SNS FTS/STS Data Acquisition & Automation

Matt Pearson
STS Control System Engineer for Instrument Systems

Computer Science & Math Workshop

June 23-24, 2022

Outline

- Who am I and what do I do?
- Current FTS data acquisition systems (hardware and software)
- Intro to EPICS (Experimental Physics and Industrial Control System)
- Current FTS control system & automation software
- Current developments at FTS (automation)
- Considerations for STS control system & scientific software
- Plans for STS data acquisition and automation
- Plans for STS computing & networking
- Plans for STS control system interfaces to scientific software

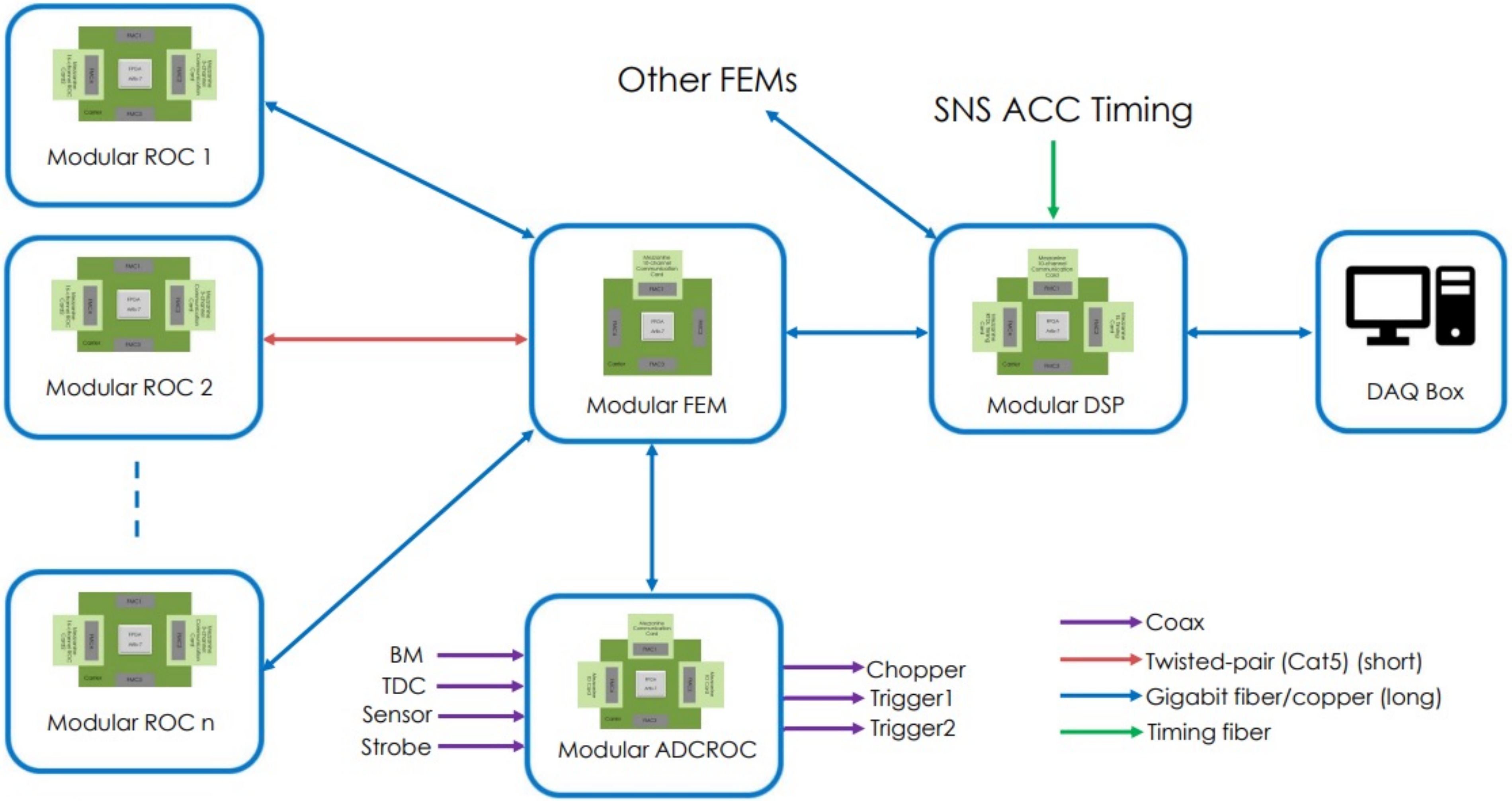
Who am I and what do I do?

- Since 2002 I have been involved in developing control system software, mostly at EPICS based sites.
- Joined SNS in 2013 in the Instrument Control & Data Acquisition group.
- Joined STS in 2020, responsible for the Integrated Control System for Instrument Systems:
  - Data acquisition electronics & software
  - Control system hardware & software
  - Control system automation and experimental user interfaces
  - Compute systems, networking & data storage (for technical systems & data analysis)

