

# Data Acquisition at the NSLS II



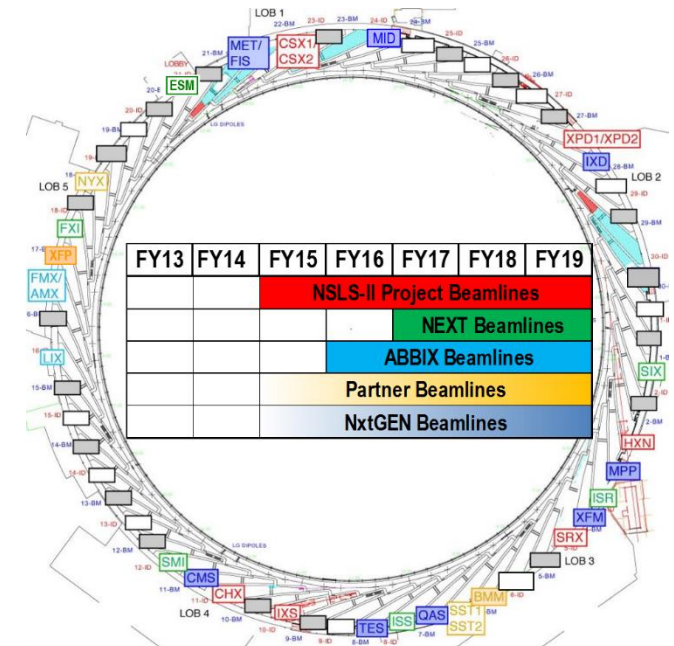
**Leo Dalesio, (NSLS II control group)  
Oct 22, 2014 (not 2010)**

# Introduction

- When we scaled some data collection to the frame size and data sets that are expected at NSLS II we found that most of the delays in processing the data were:
  - Time to find the data sets that were needed for the analysis.
  - Movement of the data into memory.
- No Xray facility has demonstrated an architecture that supports the range of requirements posed by an entire facility.

# Data Rates

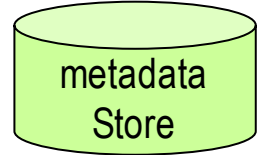
- Each beam line can produce 8 MB frames up to 1 kfps, with data sets up to 1 TB per minute.
- There will be over 60 experimental end stations when the facility is completed.
- Detectors buffer all of this data on SDD storage and does not pose a challenge.



# Data Management

- 1 data set per minute, with 60 beam lines operating 20 hours per day
- 72,000 data sets per day, over 2 M data sets in one year.
- 20,000 signals are controlled and monitored on each beam line and recorded as time series.
- ***A MetaDataStore service is being provided to support the tracking of data***
- ***A ChannelArchiver is provided to store time series data from the beam lines.***
- ***A Science Data Store is being provided to store large frame rate data sets.***

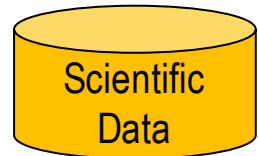
# Data Is Stored in Appropriate Repositories



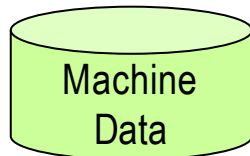
The metadata store contains all configuration and environment parameters that are used to characterize a data set. Searches for specific data sets must be optimized.



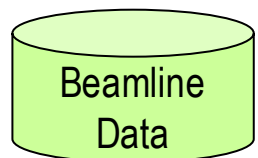
The Experiment Log Book contains information about the data collection. It may be used to log steps in the data analysis such as export file formats used. Log entries can be made by manually or programmatically.



The Scientific Data is the repository for large data sets. These data sets include matrices and image sets with the metadata required to plot or view the data. These are stored by sample and time stamp. Data can be written from experiment control or submitted from analysis. Other data sets can also be sent here such as reference or model data.



Machine data is all time series data that is taken from the accelerator that is available for use in analysis.

