



Mu2e TDAQ and slow control systems

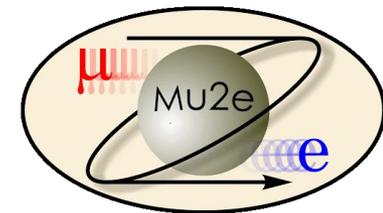
Antonio Gioiosa

Università di Pisa, INFN Pisa

*Level 3 Manager of Trigger &
Data Acquisition System Design & Test*

CPAD Instrumentation Frontier Workshop 2021

March 18, 2021

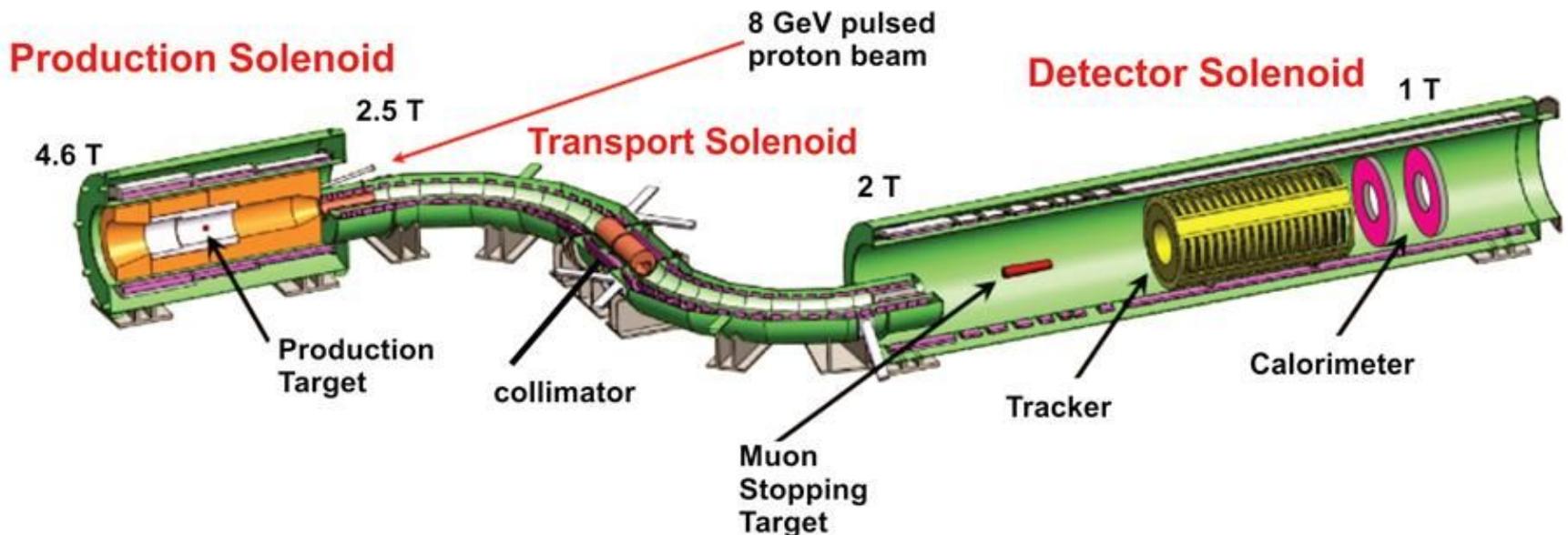


The Mu2e Experiment at Fermilab

Mu2e is an experiment under construction at **Fermilab** to measure the charged-lepton flavour violating neutrinoless conversion of a negative muon into an electron in the field of an aluminum nucleus

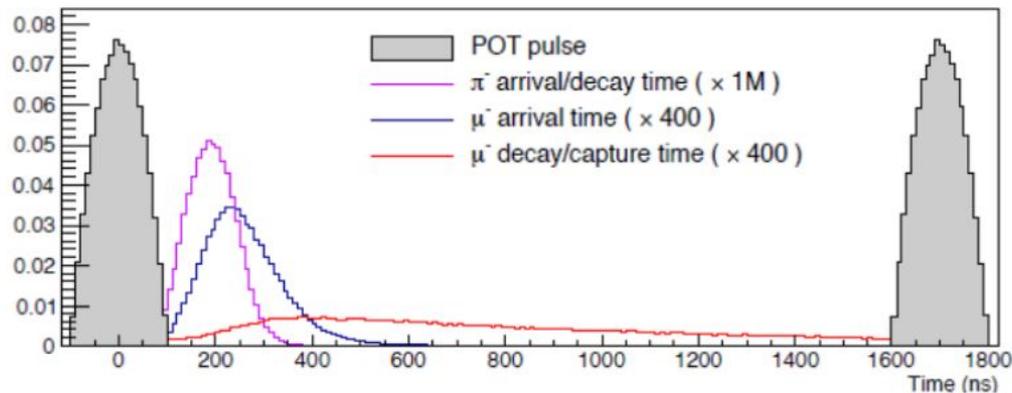
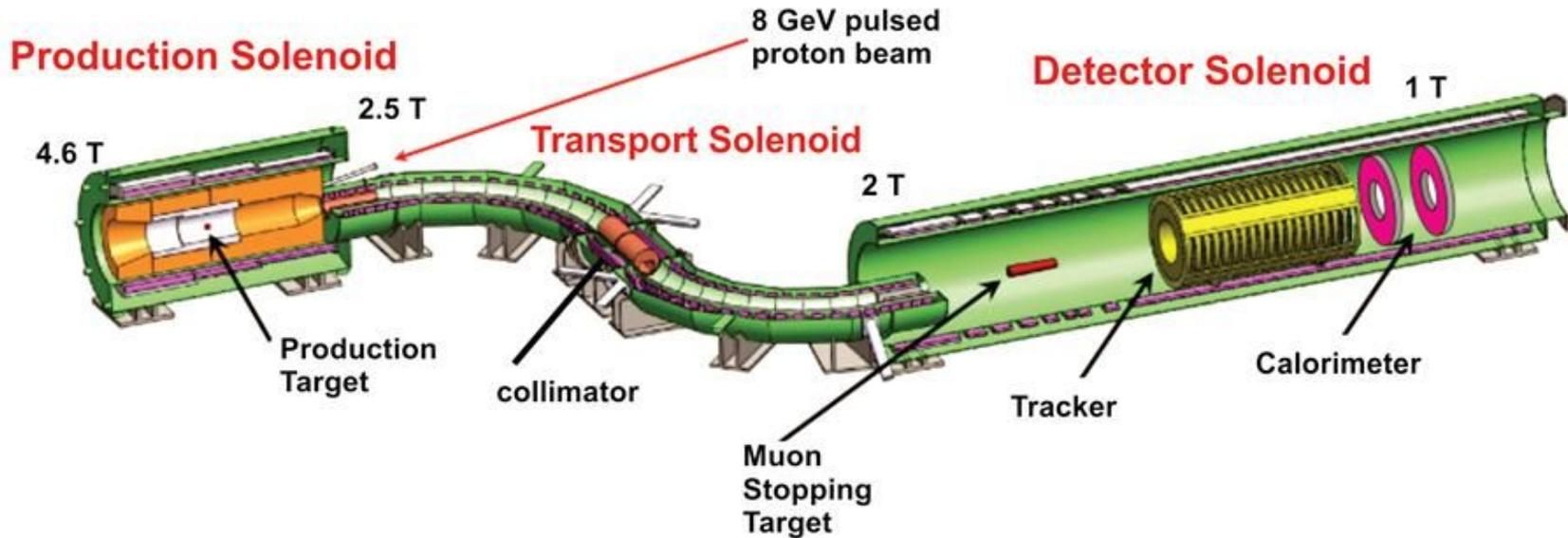
With the expected experimental sensitivity, Mu2e will improve the SINDRUM II limit ($7.0 \cdot 10^{-13}$) of four orders of magnitude

(assuming we will run for three years, with $3.6 \cdot 10^{20}$ protons, with a run time of $6.0 \cdot 10^7$ s, requiring a background under 1 event)



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)



