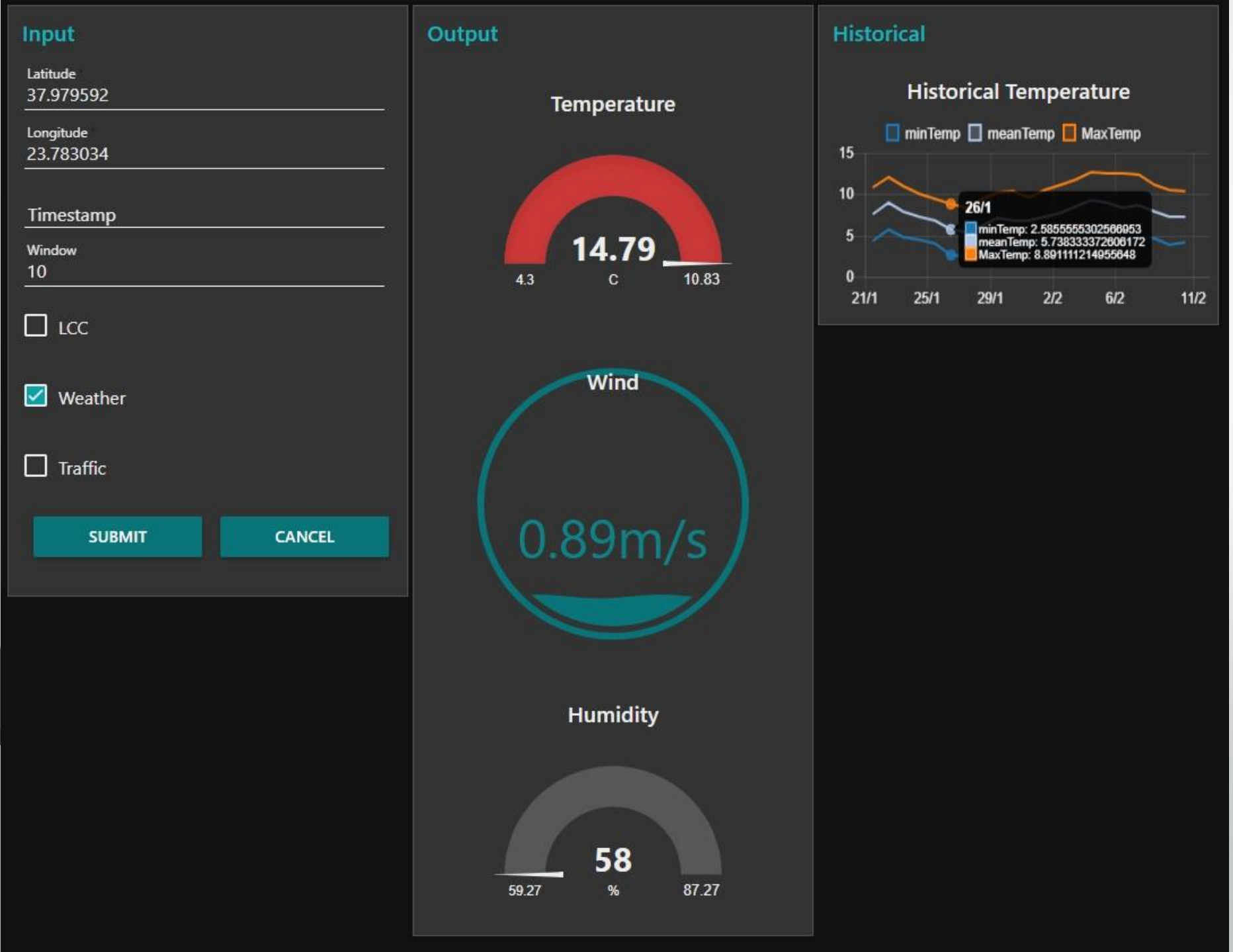
Chain Status Modeling and Visualization

Affectus Hackathon 2018

Weather Crawler

- NodeRed Implementation
- Uses the aWhere and Darksky APIs
- Collects norms about the weather conditions (temperature, wind and humidity)
- Compares current (or chosen timestamp) conditions with norms
- Emits relevant events (too cold, too hot, too humid, too dry e.t.c.)



