# Building Smart IoT Applications with Python and Spark

...

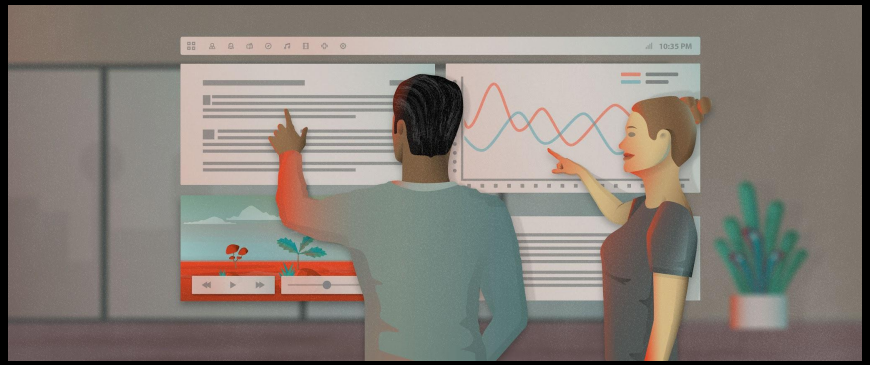
Rafael Schultze-Kraft

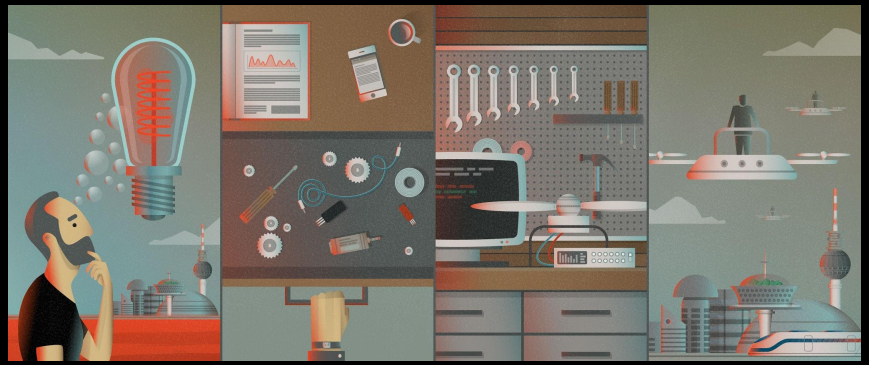
PyData Berlin 2017

WATTx

# WHO WE ARE

WATTx



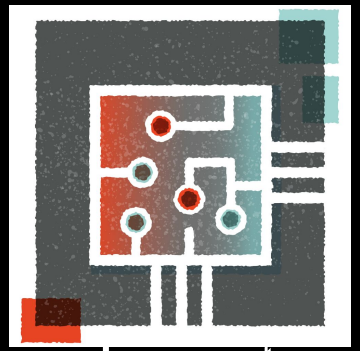


# OUR PROCESS

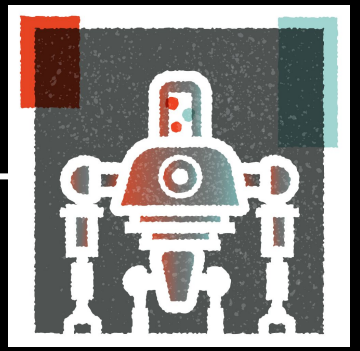
Ideation

Discovery

Prototyping



Venture

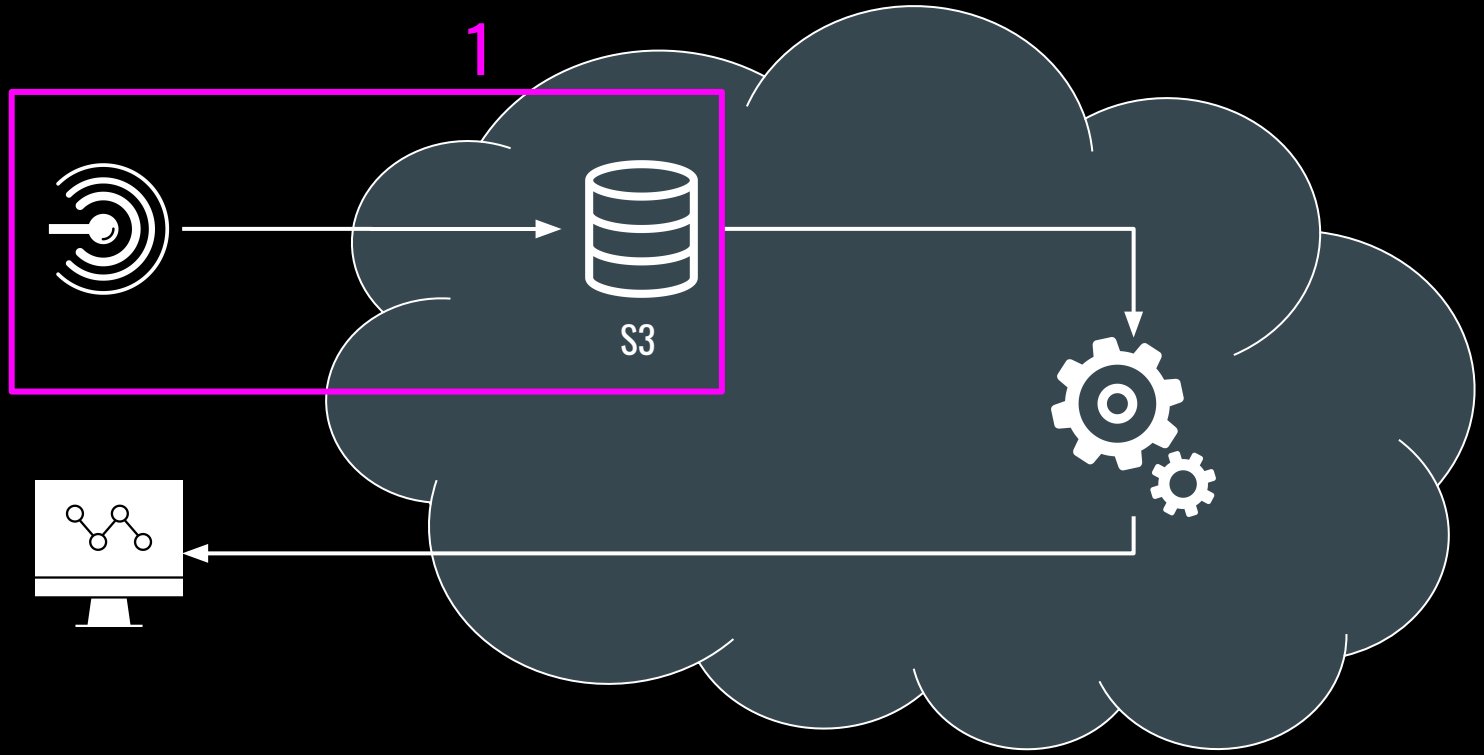


# Smart Building Infrastructure

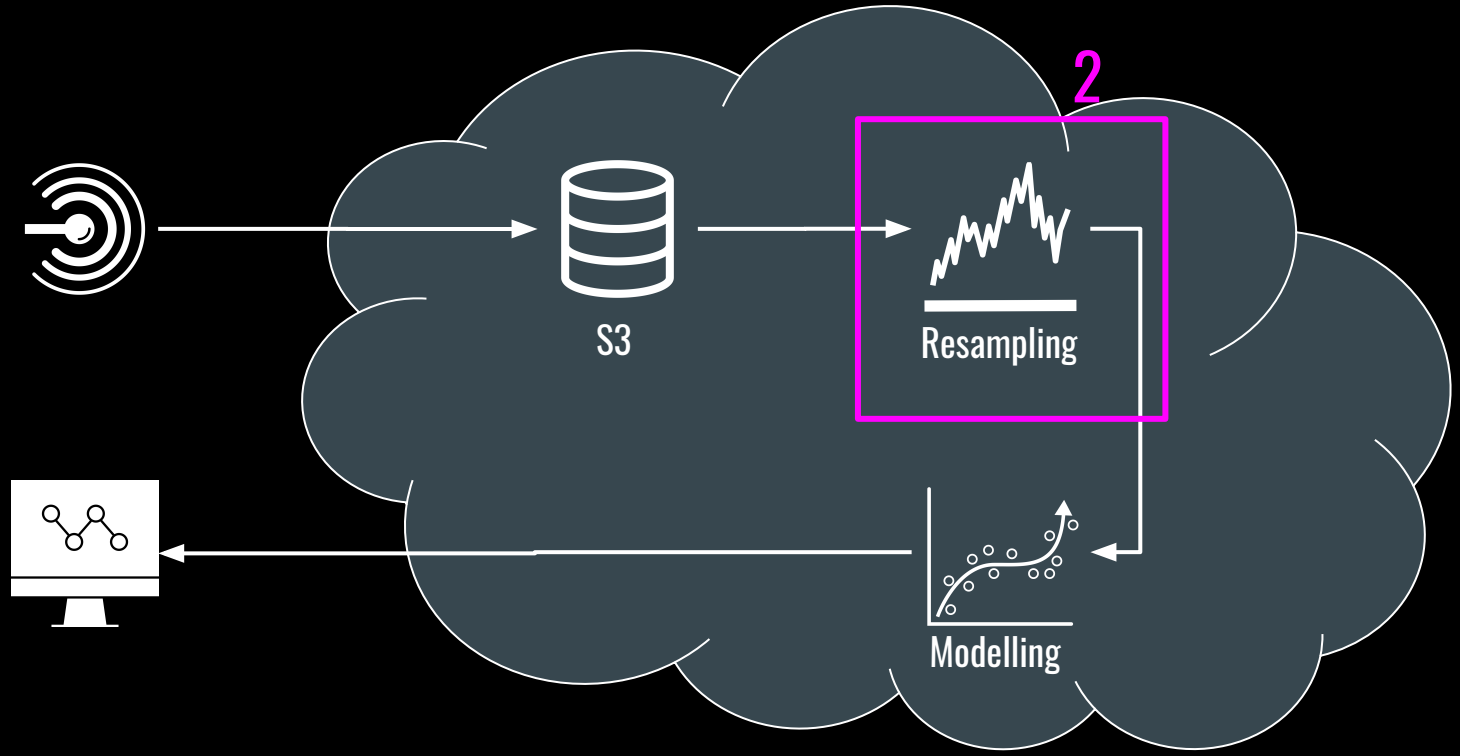

www.snuk.io

# IoT APPLICATION

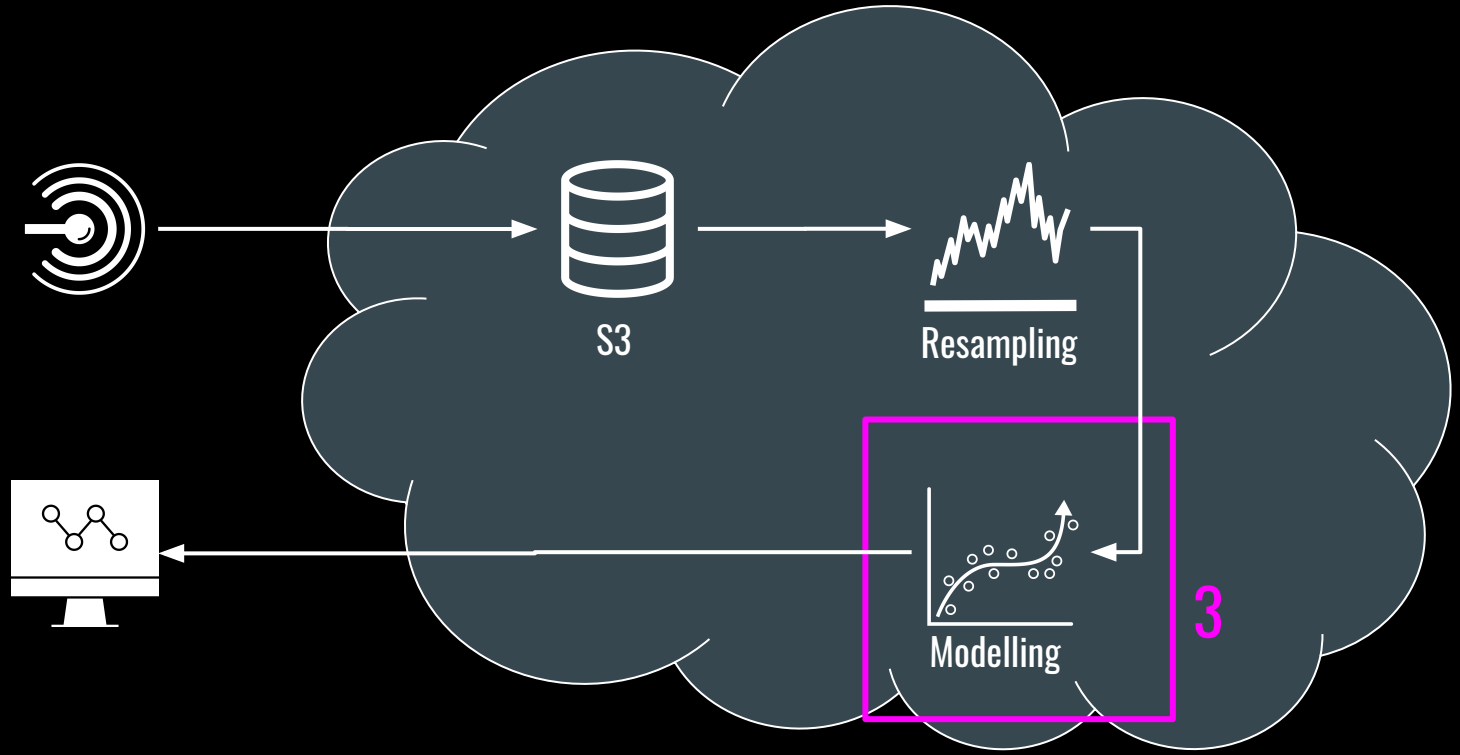
# 1 SMART OFFICE IoT SETUP



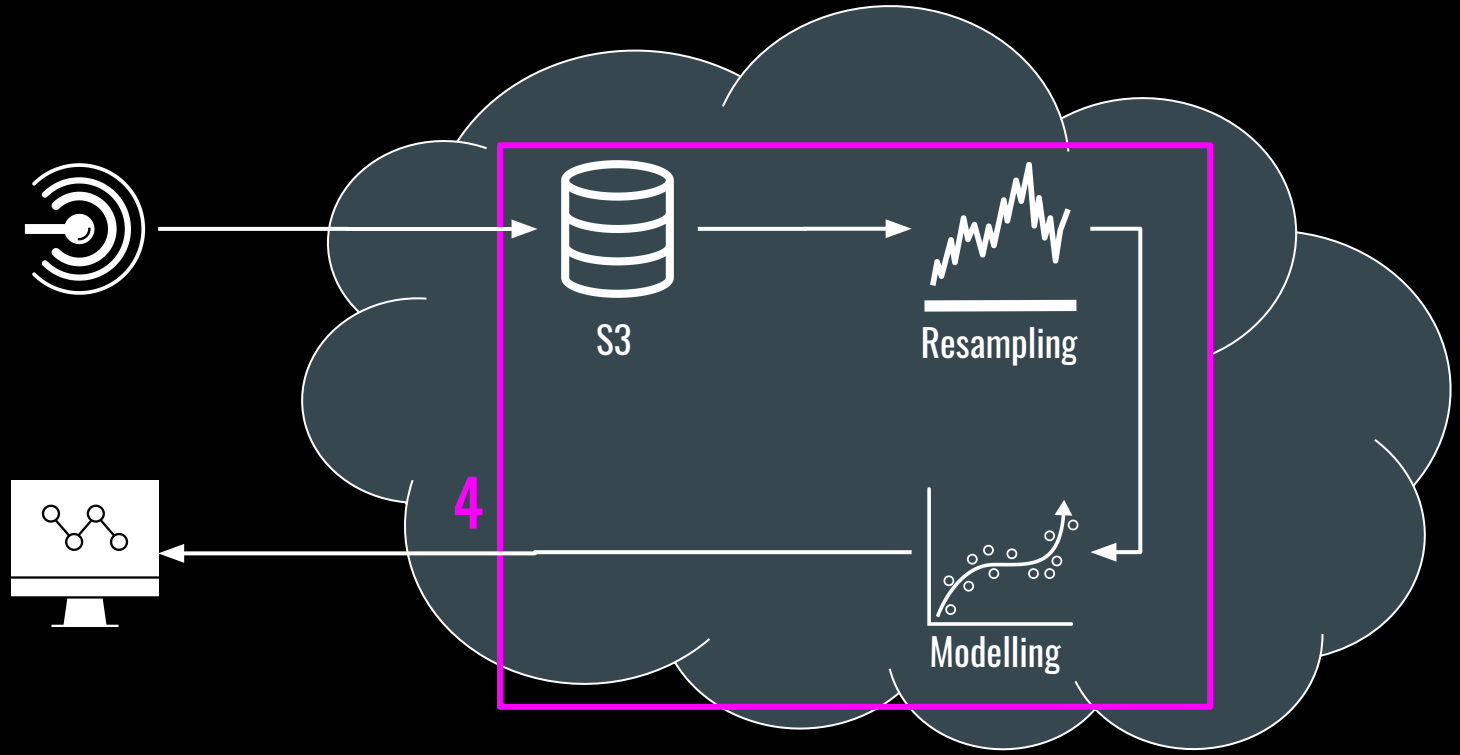
# 2 RESAMPLING IoT SENSOR DATA



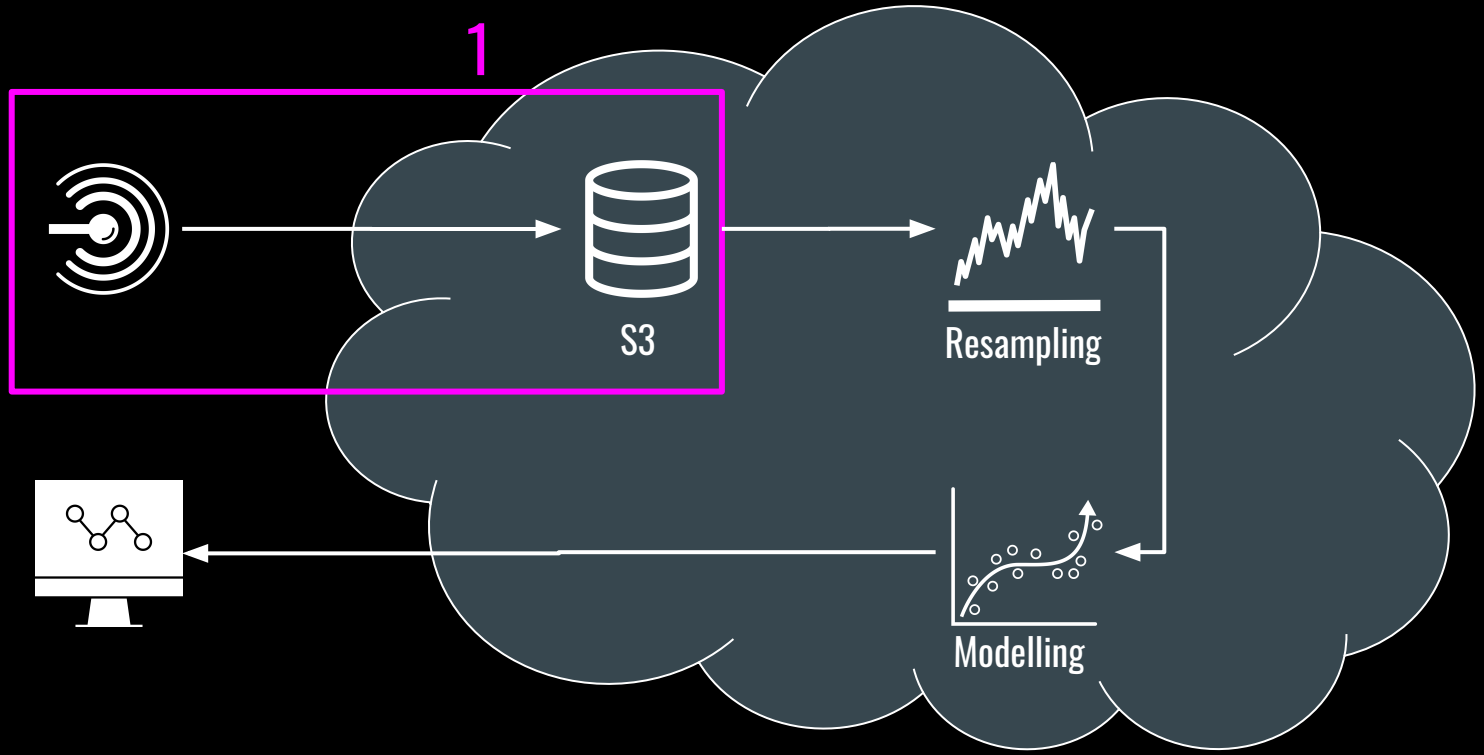