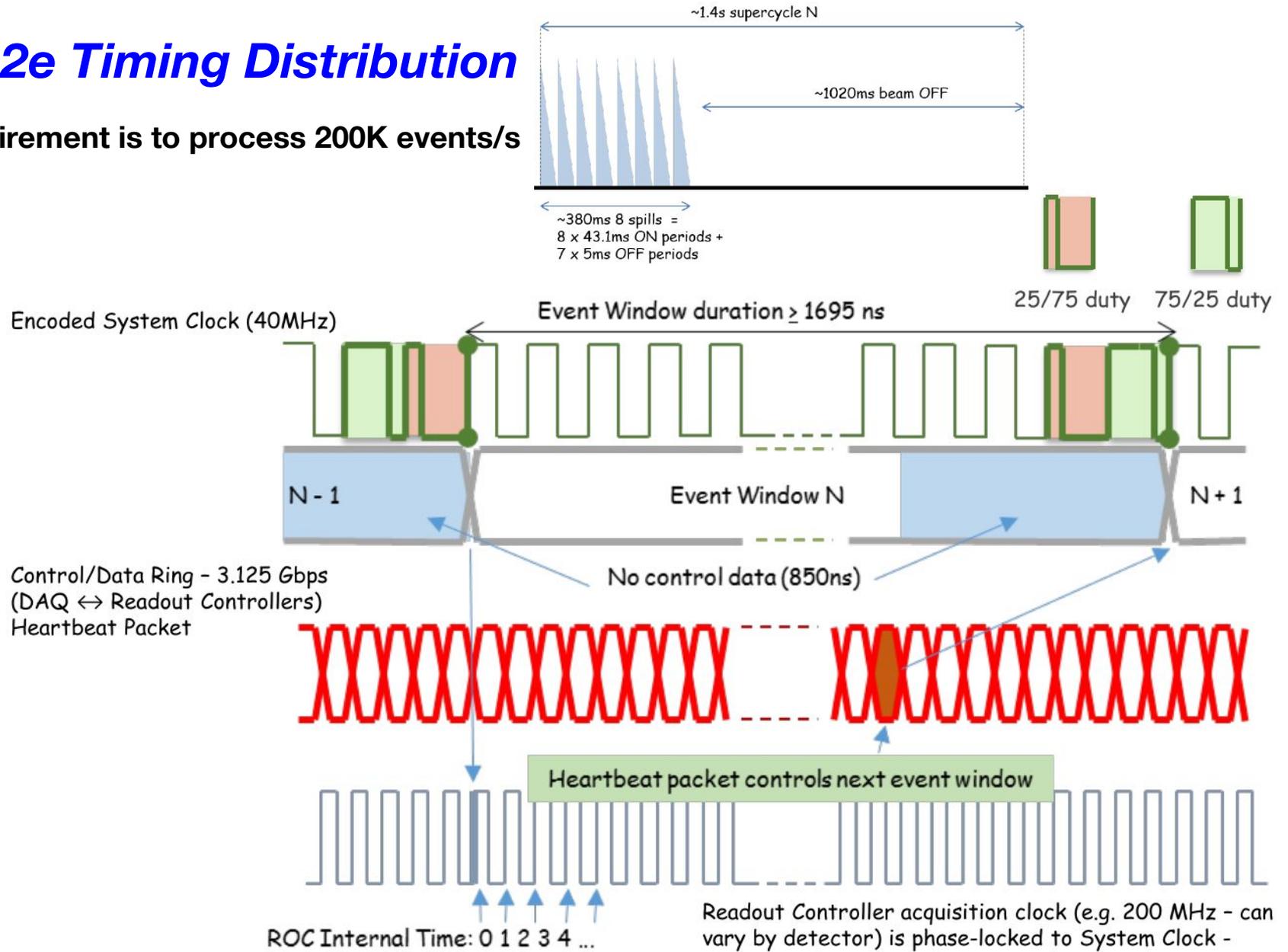
Mu2e TDAQ and Slow Control integration

Summary:

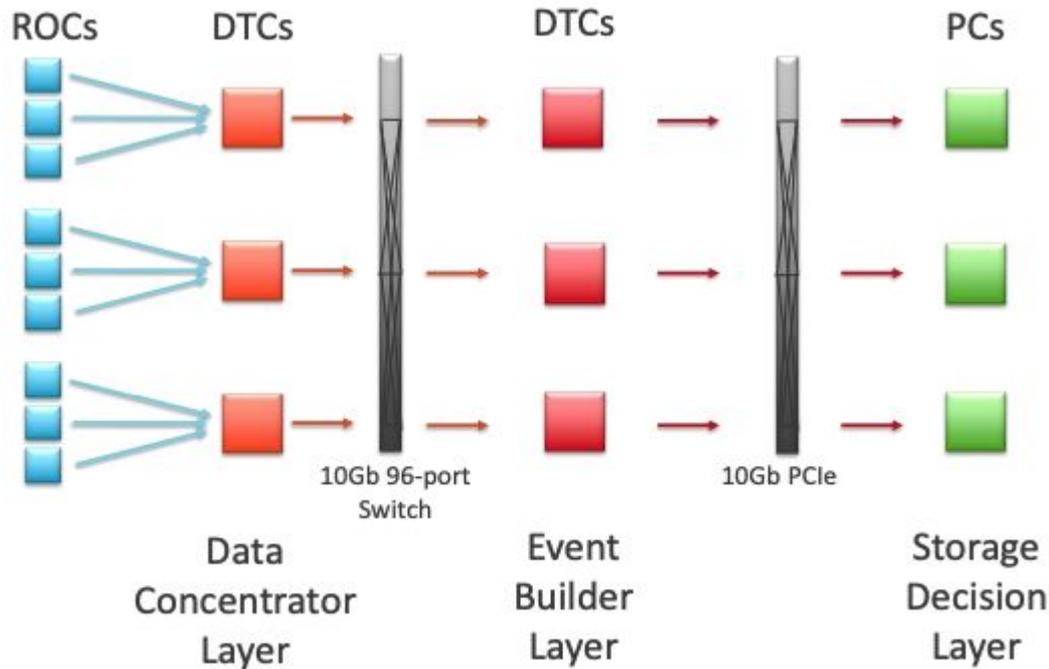
- Mu2e Timing Distribution and TDAQ Readout scheme
- Mu2e TDAQ components Diagram
- Online DAQ (***otsdaq***) overview
- Slow control and its integration in ***otsdaq***
 - **Monitoring** and Slow Controls GUI
 - Slow Controls **Integration** with ***otsdaq*** State Machine and **Alarm handling**
- Conclusions

Mu2e Timing Distribution

Requirement is to process 200K events/s

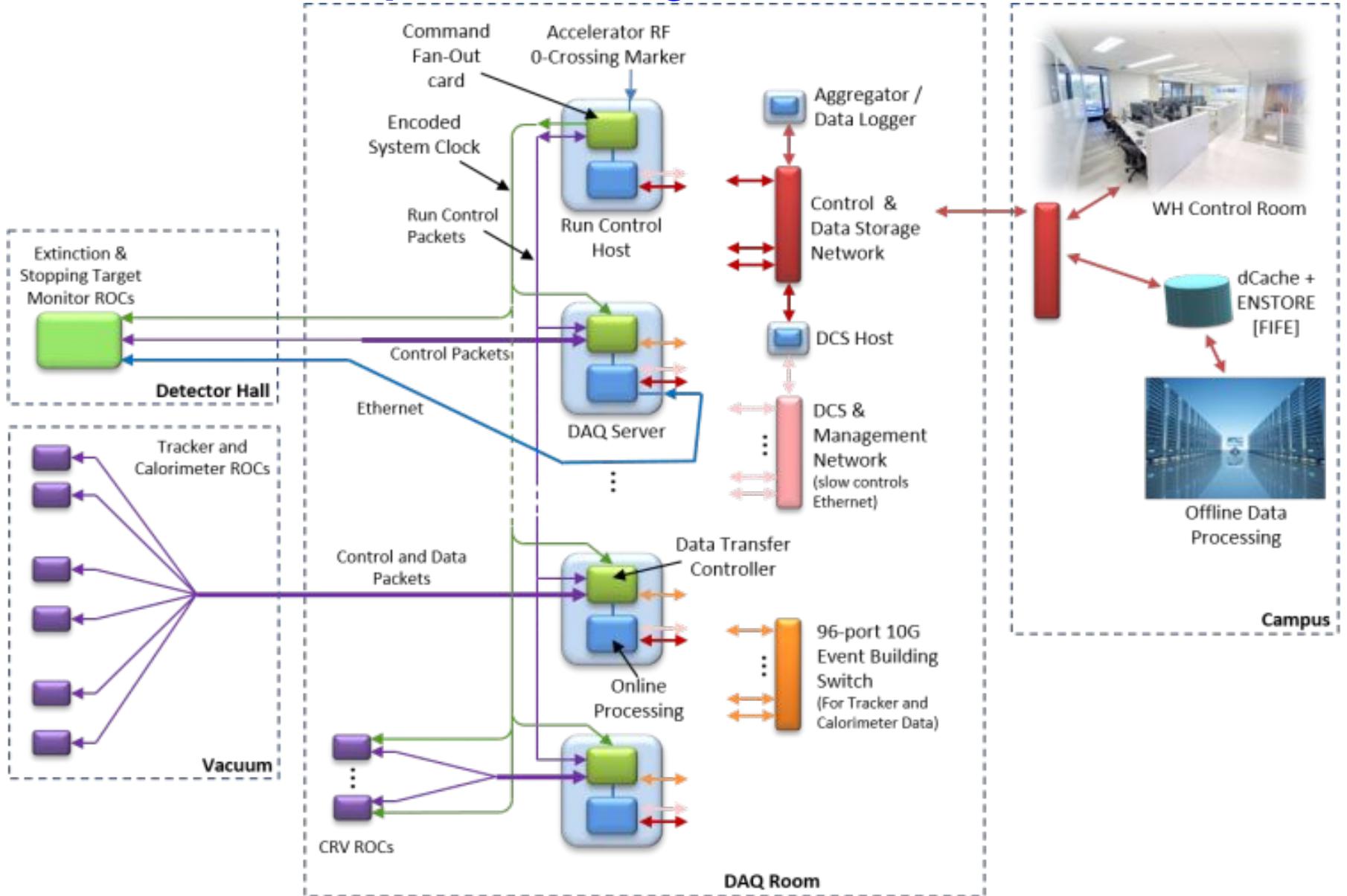


TDAQ Readout scheme



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

Mu2e TDAQ components Diagram



Mu2e TDAQ Data Flow and High Level Trigger Software

Run Control and Data Flow summary

1. Experiment defined Run Plan is coordinated by CFO. The System Clock (40MHz) and Event Window markers originate at the Command Fan-Out Card (CFO) and are distributed to ROCs
2. CFO distributes System Clock and Event Windows to DTCs with fixed latency
3. DTCs distribute System Clock and Event Windows to ROCs with fixed latency
4. ROCs respond to Data Requests
5. A slice of the detector arrives at each DTC (6 ROCs for 1 DTC)
6. DTCs forward data slice through Event Building Switch to round-robin DTC destination
7. DTCs receive full events from multiple DTC sources, pre-process, and pass through PCIe to online processing
8. Trigger decisions are made in online processing
9. Trigger accept causes readout of corresponding CRV data
10. Event data from all detectors are aggregated at Data Logger
11. Experiment data is transferred from Data Logger to persistent storage

