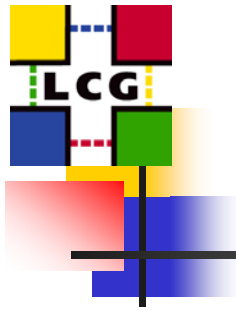
ROOT Project Status
Major developments
Directions

LCG Comprehensive Review

14 November 2005

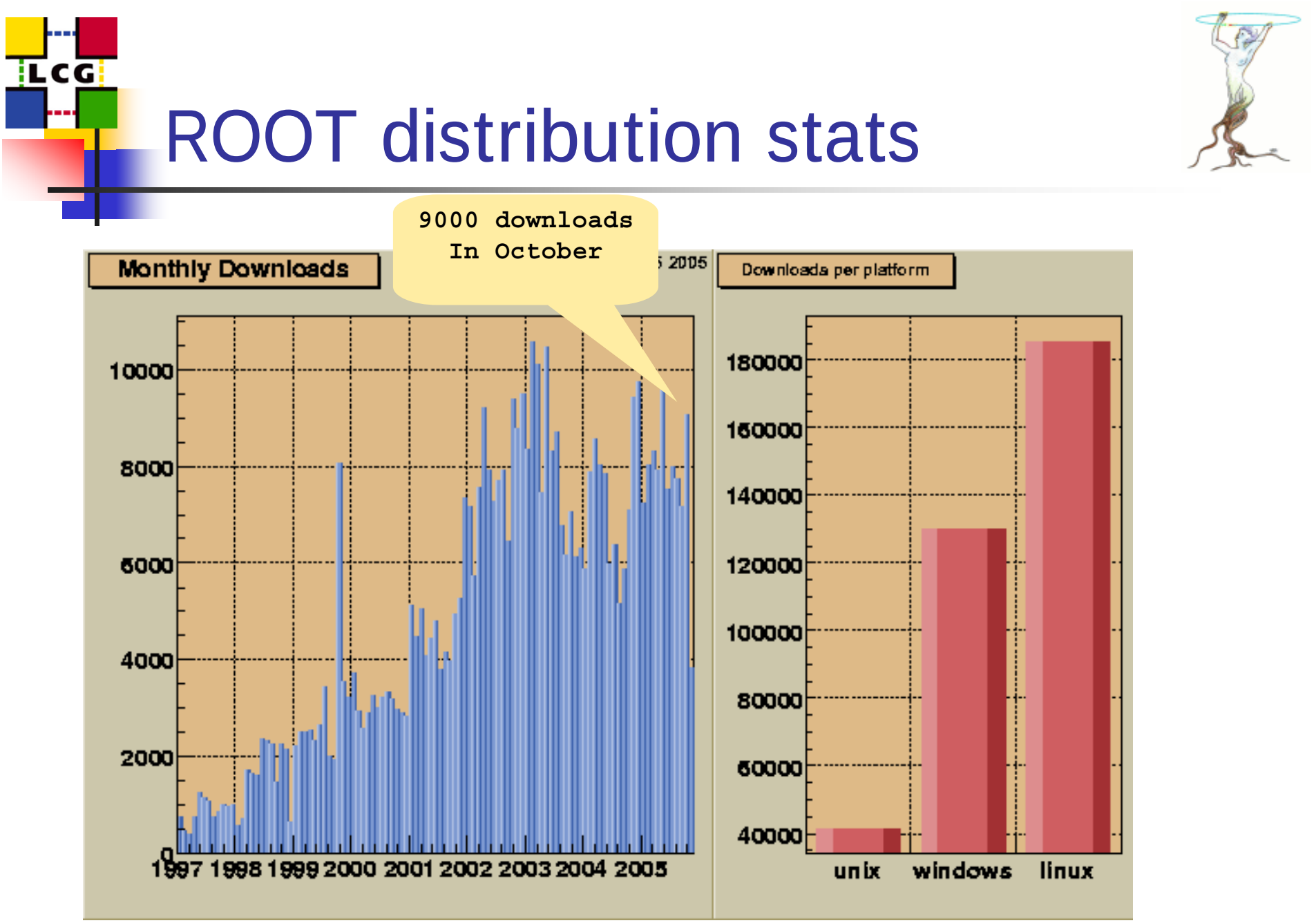
René Brun

CERN



ROOT: a 10 years old project

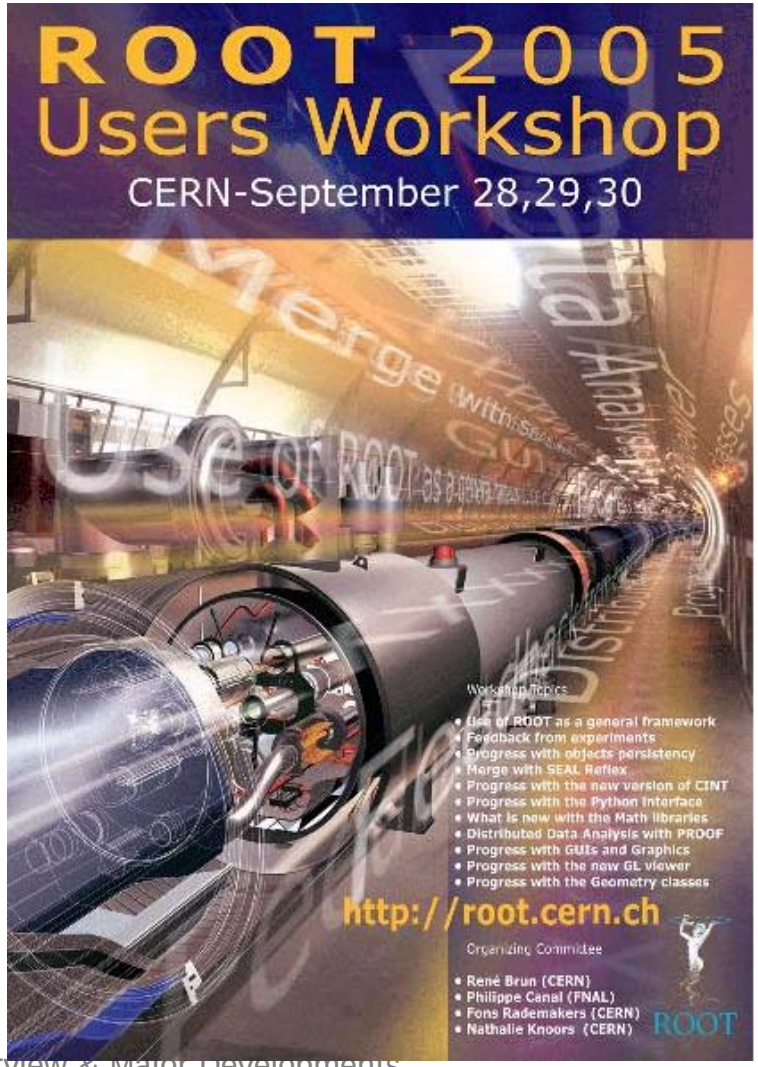
- Started in January 1995 in NA49
- First public presentation in November 95
- Cooperation with Masa Goto/CINT in April 96
- First implementation of ROOT with Alice in December 97 (TGeant3)->VMC
- FNAL chooses ROOT for I/O and data analysis
  - Announced at CHEP98 in Sept 98 in Chicago
  - RHIC experiments follow immediately
- Hoffmann computing Review in 2000/2001
- LCG project in 2002 (Blueprint RTAG)
- ROOT: LCG project in 2005



ROOT distribution stats

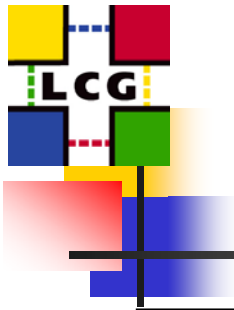
ROOT 2005 workshop September 28-30 at CERN

- 115 registered participants
- 41 talks
- 9 posters
- See talks at ROOT web page



ROOT License

- With version 5.04 we have changed the license to LGPL.
- With this change ROOT can be distributed with systems like Debian, RedHat, etc.
- The LPGL will also allow our commercial users to continue using the system



ROOT version 5

- pro release 4.04/02 3 May (new Users Guide)
- 1st dev release 5.02 28 June
- 2nd dev release 5.04 20 September
- 3rd dev release 5.06 2 Nov
- Pro release scheduled for 15 December

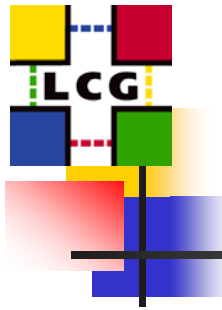
See detailed Release Notes
http://root.cern.ch/root/Version50400.news.html

See presentations at the ROOT2005 workshop

New ROOT team structure

The work-packages

SEAL Lorenzo Moneta	**DICT** Philippe Canal	**BASE** Fons Rademakers
MATH Lorenzo Moneta	**I/O & Trees** Philippe Canal	**2-D/3D graphics** Olivier Couet
GEOM/VMC Andrei Gheata	**PROOF** Fons Rademakers	**GUI** Ilka Antcheva



SEAL work-package

- Lorenzo Moneta
- Stefan Roiser

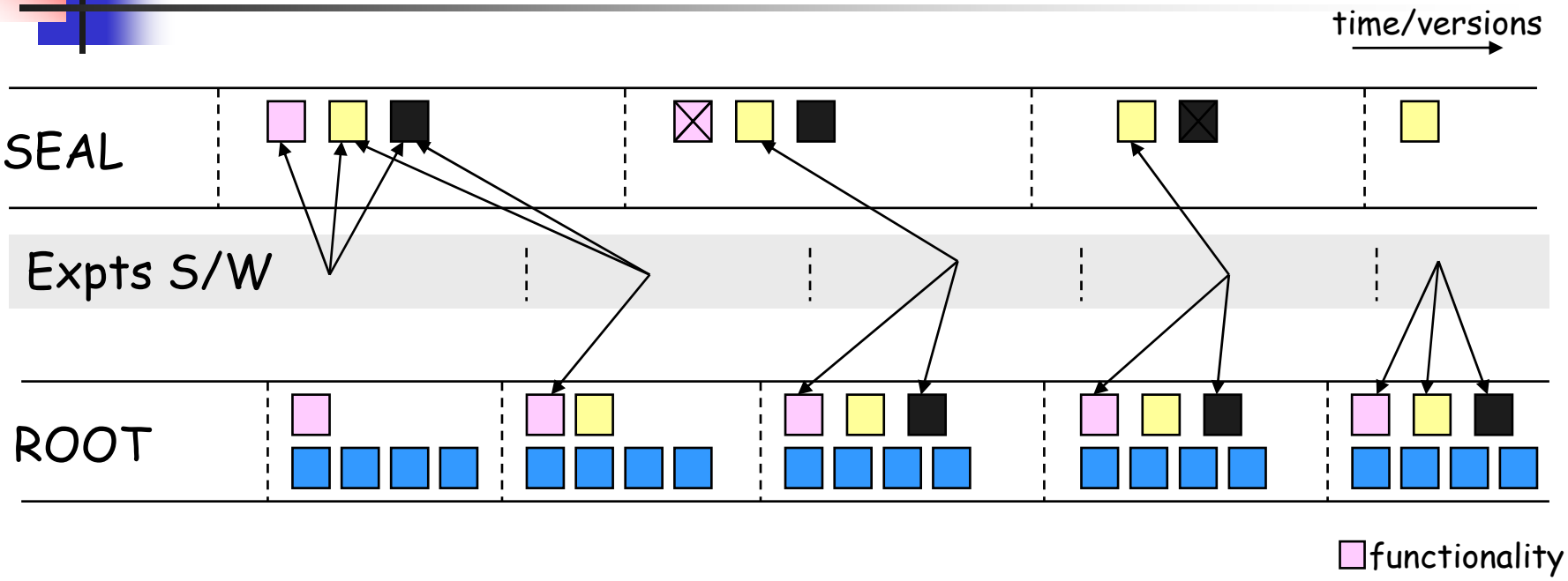
Maintenance of existing SEAL libraries

In a medium/long term keep:

