Fermilab

Managed by Fermi Research Alliance, LLC for the U.S. Department of Energy Office of Science

artdaq and *otsdaq* for SBN and DUNE

Kurt Biery, Scientific Computing Division

SBN – DUNE DAQ Workshop

20 November 2015

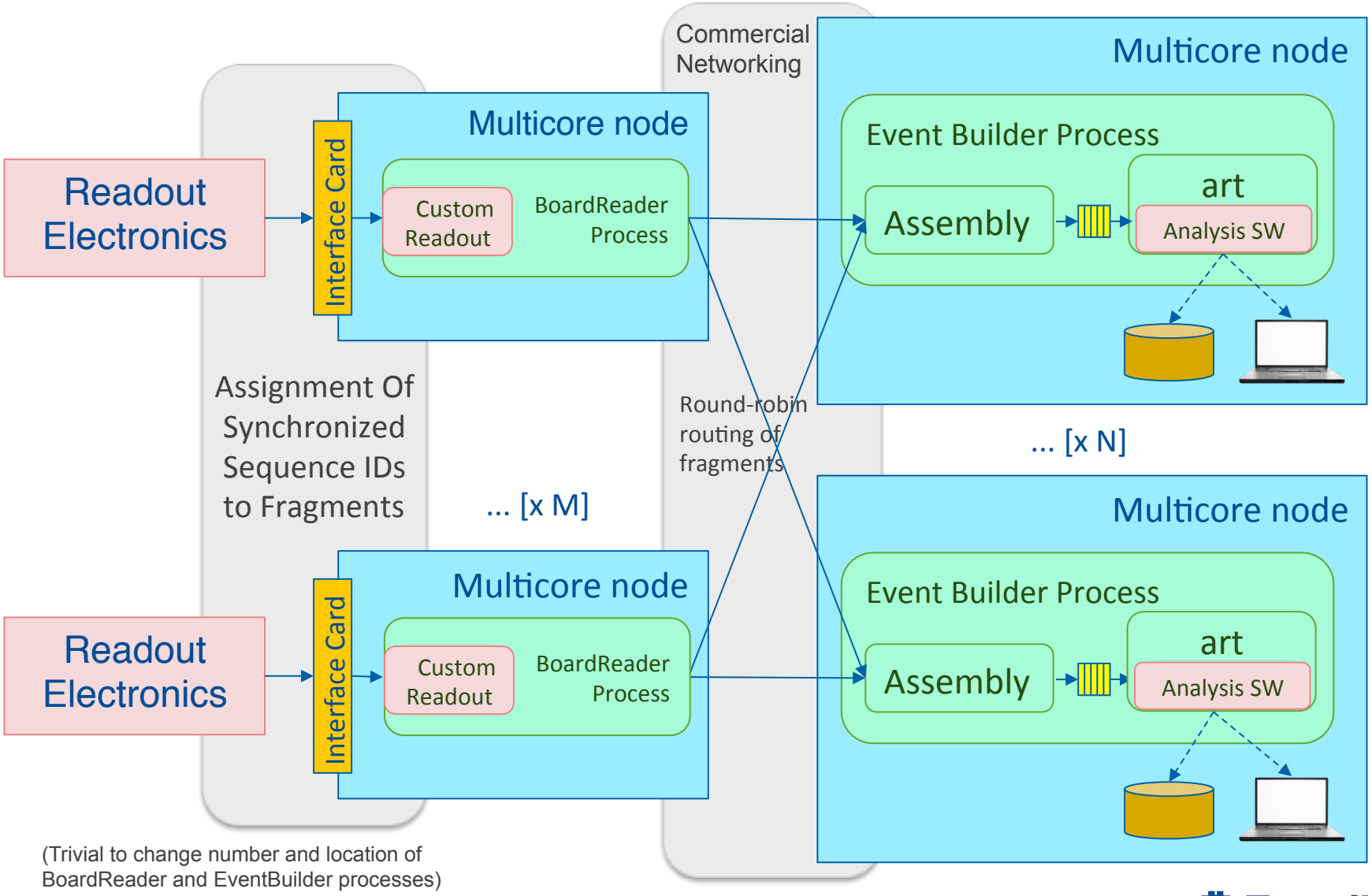
What is *artdaq*?

artdaq is a software toolkit for creating data acquisition systems

- Core DAQ software developed by Fermilab SCD
  - Provides common, reusable components
  - Based on a data-streaming architecture with software event filtering
- Integrated with the *art* framework
  - Same environment for development of online and offline algorithms

- It provides data transfer, event building, process management, system and process state behavior, control messaging, message logging, configuration infrastructure, DAQ monitoring, writing of data to disk, and infrastructure for online data quality monitoring.
- The goal is to provide the common, reusable components of a DAQ system and allow experimenters to focus on the experiment-specific parts of the system. These are the software that reads out and configures the experiment-specific front-end hardware, the analysis modules that run inside of *art*, and the data quality monitoring modules.

