

Realtime telescope data visualisation using web technologies



When performing onsite or remote observations in a telescope, it is very critical to have a good feedback about the current status of your ongoing observation.

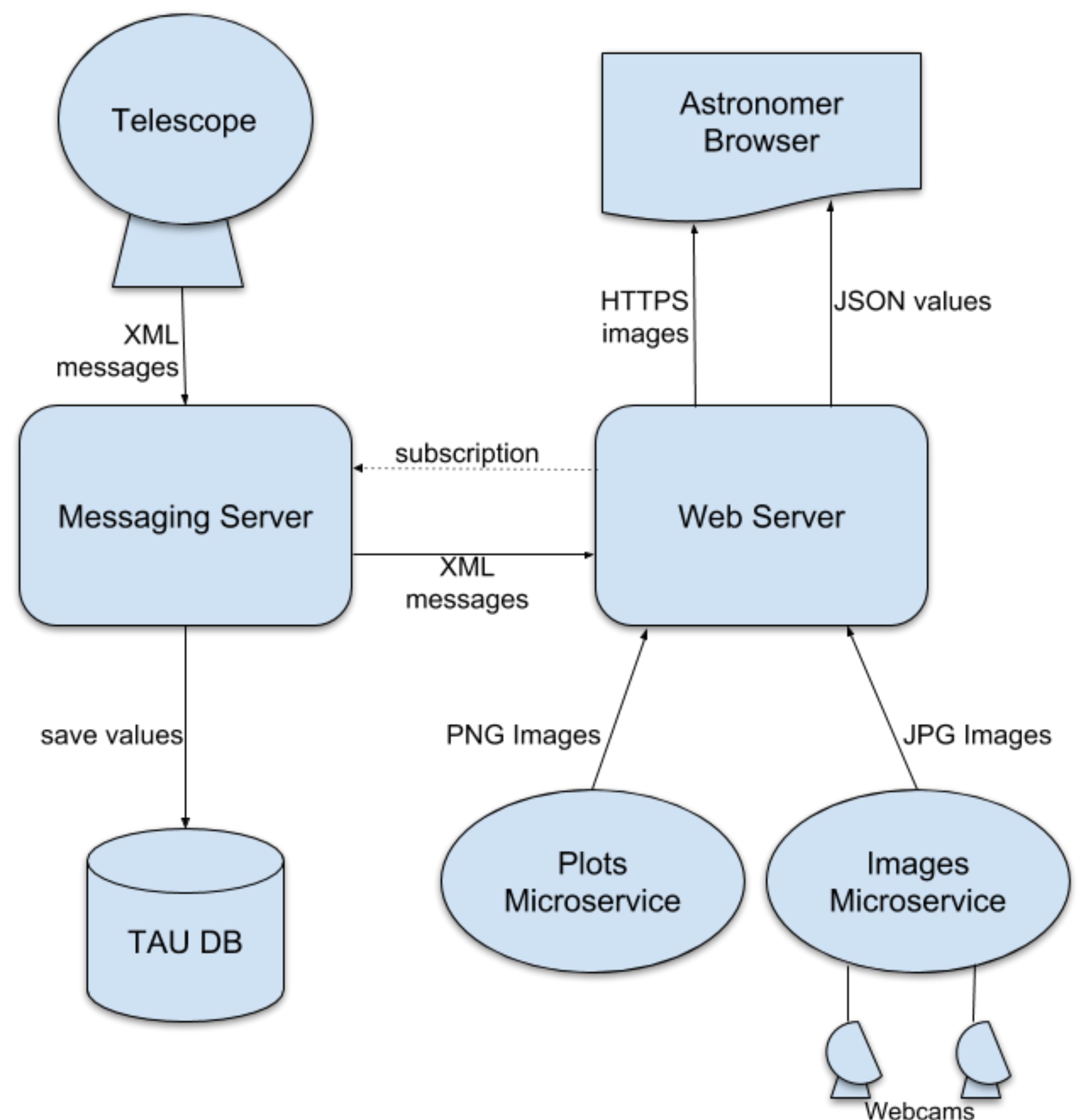
Nowadays the web technology have evolved to allow us to get data in realtime with the advantage of using just a web browser.

In our telescope we found the need of updating our previous outdated monitoring system which is currently showing information about the status of the telescope, the last scans plots, a view of the current weather conditions, etc.

