JavaScript ROOT

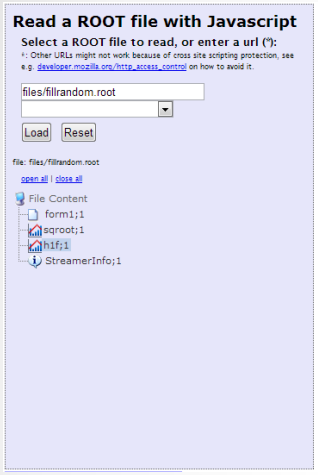
https://root.cern.ch/js/

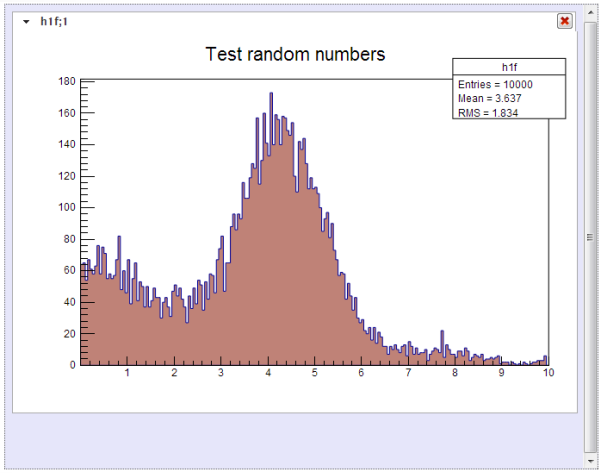
Bertrand Bellenot (CERN)
Sergey Linev (GSI)

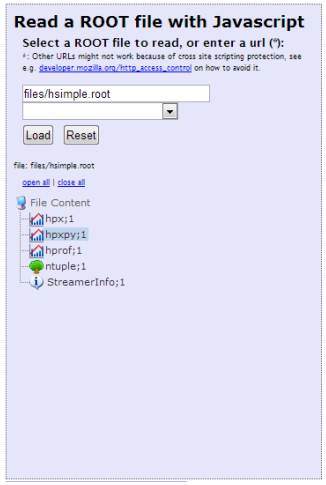
JSRootIO project

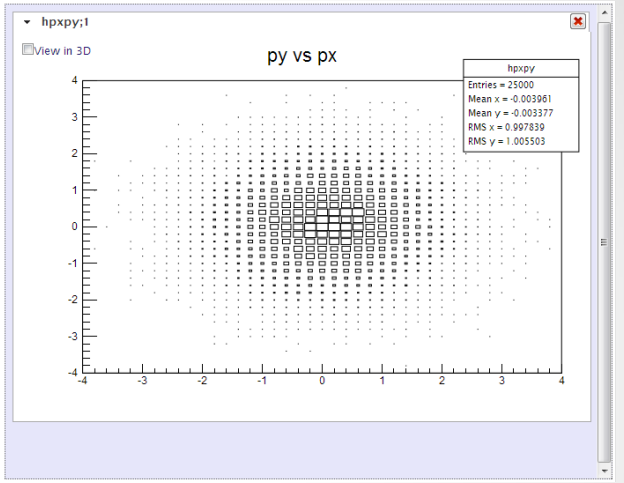
- Original project from Bertrand Bellenot
  - was presented at the ROOT workshop 2013
- ROOT I/O in JavaScript
  - reading ROOT objects using streamer infos
    - many exceptions due to custom streamers
  - source for the project name - **JSRootIO**
- Graphic with d3.js, three.js and jQuery.js
- Navigating the ROOT files content and displaying objects in modern web browsers

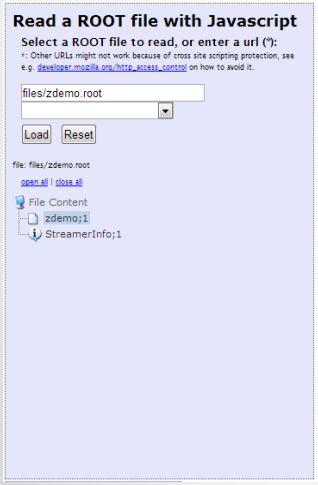
JSRootIO screenshots

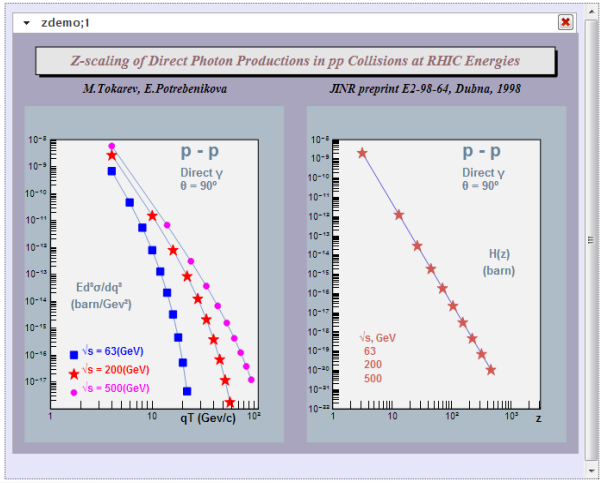


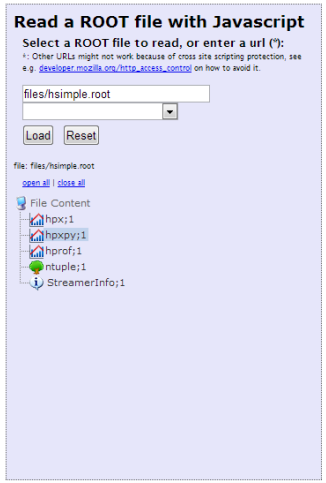


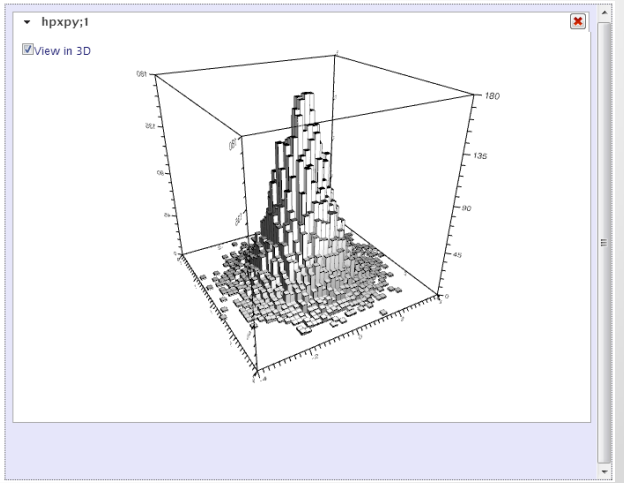












JSRootIO and http server

- Could one use JSRootIO with online ROOT application?
- In general yes, but many hidden problems and caveats:
  - difficulty with fixed HTML design
  - lack of objects update
  - flexible API was missing
- There was an intermediate solution with many workarounds until a decision was taken to redesign JSRootIO completely

JavaScript ROOT

- Preserve old functionality, including look-and-feel
- Redesign was focused on:
  - modularity
  - clear API
  - more interactive features
  - more supported classes
  - support of user classes
- Project was renamed to **JSROOT**
  - binary ROOT files reading is an optional part of the project

Main features

JavaScript ROOT provides:

- Objects reading from binary and JSON ROOT files
- Display for popular ROOT classes in web browsers
- Flexible API for usage in other projects

How to use JSROOT?

