

Status of Event Display for Open Data Access

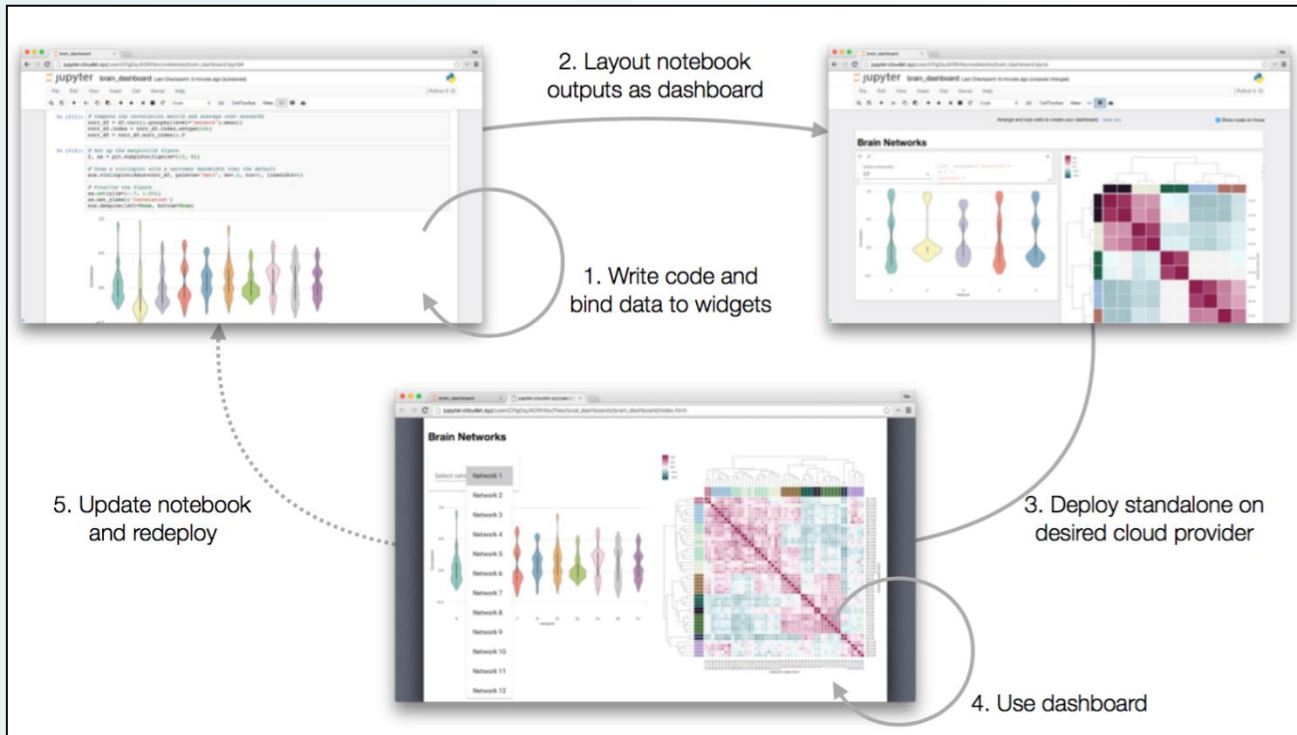
S.Dmitrievsky, JINR

Task: To implement OPERA event display (two 2D-views of an event in ED + one simplified 3D-view of microtracks near the vertex in ECC) as a GUI web-application.

The proposal of **Andrey Ustyuzhanin (AU)** from **Yandex team** was to write the new event display from scratch in **Python** language using the **Matplotlib** plotting library. (In this case no existing **C++** code from the **OpRelease** could be used directly.)

As the first step AU provided me with his examples of drawing of an ECC brick info implemented as **Jupyter notebooks** (interactive web-pages with input commands and the results of their executions displayed in the same window of a browser). Then he suggested to use a set of extensions ([dashboards layout](#), [dashboards bundlers](#), and [dashboards server](#)) for a notebook in order to convert it into a standalone web-application.

Workflow of development of a web-application from a Jupyter notebook



Some issues to be discussed next week with CERN experts:

How the Event display web-application should be integrated in the OPERA site dedicated to the open data access?

Who will be the administrator (or the responsible person) of the site in order to discuss with him the software restrictions/requirements? For example, a lot of software is still based now on **Python2** (but not on **Python3**), though support of **Python2** will probably be stopped in **2020** (?)

It's difficult for me to predict the speed of animation (interaction) of the web-application. (For example, we know that speed of interaction with GUI of the **EventViewer** running on remote machine could be very low.) There are several possibilities exists to increase the speed of python programs or to combine them with more power libraries written in **C++**. May be it would be possible to use **ROOT** and write the application in **C++** from the very beginning (it would be more easy because it's similar to our off-line **EventViewer**) since **ROOT** now has a **Jupyter** kernel..