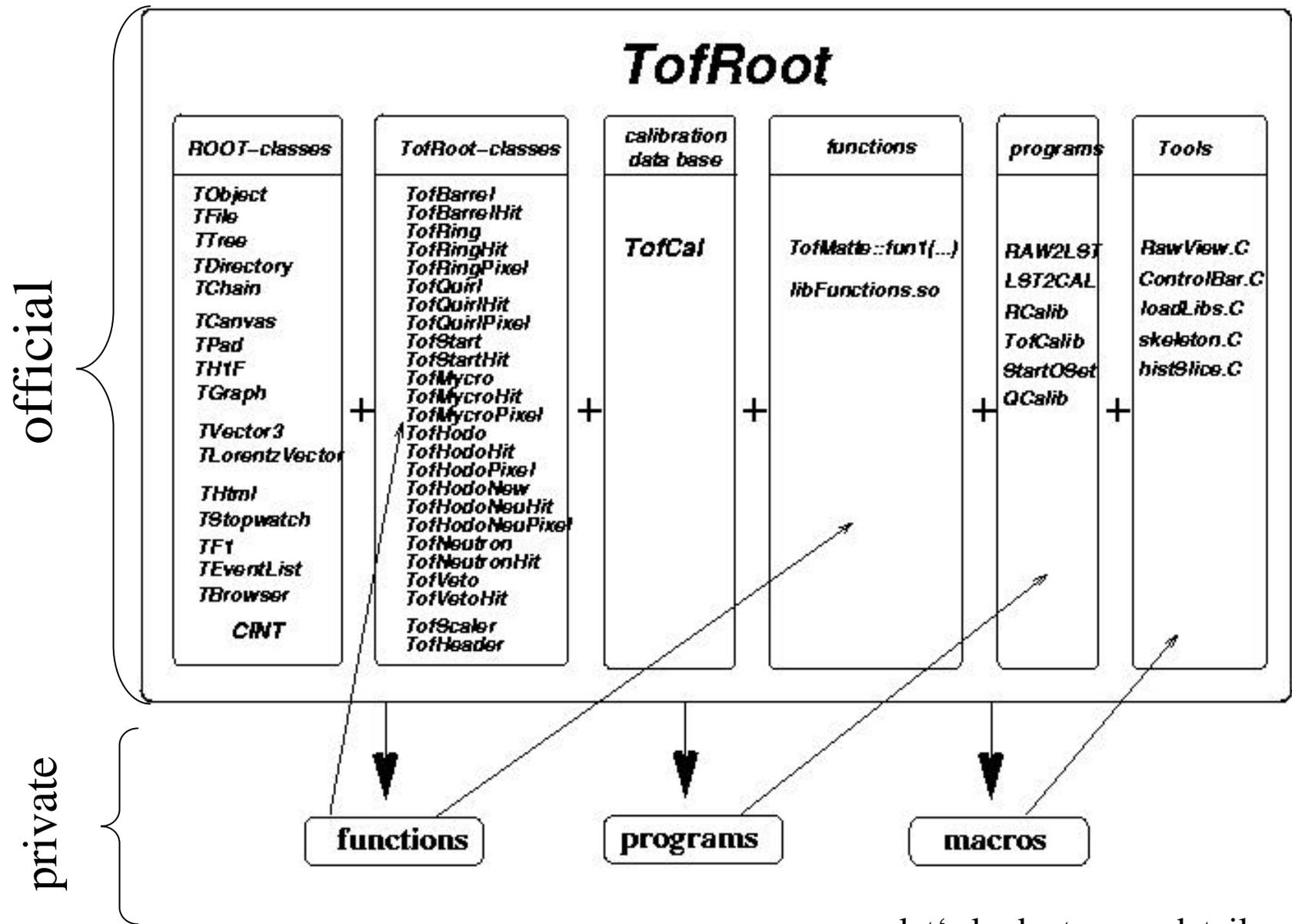
• Kombination	Class used by UPixel (Pixel reconstruction)
• SucheEineSpur	Finds a pixel from UPixel and returns pixel position
• SucheZweiSpuren	Finds two pixels from UPixel and returns pixel position
• Tabelle	Table used by UPixel (Pixel reconstruction)
• TMatte	TMatte, die funktionsammlung
• TofBarrel	COSY Tof Barrel Detector
• TofBarrelHit	COSY Tof Barrel Detector: Hits
• TofCal	Base class of COSY Tof Calibration data base
• TofCalData	Data Container of COSY Tof Calibration data base
• TofElastic	2 Particles Kinematic
• TofElasticParticle	Barrel Particle with calib functs
• TofEvent	COSY Tof Event
• TofEventTrk	COSY Tof Event: Tracks
• TofGeoData	Named Container for geometrical constants
• TofHeader	COSY Tof Event Header
• TofHodo	COSY Tof Hodo Detector
• TofHodoCluster	ein TofHodoCluster X- und Y-Lage
• TofHodoHit	COSY Tof Hodo Detector: Hits
• TofHodofeu	COSY Tof Hodofeu Detector
• TofHodofeuCluster	ein TofHodofeuCluster - X- und Y-Lage
• TofHodofeuHit	COSY Tof Hodofeu Detector: Hits
• TofHodofeuPixel	COSY Tof Hodofeu Detector (Cluster give Pixel)
• TofHodoPixel	COSY Tof Hodo-Detector (Cluster give Pixel)
• TofMycroCluster	ein MycroCluster (Ring oder Torte)
• TofMycroPixel	COSY Tof MycroStrip Detector (Cluster give Pixel)
• TofMycroStrip	COSY Tof MycroStrip Detector
• TofMycroStripHit	COSY Tof MycroStrip Detecotr: Hits
• TofNlus	COSY Tof Neutron Detector
• TofNlusHit	COSY Tof Neutron Detector: Hit in one scintillator
• TofObjDouble	Container for collectable doubles
• TofParticle	Particle description, used by TofElastic
• TofQuirl	COSY Tof Quirl Detector
• TofQuirlHit	COSY Tof Quirl Detector: Hits
• TofQuirlPixel	COSY Tof Quirl Detector: Pixel
• TofRing	COSY Tof Ring Detector
• TofRingHit	COSY Tof Ring Detector: Hits
• TofRingPixel	COSY Tof Ring Detector: Pixel
• TofRStart	COSY Tof Mars Detector
• TofRStartHit	COSY Tof Mars Detector: Hits
• TofScaler	COSY Tof TofScaler
• TofScalerData	COSY Tof Scaler: Data
• TofTorte	COSY Tof Torte Detector
• TofTorteHit	COSY Tof Torte Detector: Hits
• TofTrackParticle	info on track and particle properties
• TofTracks	COSY Tof Event => the tracks
• TofVeto	COSY Tof Veto Detectors
• TofVetoHit	COSY Tof Veto Detectors: Hits
• UPixel	COSY Tof Pixel reconstruction
• ZweierKombination	Class used by UPixel (Pixel reconstruction)

