ROOT Data Analysis Framework

In a Nutshell

Danilo Piparo, PH-SFT

These Slides

Are supposed to support our discussion

Are incomplete: they highlight the aspects we believe are relevant for today's discussion

Are not an introductory course about ROOT

CERN

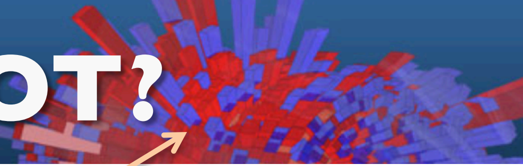
- Several sectors, during your visit mainly in contact with the "Research and Computing" sector (RC)
- Divided in three departments: Experimental Physics, Theoretical Physics and Information Technology

Who are we?

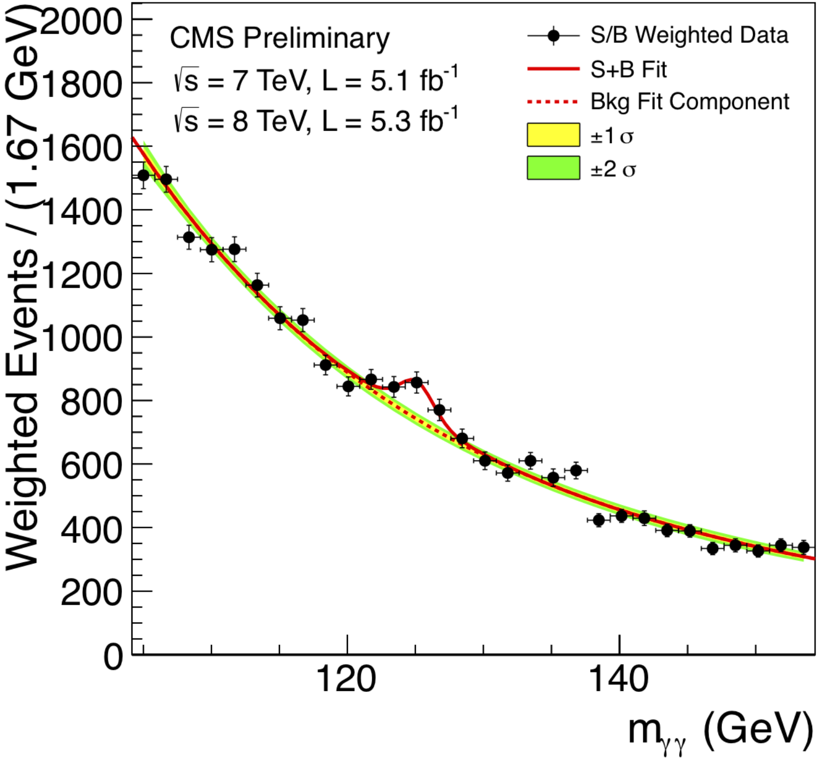
- We are part of the Software Development for Experiments group (EP-SFT)
  - The "software development unit" of CERN's Experimental Physics department
- We develop several tools, among which the ROOT framework
  - O(10.000) users, widely adopted library in HEP (also used in other sciences and industry)
- We are active also in simulation (passage of particles through matter and complex detectors), future experiments setup, distributed file systems, virtualisation, software packaging and provision, education and teaching.

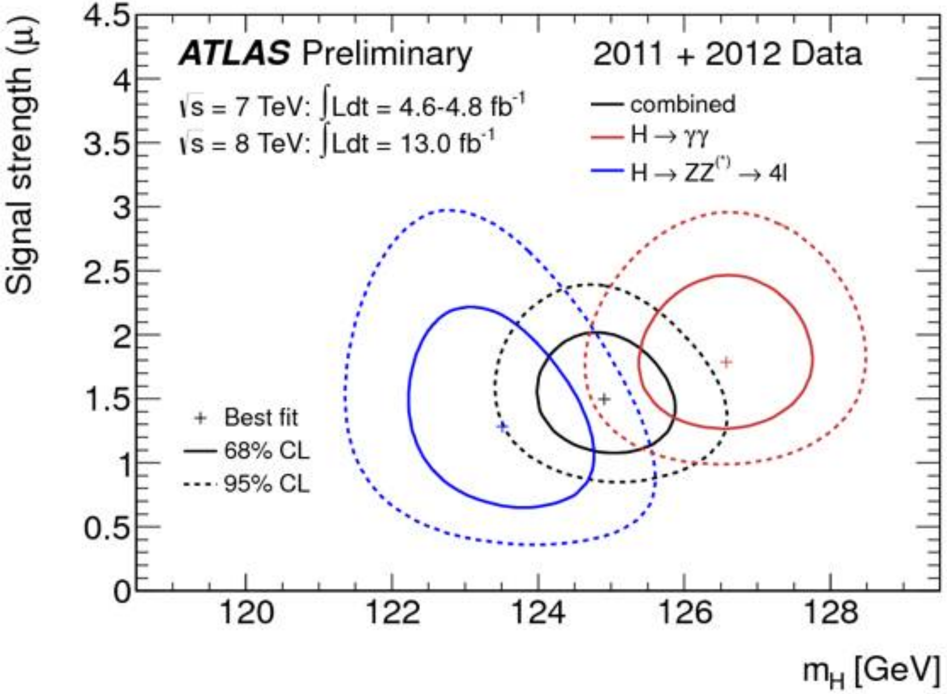
A “Quick Tour” Of ROOT

What can you do with ROOT?



LHC collision in CMS:
event display, also done with ROOT!





ROOT in a Nutshell

ROOT is a software toolkit which provides building blocks for

- Data processing
- Data analysis
- Data visualisation
- Data storage

An Open Source Project

All contributions are warmly welcome!

