Mu2e Online DAQ and Slow Control interface

Antonio Gioiosa

Università di Pisa, INFN Pisa

SIF 106° Congresso Nazionale

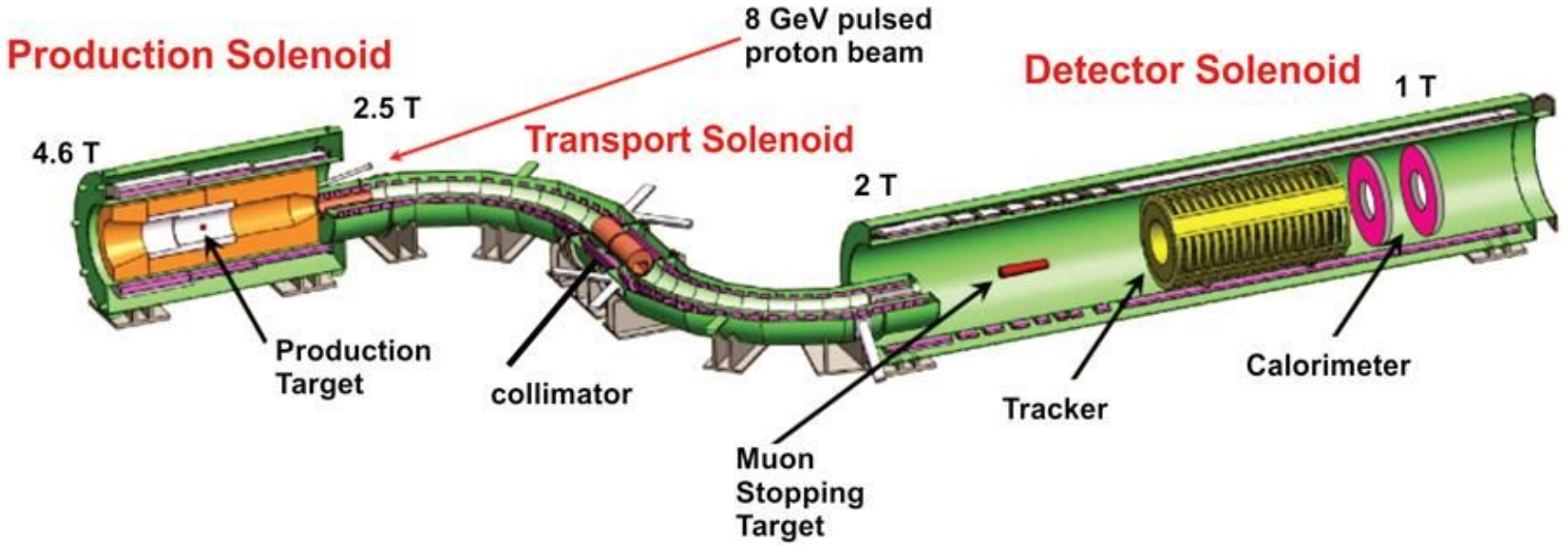
September 14, 2020

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

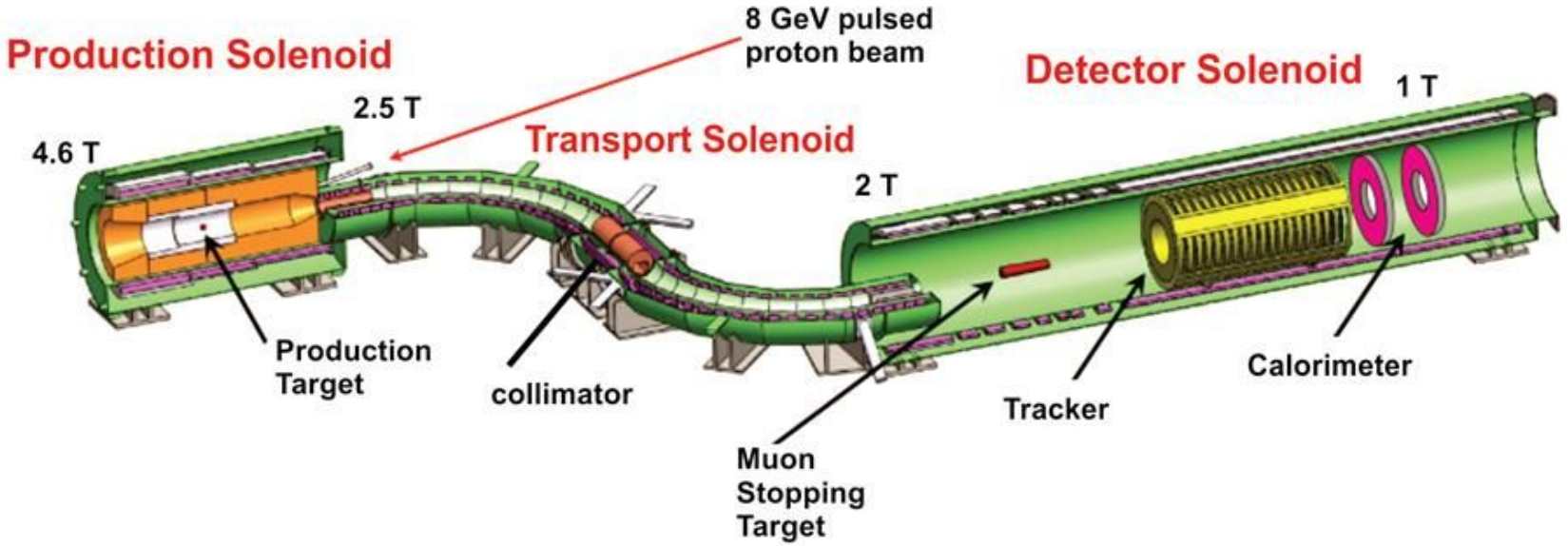
With the expected experimental sensitivity, Mu2e will improve the SINDRUM II limit ($7.0 \cdot 10^{-13}$) of four orders of magnitude

(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)



- A pulsed proton beam (from Fermilab's accelerator complex) hits the production target to produce pions which decay into muons
- The muons get transported via the transport solenoid to the detector solenoid where they get stopped at the aluminum stopping target
- If conversion electrons are produced in the stopping target, they will move through the tracker and calorimeter where they can be measured

Mu2e Online DAQ and Slow Control interface

Summary:

- Online DAQ (***otsdaq***) overview
- Slow Controls connection and EPICS plugin development in ***otsdaq***
- Slow Controls Monitoring in ***otsdaq***
  - Monitoring and Slow Controls GUI
- Slow Controls Alarm alerts and messages
- Slow Controls Integration with ***otsdaq*** State Machine and Alarm handling
  - Integration of ***otsdaq*** front-end DAQ hardware Process Variables (PVs) with EPICS (*DTC/ROCs/CFO*)
  - Integration of online data processing *output metrics* with EPICS
- Conclusion