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/>



otsdaq

[Project Homepage](#)

[Source Code Documentation](#)

[User Manual](#)

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

["First Demo" tutorial](#)



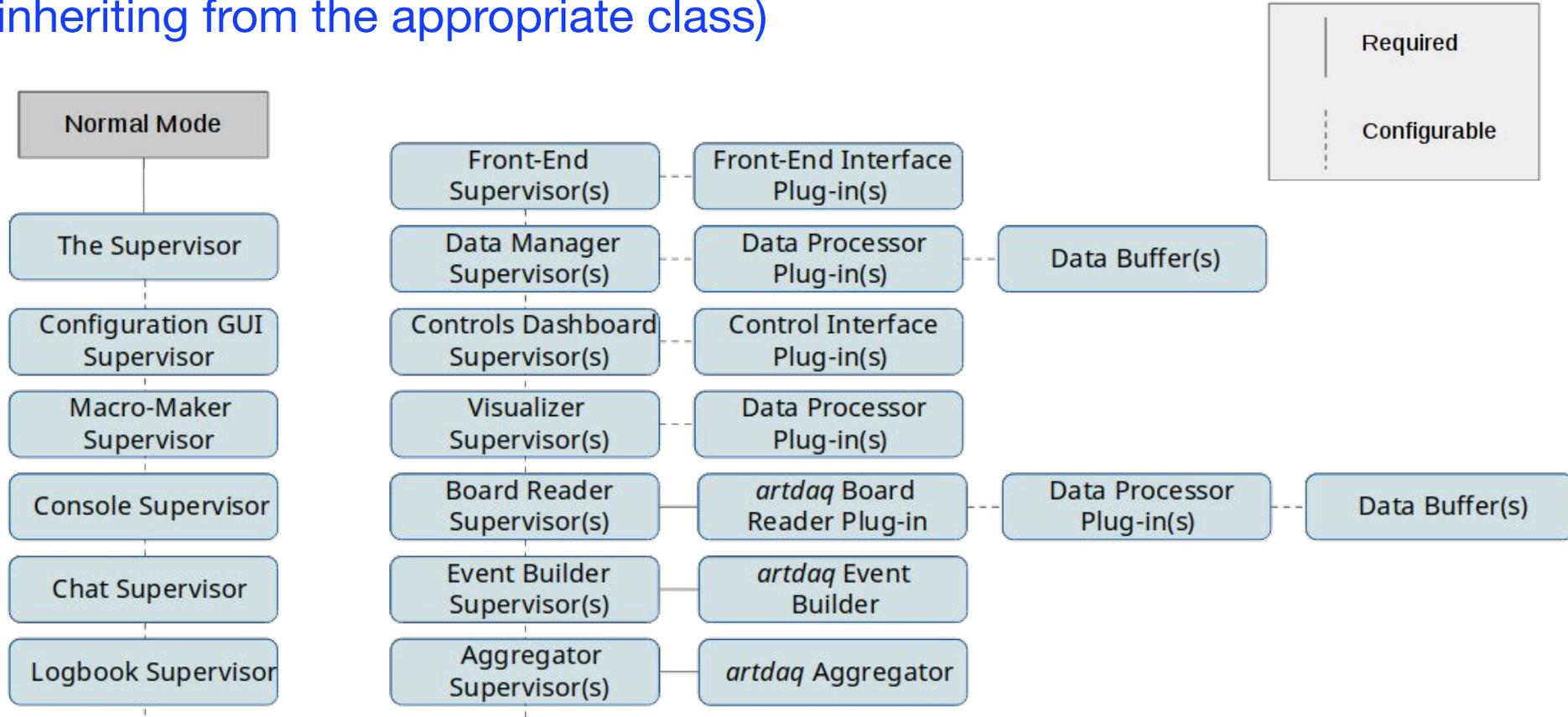
otsdaq is a Ready-to-Use data-acquisition (DAQ) solution aimed at test-beam, detector development, and other rapid-deployment scenarios. *otsdaq* uses the *artdaq* DAQ framework under-the-hood, providing flexibility and scalability to meet evolving DAQ needs. *otsdaq* provides a library of supported 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



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

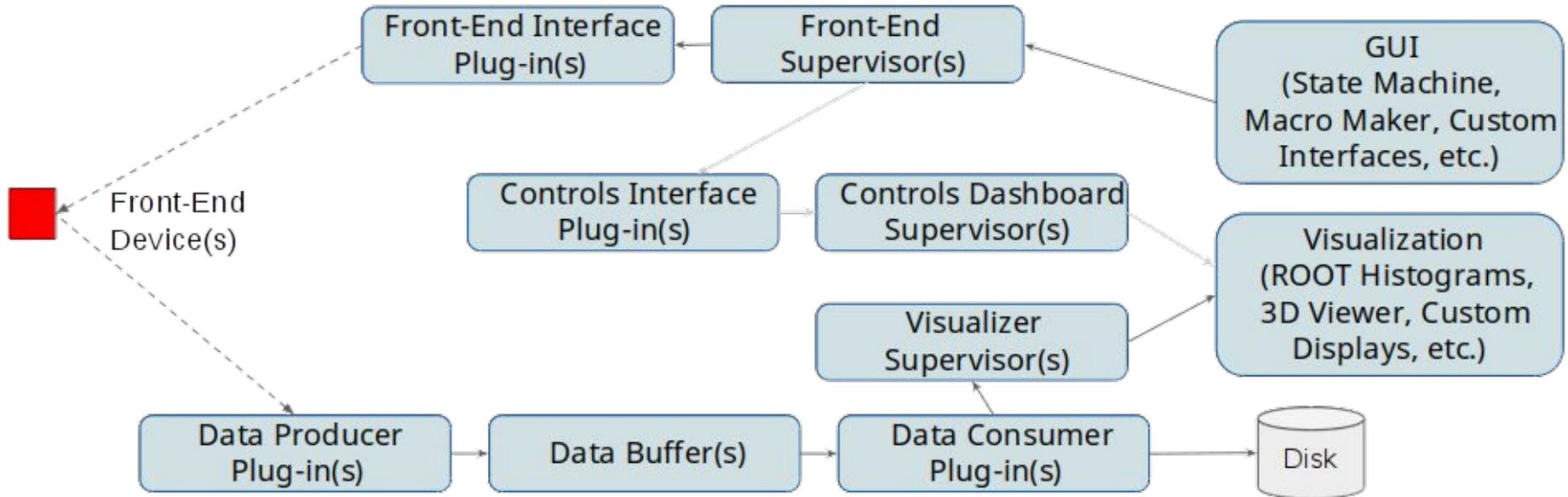
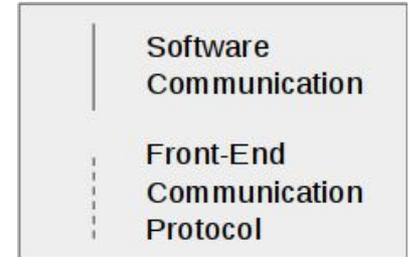


Web side is HTML and JavaScript. User code is added in the form of web-apps through .html files (including the appropriate .js and .css files)

otsdaq overview



Data Flow Block Diagram



otsdaq overview

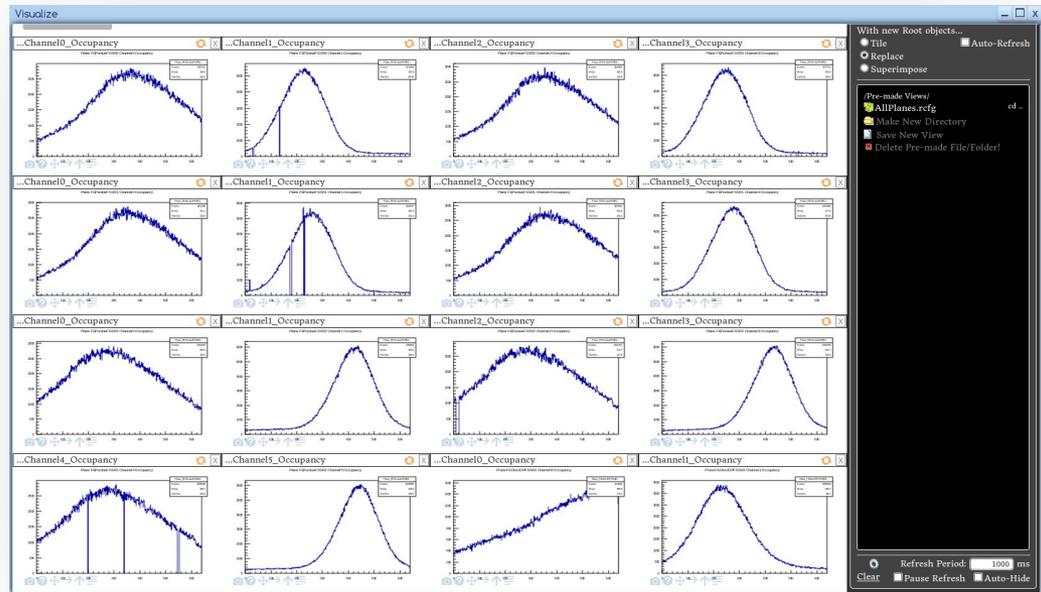
Online DAQ features

1. Context-sensitive tooltips
2. User management
3. Web desktop
4. State machine
5. Front-end interfaces
6. Macro Maker
7. FE Macros
8. Macro Maker mode
9. Iterator
10. Configuration tree
11. Configuration schema editor
12. Configuration change control
13. Configuration database
14. Data processing (and artdaq)
15. Data quality monitoring
16. Visualizer
17. Logbook
18. Chat
19. Console
20. Code editor
21. DTC Client library
22. EPICS interface
23. Slow controls
24. web-app
25. DBMS interface

