Dataflow Subsystem

Kurt Biery

DUNE DAQ Workshop

04 February 2019

The suggested topics to cover

Scope & Interfaces

Current status: planning and technical progress

Short-, medium-, and long-term needs

Highlight needs for TDR

Highlight needs by EDR (to be done/decided/tested)

What must be done at protoDUNE and when; what required?

Dataflow Components

- The data transfer from the FELIX hosts to the Event Builders
- The building of timeslice- and geographically-complete bundles of data (‘events’) [dynamic definition of ‘complete’]
- Possible filtering, compression, zero suppression, or other manipulation of the data – High-Level Filter farm
- Real-time data quality monitoring
- Storage of the data before transfer to Fermilab
- Coordination with offline computing (e.g. data transfers)

In slightly different terms…

- *‘artdaq’*
- Storage buffer
- L2 (HLF) farm
- Data management (this means what exactly?)
- Computing interface

Scope, part 1

Existing *artdaq*

- Data transfer, event building, software algorithms in *art*, data logging, delivery of data to real-time DQM
- Dataflow applications: program “main”s, state model, control message handling
- Message logging (MessageFacility and TRACE)

Other possibilities for inclusion in a DAQ framework

- Infrastructure libraries for use in data selection processes

Framework(s) for RT DQM and HLT software

- [Question: is MONET replacement in this scope?]

Scope, part 2

Storage buffer

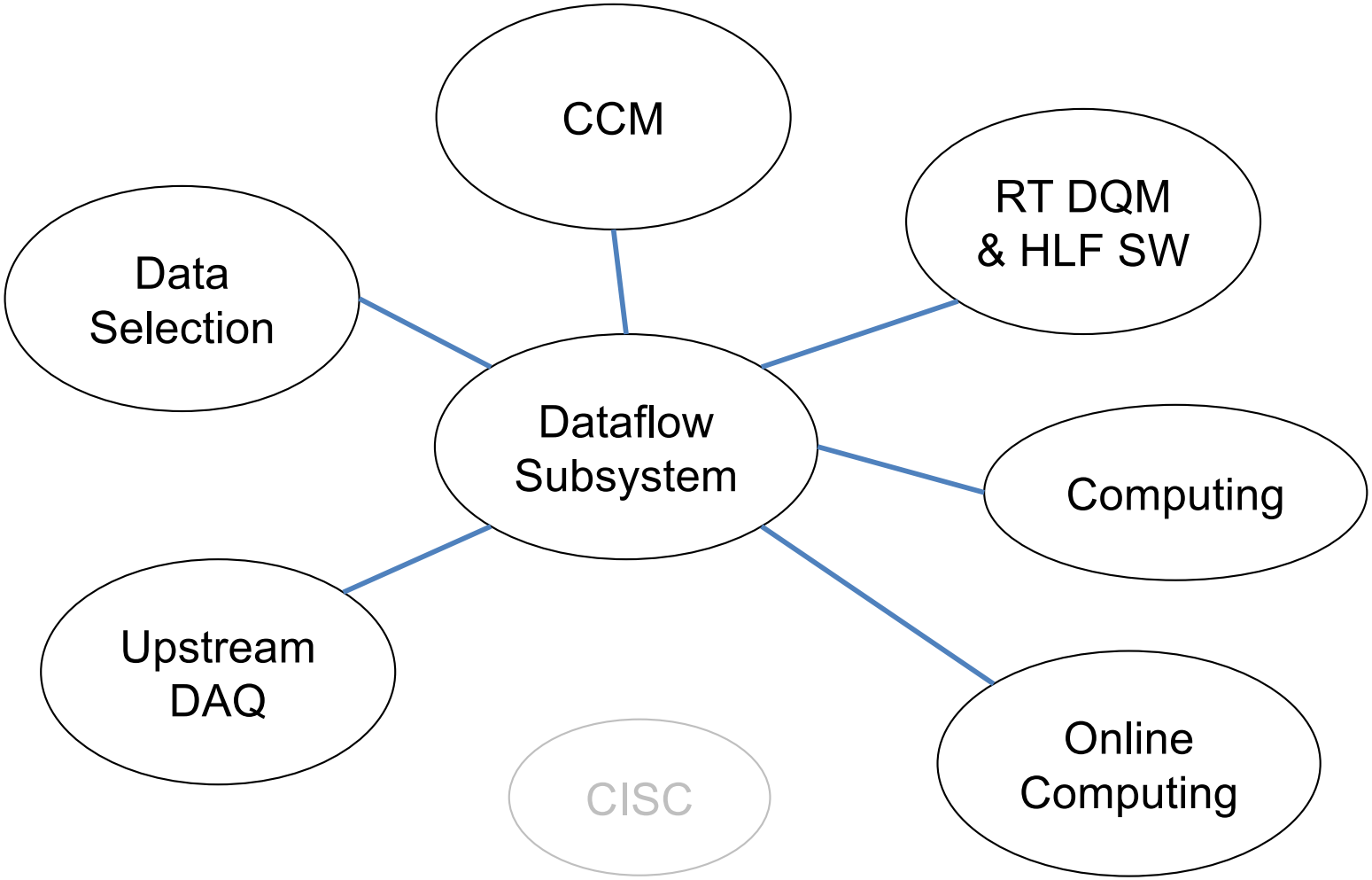
- Data logging applications (including support for streams)
- (Disk) Performance validation and monitoring
- Specification of size and performance
- Usage monitoring and management