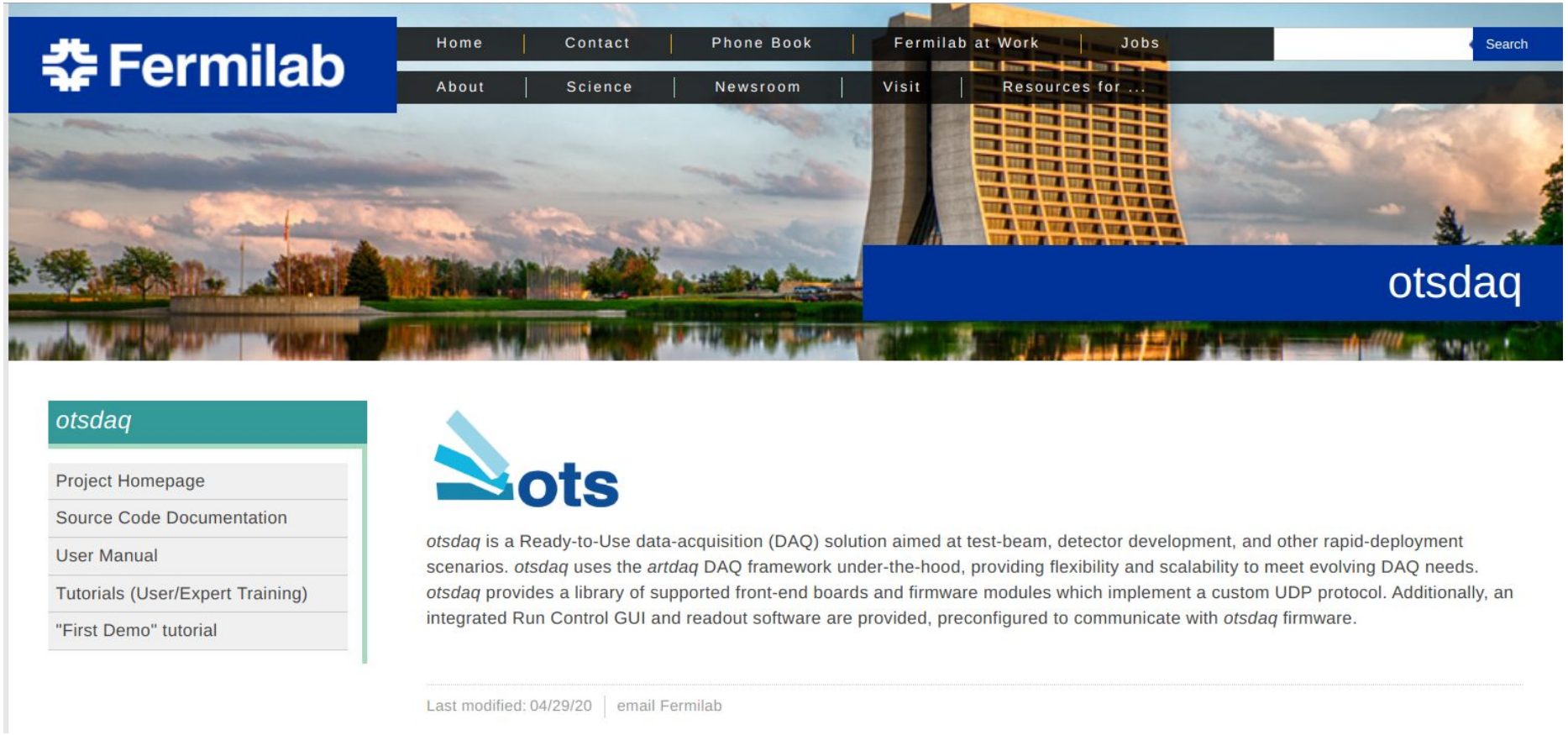
Mu2e Online DAQ solution: *otsdaq*

otsdaq overview

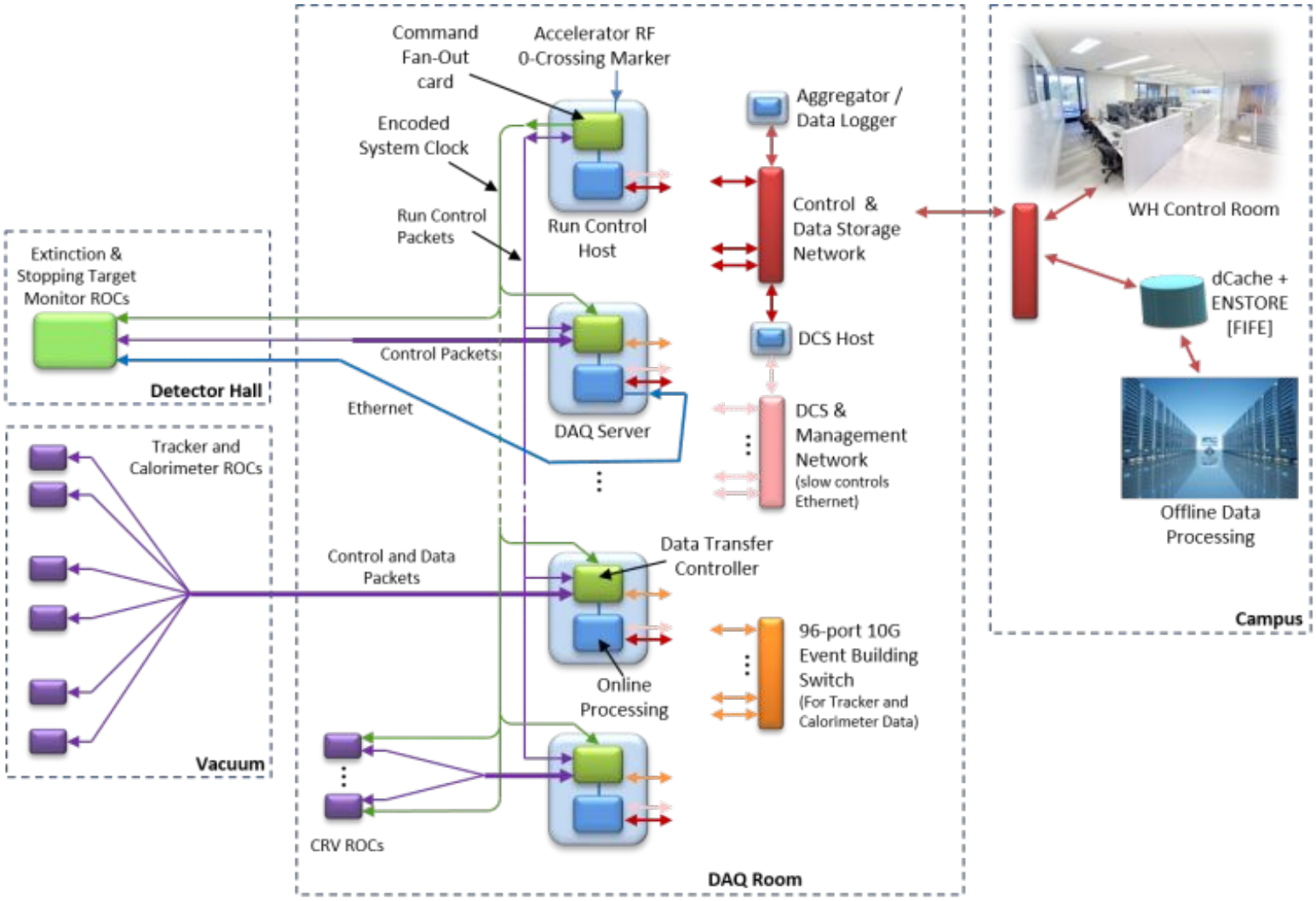
- ***otsdaq*** is a Ready-to-Use data-acquisition (DAQ) solution aimed at test-beam, detector development, and other rapid-deployment scenarios
- it uses the ***artdaq*** DAQ framework under-the-hood, providing flexibility and scalability to meet evolving DAQ needs
- ***otsdaq*** provides a library of supported front-end boards and firmware modules which implement a custom UDP protocol
- 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/