Weather Front End

Helper Flows

- Stream Processor: Creates test streams and applies some preprocessing to all streams (data ingestion)
- Delay Checker: Runs through the MongoDB data store, locates things that have surpassed the expected duration and forwards ‘delayed’ events to a web socket
- Stage Identifier: Identifies the chain stage based on a scan geo-location data by cross-referencing it with the knowledge base, and adds it to the scan data stream

Tensor Flows

- An open-source symbolic math library used for machine learning applications such as neural networks.
- Tensors are multi-dimension arrays of numerals.
- Supports classification, regression and clustering using linear regressors or deep neural networks.
- Developed in Python.

Example Data

Day	Month	Chain Stage	Product ID	Thing ID	Hot or Cold	Humid or Dry	Windy	Duration
3	10	2	653	1235	-1	1	0	5213
14	9	3	523	5243	1	0	1	1262
6	9	1	123	12	0	1	1	6342
10	7	1	63	532	-1	-1	0	123567

Example Tensor

[[3,14,6,10],[10,9,9,7],[2,3,1,1],[653,523,123,63],[1235,5243,12,532],[-1,1,0,-1],[0,1,1,0],[5213,1262,6342,123567]]

Affectus Chain Modeling

- Python Backend for TensorFlows modeling
- NodeRed front end for processing incoming data into TensorFlows and storing data
- Connected by Rest API calls using HTTP Post

Chain Modeling Backend

- Flask python server to listen for incoming model train or prediction requests
- If request is train start training process and block all other incoming requests until it is completed
- If request is prediction use the trained model to predict the expected duration that the Thing is supposed to remain in this chain stage
- Features are parametric but some are required and some optional
  - Required: Thing ID, Product ID, Day, Hour, Chain Stage
  - Optional: Temperature, Humidity, Traffic, Wind, Month, LCC status…

Chain Modeling Frontend

- Receives data points in JSON format, through a web socket
- Processes it into the correct JSON format and makes prediction calls to the backend
- If enough data points have been collected triggers a retraining to the backend
- Stores the expected duration into MongoDB

