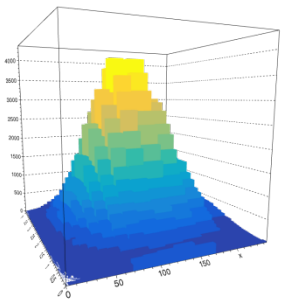
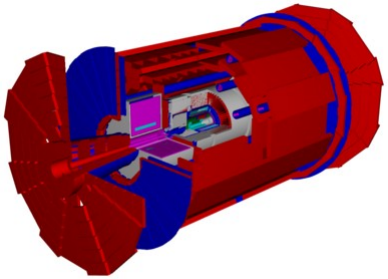
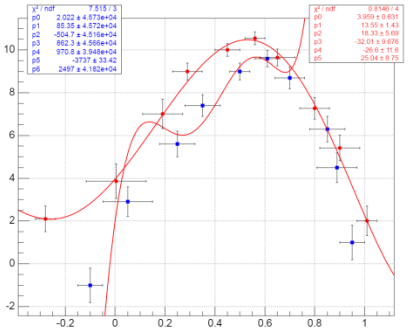
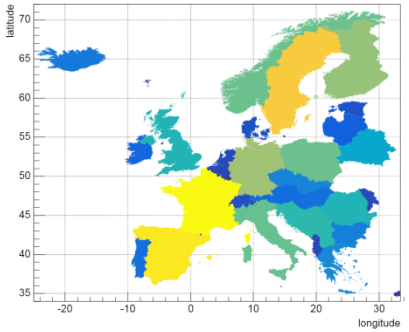
web-based graphics and GUI in ROOT7

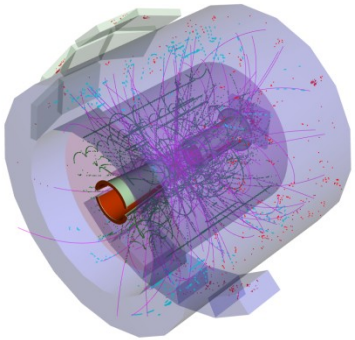




Sergey Linev (GSI, Darmstadt)







ROOT graphics

- ROOT graphics has more than 20 years of history
  - a lot of powerful components
  - but also a lot of caveats
- Make full redesign in ROOT7
  - portable
  - support multithreading
  - allow multiple views
  - remote display for all platforms

ROOT7 graphics

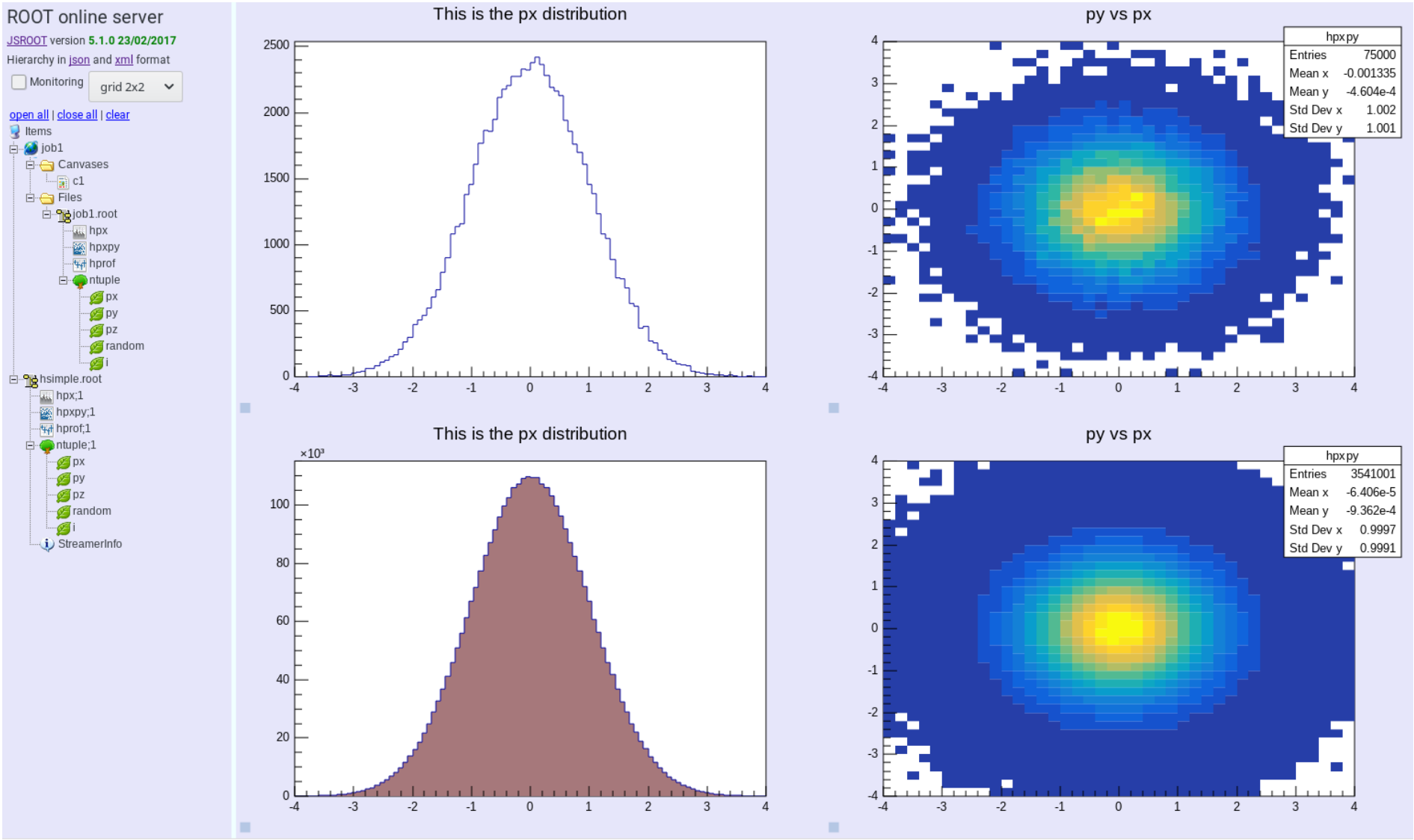
- Web-based
  - C++ server
  - JavaScript clients

- Reuse existing components
  - *THttpServer* for communication
  - *TBufferJSON* for I/O
  - *JSROOT* as code base for clients

THttpServer

- http access to running ROOT application
- civetweb or fastcgi
- visualization and live update of ROOT objects
- execution of commands and methods
- hierarchy display of objects
- possibility for fully custom UI

https://root.cern/js/latest/api.htm#httpserver_c_read_file



New in THttpServer

- Full support of websockets
  - text or binary data exchange

- Long polling as fallback
  - when websockets blocked
  - use async http requests
  - https://www.pubnub.com/blog/2014-12-01-http-long-polling/

TBufferJSON

- Converts any streamable object into JSON
- ROOT I/O remains fully on server side
- Support of custom streamers
- Optional array compression

TString json = TBufferJSON::ToJSON(obj)

var obj = JSROOT.parse(json)

- Size of gzip compressed JSON data comparable with size of binary ROOT data

New in TBufferJSON

- Now also objects reading from JSON

```
var json = JSROOT.toJSON(obj)
```

```
UserClass *ptr = nullptr;
TBufferJSON::FromJSON(ptr, json);
```

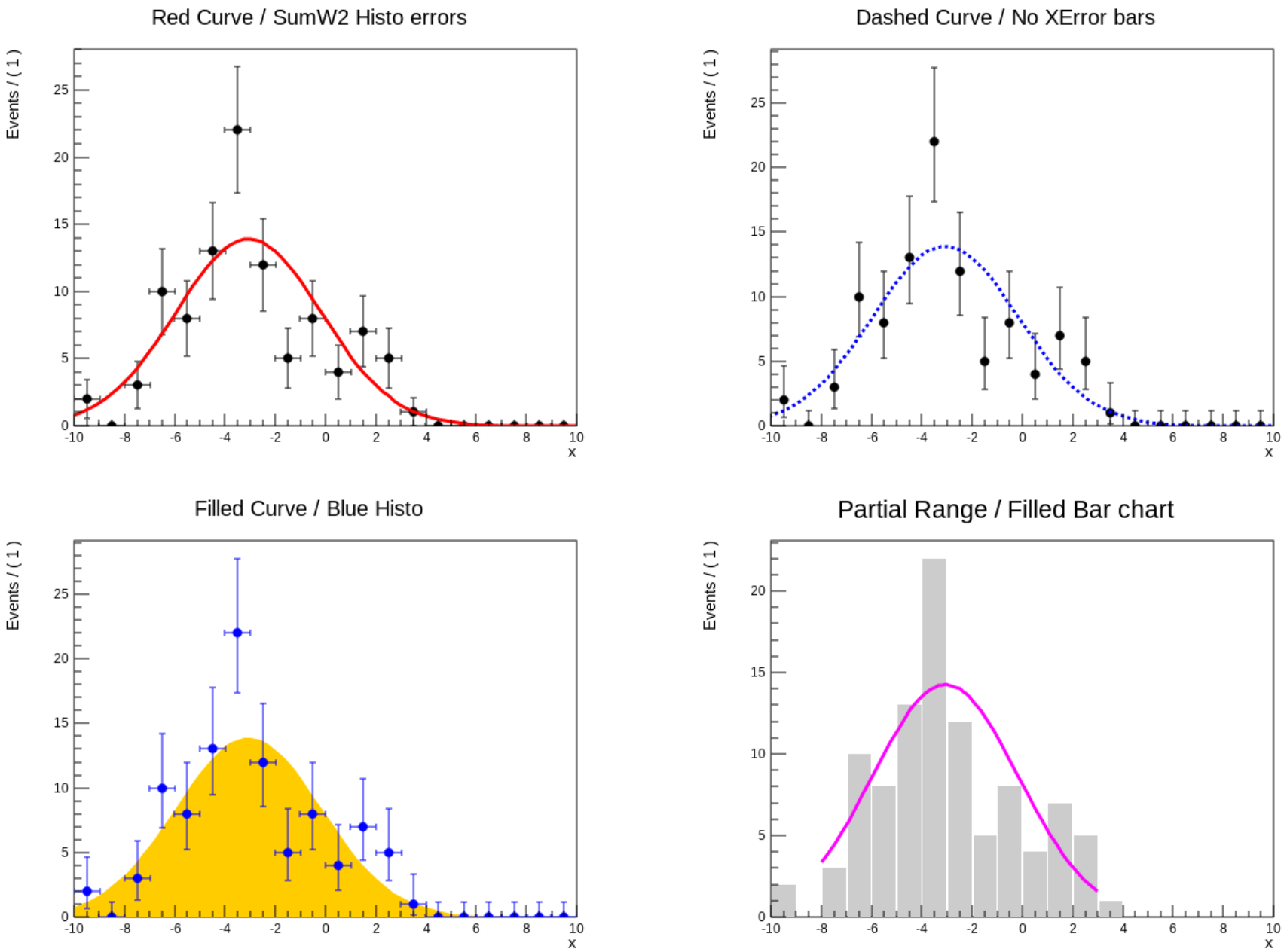
- Enables data exchange between C++ server and JavaScript-based clients