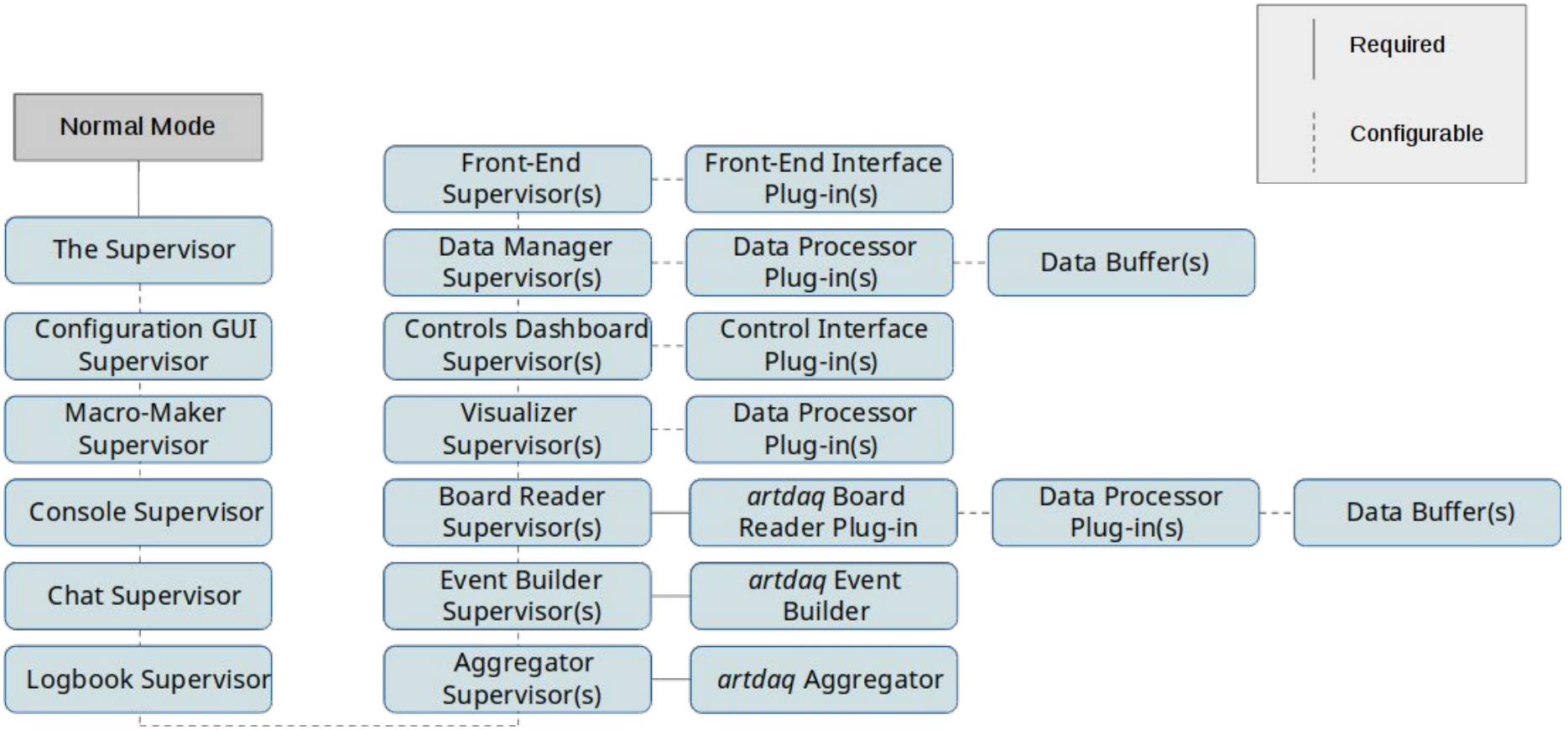


Mu2e TDAQ components Diagram



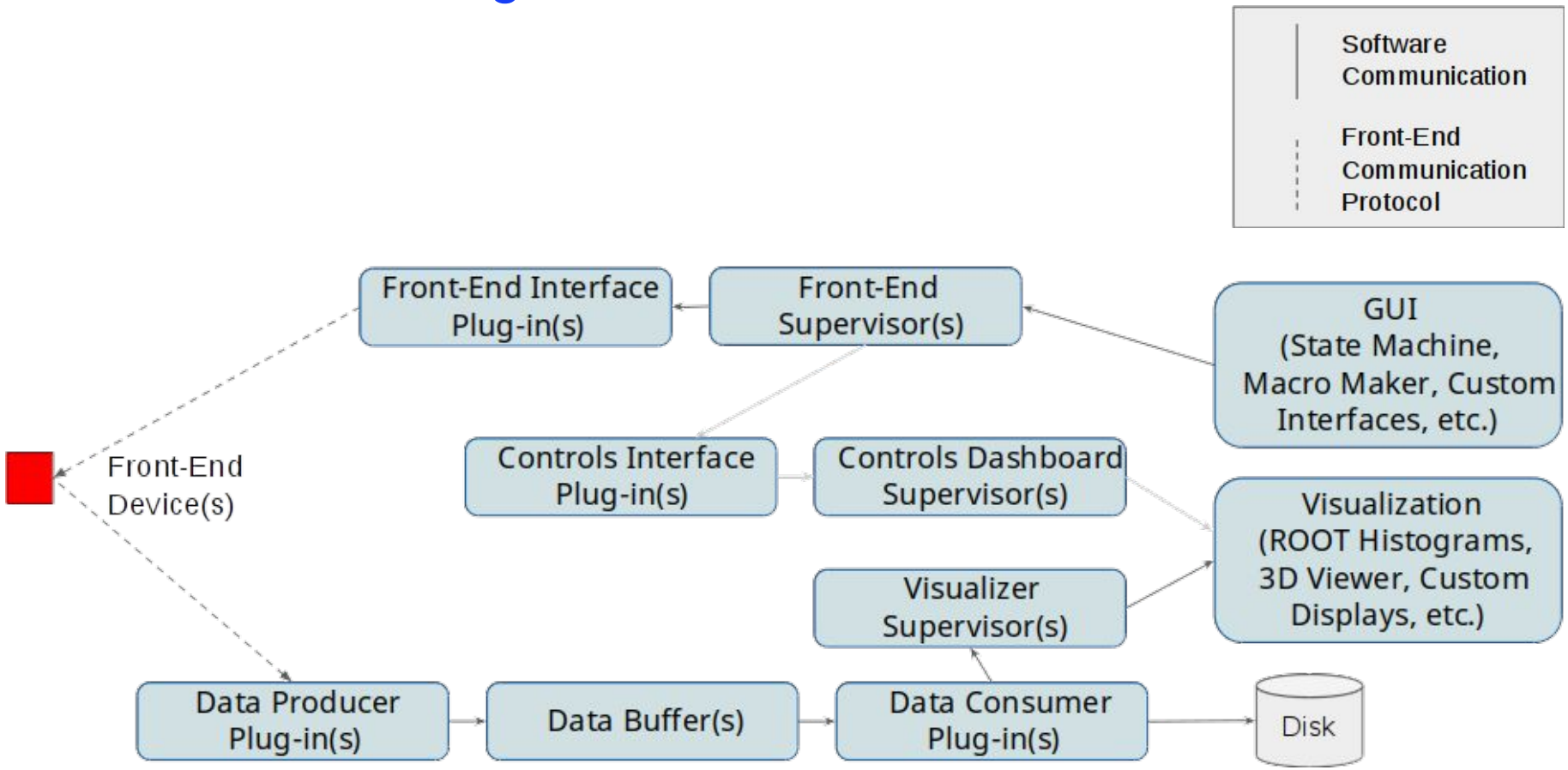
otsdaq overview

Components



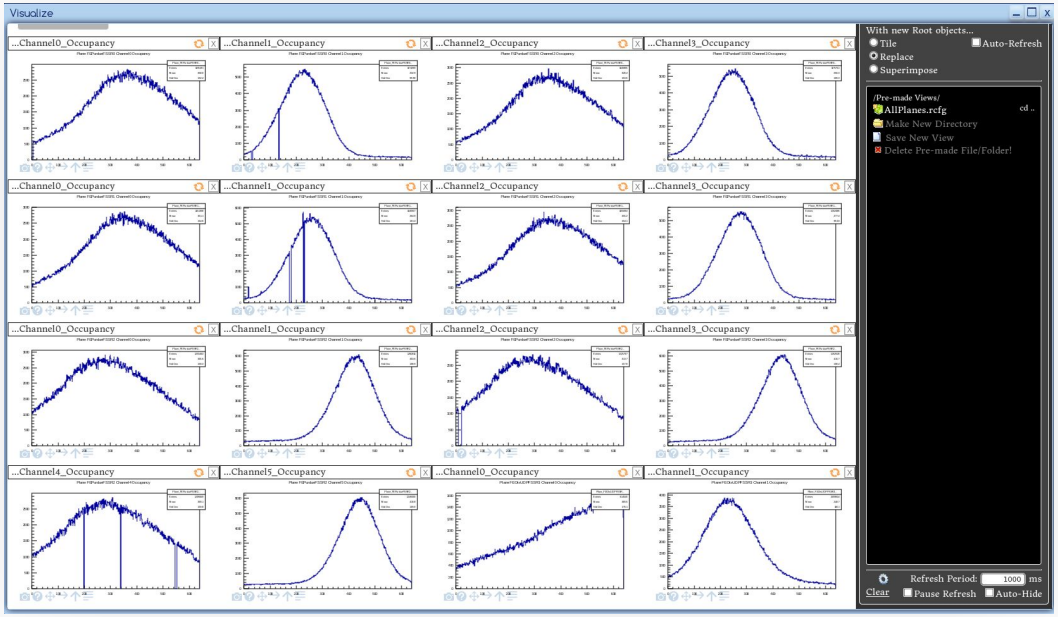
otsdaq overview

Data Flow Block Diagram



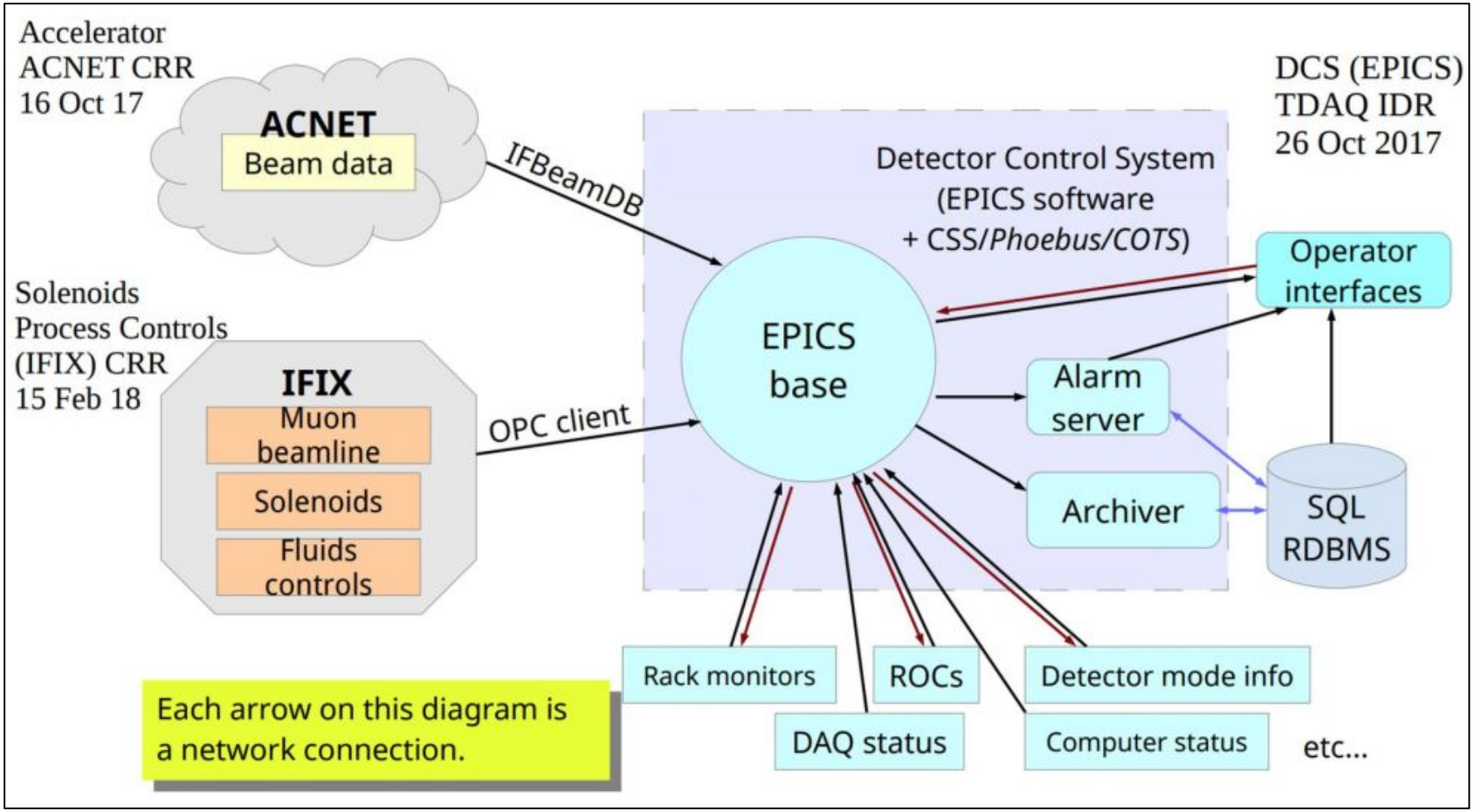
Data processing: Page example

- Data processing is the primary responsibility of the online DAQ. Mu2e's event window data will be processed through artdaq modules
- Data processor plugins is 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

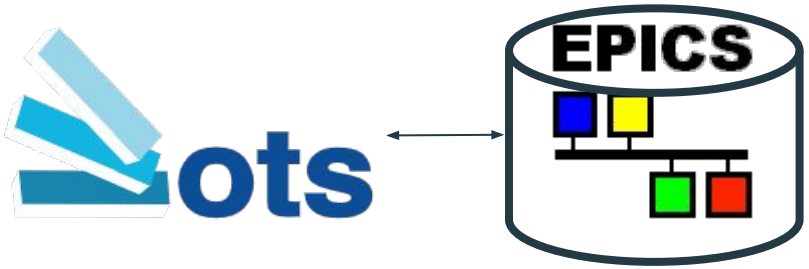
Experimental Physics and Industrial Control System



Slow Controls connection and EPICS plugin development in otsdaq

Channel subscription to EPICS

- 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
  - *dcs_archiver*
  - *dcs_alarm*
  - *dcs_log*



Slow Controls Monitoring in otsdaq

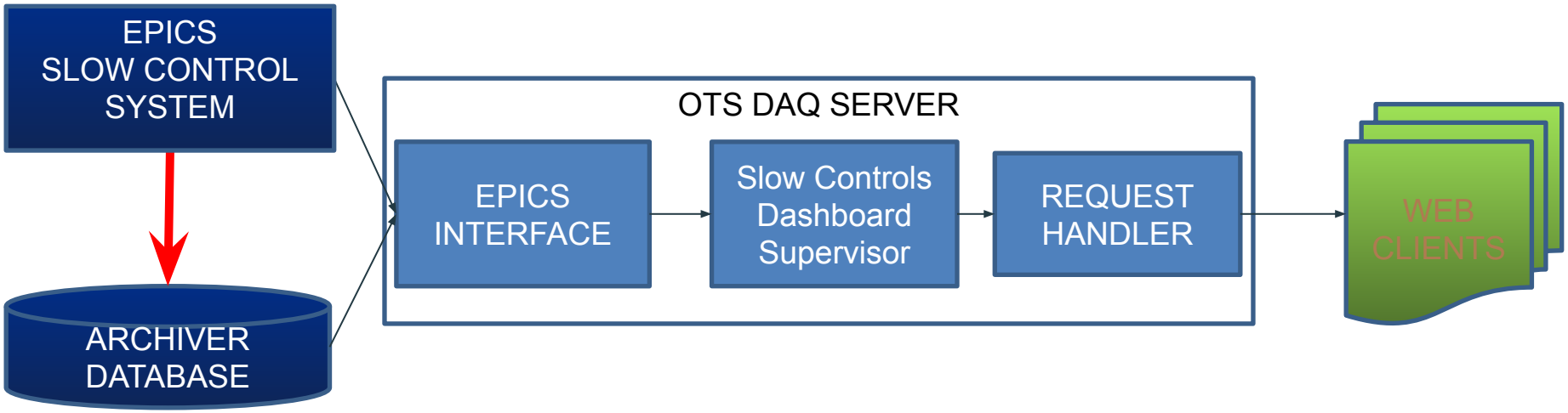
Slow Controls Software purpose

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

  Alarms, Warnings, Readouts, Timestamps, Status

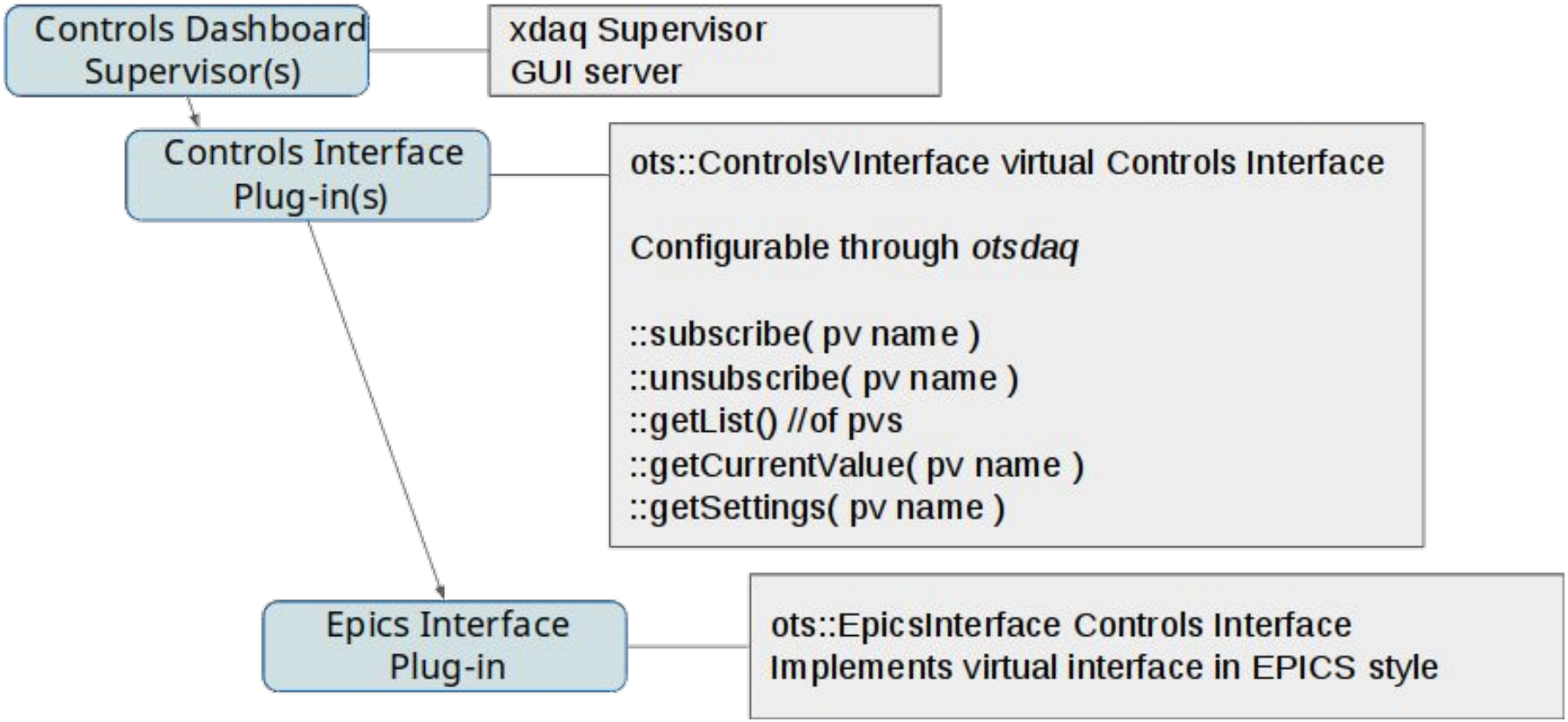
- Interact through a web interface that is:

