H2020 MSCA RISE 2018
GA 822185

Trigger-DAQ and slow control systems in the Mu2e experiment

Antonio Gioiosa

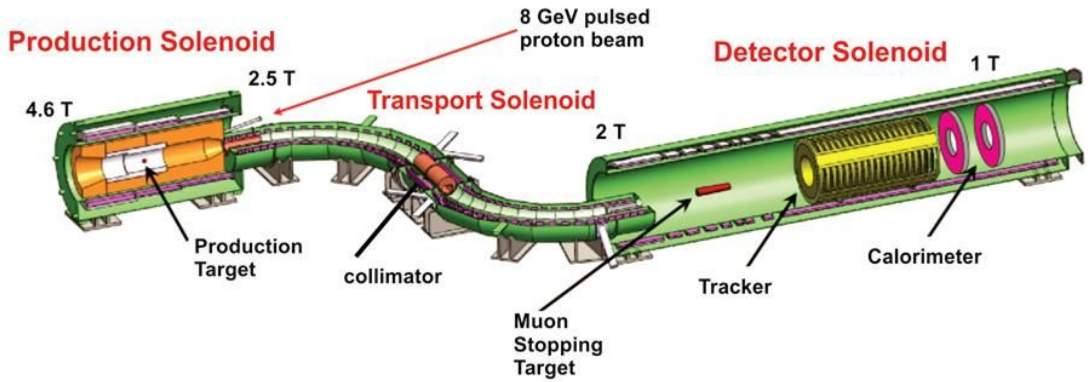
University of Molise & NFN Pisa

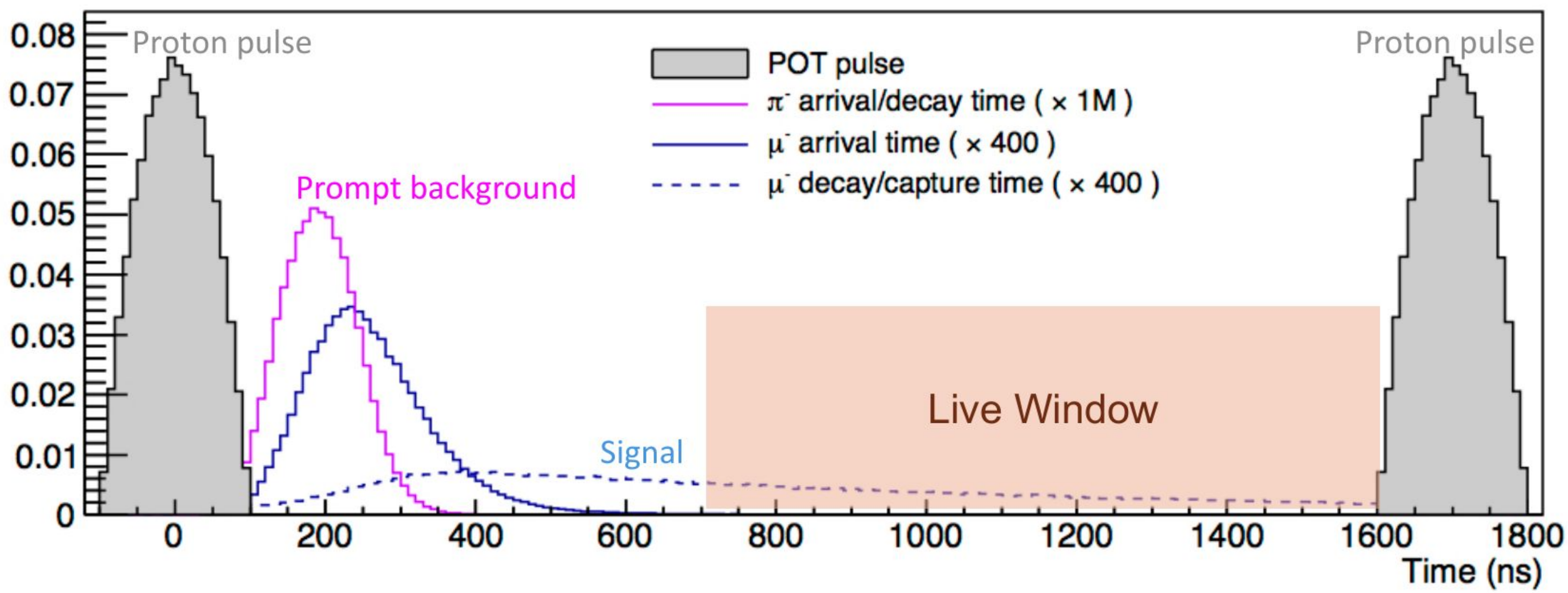
