

DEVELOPMENT OF A GUI FOR THE EXPLORATION OF HIGGS MEASUREMENTS

Internship report by Andrius Versockas
Supervisors: André David, Giovanni Petrucciani
Home institution: Vilnius University

Project

2

Project name - **Datacard Viewer**

- **Datacard** - data from the real higgs searches in CMS stored in ROOT (RooFit) format files. On top of that there is a layer of ASCII configuration files describing the contents of these ROOT files.
- **Datacard Viewer** - web application to view content of datacards.

Problematic

How users view datacards?

3

- User uses a text editor to view ASCII configuration files (.txt).
- User surfs the data checking and analysing it.
- When in need to check the ROOT histograms, user memorizes the naming conventions and the names of the needed bin/proccess/nuissance.
- User opens root browser to view the needed ROOT file (.root) and finds the required histograms.
- User uses tools to compares them .

Problematic

Manual browsing drawbacks

4

- ❑ Using text editor and root browser for one datacard.
- ❑ The datacards data can be very large, once many channels are combined. Hard to inspect manually.
- ❑ One ROOT file can contain hundreds of histograms.
- ❑ No sorting or grouping.
- ❑ Datacard can contain multiple ROOT files, really hard to compare.

Purpose of Datacard Viewer

5

❑ One tool for everything

User no longer needs to use text editors or root browser to view datacard contents. Also, no need to use any external tools to merge the wanted histograms.

❑ Searching, sorting and grouping

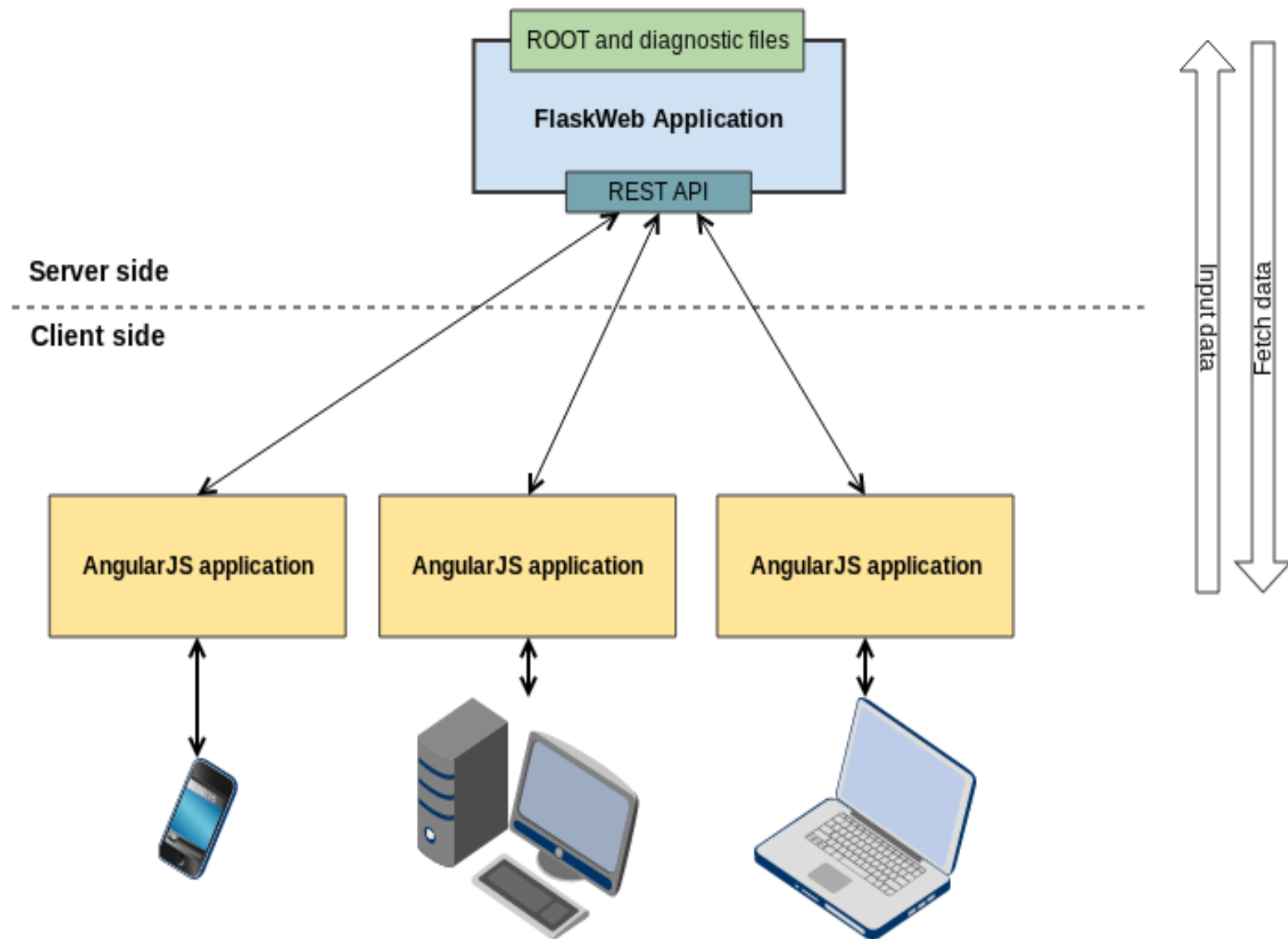
Users can easily manipulate datacards data. For now search only works with data values.

❑ Increased productivity

Simple and clear UI design allows user to easily view the data faster.

Technologies

Datacard Viewer architecture



Technologies

Flask - A Python Web Microframework

7

I chose Flask because it is :

- A lightweight python server side frameworks (has very good performance).
- Flask has big community and great documentation (easy to get started).
- Flask has many plugins (helps accomplish general tasks easier).

Technologies

Client side (frontend) technologies

8

- ❑ AngularJs (v1.2.13) - javascript framework for creating dynamic web application.
- ❑ Bootstrap (v3.1.1) - CSS framework, to make application look attractive.
- ❑ Handsontable (v0.10.5) - Handsontable is a minimalistic Excel-like data grid editor for displaying datacards contents.
- ❑ JSRootIO (v2.1) - to parse and draw histograms from ROOT files.
- ❑ And a few others javascript frameworks to make life easier.

What I have done

9



What I have done

Documentation for Datacard Viewer

10

- ❑ Created deployment descriptor. Detail instructions how to deploy application..
- ❑ Created documentation of Datacard Viewer application for future developers that will improve application.
- ❑ Code, documentation and more is stored in [Github](#) repository, accessible for all.

What I have done

Thing that should be improved by future developers

11

- *Add Multiple root files support.*
Suggestion: create an object to store: filename => keylist.
- *Figure a way out to support root files with "RooWorkspace".*
Suggestion: Update JSRootIO to the newest version when possible, maybe support will be added.
- Add "Configuration" menu to application, to adjust settings e.g. "MASS" parameter.
- Redo "3 in 1" drawing from HStack to Custom.
Reason: HStack draws plots one on top of another. Improve datacardShapeMap parser.
- Fix environment paths for Datacard.py and DatacardParser.py.
- Improve current search, to search/sort by bin/proccess/nuissance.
- More on [Github/tasks](#).

Questions

12



Special thanks to
Bertrand Bellenot