Feeding Realworld data into Prometheus Microservice Application on Kubernetes

Burkhard Noltensmeier, CEO teuto.net

Burkhard Noltensmeier

teutoStack

kubernetes
by Google

OWL (Ostwestfalen Lippe)

Smartfactory OWL

Fraunhofer IOSB

Getting as Sense of what is going on !

and

aggregate Information with context in a distributed world


4

Getting a Sense ?

Scattered Locations

Multitude of protocols

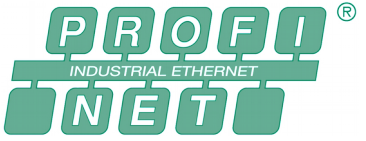
Missing Context

Overwelming Information

Modbus / Profinet

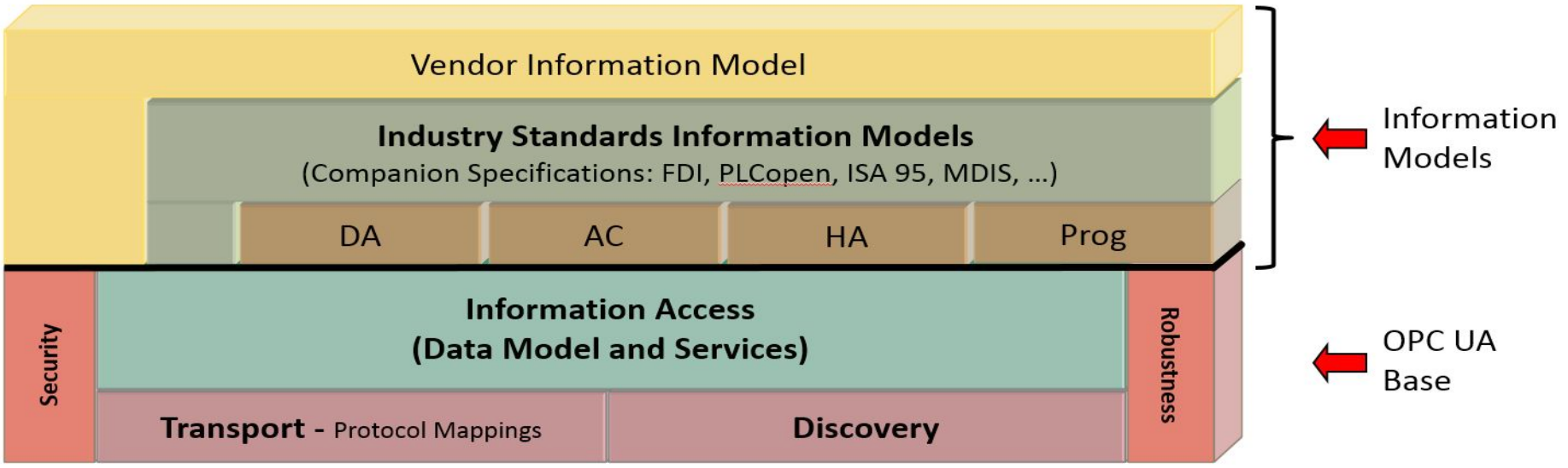
Binary Protocol

Needs cusomized exporter to add context



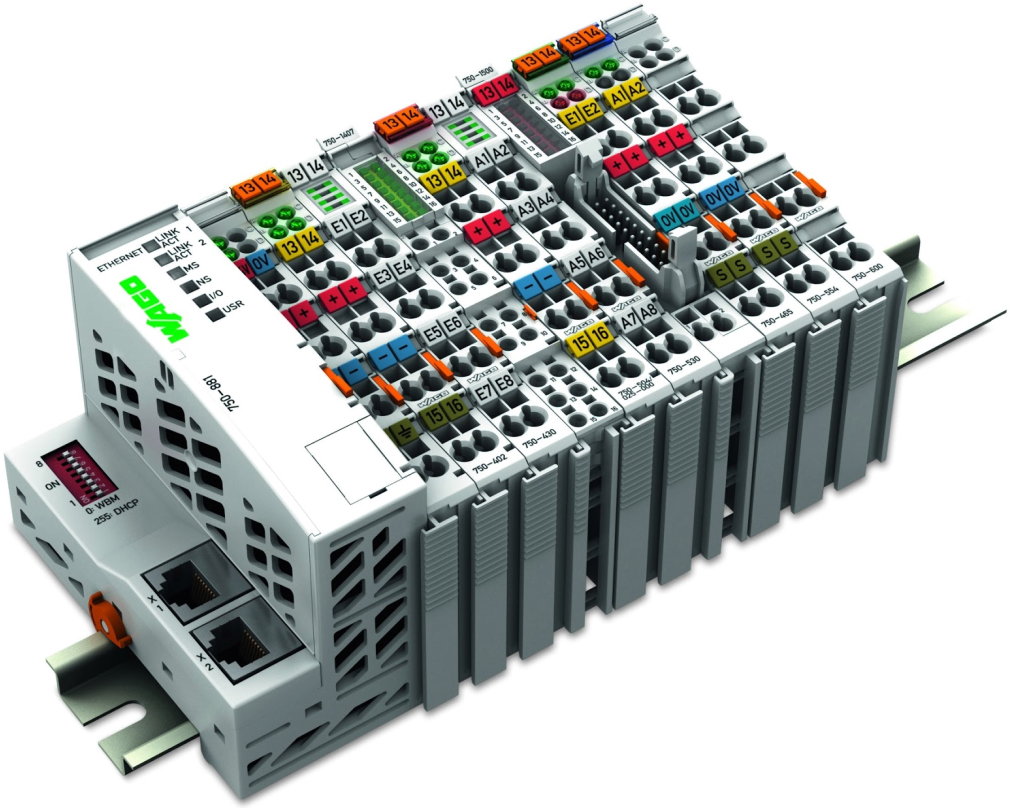
e.g. electricity meter sdm630

- Modbus RTU (Serial)
- Some context
- Serialnumber, volt, phase, wattage



Complex, High Level (XML)