# 3 MODELLING IoT SENSOR DATA



# 4 SCALING AND AUTOMATING

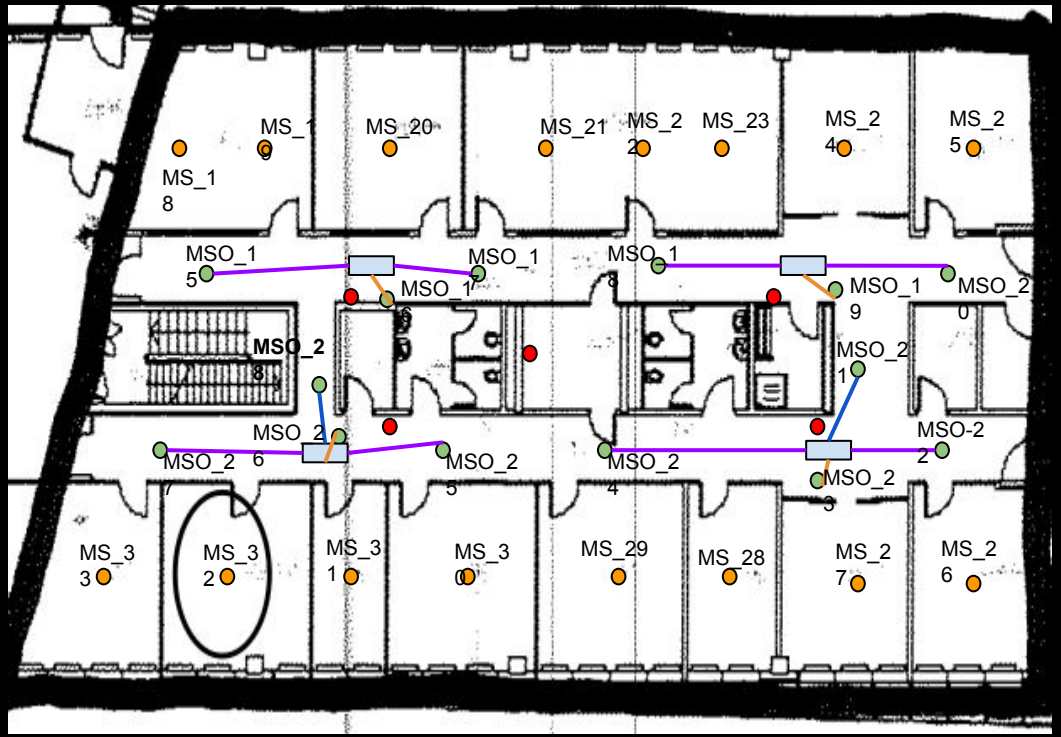


# 1 SMART OFFICE IoT SETUP



# SETUP

- 4000m$^2$ office space
- 2 floors
- 36 rooms
- 215 sensors
- 3 months of data



# SENSORS

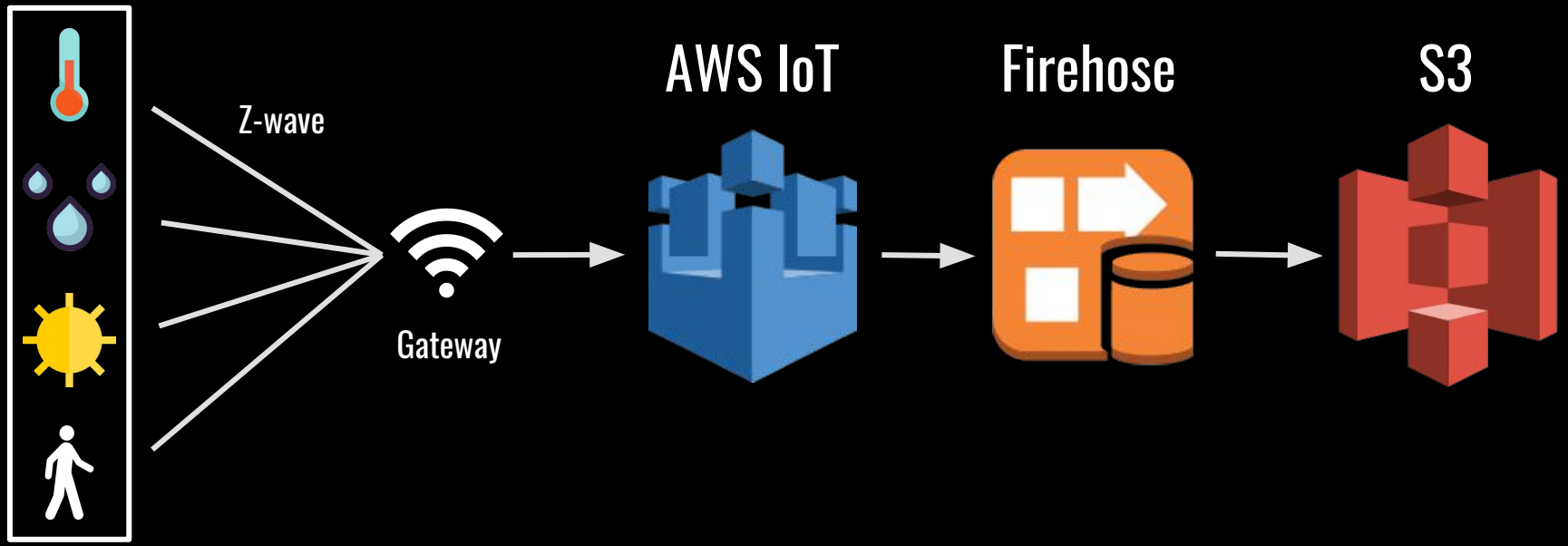
Temperature

Luminosity

Humidity

Motion

# DATA FLOW

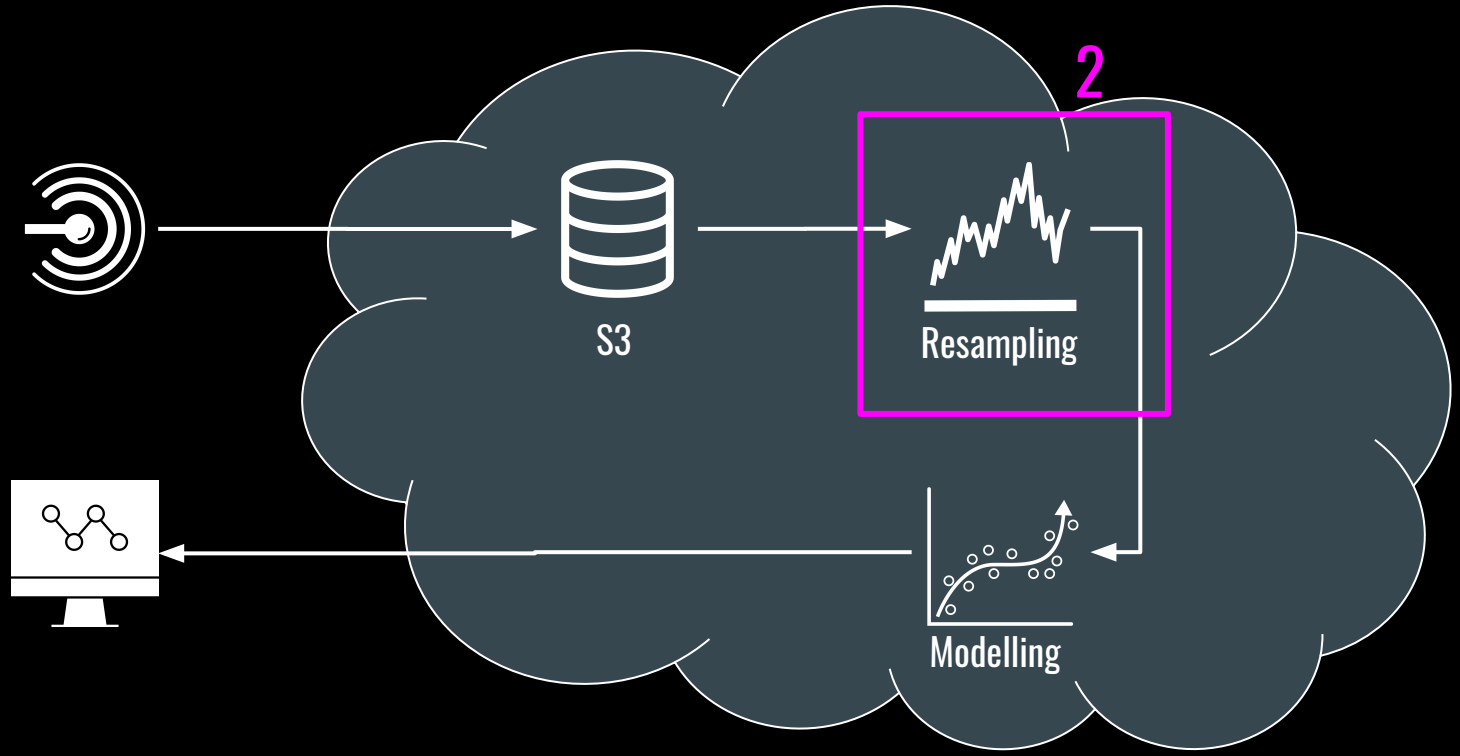


# RAW DATA

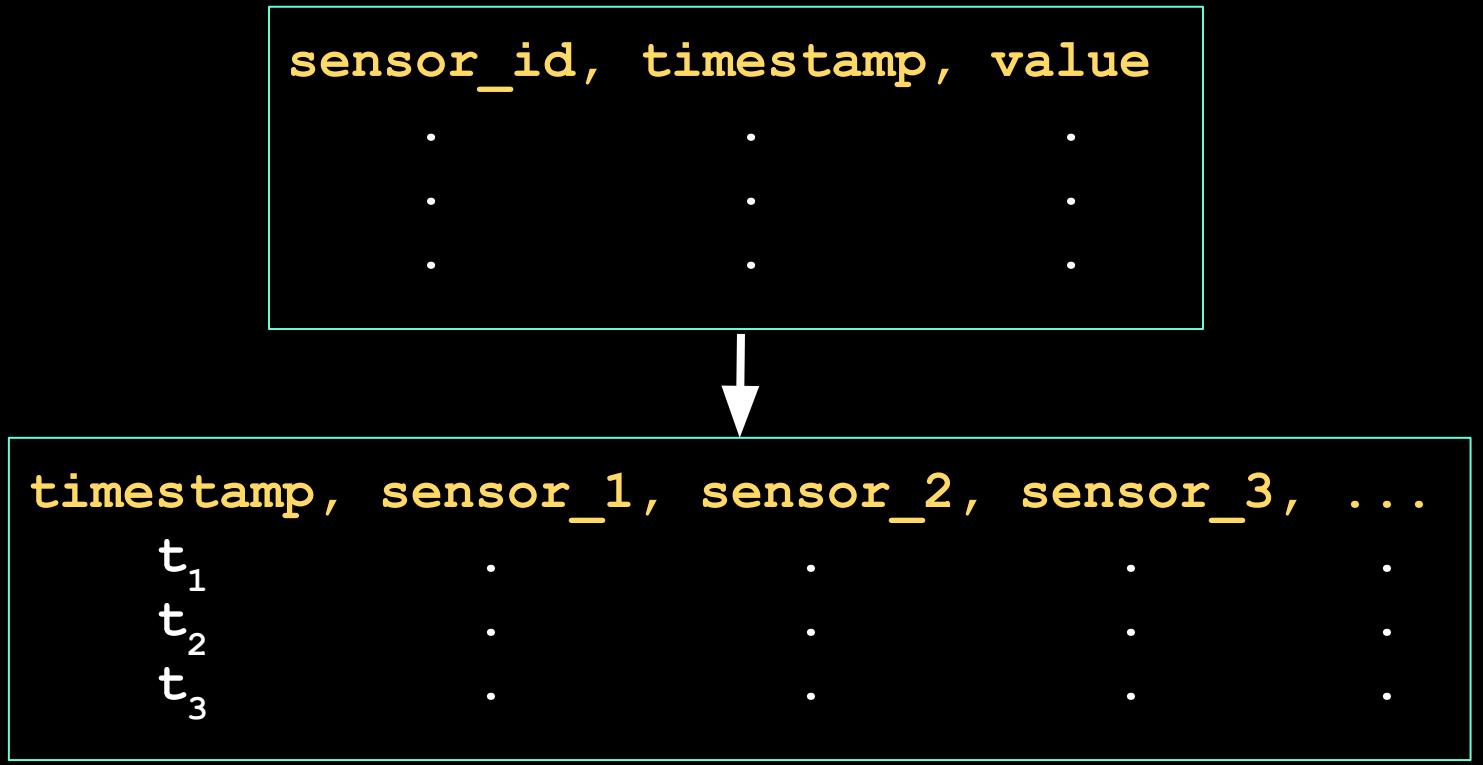
**s3://raw_data/2016/08/14/**

```
sensor_id='8514-294-220-250-m', timestamp=1471020341000, value=1.0,
sensor_id='7513-133-195-250-m', timestamp=1471020345000, value=0.0,
sensor_id='8514-294-220-250-t', timestamp=1471132284000, value=25.0,
sensor_id='7513-133-195-250-t', timestamp=1471132306000, value=23.9,
sensor_id='7513-133-195-250-t', timestamp=1471132316000, value=23.0,
