



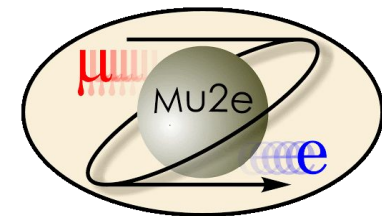
# *Trigger-DAQ and slow control systems in the Mu2e experiment*

Antonio Gioiosa

*Università di Pisa, INFN Pisa*

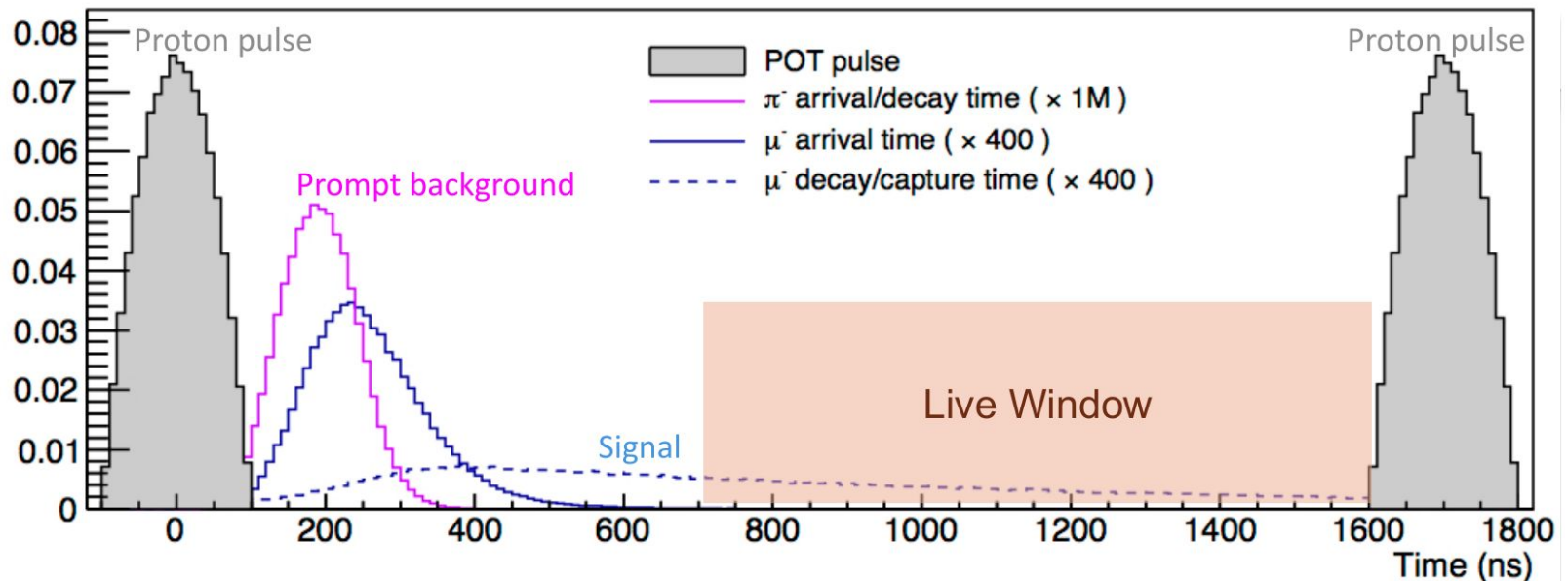
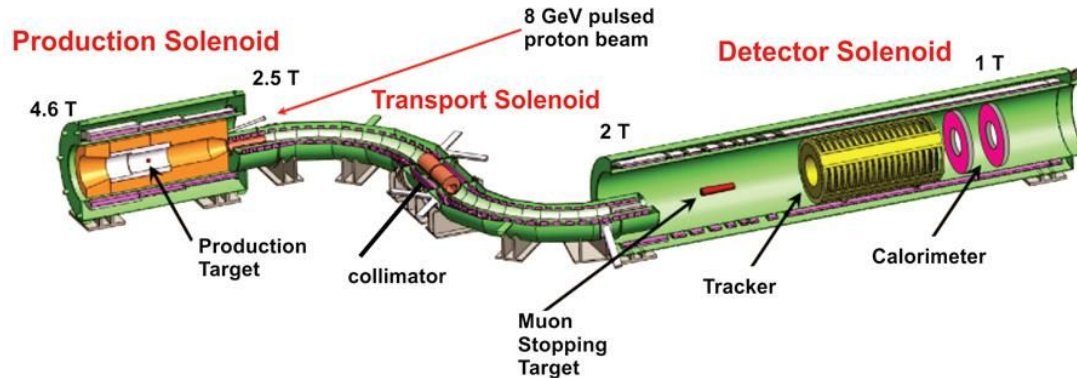
CALOR 2022

