



# ROOT 2018

EP-SFT, Axel Naumann, 2018-03-19

# Who is ROOT?

- Kim Albertsson
- Guilherme Amadio
- Bertrand Bellenot
- Philippe Canal
- Olivier Couet
- Enrico Guiraud
- Lorenzo Moneta
- Axel Naumann
- Danilo Piparo
- Oksana Shadura
- Yuka Takahashi
- Enric Tejedor
- Xavier Valls
- Vassil Vassilev
- Stefan Wunsch

# Main Contributors

- Brian Bockelman
- Zhe Zhang
- Sergey Linev
- Marsupial

# Alive?



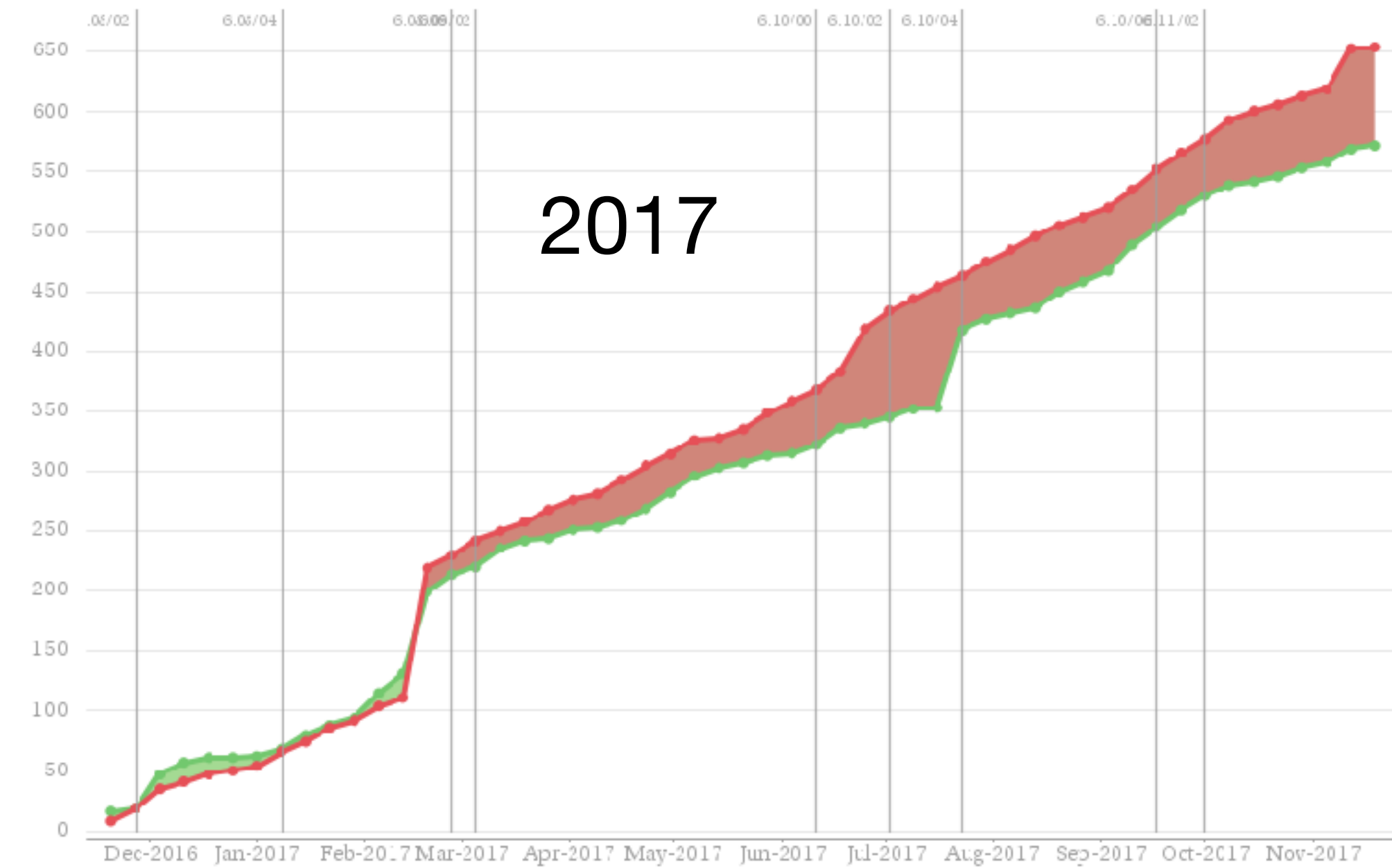
