# The BL4S Data Acquisition System

Cristóvão B. da Cruz e Silva
on behalf of BL4S Team

# Overview

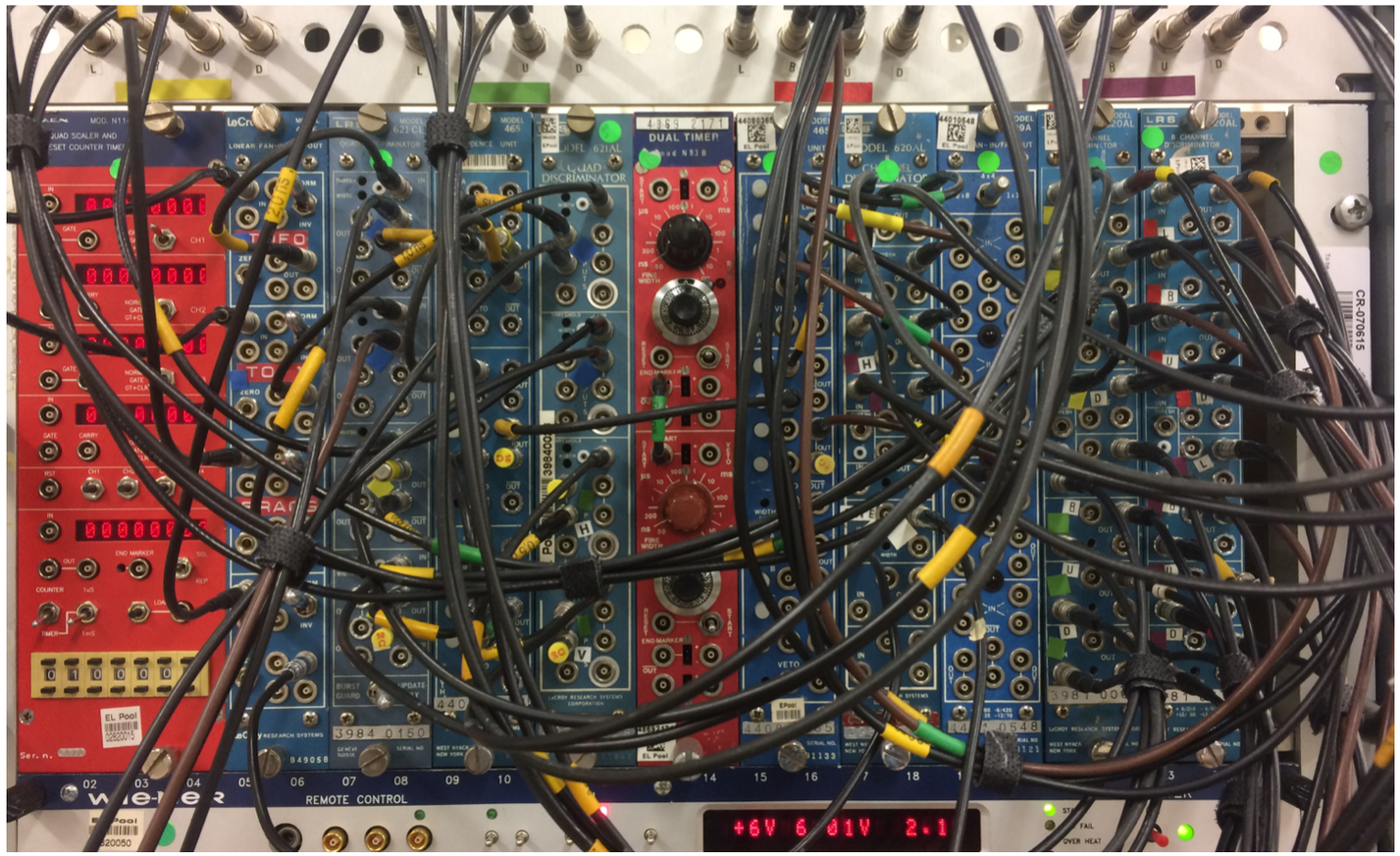
- DAQ Hardware
  - NIM
  - VME
  - Micromegas