JavaScript ROOT

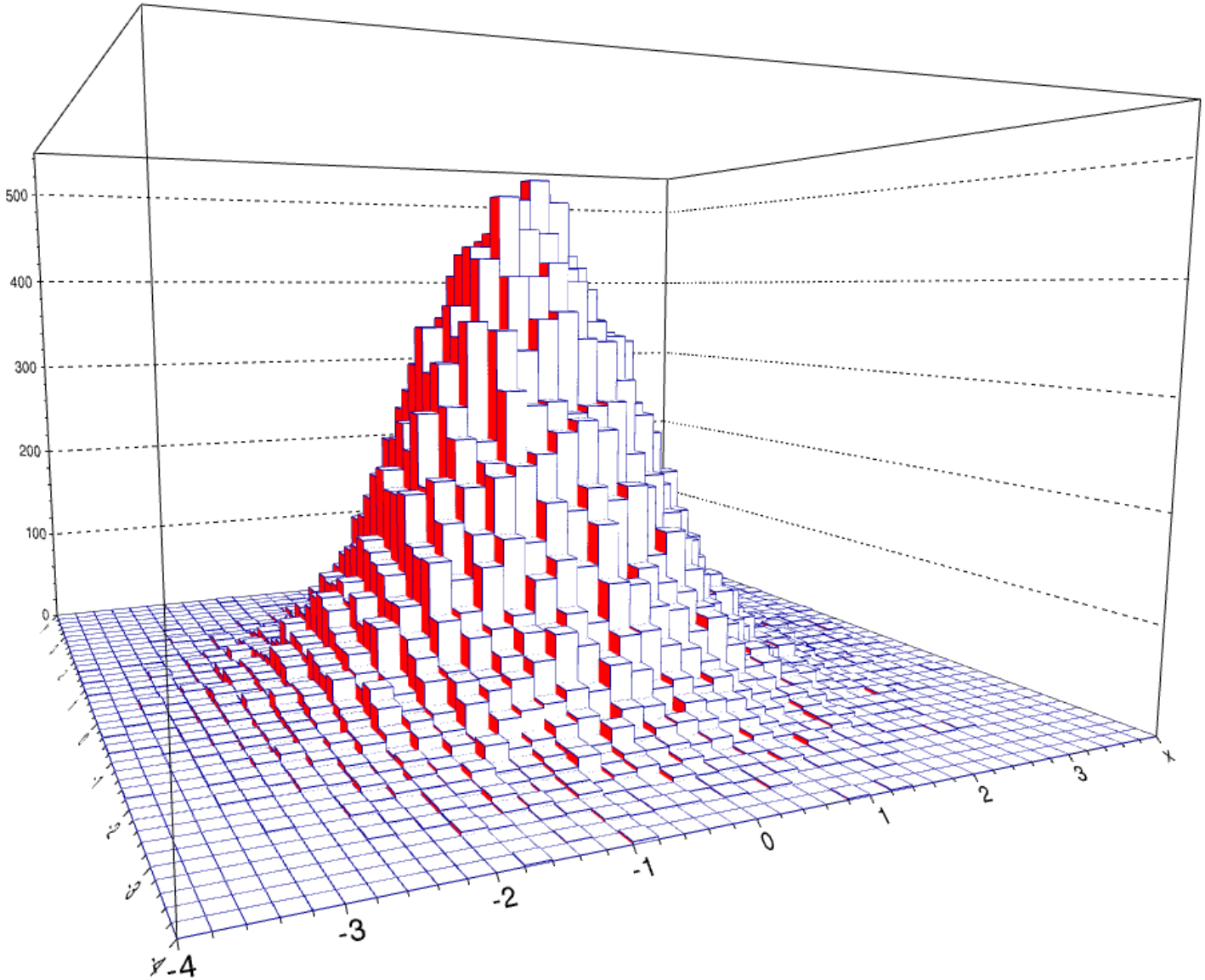
- ROOT objects display in web browsers
- Full support of ROOT I/O, including TTree
- JSON for any streamable ROOT object
- User interface for the THttpServer

- Developed since 2012
  - https://root.cern/js/
  - https://github.com/root-project/jsroot

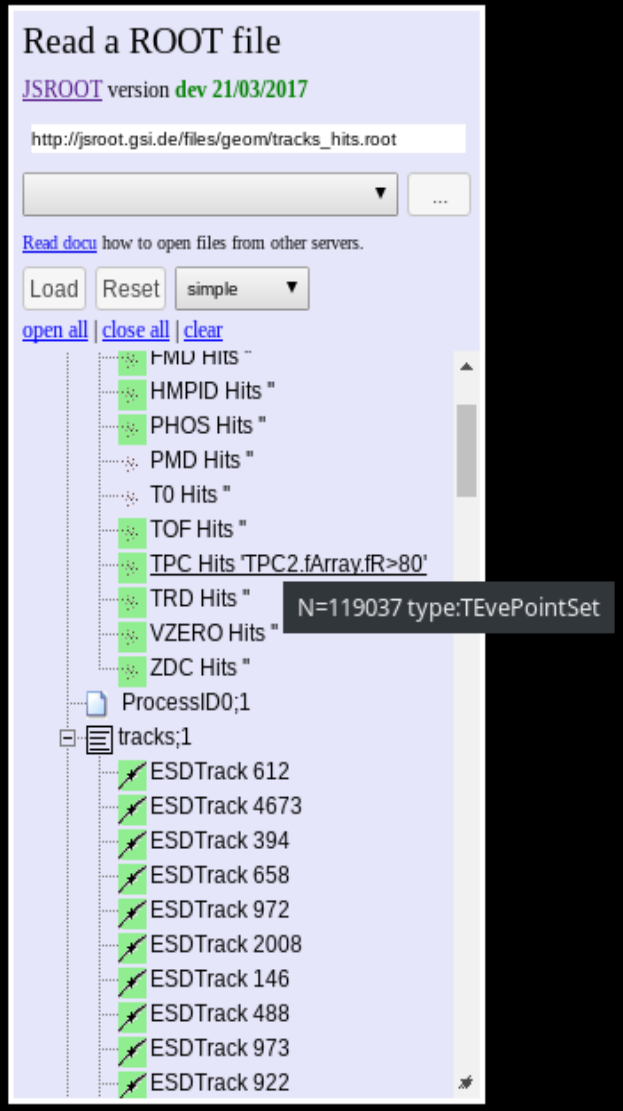
https://root.cern/js/latest/examples.htm#tcanvas_roofit

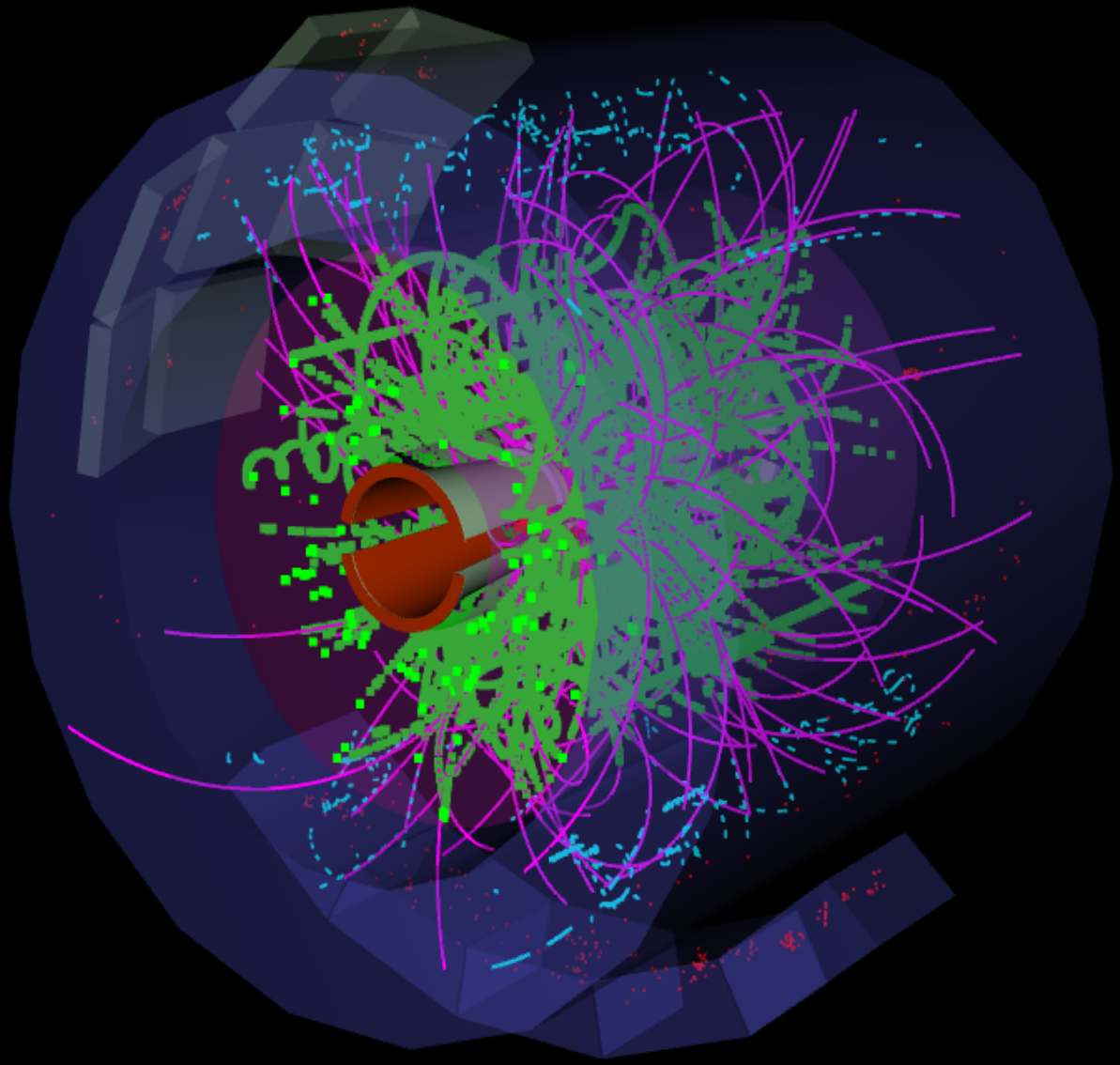


https://root.cern/js/latest/examples.htm#th2_lego1



https://root.cern/js/latest/examples.htm#tgeo_tracks+hits





New in JSROOT

- Full support of Node.js
  - reading binary ROOT files
  - creation of SVG images
  - **npm install jsroot**
- Support of more classes
  - TGraphPolar, TSpline, TPolyLine3D, …
- Special painter classes for ROOT v7
  - coexistence of v6 and v7 graphics

TWebWindow class

- Server-side instance
- Display HTML/JavaScript-based windows
- Organizes communication with multiple clients
- Provide support of batch mode

TWebWindow - server side

```
using namespace ROOT::Experimental;

// create window instance
auto window = TWebWindowsManager::Instance()->CreateWindow();

// configure html page loaded when window shown
window->SetDefaultPage("file:Main.html");

// this is call-back, invoked when message received via websocket

window->SetDataCallBack( [](unsigned connid, const std::string &arg)
 { printf("Get msg %s from %u\n", arg.c_str(), connid); } );

// configure predefined geometry
window->SetGeometry(300, 300);

// display window
window->Show();
```

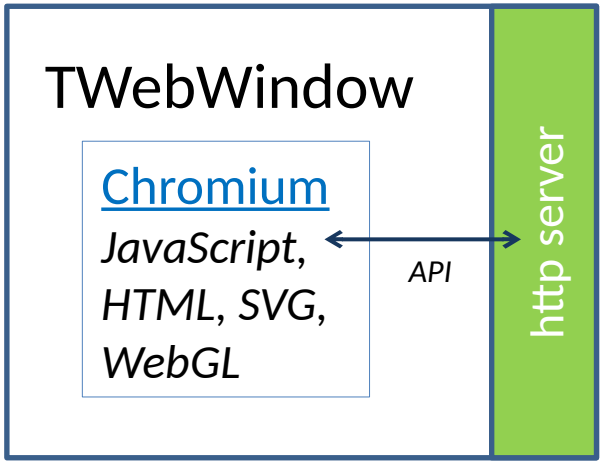
TWebWindow - client side

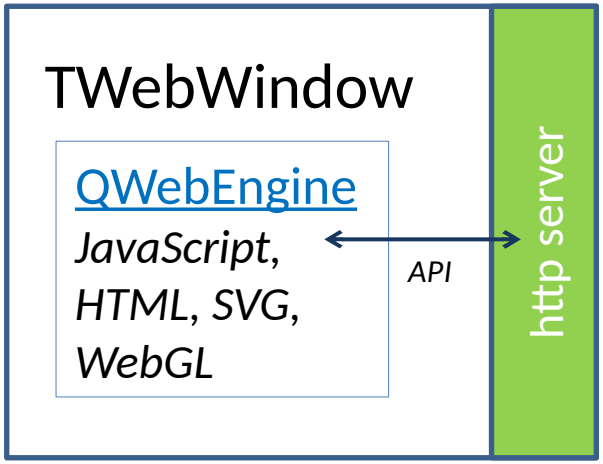
```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>TWebWindow example</title>

    <script src="/jsrootsys/scripts/JSRootCore.js"
            type="text/javascript"></script>

    <script type="text/javascript">
       function InitCustomUI(handle) {
          // assign receiver object – here dummy
          handle.SetReceiver({});
          // connect to the server
          handle.Connect();
          // start loading openui5 components
       }

       JSROOT.ConnectWebWindow({
          prereq: "openui5",
          callback: InitCustomUI
       });
    </script>
   </head>
   <body class="sapUiBody" id="content" role="application">
   </body>
</html>
```

Client – server communication

- Bi-directional
  - websockets
  - long polling
  - cefQuery (CEF, local)
  - QWebEngineUrlSchemeHandler (Qt5, local)
- Text exchange in both directions
  - binary transfer only from server to client
- Asynchronous
  - server and client sends data independent from each other
- Flow control
  - credit-based, avoid oversubscription of communication channel