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



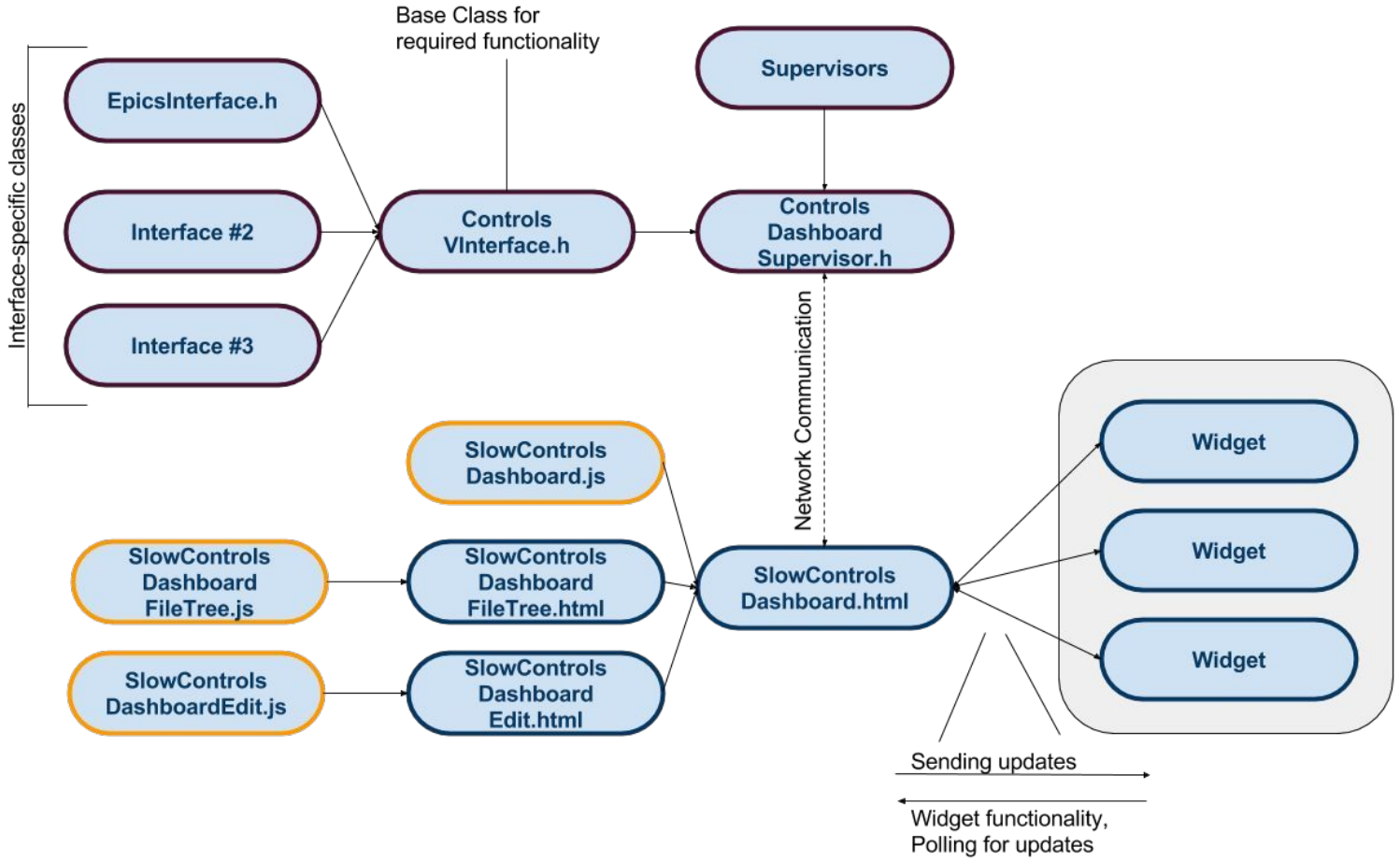
Slow Controls Monitoring in otsdaq

Slow Controls C++ Hierarchy



Slow Controls Monitoring in otsdaq

Slow Controls GUI Hierarchy

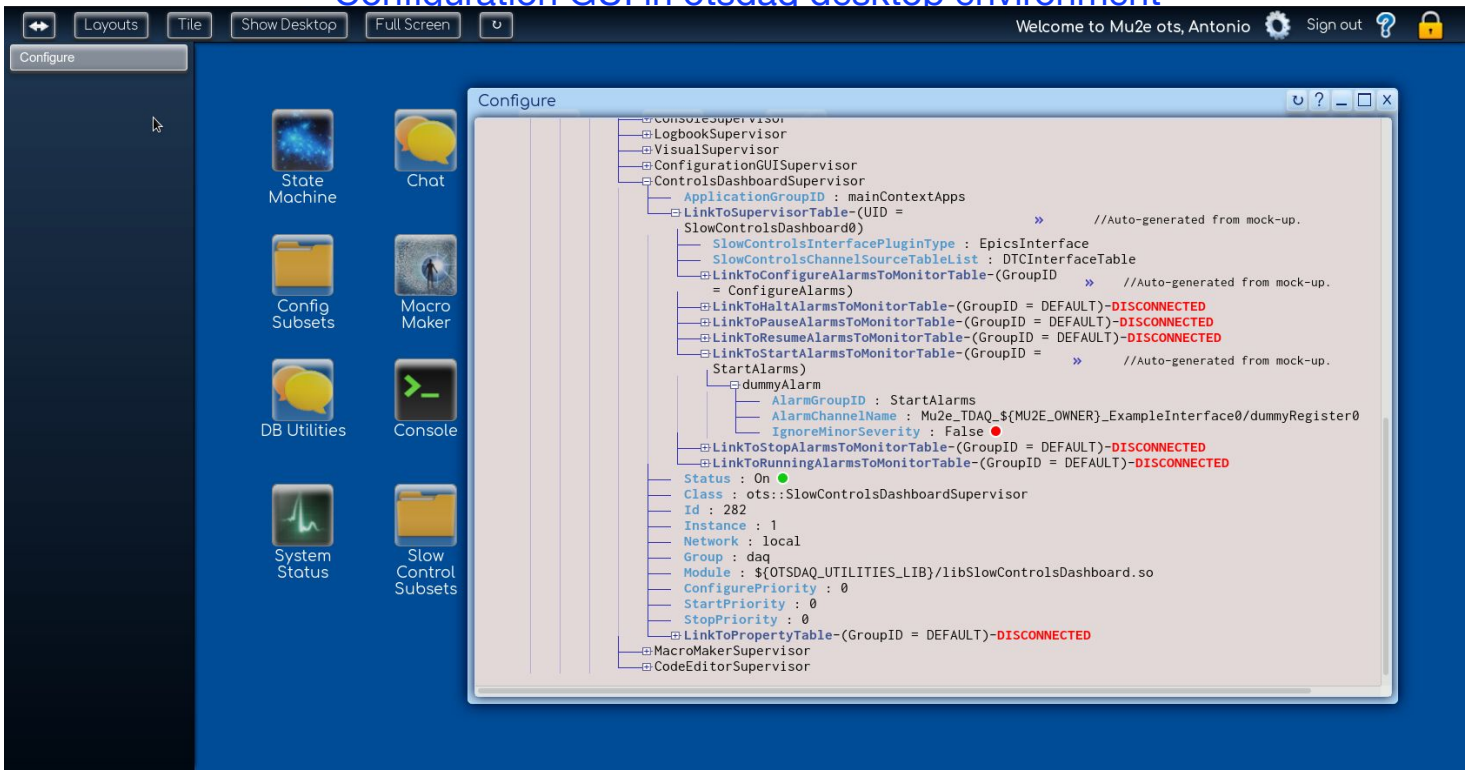


Slow Controls Monitoring in otsdaq

Configuring by specific tables in otsdaq

DesktopIconTable, XDAQApplicationPropertyTable, XDAQApplicationTable, XDAQContextTable

Configuration GUI in otsdaq desktop environment



Slow Controls Monitoring in otsdaq

Basic Widget Mechanics

- All widgets have six required methods:

  init(), getParameters(), setParameters(), setupPVs(), newWidget(), and newValue()

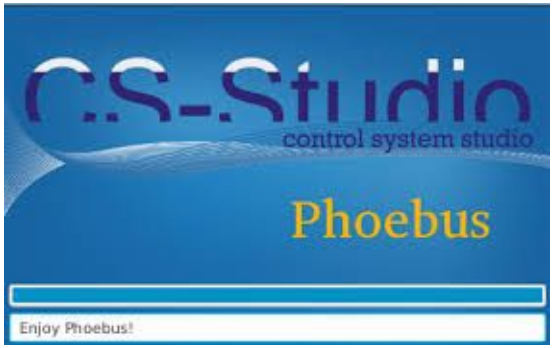
  (see also Control System overview Mu2e Document 9082-v1)

