🚀

EX-UI Notes

EX-UI uses the existing codebase of OpenMCT (and thus uses the same stack) as a basis and builds upon it by adding more feature to meet the requirements of WARR. **OpenMCT** is a mission control framework for data visualisation. It was developed by NASA and the Ames Research Center and the Jet Propulsion Laboratory (JPL).

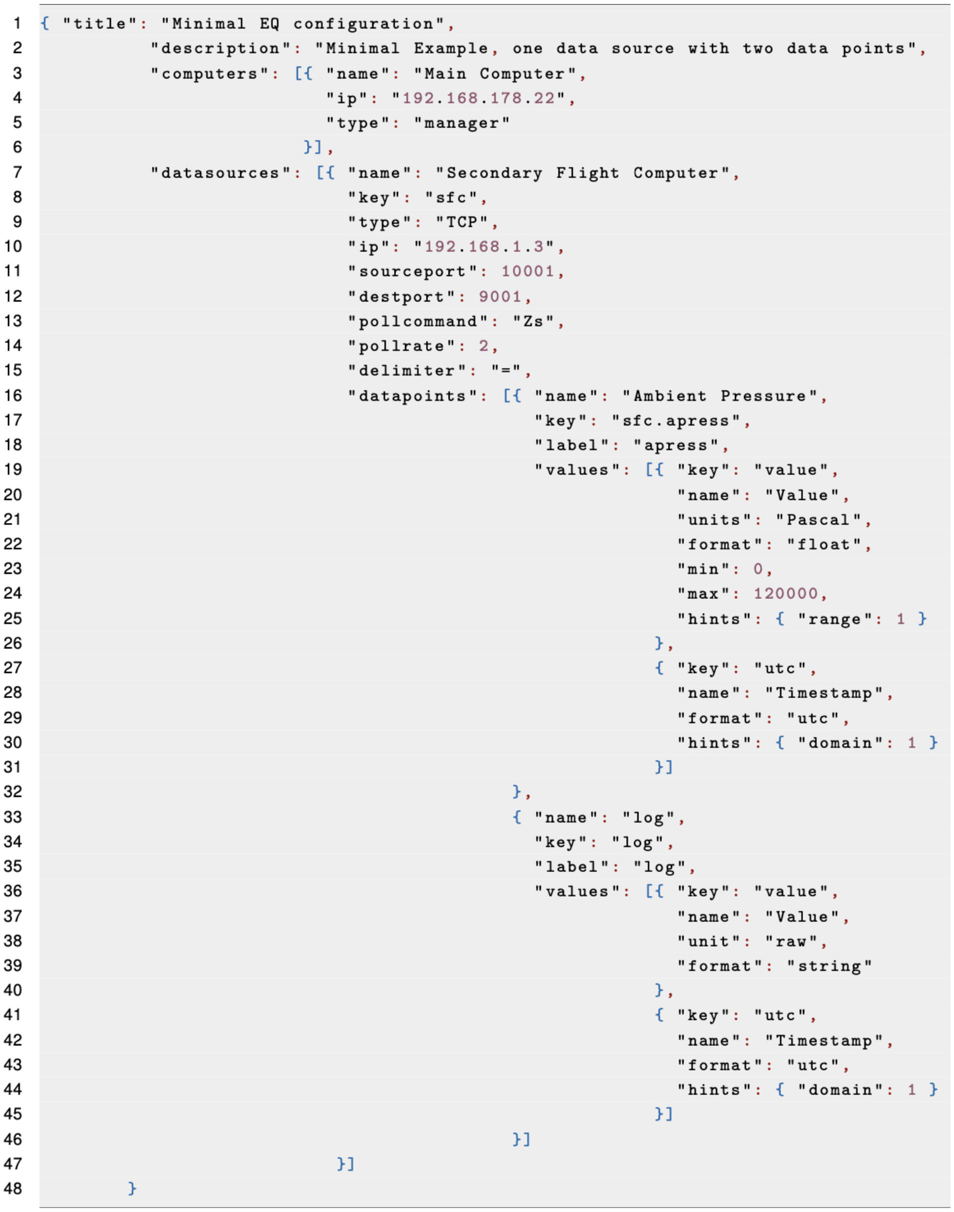
The Structure of OpenMCT

- Backend: **NodeJS, Express** and **CouchDB**
- Frontend: **VueJS** and **SASS**

While OpenMCT provides various visualisations for different types of data, it needs various extensions to support more types of data and different visualisation methods. Extending OpenMCT is simply done by developing a website for the required type of data and visualisation and embedding it into OpenMCT which is a feature supported by the framework. OpenMCT also supports modifying its UI elements by editing the `layout.vue` file and using various **SASS** files.