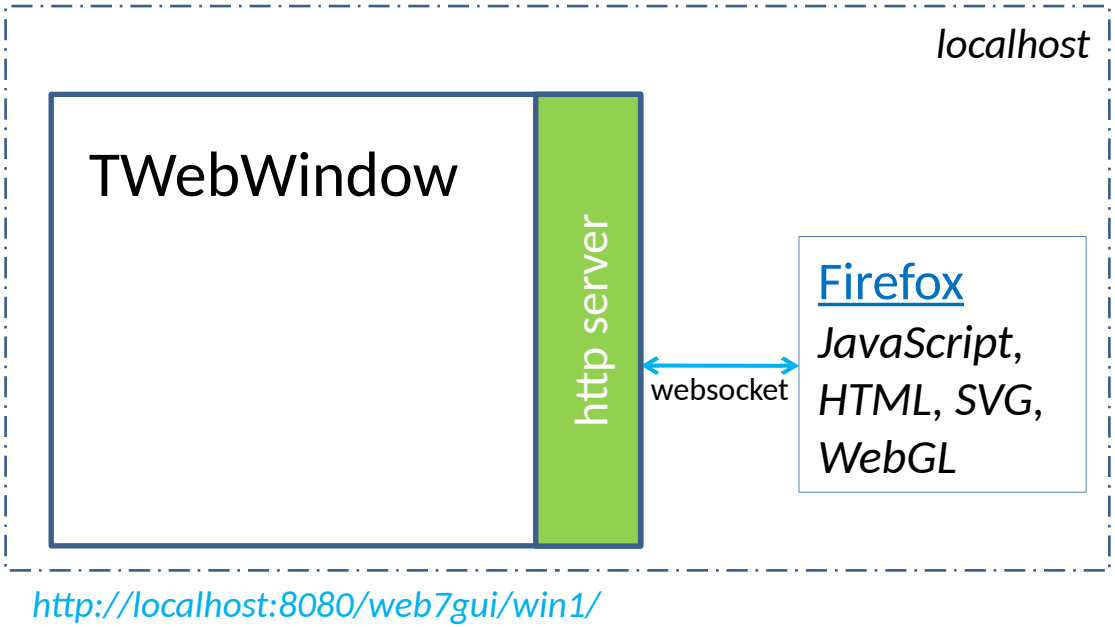
Local clients





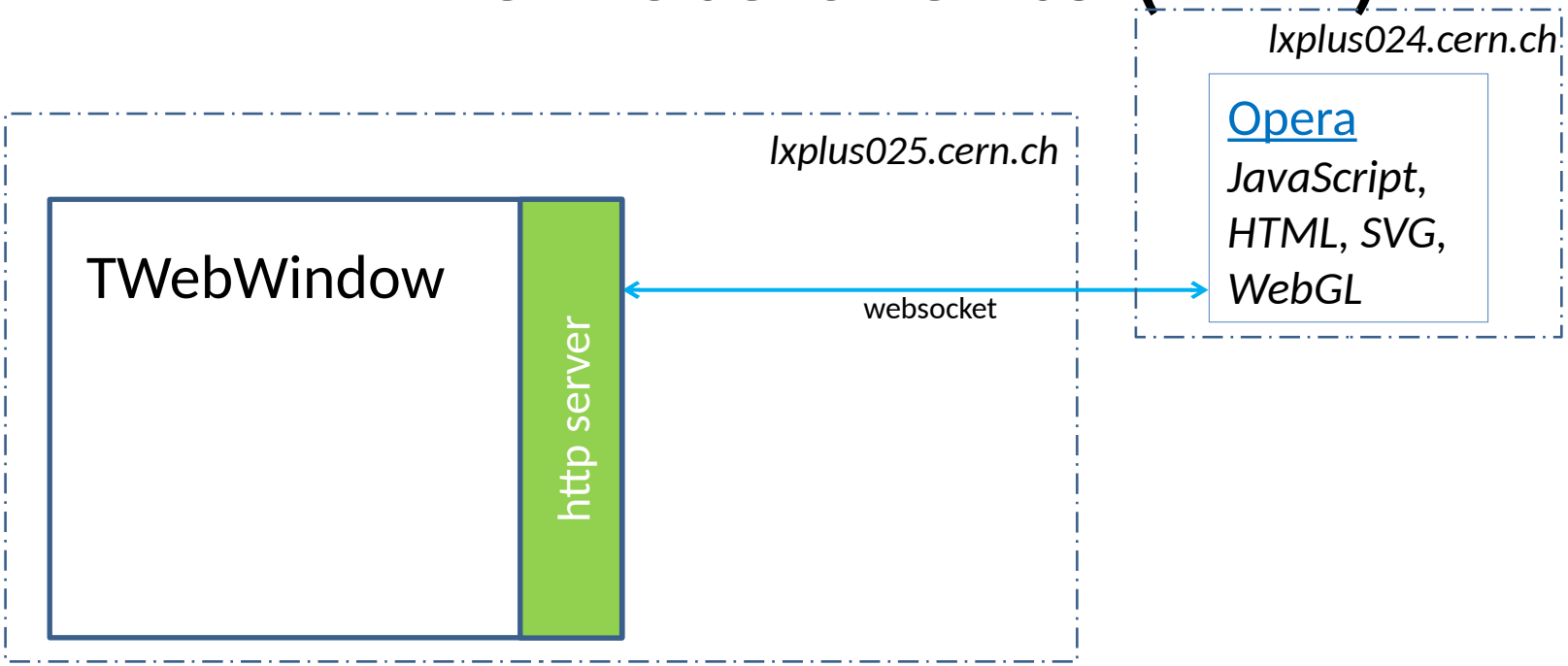
- Use Chromium Embedded Framework CEF
  see https://bitbucket.org/chromiumembedded/cef
- Create necessary window(s) directly in C++
- Communication via CEF API - no any sockets
- Qt5 with QWebEngine – also no any opened sockets

Local clients



- Run any browser on the some host
- Use THttpServer bound to loopback address
- Communication via websockets

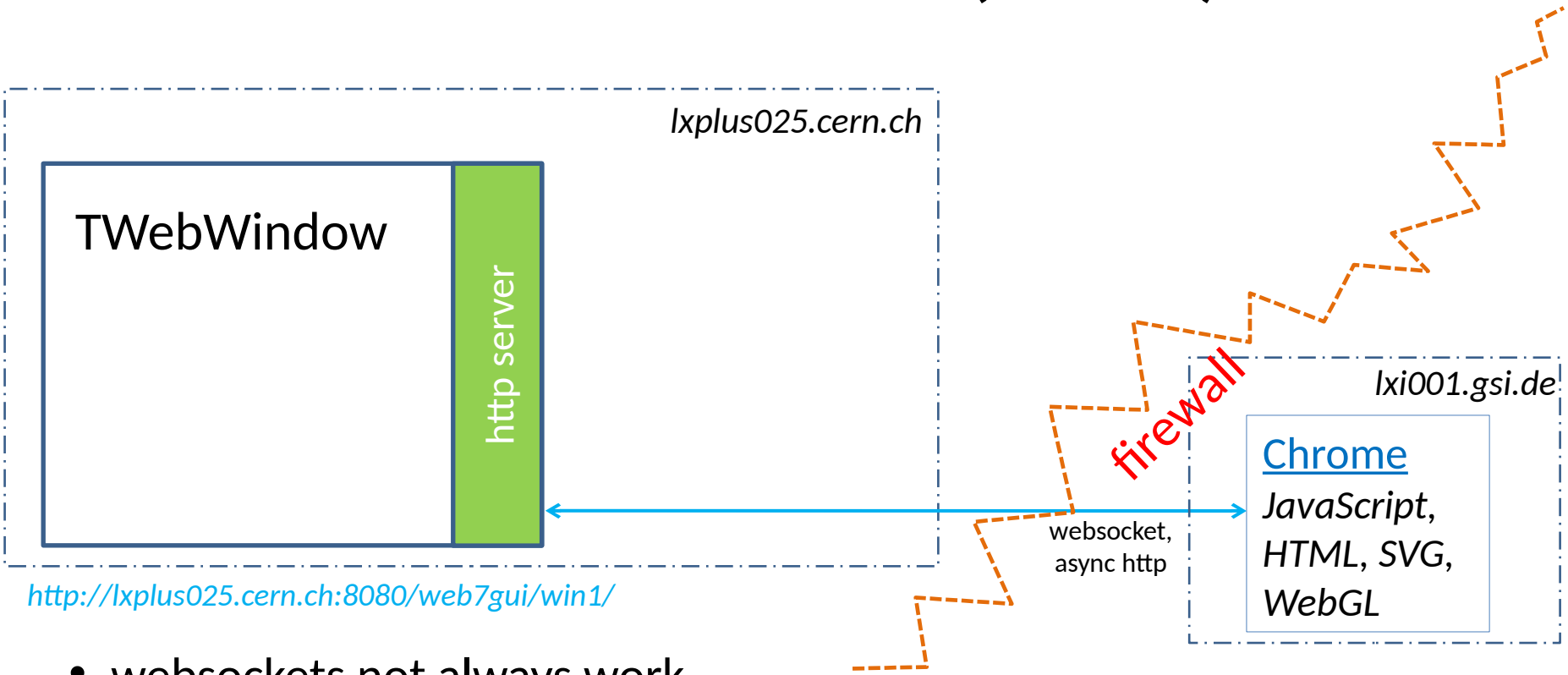
Remote clients (LAN)



http://lxplus025.cern.ch:8080/web7gui/win1/

- No any difference with local clients
- Bound THttpServer with normal IP address
- Communication via websockets

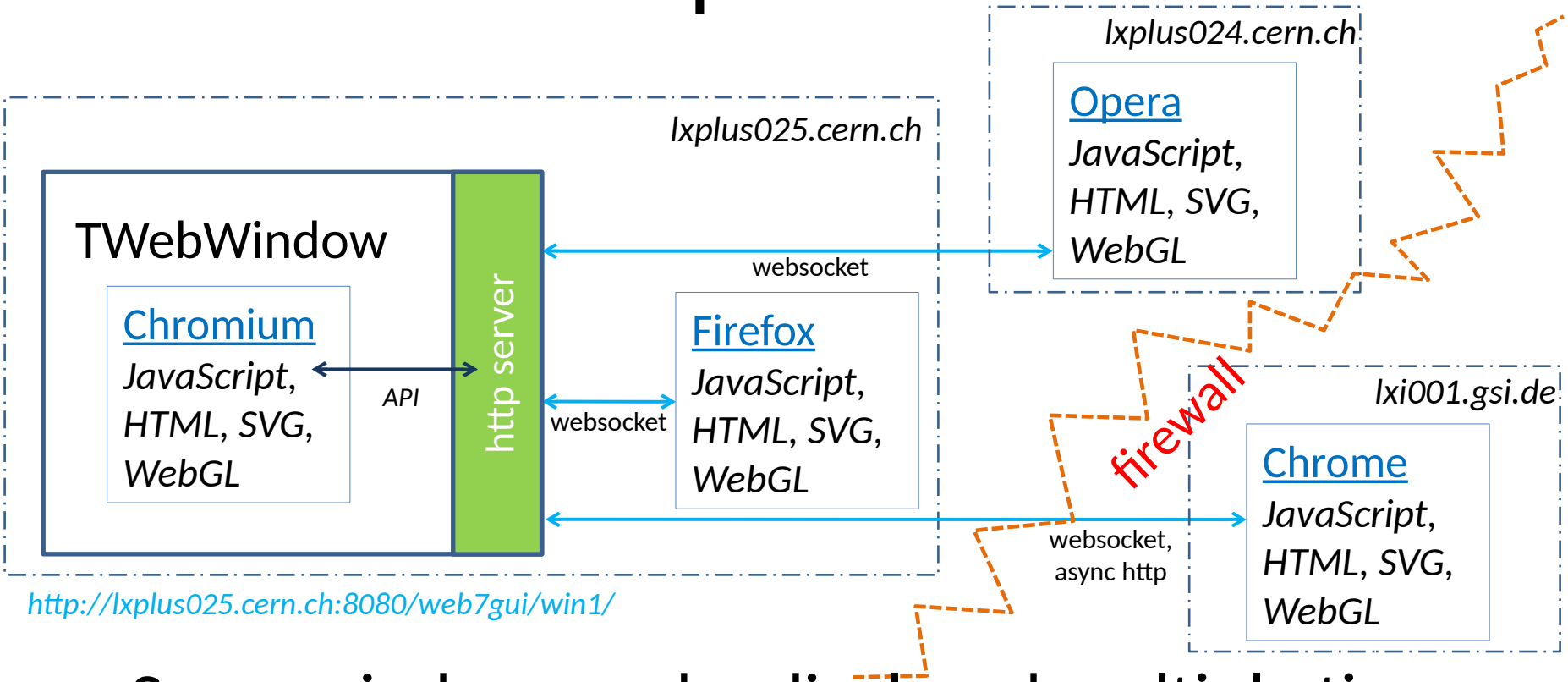
Remote clients (WAN)



