A Web-based Solution to Visualize Operational Monitoring Data in the Trigger and Data Acquisition System of the ATLAS Experiment at the LHC

G. Avolio[1], M. D'Ascanio[1], G. Lehmann Miotto[1], I. Soloviev[2]
On Behalf of the ATLAS Collaboration

[1]CERN, Geneva, Switzerland
[2]UCI, Irvine, CA, USA

1. Introduction

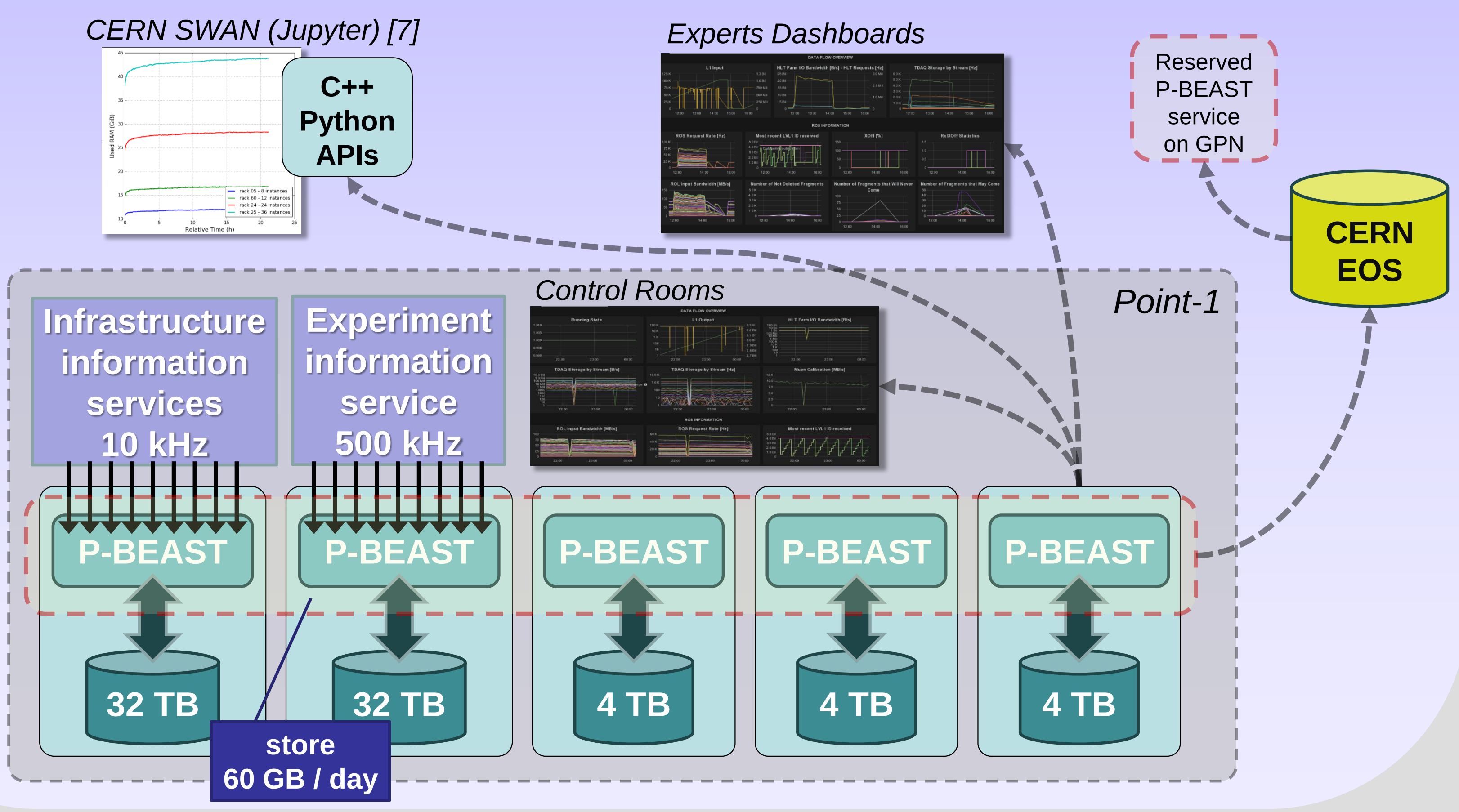
The Trigger and Data Acquisition (TDAQ) [1] system of the ATLAS detector [2] at the Large Hadron Collider (LHC) at CERN is composed of a large number of distributed hardware and software components (about 3000 machines and more than 25000 applications) which, in a coordinated manner, provide the data-taking functionality of the overall system.