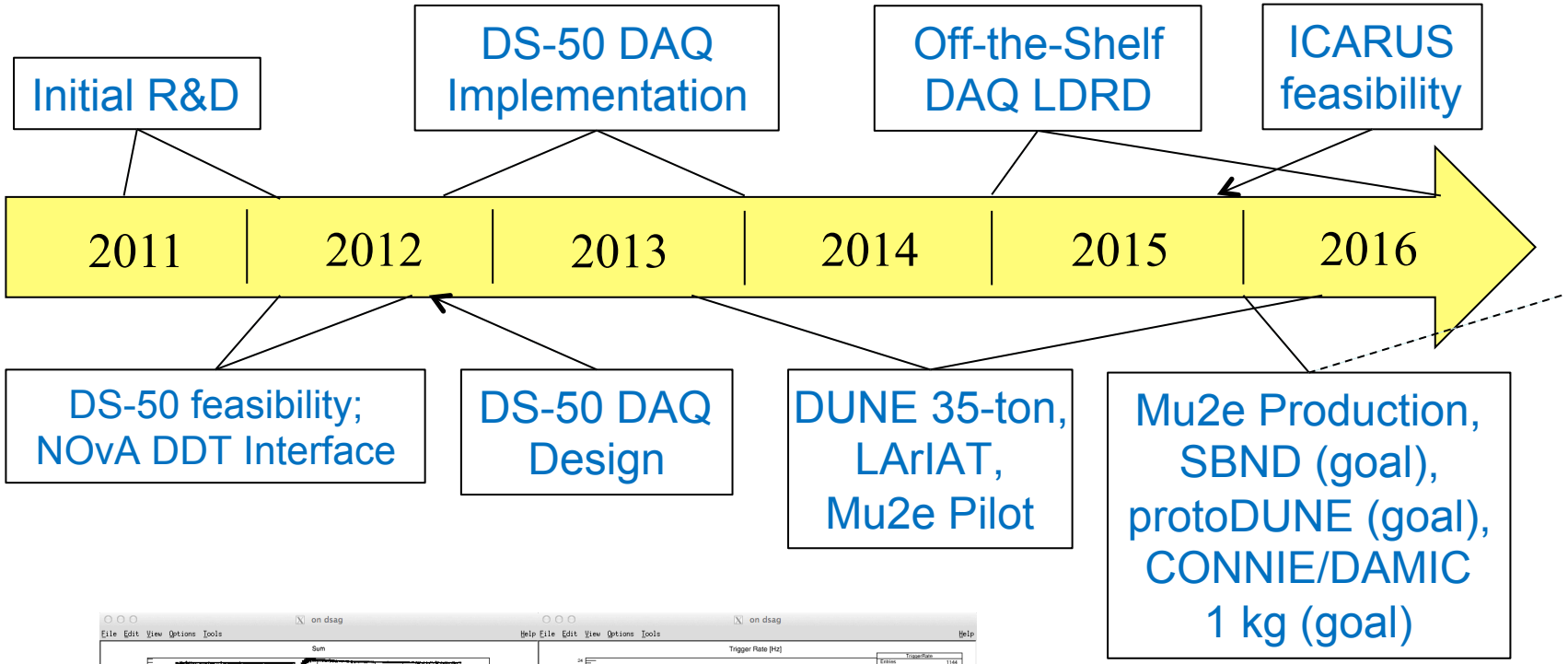
Simplified *artdaq* System

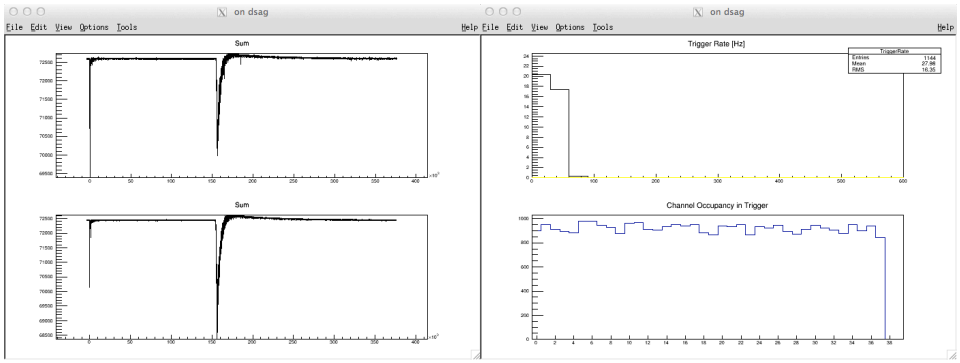


(Trivial to change number and location of BoardReader and EventBuilder processes)