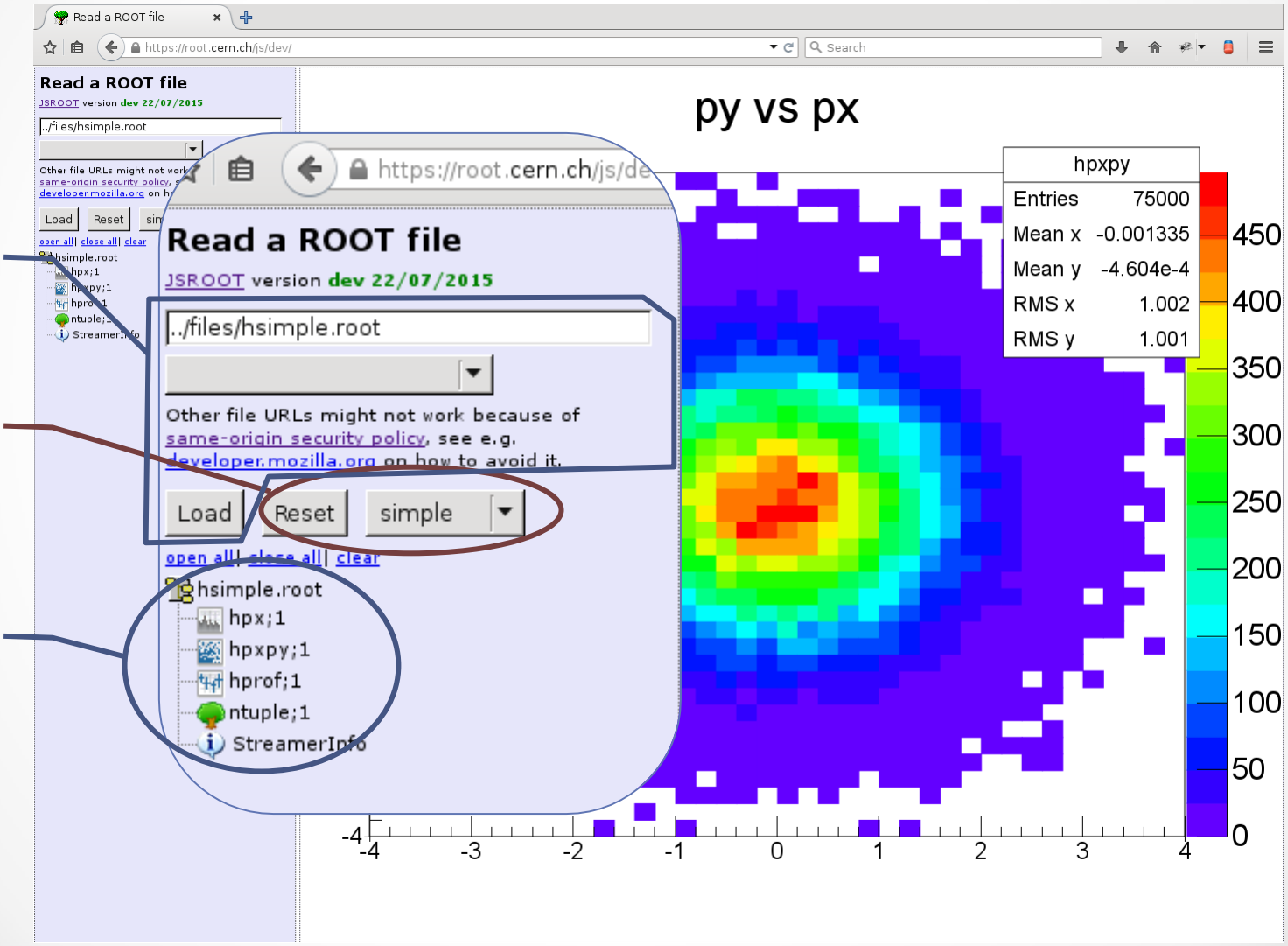
- As before, for interactive browsing of ROOT files
  - open JSROOT web page https://root.cern.ch/js/latest/
  - load file(s) from web
  - show content of the files
  - display objects from the files

