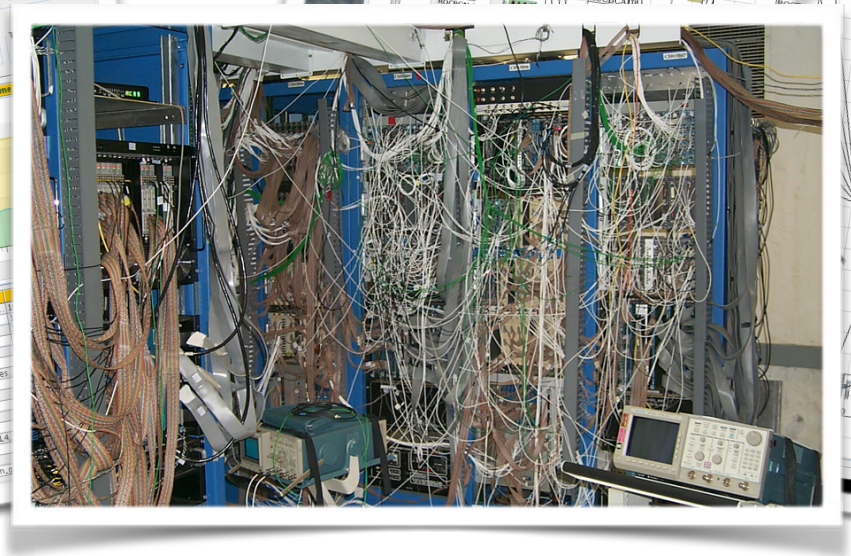
The future of DAQ

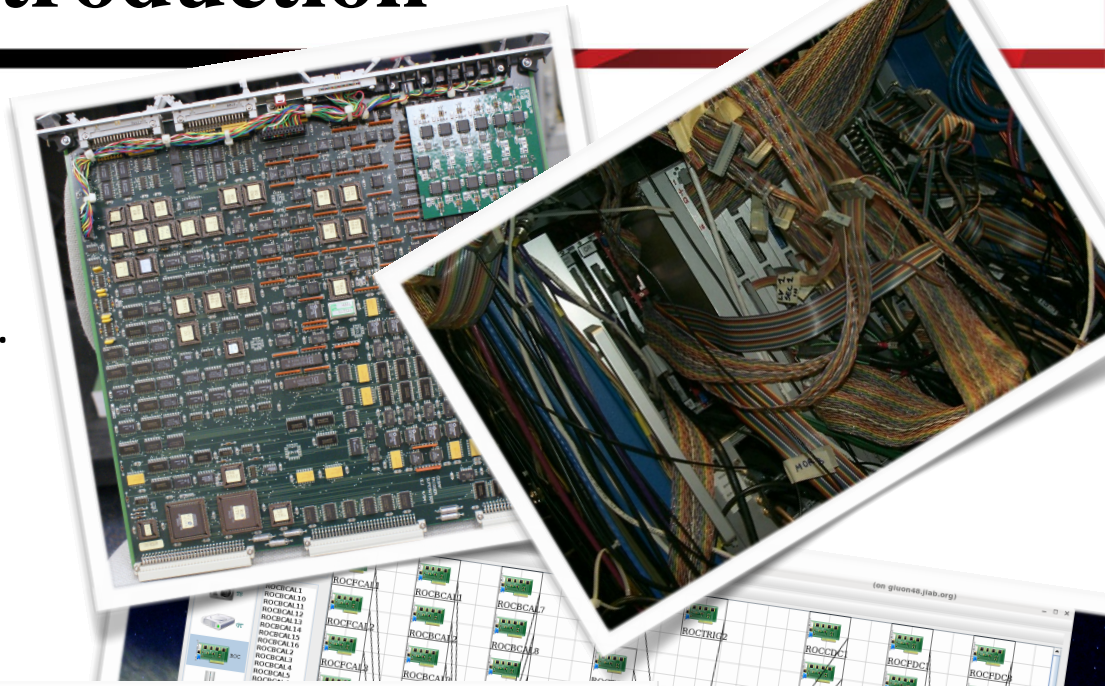
Graham Heyes

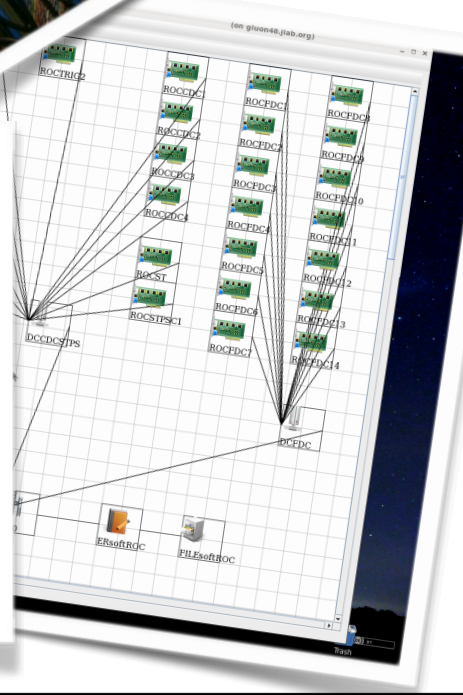
JLab Data Acquisition Support group

Introduction

- How JLab does things.
- Challenges to look forward to.
- Lessons learned to look back at.
- Solutions to think about.
- Concluding remarks.