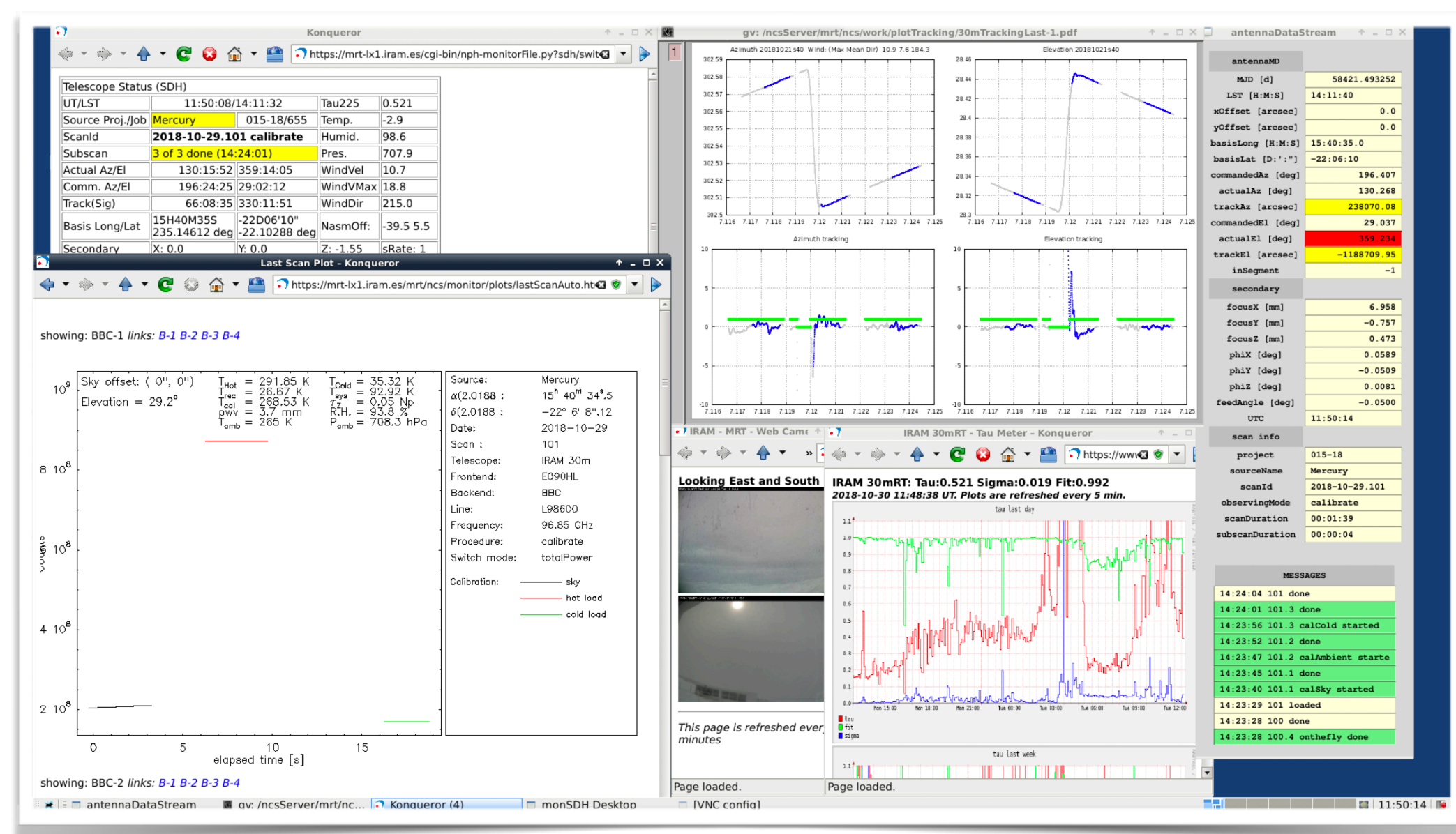
This poster shows how the data visualisation has improved by using newer technologies like microservices, websockets and messaging, as well as the structure developed to integrate them succesfully in a reliable and more attractive way.

