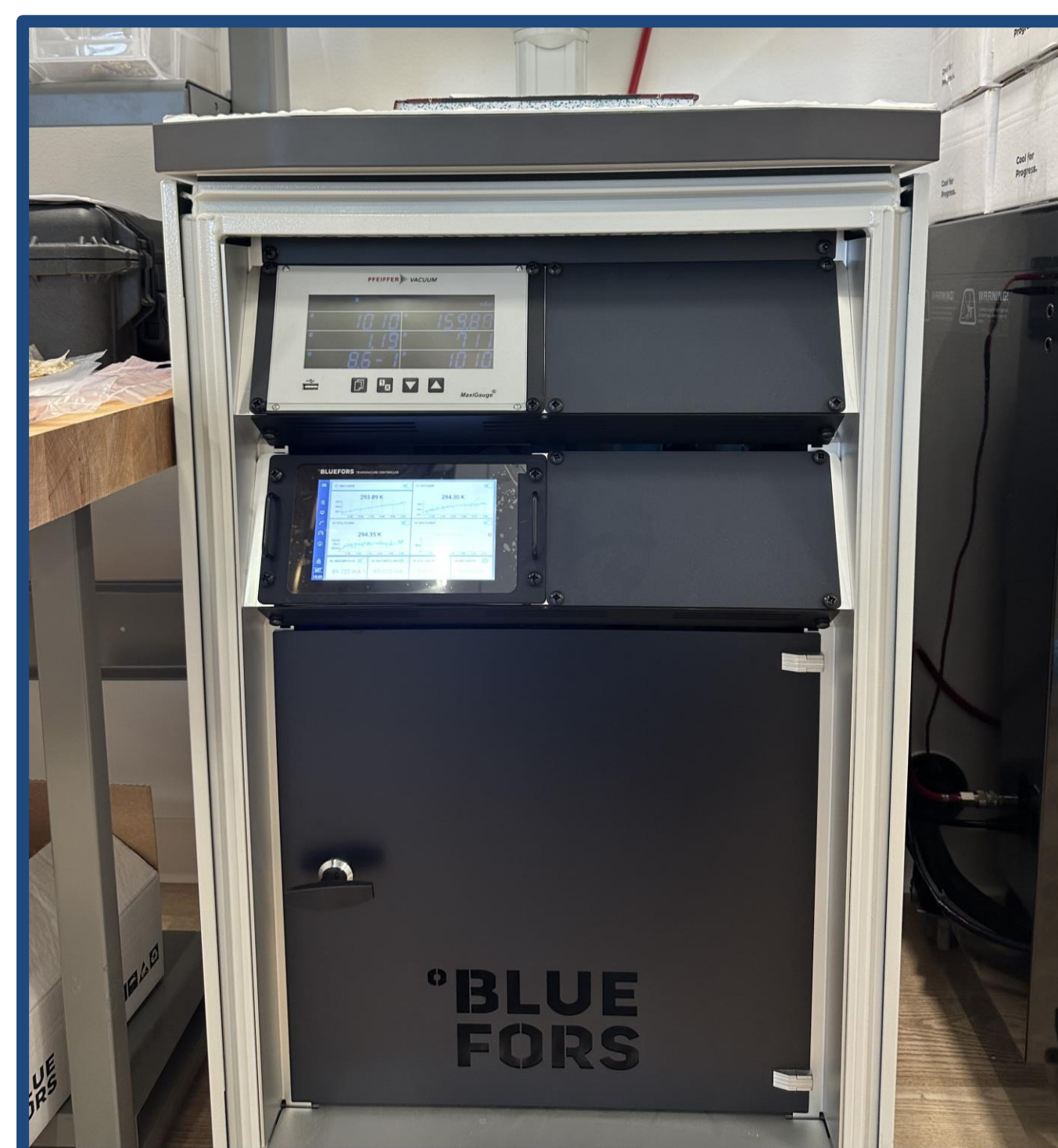


Implementation of a Web Interface to Display Real-Time Statistics of a Dilution Refrigerator

Hammad Rasheed, Fermilab & SQMS Intern, Supervisor: Nicholas Bornman

Overview

- Goal was to communicate with a Bluefors control unit, which monitors data from a dilution refrigerator.
- Display the data in a webpage with an aesthetic display.
- Update the data in real-time.



This is the Bluefors control unit that is attached to a dilution refrigerator.

```
import asyncio
from websockets.sync.client import connect

def hello():
    with connect("ws://10.225.99.17:5000") as websocket:
        websocket.send("Hello world!")
        message = websocket.recv()
        print(f"Received: {message}")

hello()
```

Initial Working Websocket Client

```
import asyncio
from websockets.server import serve

# Necessary imports to run the websocket server and run the server continuously

sync def echo(websocket):
    async for message in websocket:
        await websocket.send(message)
        print("MESSAGE: " + message)
        # Waits until a message is received

sync def main():
    async with serve(echo, "", 5000):
        await asyncio.Future() # Makes the server run continuously
        # Creates a server with the IP of the computer with a port of 5000

asyncio.run(main())
# Runs the server locally
```