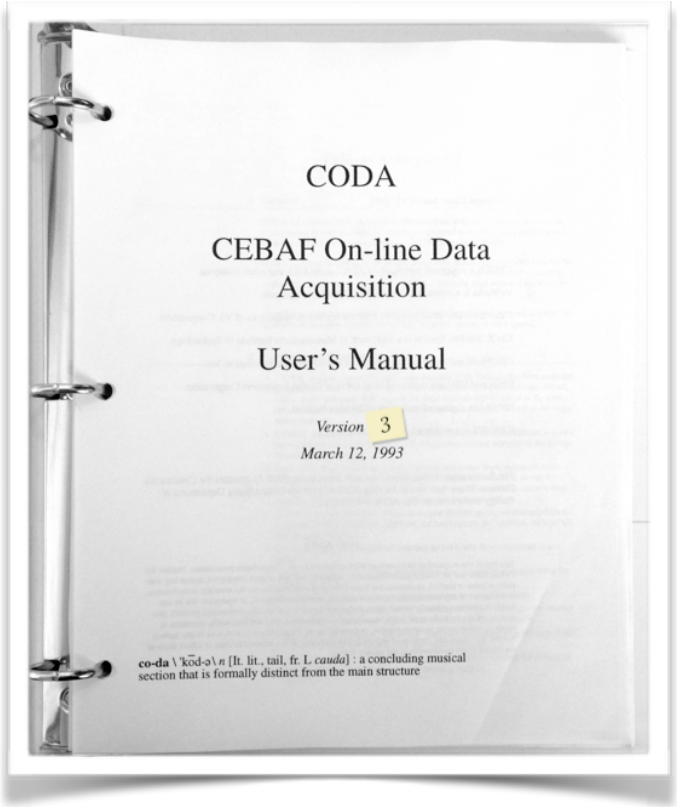




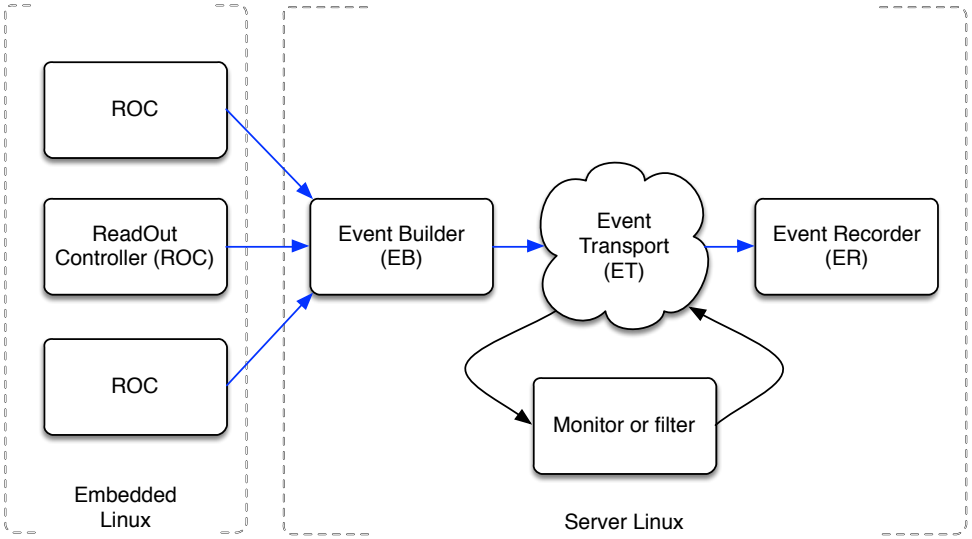


DAQ at JLab

- CEBAF Online Data Acquisition
  - Specification of recommended COTS hardware.
  - Suite of custom hardware.
  - Software:
    - Hardware drivers.
    - Embeddable Linux OS.
    - Readout software.
    - Event transport - ET.
    - Event building and storage.
    - Data format - EVIO.
    - Experiment control - AFECS.

