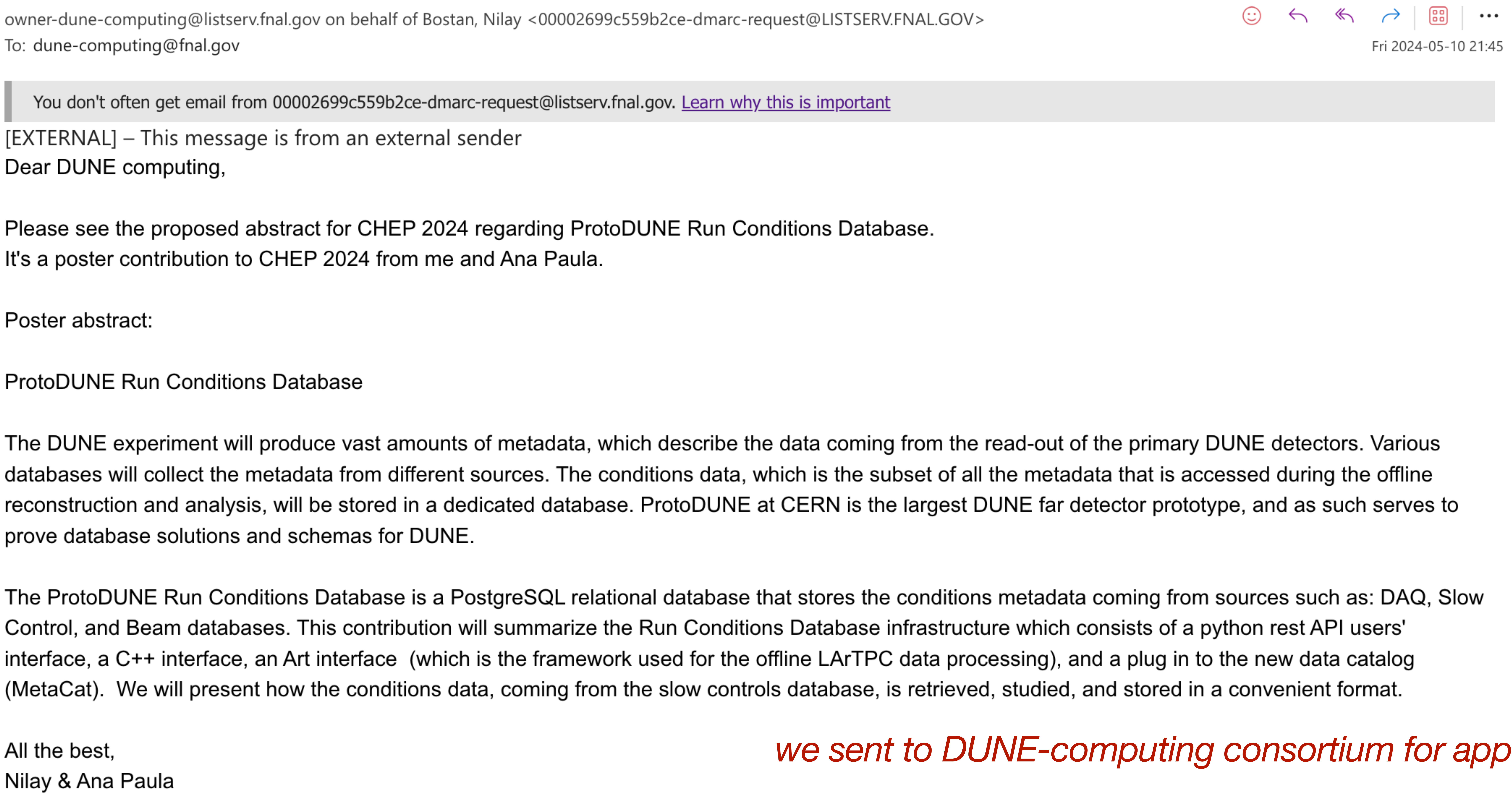
Update on ProtoDUNE DB

ProtoDUNE HD
Slow Controls DB & UConDB

Nilay Bostan, 05/14/2024

Abstract proposal for CHEP2024



owner-dune-computing@listserv.fnal.gov on behalf of Bostan, Nilay <00002699c559b2ce-dmarc-request@LISTSERV.FNAL.GOV>
To: dune-computing@fnal.gov

Fri 2024-05-10 21:45

You don't often get email from 00002699c559b2ce-dmarc-request@listserv.fnal.gov. Learn why this is important

[EXTERNAL] – This message is from an external sender

Dear DUNE computing,

Please see the proposed abstract for CHEP 2024 regarding ProtoDUNE Run Conditions Database.
It's a poster contribution to CHEP 2024 from me and Ana Paula.

Poster abstract:

ProtoDUNE Run Conditions Database

The DUNE experiment will produce vast amounts of metadata, which describe the data coming from the read-out of the primary DUNE detectors. Various databases will collect the metadata from different sources. The conditions data, which is the subset of all the metadata that is accessed during the offline reconstruction and analysis, will be stored in a dedicated database. ProtoDUNE at CERN is the largest DUNE far detector prototype, and as such serves to prove database solutions and schemas for DUNE.

The ProtoDUNE Run Conditions Database is a PostgreSQL relational database that stores the conditions metadata coming from sources such as: DAQ, Slow Control, and Beam databases. This contribution will summarize the Run Conditions Database infrastructure which consists of a python rest API users' interface, a C++ interface, an Art interface (which is the framework used for the offline LArTPC data processing), and a plug in to the new data catalog (MetaCat). We will present how the conditions data, coming from the slow controls database, is retrieved, studied, and stored in a convenient format.

All the best,
Nilay & Ana Paula

we sent to DUNE-computing consortium for approval

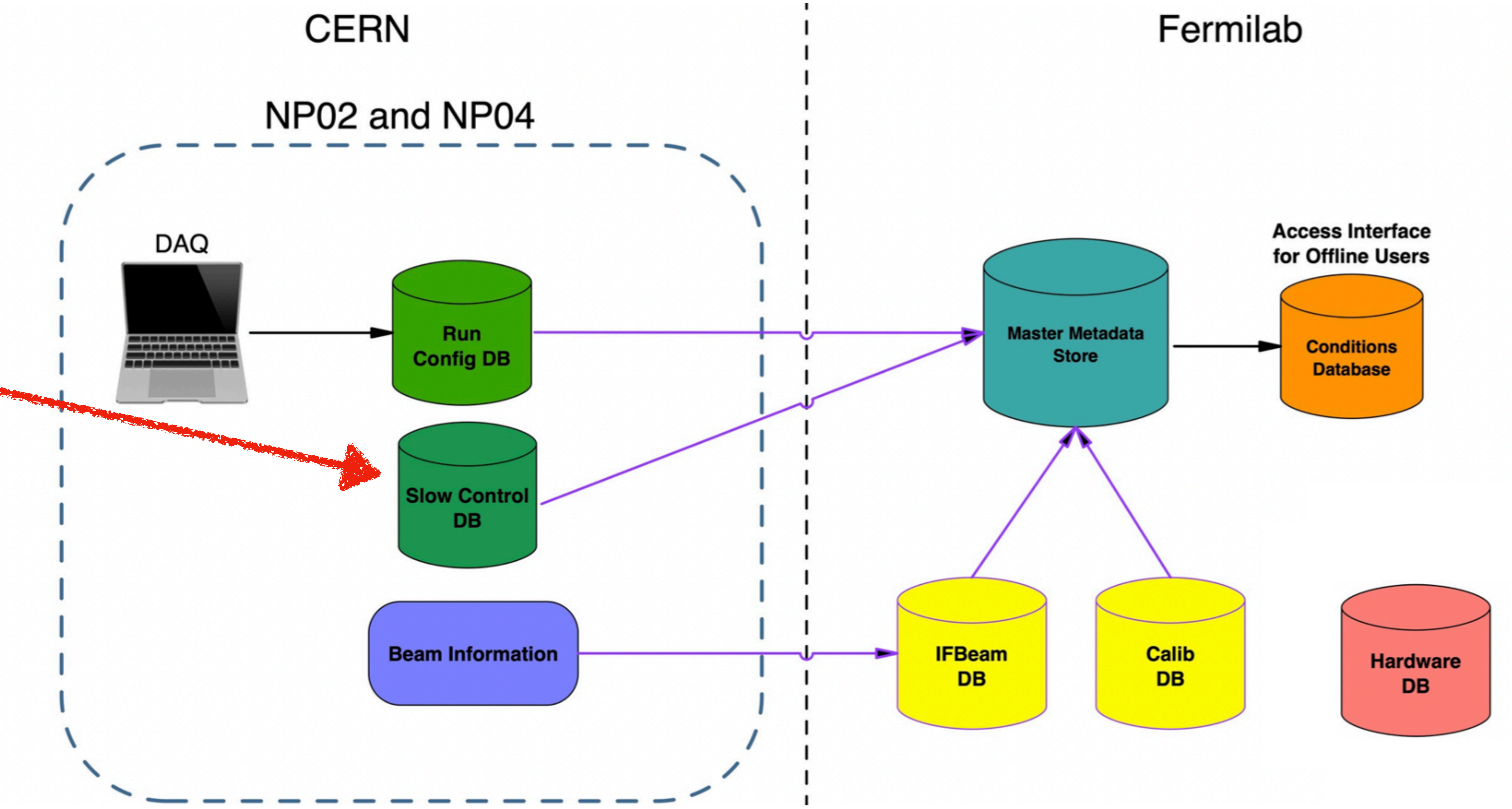
ProtoDUNE database system

- DUNE will produce vast amounts of metadata, which describe the data coming from the read-out of the primary DUNE detectors.