Fermilab

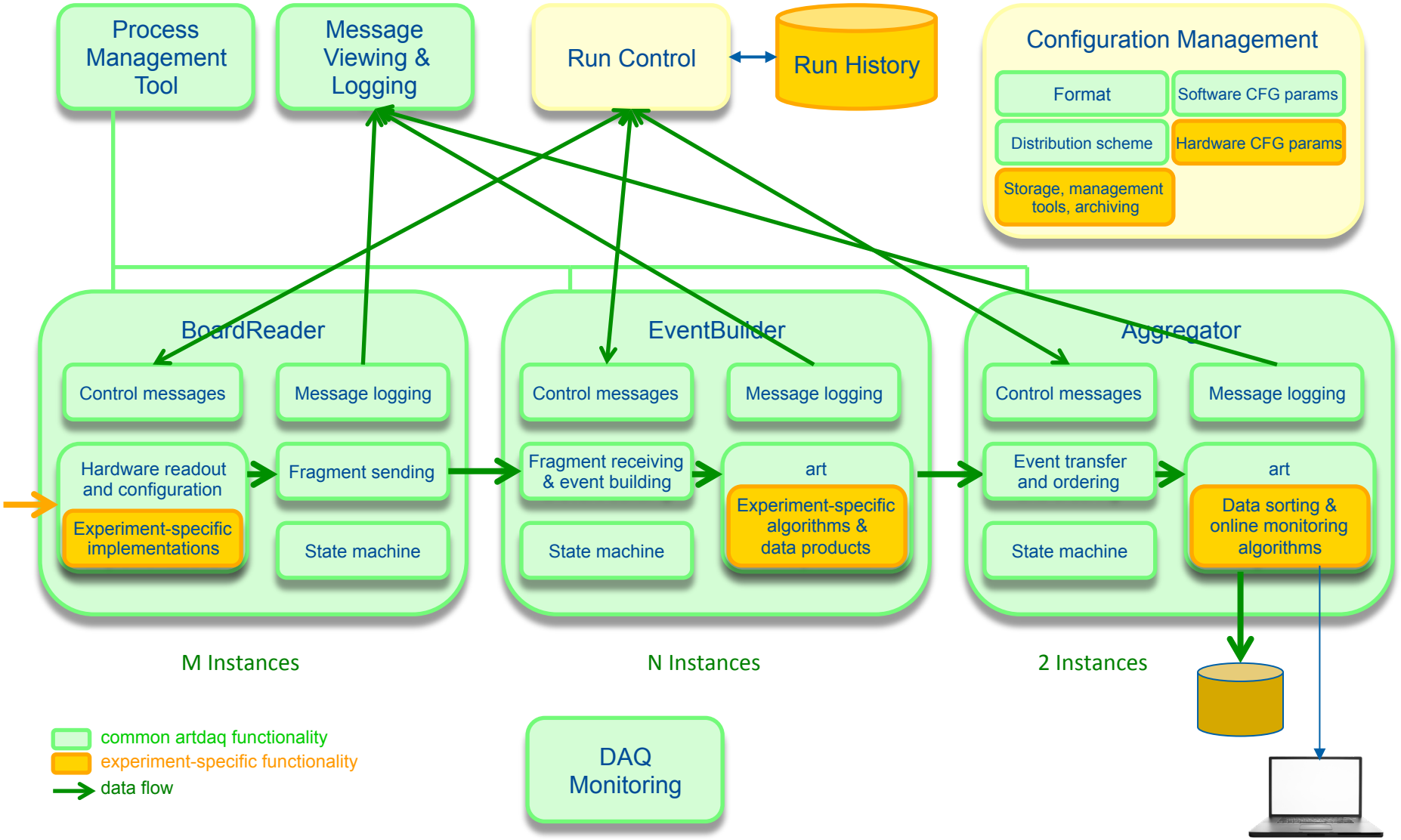
artdaq Timeline





Sample DS-50 online monitoring histograms

artdaq Software Components and Functions

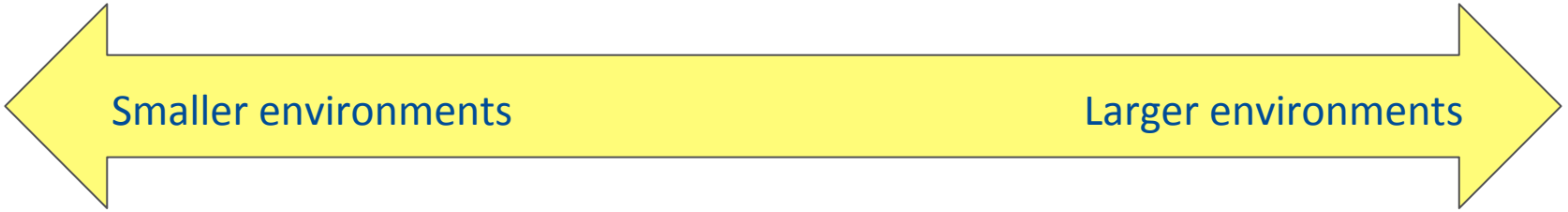


Fermilab

Off-the-Shelf DAQ LDRD

- 2 years of effort for *otsdaq* proof-of-concept:
  - Survey the market for candidate IoT boards.
  - Focus on 1 board in each range (Low, Mid, and High) to populate initial menu.
  - Develop a JavaScript GUI for control and readout using web browser.
  - Develop host and embedded APIs for socket based communication between *artdaq* and candidate boards.
  - Develop sample reusable firmware components.
  - Test and catalog available features and supported data rates.
- Beyond the LDRD, our goal is to build on the proof-of-concept to build a fully functioned, easy-to-use, off-the-shelf DAQ system (bringing together HW, FW, SW).

Scaling Up and Scaling Down



Test beam (FTBF) University test stands Detector test stands	LArIAT	protoDUNE	Full-scale experiments DUNE, SBN, Mu2e, DarkSide

otsdaq (including *artdaq*)

- Quicker and easier startup
- UDP *artdaq* receiver
- Reusable firmware
- Recommended OtS hardware
- Possibility of slow controls in data stream

artdaq with tools from *otsdaq*

- Experiment-developed components
- Collaboration on *artdaq* customizations and enhancements

Goal is smoother transition from test stands through full experiments

Developing *artdaq*-based Experiment DAQ Systems (1)

Our model is to partner with experiments

- FNAL SCD: *artdaq* training and guidance; changes to *artdaq*, if needed; system integration, testing, and debugging; assistance with experiment-specific components, upon occasion.