ROOT is written mainly in C++ (C++11 standard)

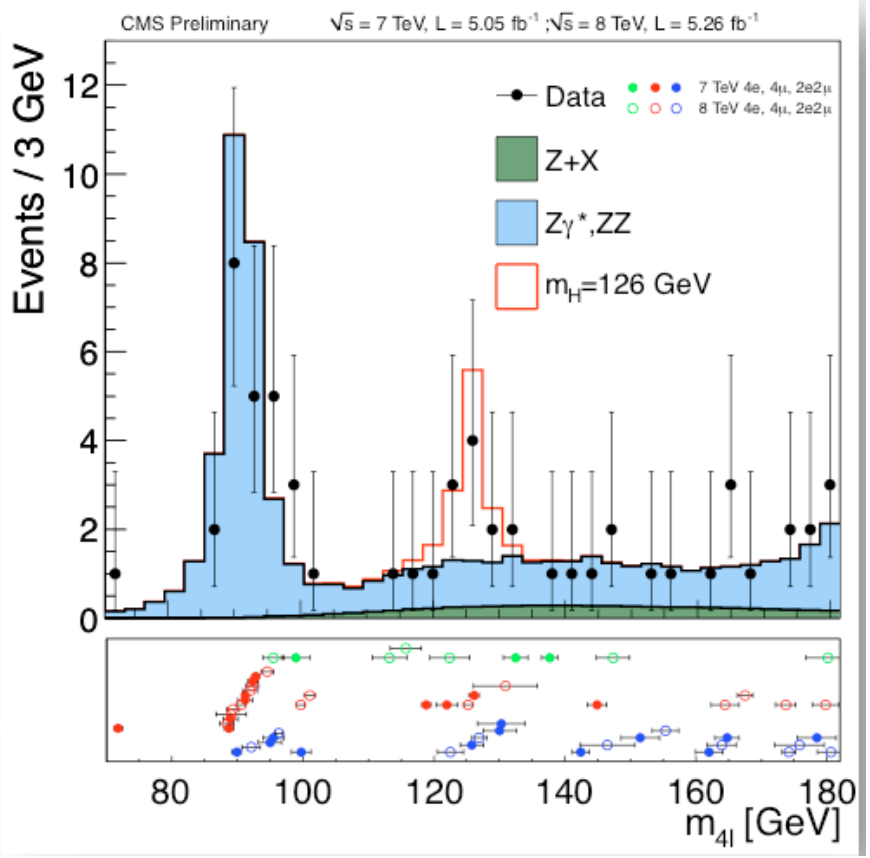
- Bindings for Python and other languages* provided

Adopted in High Energy Physics and other sciences (but also industry)

- ~250 PetaBytes of data in ROOT format on the LHC Computing Grid
- Fits and parameters' estimations for discoveries (e.g. the Higgs)
- Thousands of ROOT plots in scientific publications

