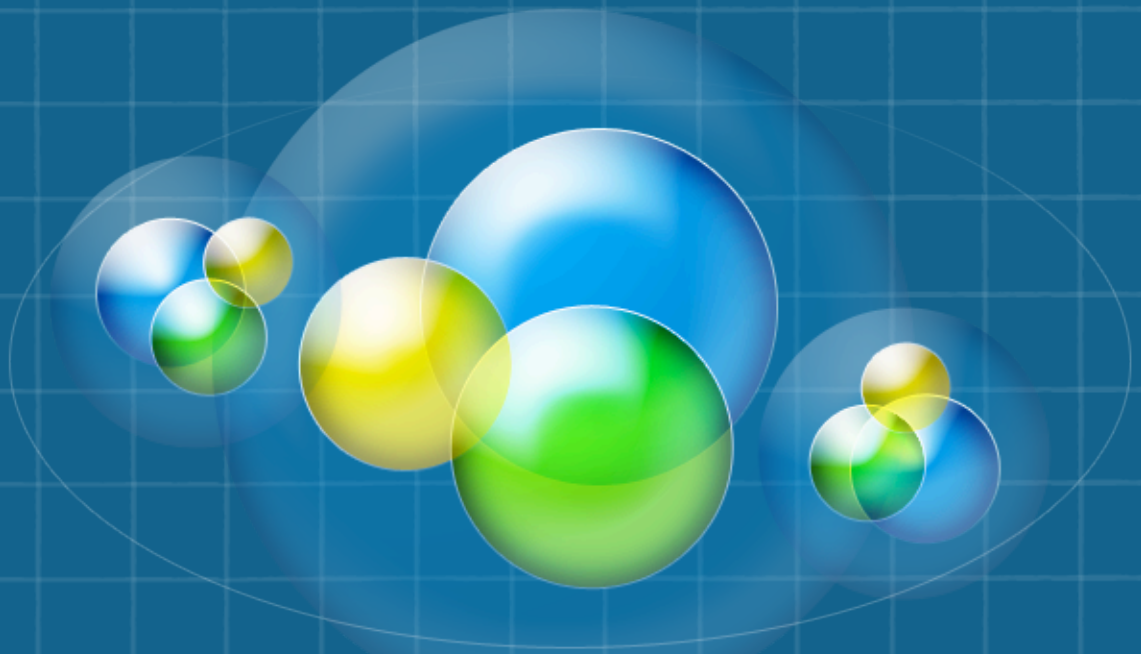
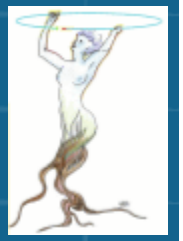


ROOT: Status and Roadmap

René Brun /SFT

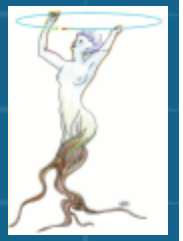
30 September 2009

Plan



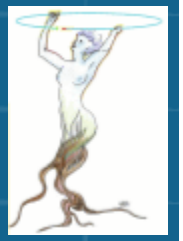
- A bit of history
- Project structure and people
- Packages status and plans
- Releases and Users
- Challenges and worries
- ROOT and SFT : wishes

Original Objectives in 95



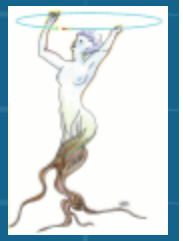
- Continue the tradition started with CERNLIB to develop packages targeted to the needs of our experiments and with a long life time.
- Eg hbook (36 years), geant3 (28), paw(25): still in use!
- Focus on data analysis (PAW in the oo world)
- Open Source model to get contributions and feedback as soon as possible.
- First ROOT prototype for NA49 in 1995, rapidly followed by many more adopters. Bottom-up (user-driven) instead of top-down (committee) approach. Users workshops.

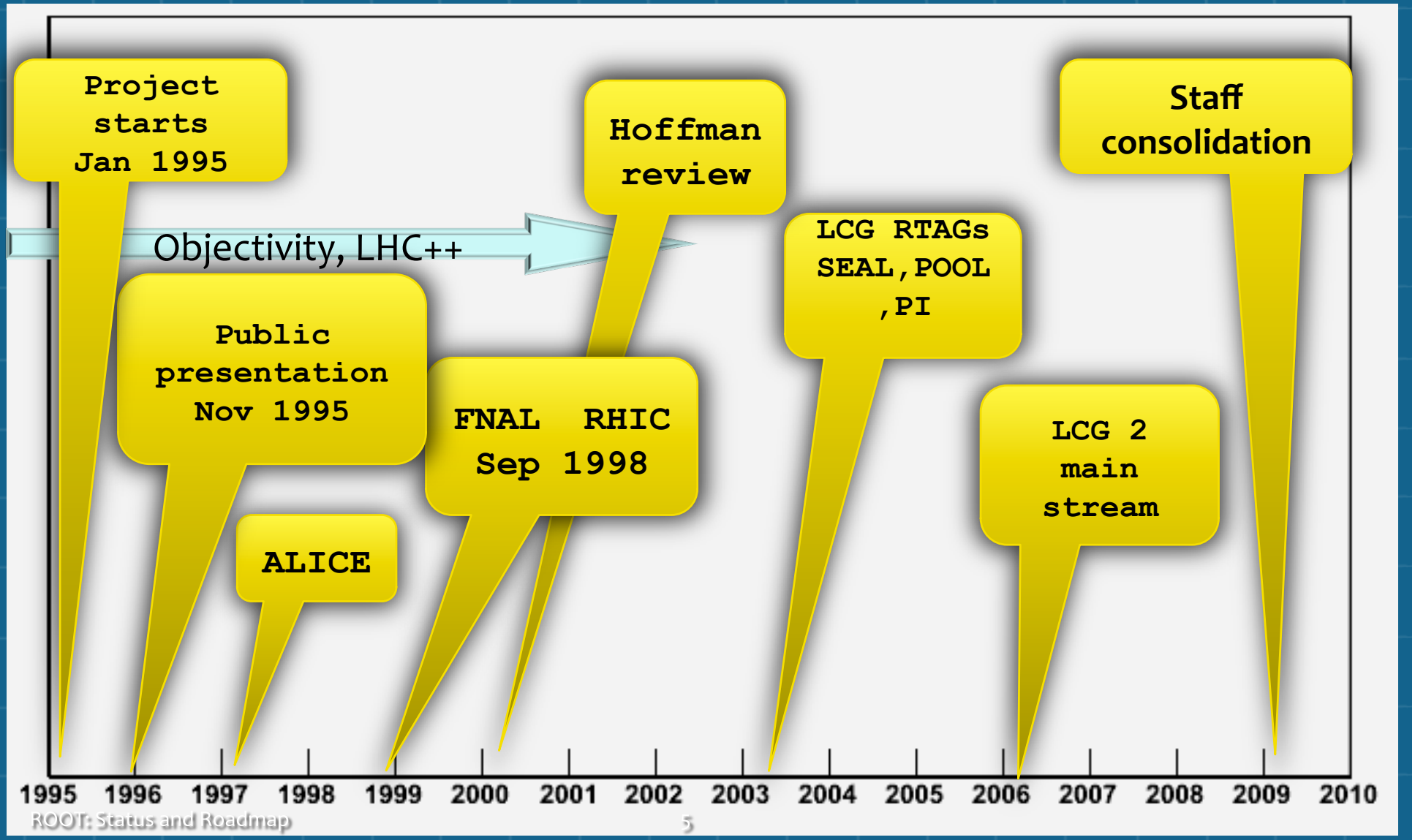
Facts



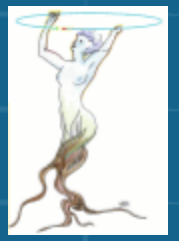
- It became rapidly clear that a general solution for object persistency was a key issue.
- In particular when FNAL, RHIC (and soon after Babar) realized in 97 that the official solution based on Objectivity was not going to work. They needed a concrete solution for 2001!
- It was vital for us to have this early experience with pre-LHC big experiments.
- We gradually implemented a general I/O system capable of writing any C++ object model and support its evolution in time.
- With the growing success of this approach, more developments were requested by users and gradually implemented when additional manpower resources became available.