During data taking runs, a huge flow of operational data is produced in order to constantly monitor the system and allow proper detection of anomalies or misbehaviors. In the ATLAS TDAQ system, operational data are archived and made available to applications by the **P-BEAST** (*Persistent Back-End for the Atlas Information System of TDAQ*) [3] service, implementing a custom time-series database.

The possibility to efficiently visualize both real-time and historical operational data is a great asset for the online identification of problems and for any post-mortem analysis. A web-based solution has been developed to achieve such a goal: the solution leverages the flexibility of the **P-BEAST** archiver to retrieve data, and exploits the versatility of the **Grafana** [4] dashboard builder (based on the **AngularJS** [5] framework) to offer a very rich user experience.

2. P-BEAST Service

P-BEAST services are running inside ATLAS experiment area (Point-1) on 5 nodes communicating over CORBA protocol. The raw data of LHC Run 2 are stored inside Point-1 and replicated on CERN EOS [6] for long term archive. The data can be visualized using **Grafana** dashboards over HTTP protocol. The C++ and Python programming interfaces are available for further complex analysis.



3. Grafana

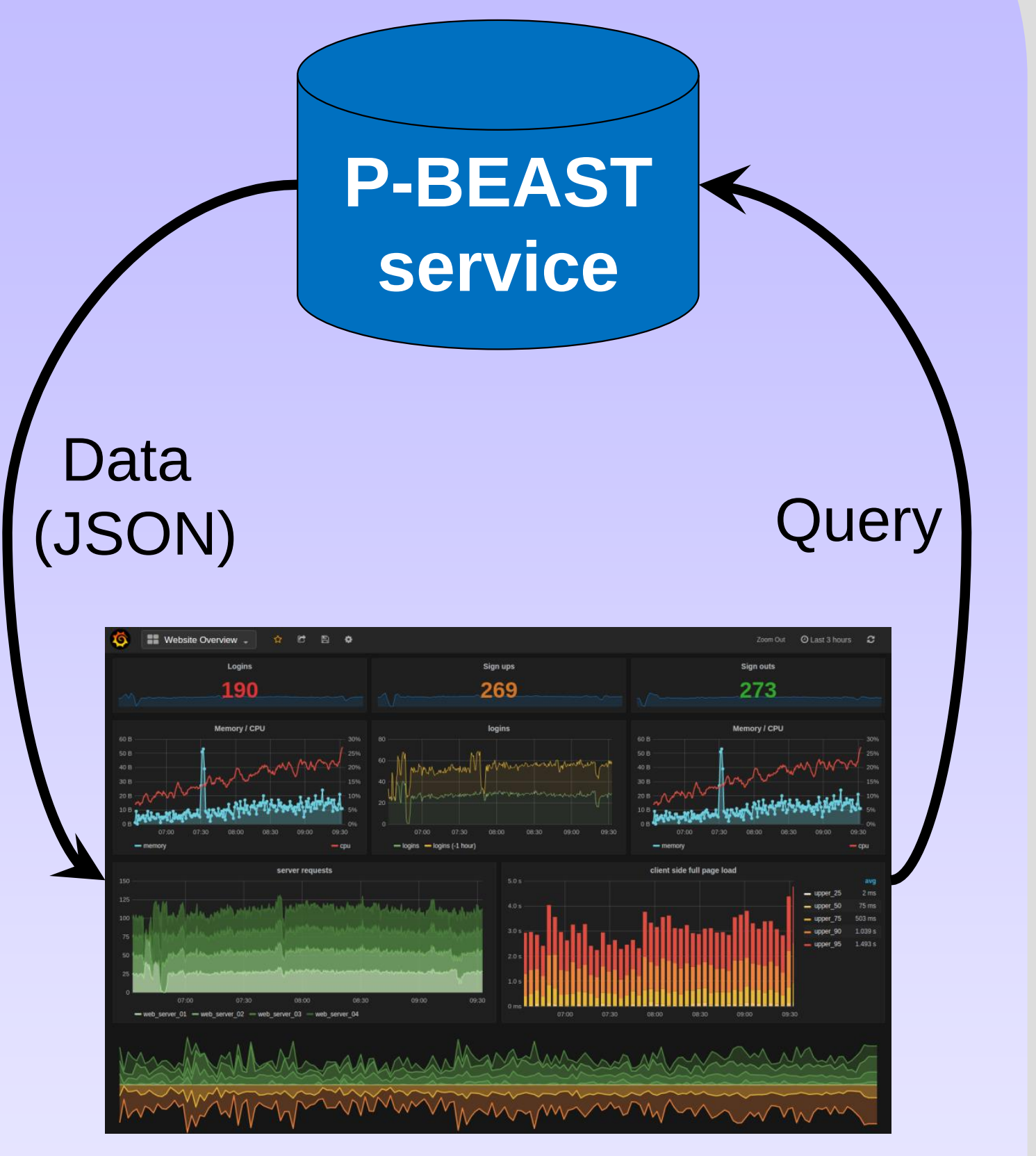
Grafana has been chosen as the tool to visualize data stored in the **P-BEAST** database.