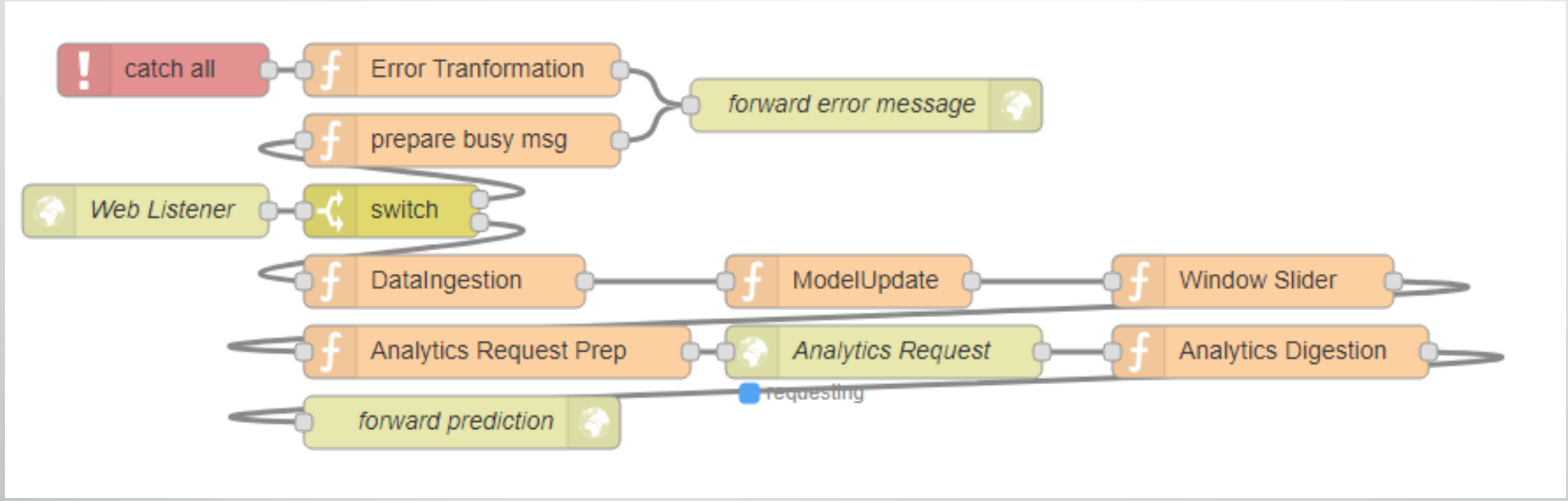
Communication Templates

Front to Back End

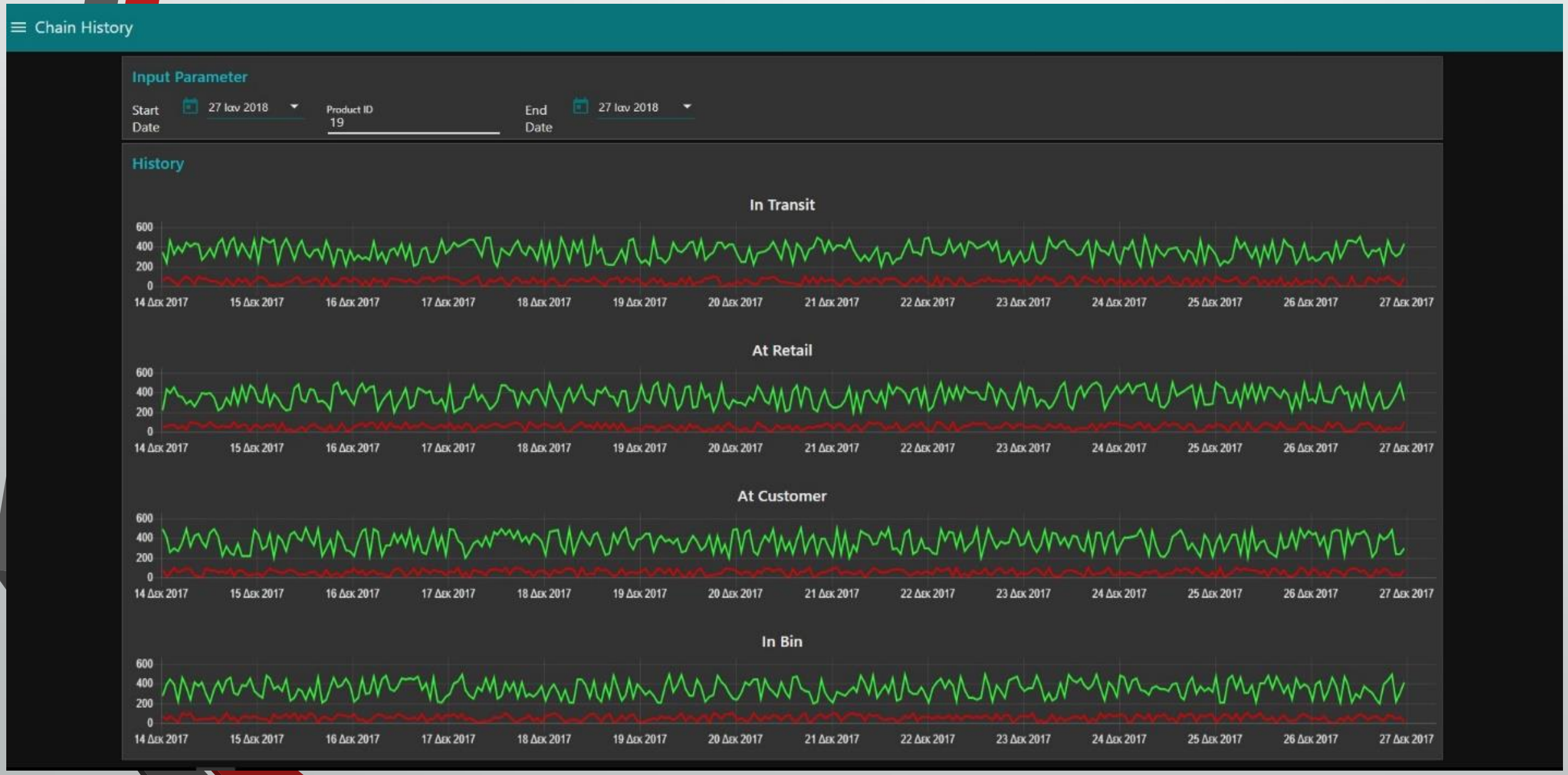

```
{"data":[{
        "thing_id":644,
        "product_id":2,
        "data":[ {
            "stage":2,
            "day":5,
            "hour":10,
             "duration":312410322.36}]}],
        "parameters":{
            "retrain":false,
            "refresh_data":false,
            "train_file":"./scan_data/train.csv",
            "model_dir":"./model_dir",
            "model_parameters":{
                 "learning_rate":0.6}}
}
```


