Upgrades for Offline Data Quality Monitoring at ATLAS

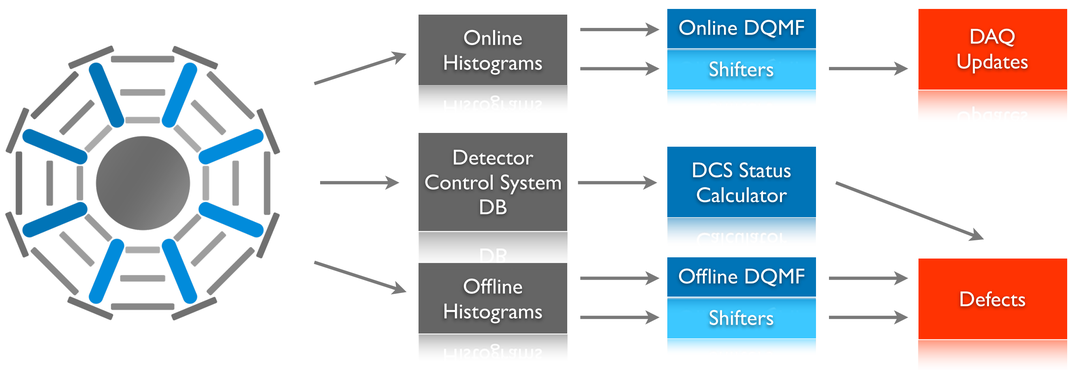
By **James Frost**, **Sami Kama**, **Jessica Leveque**, **Peter Onyisi**, **Anna Sfyrla** and **Joerg Stelzer**, presented by **Morten Dam Jørgensen** on behalf of the **ATLAS Collaboration**

Long Shutdown Plans

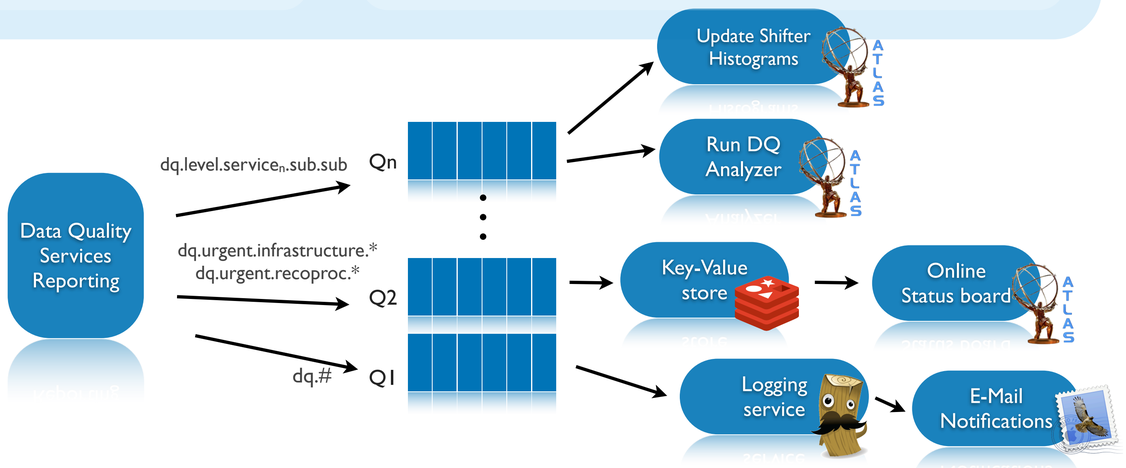
During the first major shutdown of the LHC the ATLAS offline Data Quality (DQ) team is preparing major upgrades in all areas of the data quality assurance chain

Main Upgrade Activities

- Optimization of monitoring histograms
- Automatic software validation systems
- Improved configuration of the data quality review chain
- Self-monitoring of critical infrastructure
- On-demand analysis tools (Good Run Lists)
- Improved code quality in reconstruction tools
- Introducing an offline messaging bus
- Upgrades of shifter tools adding interactivity