http://lxplus025.cern.ch:8080/web7gui/win1/

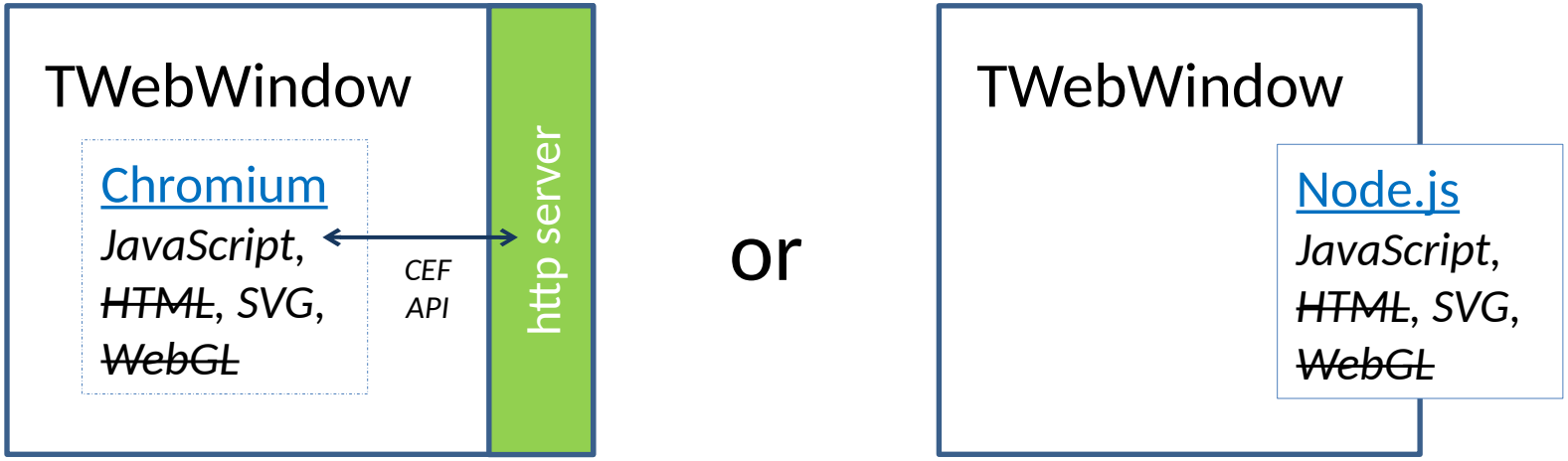
- websockets not always work
  - firewall/proxy limitation
  - not supported in fastcgi
  - automatic fall back to long polling (async http)

Multiple clients



- Same window can be displayed multiple times
- Separate websocket for each client

Batch mode



- For image production
  - SVG, PS, PNG, GIF, …
- No any window or socket created
- Reuse same JavaScript code
- Instead of WebGL use three.js SVG renderer

ROOT7 TCanvas

- Central graphics class in ROOT7
- Decouple objects from drawing attributes
- Support multiple clients

- Try to reuse JSROOT code as much as possible
- In the prototyping phase

ROOT7 GUI

- ROOT7 needs not only graphics
  - TBrowser, TFitPanel, TGEditor, …
- Required library for buttons, checkbox, list, menus, …
- After several iterations
  - SAP openui5 http://openui5.org/
- Fully supported in TWebWindow

ROOT7 WebEve* for CMS

- CMS event display is a based on ROOT. The software will continue to depend on ROOT and therefore ROOT7 in the future
- Intent to transform existing event display where current CMS code provides EVE objects and write a web client for user interaction
- Initial exploration was done by passing over TEveElements for 3D scene and tables. The demo has limited functionality: drawing only hits, geo, and tracks.
- Closed exploration event display within CMS and move to eve7 development to define protocol to communicate changes between server and client, implement synchronized selection state between multiple views, and rendering of complex physics objects like calorimeters and jets

***Alja Mrak-Tadel and Matevz Tadel*

cmsShow: data.root [1/1], event [4/10] - Mozilla Firefox

cmsShow: data.root [1/1]

xrootd.t2.ucsd.edu:5281/web7gui/win1/

Search

File

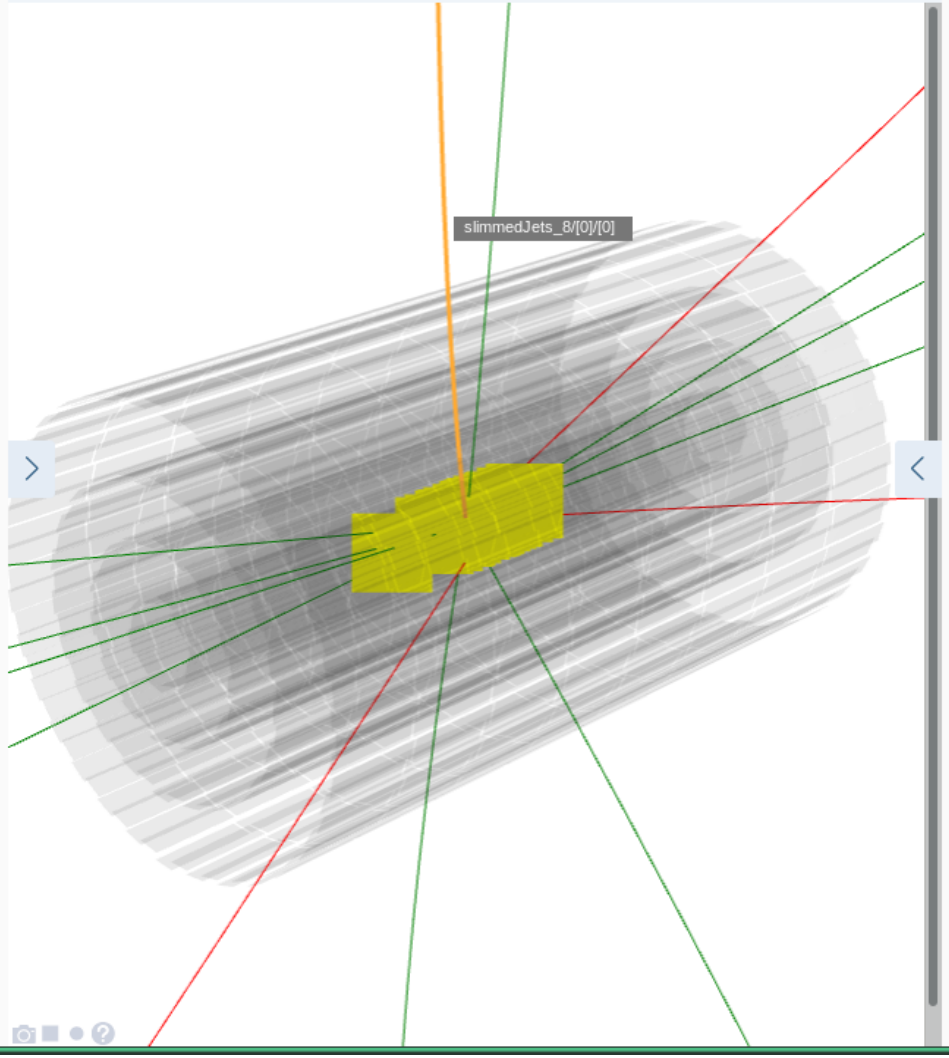
Event:

Summary

- slimmedJets [12]
- slimmedMuons [3]
- offlineSlimmedPrimaryVertices [25]

3D View

Reload



TableView slimmedJets

DATA

pt	eta	phi	el...	m...	p
84.5	-0.227	1.904	0	0	0.3
42.2	0.058	-2.25	0	0	0.3
30.4	-1.659	-0.763	0	0	0.1
29.9	-2.984	-0.034	0	0	0.2
20.6	2.572	1.159	0	0	0.5
14	-2.903	0.411	0	0	0.0
12.9	-2.133	0.818	0	0	0
11.9	-0.676	2.327	0	0	0.3
11.2	2.412	-0.145	0	0	0.4
10.6	-3.511	1.726	0	0	0
10.6	2.798	2.611	0	0	0
10.5	-0.198	-1.54	0	0	0.1

SOURCES

To next page

cmsShow: data.root [1/1]

xrootd.t2.ucsd.edu:5281/web7gui/win1/

File

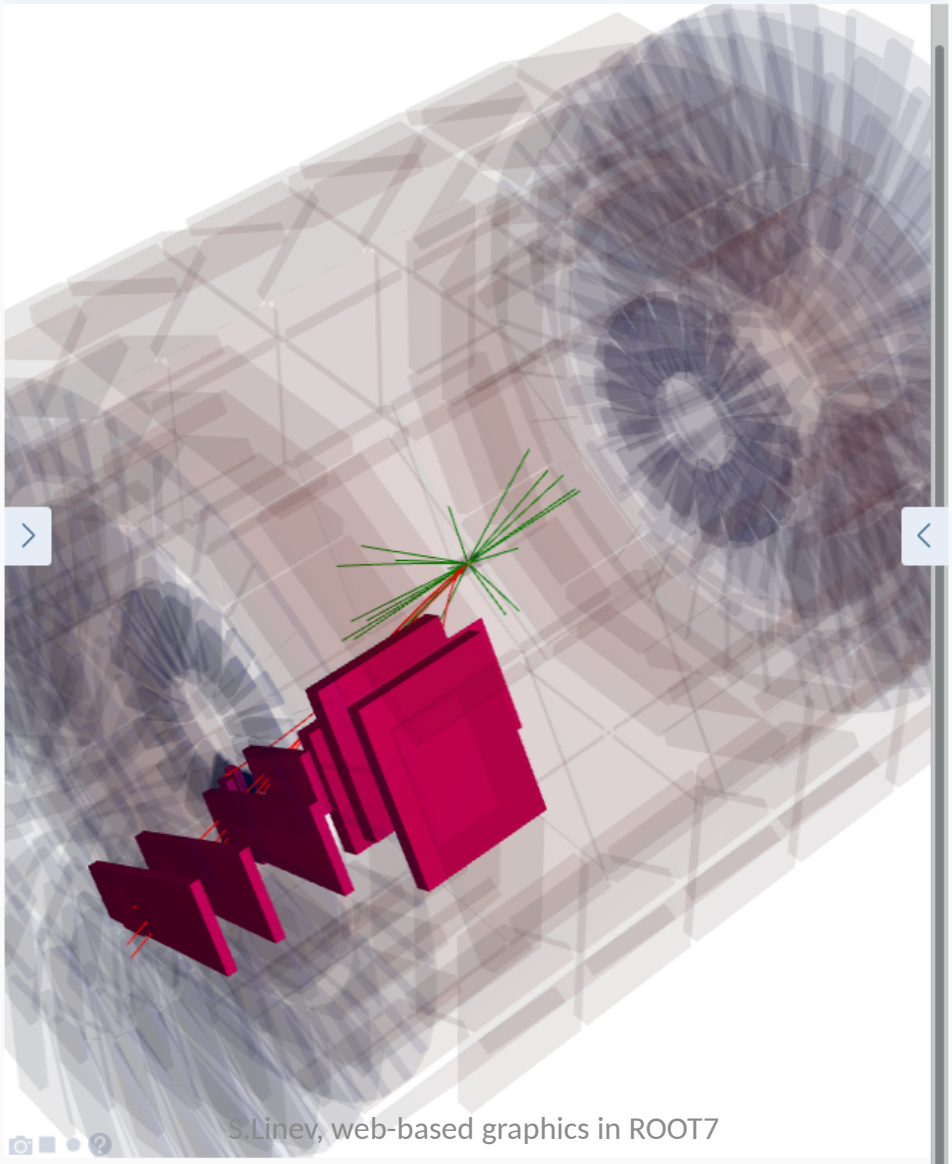
cmsShow: data.root [1/1], event [2/10]

Summary

- slimmedJets [19]
- slimmedMuons [5]
- offlineSlimmedPrimaryVertices [28]