Development Context



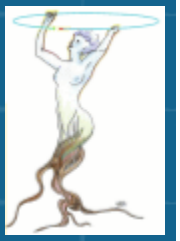


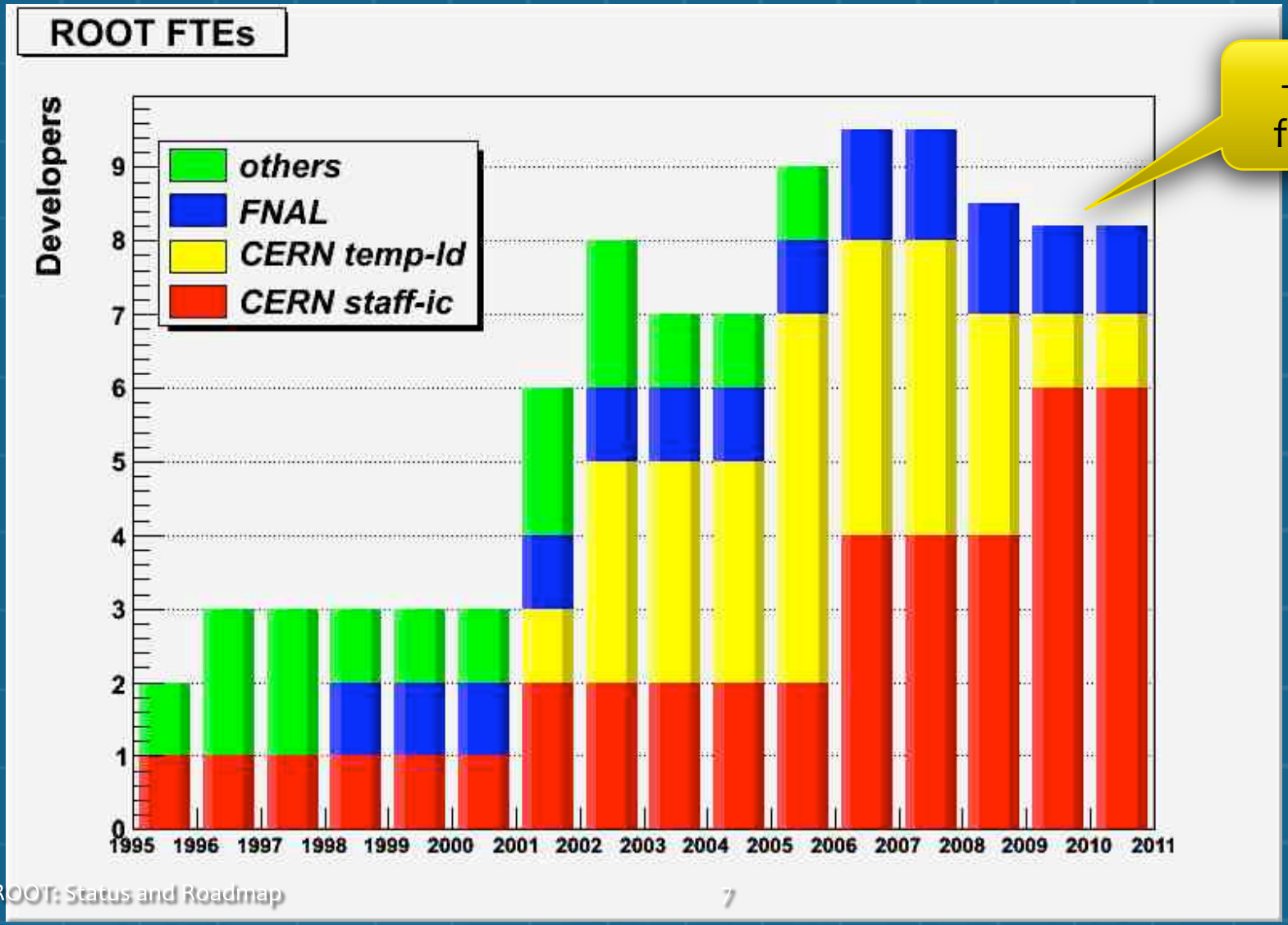
Shaping the structure



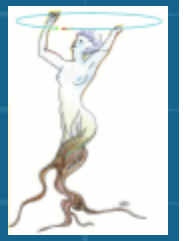
- Started with a monolithic (PAW-like) structure, we quickly moved around 2002 to a more dynamic and flexible project structure with a CORE and optional libraries dynamically linked via a plug-in manager.
- CERN manpower works on the CORE and most used libs. Many outside contributions on the optional libs, but still being careful for the medium and long term support. All libs depend on CORE for the dictionaries (trying to get CORE as stable as possible)
- Put in place a user support structure (forum, Savannah,..) and a test environment with nightly builds (with SPI).
- Several LCG reviews since 2004.

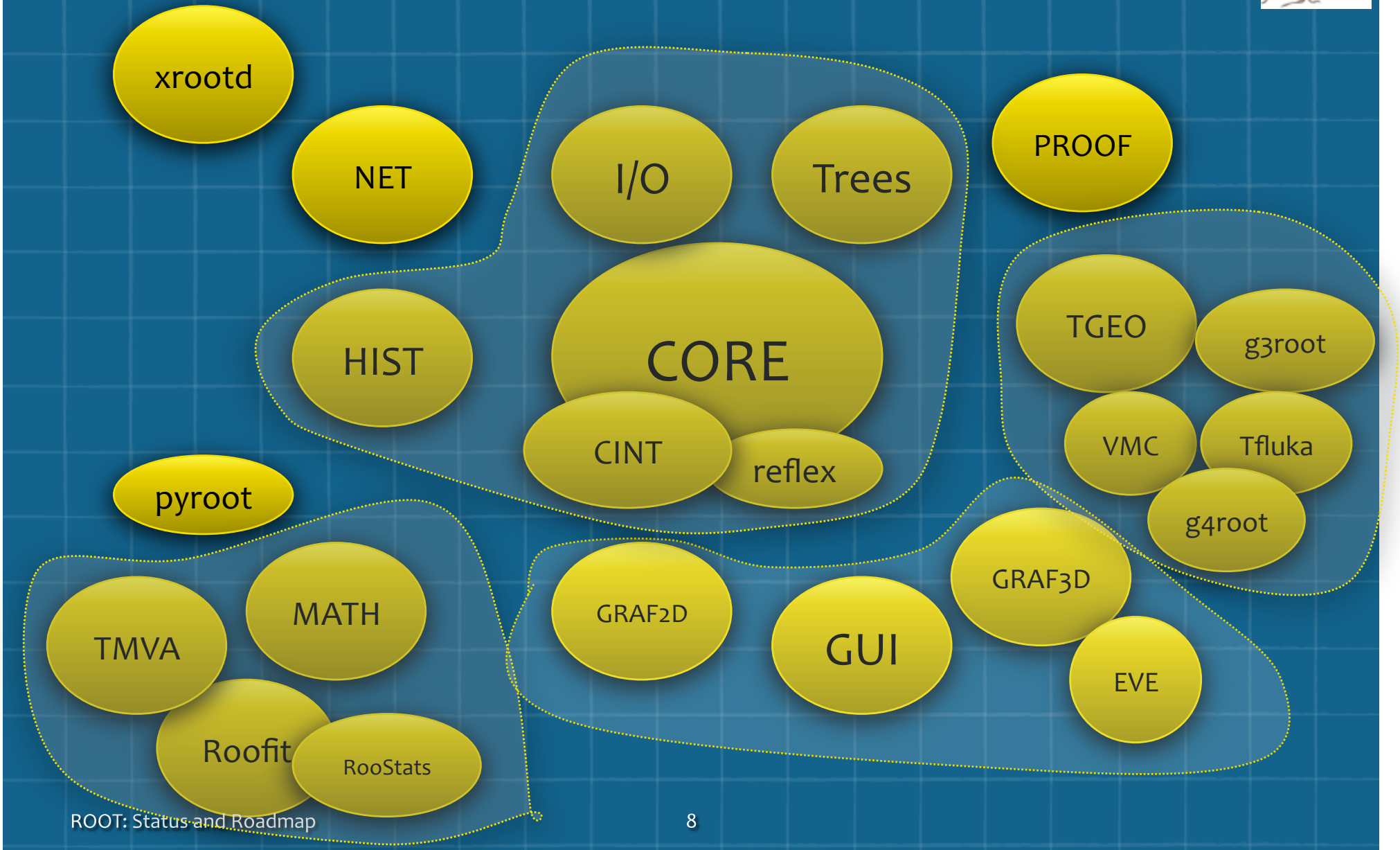
ROOT main developers



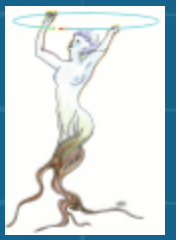


Project Structure





People



G.Ganis

xrootd

F.Rademakers

PROOF

NET

I/O

Trees

A.Gheata

R.Brun

P.Canal

TGEO

g3root

HIST

CORE

A.Naumann

W.Lavrijsen

VMC

Tfluka

CINT

reflex

pyroot

L.Moneta

g4root

GRAF3D

MATH

GRAF2D

GUI

TMVA

EVE

Roofit

RooStats

O.Couet

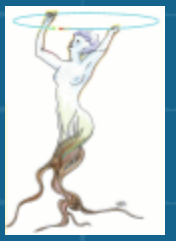
B.Bellenot

M.Tadel

A+x+y

W.Verkerke

People (many more)



Only 23 bubbles shown in this picture.
Many more people have contributed to the development of the other 85 smaller bubbles.