Equipment Configuration File

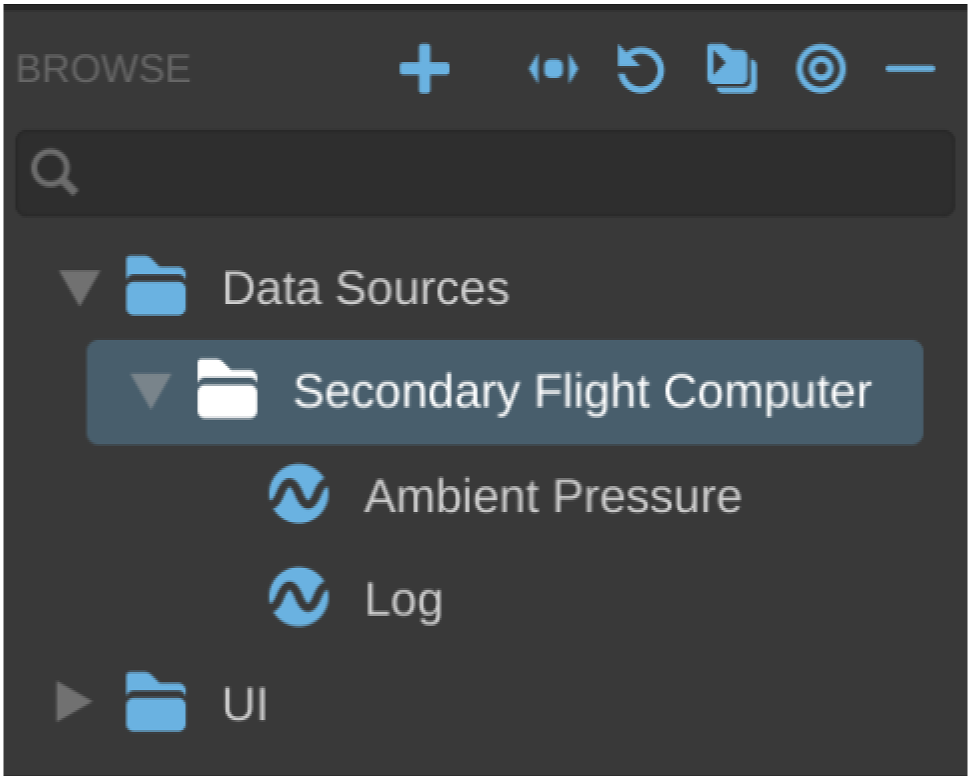
As OpenMCT visualises data from different sources, an **Equipment Configuration File** is used as an automatic configuration and unifies configs across all code domains.



```
{ "title": "Minimal EQ configuration",
          "description": "Minimal Example, one data source with two data points",
          "computers": [{ "name": "Main Computer",
                          "ip": "192.168.178.22",
                          "type": "manager"
                        }],
          "datasources": [{ "name": "Secondary Flight Computer",
                            "key": "sfc",
                            "type": "TCP",
                            "ip": "192.168.1.3",
                            "sourceport": 10001,
                            "destport": 9001,
                            "pollcommand": "Zs",
                            "pollrate": 2,
                            "delimiter": "=",
                            "datapoints": [{ "name": "Ambient Pressure",
                                             "key": "sfc.apress",
                                             "label": "apress",
                                             "values": [{ "key": "value",
                                                          "name": "Value",
                                                          "units": "Pascal",
                                                          "format": "float",
                                                          "min": 0,
                                                          "max": 120000,
                                                          "hints": { "range": 1 }
                                                        },
                                                        { "key": "utc",
                                                          "name": "Timestamp",
                                                          "format": "utc",
                                                          "hints": { "domain": 1 }
                                                        }]
                                           },
                                           { "name": "log",
                                             "key": "log",
                                             "label": "log",
                                             "values": [{ "key": "value",
                                                          "name": "Value",
                                                          "unit": "raw",
                                                          "format": "string"
                                                        },
                                                        { "key": "utc",
                                                          "name": "Timestamp",
                                                          "format": "utc",
                                                          "hints": { "domain": 1 }
                                                        }]
                                           }]
                           }]
        }
```

Picture is from Antonio's Thesis, Page 47.