Intense 2022 - MidTerm Review meeting

November 28, 2022

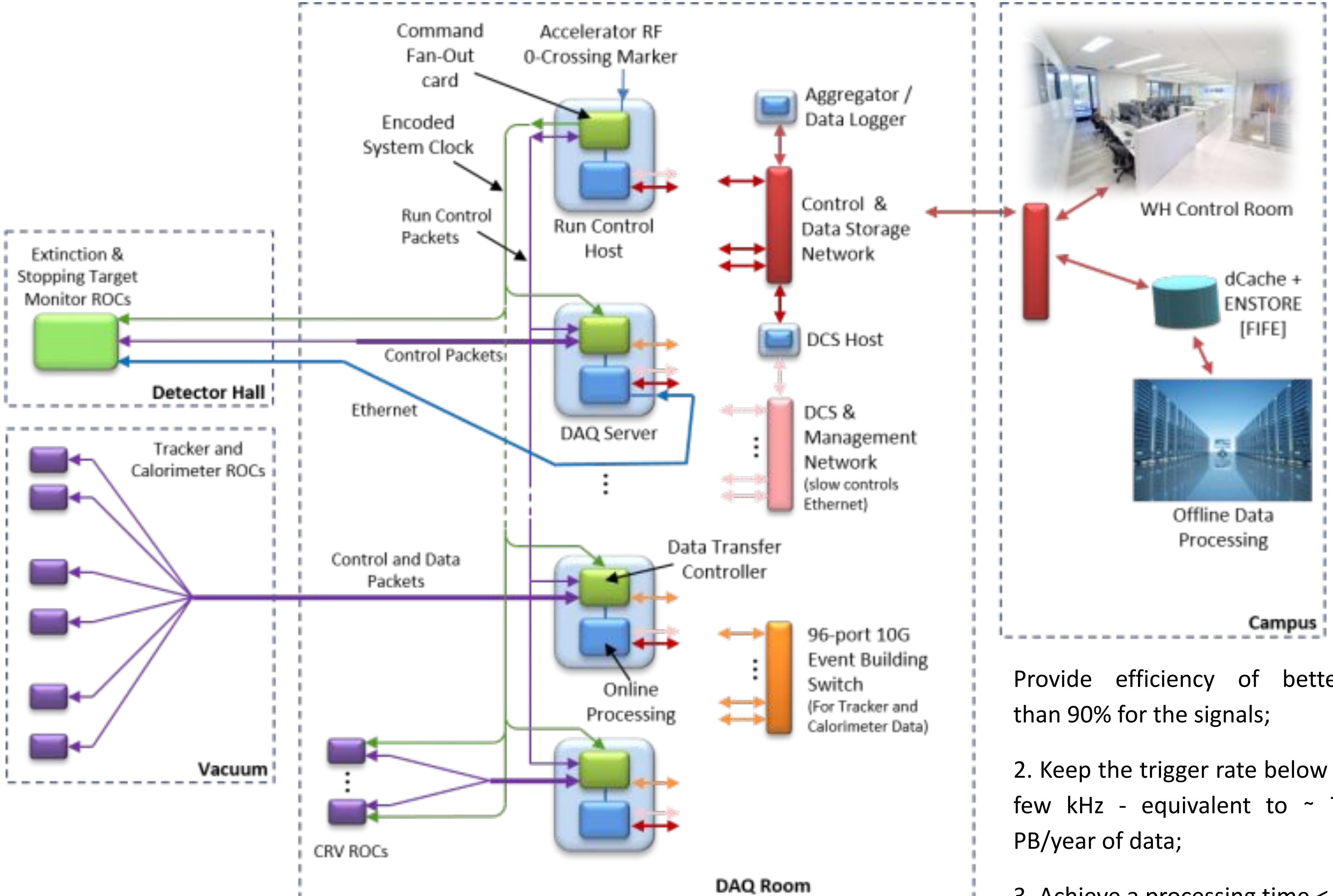
The Mu2e Experiment at Fermilab

The signal we are looking for is a delayed monoenergetic electron with an energy of just under 105 MeV (muon mass)