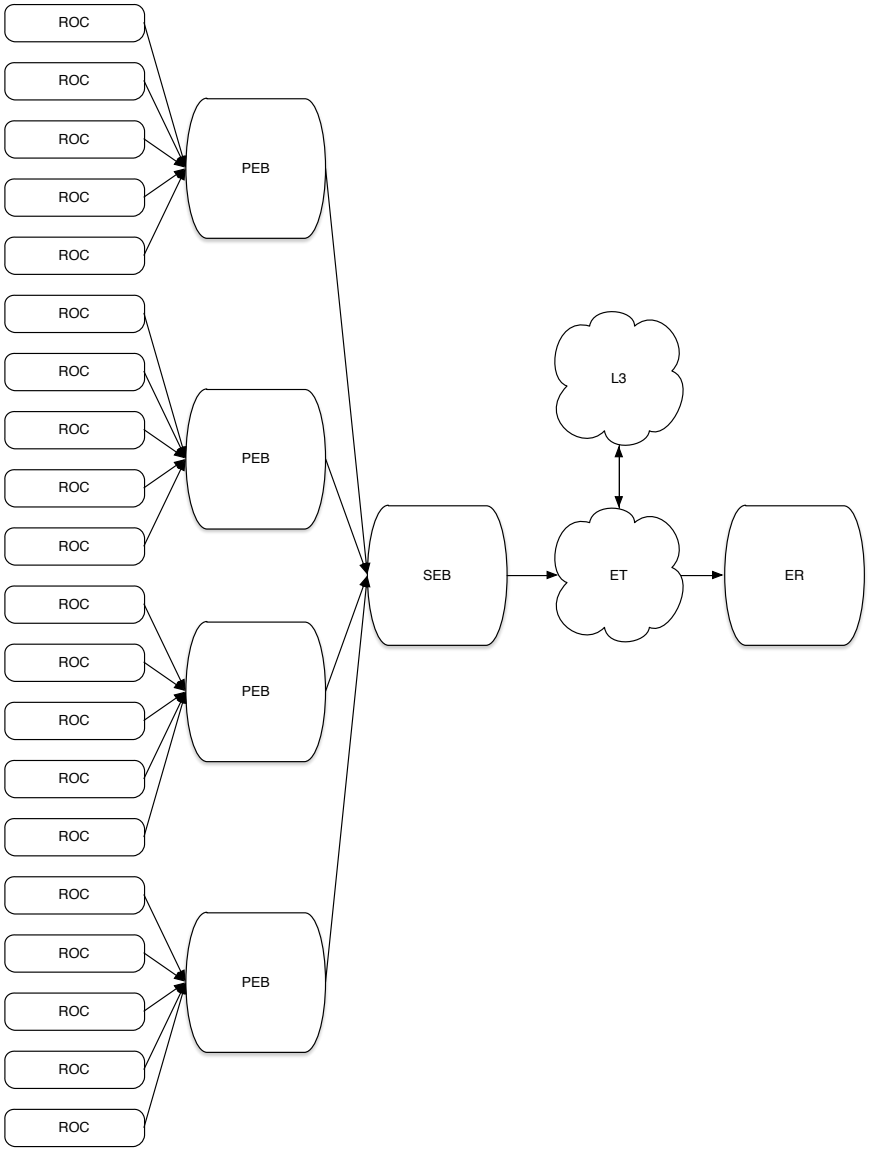


CODA, core concept



- Modular software, CODA components, with common core functionality.
- ReadOut Controller (ROC) running on embedded Linux in VME.
  - Receives trigger
  - Formats raw data
  - Sends data to EB over network.
- EB builds events from multiple ROCs.
- ET - “buffer manager”
  - Hooks for monitor & filter.
- Event Recorder - writes to disk.

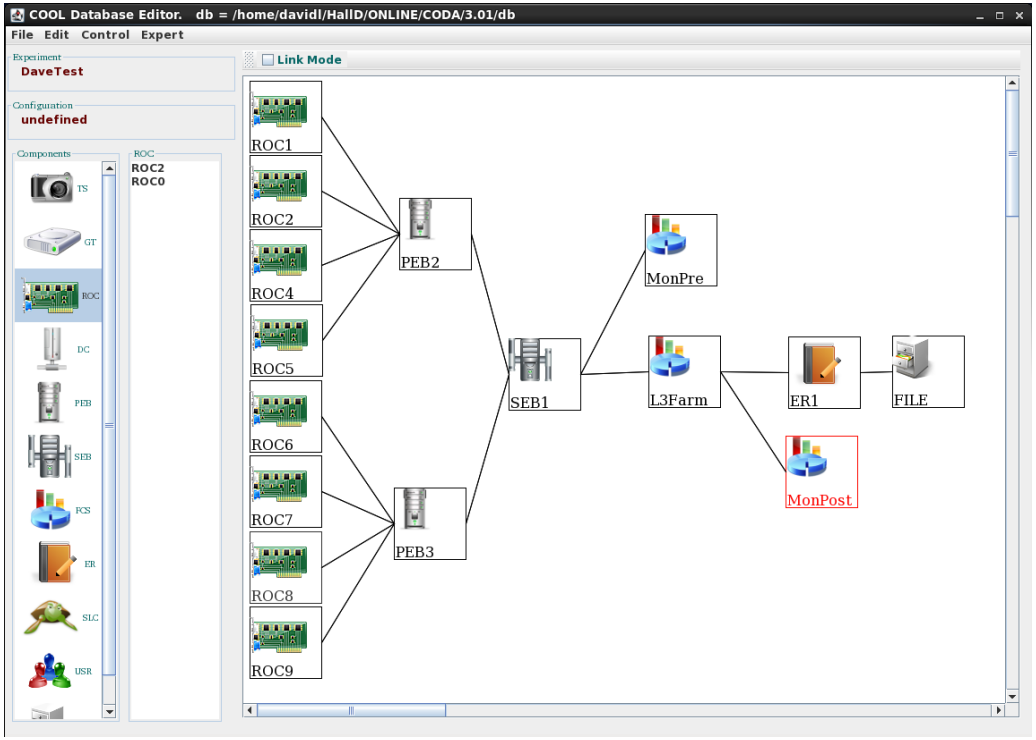
Scalability



- Modularity allows us to handle complex systems by breaking large problems into smaller chunks.
- In GLUEX instead of one EB handling fifty ROCS we have four Primary EBs (or Data Concentrators) doing partial building and one Secondary EB completing the build.

