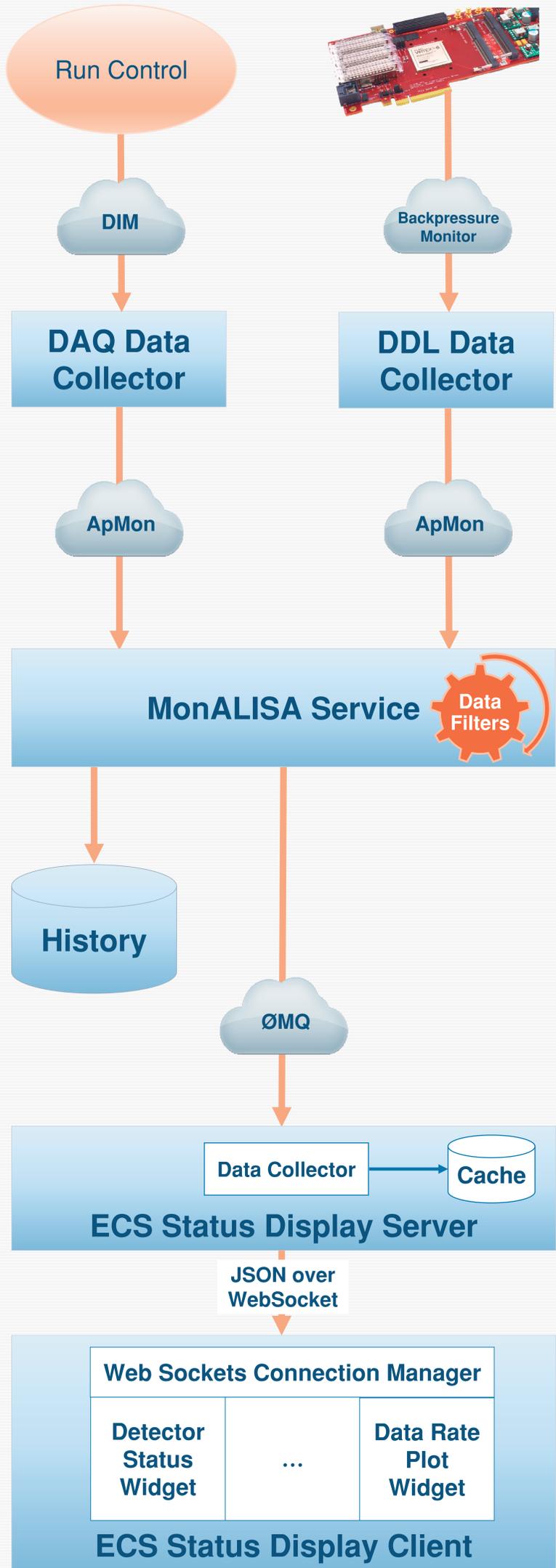
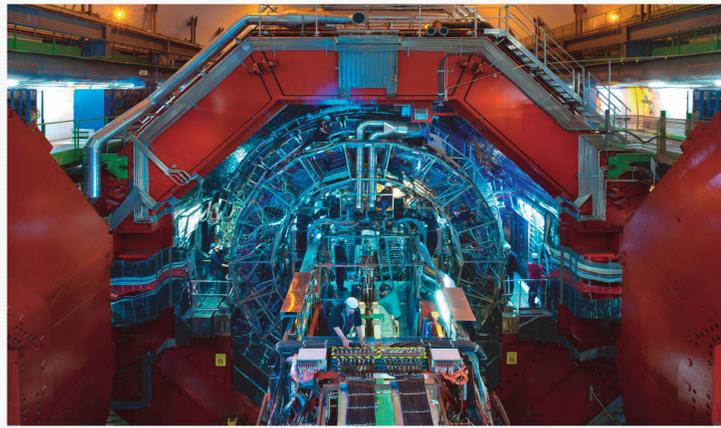


Dataflow
Collect
Process
Visualize



The goal of MAD is to provide the shift crew, the on-call experts and the Run Coordination with a clear and quasi real-time view of the **ALICE dataflow** status, thus allowing prompt reactions to abnormal situations.

MAD Data Collectors, developed in C++ and executed as daemon processes, are responsible for gathering the raw monitoring values of the base metrics. Once the raw values are collected, they are sent to the MonALISA Service as small UDP datagrams using the **ApMon** library.

Data Filters, developed in Java, run as MonALISA Service modules to aggregate base metrics into higher-level metrics.

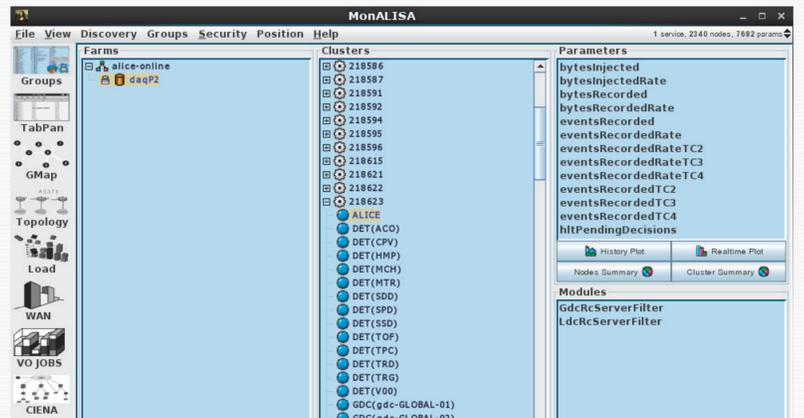
Monitoring values are transferred via ZeroMQ to the **ECS Status Display Server**, which handles client connections and registration. A local cache minimizes data traffic. The values are then pushed from the server to the clients in JSON format using the **WebSocket** protocol. Visualization is provided by modular and dynamic **Widgets** that can be combined to create thematic layouts.

ALICE shift crew, operating the experiment 24/7, have access to the layouts via **75" Full HD screens** available in the ALICE Run Control Center.

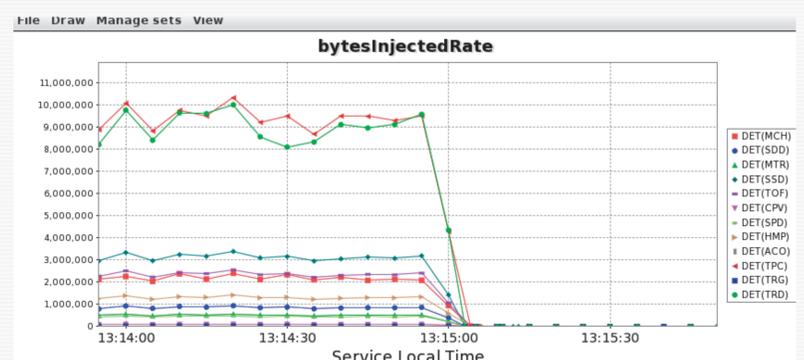
Future plans include the addition of extra metrics from several ALICE subsystems (DAQ, Trigger, Detectors), the deployment of a MonALISA Repository archive and the migration of the WebSockets server to Node.js.



ECS Status Display screenshot displaying information of an ongoing run.



MonALISA interactive client. Run numbers are displayed as Clusters, dataflow entities (detectors, LDCs, GDCs) as nodes.



MonALISA interactive client. Experts can access finer-grained plots such as per-detector readout rates to identify the root cause of an abnormal situation.