Mu2e TDAQ components Diagram

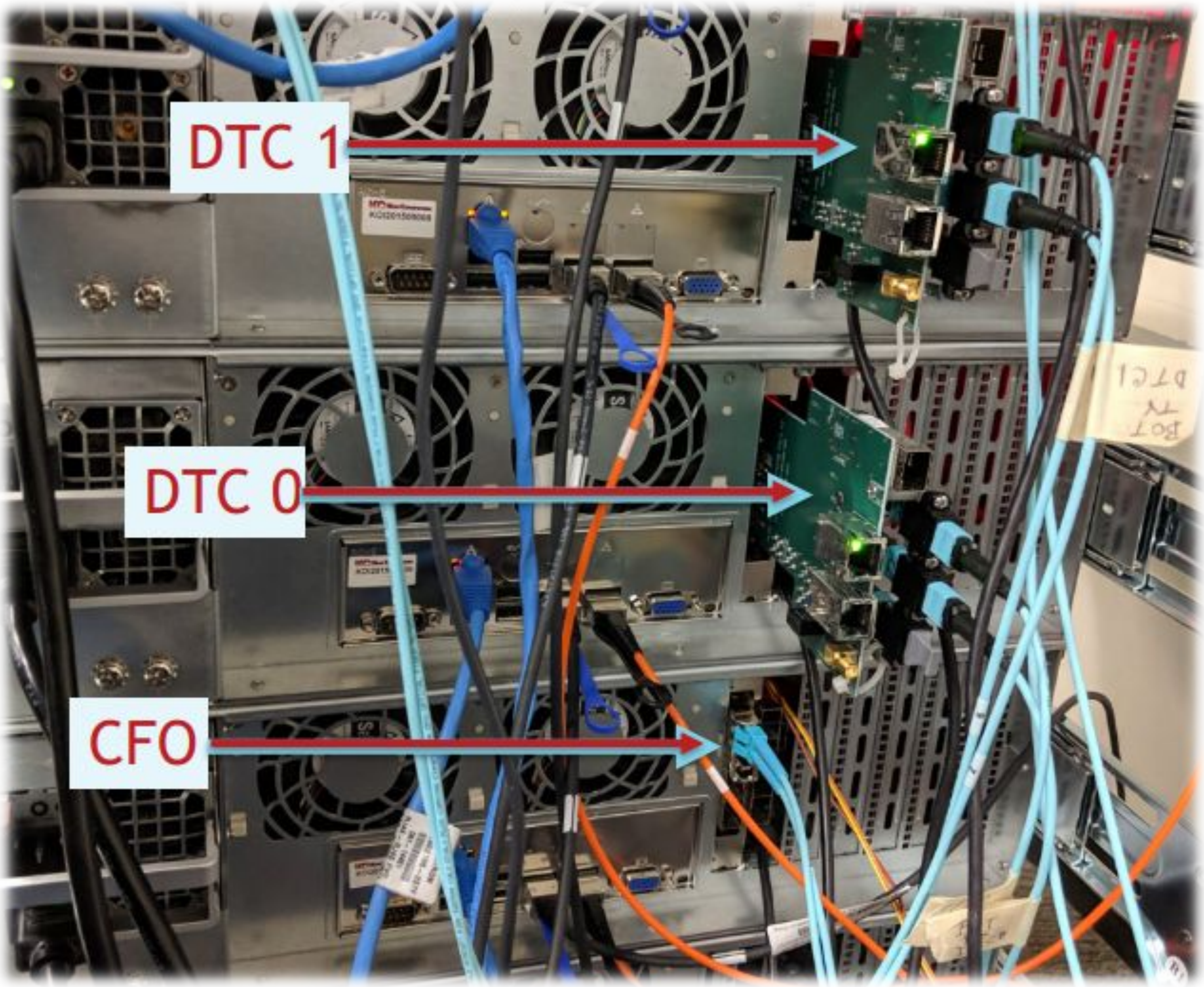


Provide efficiency of better than 90% for the signals;

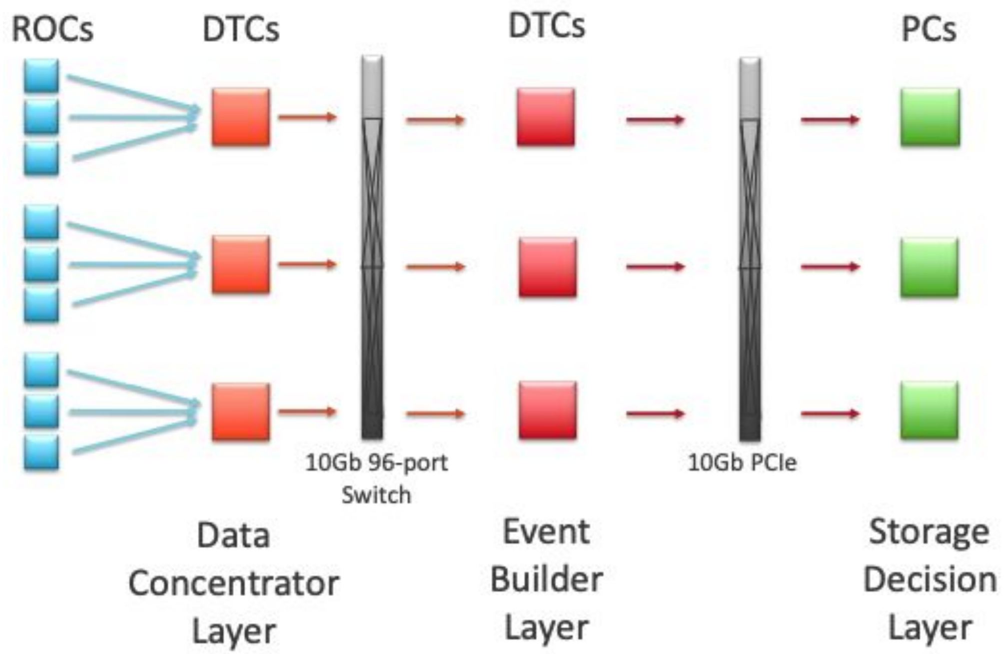
2. Keep the trigger rate below a few kHz - equivalent to ~ 7 PB/year of data;

3. Achieve a processing time < 5 ms/event.

Test Stand -> Production DAQ Room



TDAQ Readout scheme



- 396 ROCs 69 DTCs (Kintex-7) for data readout and event building
- Large front end buffers to average over long off-spill time
- 800 threads on 40 nodes for HLT → ~5 ms per event
- ~40 GB/s data read out to storage decision layer, ~280 MB/s written to disk