- **An unstructured database (UConDB)** holds the master store of metadata which collects info from all databases, allowing for maximal flexibility. **UConDB object is a BLOB*** - document (PDF, JSON, XML, CSV, text, FHICL, HDF5, …) - image - anything - The database is unaware of BLOB's internal structure, **UConDB records (independent) timelines for multiple objects, identified by name.**

https://ucondb.readthedocs.io/en/latest/

Slow Control DB:
-> Device values stored at high rate giving O(10GB)/day
-> Subset will be moved to UconDB, we need better understating of sensorlist



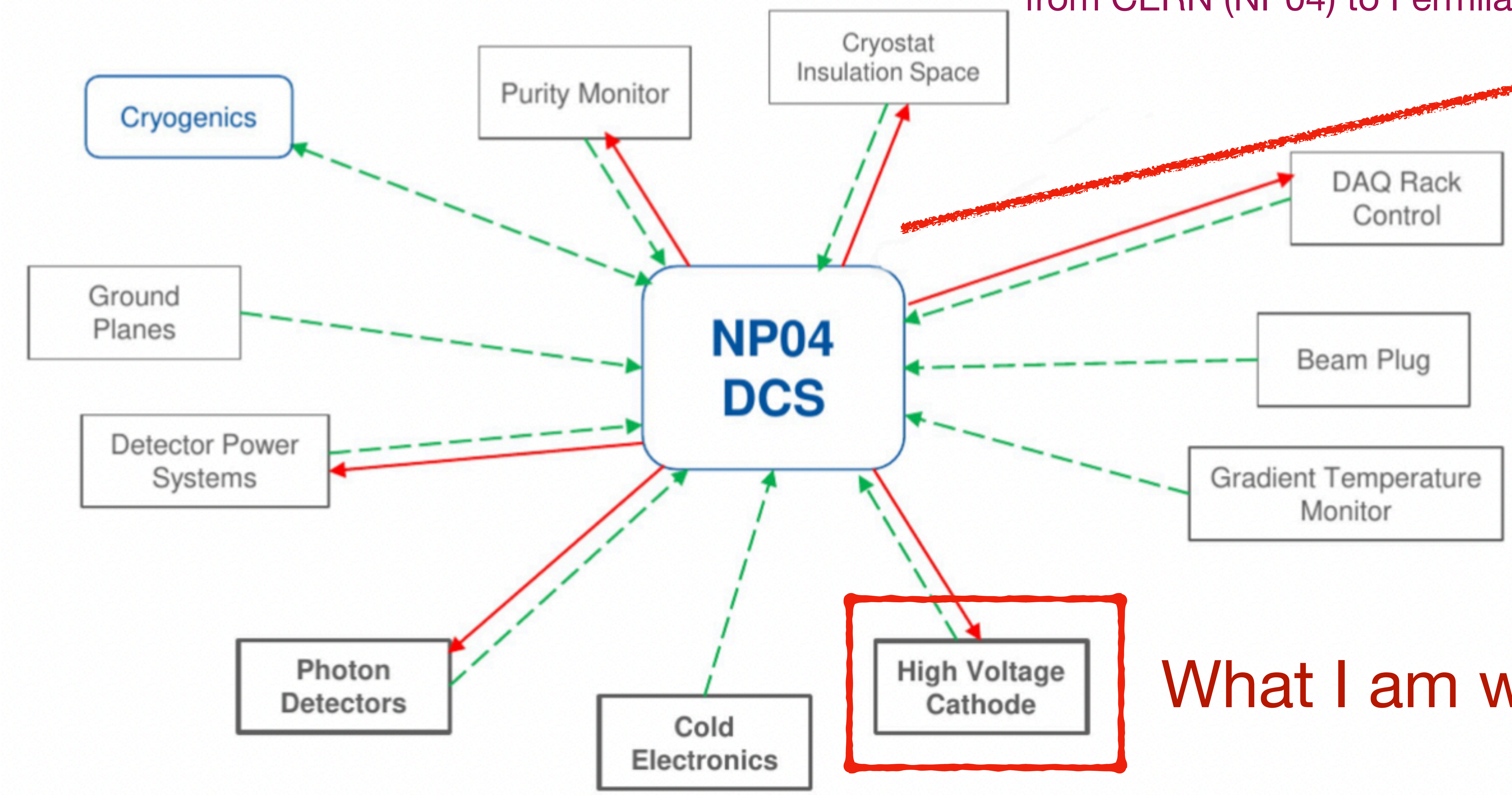
* Object storage is a computer data storage approach that manages data as "blobs" or "objects". Each object is typically associated with a variable amount of metadata, and a globally unique identifier.

Slow Controls

- Working on ProtoDUNE conditions database, investigate data from Slow Controls (Detector Control System aka ‘DCS’)
  E.g., LAr temp. & purity, high-voltage, ground impedance

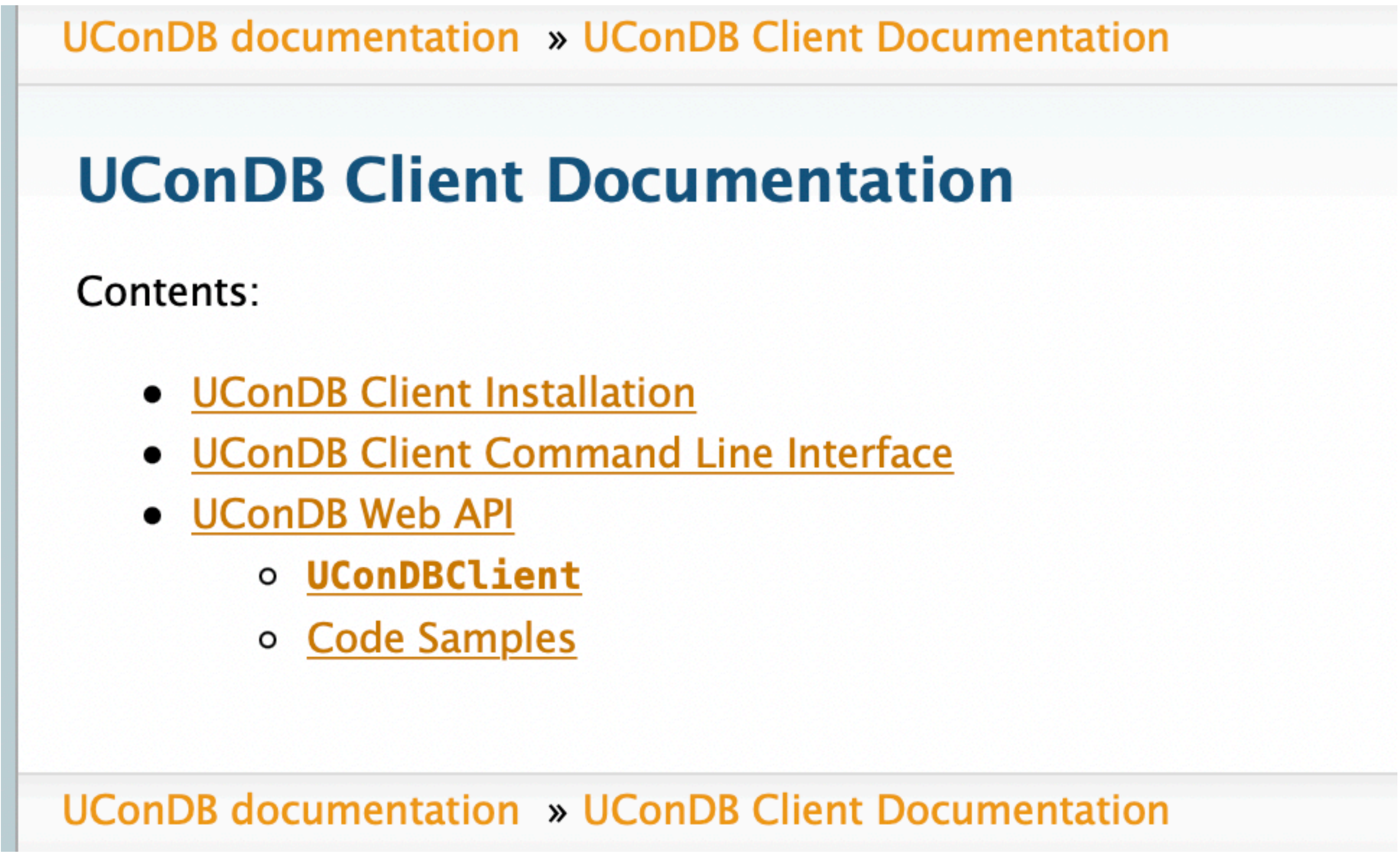
- Indexed by time stamp & stored in SC archive (‘DCS-DB’)