Widget properties

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

Load and save dashborad page in XML

Cs-Studio Phoebus compatible format

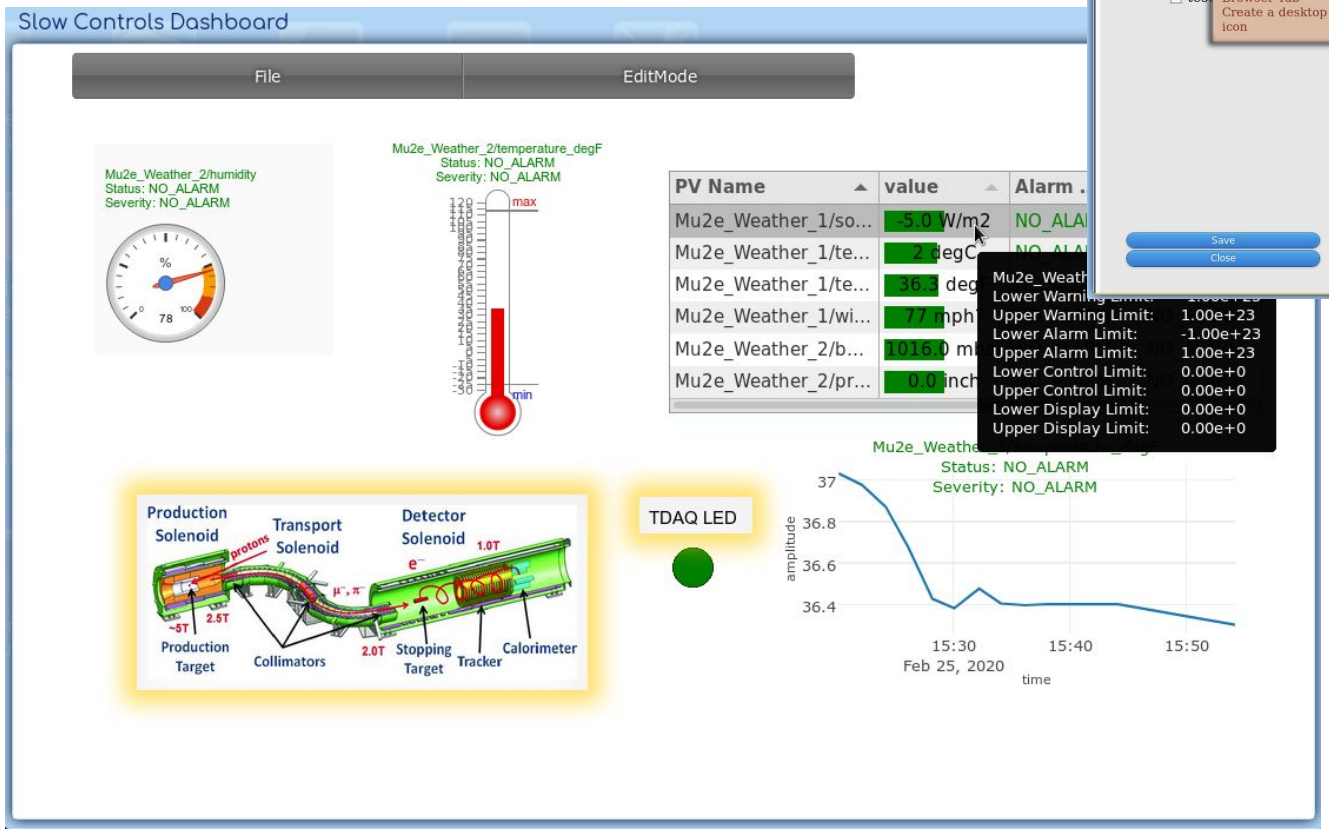


Slow Controls Monitoring in otsdaq

Examples

Example of page loading

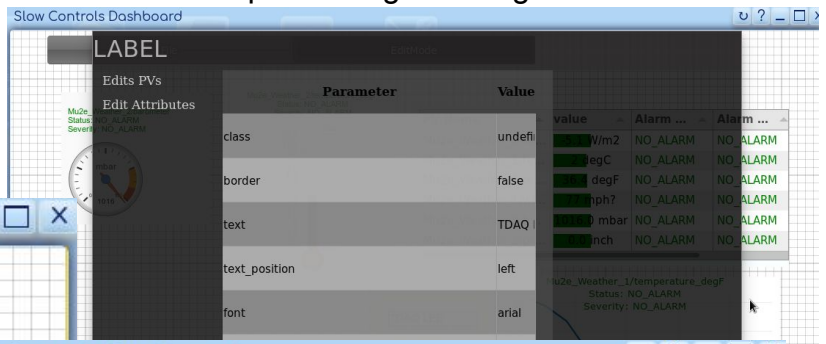
Example of loaded page



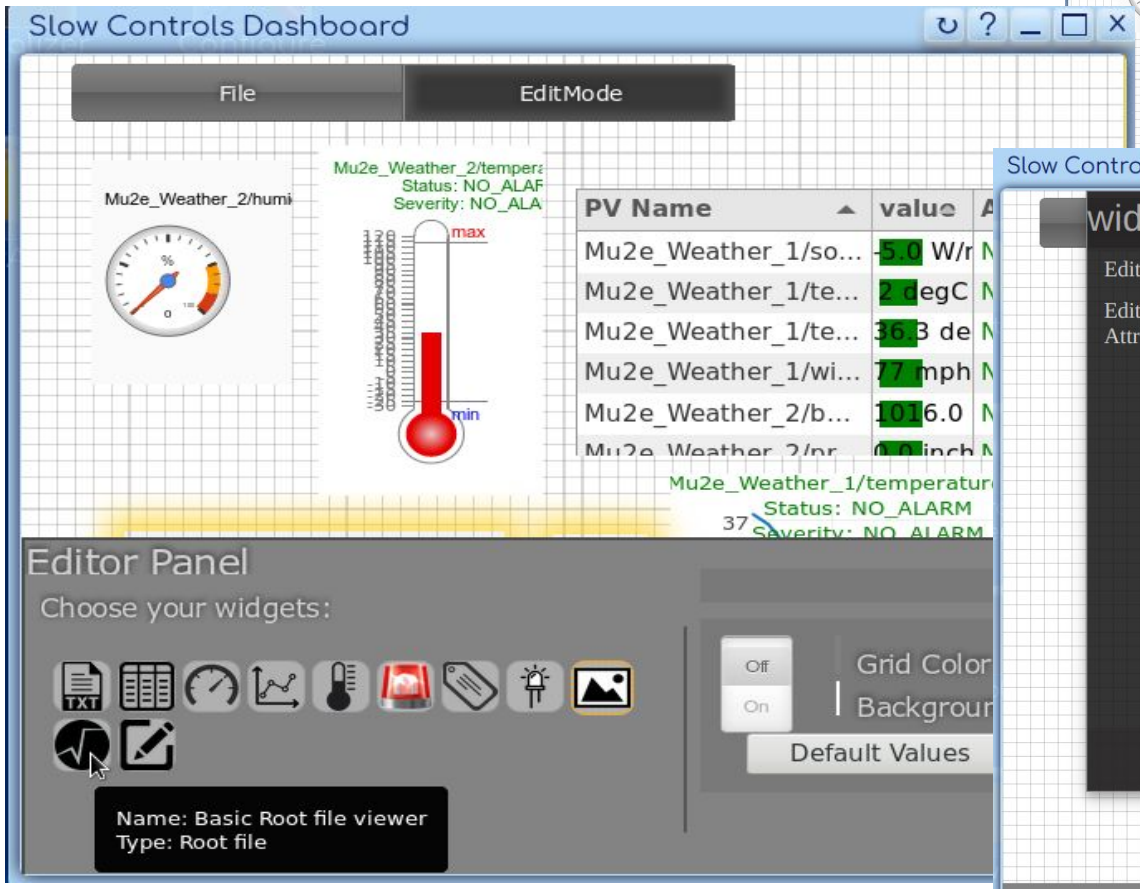
Slow Controls Monitoring in otsdaq

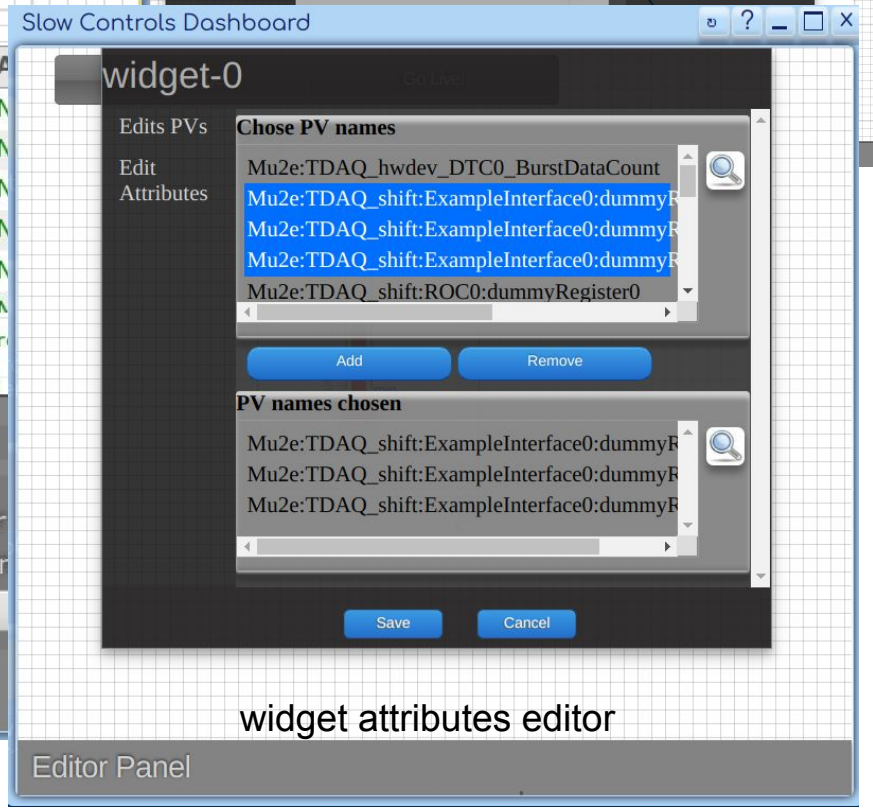
Examples

Example of widget settings window