Very High Activity

- From OpenHub: "Activity level is relative to all projects on Open Hub. Only a small percentage of projects on Open Hub will display the Very High icon. Many important, highly regarded and popular projects may show lower activity levels"

# Jira, Coverity

+ backlog of 600 issues

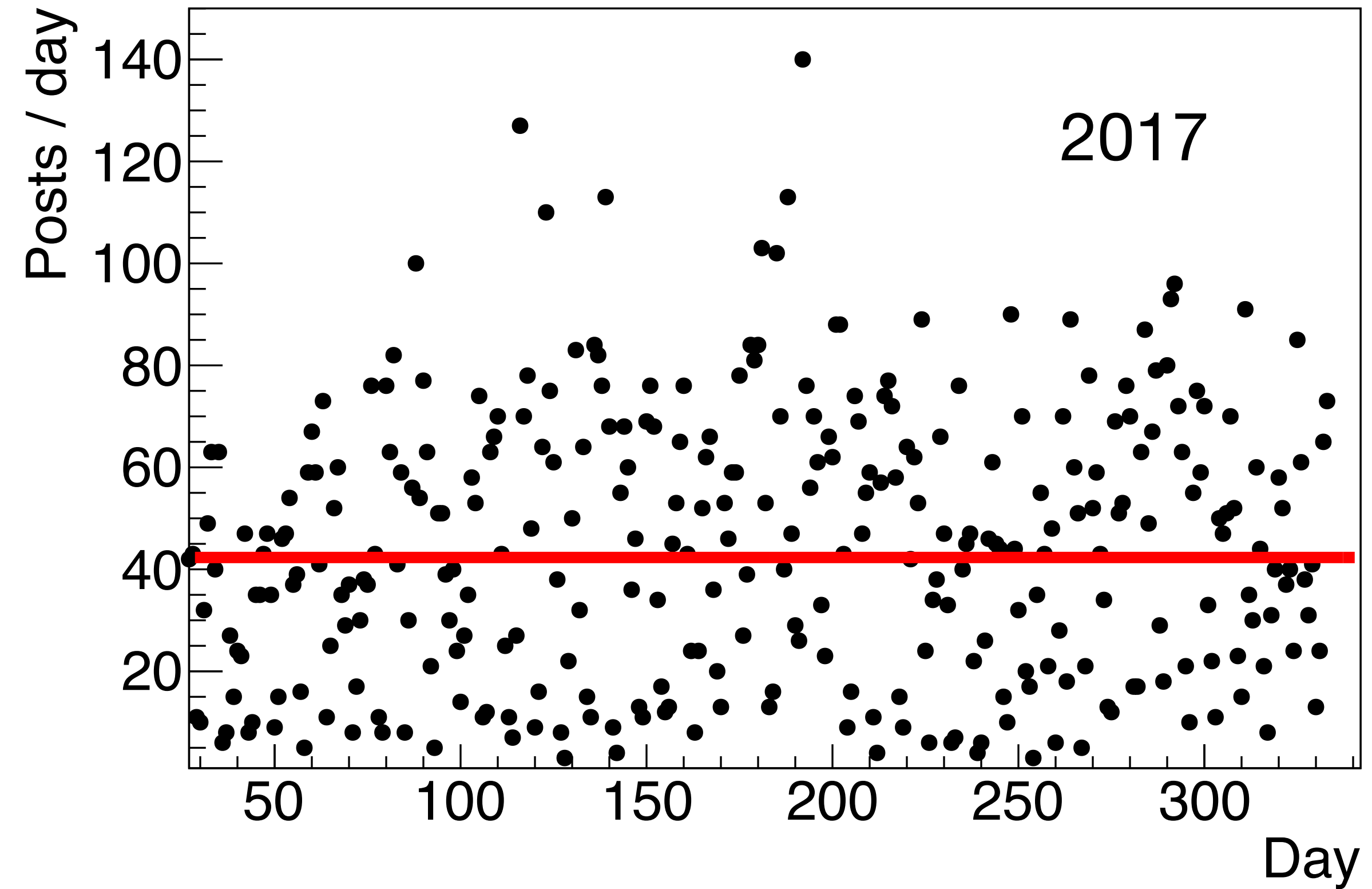
- 1230 open ROOT issues
- + Coverity (about 1000 issues)