May 19, 2021

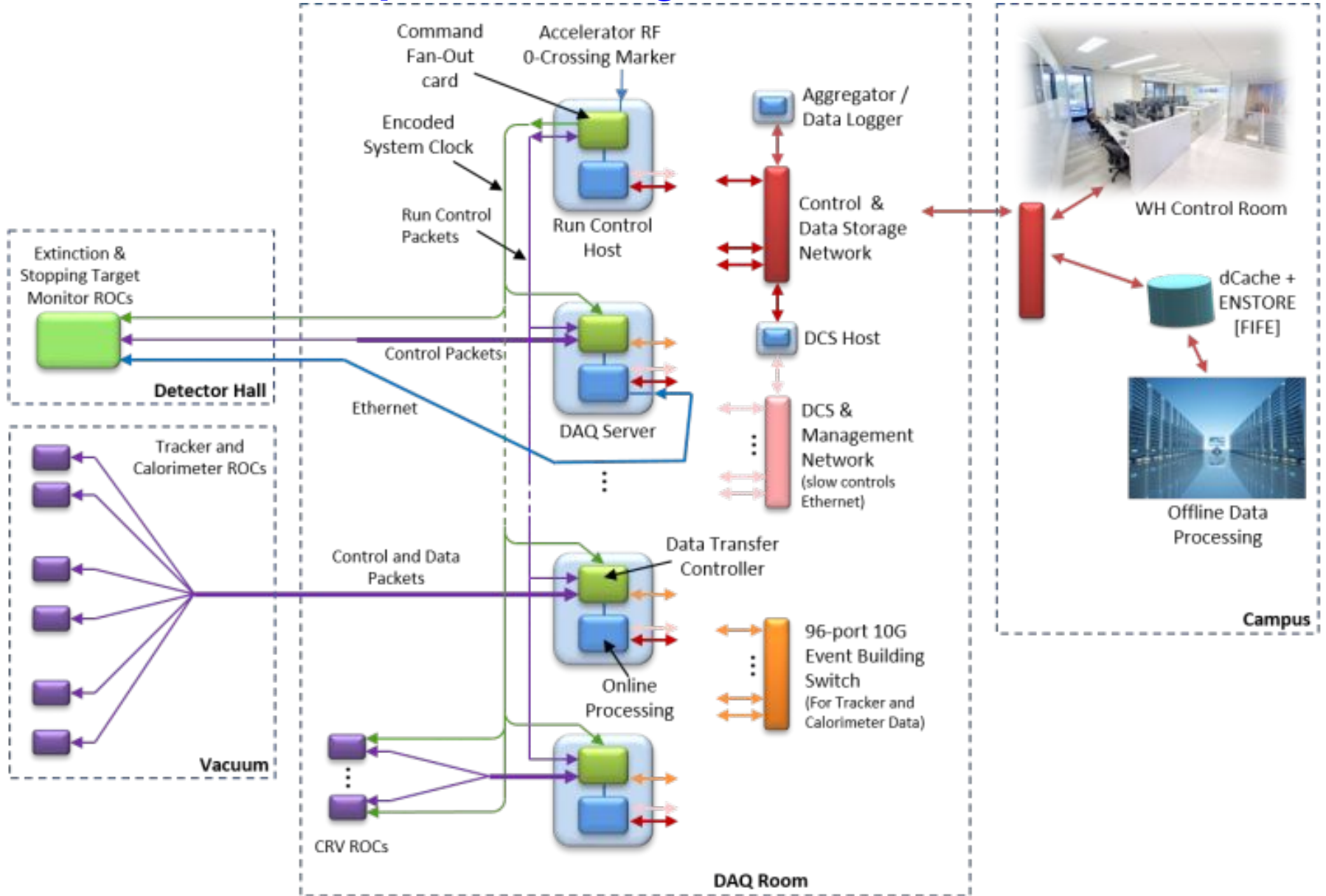


# The Mu2e Experiment at Fermilab

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

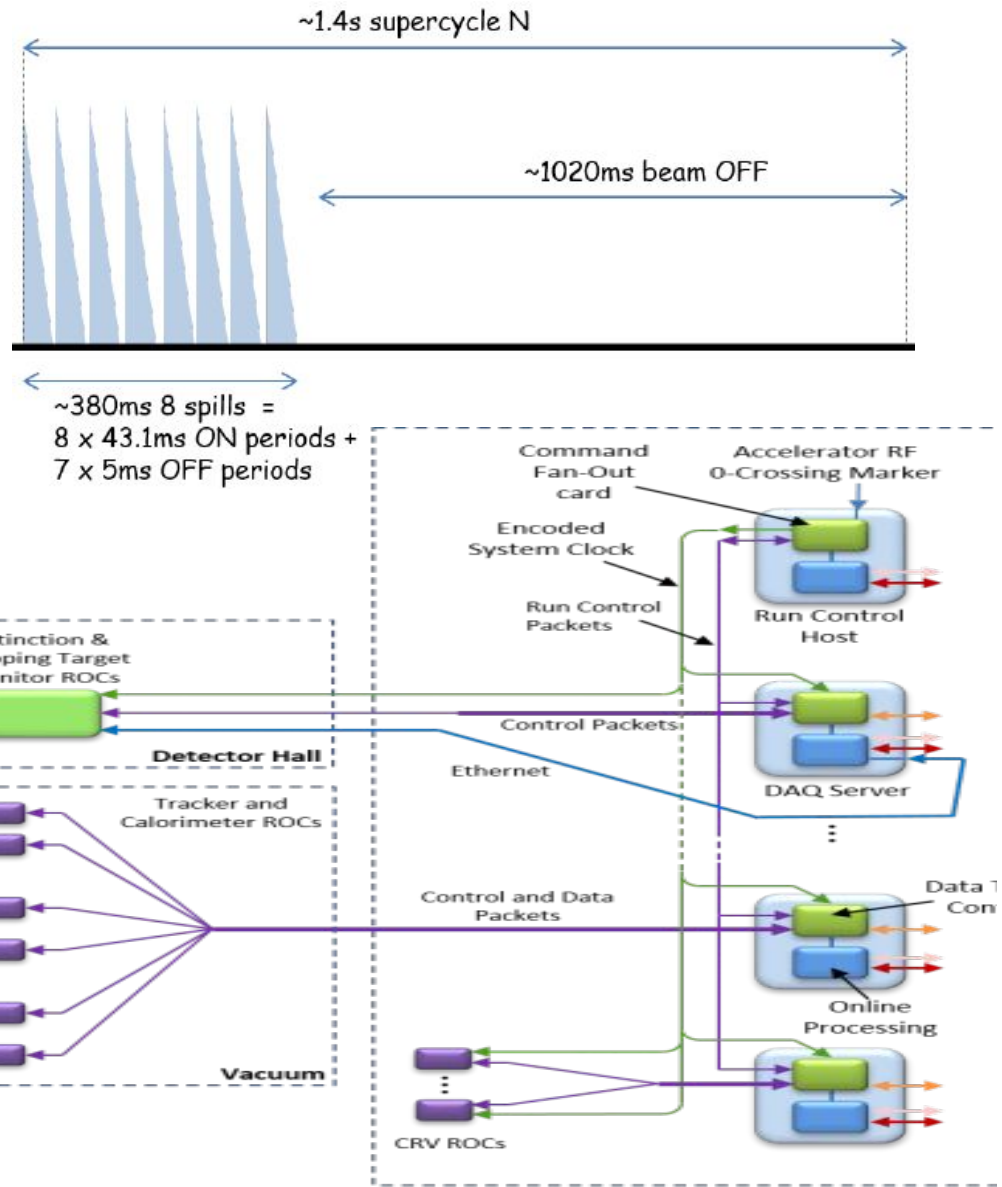




# Mu2e Timing Distribution

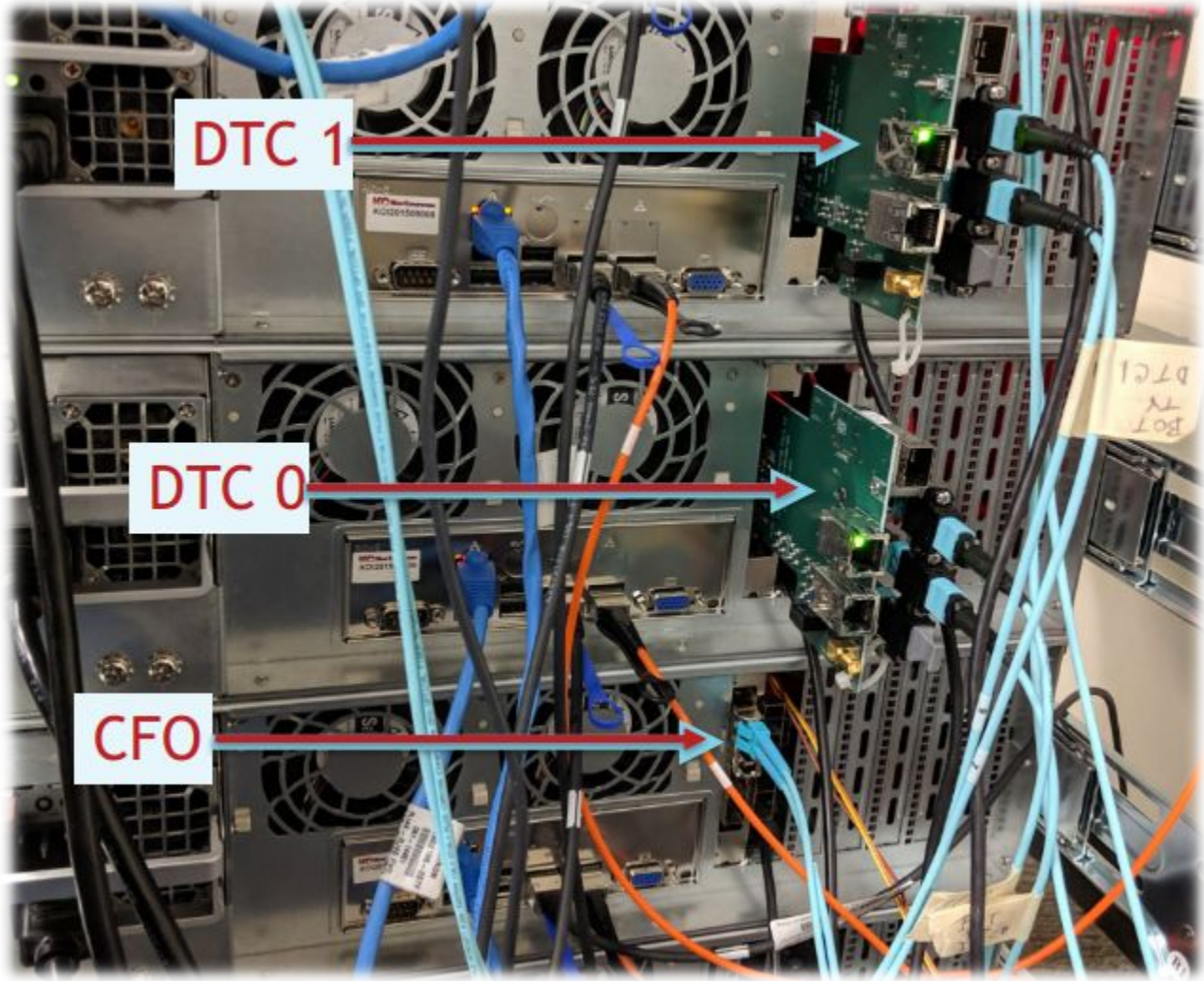
Requirement is to process 200K events/s

- 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

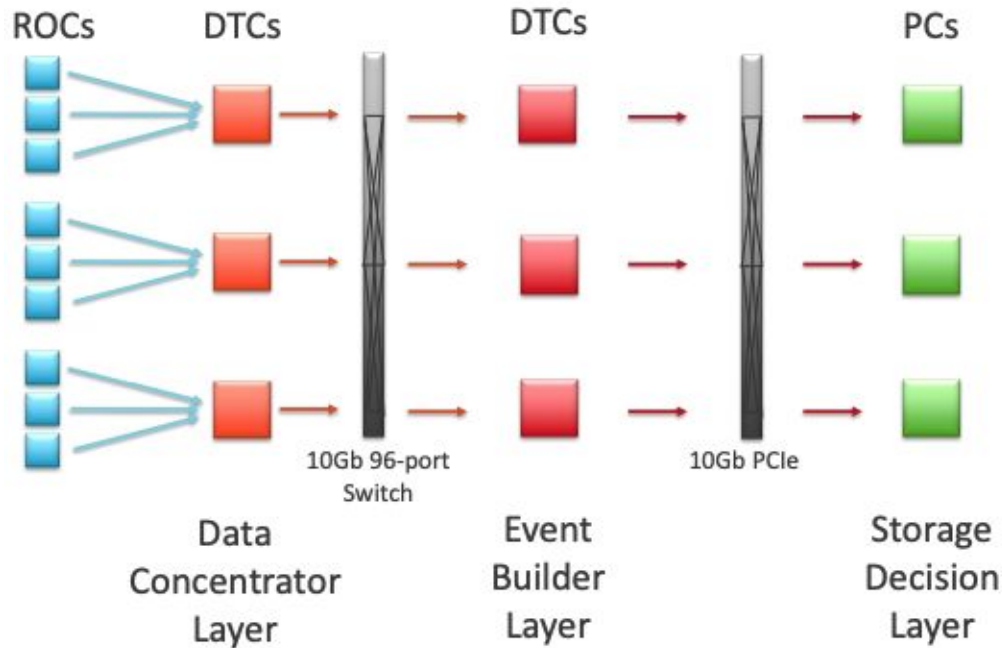


