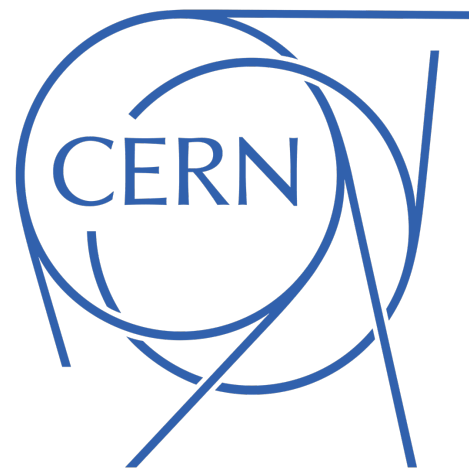


Ethan Hazelton

University of Michigan

06/01/2023

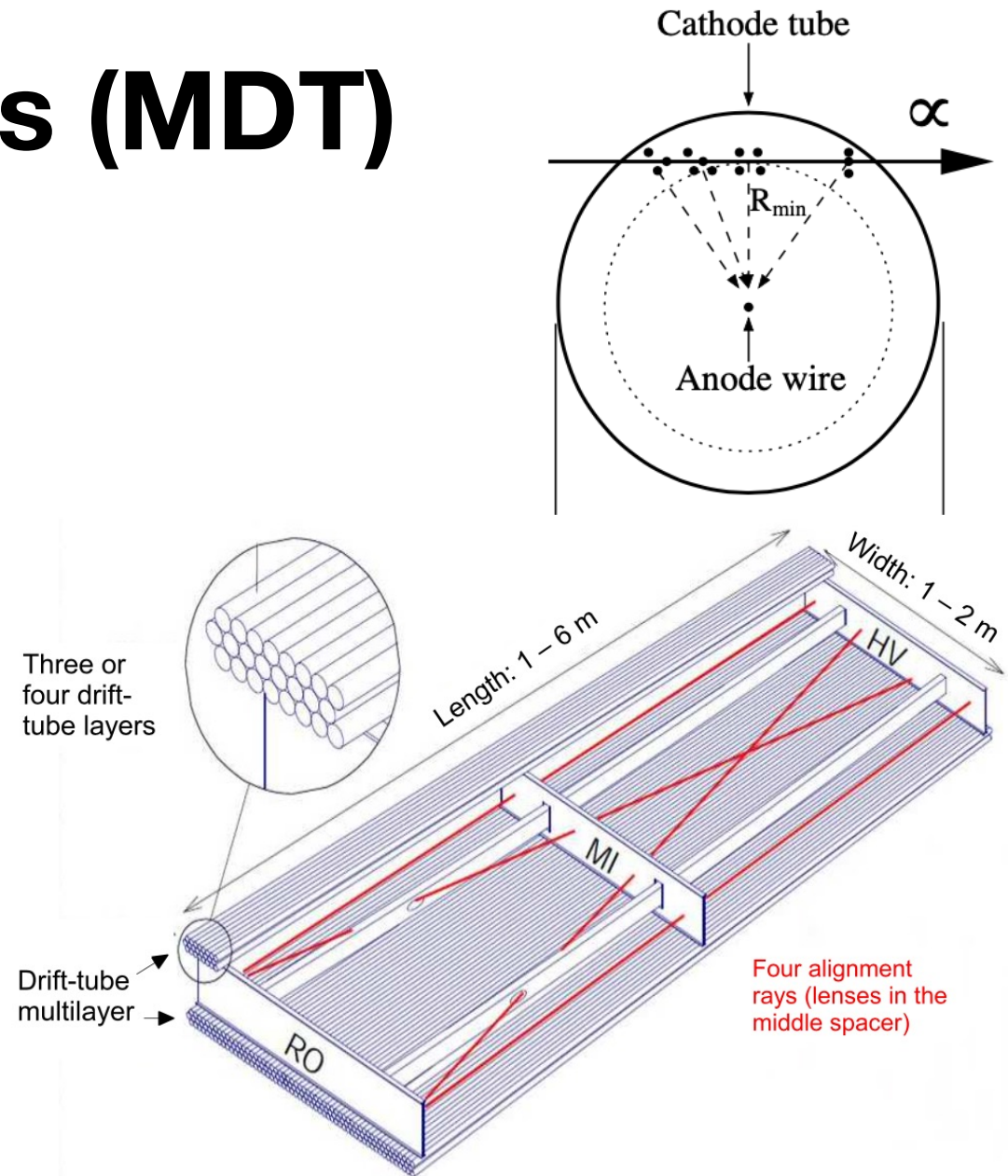