The data quality chain



Logging with messaging bus

Dynamic Monitoring Displays

Streamlined Data Quality Process

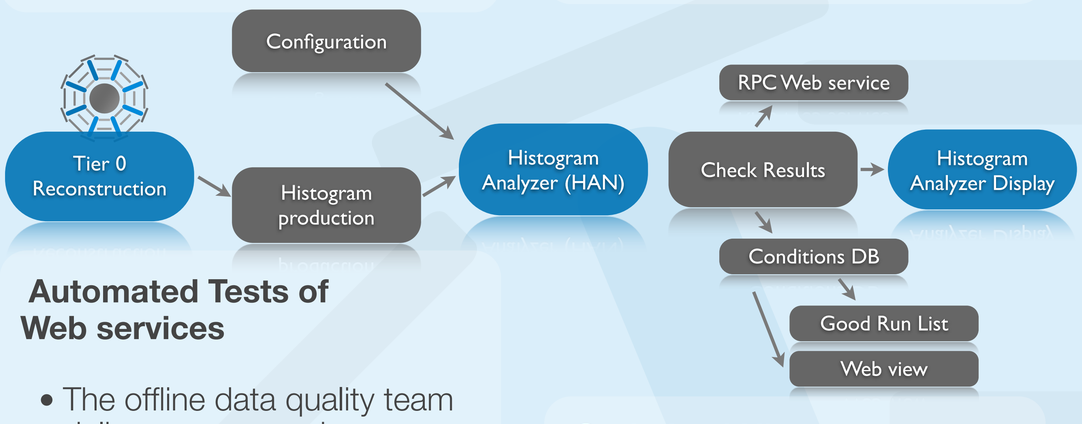
- The quality of events recorded by ATLAS is assured by a series of checks and validations, happening in real-time during data taking and offline after reconstruction.
- An integral part of the quality assurance is done by manual verification of histograms produced to show possible artifacts and deviations from nominal values.
- In the 18 month shutdown period we are heavily updating the definition, production and visualization of offline histograms to streamline the processes

Introducing Interactive Web Displays

- Allow shifters to zoom, fit and export data from monitoring histograms.

Updates to the Web Display Configuration Interface

- Introducing YAML for configuring benchmark histograms
- Automating uploading and validation of new configurations with web-interface
- Automating integration into running reconstruction code



Automated Tests of Web services

- The offline data quality team delivers many services through web interfaces. We are working on automated testing of these interfaces with tools such as Selenium

Optimized Histogram Production

- Monitoring histograms need to scale with higher luminosities
- We are working on a way to handle comparison of many runs without refilling millions of histograms

Bus-based Messaging between Data Quality Services

Connecting the Threads

- Data Quality provides multiple services and relies on even more
- Currently communication between them relies on scraping, direct messaging and manual intervention
- To strengthen the infrastructure we are introducing a messaging bus that allows services to communicate in a flexible way

RabbitMQ

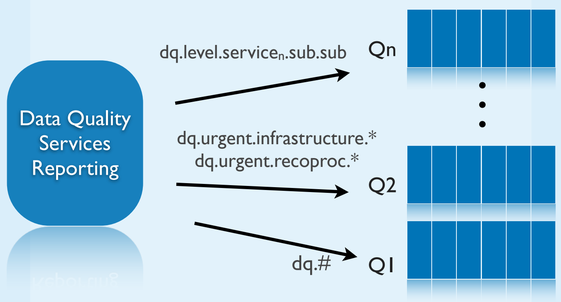
- We use the Erlang based RabbitMQ as a messaging server, due to flexibility and ease of use

Use-cases

- Broadcasting
- Logging
- Heartbeat signals
- Loosely coupled dependencies, failure at a source doesn't kill the whole chain
- Asynchronous communication
- Multiple listeners to single services

Logging and Notifications

- All services sends log notifications to a common exchange
- Notifications delegated by the exchange into queues based on routing keys formatted like a namespace, with overall domain, reporting level and service flags. In the figure below */# marks wildcards



- A log archive is implemented by subscribing a log database to a "catch-all" queue

- E-mail notifications can be handled by querying the logging service or directly by subscribing to a queue

- Online status board with Redis key-value server as state keeper