We have 177 contributors in our CREDITS file

G.Ganis

PROOF

Trees

P.Canal

A.Gheata

TGEO

g3root

VMC

Tfluka

g4root

A.Naumann

CINT

reflex

W.Lavrijsen

pyroot

L.Moneta

MATH

TMVA

Roofit

RooStats

A+x+y

W.Verkerke

GRAF2D

O.Couet

GUI

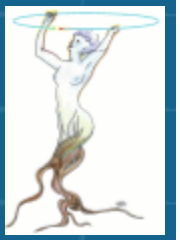
B.Bellenot

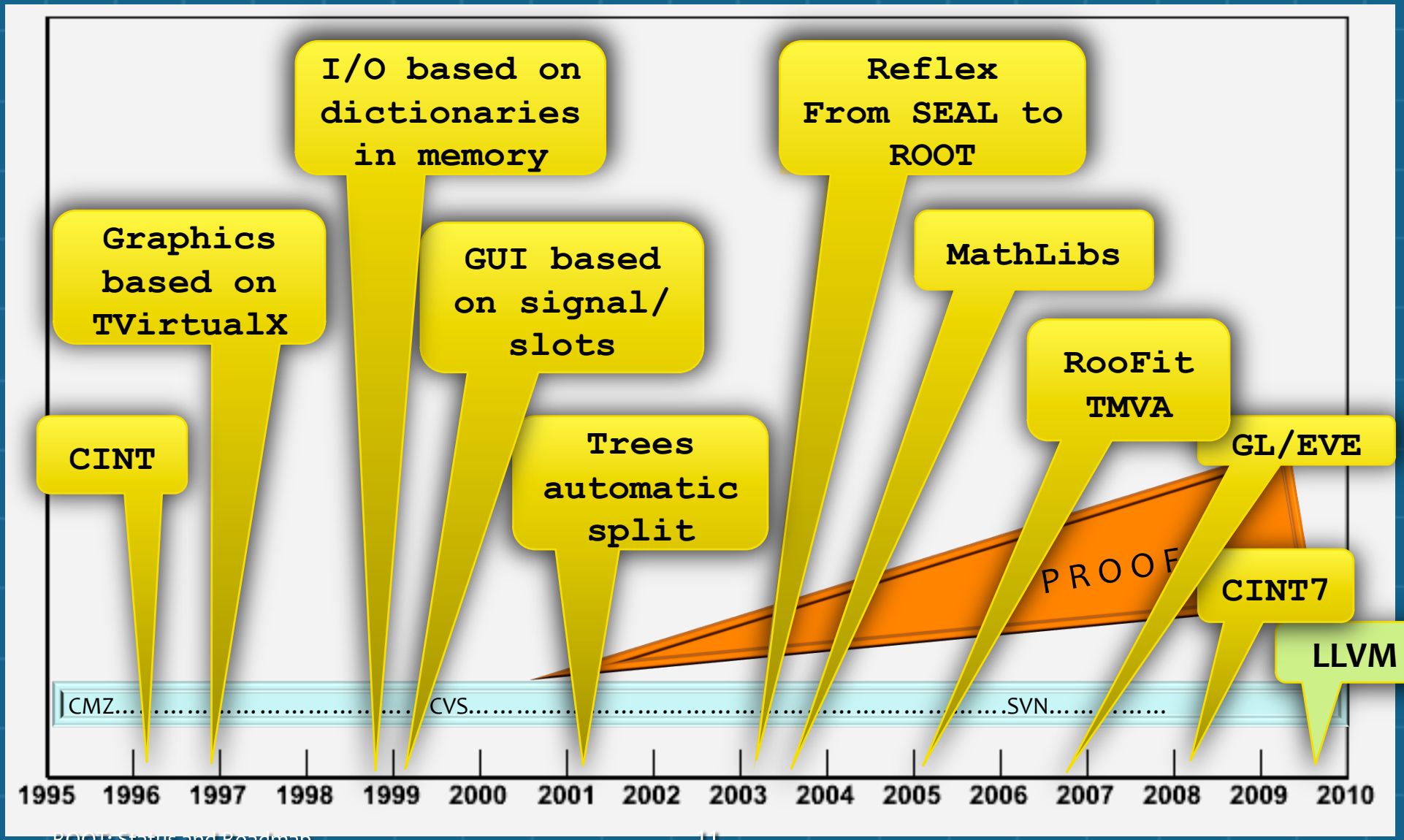
GRAF3D

EVE

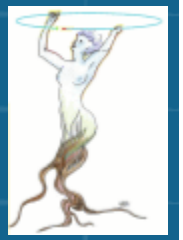
M.Tadel

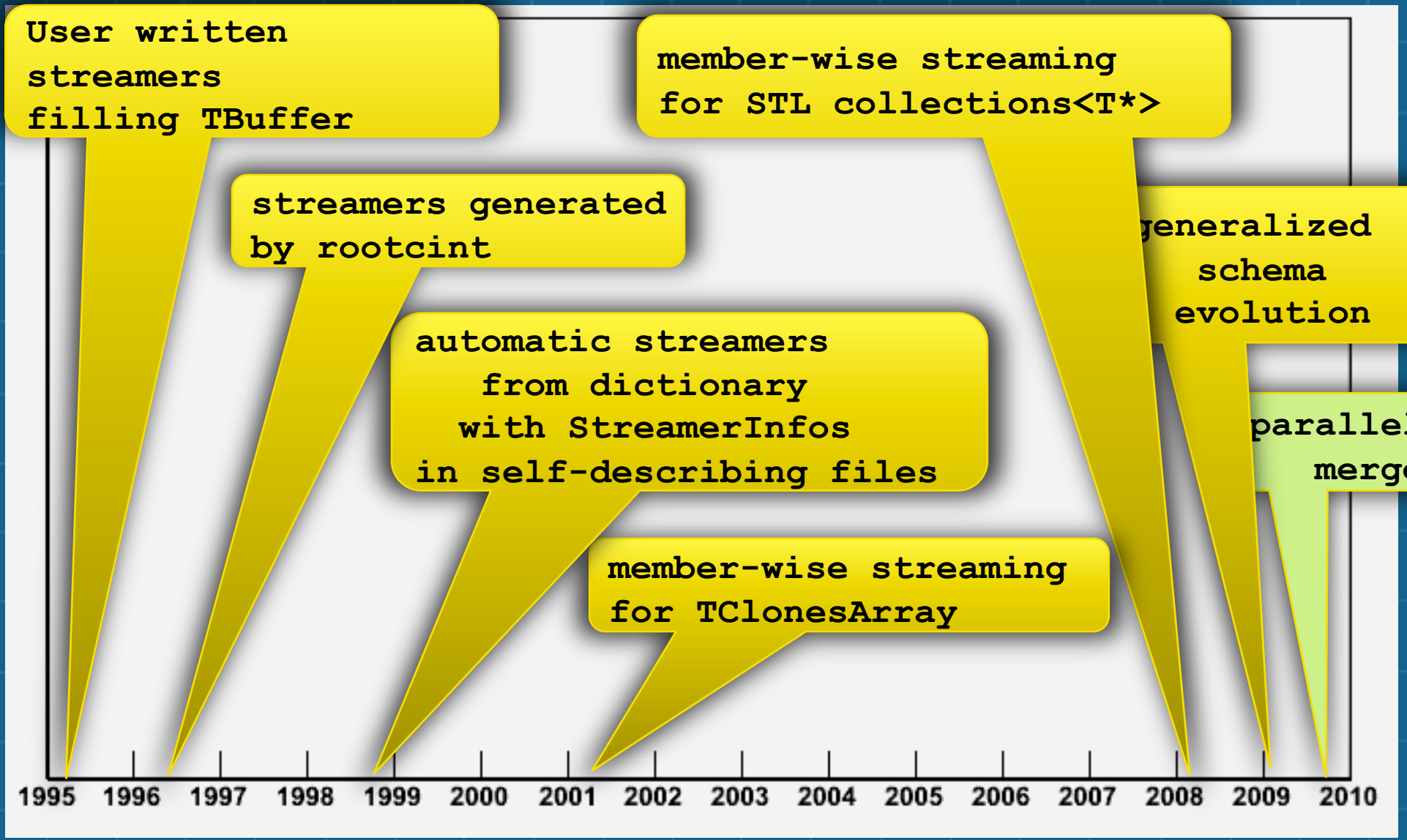
Major Technical Steps



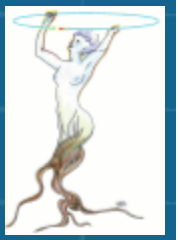


Input/Output: Major Steps





ROOT Math/Stat Libraries



The new CERNLIB/ MATHLIB

With better algorithms

High Level Analysis Libraries
- RooStats
- RooFit
- TMVA

Fitting and Minimization
- Quadp
- Linear Fitter
- TFumili
- Minuit2
- TMinuit

Statistical Libraries
- Limit classes
- MLP

Extra Libraries
- Unuran
- FFTW
- Spectrum
- Foam

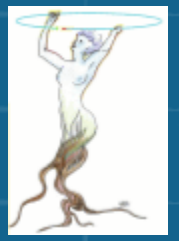
Linear Algebra
- SMatrix
- TMatrix

Physics Vectors
- GenVector
- TVector3/TLV

MathCore
- Math Interfaces
- Fitting Classes
- Basic algorithms
- TComplex
- Math functions
- TRandom

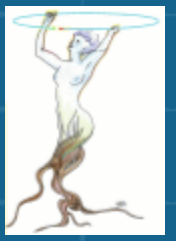
MathMore
- Random Numbers
- Extra algorithms
- Extra Math functions
- GSL

RooFit/ RooStats



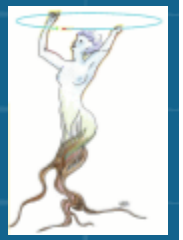
- The original Babar RooFit package has been considerably extended by Wouter Verkerke (now more than 120000 lines of code)