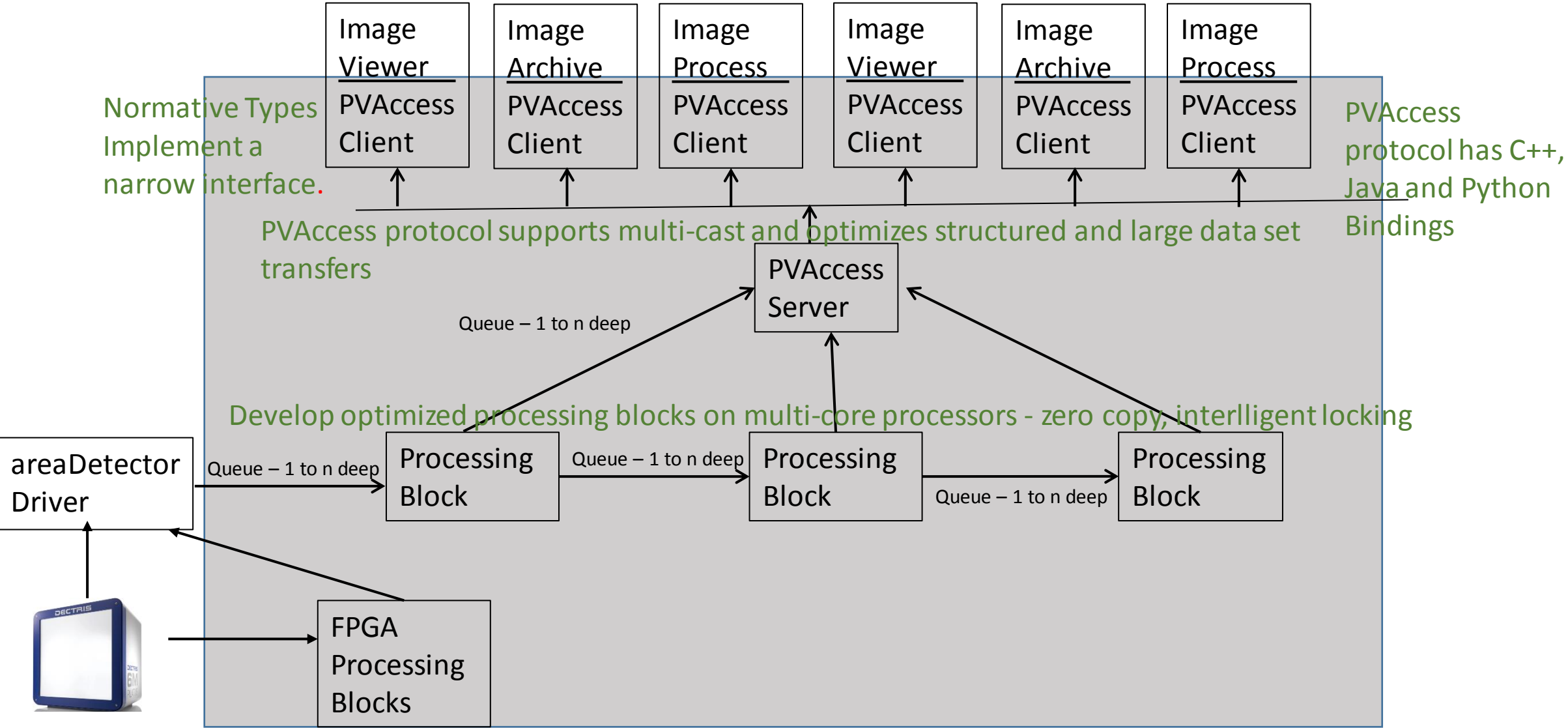


Beamline data is all time series data that is taken from the end station, beam line or accelerator that is available for use in analysis.