The new monitor gets the data following this process:

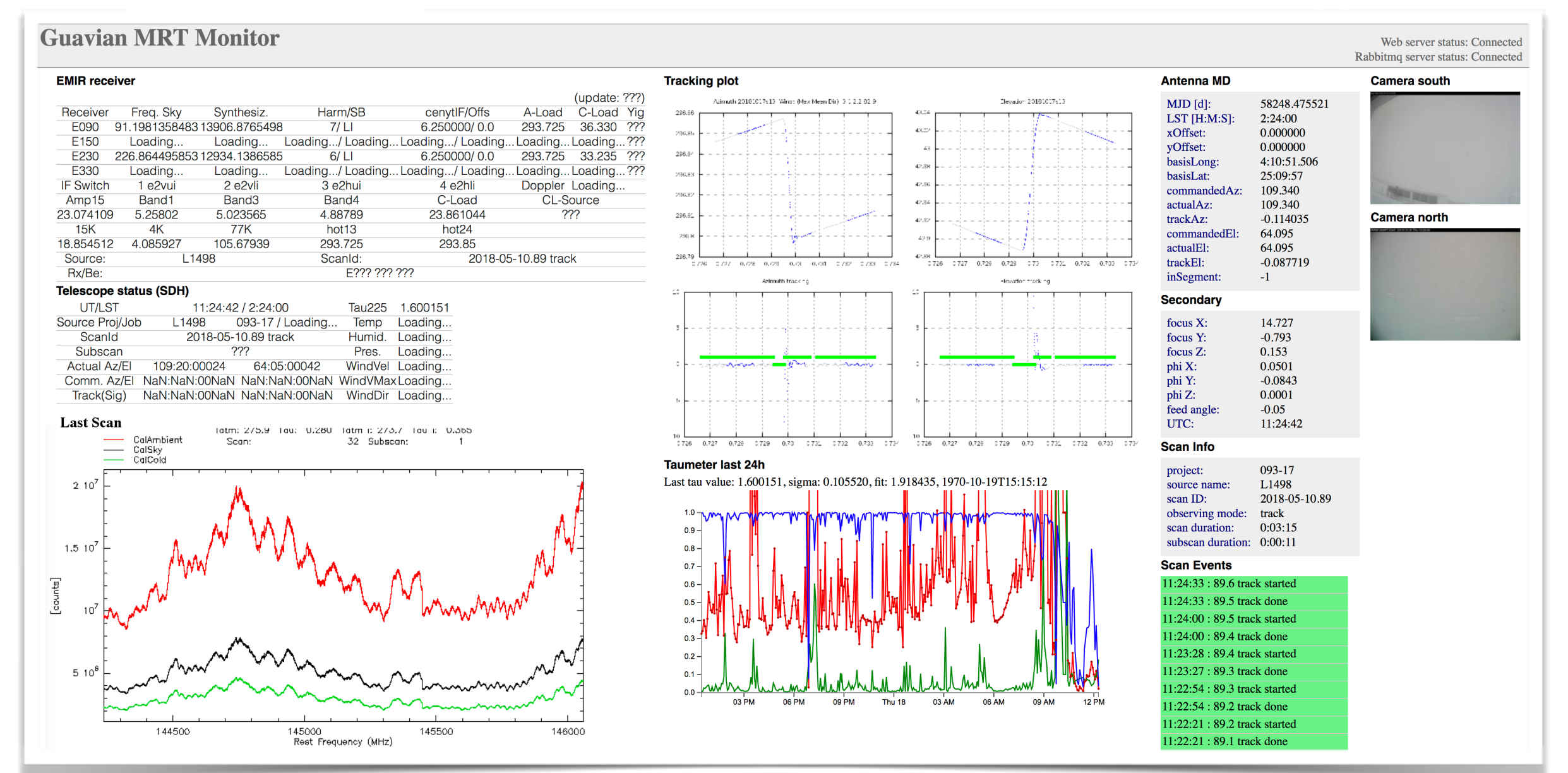
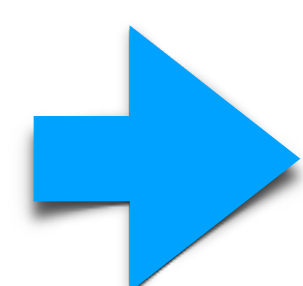
1. The telescope control software continuously sends *XML* messages to the *RabbitMQ* server about its current status.
2. The messaging server saves some values to databases like *Tau*. It also forwards some of the messages to the web server according to the already specified subscription.
3. The web server is in charge of collecting all the relevant data for the astronomer. The telescope status is obtained from the messages but some plots and images are provided by *microservices*.
4. The images, plots and values are sent to the astronomer browser using websockets. This way the server can update the information as soon as it is available, obtaining a realtime experience.



Python 3, Flask and RabbitMQ are used in the backend. *Docker* containers helped with an easy to set up development environment. The frontend is using *jQuery, Vue.js, websockets* over *socket.io* and *SVG*.



Old VNC desktop monitor



New web interface