- RooFit is the base for the new RooStats package jointly developed by ATLAS (Kyle Cranmer et al) , CMS(G.Schott et al) and the ROOT team.
- RooStats is a package providing common high level tools for statistical calculation (for signal discovery, limit calculation and analysis combinations) . Tutorials for both ATLAS and CMS together are being organized in October. Regular reviews by the CMS and ATLAS statistics specialists.
- Very nice cooperation

CINT → LLVM



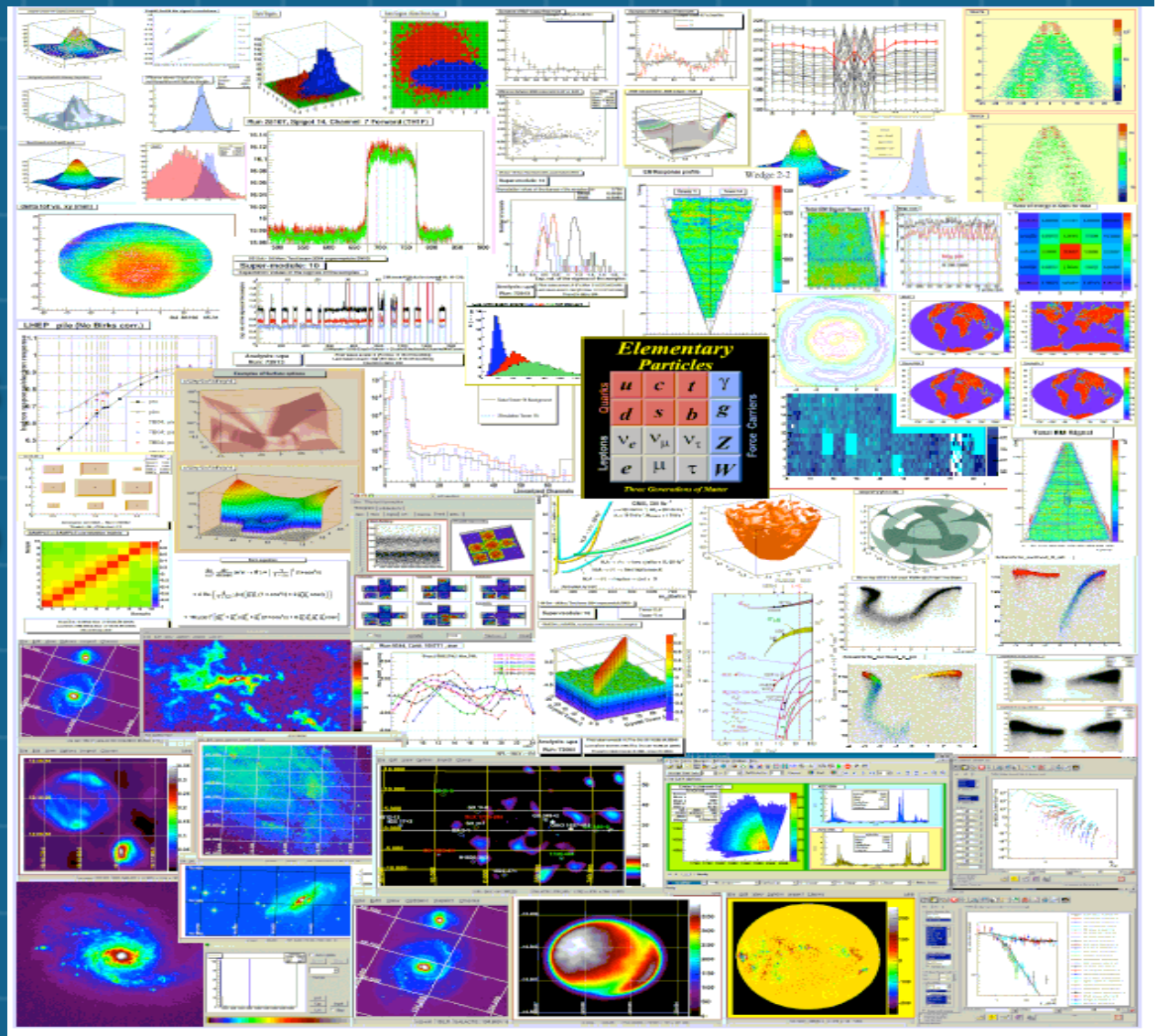
- CINT is the CORE of ROOT for
  - Parsing and interpreting code
  - Storing the class descriptions
- A new version of CINT (CINT7) based on Reflex has been in the shop for 2 years until we realized that we were building on 2 incompatible worlds with gccxml. CINT7 was also found too slow to go to production.
- We are considering an upgrade of CINT using LLVM (Apple-driven OS project). LLVM is a GCC compatible compiler featuring a JIT.
- CINT/LLVM (CLING) should be c++0x compliant. It will reduce the dictionary size (no more wrappers). It will be more robust: the interpreter is the compiler.

2-D Graphics

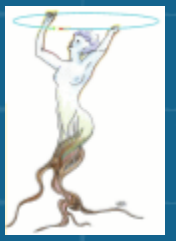


- New functions added at each new release.
- Always requests for new styles and new ways to display data.
- ps,pdf,svg,gif,

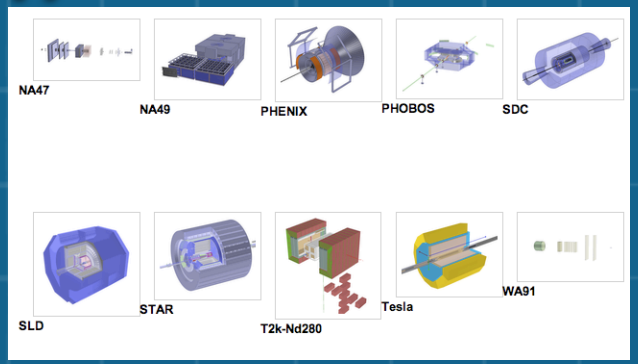
Move to GL ?