Editor

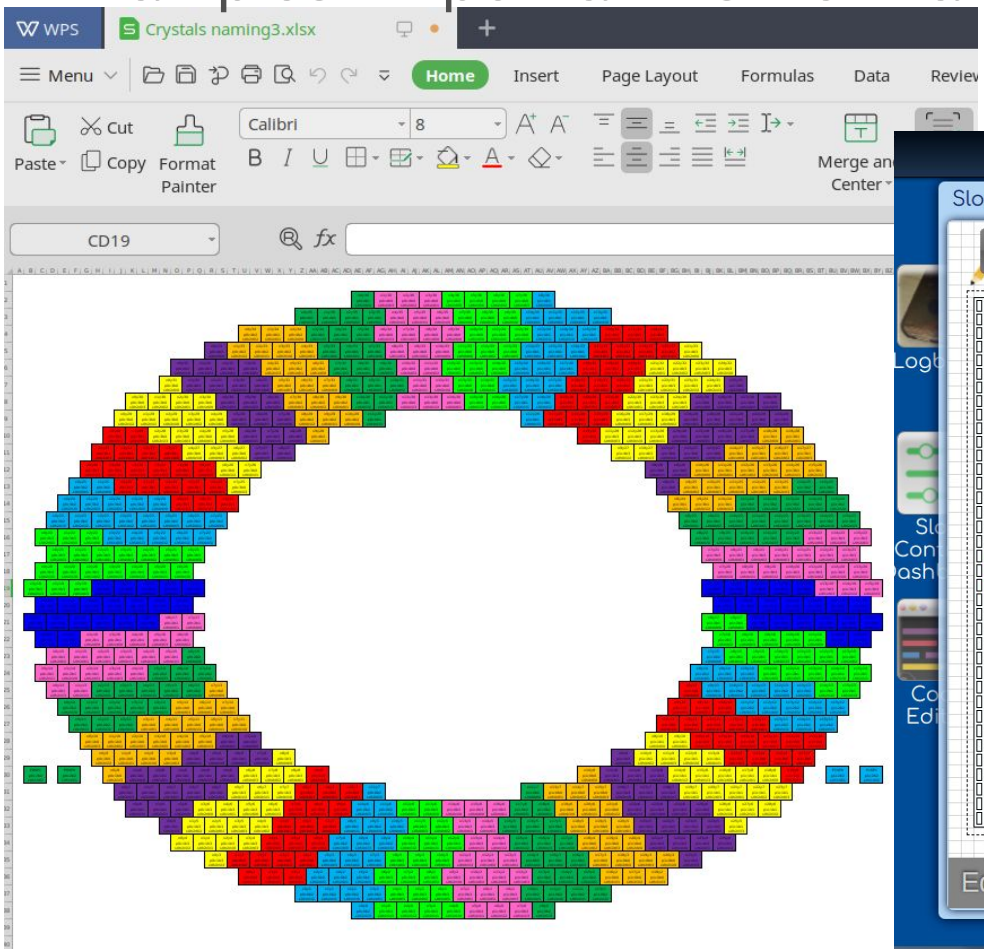


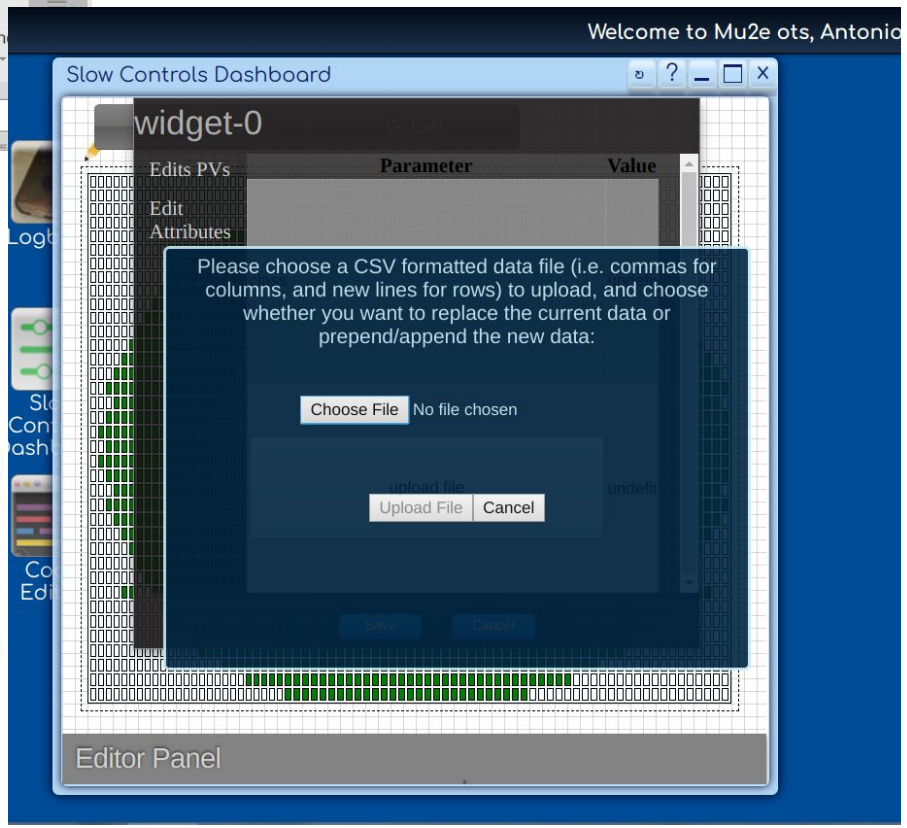


widget attributes editor

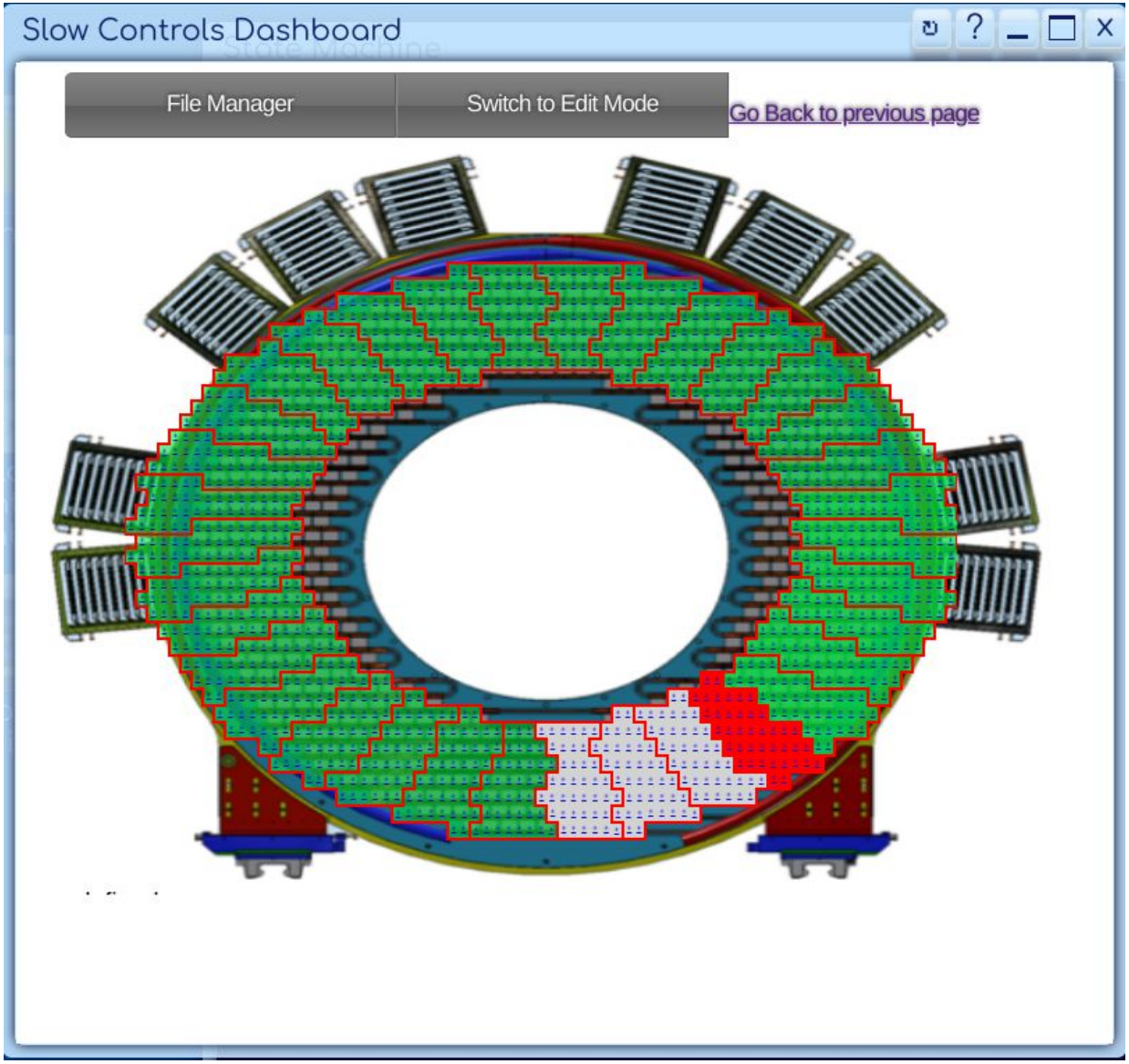
Calorimeter monitoring and the Slow Controls GUI

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



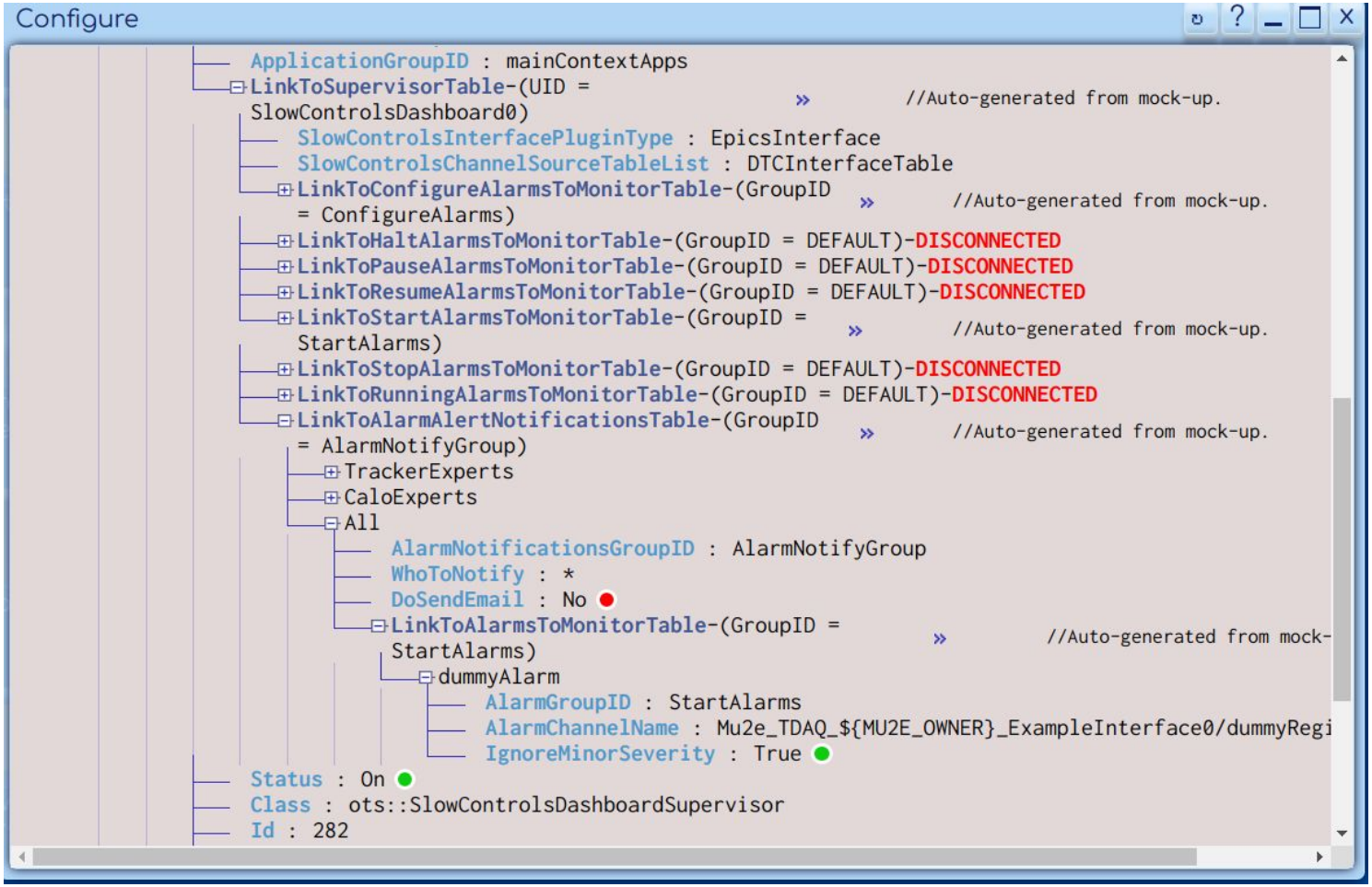


Calorimeter monitoring and the Slow Controls GUI



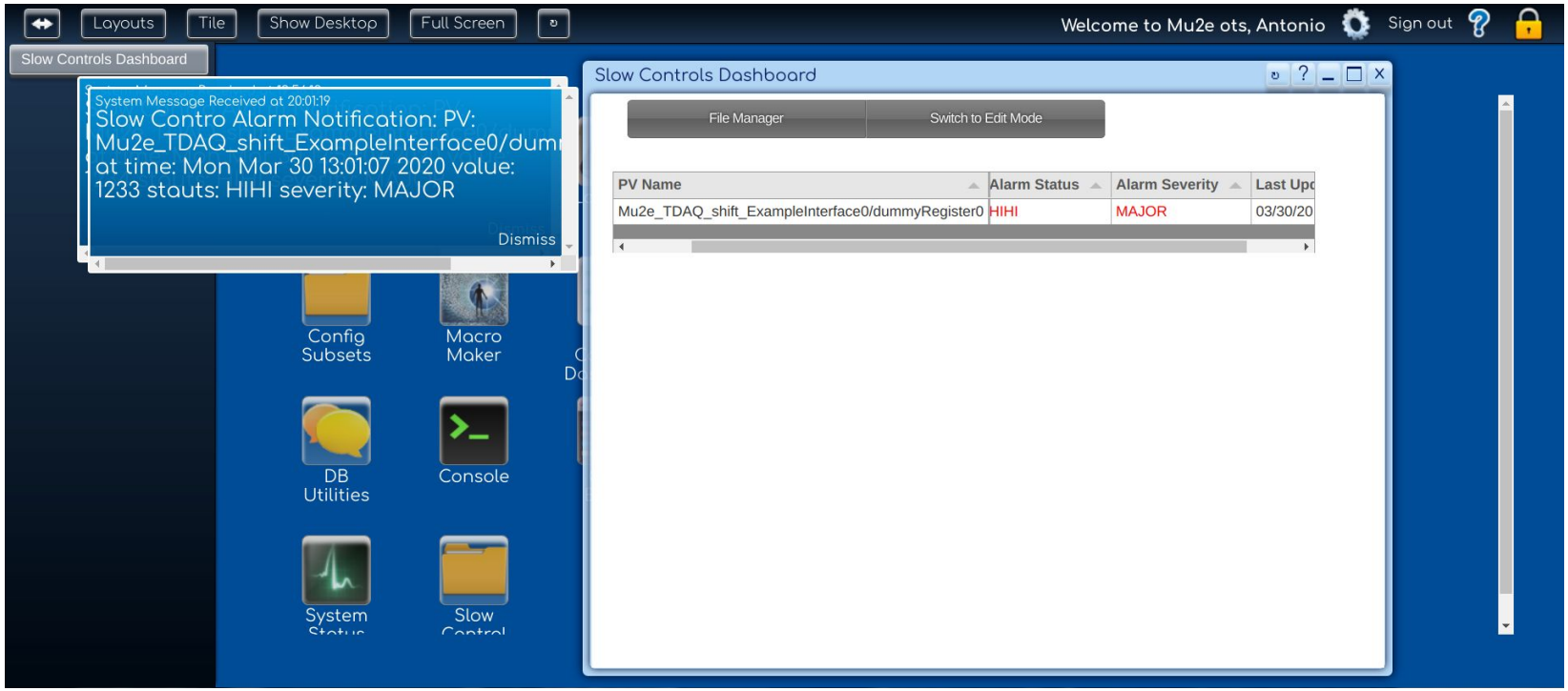
Slow Controls alarm notification by System Message

Configured by specific table in otsdaq:



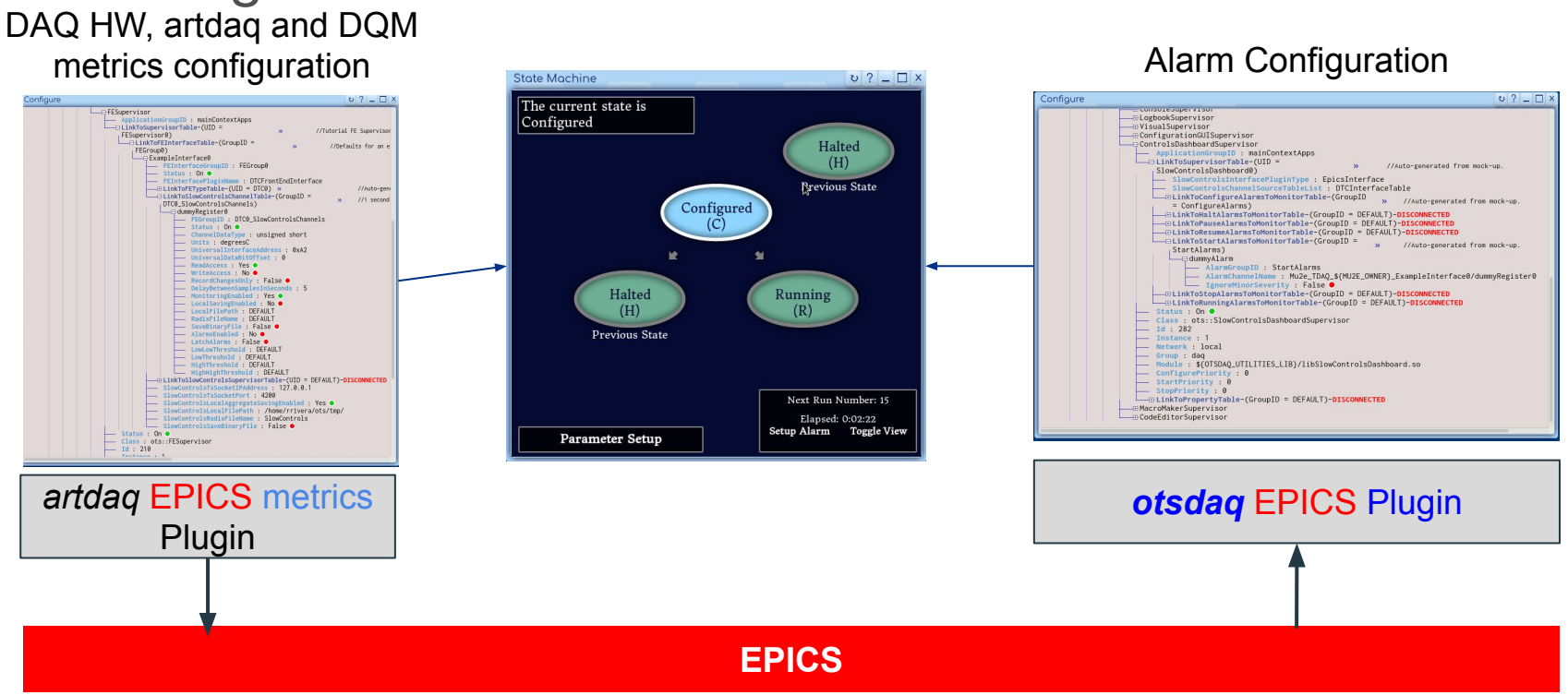
Slow Controls alarm notification by System Message

System message alarm notification example



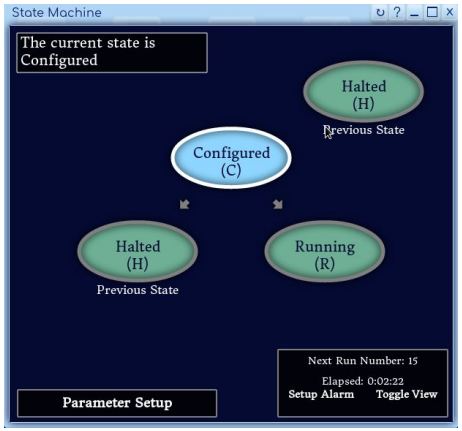
Integration with State Machine

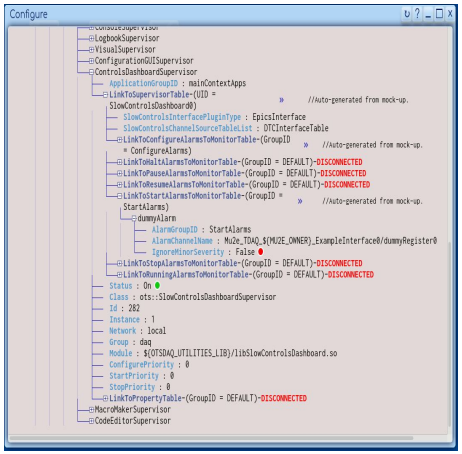
- State Machine Configuration and data subscription to EPICS
- Alarm propagation (from EPICS) and *otsdaq* State Machine handling



Integration with State Machine

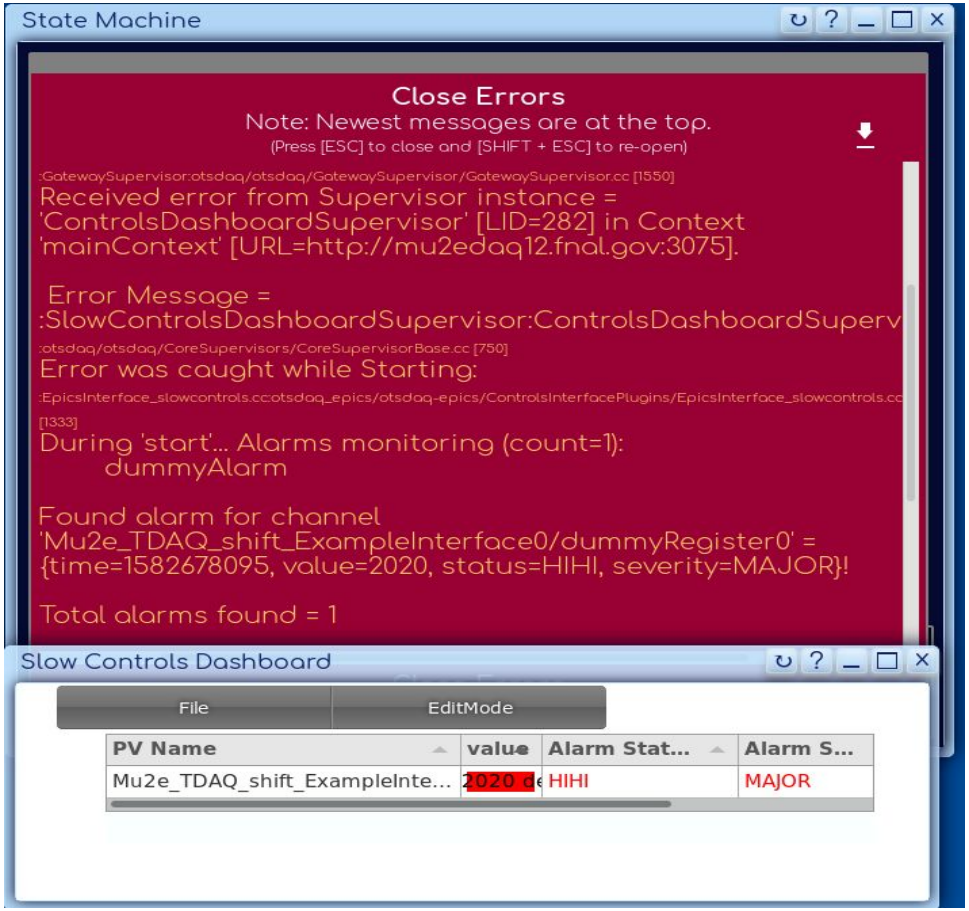
- Alarm propagation (from EPICS) and *otsdaq* state machine handling is available: needs just to identify which PV alarms, *status* and *severity* will be propagated
- *Tables and parameters designed for configuration*
  - SupervisorTable parameters:
    - *Slow Controls Interface Plugin Type*
    - *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*
    - *Running*

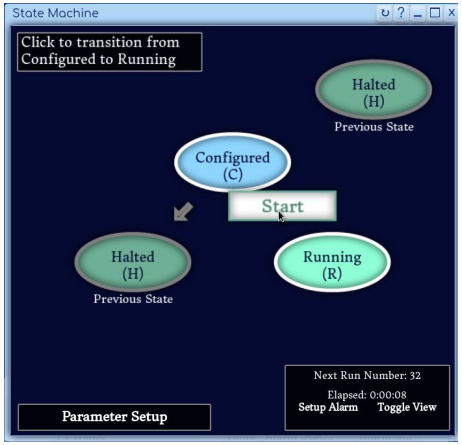


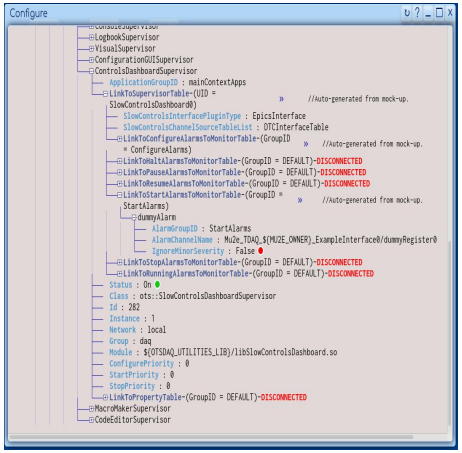


Integration with State Machine

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







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

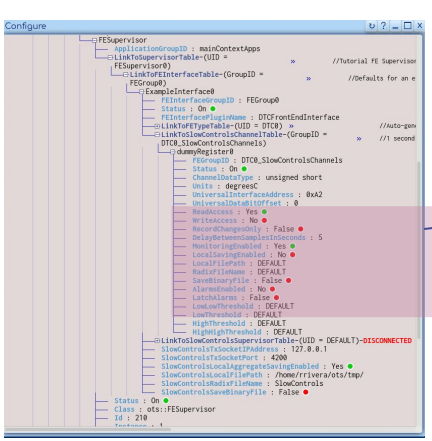
Actions designed and developed in *otsdaq*

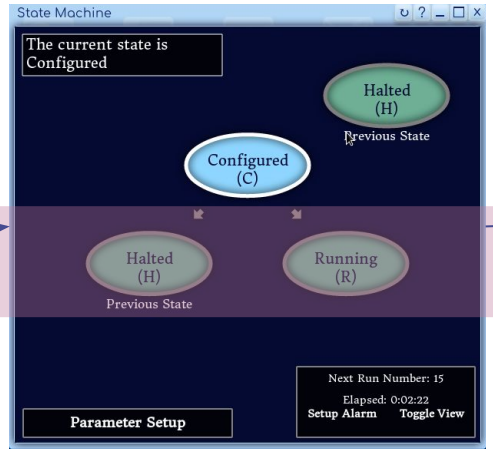
1. *otsdaq* DCS channels Front End and tables configuration
2. *otsdaq* State Machine configuration implementation
3. add/update channels info for IOC and Archiver DB
4. software IOC restarting
5. EPICS Archiver restarting
6. new *otsdaq* epics_plugin channels subscriptions to EPICS
7. Sending configured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

   otsdaq DCS channels new values → artdaq Metric Manager → software IOC → EPICS → otsdaq DCS GUI