- DAQ Software
  - GUI
  - Readout Modules
  - Monitoring

- EUDET Type Telescope DAQ
  - Hardware
  - Software

# DAQ Hardware - NIM

- HV Modules
  - Power detectors
- Discriminators
- Coincidence Unit
  - For trigger logic




**2018 NIM Crate Setup (Both BL4S experiments)**

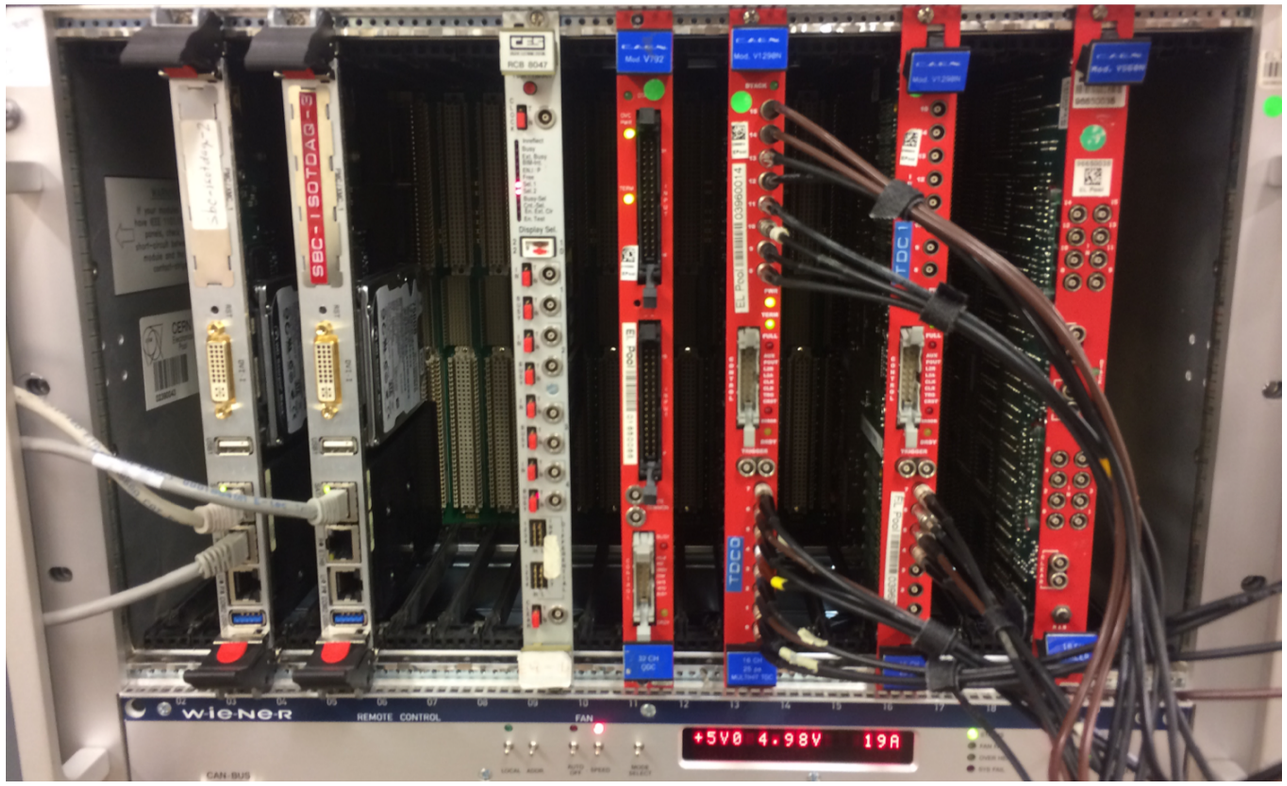
# DAQ Hardware - VME

- SBC
  - DAQ Software
  - Data storage

- TDC (CAEN v1290)
  - 25 ps resolution

- QDC (CAEN v792)