sensor_id='7505-277-172-250-m', timestamp=1471132385000, value=0.0,
sensor_id='7573-230-160-250-l', timestamp=1471132448000, value=4.0,
                                .
                                .
                                .
```

# 2 RESAMPLING IoT SENSOR DATA



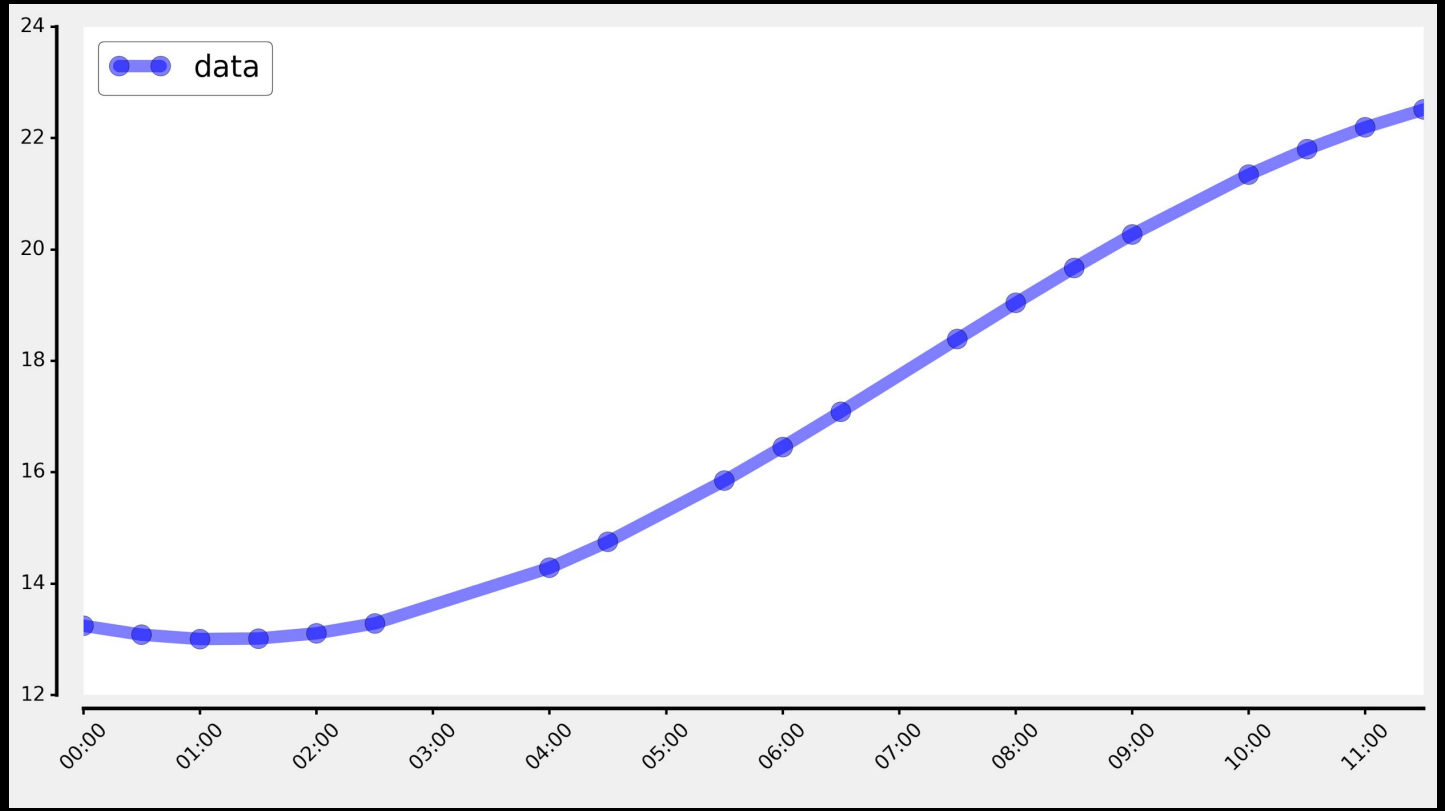
# OBSERVATIONS to TIME SERIES



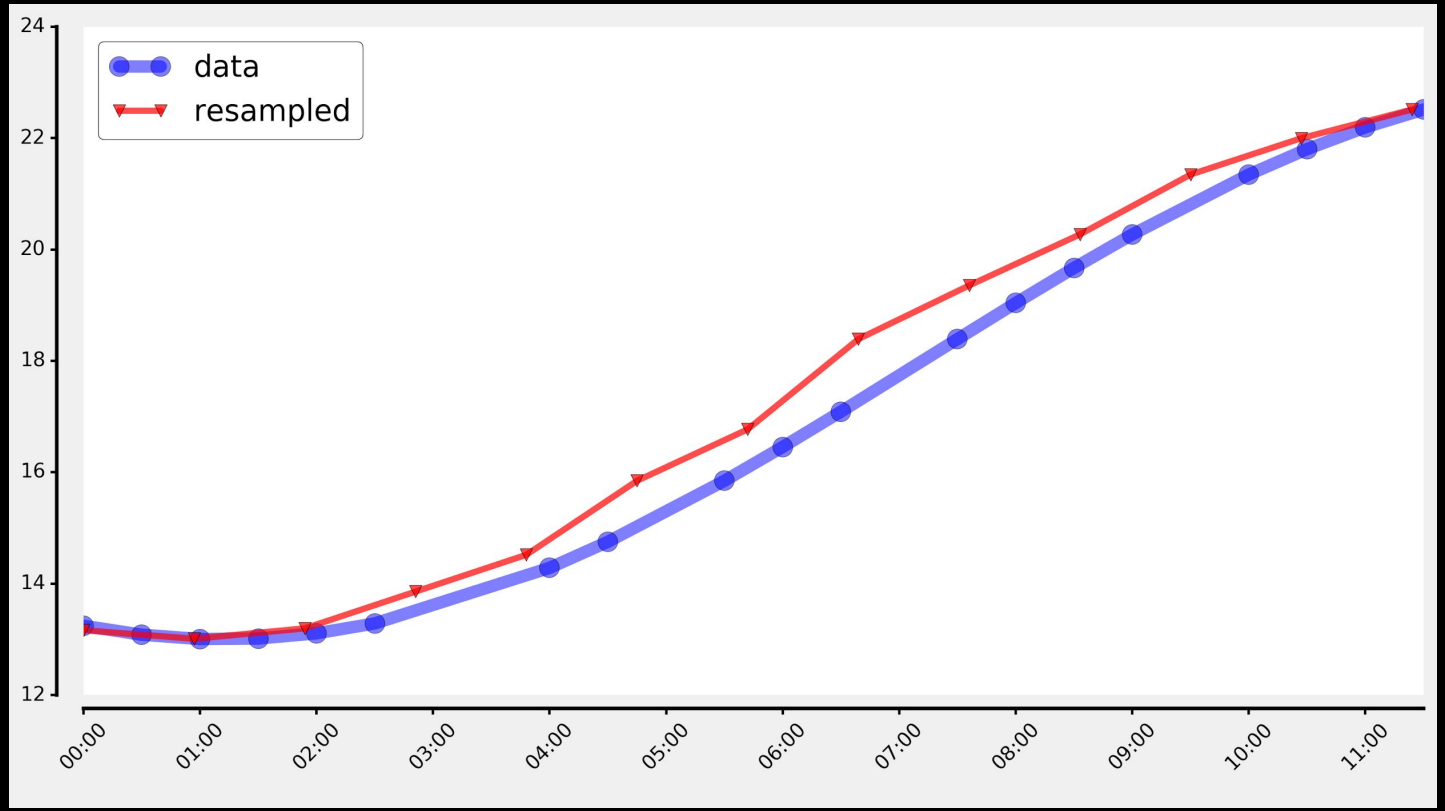
# RESAMPLE? PANDAS!

```
import pandas as pd

df = df.resample(df, '5min', 'mean')
```

# RESAMPLING


data
24
22
20
18
16
14
12
00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00

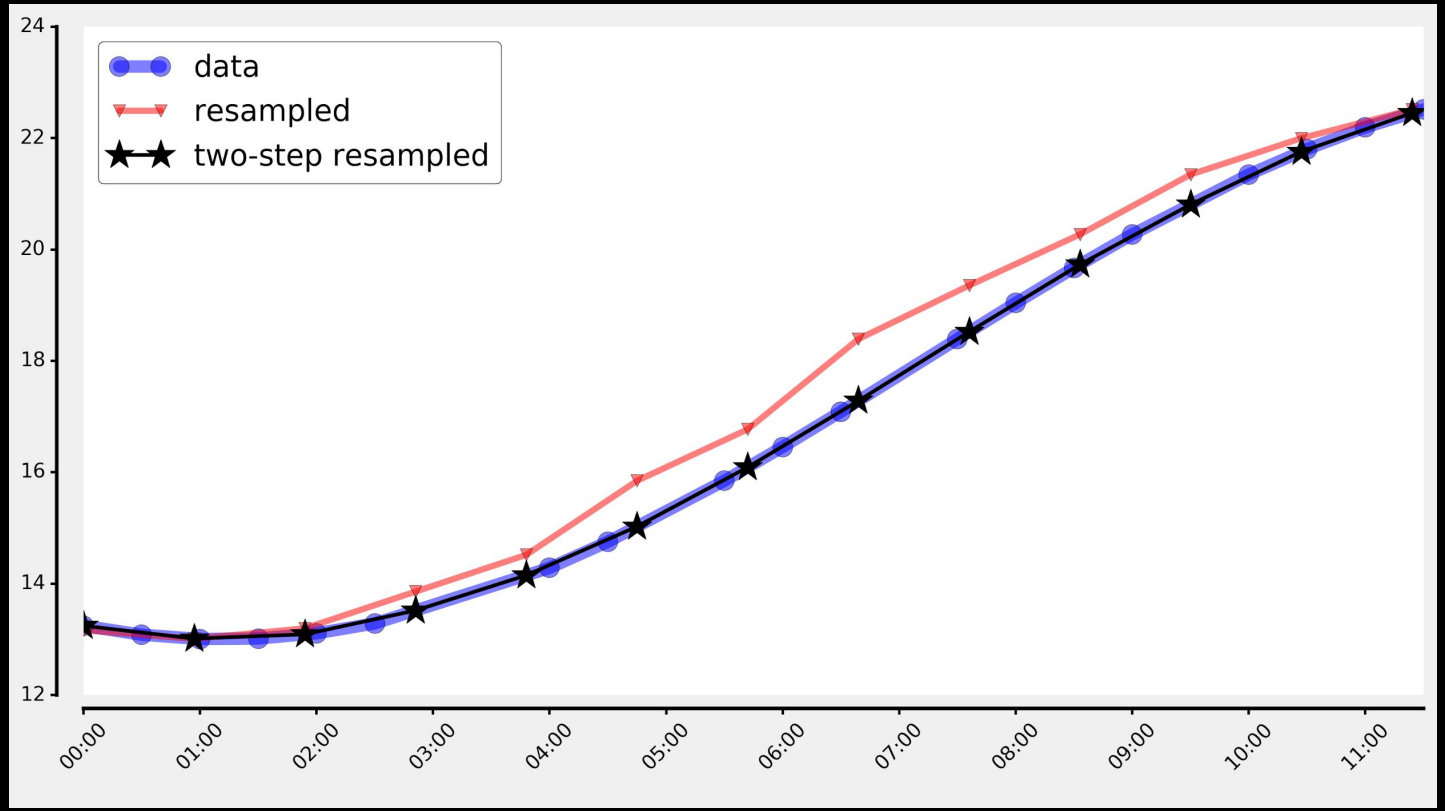
# RESAMPLING


data
resampled
24
22
20
18
16
14
12
00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00