# Data Processing

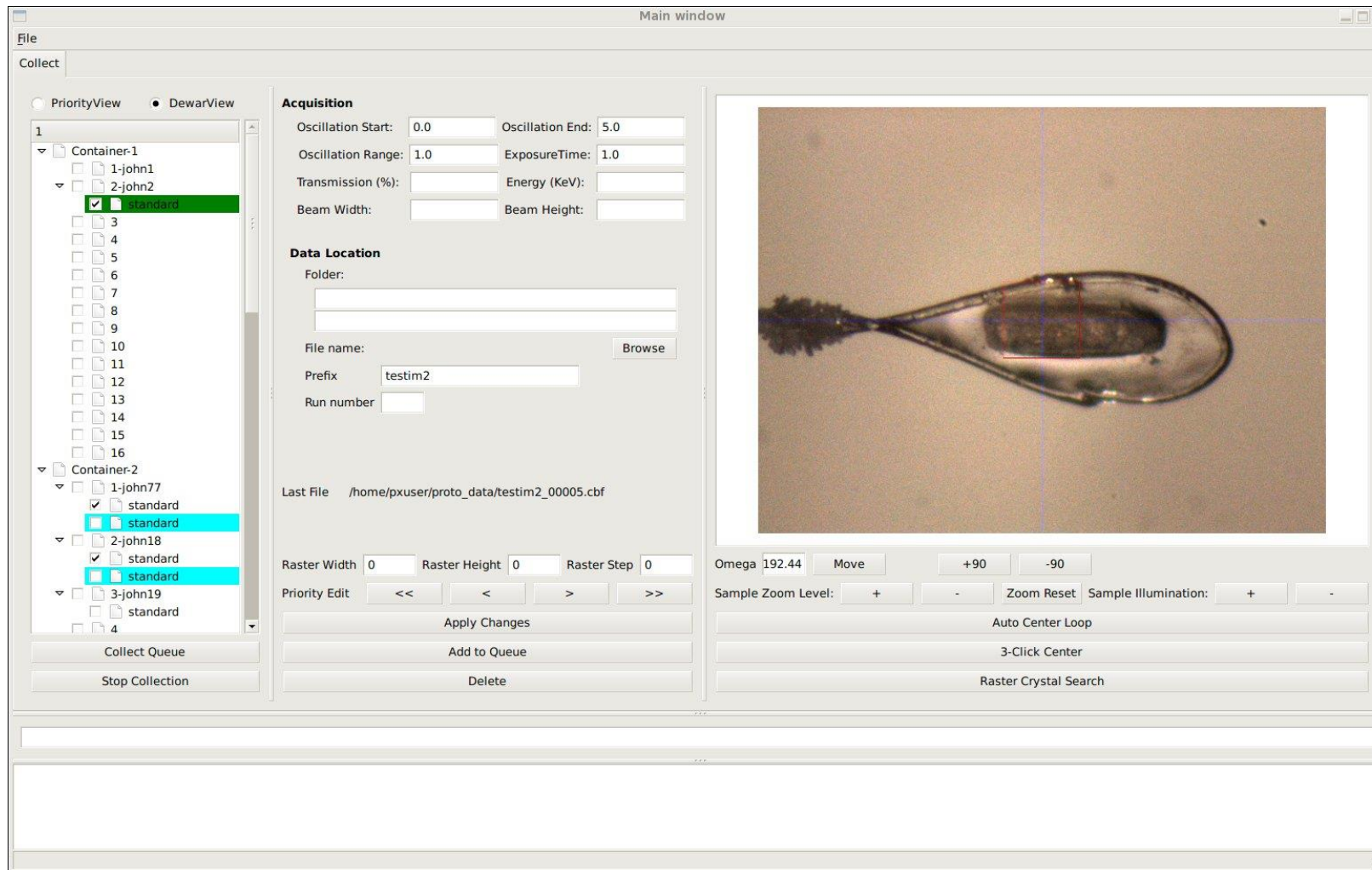
- Scientists may be at the beam line and interact with the experiment control. In this case, we want to minimize the latency required to process the data and provide information.
- Scientists may have submitted samples and interact with a scheduler and analysis report. In this case, we want to batch process the data and provide remote analysis results and reanalysis requests.
- There are many architectural elements being modified or developed to address fast sample feedback to the scientist.
  - *Zero copy transportation of large arrays in front end computers*
  - *Modified EPICS locking to use multicore front ends for array processing*
  - *Requests to vendors to provide parallel SFP ports to pipe data into our FPGAs*
  - *Multicast network protocol to send data to multiple clients*
  - *We are developing a sample management application.*