otsdaq overview

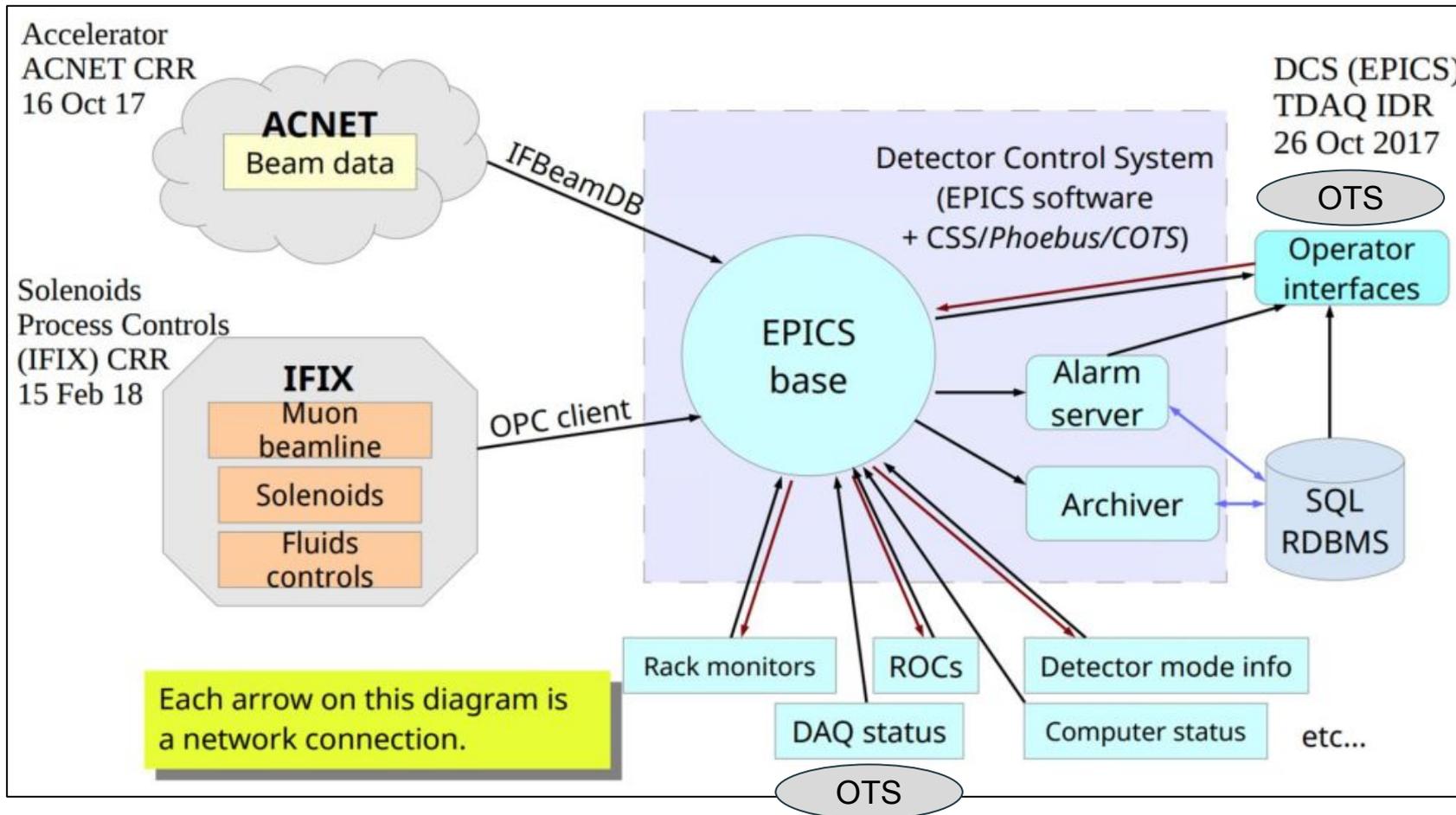
Data processing: Page example

- Mu2e's event window data will be processed through artdaq modules
- Data processor and Data Quality Monitor **DQM** plugins are provided by otsdaq core
- **DQM metrics** are distinct from event rate and data flow metrics (which artdaq provides)
- **DQM** generates data products that are sent to an **artdaq Dispatcher**, which aggregates **DQM metrics** and presents them to a visualizer application



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

Experimental Physics and Industrial Control System



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

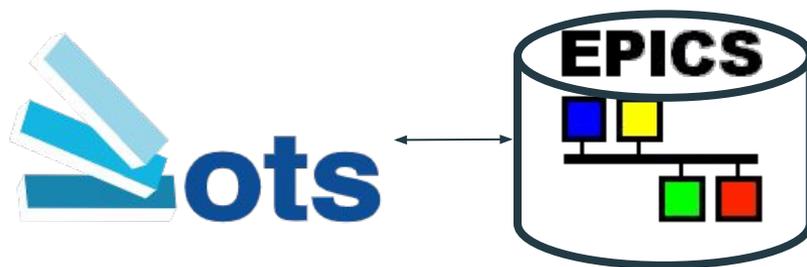
Channel subscription to **EPICS** (uses Input Output Controller **IOC**)

- Value
- Alarm (Status, Severity)
- Settings
 - *PV Unit, Lower and Upper Warning Limits, Lower and Upper Alarm Limit, Lower and Upper Control Limits, Lower and Upper Display Limits*



- Channel history and alarms retrieving from EPICS Archiver Databases

- *dc_s_archiver*
- *dc_s_alarm*
- *dc_s_log*



Slow Controls Monitoring in otsdaq

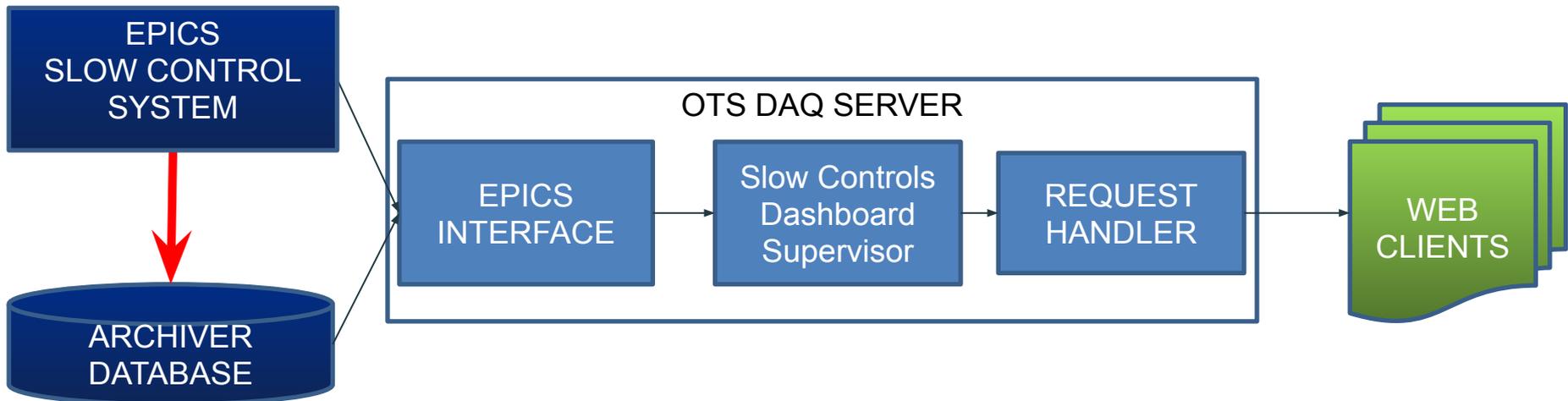
Slow Controls Software purpose

- Allow the user to monitor or interact with their own DAQ hardware. Able to see things such as:

Alarms, Warnings, Readouts, Timestamps, Status

- Interact through a web interface that is:

Lightweight, User-Friendly, Plug n' Play, Customizable



Slow Controls **WEB** Monitoring in *otsdaq*

developed in JavaScript and HTML (client side) and C++ (server side)

Basic Widget Mechanics

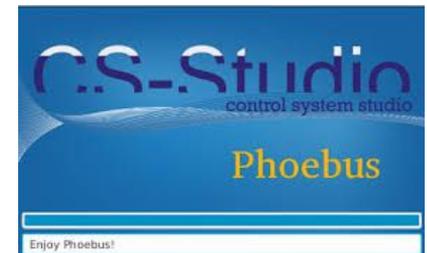
- All widgets have six required methods:
init(), getParameters(), setParameters(), setupPVs(), newWidget(), and newValue()

Widget properties

- Dynamic sizing
- Proper handling of setups
- Value error, warning and alarm handling
- Disconnection handling

Load and save dashboard page in XML

Cs-Studio Phoebus (EPICS GUI) compatible format



Slow Controls Monitoring in otstdaq

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 degC	NO_ALARM
Mu2e_Weather_1/wi...	77 mph	Upper Warning
Mu2e_Weather_2/b...	1016.0 m	Upper Alarm L
Mu2e_Weather_2/pr...	0.0 inch	Lower Control

Production Solenoid Transport Solenoid Detector Solenoid

TDAQ LED

Mu2e_Weather...
Status: NO_ALARM
Severity: NO_ALARM

Slow Controls Dashboard

Files

- Pages
 - private
 - phoebus_test1
 - test
 - Load Page
 - Load Page in new Desktop Window
 - Load Page in new Browser Tab
 - Create a desktop icon