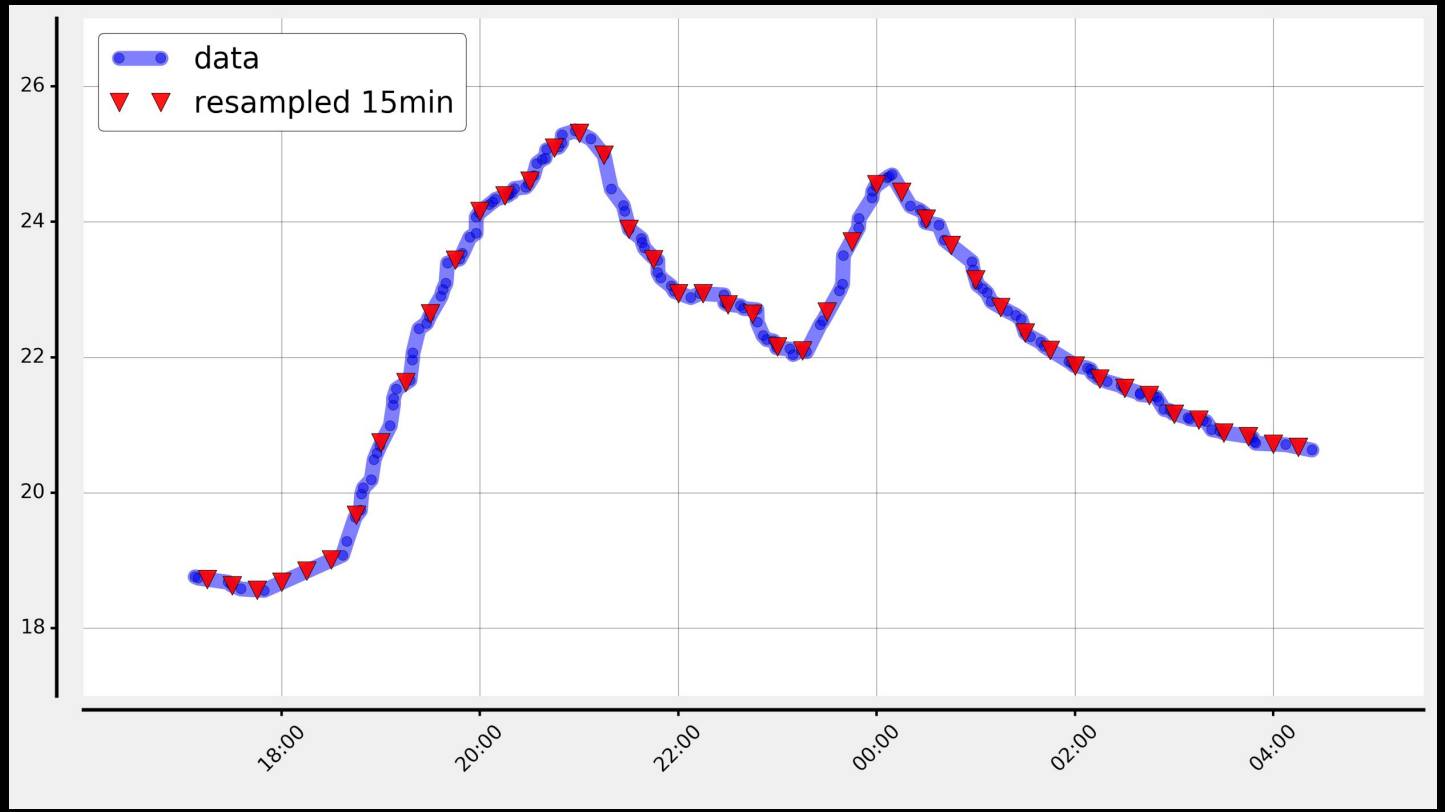
# TWO-STEP RESAMPLING PROCESS

```
# 1) upsample to high frequency
tmp = data.resample('1min', 'mean').interpolate()


# 2) downsample to desired frequency
resampled = tmp.resample('15min', 'ffill')
```

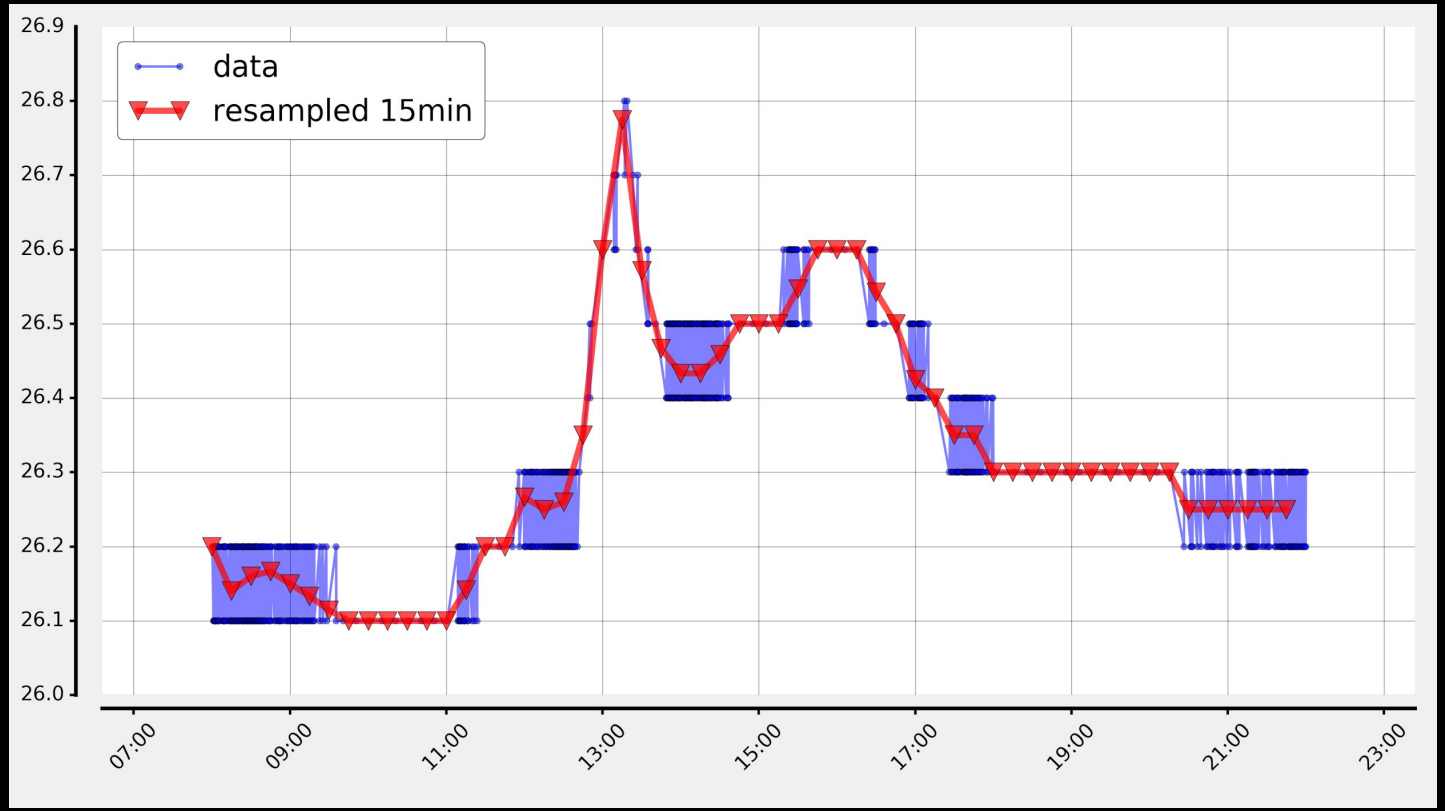
# RESAMPLING


data
resampled
two-step resampled
24
22
20
18
16
14
12
00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00

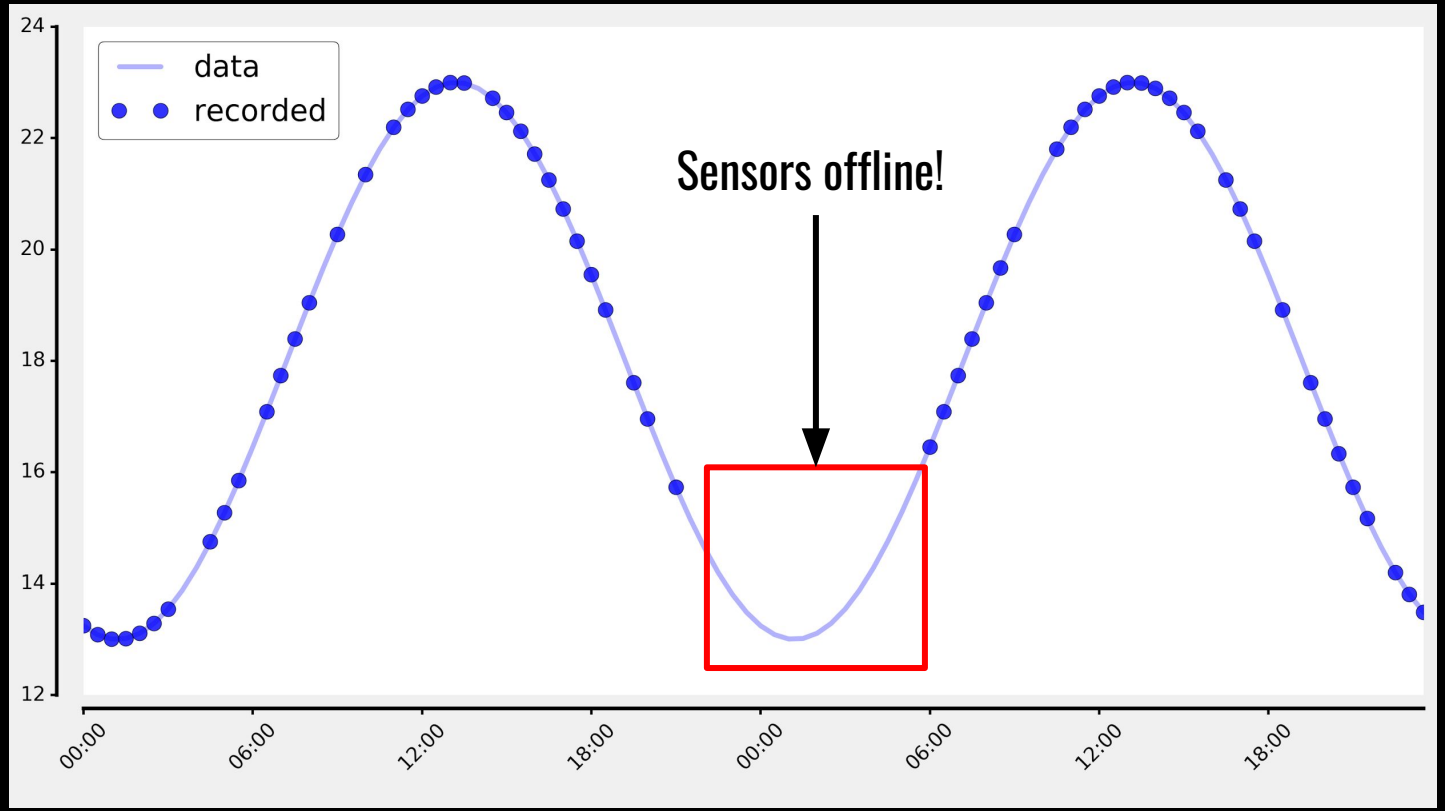
# RESAMPLING


data
resampled 15min
26
24
22
20
18
18:00
20:00
22:00
00:00
02:00
04:00

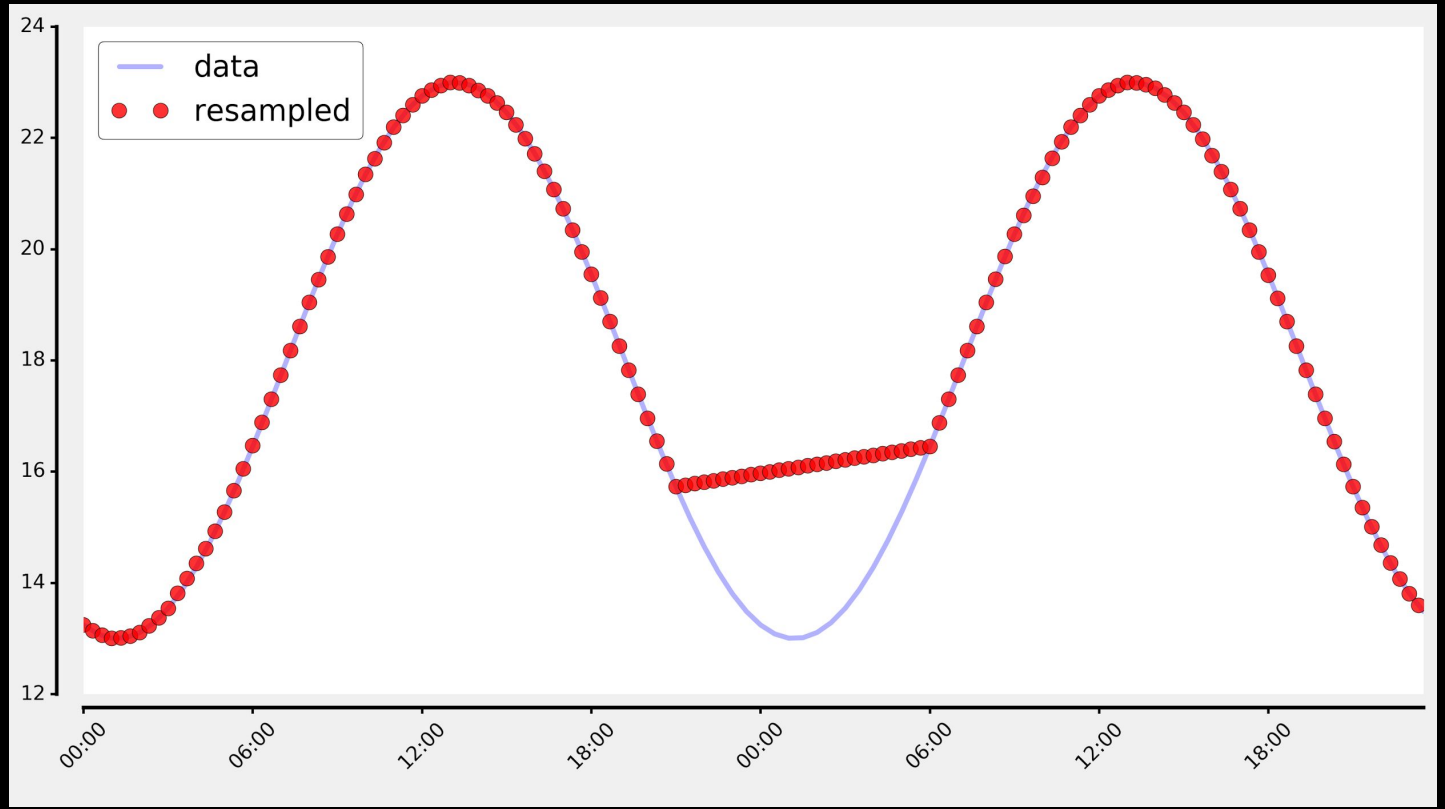
# RESAMPLING


data
resampled 15min
26.9
26.8
26.7
26.6
26.5
26.4
26.3
26.2
26.1
26.0
07:00
09:00
11:00
13:00
15:00
17:00
19:00
21:00
23:00