Foundation classes; SealBase, SealUtil

SealKernel, SealServices

Smooth transition for Experiments



- Adiabatic changes towards the experiments
- SEAL functionality will be maintained as long as the experiments require

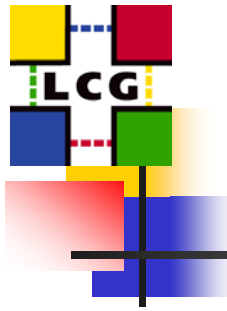
BASE work-package

- Fons Rademakers
- Ilka Antcheva (doc) (LCG1, new LCG2)
- Bertrand Bellenot (new LCG2 since 8/05)
- Philippe Canal (FNAL)
- Axel Naumann (html/doc) (new LCG2 11/05)

```
CVS, DOC, Install, Releases, QA, Mailing lists
ACLIC
System classes, Collection classes
Network, plug-in manager
```

BASE work-package : Plan

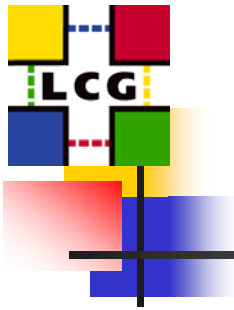
- plug-in manager extensions
- port to new platforms
- I/O thread safe
- New THtml & Help/Doc system



DICT work-package

- Philippe Canal (FNAL)
- Markus Frank (LHCb)
- Masa Goto (Agilent/Tokyo)
- Wim Lavrijsen (PyRoot) (LBL/Atlas)
- Axel Naumann (Cint->Reflex)
- Stefan Roiser (Reflex) LCG1, new LCG2

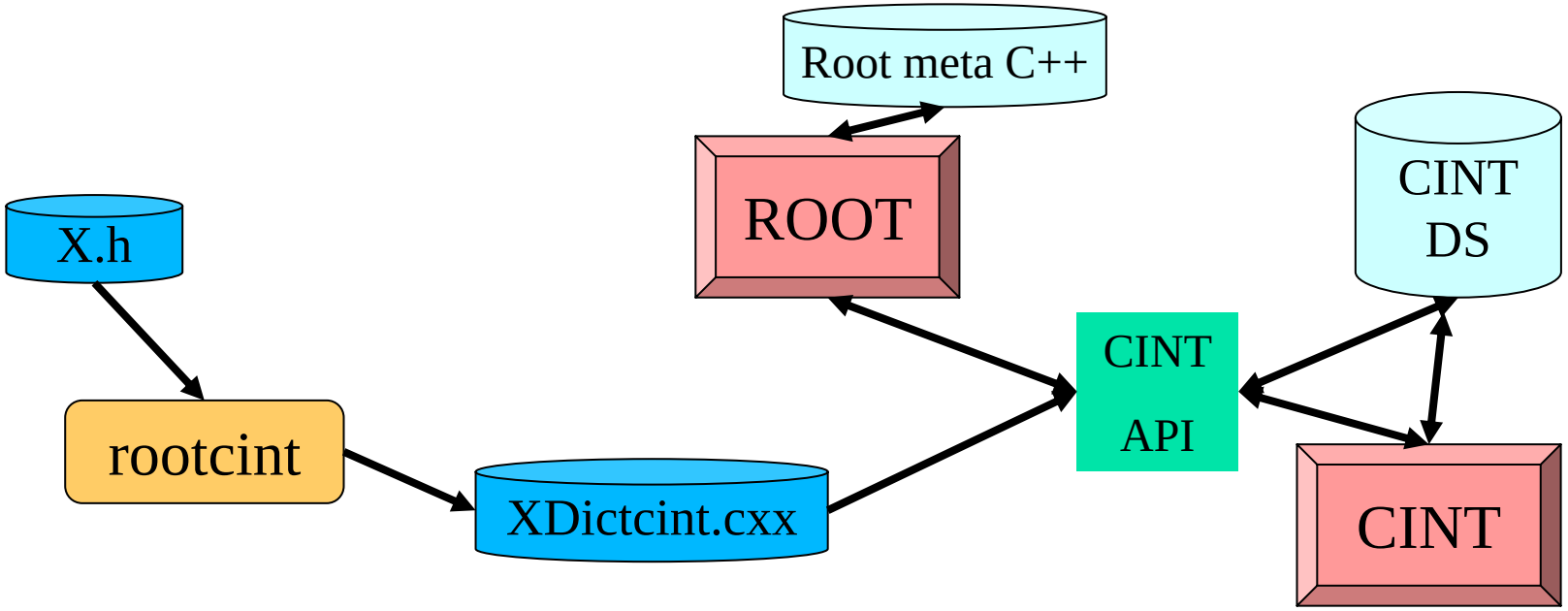
```
CINT, Reflex, ROOT meta classes

Rootcint, gccxml

Pyroot
```



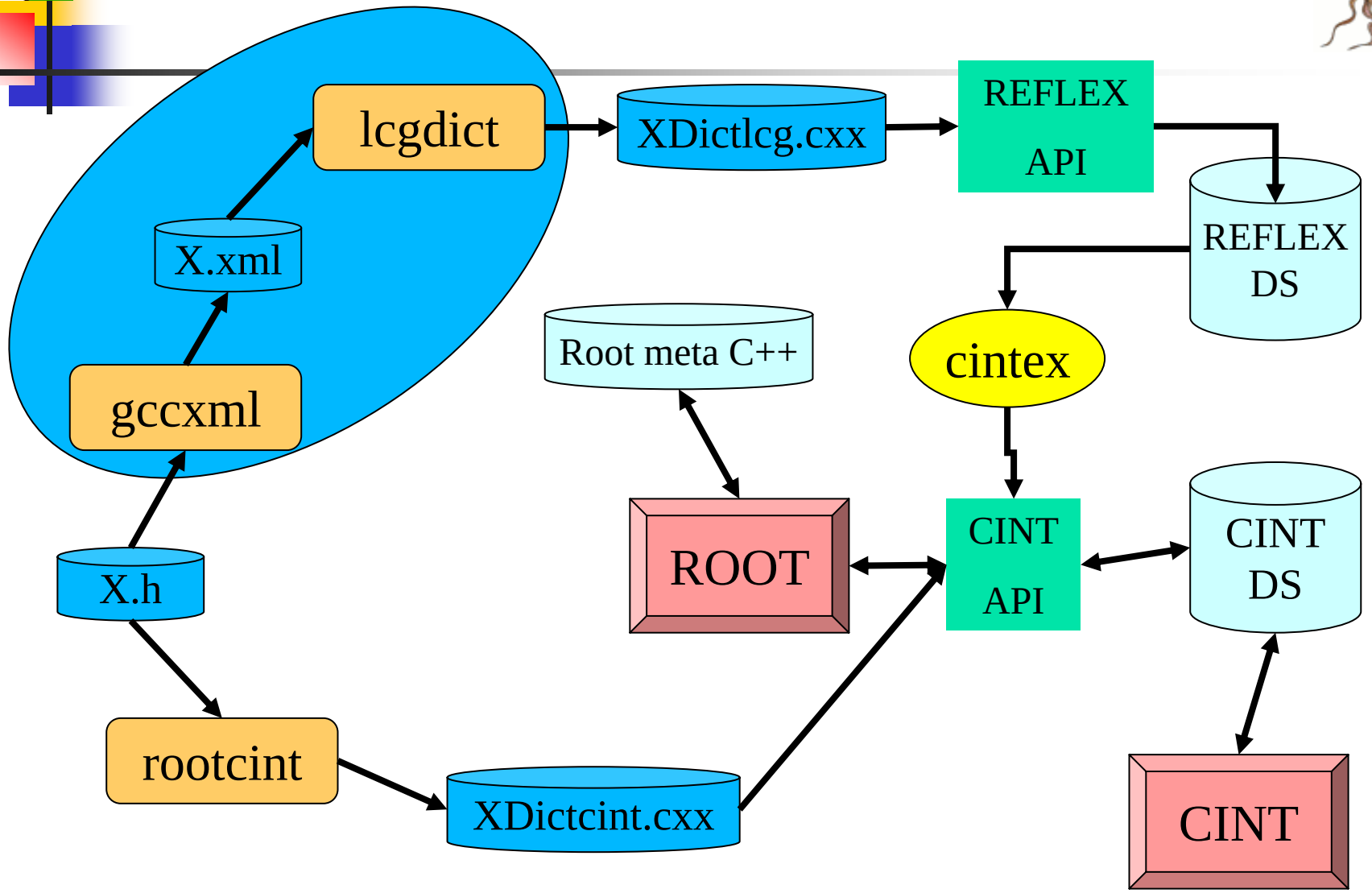
DICT work-package : Plan

- New version of Reflex
- New version of rootcint
- rootcint → CINT
- rootcint -> Reflex ->Cintex ->CINT
- rootcint ->gccxml -> Reflex ->Cintex -> CINT
- Adapt PyRoot to Reflex
- Adapt CINT to Reflex

