Data Acquisition in DM-Radio 50L

DMR Collaboration Meeting

Pam Stark

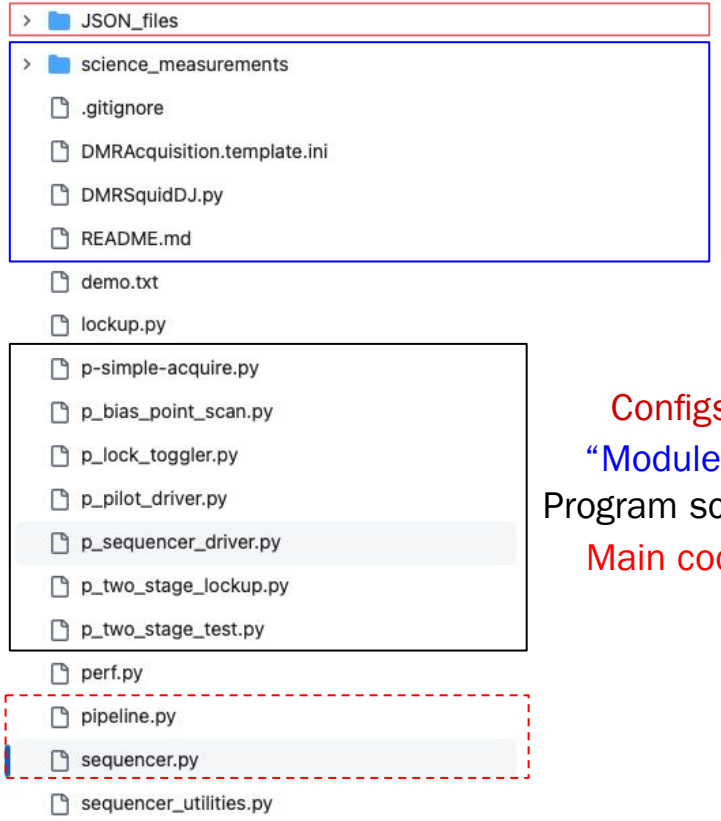
08/05/2024

How do we get science data out of DM-Radio 50L?

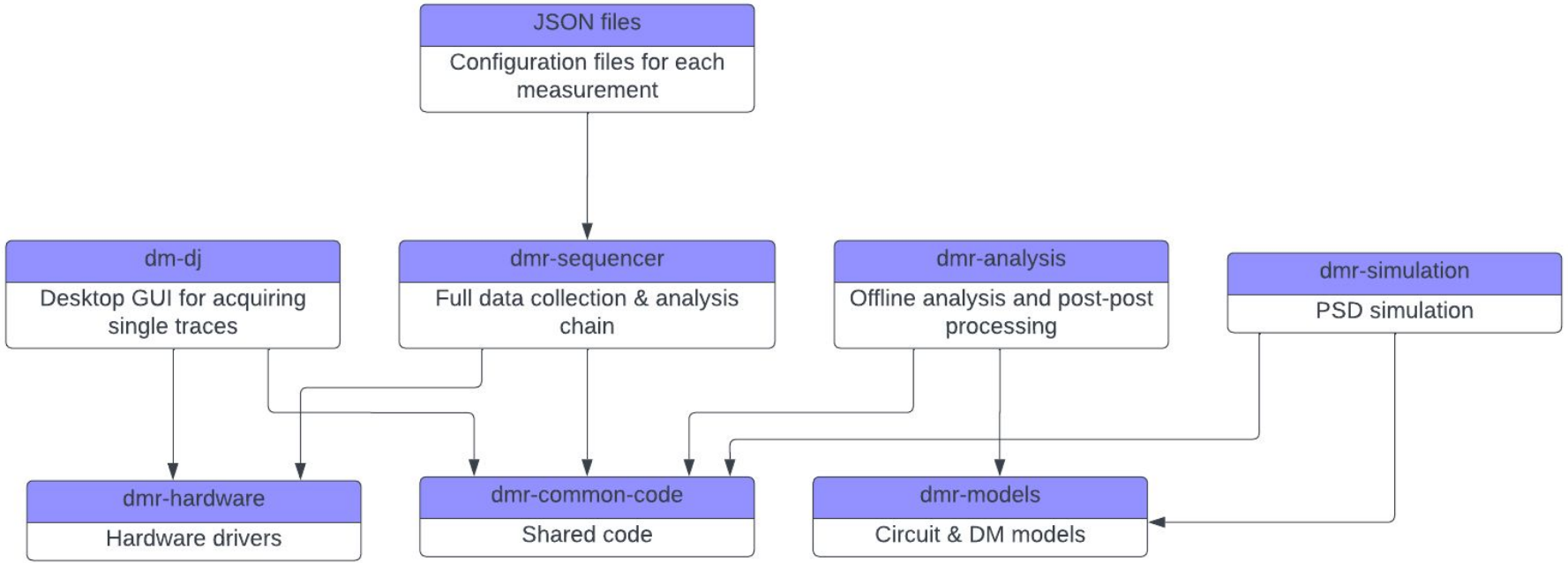
- DAQ efforts include:
  - Instrument controls & data collection
  - Run procedures
  - Remote monitoring
  - Data storage & management
- We will focus on a high-level overview of the DAQ and status updates on:
  - Sequencer software development
  - Hardware
  - Database & data storage

