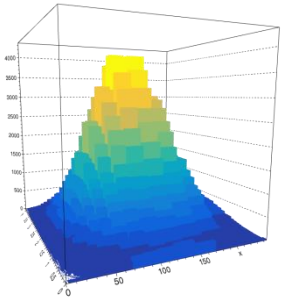
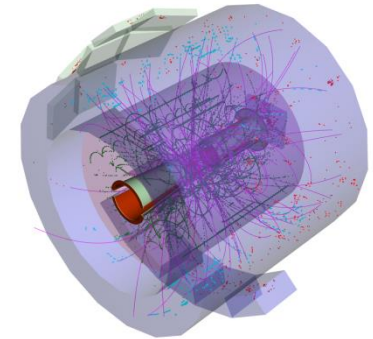
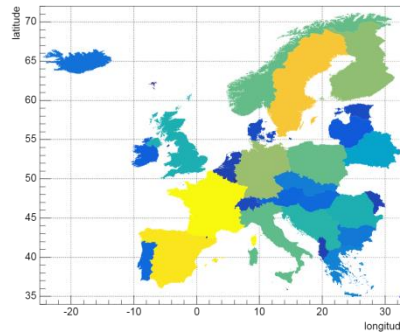
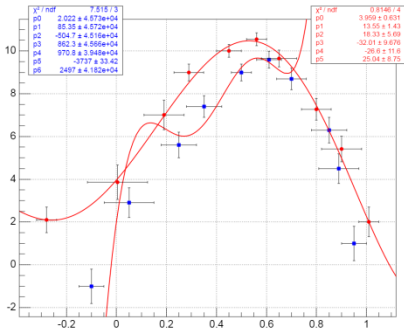
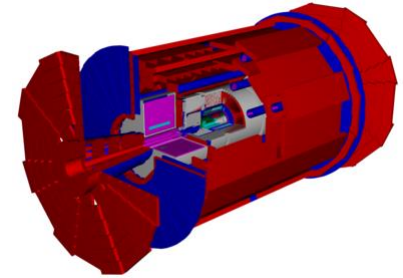


3D graphics in JavaScript ROOT



<https://root.cern/js/>

Sergey Linev (GSI, Darmstadt)



Outlook

- JavaScript ROOT project
- Histograms drawings with three.js
- Geometry display
- Tracks, hits, hierarchy
- Plans and ideas

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

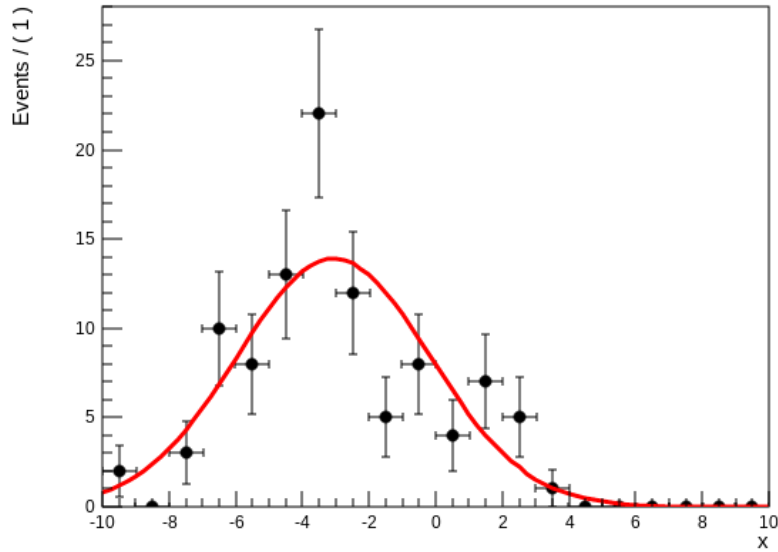
- Developed since 2012
 - <https://root.cern/js/>
 - <https://github.com/root-project/jsroot>

Supported by JSROOT

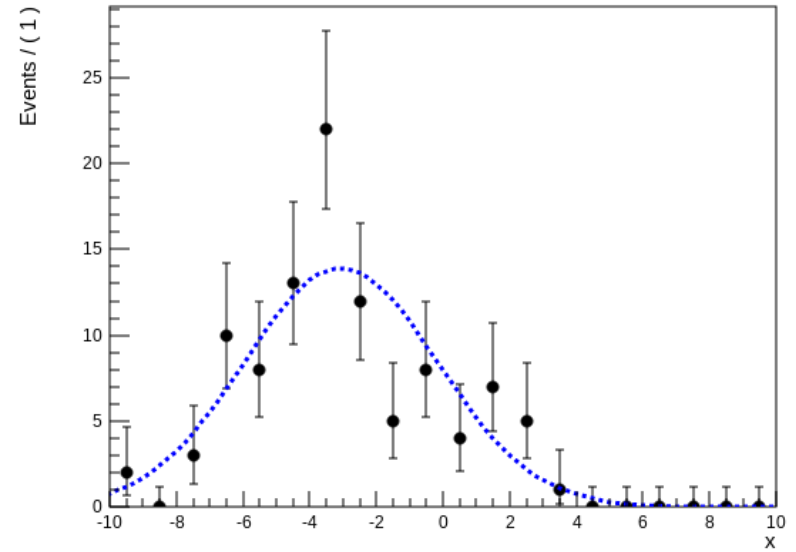
- Histograms, graphs, functions
- Canvases, stacks, multi-graphs
- Text, legend, stats
- TTree::Draw
- TGeo
- Many examples on