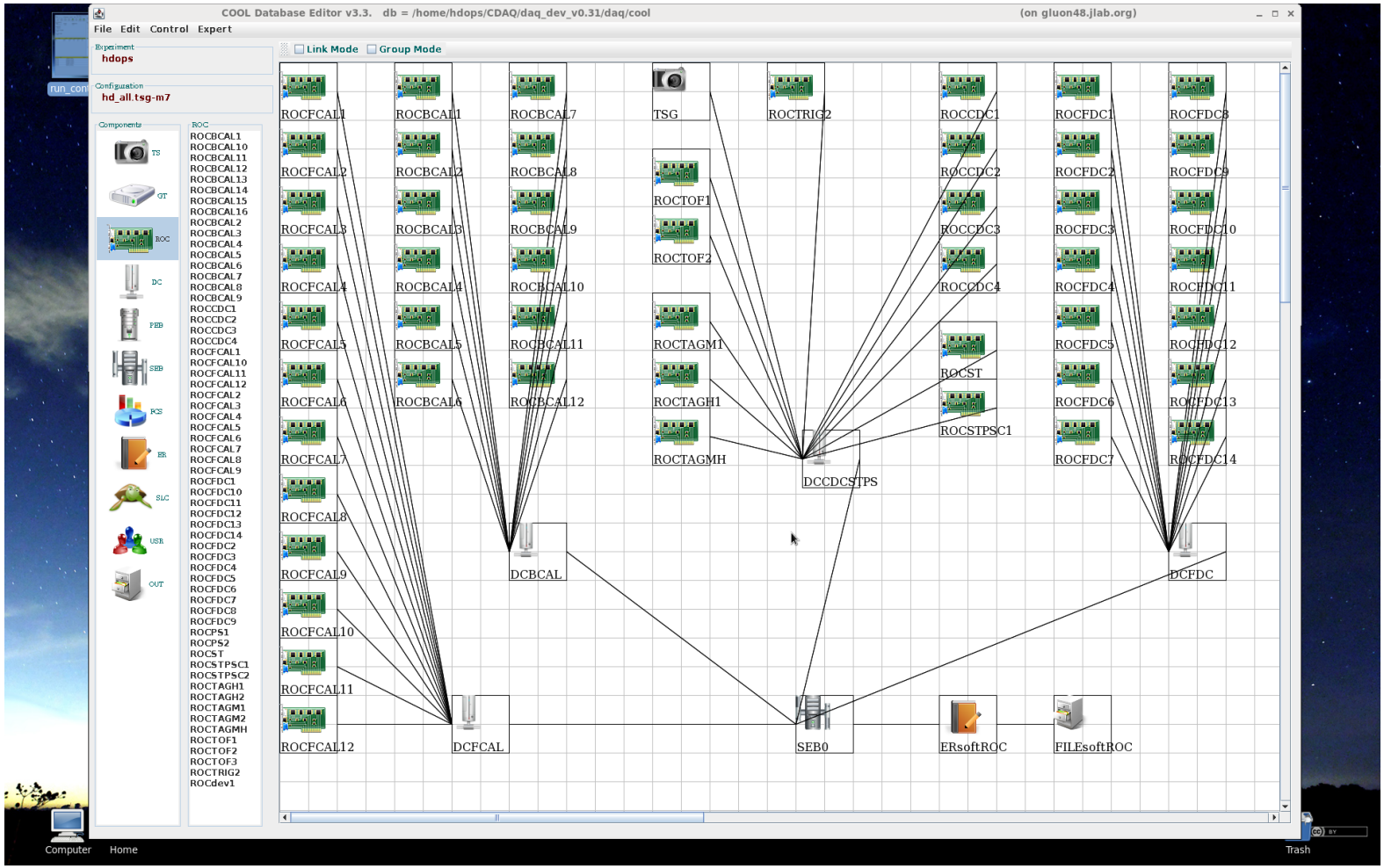
Complexity

- With complex configurations there are a lot of parameters to keep track.
- We developed an RDF based language, COOL, to describe data acquisition systems.
- Since COOL itself is complicated a graphical editor to allows configuration without knowing the language.



A large system

Editing a real GLUEX configuration:
50 ROCS, 4 Data Concentrators, 1 Secondary EB, 1 Event Recorder.



AFECS

- CODA 1.X Run Control was a monolithic program written in C++.
  - Slow, locked to running on only one machine, scalability issues.
- I read a paper back in 2001 on distributed agent based systems.
  - Break a large problem, into small pieces.
  - System distributed by design.
- At the time there was no alternative to writing something new.
- Resulting work produced AFECS “Agent Framework for Experiment Control Systems”.

Publications

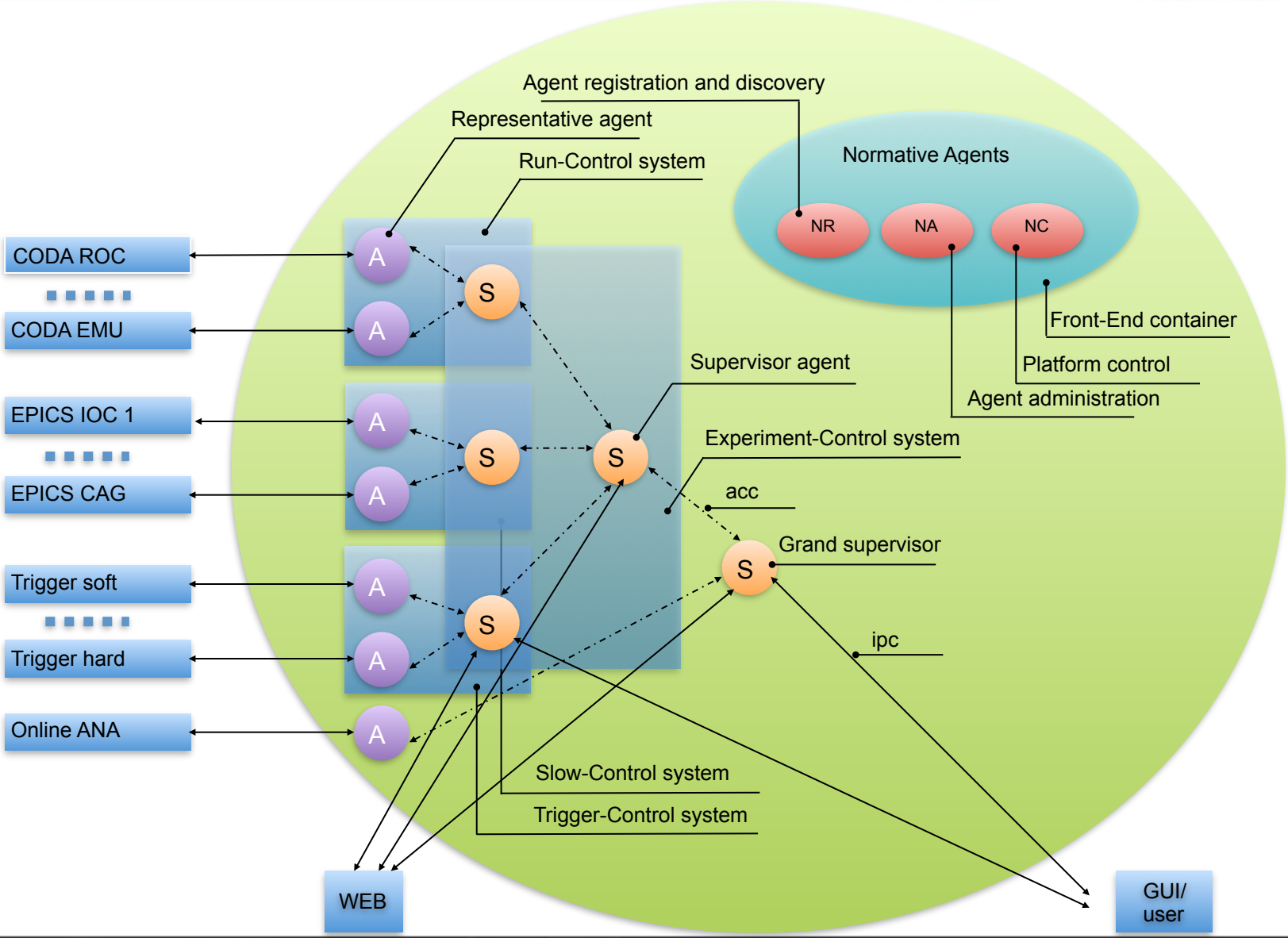
V. Gyurjyan, D. Abbott et al. “FIPA agent based network distributed control system”. arXiv:hep-ex/0305016. 12 May 2003