The Geometry Package TGEO



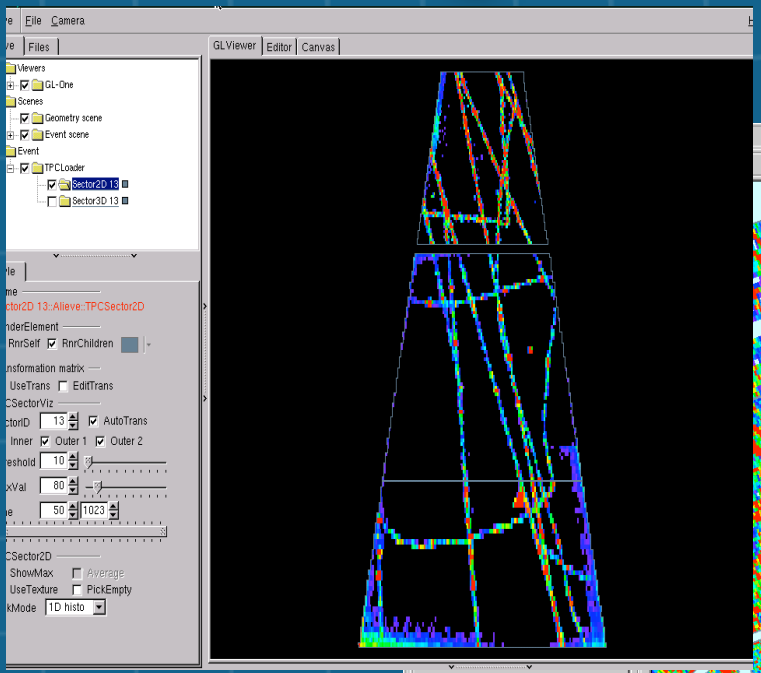
- The TGeo classes are now stable (thanks ALICE).
- Can work with different simulation engines (G3,G4,Fluka) (See Virtual Monte Carlo)
- G3->TGeo, G4->TGeo, TGeo<->GDML
- Used in online systems and reconstruction programs
- Built-in facilities for alignment
- Impressive gallery of experiments (35 detectors in $ROOTSYS/test/stressGeometry)

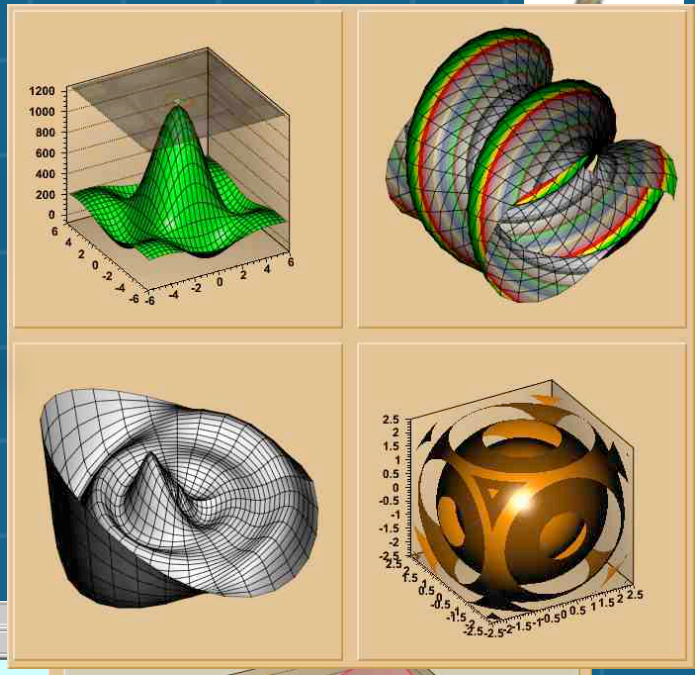


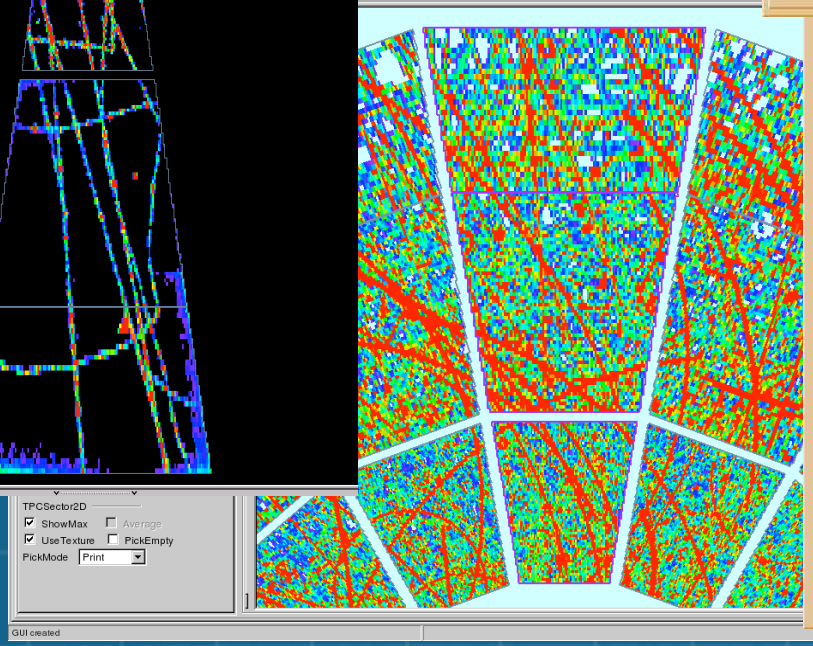
3-D Graphics

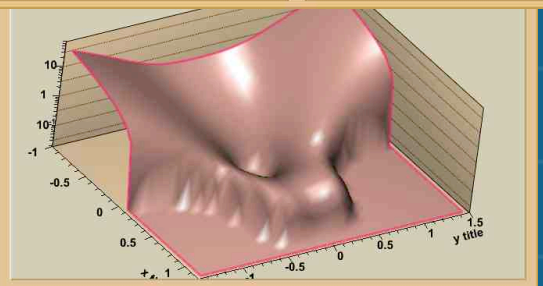
Detector hits

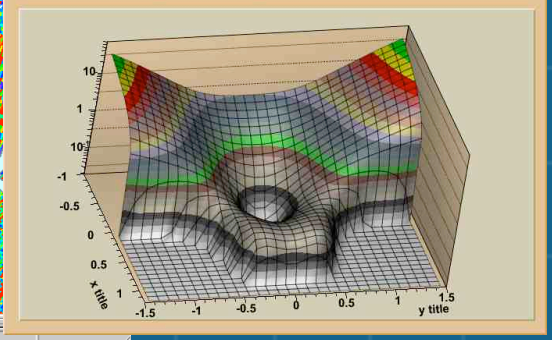
Mathematical objects



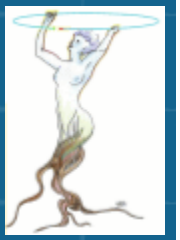




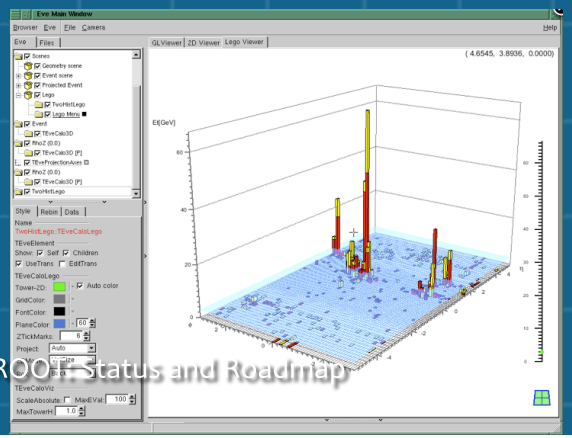


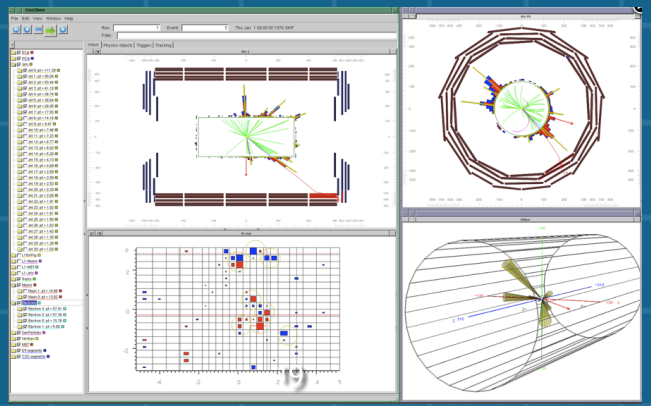


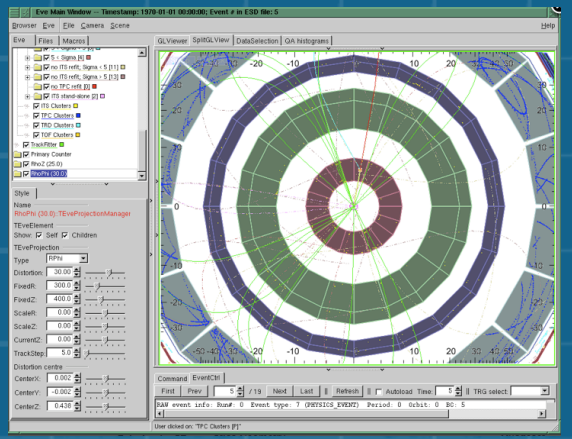
Event Display: EVE



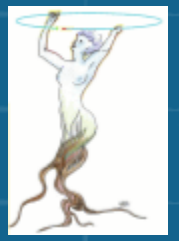
- a GL-based package for event displays.
- Developed in collaboration with ALICE (AliEve) and CMS (FireWorks).
- Provides all the GUI widgets, browsers, GL infrastructure (far better than the old OpenInventor).
- Used now by many experiments (see eg FAIRROOT, ILCROOT) to display raw data, MC events or detector oriented visualization.