<https://root.cern/js/latest/examples.htm>

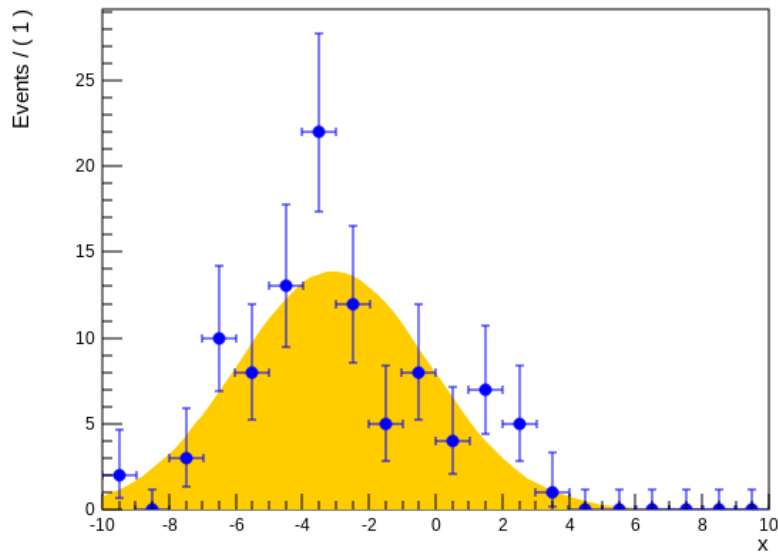
Red Curve / SumW2 Histo errors



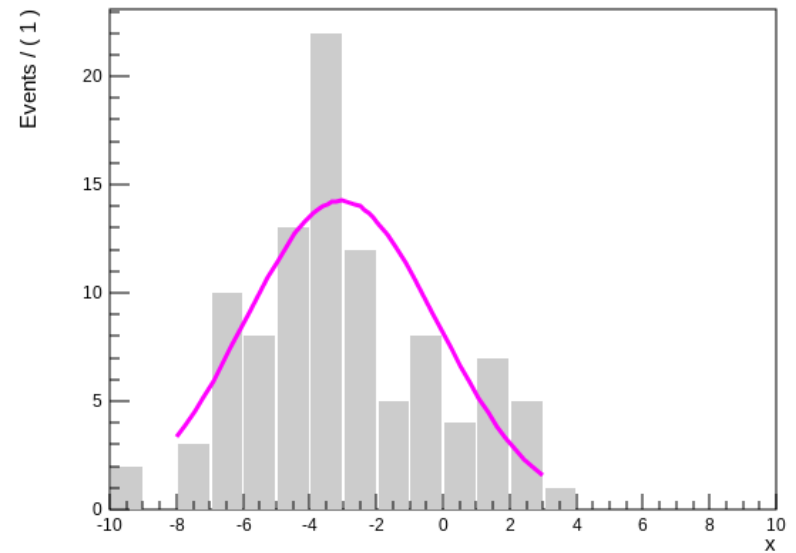
Dashed Curve / No XError bars



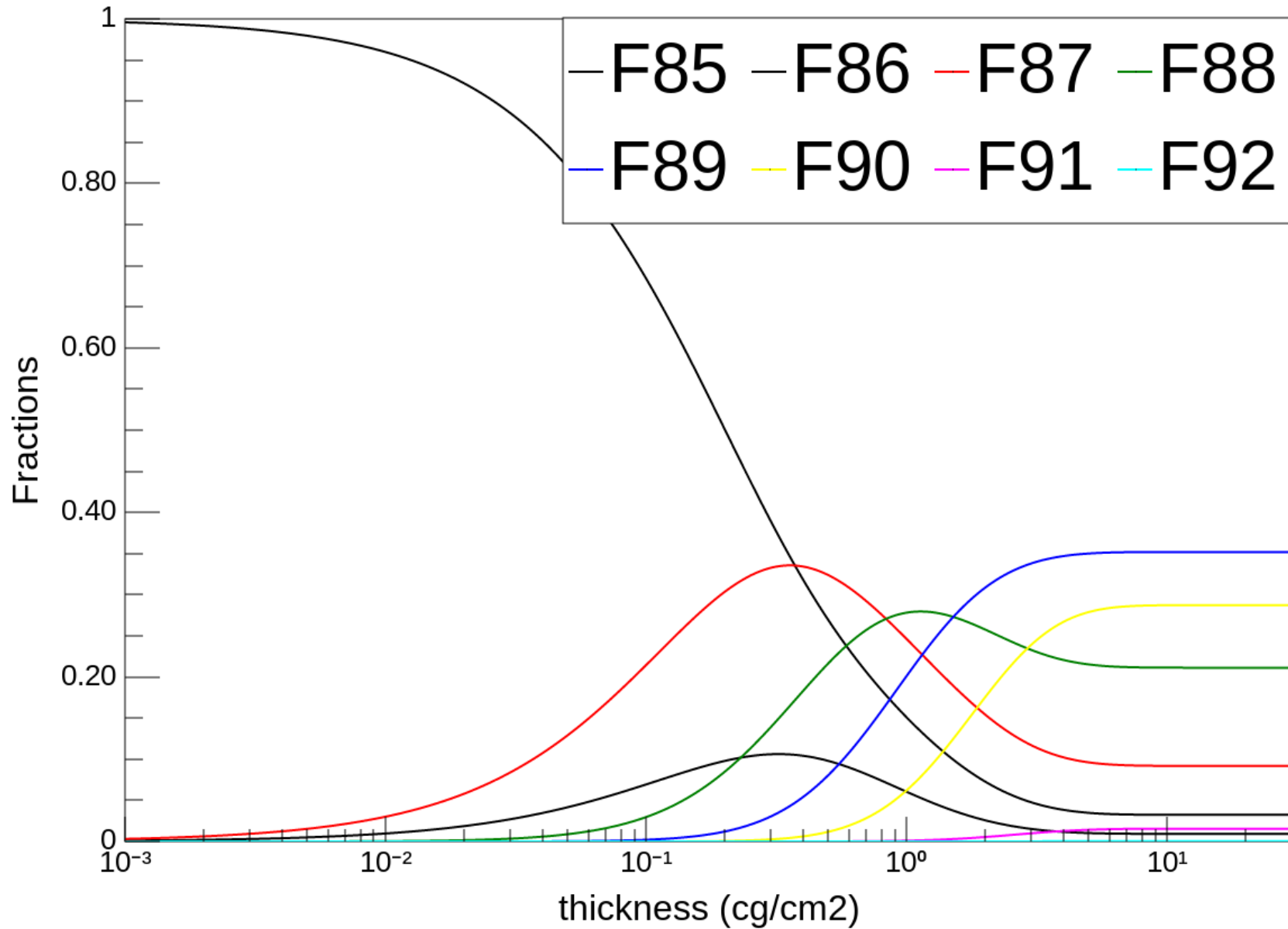
Filled Curve / Blue Histo



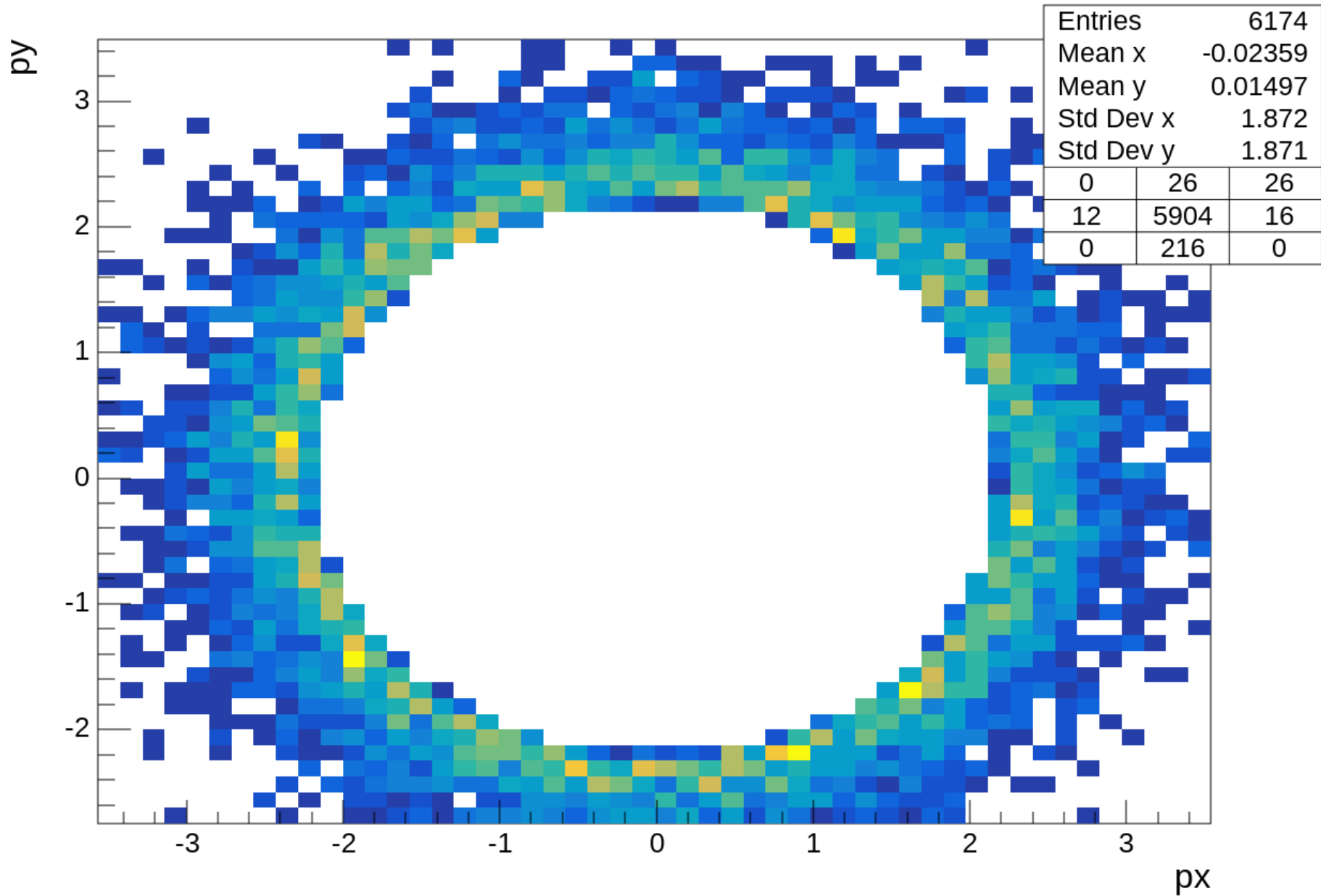
Partial Range / Filled Bar chart



U projectile at 200 MeV/u on ^{12}C target with 0 mbar pressure.