*DTC  $\leftrightarrow$  ROC Heartbeat packet (16 bytes) to specify the detail of each Event Window*

# 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*



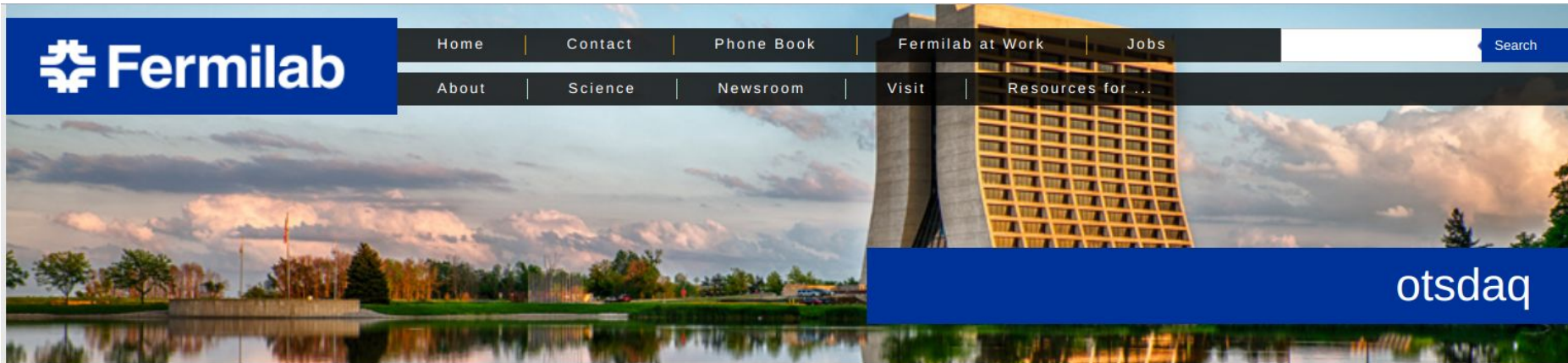
## *otsdaq* overview

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



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



## otsdaq

[Project Homepage](#)

[Source Code Documentation](#)

[User Manual](#)

[Tutorials \(User/Expert Training\)](#)

["First Demo" tutorial](#)



*otsdaq* is a Ready-to-Use data-acquisition (DAQ) solution aimed at test-beam, detector development, and other rapid-deployment scenarios. *otsdaq* 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. Additionally, an integrated Run Control GUI and readout software are provided, preconfigured to communicate with *otsdaq* firmware.

