SRO at JLab - Status

- **Objective:** Support the traditional triggered model along with streaming in one integrated DAQ framework.
  - Leverage existing hardware. Add firmware and software to support Streaming.
  - Focus on the parallels of the the JLab framework with the planned ePIC DAQ.

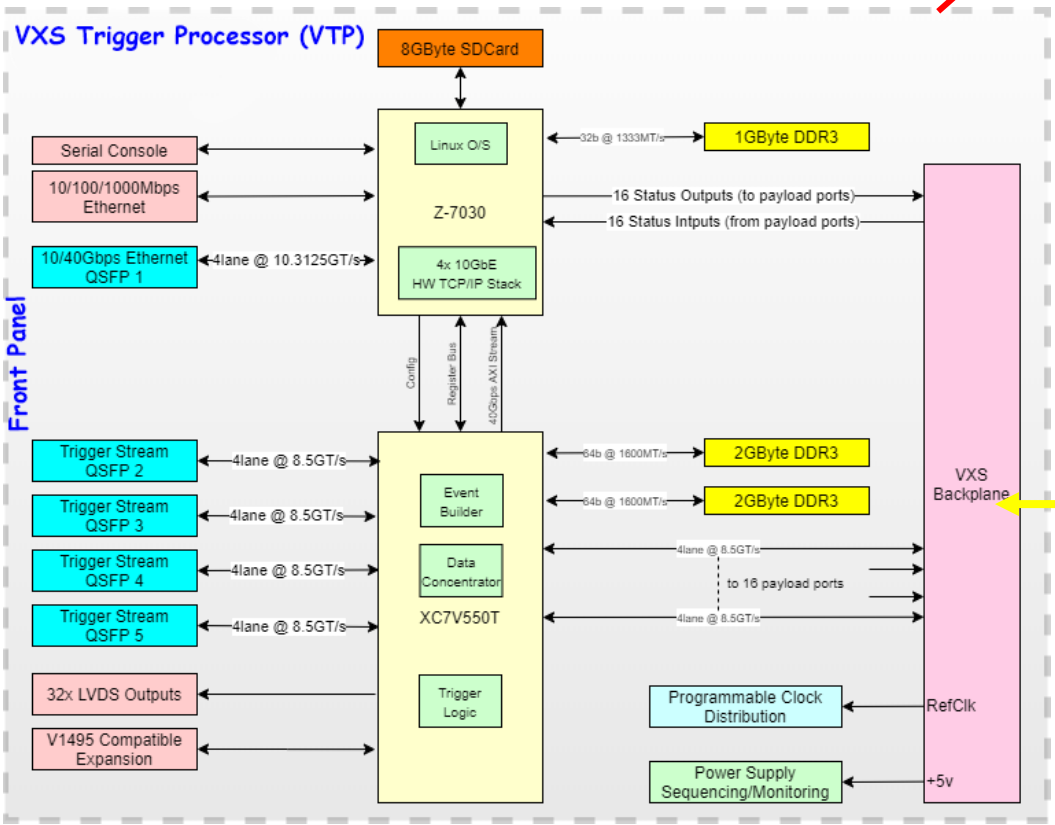
- General info on data formats we are using

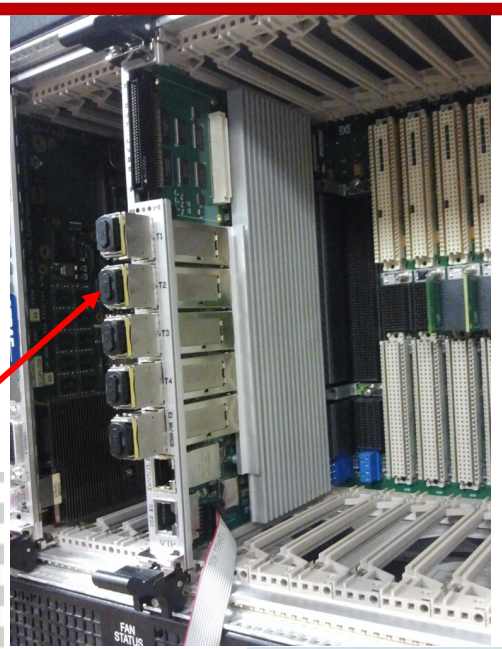
- Next steps…

JLab VXS Platform

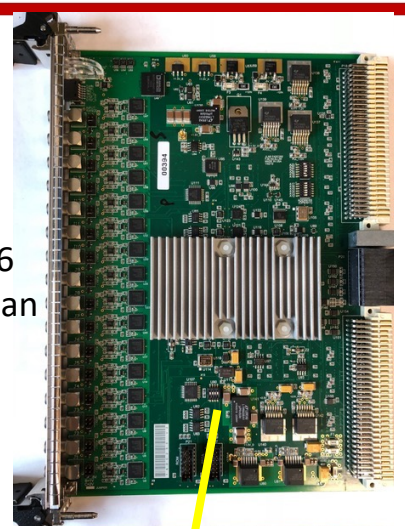
Linux OS on the Zync-7030 SoC
(2-core ARM 7L , 1GB DDR3)
10/40Gbps Ethernet output

Xilinx Virtex 7 FPGA
20x4 serial lanes from both the VXS backplane and the Front panel. 4GB DDR3 RAM