# MDT DCS Offline Monitoring



**ATLAS**  
EXPERIMENT

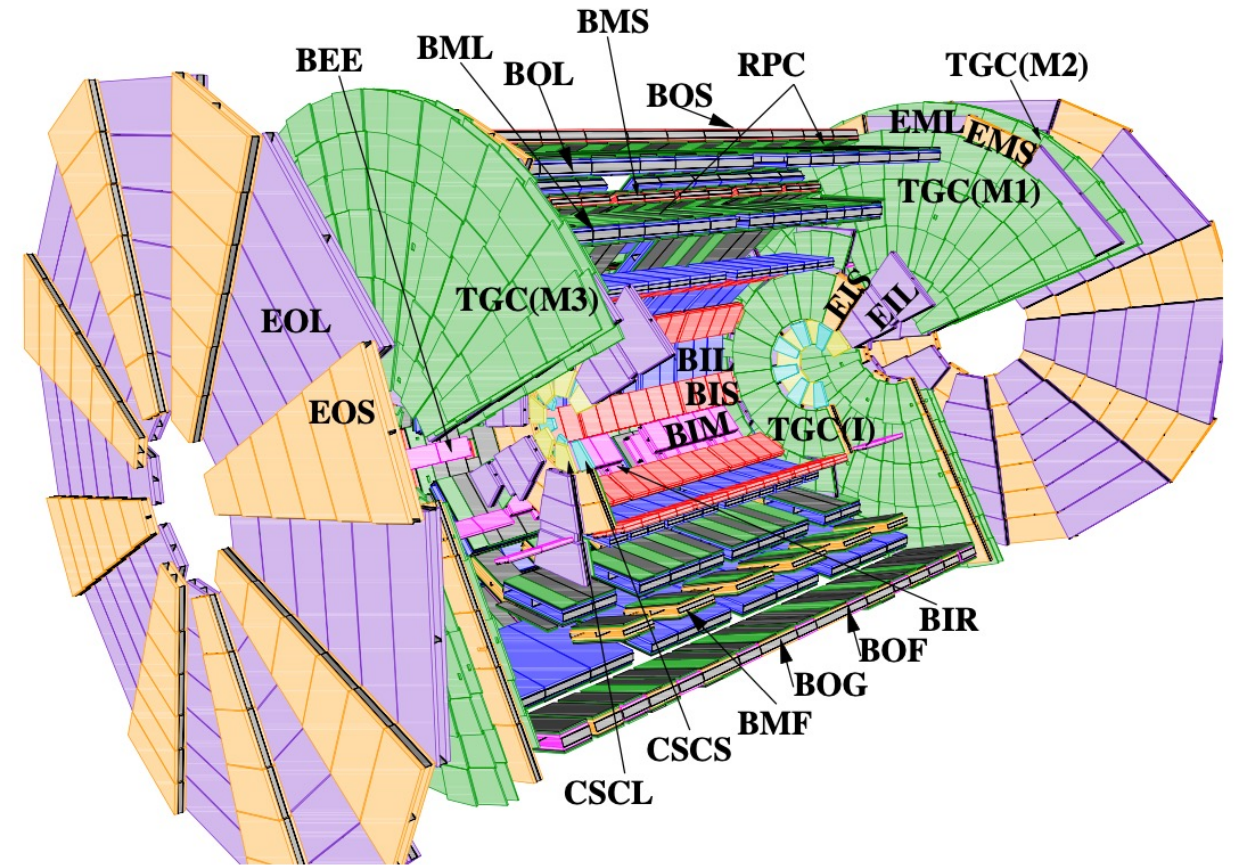
# Monitored Drift Tubes (MDT)

