FRONT-END MONITOR AND CONTROL WEB APPLICATION FOR LARGE TELESCOPE INFRASTRUCTURES: A COMPARATIVE ANALYSIS

Stefano di Frischia, Mattero Canzari (INAF-OAAb, Teramo, Italy)
Valentina Alberti (INAF-OAT, Trieste, Italy)
Athos Georgiou (CGI Scotland, Edinburgh, UK)
Hélder Ribeiro (Universidade do Porto, Porto, Portugal)

M&C Front-End Web Application: a key feature

A robust monitor and control front-end application is a crucial feature for large and scalable radio telescope infrastructures such LOFAR and SKA, whereas the control system is required to manage numerous attribute values at a high update rate. Two state-of-the-art web applications such Grafana® and Taranta are taken into account, developing a comparative analysis between the two software suites. Such a choice is motivated mostly because of their widespread use together with the TANGO Controls Framework.

Objectives

A typical use-case is analyzed, whereas an interactive dashboard is built to monitor and control a hardware device. Then, we set up some comparable metrics to evaluate the pros and cons of both platforms, regarding the technical and operational requirements, fault tolerances, developers and operators efforts, and so on. The main objective is to offer the stakeholders a basis for future choices.

Use Case: create and monitor a dashboard

It has been chosen the following use case which can be representative of a real use case in the aforementioned projects like LOFAR or SKA: monitor a dashboard that shows the image of a Printed Circuit Board (PCB) which represents a real hardware device of the observing station. The main attributes of the device (and therefore their corresponding values) are placed upon the image with the possibility by the operator to read and/or modify them.

Ideally, the steps are the following:

- create a new dashboard inside the chosen front-end framework;
- place an image of the selected PCB as background image of the dashboard;
- retrieve from the data source the list of the attributes which belong to the chosen device;
- place each pair label-values upon the image, selecting the right format, graphic, unity, etc.;
- check if the dashboard is updated correctly after the device values updating;
- modify a read/write attribute if possible directly from the GUI.

