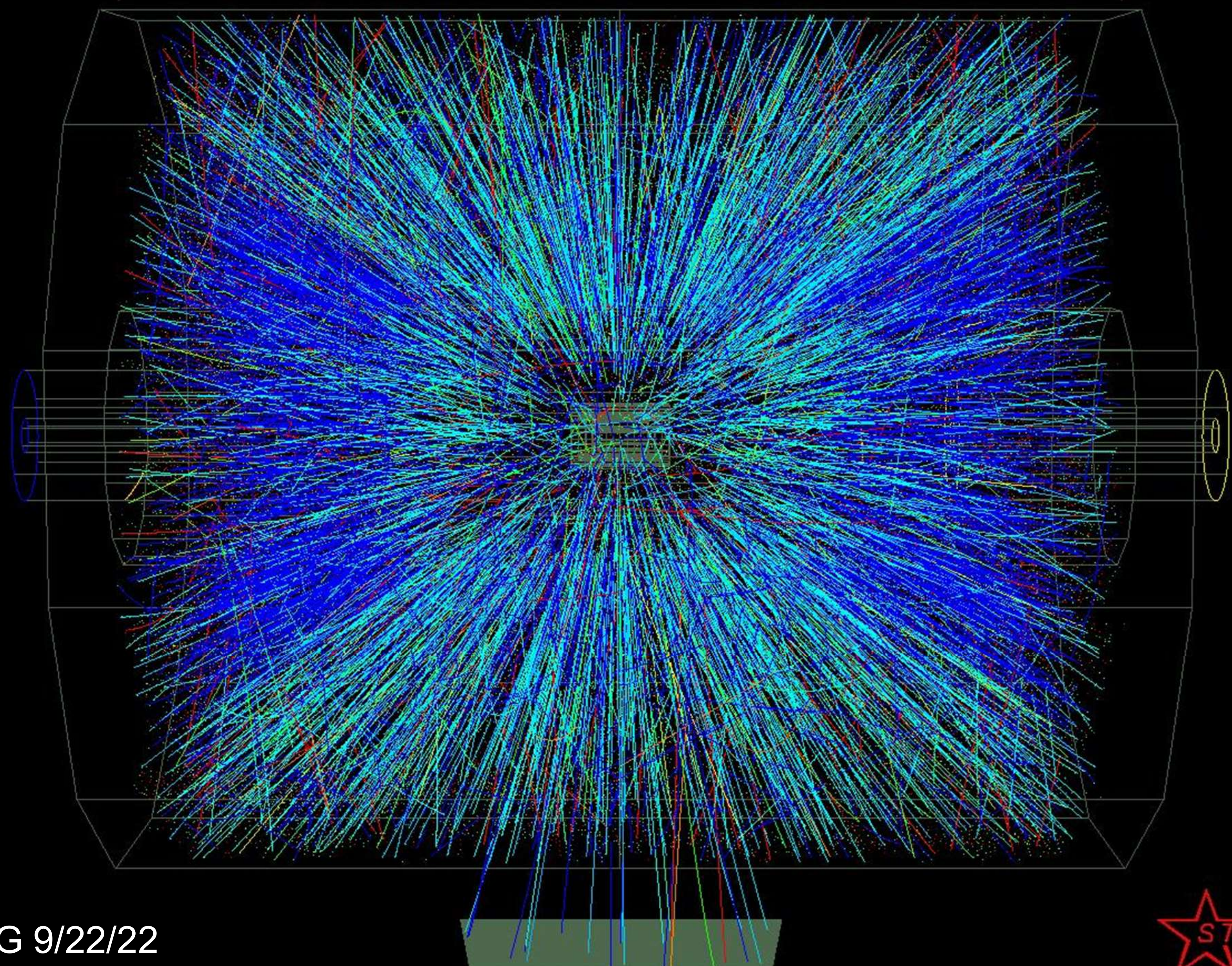
STAR Trigger / DAQ

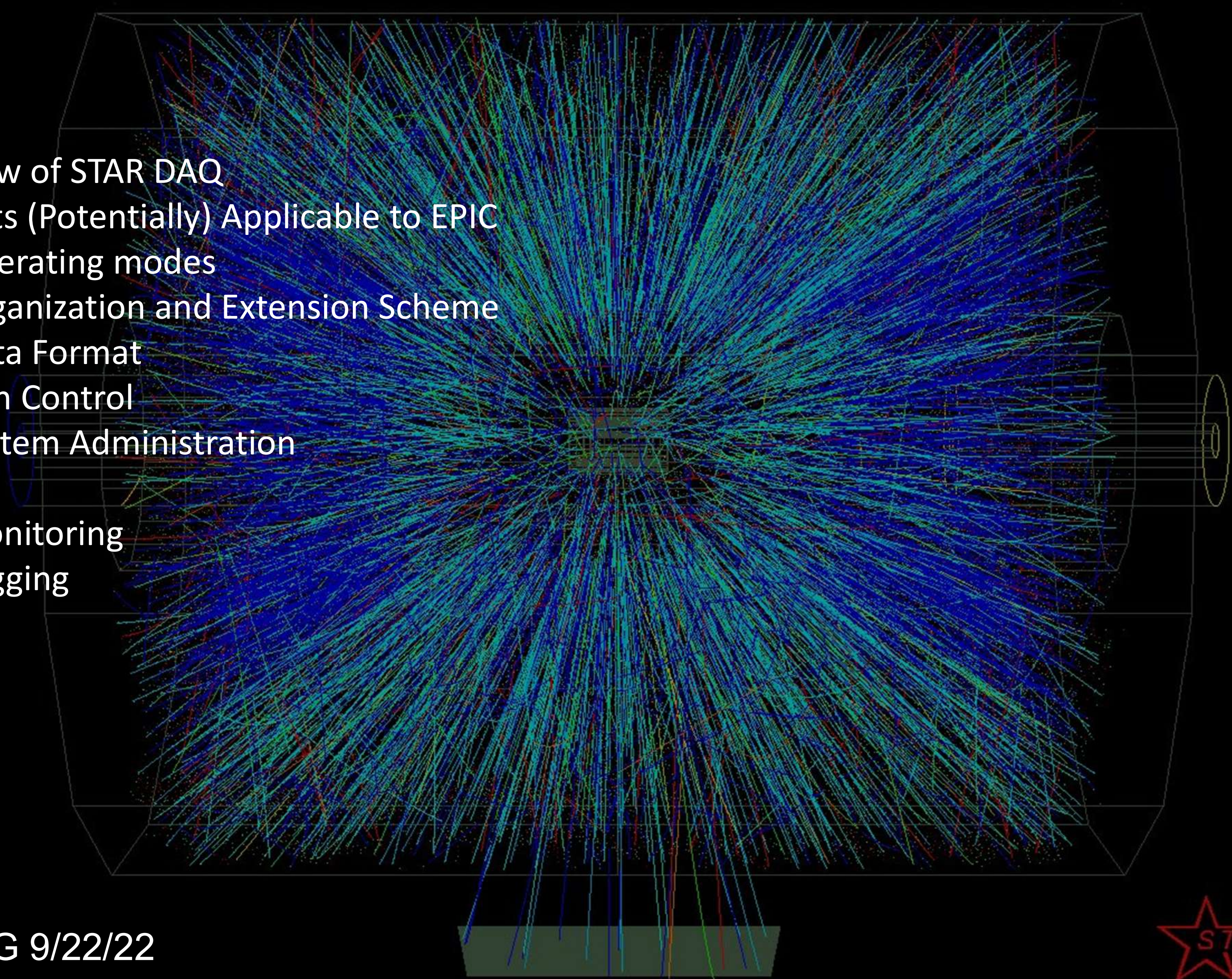


ePIC WG 9/22/22

STAR Trigger / DAQ

Jeff Landgraf

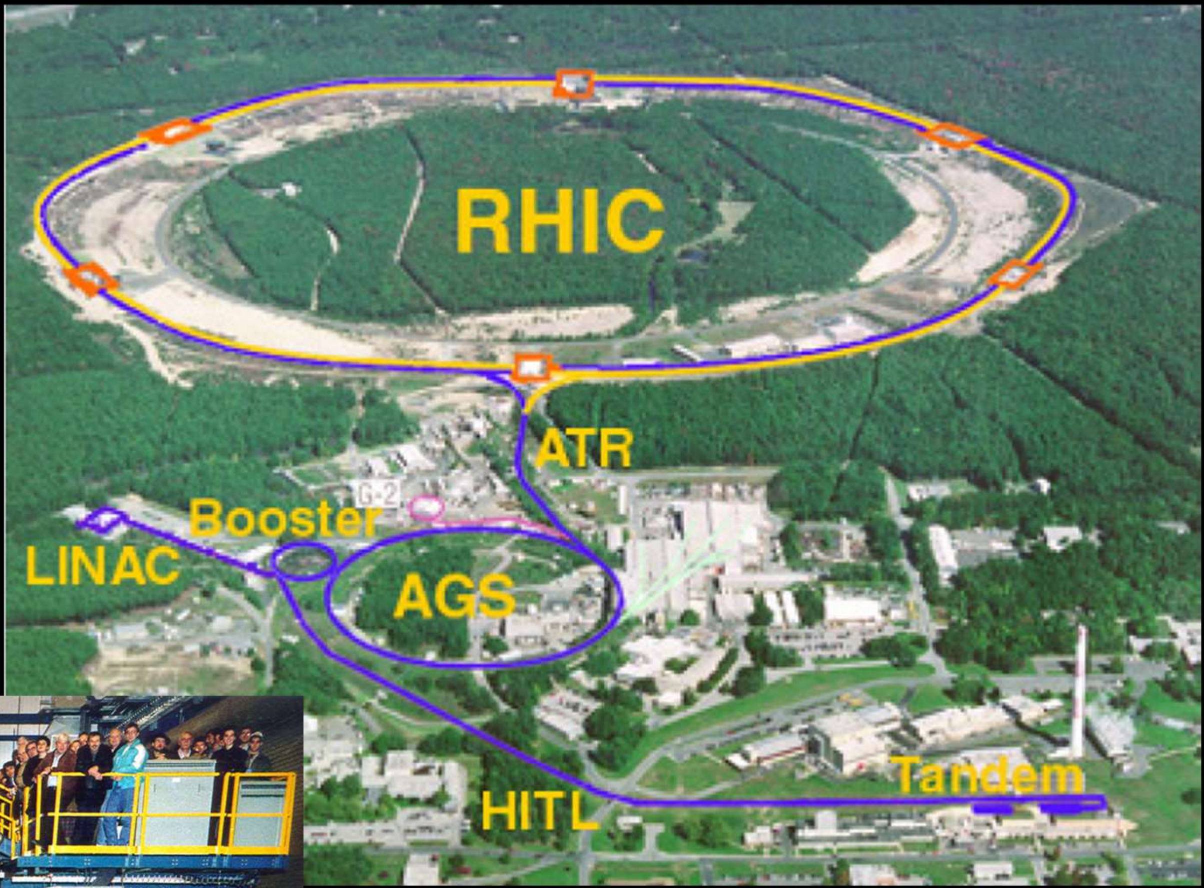
1. Overview of STAR DAQ
2. Elements (Potentially) Applicable to EPIC
    - Operating modes
    - Organization and Extension Scheme
    - Data Format
    - Run Control
    - System Administration
    - QA
    - Monitoring
    - Logging

