DUNE Far Detector DAQ Back End Dataflow

Kurt Biery

DUNE FarDet DAQ Conceptual Design Review

03 December 2018

Introduction

The DAQ Back End Dataflow elements include:

- The data transfer from the FELIX hosts to the Event Builders
- The building of timeslice- and geographically-complete bundles of data (‘events’) [dynamic definition of ‘complete’]
- Possible filtering, compression, zero suppression, or other manipulation of the data – High-Level Filter farm
- Real-time data quality monitoring
- Storage of the data before transfer to Fermilab
- Coordination with offline computing (e.g. data transfers)

Requirements

The DAQ shall provide capability for triggering on events of interest in order to limit the total bandwidth for events stored on tape. (< 30 PB/year)

- Affects the data rates and storage that need to be provided

The DAQ architecture shall provide a mechanism for triggering on galactic supernova bursts and recording neutrino interactions associated with those bursts over a 10 second period.

- Affects how data fragments are gathered and written (separate streams?)

The DAQ shall merge data into a form suitable for offline analysis. Furthermore, tags shall be provided to allow the data collection conditions at the time and the livetime to be determined.

- Affects how data is written and identified

The DAQ shall operate with deadtime that does not contribute significantly to overall loss of detector livetime.

- Affects how the system is designed

The detector shall operate stably at least 90% of the time in order to collect the required physics data.

- Affects how the system is designed

Internal Requirements

Support the overall DAQ requirements for livetime and uptime

Accept Trigger Commands and read out the data for the specified time windows and detector components

Provide the infrastructure for the High-Level Filter

Provide sufficient DAQ-specific data storage to avoid any disruption to data taking from a temporary loss of network connectivity to Fermilab

Provide ‘DAQ Kit’ software throughout the next 6+ years

Interfaces

Internal

- Data buffers on the FELIX hosts
- Control, configuration, and monitoring
- Trigger

External

- Offline data and computing models
- Analysis and DQM software
- WAN and offline
- Operating system(s) and third-party software
- Physical environment

Technical Constraints

Provide basic functionality for test environments from 2019 thru Production phase.

Provide enhanced functionality to meet the needs of the production system, especially very high uptime

- Fault tolerance
- Dynamic reconfiguration
- Automated error recover
- Automated running of calibrations, either as part of normal data taking or in parallel

Key Challenges

Provide basic functionality for test environments from 2019 thru Production phase.

Provide enhanced functionality to meet the needs of the production system, especially very high uptime

- Fault tolerance
- Dynamic reconfiguration
- Automated error recover
- Automated running of calibrations, either as part of normal data taking or in parallel