# Data Processing Pipeline for fast results





# Sample Management



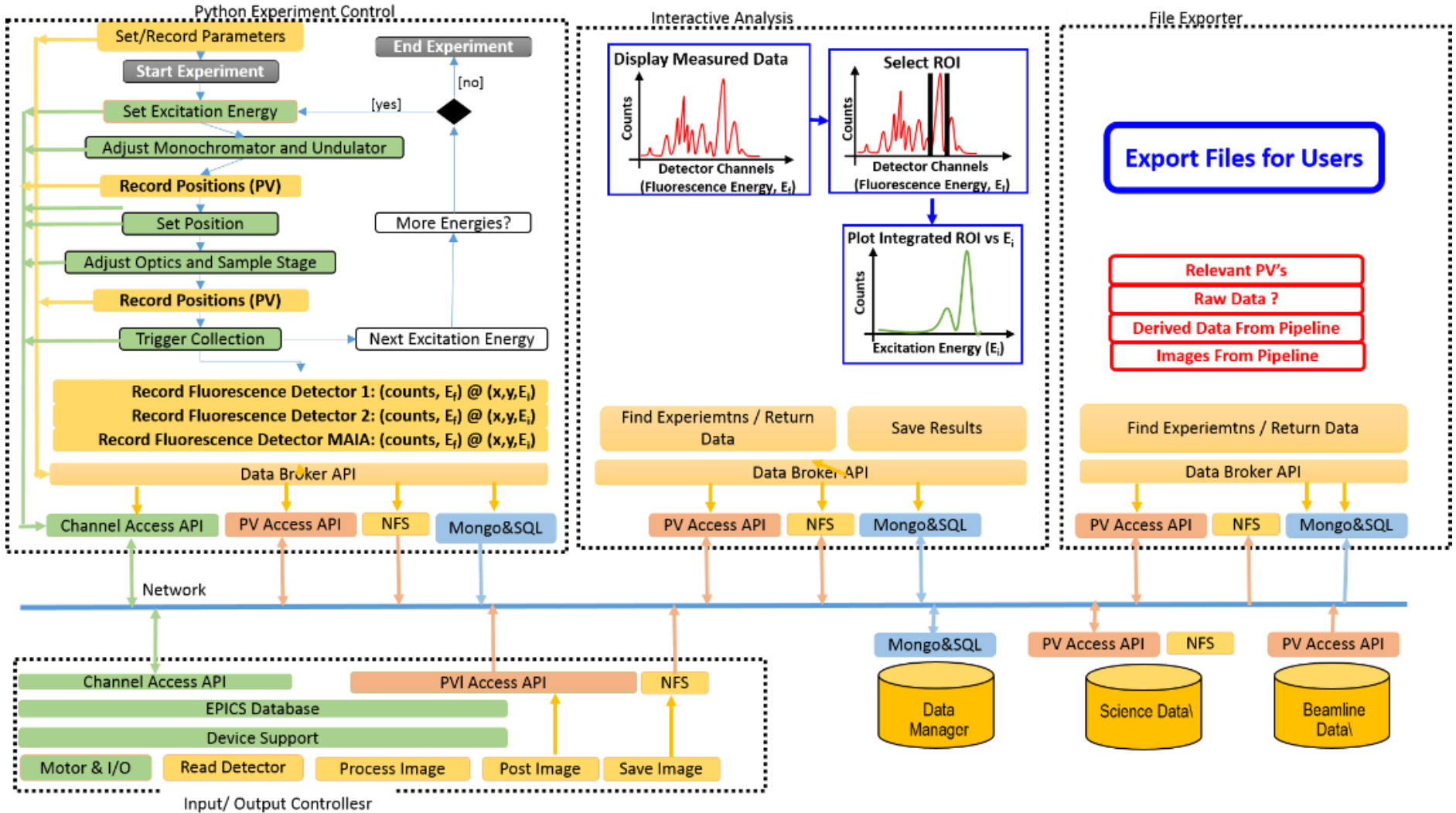
Samples are either viewed by their order in the sample changer dewar (DewarView), or by the order the user would like to collect (PriorityView). Priority The color coding for now is cyan=done, green=running, checked means requested to run when the "Collect Queue" button is pressed below the tree.