“Sequencer” is the software running the data pipeline

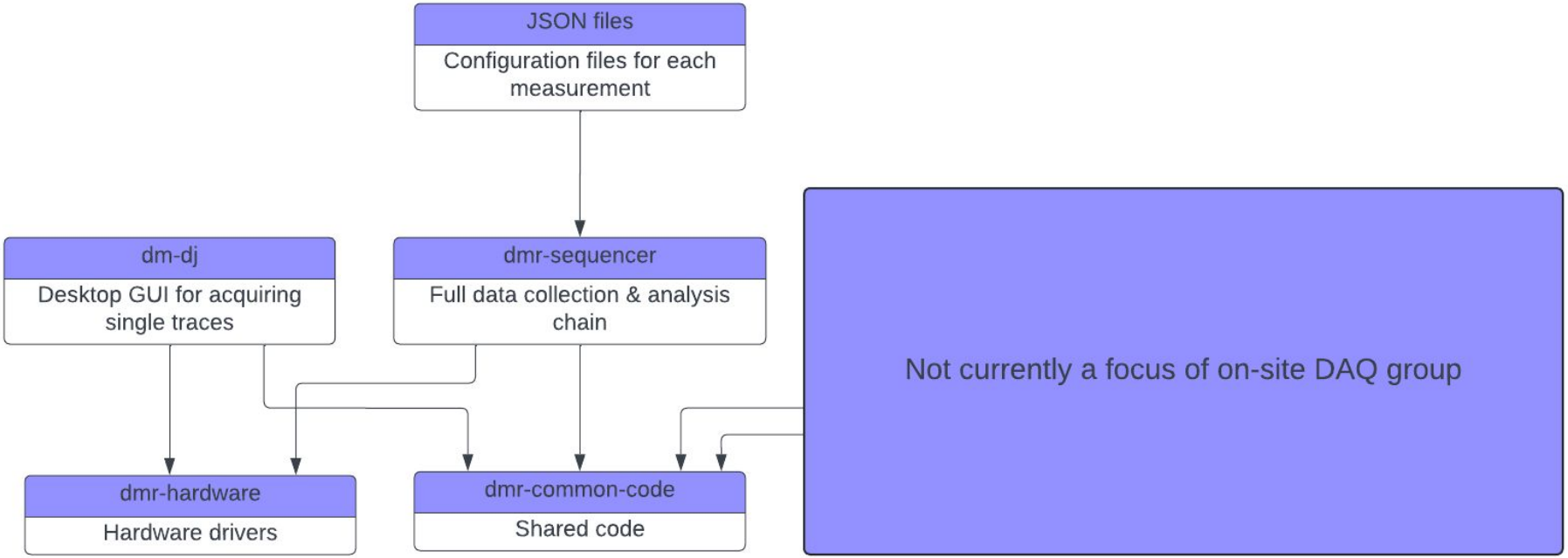
- Designed with flexibility in mind
  - **Modular** in nature for easier modification
- Processes are multi-threaded for speed-up and simultaneous operation
- Developed in Python and set up with Pycharm environment on DAQ computer