- For muon position and transverse momentum measurement
- Tubes contain Argon gas mixture
- Muons ionize the gas in each tube, causing free electrons to travel towards the anode wire
  - Relationship between electron drift time and distance to determine the position of the interacting muon



# MDT (cont.)

- Muons subject to  $B$  field causing their path to be curved with a radius of curvature  $r$
- Transverse momentum can be calculated from the muon track determined by the drift tubes:
  - $p_T = rqB$

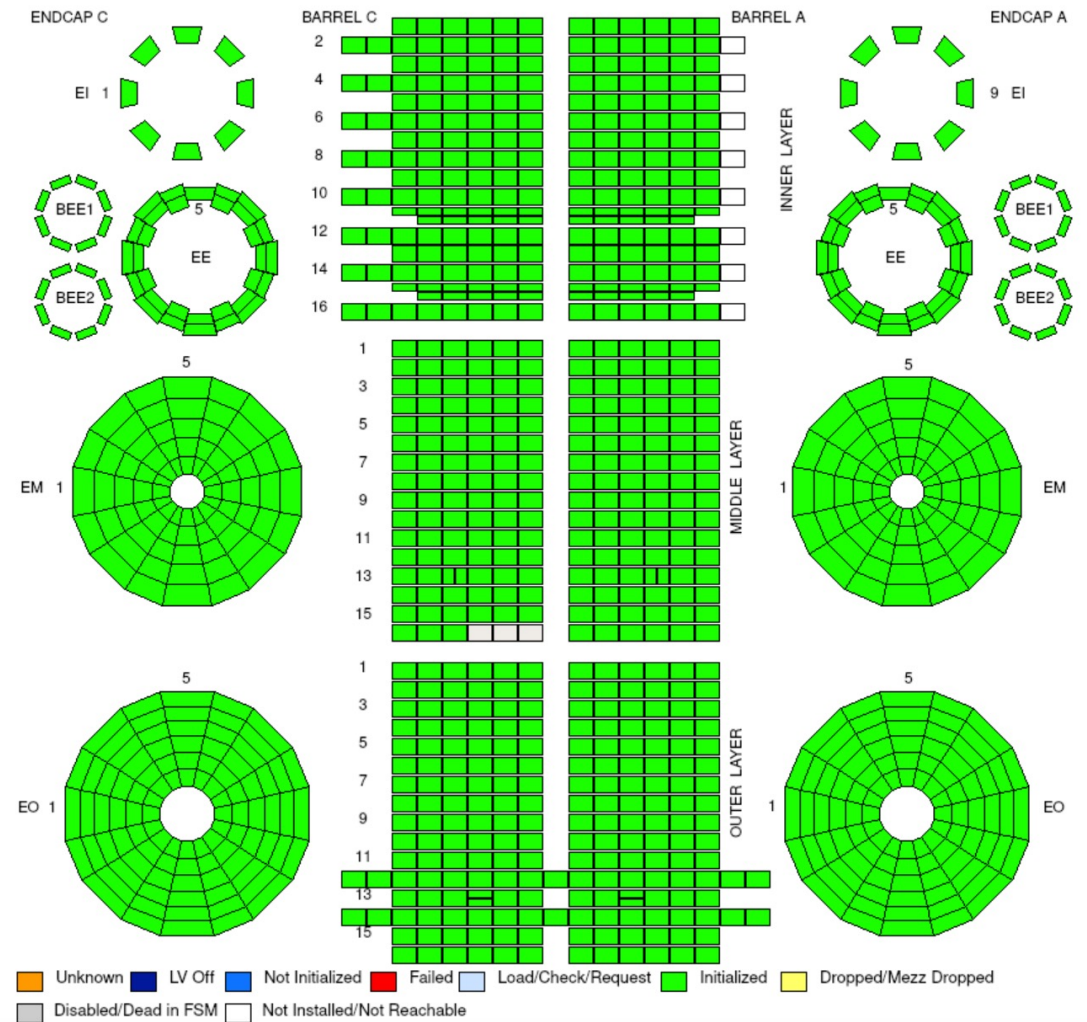




# DCS Online Monitoring

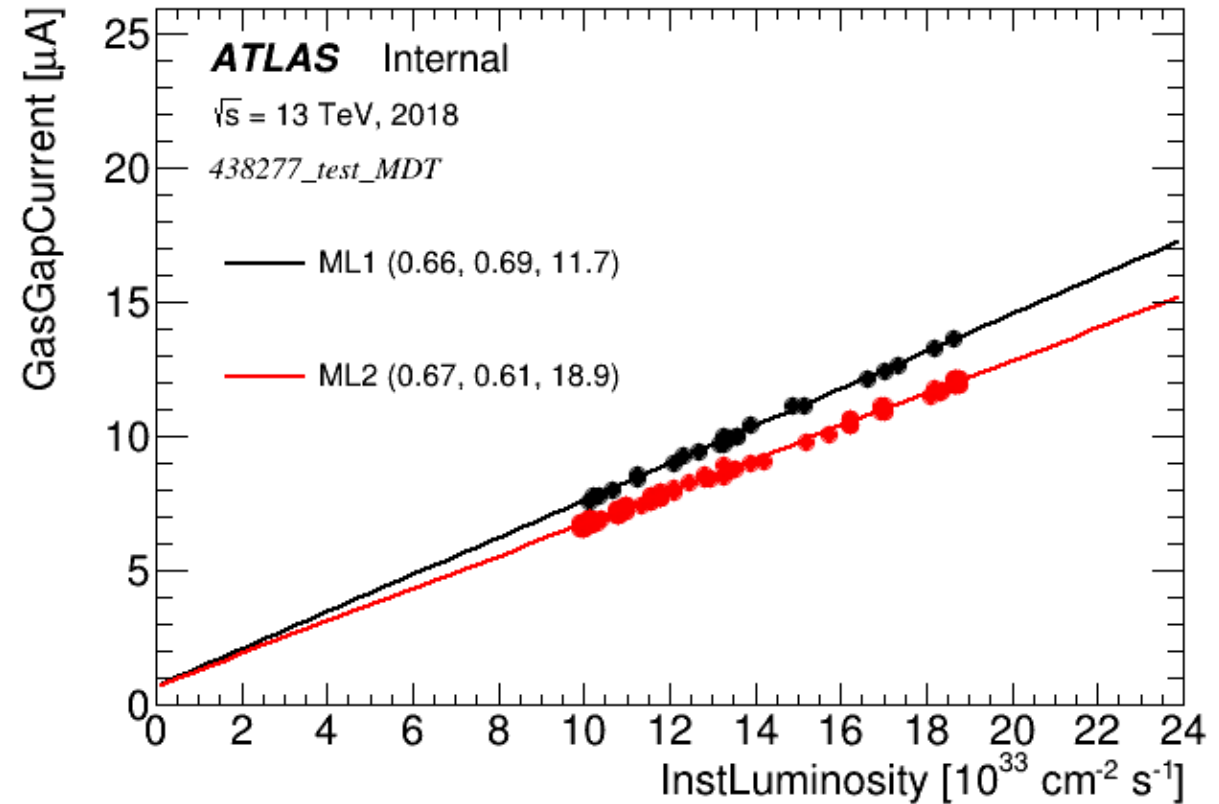
- Online monitoring system used to determine the status of the MDT detectors and front-end electronics