* Atm only Python for the 6 series

Prestigious Discoveries with ROOT



Guess who is also using it…

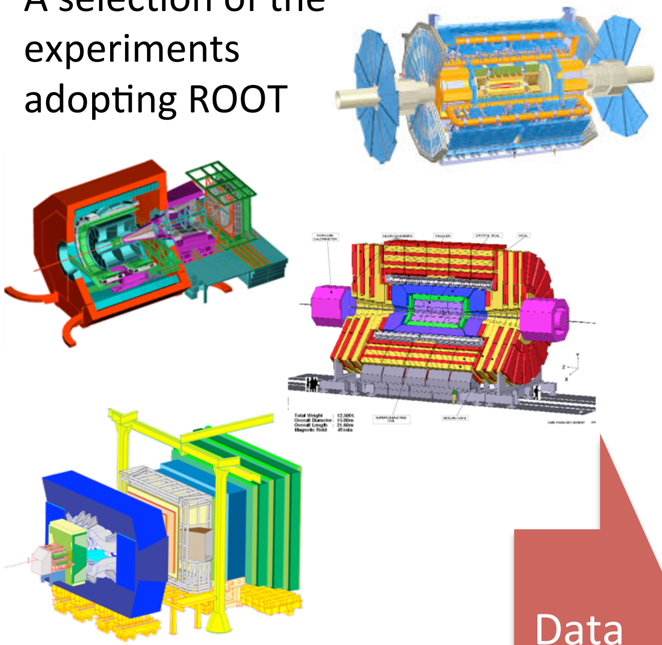
ROOT in a Nutshell

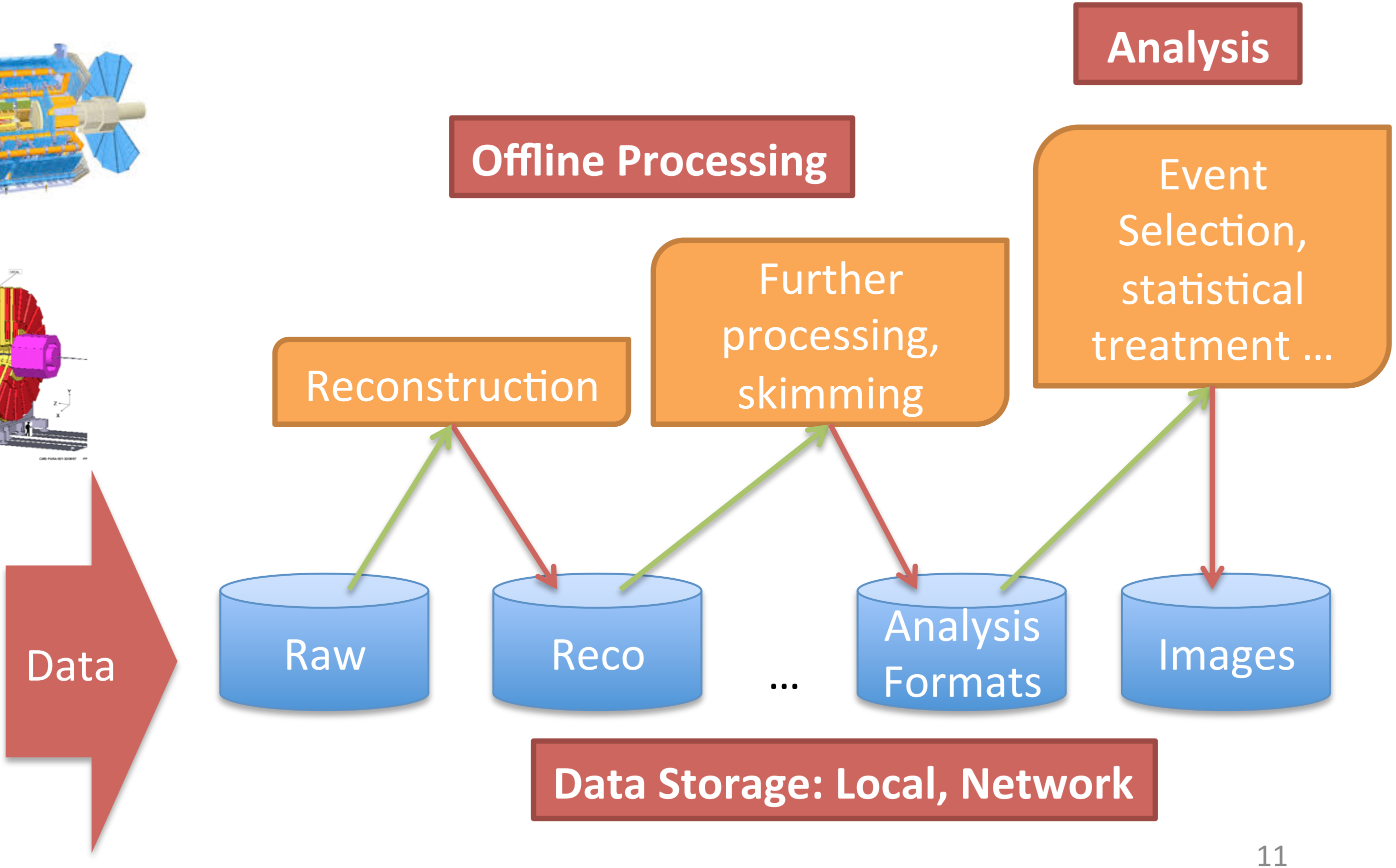
ROOT can be imagined as a family of building blocks for a variety of activities, for example:

- Data analysis: histograms, graphs, trees
- I/O: row-wise, column-wise storage of **any** C++ object
- Statistical tools (RooFit/RooStats): rich modeling and statistical inference
- Math: non trivial functions (e.g. Erf, Bessel), optimised math functions (VDT), linear algebra
- C++ interpretation: fully C++11 (and optionally C++14) compliant
- Machine Learning(TMVA): e.g. Boosted decision trees, neural networks
- And more: HTTP servering, JavaScript visualisation, advanced graphics (2D, 3D, event display).
- PROOF: parallel analysis facility

ROOT Application Domains

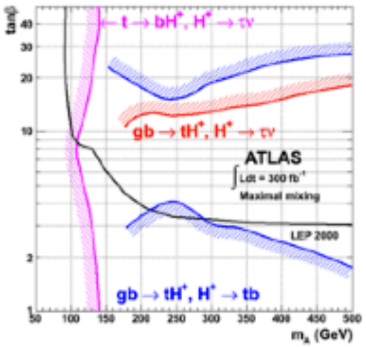
A selection of the experiments adopting ROOT

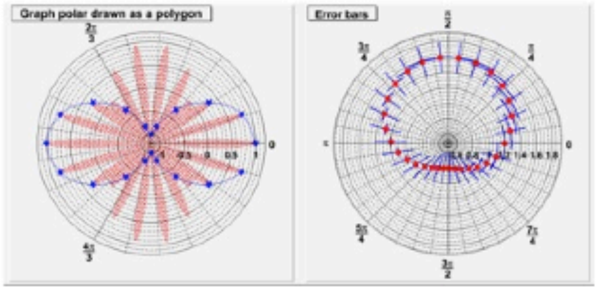


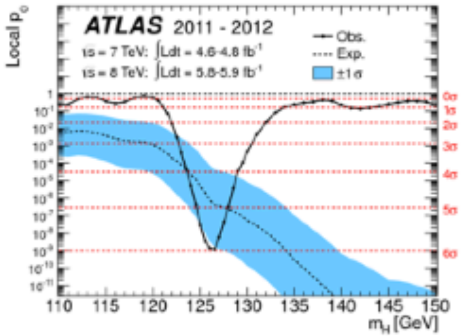


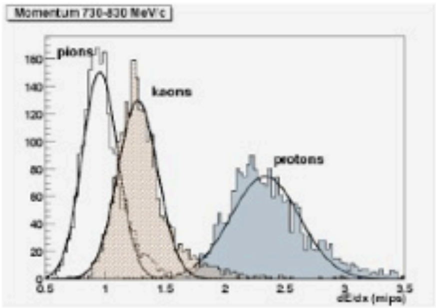
Graphics In ROOT

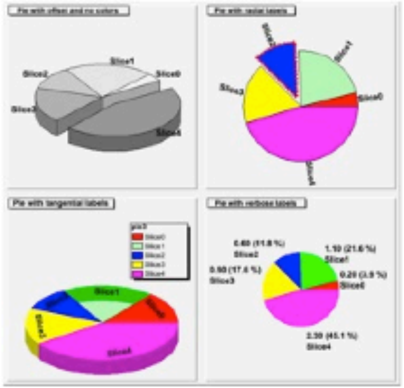
Many formats for data analysis, and not only, plots