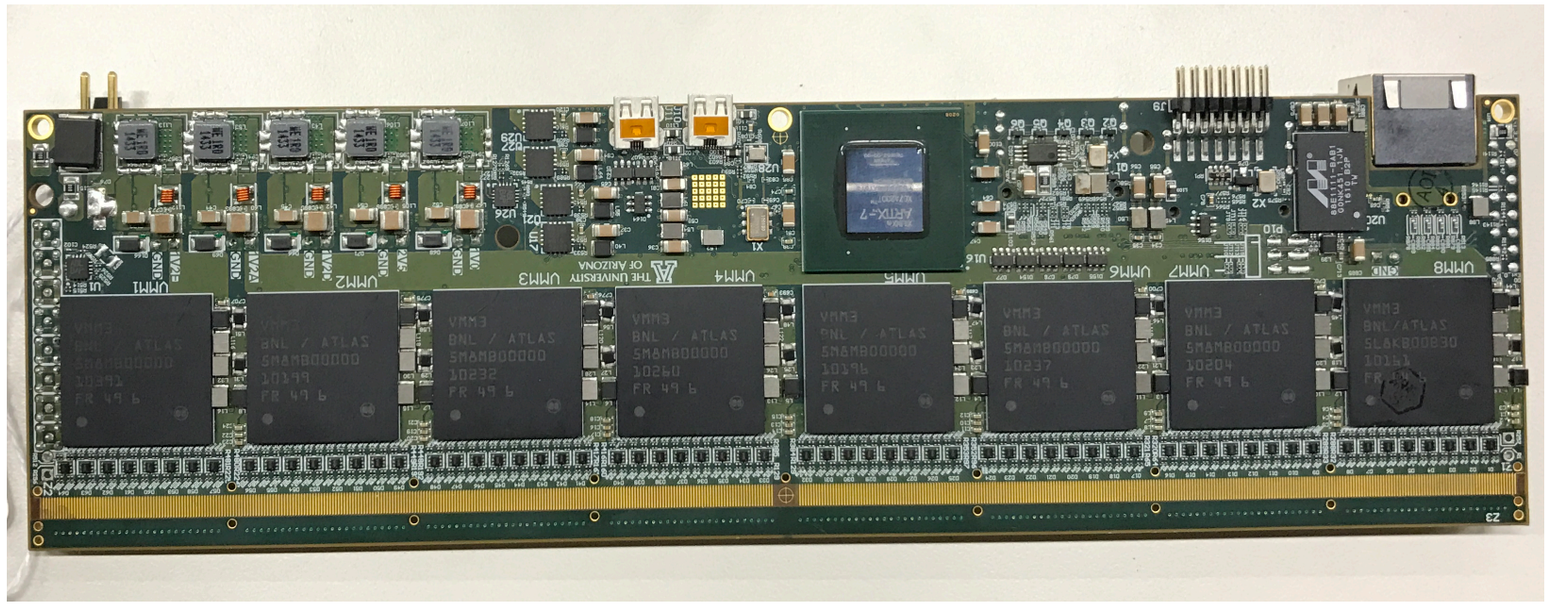
- CORBO (Trigger Board)
  - Synchronous system