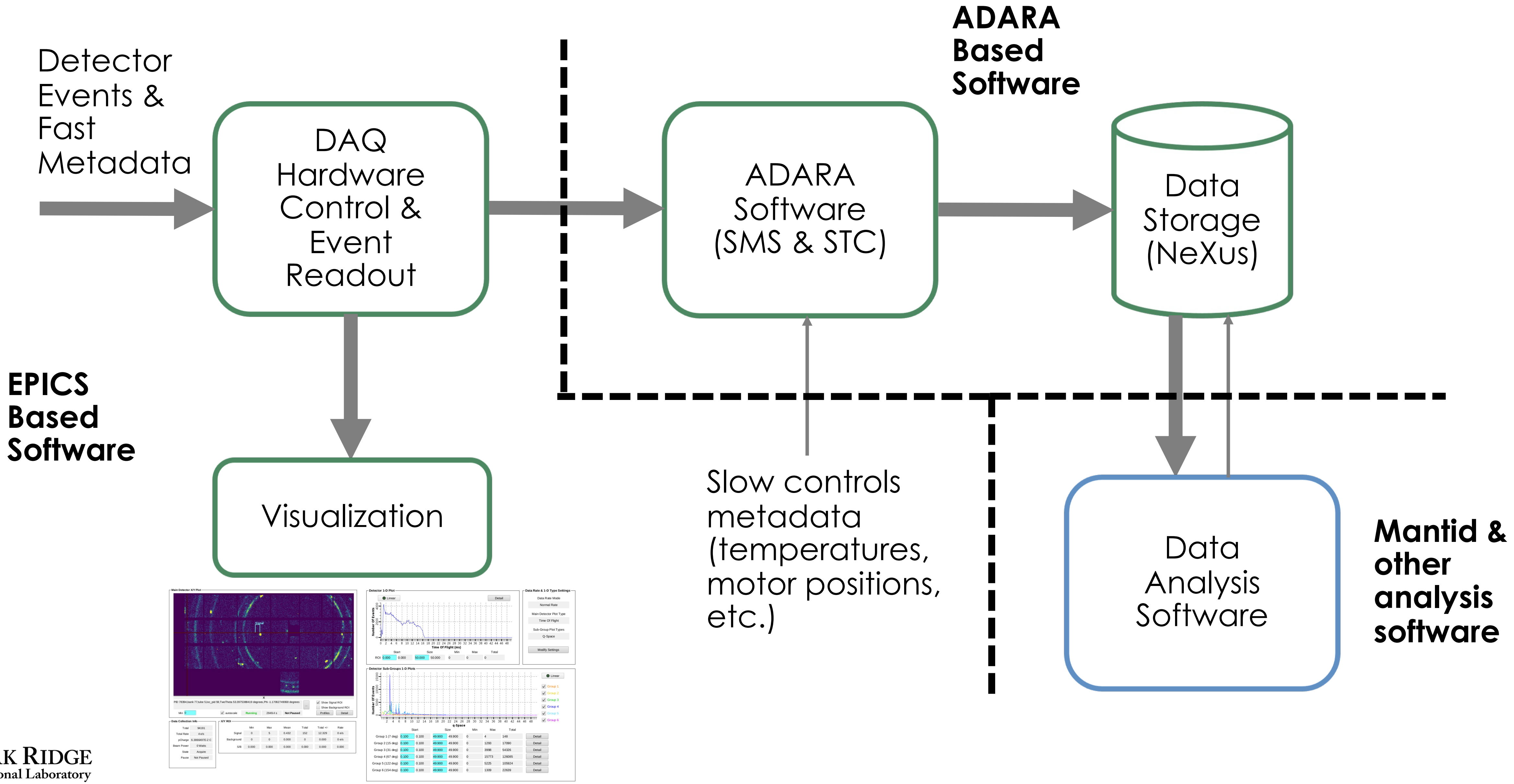
Current FTS/HFIR Data Acquisition

Custom in-house electronics to digitize detected neutron events and/or metadata and timestamp them using the SNS timing system signal.

Based on FPGA and modular mezzanine card design (2022)



Data Acquisition Software ('DAQ Box' expanded)



EPICS (Experimental Physics & Industrial Control System)

- Framework for developing highly distributed control & data acquisition systems.
- Built in support for many kinds of hardware devices (and easy to develop new support).
- Client/Server model. The logic resides in the servers, and the clients act as an interface to the servers.
- The clients and servers communicate using protocol called Channel Access (CA) or PVAccess (PVA).
- Servers are written in C/C++ or Python and have a small memory footprint. The servers run on Linux, Windows, MacOS, VxWorks, RTEMS & more. Various hardware platforms.
- Clients tend to be C/C++, Java or Python based (graphical or scripts, or anything that can talk CA or PVA).
- Extensive worldwide collaboration for 35 years.
- The SNS accelerator has an EPICS control system since before 2007, and during 2012-2019 we converted all the SNS Instruments to EPICS, and about half of HFIR beamlines, with a lot of success.