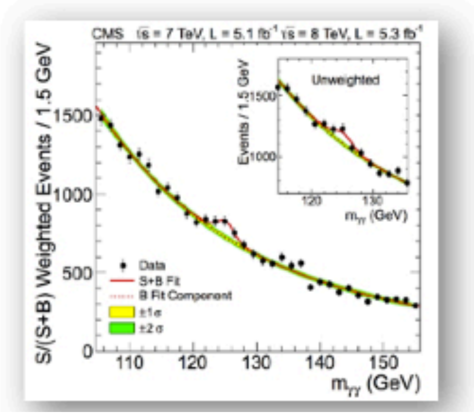










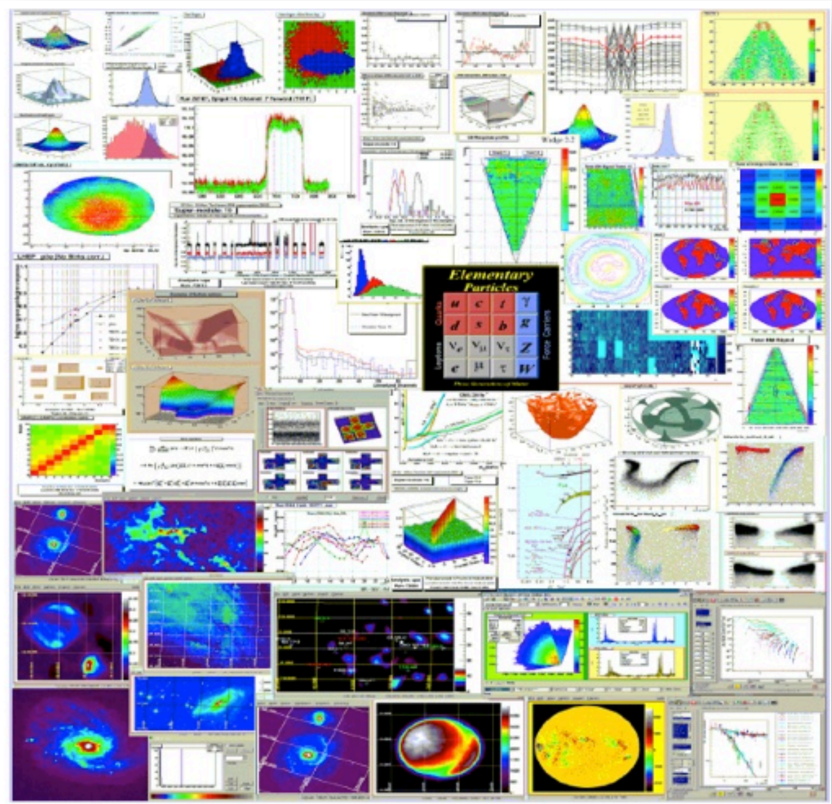


2D Graphics

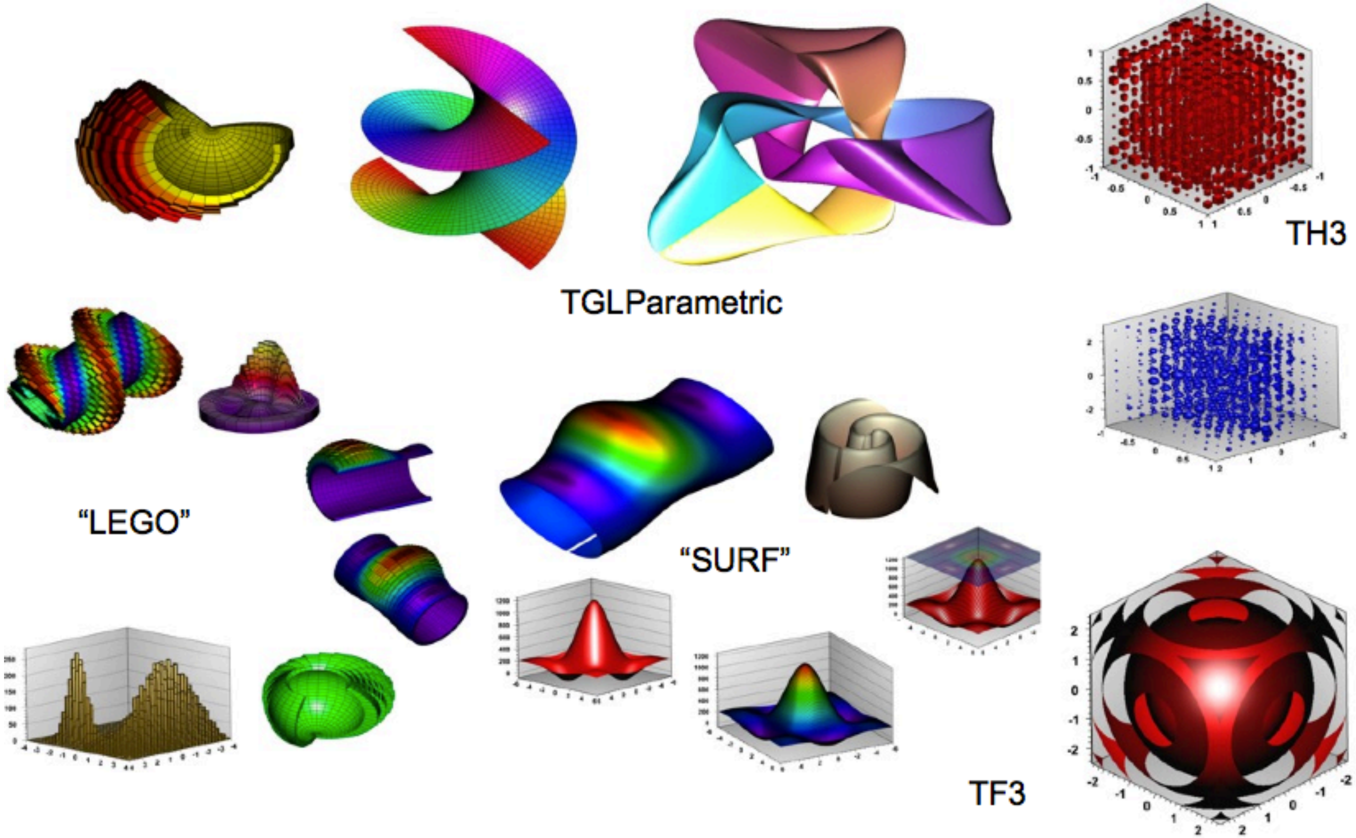
New functionalities added at every new release