As seen above, the equipment configuration file has only one data source, supplying 2 data points. The equipment configuration file can then be parsed to **Domain Objects** by a plugin called EQ. Domain Objects are objects created by OpenMCT which represent all the visualisations and data points visualised by OpenMCT. The resulting domain objects are found in the Object Tree as shown in the figure below:



Picture is from Antoni's Thesis, Page 48.

Backend Architecture

Web Server

The express web server was slightly adjusted and an endpoint `http://server-ip:8080/eq.json` was added so that any one using EX-UI can access the equipment configuration file.

Database

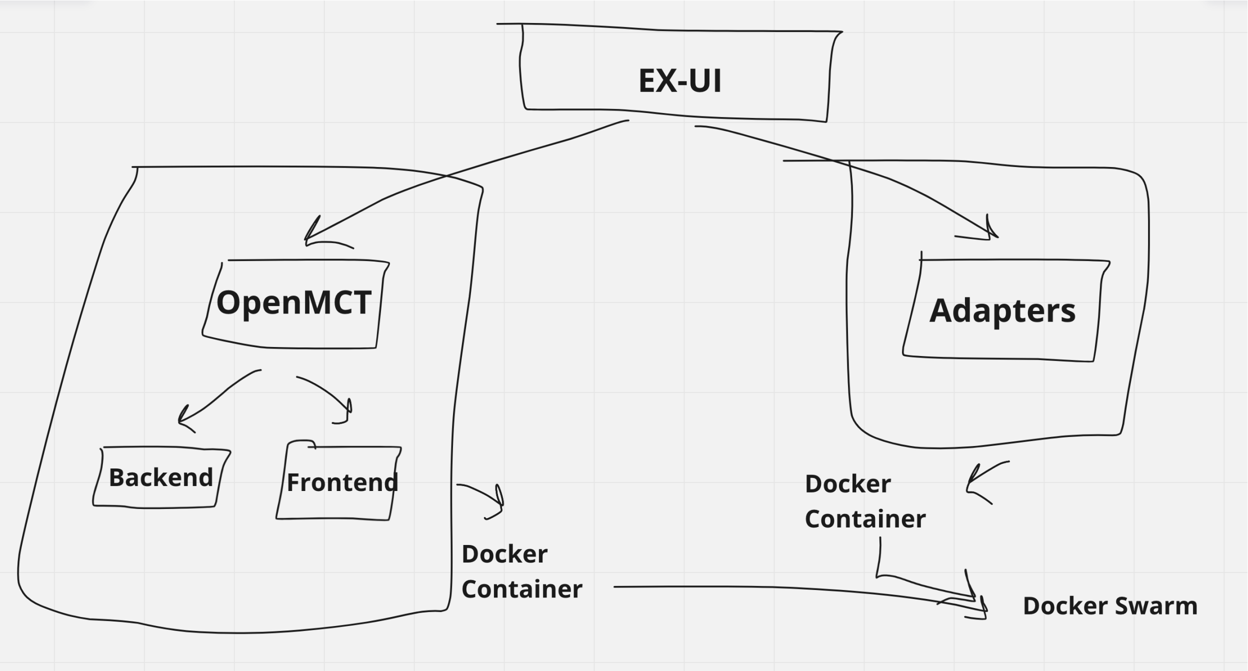
OpenMCT uses **CouchDB** to store user-created objects

Adapters

In order to integrate devices into EX-UI, their interfaces needs to be mapped by some code to an interface that a browser can support. This is done through **Adapters.** Adapters are pieces of code that help extend OpenMCT by basically taking input from a device, process and transform it to a format that OpenMCT can visualise. Each adapter is responsible only for 1 task in order to preserve the modularity of the adapters.

Distributing the System

Some adapters perform computationally extensive tasks and thus running the entirety of EX-UI on one single computer becomes impractical and the need for distributing the system arises. To solve this problem, **Docker** was used. Each of the web server and the adapter were a docker container of their own then they were replicated and distributed over multiple computers.



Flexing my amazing drawing skills xd

Installation and Startup

To freshly install EX-UI on a new system, the `install.py` script must be run. The `install.py` is responsible for installing docker and other required dependencies for EX-UI to run. To start the project and the docker swarm, run the `start.py` script.