- Experiment collaborators: software interface to detector electronics, online monitoring algorithms, compression or analysis software.

DUNE 35-ton experience

- SCD: initial setup of the lbne-raw-data and lbne-artdaq git repos; advice and contributions for development of RCE, SSP, and PTB interface software; development of RunControl-to-*artdaq* interface application; core *artdaq* changes; builds of lbne_raw_data for offline; system testing (e.g. max throughputs); lbne-artdaq development (e.g. stats to RC).
- DUNE scientists: RCE, SSP, and PTB interface software; *art*-based analysis and filtering software; online monitoring software; RunControl; configuration management.

Developing *artdaq*-based Experiment DAQ Systems (2)

DarkSide experience

- SCD: core *artdaq* development; RunControl scripts; collaboration on hardware interface and compression software; commissioning effort; assistance with builds and deployments.
- DarkSide scientists: CAEN V1720, V1724, V1190, and trigger board interface software; graphical RunControl application, online monitoring software, primary responsibility for testing and updates to new versions of the software.

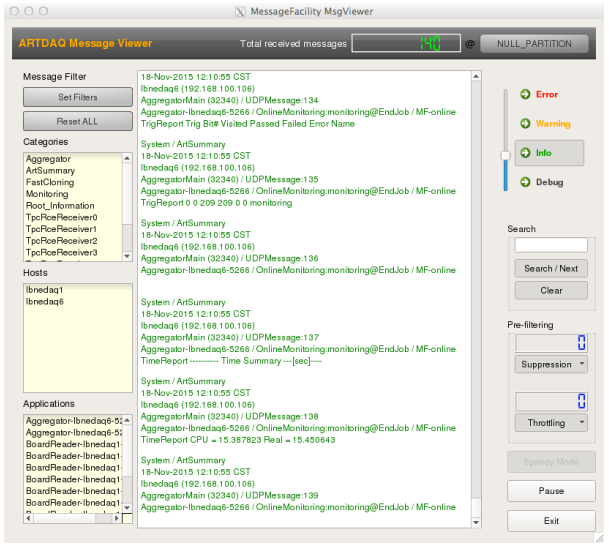
Working model for SBND (Wes' talk on 09-Nov)

- Reuse of hardware interface libraries from uBooNE.
- *artdaq* infrastructure.
- PMT, trigger, and laser readout in parallel with TPC – a critical piece of this will be a common sequence ID.
- Update CRT (cosmic ray tagger) teststand at Bern to use *artdaq*.
  - Build partial events consisting of just CRT data – sequence ID important here, too.
- Path forward includes *artdaq* teststands, core *artdaq* enhancements, Run Control