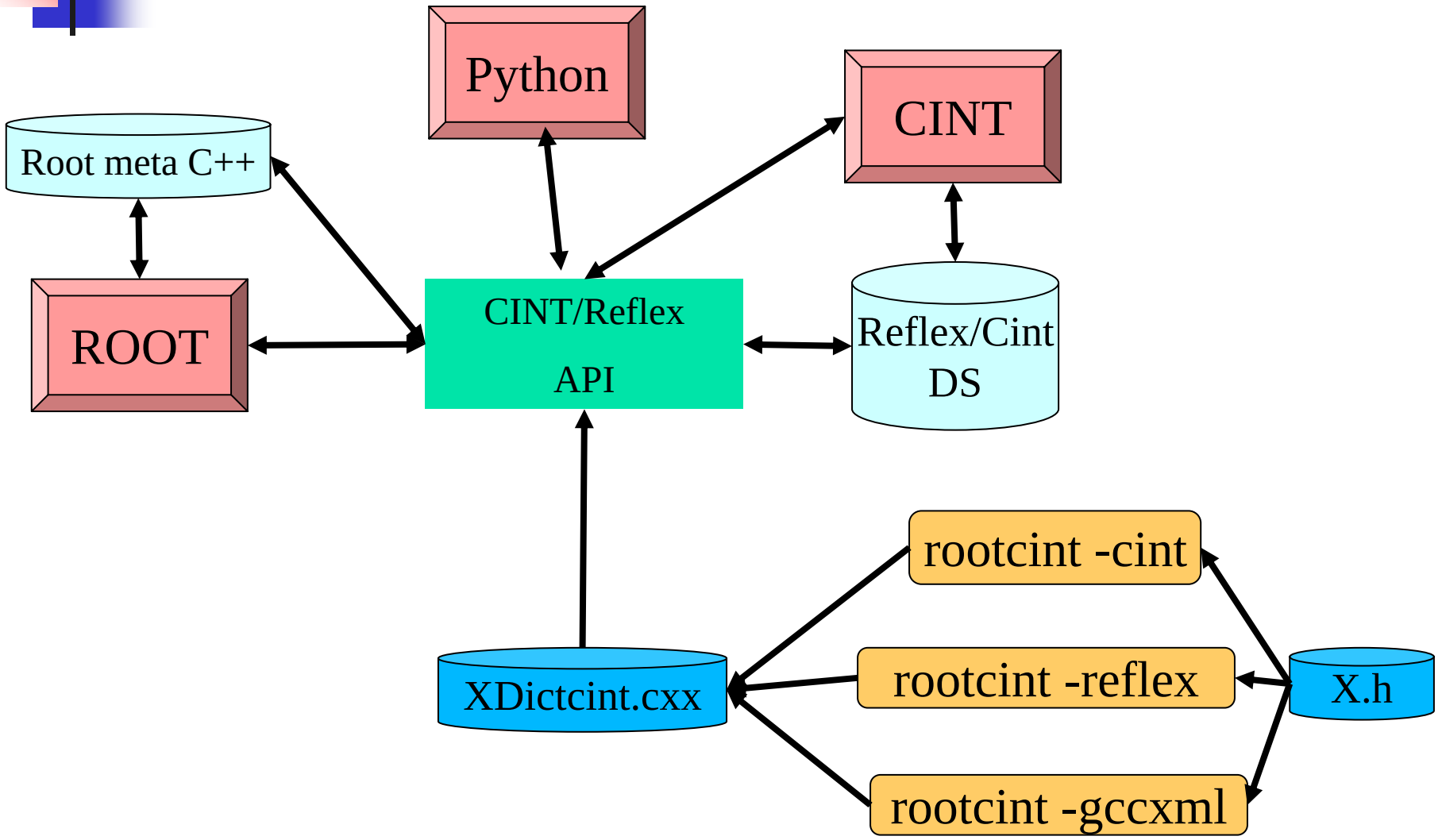
Dictionaries : root only

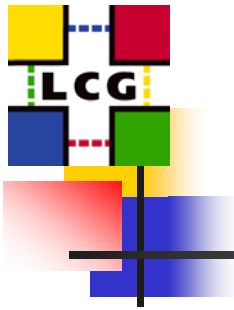


Dictionaries : situation today



Dictionaries : situation in the future





IO & Trees work-package

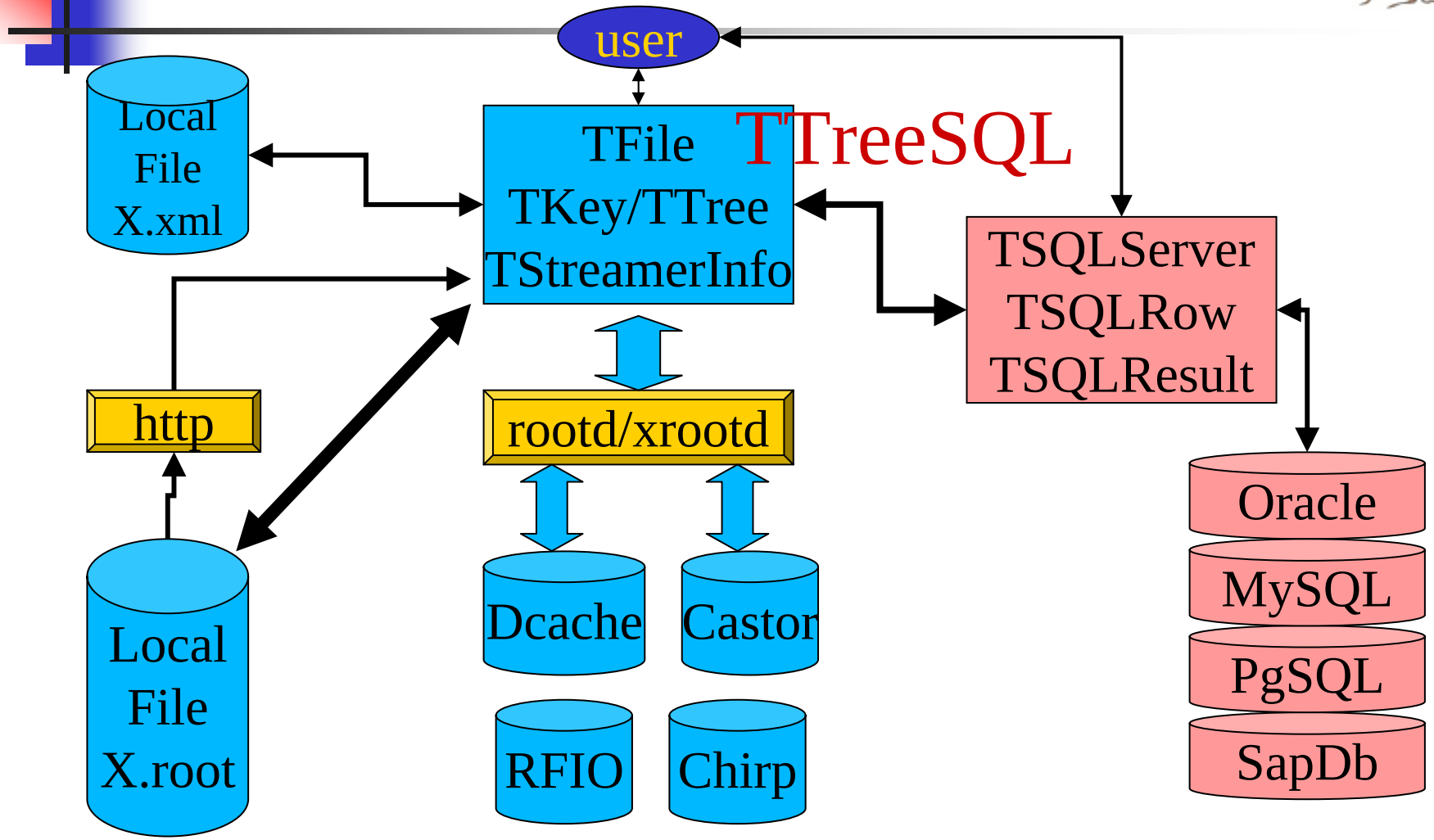
- Philippe Canal (FNAL)
- Markus Frank (LHCb <50%)
- Paul Russo (FNAL new since September)

```
Basic I/O, Auto schema evolution
CINT/rootcint/reflex interfaces
Trees, TreeSQL, Tree queries
Bitmap indices
```

IO work-package : Plan

- Consolidation, Consolidation, Consolidation
- More cases in auto schema evolution
- Better support for references
- read ahead with large caches
- TreeSQL, TSQLFile

File types & Access in 5.04



Bitmap Indices

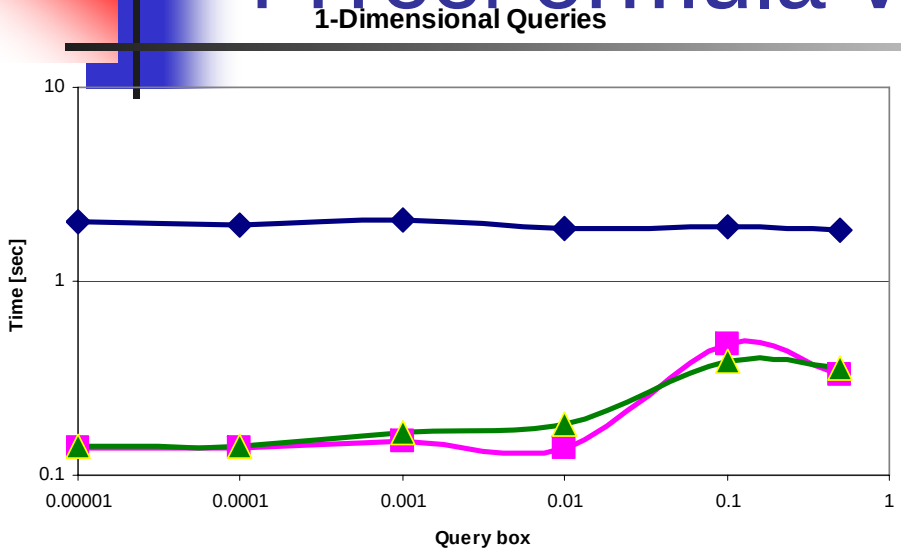
- Bitmap indices are efficient data structures for accelerating multi-dimensional queries:
  - E.g. pT > 195 AND nTracks < 4 AND muonTight1cm > 12.4
- Supported by most commercial database management systems and data warehouses
- Optimized for read-only data
- However, because an efficient index may be as big as the data, we think that it is only appropriate for things like event meta data catalogues