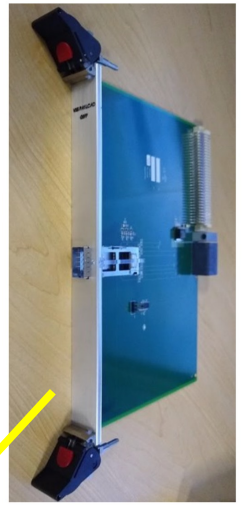


RDOs
Local Remote

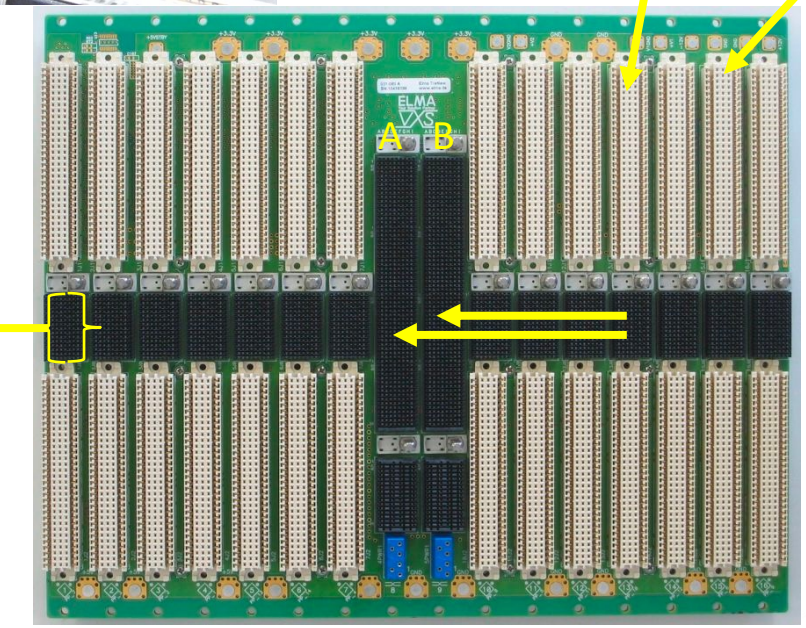


16 Chan

JLAB 250MHz FADC



QSFP Extender

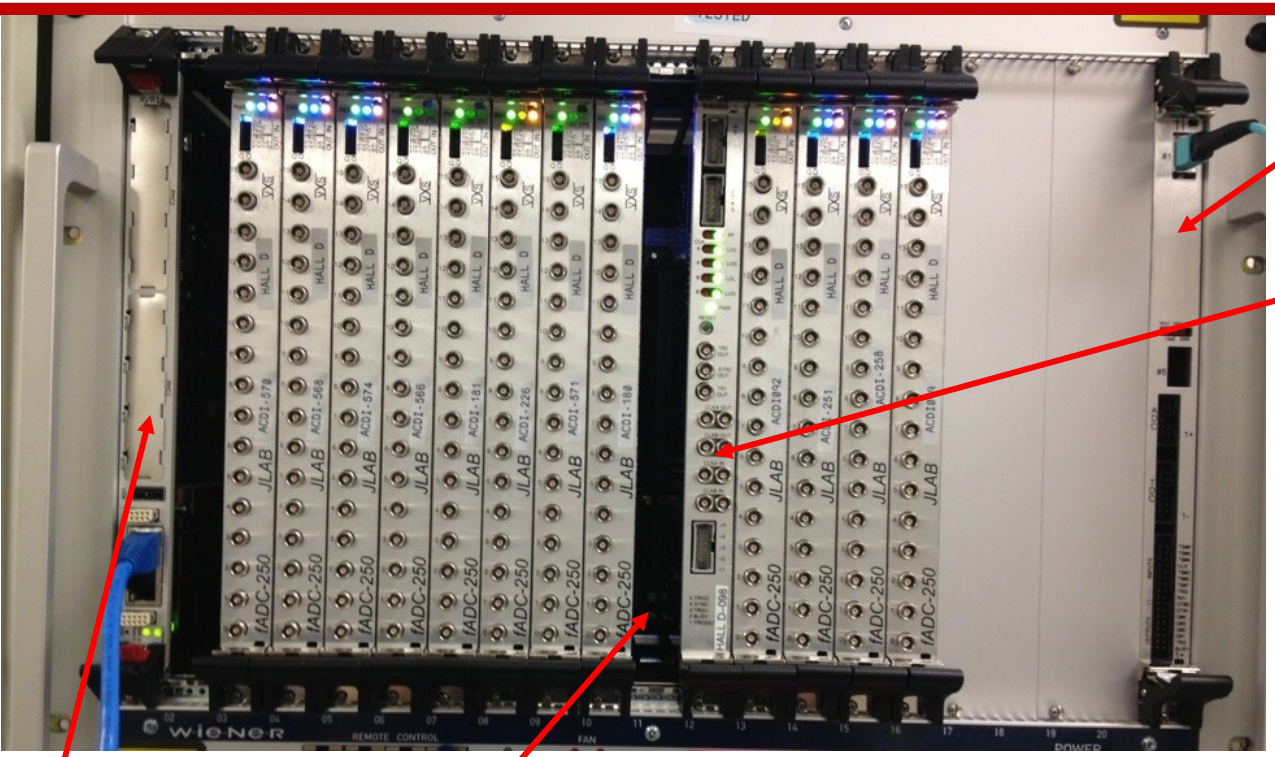


16 payload ports supported

VXS Backplane
4 serial lanes to each switch slot – **20 Gbps**

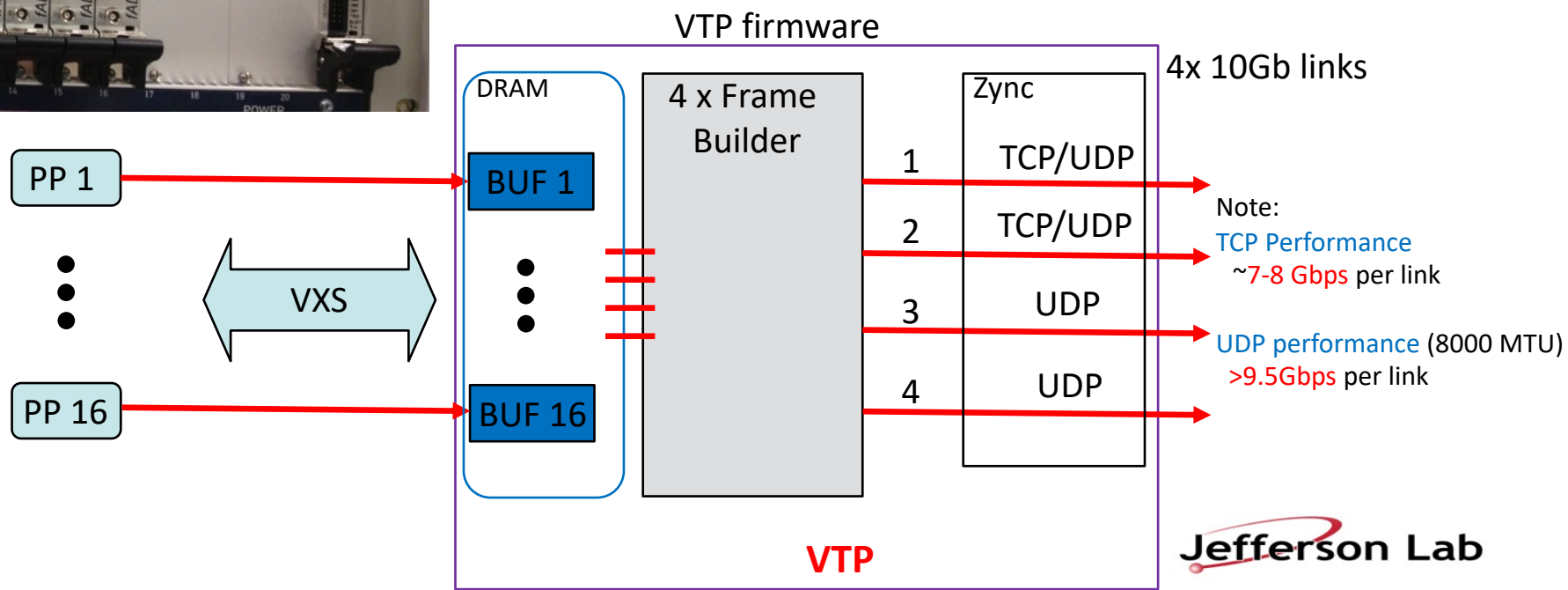
Jefferson Lab

Single Crate System



Trigger Interface Module (TI)
Local clock/trigger/sync source or remote (via MTP fiber)

Signal Distribution (SD)
Delivers all TI signals to each payload port and payload busy back to the TI

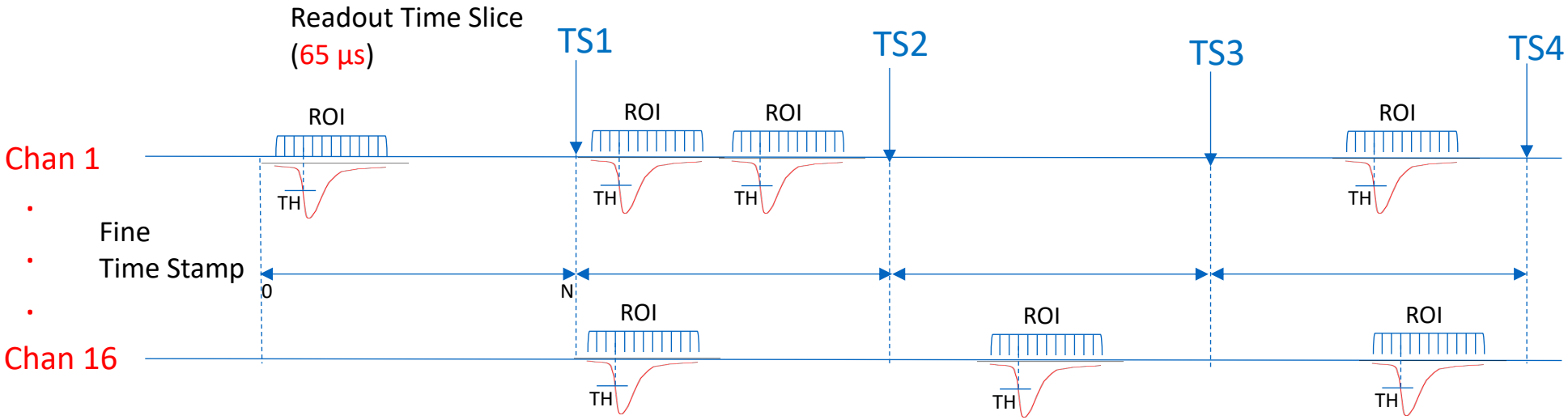


Jefferson Lab

JLAB FADC as an RDO (Streaming mode)