MDT JTAG Initialization State



# Project: MDT DCS Monitoring

- Working with Zhen Yan (Boston University) and Tiesheng Dai (University of Michigan)
- **Goal of the project:** Generate needed graphics using DCS data for offline monitoring of MDT detectors/electronics
  - Build web application to display data
  - **Ex:** HV iMon vs LHC Instantaneous Luminosity



Example and not mine

# Project: Database Query

- In order to query the large database containing MDT DCS data I had to learn SQL (Structured Query Language) and Apache Spark (pyspark)
- SQL is a language used to manage databases
- Apache Spark is an analytics engine for large-scale data processing
  - Pyspark is the python API for the use of Apache Spark

```
In [2]: %run -i mapping.py
```

```
In [*]: #x = mapping('451295')  
x = mapping('440613')  
x.find_run_interval()
```



The screenshot shows the Apache Spark web interface with a summary bar at the top: Apache Spark, 1 EXECUTORS, 4 CORES, Jobs: 1 RUNNING, 5 COMPLETED. Below is a table of job details.

ID	Job	Status	Stages	Tasks	Submission Time	Duration
▶ 0	parquet at NativeMethodAccessorImpl.java:0	COMPLETED	0/1(1) active	1/1	12 seconds ago	1.9s
▶ 1	parquet at NativeMethodAccessorImpl.java:0	COMPLETED	0/1(1) active	1/1	10 seconds ago	148ms
▶ 2	parquet at NativeMethodAccessorImpl.java:0	COMPLETED	0/1(1) active	1/1	10 seconds ago	102ms
▶ 3	collect	COMPLETED	0/1(1) active	4/4	2 seconds ago	2s
▶ 4	collect	COMPLETED	0/2(2) active	7/7	1 second ago	1.2s
▶ 5	count at NativeMethodAccessorImpl.java:0	RUNNING	0/2(2) active	37/191 (4)	7 seconds ago	-


# Project: CSV generation

- In order to extract the data from the database, I had to create an algorithm to query the database for the selected data and generate a .csv file with the data
- I have chosen to divide the data by **run number**


SWAN > My Projects > mdt\_dcs > mdt\_dcs\_mapping > csv

CSV ↑

NAME ▾

 run440613\_fsm.csv

 run440613\_HViMon.csv

 run451295\_fsm.csv

run451295\_HViMon.csv

# Project: Current Work & Next Steps

- Detect JTAG, HV, and LV fsm(finite state machine) failures from a selected run
  - Create some kind of graphic showing this
- Create algorithm to generate HV iMon vs Luminosity plots
- Start building web application to display graphics (streamlit.io)

```
#FAIL FINDER
filename = "../csv/run451295_fsm.csv"
fsm = loadDf(filename)
```