We need to send the data
from CERN (NP04) to Fermilab (UConDB)



What I am working on

From Manuel J. Rodriguez



UConDB documentation » UConDB Client Documentation

UConDB Client Documentation

Contents:

- UConDB Client Installation
- UConDB Client Command Line Interface
- UConDB Web API
  - UConDBClient
  - Code Samples

UConDB documentation » UConDB Client Documentation

What is my responsibility on this project for now?

- Getting familiarized with the slow control data -> ✓
- Extract the data from database in different timestamps, convert unix time to date time, and then plot -> ✓
- After the data extraction from NP04, sending the data to UconDB and store data there -> ✓

-> In UConDB there is a test folder, and a test object, that I just sent the DB in lxplus to test UConDB in dunegpvm. I tried this with admin password and it was successful -> ✓

(But in reality, I need to send to the protodune_conditions in UConDB, and the object: slow_controls, actually I have to create this for the first time)

- I combined all of these above in a one code from scratch. -> ✓
- After the new comprehensive code, I will transfer **all** necessary data from DCS-DB to Masterstore (from CERN to Fermilab) automatically. -> ✓

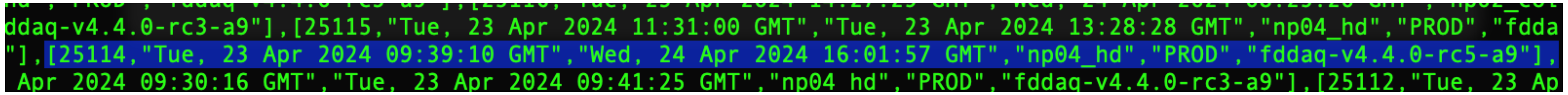
-> Once that its working correctly we should merge my code with the cron jobs that Ana Paula has set up, so that every time that there is a new run my code is run and the HV is uploaded. But we don't need to redo the cron jobs, just add to them -> NOT YET

- What I just created the script to get the HV data in different time periods, then plotted, I just get the mean and std of that values -> ✓

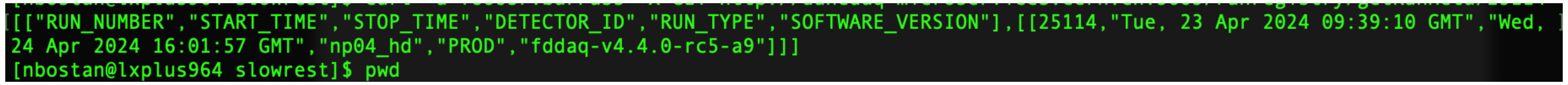
-> Then, after creating the script to get mean and std, we need to send the HV mean and std to the conditions db (ConDB) -> NOT YET

Visualization

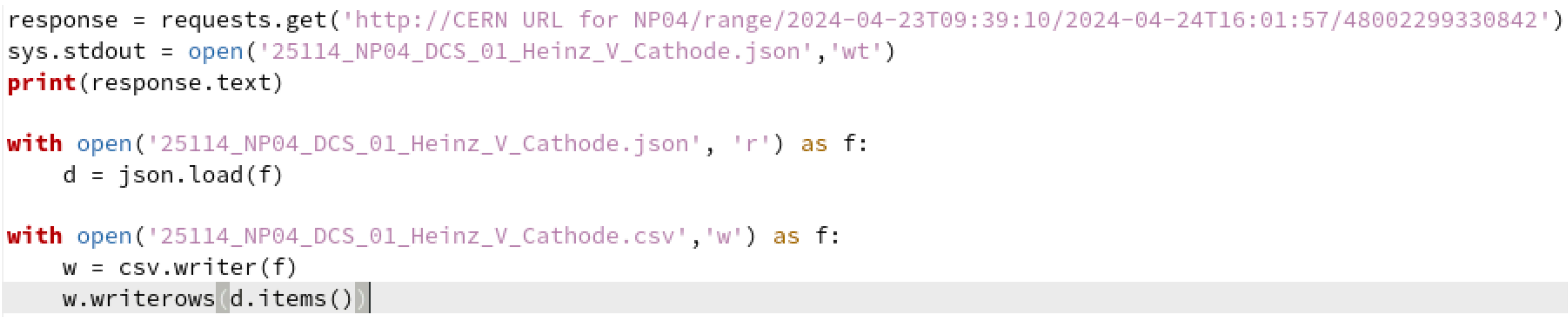
- Extract the data from database in different timestamps, convert unix time to date time, and then plot -> ✓



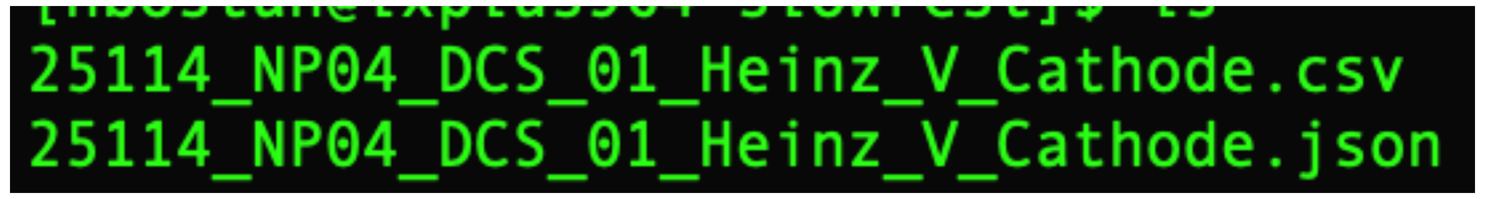
```
ddaq-v4.4.0-rc3-a9"],[25115,"Tue, 23 Apr 2024 11:31:00 GMT","Tue, 23 Apr 2024 13:28:28 GMT","np04_hd","PROD","fdda
"],[25114,"Tue, 23 Apr 2024 09:39:10 GMT","Wed, 24 Apr 2024 16:01:57 GMT","np04_hd","PROD","fddaq-v4.4.0-rc5-a9"],
 Apr 2024 09:30:16 GMT","Tue, 23 Apr 2024 09:41:25 GMT","np04_hd","PROD","fddaq-v4.4.0-rc3-a9"],[25112,"Tue, 23 Ap
```