V. Gyurjyan, D. Abbott et al. “Jefferson Lab Data Acquisition Run Control System”. CERN-2005-002, Volume 1, pp 151-154

V. Gyurjyan, D. Abbott et al. “AFECS: Multi-Agent Framework for Experiment Control Systems”. Journal of Physics: Conference Series 119 (2008) 022025

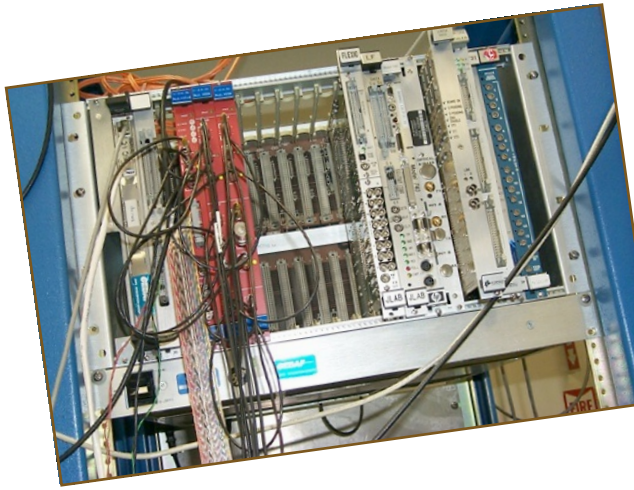
V. Gyurjyan, D. Abbott et al. “A Multi-Agent System For Building Large-Scale Distributed, Hierarchical Control Systems”. Proceedings of ICALEPCS 2009, Kobe, Japan.

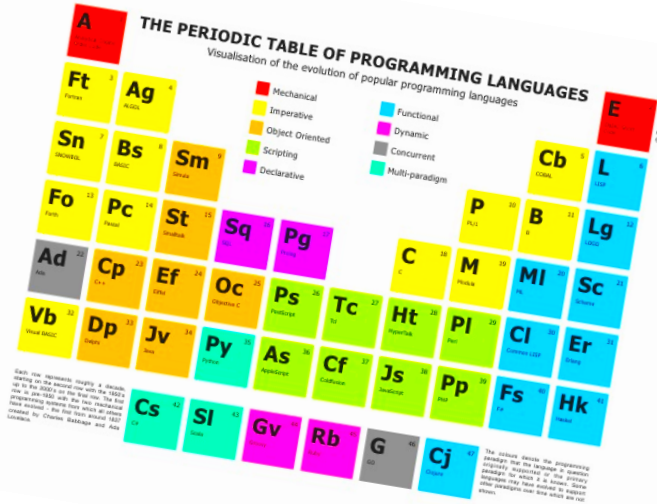
AFECS Based Experiment Control



Challenges to look forward to

- VME and busses in general
  - We like to avoid building custom hardware, vendors are dwindling.
  - Bus speed wasn't driven by nuclear physics but was always enough. This is changing and VME may not be any faster than it is now and may even be slower.
  - Move to standards like VXS - high speed serial fabrics rather than traditional parallel backplanes.
    - VXS works well for us but we are one of only a few adopters.
    - Any solution requires custom FPGA based hardware.
  - We are entering the post VME era.
- Software
  - Languages and techniques are constantly evolving, driven by industry. Need to be adaptable to new ideas.
  - Much of CODA is written in Java, where will that be in ten years?
    - Take care not to jump on a bandwagon with uncertain future.





Challenges continued…

- Experiments are moving towards very loose triggers, or no trigger at all.
  - G byte/s rates at the ROC level.
    - Firmware instead of software.
    - Data has to be tagged with a timestamp from a high speed global clock.
    - Bandwidth constraints on possible architectures.
  - Event building at these rates becomes a hard problem again.
  - Storage is currently a bottleneck
    - Heavy reliance on online data filtering and compression.
      - Large compute clusters online with high bandwidth network.
    - Media costs are reasonable but have to run many drives in parallel to store gigabytes per second.
    - Novel storage solutions? Maybe but can't see anything replacing tape soon.