User interface

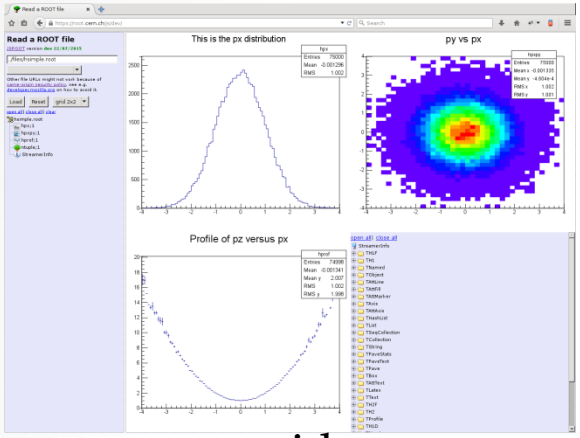
file loading

layout selection

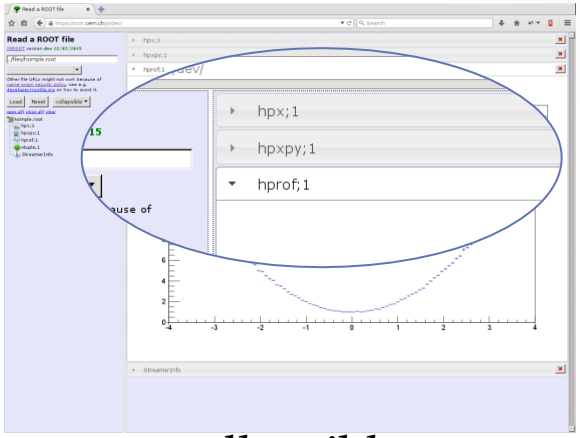
file content



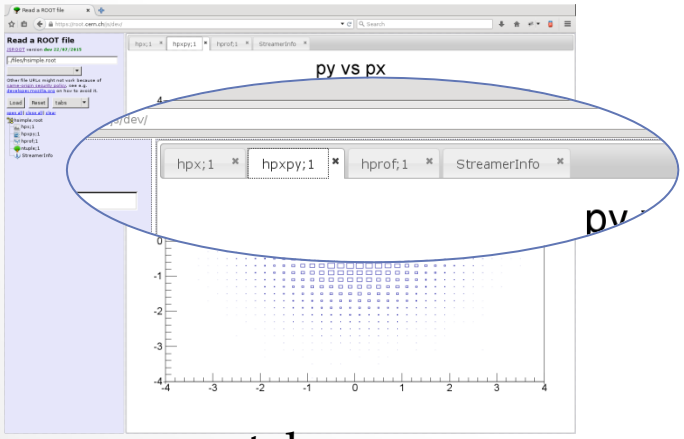
Different layouts



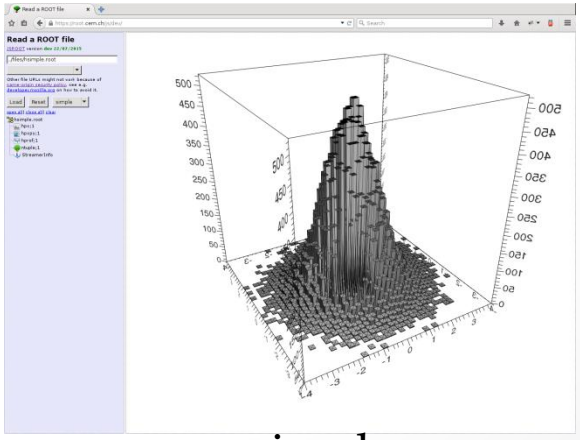
grid



collapsible

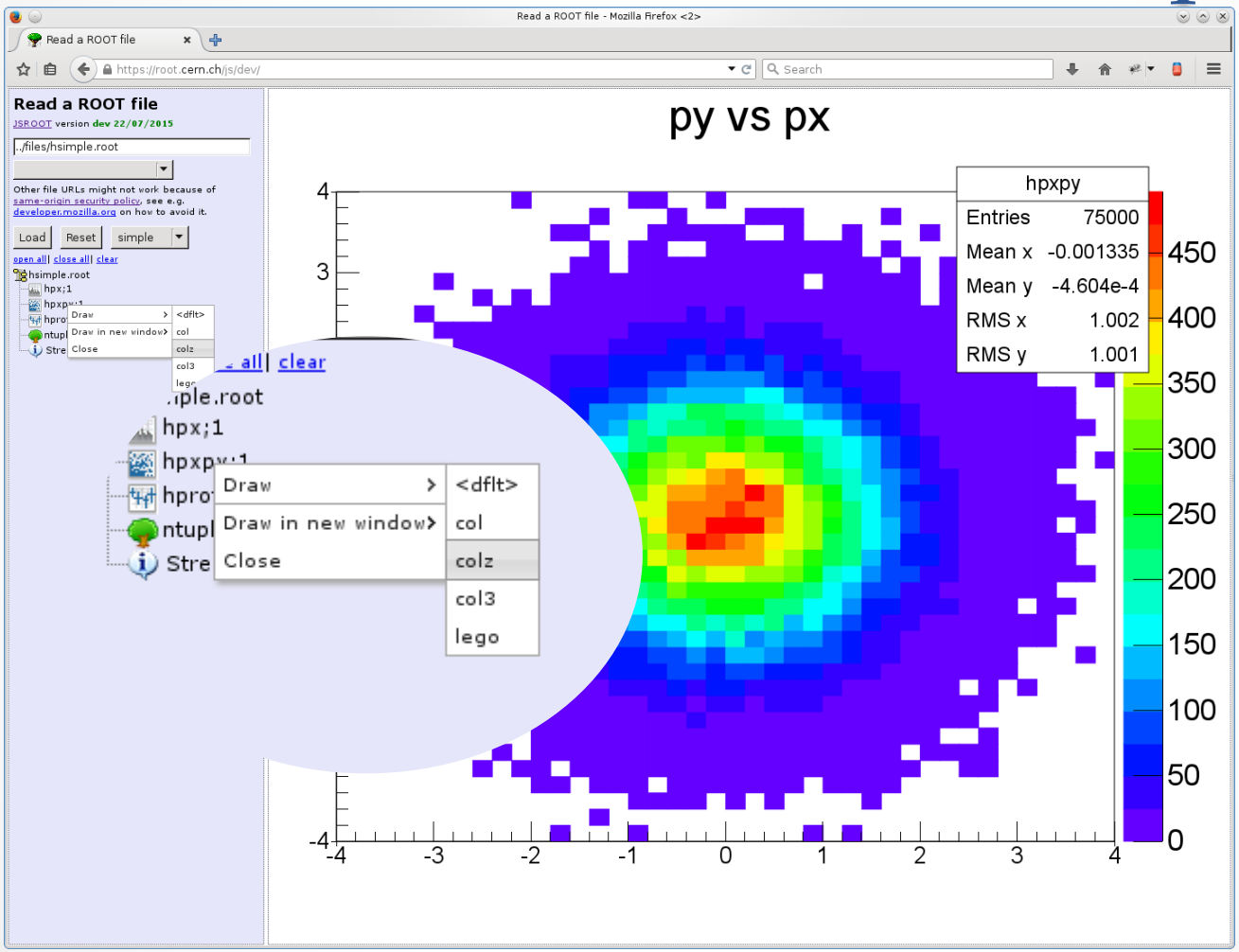


tabs

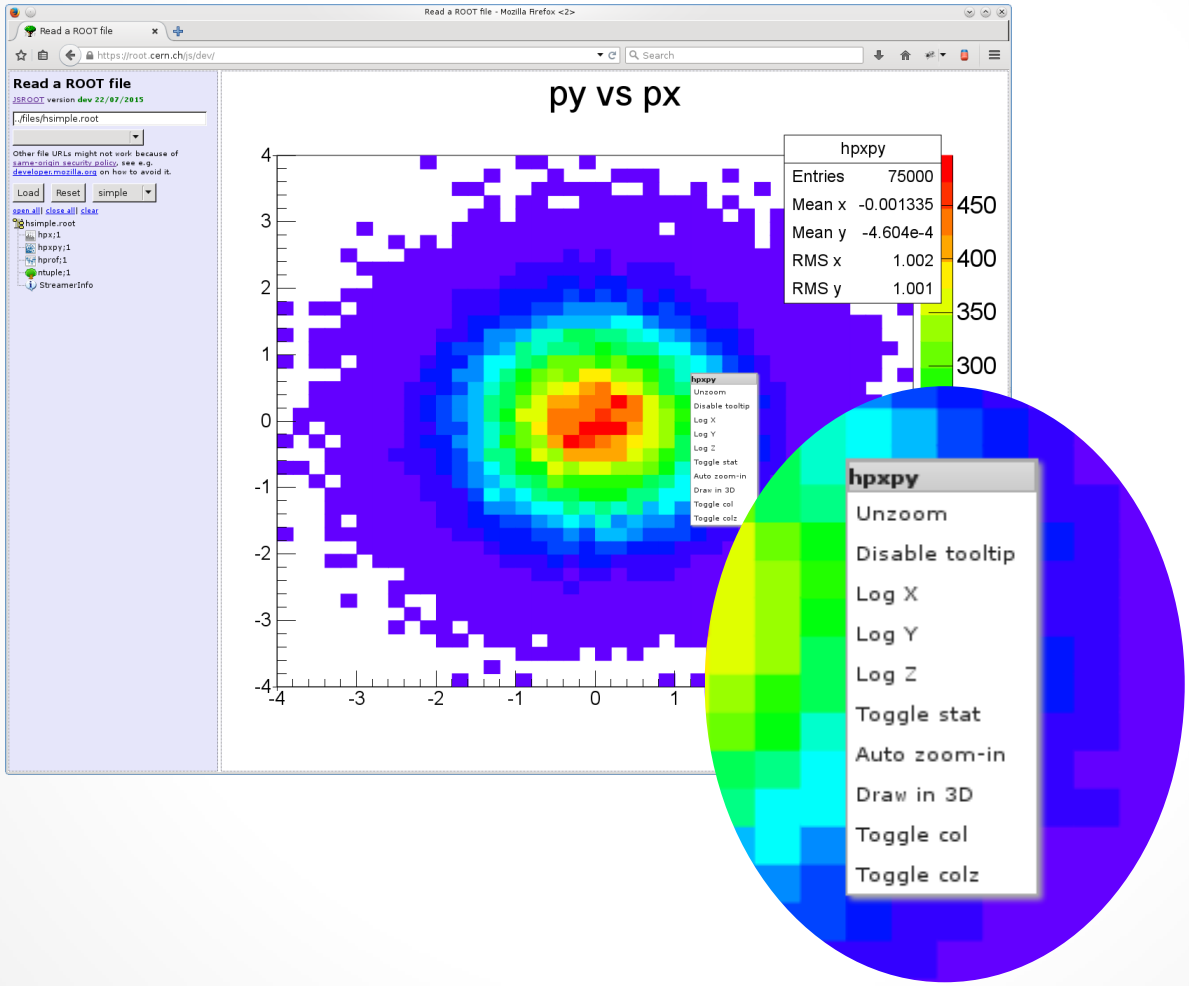


simple

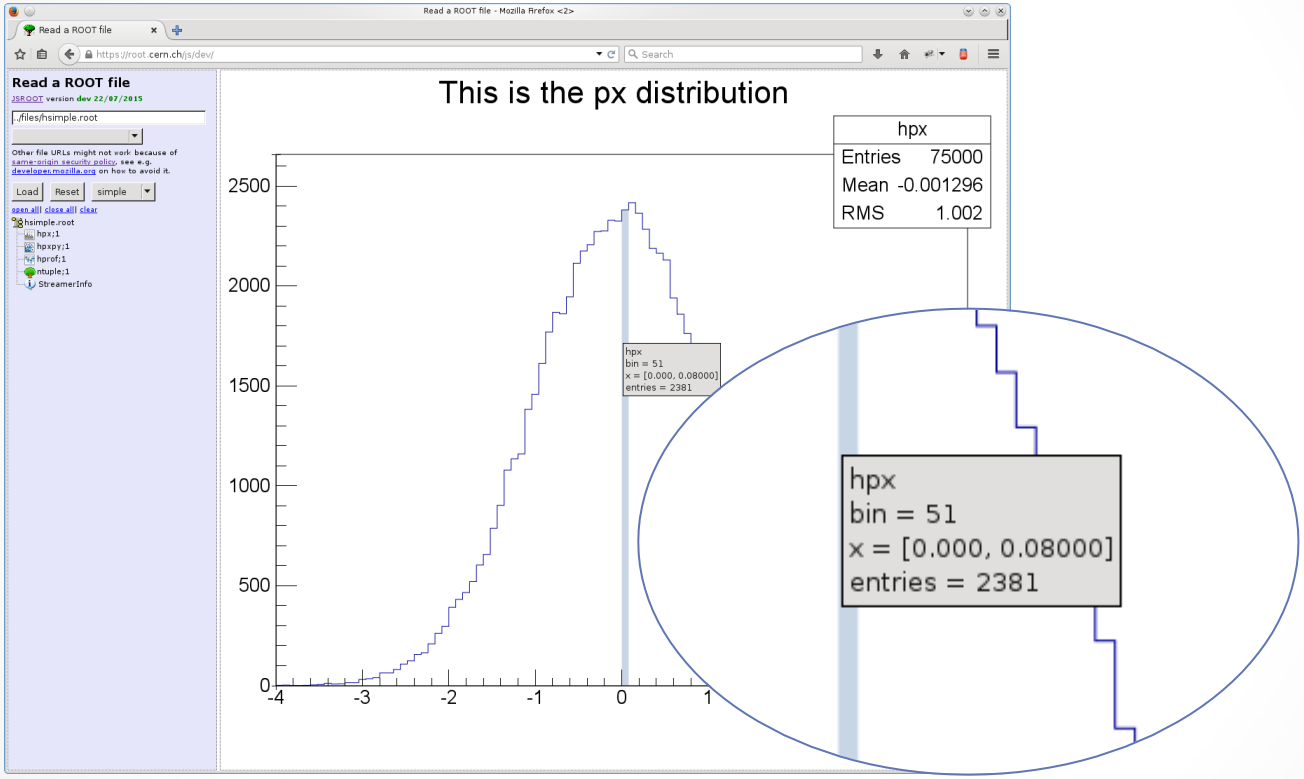
