

# Predictive Maintenance & Data Logger for Industry 2.0

James Conlon, Animisha  
Sharanappa

# Overview:

## Predictive Maintenance

- Using Machine Learning to predict a machine part failure before it happens
- Eg: situation where operator can see rust on a pipe before it bursts and explodes and ruins everything
- They should replace it **before** it bursts.
- In real life “rust” is hard to see
- Requires high frequency data ( $\sim 100\text{Hz}$ ) as we model things no operator could see like Amplitude deviations measured with an accelerometer.

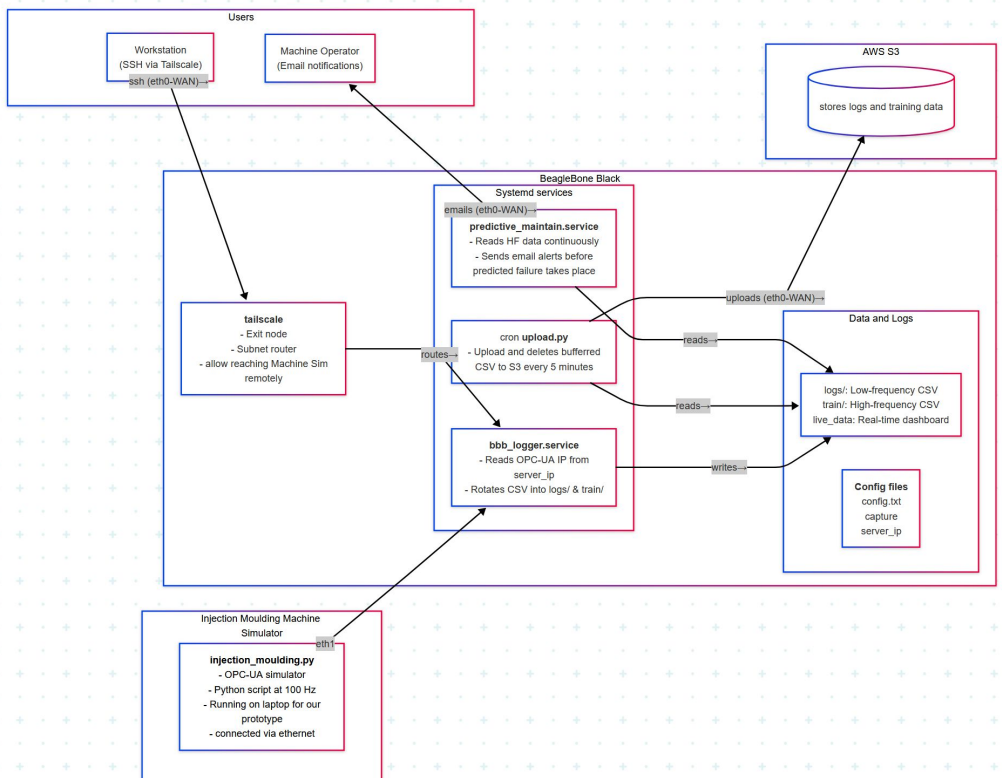
### Injection Molding Machine



# Method

Goal: improve on data loggers and current maintenance techniques, to be configurable by machine operator, save on cloud costs for HF data, work on bad network connection

- ML done locally (on the “edge device”)
- Monitor, Log data, look ahead
- Main components:
  - Injection moulding simulator for ML
  - Buffering csv data files locally
  - Sending buffered data to AWS
  - ML on device (Predictive Maintenance with operator notifications)



# Results

- The system did not fail overnight and continuously logged data. Every 5 mins it sent good sized files as defined in the cron job.
- The ML model could somewhat accurately predict failures before they happen and send alert notifications
- Early warning prediction ~10-15s. While we currently model a 20% failure rate per cycle, we got many false positives. (around 4 / 10 alerts are false positive)
- In reality, you would also like higher lead time on alerts, but neither our ML model or data simulator is able to forecast that far in advance.

	Name	Type	Last modified
	<a href="#">log_1745910296.csv</a>	csv	April 29, 2025, 03:10:44 (UTC-04:00)
	<a href="#">log_1745910409.csv</a>	csv	April 29, 2025, 03:10:44 (UTC-04:00)
	<a href="#">log_1745910523.csv</a>	csv	April 29, 2025, 03:10:44 (UTC-04:00)
	<a href="#">log_1745910637.csv</a>	csv	April 29, 2025, 03:15:44 (UTC-04:00)
	<a href="#">log_1745910750.csv</a>	csv	April 29, 2025, 03:15:45 (UTC-04:00)
	<a href="#">log_1745910863.csv</a>	csv	April 29, 2025, 03:20:45 (UTC-04:00)
	<a href="#">log_1745910977.csv</a>	csv	April 29, 2025, 03:20:45 (UTC-04:00)
	<a href="#">log_1745911090.csv</a>	csv	April 29, 2025, 03:20:45 (UTC-04:00)
	<a href="#">log_1745911203.csv</a>	csv	April 29, 2025, 03:25:46 (UTC-04:00)
	<a href="#">log_1745911316.csv</a>	csv	April 29, 2025, 03:25:46 (UTC-04:00)
	<a href="#">log_1745911430.csv</a>	csv	April 29, 2025, 03:25:46 (UTC-04:00)
	<a href="#">log_1745911544.csv</a>	csv	April 29, 2025, 03:30:47 (UTC-04:00)
	<a href="#">log_1745911657.csv</a>	csv	April 29, 2025, 03:30:47 (UTC-04:00)
	<a href="#">log_1745911771.csv</a>	csv	April 29, 2025, 03:35:49 (UTC-04:00)
	<a href="#">log_1745911884.csv</a>	csv	April 29, 2025, 03:35:48 (UTC-04:00)
	<a href="#">log_1745911997.csv</a>	csv	April 29, 2025, 03:35:48 (UTC-04:00)
	<a href="#">log_1745912110.csv</a>	csv	April 29, 2025, 03:40:48 (UTC-04:00)
	<a href="#">log_1745912224.csv</a>	csv	April 29, 2025, 03:40:48 (UTC-04:00)
	<a href="#">log_1745912336.csv</a>	csv	April 29, 2025, 03:45:49 (UTC-04:00)
	<a href="#">log_1745912448.csv</a>	csv	April 29, 2025, 03:45:49 (UTC-04:00)
	<a href="#">log_1745912562.csv</a>	csv	April 29, 2025, 03:45:49 (UTC-04:00)
	<a href="#">log_1745912675.csv</a>	csv	April 29, 2025, 03:50:50 (UTC-04:00)
	<a href="#">log_1745912790.csv</a>	csv	April 29, 2025, 03:50:50 (UTC-04:00)
	<a href="#">log_1745912903.csv</a>	csv	April 29, 2025, 03:50:50 (UTC-04:00)
	<a href="#">log_1745913016.csv</a>	csv	April 29, 2025, 03:55:51 (UTC-04:00)
	<a href="#">log_1745913129.csv</a>	csv	April 29, 2025, 03:55:51 (UTC-04:00)

# Conclusions/Future Work

- The project presents and demonstrates a working prototype for predictive maintenance and data logging.
- Tuning the ML model to be faster and more accurate.
- Would require real data or a better data set – our data set is very small compared to a real one that may have dozens or hundreds of data points, not 6.
- Running on multithreaded CPU (not beaglebone) so no blocking threads between data collector and ML
- Making the data points configurable in the config.txt instead of hard coding them in the binary
- Making a better frontend interface