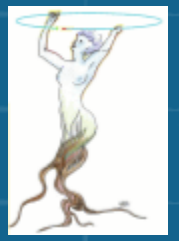


GUI



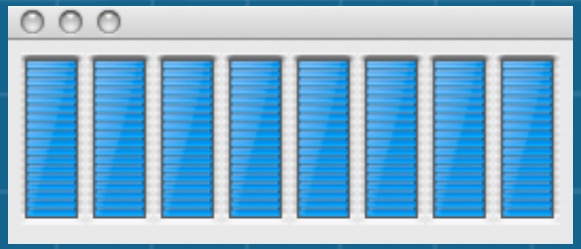
- Portable GUI with all widgets found in modern UI toolkits.
- Many enhancements in the GUI classes: browser, html browser, TABs, EVE widgets, FitPanel.
- GUI builder with C++ code generator.
- **Session recorder tool (class TRecorder)**
  - can store and replay a GUI session:
  - All mouse events
  - Keyboard input, including macro execution
- QT interfaces: a big pain, difficult to maintain with the successive versions of Qt.

PROOF

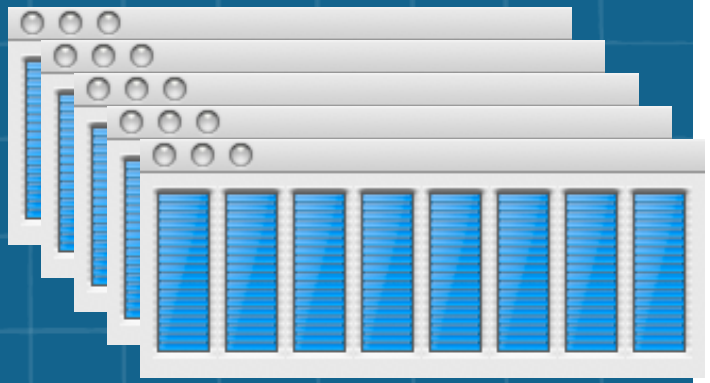


- Massive data analysis with parallel systems. Now in production in many places (see reserve slides). Growing success.
- A lot of developments for a better integration with dataset management systems and to support priority and accounting management.
- PROOF-LITE is a mini version to run on a multi-core.

ROOT: one core only

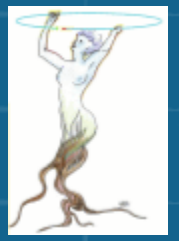


with PROOF-LITE



with PROOF

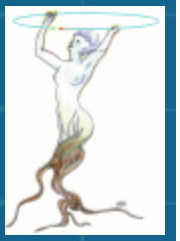
Supported Platforms



- **Linux (RH, SLC4, SLC5, Suse, Debian, Ubuntu)**
  - gcc3.4,…, gcc4.4 (32 and 64 bits)
  - icc10.1, icc11.1
- **MacOS X (Intel/PPC, 10.4, 10.5, 10.6)**
  - gcc4.0.1, gcc4.2.1,gcc4.4.1, llvm-g++
  - icc10.1, icc11.1
- **Windows (XP, Vista)**
  - VC++7.1, VC++9
  - Cygwin gcc3.4, 4.3
- **Solaris + OpenSolaris**
  - CC5.2
  - gcc3.4, gcc4.4

Support for many compilers adds robustness at a minor development cost.
It facilitates the migration to new compiler versions.

Supporting Infrastructure

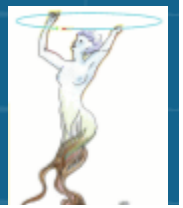


- **Flexible ./configure;make based build system**
  - **make -j 24 takes 2.50 min for all 2.2 million lines of code on a 24-core machine**
- **RPM and Debian packaging support.**
- **ROOT about to be distributed by native RedHat, Debian and Ubuntu**
- **Builds shared and static versions**
- **ROOT website moved from static web pages to collaborative Drupal based content management system**

Simple and standard solution in most OS projects

http://root.cern.ch

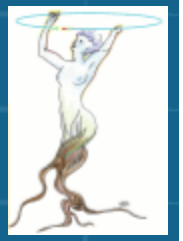
Testing, Nightly builds



Project	Version	slc4_ia32_gcc34_dbg (Fri Sep 25 04:36 2009)		slc4_amd64_gcc34_dbg (Fri Sep 25 04:16 2009)		slc4_amd64_gcc34 (Fri Sep 25 06:40 2009)		slc4_amd64_gcc43 (Fri Sep 25 08:47 2009)		i686-slc5-gcc34-opt (Fri Sep 25 05:11 2009)		i686-slc5-gcc43-opt (Fri Sep 25 07:36 2009)		x8 (F
LCGCMT	LCGCMT-preview	build	tests	build	tests	build	tests	build	tests	build	tests	build	tests	
ROOT	ROOT_today	build	tests	build	tests	build	tests	build	tests	build	tests	build	tests	
RELAX	RELAX-preview	build	tests	build	tests	build	tests	build (30)	tests	build	tests	build (30)	tests	
CORAL	CORAL-preview	build (12)	tests (42)	build	tests (4)	build	tests (9)	build	tests (10)	build	tests (5)	build	tests (8)	
POOL	POOL-preview	build	tests (10)	build	tests	build	tests	build	tests (8)	build	tests	build	tests (8)	
COOL	COOL-preview	build	---	build	---	build	---	build	---	build	---	build	---	
GAUDIATLAS	GAUDI_ATLAS	build (56)	tests	build (56)	tests	build (56)	tests	build (103)	tests	build (56)	tests	build (103)	tests	
GAUDI	GAUDI_HEAD	build	tests	build	tests	build	tests	build (42)	tests	build	tests	build (42)	tests	

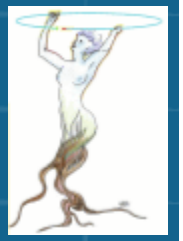
- Several thousand tests run every night on many platforms.
- Code checking with a coding convention rule checker
- Code coverage to be improved, but already much better than any other package reviewed.
- Frequent valgrinding and profiling
- Excellent cooperation with SPI

Releases Policy



- 2 production releases per year (June and December)
  - 5.24 in June, 5.26 in December
  - With patch releases with bug fixes only
  - 5.22a,b,c,d, 5.24a
- Development releases every 2 months
  - 5.23/02 feb 26
  - 5.23/04 apr 23
  - 5.25/02 sep 30
  - 5.25/04 nov 25

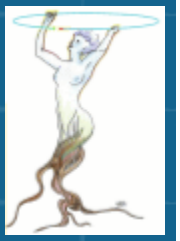
Releases and Users



- It may take a few months before large experiments move to a new version of ROOT. Currently:
  - ALICE -> 5.24 and trunk
  - LHCb -> 5.22d (testing 5.24 and trunk)
  - CMS -> 5.22d (testing trunk)
  - ATLAS ->5.22d
- End-users typically working with very recent versions
  - To use the latest graphics features
  - To use the latest TMVA , RooFit, EVE, GUI, Maths,etc