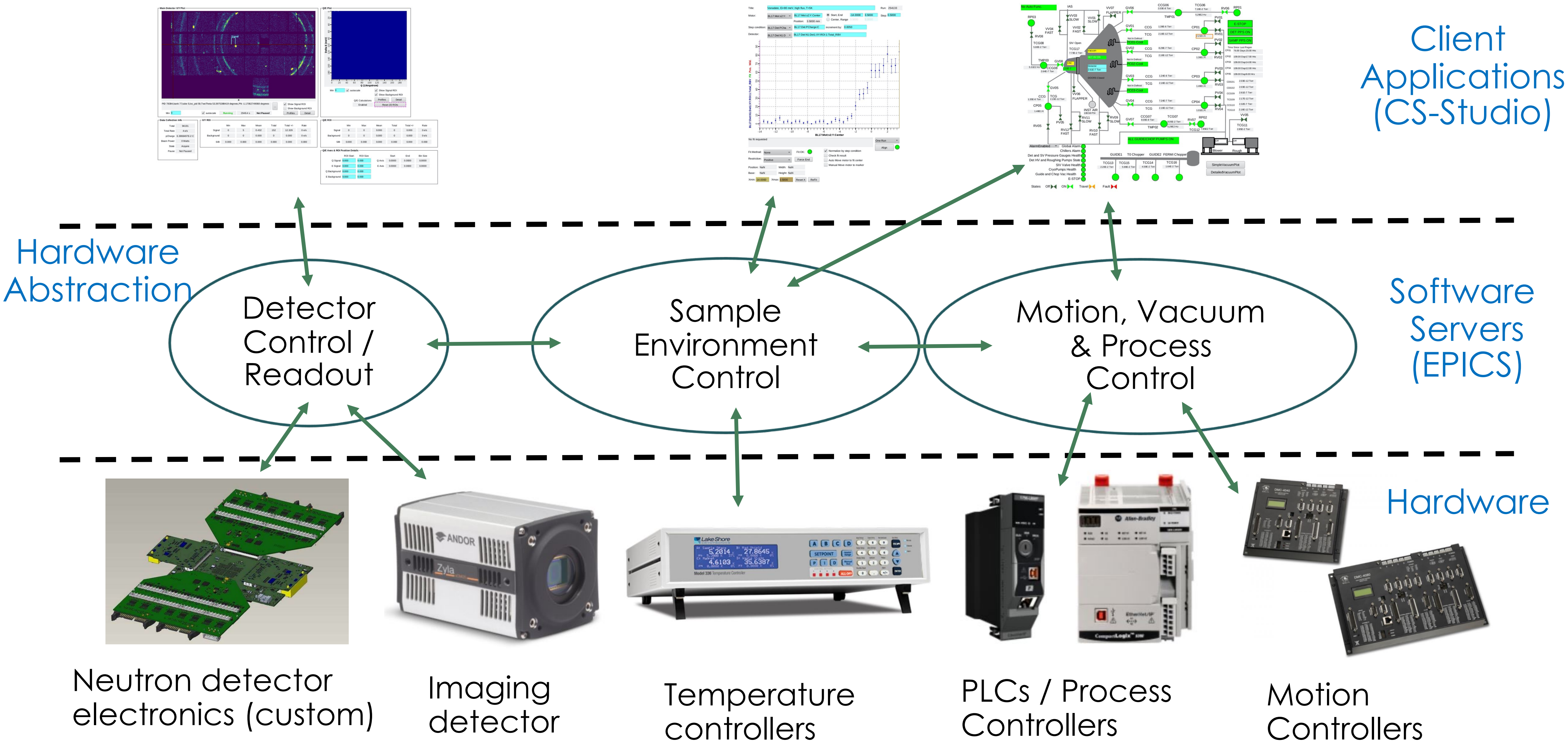
ADARA (Accelerating Data Acquisition, Reduction, and Analysis)

This is currently in use on all SNS and some HFIR Instruments, and we will reuse it for STS.

Applications & network protocol for streaming neutron events:

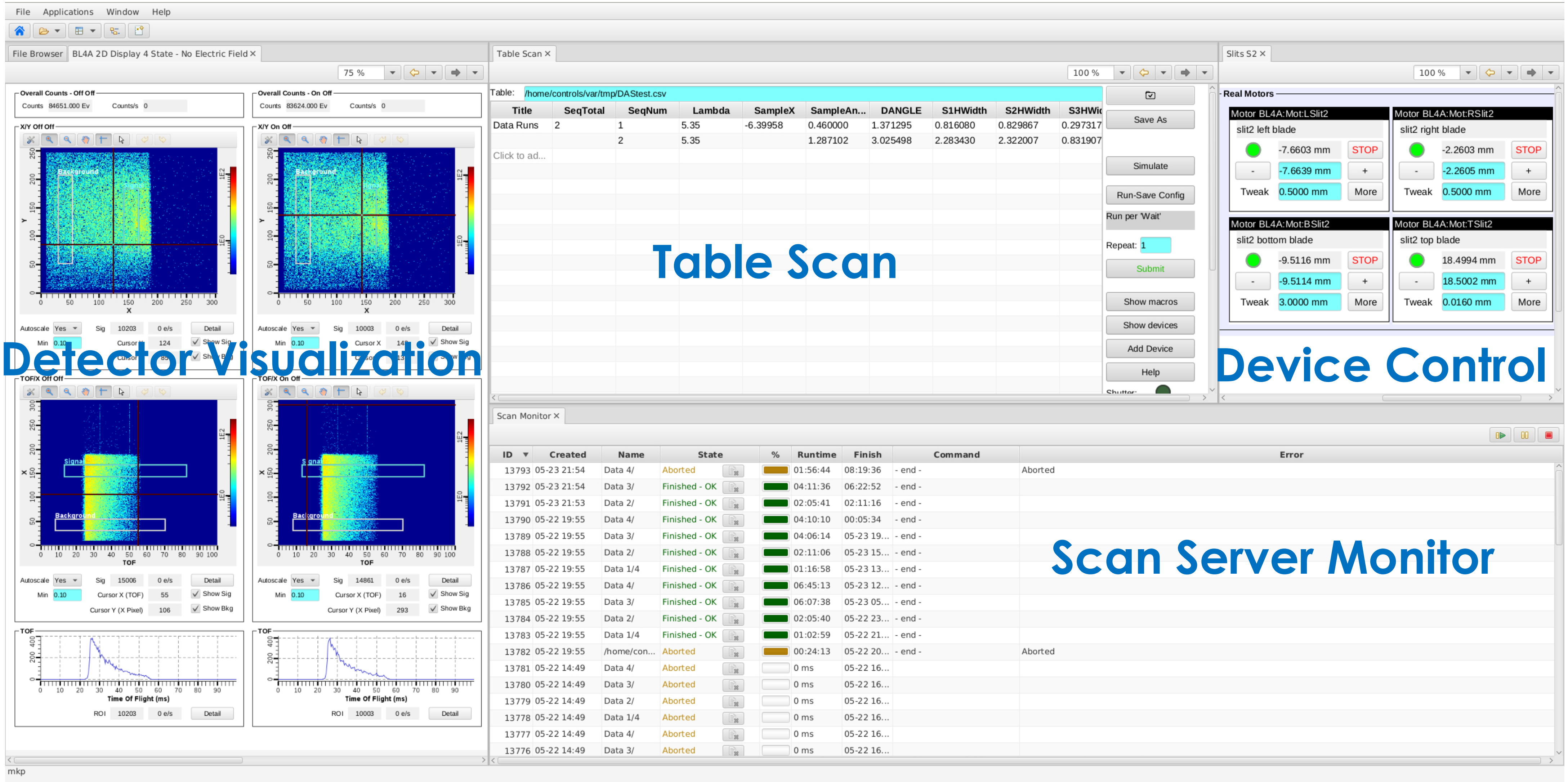
- SMS (Streaming Management Service)
  - Receive detector data (pixel ID, Time Of Flight) & other fast event data from EPICS detector software.
  - Receive metadata packets from the PVSD service (motor positions, sample environment data).
- PVSD (PV Streaming Daemon)
  - Monitor EPICS PV (channel access) and send updates to SMS
- STC (Streaming Translation Client)
  - Receive the aggregated data stream from SMS and write into NeXus file. Notify the analysis software so data reduction can start.