Last modified: 04/29/20 | [email Fermilab](#)

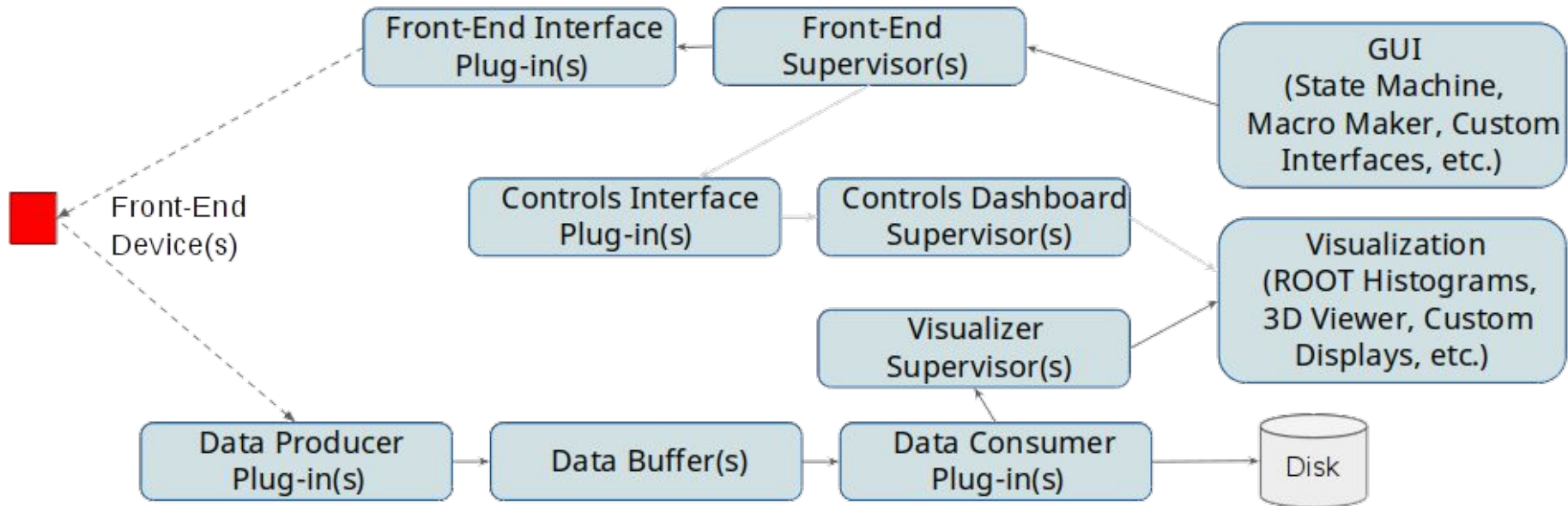
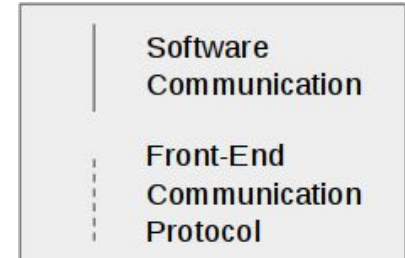


# otsdaq overview



## Data Flow Block Diagram

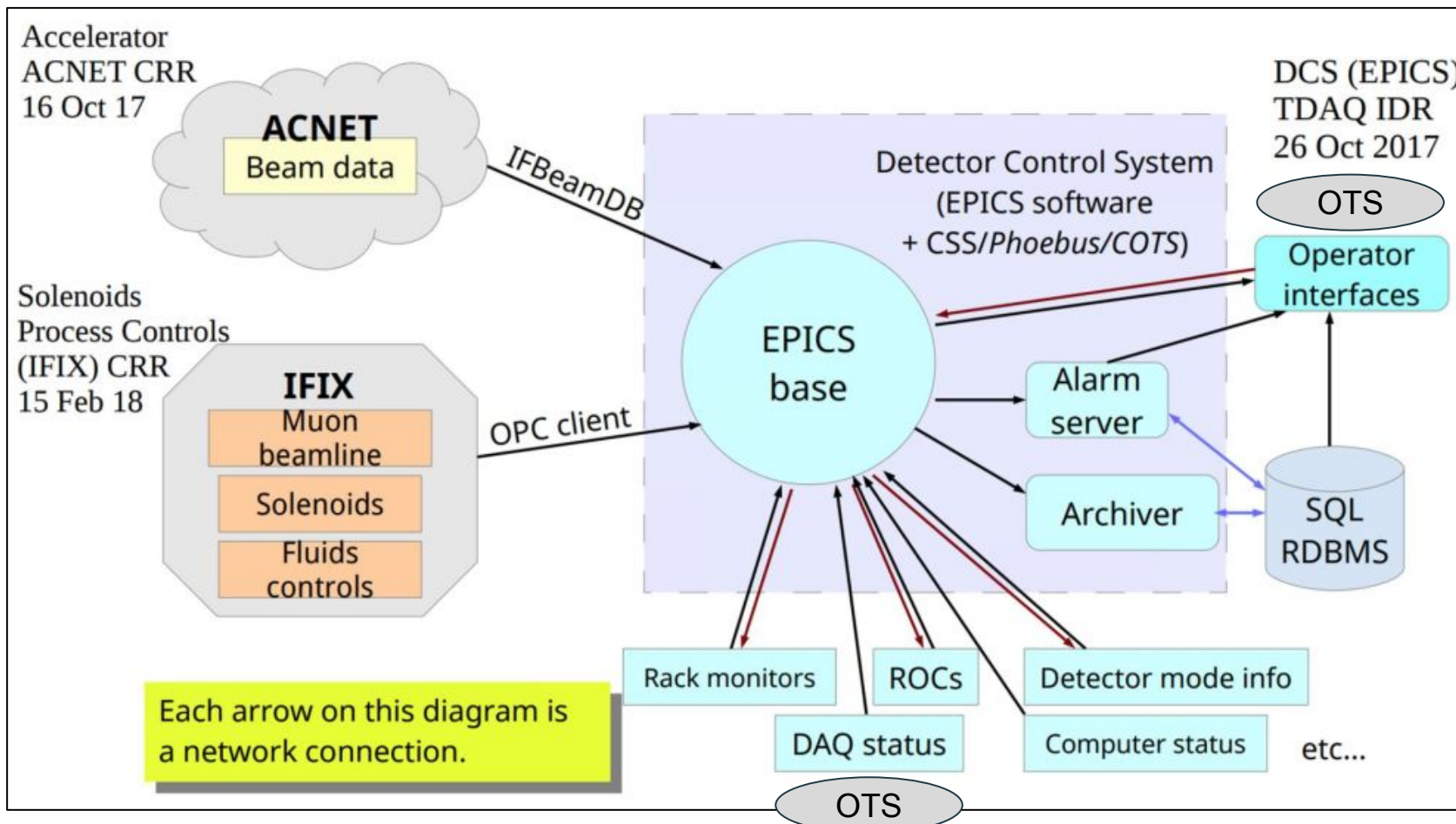
Server side is C++. User code is added through plugins (C++ classes inheriting from the appropriate class)