Coordination with offline computing for downstream file-transfers, local deletion

Coordination with Computing for specifications on streams, metadata, derived data formats(?)

elog?

Interfaces


CCM
RT DQM
& HLF SW
Data
Selection
Dataflow
Subsystem
Computing
Upstream
DAQ
CISC
Online
Computing

Dataflow Subsystem Interfaces

Data selection system

Upstream DAQ system

Control, configuration, and monitoring systems

Slow controls or other sources of ‘metadata’

Offline data and computing models

Analysis and RT DQM software

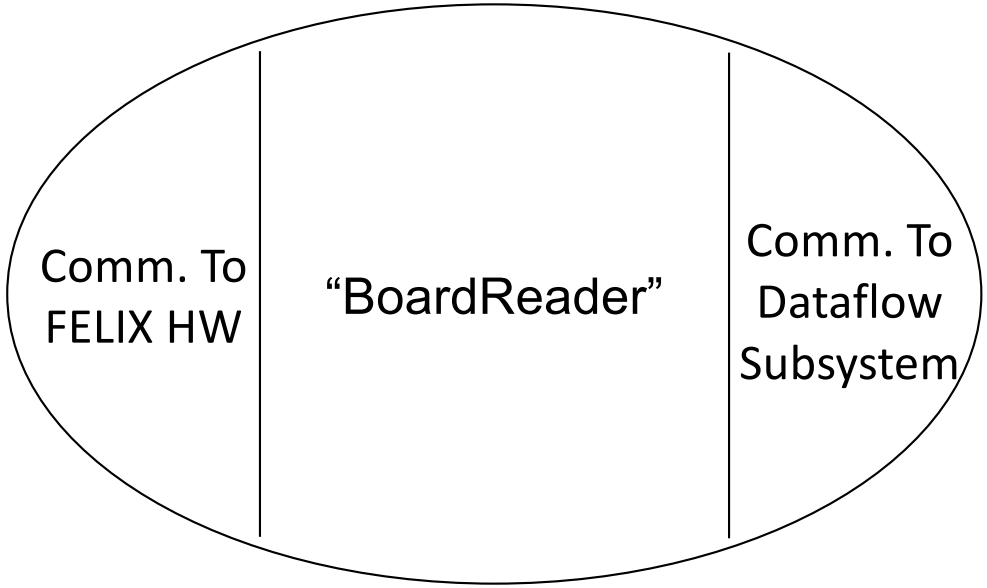
Computer, storage, and network hardware

Operating system(s) and third-party software

Physical environment in computer room(s)