Sample Enhancements to *artdaq* for DUNE 35-ton

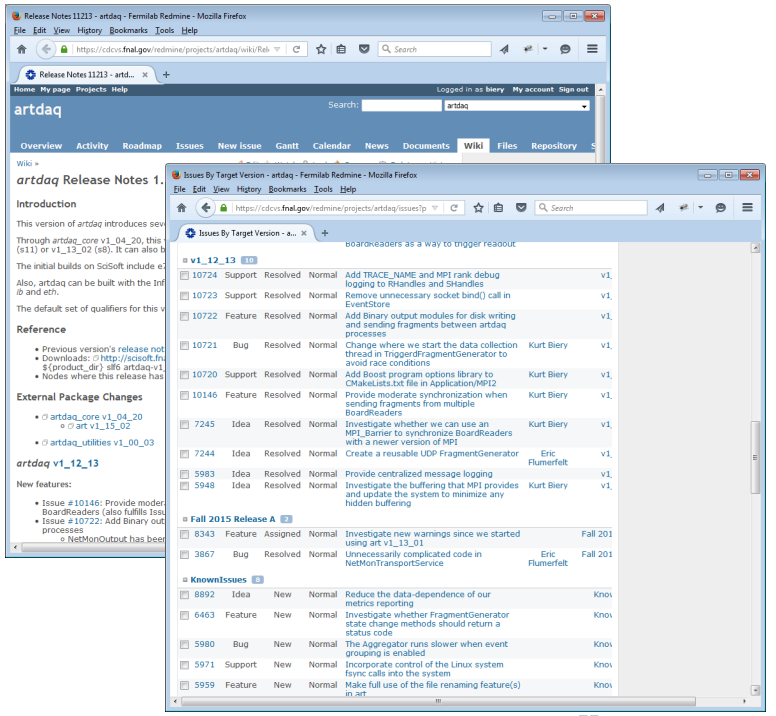
- DAQ performance monitoring using Ganglia
- Graphical viewer for log messages
- Fragment synchronization to prevent readout of some parts of the detector running far ahead
- Error state for *artdaq* processes to indicate a fatal error in one part of the DAQ system

- All of these have been on the *artdaq* to-do list for a while, and their implementation was motivated by the needs of the 35-ton DAQ.



Core *artdaq* Development

- Team of 6 software professionals with backgrounds in HEP and engineering working on DAQ software
  - Core *artdaq*, experiment-specific customizations, driver software, hardware interface libraries, system design and testing
  - With hardware and firmware engineers, developing off-the-shelf DAQ software, firmware, software, and recommended OtS HW
- Releases of *artdaq* scheduled based on needs of experiments, and needs of projects such as *otsdaq*
  - Release notes: https://cdcvs.fnal.gov/redmine/projects/artdaq/wiki
  - Issue tracking: https://cdcvs.fnal.gov/redmine/projects/artdaq/issues
- *artdaq*-demo for education and starting point for experiment-specific software packages
  - Wiki documentation: https://cdcvs.fnal.gov/redmine/projects/artdaq-demo/wiki



Fermilab

Coordinating with Other Groups at Fermilab

- We're working with SCD/ECF/SLAM group (Rennie Scott et al) to define standard patterns for DAQ teststand and pilot system clusters. (Rennie's group has already developed a robust model for production DAQ clusters.)
- We work closely with the *art* development team to prepare new versions of *artdaq* coupled with new versions of *art*, request and develop changes to either package based on the needs of the other, and get advice on best ways to modify parts of *artdaq* that are closely related to art.
- We work with the Neutrino Division Online Support Group to build and configure DAQ clusters and share information regarding support of experiment DAQ systems.

Enhancements and Additions

- Multi-layer *artdaq* systems – motivated by needs of DUNE, SBND, upgraded uBooNE, Mu2e.
- Run Control – ability to control/monitor multiple "experiments" has come up in discussions with both SBN and DUNE; prototype graphical interface developed as part of *otsdaq*.
- Configuration Management – managing *artdaq* system and detector electronics configurations; *otsdaq* work; discussions with 35t folks.
- Run-level data products – storing configuration information in the data file.
- Multiple Linux variants – working with Wes now on Ubuntu.
- Integration or interfacing with slow controls.
- Publish/subscribe messaging system – for data transfer and control messages; better separation of online monitoring from primary data flow; load balancing of events into the event builder farm.
- System-wide coordination of back-pressure – maybe best left to hardware, but could be worth investigating.

Multi-layer *artdaq* Systems

SBND and Upgraded uBooNE:

- Writing of local supernova streams.
- Aggregation of cosmic ray tracker data into sub-events.