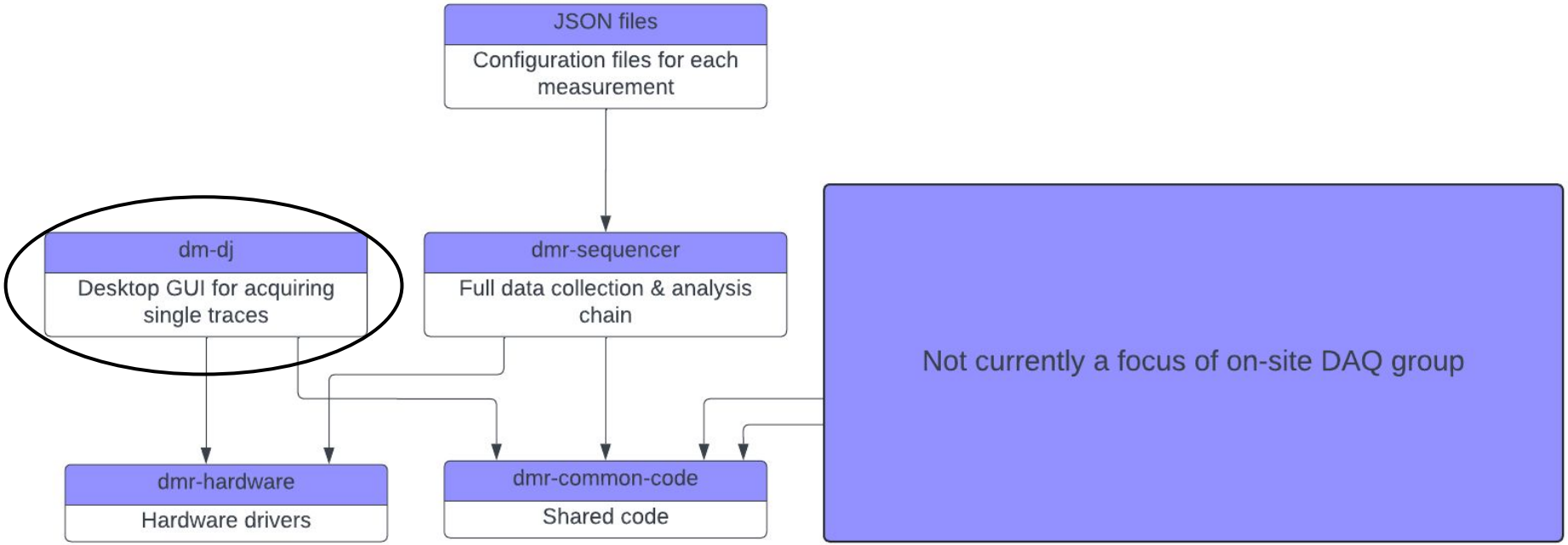
Sequencer Dependencies



Sequencer Dependencies



Sequencer Dependencies



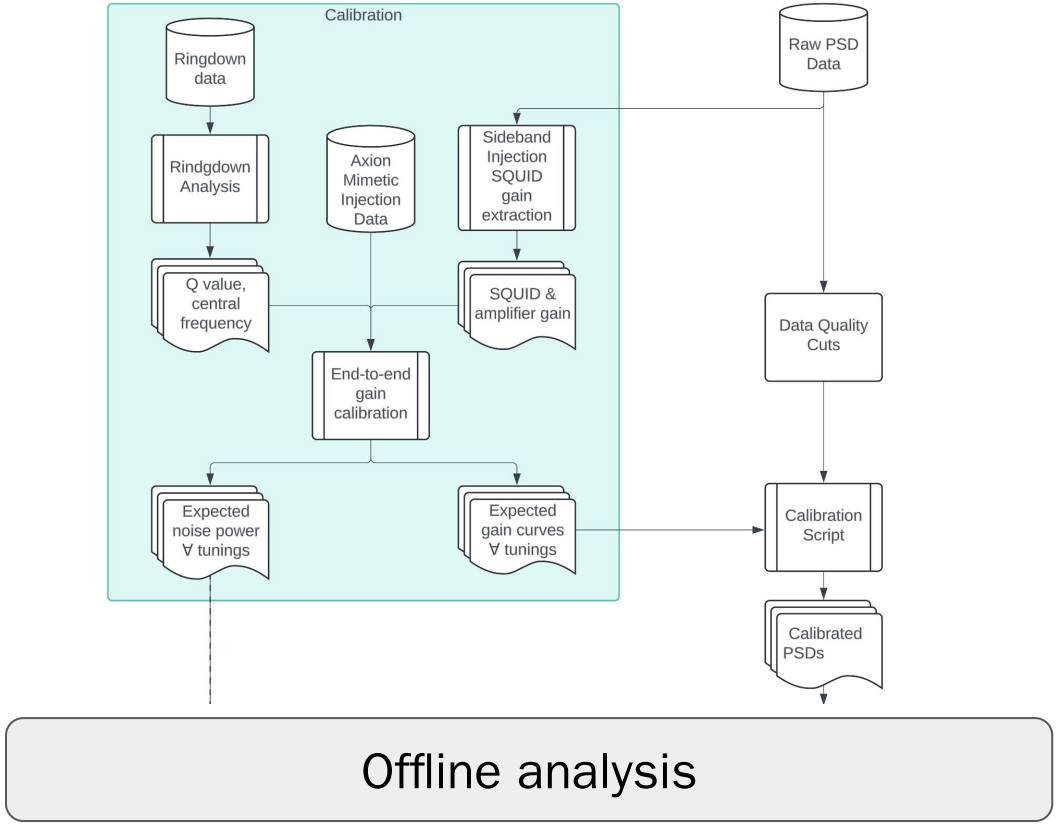
Sequencer Terminology

- A **sequencer step**:
    1) Performs a tuning step
    2) Sets SQUID bias point
    3) Perform measurements for each bias point

Note: sequencer will take data at several bias points

- A **measurement** is an on-line analysis module (ringdown, sideband, etc.) run sequentially in pipeline
- A **sequencer run** has several steps with different **measurements** & **measurement parameters**
- **Modular:** anyone can build an in-line analysis module by writing a "science measurement" script, defining a config file, and calling the measurement

Sequencer handles data-taking & “in-line” analysis



Sequencer will interface between several machines

Digitizer computer collects data and runs sequencer, slow controls computer connects to instruments, AWS hosts the database

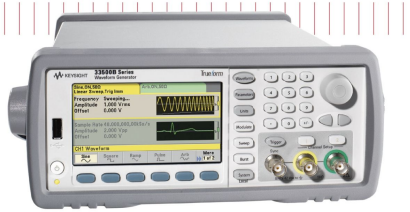
Networking

- Extensive conversation has gone into deciding which computers are on which network and remotely accessible
  - DAQ Computer: Development, Stanford subnet → accessible with VPN
  - Slow Controls: Hardware & SQUID controls, Stanford subnet → accessible with VPN