**2018 VME Crate (Mid setup)**

# DAQ Hardware - MM

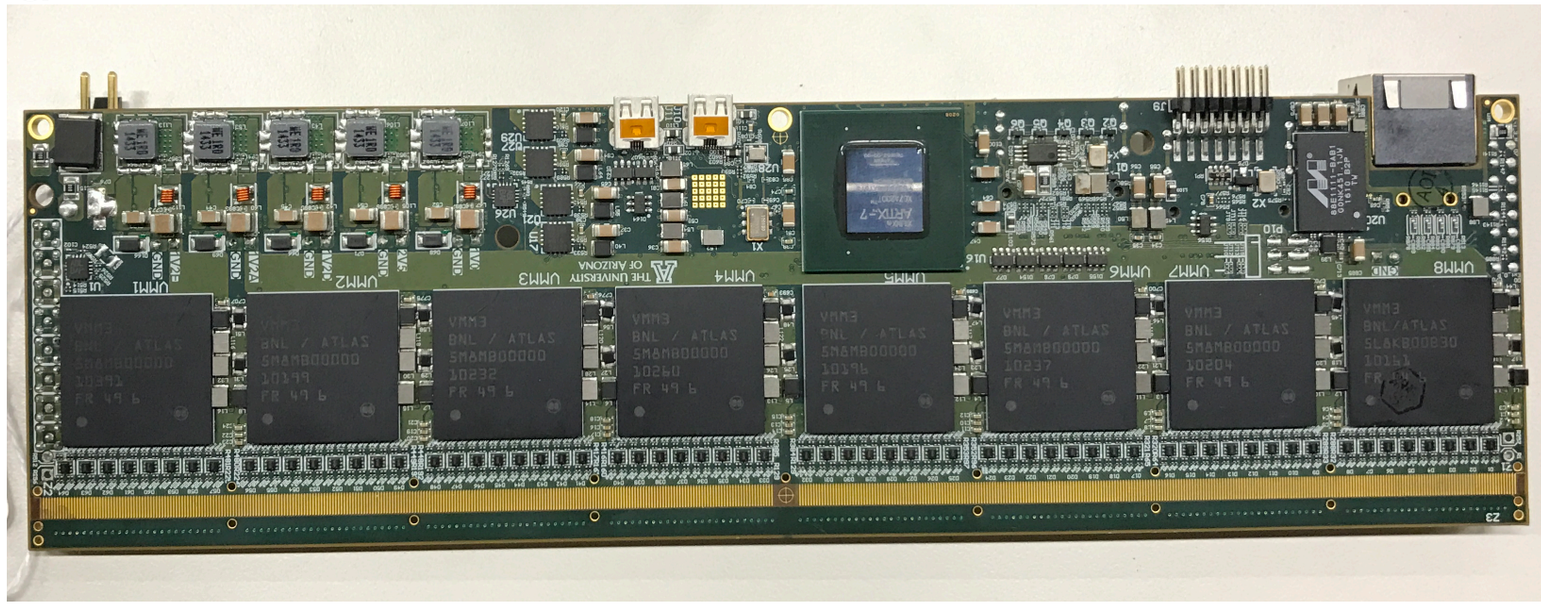
- Micromegas design based on the Atlas NSW MM design
- In 2019, readout was changed to MMFE8 (used APV previously)
  - Trigger is distributed through a dedicated NIM module
    - Clock and trigger are sent over HDMI cable
- Data is sent out through UDP over ethernet