Control System Hardware, Servers, Clients



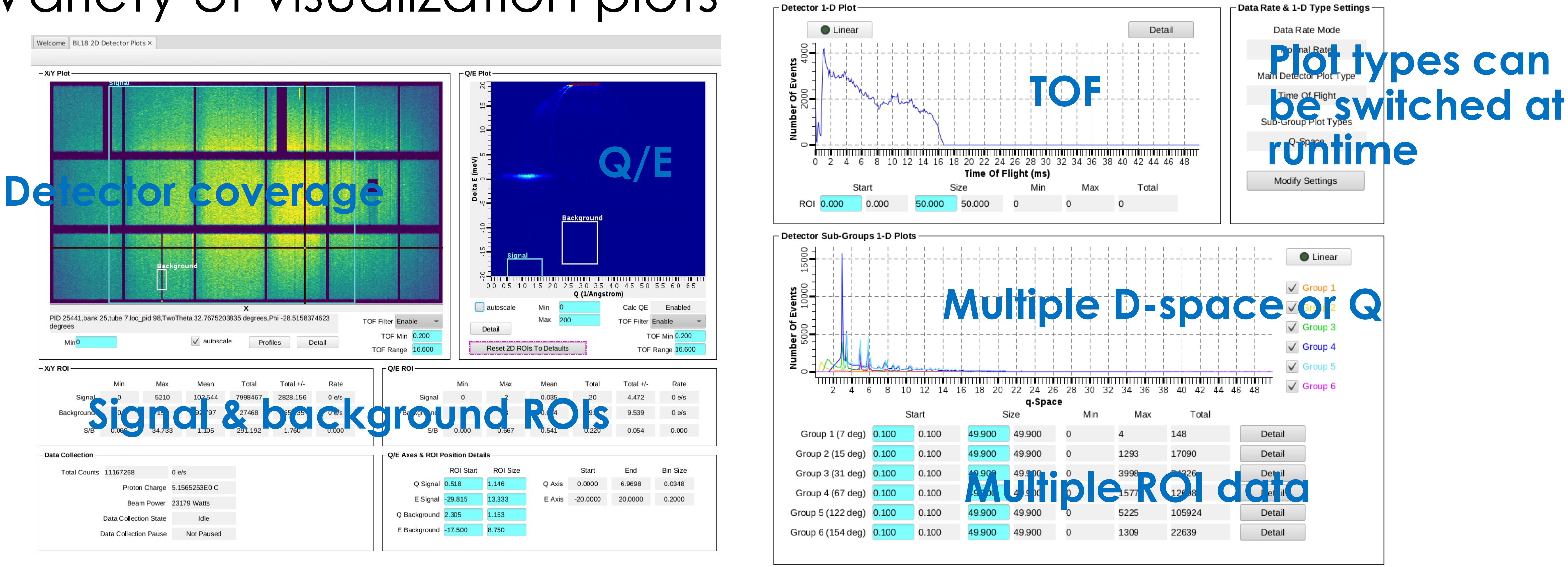
Many hardware protocols (<1Hz to MHz, ethernet, serial, PCIe, USB, etc)