---

- containers
- calibration data base
- function collection
- event & track
- helpers

Last update: Thu Jun 7 17:53:31 2001

... What is TofRoot? ...



... let's look at some details ...

# Container Classes

```
TofBarrel : public TObject
```

```
private :
```

```
  Int_t fNofBarrel  
  static TClonesArray fgBarrelHit
```

```
public :
```

```
  TClonesArray* GetHits() const  
  Double_t      GetQdc(Int_t position)  
  Int_t         GetBarrelTotal()  
  TClonesArray* GetHits() const  
  Int_t         MakeBarrelArray(Double_t* carry)  
  void          AddHit(Double_t tdc1, Double_t qdc1, Double_t tdc2, Double_t qdc2, Int_t rss)  
  ...
```

Getters for data in ClonesArray

simple functions

```
TofBarrelHit
```

```
private :
```

```
  Double_t  ftdc1  
  Double_t  ftdc2  
  Double_t  fqdc1  
  Double_t  fqdc2  
  Double_t  fSs
```

```
public :
```

```
  Int_t GetSs()  
  Int_t GetTdc1()  
  ...
```

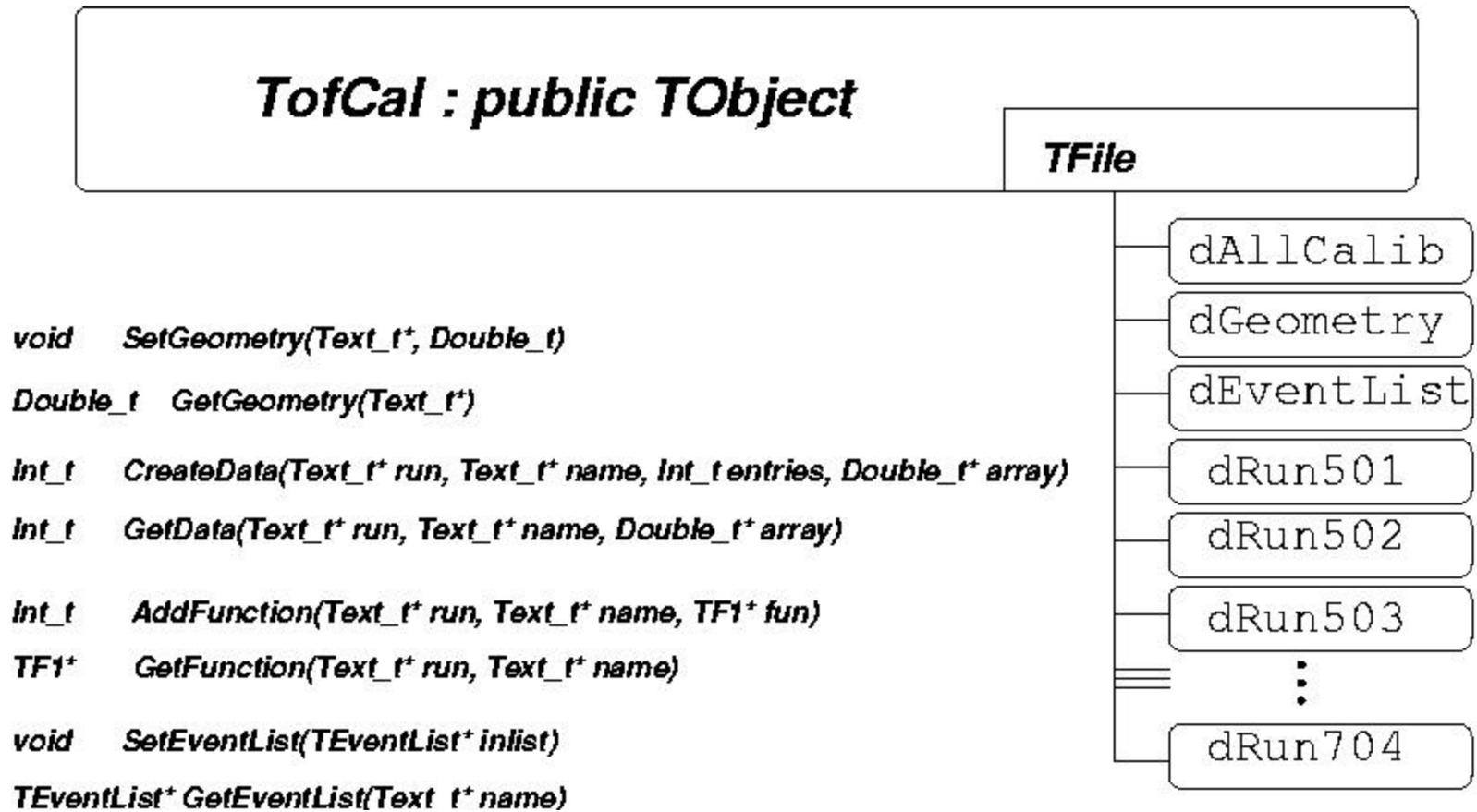
data

only getters

... what about the calibration data base? ...

... let's look at our file format ...