Initial Working Websocket Server

Initial Goals

- Understanding the websockets library in Python.
- Communicating between two different computers using a websocket server and websocket client.
- Learning how to efficiently use Python and JavaScript.

```
{
  "command": "read",
  "id": "6352827e-1ac3-11ec-bdf6-14dae904baea",
  "data": {
    "target": "mapper.bf.flow"
  }
}
```

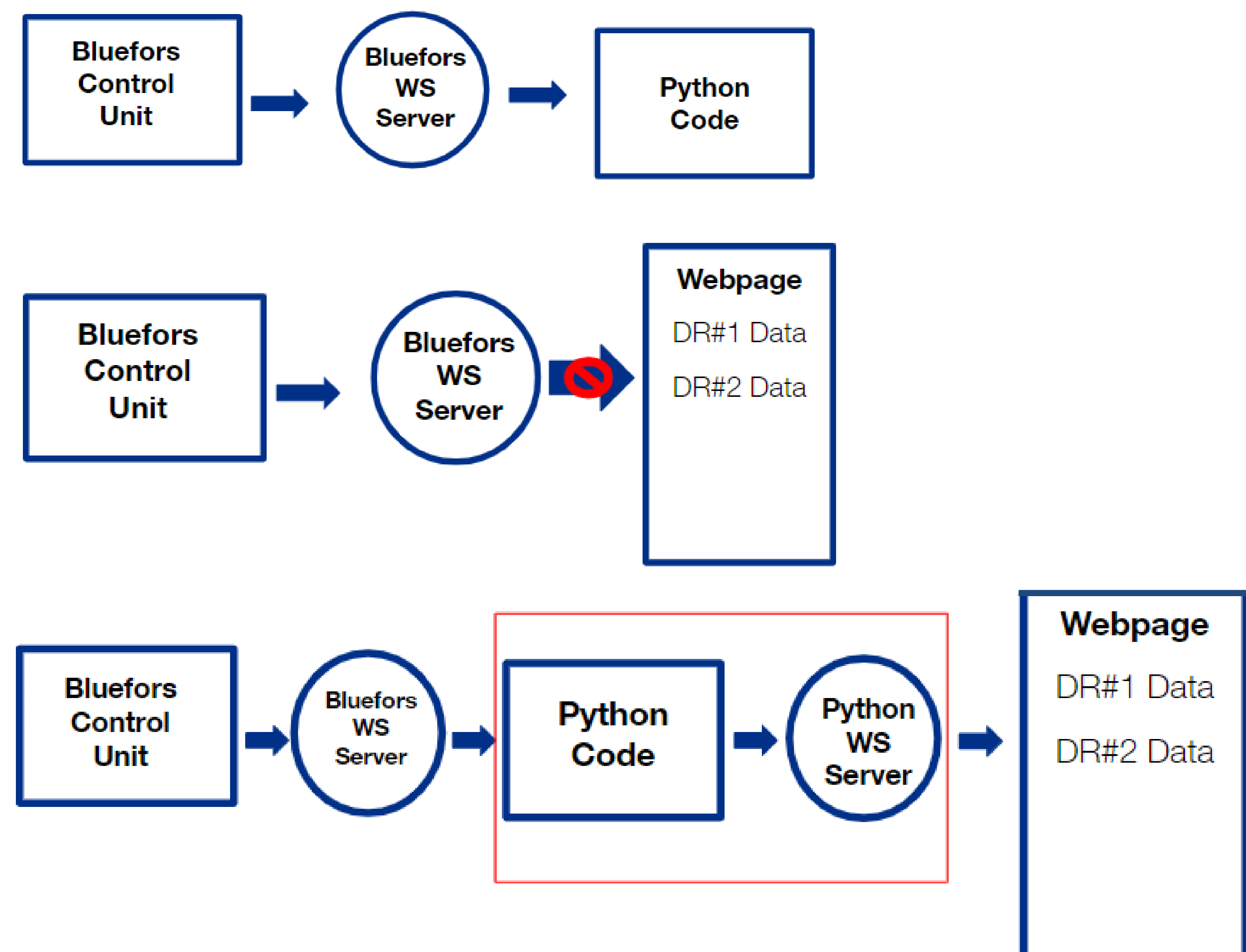
```
Received
{
  "id": "6352827e-1ac3-11ec-bdf6-14dae904baea",
  "status": "RECEIVED",
  "data": {
    "command": "read",
    "id": "6352827e-1ac3-11ec-bdf6-14dae904baea",
    "data": {
      "target": "mapper.bf.flow"
    }
  }
}

Received
{
  "id": "6352827e-1ac3-11ec-bdf6-14dae904baea",
  "status": "SUCCEEDED",
  "data": {
    "name": "mapper.bf.flow",
    "type": "Value.Number.Float",
    "content": {
      "read only": true,
      "maximum_age": 5000,
      "lockable": false,
      "locked": true,
      "owner": "driver.vc.flow",
      "latest valid value": {
        "value": "1.23",
        "outdated": false,
        "date": "1632218701084",
        "status": "SYNCHRONIZED",
        "exception": ""
      }
    }
  }
}
```

Understanding Bluefors API

- Read the Bluefors control software user manual to understand how to communicate with the equipment.
- Understand how to send JSON requests and handle received JSON objects.
- Figure out how to access the websocket servers hosted by the control unit.

Progress and Change in Methods



This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTs) under the Community College Internships Program (CCI)