Typical instrument user interface using CS-Studio



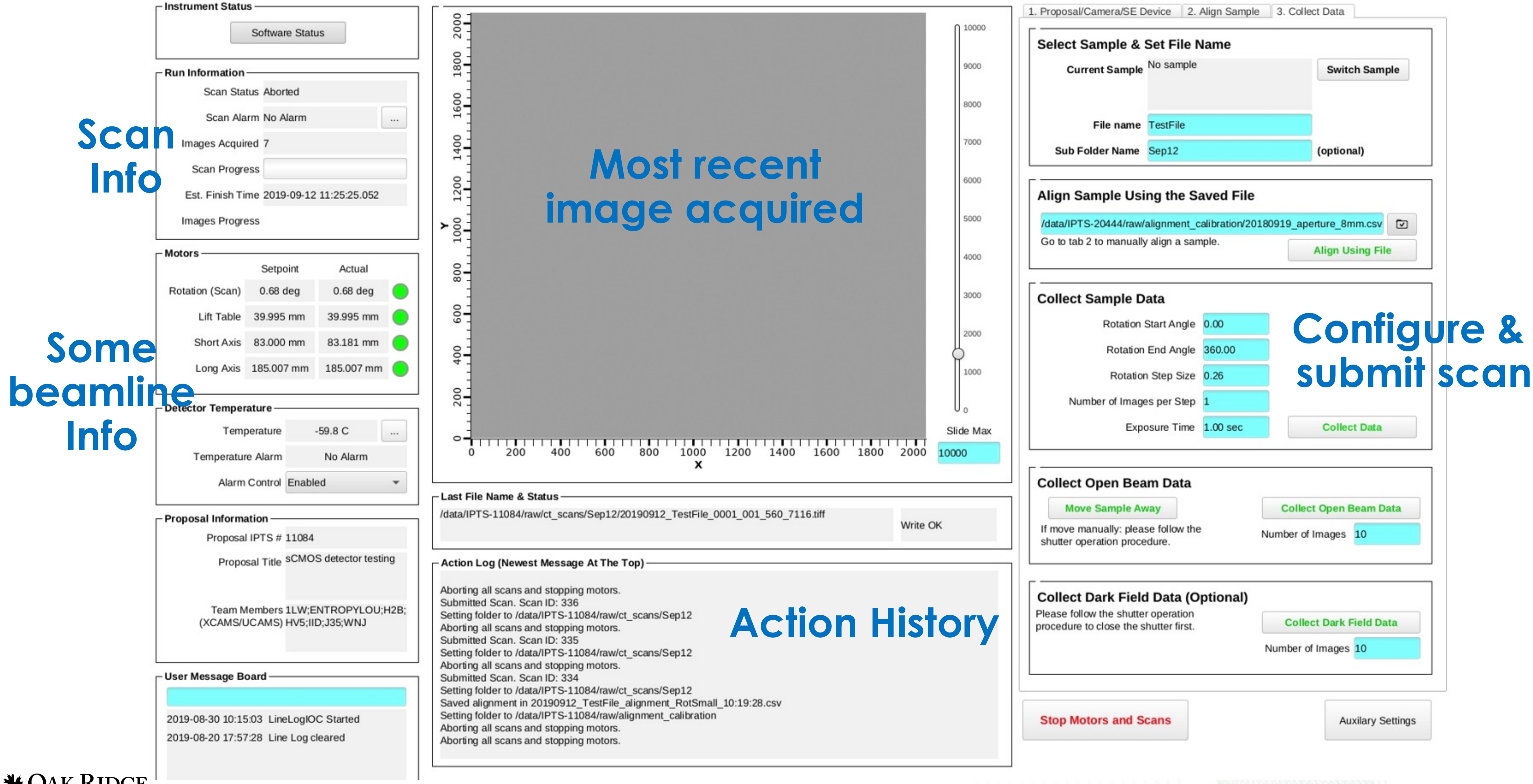
CS-Studio: https://controlsystemstudio.org/

Variety of visualization plots



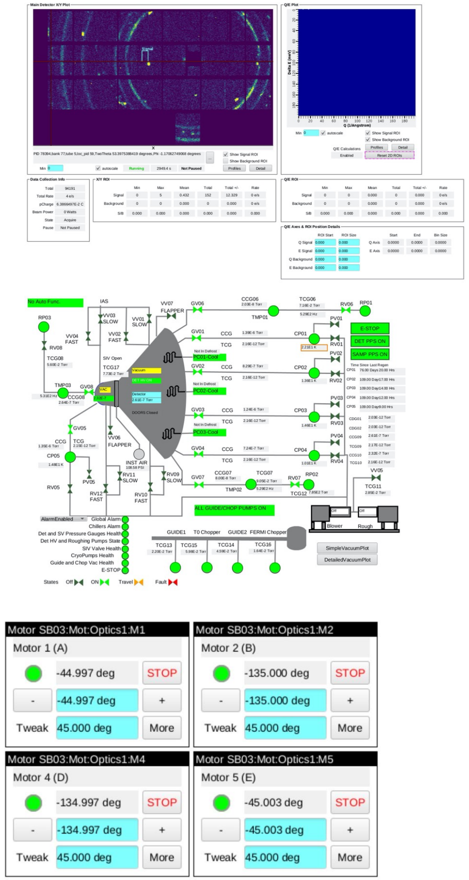
- The visualization aims to be as efficient as possible and pre-compute what we can.
- Can deal with several million events/s for complex plots (like Q/E) or 10-20Me/s for simple detector coverage, TOF, d-space, etc.
- Some beamlines are running up against CPU limits though.
- Investigating more multi-threaded solutions and/or GPU use.
- Eventually expect to be limited by what computing we can afford.

Example of an experiment specific user interface (CG1D Imaging)

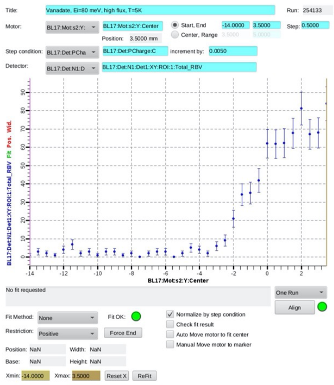


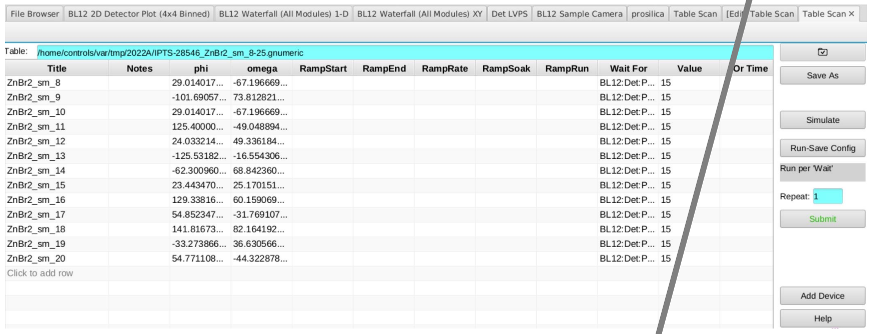
Variety of EPICS Client Applications

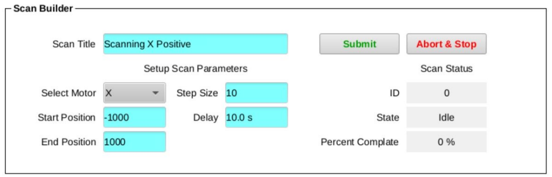
Control System Studio (CS-Studio)