3D View

Reload



TableView slimmedJets

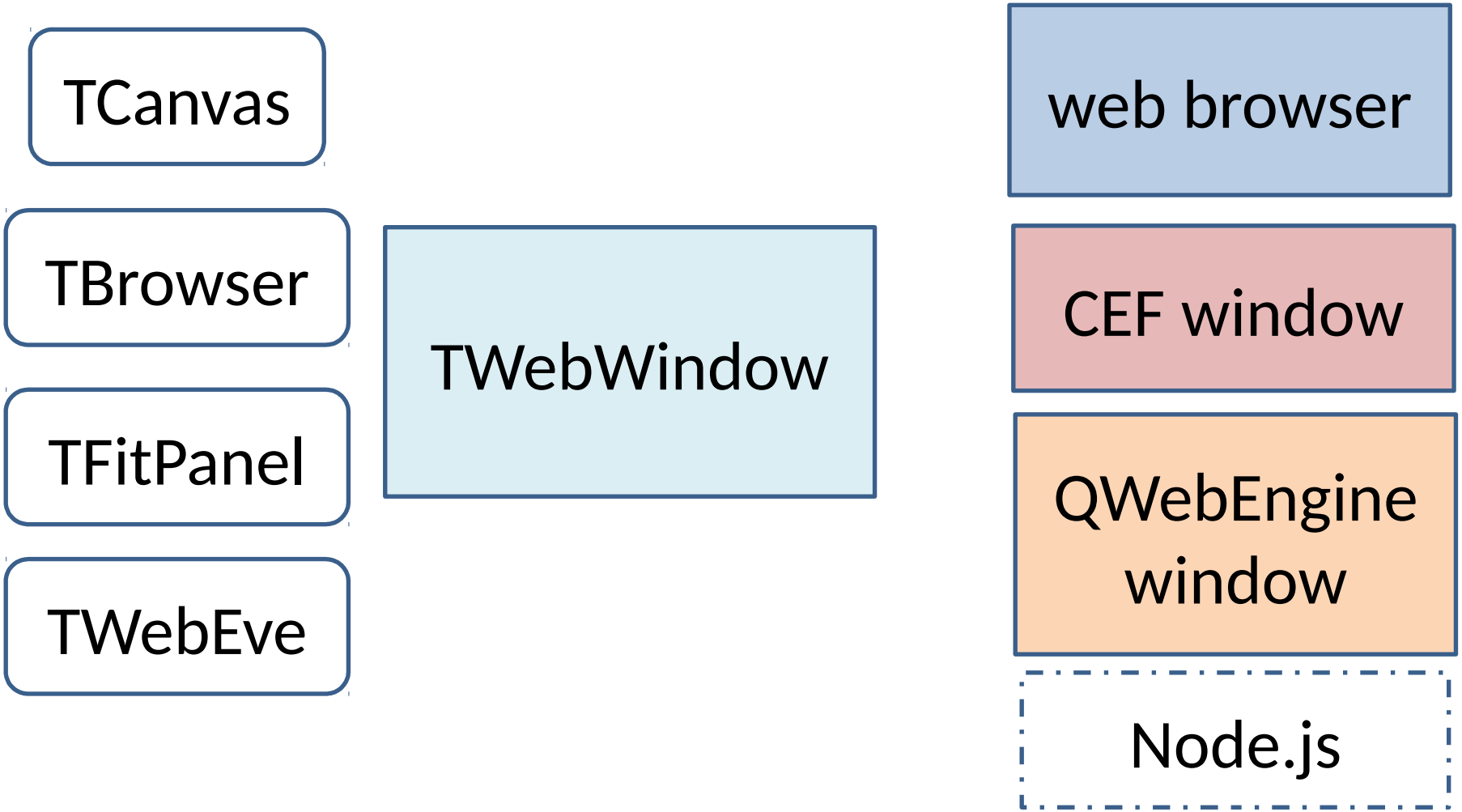
DATA

pt	eta	phi	el...	m...	p.
10.3	3.382	2.03	0	0	0
10.5	-1.374	-2.978	0	0	0.26
10.6	-2.4	-0.974	0	0	0.20
10.8	-2.008	2.185	0	0	0
11.1	1.961	0.082	0	0	0
11.8	3.601	1.358	0	0	0
12	2.507	0.583	0	0	0.17
12.6	-2.92	1.433	0	0	0
12.9	-4.05	-1.843	0	0	0
16	0.225	2.436	0	0	0.45
20.7	0.096	-2.885	0	0	0.25
34	1.211	1.128	0	0	0.57
36.9	-1.14	1.646	0	0	0.11
55	-1.21	0.326	0.505	0	0.05
62.1	-1.171	1.014	0.455	0	0.08
66.5	-0.791	2.197	0	0.123	0.34
190.6	-1.519	2.594	0	0.316	0.02
255.9	0.515	0.093	0	0	0.23
325	0.354	-1.974	0	0	0.16