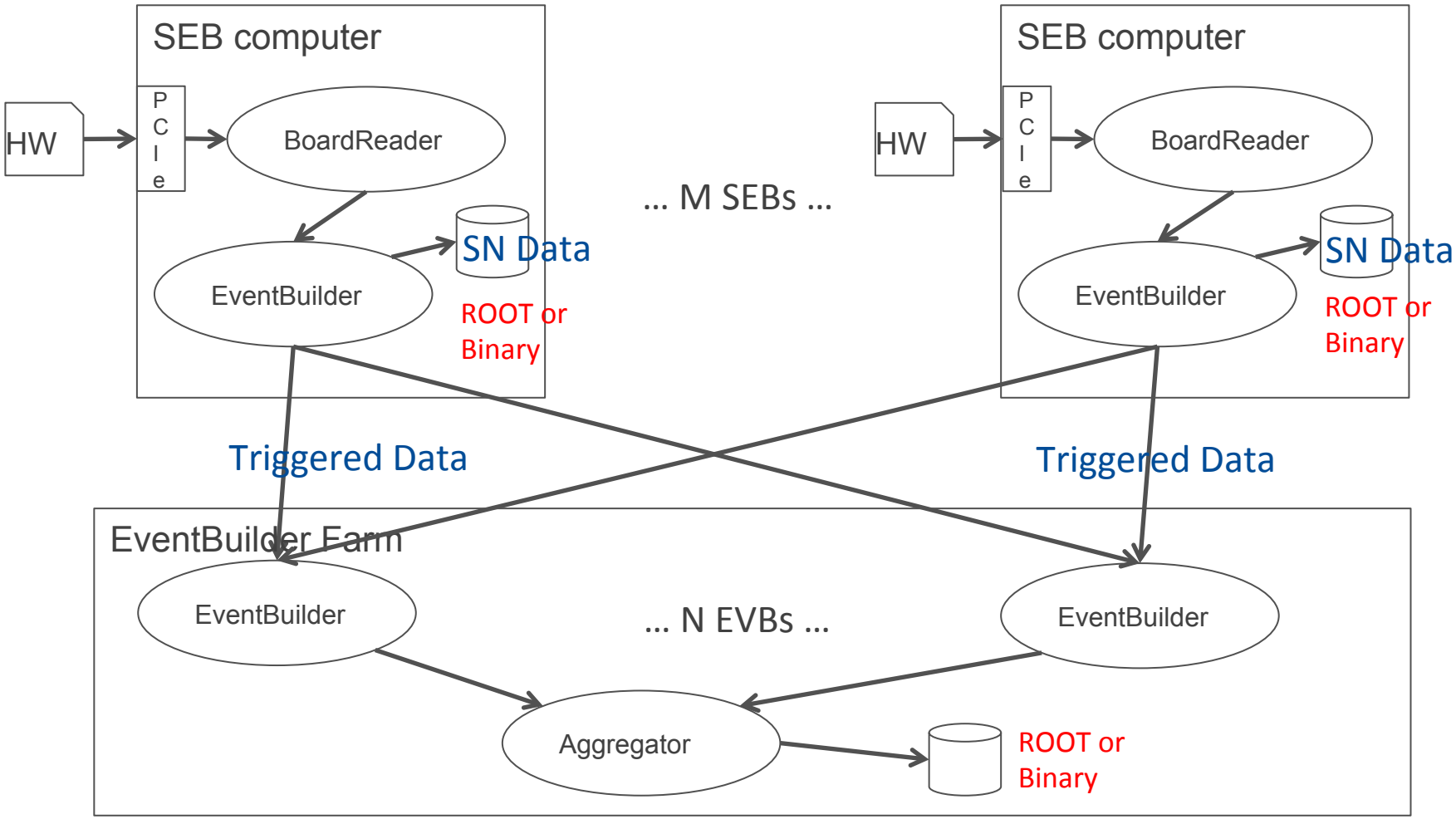
DUNE:

- Readout of zero-suppressed event data into a software trigger farm (first layer).
- For accepted events, the system would request the non-zero-suppressed data (second layer).

Mu2e:

- Readout of tracker and calorimeter into the software trigger farm (1st layer)
- Subsequent readout of cosmic ray veto data only for events which are accepted by the software trigger in the 1st layer (2nd layer)

Multi-Layer *artdaq* for uBooNE-Like System



Run Control and Configuration Management

- Opportunities here for common effort and solutions.
- Similar themes have been heard in SBN and DUNE discussions.
- Common tool(s) for use in *artdaq* would likely benefit SBN, DUNE, *artdaq* itself, and future users of *artdaq*.

Configuration Management:

- We've had some preliminary discussions between DUNE 35t and *artdaq*/*otsdaq* folks about common needs and ideas.

Run Control:

- Maybe a workshop with wide participation?