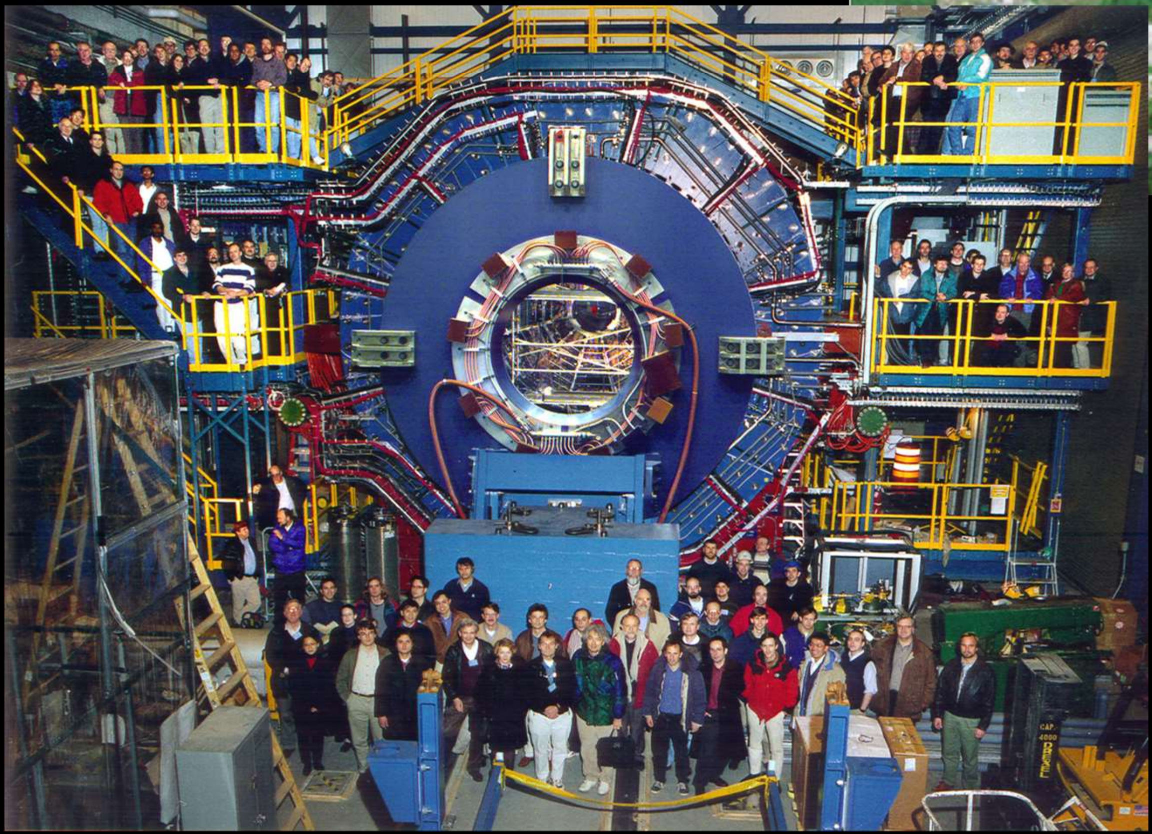


ePIC WG 9/22/22

The STAR Collaboration at RHIC

STAR is:
- 753 collaborators
- 70 institutions / 14 countries

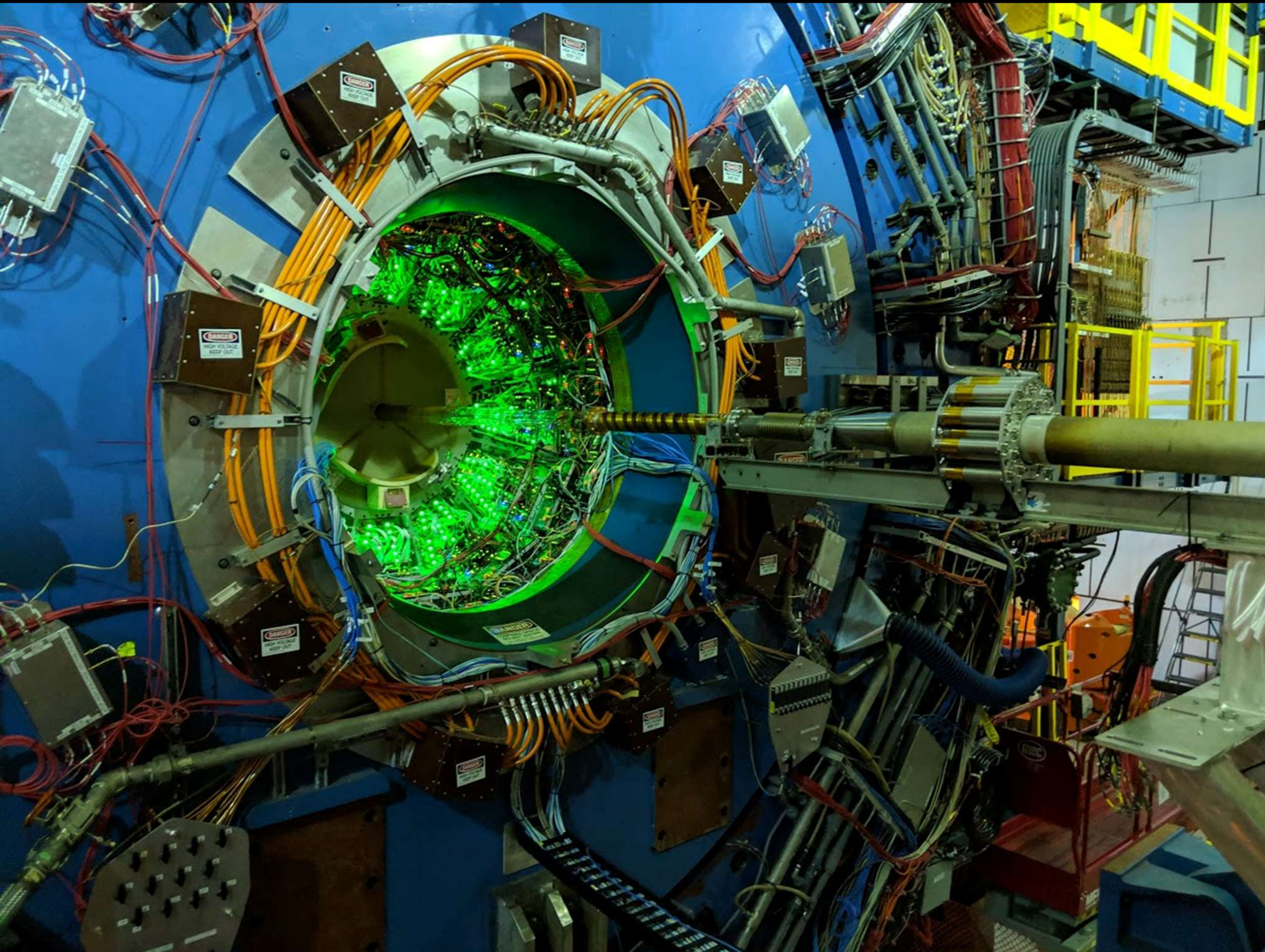




RHIC collides protons, gold and anything in between.

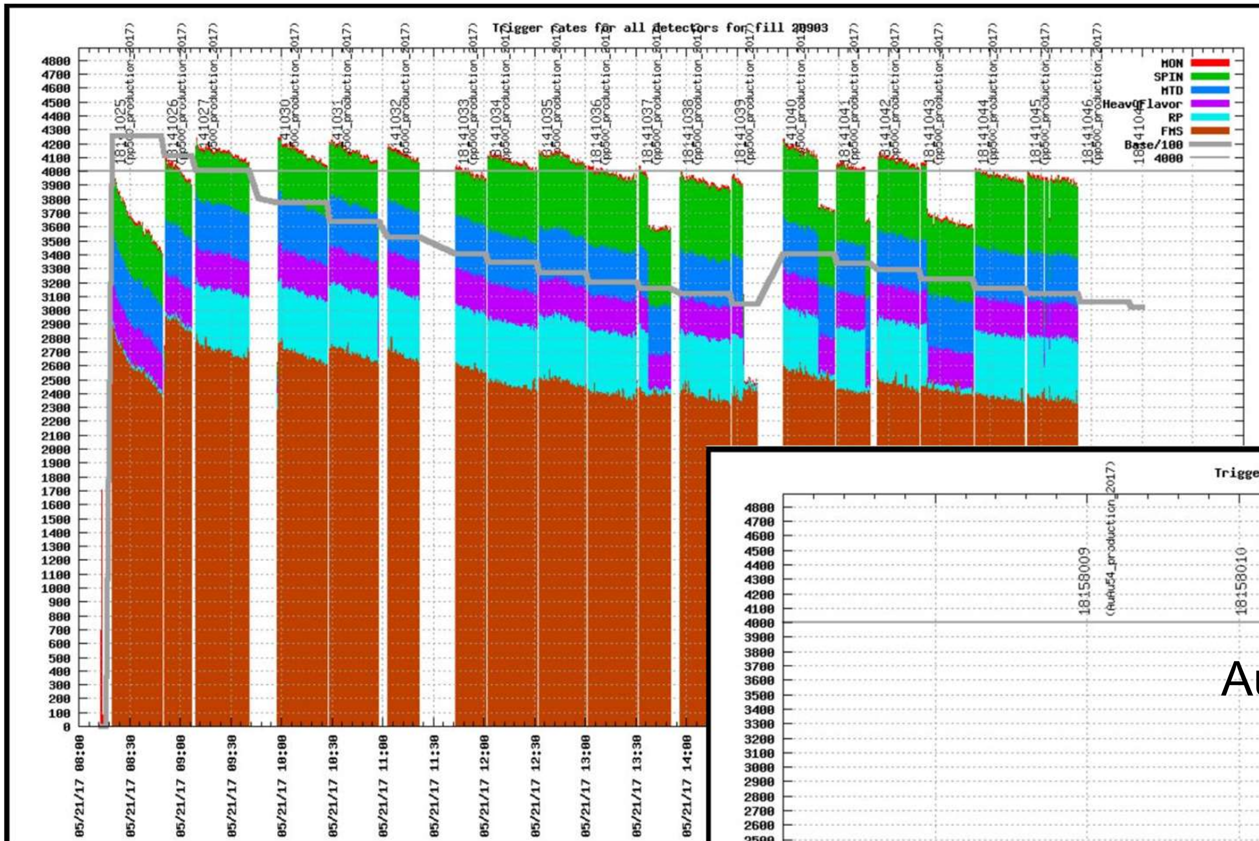
Maximum energy:
- 200 GeV / nucleon
- 500 GeV for protons