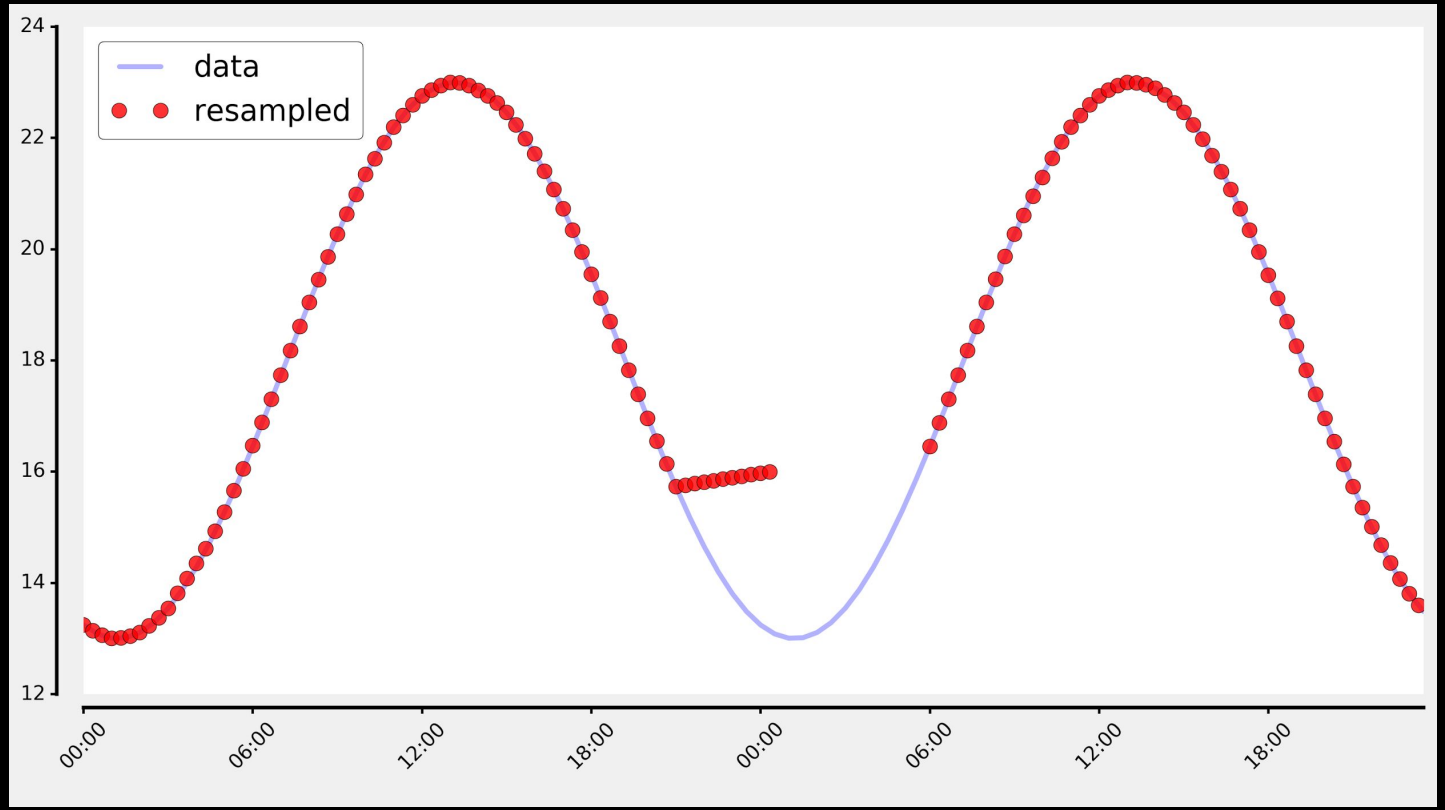
# PROBLEM: OFFLINE SENSORS



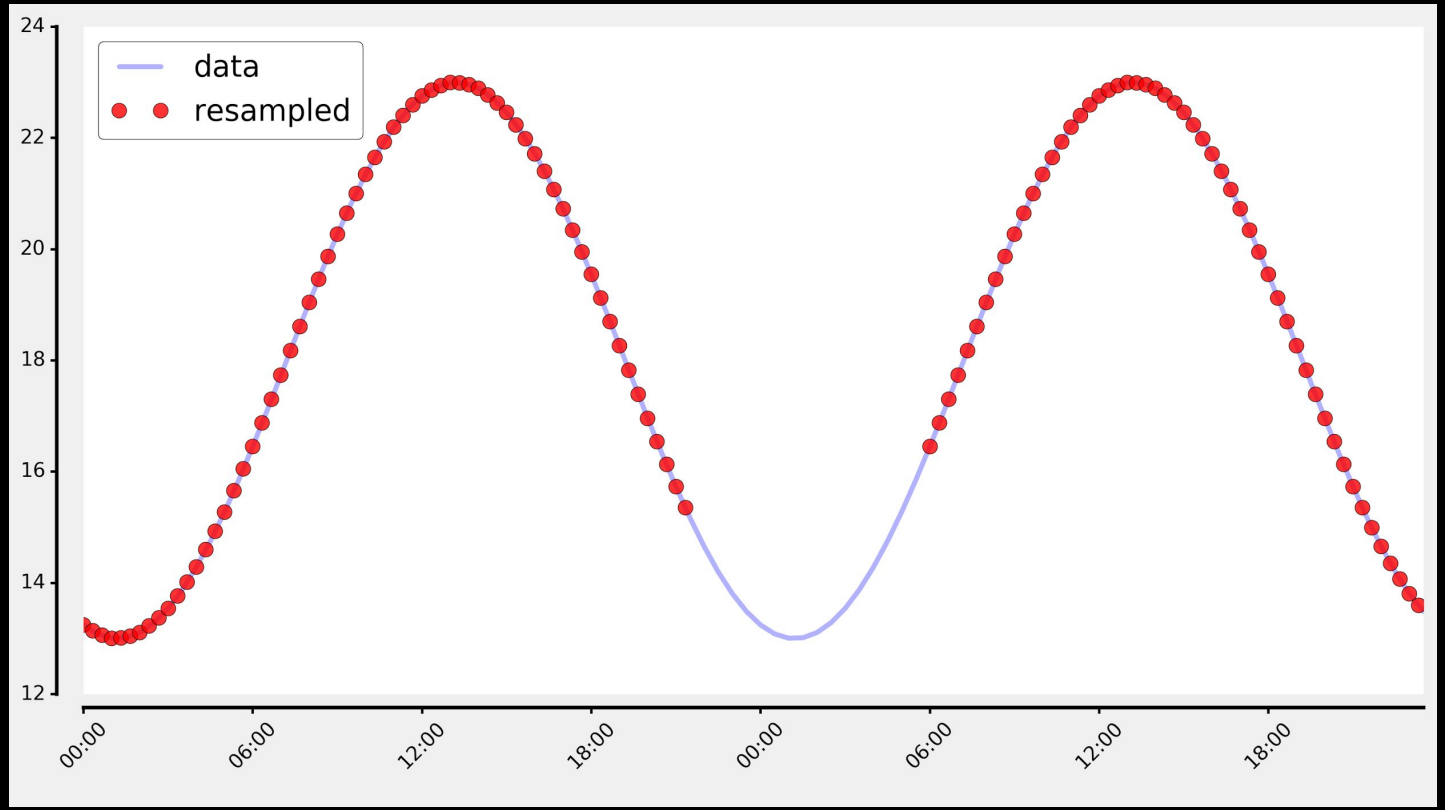
# OFFLINE SENSORS



# OFFLINE SENSORS



# OFFLINE SENSORS



# TIME SERIES

**2016/06/20**

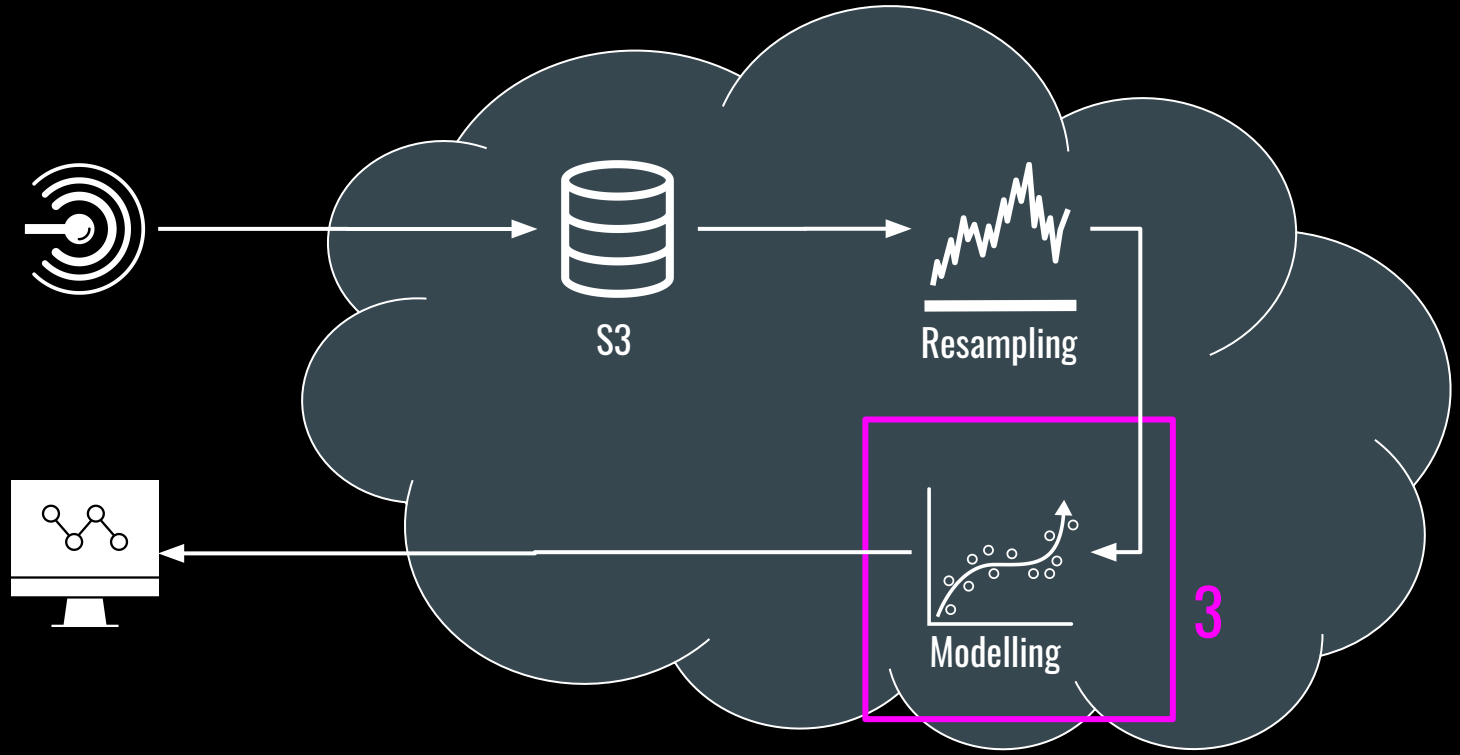
| | occ-7115 |
|---|---|
| 2016-06-20 11:00:00 | 0.000000 |
| 2016-06-20 11:05:00 | 0.000000 |
| 2016-06-20 11:10:00 | 0.056667 |
| 2016-06-20 11:15:00 | 0.770000 |
| 2016-06-20 11:20:00 | 0.766667 |
| 2016-06-20 11:25:00 | 0.880000 |
| 2016-06-20 11:30:00 | 0.060000 |
| 2016-06-20 11:35:00 | 0.446667 |
| 2016-06-20 11:40:00 | 0.790000 |
| 2016-06-20 11:45:00 | 0.860000 |
| 2016-06-20 11:50:00 | 0.800000 |
| 2016-06-20 11:55:00 | 0.440000 |
| 2016-06-20 12:00:00 | 1.000000 |