SOURCES

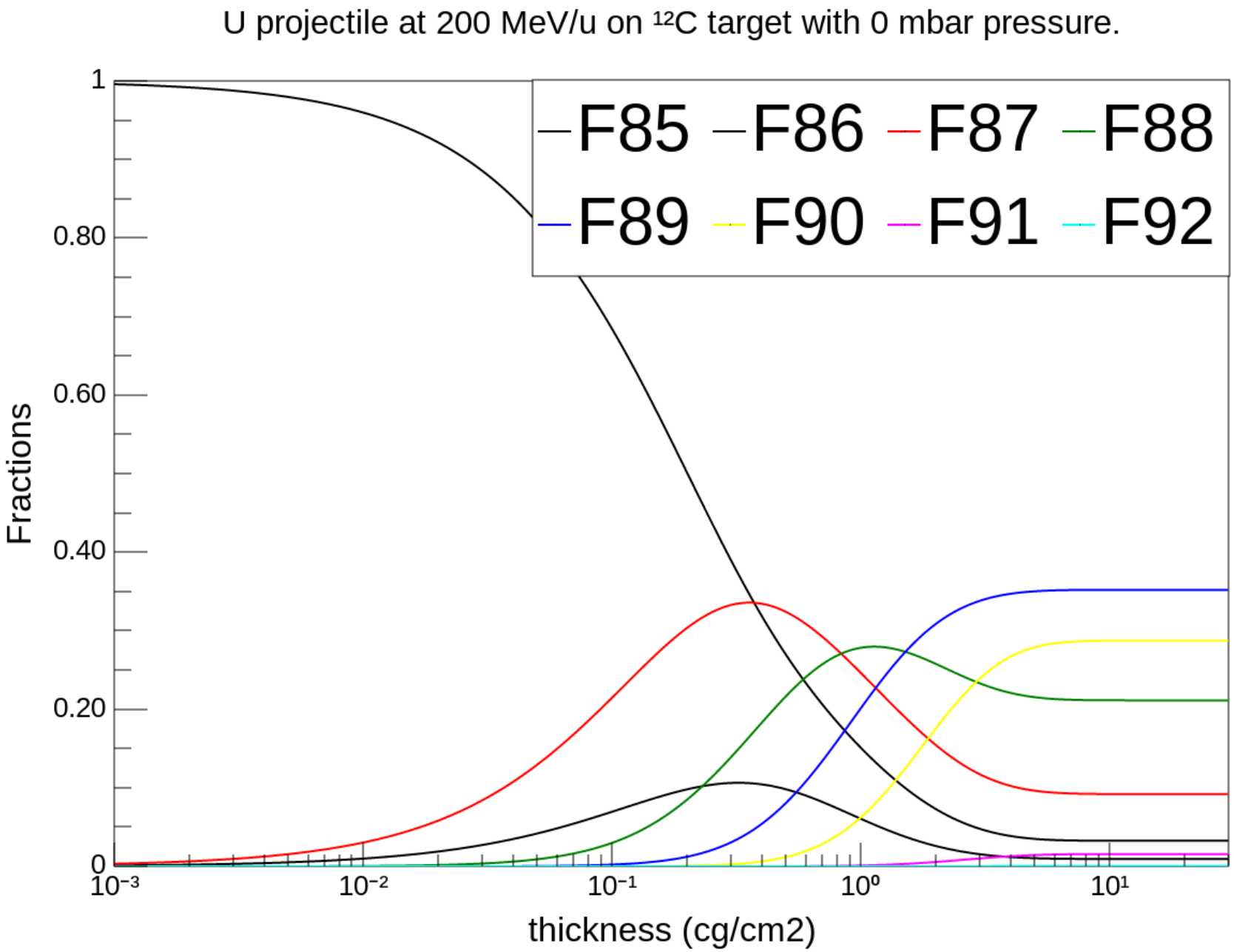
To next page

ROOT7 – coming soon

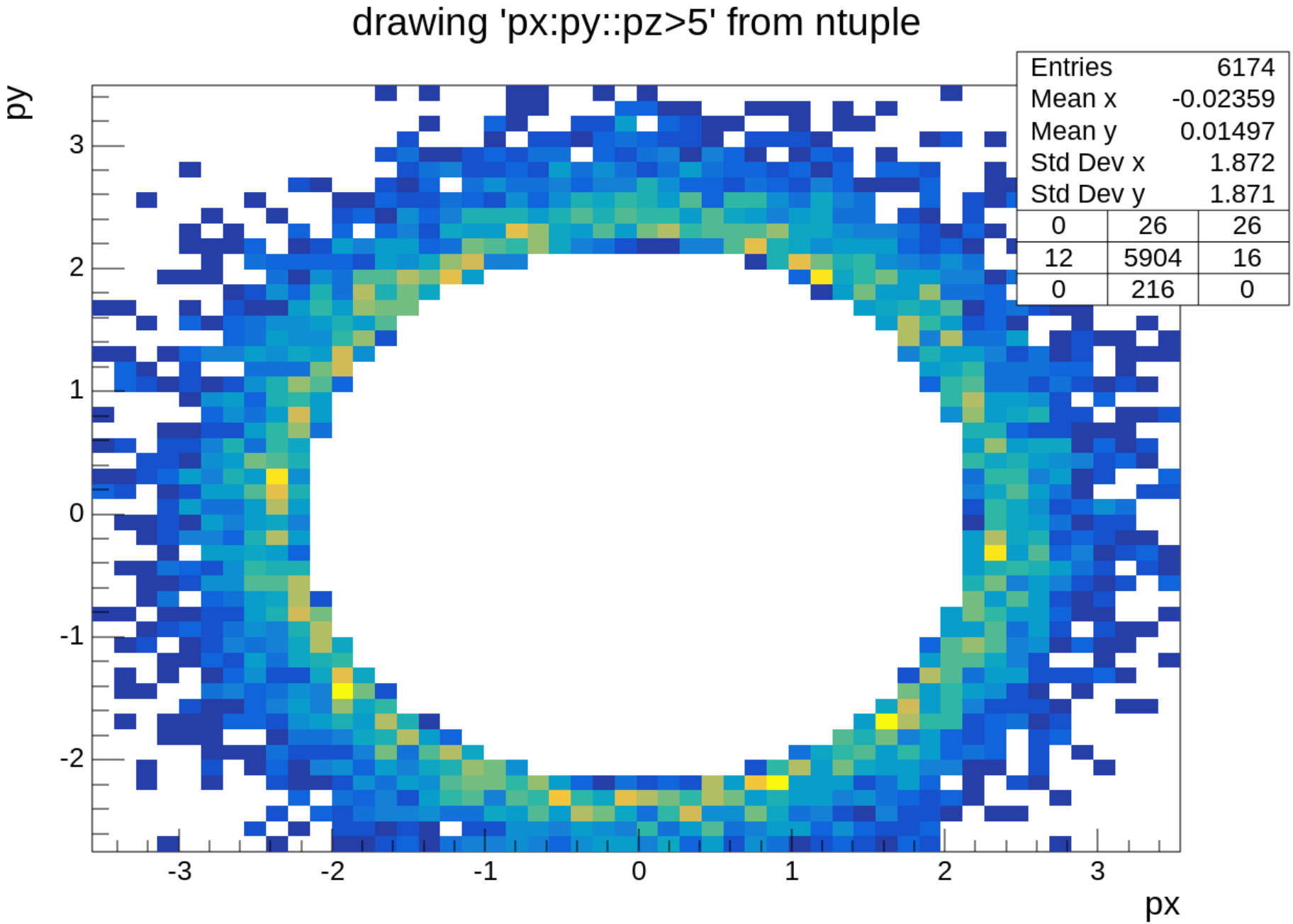


Backup slides

https://root.cern/js/latest/examples.htm#tf1_ff2



https://root.cern/js/latest/examples.htm#ttree_cut



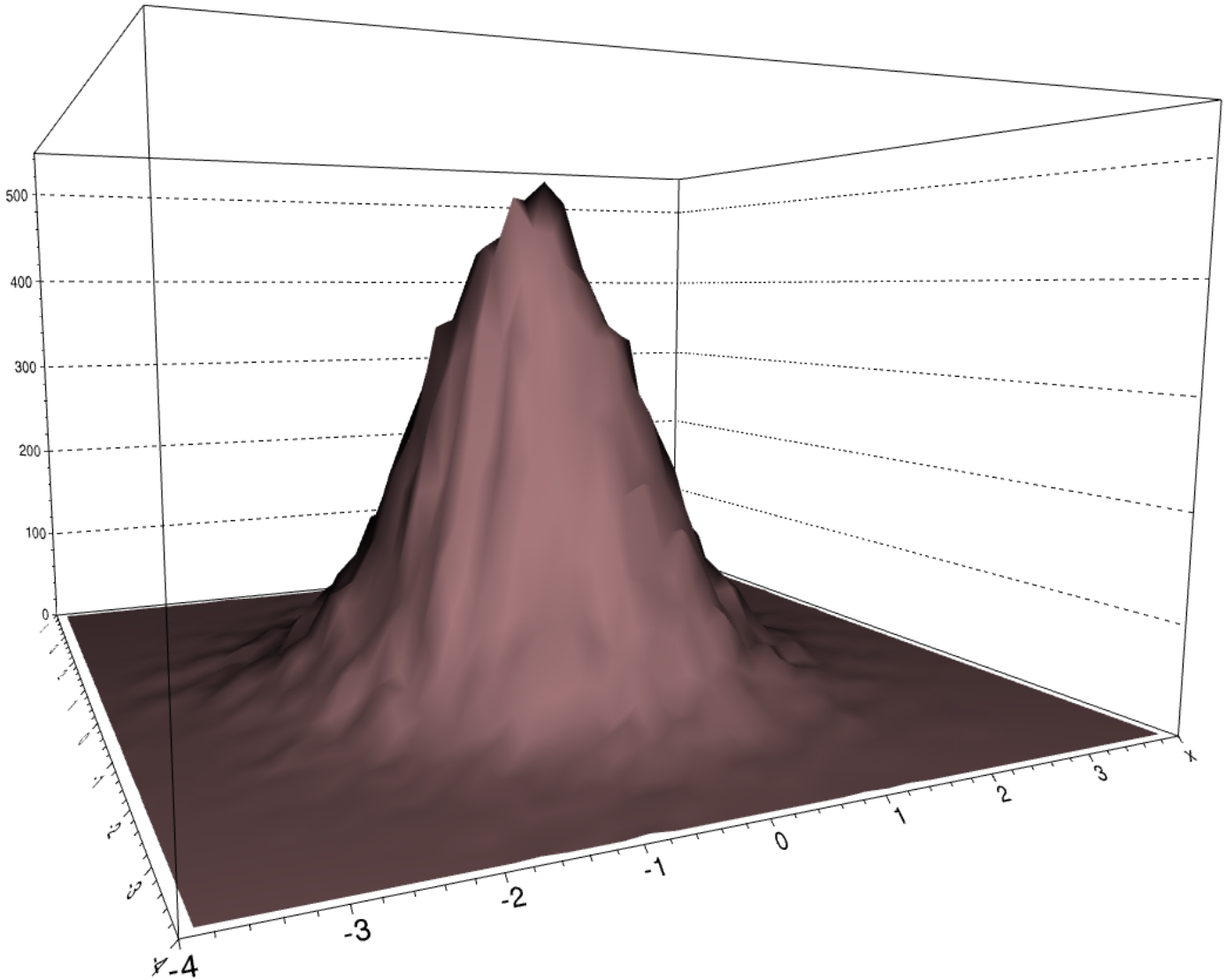
JSROOT - Simple API

- Few lines of JavaScript code to insert JSROOT graphics on any HTML page

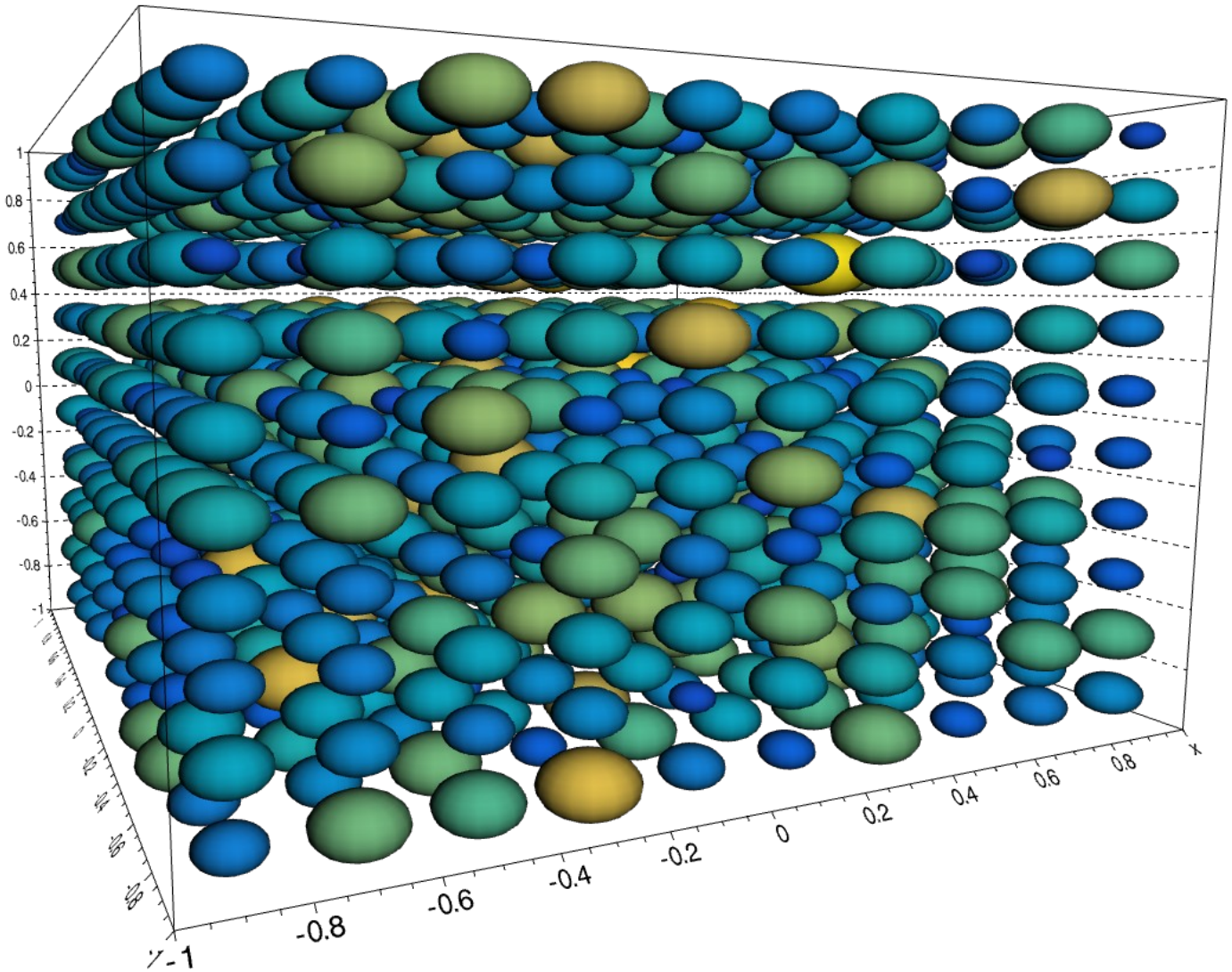
```
<script src="https://root.cern/js/latest/scripts/JSRootCore.min.js"
        type="text/javascript"></script>

<script type='text/javascript'>
    JSROOT.NewHttpRequest("hpx.json", 'object', function(obj) {
       JSROOT.draw("drawing", obj, "hist");
    }).send();
</script>


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

- Examples https://root.cern/js/latest/api.htm

https://root.cern/js/latest/examples.htm#th2_surf4



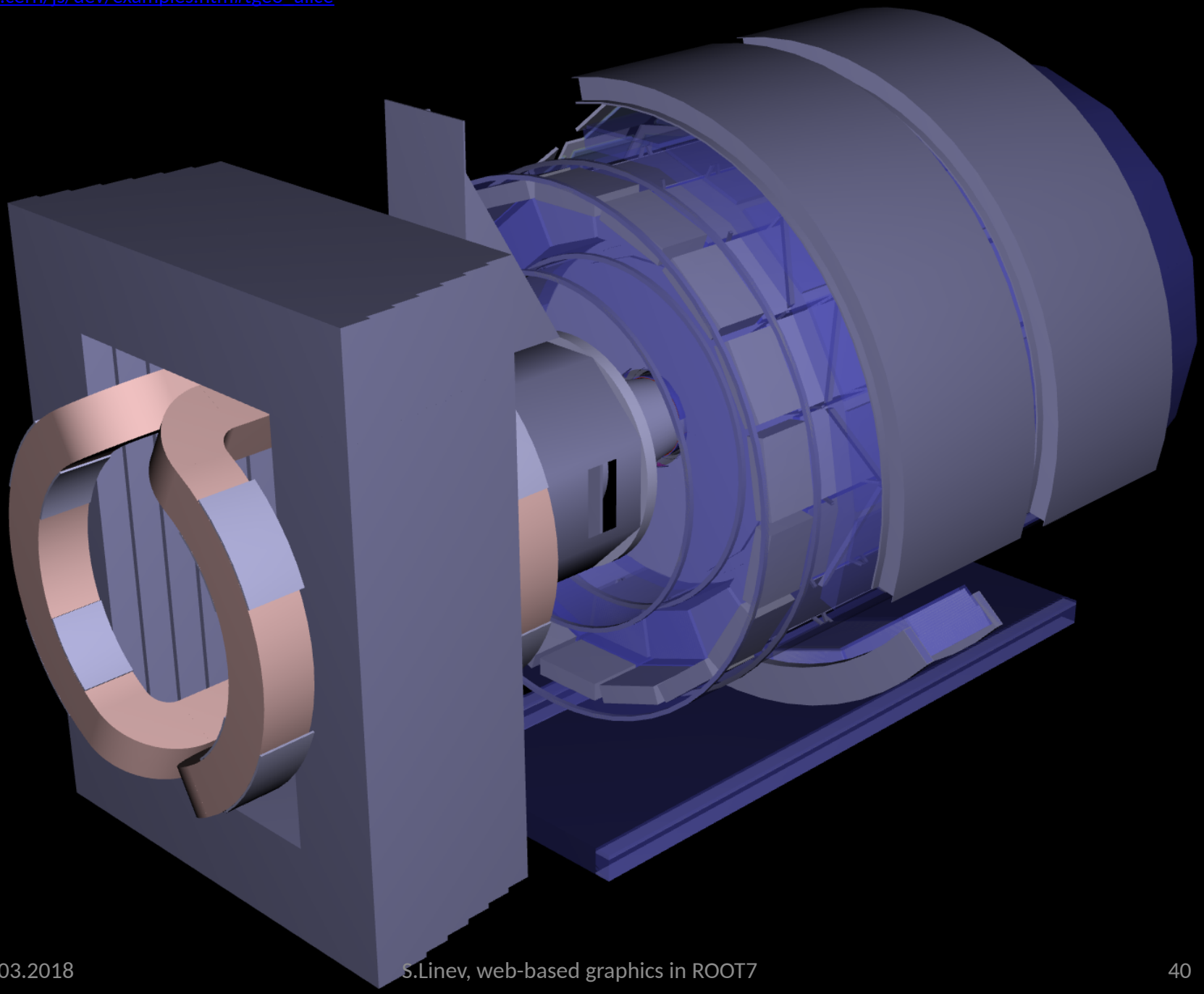
https://root.cern/js/latest/examples.htm#th3_glbox2



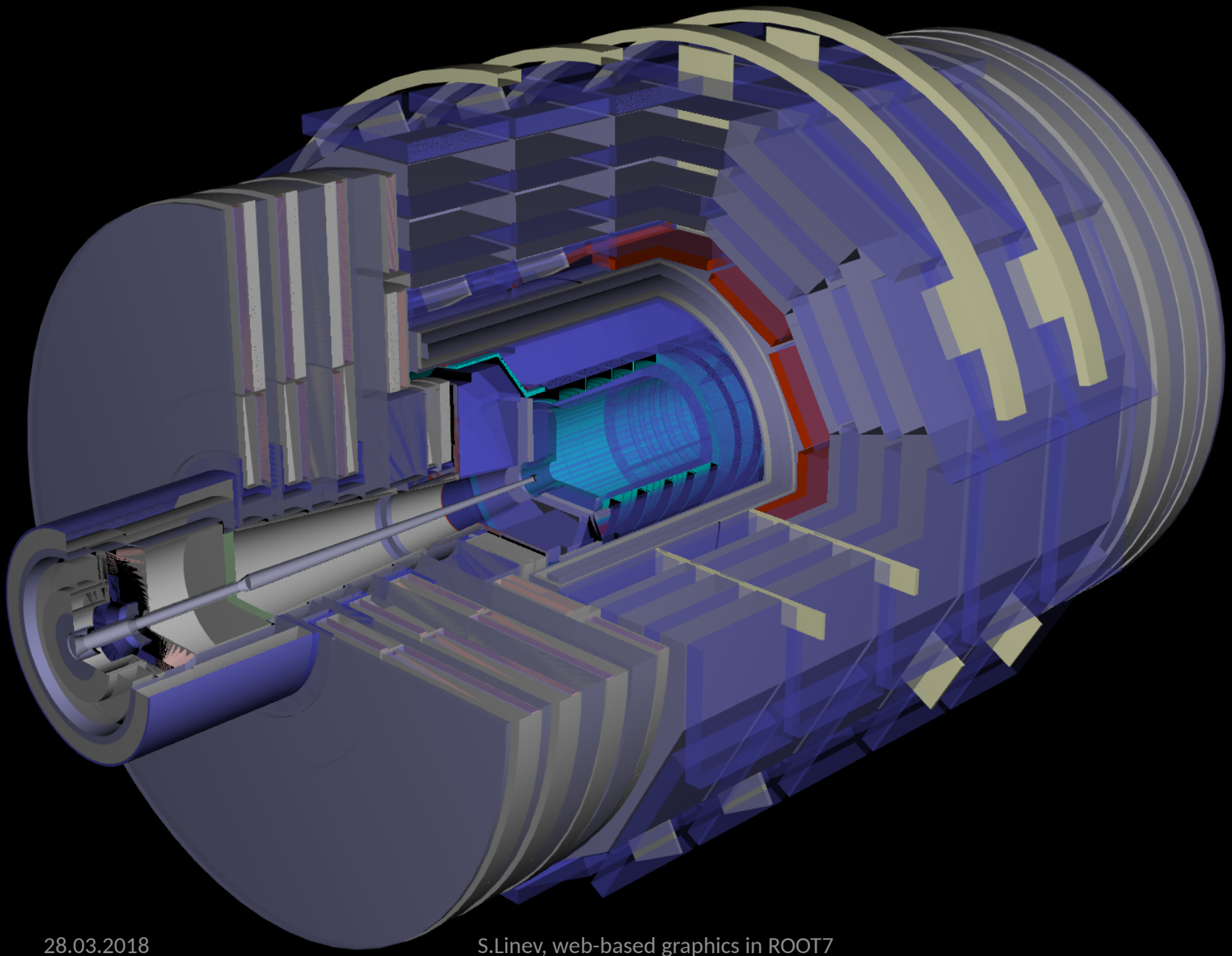
Geometry display