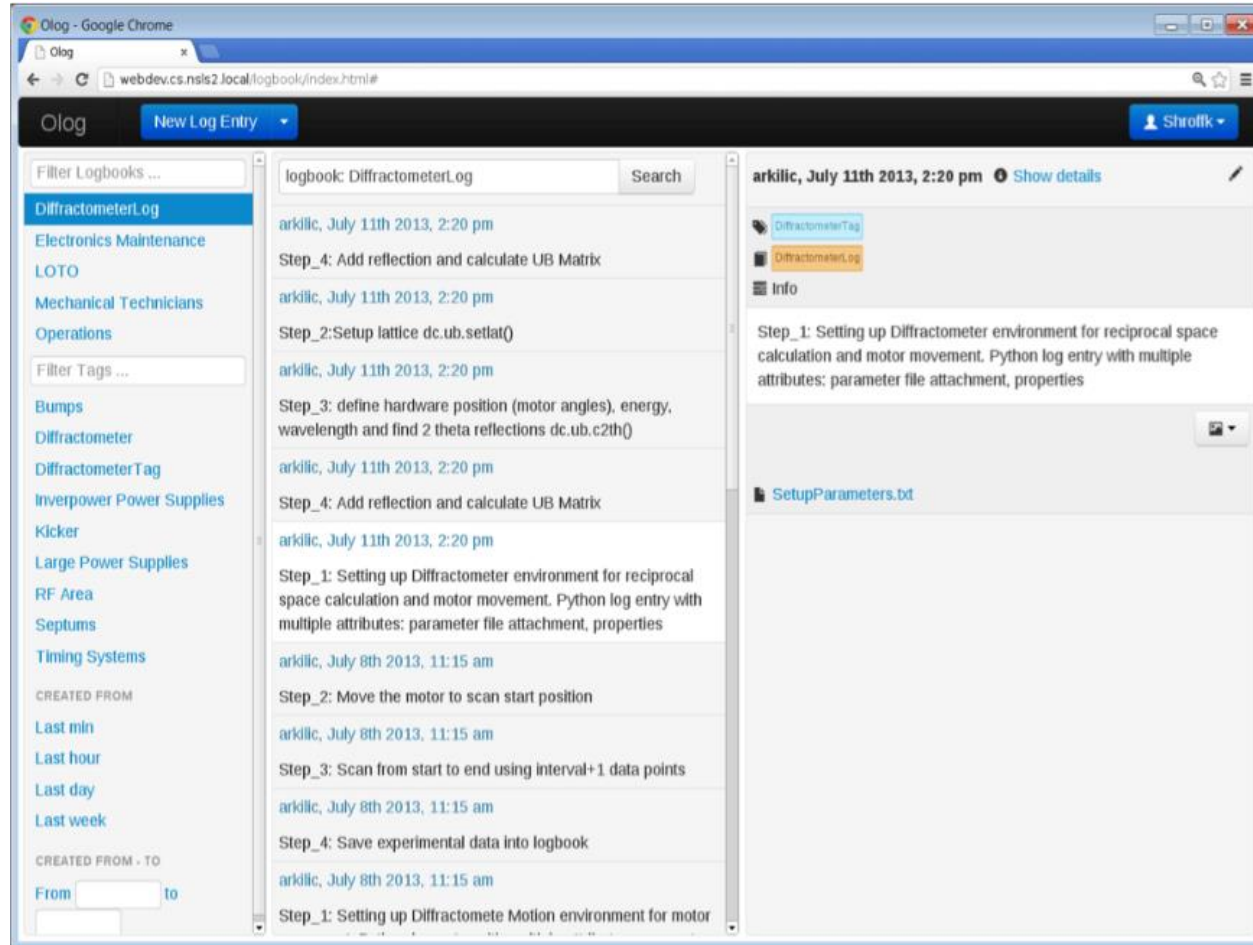
# Data Analysis

- There are many techniques used to study various samples at NSLS II.
- Closer to the raw data, many of the algorithms used are common across beam lines.
- There are some very specialized analysis routines required as well
- Provenance of data must be preserved to validate the scientific process.
- “Scientists would rather use someone else’s toothbrush than their analysis routines”
- Users will want to use different, competing, evolved, higher performance, or just better codes.
  - To allow access to raw and processed data, a Data Broker API is provided
  - A standard library of python processing routines is produced and supported.
  - Processing routines use standard data types
  - Results from processing can be stored back into the data management system
  - A visualization library is supported for 1D and 2D data and data sets.
  - A data flow system is being developed to support the development of data analysis. VisTrails is being evaluated as a tool to provide this.

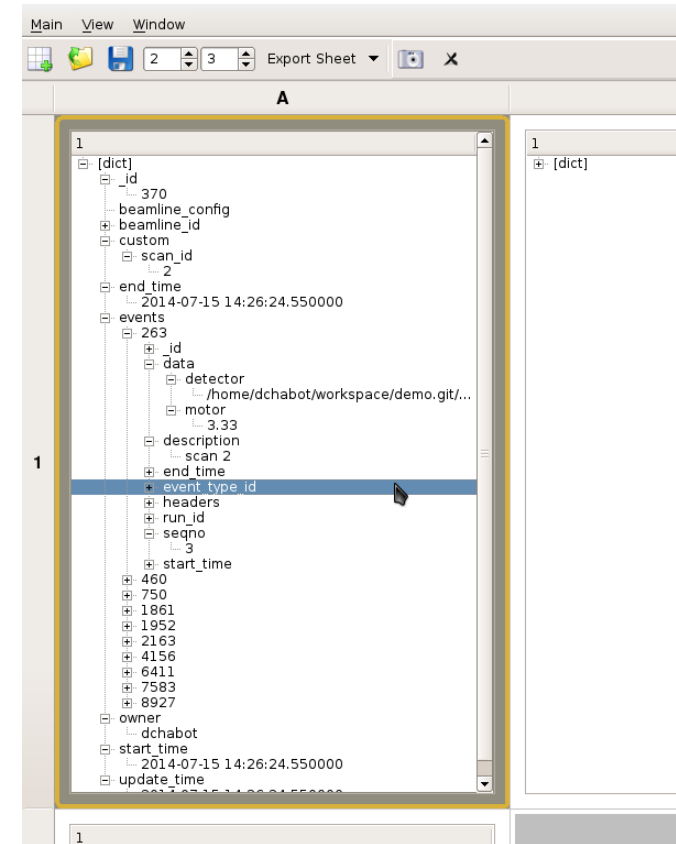
# Design Status – Experiment Control (1 OF 4)