Context menu with draw options



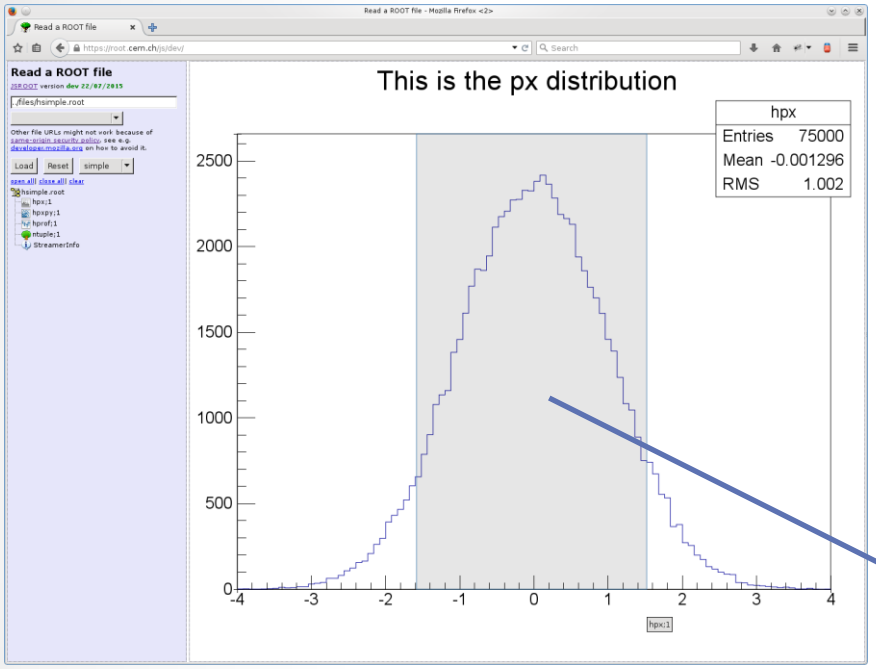
Context menu for drawn object

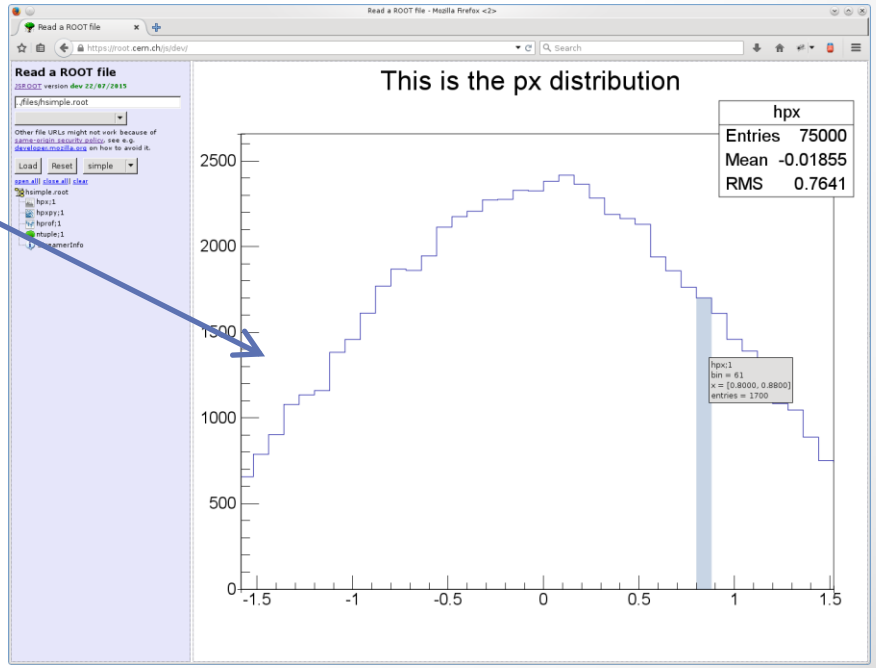


Informative tooltips



Intuitive zooming





How to share results?

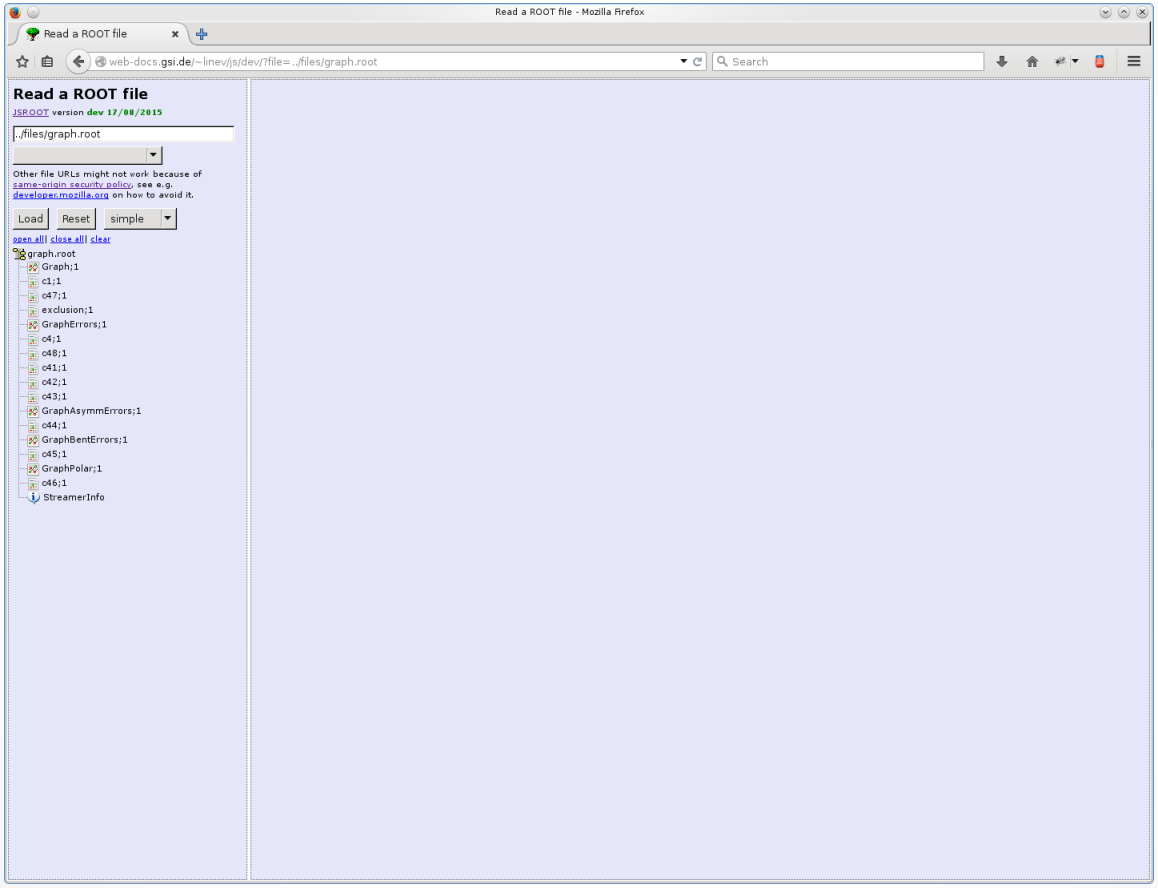
- Before
  - create and send PNG image (static)
  - or create and send ROOT file with canvas (interactive)
    - one requires ROOT installed everywhere
- With JSROOT
  - copy your ROOT file on web server and send link to the file
  - open main page https://root.cern.ch/js/latest/
  - enter file name (like https://root.cern.ch/js/files/hsimple.root)
  - find and draw histogram or canvas
- Same actions repeat many times again ☹
  - interactive not always mean better
  - are there alternatives?
- Solution - use JSROOT UI with URL parameters!