  - Snoopy: Fridge monitoring & controls, internal lab network
- DAQ & slow controls computers must communicate during runs
- All three physical computers must be able to push data to the database and can communicate with AWS server

Hardware interface with sequencer code

- No hardware should be directly connected to Digitizer computer to minimize load on computer & chance of error
  - Exception: direct remote connections between sequencer and instruments on the network
- Hardware communication between digitizer and slow controls computers enabled through Pyro servers

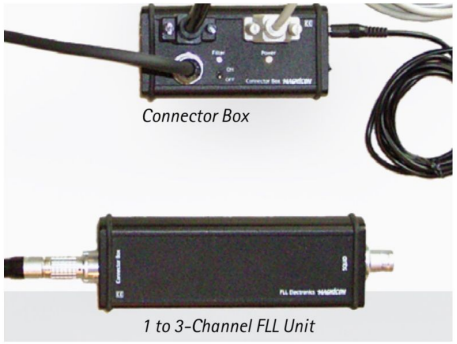
Hardware Overview



Function Generator

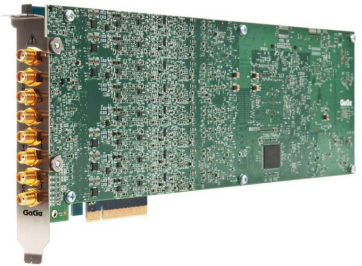
TBD: Will we have more than one piezo?

Attocube Piezo Motion Controller



Magnicon

Oscilloscope



Digitizer

Status updates on key instruments

- **Digitizer:**
  - Preliminary longevity tests show errors → on-site debugging in progress
- **Function generator:**
  - Direct connection tested with sequencer calibration runs
  - Remote connection established, not yet incorporated into software
- **Attocube*:**
  - Attocube hardware module written & incorporated into sequencer
  - Remote connection established and tested

*Special shoutout to Joe for sharing attocube resources!