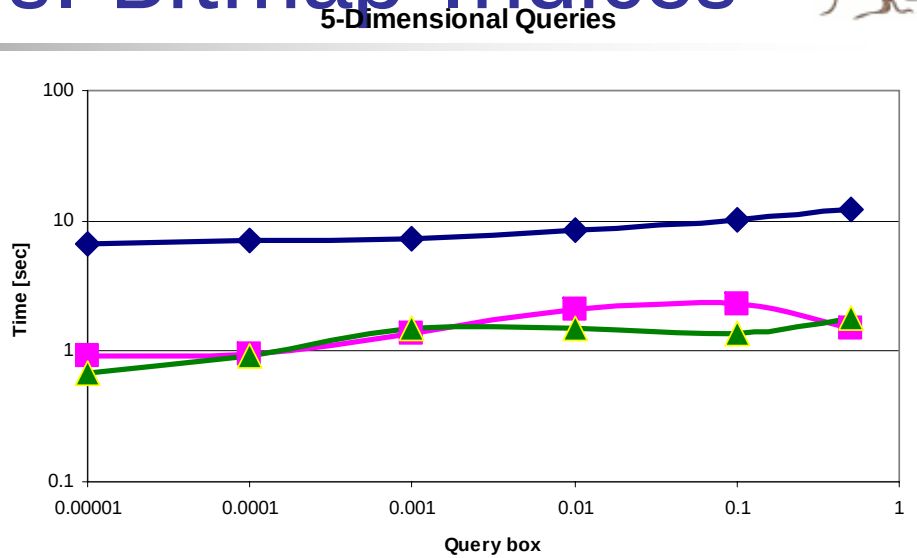
FastBit: A compressed bitmap indexing technology for efficient searching of read-only data

http://sdm.lbl.gov/fastbit

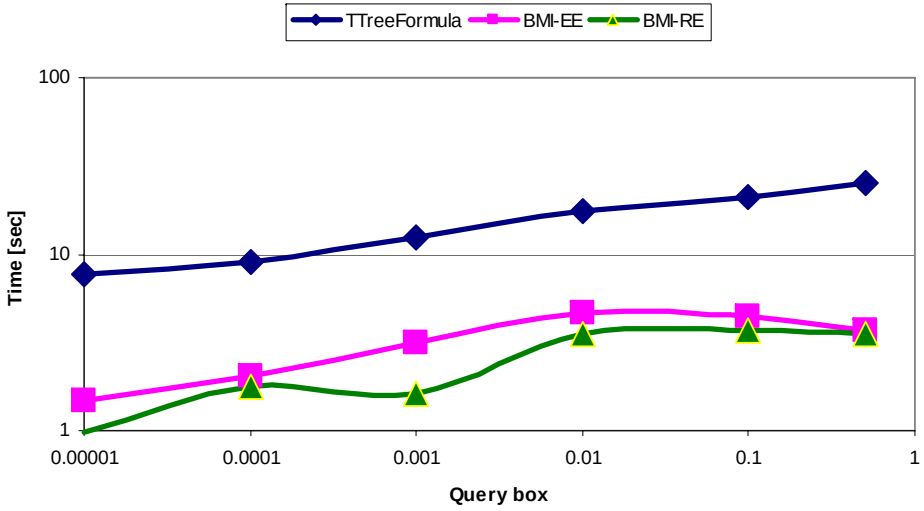
LBNL holds the copyright of the FastBit software and a US patent on the core compression technique used in FastBit. LBNL intends to seek opportunities to commercialize the searching technology and the compression technique. However, since the ROOT framework is essential to high-energy physics experiments funded by US Department of Energy, which also funded the development of FastBit and the related compression technique, LBNL has agreed to develop a license to grant ROOT users free use of the FastBit searching technology as long as FastBit is only accessed through ROOT framework. FastBit source code may also be distributed with ROOT so long as it is only used through ROOT.

Query Performance - TTreeFormula vs. Bitmap Indices

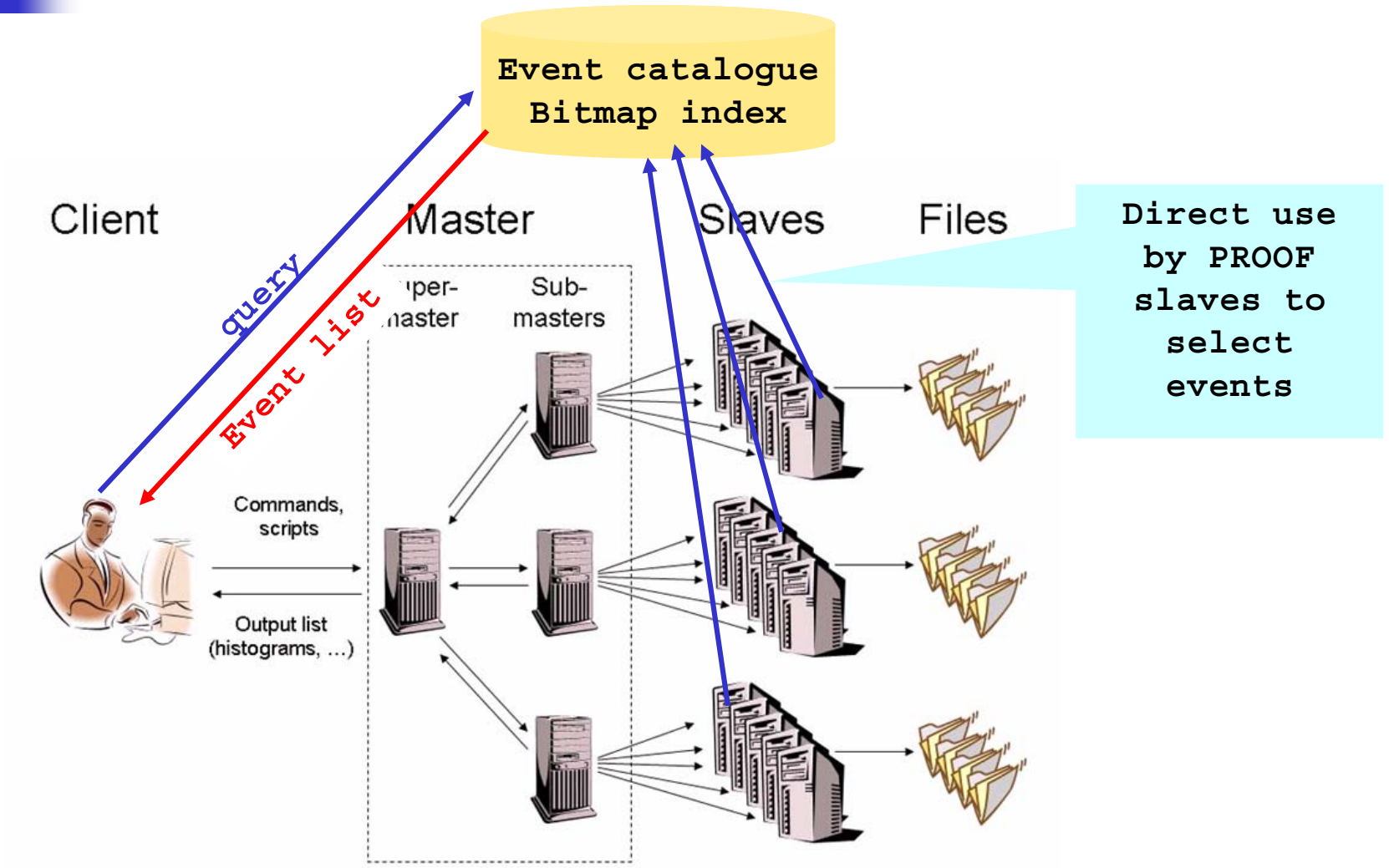


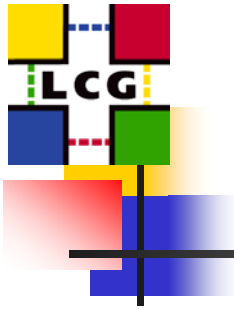


Bitmap indices 10X faster than TTreeFormula



Data analysis with bitmap indices



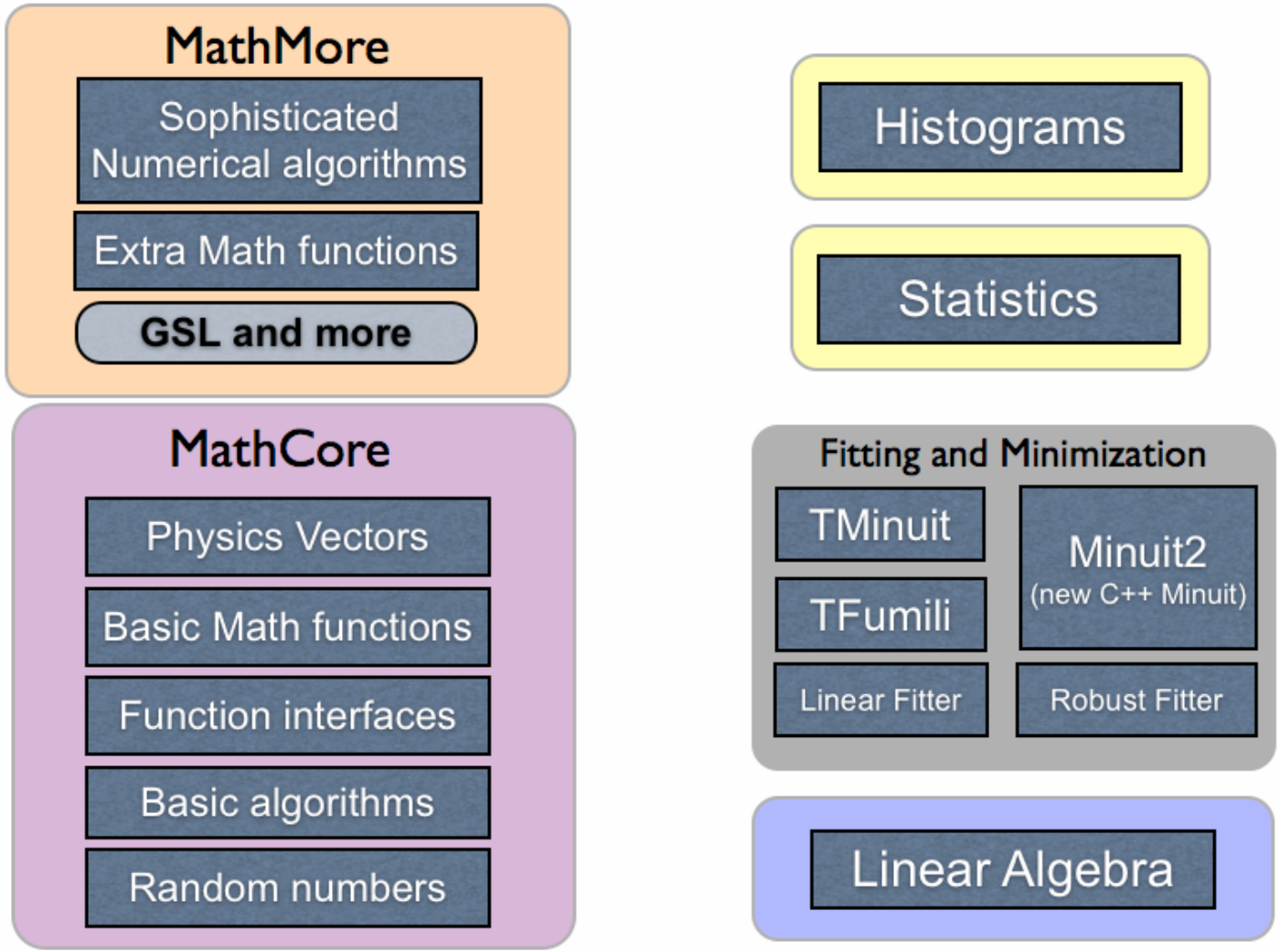


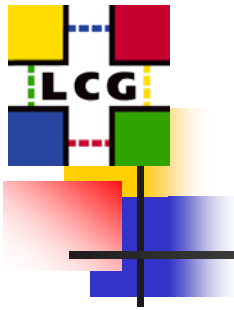
MATH work-package

- Lorenzo Moneta
- Eddy Offermann (Rentec/New York)
- Anna Kreshuk --.....---
- FNAL for MathCore

```
MathCore, MathMore, TMath, TF1, TH1
TMinuit, TFumili, new Minuit, roofit, TVirtualFitter
Stats classes, Linear Algebra
```