Challenges continued…

- Experiment complexity is increasing.
  - Even with the COOL language configuration of a system the size of GLUEX requires care to avoid errors that impact data taking.
  - AFECS is a good step forward compared with the previous CODA run control but its inherent complexity makes debugging an issue.
    - When it works it works very well but when it doesn't it can be hard to find out why.
  - Transitions that used to be instantaneous take time.
  - User's find it harder to understand what is going on.
    - In distributed systems symptoms can be disconnected from causes.
  - Long term stability
    - Unpredicted interactions and behaviors.
    - "rare" once a shift events become once a run.

Lessons Learned

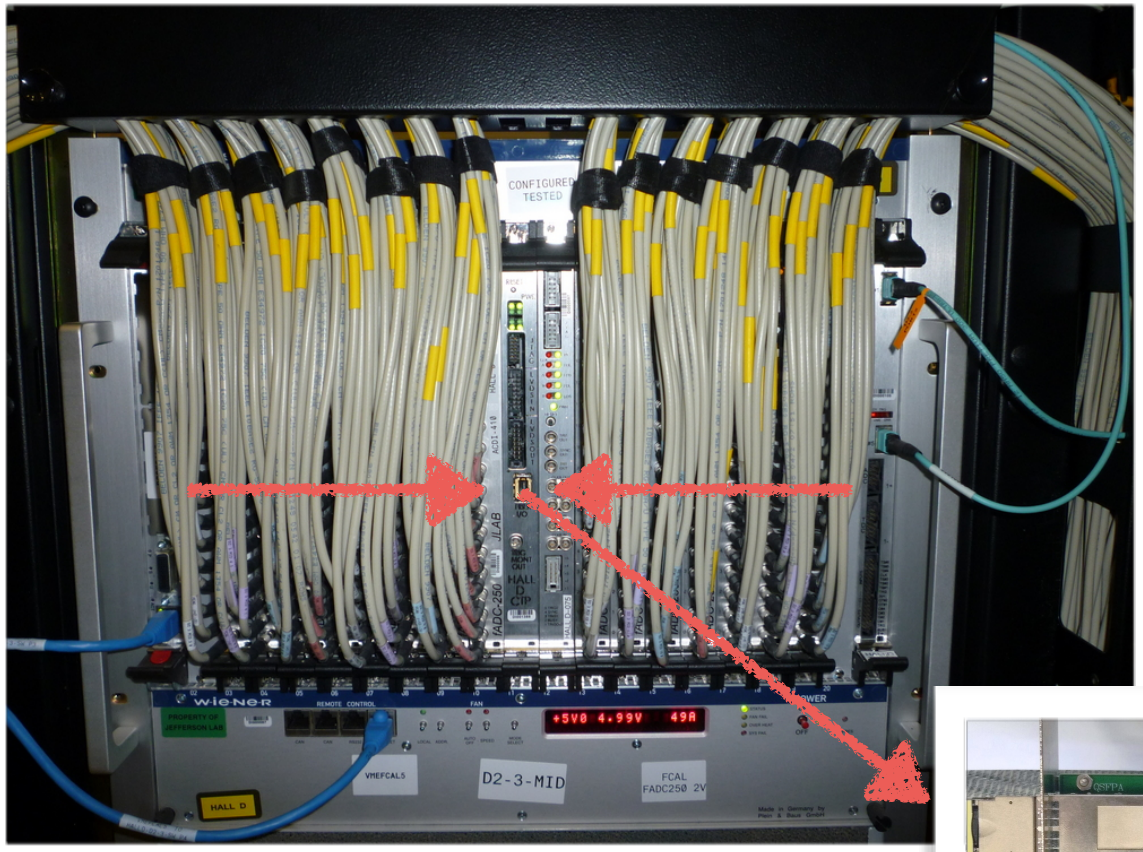
- Complexity
  - Too many features is not necessarily a good thing.
  - Don't panic the shift taker.
    - Interfaces and system behavior should be simple.
- Closed boxes and walled gardens can be a bad think.
  - Users want to runs scripts etc based on what the DAQ is doing.
  - Users want to put ad-hoc data into the data stream:
    - Data from slow control.
    - Data from interaction with the shift crew
- Users like using the same DAQ in the counting house that they used in testbeds.
- CODA has been very successful because it is modular
  - Scalable from small to large systems.
  - Parallelism is built in.

Solutions to think about

- Move the “business end” of the ROC from software to firmware.
- Avoid VME bus bandwidth limitations using a serial backplane or some other medium for data transfer.
  - Read data over serial backplane to a board in switch slot.
  - Firmware on board formats the data.
  - A high speed fiber link sends data downstream.
  - VME and CPU in slot 1 is only used for configuration and diagnostics.
- The CTP, crate trigger processor, board in use by GLUEX could easily be modified to perform this role.
  - under investigation for SoLID DAQ.