drawing 'px:py::pz>5' from ntuple



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>

THttpServer

- online 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

ROOT online server

JSROOT version 5.1.0 23/02/2017

Hierarchy in [json](#) and [xml](#) format

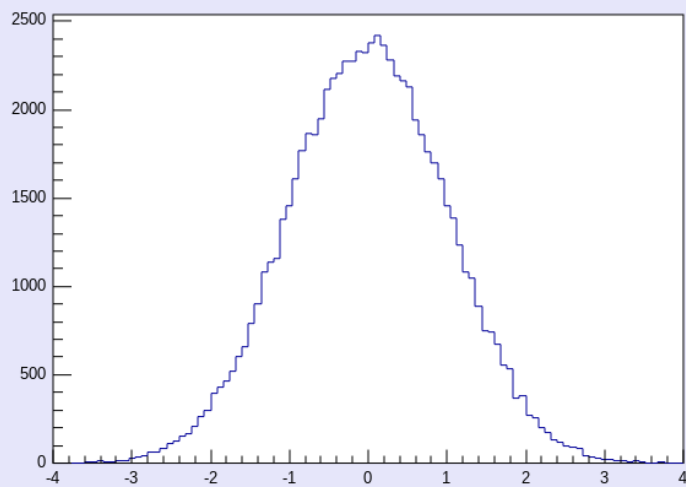
Monitoring grid 2x2 ▼

[open all](#) | [close all](#) | [clear](#)

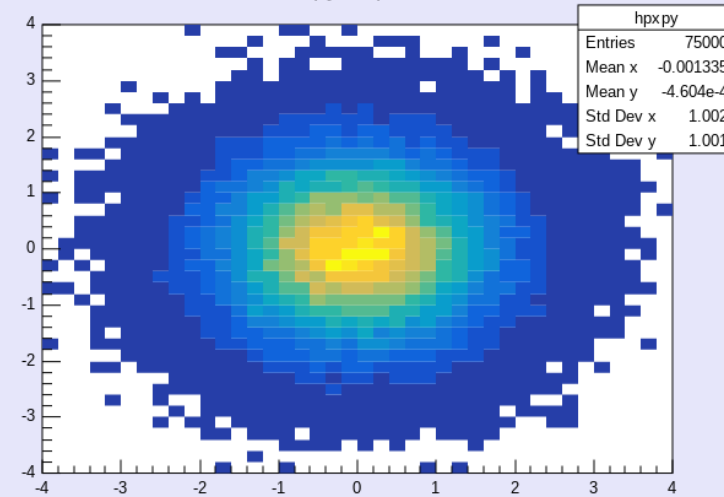
Items

- job1
 - Canvases
 - c1
 - Files
 - job1.root
 - hpx
 - hpxpy
 - hprof
 - ntuple
 - px
 - py
 - pz
 - random
 - i
- hsimple.root
 - hpx;1
 - hpxpy;1
 - hprof;1
 - ntuple;1
 - px
 - py
 - pz
 - random
 - i
 - StreamerInfo