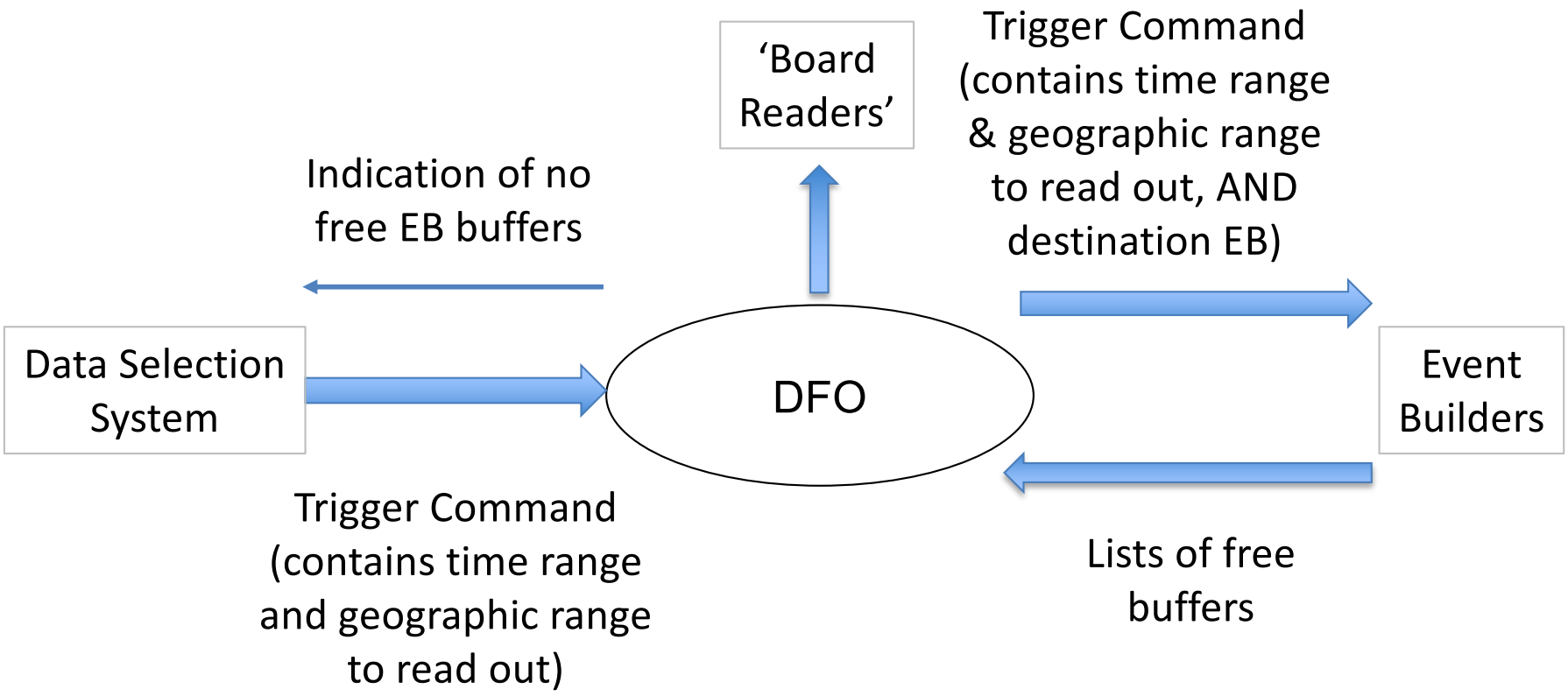
WAN and file transfers

Interfaces in the FELIX host SW process



Discussed earlier in the workshop – let’s defer this to a later Discussion between Upstream DAQ and Dataflow Subsystem.

Dataflow Orchestrator (DFO)



We could say that the ownership of the DFO can be discussed later between the DataSel and Dataflow Subsystem groups, but we could also decide now. Prior expectation was Dataflow Subsystem…

What is needed from other blocks

Run number, configuration snippets, control commands

Servers, OS, disks, NFS or something else

- This needs to be reliable

Partition management (Dataflow is largely agnostic)

[need to think of more items for this list]

What is provided to other blocks

‘DAQ configuration’ delivery

Readout

Trigger record data organized by streams

Support! (diagnosing system issues)

Libraries?

[need to think of more items for this list]

Current Status

Technical status:

- Debugging known issues at protoDUNE
- Implementing *artdaq* changes motivated by ‘lessons learned’
- Validating existing *artdaq* fixes and changes

Status of planning:

- Gathering requirements, task lists
- Planning for adiabatic changes at protoDUNE
- Focusing on practical aspects, not so much on administration or major redesign(s)
- [Forum for coordinating protoDUNE changes?]