CTP board

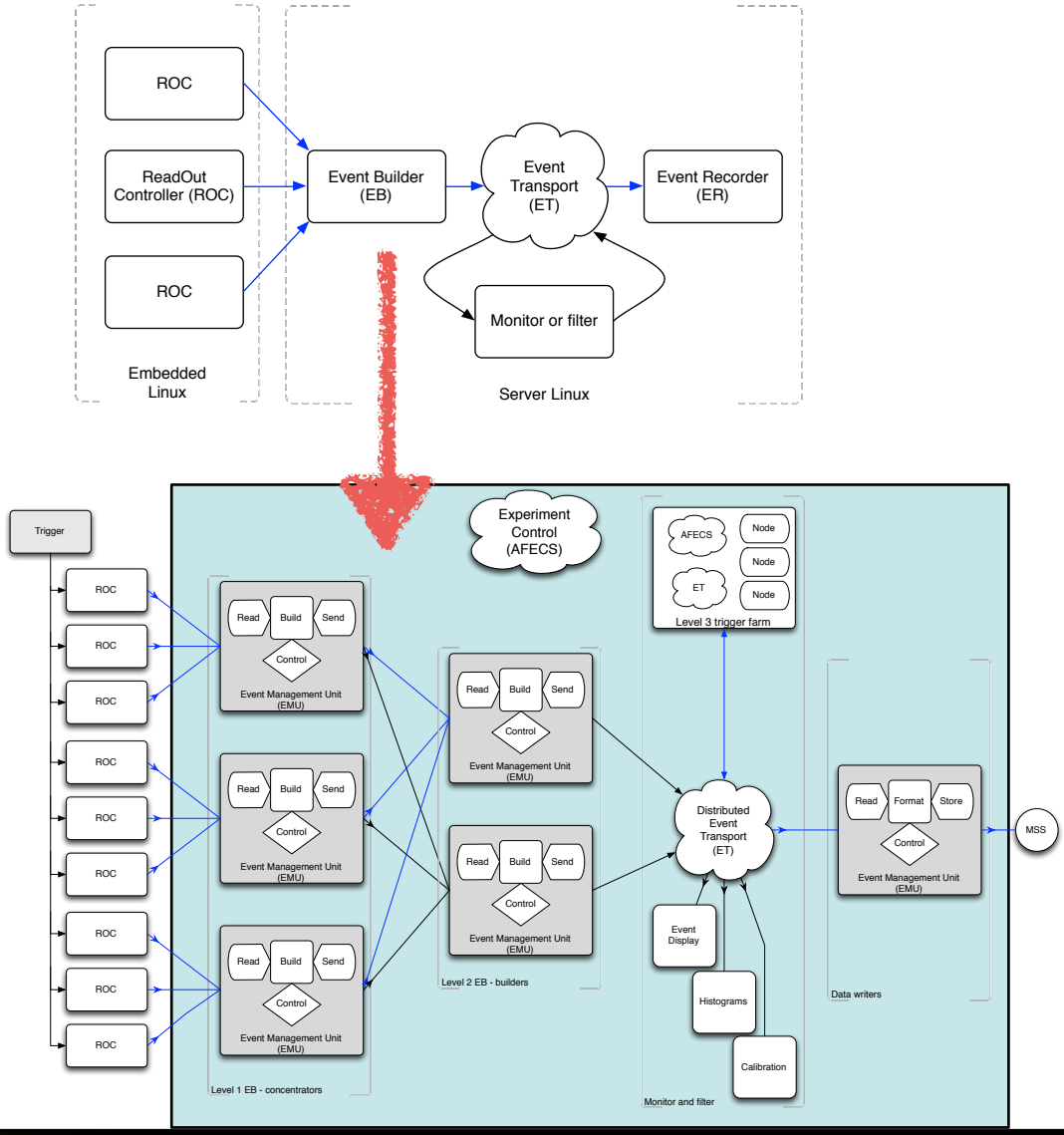
CTP and TI boards



- Designed as parts of a general purpose trigger.
- Data is transferred over serial backplane to CTP
- CTP sends data over fiber.
- PCI version of TI could be modified as a receiver of data from CTP.

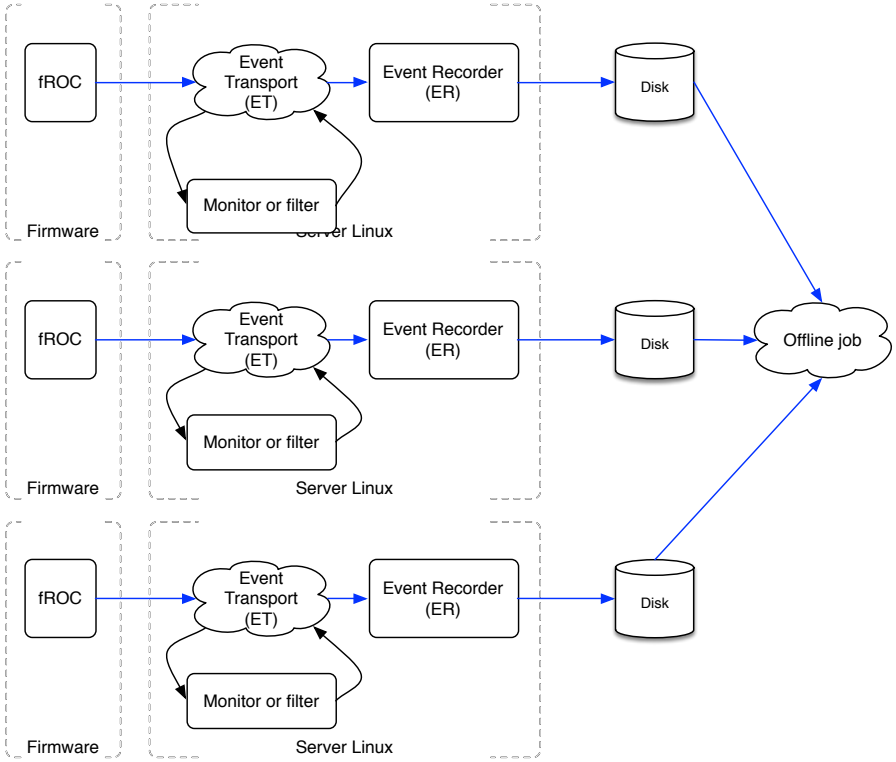
Not so novel architectures

- In a simple CODA system the EB, monitor and ER run on a single physical machine.
- In GLUEX the individual CODA components each run on their own machine with Infiniband links between them.
  - Network bandwidth
  - CPU load for event building
- Given a sufficiently fast machine we could regress and run everything on one machine.
  - Simpler configuration
  - More robust



Totally the opposite direction

- Instead of building events online store everything and figure it out later.
- Proposed for SoLID in one of it's modes with a high rate of single track events.
  - 30 segments of GEM detectors.
  - Effectively run 30 DAQ systems in parallel.
- Could work for other experiments with more complex events.
  - Event building and level III trigger in software "offline" or "near-line"



One step further

- A light field or plenoptic camera, captures the intensity of light and the direction that the light rays are traveling in space.
- Physical optics that generates a traditional focussed 2D image is replaced by software that calculates what the image would have been in a particular focal plane.