Always requests for new style of plots

Can save graphics in many formats: *ps, pdf, svg, jpeg, LaTex, png, c, root*



3D Graphics



At the Core

ROOT is very modular but there is synergy among its components

Slim Core ("orchestration framework") containing

- Management of global configuration settings
- A type system: classes, functions, methods, enumerators known to ROOT (highly non C++ standard!)
- Plugin registration mechanism: no need to load all libraries at start-up (link), just if needed, at runtime
- Seamless interaction with interpreter functionalities

Interpreter

ROOT is shipped with an interpreter, CLING

- C++ interpretation: highly non trivial and not foreseen by the language!
- One of its kind: Just In Time (JIT) compilation
- A C++ interactive shell.

```
$ root -b
root [0] 3 * 3
(const int)9
```

Can interpret "macros" (non compiled programs)

- Rapid prototyping possible

ROOT provides also Python bindings:

- Can use Python interpreter directly after a simple *import ROOT*
- Possible to "mix" the two languages (see more in the following slides!)

Interpreter: interactive shell

~> root

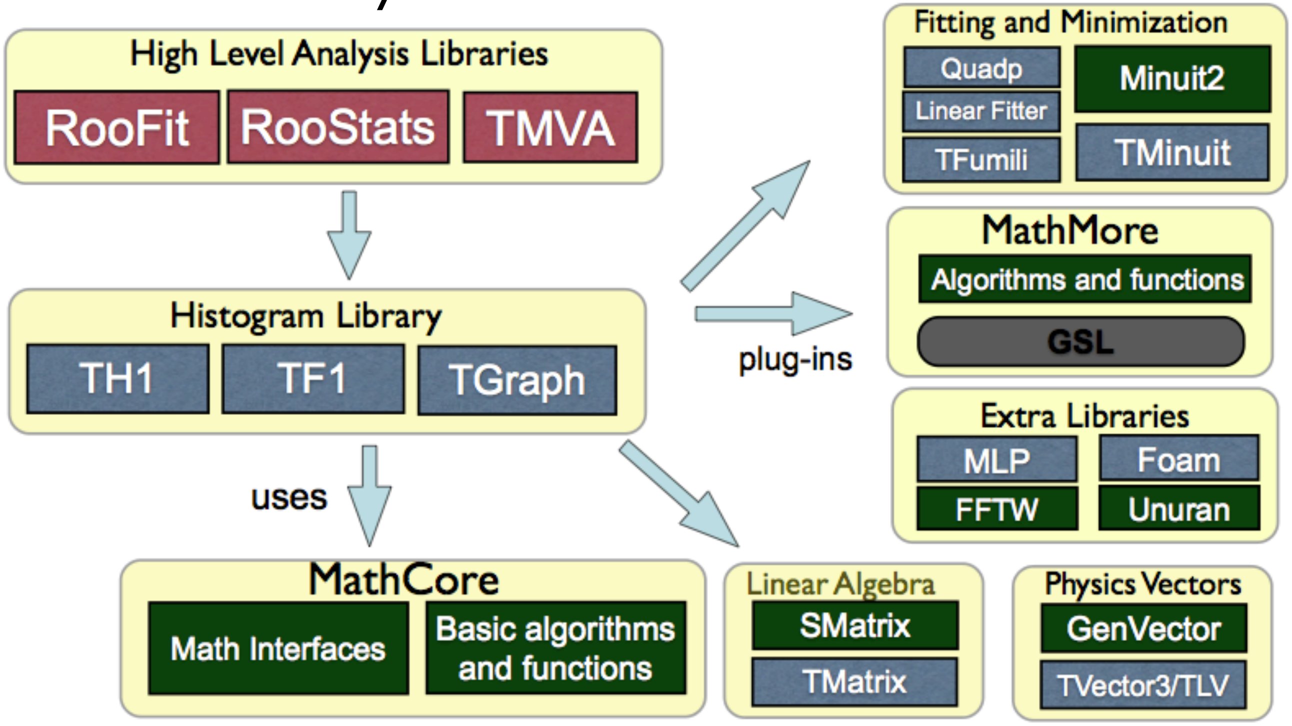
```
~-> root
root [0] i = 4
(int) 4
root [1] TH1F h("myHisto","Histogram Title",64,-4,4)
(TH1F &) Name: myHisto Title: Histogram Title NbinsX: 64
root [2] h.FillRandom("gaus")
root [3] h.GetSkewness()
(Double_t) 0.00699916
…
```

And Python too…

```
python
Python 2.7.6 (default, Sep  9 2014, 15:04:36)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.39)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import ROOT
>>> h = ROOT.TH1F("h","h",64,-4,4)
>>> h.Draw()
```

ROOT Math/Stats Libraries

ROOT provides a reach set of mathematical libraries and tools needed for sophisticated statistical data analysis



Persistency (I/O)

ROOT offers the possibility to write C++ objects into files

- Exceptional: impossible with C++ alone!
- Used for tens of petabytes/year rates of LHC detectors.
- Of course simple, flat n-tuples are supported (in fact, complex objects are automatically decomposed in simple columns)

Achieved with serialization of the objects using the reflection capabilities, ultimately provided by the interpreter

- Raw and column-wise streaming

Cornerstone for storage of experimental data

As simple as this for ROOT objects: one method - *TObject::Write*

And for non ROOT objects: *myFile.ReadObject(myObjPointer, "objName")*

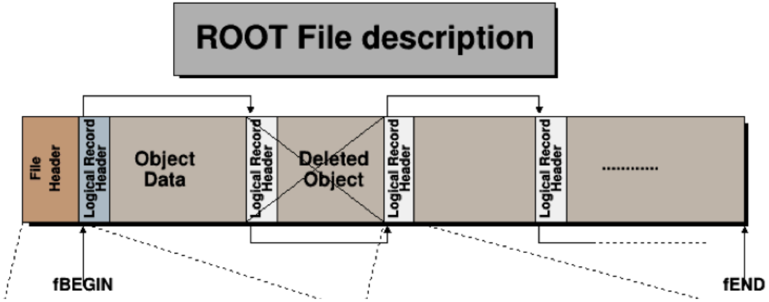
The ROOT file Format

The ROOT file format is the format of HEP

- Some sporadic exception in very specialised and controlled environments (e.g. Raw data of LHCb)

This is its complete description.