# Design Status – Experiment Control (2 OF 4)



The screenshot shows a web browser window titled "Olog - Google Chrome" displaying a logbook interface. The URL is "webdev.cs.nsls2.local/logbook/index.html#". The interface includes a navigation menu on the left with categories like "DiffractometerLog", "Electronics Maintenance", "LOTO", "Mechanical Technicians", "Operations", "Bumps", "Diffractometer", "Diffractometer Tag", "Inverpower Power Supplies", "Kicker", "Large Power Supplies", "RF Area", "Septums", and "Timing Systems". The main content area shows a list of log entries for "logbook: DiffractometerLog" with a search bar. The selected entry, dated "July 11th 2013, 2:20 pm", includes steps such as "Step\_4: Add reflection and calculate UB Matrix", "Step\_2: Setup lattice dc.ub.setlat()", "Step\_3: define hardware position (motor angles), energy, wavelength and find 2 theta reflections dc.ub.c2th()", and "Step\_1: Setting up Diffractometer environment for reciprocal space calculation and motor movement. Python log entry with multiple attributes: parameter file attachment, properties". A "SetupParameters.txt" file is attached to the entry.



The screenshot shows a spreadsheet application window with a menu bar (Main, View, Window) and a toolbar. The spreadsheet displays a hierarchical data structure in a tree view. The root node is a dictionary [dict] with the following structure:

- dict
  - \_id: 370
  - beamline\_config
  - beamline\_id
  - custom
    - scan\_id: 2
  - end\_time: 2014-07-15 14:26:24.550000
  - events
    - 263
      - id
      - data
        - detector: /home/dchabot/workspace/demo.git/...
        - motor: 3.33
      - description: scan 2
      - end\_time
      - event\_type\_id
      - headers
      - run\_id
      - seqno: 3
      - start\_time
    - 460
    - 750
    - 1861
    - 1952
    - 2163
    - 4156
    - 6411
    - 7583
    - 8927
  - owner: dchabot
  - start\_time: 2014-07-15 14:26:24.550000
  - update\_time