A 250 MHz FADC generates a 12 bit sample every 4ns. That's 3 Gb/s for one channel. 16 channels is 48 Gb/s. Currently, we identify a threshold crossing (hit) and integrate charge over a ROI and send only a **sum** and **timestamp** for each hit.

Available VXS bandwidth will allow for 1 hit every 32ns from all channels sent to the DAM.
A data frame (**Time Slice**) for all available hits is generated in the DAM (currently every 65µs



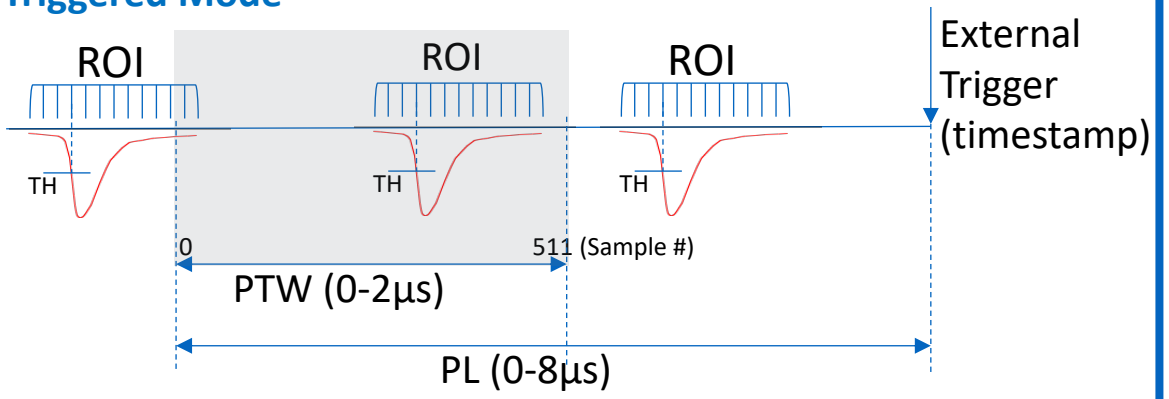
2 FPGAs manage the FADC processing

Note: The FADC can still simultaneously operate in triggered mode with an 8µs pipeline and 2µs readout window. With readout over the VMEBUS

Jefferson Lab

FADCs - Triggered vs Streaming

Triggered Mode

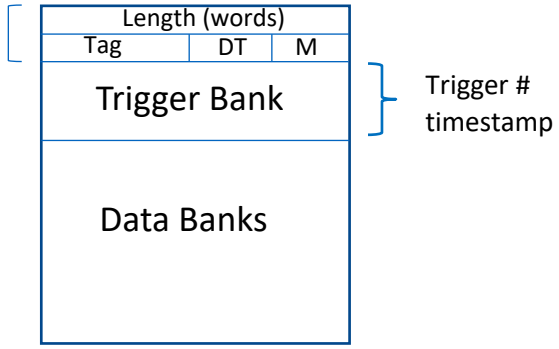


PL: Programmed Lookback
PTW: Time window

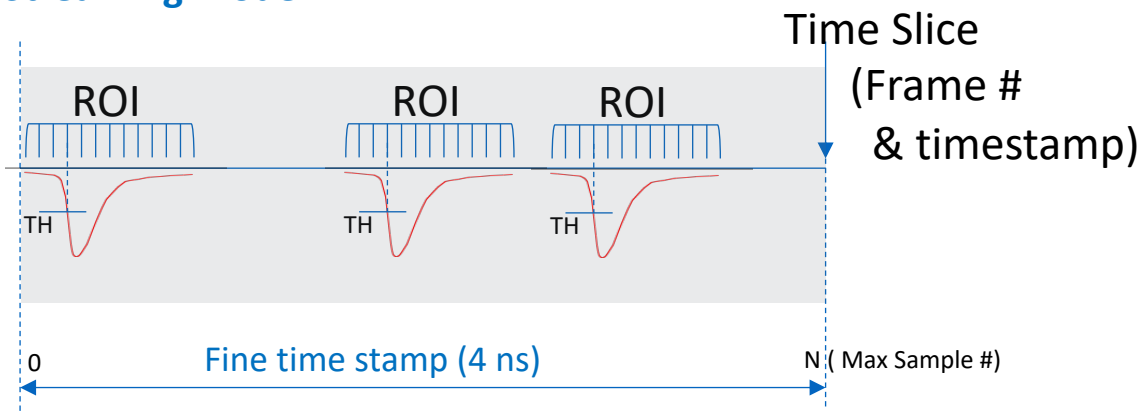
Data we get on a trigger:

- All waveform samples for the ROI
- Threshold Sample # (hit time within the PTW)
- Trigger absolute timestamp

ROC Data Format



Streaming Mode

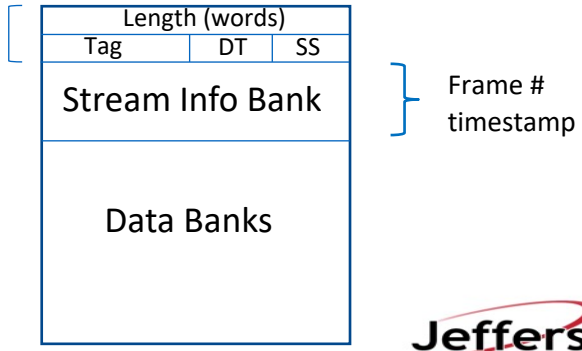


1 Frame = N Clocks (up to 16bits, currently 65536 ns)

Data we get for a Frame:

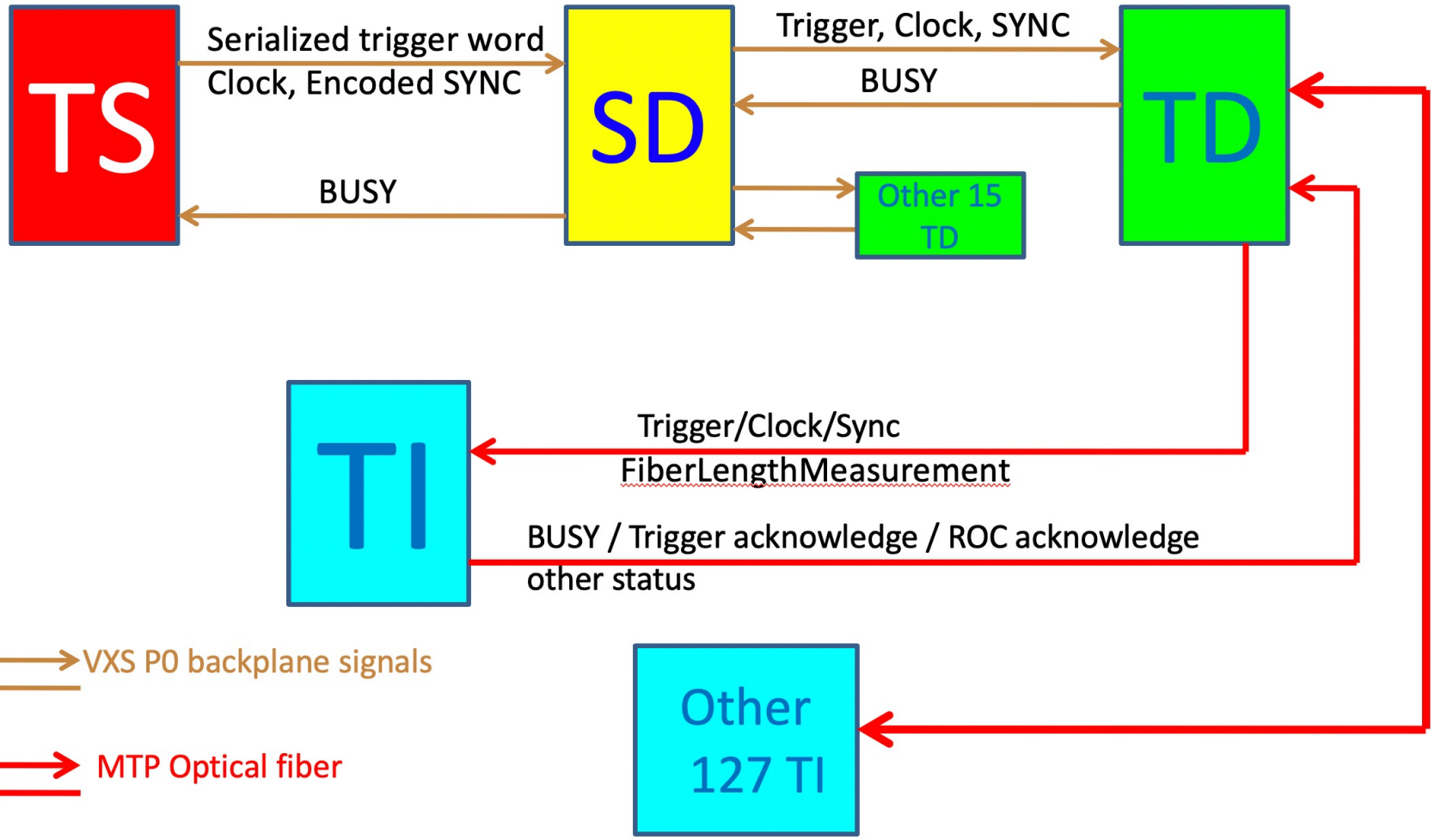
- Pedestal subtracted **sums** over an ROI for every hit over threshold
- Threshold sample # (fine time stamp for each hit)
- Frame # and absolute timestamp for the frame

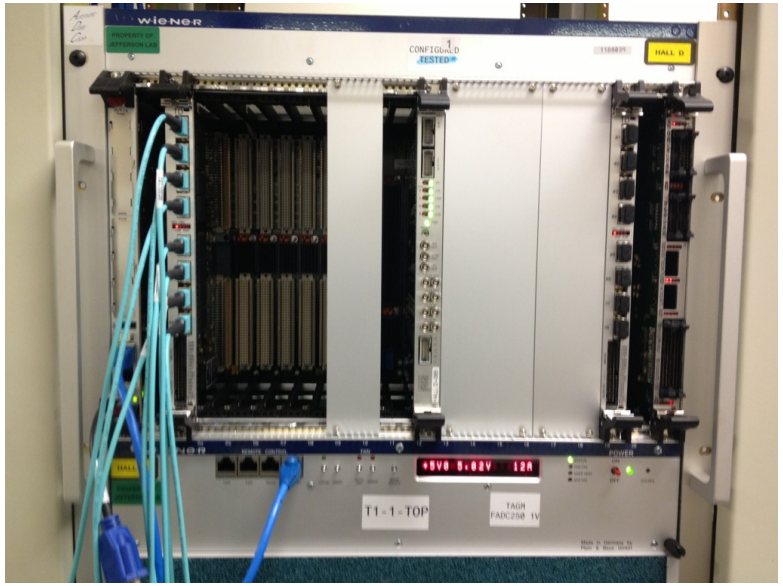
ROC Data Format



Jefferson Lab

JLab Timing System Components (GTU)

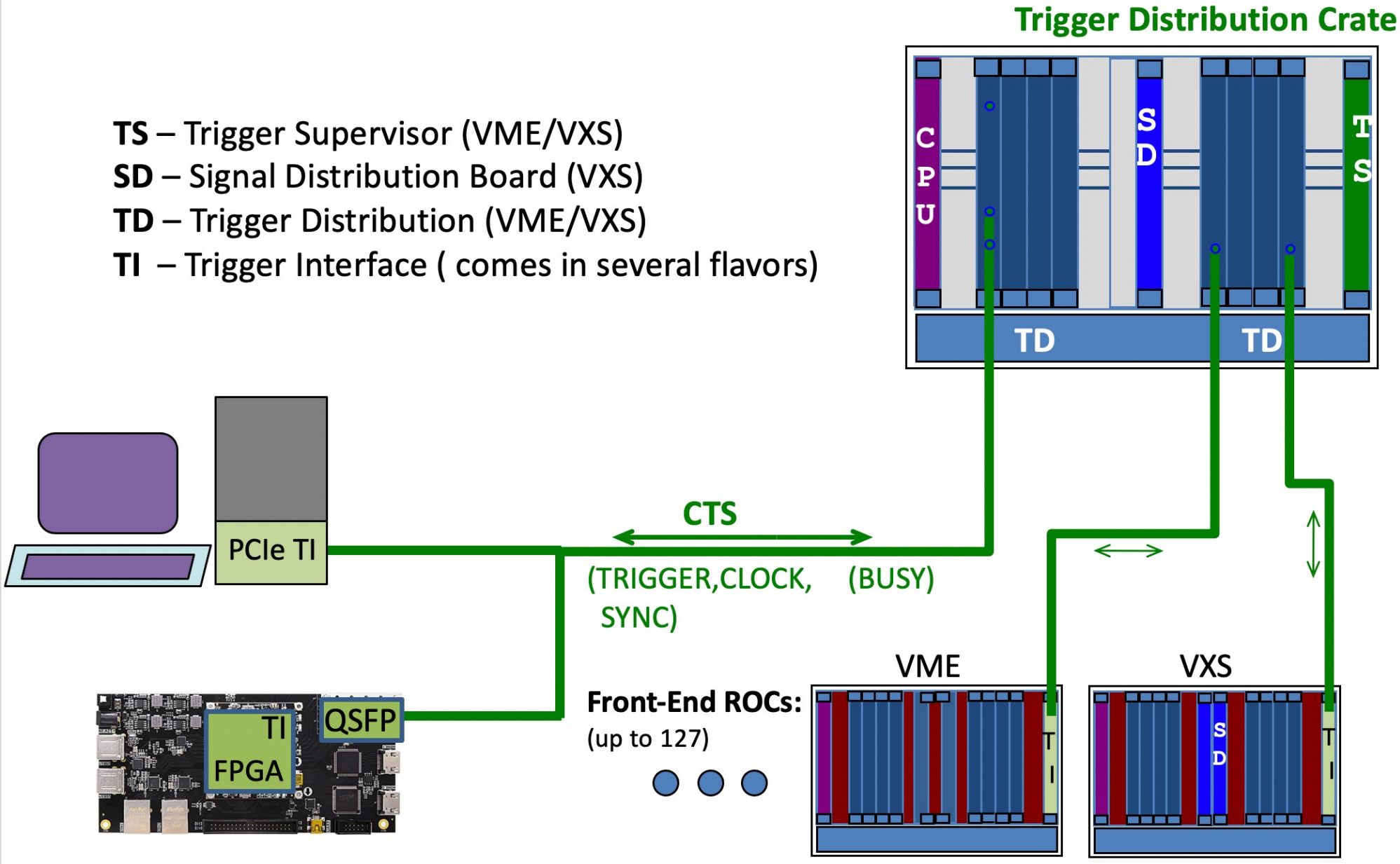




Jefferson Lab

JLAB Clock and Trigger Distribution System

TS – Trigger Supervisor (VME/VXS)
SD – Signal Distribution Board (VXS)
TD – Trigger Distribution (VME/VXS)
TI – Trigger Interface (comes in several flavors)