- three.js was there – just use it
- All kind of ROOT TGeo classes
- All kinds of shapes
  - composite with enhanced ThreeCSG.js
- THREE.BufferGeometry
- Interactive:
  - rotation and zoom
  - volumes highlight and tooltip
  - context menu
  - clip panels

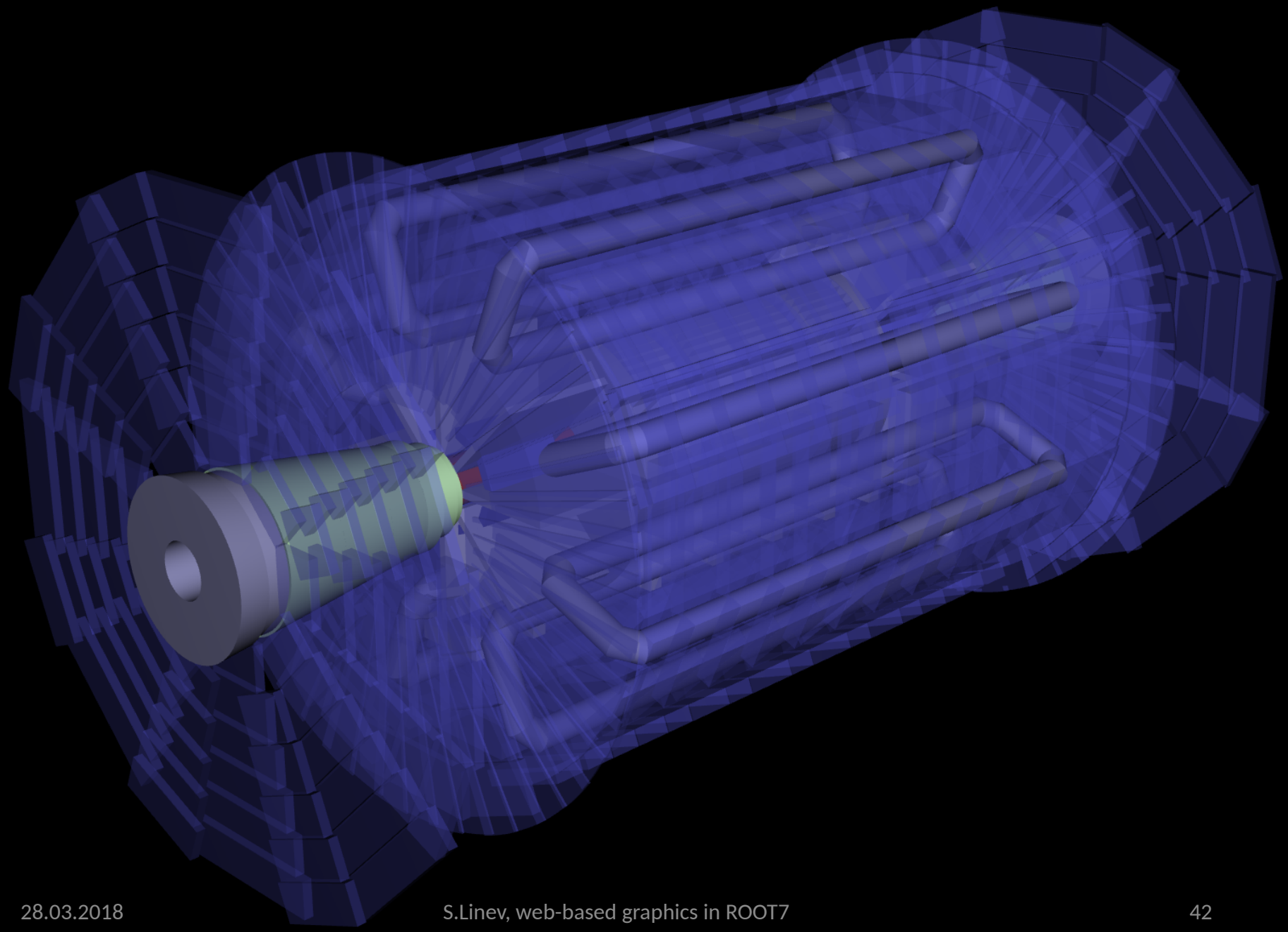
https://root.cern/js/dev/examples.htm#tgeo_alice



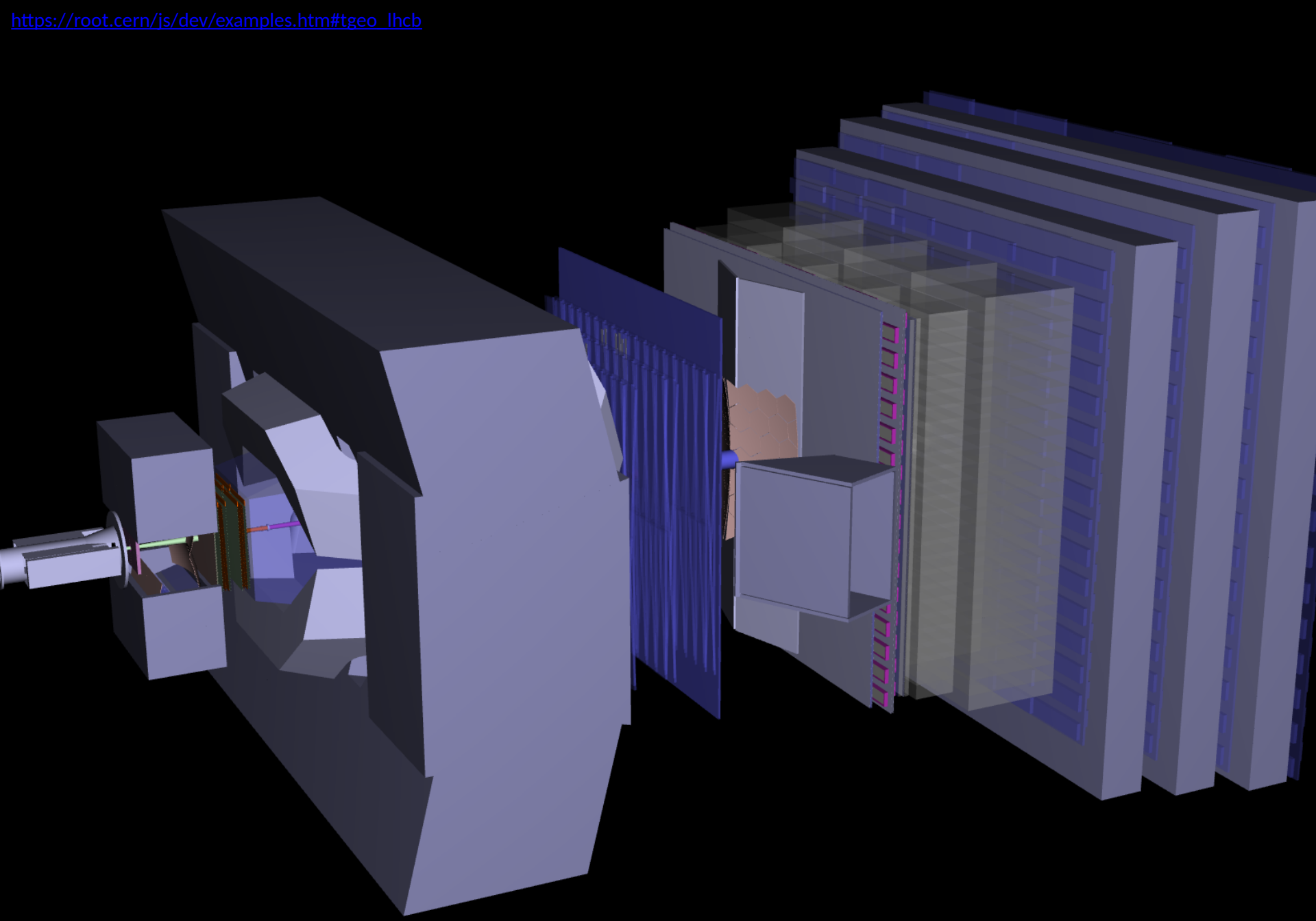
https://root.cern/js/dev/examples.htm#tgeo_cms



https://root.cern/js/dev/examples.htm#tgeo_atlas



https://root.cern/js/dev/examples.htm#tgeo_lhcb



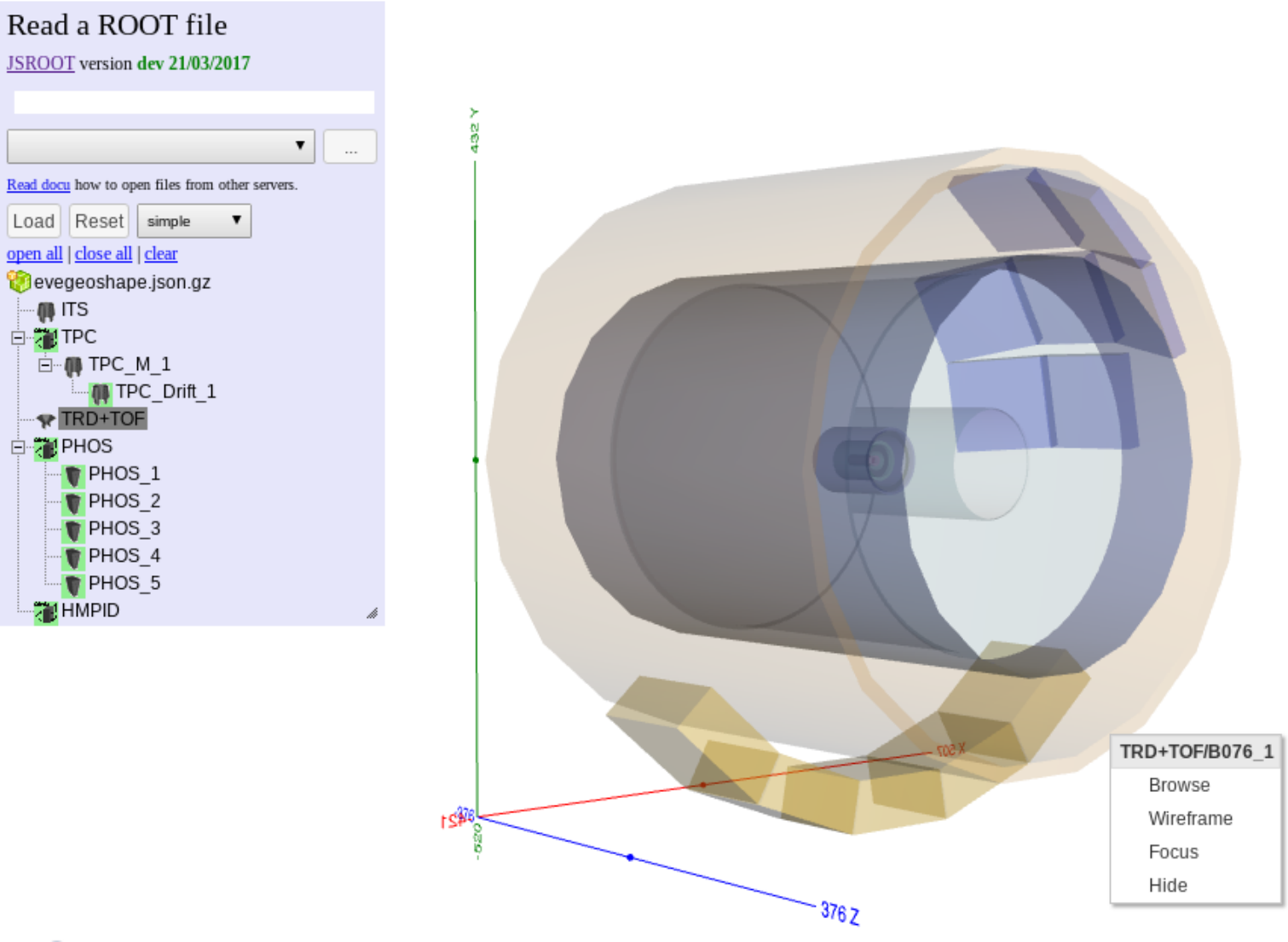
Behind the scene

- Select most significant visible volumes
  - try limit model by 10^5 faces
  - take into account view frustum (optionally)
  - create compact model descriptor
  - takes less then 0.5s for large geometries

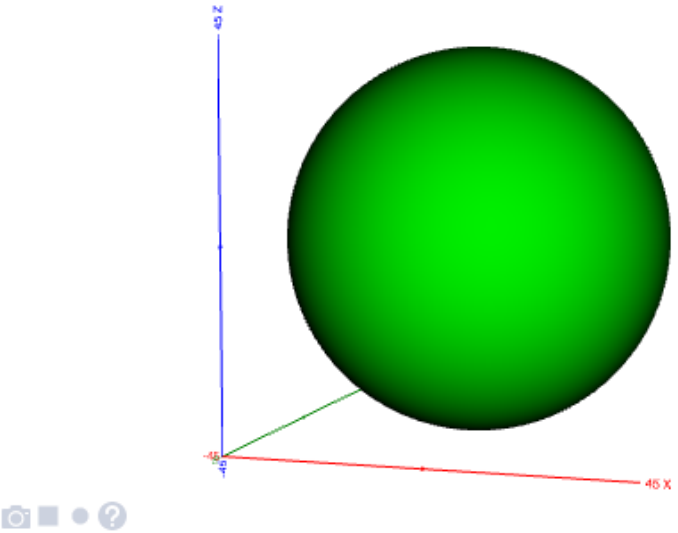
- Build three.js model
  - reproduce TGeo hierarchy in THREE.Object3D
  - create BufferGeometry for each unique shape
    - reuse build geometries when possible