Here is a list of the **Grafana** main features driving our choice:

- Dashboard management made easy;
- Native support for time series;
- Rich graphing;
- Support for template variables;
- Support for dynamic dashboards via *JavaScript*;
- Plug-in like architecture allowing integration of custom data sources;
- Pure client-side application, facilitating its deployment.

Grafana natively includes support for several time series databases (*Graphite*, *InfluxDB*, *ElasticSearch*, *OTSDB*). The integration of the **P-BEAST** back-end as an additional data source was the main challenge faced during the development phase. That included:

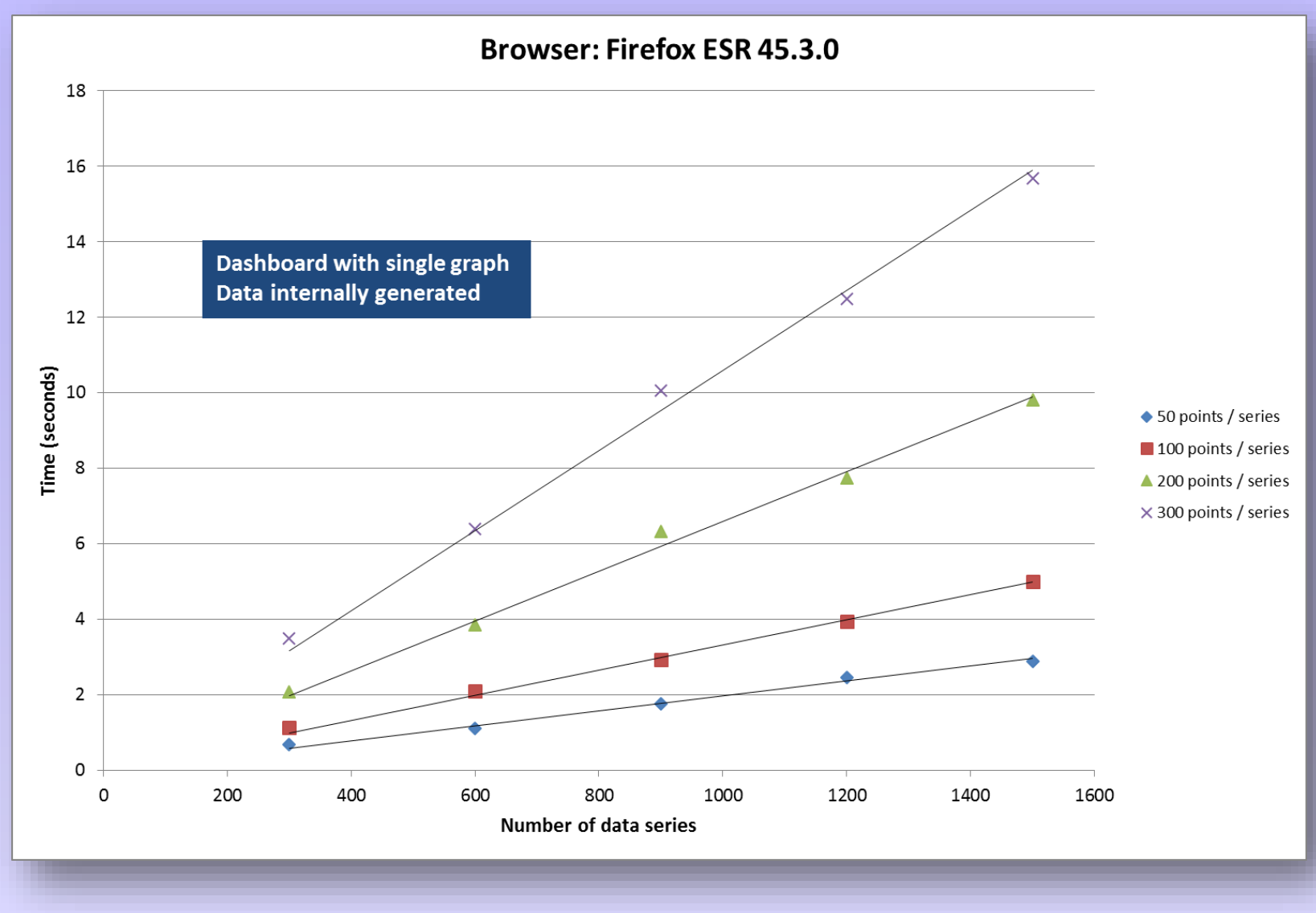
- Implementation of proper REST calls to the **P-BEAST** server;
- Data post-processing in order to comply to **Grafana**'s data format;
- Building of dedicated panels to configure and set query specific parameters.