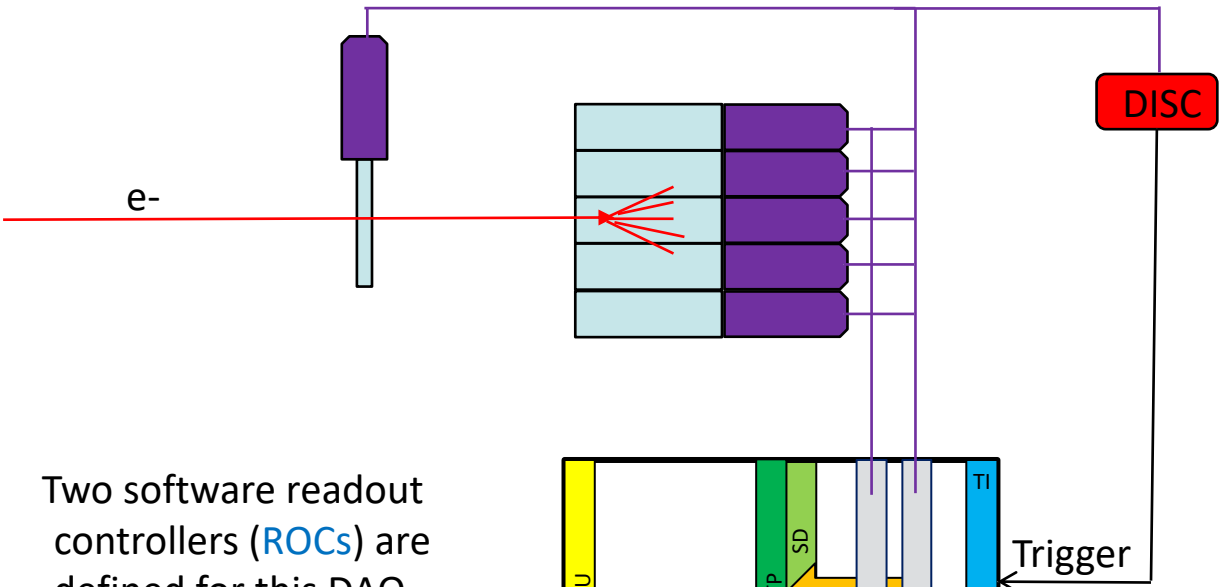


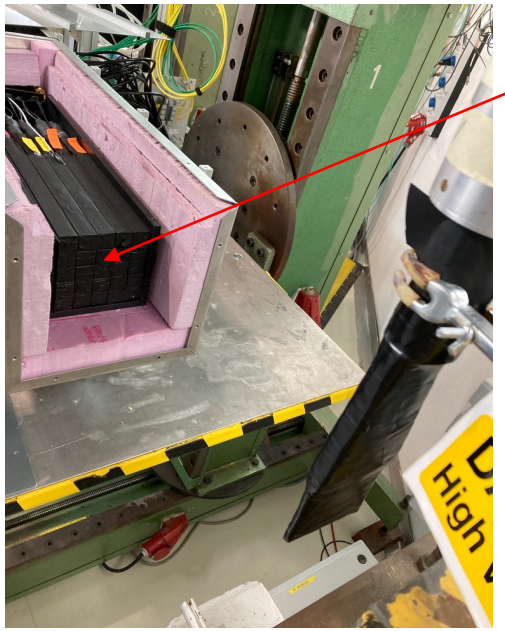
For large DAQ systems with many front-ends (DAMs)

The TI board can be used as a TS for small (up to 9) front-ends

Jefferson Lab

Simple Hybrid DAQ System



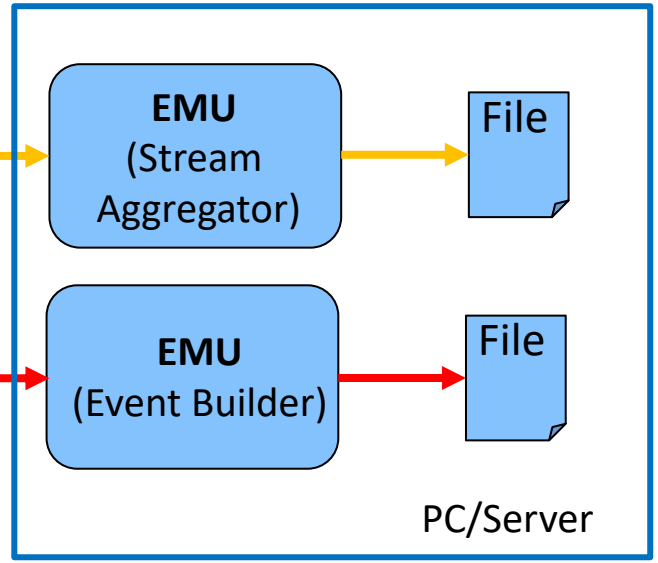


5x5 lead glass array (+ phototubes)