High Level Trigger Software

Mu2e Online DAQ solution: *otsdaq*

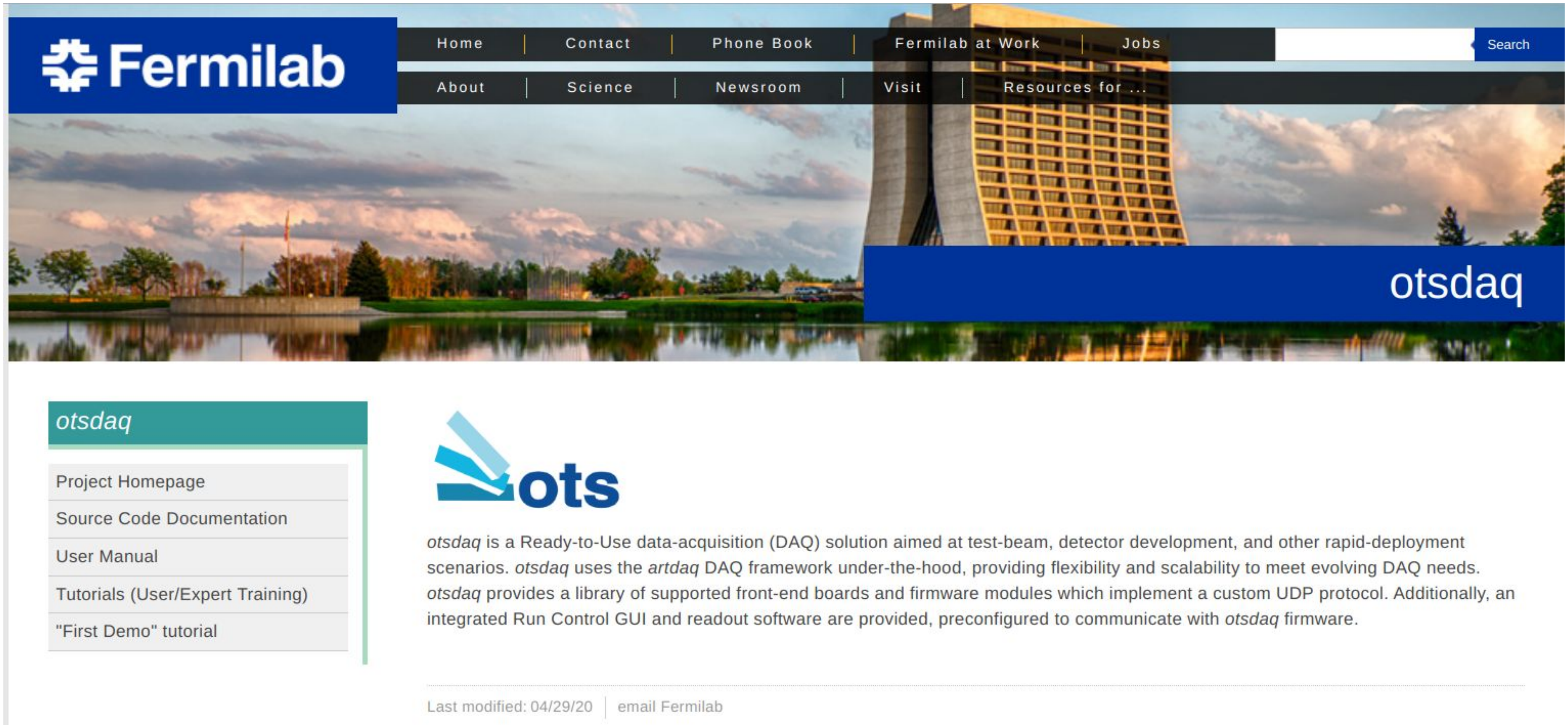
otsdaq overview

Acronym for "off-the-shelf data acquisition."

- ***otsdaq*** is a Ready-to-Use data-acquisition (DAQ) solution aimed at test-beam, detector development, and other rapid-deployment scenarios
- it uses the ***artdaq*** DAQ framework under-the-hood, providing flexibility and scalability to meet evolving DAQ needs
- ***otsdaq*** provides a library of supported front-end boards and firmware modules which implement a custom UDP protocol
- Developments are in two directions: **server** side and **web** side.
- An integrated Run Control GUI and readout software are provided, preconfigured to communicate with ***otsdaq*** firmware