URL parameters in JSROOT

file(s)	– name of file(s) to open
json	– name of json file to open
item(s)	– item name(s) to display
opt(s)	– drawing option for the item(s)
layout	– layout for drawings like grid or tabs
nobrowser	– do not display objects hierarchy
mathjax	– enable usage of MathJax.js
interactive	– enable/disable interactive features
load	– name of extra JavaScript to load
optimize	– drawing optimization (0: off,1: large histos, 2: always)

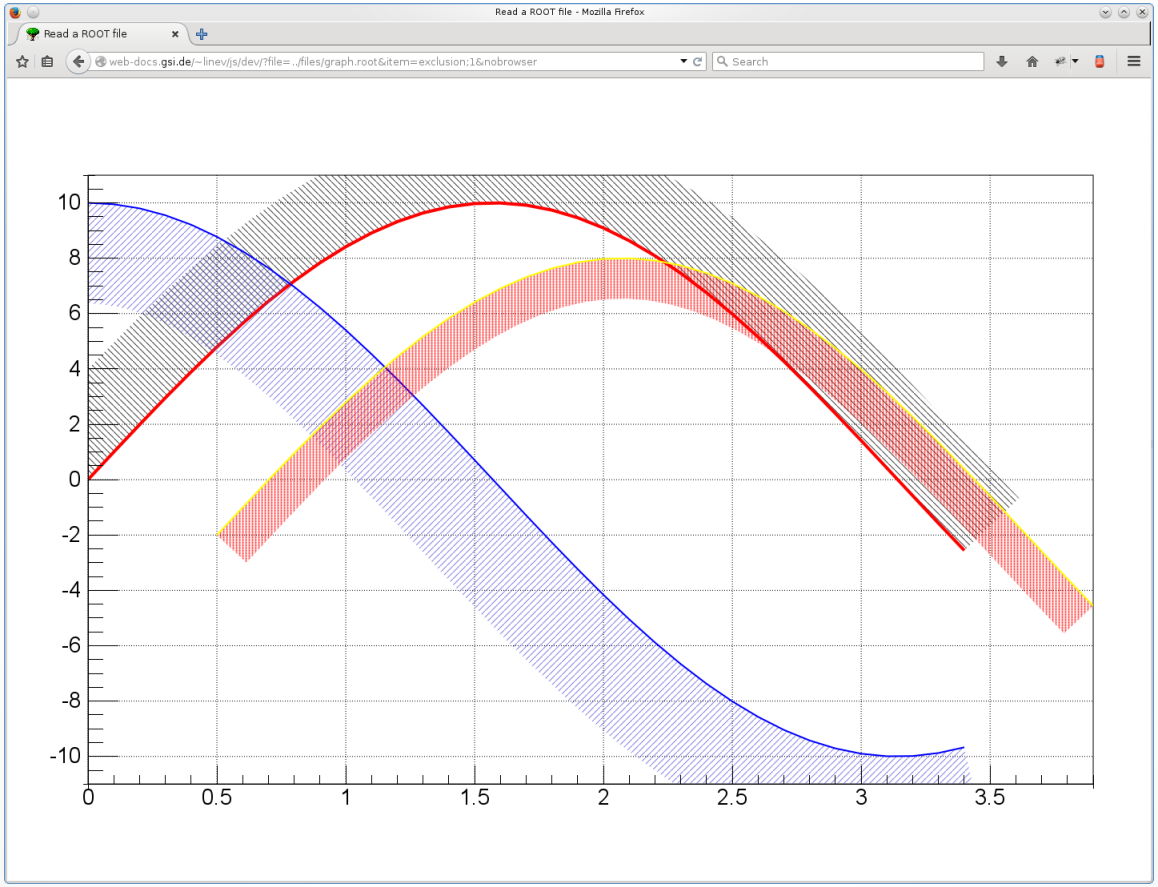
URL: open file

https://root.cern.ch/js/latest/?**file**=../files/graph.root



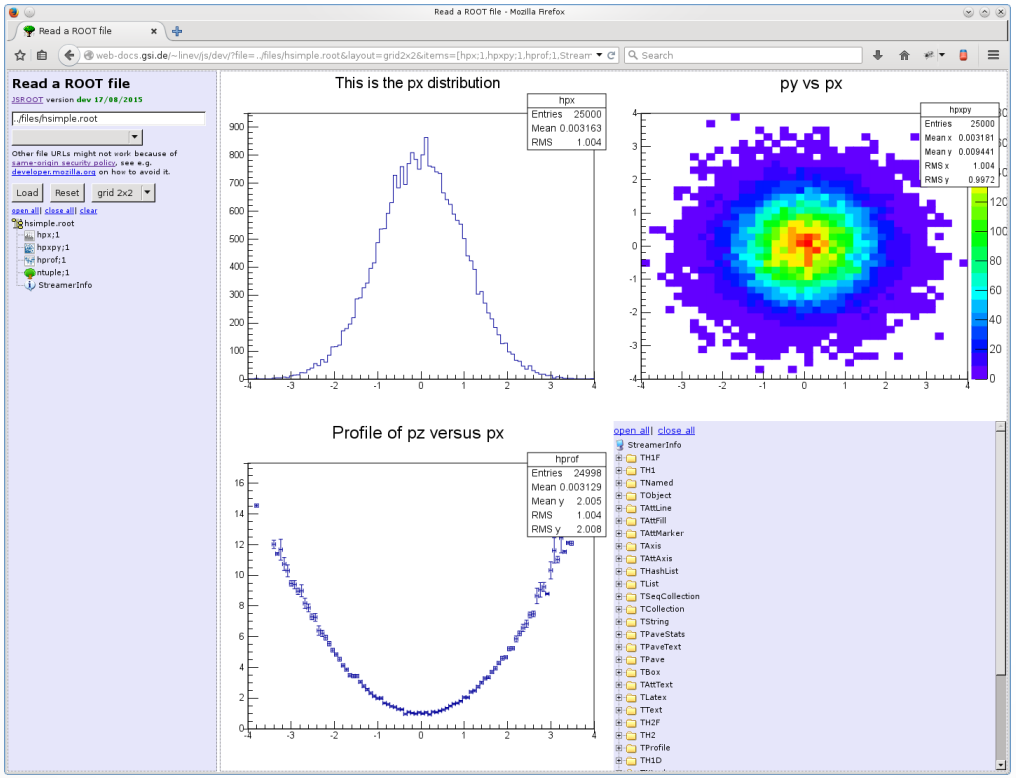
URL: display canvas from file

https://root.cern.ch/js/latest/?**file**=../files/graph.root&**item**=exclusion;1&**nobrowser**