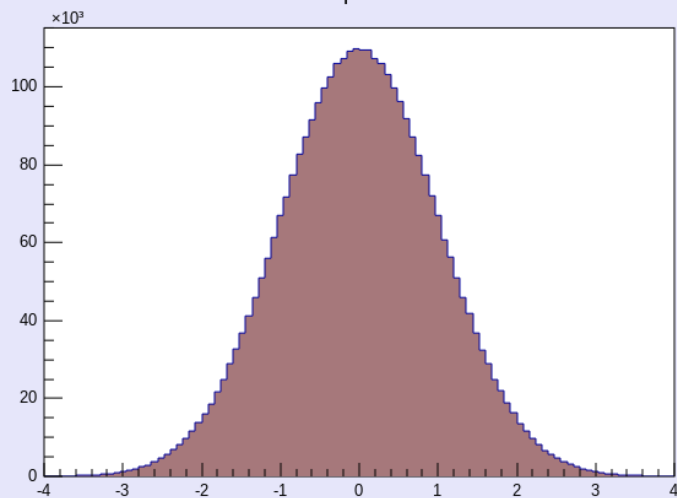
This is the px distribution



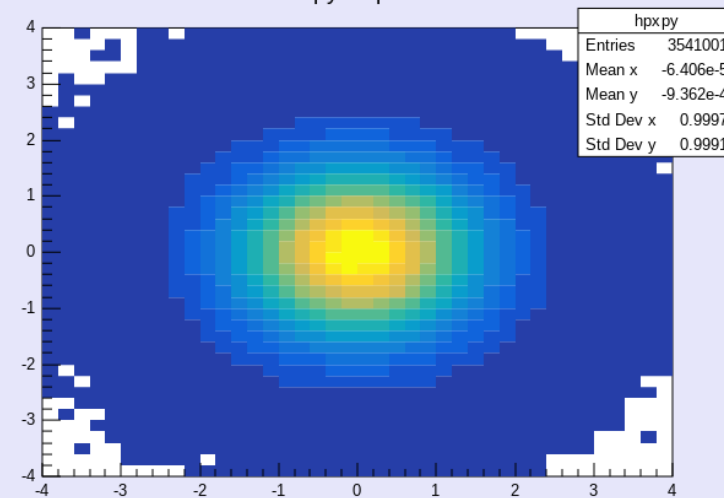
py vs px



This is the px distribution



py vs px

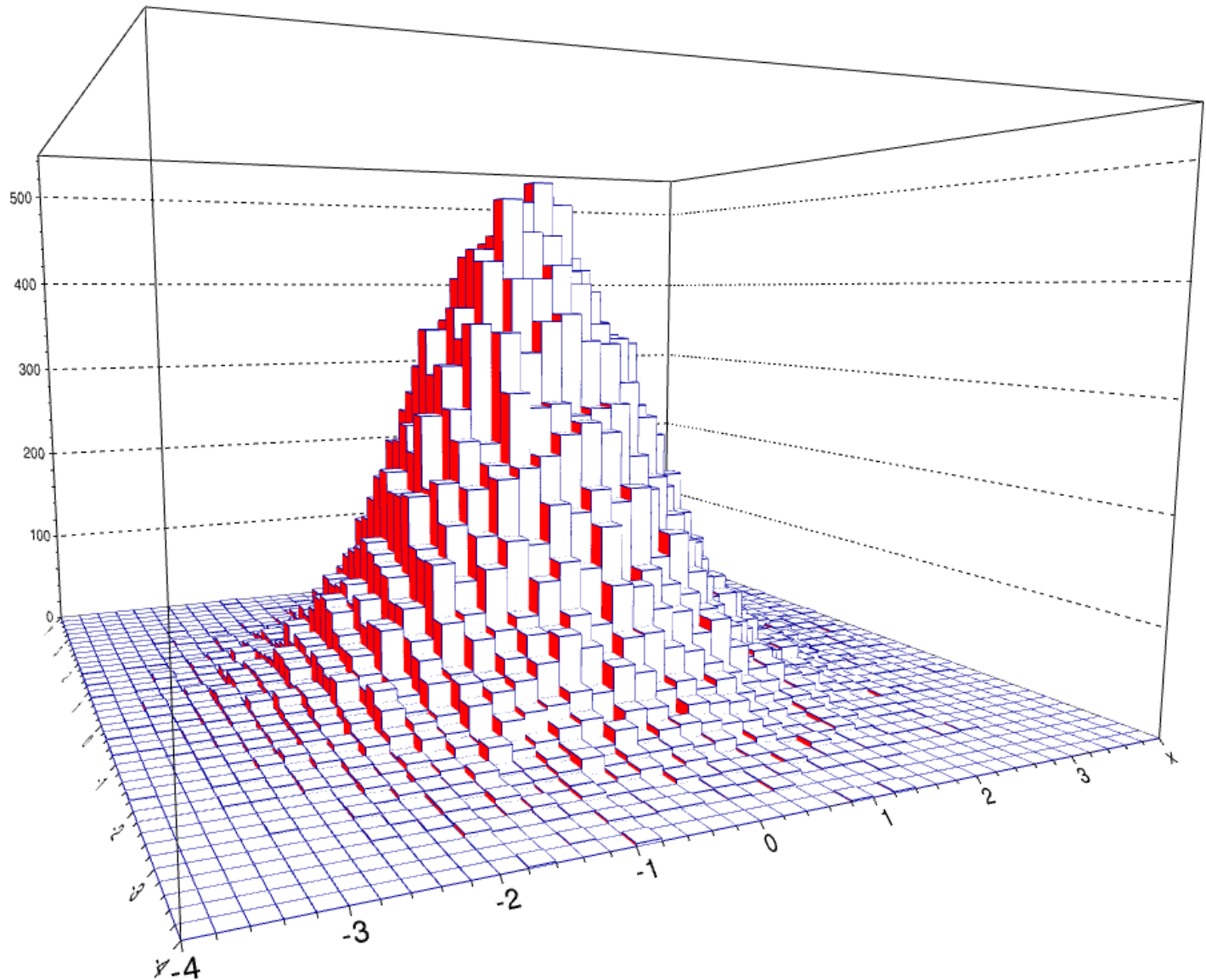


TBufferJSON

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

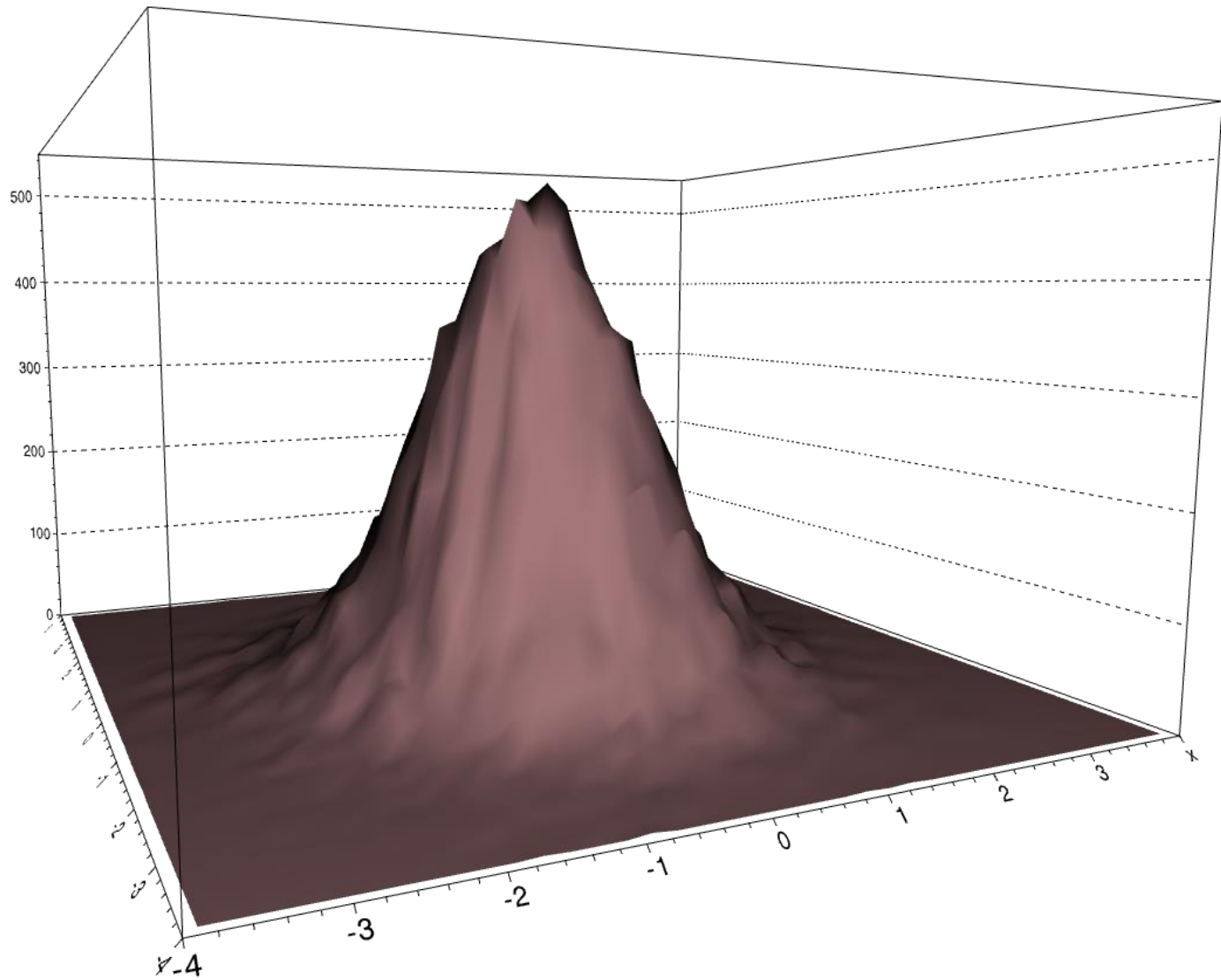
- Size of compressed JSON files comparable with size of normal ROOT files