Two software readout controllers (ROCs) are defined for this DAQ System

VME ROC
Runs on the VME CPU
Reads out Triggered data

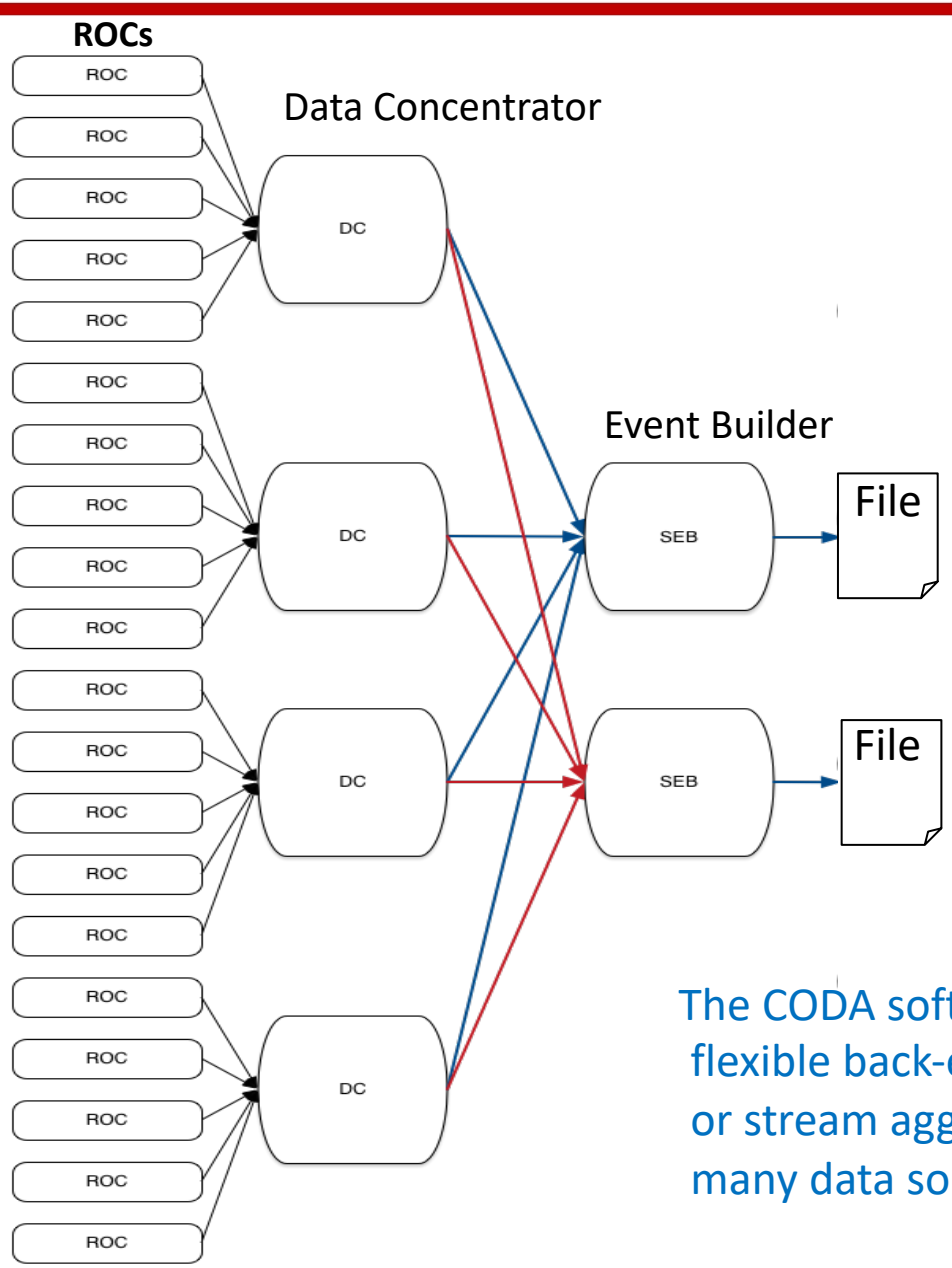
VTP ROC
Configures and manages the stream data



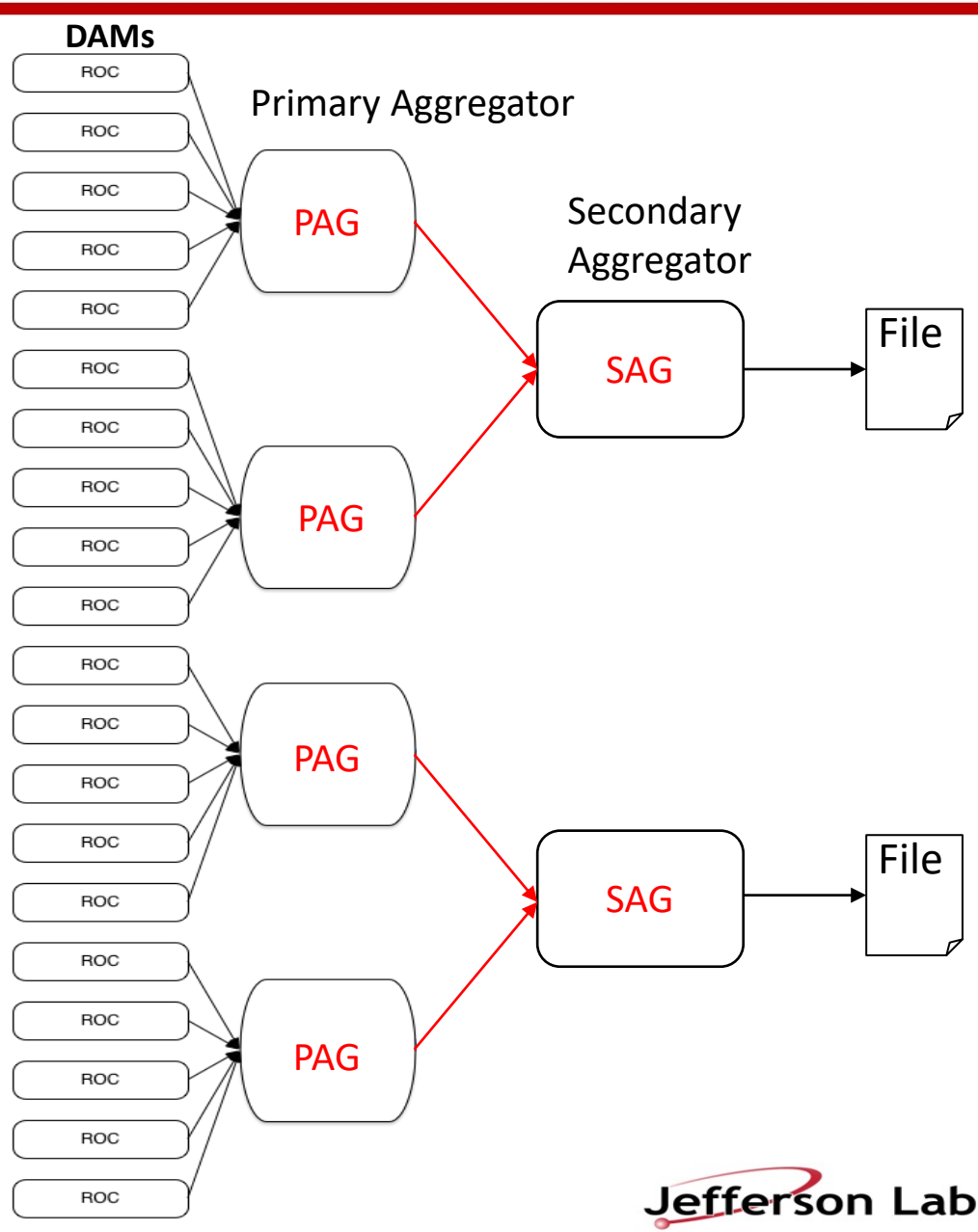
Jefferson Lab

Scaling up the DAQ

Triggered DAQ



Streaming DAQ



The CODA software toolkit allows flexible back-end event building or stream aggregation from many data sources.

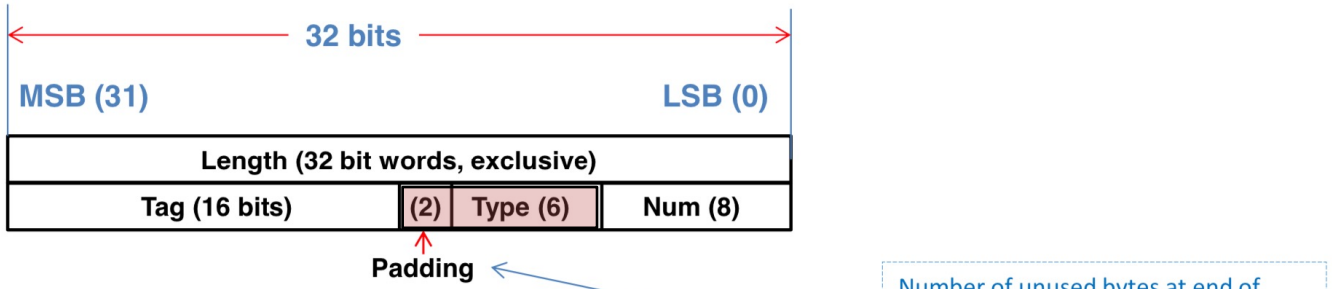
Jefferson Lab

EVIO Primitive Data Structures

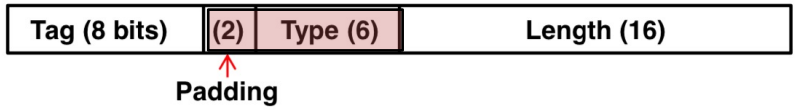
- EVIO data formats are based on 32 bit words