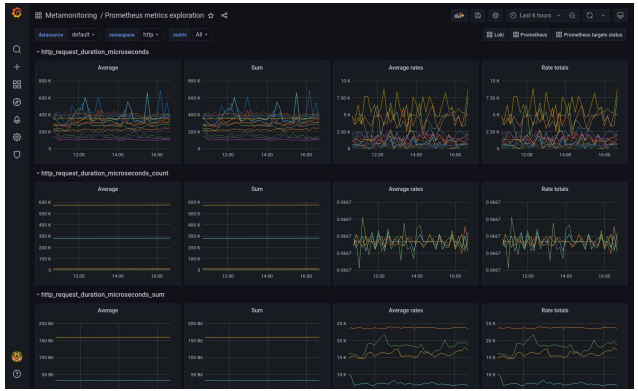
Sequencer stores and outputs data in hdf5 files

- Each run outputs a collection of hdf5 files containing measurements parameters (‘metadata’) and raw/PSD data
  - Includes config file inputs & hardware params
- No data will be stored in the databases: instead, will store run parameters/metadata with a pointer to these hdf5 data files
- More on this during the DAQ-a-thon….

How can remote analyzers interface with the data?

One-way Communication

- Grafana dashboard: front-end for data monitoring [3.18.210.26:3000](3.18.210.26:3000)



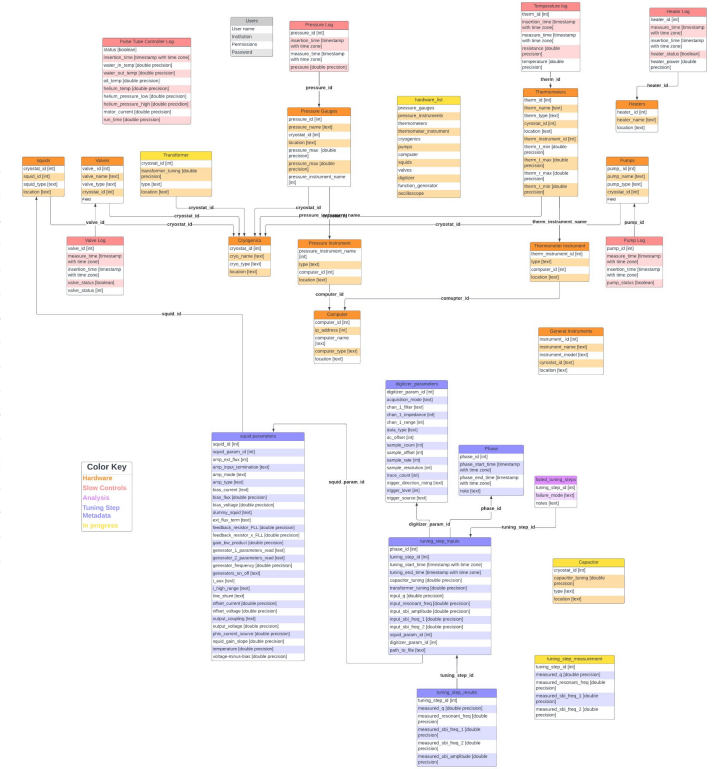
- PSQL database: for metadata storage & slow monitoring