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

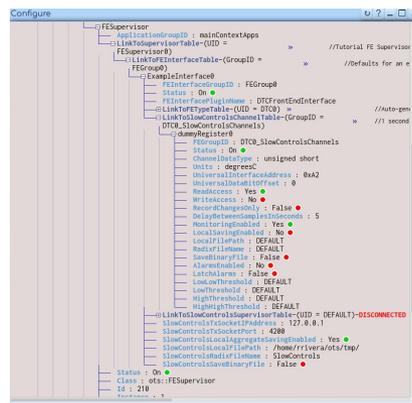
Slow Controls Dashboard

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

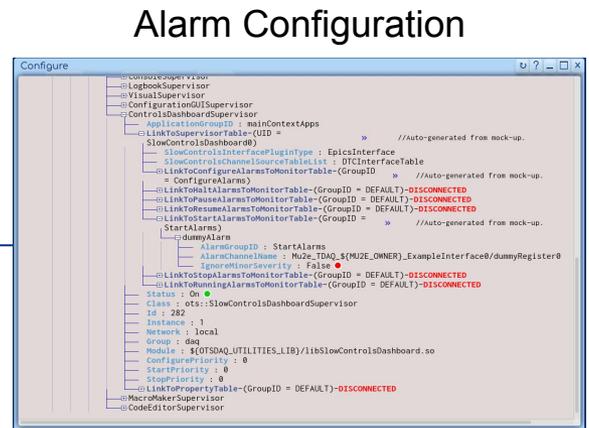
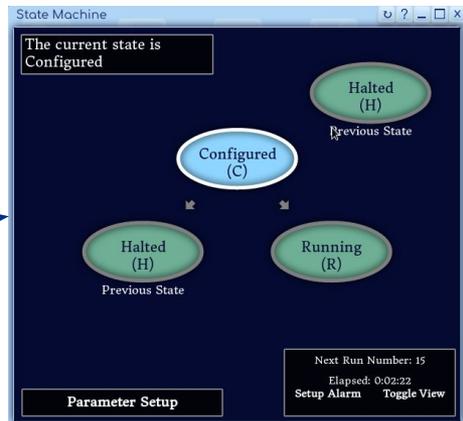
Calorimeter monitoring in the slow control GUI

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



artdaq EPICS metrics Plugin



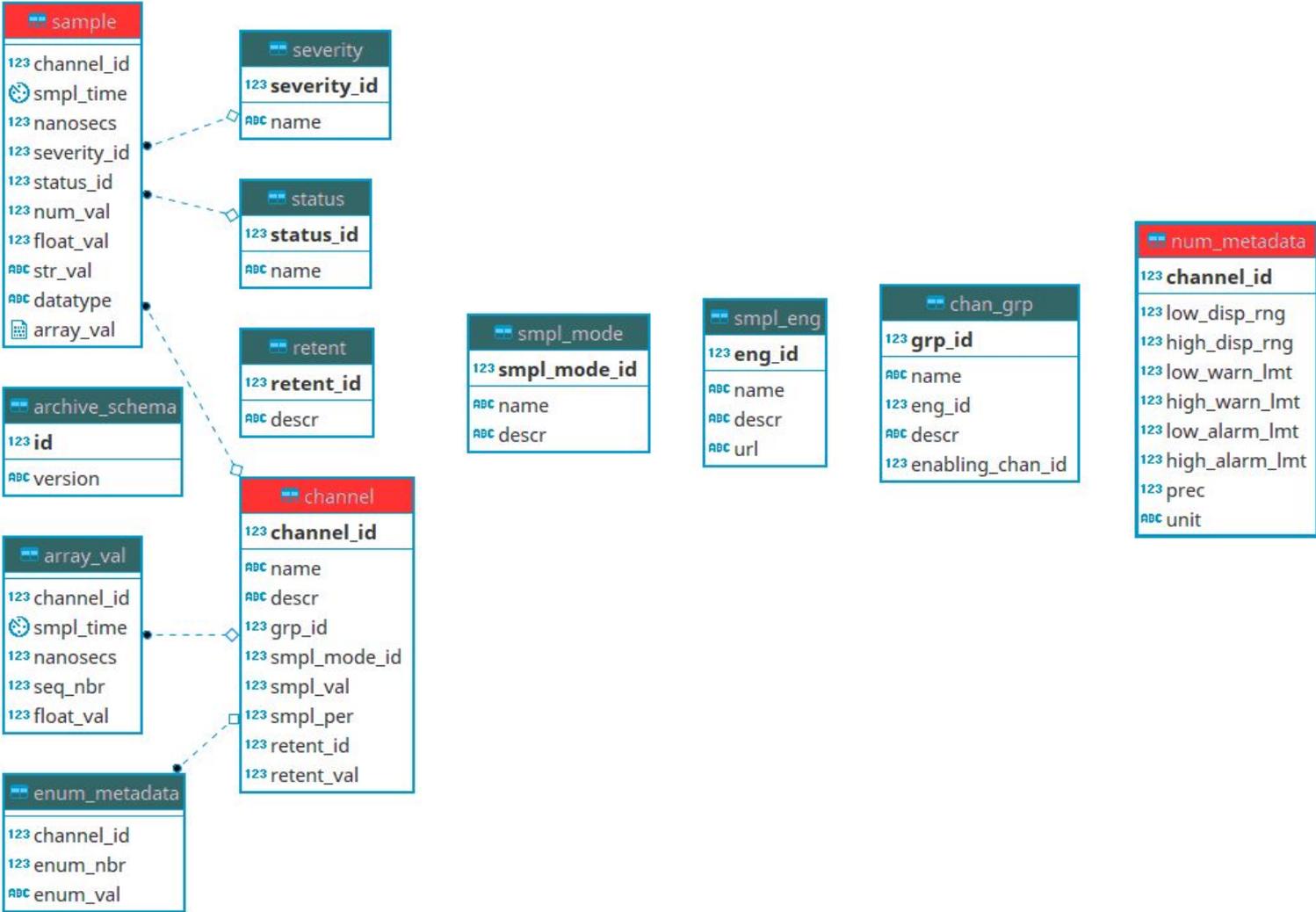
otsdaq EPICS Plugin

Alarm Configuration



EPICS Database

- Postgres DBMS



Conclusions



- Mu2e Experiment is under construction at Fermilab and will be ready for data taking next years
- Mu2e TDAQ and slow control are in large part developed according to the requirements (200K events/s for data taking) and hardware tests are 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 informations to **EPICS**, and a real configuration and Integration with the *otsdaq* State Machine

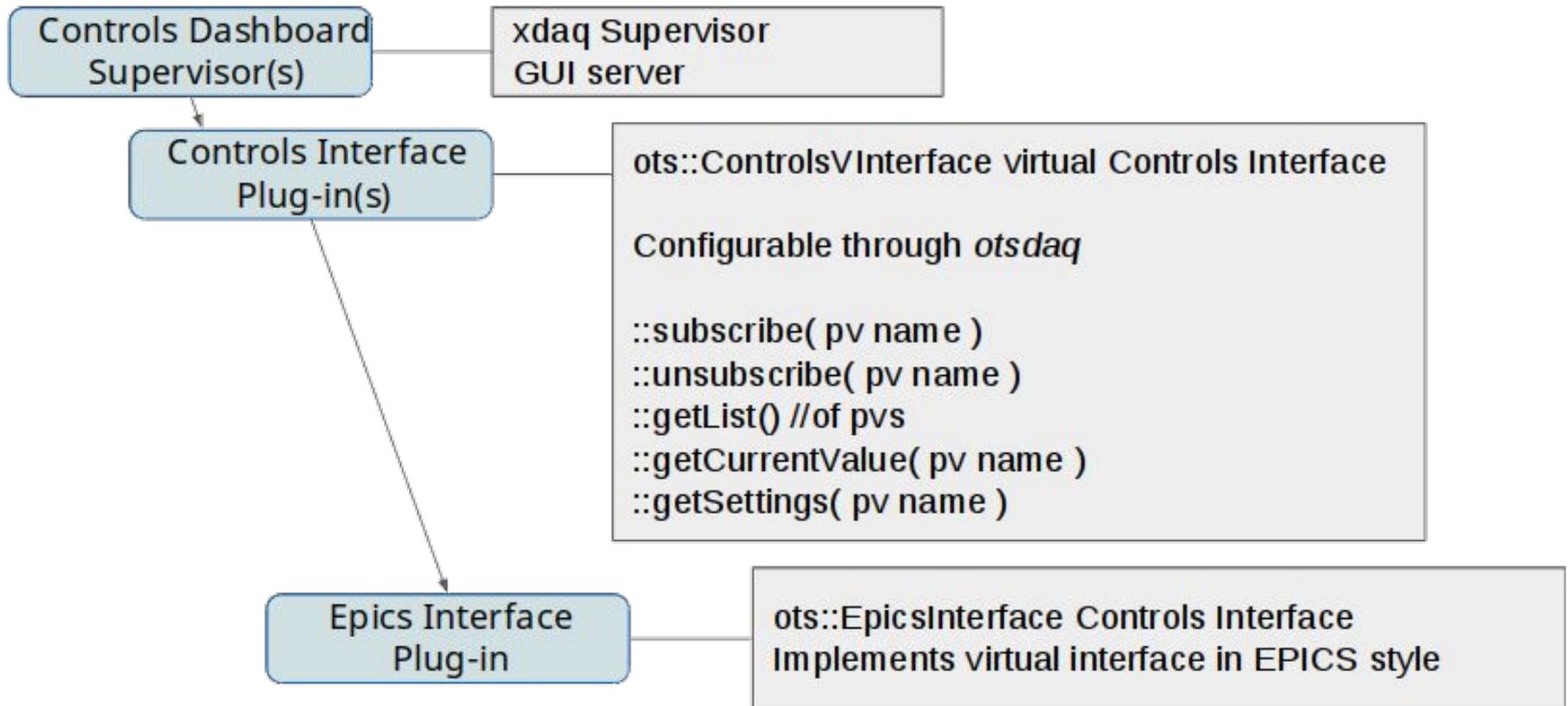
This work was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement no 734303, 822185, 858199, 101003460