Digression – pDataSelInfr at pDUNE

Phil, Wes, Pierre, DavidC, KB met last Thursday

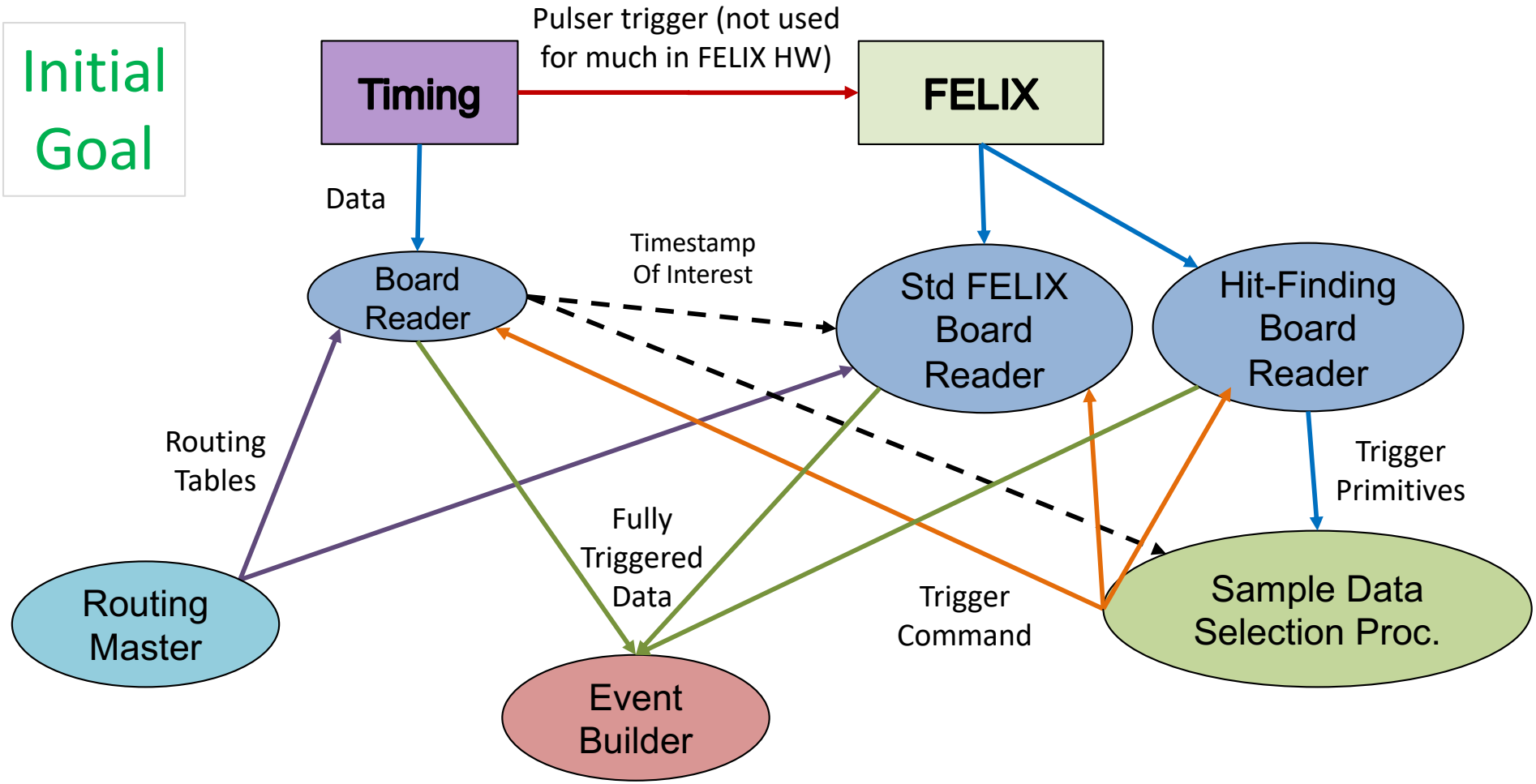
Motivated by Phil's (et al) inquiries about extending his FELIX hit-finding BoardReader code to perform software triggering

We have a plan for incremental changes to the protoDUNE system

Some simplifications:

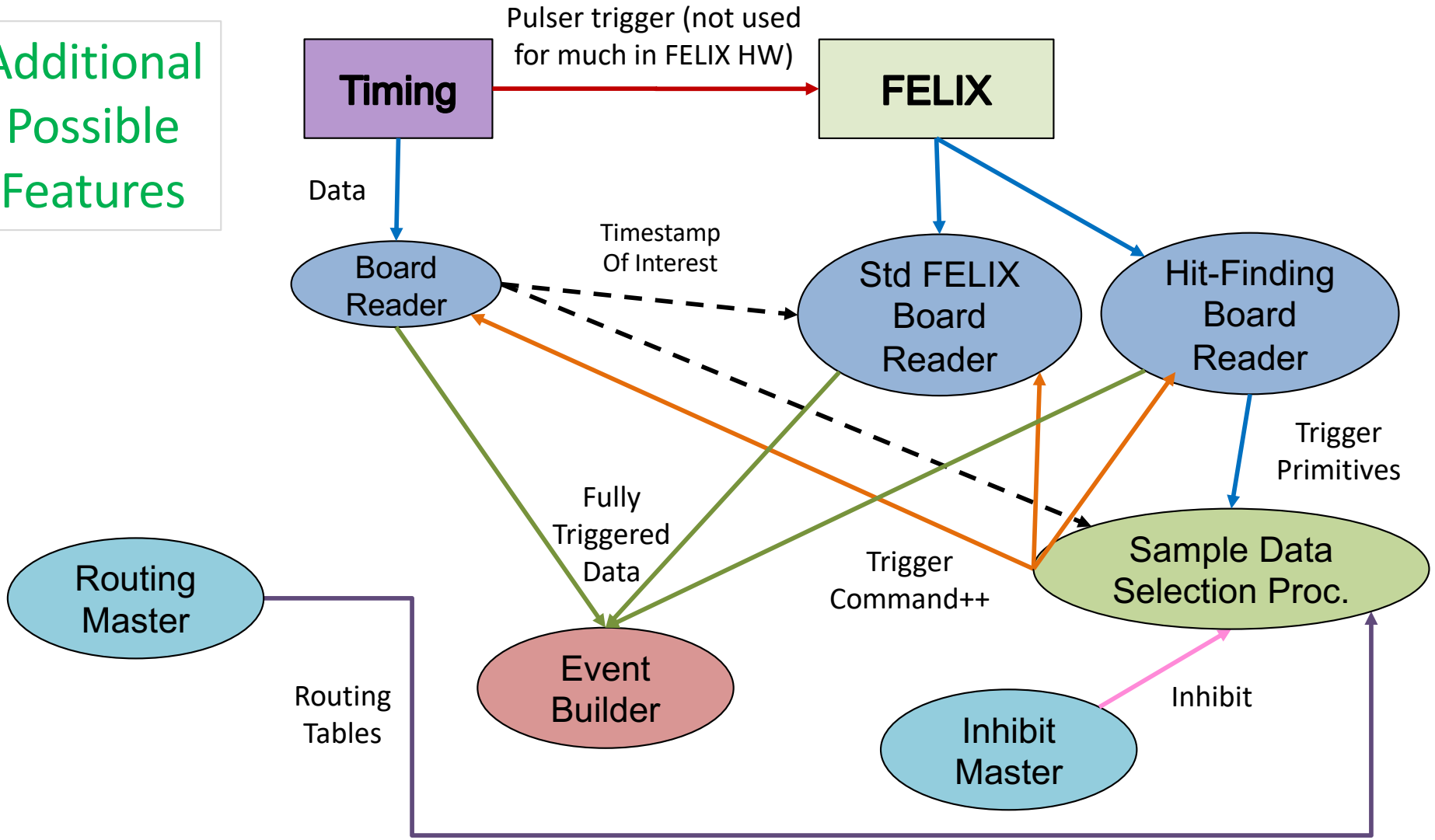
- Start with just the Timing System (& its BR) and FELIX (& its BR(s))
- Configure the Timing System to run at a non-zero pulser rate
- Look for hits, build TriggerPrimitives, etc, within the pulser windows
- The advantages of this pulser-based model:
  - The Timing BR has something meaningful to send to an EventBuilder when it receives a Trigger Command
  - We can keep the existing functionality inside the existing FELIX BR which only compresses the data that corresponds to the time windows specified by messages from the Timing System BR
  - We can (gradually?) increase the pulser rate to essentially achieve DC operation

Digression – pDataSelInfr at pDUNE



Digression – pDataSelInfr at pDUNE

Additional Possible Features



Time Frames

Relevant anchors?