URL: display several items

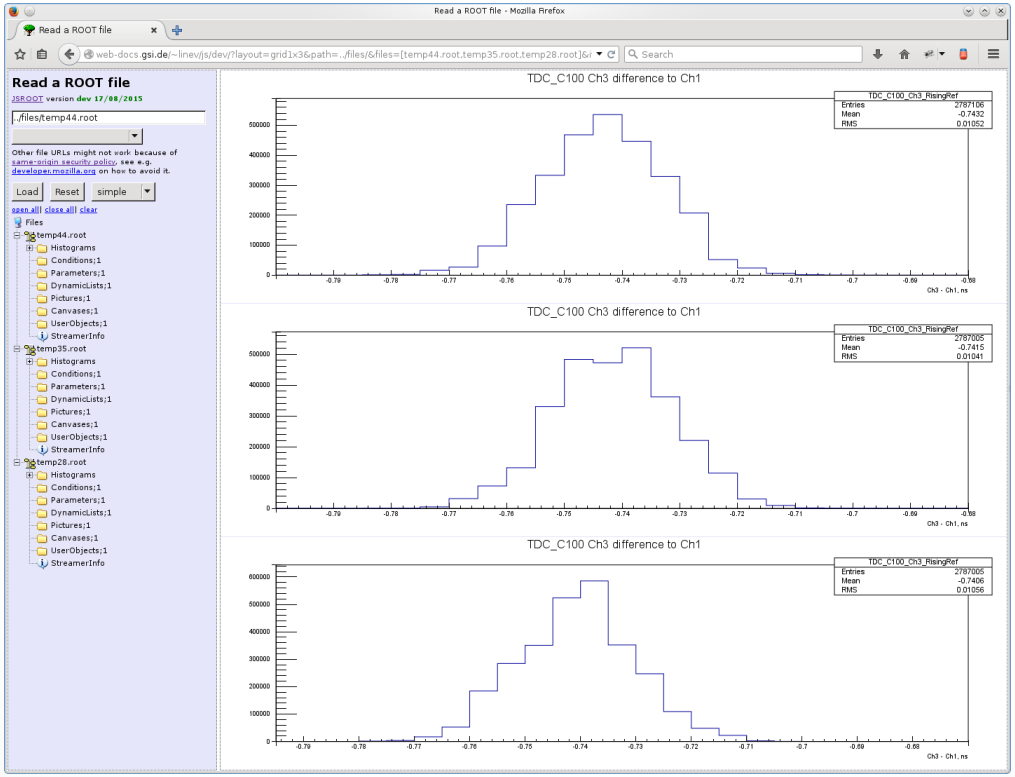
https://root.cern.ch/js/latest/?**file**=../files/hsimple.root&**layout**=grid2x2&**items**=[hpx;1,hpxpy;1,hprof;1,StreamerInfo]&**opts**=[hist,colz,e1,any]



http://bit.ly/1EBp349

URL: display histograms from different files

https://root.cern.ch/js/latest/?**layout**=grid1x3&**path**=../files/&**files**=[temp44.root,temp35.root,temp28.root]&**items**=[temp44.root/Histograms/TDC_C100/Ch3/TDC_C100_Ch3_RisingRef;1,temp35.root/_same_,temp28.root/_same_]&**opts**=[autozoom,autozoom,autozoom]



http://bit.ly/1L5cvyJ

I/O improvements

- Make logic closer to original ROOT I/O
  - introduce JSROOT.TBuffer class
  - always use checksum to verify content
- Handle all custom streamers in central place
  - all kind of ROOT collections
  - TCanvas, TObjString, TStreamer... classes
  - make it easier to support user classes with custom streamers
- support ROOT4, ROOT5 and ROOT6 binary files
- support files reading from other web server
  - CORS headers should be enabled on the server
  - one can read files from local file system
- I/O fully independent from graphics
  - vice versa is also true

Graphics improvements

- Full code reorganization
- Introduce painter classes
  - somehow similar to original ROOT
- Make several SVG layers
  - axis, main drawing, labels
  - easier to overlap objects
- Comfort zooming and stat box update
- Context menu for additional functionality
- Significant performance increase
- Use of MathJax.js for equation drawings

Supported ROOT classes

- histograms:
  - TH1, TH2, TH3, TProfile
- graphs:
  - TGraph, TCutG, TGraphErrors, TGraphAssymErrors, TGraphBentErrors
- superposition:
  - THStack, TMultiGraph
- functions:
  - TF1
- text:
  - TLatex, TMathText, TPaveText, TPaveStats, TPaveLabel
- containers:
  - TCanvas, TPad

Modularity

- Code divided on several modules
  - **core**, **2d**, **3d**, **io**, **gui**
- Modules loaded when required
  - in simple case only three JSROOT scripts are loaded instead of 10 before
  - could be specified when loading central JSROOT script

```
<script type="text/javascript“
  src="https://root.cern.ch/js/latest/scripts/JSRootCore.js?2d&io"></script>
```

- One could use require.js (optional)
  - example https://root.cern.ch/js/latest/demo/example_require.htm
- Minified version of scripts are provided

Use in other HTML pages

- Simplest solution - <iframe> tag

```
…
<iframe width="800" height="500"
src="https://root.cern.ch/js/latest/?file=../files/hsimple.root&
item=hpx;1&nobrowser">
</iframe>
…
```

- Not the first choice when many objects should be shown on the same page

Use in other HTML pages

- Load required functionality:

```
<script type="text/javascript"
   src="https://root.cern.ch/js/latest/scripts/JSRootCore.js?2d&io"></script>
```

- Provide place for drawing object:

```
<div id="drawing" style="width:800px; height:600px"></div>
```

- Retrieve object and call:

```
JSROOT.draw("drawing", obj, "colz");
```

Display object from ROOT file

```
var filename = “https://root.cern.ch/js/files/hsimple.root";
new JSROOT.TFile(filename, function(file) {
    file.ReadObject("hpxpy;1", function(obj) {
        JSROOT.draw("drawing", obj, "colz");
    });
});
```

- See https://root.cern.ch/js/latest/demo/example_file.htm

Display object from JSON file

- TBufferJSON can create JSON representation
  - no need for binary ROOT I/O in JavaScript
  - more details in the THttpServer presentation on Friday

```
JSROOT.NewHttpRequest("hpx.json", "object", function(obj) {
    JSROOT.draw("drawing", obj, "hist");
}).send();
```