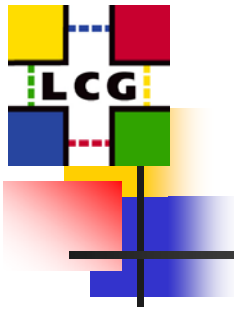
New Math Libraries organization





MATH work-package : News

- MathCore library with basic Math functionality
  - Basic Special and statistical functions
  - Physics and geometry vectors
- MathMore library
  - C++ interface to function and algorithm from GSL
    - Extra math functions, Adaptive integration, derivation, root finders
- Minuit2
  - New OO implementation of Minuit
  - Interface to ROOT TVirtualFitter
- Linear and Robust Fitter
- sPlot



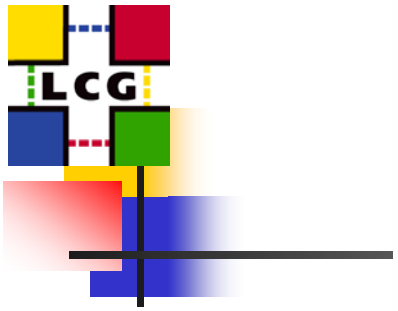
MATH work-package : Plan

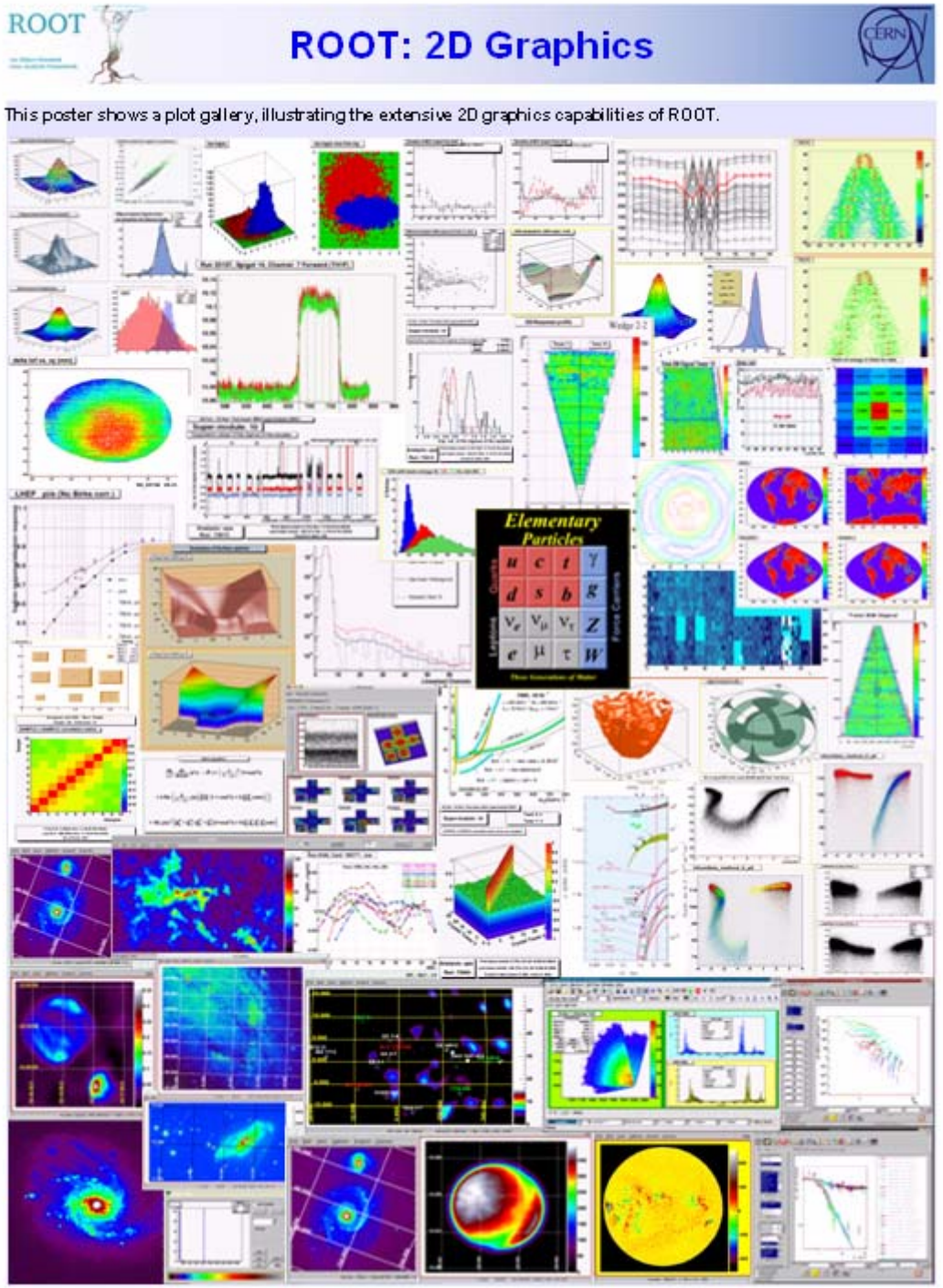
- Complete MathCore with Random numbers
- Adapt ROOT classes to MathCore
- TF1,2,3, Fitting
- Virtual Fitter extensions
  - corresponding changes in ROOT fitting and roofit
- Fully integrate and extend new Minuit
- Fitting GUI
- Box plots, qqplots
- Many new tools required for LHC Physics analysis (PHYSTAT05 Oxford)

Graphics work-package

- Olivier Couet (2-D graphics)
- Andrei Gheata (geometry) /ALICE
- Richard Maunder (GL)
- Timur Pocheptsov (GL) JINR/Dubna

```
2-D graphics, histpainter, graphs, TLatex
3-D graphics
X3D & GL viewers
Image processing classes
```




ROOT
ROOT: 2D Graphics
CERN
This poster shows a plot gallery, illustrating the extensive 2D graphics capabilities of ROOT.
Elementary Particles

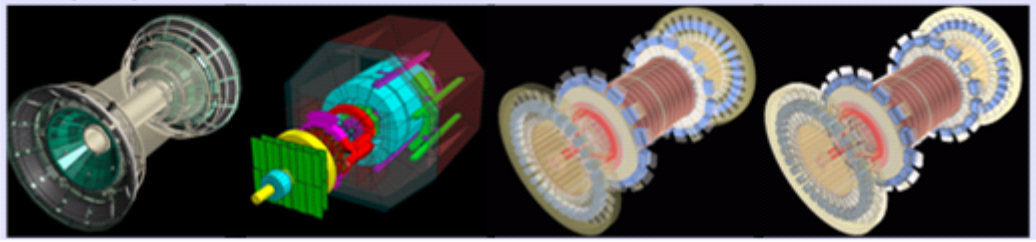
ROOT: 3D Graphics

R. Maunder [1], T. Pocheptsov [2]
[1] CERN – European Organization for Nuclear Research, Geneva, Switzerland
[2] Joint Institute for Nuclear Research, Dubna, Russia

GL Viewer: Can now be used as a standalone viewer and as built in pad. TPad can render classical 2D graphics, using X11 or Windows graphics and 3D graphics using OpenGL. We intend to exploit this to offer wide new range of data representation techniques.

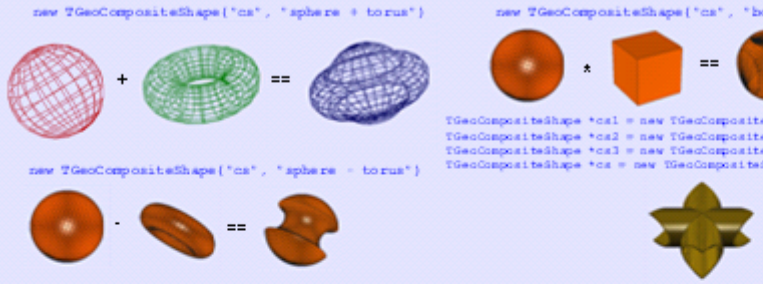
```
root[0] gStyle->SetCanvasPreferGL(kTRUE);
root[1] .x shapes.C
```

Draw Style: Objects can be drawn as solids and solids with outlines:

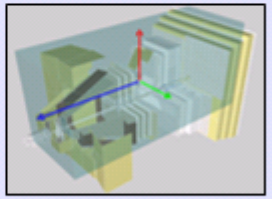


Composite Shapes: ROOT 's composite shapes (TGeoCompositeShape) can now be visualized with GL viewer:

```
new TGeoCompositeShape("cs", "sphere + torus")
new TGeoCompositeShape("cs", "box * sphere")
new TGeoCompositeShape("cs", "sphere - torus")
TGeoCompositeShape *cs1 = new TGeoCompositeShape("cs1", "tub1+tub1:r1");
TGeoCompositeShape *cs2 = new TGeoCompositeShape("cs2", "tub2+tub2:r1");
TGeoCompositeShape *cs3 = new TGeoCompositeShape("cs3", "cs1-cs2");
TGeoCompositeShape *cs = new TGeoCompositeShape("cs", "cs3*box:r2");
```



Clipping: Remove subset of objects to show detector internals etc.
Two techniques



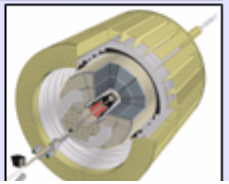
•OpenGL® Clip Planes: Multiple renders, each with one or more clip planes, combine together e.g. 3 sides of box:

Pro: Fast and simple, interactive (few planes) Con: Accurate only for shapes described by planes - approx otherwise.
Clipped solids not capped – hollow.