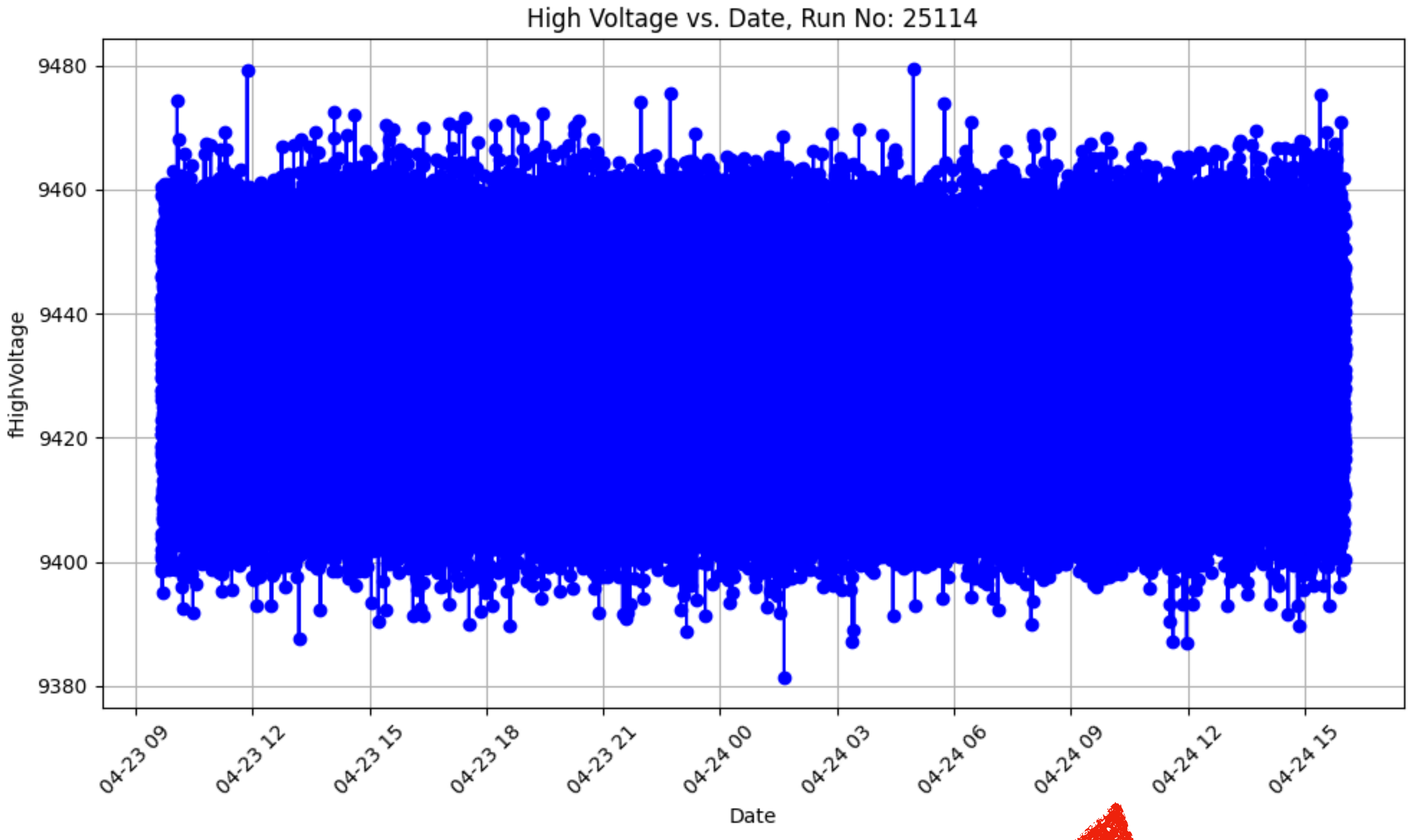
```
[[["RUN_NUMBER","START_TIME","STOP_TIME","DETECTOR_ID","RUN_TYPE","SOFTWARE_VERSION"],[[25114,"Tue, 23 Apr 2024 09:39:10 GMT","Wed,
 24 Apr 2024 16:01:57 GMT","np04_hd","PROD","fddaq-v4.4.0-rc5-a9"]]]
[nbostan@lxplus964 slowrest]$ pwd
```



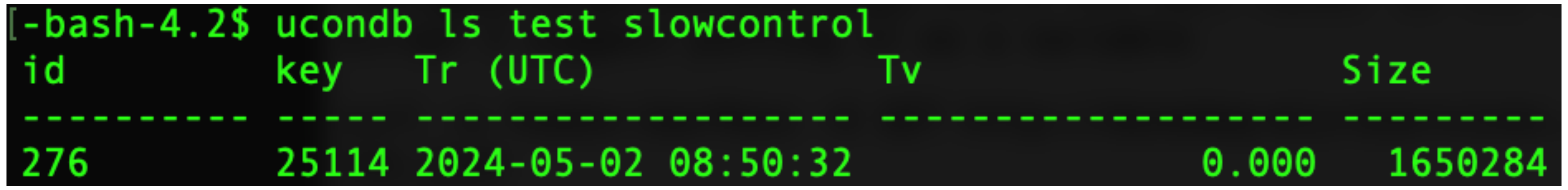
```
response = requests.get('http://CERN URL for NP04/range/2024-04-23T09:39:10/2024-04-24T16:01:57/48002299330842')
sys.stdout = open('25114_NP04_DCS_01_Heinz_V_Cathode.json','wt')
print(response.text)

with open('25114_NP04_DCS_01_Heinz_V_Cathode.json', 'r') as f:
    d = json.load(f)

with open('25114_NP04_DCS_01_Heinz_V_Cathode.csv','w') as f:
    w = csv.writer(f)
    w.writerows(d.items())
```



```
25114_NP04_DCS_01_Heinz_V_Cathode.csv
25114_NP04_DCS_01_Heinz_V_Cathode.json
```

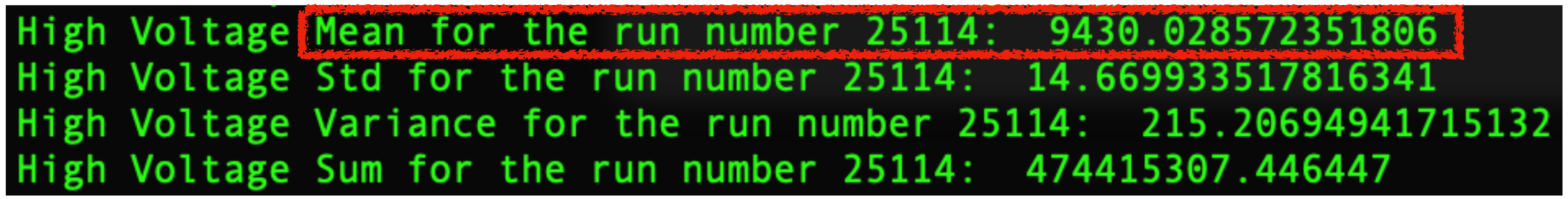


- After the data extraction from NP04, sending the data to UconDB and store data there -> ✓



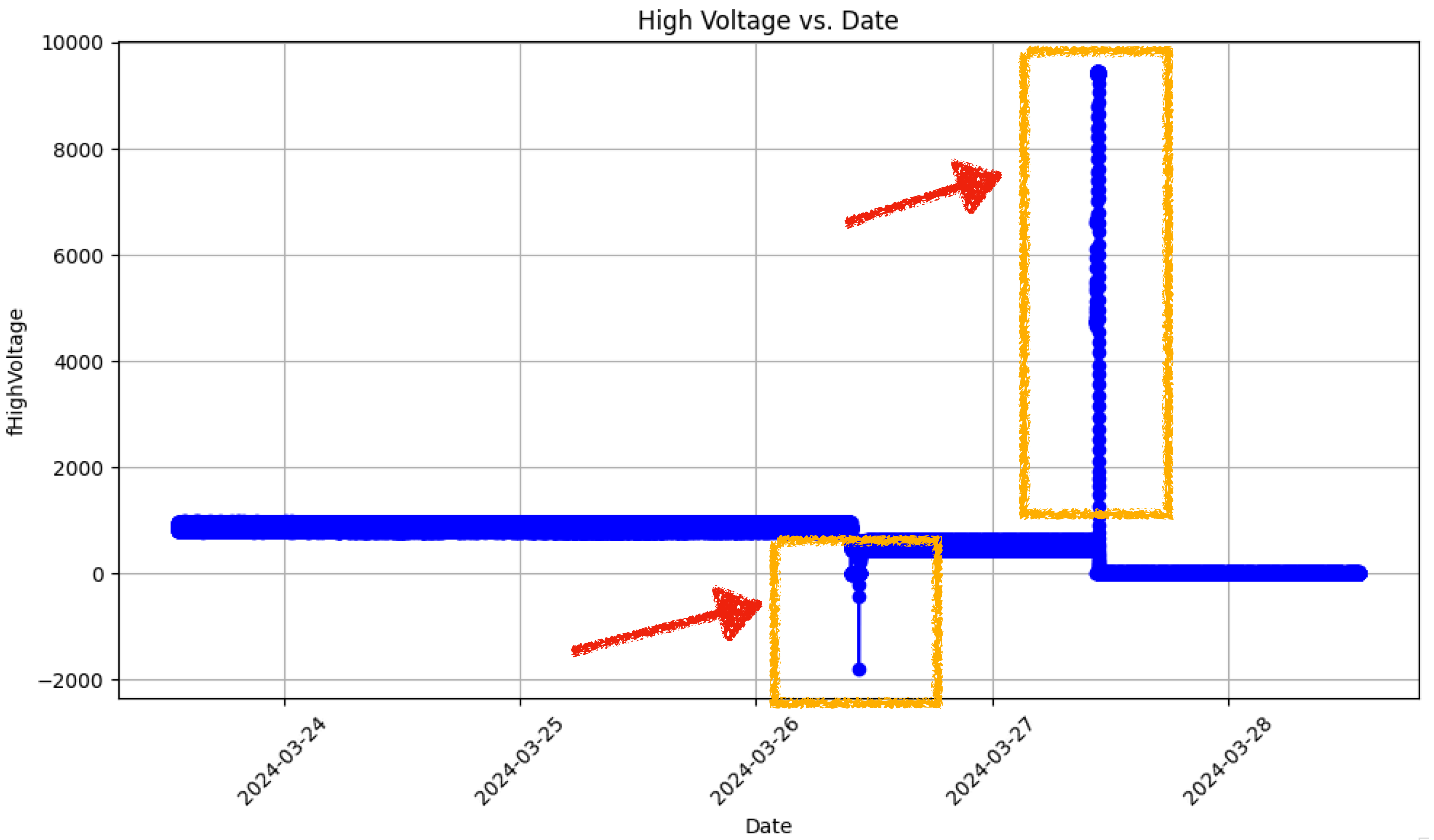
```
-bash-4.2$ ucondb ls test slowcontrol
id          key   Tr (UTC)              Tv                    Size
----------  ----- -------------------   -------------------   ---------
276         25114 2024-05-02 08:50:32                  0.000    1650284
```