# TIME SERIES

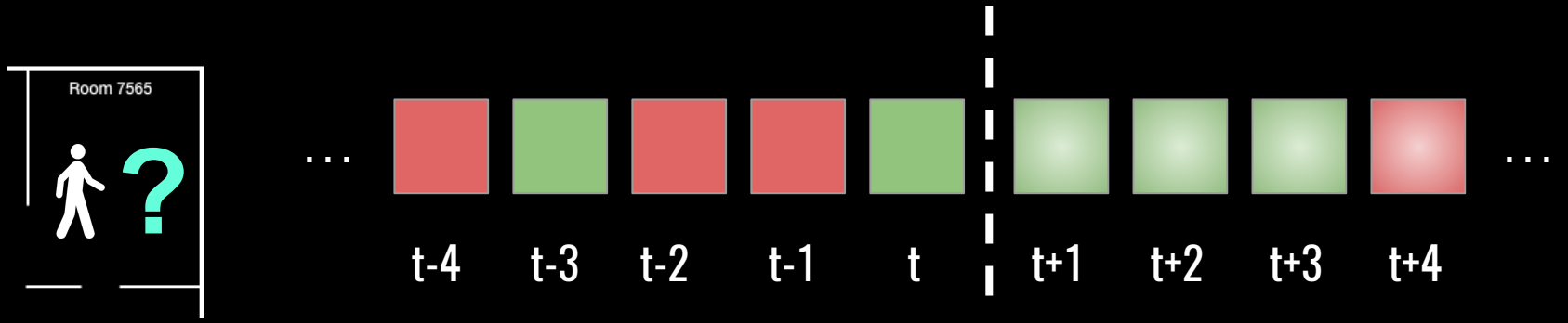
**2016/06/20**

| | occ-7115 | tmp-7115 | |
|---|---|---|---|
| 2016-06-20 11:00:00 | 0.000000 | 23.100000 | |
| 2016-06-20 11:05:00 | 0.000000 | 23.120000 | |
| 2016-06-20 11:10:00 | 0.056667 | 23.566000 | |
| 2016-06-20 11:15:00 | 0.770000 | 23.800000 | |
| 2016-06-20 11:20:00 | 0.766667 | 23.810000 | |
| 2016-06-20 11:25:00 | 0.880000 | 23.810000 | . . . |
| 2016-06-20 11:30:00 | 0.060000 | 23.810000 | |
| 2016-06-20 11:35:00 | 0.446667 | 23.950000 | |
| 2016-06-20 11:40:00 | 0.790000 | 23.950000 | |
| 2016-06-20 11:45:00 | 0.860000 | 23.955500 | |
| 2016-06-20 11:50:00 | 0.800000 | 23.960000 | |
| 2016-06-20 11:55:00 | 0.440000 | 23.980000 | |
| 2016-06-20 12:00:00 | 1.000000 | 24.111200 | |

# 3 MODELLING IoT SENSOR DATA

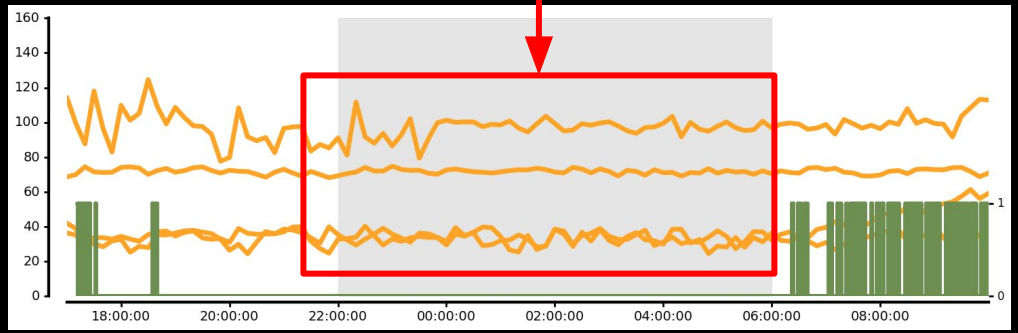


# PREDICTING OCCUPANCY



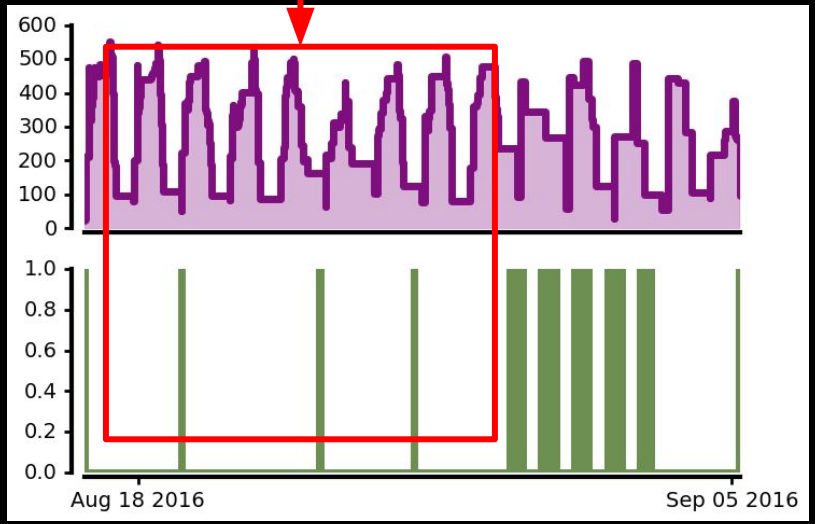
# MOTIVATION

*Lights are left on at night when nobody is present*



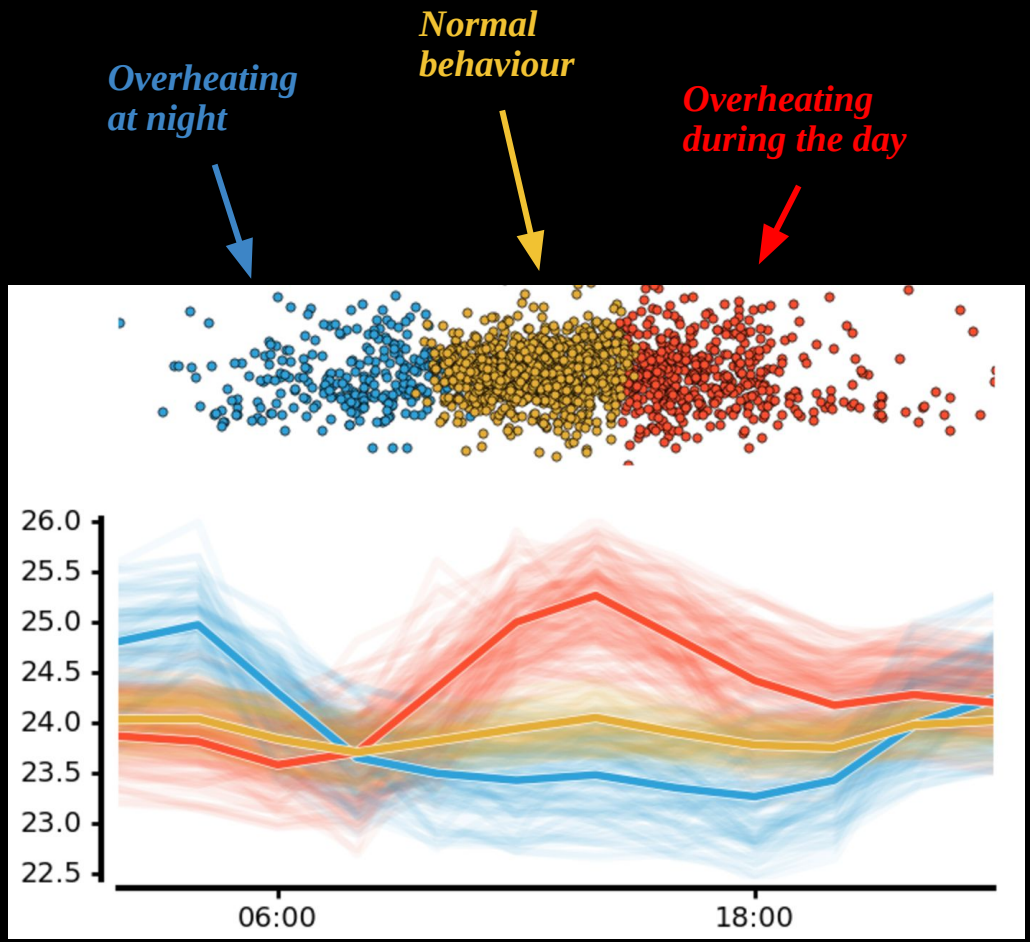

***Light intensity*** ***Occupancy***

*Rooms are being cooled at full intensity when nobody is present*

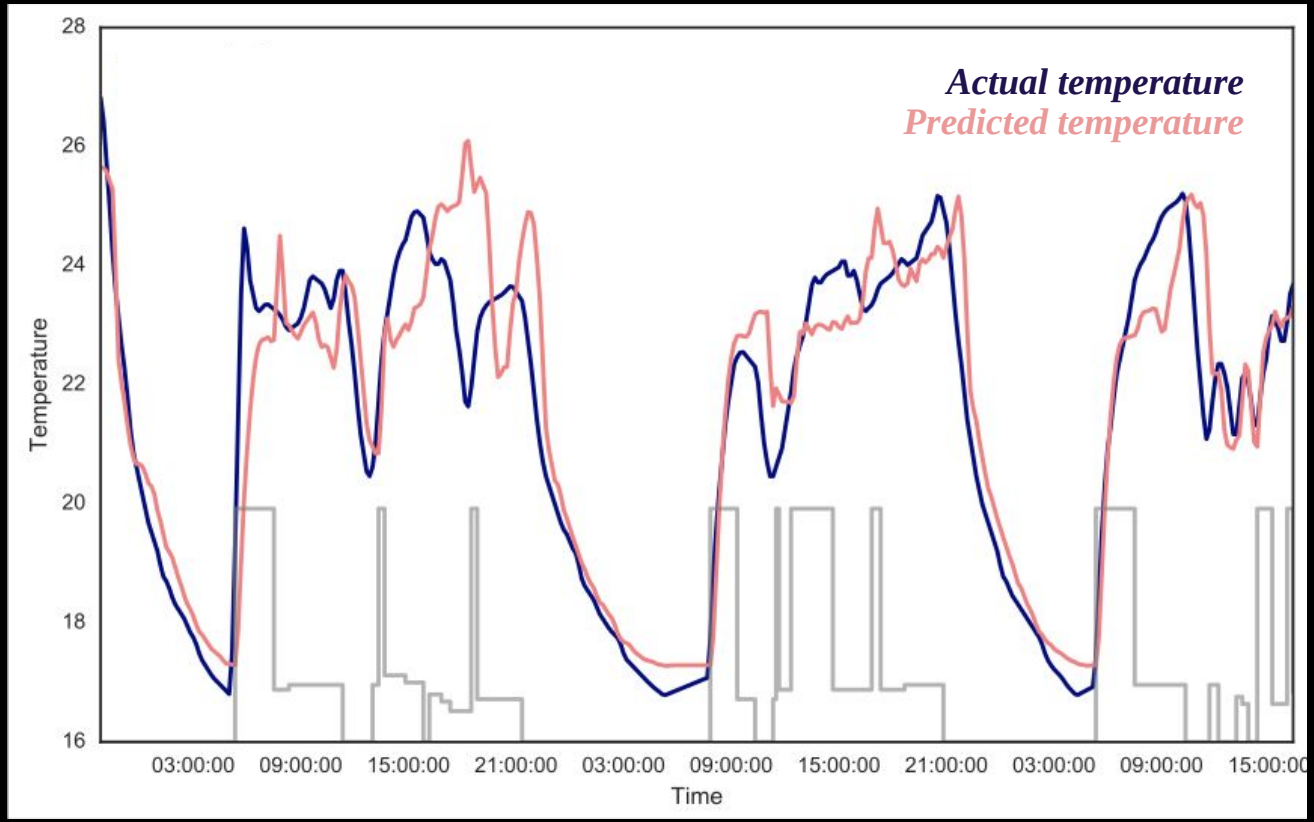



***Ventilation ($m^3/h$)*** ***Occupancy***

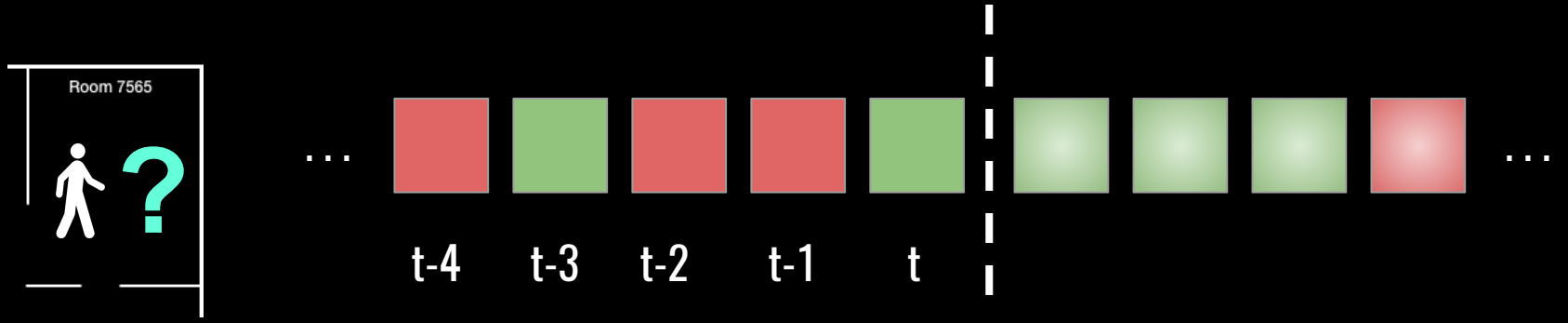
# MOTIVATION


Overheating at night
Normal behaviour
Overheating during the day
26.0
25.5
25.0
24.5
24.0
23.5
23.0
22.5
06:00
18:00

# MOTIVATION


Actual temperature
Predicted temperature
Temperature
28
26
24
22
20
18
16
03:00:00
09:00:00
15:00:00
21:00:00
03:00:00
09:00:00
15:00:00
21:00:00
03:00:00
09:00:00
15:00:00
Time

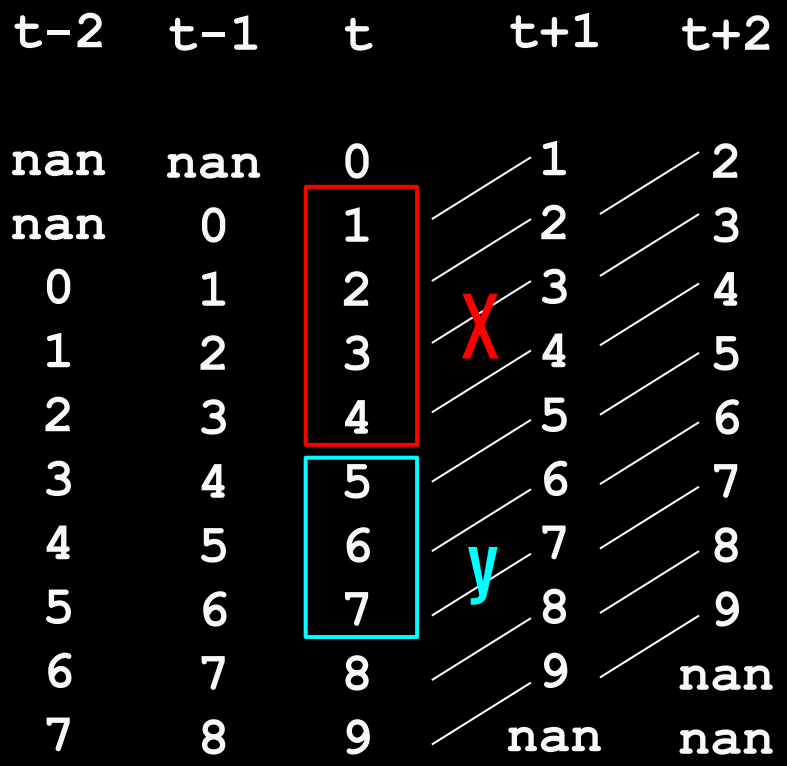
# MULTI-LABEL PREDICTION




$$[x_{t+s}, \ldots, x_{t+1}] = f(x_t, \ldots, x_{t-n+1}) + \epsilon$$

# TIME-SERIES TO SUPERVISED LEARNING

```
import pandas as pd

df = DataFrame()
df['t'] = range(10)
df['t+1'] = df.t.shift(-1)
df['t+2'] = df.t.shift(-2)
df['t-1'] = df.t.shift(1)
df['t-2'] = df.t.shift(2)
```