Heavy Ion & Polarized Proton Programs

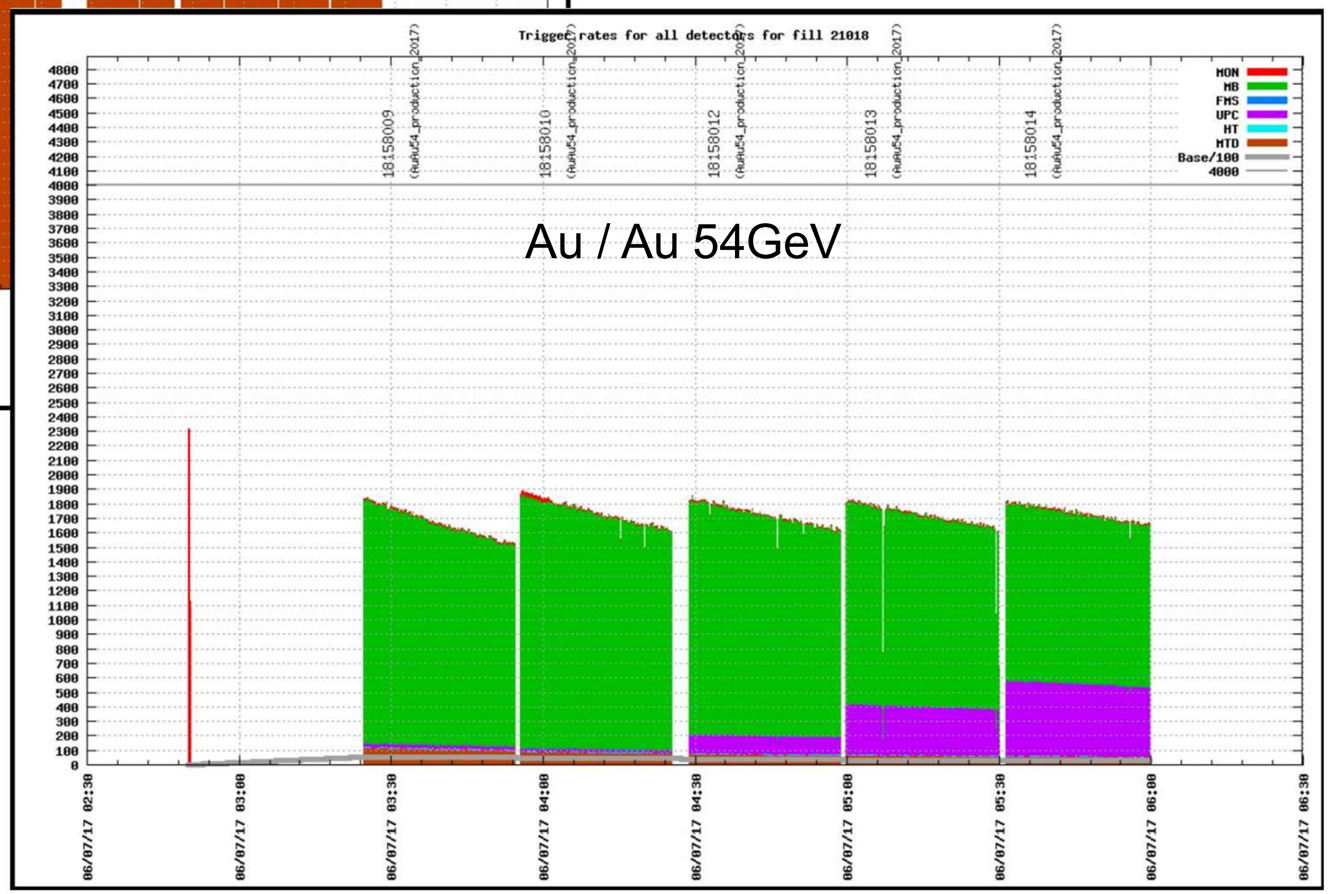


January 23, 2019

Proton / Proton 500GeV (2017)



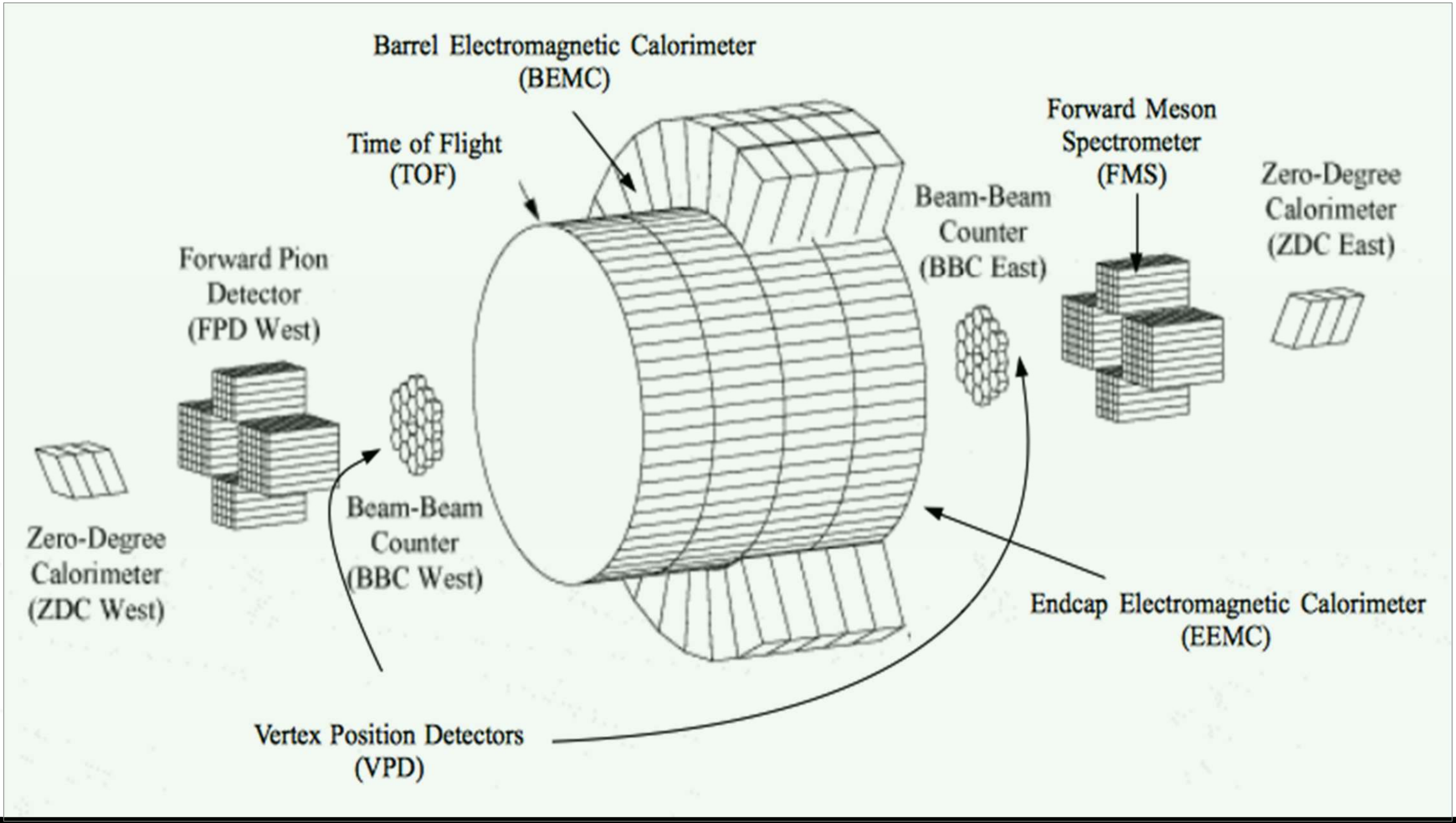
- Long Fills
- High Rates
- Complex Trigger
- Radiation Issues
    * Detector Failures
    * Detector Omission



Au / Au 54GeV

- Short Fills
- Lowish Rates
- Simple Trigger
- Few failures

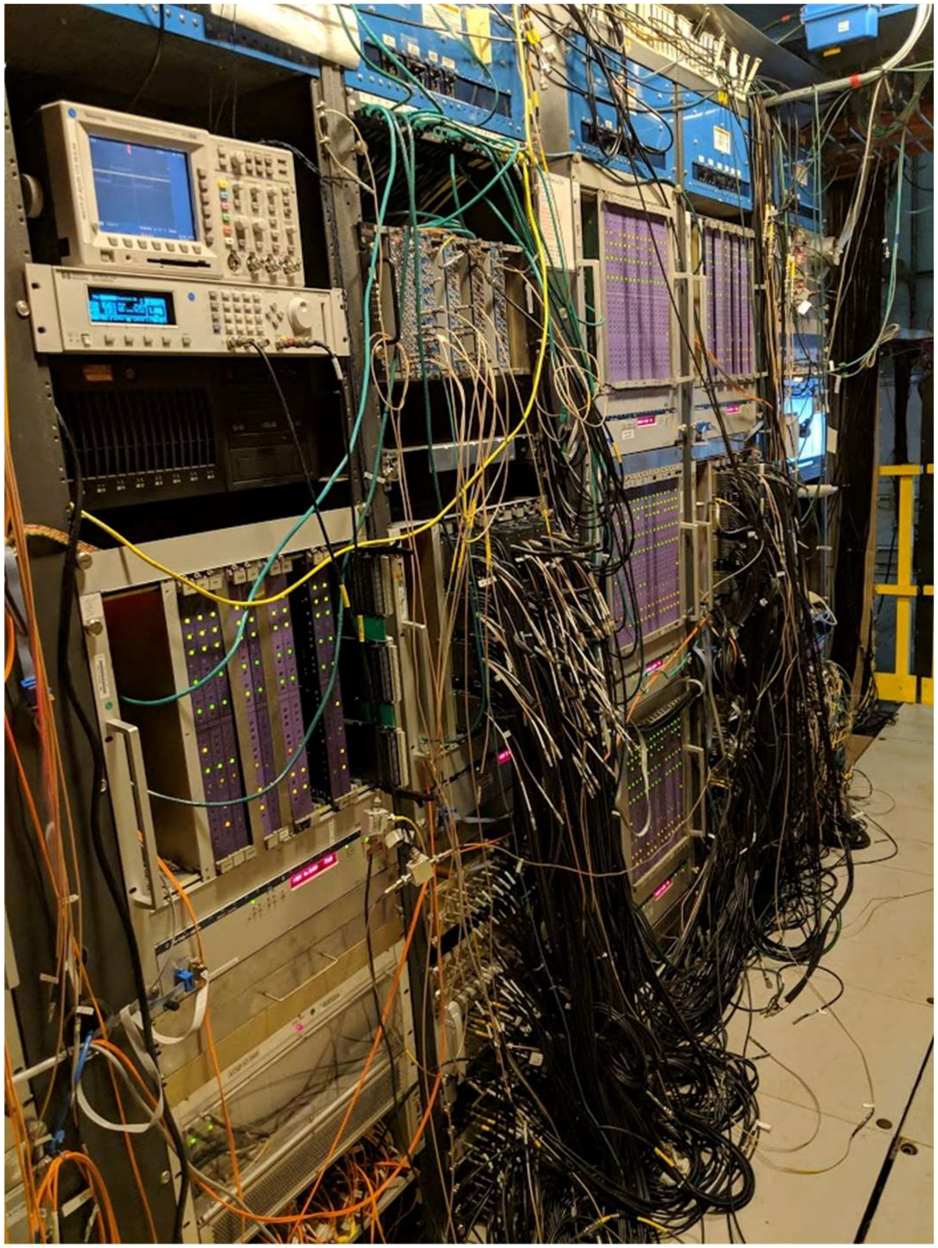
The Trigger Detectors:

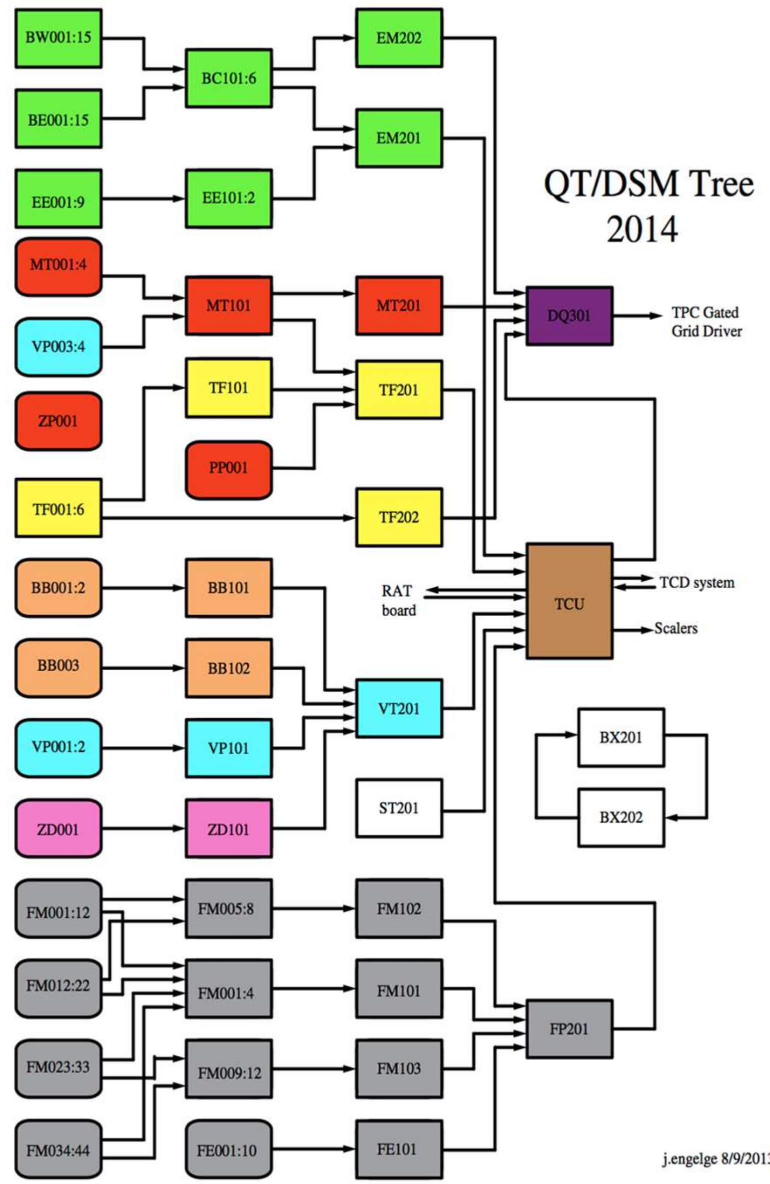


Trigger detectors are capable of reading out every bunch crossing!

- Some read out all of their data each bunch crossing (ZDC, BBC, EPD, VPD…)
- Others read out a useful portion of their data each bunch crossing (EMC's, TOF)

Getting The 128 Bits
To the TCU (The DSM / QT Tree):





Getting The 128 Bits
To the TCU (The DSM / QT Tree):

- DSM are programmable FPGA based boards with 128 inputs bits and 32 bits outputs.

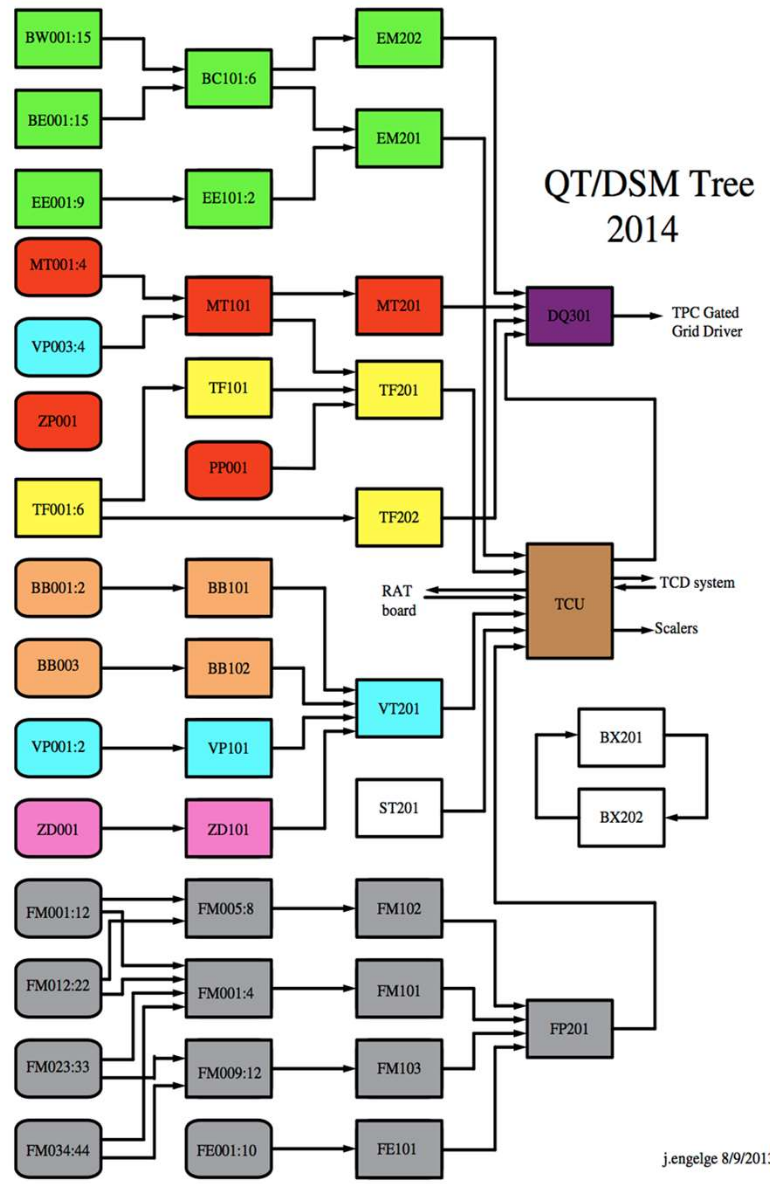
- The QT have a similar role, but can handle some analog input data.

- They receive the data directly from the electronics Of the various trigger detectors.

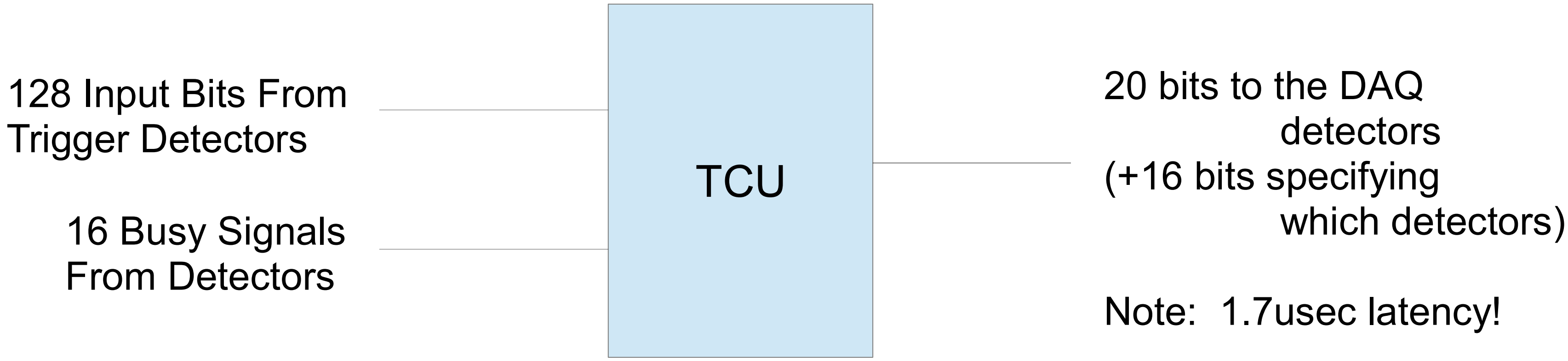
- They are arranged in 3 layers, and the boards within a layer have no information from the other boards in the same layer.

- They can have different code running each time A run is configured, so we have a protocol To group the firmware with along with appropriate Labels in the run control software.

- The DSM crates are connected to a computer called L2 using a custom network called STP. L2 gathers the data from all of the DMS/QT boards And ships it to DAQ.



The Brains of the L0 Trigger (Trigger Control Unit):



UPC-Jpsi-zdc condition 0 toggle enabled

MTD_th1	BBC-TAC		BEMC_HiTwr-th0	revtick-1	FMSsmall-BS3	- Laser-protection	
MTD_th2	- BBC-E		BEMC_HiTwr-th1	revtick-2	FMSsmall-BS2	Laser-fire	
RP_ET	- BBC-W		BEMC_HiTwr-th2	revtick-3	FMSsmall-BS1	Laser-lamp	
TOF-UPC	EPD-TAC		BEMC_HiTw4-th3	revtick-4	FMSlarge-BS3	Laser-diode	
TOFmult0	EPD-E		BEMC_HiTwr-th4	Yellow-filled	FMSlarge-BS2		
TOFmult1	EPD-W		Unused	QLINK4T-1	FMSlarge-BS1		
- TOFmult2	ZDC-TAC		Unused	Blue-filled	FMS-DiBS		
TOFmult3	+ ZDC-E		Unused	QLINK4T-3	FMS-JP2		
MTD1-T-Cosmic	ZDC-W		Unused	QLINK4T-4	FMS-JP1		
TOFsector0_3	ZDC-EW		+ EMC-UPCtopo	QLINK4T-5	FMS-JP0		
TOFsector1_4	Minimum-Bias		Unused	TTLIN-0	FMS-DiJet		
TOFsector2_5	Preceded		Unused	EPD-CR-Trigger	Unused		
RP_IT	VPD-TAC2		Unused	TTLIN-2	Unused		
RP_EOR	VPD-TAC		EEMC_HiTwr-th0	TTLIN-3	Unused		
RP_WOR	VPD-E		EEMC_HiTwr-th1	TTLIN-4	Unused	Zero-bias	
MTD-Cosmic	VPD-W		unused	TTLIN-5	Unused	Random	

UPC-Jpsi-zdc condition 1 toggle enabled

MTD_th1	BBC-TAC		BEMC_HiTwr-th0	revtick-1	FMSsmall-BS3	- Laser-protection	
MTD_th2	- BBC-E		BEMC_HiTwr-th1	revtick-2	FMSsmall-BS2	Laser-fire	
RP_ET	- BBC-W		BEMC_HiTwr-th2	revtick-3	FMSsmall-BS1	Laser-lamp	
TOF-UPC	EPD-TAC		BEMC_HiTw4-th3	revtick-4	FMSlarge-BS3	Laser-diode	
TOFmult0	EPD-E		BEMC_HiTwr-th4	Yellow-filled	FMSlarge-BS2		
TOFmult1	EPD-W		Unused	QLINK4T-1	FMSlarge-BS1		
- TOFmult2	ZDC-TAC		Unused	Blue-filled	FMS-DiBS		
TOFmult3	ZDC-E		Unused	QLINK4T-3	FMS-JP2		
MTD1-T-Cosmic	+ ZDC-W		Unused	QLINK4T-4	FMS-JP1		
TOFsector0_3	ZDC-EW		+ EMC-UPCtopo	QLINK4T-5	FMS-JP0		
TOFsector1_4	Minimum-Bias		Unused	TTLIN-0	FMS-DiJet		