Backup Slides

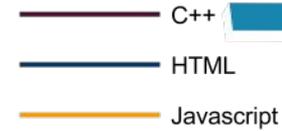
Slow Controls Monitoring in otsdaq



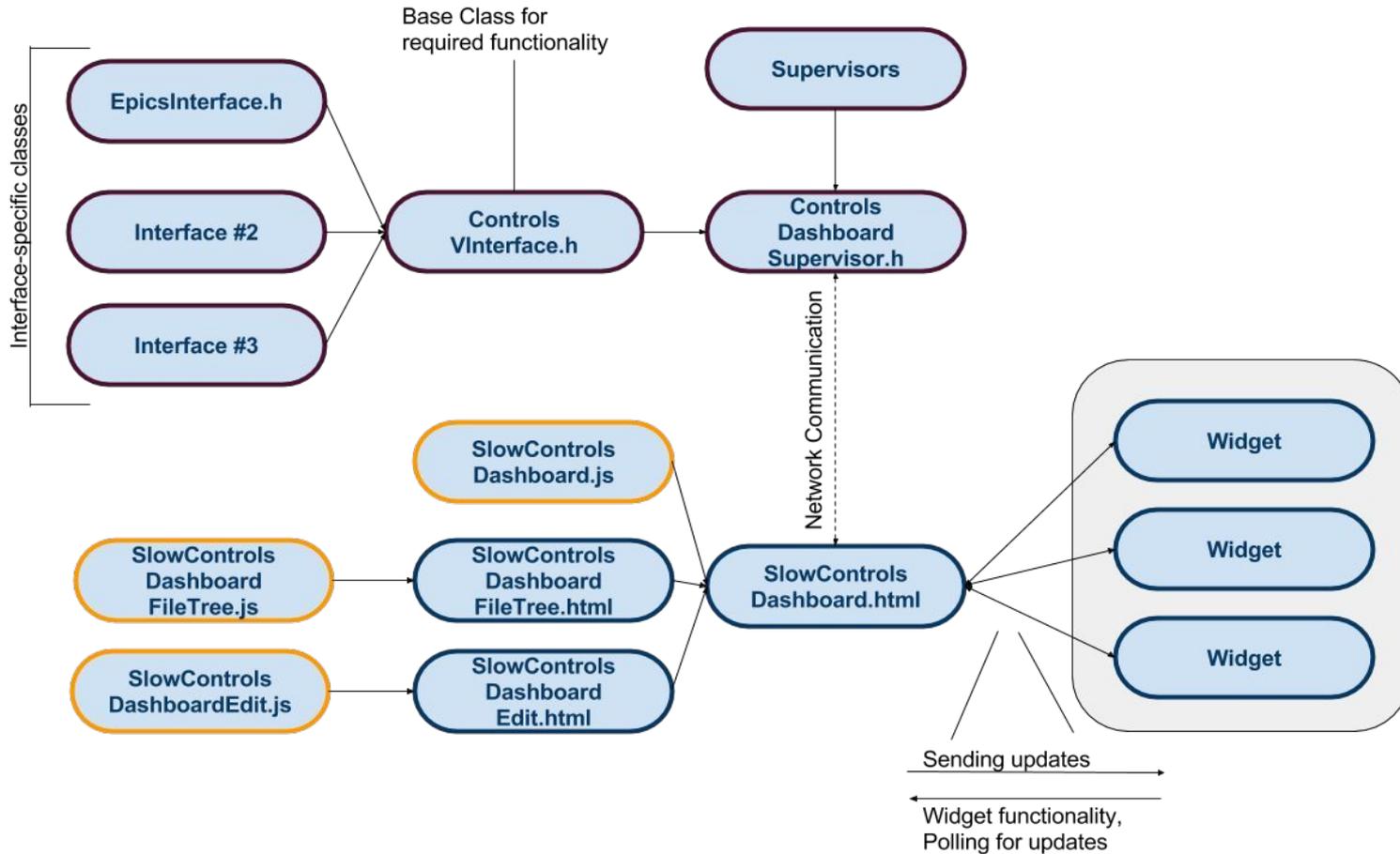
Slow Controls C++ Hierarchy



Slow Controls Monitoring in otsdaq



Slow Controls GUI Hierarchy

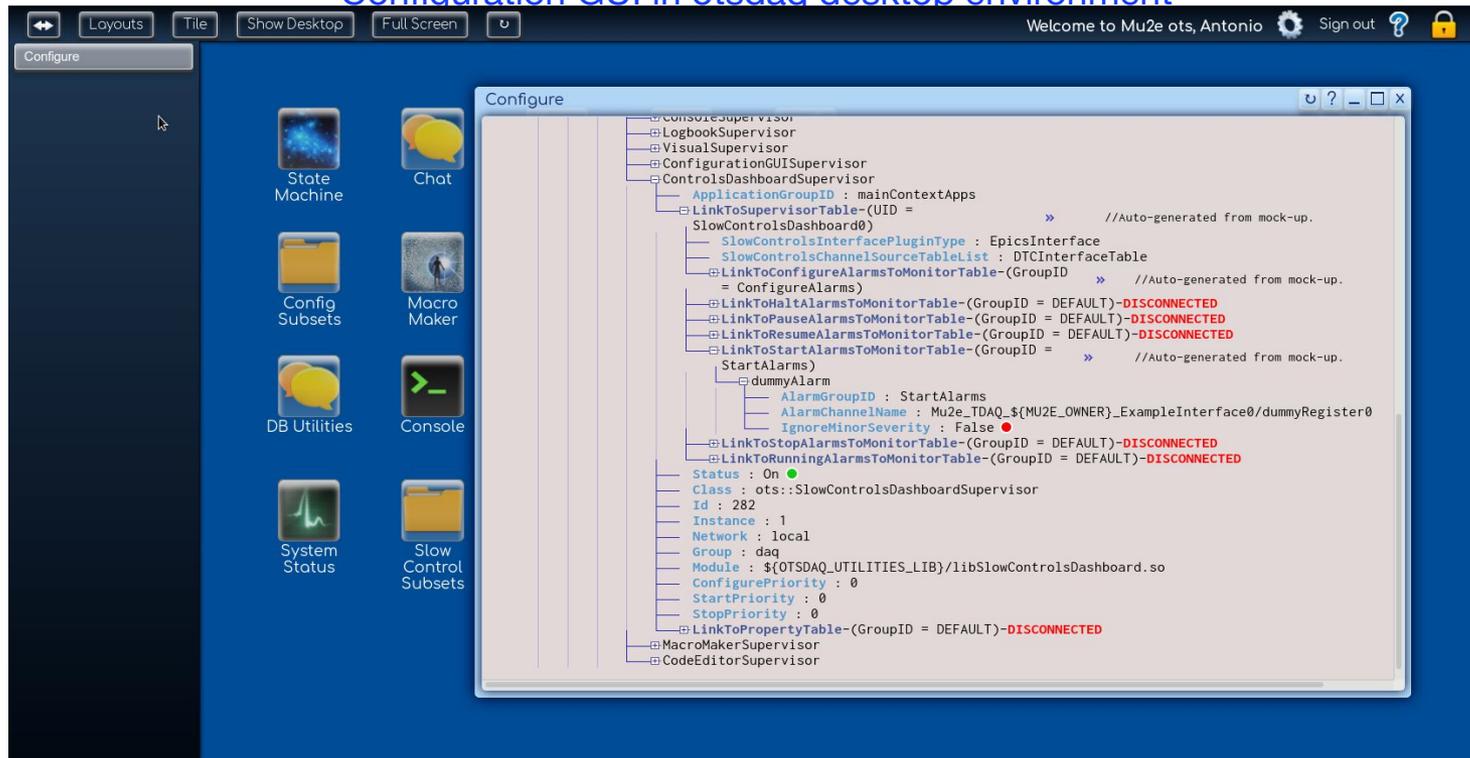


Slow Controls Monitoring in otsdaq

Configuring by specific tables in otsdaq

DesktopIconTable, XDAQApplicationPropertyTable, XDAQApplicationTable, XDAQContextTable

Configuration GUI in otsdaq desktop environment



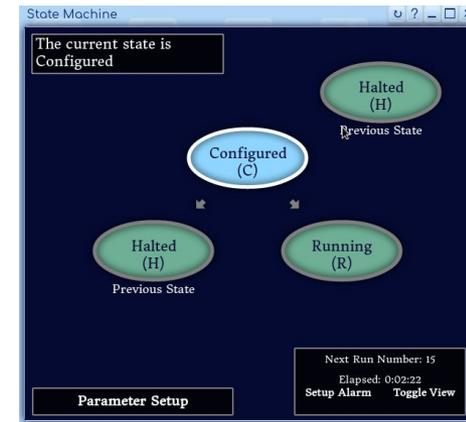
*Integration of **otsdaq** front-end DAQ hardware and **artdaq** metrics with **EPICS***

Actions designed and developed in *otsdaq*

1. *otsdaq* DCS channels Front End and tables configuration
2. *otsdaq* State Machine configuration implementation
3. add/update channels info for **IOC** and **Archiver** DB
4. software **IOC** restarting
5. **EPICS Archiver** restarting
6. new *otsdaq* *epics_plugin* channels subscriptions to EPICS
7. Sending configured channels values to **EPICS**:
otsdaq DCS channels new values → *artdaq Metric Manager*
→ *software **IOC*** → **EPICS** → *otsdaq DCS GUI*