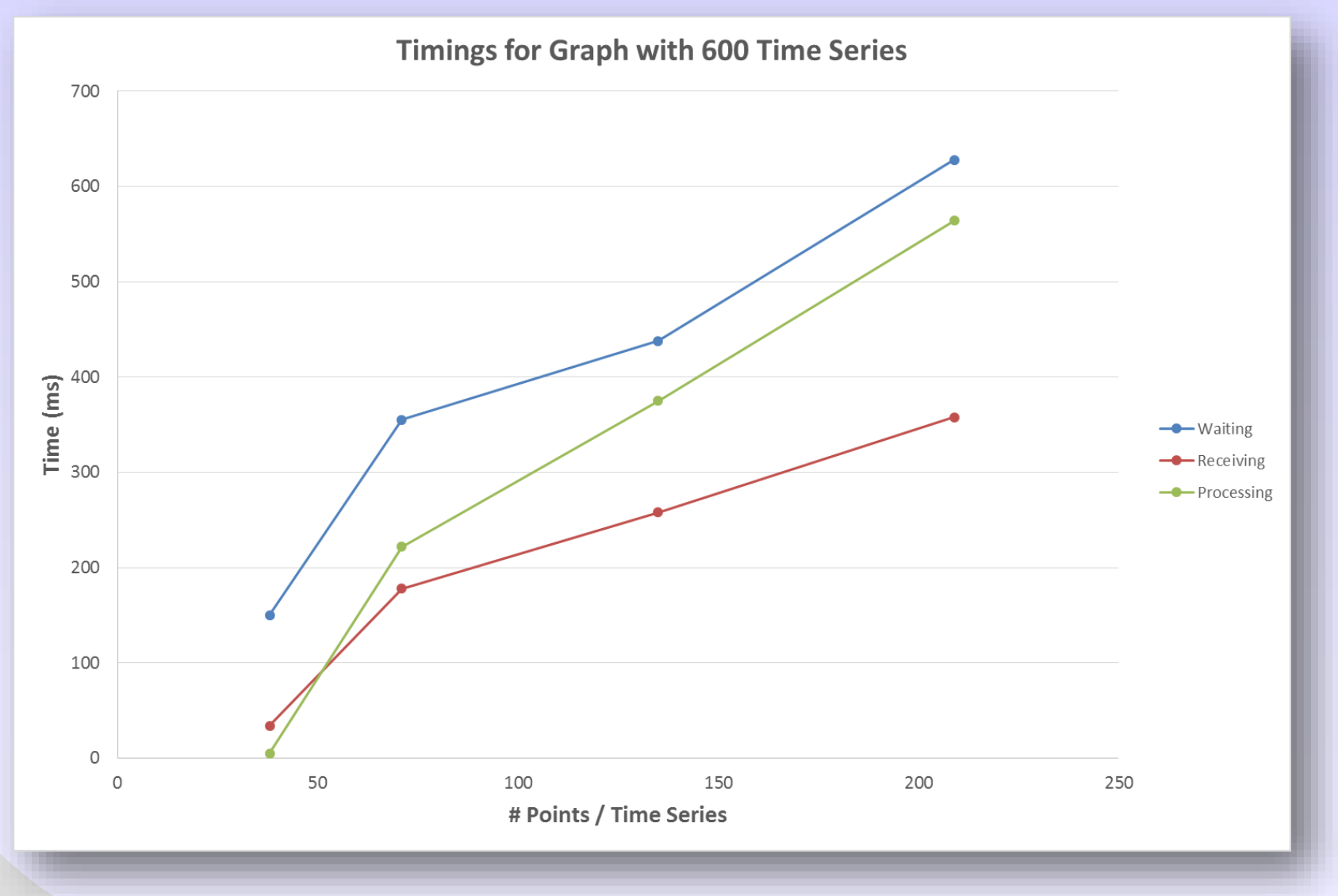


Particular attention has been paid to the definition of the JSON structure representing the **P-BEAST** data. Given the huge amount of data to be shown, a too expressive JSON format proved to be a bottleneck in performance.

4. Performance

The plot on the right shows the time needed by the browser to render a dashboard with a single graph, as a function of the number of drawn time series and for different number of data points per series. Data are internally generated, so that the results do not depend on the performance of the **P-BEAST** back-end (including the network latency), but just on the **Grafana** software stack and the browser's *JavaScript* engine. The tests were performed using **Firefox ESR 45.3.0** and **Grafana 1.9.1**. As it can be noticed, the rendering time greatly depends on the number of data points per time series. Taking into account those results, it was decided to parametrize the queries to **P-BEAST** using a *maximum number of data points* per time series. In this way the user is able to configure reach and complex dashboards keeping performance under control.