- Analogously a trigger-less detector could capture everything in a volume of space and “events” extracted from the data by pattern recognition.
- Don’t try to find tracks online just store raw data tagged by space and time coordinates and process near-line or offline with a sufficiently powerful machine, maybe terascale.

Control issues

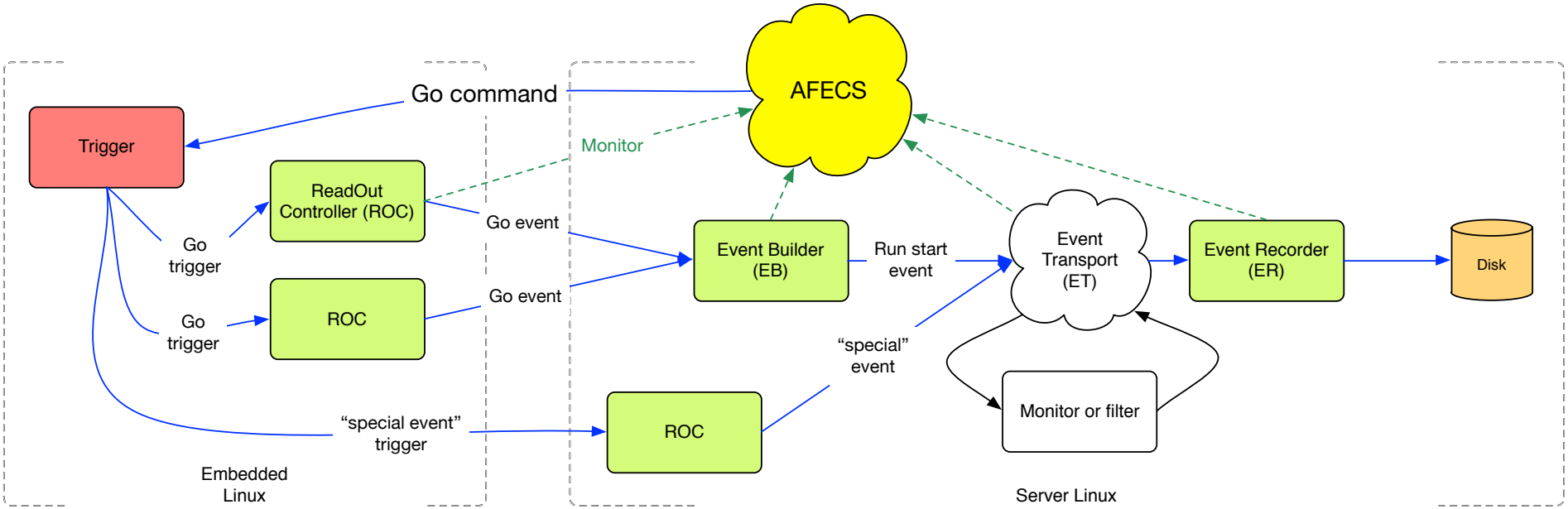
- In each version of CODA up to now every CODA component has followed a simple internal state machine:
  - configure, download, pre-start, go, and end.
- This was developed 25 years ago when individual VME (dare I say Fastbus) CPUs were slow and had limited network bandwidth.
- Issues:
  - Every component has to transition in the correct order.
  - Have to work hard to keep the transitions synchronous with what the data stream is doing.
    - ROC sends an “end event” to mark the end of data. The EB and ER can’t complete their end transition until these events make it to disk.
  - Shutting down cleanly ready for the next run is problematic.
- These issues are magnified in the more exotic configurations such as the one envisaged for SoLID.

Configuration

- The control transitions “configure” and “download” were conceived when the cost of starting a new process, particularly in the VxWorks OS in use at the time was high.
  - The idea was to allow rapid reconfiguration of the DAQ.
  - Download caused running program to dynamically unload a plugin “readout list” and load a new one for a different run type.
  - In reality, in an abundance of caution, most users now completely shut down the DAQ and restart everything when they change run configuration.
- In the new system AFECS would start all the DAQ components with configuration parameters passed once at startup.
- A configuration change would cause a complete shutdown and restart.

Data driven

- A further solution to control related issues could be a move to a data driven model.
  - Make CODA components, as viewed from outside, stateless.
    - AFECS then only cares that they are running and ready to take data.
  - The trigger hardware already generates “special” trigger types that ROCs use to generate “marker” events at the start and end of runs.
  - Components would respond to marker events instead of AFECS commands.



Concluding remarks

- CODA has been very successful
  - A common DAQ for all halls and testbeds is very useful.
  - Modularity leads to scalability but also allows incremental improvement.
  - Allows researchers to move from one hall to another and not have to re-learn the system.
- Hindsight is a useful tool and we have plenty of it to work with.
- Things like VME that have served us well in the past are becoming less useful. Careful selection of replacement technology is needed.
- Future experiments will keep the field interesting as will emerging technologies.
- Looking to the future there will certainly be challenges but they are not insurmountable with the technologies that we expect to be available.