# TIME-SERIES TO SUPERVISED LEARNING

```
import pandas as pd

df = DataFrame()
df['t'] = range(10)
df['t+1'] = df.t.shift(-1)
df['t+2'] = df.t.shift(-2)
df['t-1'] = df.t.shift(1)
df['t-2'] = df.t.shift(2)
```

X

| t-2 | t-1 | t |
|---|---|---|
| nan | nan | 0 |
| nan | 0 | 1 |
| 0 | 1 | 2 |
| 1 | 2 | 3 |
| 2 | 3 | 4 |
| 3 | 4 | 5 |
| 4 | 5 | 6 |
| 5 | 6 | 7 |
| 6 | 7 | 8 |
| 7 | 8 | 9 |

y

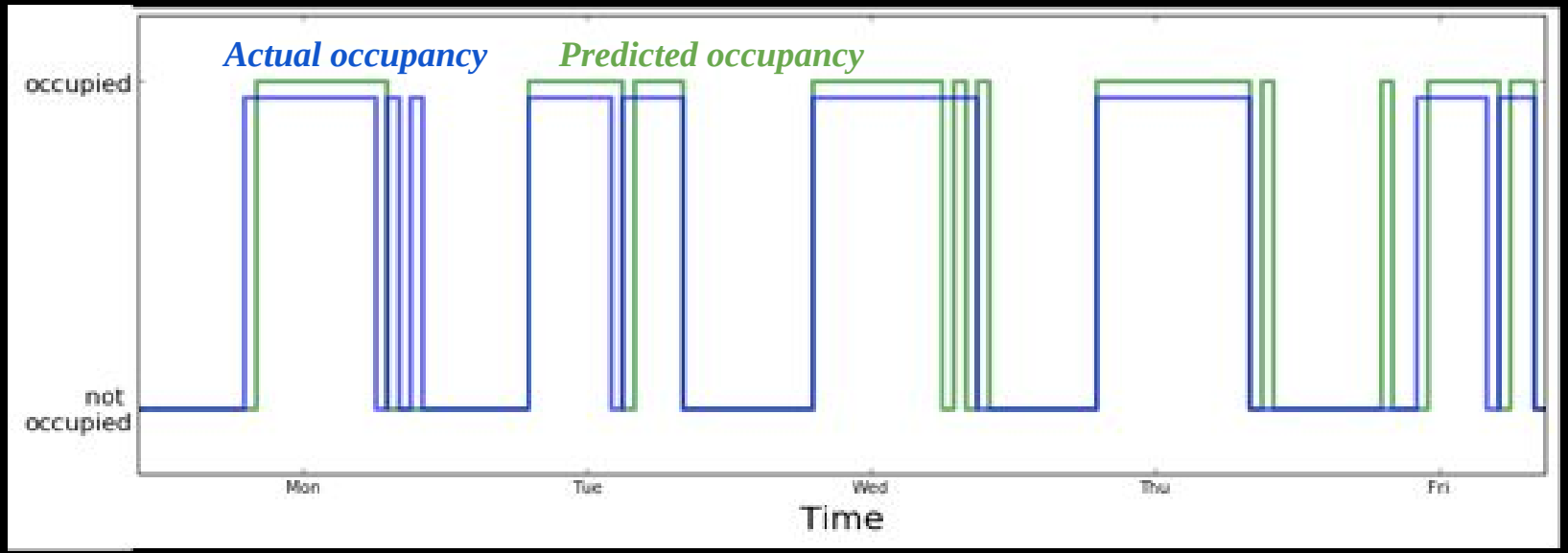
| t+1 | t+2 |
|---|---|
| 1 | 2 |
| 2 | 3 |
| 3 | 4 |
| 4 | 5 |
| 5 | 6 |
| 6 | 7 |
| 7 | 8 |
| 8 | 9 |
| 9 | nan |
| nan | nan |

# TIME-SERIES TO SUPERVISED LEARNING

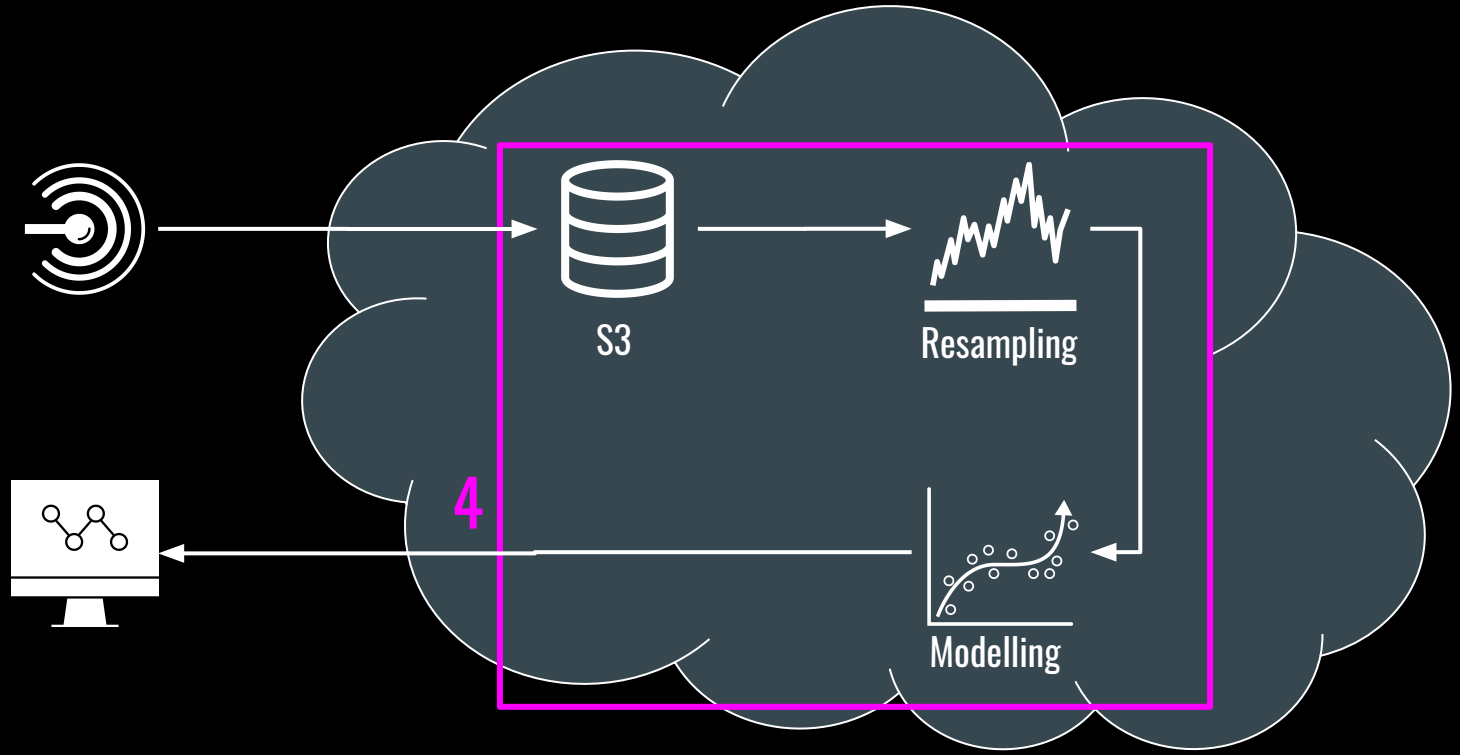
```
import pandas as pd

df = DataFrame()
df['t'] = range(10)
X, y = tseries_to_ml(
    df, bsteps=2, fsteps=2)


RandomForestClassifier().fit(X, y)
```

| X | | | y | |
|---|---|---|---|---|
| t-2 | t-1 | t | t+1 | t+2 |
| nan | nan | 0 | 1 | 2 |
| nan | 0 | 1 | 2 | 3 |
| 0 | 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | 4 | 5 |
| 2 | 3 | 4 | 5 | 6 |
| 3 | 4 | 5 | 6 | 7 |
| 4 | 5 | 6 | 7 | 8 |
| 5 | 6 | 7 | 8 | 9 |
| 6 | 7 | 8 | 9 | nan |
| 7 | 8 | 9 | nan | nan |

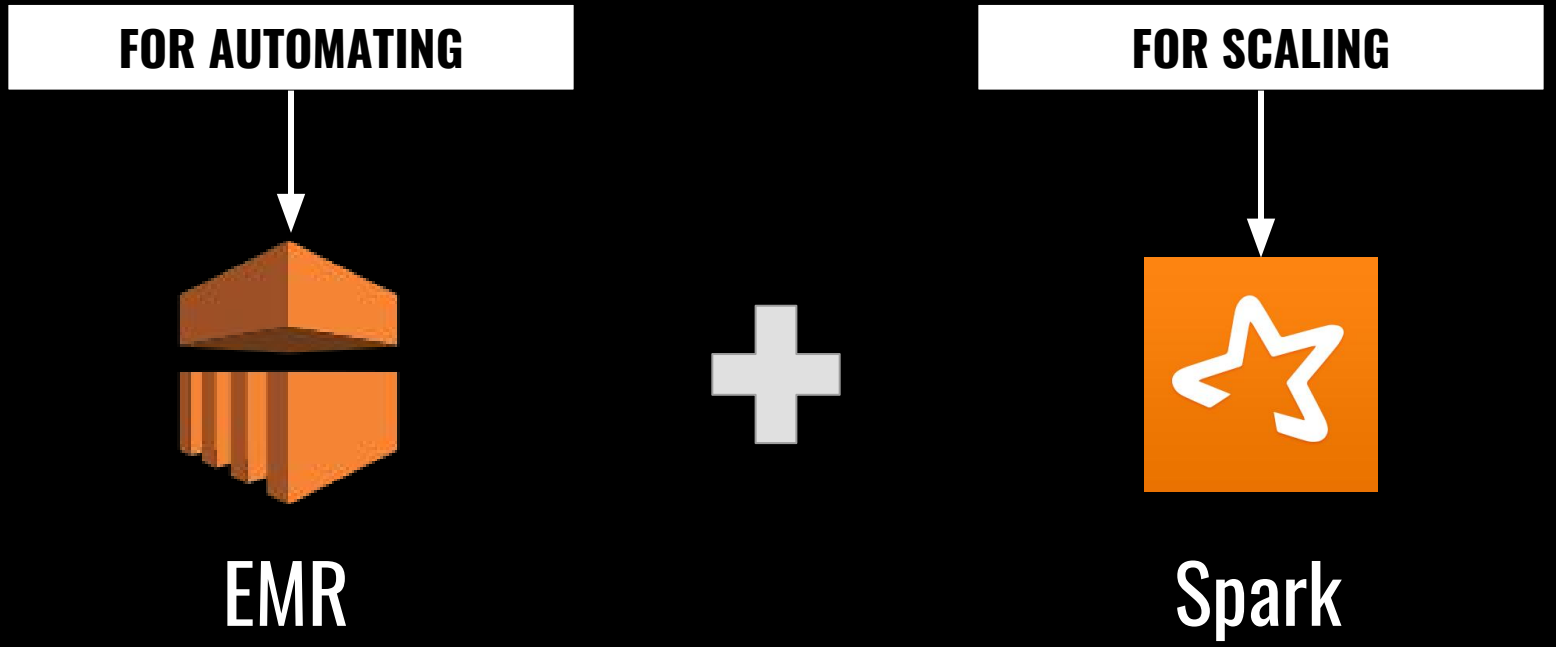
# OCCUPANCY PREDICTION



# 4 SCALING AND AUTOMATING



# SCALING AND AUTOMATING



# SAMPLE SPARK APP

```
from spark import SparkContext

def main():
    sc = SparkContext(appName='TrainModels')
    room_sensors = grouped_sensors(gby='room', type='motion')
    rdd = sc.parallelize(room_sensors)
    rdd.map(load_sensor_data) \
       .map(train_predictor)
    sc.stop()

if __name__ == '__main__':
    main()
```

```
[(room_08, [ms_12, ms_14, m_23]),
 (room_13, [ms_15, ms_18]), ...]
```

# SAMPLE EMR JOB

```
import boto3

emr = boto3.client('emr')

emr.run_job_flow(
    Name='MyCluster',
    ReleaseLabel='emr-5.4.0',
    Instances={
        'MasterInstanceType': 'm3.xlarge',
        'SlaveInstanceType': 'm3.xlarge',
        'InstanceCount': 5,
    },
    BootstrapActions=BOOTSTRAP_ACTIONS,
    Steps=JOB_STEPS)
```

# JOB STEPS

```
JOB_STEPS = [{
    'Name': 'Resample Timeseries',
    'HadoopJarStep': {
        'Jar': 'command-runner.jar',
        'Args': ['spark-submit',
                 '/home/hadoop/resample_tseries.py'],
    },
},{
    'Name': 'Train Occupancy Models',
    'HadoopJarStep': {
        'Jar': 'command-runner.jar',
        'Args': ['spark-submit',
                 '/home/hadoop/train_occupancy.py'],
    },
}]
```