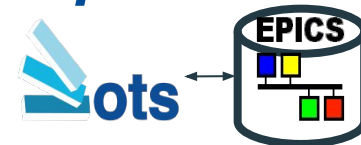
Web side is HTML and JavaScript. User code is added in the form of web-apps through .html files (including the appropriate .js and .css files)

# Slow Controls connection and **EPICS** plugin development in *otsdaq*

Experimental Physics and Industrial Control System



# Slow Controls connection and **EPICS** plugin development in *otsdaq*

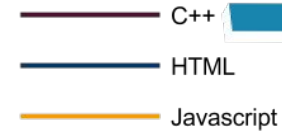


Channel subscription to **EPICS** uses Input Output Controller (**IOC**)

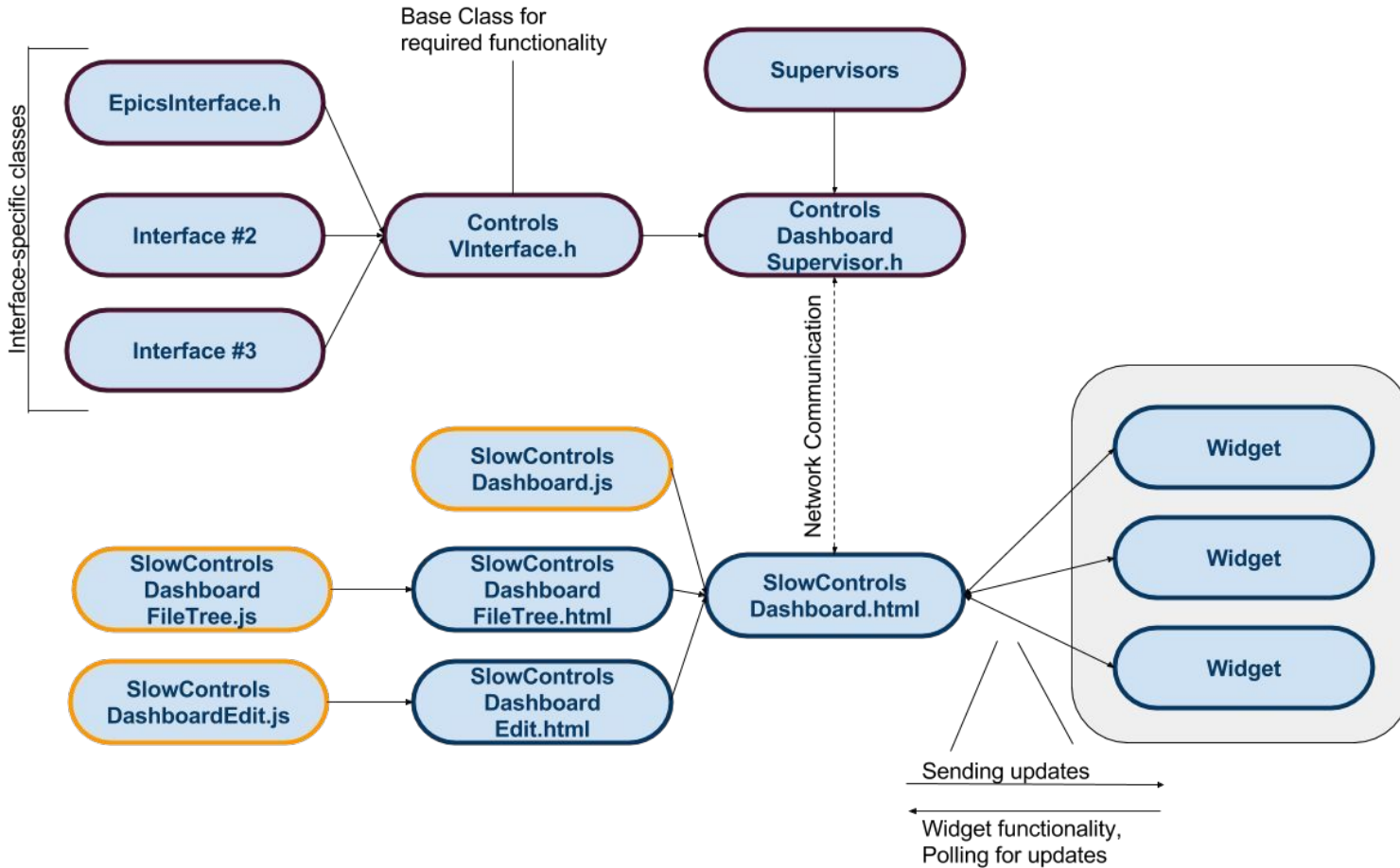
- integration of slow control in the online daq uses the same Interface plugin for:
  - a. Monitoring of all mu2e slow control channels
  - b. Sending Process Variables (PVs) of DAQ hardware info as **EPICS** channels and PVs settings into **EPICS** databases
- The Interface plugin:
  - a. Performs channel 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