Two-way Communication

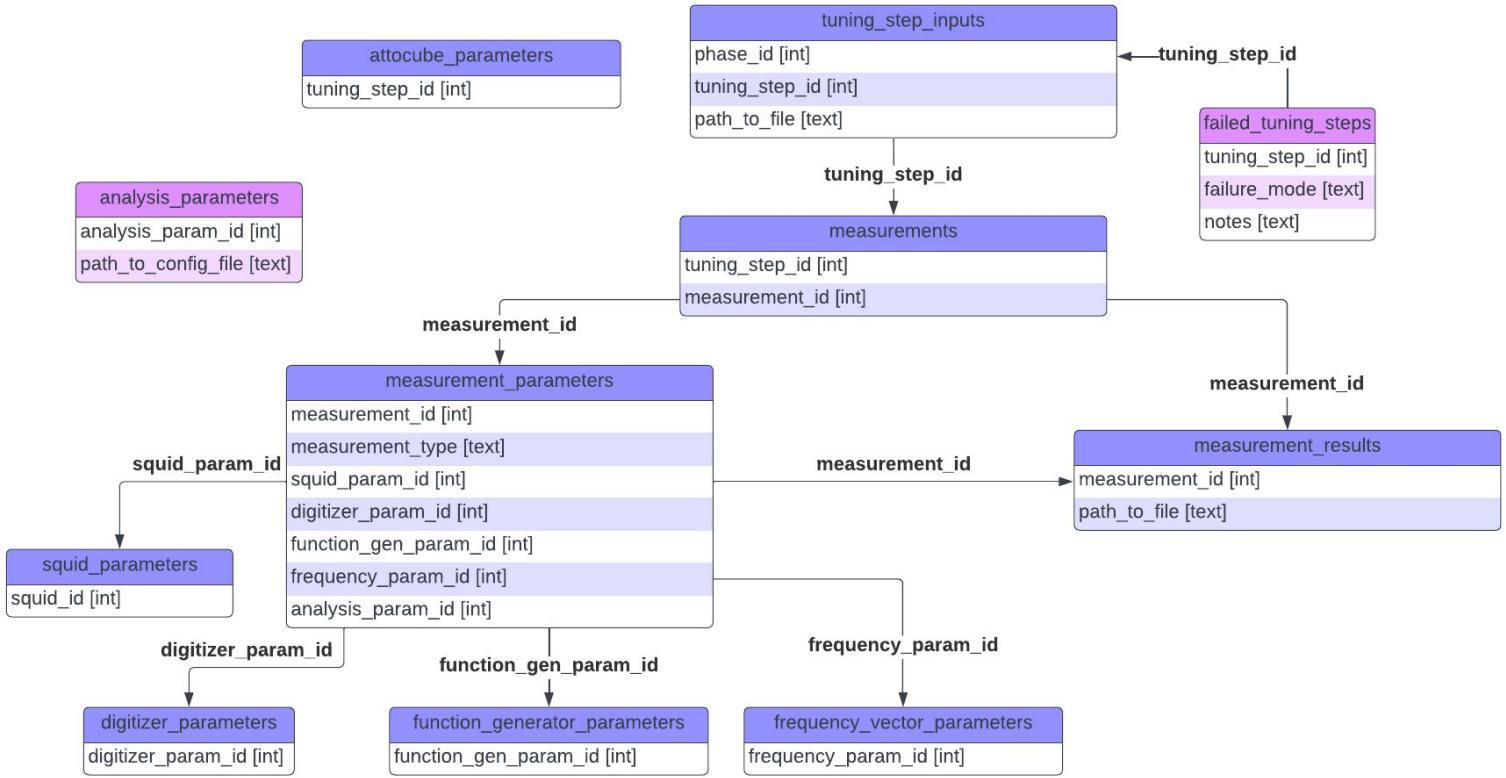
- Direct connections to DAQ computer & sequencer → not encouraged
- Remote controls website hosted on Stanford web pages

PSQL database tracks four types of data

- Again: no actual science data is being saved in the database
- Grafana dashboard interfaces with slow controls & hardware tables to allow slow monitoring



Zooming in on the tuning step tables…

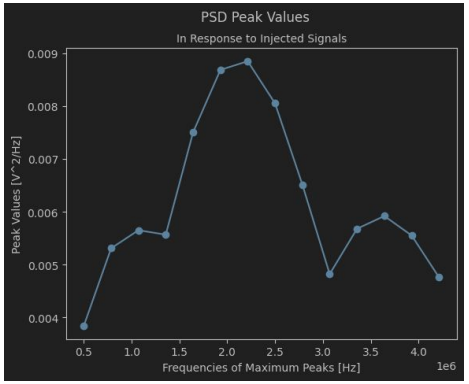


Collaborators will have access to Sherlock & Oak

- Oak: Stanford's high-performance computing data storage service
- Sherlock: Stanford's HPC cluster
  - Collaborators will have access to Sherlock for analysis & computing needs
  - Sherlock & Oak are integrated for easy data access

Open Questions & Timelines:

- Several open questions:
  - SQUID tuning procedure: take full IV curve once per cooldown, otherwise adjust at start of sequencer run?
  - Is there any hardware we are missing?
  - How to make database flexible enough so we don't have to constantly write in new tables?
- First fully automated studies have been run on Celeste's calibration loop
- Sequencer development continues, with plans for full deployment by late September



Thank you!