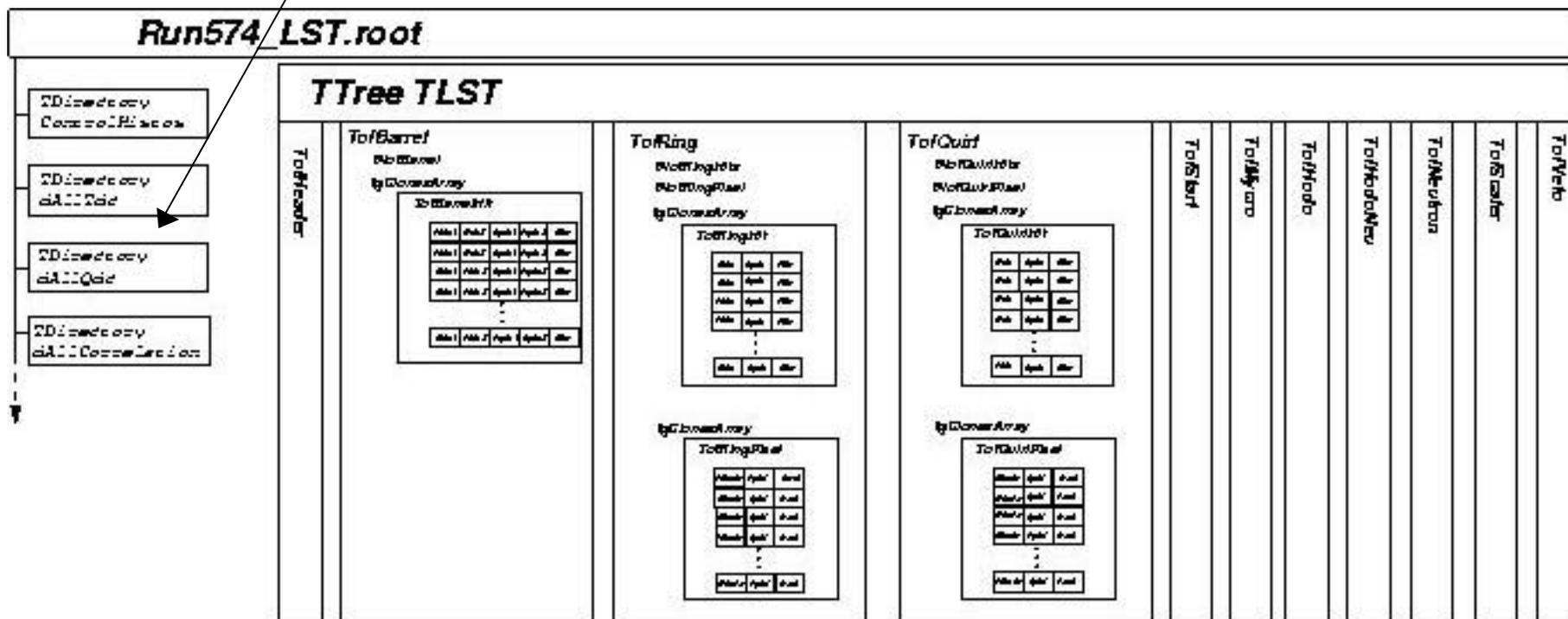
## TofCal – our data base



data are `named doubles` and `named arrays of doubles`  
retrieving data through directory and variable `name`