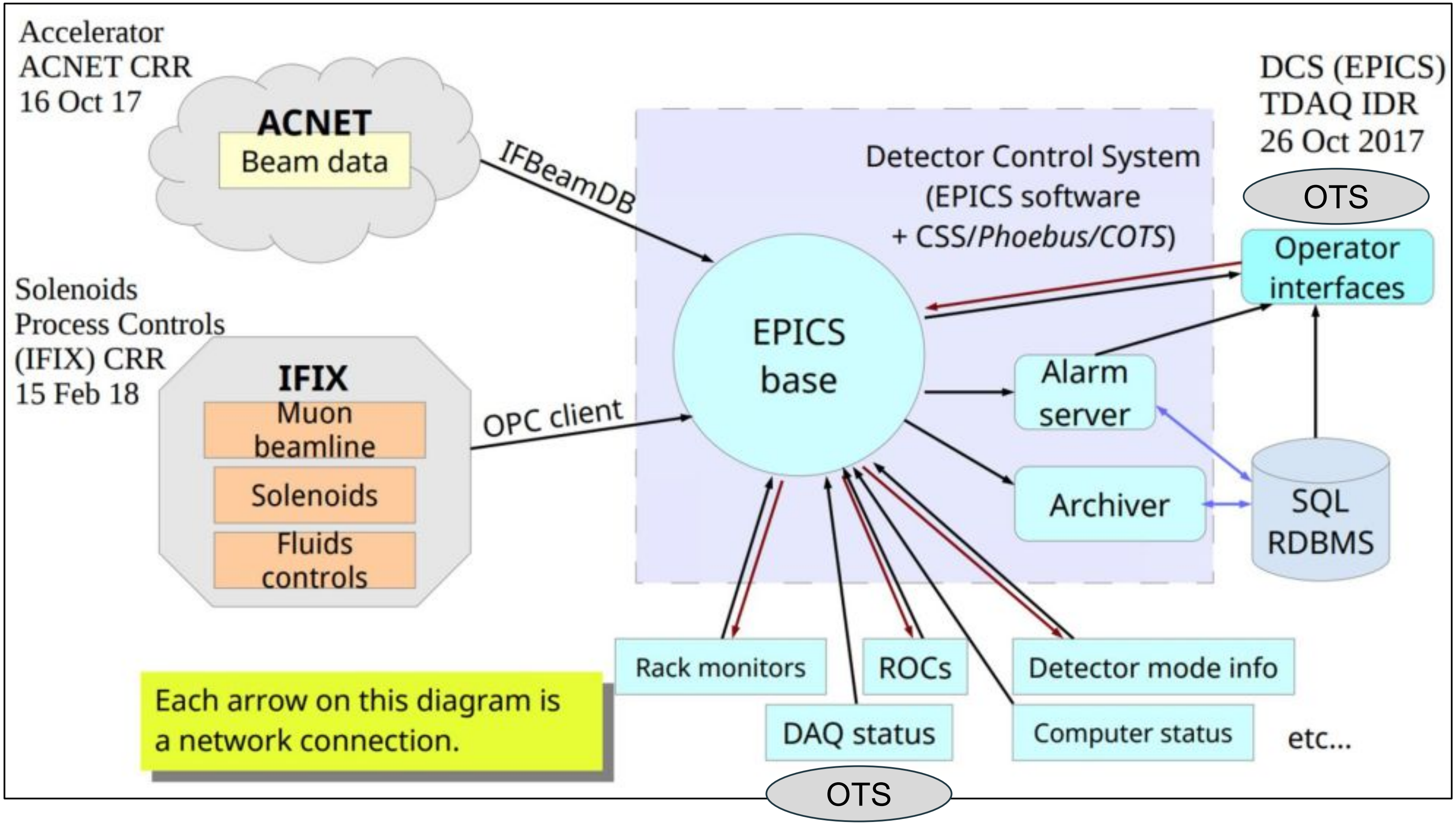
otsdaq overview

More info at ***otsdaq*** web page https://otsdaq.fnal.gov/



Slow Controls connection and EPICS plugin development in otsdaq

Experimental Physics and Industrial Control System



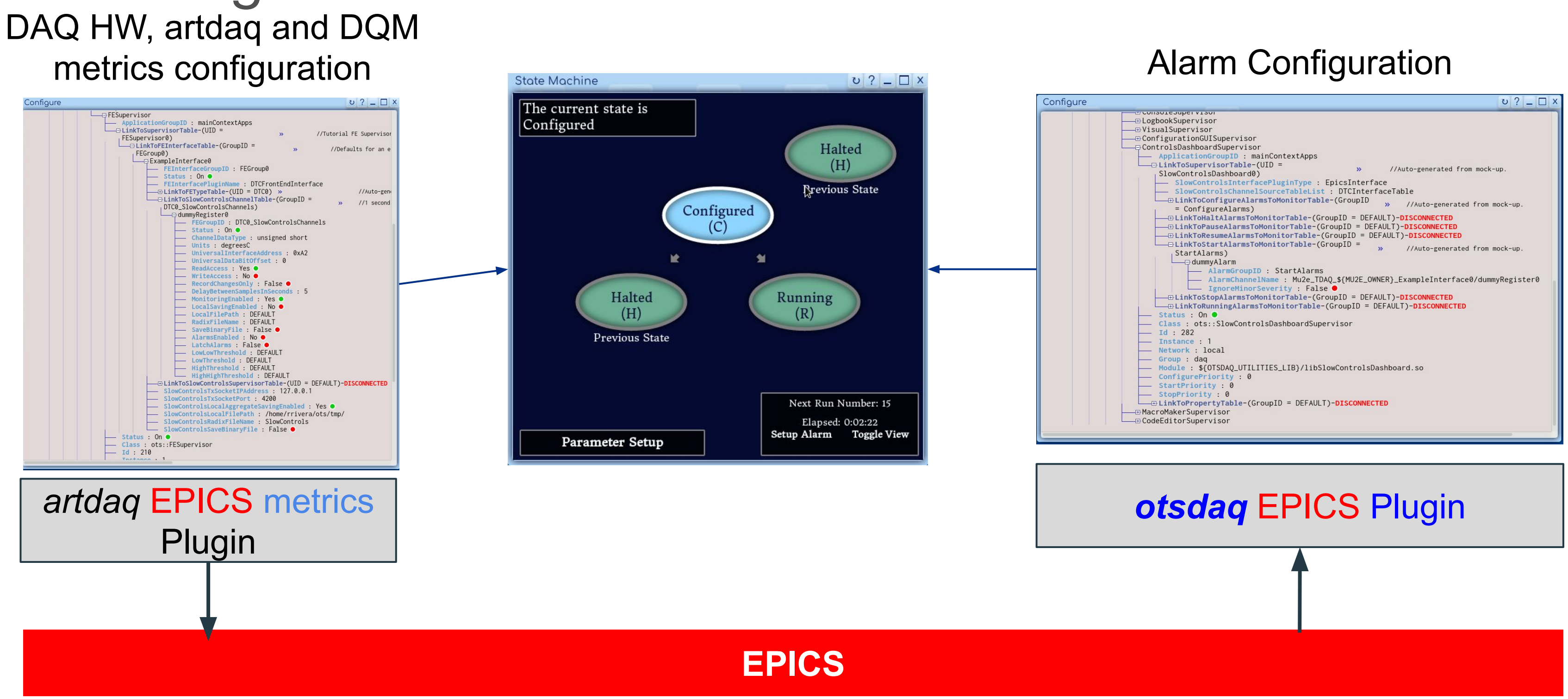
Slow Controls connection and EPICS plugin development in otsdaq

Channel subscription to EPICS uses Input Output Controller (IOC)

- integration of slow control in the online daq uses the same Interface plugin for:
  a. Monitoring of all mu2e slow control channels
  b. Sending Process Variables (PVs) of DAQ hardware info as EPICS channels and PVs settings into EPICS databases
- The Interface plugin:
  a. Performs channel subscription to EPICS using Channel Access EPICS C++ libraries to send and retrieve slow control data information like: Value, Alarm (Status, Severity), Settings
  b. Uses Postgres database C++ libraries to set channels and retrieve channels and alarms histories from EPICS databases