A. protoDUNE DP cosmic run
B. Draining of LAr
C. EDR (Q1 2021)
D. Second beam run

Short- and medium term are between now and [C]?

Long-term is [D] and beyond?

The terminology probably doesn't matter…

Areas of Work (more detail in appendices)

1. protoDUNE debugging [A]
2. protoDUNE enhancements [A-B]
3. *artdaq* enhancements related to protoDUNE [A-B]
4. protoDUNE support [A,B,D]
5. Changes needed for DAQ Kit [A]
6. Queued *artdaq* changes [A]
7. *artdaq* infrastructure changes [D]
8. Offline interface documentation and pDUNE improvements [A]
9. Dynamic reconfiguration [A,B,D+]
10. Fault tolerance [A,B,D+]
11. Other future features [D+]

A couple of DAQ Kit comments

How committed to this are we? (seems quite useful)

It would be worthwhile to understand what Dataflow functionality is needed, nice to have, etc. – Don't know that any work has been done on this.

One positive note: it sounds like the ICEBERG setup at Fermilab is working. (John Freeman and others helped with the software there.)

Opportunity for new Dataflow Subsystem contributors?

Needs for TDR

1. It seems like we're in good shape here…

Needs for EDR

1. Full understanding of needs and plans for fault tolerance and dynamic reconfiguration.
2. Demonstration of progress
3. Prototyping of longer-term developments

4. [In discussions during the talk, we were reminded that the interactions with a partition-aware Data Selection System need to be defined and tested at protoDUNE.]
5. [For later discussion, how to write trigger primitives.]

Milestones, now to EDR

1. [coordinate with Phil et al to define milestones for prototyping data selection infrastructure at protoDUNE]
2. Complete ‘protoDUNE Debugging’ tasks by end of March 2019?
3. Complete selected ‘protoDUNE Enhancements’ by end of May 2019?
4. Ditto selected ‘artdaq enhancements related to pDUNE’
5. Support will be ongoing
6. Update DAQ/Computing Interface document by 28-Feb
7. Improve delivery of metadata to offline by PDDP cosmic run?
8. Perform initial investigations of fault-tolerance and dynamic reconfiguration ideas by the EDR.

Resources and Planning

My view of the resources that are available to work on this…

A practical question:

- Are there special HW, FW, or SW tests that we want/need to do during dedicated DAQ testing times at protoDUNE (e.g. early March and early May)? What level and type of support would be needed for those?

Appendices

The following slides list specific tasks within various areas of work…

Appendix 1 – protoDUNE Debugging

1. Improve run stops/starts
2. Support longer Trigger system readout windows
3. Switch back to delivering events to OnMon via the Dispatcher (as opposed to to disk files)
4. Provide basic fault-tolerance of killing an EB (or other non-critical process, e.g. Dispatcher) and have the system continue running smoothly
5. Investigate occasional long delays in delivering data from BRs to Ebs
6. Rate tests? (data-logging bottlenecks?)

Appendix 2 – protoDUNE Enhancements

1. Change configurations (reconfigure) without restarting the system
2. Provide better visibility into dataflow problems (catch BRs sending empty fragments, catch EBs sending incomplete events)
3. Upgrade/replace Online Monitoring system? (MONET)
4. More service-oriented approach to host assignment (for processes which support that)
5. Self-triggering on data
6. Dynamically exclude components – incremental steps

Appendix 3 – '*artdaq*' Enhancements Related to pDUNE

Practical:

1. Deploy DAQInterface advanced memory usage (max buffer sizes specified in BR configs).
2. Move to sub-configurations as a prototype of better configuration information organization.

More forward-looking:

1. Libraries or other infrastructure for Data Selection software applications?
2. Understand the event-rate and -type needs of OnMon.

Appendix 4 – protoDUNE Support

Planning & changes for extremely large readout windows

- [offline request] Organize data to allow smaller memory usage (Different fragment types and different collections)

Work with folks at CERN to support better remote access to np04 DAQ cluster?

protoDUNE Dual-Phase DAQ

System debugging

Etc, etc.

Appendix 5 – DAQ Kit Preparation & Support

Improve reliability and visibility to errors?

Convenient packaging and deployment

What else?