Evio Header Formats

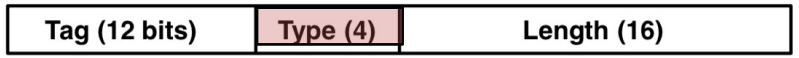
Bank :



Segment :



Tag Segment :

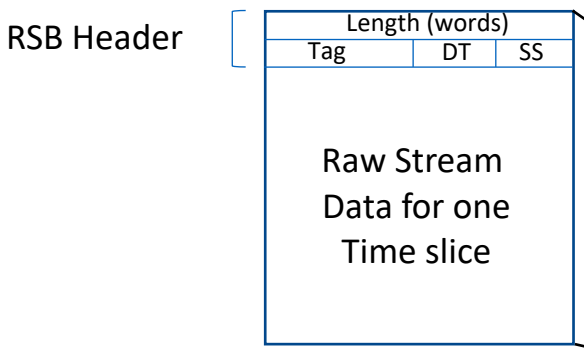


Evio Content Type Codes

Content Type	Primitive Data Type
0x0	32 bit unknown (not swapped)
0x1	32 bit unsigned int
0x2	32 bit float
0x3	8 bit char* (string)
0x4	16 bit signed int
0x5	16 bit unsigned int
0x6	8 bit signed int
0x7	8 bit unsigned int
0x8	64 bit double
0x9	64 bit signed int
0xa	64 bit unsigned int
0xb	32 bit signed int
0xc	Tag Segment
0xd	Segment
0xe	Bank
0xf	Composite
0x10	Bank
0x20	Segment

Stream Aggregation – Data formats

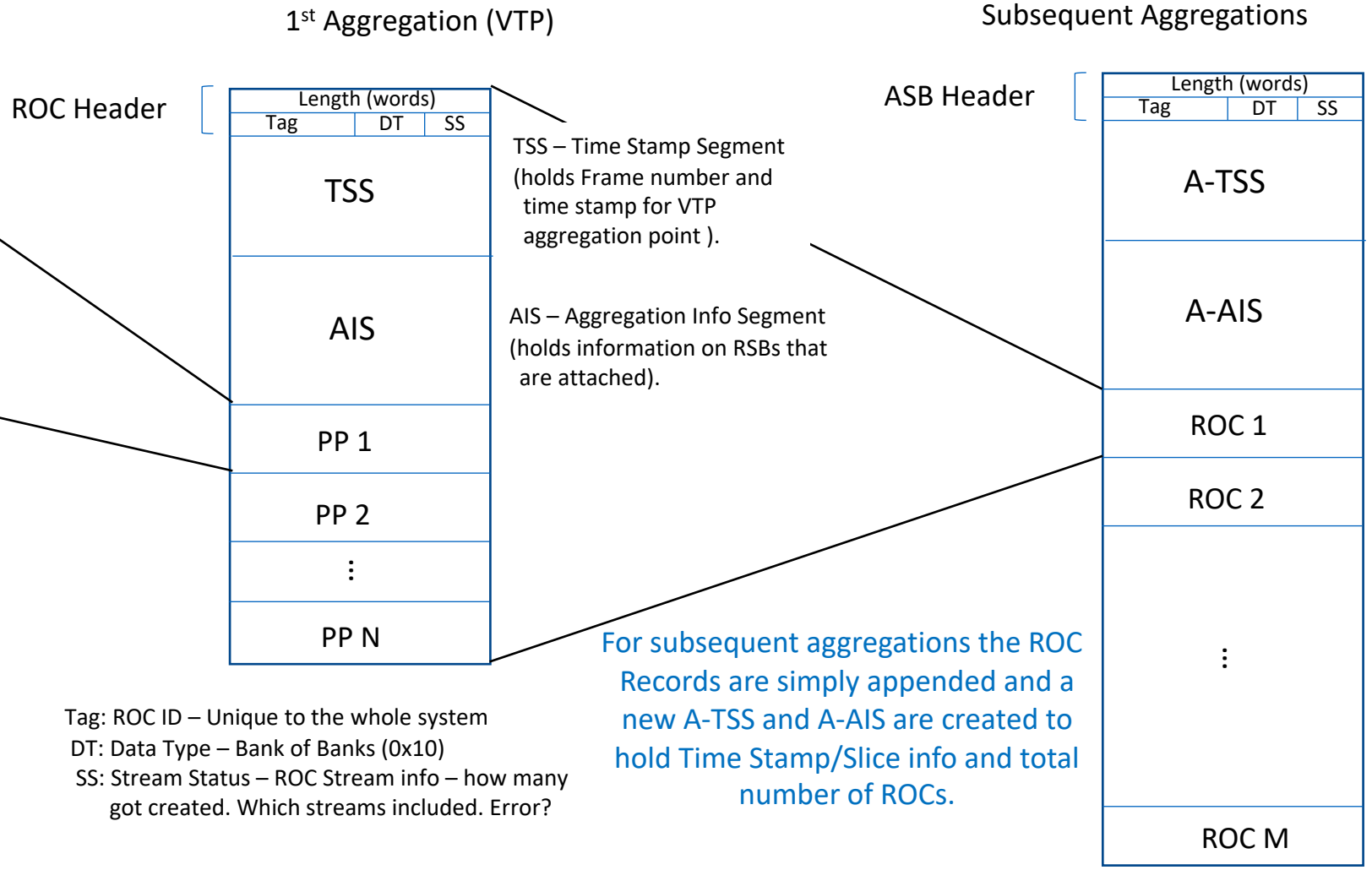
Raw Stream Bank (RSB)
(Payload Port – e.g. FADC)



Tag: Payload port ID – Unique just to the FEE/Digitizer Module Detector/Channel info
DT: Data Type – User/FEE specified
SS: Stream Status – Raw Bank, Did data get dropped? Clock info, Error?

Note: It is important for the "Front-End" Electronics/Digitizers to be integrated with the timing distribution.
- Data Frames are defined by the system clock
- Local time stamps are synced with the system
- Can streams be started and stopped synchronously?