Cancel Ok

The Configuration File:

- The TCU supports up to 64 independent triggers running at the same time
- The concept of a trigger in STAR requires that ALL of the conditions L0 / L1 / L2 and HLT be fully specified.
- Rates are specified by a single prescale for every run, but that prescale can be specified by a program.

production_isobar_2018 (on rts02.starp.bnl.gov)

CFG.TRG_SETUP -

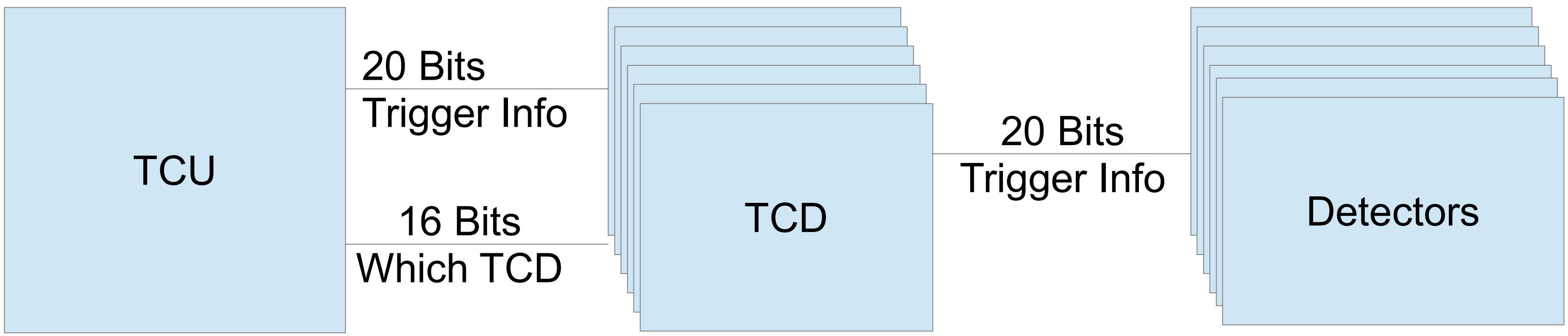
triggers | evpGroup | dataStreamNames | plotGroupNames | labels | contaminationDef | extra

idx	used	id	name	detectors	definition	L1	L2	L4	stream	plot grp	extra	ps
0	✔	600711	UPC-JPsi	custom	+(TOFmult...	accept	accept	accept	upc	UPC	...	Expected...
1	✔	600712	UPC-Jpsi-zdc	custom	+(ZDC-E,E...	accept	accept	accept	upc	UPC	...	Expected...
2												
3												
4												
5												
6												
7												
8		600003	vpdmb	custom	+(ZDC-TA...	accept	l2-emc-ped	accept	physics	MB	...	Expected...
9		10	vpdmb-hlt	none	+(ZDC-TA...	accept	accept	l4_hltgoo...	physics	MON	---	---
10	✔	600031	vpdmb-30	custom	+(VPD-TA...	accept	l2-emc-ped	accept	physics	MB	...	Expected...
11	✔	600042	vpdmb-30-hlt	none	+(VPD-TA...	accept	accept	l4_hltgoo...	physics	MB	---	---
12	✔	13	vpdmb-30-itpc	none	+(VPD-TA...	accept	iTPC_data	accept	itpc	MB	---	---
13												
14												
15	✔	600231	bht1-vpd30	custom	+(VPD-TA...	accept	accept	accept	hf	HF	...	Expected...
16	✔	600232	bht1-vpd100	custom	+(ZDC-TA...	accept	accept	accept	hf	HF	...	Expected...
17	✔	600213	bht2	custom	+(BEMC_H...	accept	accept	accept	hf	HF	...	Expected...
18												
19	✔	600214	bht2-l2gamma	custom	+(BEMC_H...	accept	l2-btow-g...	accept	hf	HF	...	Expected...
20												
21	✔	600601	dimuon	custom	+(MTD_th...	accept	accept	accept	mtd	MTD	...	Expected...
22												
23												
24												
25												
26	✔	27	epd	test	+(EPD-TA...	accept	accept	accept	physics	MON	...	Expected...
27	✔	28	epd-west	test	+(EPD-W) ...	accept	accept	accept	singles	MON	...	Expected...
28	✔	29	epd-east	test	+(EPD-E) ...	accept	accept	accept	singles	MON	...	Expected...
29	✔	30	bbc-west	test	+(BBC-W) ...	accept	accept	accept	singles	MON	...	Expected...
30	✔	31	bbc-east	test	+(BBC-E) ...	accept	accept	accept	singles	MON	...	Expected...
31	✔	32	BBCW-notBBCE-ZDCW-lowTOF	test	+(BBC-W,...	accept	accept	accept	singles	MON	...	Expected...
32	✔	33	BBCE-notBBCW-ZDCE-lowTOF	test	+(BBC-E,Z...	accept	accept	accept	singles	MON	...	Expected...

status: [move] ctrl to copy

Cancel | Save As | Ok

After the TCU decides to trigger an event:



The Trigger Info is:

12 Bits Token. A unique identifier for the event until the all of it's components can be assembled.

4 Bits Trigger command (laser / pulser / configuration evt/ physics)

4 Bits DAQ command (read raw)

*** The detectors have no information as to what trigger fired!

Not So Random Detector Example (TPX – the outer part of the TPC):

- 96 RDO's contain a fiber back to TPX DET computer.

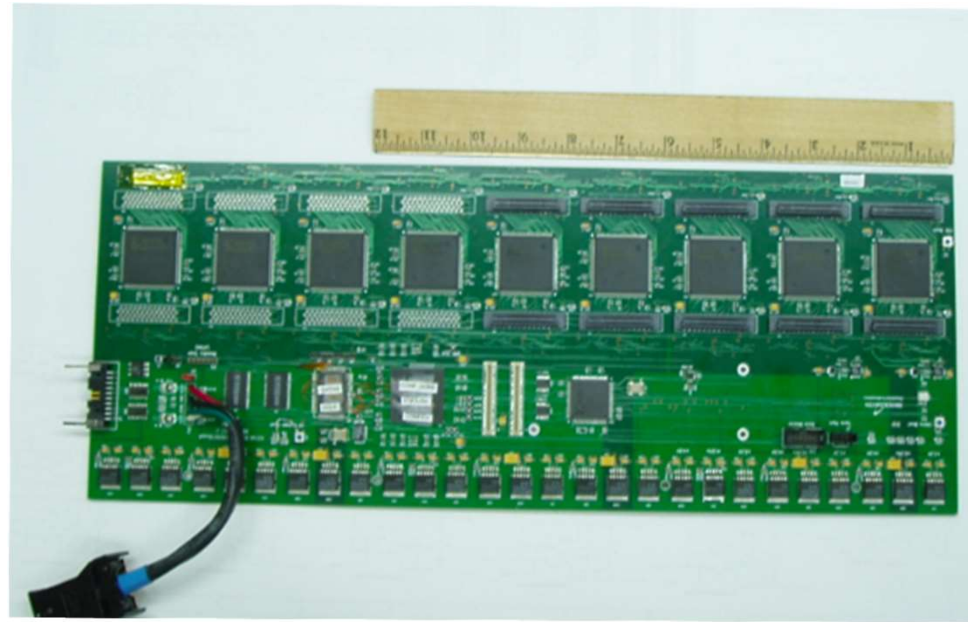
- 18 FEE's / RDO

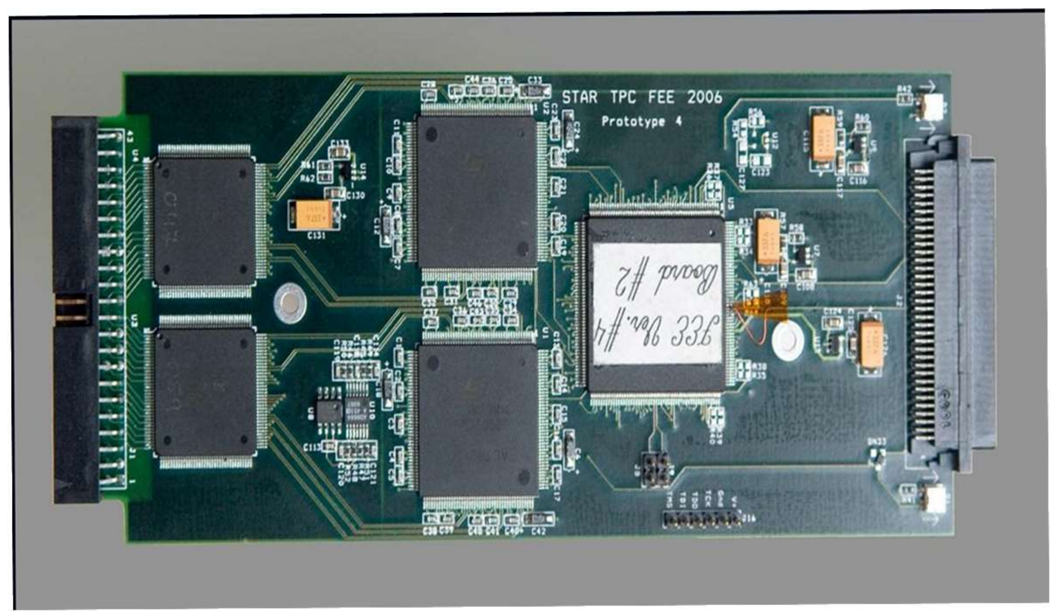
- Black Event ~50MB

- The fee's use ALICE's ALTRO chip
  * digitization
  * pedestal subtraction
  * tail cancellation
  * zero suppression

- Electronics alone reduces event Size to about 2-12MB

- 36 TPX DET computers perform 2 dimensional cluster finding to Reduce data volume down to .25 – 2 MB/event.





The Event Building Network:

* The computers that receive the data from
The detector fibers are called DETS. There
Are roughly 75 DET computers.

* The event builders consists of 14
computers. Each containing 10-24 TB
of buffer disk.