**Prototype MMFE8 w/ Ethernet readout**

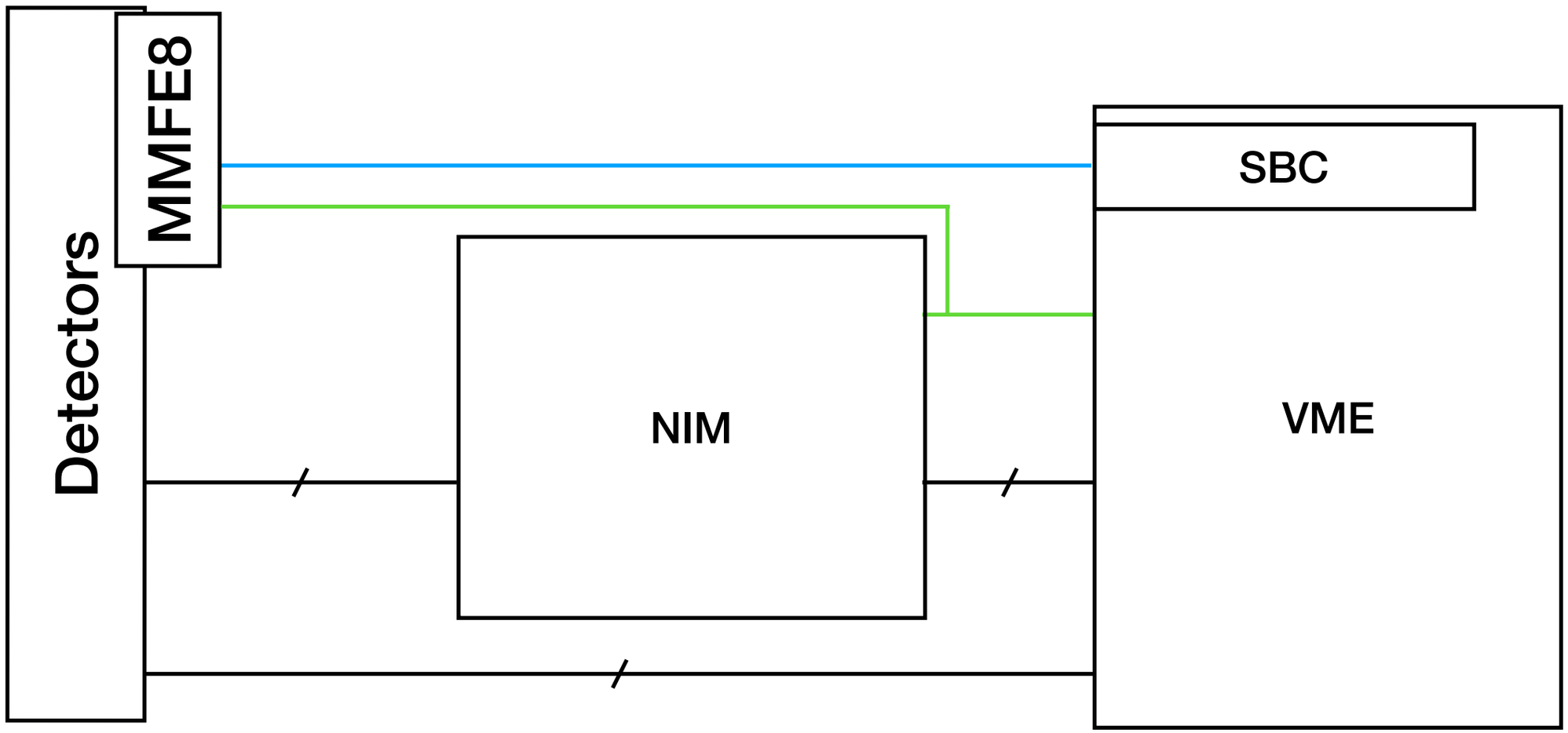
# DAQ Hardware - MM

- Pros:
  - Very reduced development time
  - Extremely capable readout module
- Cons:
  - Constrained by Atlas design, eg: complex system to set up
  - Issues with connection of the channels from detector to board