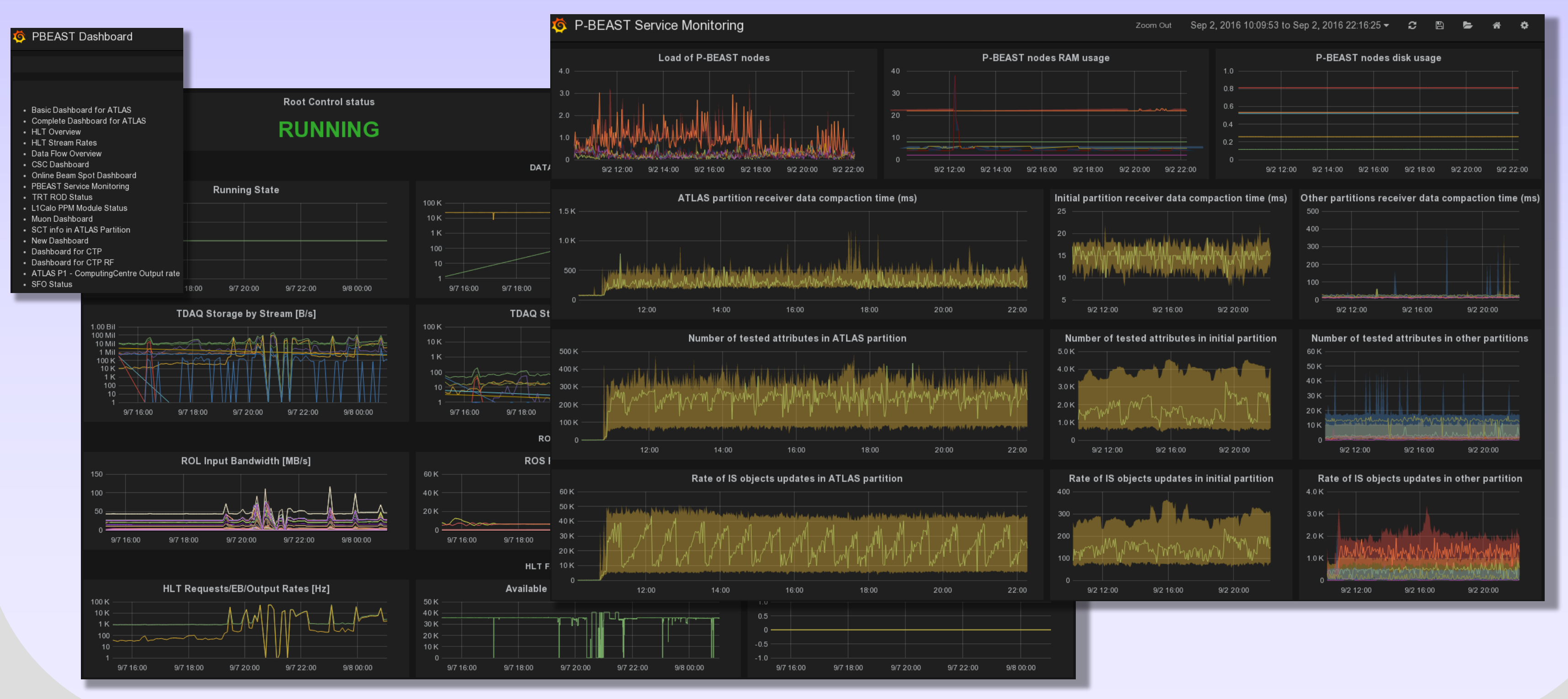
The plot on the left shows typical times for the interaction between the **Grafana** dashboard and the **P-BEAST** back-end. Tests were performed with a single graph with 600 time series and varying the number of data points per series, using the same browser and **Grafana** versions previously mentioned. Reported times are:

- *Waiting*: time waiting for a response from **P-BEAST**;
- *Receiving*: time taken to read the entire response from **P-BEAST**;
- *Processing*: time taken to properly format data received from **P-BEAST** and inject them into the **Grafana** engine.