CS-Studio Scan (Automation) Applications







These use the SNS 'Scan Server'

Client Side Scripts (Automation)

```
from epics import caget, caput

# Read a PV.
value = caget('motor_x')

# Write a channel.
# Usually best to wait for completion.
caput('motor_x', 8, wait=True)
```

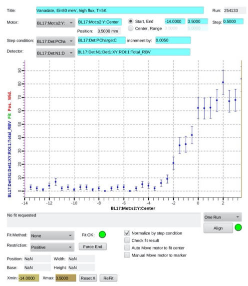
Server Side Scripts

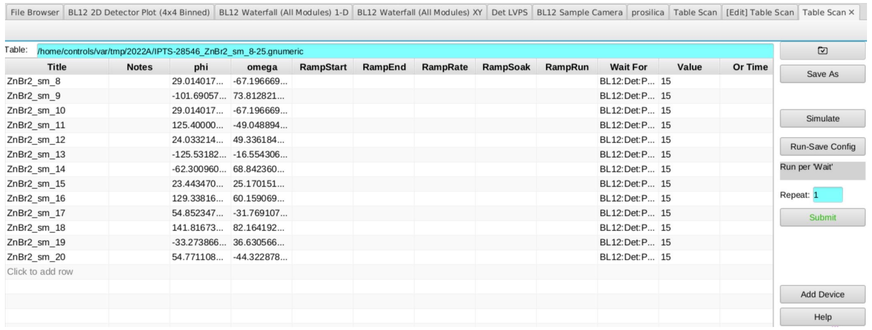
```
from scantools.scripting import *
Set("MotorZ", 18.2)
for x in frange(0.5, 2.5, 0.1):
    Set("BL12:Mot:x", x)
    Run("Sample run z=18.2 x=%f" %x, wait_seconds=300)
Submit()
```

How does the scan server work?

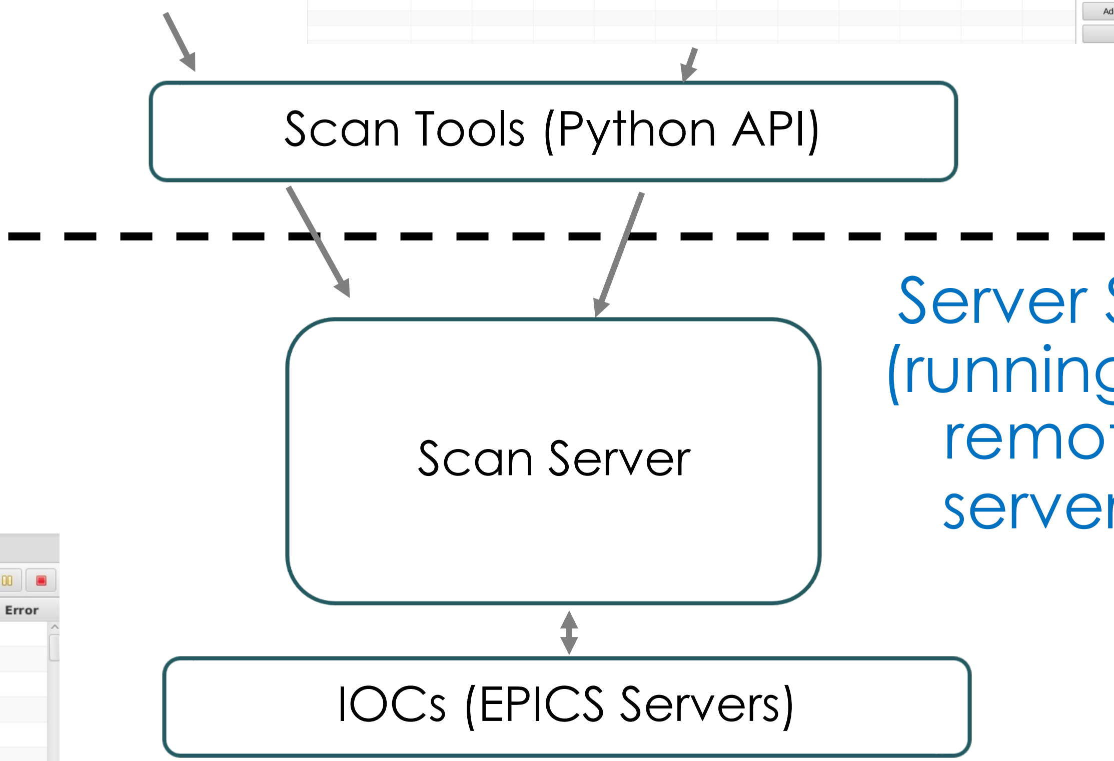
- Clients can submit jobs (scans) to the scan server. A job is a list of commands.
- There is a ‘scan tools’ layer to ensure consistency of behavior.
- Scans can be queued up.
- Scans can be edited, paused or cancelled after submitting.
- Scans run remotely on a server.
- Very reliable (10 years of operation)
- Defined API and Python client library.
- Multiple clients can see what is going on via the ‘Scan Monitor’.