# Design Status – Experiment Control (3 OF 4)

Query example <@xf23id1-ws3>

Query

scan\_id

owner edill

start\_time 2014-08-22 00:00:00

end\_time 2014-08-22 00:00:00

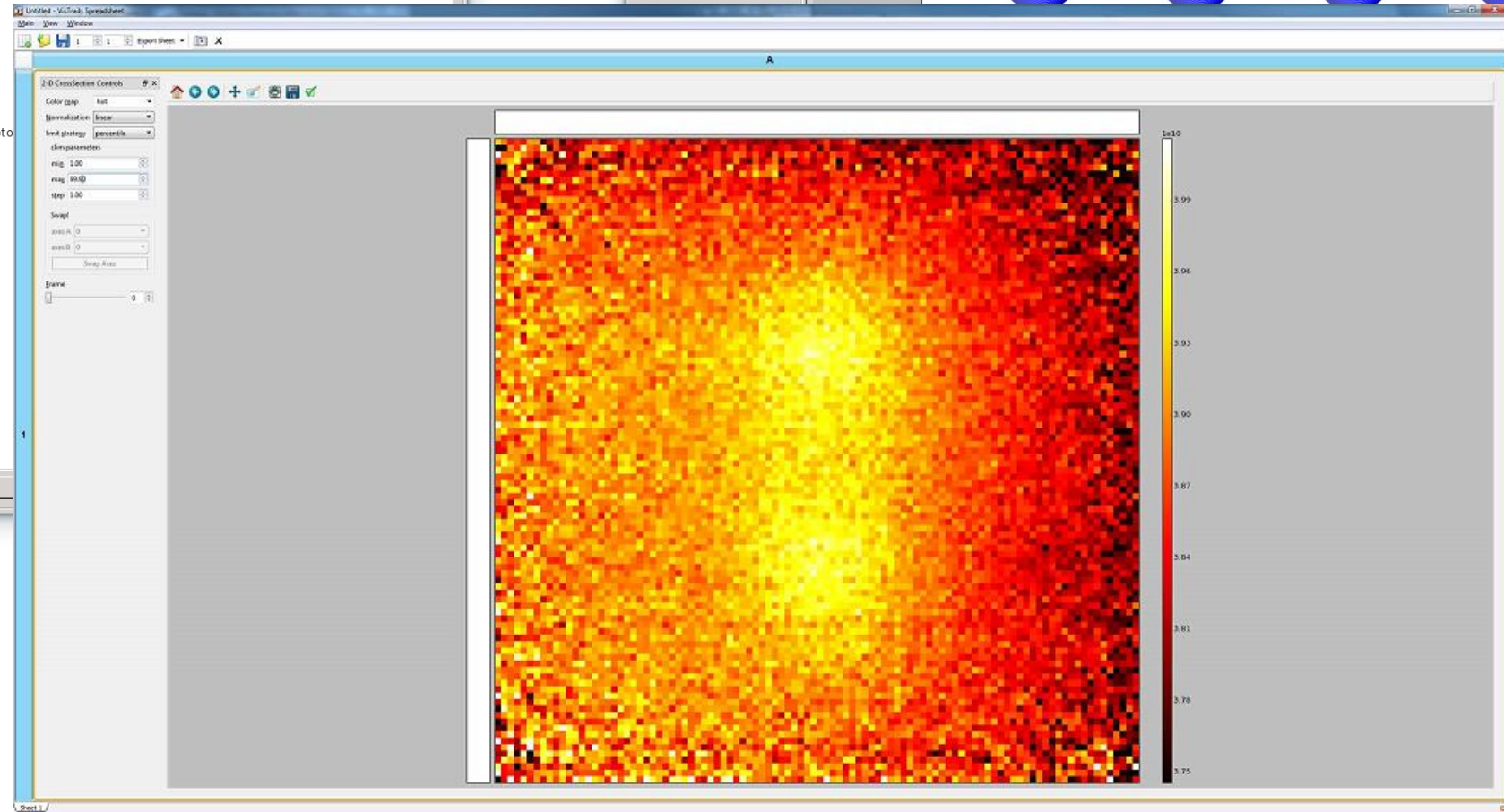
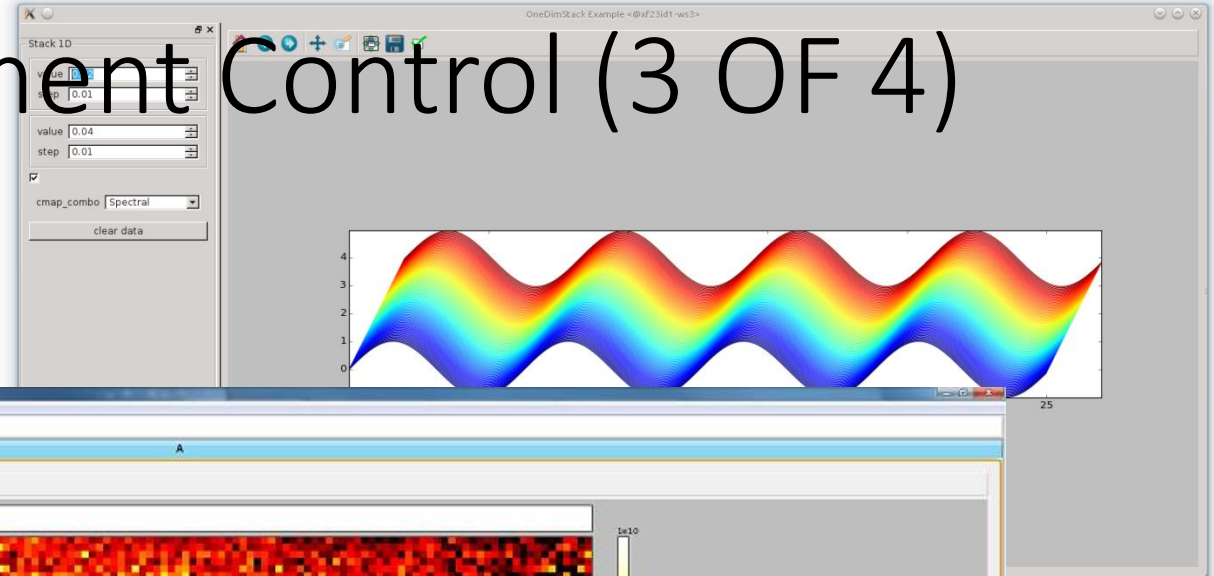
num\_header

data

Search

Results

- 1
  - header\_0
  - header\_1
  - header\_2
    - id
    - beamline\_id
    - configs
    - custom
    - end\_time
    - event\_descriptor\_0
      - id
      - descriptor\_name
      - event\_type\_id
      - events
        - event\_0
          - id
          - data
            - name
            - value
          - description
          - event\_descriptor
          - header\_id
          - owner
          - seq\_no
        - event\_2
          - header\_id
    - header\_id
    - tag
    - type\_descriptor
    - owner
    - scan\_id
    - start\_time
    - status
    - header\_3
    - header\_4
    - header\_5