# Slow Controls Monitoring in otsdaq



## Slow Controls GUI Hierarchy



# Slow Controls Monitoring GUI in otsdaq

Example of page loading

## Examples

Example of loaded page

Slow Controls Dashboard

File EditMode

Mu2e\_Weather\_2/humidity  
Status: NO\_ALARM  
Severity: NO\_ALARM

Mu2e\_Weather\_2/temperature\_degF  
Status: NO\_ALARM  
Severity: NO\_ALARM

PV Name	value	Alarm ...
Mu2e_Weather_1/so...	5.0 W/m2	NO_ALARM
Mu2e_Weather_1/te...	2 degC	NO_ALARM
Mu2e_Weather_1/te...	36.3 deg	NO_ALARM
Mu2e_Weather_1/wi...	77 mph?	NO_ALARM
Mu2e_Weather_2/b...	1016.0 m	NO_ALARM
Mu2e_Weather_2/pr...	0.0 inch	NO_ALARM

Production Solenoid    Transport Solenoid    Detector Solenoid

TDAQ LED

Mu2e\_Weather...  
Status: NO\_ALARM  
Severity: NO\_ALARM

Slow Controls Dashboard

Files

- Pages
  - private
    - phoebus\_test1
    - test1
    - test2
    - test3
    - test4
    - test5

PV Name	value	Alarm...	Alarm...
Mu2e_Weather_1/so...	5.0 W/m2	NO_ALARM	NO_ALARM
Mu2e_Weather_1/te...	2 degC	NO_ALARM	NO_ALARM
Mu2e_Weather_1/te...	36.3 degF	NO_ALARM	NO_ALARM
Mu2e_Weather_1/wi...	77 mph?	NO_ALARM	NO_ALARM
Mu2e_Weather_2/b...	1016.0 mbar	NO_ALARM	NO_ALARM
Mu2e_Weather_2/pr...	0.0 inch	NO_ALARM	NO_ALARM

Save Close

Slow Controls Dashboard

File Manager    Switch to Edit Mode    [Go Back to previous page](#)

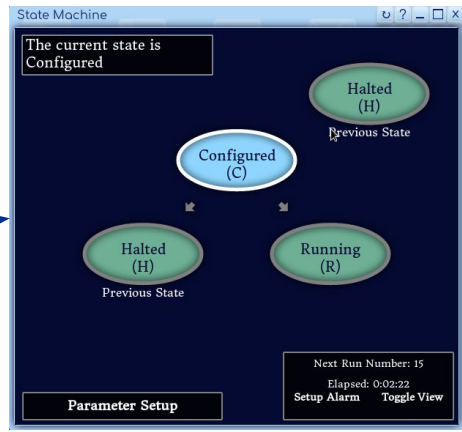
Calorimeter monitor in the slow control GUI

# Integration with State Machine

- **State Machine** Configuration and data subscription to **EPICS**
- Alarm propagation (from **EPICS**) and **otsdaq** State Machine handling  
DAQ HW, artdaq and DQM metrics configuration

```
Configure
- FE Supervisor
  - ApplicationGroupID : mainContextApps
  - LinkToSupervisorTable (UID = ...) //Tutorial FE Supervisor
  - LinkToInterfaceTable (GroupID = ...) //Defaults for an e
  - ExampleInterface
    - FEInterfaceGroupID : FEGroup0
    - Status : On
    - FEInterfacePluginName : DTFromEndInterface
    - LinkToPropertyTable (UID = DTCS) //Auto-gen
    - LinkToSlowControlChannelTable (GroupID = ...) //1 second
    - DTCS_SlowControlChannels
      - SlowControlChannels
        - ChannelType : assigned short
        - Units : degreesC
        - UniversalInterfaceAddress : 0xA2
        - UniversalInterfacePort : 0
        - ReadAccess : Yes
        - WriteAccess : No
        - RecordChangesOnly : False
        - DelayBetweenEventsInMilliseconds : 5
        - MonitoringEnabled : Yes
        - LocalMonitoringEnabled : No
        - LocalFilePath : DEFAULT
        - HostName : DEFAULT
        - SaveBinaryFile : False
        - AlarmEnabled : No
        - LatchAlarms : False
        - LoadThreshold : DEFAULT
        - HighThreshold : DEFAULT
        - HighPulseThreshold : DEFAULT
        - LinkToSlowControlSupervisorTable (UID = DEFAULT)-DISCONNECTED
        - SlowControlSocketAddress : 127.0.0.1
        - SlowControlSocketPort : 4208
        - SlowControlLocalAggregateStatusEnabled : Yes
        - SlowControlLocalFilePath : /home/rtviera/otsdaq/
        - SlowControlSubAlarmName : SlowControls
        - SlowControlSaveBinaryFile : False
      - Status : On
      - Class : ots:FESupervisor
      - Id : 219
```