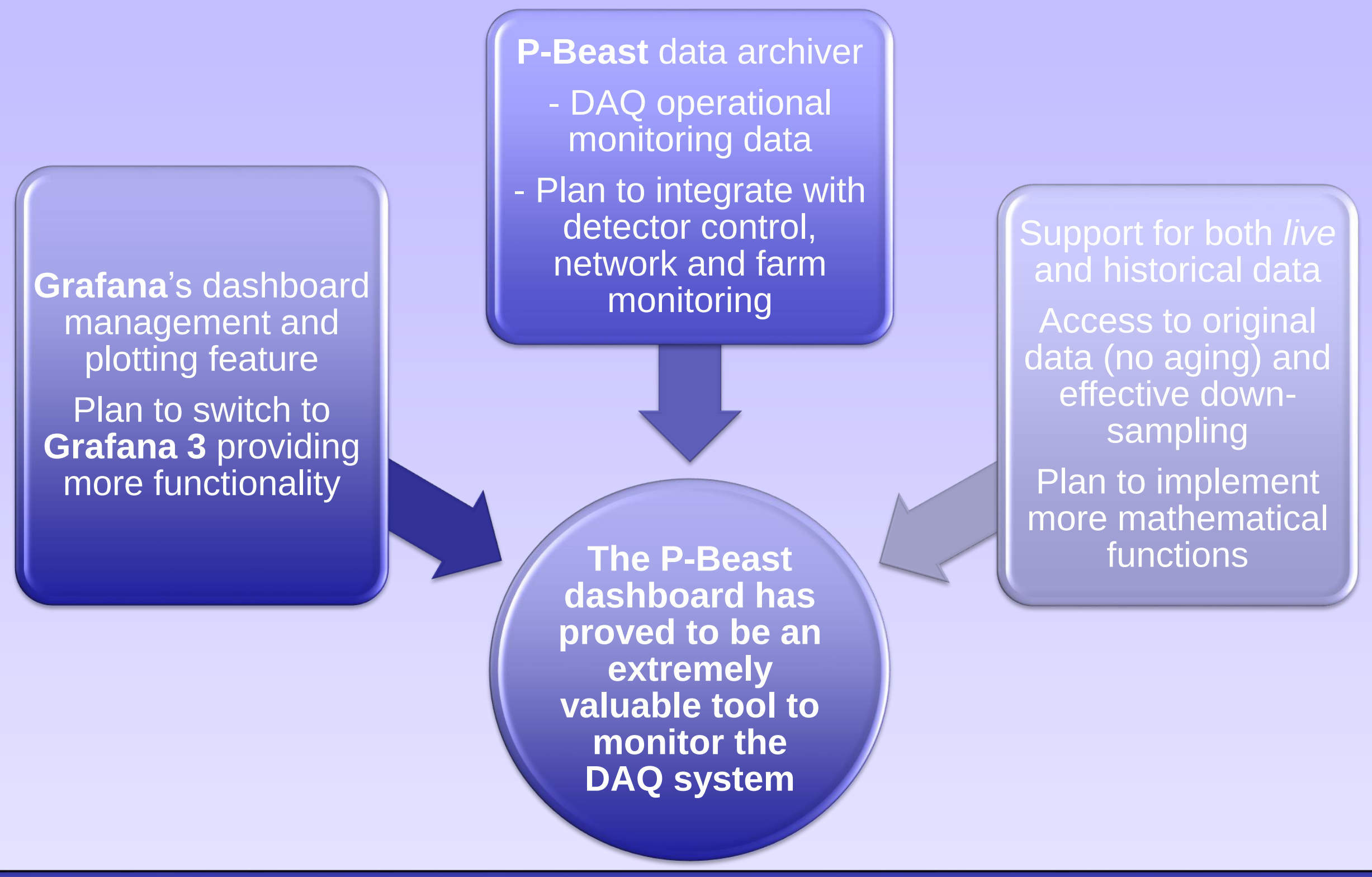
As it can be noticed, the contribution of the processing time becomes less important increasing the number of data points per time series.

5. User Dashboards

The ATLAS experts have a possibility to construct dashboards and to store them into a central place, that is accessible by all members of the experiment. The dashboards can present online data as well as archived ones without loosing data precision. For requests over large time interval, the down-sampling of the data is performed by the **P-BEAST** service using a persistent cache on the server side. This allows to **P-BEAST** service response time relatively short for any time intervals and to pass amount of data small enough keeping user's internet browser responsive even for large number of plots per page.
Currently about 20 different dashboards are configured and regularly consulted by experts and shifters in the ATLAS control room. The richest dashboards are made up of more than 20 plots with single graphs showing up to 1000 time series.



6. Summary and Plans


P-Beast data archiver
- DAQ operational monitoring data
- Plan to integrate with detector control, network and farm monitoring
Grafana's dashboard management and plotting feature
Plan to switch to **Grafana 3** providing more functionality
Support for both *live* and historical data
Access to original data (no aging) and effective down-sampling
Plan to implement more mathematical functions
The P-Beast dashboard has proved to be an extremely valuable tool to monitor the DAQ system

References

1. The ATLAS Collaboration, 2002, *ATLAS high-level trigger, data-acquisition and controls: Technical Design Report*
2. The ATLAS Collaboration, 2008, *The ATLAS Experiment at the CERN Large Hadron Collider*, J. Instrum. 3
3. A. Sicoe, I. Soloviev, *New Persistent Back-End for the ATLAS online information service,* Real Time Conference (RT), 2014 19th IEEE-NPSS
4. Grafana, *http://grafana.org*
5. AngularJS, *https://angularjs.org*
6. EOS Storage, *http://eos.readthedocs.io*
7. The Swan Service, *http://swan.web.cern.ch*