•CSG Operation: Add all object meshes (o1..on), subtract clipping object mesh (c):

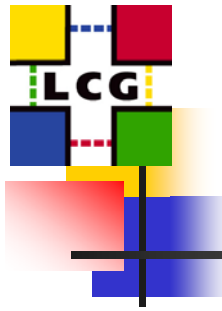
o1 + o2 + ... + on – c

Pro: Any arbitrary clipping shape possible Con: Cannot adjust interactively
Slow, complex
Proper capping of solids



Graphics work-package : Plan

- Many features planned for 2D graphics. They are all listed here: http://cern.ch/couet/POW.htm . They range from a few days to several weeks/months of work (like re-implement TGaxis).
- GL Viewer with new GUI
- GL for dynamic tracks
- GL in TPad (with PostScript output)
- Event Display infrastructure



GUI work-package

- Ilka Antcheva (0.7)
- Bertrand Bellenot (0.2)
- Fons Rademakers (0.1)
- Valeri Fine (Qt) BNL/STAR
- Valeriy Onuchin (finished in July, now Protvino)

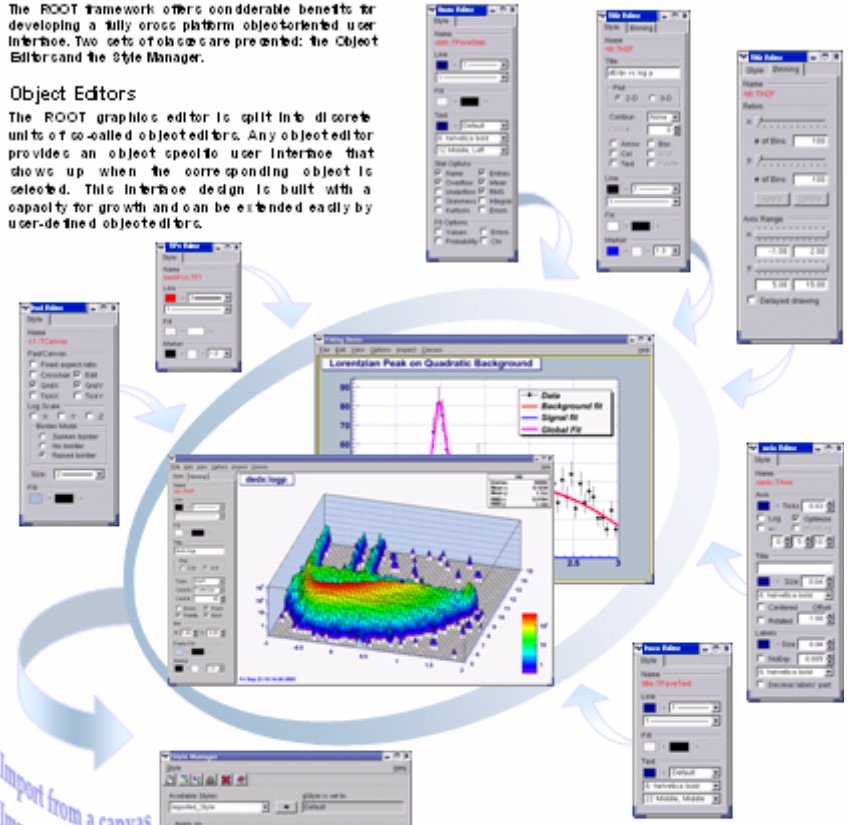
```
Low-level GUI interface (TVirtualX implementations)
High level widgets (Editors, Browsers, TreeViewer)
GUI builder
```

ROOT Graphical User Interface

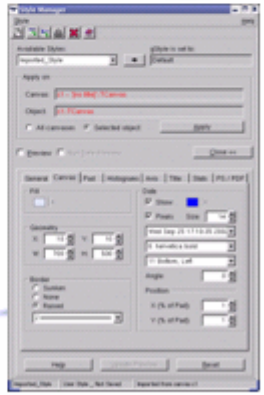
The ROOT framework offers considerable benefits for developing a fully cross platform object-oriented user interface. Two sets of classes are presented: the Object Editors and the Style Manager.

Object Editors

The ROOT graphics editor is split into discrete units of so-called object editors. Any object editor provides an object specific user interface that shows up when the corresponding object is selected. This interface design is built with a capacity for growth and can be extended easily by user-defined object editors.



Import from a canvas
Import from a macro
Create a new style
Delete a style
Edit selected style
Export to a macro



Style Manager

This new Graphical User Interface is created to manage different styles in a ROOT session. It allows users to import a style from a canvas or a macro, to select a style for editing, to export it in a C++ macro, to apply a currently selected style on a selected object in a canvas or on all canvases, to set it as the `gStyle`.

This interface is composed of two parts:
- the top level interface manages a list of all available styles for the current ROOT session and shows the currently selected one;
- the style editor deals with the settings of the currently selected style.

A preview of the selected canvas helps for precision work. It can be updated dynamically at run-time or by request to show how the edited style looks.

All changes made in the style editor can be cancelled and the edited style can be restored to the last saved state in a macro.

For more information see: http://root.cern.ch
For any questions please use the address: rootdev@root.cern.ch

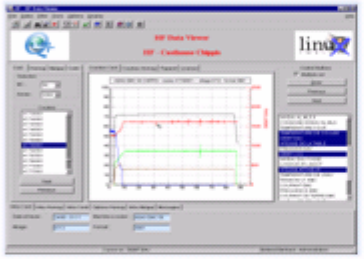
ROOT Graphical User Interface

GUI Applications Examples

The screens below illustrate the powerful Graphical User Interface capabilities of ROOT. They come from several concrete applications used in the aluminum industry (ALCAN Aluminium Valais SA).

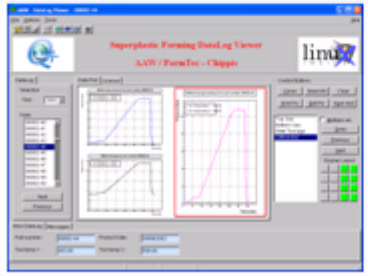
Data Visualization

The following applications are mainly used to visualize data coming from different facilities, on different platforms (Windows NT and GNU).
Data are collected on a Windows server, converted into Root format, validated, then archived on CD once a year.



The main application used in the aluminium casting plant is `HFviewer`, which regroup data from :
- liquid metal treatment (Ar + Cl2)
- spectrometric analysis (alloy composition)
- casting process
- homogenization

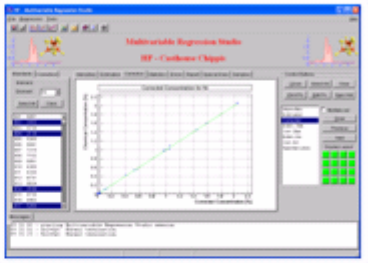
It is also used for validation of the product before expedition.



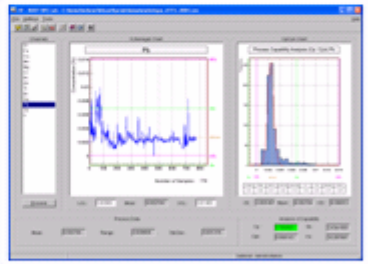
`DigView` is another data visualization application used to show data coming from the superplastic forming process. This process uses air pressure to form metal sheets (special aluminium alloy) into specific complicated shapes (e.g. car body parts).

Data Analysis

The next applications are for data analysis. These use statistics, i.e. SPC (statistical Process Control) and MVR (multivariate regression). They are used for Quality Insurance and Spectrometry.



The multivariate regression (MVR) calculation is an empirical correction procedure to minimize, in a multicomponent matrix, the influence of interfering elements on an analyte.
The MVR's options provide facilities and flexible mathematical algorithms to compute simultaneously the basic curve polynomials and coefficients for additive and/or multiplicative corrections.
The calculation is performed on the intensities and concentrations of a set of samples used as standards.



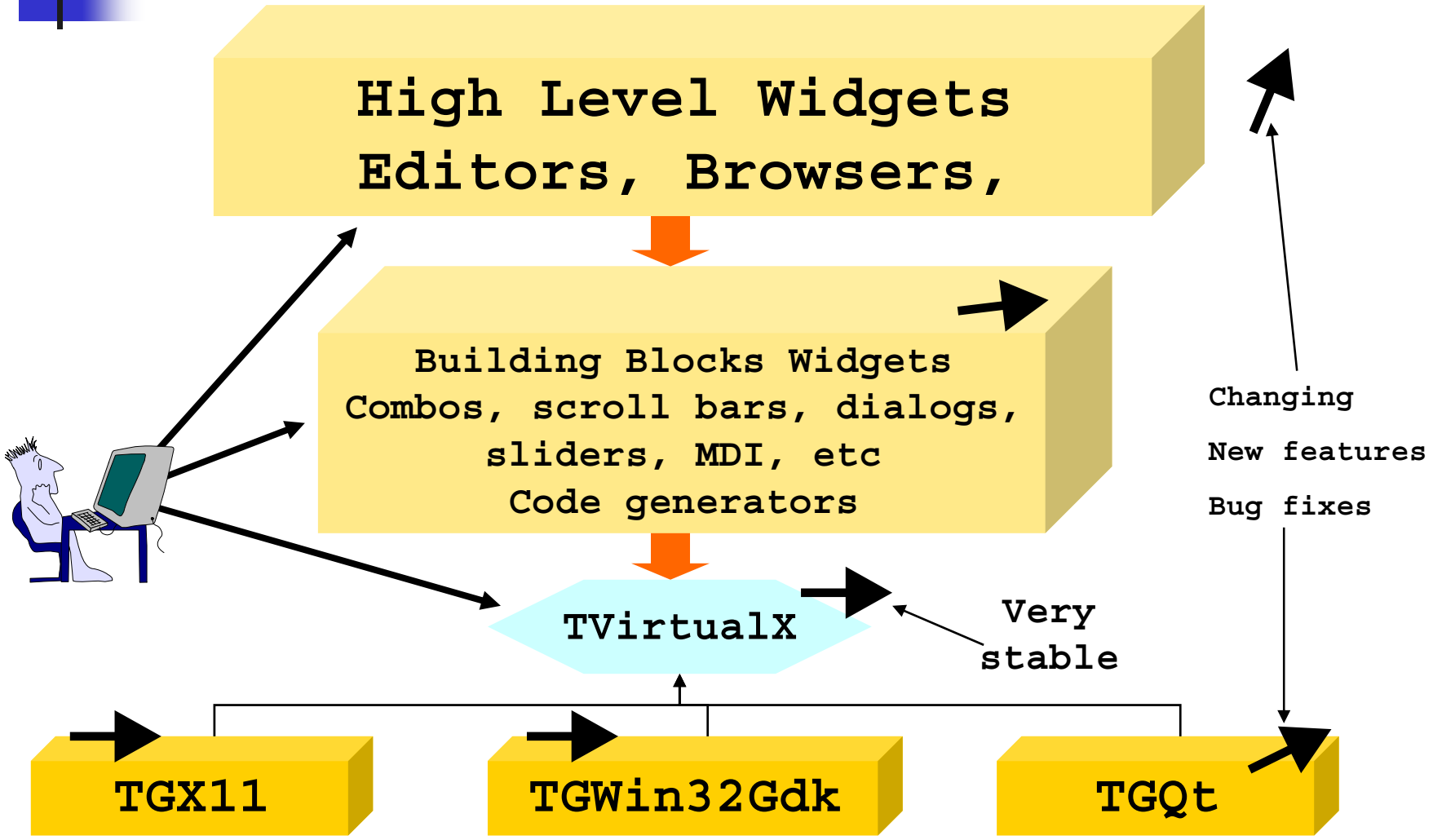
The `SPCLab` application determines the process capability of spectrometry analysis.