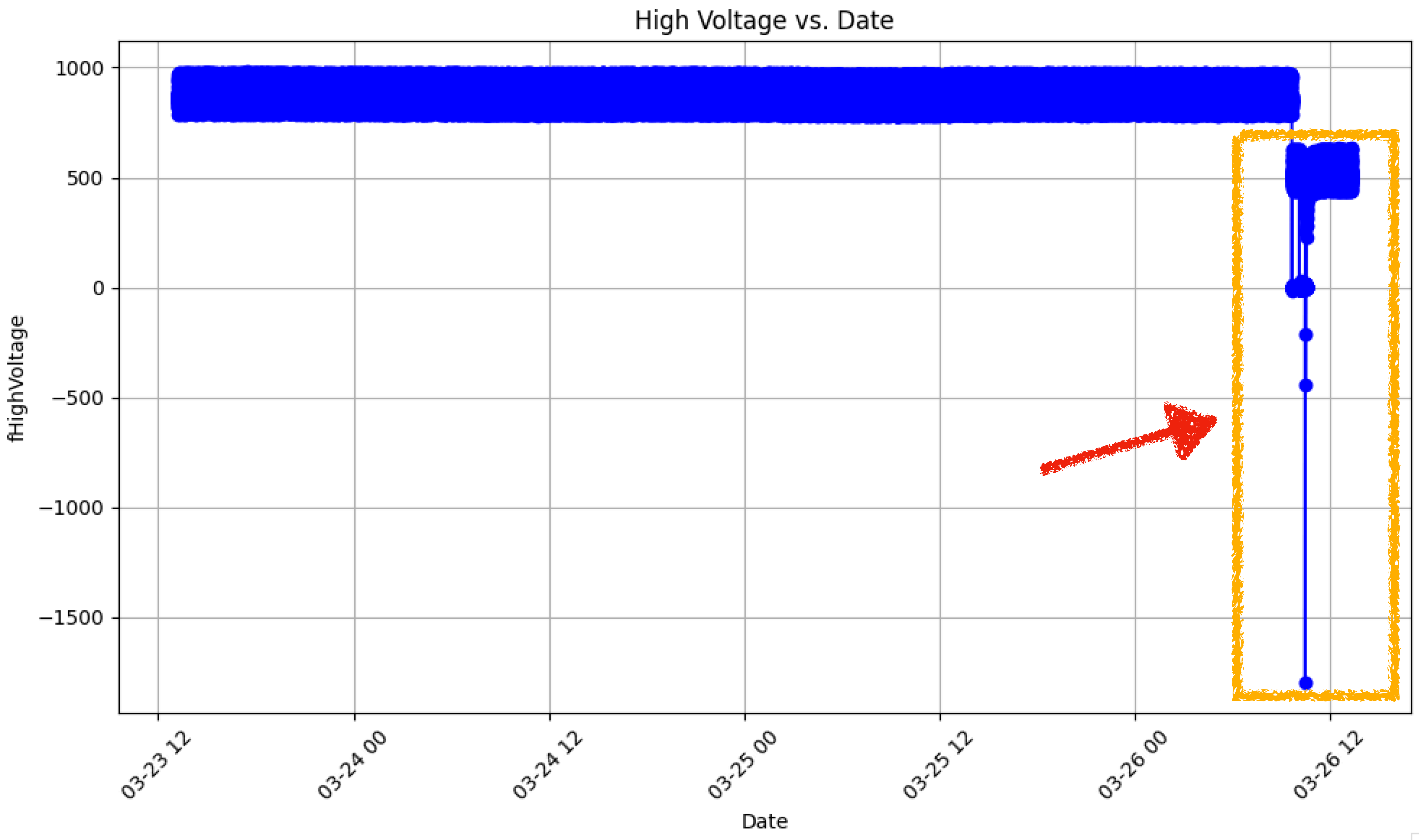
- I just get the mean and std of the HV for specific run numbers: -> ✓



```
High Voltage Mean for the run number 25114:  9430.028572351806
High Voltage Std for the run number 25114:  14.669933517816341
High Voltage Variance for the run number 25114:  215.20694941715132
High Voltage Sum for the run number 25114:  474415307.446447
```

Some weird periods in the HV that I noticed





I talked to some ProtoDUNE detector operations expert and they mentioned that

-> The identified periods where the HV is changing quickly (basically the vertical lines in the plot) and we can mark them as 'unstable' so that they can be excluded from further analysis.

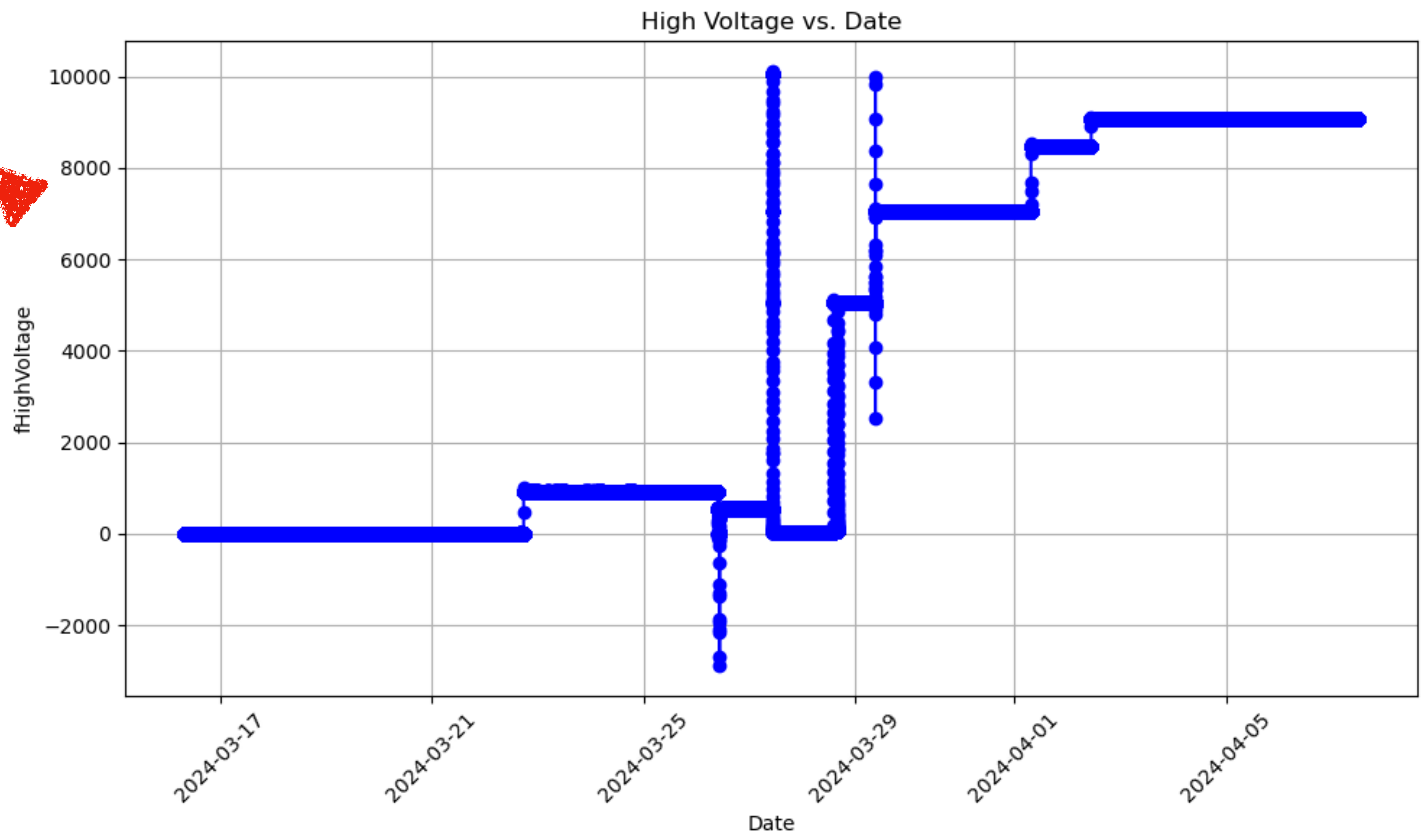
-> If there are negative values in HV, then the controls are disconnected from the power supply. The power supply delivers 2 analogue signals 0-10 V equivalent to the maximum range for Voltage and Current.