Aggregate Stream Records

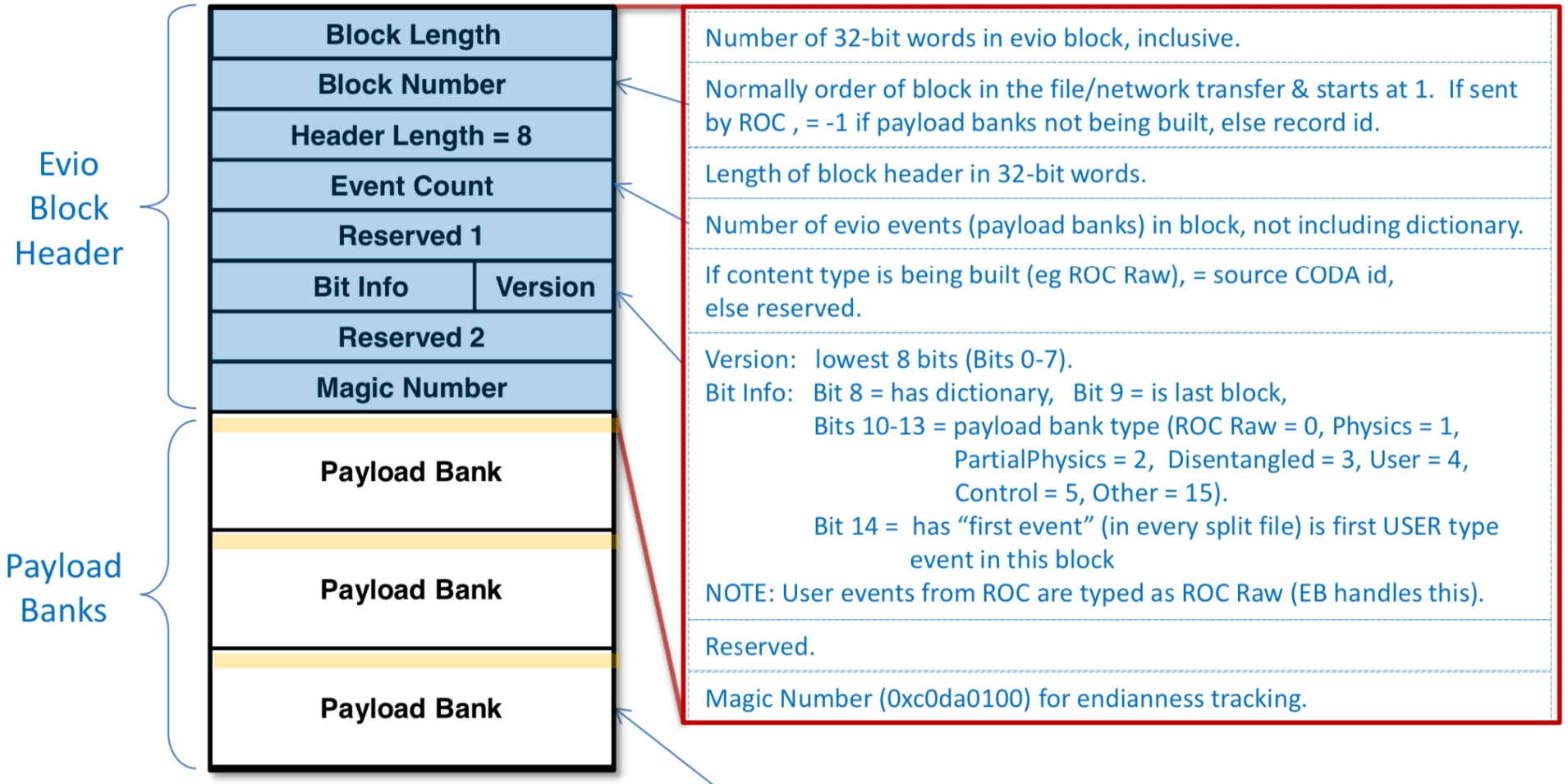


Tag: ROC ID – Unique to the whole system
DT: Data Type – Bank of Banks (0x10)
SS: Stream Status – ROC Stream info – how many got created. Which streams included. Error?

For subsequent aggregations the ROC Records are simply appended and a new A-TSS and A-AIS are created to hold Time Stamp/Slice info and total number of ROCs.

Jefferson Lab

CODA Data Transport and Files



Format used when sending all types of online CODA data over the network. They are in standard evio buffer/file output format with block headers.

Jefferson Lab

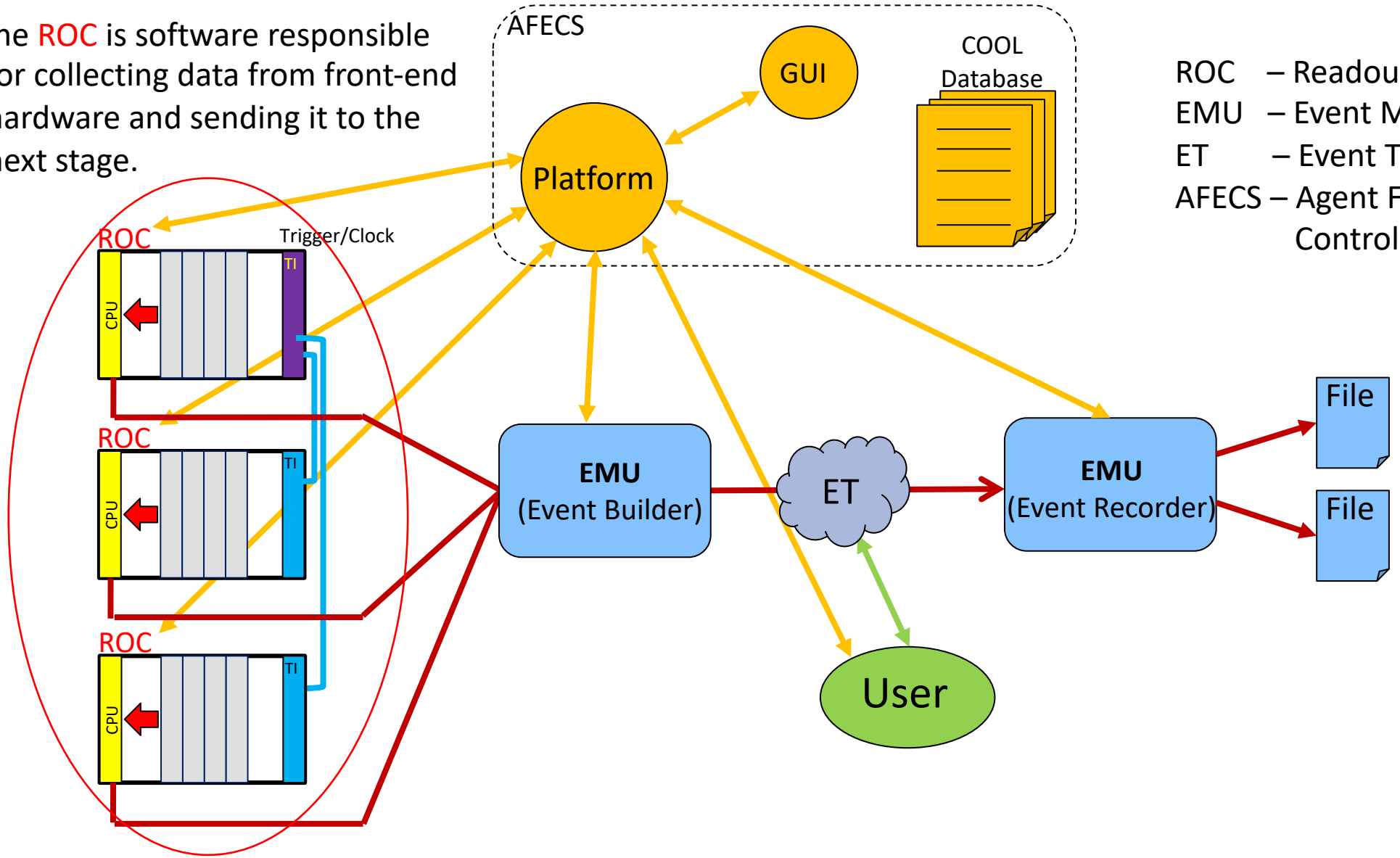
Where do we go from here?

- Expand current JLab FADC functionality to support waveform readout.
  - This will require new congestion management on the FADC FPGA.
- Set up larger scale (multiple crate) streaming tests using existing detector systems in the experimental halls.
  - This will give analysis software groups some real stream data to chew on.
- Integration of other ASIC-based front-end electronics within the JLab streaming environment still needs to be developed (“remote” RDOs).
- Migrate existing firmware and software libraries to new “DAM” hardware (e.g. FELIX 182).

Backup Slides

The CODA Data Acquisition Toolkit

The ROC is software responsible for collecting data from front-end hardware and sending it to the next stage.



ROC – Readout Controller
EMU – Event Management Unit
ET – Event Transport (Shared memory)
AFECS – Agent Framework Experiment Control System