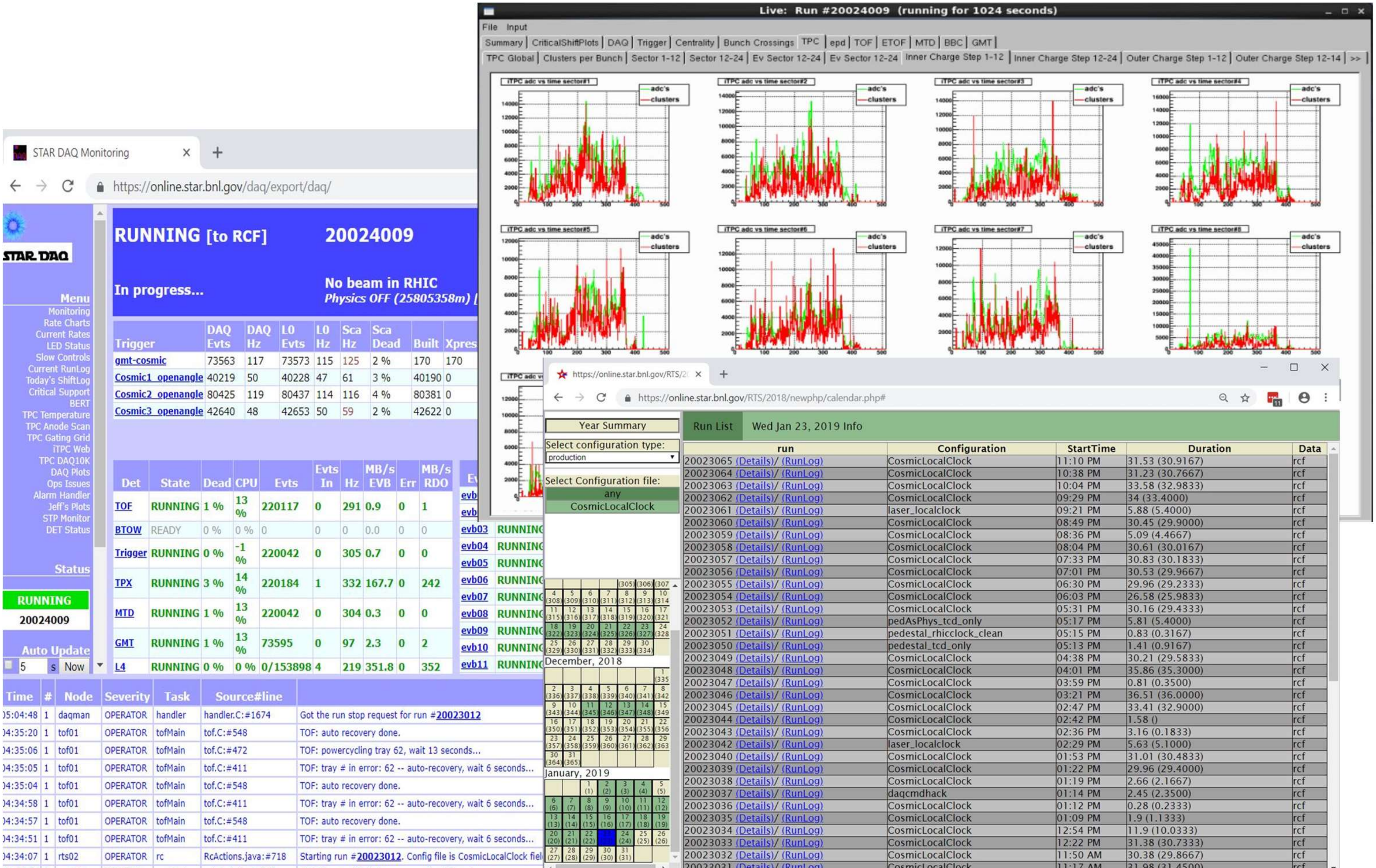
* The event building network is a hybrid of
Gb and 10Gb ethernet. The Event Builders
and HLT computers are on 10Gb ports.
The DETS use Gb ethernet, though the
Switches have 10Gb uploads to the event
Builders

* The HLT trigger is on
A separate 10Gb network along with the
Event Builders. The EVBs send fully
Assembled events to HLT, and receive
An appropriate trigger decision.

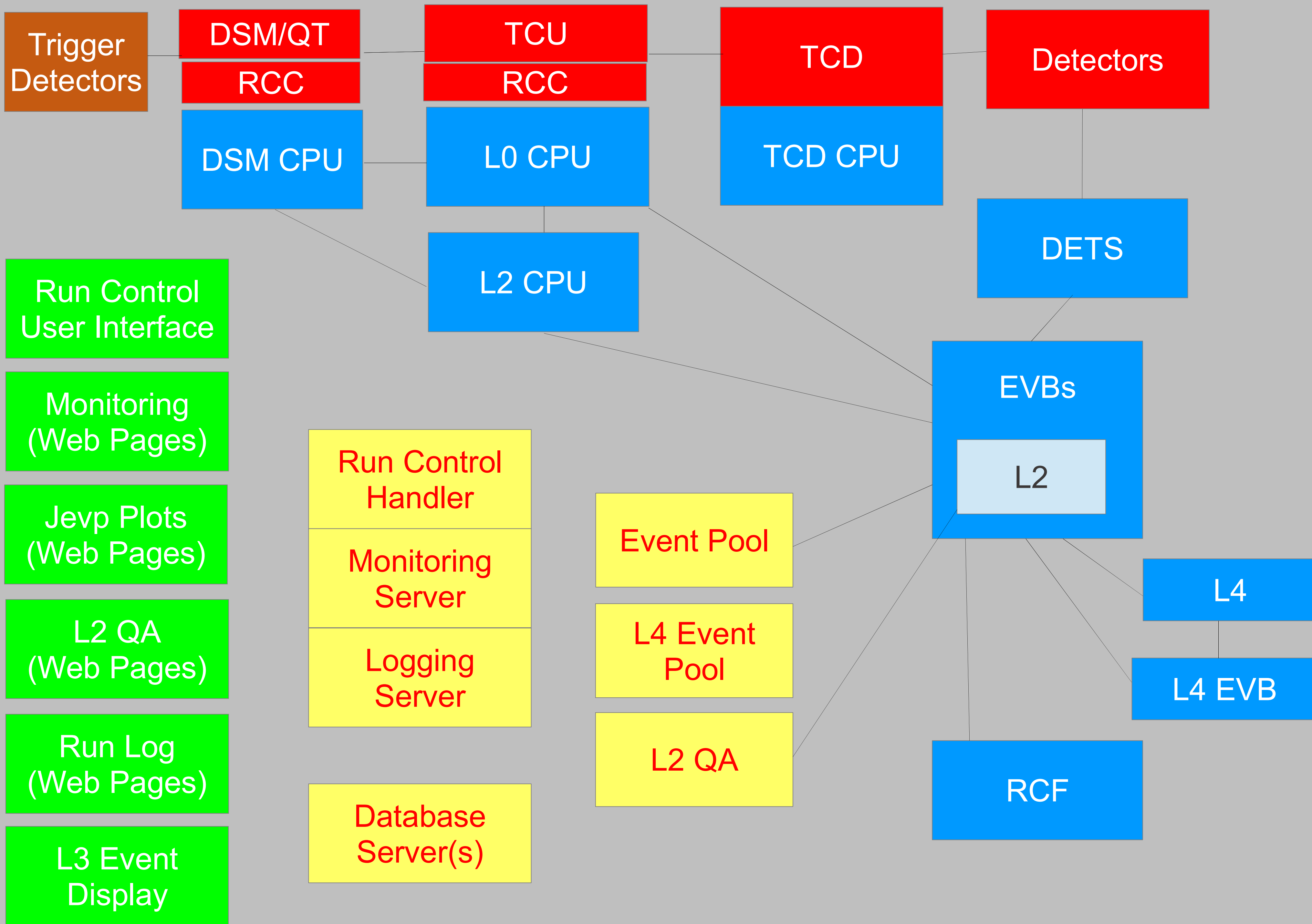
* The aggregate bandwidth of the system
Is about 2000MB / second.

* Completed events are shipped to a tape robot
system called HPSS.

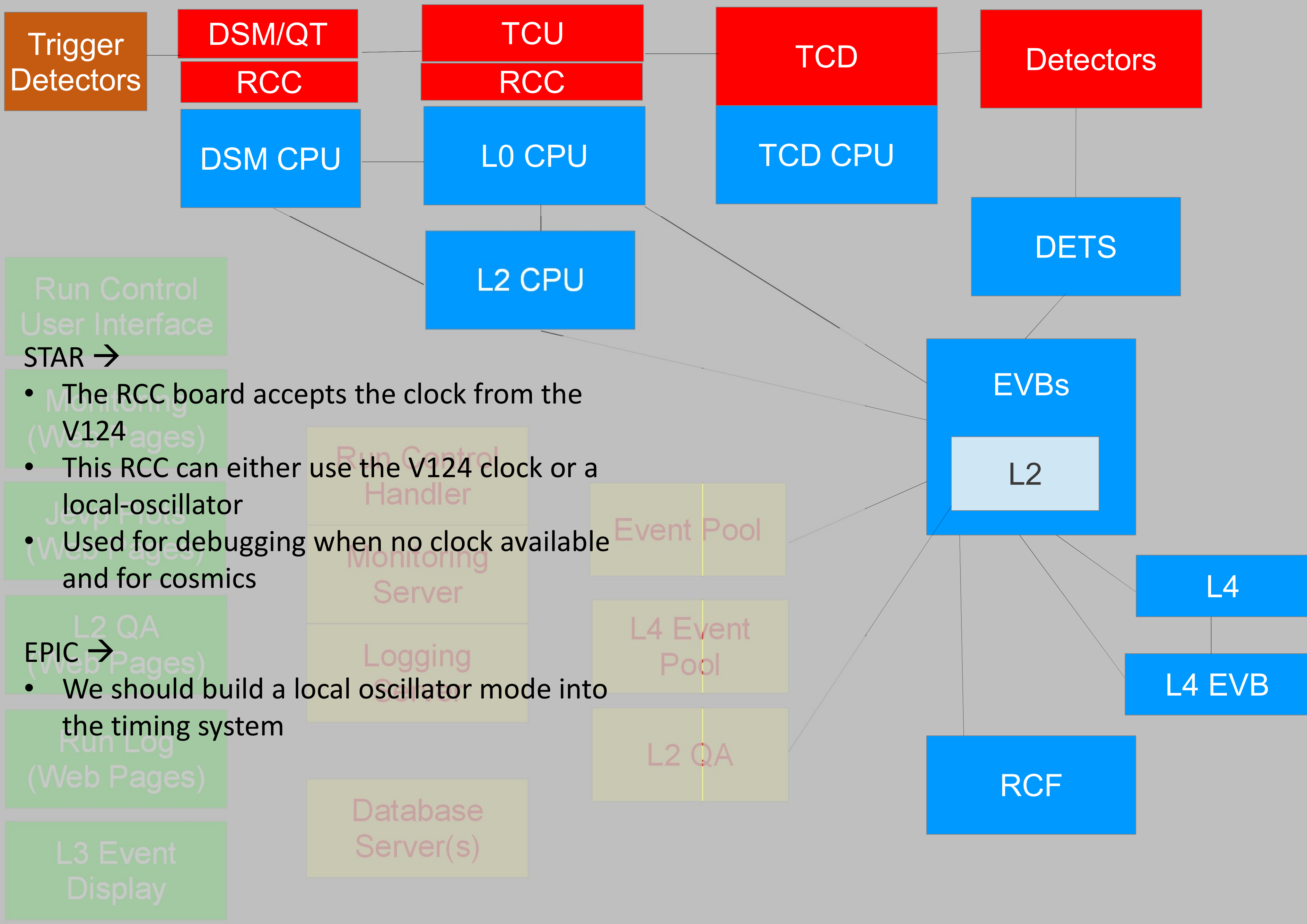
Monitoring / QA / Databases:



Full (Color Coded) Trigger/DAQ Data Flow:



RHIC Clock vs Local Clock Running



STAR →

- The RCC board accepts the clock from the V124
- This RCC can either use the V124 clock or a local-oscillator
- Used for debugging when no clock available and for cosmics