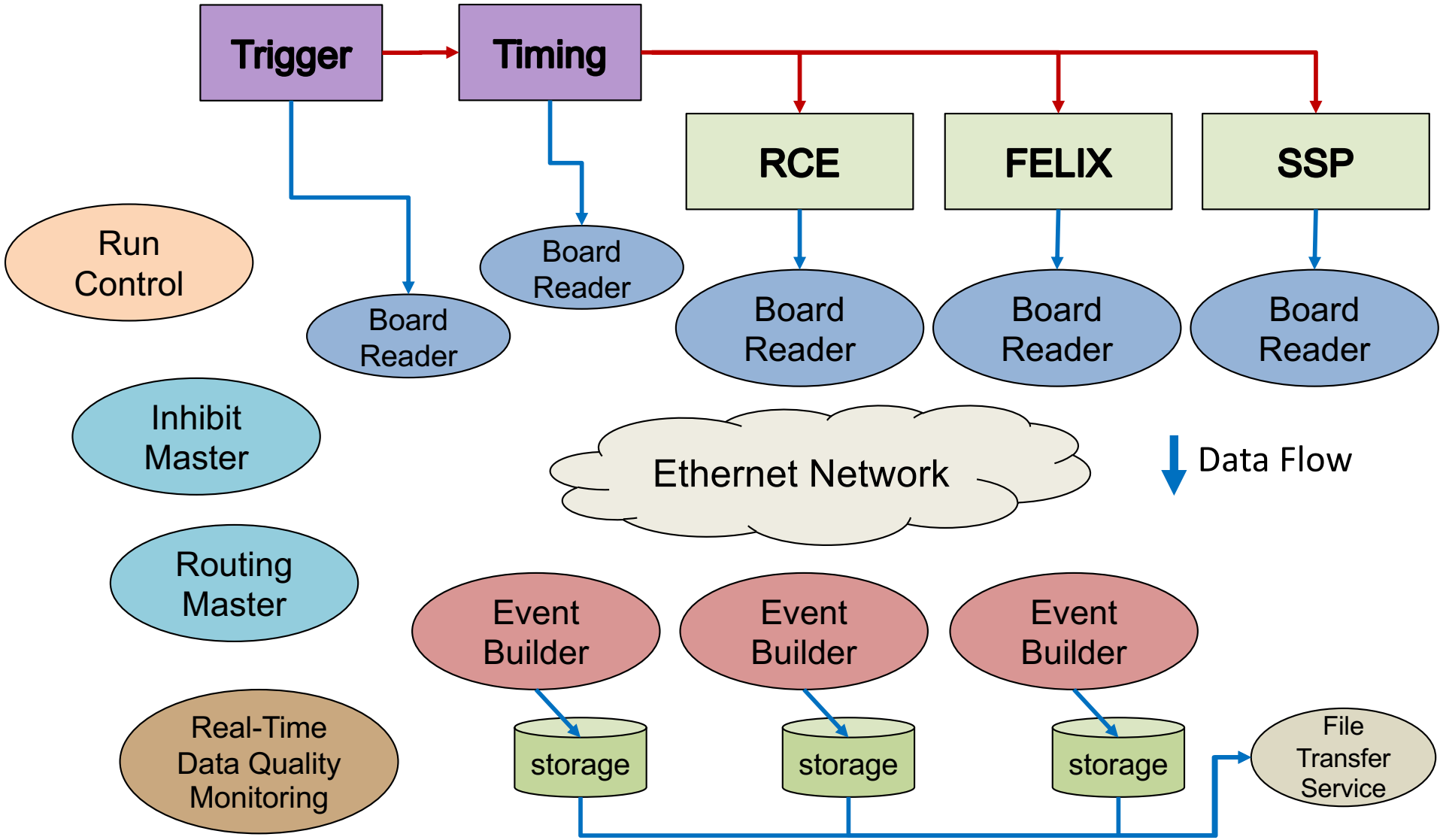
System Design

Use of the *artdaq* DAQ toolkit, with needed enhancements, to provide data transfer, event building, infrastructure for the DQM algorithms, deliver of data to DQM processes, and data logging.