  - can be offload to HTML5 Worker
  - ~3s for typical LHC geometry

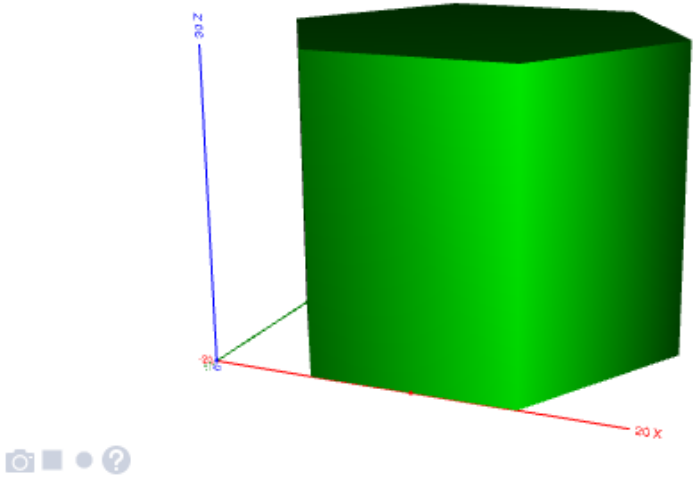
- three.js model can be used without JSROOT display
  - https://root.cern/js/dev/api.htm#custom_html_geometry

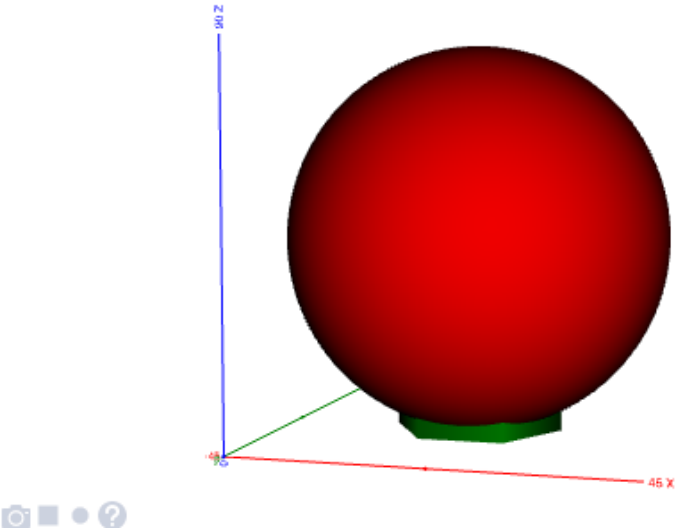
https://root.cern/js/dev/examples.htm#tgeo_eve

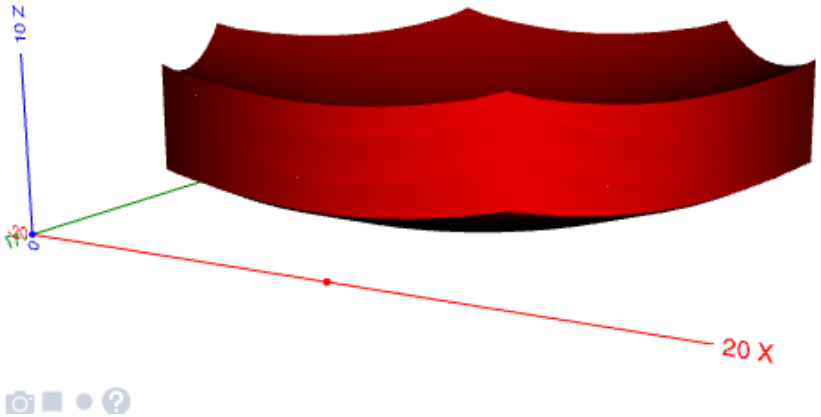


https://root.cern/js/dev/examples.htm#tgeo_comp

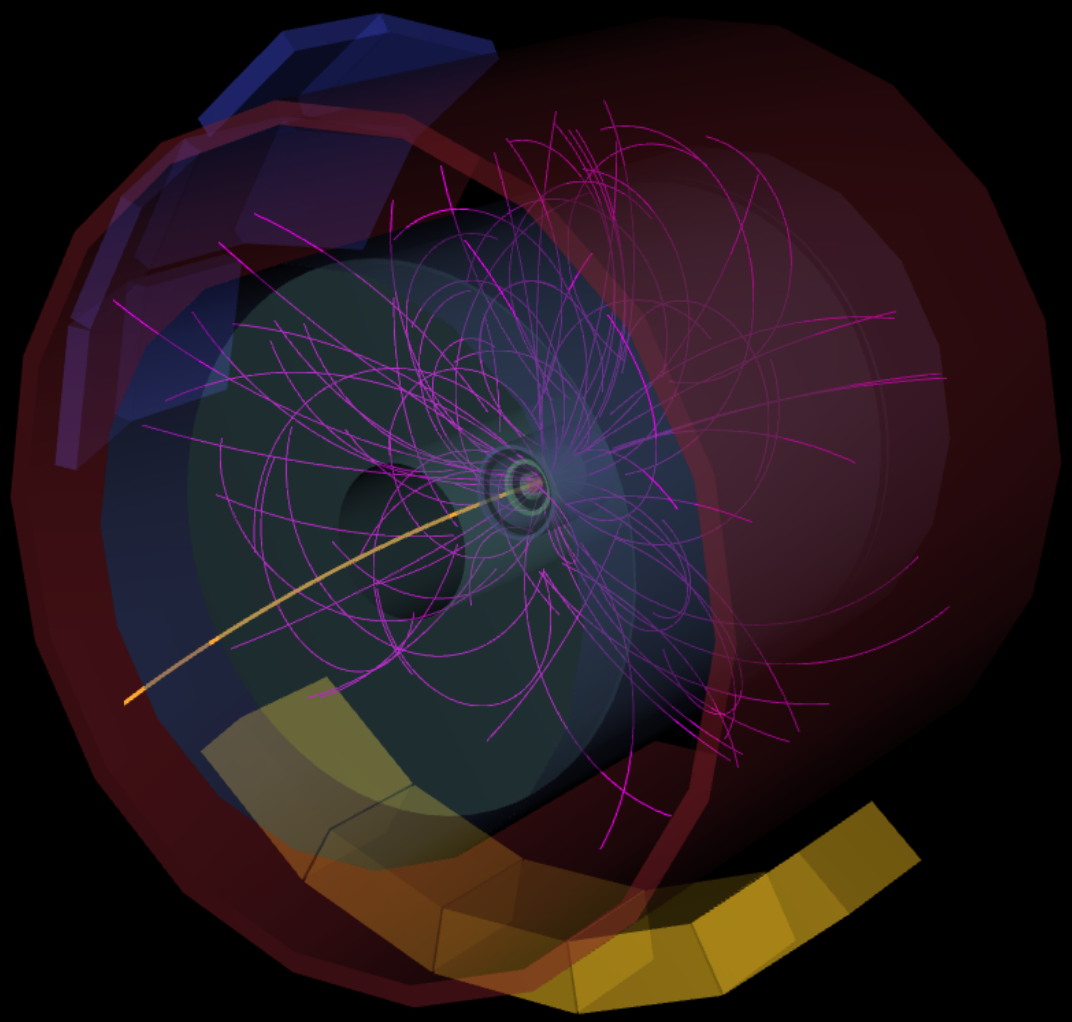


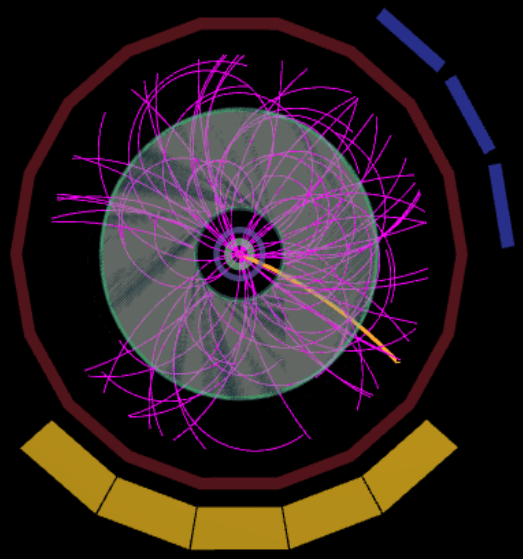


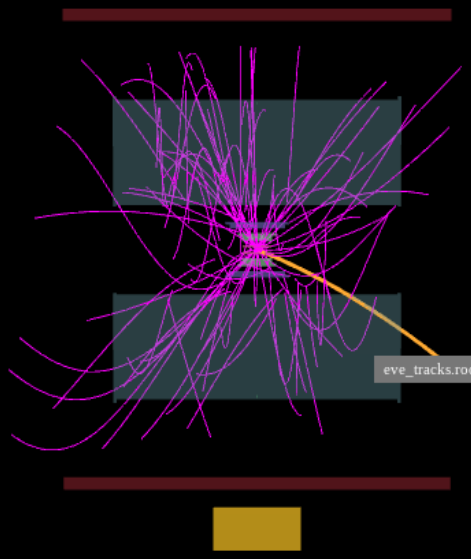




https://root.cern/js/latest/examples.htm#tgeo_proj







TTree::Draw

- Access to data directly from the browser
  - no need for MakeClass or MakeSelector
- All branches types are supported
  - including splitted STL containers
  - and old TBranchObject
- Fast – multi-range HTTP requests
- Direct dump of branch data
- Complex TTree::Draw syntax
  - expressions
  - cut condition
  - arrays indexes
  - Math functions
  - Class functions
  - histogram parameters
- TSelector-like user API