**Prototype MMFE8 w/ Ethernet readout**

# DAQ Hardware



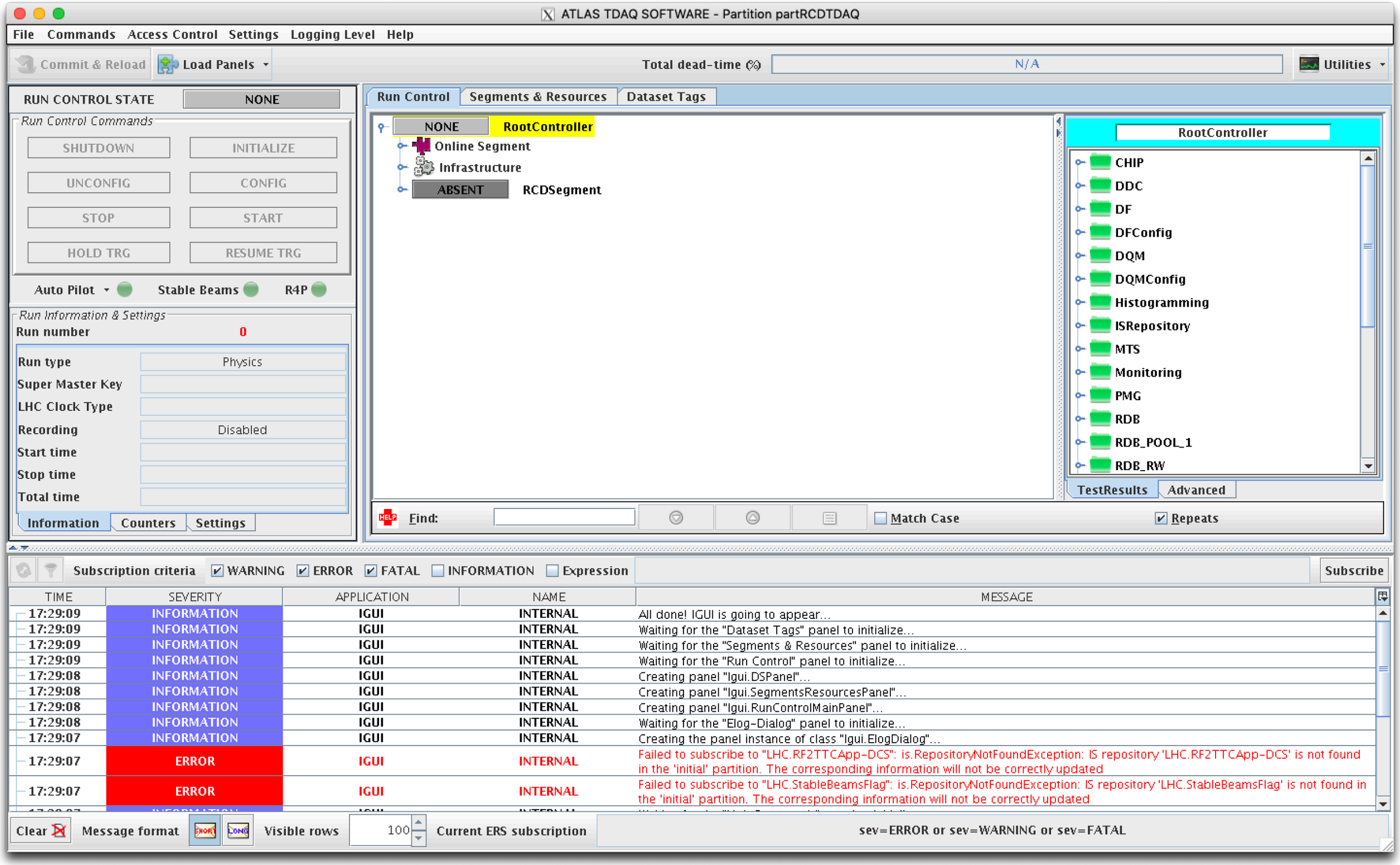
# DAQ Software

- Based on the Atlas “ROD Crate DAQ”:
  - RCDAQ designed to be used in a wide set of environments
  - Can run in a multiuser and multi-PC environment, i.e. it is a scalable DAQ
    - Typically: SBC runs the DAQ processes; run control PC runs GUI interface; monitoring PC runs monitoring processes and second GUI interface

- Custom readout modules for the VME modules and for the micromegas

- Custom monitoring programs
  - Monitor module performance, e.g. a monitor for the TDC
  - Monitor physics data, e.g. a monitor for the DWC detector