EPIC →

- We should build a local oscillator mode into the timing system

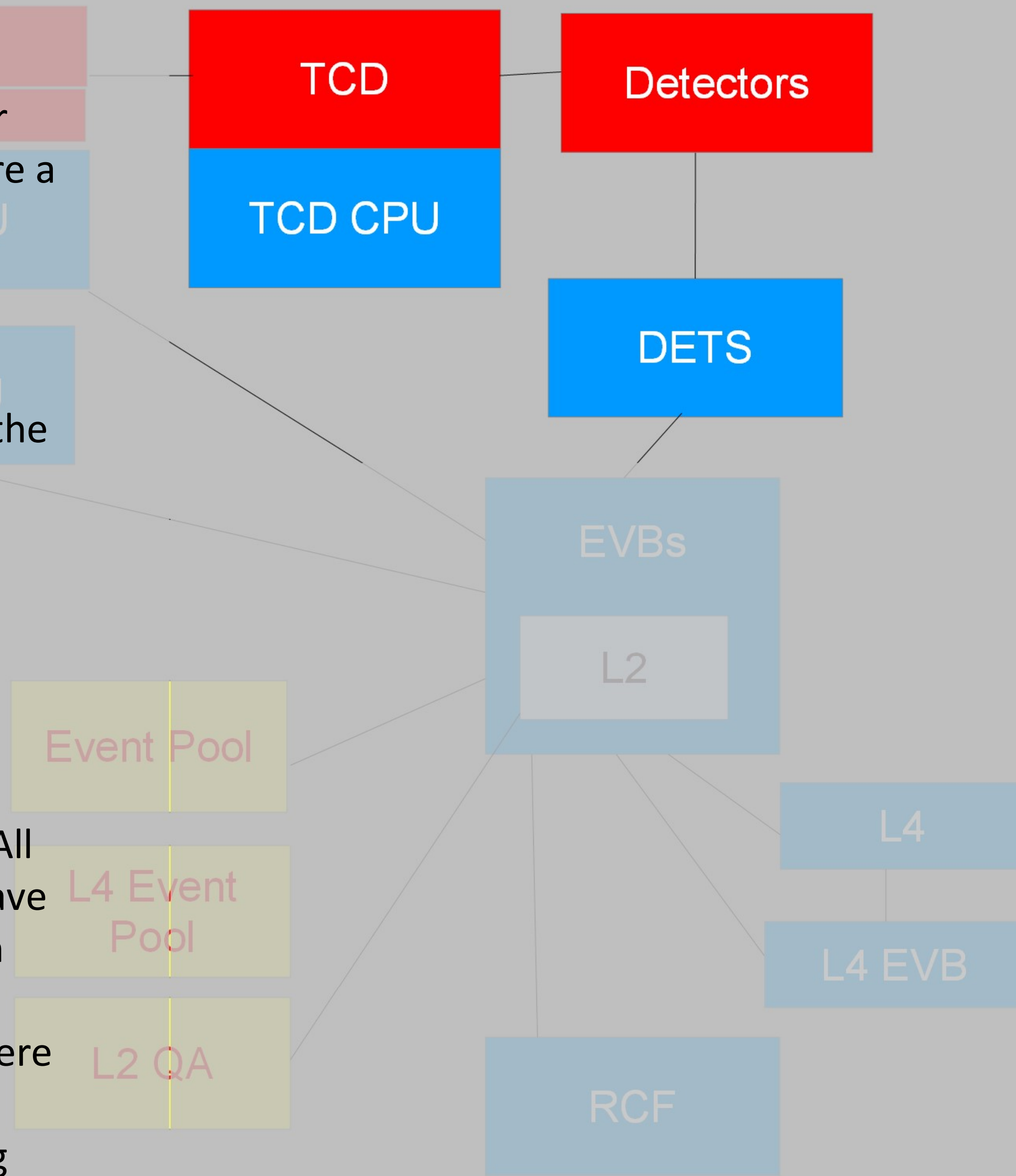
Stand-Alone Running Modes

STAR →

- There is a separate TCD for each detector
- The TCD's have a stand alone mode where a local oscillator is used for the clock and triggers are issued via software or a web interface
- This gives full local debugging of the full DAQ electronics chain, without needing the trigger system, and without impacting ongoing beam operations.
- Data format supports independent files without effort

EPIC →

- Similar emulation can and should be implemented in the readout computer. All timing input to the DAM board should have the capability of being set via software in emulated mode
- From the DAQ electronics perspective there is no distinction from regular running
- The natural independence of a streaming DAQ would make this difficult to break!



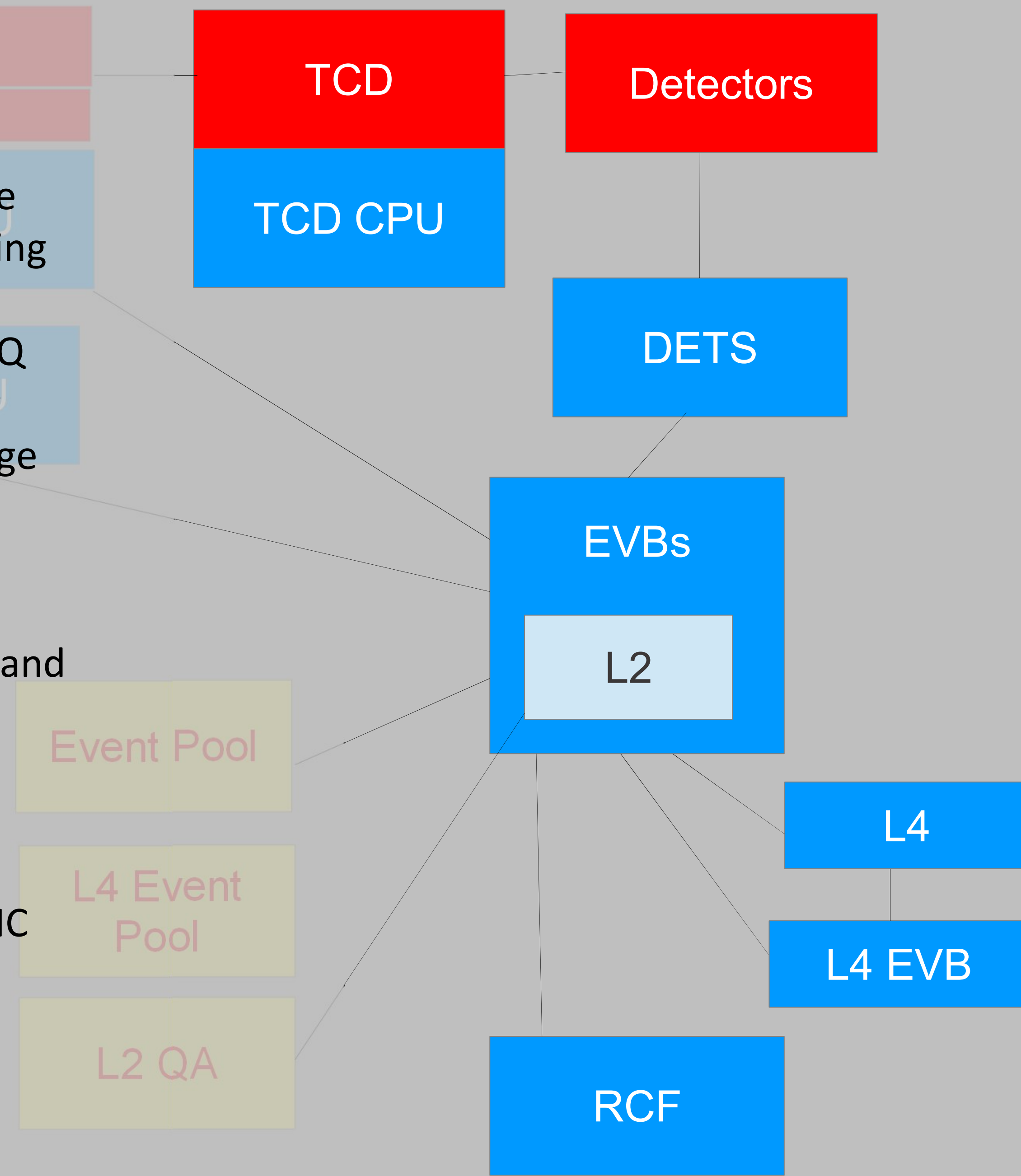
TCD Only Runs…

STAR →

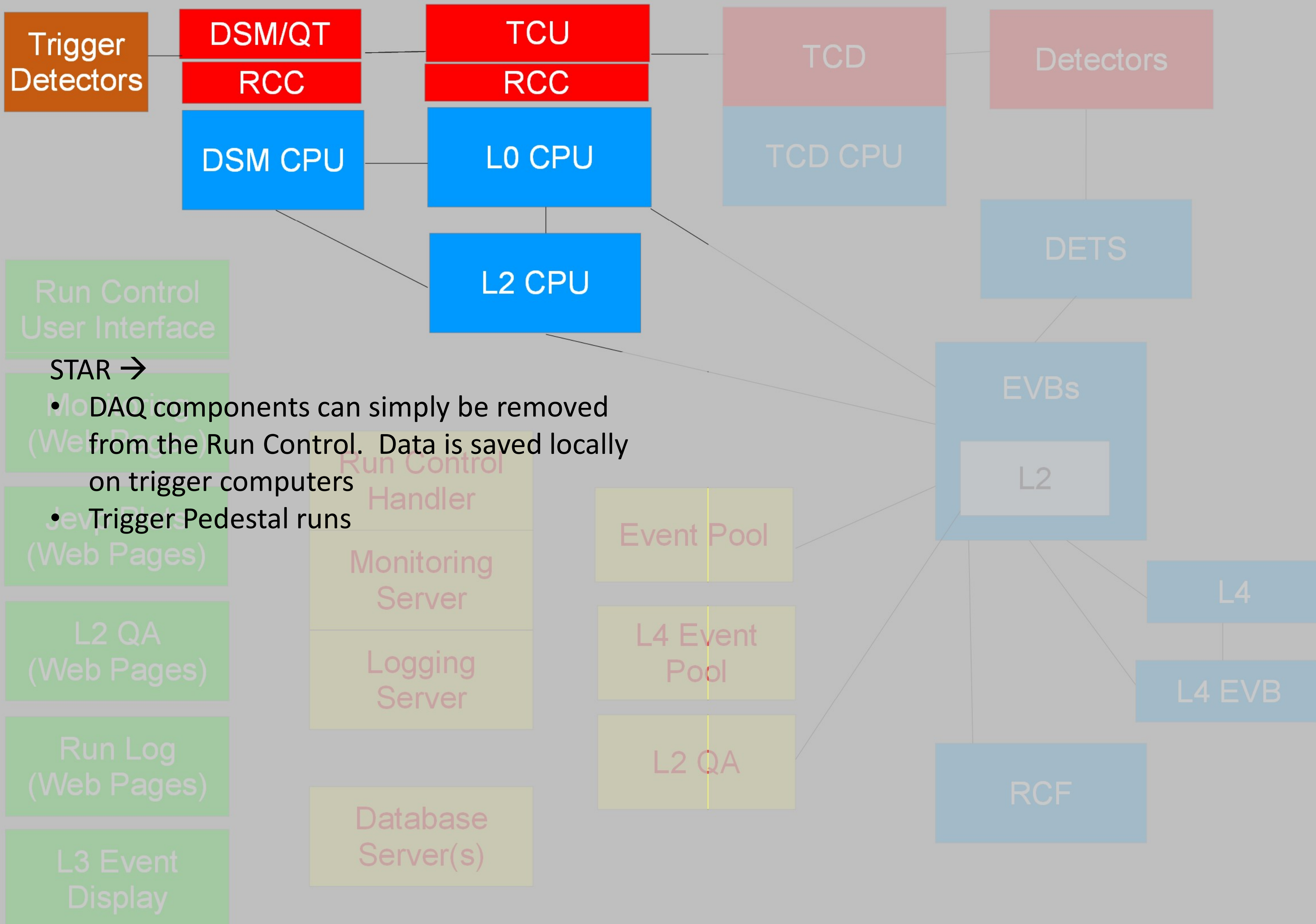
- There is similar capability to bypass the trigger using TCD-ONLY runs. In this case the TCD CPU emulates the trigger including multiple defined triggers, trigger rates, detector mixes and the like. The full DAQ chain is run via run control and including QA, high level triggers, and normal storage to HPSS
- All trigger nodes are bypassed
- Dummy trigger databanks are built
- Frequently use this mode for debugging and for noise check runs and for detector pedestal runs.