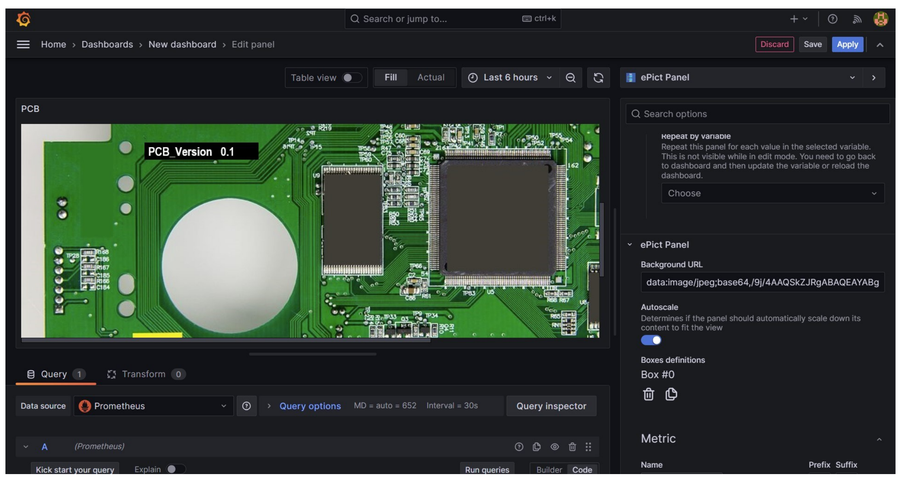

Fig.1: snapshot of Grafana ® edit panel window

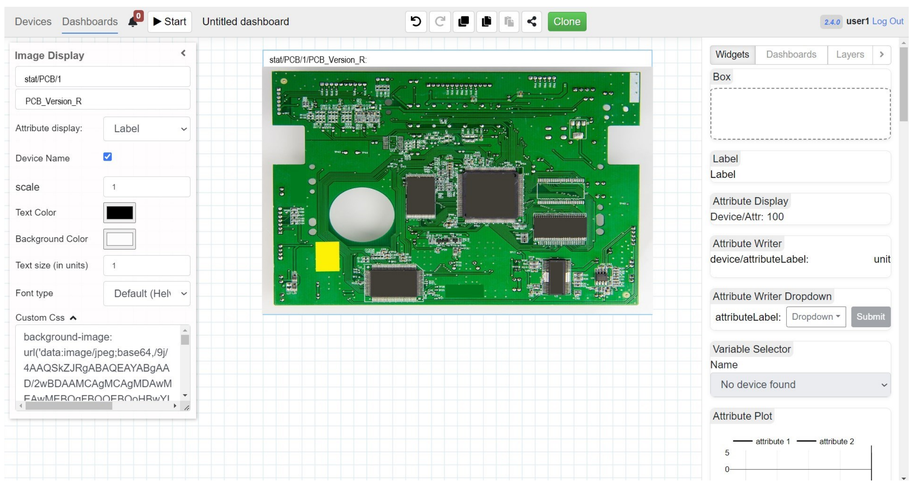

Fig.2: snapshot of Taranta create dashboard window

COMPARATIVE TABLE RESULTS

	GRAFANA®	TARANTA
Repository	https://github.com/grafana/grafana	https://gitlab.com/tango-controls/web/taranta-suite
Version	10.1.0	2.4.0
License	AGPL-3.0-only	LGPL 3.0
Target platforms	All	Windows not supported
Main frontend programming language	Typescript	React
Contributors	Around 2100	Around 40
Installation	Pretty straightforward	It may require to tweak some parameters in Docker Compose file
Other software needed for present use case	Prometheus, Prometheus-node Exporter	TangoGQL, MongoDB and other accessory tools are automatically installed
TANGO Controls support	Through Prometheus, or directly inject TangoDB as data source (f.e. MariaDB). Needs proper configuration.	Tailored on TangoDB, it needs only its host address to connect
Data Sources	Supports a multitude of data sources like Prometheus, Loki, Elasticsearch, InfluxDB, Postgres and many more.	Tailored on TangoDB
Developer preparation for dashboard creation	It may require to modify JSON files as well as write queries in the data source format	Minimum
End user preparation for dashboard interaction	Minimum	Minimum
On-line Support	Slack channels, community forums and dedicated Grafana Labs contact support	Developer community contacts and slack channesl
Documentation	Official docs along with tutorials, webinars, videos and blogs	Repository and Tango community documentation
Tool Customization	Lots of extensions and plugins developed by the community	No plugins outside the official suite
GUI usability	Highest. UX/UI dedicated development	High. Simple and straightforward
Mobile responsiveness	High. Dedicated development	Basic
Dashboard visual appeal	High. Many customizable graphic options.	Medium. It focuses on readability rather than graphic embellishment
Data visualization tools	A broad choice of widgets that should meet all data format needs. Less customizable from the point of view of the Tango device monitor and control.	A minor number of widgets but each one of them tailored on a specific Tango Controls feature
Dashboard customization	High, but it required more effort to adapt it to the present use case	High, it required less effort since it met easily the present use case
Alerting	Grafana Alerting System	Taranta Alerting features
Authentication and Authorization	Grafana Auth or other auth providers	Taranta Auth package
TANGO command support	Not present. Not allowed through Prometheus	Present. Dedicated widget
Dashboard minimum refresh rate	Between 1s and 5s	Handled by websocket through an asynchronous event management
Suitability for business companies	Suitable both for small and large companies	Suitable for large companies which adopt Tango Controls framework
Scalability	Dedicated Grafana Cloud platform suitable for different requirements	Minikube and Docker as preferred deployment tools

ACKNOWLEDGEMENTS

Many thanks to the INAF staff, the ASTRON staff, the SKAO developers community and the TANGO community for their support, great work and ideas.