Integration with State Machine

- **Alarm** propagation (from **EPICS**) and *otsdaq* state machine handling is available: needs just to identify which **PV alarms**, *status* and *severity* will be propagated
- *Tables and parameters designed for configuration*
 - SupervisorTable parameters:
 - *Slow Controls Interface Plugin Type*
 - *Slow Controls Channel Source Table List (HW list i.e. DTC Interface, CFO Interface)*
 - Alarms To Monitor Tables for transition to states:
 - *Configure*
 - *Halt*
 - *Pause*
 - *Resume*
 - *Start*



```
Configure
├── LogbookSupervisor
├── VisualSupervisor
├── ConfigurationUISupervisor
├── ControlDashboardSupervisor
├── ApplicationGroupID : mainContextApps
├── LinkToSupervisorTable (UID = ...) //Auto-generated from mock-up.
├── SlowControlDashboard
├── SlowControlInterfacePluginType : EpicsInterface
├── SlowControlChannelSourceTableList : DTCInterfaceTable
├── LinkToConfigureAlarmsToMonitorTable (GroupID = ...) //Auto-generated from mock-up.
├── * ConfigureAlarms
├── LinkToHaltAlarmsToMonitorTable (GroupID = DEFAULT)-DISCONNECTED
├── LinkToPauseAlarmsToMonitorTable (GroupID = DEFAULT)-DISCONNECTED
├── LinkToResumeAlarmsToMonitorTable (GroupID = DEFAULT)-DISCONNECTED
├── LinkToStartAlarmsToMonitorTable (GroupID = ...) //Auto-generated from mock-up.
├── StartAlarms
├── EmptyAlarm
├── AlarmGroupID : StartAlarms
├── AlarmChannelName : Mu2e_TDAQ_${MIZE_OWNER}_ExampleInterface/dummyRegister
├── IgnoranceSeverity : False
├── LinkToStopAlarmsToMonitorTable (GroupID = DEFAULT)-DISCONNECTED
├── LinkToRunningAlarmsToMonitorTable (GroupID = DEFAULT)-DISCONNECTED
├── Status : On
├── Class : etsi:SlowControlDashboardSupervisor
├── Id : 282
├── Instance : 1
├── Network : local
├── Group : daq
├── Module : ${OTSDAQ_UTILS_LIB}/libSlowControlDashboard.so
├── ConfigurePriority : 0
├── StartPriority : 0
├── StopPriority : 0
├── LinkToRegisterTable (GroupID = DEFAULT)-DISCONNECTED
├── MacroMakerSupervisor
├── CodeEditorSupervisor
```

Integration with State Machine

- **Alarm** propagation (from **EPICS**) and *otsdaq* state machine handling: Example on “Start” transition

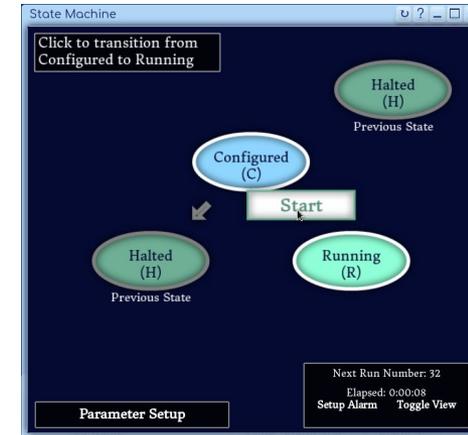
State Machine

Close Errors
Note: Newest messages are at the top.
(Press [ESC] to close and [SHIFT + ESC] to re-open)

```
:GatewaySupervisor:otsdaq/otsdaq/GatewaySupervisor/GatewaySupervisor.cc [1550]  
Received error from Supervisor instance =  
'ControlsDashboardSupervisor' [LID=282] in Context  
'mainContext' [URL=http://mu2edaq12.fnal.gov:3075].  
  
Error Message =  
:SlowControlsDashboardSupervisor:ControlsDashboardSuperv  
:otsdaq/otsdaq/CoreSupervisors/CoreSupervisorBase.cc [750]  
Error was caught while Starting:  
:EpicsInterface_slowcontrols.ccotsdaq_epics/otsdaq-epics/ControlsInterfacePlugins/EpicsInterface_slowcontrols.cc  
[1333]  
During 'start'... Alarms monitoring (count=1):  
dummyAlarm  
  
Found alarm for channel  
'Mu2e_TDAQ_shift_ExampleInterface0/dummyRegister0' =  
{time=1582678095, value=2020, status=HIHI, severity=MAJOR}  
  
Total alarms found = 1
```

Slow Controls Dashboard

PV Name	value	Alarm Stat...	Alarm S...
Mu2e_TDAQ_shift_ExampleInte...	2020	HIHI	MAJOR



Configure

```
...
- SlowControlsDashboardSupervisor
  - ApplicationGroupID : mainContextApps
  - SlowControlsDashboard
    - SlowControlsInterfacePluginType : EpicsInterface
    - SlowControlsChannelSourceTableList : DDCInterfaceTable
    - LinkToSlowControlsAlarmsMonitorTable-(GroupID) //Auto-generated from mock-up.
      - ConfigureAlarms
        - LinkToAlarmsMonitorTable-(GroupID = DEFAULT)-DISCONNECTED
        - LinkToPhaseAlarmsMonitorTable-(GroupID = DEFAULT)-DISCONNECTED
        - LinkToSumAlarmsMonitorTable-(GroupID = DEFAULT)-DISCONNECTED
        - LinkToStartAlarmsMonitorTable-(GroupID = ...) //Auto-generated from mock-up.
      - StartAlarms
        - dummyAlarm
          - AlarmGroupID : StartAlarms
          - AlarmChannelName : Mu2e_TDAQ_$(MU2E_OWNER)_ExampleInterface0/dummyRegister0
          - IgnoranceSeverity : False
        - LinkToTopAlarmsMonitorTable-(GroupID = DEFAULT)-DISCONNECTED
        - LinkToRunningAlarmsMonitorTable-(GroupID = DEFAULT)-DISCONNECTED
  - Status : On
  - Class : ota:SlowControlsDashboardSupervisor
  - Id : 282
  - Instance : 1
  - Network : local
  - Group : daq
  - Module : $(OTSDAQ_UTILITIES_LIB)/libSlowControlsDashboard.so
  - ConfigurePriority : 0
  - StartPriority : 0
  - StopPriority : 0
  - LinkToRegisterTable-(GroupID = DEFAULT)-DISCONNECTED
  - MacroMakerSupervisor
  - CodeEditorSupervisor
...

```

Slow Controls Monitoring in otstdaq

Examples

Editor

Slow Controls Dashboard

File EditMode

Mu2e_Weather_2/humi

Mu2e_Weather_2/temper: Status: NO_ALARM Severity: NO_ALARM

PV Name	value	Alarm
Mu2e_Weather_1/so...	5.0	W/r
Mu2e_Weather_1/te...	2	degC
Mu2e_Weather_1/te...	36.3	de
Mu2e_Weather_1/wi...	17	mph
Mu2e_Weather_2/b...	1016.0	M
Mu2e_Weather_2/rr...	0	inch

Mu2e_Weather_1/temperatur: Status: NO_ALARM Severity: NO_ALARM

Editor Panel

Choose your widgets:

Grid Color Background

Default Values

Name: Basic Root file viewer
Type: Root file

Example of widget settings window

Slow Controls Dashboard

LABEL

Parameter	Value
class	undefi
border	false
text	TDAQ
text_position	left
font	arial

value	Alarm ...	Alarm ...
degC	NO_ALARM	NO_ALARM
degF	NO_ALARM	NO_ALARM
mph7	NO_ALARM	NO_ALARM
mbar	NO_ALARM	NO_ALARM
inch	NO_ALARM	NO_ALARM

Slow Controls Dashboard

widget-0

Chose PV names

- Mu2e:TDAQ_hwdev_DTC0_BurstDataCount
- Mu2e:TDAQ_shift:ExampleInterface0:dummyR
- Mu2e:TDAQ_shift:ExampleInterface0:dummyR
- Mu2e:TDAQ_shift:ExampleInterface0:dummyR
- Mu2e:TDAQ_shift:ROC0:dummyRegister0

Add Remove

PV names chosen

- Mu2e:TDAQ_shift:ExampleInterface0:dummyR
- Mu2e:TDAQ_shift:ExampleInterface0:dummyR
- Mu2e:TDAQ_shift:ExampleInterface0:dummyR