Back to Front End

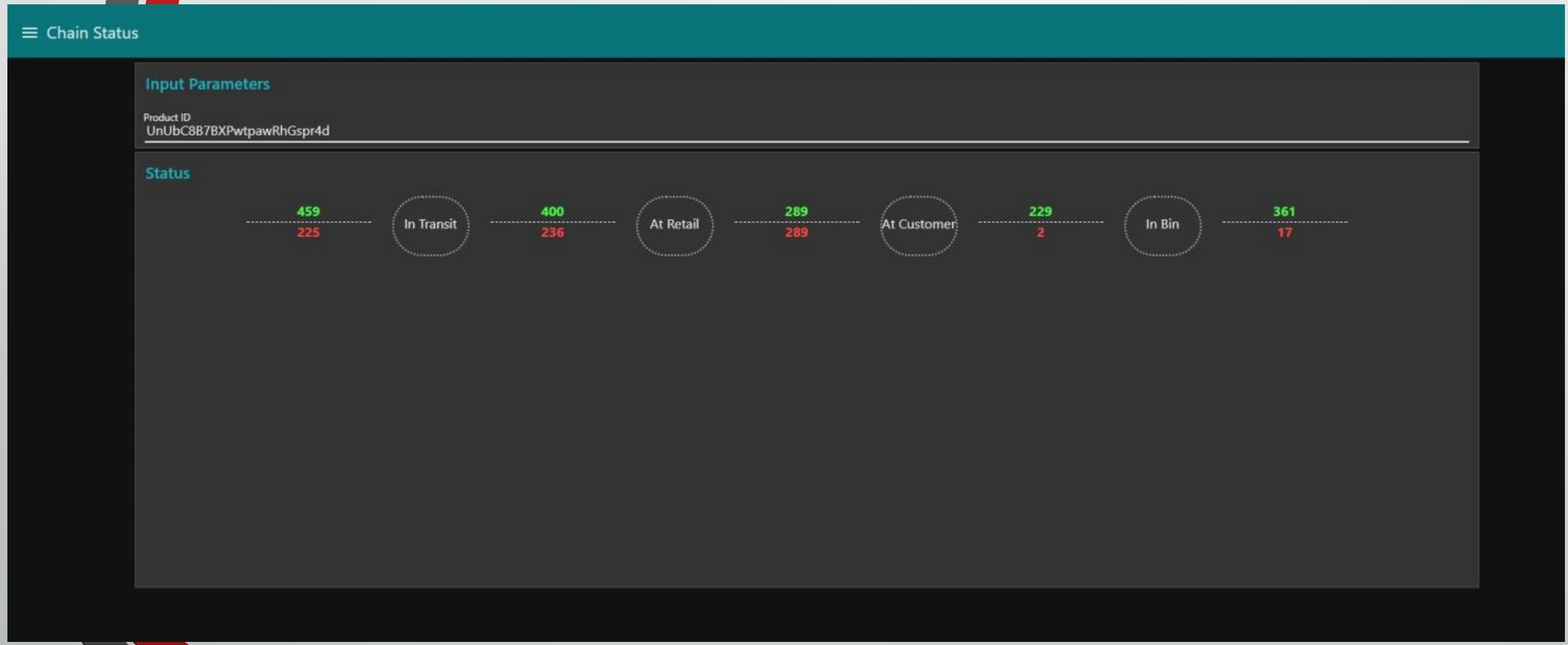

```
{
   "predictions":{
      "predictions":[[3124103.75]],
      "costs":[[52.64]],
      "actuals":[[312410322.36]],
      "error ": "NullPointerException"},
   "retrain_output":"Training already running."
}
```


catch all
Error Tranformation
forward error message
prepare busy msg
Web Listener
switch
DataIngestion
ModelUpdate
Window Slider
Analytics Request Prep
Analytics Request
Analytics Digestion
requesting
forward prediction