This must be improved on all sides. Conflict between stability and new features

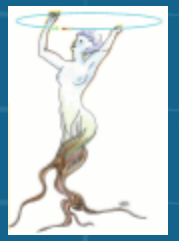
User Support

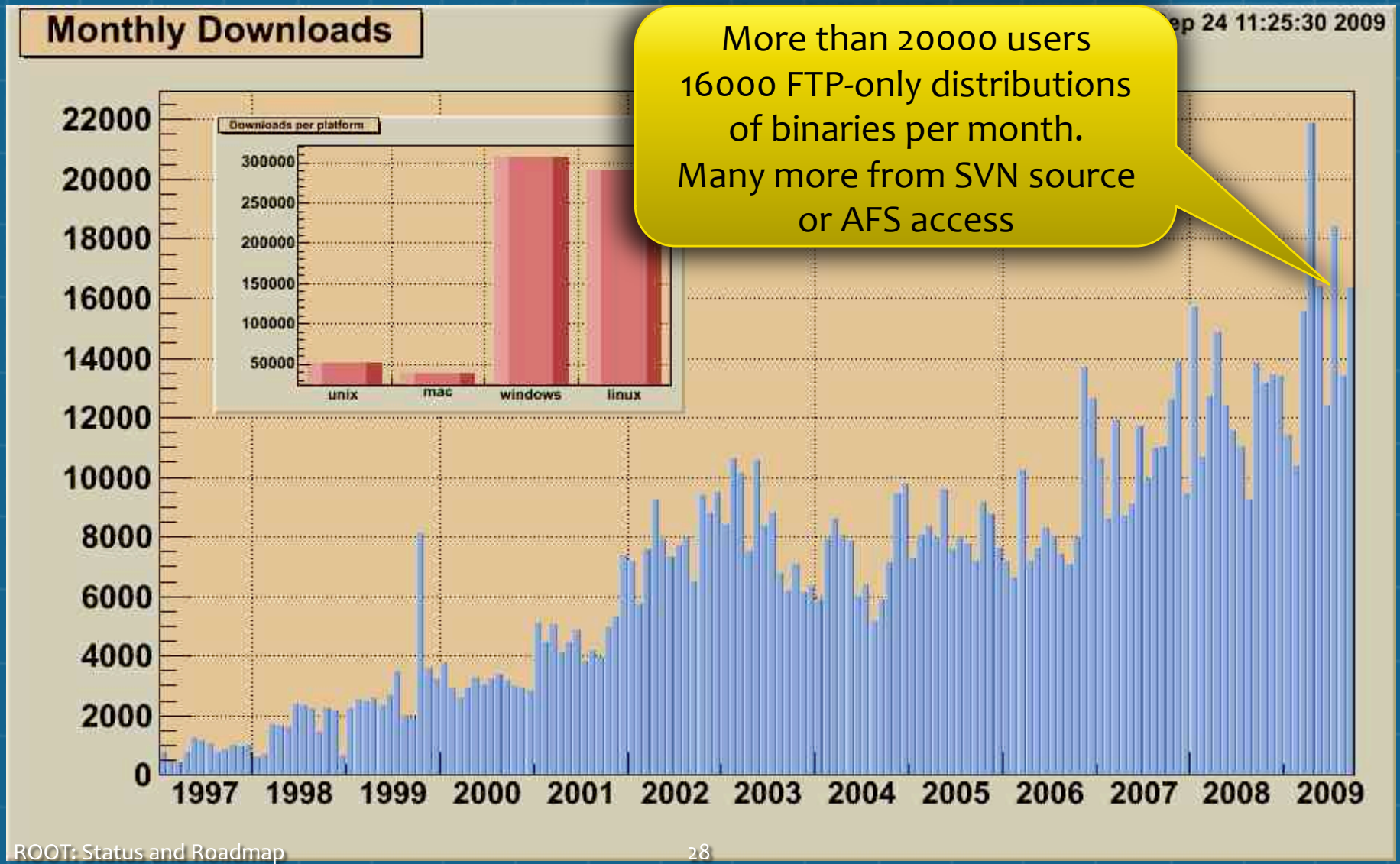


A large fraction of the developers time

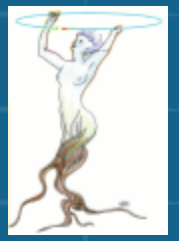
- We have several mailing lists, Forum, Savannah reporting systems. The speed to answer is critical. One hour is considered good, one day very bad.
- We are discouraging the use of the roottalk mailing list (spam). 4600 mails in 2000, 937 in 2009
- Many mails to rootdev (about 5000 per year)
- Rootforum success: 4040 registered users, 38840 articles
- Savannah for bug reports: 2394 reports with 108 still open. Thanks SPI for the good support.

Distribution Stats





Example of use



From a mail received this morning :☺

Subject: **[ROOTDEV] HRS Computing**
From: Stefano Carrazza <stefanocarrazza@vodafone.it>
Sender: owner-rootdev@root.cern.ch
Date: 09:33
To: rootdev@root.cern.ch

Hi rooters!

So, finally I'm glad to announce a new application for nonlinear optics that uses Root, it's available at:
Website: http://hrscomputing.sourceforge.net

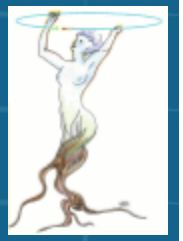
Could you please add to Root's webpage, in the example applications section, the following:
HRS Computing, an example of the use of ROOT in nonlinear optics, contact person: Stefano Carrazza

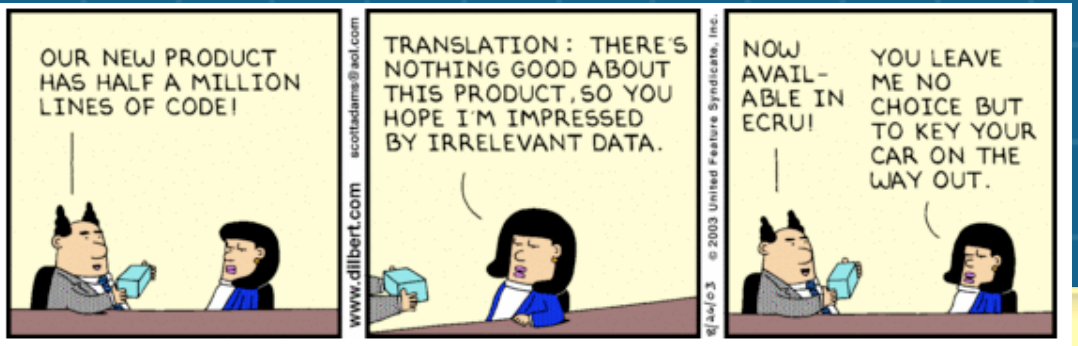
I will also post a message at roottalk with more information.

Thank you very much,
Best Regards,
Stefano Carrazza

Sloccount View

http://www.dwheeler.com/sloccount/



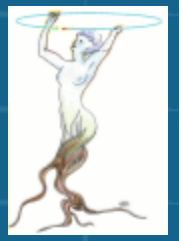


Estimation by a public tool of manpower, time to develop and total cost

Total Physical Source Lines of Code (SLOC) = 2,281,981
Estimated Person-Years(Months) = 827.18
(9,926.20)
Schedule Estimate, Years (Months) = 6.88 (82.55)
Estimated Average Number of Developers = 120.24
Total Estimated Cost to Develop = $ 111,741,241

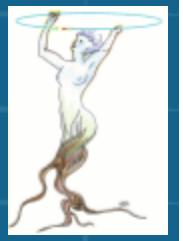
(Basic COCOMO model, Person-Months = 2.4 * (KSLOC**1.15))
(Basic COCOMO model, Months = 2.5 * (person-months**0.38))
(average salary = $56,286/year, overhead = 2.40).

Our Challenges



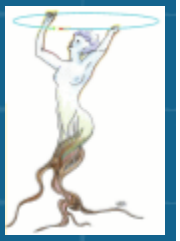
- Keep a stable core, improving robustness
- At the same time follow the technology changes
  - Distributed and parallel computing
  - Multi-core systems
  - GPU processing (OpenCL)
  - New version of C++, etc
- Improve algorithms (quality and speed)
- Improve doc , user interface, user support
- Add support for new communities, LC,etc
- And, of course, provide a robust and efficient data analysis tool for the LHC experiments.

Some worries



- The CORE packages are in the hand of unique experts who have invested many years in building the system and understanding how it is used by the experiments. What will happen if one of these persons leaves.
- CINT/Reflex/LLVM in the hands of Axel (LD), hoping IC
- I/O, Trees: essential contribution from Philippe (FNAL), but no guarantee for the medium/long term.
- Replacement for a senior staff about to leave :☺
- It takes years to train new people to the job. One should not introduce instability in a system that is going to be used massively for the LHC analysis.