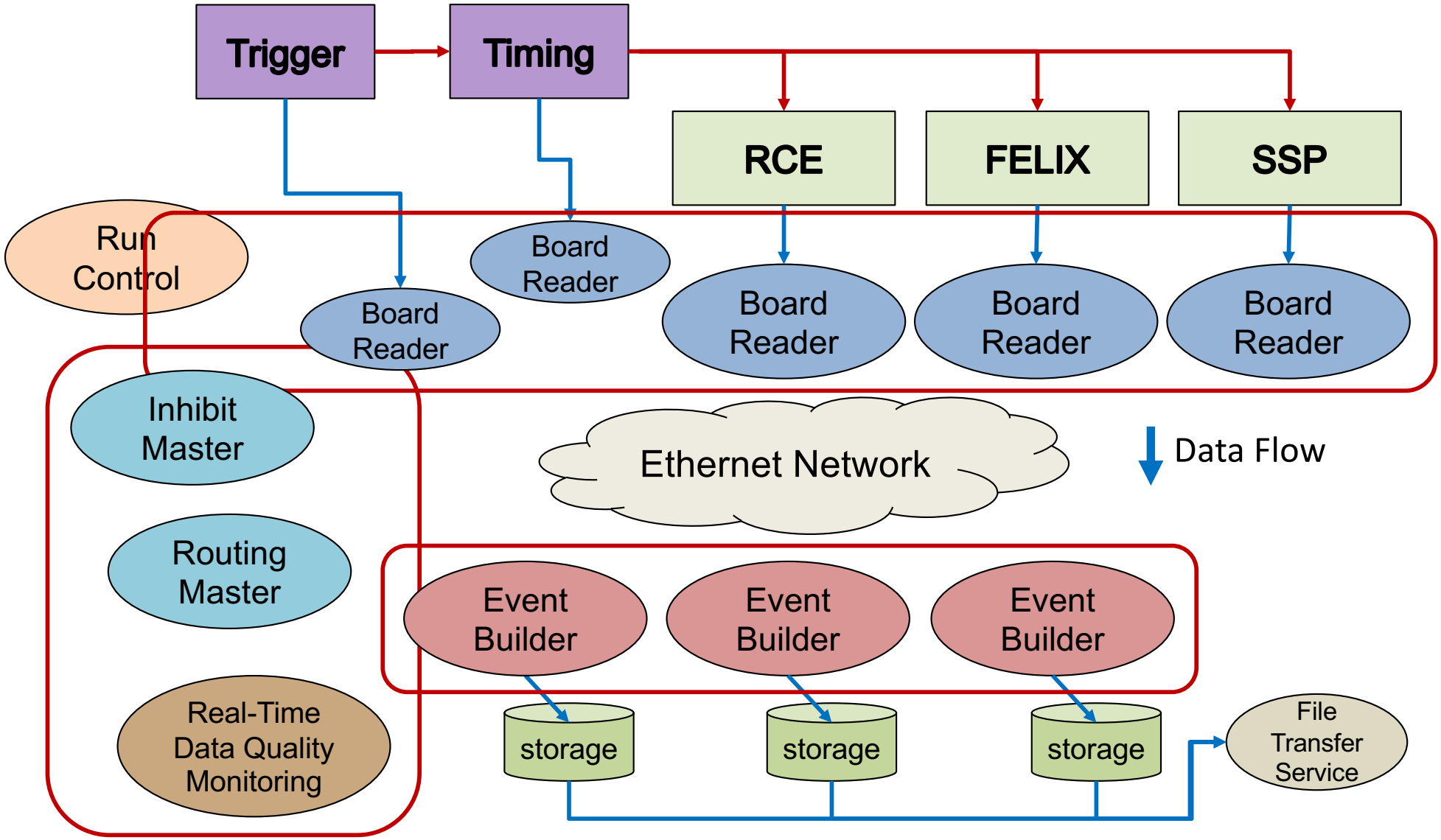
Extension of the data processing functionality in *artdaq* to support the High-Level Filter processing step.

Evolution of the components used at protoDUNE for teststands, etc.