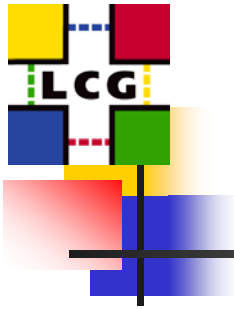
For more information see: http://root.cern.ch
For any questions please use following address: rootdev@root.cern.ch

GUI work-package : Plan

- zillions of micro/mini features
  - (see Ilka url)
  - http://antcheva.home.cern.ch/antcheva/
- New object editors
- Undo/Redo tools
- New Fit panel
- GUI Builder completion
- New GUI widgets:
  - TGHtml
  - TGTable
  - GUI skinning, etc.

GUI work-package : Plan



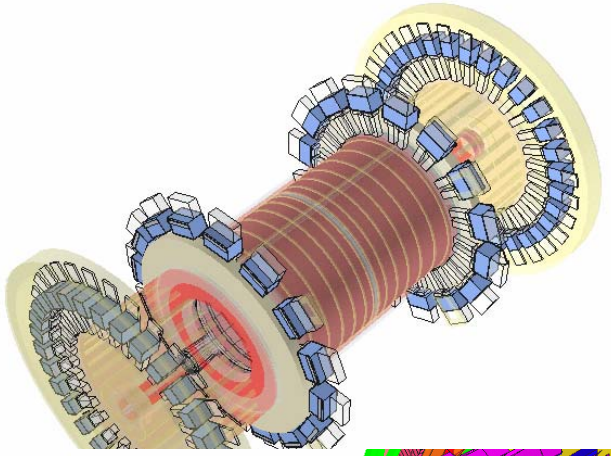


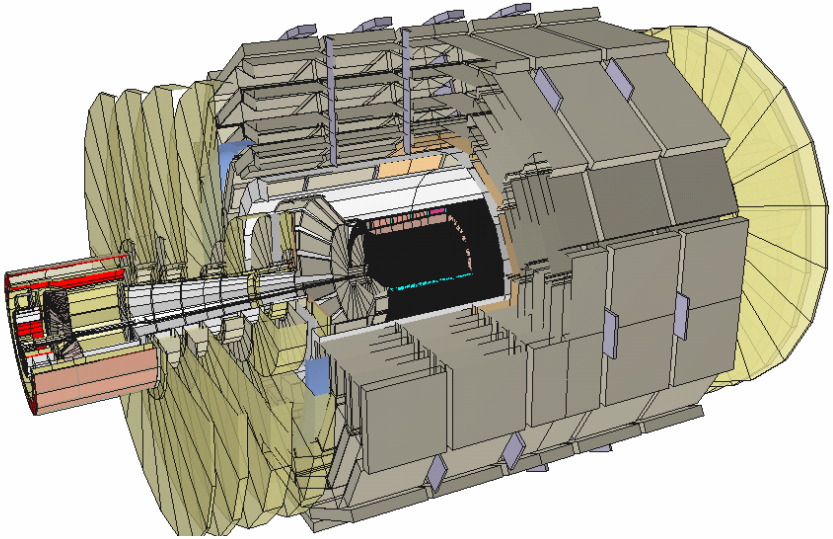
GEOM/VMC work-package

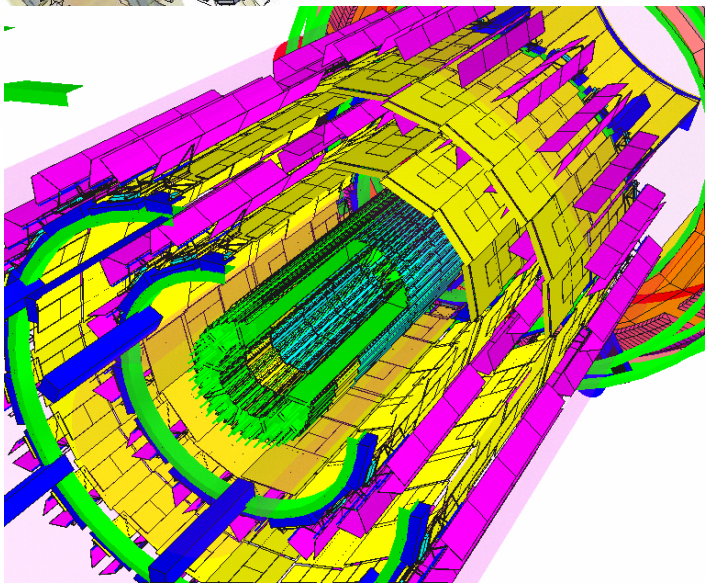
- Andrei Gheata /ALICE
- Mihaela Gheata /ALICE
- Ivana Hrivnocova /ALICE/Orsay (VMC)

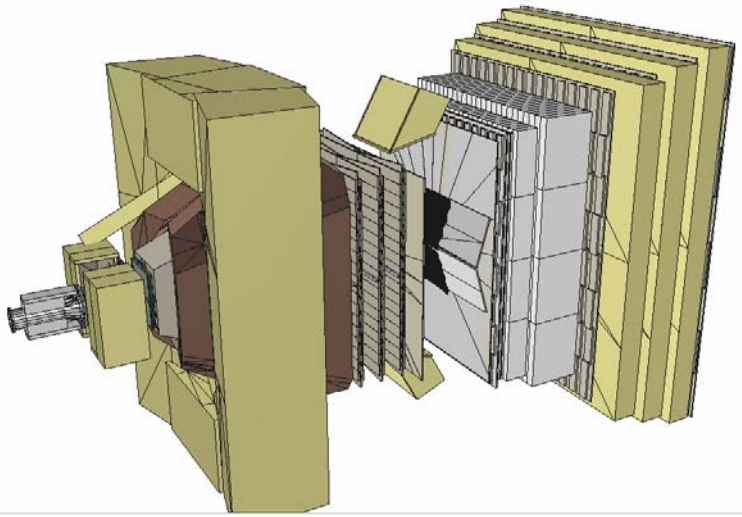
```
Detector Geometry (modeling & navigation)
Interfaces with Geant3, Geant4 and Fluka (VMC)
Graphics interface
```

LHC detectors in ROOT geom









GEOM work-package : Plan

- Support for parameterized shapes. This will reduce the geometry size in memory for certain geometries defined in G3 style.
- Interface with G4
- CAD geometry import
- Geometry builder GUI

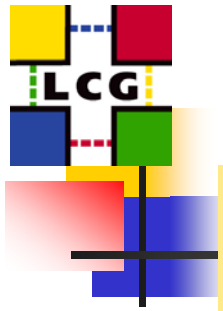
PROOF work-package

- Fons Rademakers
- Maarten Ballantijn (MIT/Phobos/CMS)
- Bertrand Bellenot (GUI)
- Gerri Ganis (LCG1, new LCG2)
- Guenter Kickinger (DS/ALICE)
- Derek Feichtinger (ARDA project)/CMS
- Andreas Peters (ARDA project)/ALICE

PROOF development

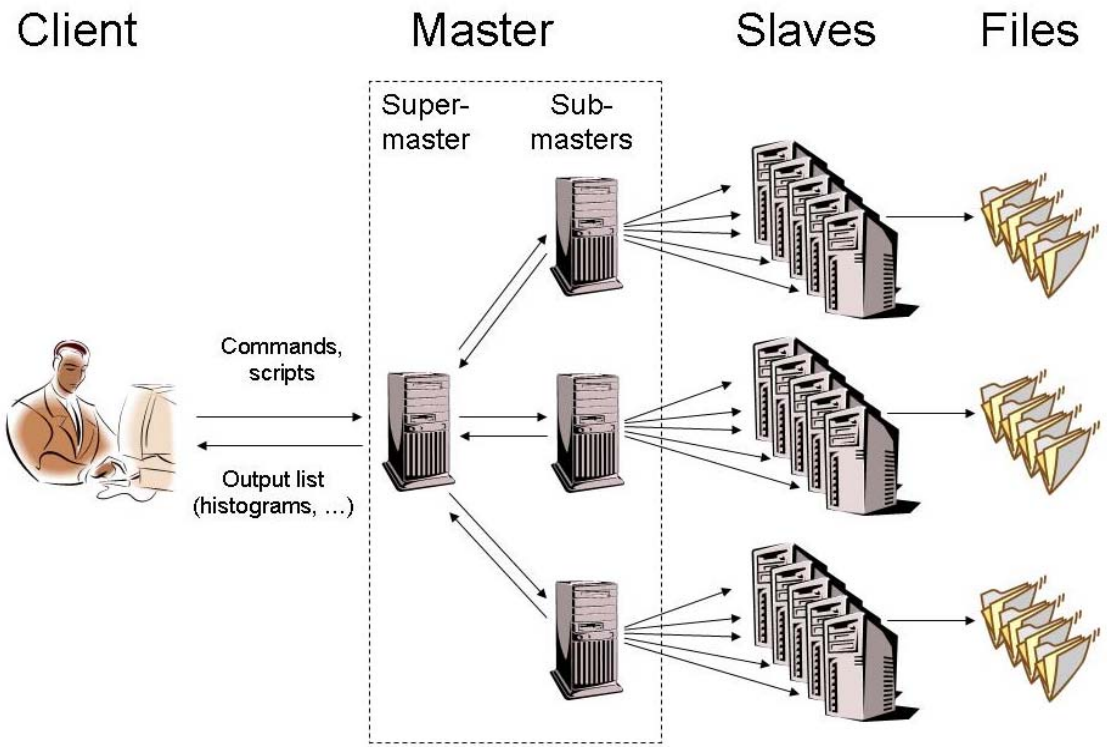
PROOF test bed

Help LHC experiments to start with PROOF

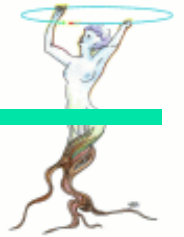


PROOF – Multi-tier Architecture

- Optimize for data locality
- If not possible, remote data via (x)rootd, rfiod, dCache …



PROOF – data analysis



Normal ROOT

- **TChain**: collection of **TTree**
- **TSelector**: :**Begin()**,**Process()**,**Terminate()**

Local processing

```
TChain a("h42");
{// Define the data set
   a.Add("root://oplapro62.cern.ch//tmp/dstarmb.root");
   a.Add("root://oplapro62.cern.ch//tmp/dstarp1a.root");
   a.Add("root://oplapro62.cern.ch//tmp/dstarp1b.root");
   a.Add("root://oplapro62.cern.ch//tmp/dstarp2.root");
   // Process the selector
   a.Process("h1analysis.C");
}
```