Chain Stage History Viewer

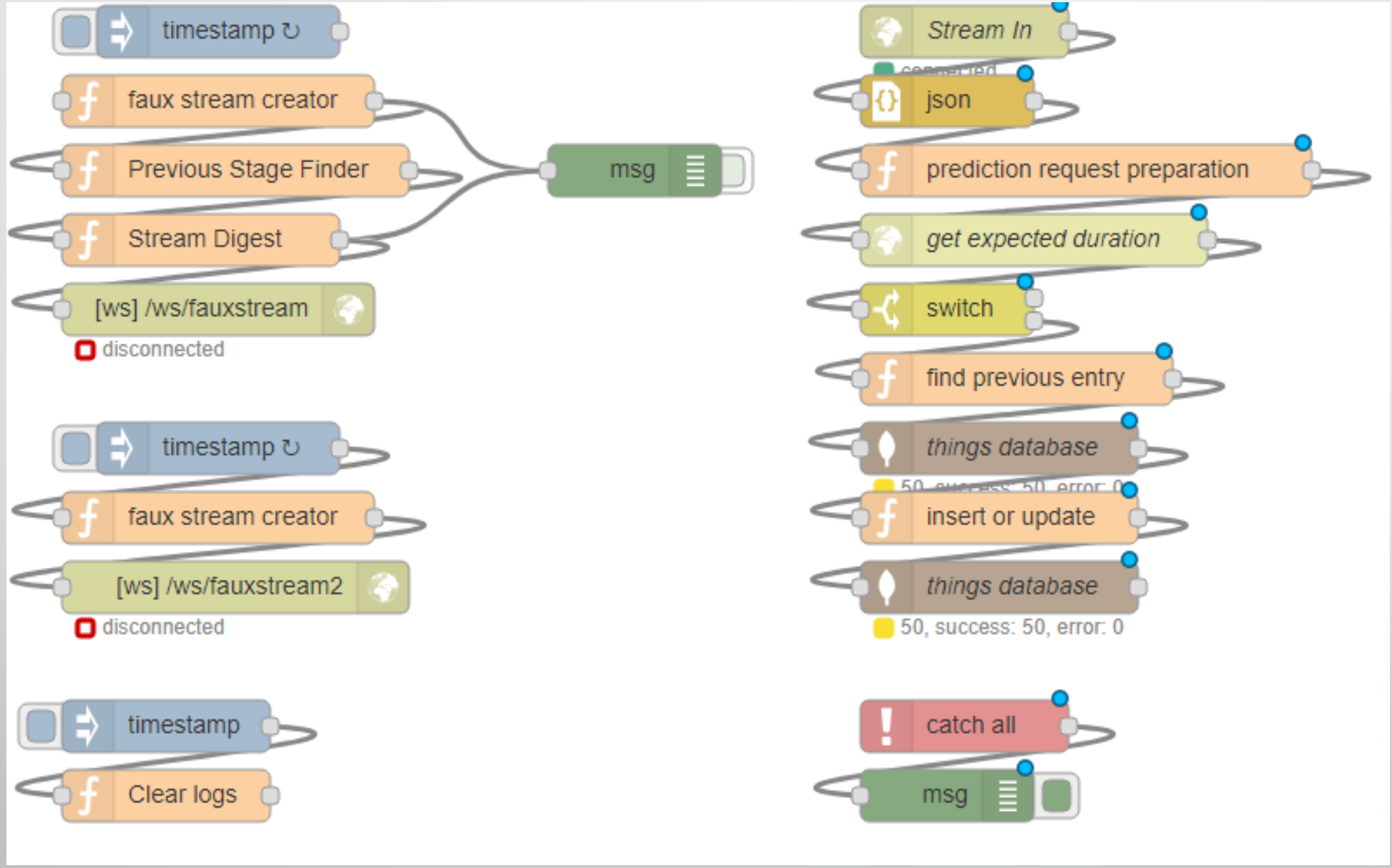


Chain Stage Live Monitoring



Test Data

- Experimental stream created using random number generator
- NodeRed flow used: Stream Processor
- A new datapoint (scan) is created every 10 seconds
- Each scan contains: timestamp, thing id, product id, location, stage.
- The scan is forwarded to a websocket and triggers all other flows.


timestamp
faux stream creator
Previous Stage Finder
Stream Digest
[ws] /ws/fauxstream
disconnected
msg
timestamp
faux stream creator
[ws] /ws/fauxstream2
disconnected
timestamp
Clear logs
Stream In
json
prediction request preparation
get expected duration
switch
find previous entry
things database
insert or update
things database
50, success: 50, error: 0
catch all
msg

Questions ?