# Data Storage

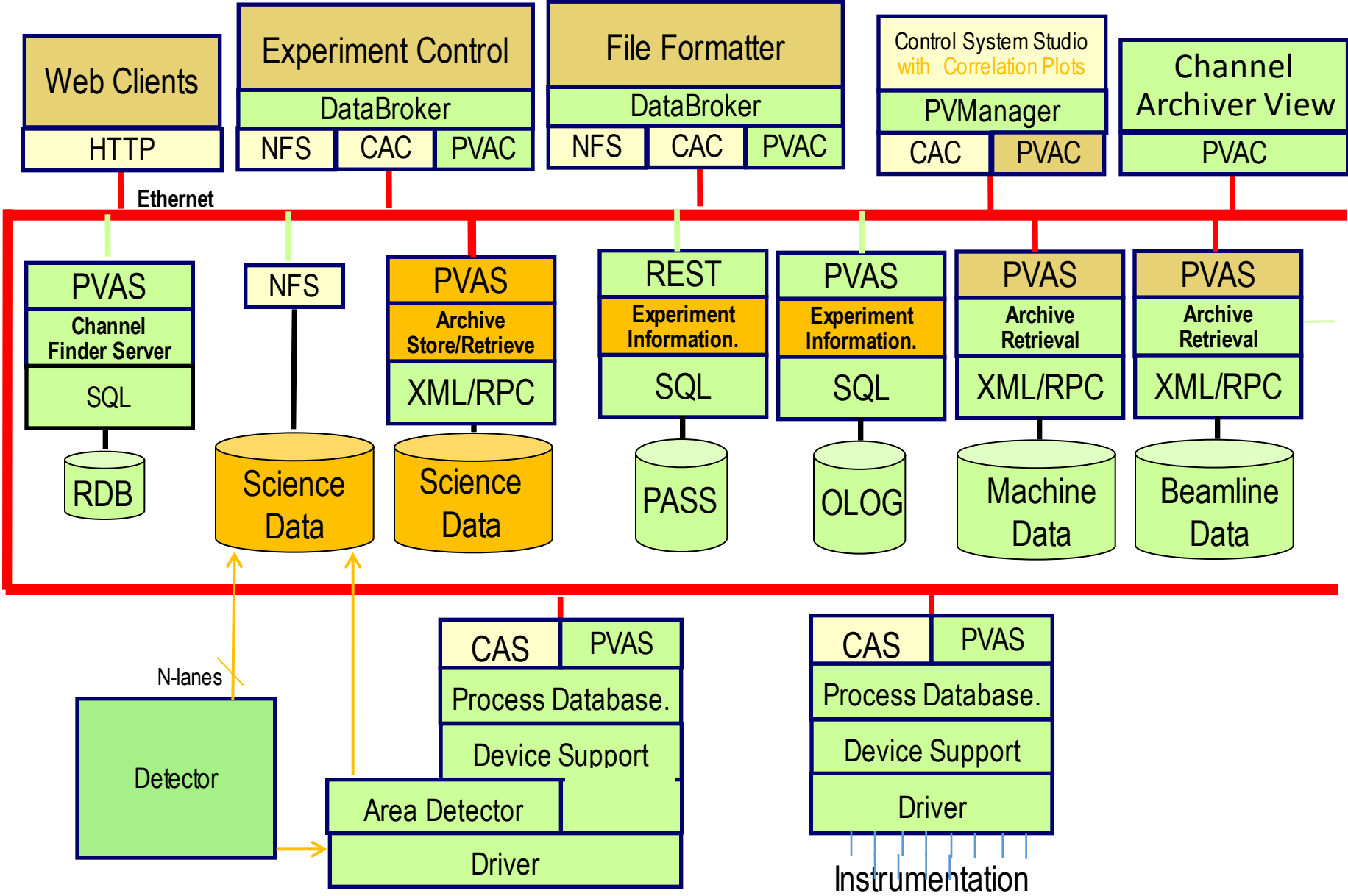
- Data must be kept online for some time and may be stored to tape for some time
- In 2016 we will need 2 PB to store one year of data, by 2020 35 PB will be needed.
- There is a requirement to track data provenance and store intermediate analysis results.
- Commercial solutions exist for this scale of data.
- Tracking Provenance is being considered in workflow tools and the metaDataStore



# Data Export

- Scientists need to put together frame rate, time series, and configuration data in specific formats for existing analysis codes.
- Scientists may want to export the data life cycle – from raw data through all analysis.
- The DataBrokerAPI can be used in Python to access all data. Code must be developed to repackage data from different data stores into a desired export format.

# Experimental Data Architecture



# Remote data access problem

- There is 1-10 Gbps link into BNL
- Our users sit at universities, coffee shops, small offices (not on a high performance network port).
- Analysis computers are being planned at BNL
- A user interface to the analysis codes developed and maintained at BNL is being developed.

# Conclusions

- Two distinct areas of concern must be covered:
  - Providing immediate feedback to allow scientists to effectively use their beam time.
  - Provide an analysis and data management framework that supports deep analysis after the experiment is run.
- No facility has demonstrated an architecture that supports the range of requirements posed by an entire facility.
- A service based architecture is being developed to support the DAQ and data analysis needs of NSLS II.
- Infrastructure is being used from an existing collaboration group that provides some strong basic capabilities.
- There is a strong set of software architects, analysis, visualization, and fpga programmers working on these issues.