EPIC →

- I'm not sure there is a close analog in EPIC



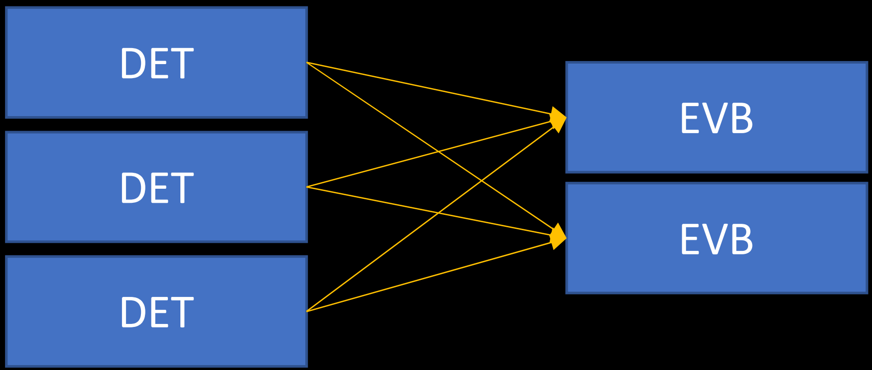
Trigger Only Runs…



STAR →

- DAQ components can simply be removed from the Run Control. Data is saved locally on trigger computers
- Trigger Pedestal runs

Organization and Detector Extension



The common elements of the readout
Are the DETs and the EVBs

The DETs have numerous detector specific
Functions, but The interface to EVB is
uniform:

```
struct iccp2k {
  u_int words;        // words of the bank to follow (not including this header
  u_short srcNode;      // source node
  u_short dstNode;      // destination node
  u_char srcTask;
  u_char dstTask;
  u_short token;
  u_char cmd;
  u_char pad1;
  u_short payload_words;   // Number of words of payload.
};
```

- TCP/IP push using iccp2k header
- Data shipped in SFS data file format
- SFS path provided according to source node
- Routing from DET to EVB handled according to static mapping of token <-> EVB
- Flow control handled by speed factor for each EVB used to construct token map

In practice, EVBs contain significant detector specific processing but this is modularized in trigger algorithms, format checkers. In order to incorporate a new detector the mapping from the srcNode to the path is the only code that needs to be added.

Data File Format: SFS File System

Properties:

- Data banks are specified by an arbitrary linux-like file path, data in the form of a file
- There is a linux filesystem like interface
- Official DAQ Reader for each detector navigates to appropriate bank by path
- Data writing files can be "streamed"
  - No need to change earlier data (sizes for example) beyond current record while writing
  - Files can be appended and maintain valid files
  - Files can be separated and maintain valid files (on absolute paths)

```
[evp] /a/23103023/> fs 12452 ls -r
 [ 173204 bytes] /#2402088/
 [     24 bytes] /#2402088/EvbSummary
 [    112 bytes] /#2402088/EventSummary
 [    256 bytes] /#2402088/TRGID
 [   9972 bytes] /#2402088/btow/
 [   9972 bytes] /#2402088/btow/sec01/
 [   9972 bytes] /#2402088/btow/sec01/rb01/
 [   9972 bytes] /#2402088/btow/sec01/rb01/raw
 [   2100 bytes] /#2402088/etow/
 [   2100 bytes] /#2402088/etow/sec01/
 [   2100 bytes] /#2402088/etow/sec01/rb01/
 [   2100 bytes] /#2402088/etow/sec01/rb01/raw
 [  86604 bytes] /#2402088/fcs/
 [   2852 bytes] /#2402088/fcs/sec01/
 (etc...)
```

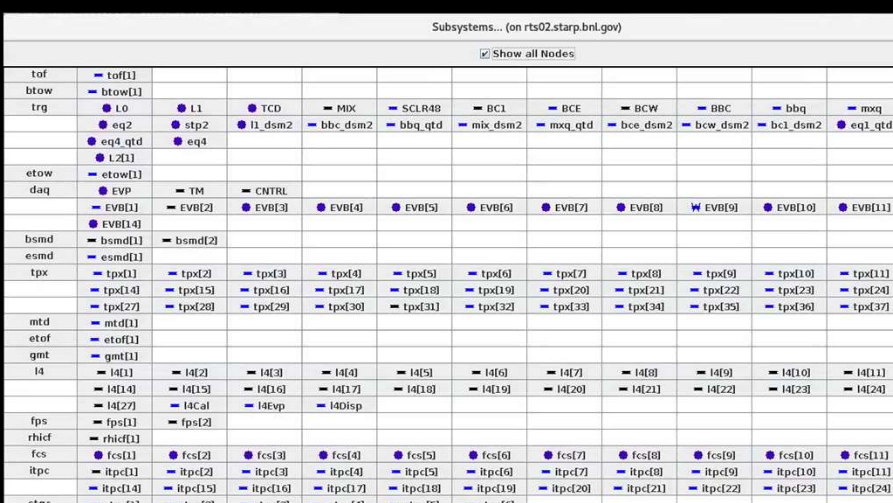
type = "FILE"		
byte_order = 0x04030201		
sz		
head_sz	attr	reserved
name....		
name (continued)....		

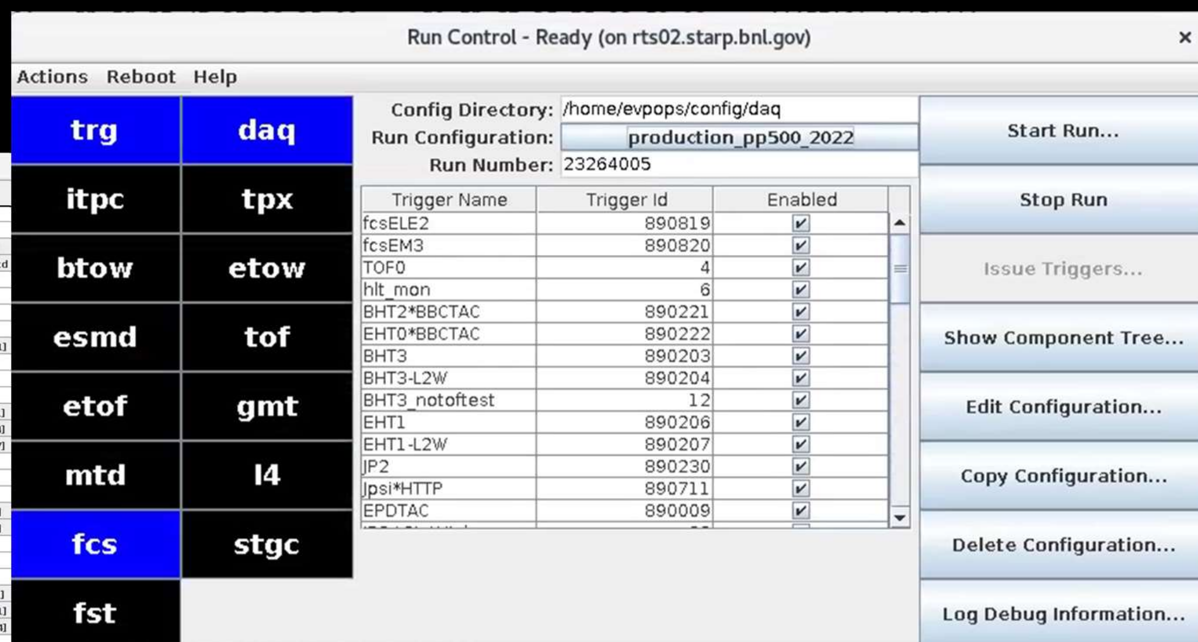
/xxx/xxx/yyy	/xxx/xxx/ is the "directory part" yyy is the "file" part
/xxx	absolute path
xxx or xxx/xxx	path is relative to the directory part of the previous entry
xxx/	directory
0 length	the name of this file is used to reset the "previous entry", but contains no data

- One doesn't know for certain when a directory is finished being added, so in practice I add the rule that a change of the base directory closes that directory.

Run Control / Configuration

- Run Control Handler is a server maintaining states
- Run Control is a java client users use to start/stop runs and access states and configuration
- Configuration Manager is the same application without the control features
- Clients follow simple ethernet protocol (or inherit from class demanding a handful of member functions: “Start Run”, “Stop Run”, “Send Config”, “Force Stop”, “Reboot”)
- XML configuration file
- Mysql relational DB for “old” parameters
- Mongo object DB for “all” parameters
- Mongo DB records, and XML files, and the bulk of the Configuration GUI are generated via Java’s “reflection” classes directly from the java configuration file definition.





System Administration

- We use PXE boot from static NFS disk images for all DETs and EVBs
- Originally based upon SL4's "livecd" features, but these features lost support
  - I now maintain a set of (not quite) turnkey scripts of my own to build images from virtual machine linux installations
  - System disk uses AUFS which does limit my kernel choices
  - Have plans to update the union filesystem when I next upgrade Linux versions
  - Machine configuration is done in several steps
    - IP addresses and names come from the statically defined DHCP entries
    - Grub configuration based on these names define boot parameters which include an "etc directory" path for each type of computer which is copied over the static etc directory built into the images before linux boot
    - Ethernet configuration (of the non-boot interfaces) is done from a master file so I can access it from one place.
    - A user script running at the end of the systemd boot process
  - Has worked well thus far… I don't know if I save time in the end, but the time I spend is during shutdowns, not during runs, and all machines are setup the same.

Online QA

Maintain 2 separate but nearly identical QA systems, one based on the HLT results and one developed on direct detector data

Maintain a pool of data accumulating at ~10-20hz(500hz). This data is stored for approximately 2-3 weeks and rolled over as data accumulates. Data mix is defined in run control by trigger.

JevpBuilders are defined by detector groups. These create the histograms using root. I force a wrapper "JevpPlot" class to be used because of root's association between histograms, canvas's and global parameters for various display features. The JevpBuilders can be run stand-alone to produce pdf files&root files for development, debugging, and sometimes analysis.

A production server calls each of the JevpBuilders and periodically writes all plots to a large ceph filesystem accessible to the web.

Javascript/PHP display for all histograms.