Client Side (desktop)





Title	Notes	phi	omega	RampStart	RampEnd	RampRate	RampSoak	RampRun	Wait For	Value	Or Time
ZnBr2_sm_8		29.014017...	-67.196669...						BL12:Det:P...	15	
ZnBr2_sm_9		-101.69057...	73.812821...						BL12:Det:P...	15	
ZnBr2_sm_10		29.014017...	-67.196669...						BL12:Det:P...	15	
ZnBr2_sm_11		125.40000...	-49.048894...						BL12:Det:P...	15	
ZnBr2_sm_12		24.033214...	49.336184...						BL12:Det:P...	15	
ZnBr2_sm_13		-125.53182...	-16.554306...						BL12:Det:P...	15	
ZnBr2_sm_14		-62.300960...	68.842360...						BL12:Det:P...	15	
ZnBr2_sm_15		23.443470...	25.170151...						BL12:Det:P...	15	
ZnBr2_sm_16		129.33816...	60.159069...						BL12:Det:P...	15	
ZnBr2_sm_17		54.852347...	-31.769107...						BL12:Det:P...	15	
ZnBr2_sm_18		141.81673...	82.164192...						BL12:Det:P...	15	
ZnBr2_sm_19		-33.273866...	36.630566...						BL12:Det:P...	15	
ZnBr2_sm_20		54.771108...	-44.322878...						BL12:Det:P...	15	
Click to add row											

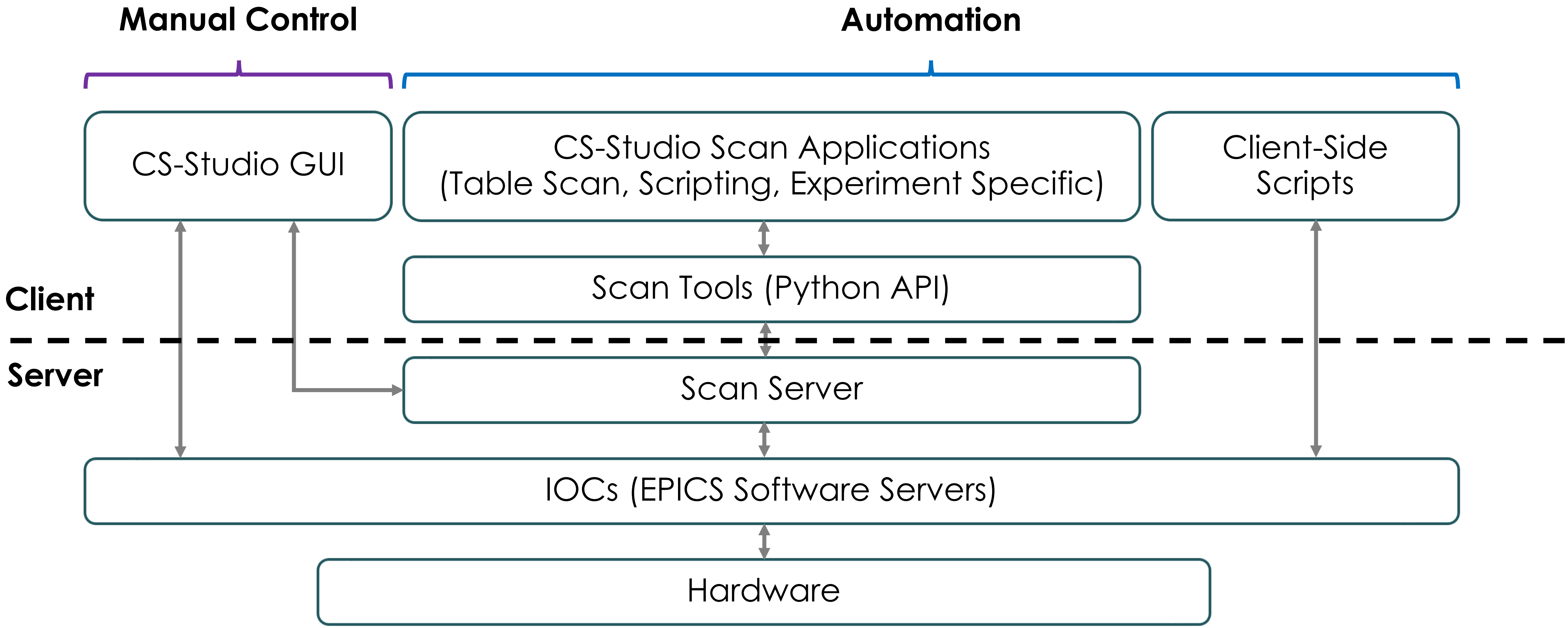


Server Side (running on remote servers)