# Forum



- Almost 50 posts per day
- More and more users



# ROOT & Experiments 2017

- ROOT Planning: regular meeting for two-way communication
- Invitations, presentations at experiments' software weeks
- Inter-experiment Machine Learning meetings
- Participation at experiments' software reviews

# Courses 2017

- Two full day ROOT CERN courses
- Sanofi Pasteur
- Summer student workshop
- Experiments' intro tutorials
- CERN School of Computing



# Conferences 2017

- Seven (!) ACAT presentations in 2017 (with proceedings)
- two/three “best poster” prizes for ROOT: Guilherme + Xavi!

# 2017 Highlights

- TDataFrame, TDataSource
- More parallelism: fitting, I/O (notably TBufferMerger), interpreter
- TMVA CNN + RNN, cross-validation; TFormula composition
- WebGUI prototype; several graphics improvements
- C++17, GCC7, Windows support
- rootbench.git
- Runtime modules

2017 Plan of Work:  
[https://indico.cern.ch/event/598604/  
contributions/2419269/attachments/  
1393870/2124204/ROOT-Plans-20170111.pdf](https://indico.cern.ch/event/598604/contributions/2419269/attachments/1393870/2124204/ROOT-Plans-20170111.pdf)

# Plan of Work 2018

- Drafted by team, collected feedback from contributors and experiments
- see [https://docs.google.com/spreadsheets/d/1pHFbCDazLF8wuICOW8C4G7IjOP\\_p5CTi94HdBh9shnI/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1pHFbCDazLF8wuICOW8C4G7IjOP_p5CTi94HdBh9shnI/edit?usp=sharing)
- Usual disclaimer: ROOT is inherently agile
  - we adjust to changing priorities
  - we expect shifts in this plan

# Outline

- Showing plans per ROOT part:
  - Math, I/O, Graphics, Parallel Analysis, Infrastructure
- Additional collaborative R&D projects, for instance with OpenLab

# The Foundation

# Our Baseline Job

- Support
  - forum
  - bug reports
  - emails
- Consulting
- Teaching

# Our Baseline Job

- Support
  - forum
  - bug reports
  - emails
- Consulting
- Teaching

# Our Baseline Job

- Support
  - forum
  - bug reports
  - emails
- Consulting
- Teaching



# ROOT & Experiments 2018

- ROOT Planning: regular meeting for two-way communication
- Invitations, presentations at experiments' software weeks
- Inter-experiment Machine Learning meetings
- Participation at experiments' software reviews

# Courses 2018

- Summer student workshop
- Experiments' intro tutorials
- CSC

# Conferences 2018

- Seven (!) ACAT presentations in 2017 (with proceedings)
  - two/three “best poster” prizes for ROOT: Guilherme + Xavi!
- At least seven (!) accepted CHEP presentations in 2018
- Shows ROOT's impact in community

Math

# Math Libraries

- Finalize parallelization of fitting
- Common PRNG library for the community
  - specifically for ROOT and simulation
  - parallelism-friendly, vectorized, state-of-the-art generators