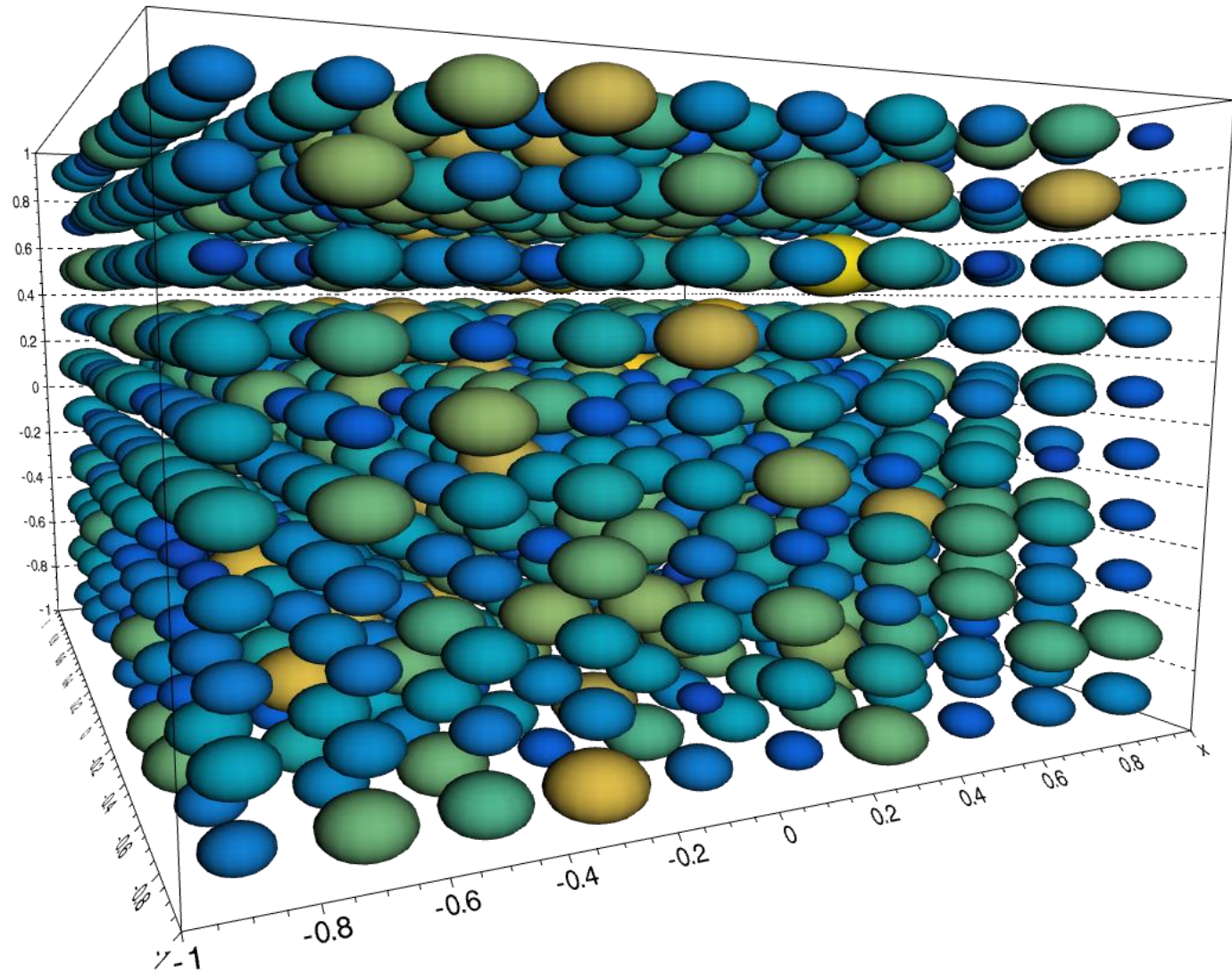
- JSROOT.parse(json) function to decode



histograms 3D display

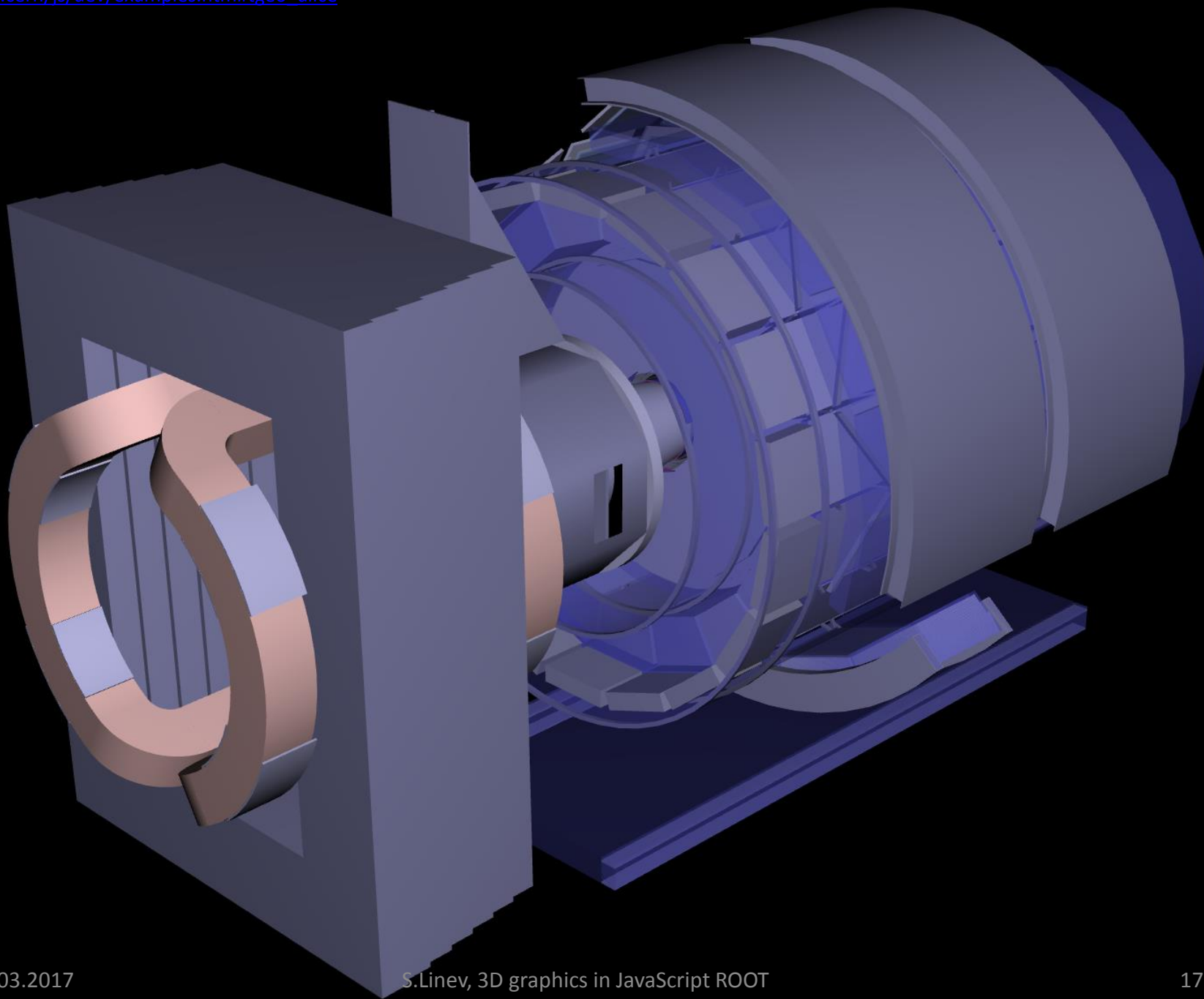
- three.js
- Supports TH1/TH2/TH3
- Interactive zooming
- Tooltips and bins highlight
- Good performance with 200x200 TH2
 - was not a case in earlier JSROOT versions
 - THREE.BufferGeometry