# RCDAQ GUI


ATLAS TDAQ SOFTWARE - Partition partRCDTDAQ
File Commands Access Control Settings Logging Level Help
Commit & Reload
Load Panels
Total dead-time (%)
N/A
Utilities
RUN CONTROL STATE
NONE
Run Control Commands
SHUTDOWN
INITIALIZE
UNCONFIG
CONFIG
STOP
START
HOLD TRG
RESUME TRG
Auto Pilot
Stable Beams
R4P
Run Information & Settings
Run number
0
Run type
Physics
Super Master Key
LHC Clock Type
Recording
Disabled
Start time
Stop time
Total time
Information
Counters
Settings
Run Control
Segments & Resources
Dataset Tags
NONE
RootController
Online Segment
Infrastructure
ABSENT
RCDSegment
RootController
CHIP
DDC
DF
DFConfig
DQM
DQMConfig
Histogramming
ISRepository
MTS
Monitoring
PMG
RDB
RDB_POOL_1
RDB_RW
TestResults
Advanced
Find:
Match Case
Repeats
Subscription criteria
WARNING
ERROR
FATAL
INFORMATION
Expression
Subscribe
TIME
SEVERITY
APPLICATION
NAME
MESSAGE
17:29:09 INFORMATION IGUI INTERNAL All done! IGUI is going to appear...
17:29:09 INFORMATION IGUI INTERNAL Waiting for the "Dataset Tags" panel to initialize...
17:29:09 INFORMATION IGUI INTERNAL Waiting for the "Segments & Resources" panel to initialize...
17:29:09 INFORMATION IGUI INTERNAL Waiting for the "Run Control" panel to initialize...
17:29:08 INFORMATION IGUI INTERNAL Creating panel "Igui.DSPanel"...
17:29:08 INFORMATION IGUI INTERNAL Creating panel "Igui.SegmentsResourcesPanel"...
17:29:08 INFORMATION IGUI INTERNAL Creating panel "Igui.RunControlMainPanel"...
17:29:08 INFORMATION IGUI INTERNAL Waiting for the "Elog-Dialog" panel to initialize...
17:29:07 INFORMATION IGUI INTERNAL Creating the panel instance of class "Igui.ElogDialog"...
17:29:07 ERROR IGUI INTERNAL Failed to subscribe to "LHC.RF2TTCApp-DCS": is.RepositoryNotFoundException: IS repository 'LHC.RF2TTCApp-DCS' is not found in the 'initial' partition. The corresponding information will not be correctly updated
17:29:07 ERROR IGUI INTERNAL Failed to subscribe to "LHC.StableBeamsFlag": is.RepositoryNotFoundException: IS repository 'LHC.StableBeamsFlag' is not found in the 'initial' partition. The corresponding information will not be correctly updated
Clear
Message format
SHORT
LONG
Visible rows
100
Current ERS subscription
sev=ERROR or sev=WARNING or sev=FATAL

# RCDAQ Readout Modules

- One readout module for each VME module type and one for MicroMegas; readout modules written in C++

- Each readout module implements the state transitions for the associated hardware

- Each readout module also implements data retrieval, placing the relevant data into a ROD fragment, which is then placed in the output data stream
  - BL4S has settled into a common format for the ROD fragment

- Each readout module has O(500) lines of code (only cpp files)

# RCDAQ Monitoring

- Events are sampled on a best-effort basis, i.e. monitoring does not affect deadtime

- A monitor is a stand alone program that runs independently from the DAQ, but subscribes to the DAQ
  - Each monitor receives the full RAW data event
  - Monitors can also be ran over the RAW data files, producing a root file with histograms and other info

- Monitors can publish histograms, these are then available through the RCDAQ GUI

- Typically 400 to 900 lines of code

# RCDAQ Pros & Cons

- RCDAQ inherently contains several desired features, e.g. best effort monitoring, multi-PC environment, inter process communication over network

- RCDAQ provides base infrastructure for readout modules and monitors, enabling compact code and quick development

- Very complex system, have to rely on help from experts when experiencing more exotic errors