-> In operation HV must be always positive, if it goes in negative for short period is because perhaps, they changed the power supply unplugging and plugging the connectors.

Conclusion

So far, so good !...

- For the histogram I made for the ProtoDUNE HV data:

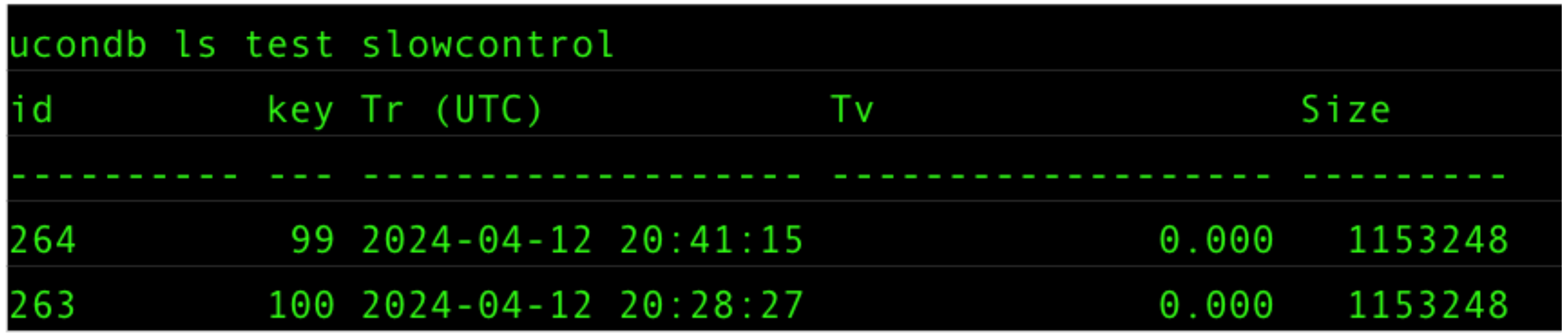


Filippo Resnati @CERN:

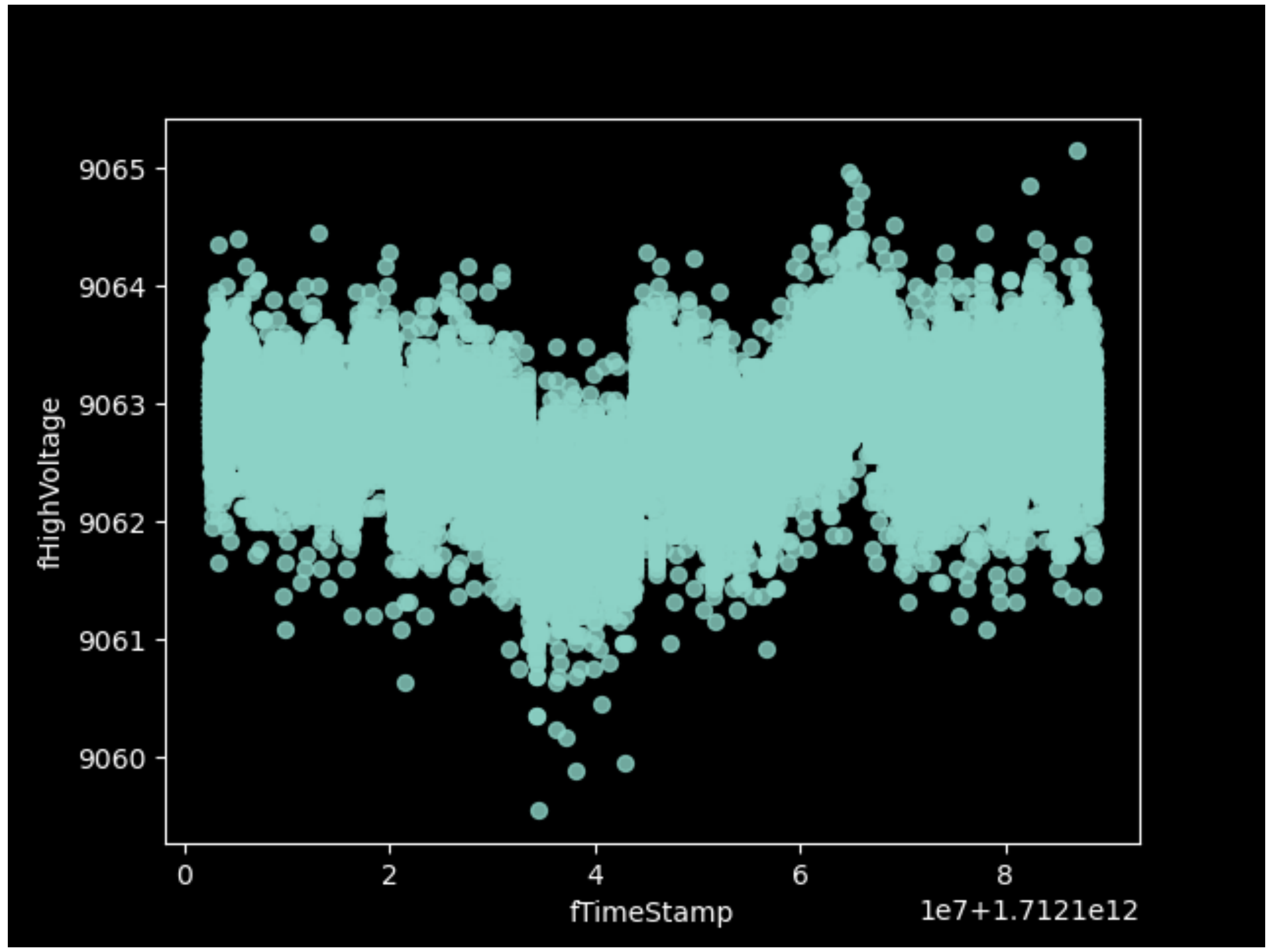
"Yes, the plot makes sense. The HV on the cathode is slightly on."

For now:

- I can extract HV data from the database in between different timestamps
- I can save the HV data to a CSV file from JSON
- I can send the ProtoDUNE data to UConDB: from CERN machines (lxplus) to Fermilab machines (dunegpvm)
- I tried to send one of the CSV files which includes HV data in between different timestamps to UConDB in dunegpvm
- To the test folder in UConDB, I was able to send the data CSV file successfully
- Getting the mean and std for specific runs



```
ucondb ls test slowcontrol
id          key Tr (UTC)                 Tv                   Size
---------- --- ------------------- ------------------- ----------
264          99 2024-04-12 20:41:15                0.000    1153248
263         100 2024-04-12 20:28:27                0.000    1153248
```



Slow Controls Data @CERN

SCADA system writes raw data to DCS-DB (Oracle)