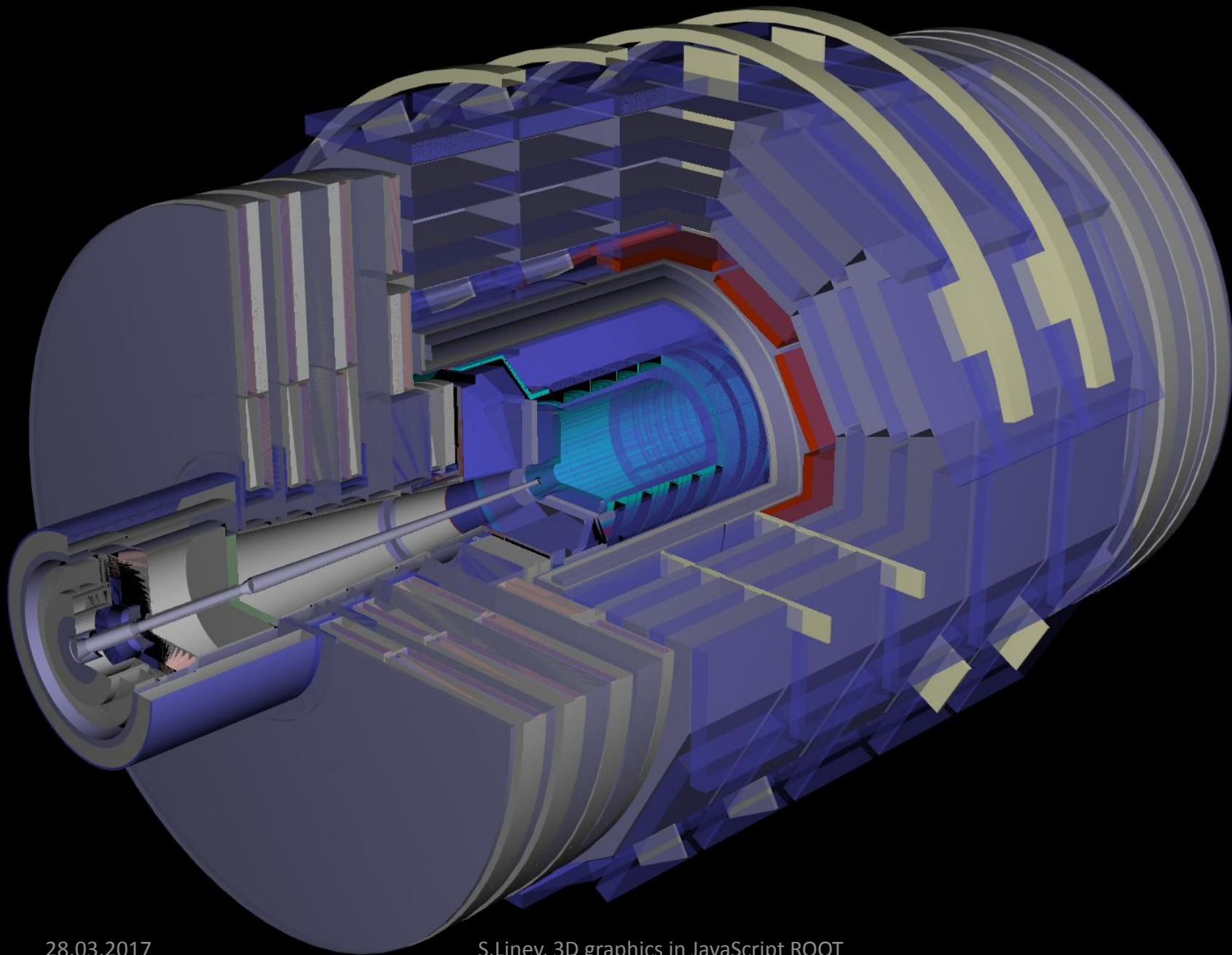


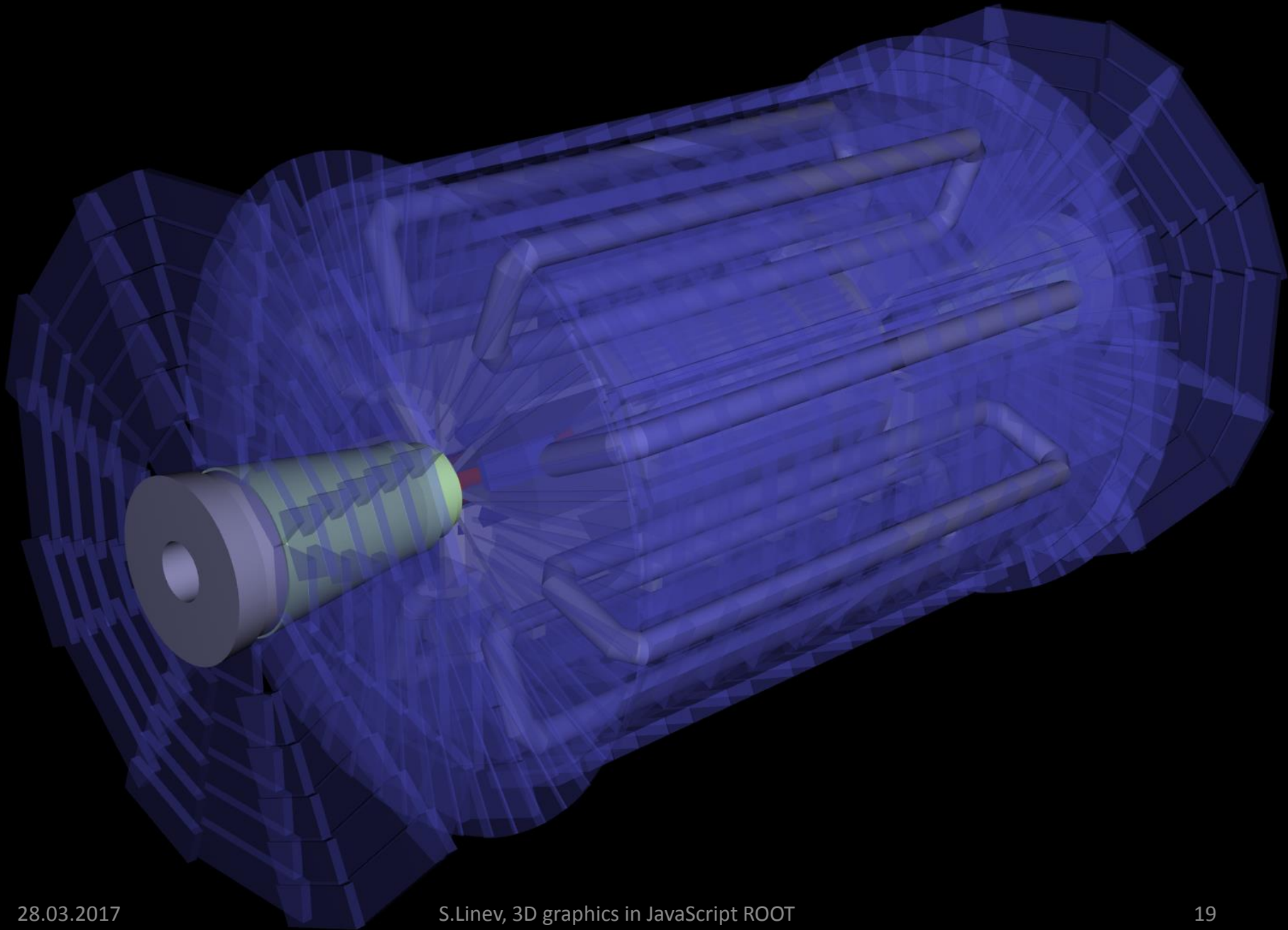


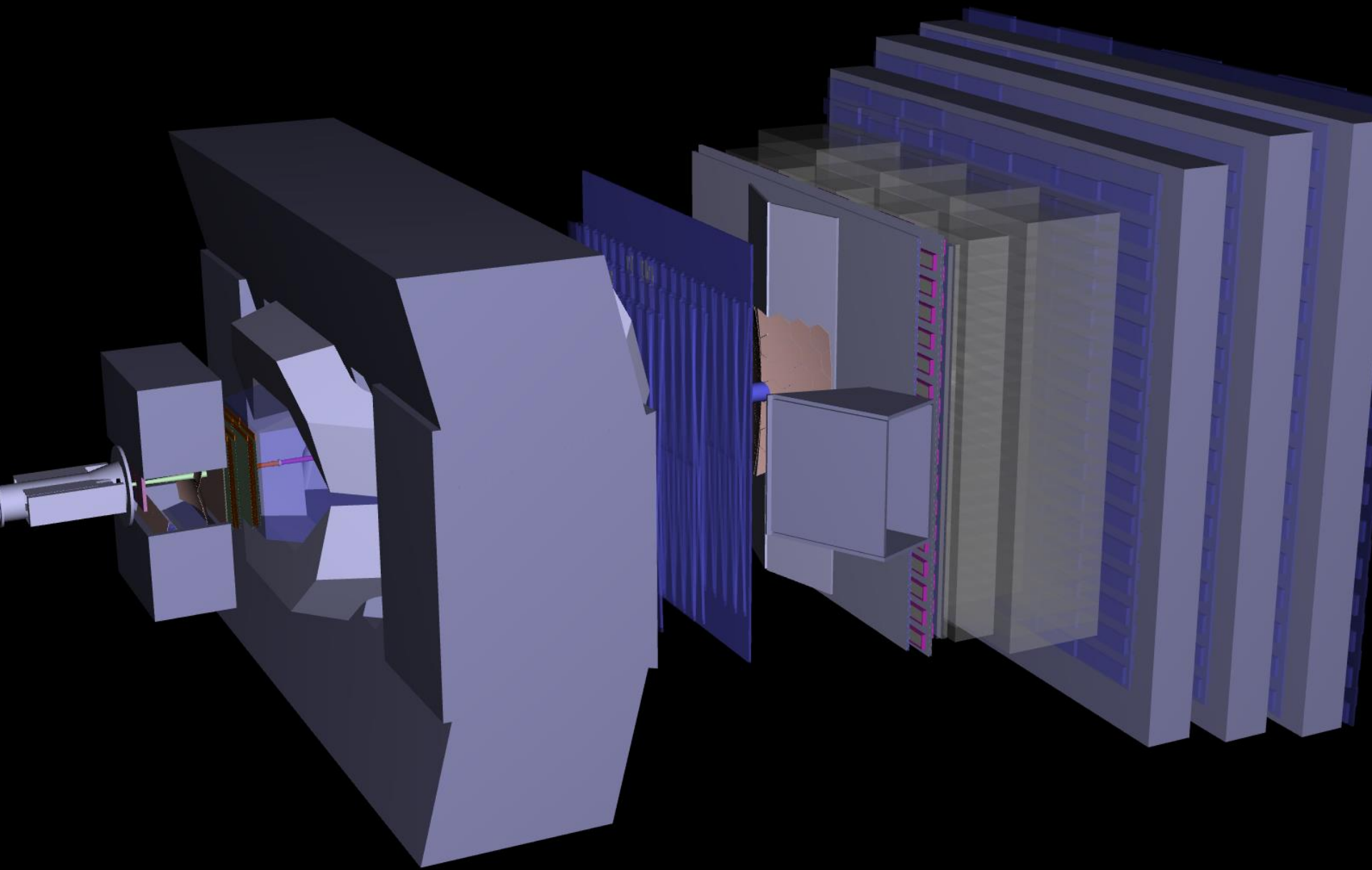
Geometry display

- 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









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

Hierarchy browser

- Explore nodes hierarchy
 - including components of composite shapes
- Display any sub-component
- Mark highlighted volumes
- Toggle visibility flags