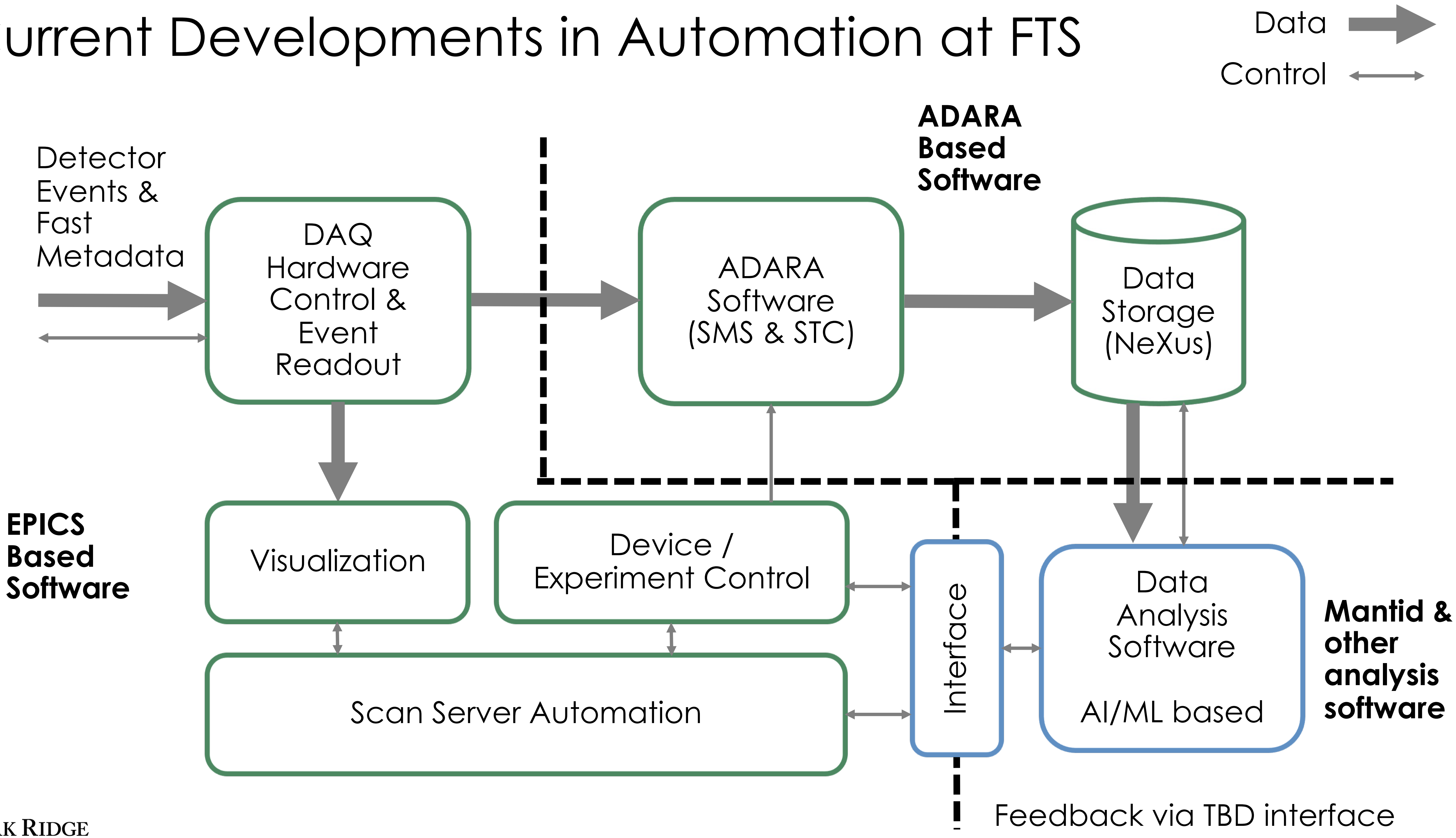
ID	Created	Name	State	%	Runtime	Finish	Command	Error
406	11:47:47	<not saved to file>	Running		00:00:36	11:48:22	Delay 100.0 sec. Remaining: 00:01:03	
405	11:47:38	<not saved to file>	Finished - OK		00:00:01	11:47:39	- end -	
404	11:47:35	<not saved to file>	Finished - OK		00:00:01	11:47:36	- end -	
403	11:47:32	<not saved to file>	Finished - OK		00:00:01	11:47:33	- end -	
402	11:47:29	<not saved to file>	Finished - OK		00:00:01	11:47:30	- end -	
401	11:46:32	<not saved to file>	Finished - OK		00:00:01	11:46:33	- end -	

Scan Monitor (panel in CS-Studio)

How does it all fit together?



Current Developments in Automation at FTS



Considerations for STS

In terms of the control system software and data acquisition systems we will be focusing on satisfying the Instrument requirements, reliability and compatibility with existing FTS systems.

One of the STS global requirements is:

R10 - STS facility shall be designed to maintain compatibility with the SNS Facility

Once completed the STS will transition back into the Neutron Sciences Directorate and will be operated in conjunction with the SNS facility. Consequently, it is imperative that designs of technical systems are compatible with the existing systems on the SNS facility. The STS management team and technical managers liaise continuously with their SNS facility counterparts and include them in design reviews to ensure compatibility.

⇨ STS is not a green field site, and if we implement something new then it must be something that FTS can adopt and something that Neutron Sciences is able to eventually support.

Plans for STS Controls & Data Acquisition

- Re-use existing FTS DAQ designs with possible hardware refresh.
  - FTS has 15+ years of experience designing and building the data acquisition electronics.
  - As much as possible aim for common design across FTS and STS.
- Plan to support the higher data rates that we will see at STS.
  - Higher speed data acquisition networking links
  - Parallel data acquisition streams
  - Higher speed storage systems
- (Better) support for multimodal measurements so we can capture all kinds of fast-metadata in parallel with neutron detector events.
- Continue use of SNS scan server and automation tools to be able to support a variety of scan applications.
- EPICS/ADARA based control system and data acquisition using CS-Studio as the main user interface.

Plans for STS Computing & Network

- New dedicated server room in STS Instrument Hall
  - 18 racks for networking, compute and storage
  - Higher density compute compared to FTS (to support GPU based systems)
  - Expected to support the eventual 22 Instruments
- High speed network links to each Instrument (10 to 100 Gbit/s)
- Each Instrument will likely have:
  - High speed networking within Instrument (10 to 40 Gbps)
  - 1 or 2 dedicated compute racks, 1 networking rack
  - GPU-based local analysis compute for AI/ML feedback
  - Remote experiment and automation support (following FTS experience)

Plans for Control System / Scientific Software Interfaces

The control system & data acquisition system is expected to interface to the scientific software in the following ways:

- At the data storage systems (file formats, post-processing jobs, etc)
- To support AI/ML experiments (so an experiment can be driven by the data analysis software, which would have to interface to EPICS)

These interfaces are still TBD but should follow developments on FTS or be eventually compatible with FTS.

Questions?