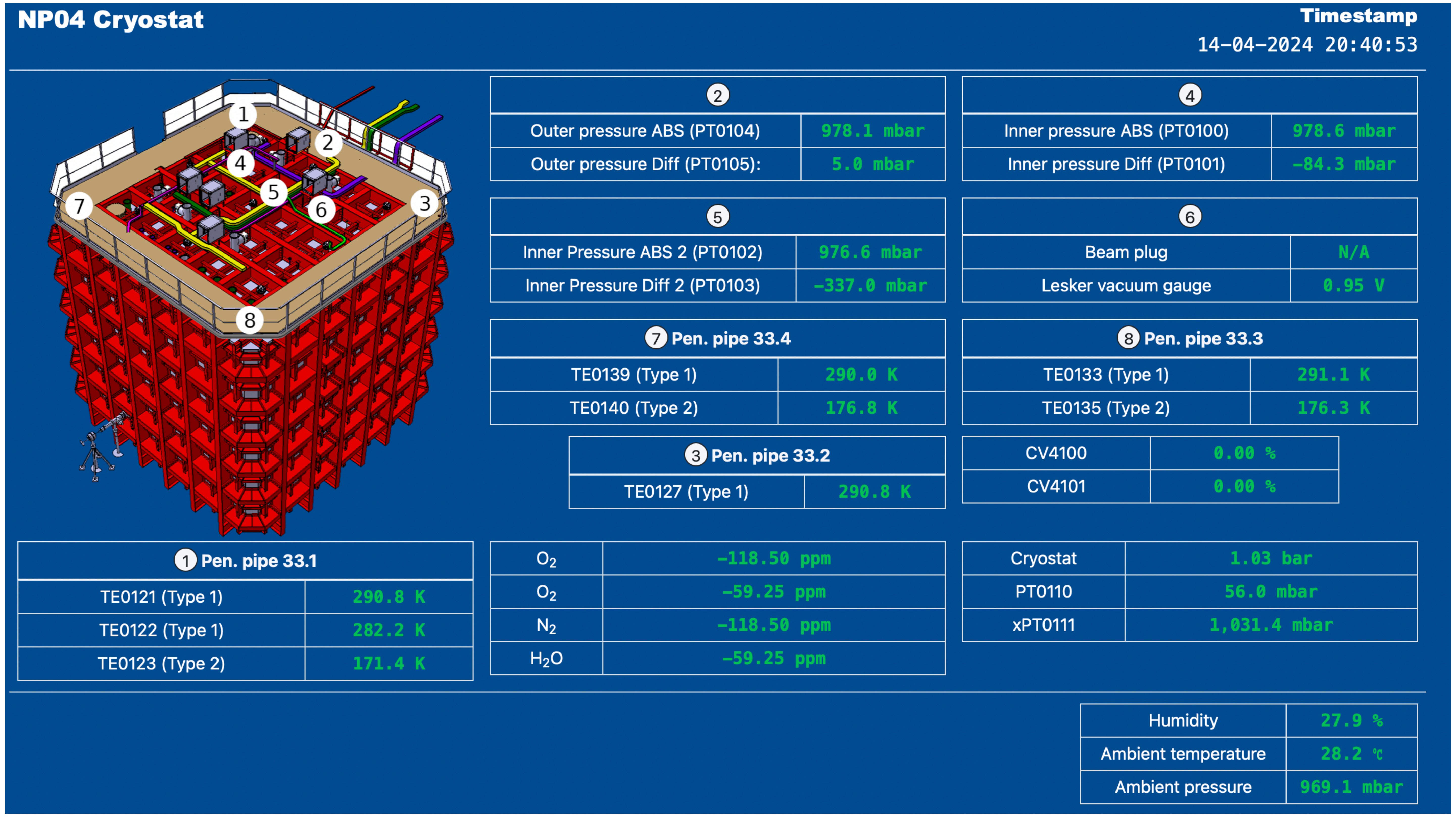
Detector experts access that data via a website

- https://np04-slow-control.web.cern.ch/

DCS-DB is accessed via read-only rest API

- https://gitlab.cern.ch/ep-dt-di/dcsdb-rest

The histogram from temperature map
I extracted for 04-14-2024
and different timestamps:
TE0121 (Type 1) 290.7 K:



NP04 Cryostat

Timestamp
14-04-2024 20:40:53

2	
Outer pressure ABS (PT0104)	978.1 mbar
Outer pressure Diff (PT0105):	5.0 mbar

4	
Inner pressure ABS (PT0100)	978.6 mbar
Inner pressure Diff (PT0101)	-84.3 mbar

5	
Inner Pressure ABS 2 (PT0102)	976.6 mbar
Inner Pressure Diff 2 (PT0103)	-337.0 mbar

6	
Beam plug	N/A
Lesker vacuum gauge	0.95 V

7 Pen. pipe 33.4	
TE0139 (Type 1)	290.0 K
TE0140 (Type 2)	176.8 K

8 Pen. pipe 33.3	
TE0133 (Type 1)	291.1 K
TE0135 (Type 2)	176.3 K

3 Pen. pipe 33.2	
TE0127 (Type 1)	290.8 K

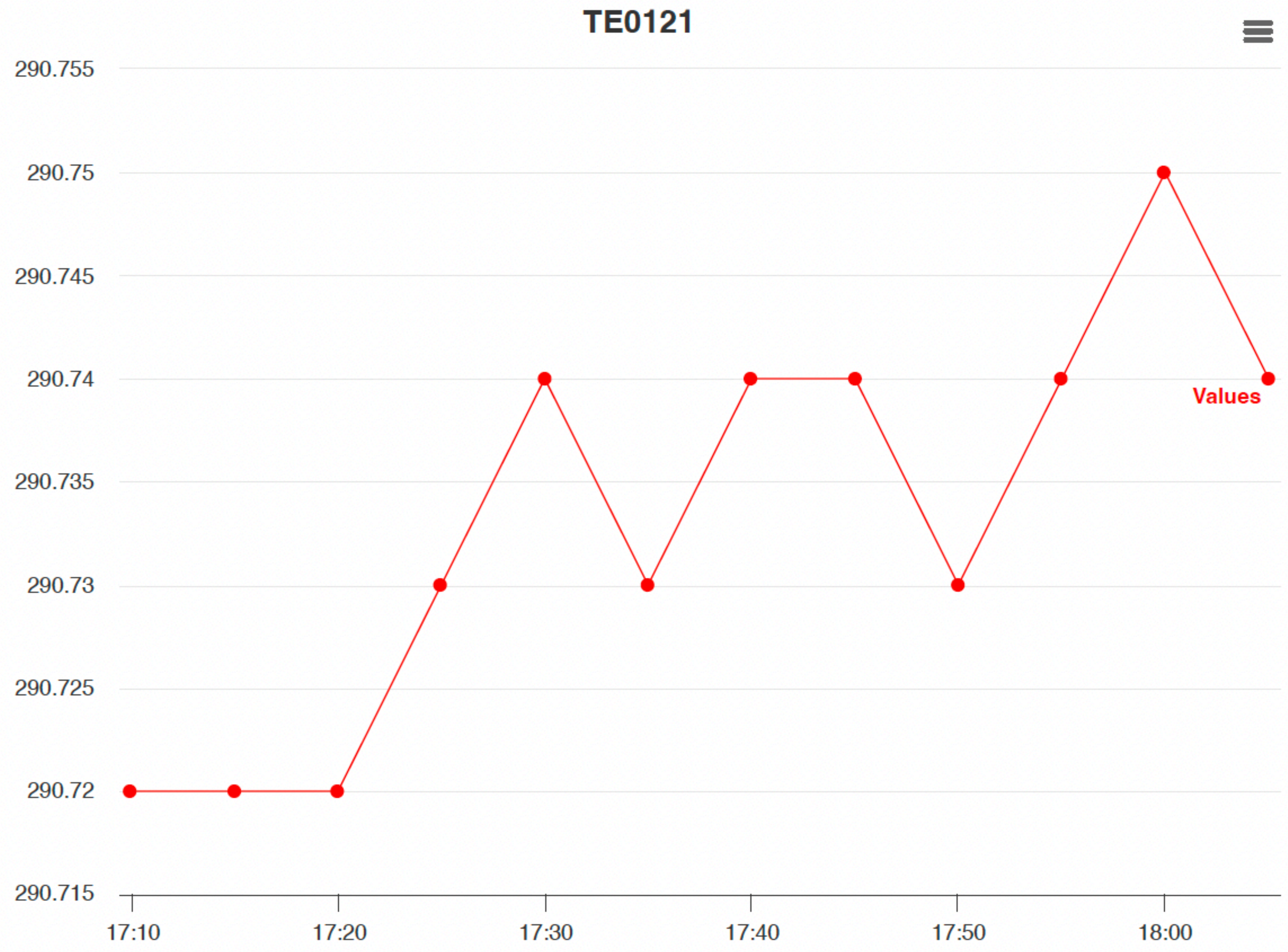
CV4100	0.00 %
CV4101	0.00 %

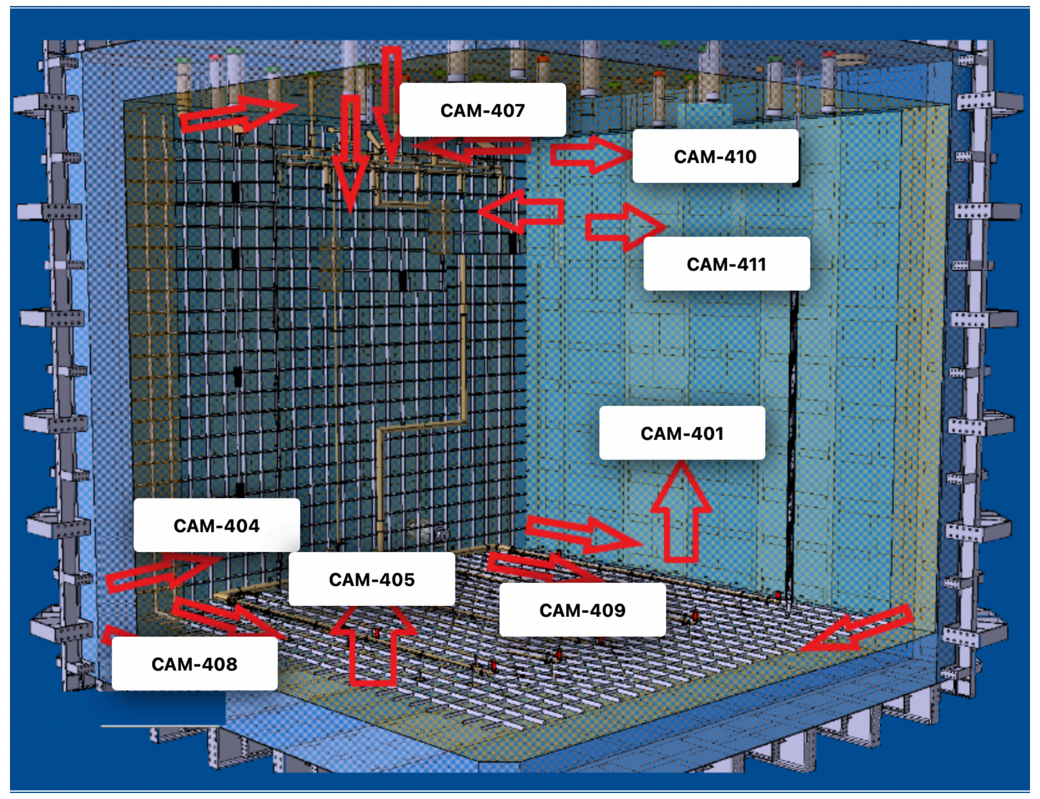
1 Pen. pipe 33.1	
TE0121 (Type 1)	290.8 K
TE0122 (Type 1)	282.2 K
TE0123 (Type 2)	171.4 K

O_2	-118.50 ppm
O_2	-59.25 ppm
N_2	-118.50 ppm
H_2O	-59.25 ppm

Cryostat	1.03 bar
PT0110	56.0 mbar
xPT0111	1,031.4 mbar

Humidity	27.9 %
Ambient temperature	28.2 ℃
Ambient pressure	969.1 mbar





SCADA: Supervisory Control and Data Acquisition

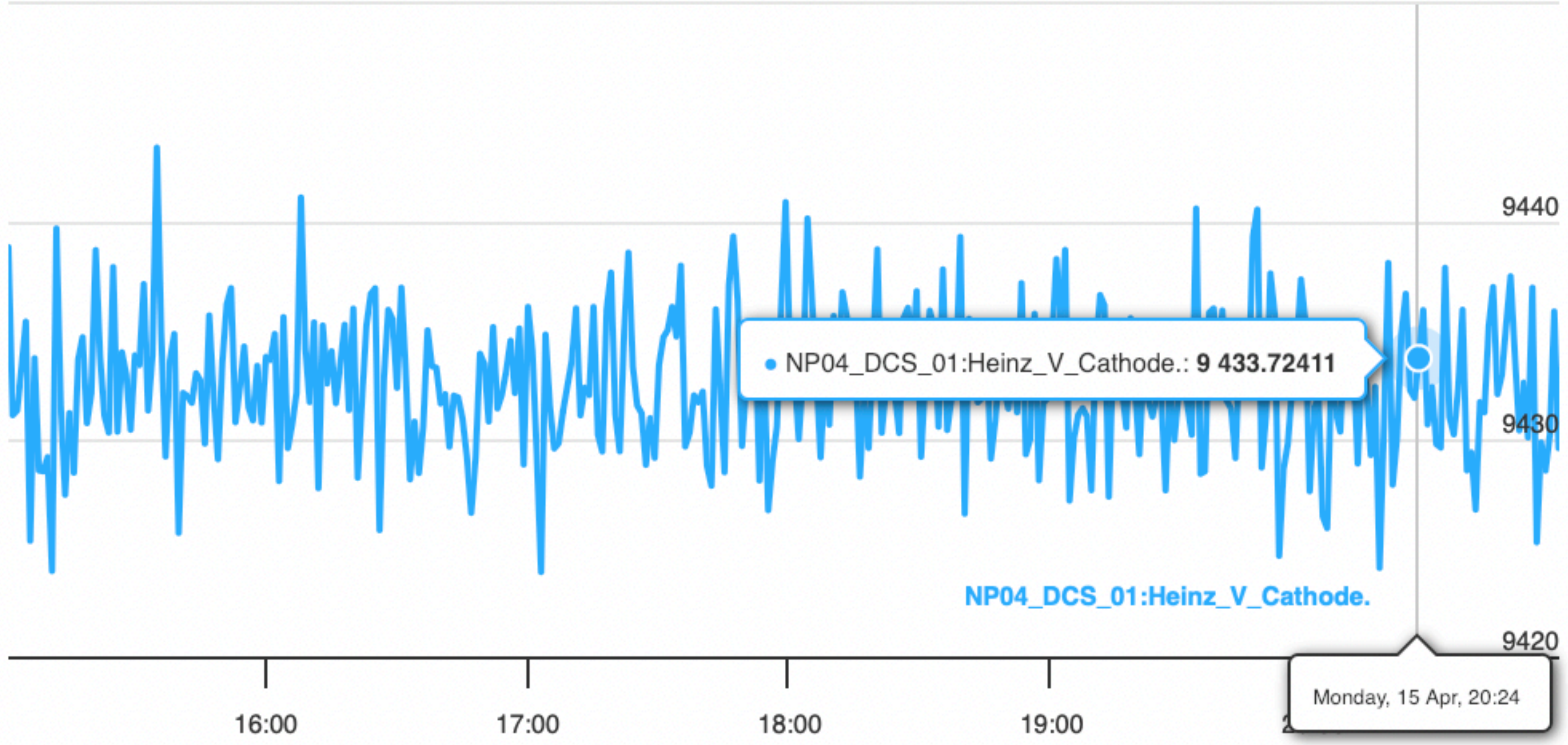
ProtoDUNE Slow Controls HD Data

This list aims to document what information from the slow controls archive is needed for offline data processing.
Horizontal Drift: A more comprehensive list of all metadata needed for offline reconstruction.

The following data is needed:
LAr purity (= electron lifetime)
LAr temperature HV (drift-inducing & to find unstable periods)
Laser System? (independent system, not yet connected to the DCS-DB)

Description	Sensor id:name(s) in DCS-DB	Needed granularity	Needed precision
LAr purity	"47928748016410" : "NP04_DCS_01:PrM0.PrM_corrected_e_lifetime", "47928764793626" : "NP04_DCS_01:PrM1.PrM_corrected_e_lifetime", "47928781570842" : "NP04_DCS_01:PrM2.PrM_corrected_e_lifetime"		
LAr temperature	Mean(s): "47931130380570" : "NP04_DCS_01:top_temp_mean.", "47931147157786" : "NP04_DCS_01:bottom_temp_mean." 48 individual sensors: "47890261082394" : "NP04_DCS_01:TE0001.", … "47891049611546" : "NP04_DCS_01:TE0048.",		
High Voltage	drift-inducing: "47894774153498" : "NP04_DCS_01:Heinz_V." to find unstable periods "47894757376282" : "NP04_DCS_01:Heinz_I.", "48005101125914" : "NP04_DCS_01:Heinz_I_Filtered.", "48002299330842" : "NP04_DCS_01:Heinz_V_Cathode.", "48001913454874" : "NP04_DCS_01:Heinz_V_Raw.", (red ones, Nilay added from sensor-dict in NP04) "48000437059866": "NP04_DCS_01:Heinz_Limit." "48002852978970": "NP04_DCS_01:Heinz_V_Cathode_97MOhms." "48001980563738": "NP04_DCS_01:Heinz_OnOff_Sts."		

For example, HV values of
the Heinz_V_Cathode for today


16:00
17:00
18:00
19:00
• NP04_DCS_01:Heinz_V_Cathode.: 9 433.72411
NP04_DCS_01:Heinz_V_Cathode.
Monday, 15 Apr, 20:24