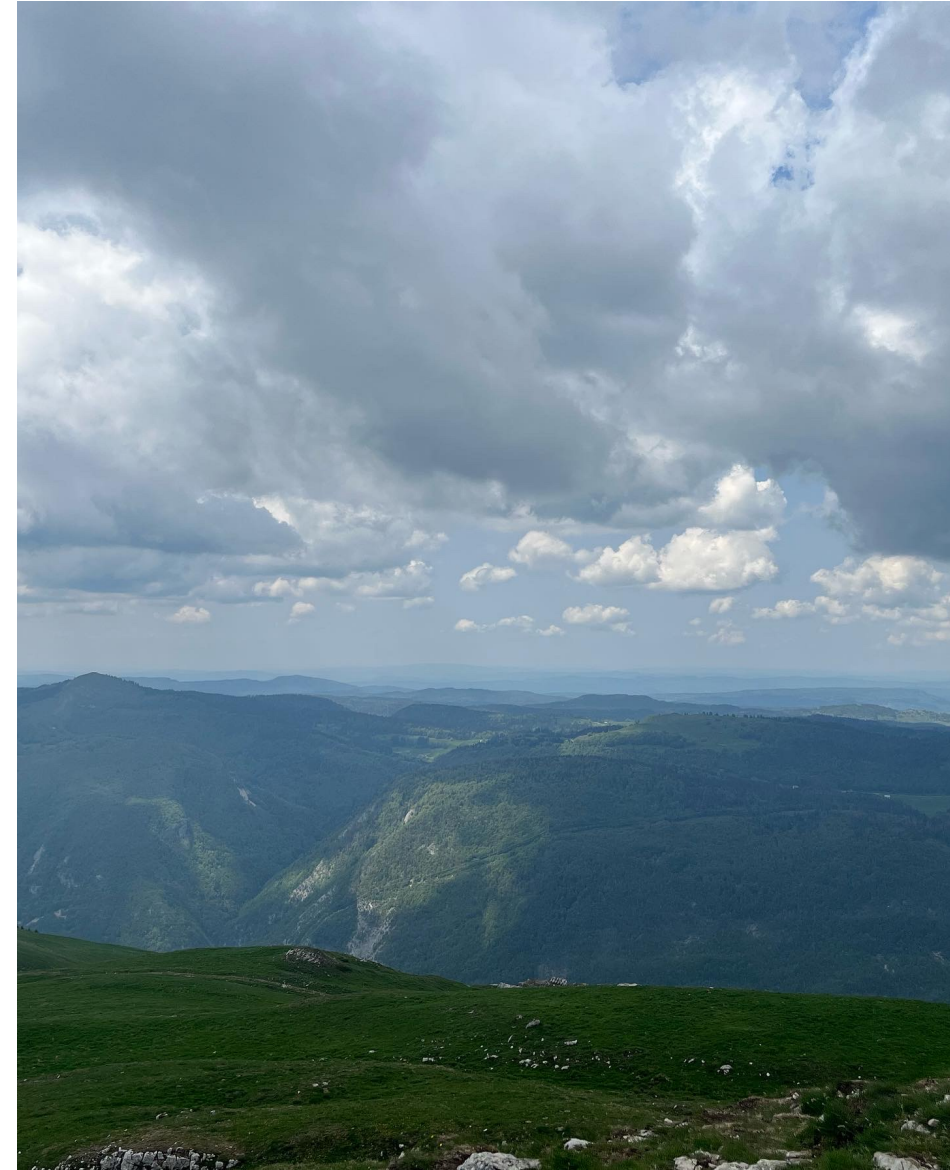
```
def JTAG_fail_finder(db):
    for k in db.index:
        val_string = db['value_string'][k]
        element_name = db['element_name'][k]
        if(not ("luminosityBlock" in element_name) and ("JTAG" in element_name) and va
            ts = db['ts'][k]
            print(f"JTAG CHAMBER ERROR: {element_name[15:-17]}; {val_string}; {ts}")
```

```
JTAG_fail_finder(fsm)
```

```
JTAG CHAMBER ERROR: BC_JTAG_INNER|BIS2C12; CAN_ELMB_NOOP; 2023-05-03 05:56:52.101000
JTAG CHAMBER ERROR: BC_JTAG_INNER|BIS3C12; CAN_ELMB_NOOP; 2023-05-03 05:56:52.101000
JTAG CHAMBER ERROR: BC_JTAG_INNER|BIS1C12; CAN_ELMB_NOOP; 2023-05-03 05:56:52.101000
JTAG CHAMBER ERROR: BC_JTAG_INNER|BIS5C12; CAN_ELMB_NOOP; 2023-05-03 05:56:52.110000
JTAG CHAMBER ERROR: BC_JTAG_INNER|BIS4C12; CAN_ELMB_NOOP; 2023-05-03 05:56:52.110000
JTAG CHAMBER ERROR: BC_JTAG_INNER|BIS6C12; CAN_ELMB_NOOP; 2023-05-03 05:56:52.112000
JTAG CHAMBER ERROR: BC_JTAG_OUTER|BOL1C09; FAILED; 2023-05-03 05:56:58.672000
```



# Outside of Work



# Thank you!