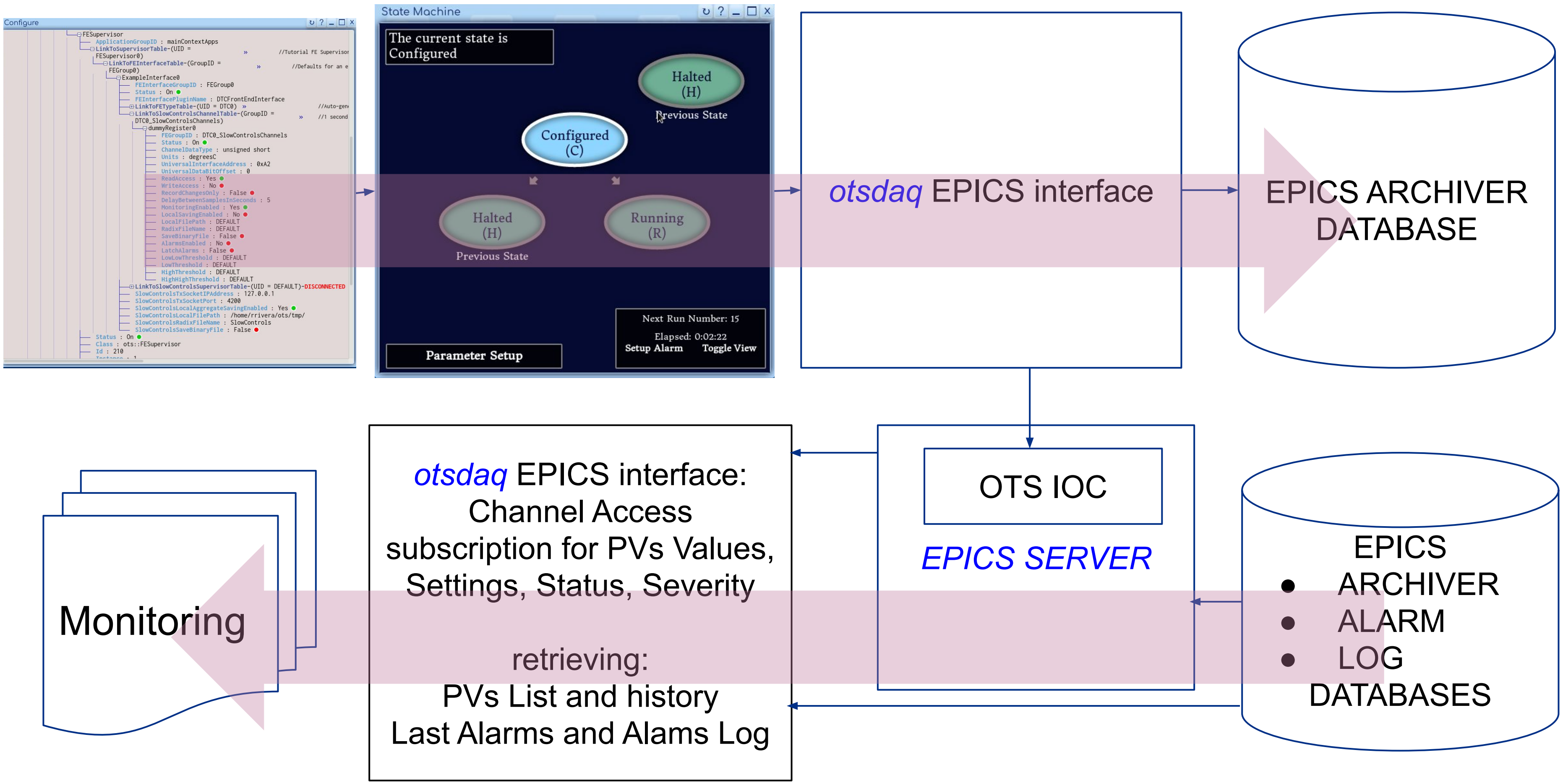
Integration with State Machine

- State Machine Configuration and data subscription to EPICS
- Alarm propagation (from EPICS) and *otsdaq* State Machine handling



Integration with State Machine

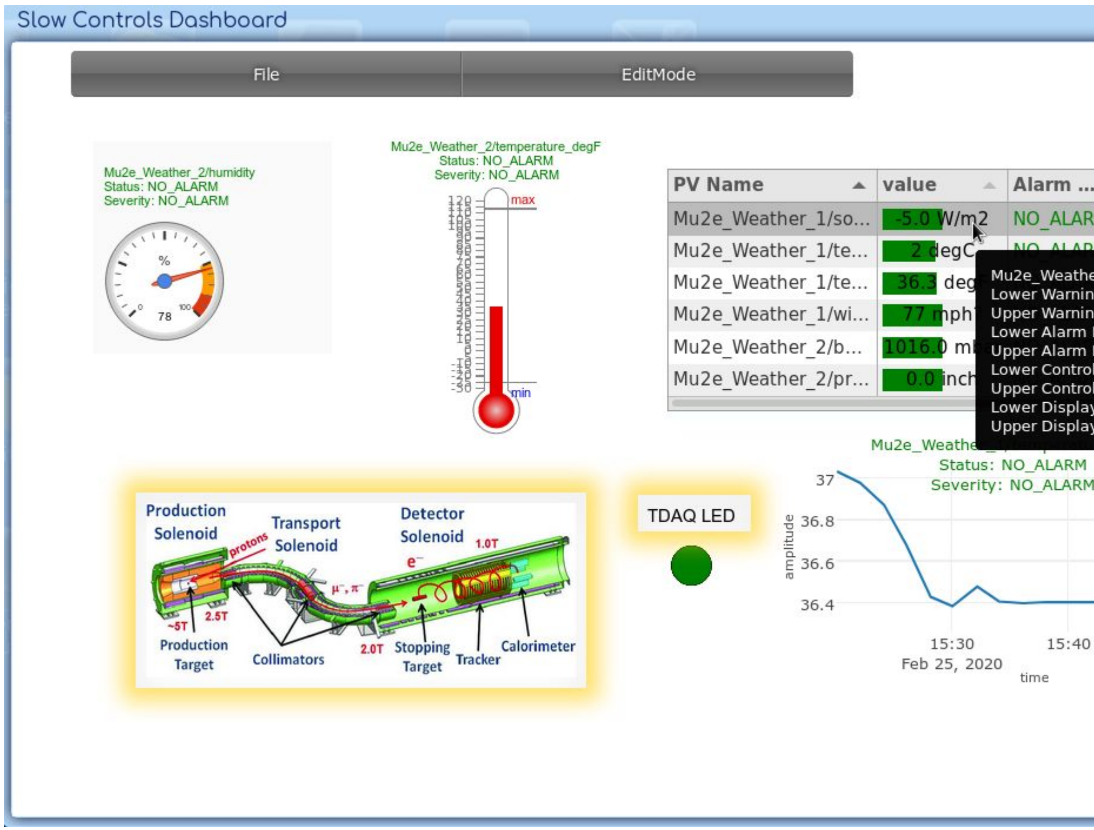
- *otsdaq FE (DTC/ROC/CFO) / artdaq metric new channel or new slow control setting → configuring State Machine → EPICS DBs and IOC* configuration
- *otsdaq Interface→ otsdaq* CA subscription and DBs select → Monitoring