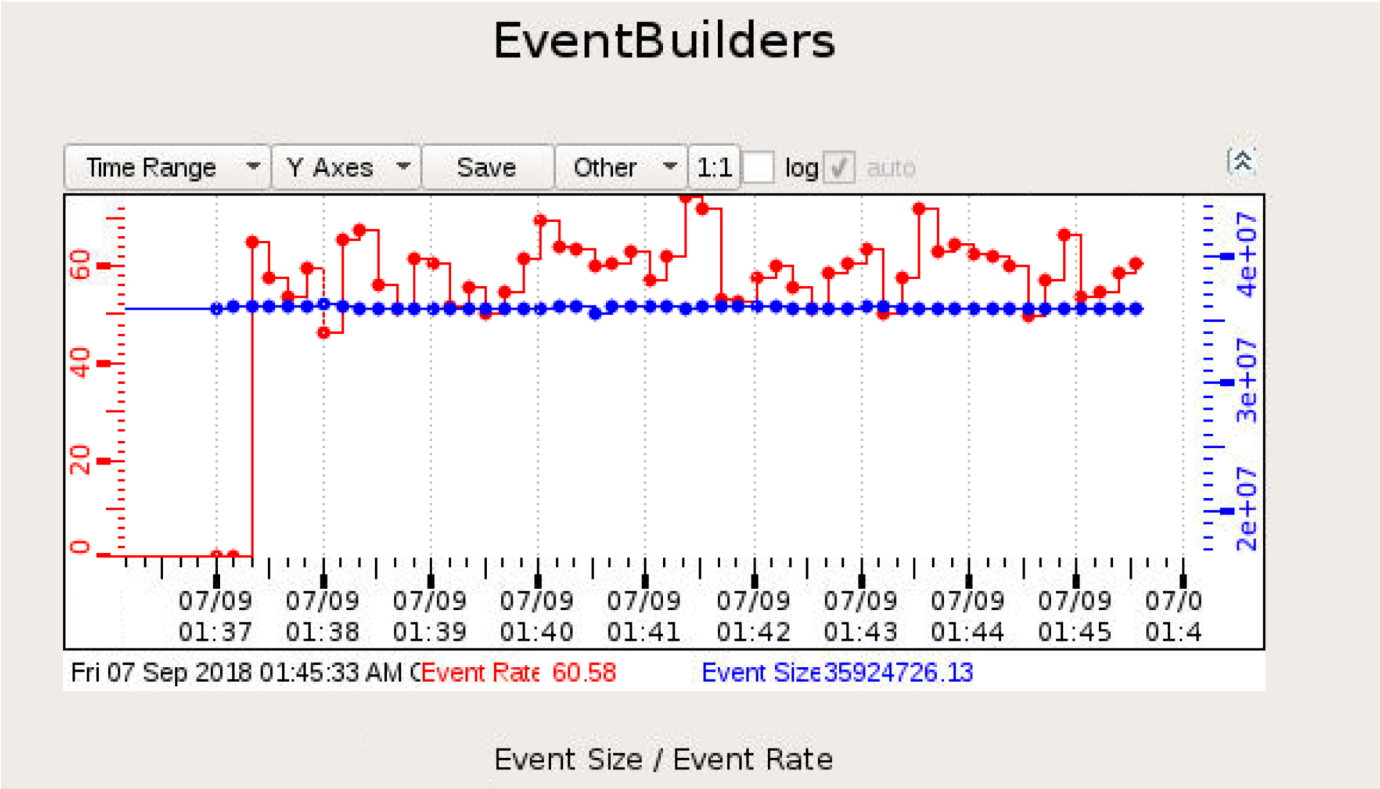
Simplified PDSP DAQ Diagram



artdaq Components in PDSP DAQ

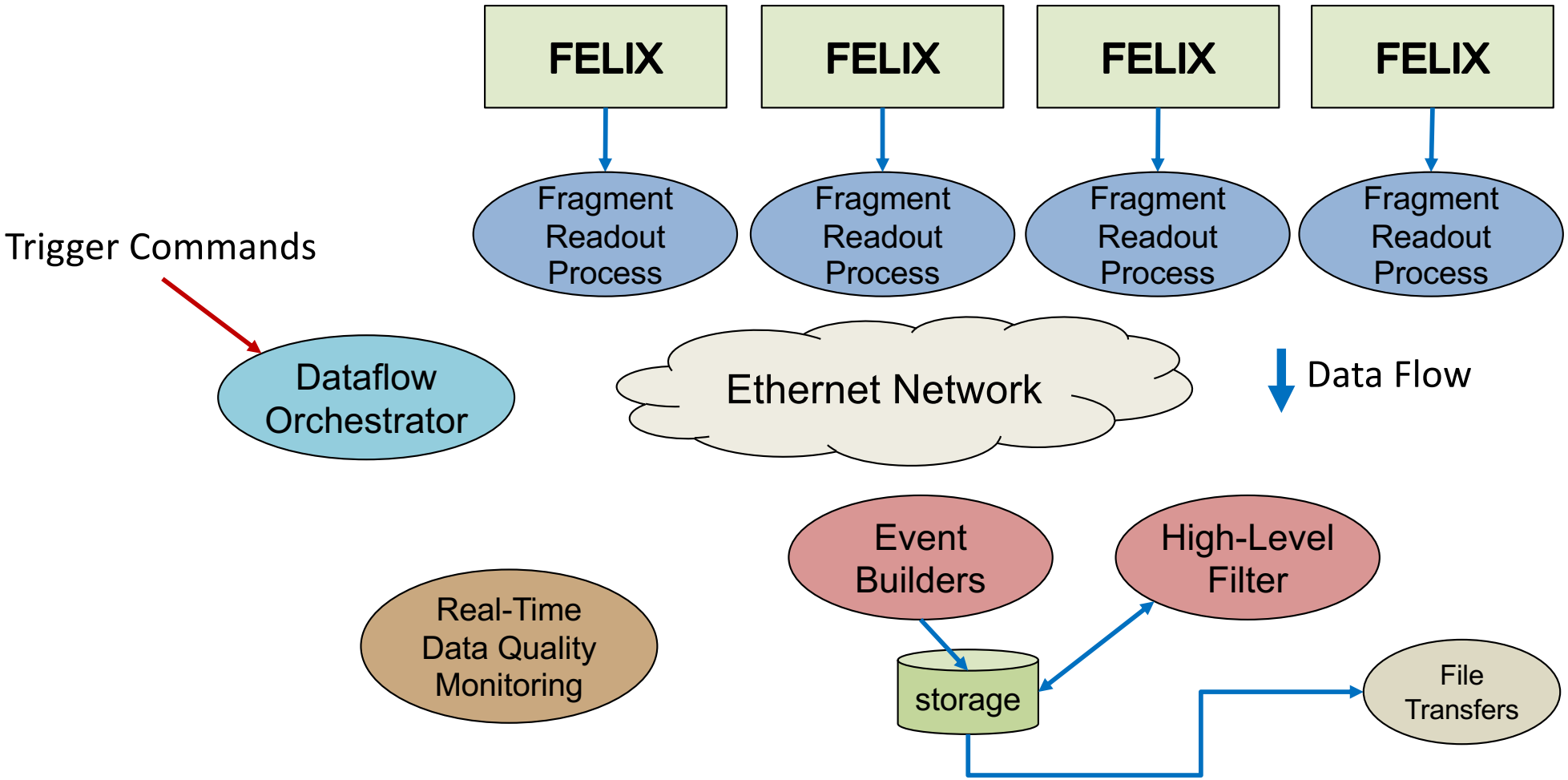


Successful Operation at protoDUNE SP



Already incorporating lessons learned

Simplified Back End Dataflow Diagram



Development Plan

Two thrusts:

1. Development and support for VST and integration test environments, data taking at protoDUNE, ‘DAQ Kits’
2. Development and validation of required new features

Plan is to take advantage of this situation:

- Evolutionary changes to the working Back End Dataflow software/system at protoDUNE
- This will provide the components for the DAQ Kits as well as provide the test environments for new Back End Dataflow features
- Incremental changes within the Back End DAQ Dataflow will reduce integration issues

Development Plan

Short-term plans:

- Design discussions at next level of detail
  - For example, incorporating the conclusions reached by the Data Model Task Force into the Back End Dataflow design
- Make improvements to protoDUNE DAQ based on lessons learned
- Work on features needed for DAQ Kits (reliability, usability, visibility to sources of errors, etc.)

Development Plan

Longer-term plans:

- Dynamic reconfiguration
- Fault tolerance
- High reliability
- Extensive operational monitoring
- Auto-recovery mechanisms