PROOF

- Same chain, same selector

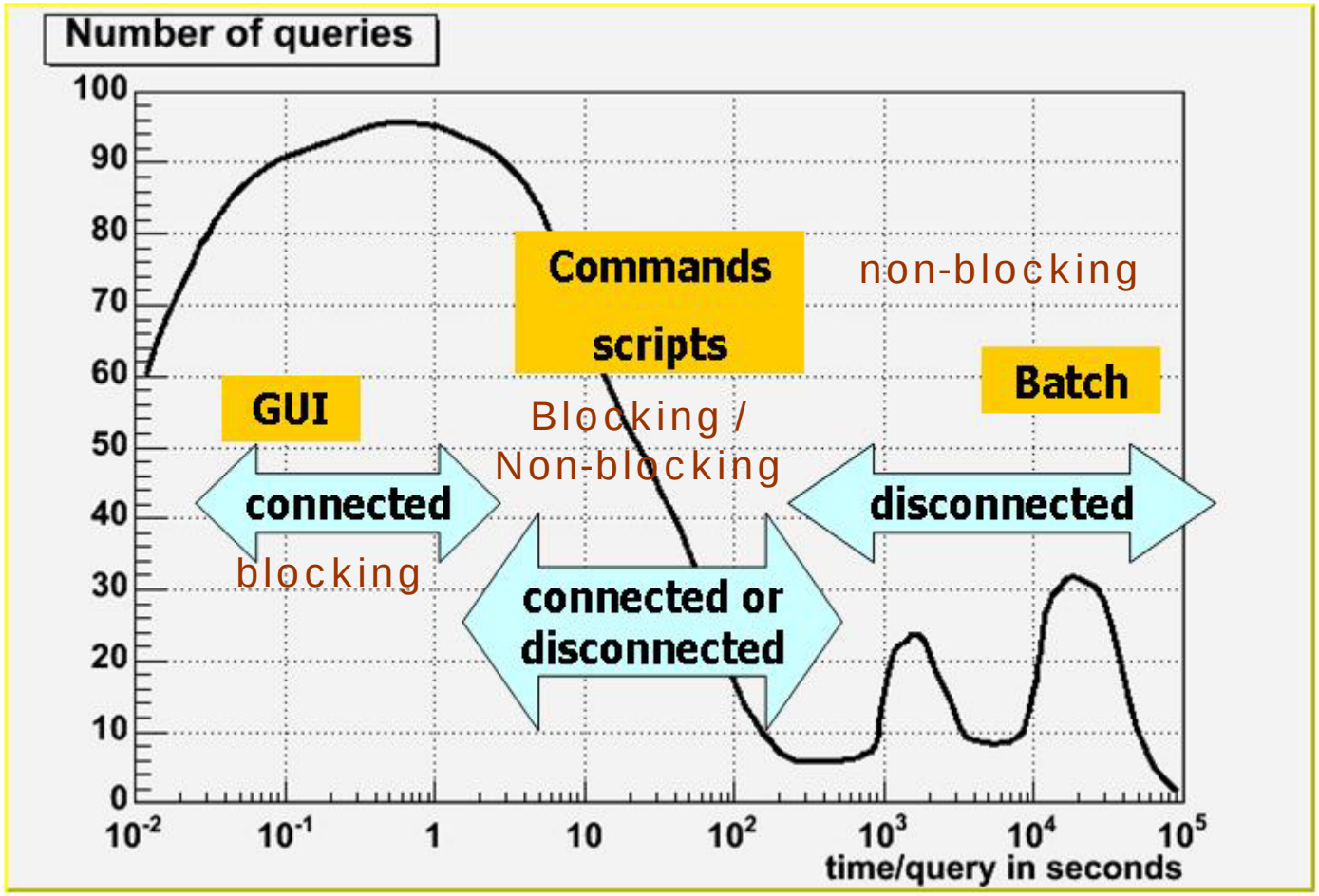
Remote processing

```
{// Open PROOF
   TProof proof("master");
   // Process the selector
   a.Process("h1analysis.C");
}
```

PROOF – User Sandbox

- User's have their own sandbox on each worker node
- File transfers minimized
  - cache packages, selector
  - File integrity: MD5 checksums, timestamps
- Package manager to upload files or packages
  - binary or source
  - PAR (PROOF Archive, like Java jar)
    - provides ROOT-INF directory, BUILD.sh, SETUP.C to control setup in each worker
  - `TProof` API to handle all this

Typical query-time distribution



G. Ganis, ROOT05, 29 Sept 2005

Analysis Session Example

AQ1: 1s query produces a local histogram

AQ2: a 10mn query submitted to PROOF1

AQ3->AQ7: short queries

AQ8: a 10h query submitted to PROOF2

Monday at 10h15 ROOT session on my laptop

BQ1: browse results of AQ2

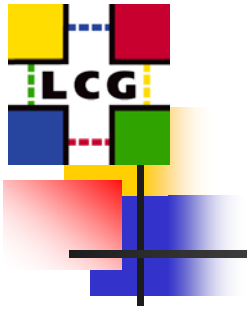
BQ2: browse temporary results of AQ8

BQ3->BQ6: submit 4 10mn queries to PROOF1

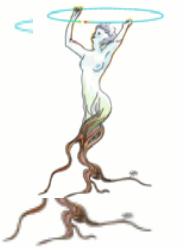
Monday at 16h25 ROOT session on my laptop

CQ1: Browse results of AQ8, BQ3->BQ6

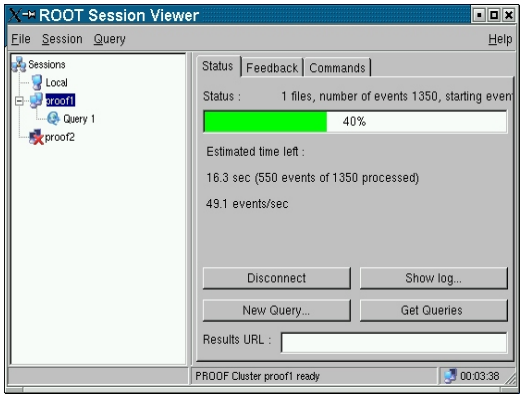
Wednesday at 8h40 Carrot session on any web browser



GUI manager

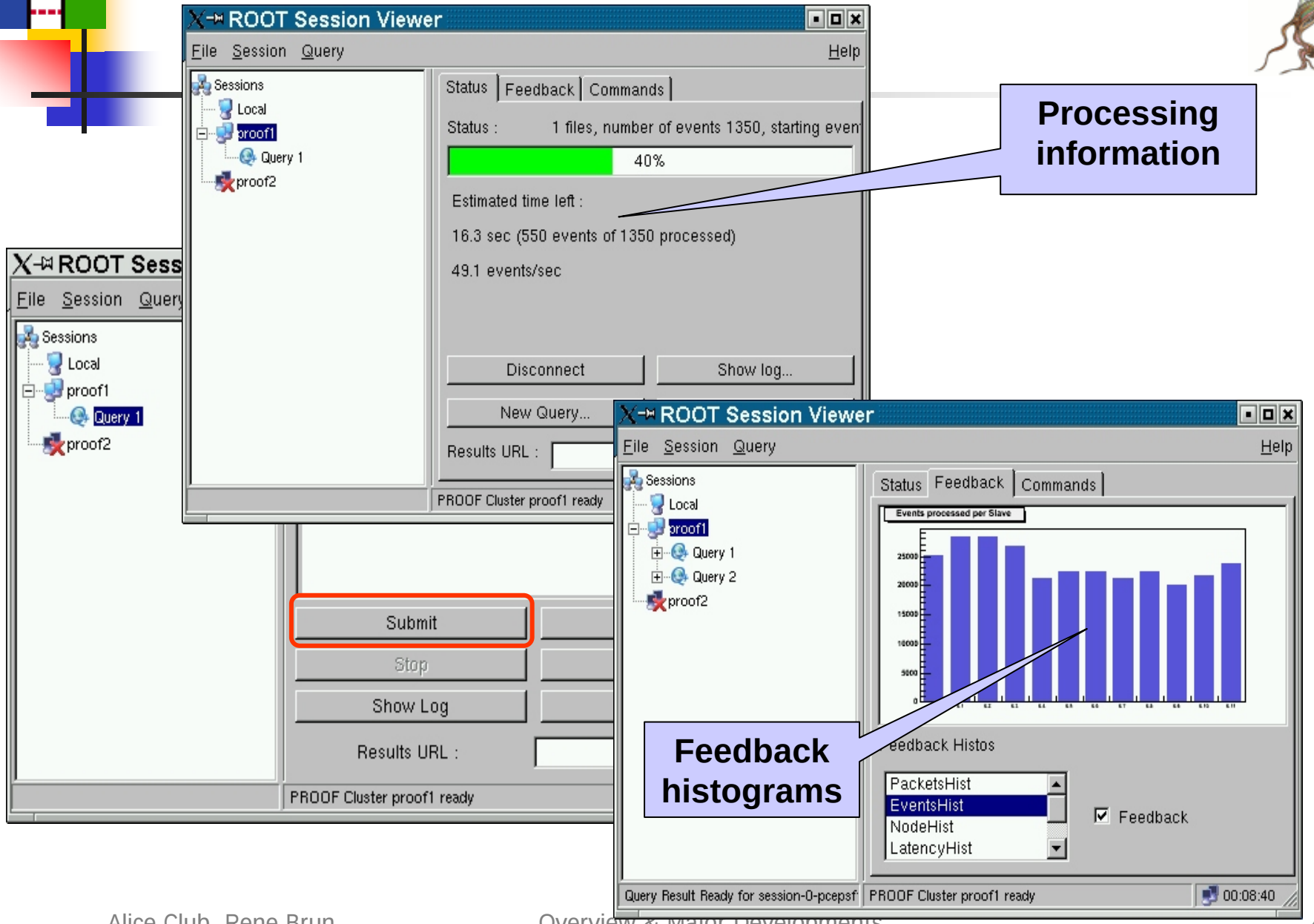


Allows full *on-click* control on everything



- define a new session
- submit a query, execute a command
- query editor
  - execute macro to define or pick up a **`TChain`**
  - browse directories with selectors
- online monitoring of feedback histograms
- browse folders with results of query
- retrieve, delete, archive functionality

Query processing



PROOF testbed

- Since September we have access to a dedicated PROOF farm (32 dual processor nodes) in 513.
- These nodes are slow machines (800 Mhz). They are good for developing the system.
- We will need a more powerful farm early next year to perform more realistic tests and welcome users to give feedback with concrete analysis tasks.

PROOF and XROOTD

- XROOTD is already playing a very important role in PROOF. It will continue to play a growing role.
- The PROOF and XROOTD teams are cooperating to get even more from XROOTD: caching, read ahead, new XROOTD services.
- Still a lot to do to have a good integration of XROOTD with other services like CASTOR.

Summary

- After 10 years of development, ROOT is widely used in HEP and elsewhere.
- The team has been extended with LCG2. It cooperates with many external developers.
- ROOT/SEAL merge is a success
- Consolidation phase for I/O and Trees
- Intensive developments in most packages
- Pushing PROOF data analysis model