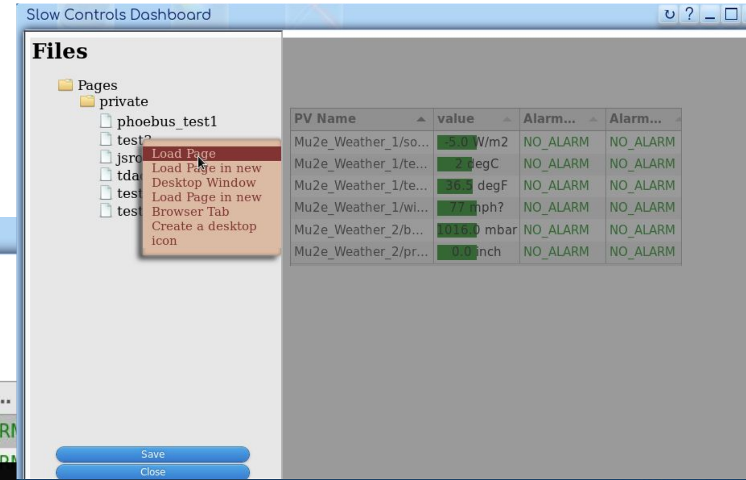
Slow Controls Monitoring GUI in otsdaq

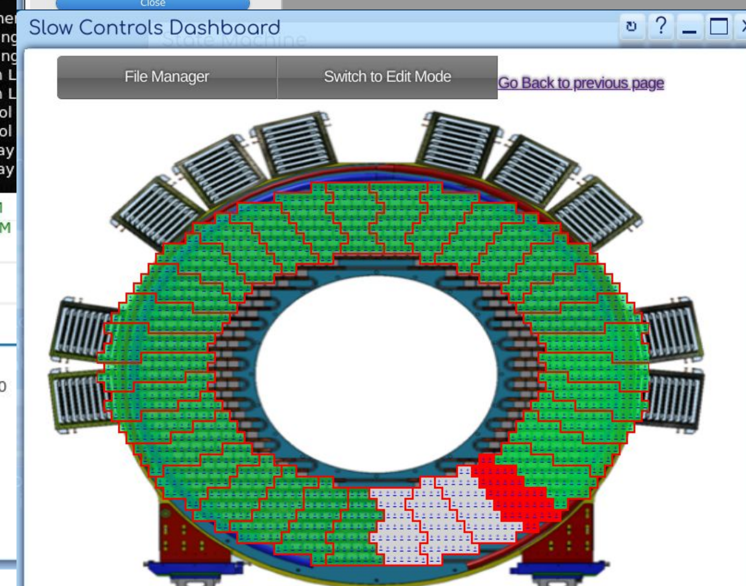
Examples

Example of loaded page



Example of page loading





Calorimeter monitor in the slow control GUI

Conclusions

- Mu2e Experiment is under construction at Fermilab and will be ready for data taking in ~ two years
- Mu2e TDAQ and slow control are in large part developed according to the requirements (200K events/s for data taking) and the installation in the daq room is going on
- Slow control integration in the online DAQ system, *otsdaq*, provides an advanced slow controls monitoring, an interface to send ***otsdaq*** front-end DAQ hardware, data processing, and DQM slow controls information to **EPICS**, and a real configuration and Integration with the *otsdaq* State Machine