Read a ROOT file

JSROOT version dev 21/03/2017

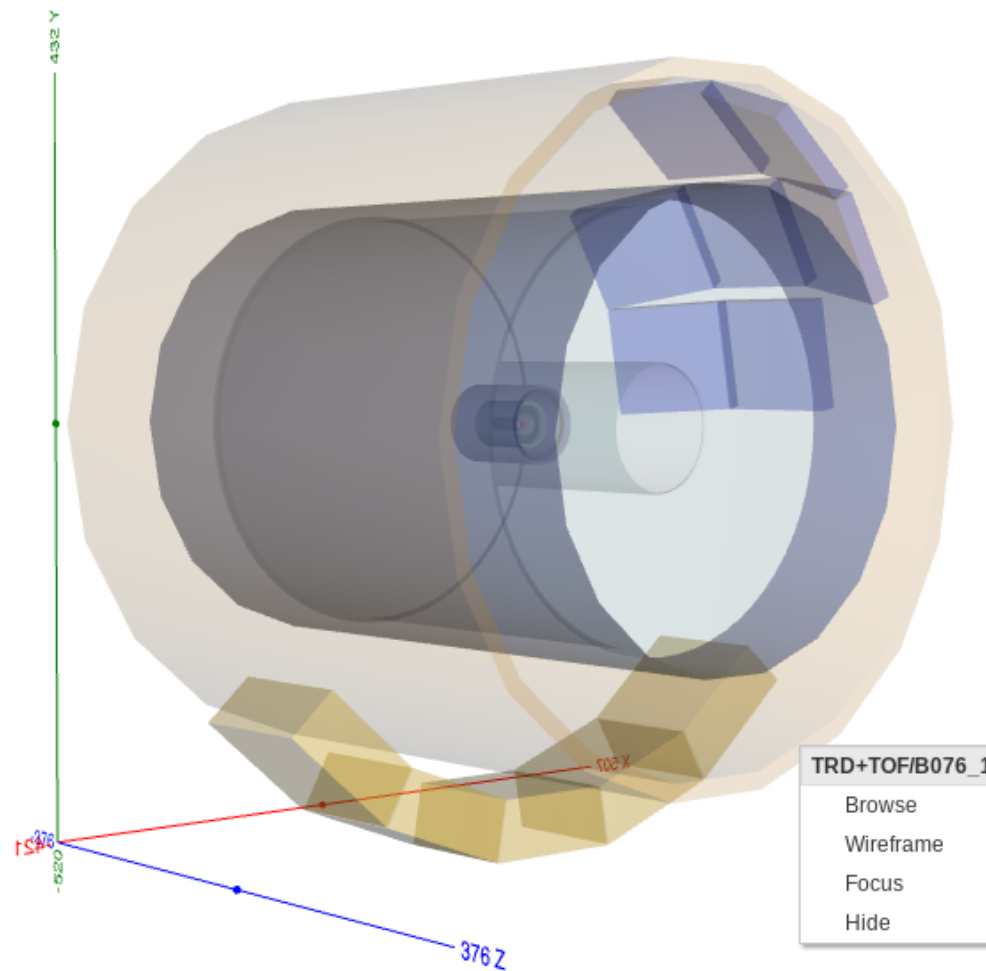
[Read doc](#) how to open files from other servers.

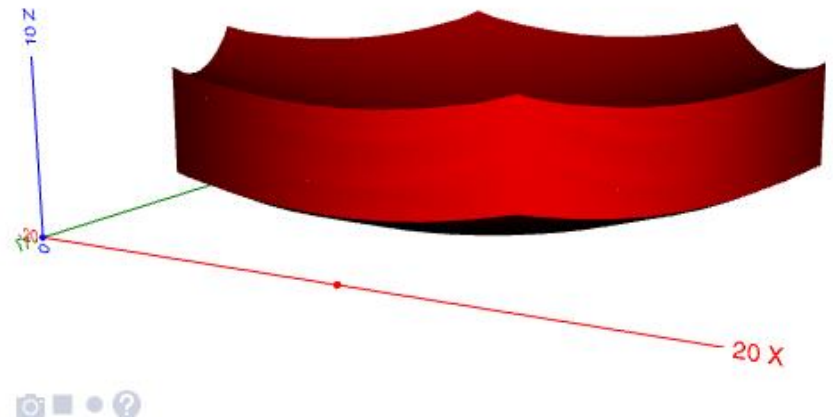
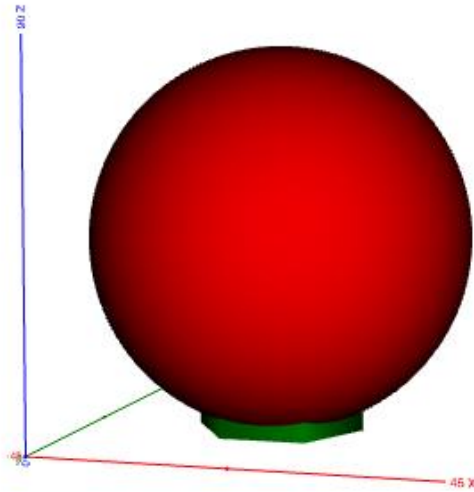
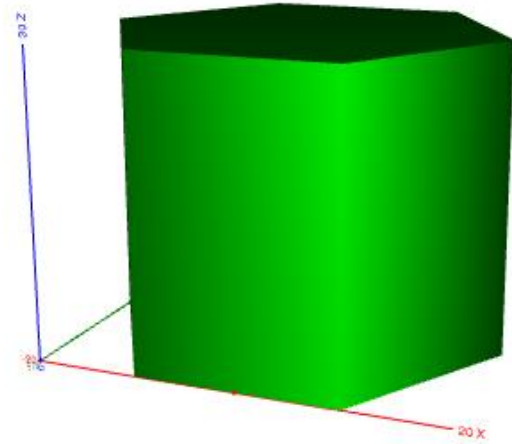
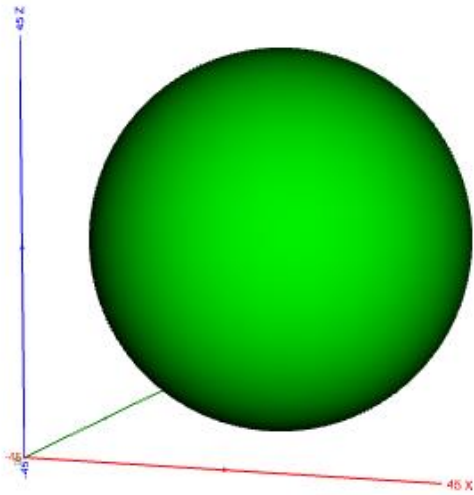
Load Reset simple

[open all](#) | [close all](#) | [clear](#)

evegeoshape.json.gz

- ITS
- TPC
 - TPC_M_1
 - TPC_Drift_1
- TRD+TOF
- PHOS
 - PHOS_1
 - PHOS_2
 - PHOS_3
 - PHOS_4
 - PHOS_5
- HMPID





Tracks and hits

- Overlay with geometry drawing
 - TGeoTrack, TEveTrack, TEvePointSet
 - extract tracks/hits from TTree
 - list can be easily extend
- Hierarchy browser
 - highlight on 3D scene when selected
 - toggle visibility flags