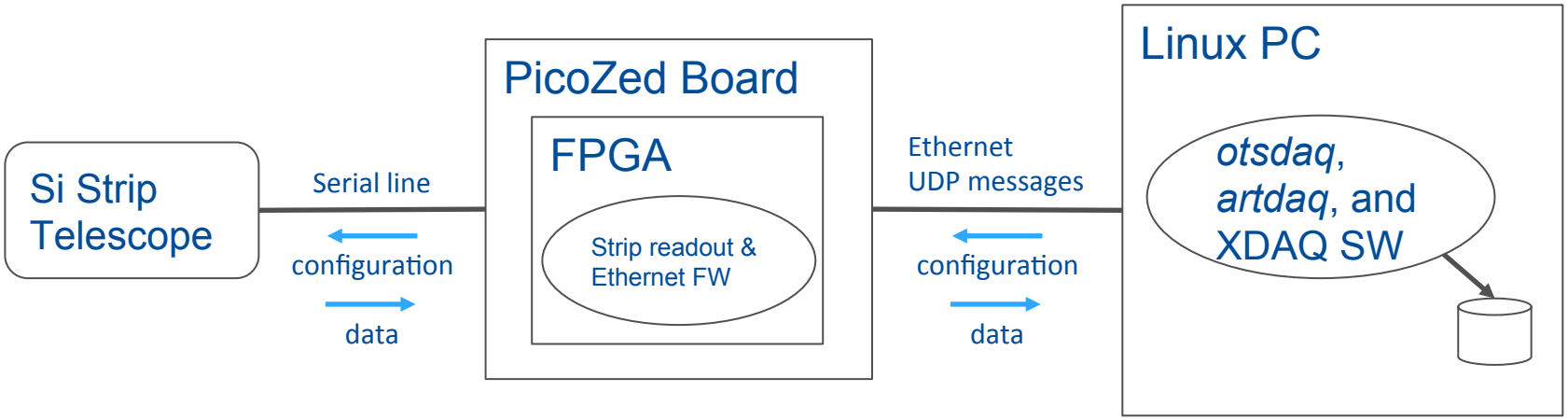
Collaboration on *artdaq*

- *artdaq* is a toolkit, and adding new tools to the toolkit is beneficial to everyone.
- *artdaq* is an open project – contributions from collaborators are welcome, and these are a natural extension of the collaborative work that we do on experiment-specific DAQ systems.

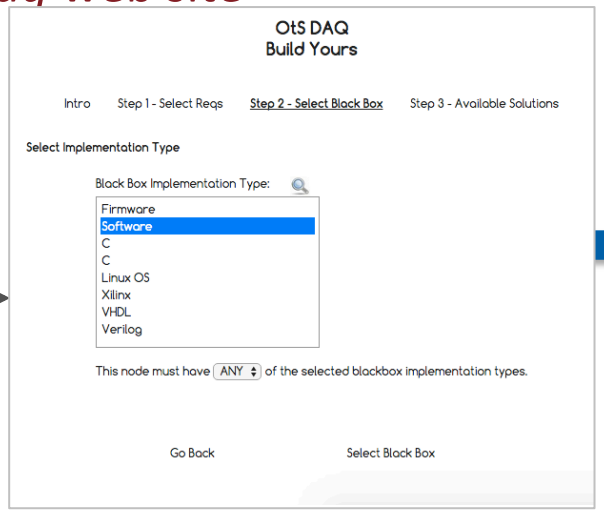
Backup Slides

Off-the-Shelf DAQ Status/Plans

Demonstration of detector readout (at FTBF)



Development of the *otsdaq* web site



To: User
Subject: Your OtS DAQ

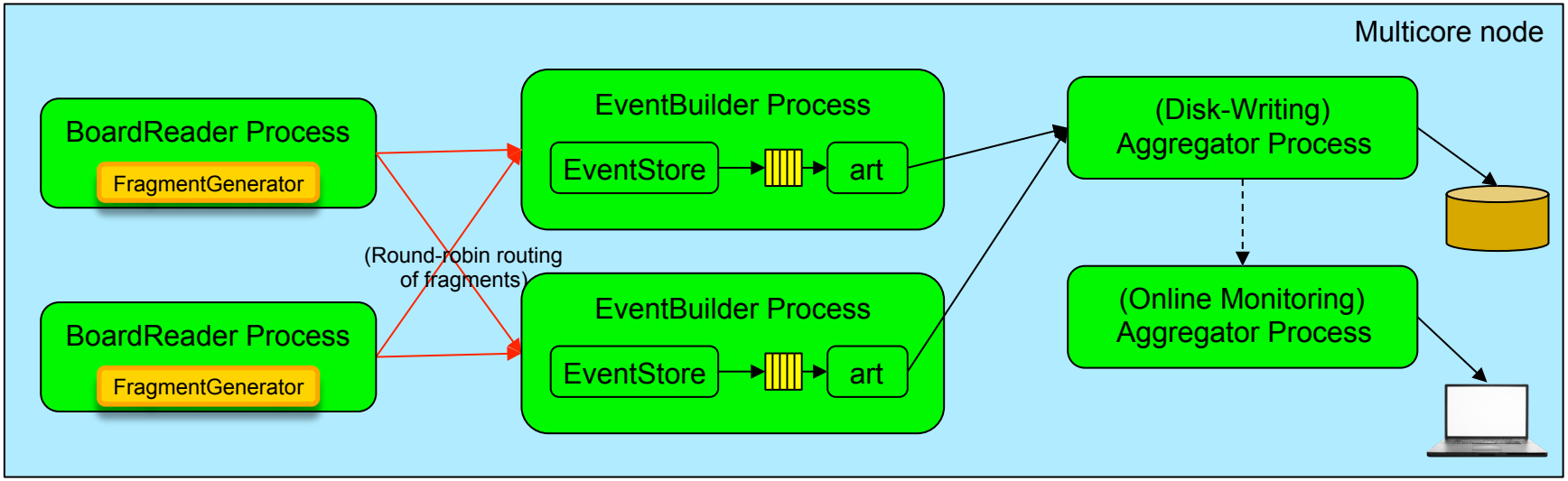
Zip files can be downloaded from here:
Location of FW: URL1
Location of SW: URL2