Appendix 6 – Pending *artdaq* Changes

In the time period leading up to beam running, we were careful about what *artdaq* changes were deployed to protoDUNE.

Some fraction of those were implemented to fix issues that were noticed in testing at Fermilab and will improve reliability at protoDUNE.

We're validating these and deploying them in a controlled way now, as time permits.

Appendix 7 – *artdaq* Infrastructure Changes

Possible changes coming in

- Analysis framework
- Configuration language
- Build and packaging tools

and *artdaq* will need to incorporate these, if they come to pass.

Haven't heard anything about

- Data format

Appendix 8 – Offline Interface Work

1. Update DAQ/Computing interface document
   a. Include statement about providing needed metadata
   b. Add information about nearline monitoring? (L2 farm?)
   c. Add information about shared software packages (e.g. just one data format package?)
2. Work with Computing Consortium folks (et al) to improve the delivery of metadata at PDSP
   a. Get readout window size into the 'runs' DB

[Side note: I've started writing test data to /dataN/test so that it doesn't get copied downstream, and will try to remember to keep doing this.]

Appendix 9 – Dynamic Reconfiguration Work

1. Investigate, plan, and implement additional config steps in *artdaq* (+ other needed/desired CFG structural changes, e.g. separate system, HW, and SW config)
2. Starting planning, investigating, prototyping naming service(s)
3. Look into what changes will be needed for subruns

Appendix 10 – Fault Tolerance Work

1. At protoDUNE, see how far we can get with the existing system and identify the issues
   a. Kill an EB process and see if the system keeps running
   b. Kill a BoardReader process and identify what needs to be changed to have the system continue gracefully
2. Move toward a model in which BRs and EBs can be killed, recovered, and re-inserted.

3. Investigate database as way to communicate dataflow state(s) between processes (for things like notifications for approaching backpressure)

Appendix 11 – Other Future Work

1. Understand the plan for aggregating data from SNB triggers; plan and prototype the changes needed
2. Consider the separation of configuration and readout functions in *artdaq* BoardReaders
3. Consider an interface at the BR-level to spy on the data out-of-band.
4. Correlated monitoring of DAQ parameters along with server and networking parameters
5. Development of training materials and documenting best practices
6. Collaboration on Dataflow Subsystem R&D projects (e.g. Giles' escalator protocol)
7. Contributions to SW development and packaging plans

References

Karol's 'Lessons Learned' talk at the CDR (Dec 2018)

Giovanna's 'Improve NP04' talk on 29-Jan-2019

Rob and Karol's 'protoDUNE planning' talks last week

My CDR talk (Dec 2018)

Backup

Some slides that I copied from my presentation at the CDR…

Internal Requirements

Support the overall DAQ requirements for livetime and uptime

Accept Trigger Commands and read out the data for the specified time windows and detector components

Provide the infrastructure for the High-Level Filter

Provide sufficient DAQ-specific data storage to avoid any disruption to data taking from a temporary loss of network connectivity to Fermilab

Provide ‘DAQ Kit’ software throughout the next 6+ years

Key Challenges

Provide basic functionality for test environments from 2019 thru Production phase.

Provide enhanced functionality to meet the needs of the production system, especially very high uptime

- Fault tolerance
- Dynamic reconfiguration
- Automated error recover
- Automated running of calibrations, either as part of normal data taking or in parallel

Development Plan

Two thrusts:

1. Development and support for VST and integration test environments, data taking at protoDUNE, ‘DAQ Kits’
2. Development and validation of required new features

Plan is to take advantage of this situation:

- Evolutionary changes to the working Back End Dataflow software/system at protoDUNE
- This will provide the components for the DAQ Kits as well as provide the test environments for new Back End Dataflow features
- Incremental changes within the Back End DAQ Dataflow will reduce integration issues

Development Plan

Short-term plans:

- Design discussions at next level of detail
  - For example, incorporating the conclusions reached by the Data Model Task Force into the Back End Dataflow design
- Make improvements to protoDUNE DAQ based on lessons learned
- Work on features needed for DAQ Kits (reliability, usability, visibility to sources of errors, etc.)

Development Plan

Longer-term plans:

- Dynamic reconfiguration
- Fault tolerance
- High reliability
- Extensive operational monitoring
- Auto-recovery mechanisms