Integration with State Machine

- *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





otsdaq EPICS interface

EPICS ARCHIVER DATABASE

OTS IOC

EPICS SERVER

EPICS
- ARCHIVER
- ALARM
- LOG

DATABASES

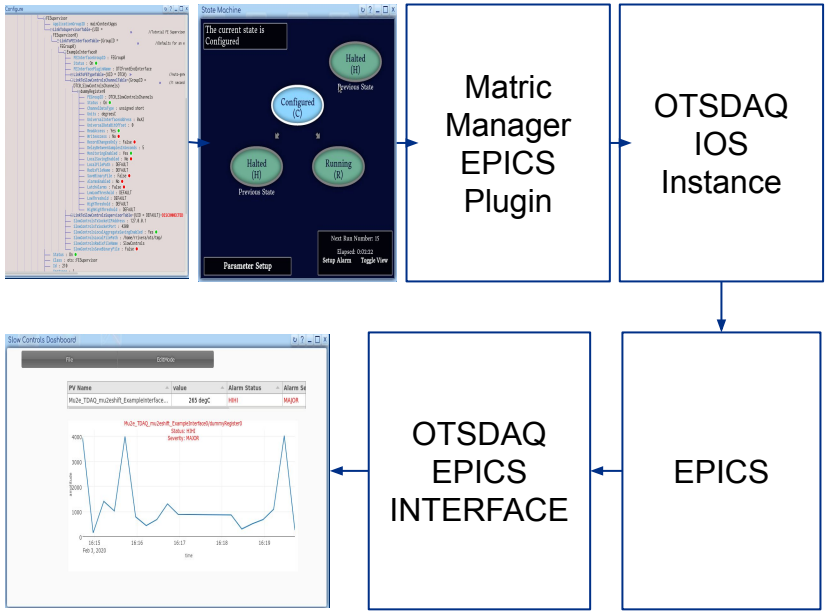
otsdaq EPICS interface:
Channel Access subscription for PVs Values, Settings, Status, Severity

retrieving:
PVs List and history
Last Alarms and Alams Log

Monitoring

Online data processing output metrics and EPICS

- *artdaq* tracks a large number of metrics about event rate and dataflow, which can be enabled at the metric plugin level
- the user can send a subset of metrics to EPICS



Artdaq Builtin Metric Inventory

Notes: %(RANK) will be replaced with appropriate ranks. Some plugins may perform transformations on these names, for example, graphite_metric replaces spaces with underscores.

artdaq_utilities

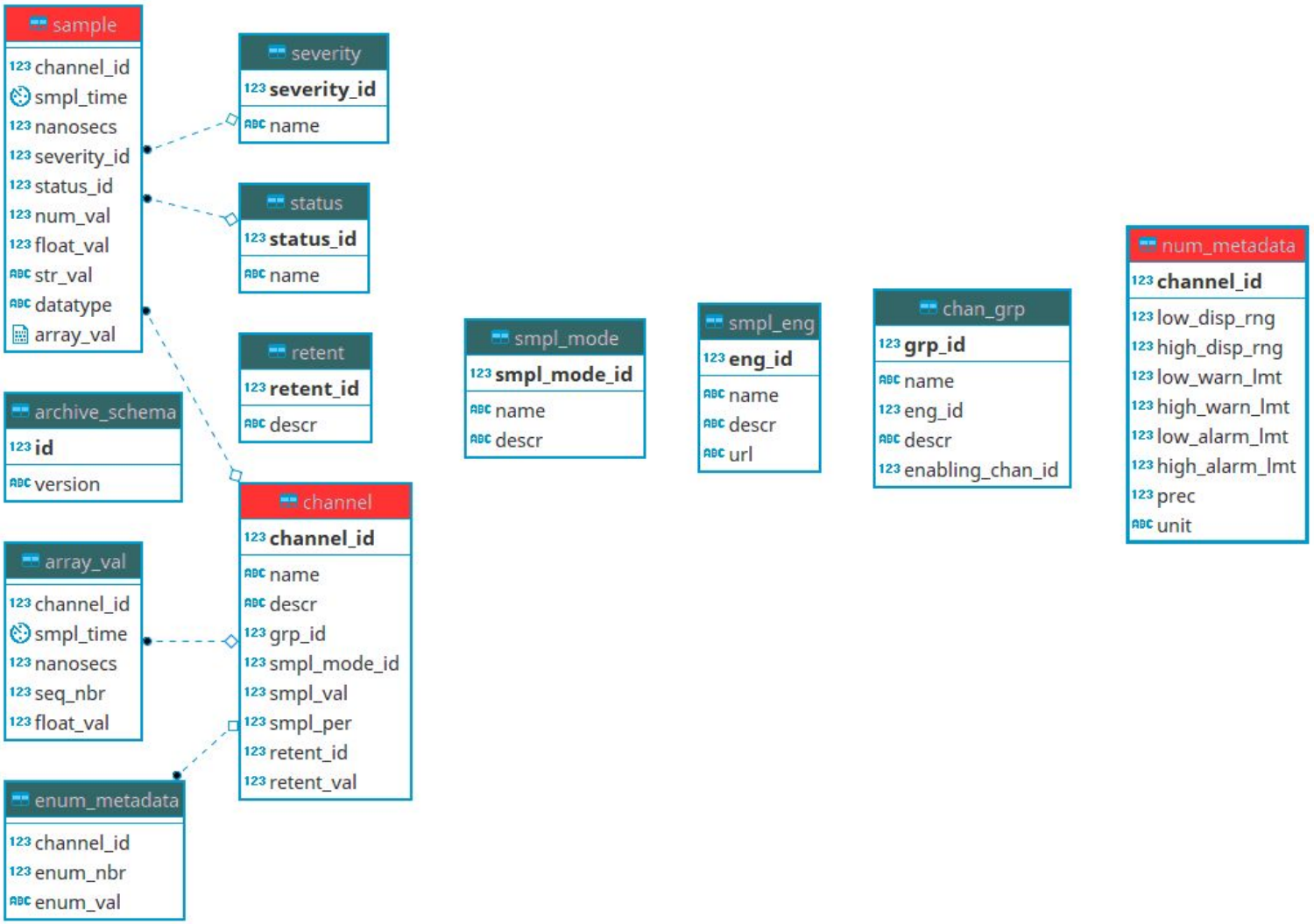
Name	Level	Units	Mode(s)	Source Class	Notes
Metric Calls	4	metrics	AccumulateAndRate	MetricManager	Tracks the number of sendMetric calls since the last time the metric dispatch thread ran
Missed Metric Calls	4	metrics	AccumulateAndRate	MetricManager	Number of discarded sendMetric calls due to queue overflow

artdaq

Name	Level	Units	Mode(s)	Source Class	Notes
Fragment Count	1	fragments	LastPoint	BoardReaderCore	
Fragment Rate	1	fragments/sec	Average	BoardReaderCore	
Average Fragment Size	2	bytes/fragment	Average	BoardReaderCore	
Data Rate	2	bytes/sec	Average	BoardReaderCore	
Avg Input Wait Time	3	seconds/fragment	Average	BoardReaderCore	Amount of time spent in CommandableFragmentGenerator::getNext
Avg BoardReader Sync Wait Time	3	seconds/fragment	Average	BoardReaderCore	Currently unused (always 0)
Avg Output Wait Time	3	seconds/fragment	Average	BoardReaderCore	Amount of time spent in DataSenderManager::sendFragment
Avg Frags Per Read	4	fragments/read	Average	BoardReaderCore	
DAQ Transition Time	4	s	Accumulate	Commandable	Records the amount of time spent in transitions
Average Sender Acknowledgement Time	3	seconds	Average	RoutingMasterCore	
Avg Table Acknowledge Time	3	seconds	Average	RoutingMasterCore	
Receiver Token Rate	1	updates/sec	Average	RoutingMasterCore	
Table Update Rate	1	updates/sec	Average	RoutingMasterCore	
Total Receiver Token Wait Time	3	seconds	Average	RoutingMasterCore	
Receiver Token Count	1	updates	LastPoint	RoutingMasterCore	
Table Update Count	1	updates	LastPoint	RoutingMasterCore	
bytesRead	3	B	LastPoint	SharedMemoryReader	
Avg Input Wait Time	3	s	Average	SharedMemoryReader	
Avg Processing Time	2	s	Average	SharedMemoryReader	
Avg Read Time	3	s	Average	SharedMemoryReader	
queue%Used	5	%	LastPoint	SharedMemoryReader	
Data Receive Size From Rank %(RANK)	5	B	Accumulate	DataReceiverManager	
Data Receive Time From Rank %(RANK)	5	s	Accumulate	DataReceiverManager	
Header Receive Size From Rank %(RANK)	5	B	Accumulate	DataReceiverManager	
Header Receive Time From Rank %(RANK)	5	s	Accumulate	DataReceiverManager	
Total Receive Size From Rank %(RANK)	5	B	Accumulate	DataReceiverManager	
Total Receive Time From Rank %(RANK)	5	s	Accumulate	DataReceiverManager	
Total Shared Memory Wait Time From Rank %(RANK)	3	s	Accumulate	DataReceiverManager	
Avg Fragment Wait Time From Rank %(RANK)	3	s	Average	DataReceiverManager	
Avg Shared Memory Wait Time From Rank %(RANK)	3	s	Average	DataReceiverManager	
Data Receive Rate From Rank %(RANK)	5	B/s	Average	DataReceiverManager	
Header Receive Rate From Rank %(RANK)	5	B/s	Average	DataReceiverManager	
Total Receive Rate From Rank %(RANK)	5	B/s	Average	DataReceiverManager	
Data Receive Count From Rank %(RANK)	3	fragments	LastPoint	DataReceiverManager	
Data Send Size to Rank % (RANK)	5	B	Accumulate	DataSenderManager	
Data Send Time to Rank % (RANK)	5	s	Accumulate	DataSenderManager	
Data Send Rate to Rank % (RANK)	5	B/s	Average	DataSenderManager	
Routing Wait Time	2	s	Average	DataSenderManager	
Data Send Count to Rank % (RANK)	3	fragments	LastPoint	DataSenderManager	
Routing Table Size	2	events	LastPoint	DataSenderManager	
Request Response Time	2	seconds	Average	RequestReceiver	
Average Event Size	1	Bytes	Average	SharedMemoryEventManager	
Tokens sent	2	Tokens	LastPoint	SharedMemoryEventManager	
Events Released to art this run	1	Events	LastPoint	SharedMemoryEventManager	
Incomplete Event Count	1	events	LastPoint	SharedMemoryEventManager	
Incomplete Events Released to art this run	1	Events	LastPoint	SharedMemoryEventManager	
Pending Event Count	1	events	LastPoint	SharedMemoryEventManager	
Run Number	1	Run	LastPoint	SharedMemoryEventManager	
Shared Memory Available %	2	%	LastPoint	SharedMemoryEventManager	
Shared Memory Available Buffers	2	buffers	LastPoint	SharedMemoryEventManager	
Shared Memory Full %	2	%	LastPoint	SharedMemoryEventManager	
Shared Memory Full Buffers	2	buffers	LastPoint	SharedMemoryEventManager	
Shared Memory Pending Buffers	2	buffers	LastPoint	SharedMemoryEventManager	
Shared Memory Reading Buffers	2	buffers	LastPoint	SharedMemoryEventManager	
Incomplete Event Rate	3	events/s	Rate	SharedMemoryEventManager	
Event Rate	1	Events/s	Rate	SharedMemoryEventManager	
Input Fragment Rate	1	Fragments/s	Rate	SharedMemoryEventManager	
Avg Data Acquisition Time	3	s	Average	CommandableFragmentGenerator	
Buffer Depth Bytes	1	bytes	LastPoint	CommandableFragmentGenerator	
Buffer Depth Fragments	1	fragments	LastPoint	CommandableFragmentGenerator	
Last Timestamp	1	Ticks	LastPoint	CommandableFragmentGenerator	

EPICS Database

- Postgres DBMS



Conclusion

Mu2e TDAQ system and slow controls integration status

- Advanced Slow Controls Monitoring in *otsdaq*
- Real configuration to send *otsdaq* front-end DAQ hardware, data processing and DQM in EPICS
- Real configuration and Integration with State Machine
- hardware tests are going on
- connection with *otsdaq* @ mu2e building will be done in the next years 2021-2022