TDirectories storing control histos

# Data Format



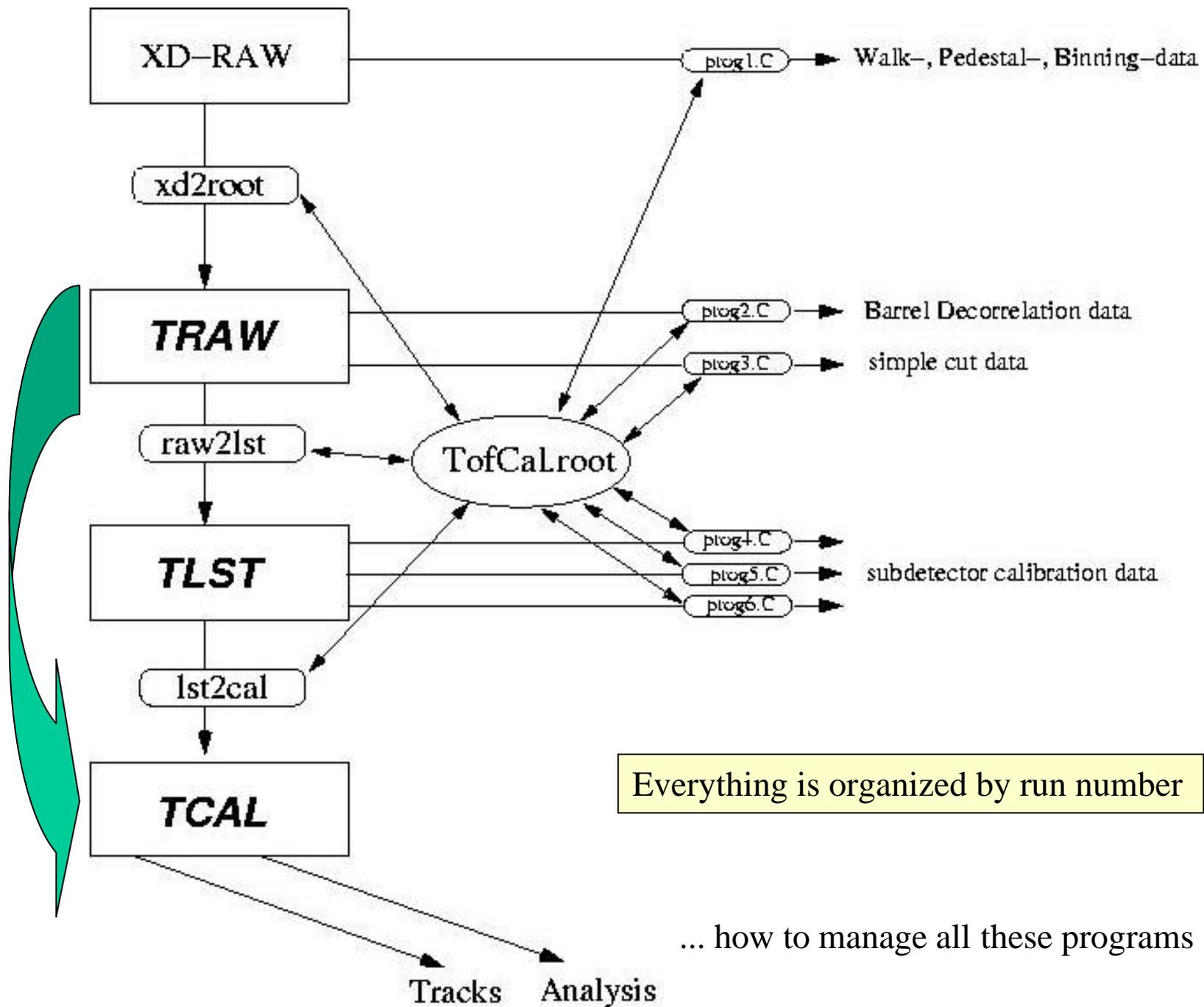
**One** fixed format for **different** levels of calibration :