Read a ROOT file

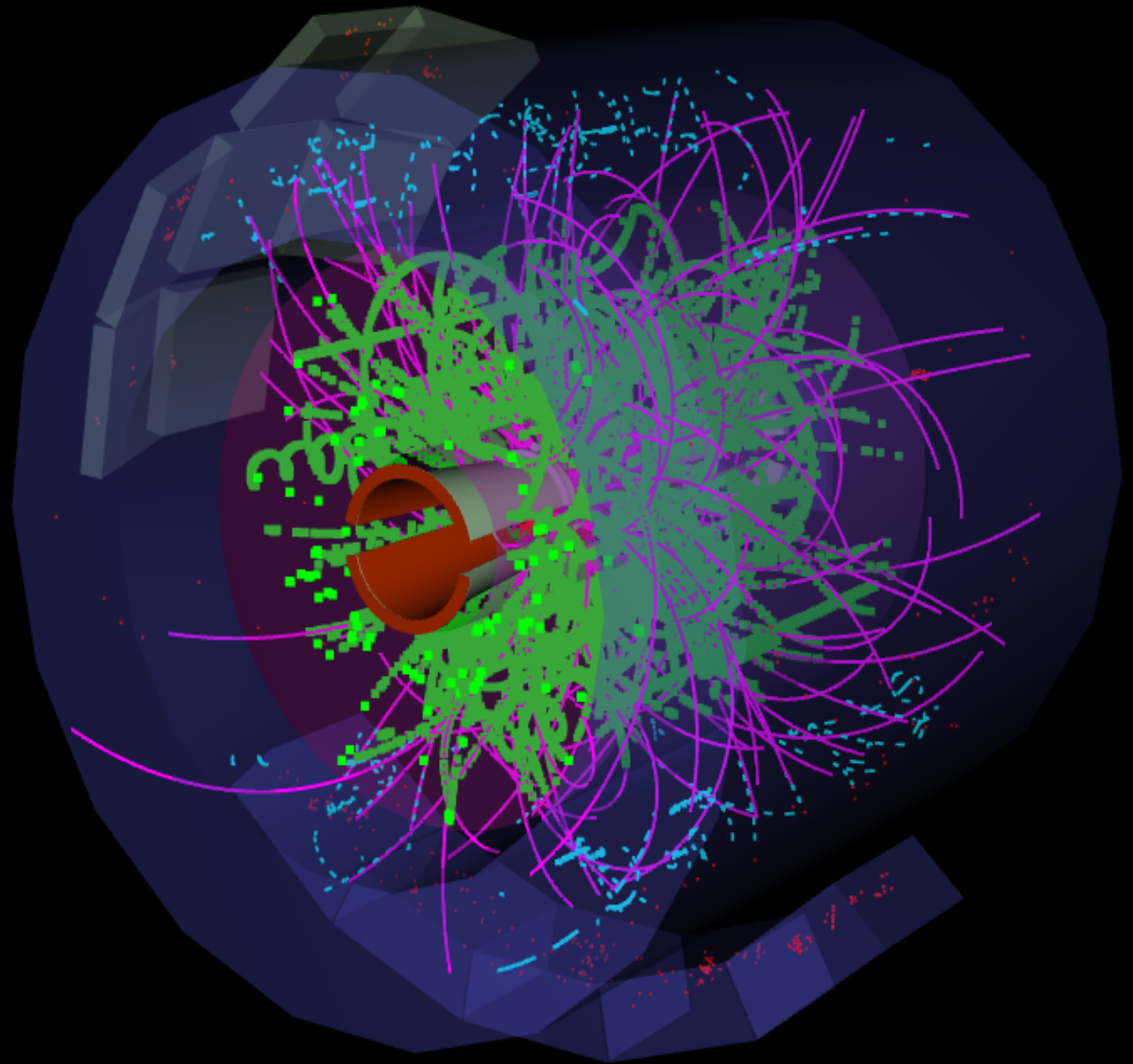
JSROOT version dev 21/03/2017

[Read docu](#) how to open files from other servers.

[open all](#) | [close all](#) | [clear](#)

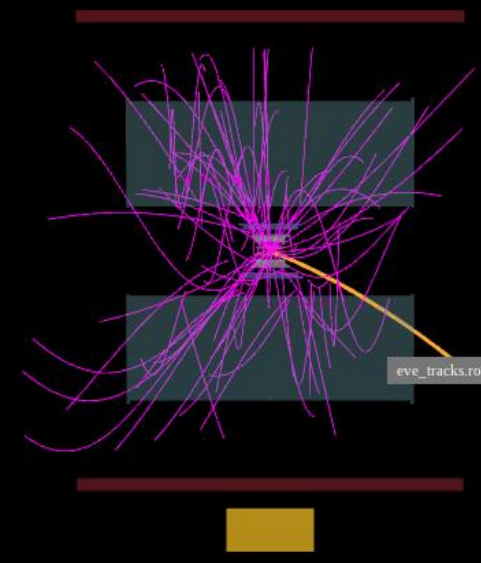
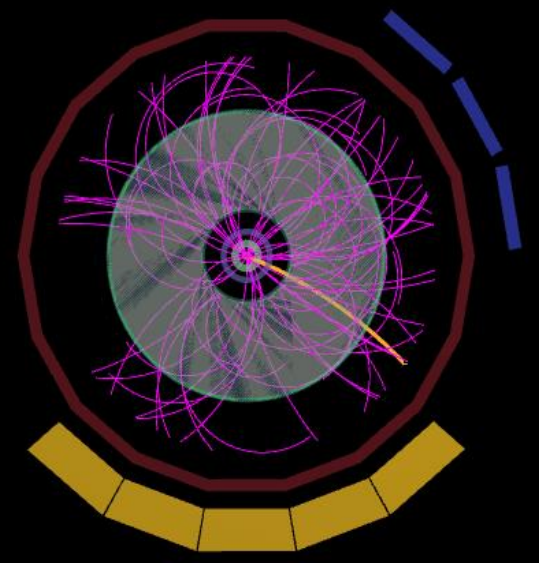
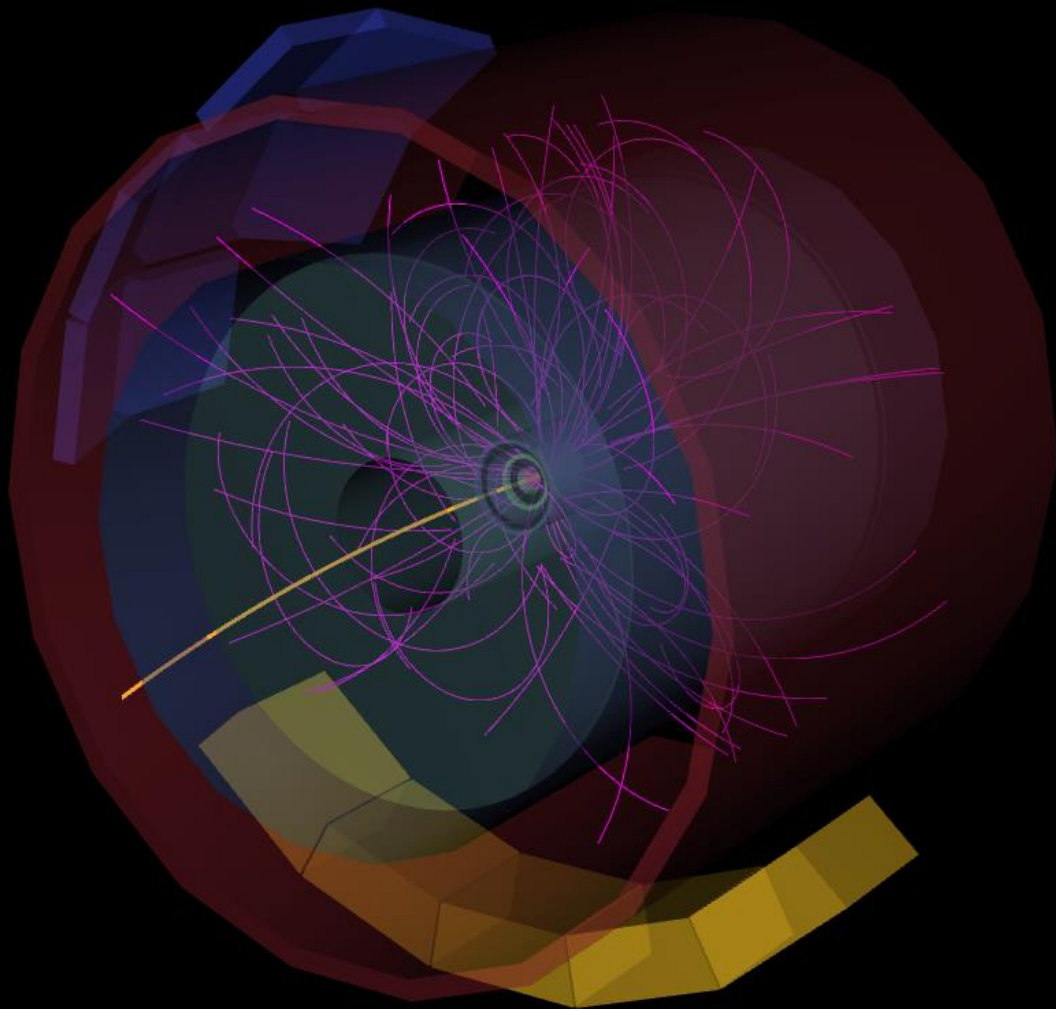
- FMD Hits "
- HMPID Hits "
- PHOS Hits "
- PMD Hits "
- TO Hits "
- TOF Hits "
- TPC Hits 'TPC2.fArray.fR>80'
- TRD Hits "
- VZERO Hits "
- ZDC Hits "
- ProcessID0;1
- tracks;1
 - ESDTrack 612
 - ESDTrack 4673
 - ESDTrack 394
 - ESDTrack 658
 - ESDTrack 972
 - ESDTrack 2008
 - ESDTrack 146
 - ESDTrack 488
 - ESDTrack 973
 - ESDTrack 922

N=119037 type:TEventPointSet



Event display?

- Not yet, but many components are there
 - Geometry and projections
 - Tracks and hits
 - Hierarchy browser
 - THttpServer and JSON for online
- Missing custom components:
 - attributes editors
 - browser actions
 - GUI



eve_tracks.root/tracks;1/[2]



Integration with EVE

- Reuse code as much as possible
- Crucial point – extraction of event data
 - most of EVE classes are not streamable
- Introducing interface to THttpServer

Next plans

- SVG for 2D projections
 - reuse TH2 drawing code
- EVE, JSROOT and THttpServer
- Evaluation of several JS GUI libraries
 - ext.js, webix, Dojo, ...
- Reliable file-based monitoring

Conclusion

JSROOT provides powerful tools for building web applications on base of ROOT

- access to all kind of ROOT data
- interactive display for ROOT classes
- integration with THttpServer

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