Resamples data per sensor

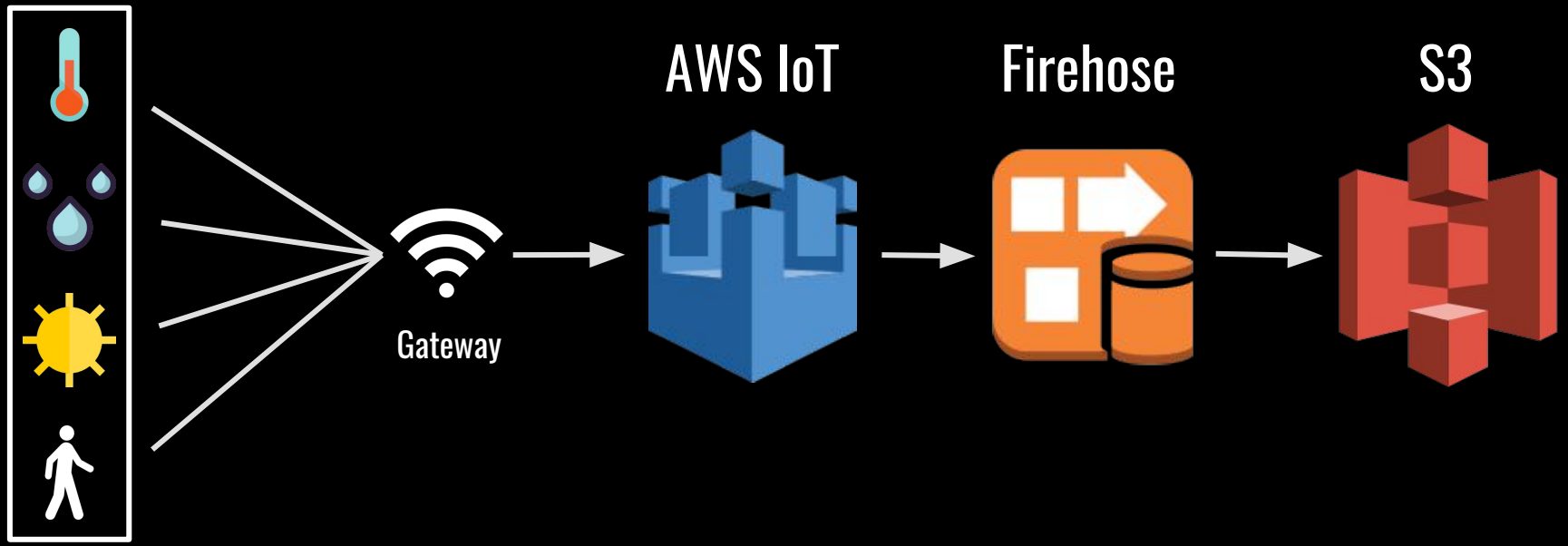
Trains predictor per room

# BOOTSTRAP ACTIONS

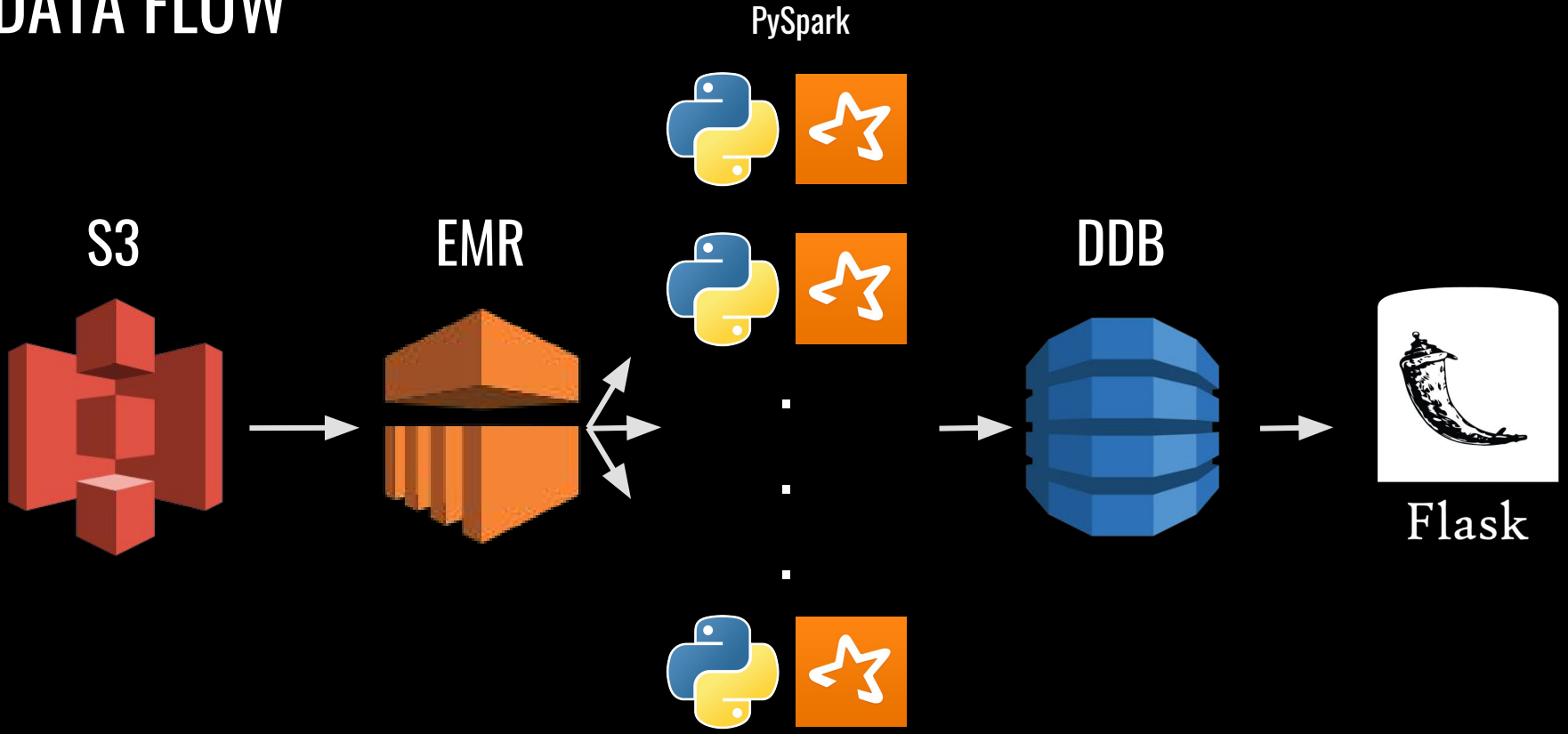
```
BOOTSTRAP_ACTIONS = [{
    'Name': 'Install Python Libraries',
    'ScriptBootstrapAction': {
        'Path': 's3://path/to/code/install_libraries.sh',
        'Args': []
    }
}]
```

```
pip install numpy pandas scikit-learn
```

# DATA FLOW



# DATA FLOW



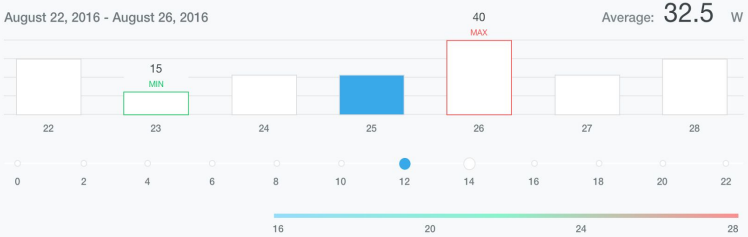
SMART OFFICE: MUC AIRPORT

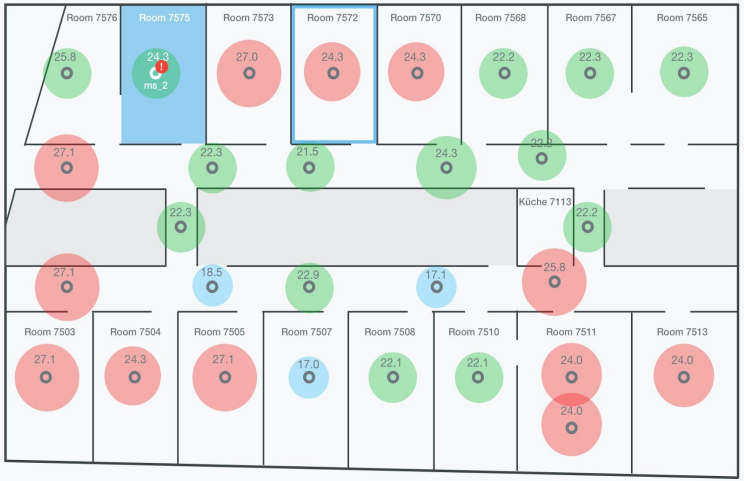
Jörg Weber

7th Floor | Occupancy | Lighting | Temperature

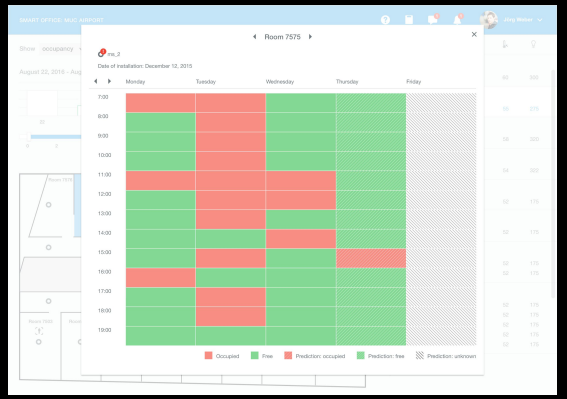
August 22, 2016 - August 26, 2016

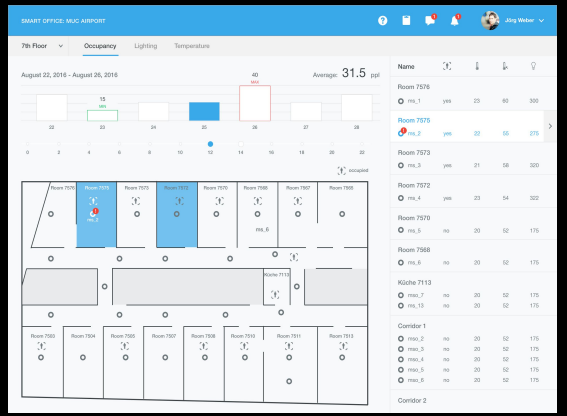
Average: 32.5 W




| Name | | | | |
|---|---|---|---|---|
| **Room 7576** | | | | |
| ms_1 | yes | 23 | 60 | 300 |
| **Room 7575** | | | | |
| ms_2 | yes | 22 | 55 | 275 |
| **Room 7573** | | | | |
| ms_3 | yes | 21 | 58 | 320 |
| **Room 7572** | | | | |
| ms_4 | yes | 23 | 54 | 322 |
| **Room 7570** | | | | |
| ms_5 | no | 20 | 52 | 175 |
| **Room 7568** | | | | |
| ms_6 | no | 20 | 52 | 175 |
| **Küche 7113** | | | | |
| mso_7 | no | 20 | 52 | 175 |
| ms_13 | no | 20 | 52 | 175 |
| **Corridor 1** | | | | |
| mso_2 | no | 20 | 52 | 175 |
| mso_3 | no | 20 | 52 | 175 |
| mso_4 | no | 20 | 52 | 175 |
| mso_5 | no | 20 | 52 | 175 |
| mso_6 | no | 20 | 52 | 175 |
| **Corridor 2** | | | | |



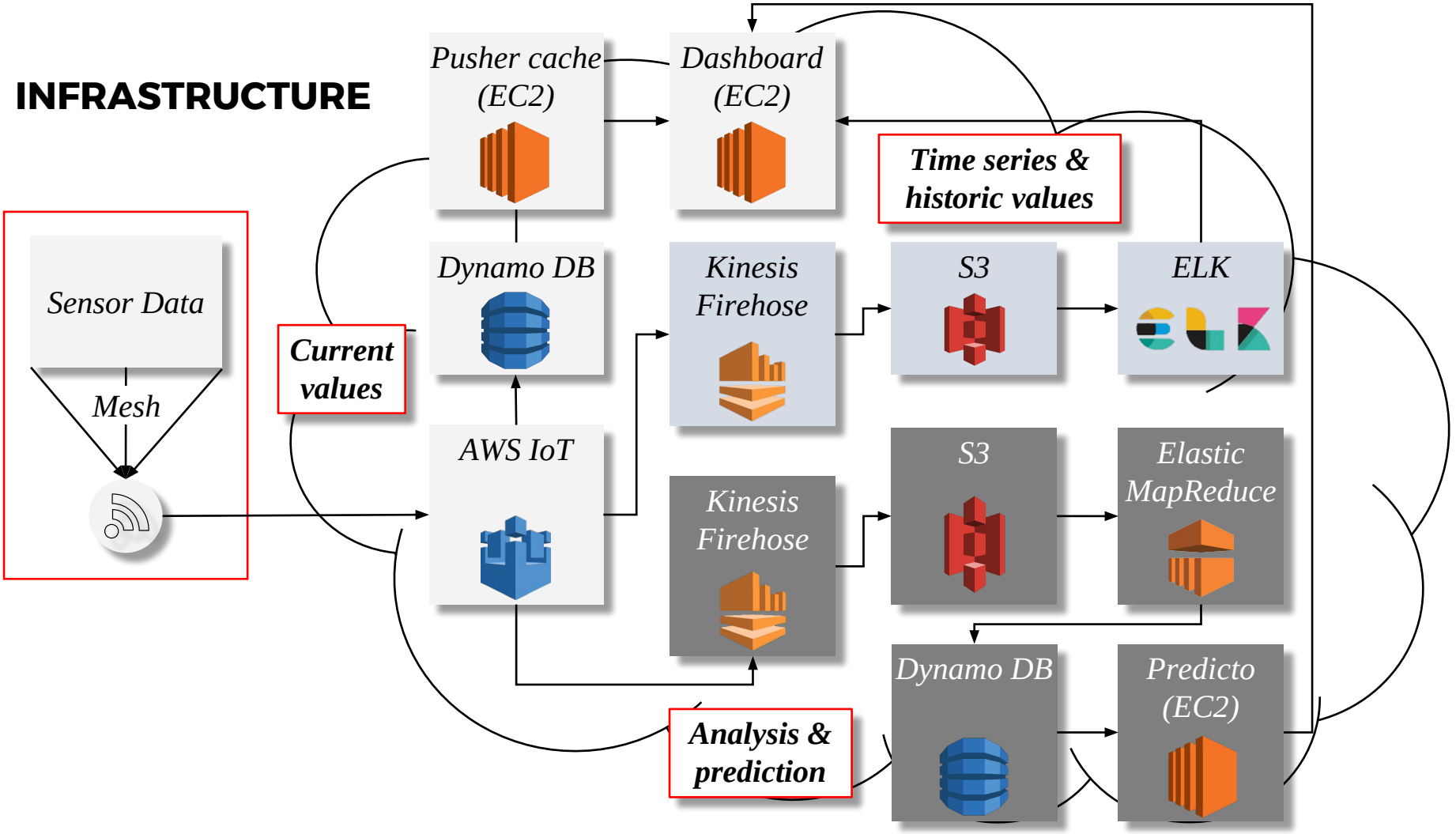

www.wattx.io

Rafael Schultze-Kraft | Data Scientist | rafael@wattx.io

# OUTLOOK

- Going Real-time: Stream Processing
- Going Big: SparkML vs Scikit-Learn
- Modelling: MCMC, LSTMs, ...
- Closing the Loop: Control of an Actuator

# INFRASTRUCTURE



# OCCUPANCY PREDICTION