RAW : raw data

LST : (LiST) – some calibration lists applied

CAL : (CALibrated) – all calibration lists applied

... How does this all work together? ...



... high quality of programs is needed, so ...

## TofRoot : Raw2Cal

```
if [ 666 -eq 1 ] ; then
  RunRAW2LST 551
  RunRAW2LST 552
  RunRAW2LST 553
  RunRAW2LST 554
  RunRAW2LST 555
  RunLeosQuirlRingNus 551 555
  cd ./LST2CALtemp
  RunLST2CALtemp 551
  RunLST2CALtemp 552
  RunLST2CALtemp 553
  RunLST2CALtemp 554
  RunLST2CALtemp 555
fi
#
```

```
#!/bin/bash
raus=0
num=$1

if [ -z $num ]
then
  echo
  echo " du MUSST schon eine runnummer angeben!! "
  raus=1
fi
if [ $raus == 0 ]
#if [ ! $raus ]
then
  inpath="/data_leo/Jan00_rootfiles/R"
  outpath="/data_martin/Jan00_rootfiles/R"
  RAW="_RAW.root"

  infile=$inpath$num$RAW
  echo
  echo ich werde $infile lesen
  LST="_LST.root"

  outfile=$outpath$num$LST
  echo und heraus wird $outfile kommen
  logpath="./ApplyList_logfiles/r"
  AL="_AL.log"
  logfilename=$logpath$num$AL
  rm $logfilename
  echo logfilename wird $logfilename sein

  echo " OK, starte ApplyList_comp fuer run nr. $num "
  ApplyList_comp $num 0 999999999 $infile $outfile > $logfilename
fi
# das ist der ganze befehl :
#           ApplyList_comp 599 0 22210000
../xd2root_msw/Rootfiles/R599_RAW.root outfile.root intree outtree > r599_AL.log
```

All programs called by shell scripts

# TofRoot : set of rules

- use ROOT conventions (kXyz, gXyz, fXyz, ...)
- use TOF conventions (evXyz, gTofXyz, ...)
- robustness is more important than speed
- write comments
- provide debug info (`#ifdef DEBUG ...`) in every official code
- use official includes, use official libs
- nothing hardcoded (get all info out of TofCal-data base)
- run number as the ruling parameter
- avoid parallel development : converge !!

... but people are lazy ...

# TofRoot : set of 'enforcements'

- Code development :  
from a simple macro into the official TofRoot environment

1. Macro using **rootcint**
2. Convert to a full program :

- enforcement  
needed** {
- a) provide control histos in xxxlog.root
  - b) provide output for logfile
  - c) write macros to easily examine histos
  - d) HTML-documentation

3. Implementation into the official TofRoot environment

... this all leads to ...

# TofRoot : set of standards

today (June, 2001) we have “frozen“ :

- ~ 200 functions
- ~ 50 classes
- fixed file format

...this was the talk on TofRoot. Our wishes ...

# Our wishes

- cross referencing of THtml output into different directories using source code which comes from different directories
- better info on important changes (e.g. const in v3.00)  
(search button for old release notes)
- better search method for root homepage

... ok, let's come to an ...

ROOT is very nice, ...

have a lot of fun, ...

have a nice workshop, ....

... ..

**\*\*\* Break \*\*\* segmentation violation**

**THE END**

## What we have learned :

Use C++ features for organisation of software.

Use C++ features for items of general interest (e.g. tools, modules, interfaces,...)

Avoid C++ features when writing calibration and analysis programs (functions) !

example: avoid *while((detector->Next())->Eval())*

(unless you (and all the others) are absolutely sure)

**DOCUMENTATION** (THtml, but also comments and some written notes)

use **ROOT-features** where possible (TMath, Histos, random numbers, ...)