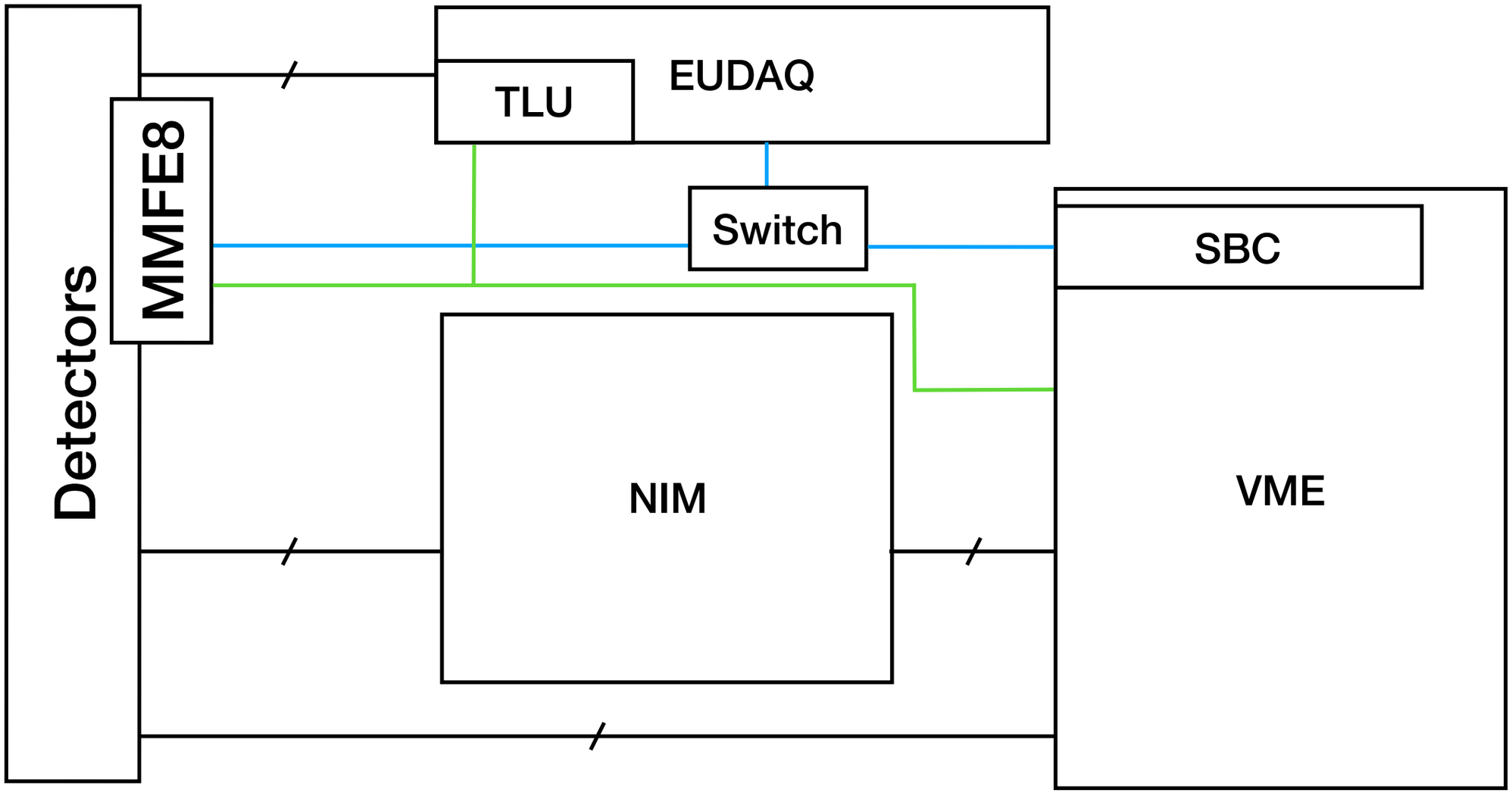
- RCDAQ is very sensitive to the network configuration
  - Changing IP/location (e.g. CERN to DESY)

# EUDET Type Telescope DAQ

- In 2019, with the move to DESY, the DESY beam telescopes became available
- Options for integration with previous DAQ system:
  - Re-implement EUDAQ functionality in RCDAQ
  - Re-implement RCDAQ functionality in EUDAQ
  - Run both DAQ systems in parallel and synchronously

# EUDET Type Telescope DAQ Hardware

- EUDAQ responsible for triggering, but sends data to RCDAQ
  - BUSY signal from VME sent to TLU



# EUDET Type Telescope DAQ Software

- EUDAQ Producer
  - Take Mimosa data and encapsulate it in a frame
  - Send frame over UDP

- RCDAQ Readout Module
  - Receive UDP packet
  - Reformat frame into ROD fragment

Implemented
~ 1 week

- RCDAQ Monitors
  - Hit monitor
  - Correlation monitor (between planes and between other tracking detectors)
  - 3D Event monitor

# Conclusion

- The standard BL4S DAQ is modular and flexible, thanks to its roots in RCDAQ
  - It can be adapted to a wide variety of different experiments
- New modules/detectors and monitoring programs can be implemented with a quick turnaround
- Sucessfully combined RCDAQ and EUDAQ into a hybrid DAQ

# Thank you!

*Questions?*