- See https://root.cern.ch/js/latest/demo/example_json.htm

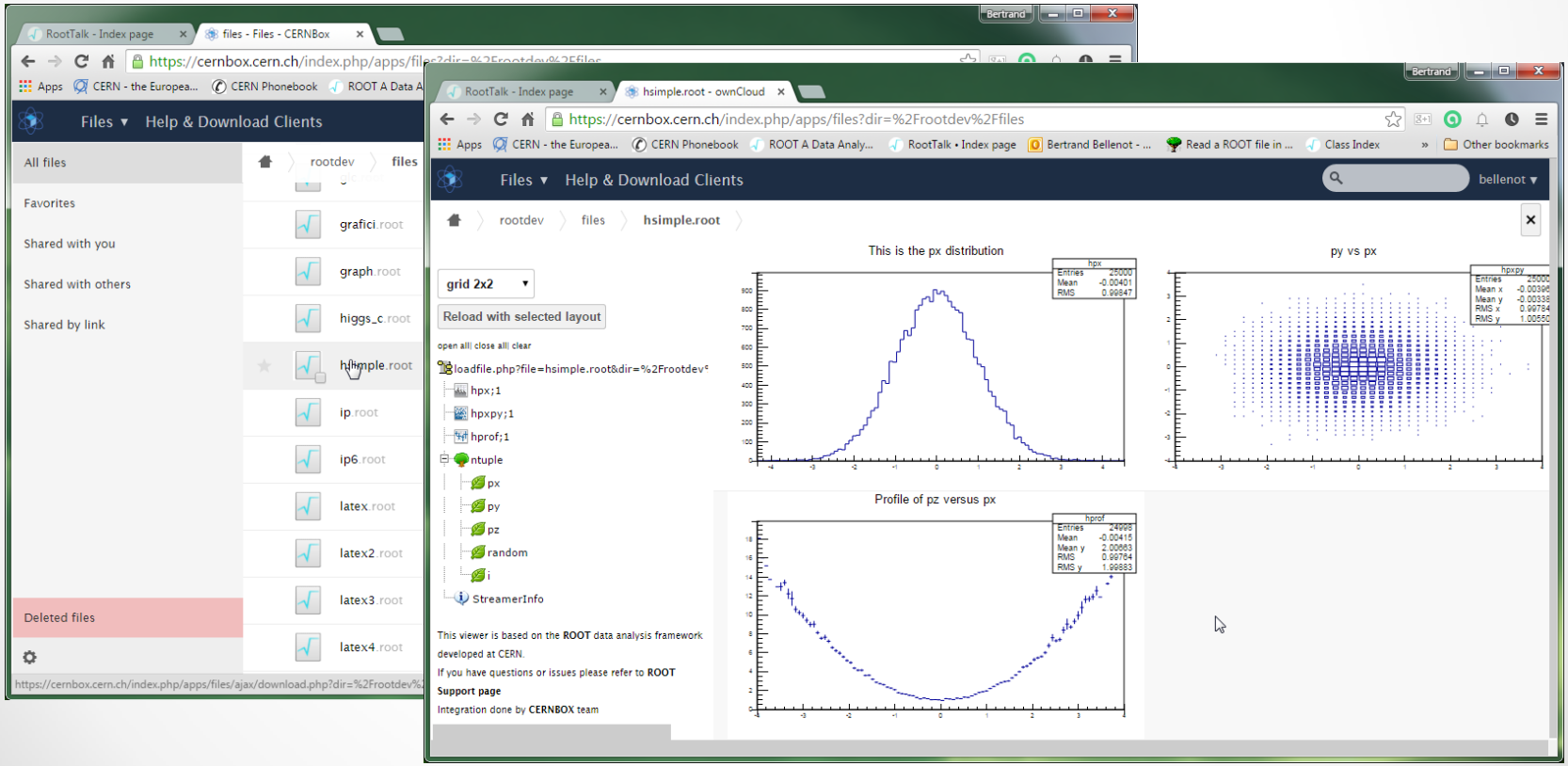
Update drawing from JSON

```
var cnt = 0;
setInterval(updateGUI, 2000);
...
function updateGUI() {
   var addr = "root" + (cnt++ % 20) + ".json";
   JSROOT.NewHttpRequest(addr, "object", function(histo) {
      JSROOT.redraw("drawing", obj, "hist");
   }).send();
}
```

- See https://root.cern.ch/js/latest/demo/demo.htm

CERNBox integration

CERNBox provides a functionality analogous to Dropbox™ or similar system, and is managed by CERN IT department (http://cernbox.web.cern.ch)
It now integrates JSROOT, allowing to display ROOT files content



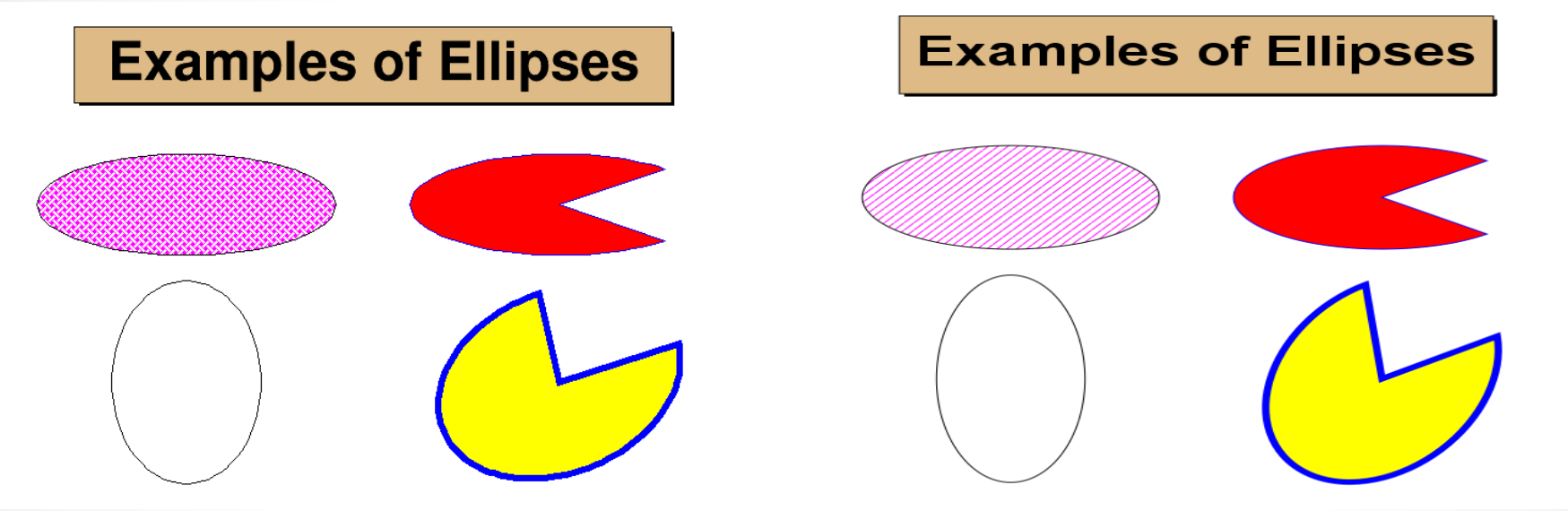
Support of user classes

- One needs to implement and register a drawing function
- Load the script together with JSROOT
- If necessary, provide a custom streamer for it

- Example with TEllipse class
  - JavaScript code (~70 lines)
    - http://jsroot.gsi.de/dev/demo/ellipse.js
  - Canvas from ROOT reference
    - http://jsroot.gsi.de/dev/index.htm?**file**=../files/ellipse.root&**item**=c1;1&**load**=demo/ellipse.js

- More examples in go4 framework
  - see Joern talk on Friday

TEllipse example



ROOT canvas

JSROOT canvas

Useful links

- Developers repository
  - https://github.com/linev/jsroot
- Latest stable version in ROOT
  - $ROOTSYS/etc/http
- All versions with documentation and examples:
  - https://root.cern.ch/js/
  - http://jsroot.gsi.de/