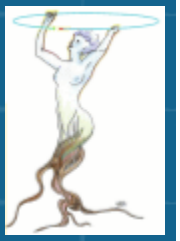
ROOT and SFT : wishes



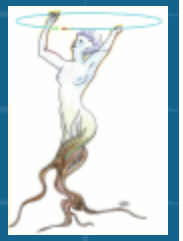
- It would be nice if the G4 and ROOT teams could improve their cooperation on items of common interest like:
  - With the interpreters CINT or/and pyroot
  - With the I/O system
  - With the Math libs, matrices, random numbers
  - With the geometry package and VMC
  - With the interfaces to event generators (common data structures)
  - With the graphics (2D & 3-D) + GUI + event displays
  - With the configuration and documentation system
  - With tuning performance and multi-core support
- This would optimize manpower, the synergy between teams and would make many users happy.

Do "+" with "<="

Summary

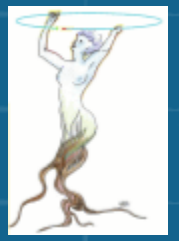


- After 15 years of development, good balance between consolidation and new developments.
- ROOT is seen as the de facto new CERNLIB.
- The ROOT main packages (I/O & Trees) are entering a consolidation, optimization phase.
- We plan on upgrading CINT with the LLVM/CLING (C++0x compatible) compiler.
- PROOF becoming main stream for LHC analysis
- User Support: time consuming but essential task
- Request stable and sufficient manpower at a critical time for LHC analysis.



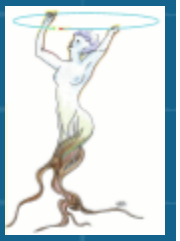
Reserve slides

New in ROOT Math



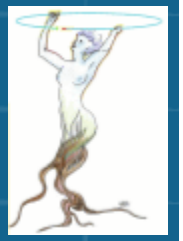
- New numerical algorithm classes and common interfaces for function evaluation
  - mechanism to use the different implementations via the plug-in manager
- New ROOT fitting classes
  - re-engineered fitting and minimization classes
  - multiple implementations of Minimizer interfaces
    - Minuit, Minuit2, Fumili, GSL minimizers
  - support for parallel fitting and minimization
- New GUI for fitting all ROOT data analysis objects (histograms, graphs, trees)

PROOF



- Parallel coordination of distributed ROOT sessions
  - Transparent: extension of the local shell
  - Scalability: small serial overhead
- Multi-Process Parallelism
  - Easy adaptation to broad range of setups
  - Less requirements on user code
- Process data where they are, if possible
  - Minimize data transfers
- Event-level dynamic load balancing via a pull architecture
  - Minimize wasted cycles
- Real-time feedback
  - Output snapshot sent back at tuneable frequency
- Automatic merging of results
- Optimized version for multi-cores (PROOF-Lite)

PROOF installations



ALICE

CERN Analysis Facility

- 112 cores, 35 TB
  - Target: 500 cores, 110 TB
- Prompt analysis of sele cted data, calibration, alignment, fast simulation
- 5-10 concurrent users
  - ~80 users registered

GSI Analysis Facility, Darmstadt

- 160 cores, 150 TB Lustre
- Data analysis, TPC calibration
- 5-10 users
- Performance: 1.4 TB in 20 mins

Other farms: JINR, Turin

ATLAS

Wisconsin

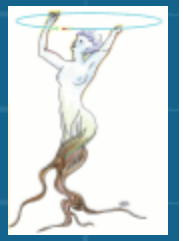
- 200 cores, 100 TB, RAID5
- Data analysis (Higgs searches)
- I/O perfomance tests w/ multi-RAID
- PROOF-Condor integration
- ~20 registered users

BNL

- 112 cores, 50 TB HDD, 192 GB SSD
- I/O perfomance tests with SSD, RAID
- Tests of PROOF cluster federation
- ~25 registered users

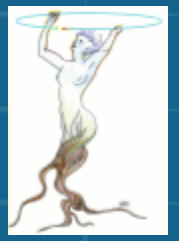
Test farms at LMU, UA Madrid, UTA, Duke, Manchester

PROOF: more installations



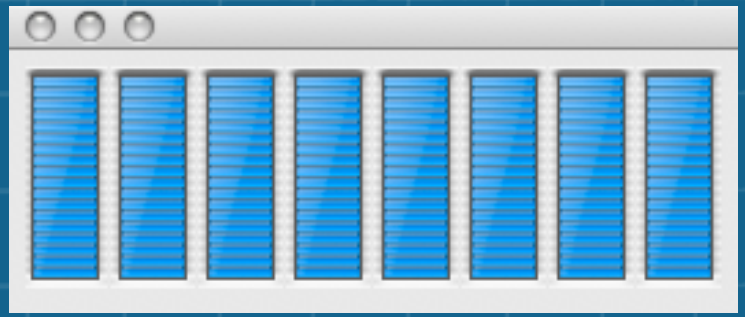
- NAF: National Analysis Facility at DESY
  - ~900 cores shared w/ batch under SGE
  - ~80 TB Lustre, dCache
  - Data analysis for ATLAS, CMS, LHCb et ILC
    - PROOF tested by CMS groups
  - ~300 registered users
- CC-IN2P3, Lyon
  - 160 cores, 17 TB HDD
  - LHC data analysis
- Purdue University, West Lafayette, USA
  - 24 cores, dCache storage
  - CMS Muon reconstruction
- ...

PROOF-LITE



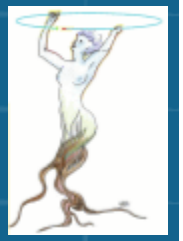
- PROOF optimized for single many-core machines
- Zero configuration setup. No config files and no daemons
- Workers are processes and not threads for robustness
- Like PROOF it can exploit fast disks, SSD's, lots of RAM, fast networks and fast CPU's
- Once your analysis runs on PROOF Lite it will also run on PROOF (with exactly the same user code)

ROOT: one core only



with PROOF-LITE

ROOT libs



- Granularity: More than 100 shared libs
- You load what you use
- root.exe links 6 shared libs (VM < 20 Mbytes)

```
(macbrun2-3) [357] ls *.so
libASImage.so          libGenVector.so        libMetaTCint.so        libRecorder.so         libTree.so             libXrdSecpwd.so
libASImageGui.so       libGeom.so             libMetaTCint_7.so      libReflex.so           libTreePlayer.so       libXrdSecsss.so
libCint.so             libGeomBuilder.so      libMinuit.so           libReflexDict.so       libTreeViewer.so       libXrdSecunix.so
libCint7.so            libGeomPainter.so      libMinuit2.so          libRint.so             libUnuran.so           libXrdSut.so
libCintex.so           libGpad.so             libNet.so              libRooFit.so           libVMC.so              libdequeDict.so
libCore.so             libGraf.so             libNetx.so             libRooFitCore.so       libX3d.so              liblistDict.so
libEG.so               libGraf3d.so           libNew.so              libRooStats.so         libXMLIO.so            libmap2Dict.so
libEGPythia6.so        libGui.so              libPhysics.so          libRootAuth.so         libXMLParser.so        libmapDict.so
libEGPythia8.so        libGuiBld.so           libPostscript.so       libRuby.so             libXrdBwm.so           libminicern.so
libEve.so              libGuiHtml.so          libProof.so            libSPlot.so            libXrdClient.so        libmultimap2Dict.so
libFFTW.so             libHbook.so            libProofDraw.so        libSQL.so              libXrdCrypto.so        libmultimapDict.so
libFTGL.so             libHist.so             libProofPlayer.so      libSessionViewer.so    libXrdCryptossl.so     libmultisetDict.so
libFitPanel.so         libHistPainter.so      libProofx.so           libSmatrix.so          libXrdOfs.so           libsetDict.so
libFoam.so             libHtml.so             libPyROOT.so           libSpectrum.so         libXrdProofd.so        libvalarrayDict.so
libFumili.so           libKrb5Auth.so         libQuadp.so            libSpectrumPainter.so  libXrdRootd.so         libvectorDict.so
libGX11.so             libMLP.so              libRGL.so              libSrvAuth.so          libXrdSec.so
libGX11TTF.so          libMathCore.so         libRIO.so              libTMVA.so             libXrdSecgsi.so
libGdml.so             libMathMore.so         libRLDAP.so            libTable.so            libXrdSecgsiGMAPLDAP.so
libGed.so              libMatrix.so           libRODBC.so            libThread.so           libXrdSeckrb5.so
```