Byte Range	Record Name	Description
1->4	"root"	Root file identifier
5->8	fVersion	File format version
9->12	fBEGIN	Pointer to first data record
13->16 [13->20]	fEND	Pointer to first free word at the EOF
17->20 [21->28]	fSeekFree	Pointer to FREE data record
21->24 [29->32]	fNbytesFree	Number of bytes in FREE data record
25->28 [33->36]	nfree	Number of free data records
29->32 [37->40]	fNbytesName	Number of bytes in TNamed at creation time
33->33 [41->41]	fUnits	Number of bytes for file pointers
34->37 [42->45]	fCompress	Compression level and algorithm
38->41 [46->53]	fSeekInfo	Pointer to TStreamerInfo record
42->45 [54->57]	fNbytesInfo	Number of bytes in TStreamerInfo record
46->63 [58->75]	fUUID	Universal Unique ID



Other ROOT Features

Geometry Toolkit

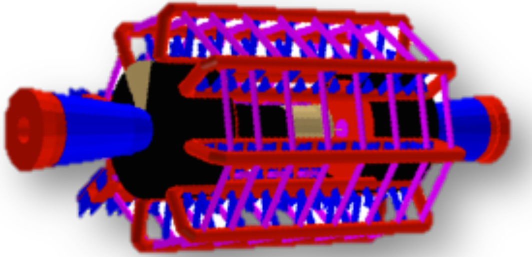
- Represent geometries as complex as LHC detectors

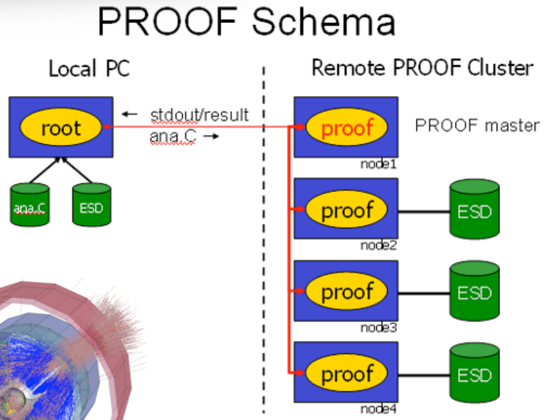
Event Display (EVE)

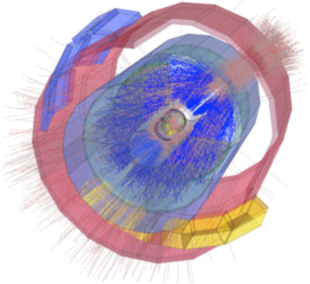
- Visualise particles collisions within detectors

PROOF: Parallel ROOT Facility

- Multi-process approach to parallelism
- A system to run ROOT queries in parallel on a large number of distributed computers
- Proof-lite: does not need a farm, uses all the cores on a desktop machine







Jupyter

- Allow physicscists to express analyses in form of notebooks
  - Code, plots and explanations in the same document
  - Reproducibility, easy to share
- We want ROOT well integrated with Jupyter notebooks
  - ROOT Kernel (mainly C++)
  - Integration of ROOT Python modules in Python notebooks
- We want to offer "Data Mining as a Service" at CERN
  - Jupyterhub
  - ROOT, R, Python, …
  - CERN authentication, storage, batch: combine Jupyterhub with the CERN services portfolio (More details tomorrow!)

Notebooks, Documentation, Education

From the ROOT website:

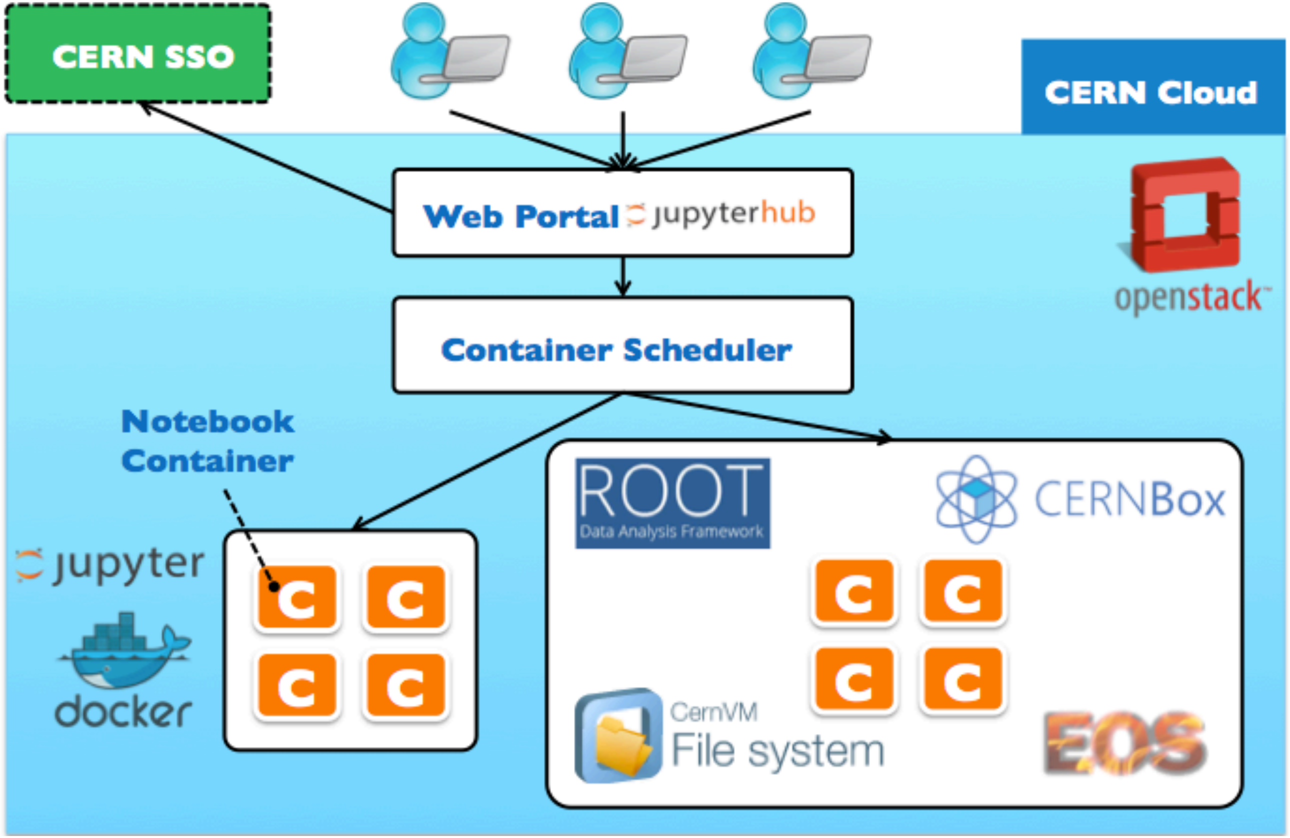
- https://root.cern.ch/howtos#Jupyter%20Notebooks
- https://root.cern.ch/code-examples#rootbooks