# Fitting + Histograms

- Parallelization of relevant histogram operations
- Fitting of new ("v7") histograms

# TMVA

- Rework data transfer: `TMVA::Event`, build I/O, numpy conversion, replace `TTreeFormula` by `TDataFrame`
- CNN, RNN, DAE integration; improved interfaces to external tools, both eval and training in TMVA or only eval; better out-of-the-box cross-validation support
- BDT with internal parallelism, DL with GPU + CPU support, process parallelism for cross-validation + hyper-parameter tuning
- Documentation! Esp. users' guide

# Vectorization

- Numerical integration and differentiation
- ROOT to depend on VecCore; move vectorized, templated math there
- ROOT to be SIMD arch agnostic; have arch-specific implementation libs



I/O

# Features

- Support for `shared_ptr`, `variant`, `optional`
- I/O of interpreted classes
- new ("v7") TTree prototype

# Performance

- Reduction of virtual function calls
- Improvements to OptimizeBasket
- Offsets optimizations in data format
- Performance optimization in TBranchProxy / TTreeReader

# Compression

- LZ4 to become default
- Investigate keeping compression dictionary across baskets
- Investigate Zstd

# Parallelism

- Parallel unzipping
- Improved thread-safety / performance (e.g. RecursiveRemove)
- Improve parallel buffer / tree merge, task-oriented + performance

Graphics

# WebGUI Fit Panel: Server / C++

- New "v7" TCanvas, TPad, TFrame (coordinate systems), TDrawable
- Basic graphics primitives, e.g. TLine, TText and their attributes; styles
- Drawables for histograms, graphs etc

# WebGUI Fit Panel: Client / JS

- Painters for high-level objects (re-use JSROOT)
- Interactivity, sending updates to server
- Settle on solution for graphics in batch



# GUI

- UI (menus, buttons) implemented in OpenUI5
- Graphics editors: line, text etc attributes

# Geometry, Eve

- Implementation of a web-based geometry viewer
- Investigation of WebGUI-version of Eve

# Parallel Analysis

# TDataFrame

- Operations on collections: DeltaR etc
- Consider new data sources: e.g. Parquet, bulk I/O
- Decouple number of slots versus workers
- Usability improvements: error messages etc
- Support for new "v7" histograms
- Pass python callables as cuts, defines

# Task + MP Parallelism

- Documentation of threading behavior
- Executors: NUMA-aware, sequential, investigate distributed (PROOF successor); improved chunking
- Investigate PyROOTSpark

# Interpreter Bindings, Platforms

# PyROOT, cling

- PyROOT maintainance
- Rvalue reference support in PyROOT and TClass
- Conversion of Python iterables to C++ containers
- Memory buffers, less copied for C++ / python object passing
- Improved support for unloading in cling

# Windows

- 32bit with tests of supported packages passing
- Investigation of 64bit port



Infrastructure

# Build System

- Remove configure/make
- Redesign for modern, modular CMake
- Late build of sub-packages
- Deprecate parts of ROOT that have not had build support with CMake

# Monitoring

- Deploy Jenkins pipeline
- Continuous performance monitoring
- Add I/O related benchmarks to rootbench.git

# Runtime C++ Modules

- Reduce deserializations in ROOT and LLVM
- Rethink rootmap-related code in terms of cxxmodules
- Consider building modules independently of cling
- Track module-related regressions in ROOT and clang
- Define adoption path for experiments, provide migration support

# Misc

- Provide Prototype for ROOT-based package manager
- Provide Containers and Virtual Machines as development environments
- Decide on use of "Evolution" proposal system
- Continue offloading clang patches

# Summary

# Shopping List or Vision?

- Both: you should be able to discern the focus areas
  - vision is directing development items
- Incredible stretch and work load
  - vision provides focus and priorities

# Where are we going?

- Remain at the heart of HEP data analysis
- Rechargeable batteries included
- ongoing rejuvenation to cover the next 20 years



# ROOT's Theme

- Simplicity
- Robustness
- Speed