artdaq EPICS metrics Plugin



```
Configure
- SlowControlDashboardSupervisor
  - ApplicationGroupID : mainContextApps
  - SlowControlDashboard
    - SlowControlInterfacePluginType : EpicsInterface
    - SlowControlChannelSourceTableList : DTCSInterfaceTable
    - LinkToConfigureAlarmsToMonitorTable (GroupID = ...) //Auto-generated from mock-up.
    - ConfigureAlarms
      - LinkToStartAlarmsToMonitorTable (GroupID = DEFAULT)-DISCONNECTED
      - LinkToPauseAlarmsToMonitorTable (GroupID = DEFAULT)-DISCONNECTED
      - LinkToResumeAlarmsToMonitorTable (GroupID = DEFAULT)-DISCONNECTED
      - LinkToStartAlarmsToMonitorTable (GroupID = ...) //Auto-generated from mock-up.
    - StartAlarms
      - StartAlarm
        - AlarmGroupID : StartAlarms
        - AlarmChannelName : M2e_TDAQ_$(M2E_OWNER)_ExampleInterface/dummyRegister0
        - IgnoreMinorSeverity : False
    - Status : On
    - Class : ots:SlowControlDashboardSupervisor
    - Id : 282
    - Instance : 1
    - Network : Local
    - Group : daq
    - Module : ${OTSDAQ_UTILITIES_LIB}/libSlowControlDashboard.so
    - ConfigurePriority : 0
    - StartPriority : 0
    - StopPriority : 0
    - LinkToPropertyTable (GroupID = DEFAULT)-DISCONNECTED
  - MacroMakerSupervisor
  - CodeEditorSupervisor
```

otsdaq EPICS Plugin

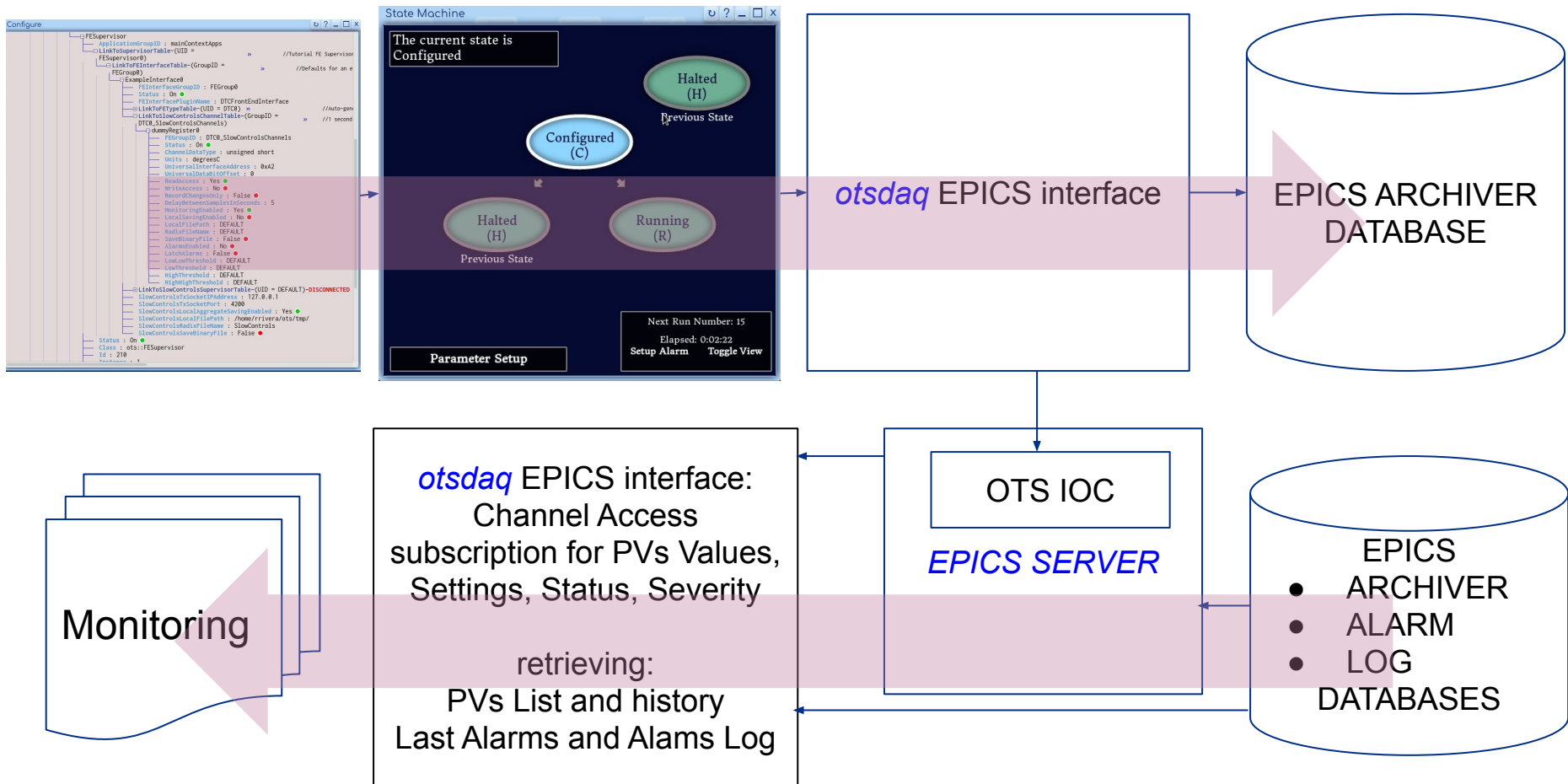
## Alarm Configuration





# 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



# 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

---

This work was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement no 734303, 822185, 858199, 101003460