From the ROOT Summer Students' tutorial

- https://indico.cern.ch/event/395198
- 50 participants with various technical skills (age 22-25) in this session, a single powerful server, jupyterhub, ROOT

  → A success!

Jupyterhub at CERN



Notebooks on the Agenda and Demo

- Indico: CERN conference management system
- Jupyter notebooks rendering

BACKUP

The ROOT Prompt

C++ is a compiled language

- A compiler is used to translate source code into machine instructions

ROOT provides a C++ **interpreter**

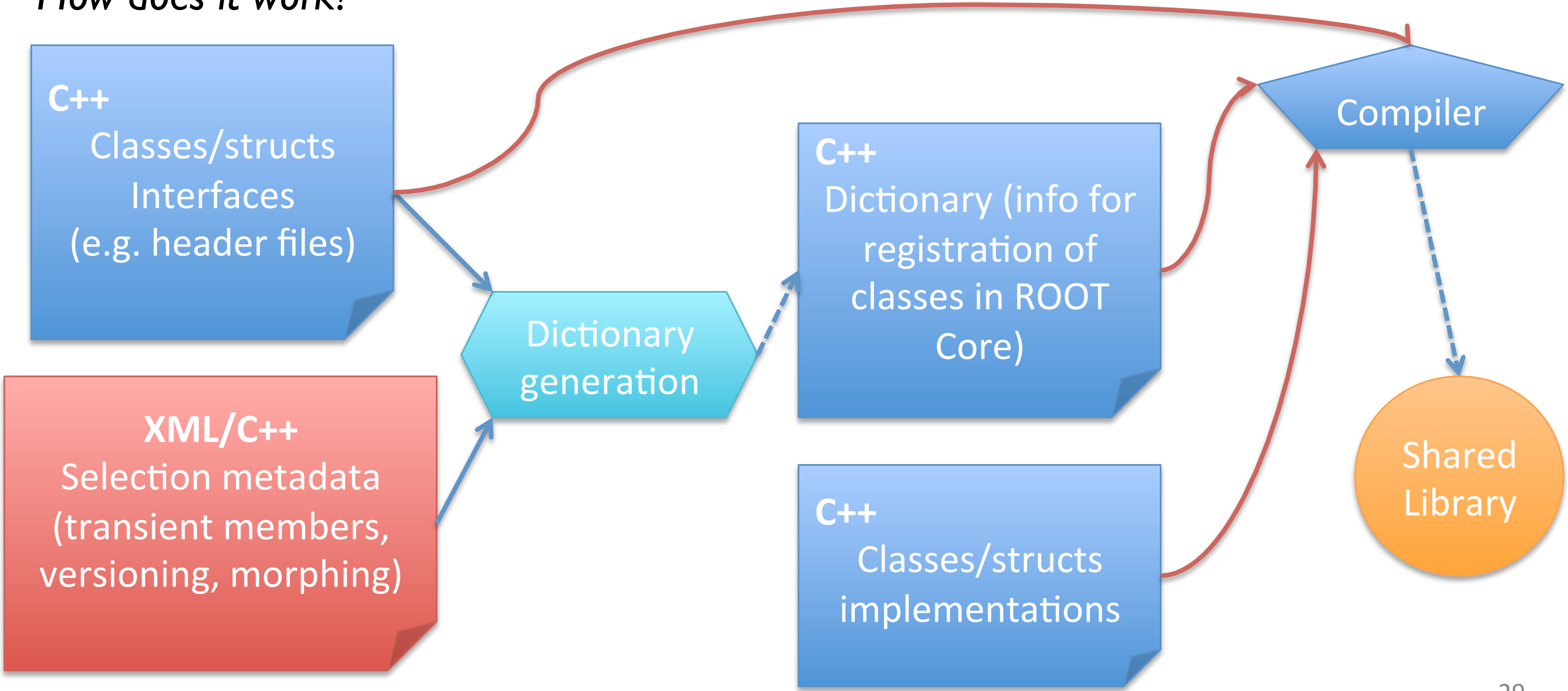
- Interactive C++, w/o the need of a compiler, like Python, Ruby, Haskell …
- Allows reflection (inspect at runtime layout of classes)
- Can be booted with the command:

```
root
```

- The interactive shell is also called "ROOT prompt" or "ROOT interactive prompt"