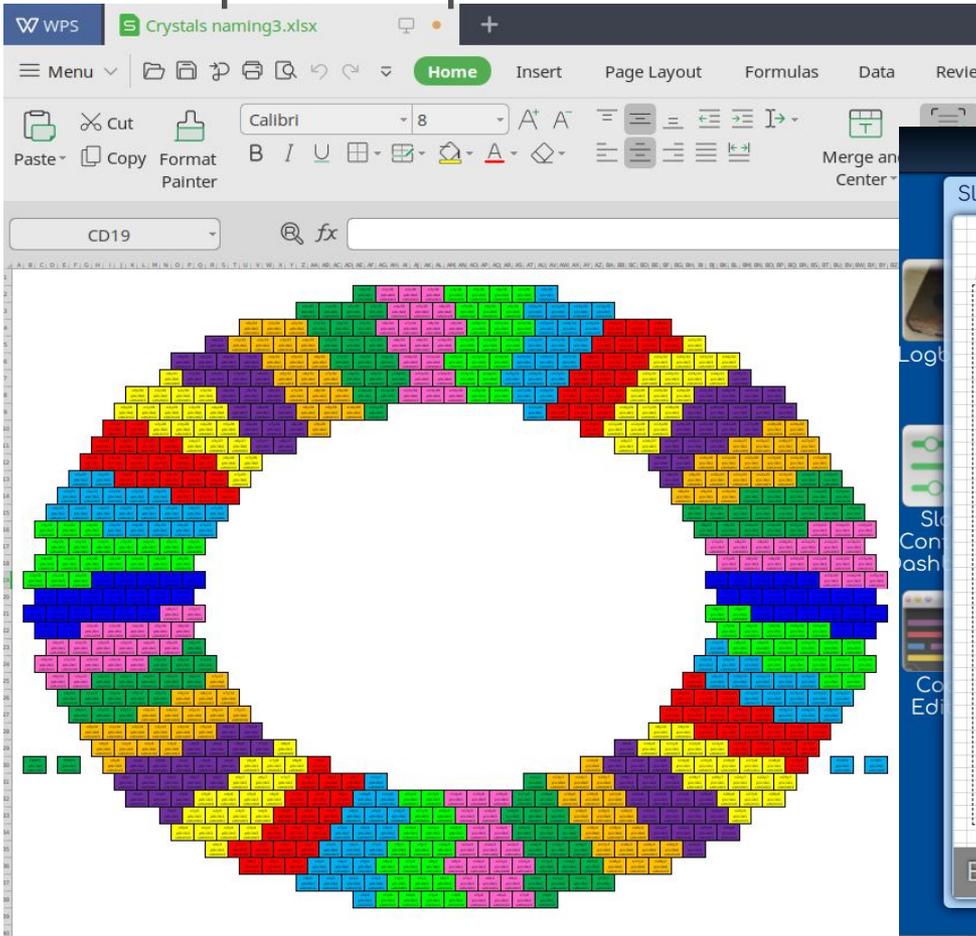
Save Cancel

Editor Panel

widget attributes editor

Calorimeter monitoring and the Slow Controls GUI

Examples: Import an xls file in a 2D-stop light widget



A screenshot of the 'Slow Controls Dashboard' GUI. The window title is 'Slow Controls Dashboard' and it includes a 'Welcome to Mu2e ops, Antonio' message. The main area shows a 'widget-0' with a table structure:

Edits PVs	Parameter	Value
Edit		
Attributes		

A dialog box is overlaid on the dashboard, containing the following text:

Please choose a CSV formatted data file (i.e. commas for columns, and new lines for rows) to upload, and choose whether you want to replace the current data or prepend/append the new data:

No file chosen

At the bottom of the dashboard, there is an 'Editor Panel'.

Slow Controls alarm notification by System Message

System message alarm notification example

The screenshot displays the 'Slow Controls Dashboard' interface. A system message notification is shown in a blue box, indicating a 'Slow Control Alarm Notification' for the PV 'Mu2e_TDAQ_shift_ExampleInterface0/dummyRegister0' with a 'MAJOR' severity. The notification also includes the time 'Mon Mar 30 13:01:07 2020' and the value '1233 stouts: HIHI'. A 'Dismiss' button is visible at the bottom right of the notification box.

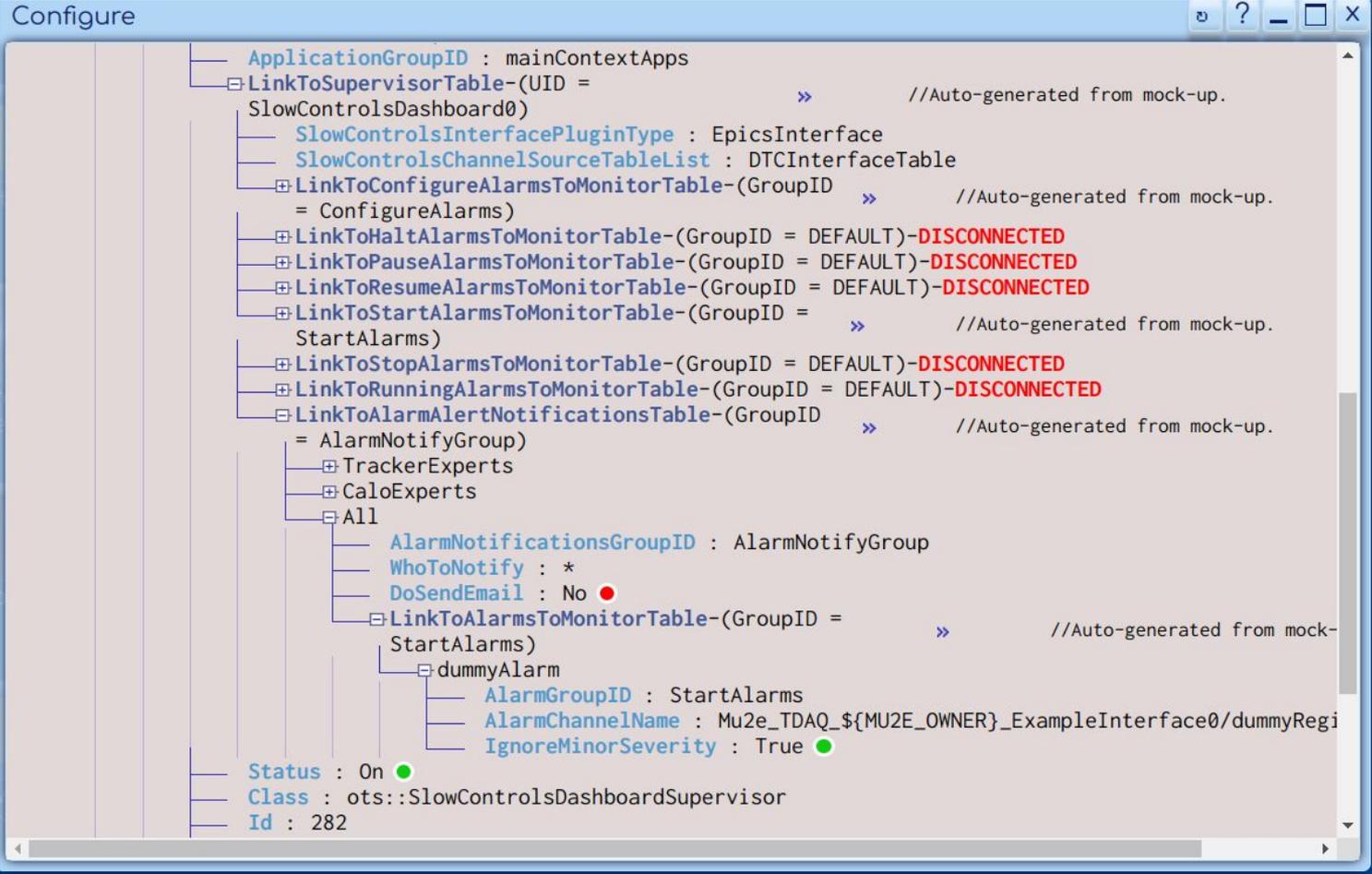
The dashboard also features a table with the following data:

PV Name	Alarm Status	Alarm Severity	Last Update
Mu2e_TDAQ_shift_ExampleInterface0/dummyRegister0	HIHI	MAJOR	03/30/20

The dashboard includes a sidebar with icons for 'Config Subsets', 'Macro Maker', 'DB Utilities', 'Console', 'System Status', and 'Slow Control'. The top navigation bar contains buttons for 'Layouts', 'Tile', 'Show Desktop', and 'Full Screen', along with a 'Welcome to Mu2e ops, Antonio' message and 'Sign out' options.

Slow Controls alarm notification by System Message

Configured by specific table in otsdaq:



```
Configure
├── ApplicationGroupID : mainContextApps
├── LinkToSupervisorTable-(UID = SlowControlsDashboard0) //Auto-generated from mock-up.
│   ├── SlowControlsInterfacePluginType : EpicsInterface
│   ├── SlowControlsChannelSourceTableList : DTCInterfaceTable
│   ├── LinkToConfigureAlarmsToMonitorTable-(GroupID = ConfigureAlarms) //Auto-generated from mock-up.
│   ├── LinkToHaltAlarmsToMonitorTable-(GroupID = DEFAULT)-DISCONNECTED
│   ├── LinkToPauseAlarmsToMonitorTable-(GroupID = DEFAULT)-DISCONNECTED
│   ├── LinkToResumeAlarmsToMonitorTable-(GroupID = DEFAULT)-DISCONNECTED
│   ├── LinkToStartAlarmsToMonitorTable-(GroupID = StartAlarms) //Auto-generated from mock-up.
│   ├── LinkToStopAlarmsToMonitorTable-(GroupID = DEFAULT)-DISCONNECTED
│   ├── LinkToRunningAlarmsToMonitorTable-(GroupID = DEFAULT)-DISCONNECTED
│   └── LinkToAlarmAlertNotificationsTable-(GroupID = AlarmNotifyGroup) //Auto-generated from mock-up.
│       ├── TrackerExperts
│       ├── CaloExperts
│       └── All
│           ├── AlarmNotificationsGroupID : AlarmNotifyGroup
│           ├── WhoToNotify : *
│           ├── DoSendEmail : No ●
│           └── LinkToAlarmsToMonitorTable-(GroupID = StartAlarms) //Auto-generated from mock-up.
│               └── dummyAlarm
│                   ├── AlarmGroupID : StartAlarms
│                   ├── AlarmChannelName : Mu2e_TDAQ_${MU2E_OWNER}_ExampleInterface0/dummyRegi
│                   └── IgnoreMinorSeverity : True ●
├── Status : On ●
├── Class : ots::SlowControlsDashboardSupervisor
└── Id : 282
```