Backup slides

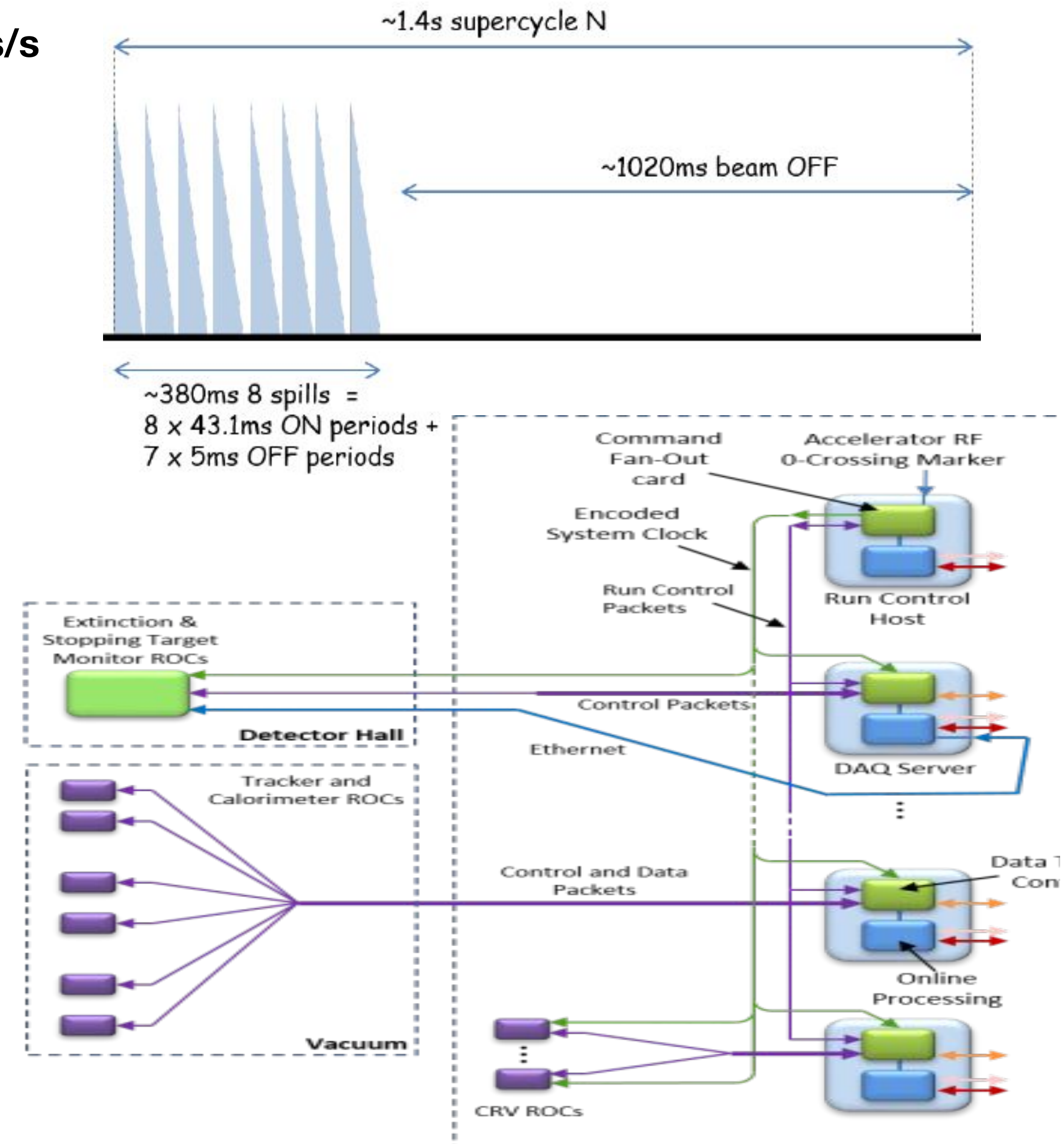
Mu2e Timing Distribution

Requirement is to process 200K events/s

- Mu2e Runs are broken up into contiguous Event Windows
- Experiment defined Run Plan is coordinated by the Command Fan-Out Card (CFO)
- The System Clock (40MHz) and Event Window markers originate at the CFO ...and are distributed to ROCs:

1. CFO distributes System Clock and Event Windows to DTCs with fixed latency
2. DTCs distribute System Clock and Event Windows to ROCs with fixed latency
3. ROCs respond to Data Requests

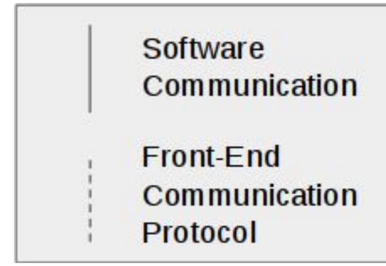
DTC⇆ROC Heartbeat packet (16 bytes) to specify the detail of each Event Window

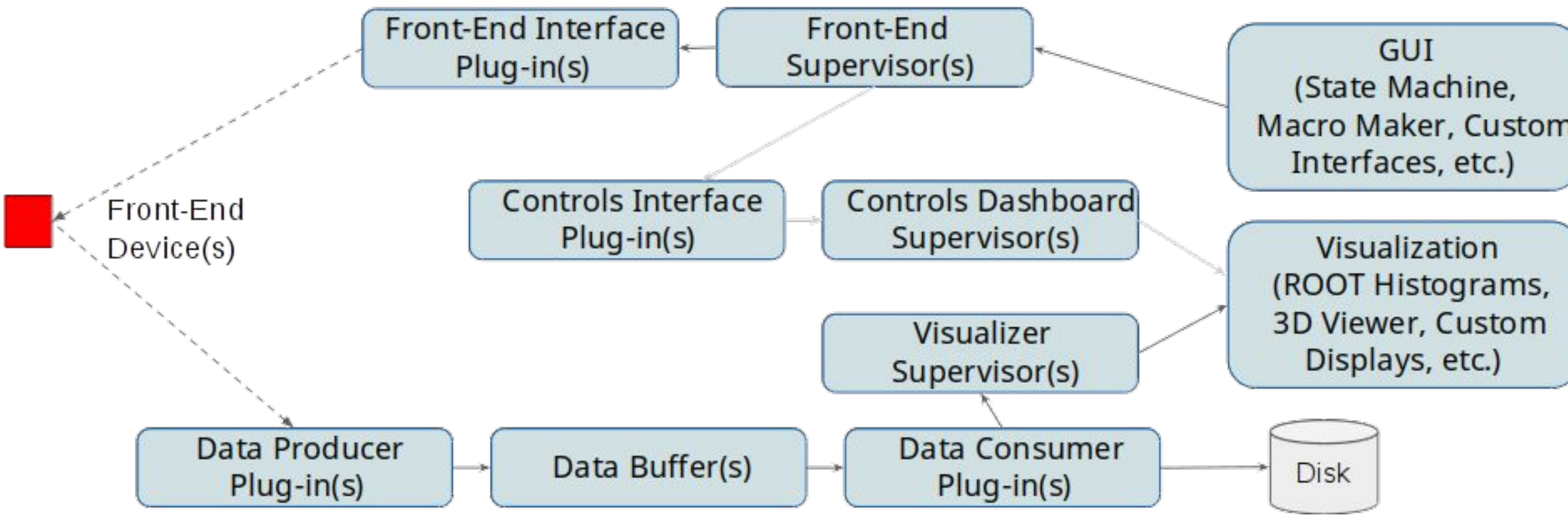


otsdaq overview

Data Flow Block Diagram

Server side is C++. User code is added through plugins (C++ classes inheriting from the appropriate class)





Web side is HTML and JavaScript. User code is added in the form of web-apps through .html files (including the appropriate .js and .css files)

Slow Controls Monitoring in otsdaq

Slow Controls GUI Hierarchy