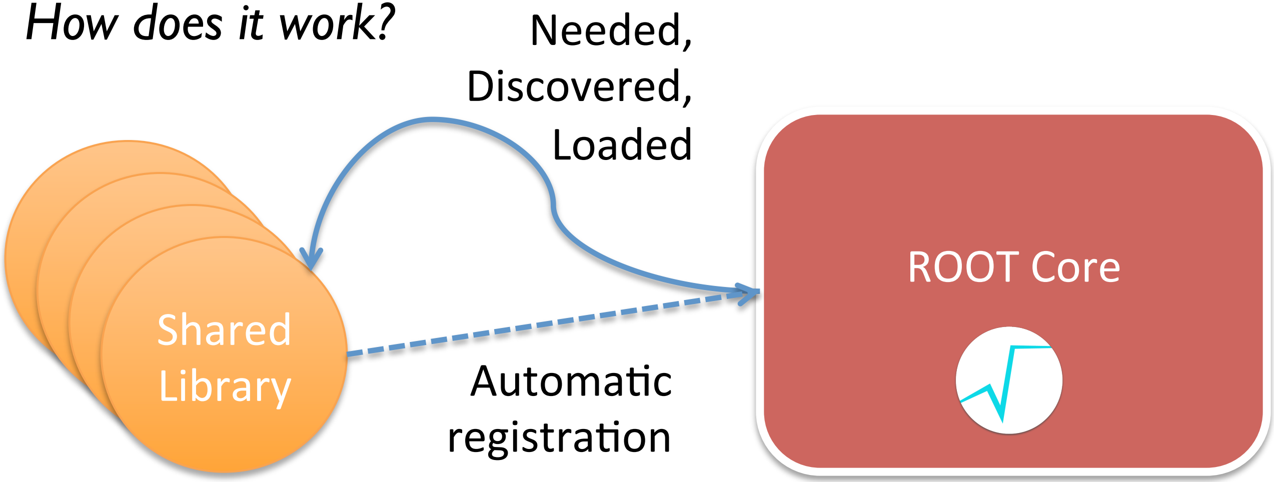
Persistency (I/O)

How does it work?



Persistency (I/O)

How does it work?



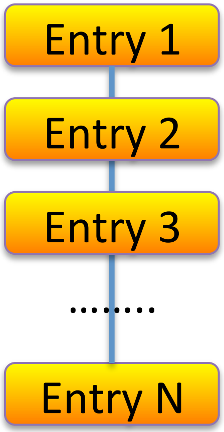
Now ROOT “knows” how to serialise the instances implemented in the library (series of data members, type, transiency) and to write them on disk in row or column format.

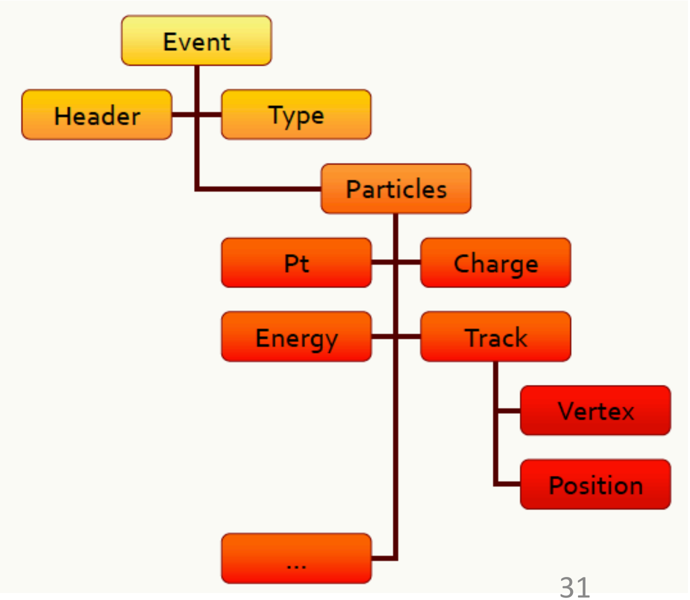
Trees

- The TTree is the data structure ROOT provides to store large quantities of same types objects
- Organised in branches, each one holding objects: partial reads possible
- Organised in independent events, e.g. collision events
- Efficient disk space usage, optimised I/O runtime

x	y	z
-1.10228	-1.79939	4.452822
1.867178	-0.59662	3.842313
-0.52418	1.868521	3.766139
-0.38061	0.969128	1.084074
0.552454	-0.21231	0.350281
-0.18495	1.187305	1.443902
0.205643	-0.77015	0.635417
1.079222	-0.32739	1.271904
-0.27492	-1.72143	3.038899
2.047779	-0.06268	4.197329
-0.45868	-1.44322	2.293266
0.304731	-0.88464	0.875442
-0.71234	-0.22239	0.556881
-0.27187	1.181767	1.470484
0.886202	-0.65411	1.213209
-2.03555	0.527648	4.421883
-1.45905	-0.464	2.344113
1.230661	-0.00565	1.514559
		3.562347

LEP style flat n-tuples evolved in more efficient trees (fast access, read ahead)





Preemptive Trouble Shooting

? *What could be the advantage of learning this software technology?*

! **1.** Batteries included: you have all the tools to process, store, analyse and visualise data in one single kit.

! **2.** You join a huge community, O(10^4) users + a very supportive team of core developers

? *Why C++ and not a scripting language?!*

! Performance. Support for languages like Python

? *Why prompt and libraries instead of a GUI?*

! ROOT is a programming framework, not an office suite.

Trees

Does it scale? Yes – it does

- Files can be as big as a few gigabytes (10? 20?)
- LHC experiments write and read ROOT files
- 250PB of data registered on the WLCG in ROOT format: much more around (university clusters, private ntuples, non LHC experiments!)

From Ian's presentation:

- A full genome: 200GB
- Genes (5%): 10GB
- A "study group": 100 individuals

From Nature:

- 4 Nucleobases
- Homo Sapiens: 3 billion bases / genome
- Homo Sapiens: 25k Genes

The ROOT file Format

Provided and desireble features

- Simple: “keys” on disk, easily described and open source
- Demonstrated reliability: easy recovery, self describing (description of data stored with the data)
- Efficient: compressed data, options available for compression algorithms
- Remote reading over the network over already available protocols
- Backward/forward compatibility