Fermilab

The *artdaq*-demo

Demo package to illustrate *artdaq* use

- Instructions for downloading, building, and running a sample system
- More information here:
  - https://cdcvs.fnal.gov/redmine/projects/artdaq-demo/wiki
- An easy way to try out *artdaq* and learn more about it
- Currently targeted for Scientific Linux



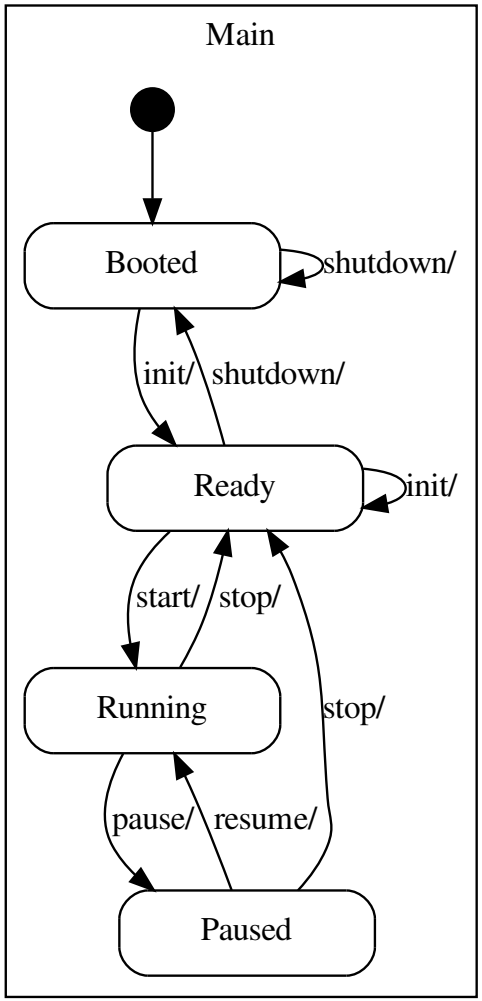
State Behavior & Control Messaging

State behavior part of *artdaq* core

- Standard states:
  - Booted, Ready (configured), Running, Paused, Error
- Standard transitions:
  - Init(cfgString), start(runNum), stop, pause, resume, shutdown
- State Machine Compiler tools
- Overall system state managed by Run Control

Control messaging infrastructure also included

- Standard commands:
  - State transitions & status queries
- XMLRPC
- Command-line Run Control scripts provided



Configuration Documents

Configuration infrastructure part of core *artdaq*

- Structured text string sent at “init” transition
  - hardware, software, system, *art* configuration
- Interpretation of software and system parameters provided in core *artdaq*
- Hardware parameters handled by experiment-specific code
- Archiving and management of configuration data – not yet part of core *artdaq* – but may be in the future

```
daq: {  # red = group identifiers
  max_fragment_size_words: 2097152
  fragment_receiver: {
    mpi_buffer_count: 40
    generator_ds50: {
      fragment_id: 2
    }    # green = software params
    generator: V1495Driver
    first_event_builder_rank: 3
    event_builder_count: 5
    rt_priority: 2  # system params
    link_type: PCIE
    link_number: 0 # hardware params
    vme_base: 0x01000000
    pulser_frequency: 0.0
    laser_frequency: 0.0
    random_triggers: false
    acquisition_gate_us: 400.0
    low_threshold: 5
  }
}
```

Performance

- *artdaq* overheads are small – networking, disk writing, and algorithm performance are bigger effects.
- Some sample numbers from DarkSide:
  - 6-12 BoardReaders on 4 computers, 16 EventBuilders on 4 computers, 40 Gbps Infiniband network.
  - Loss-less compression in software (*art* in EventBuilders) reduces ADC data size by approximately a factor of 5.
  - Up to 500 MB/s from BoardReaders to EventBuilders
  - 1.5+ kHz event rate for laser runs (200 MB/s)
- Initial *artdaq* performance studies showed 3.5+ GB/s overall throughput with 4+ EB nodes and 40 Gbps Infiniband.
- We'll continue to document performance results as we have opportunities to run tests in new DAQ environments (e.g. Mu2e Pilot system).