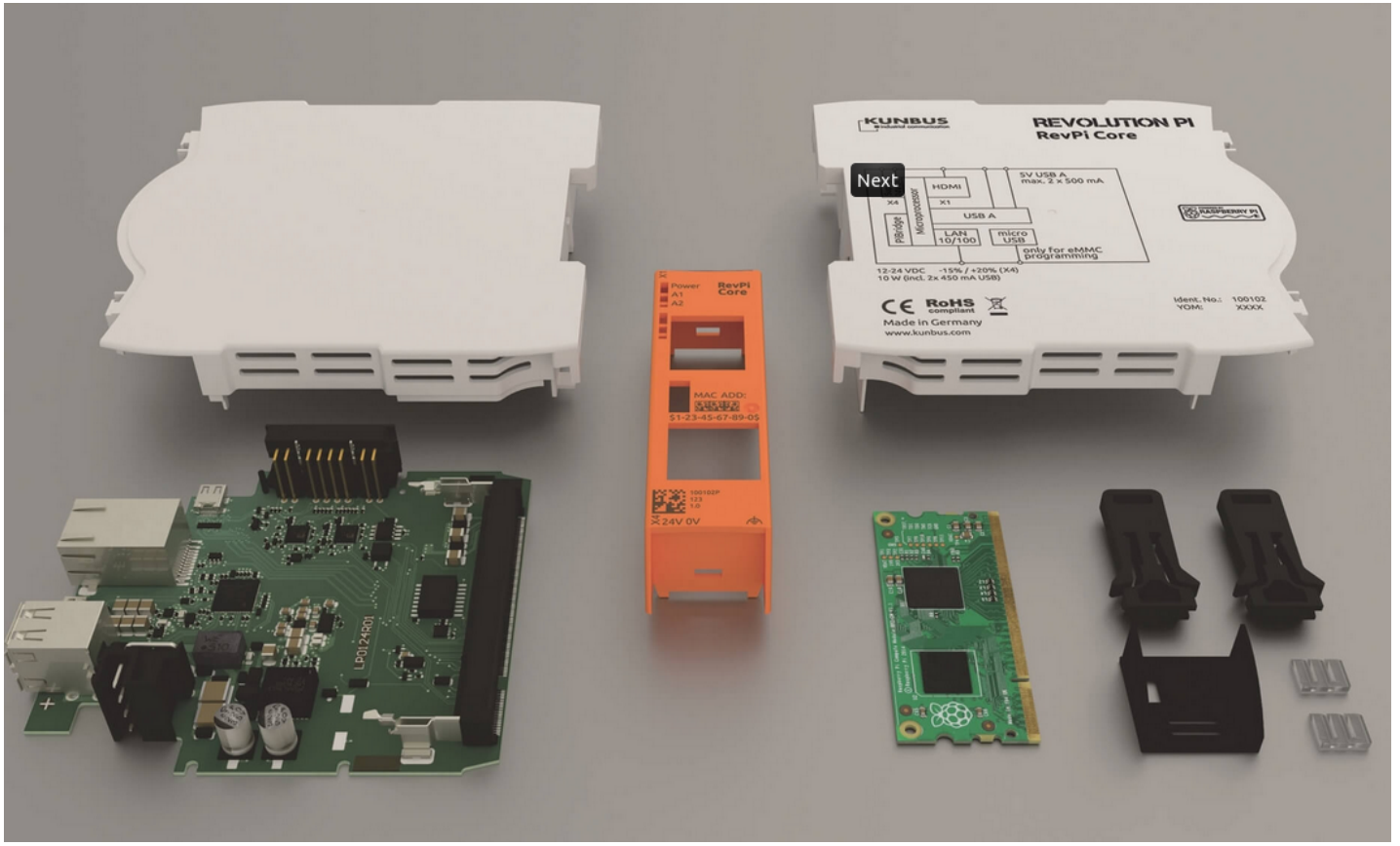
e.g. PLC Controller Wago



Edge Devices for Protcol Conversion

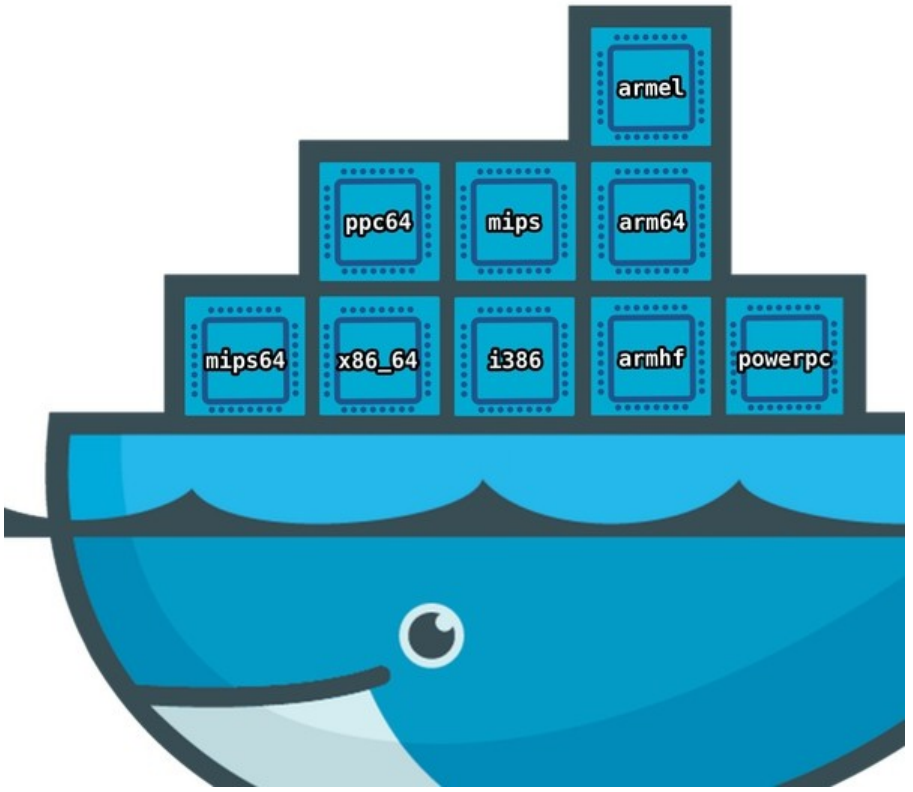
- Kunbus RevolutionPI

Edge Device Raspberry PI (of course)



How to do Continous Integration (for ARM)

- https://github.com/multiarch/qemu-user-static
- Qemu ARM emulation for x86
- building x86 and arm container in one CI Run



GitLab

Howto do Remote Management

- Still Struggling
- Eliot looks promising

https://github.com/ernoaapa/eliot

- Or even kubernetes federation on Raspberry PI ?

How to build the Platform?

- Cloud Native to the Rescue
- Taking Prometheus as a Building Block
- Build Microservices with Open Source components
- Adding a Control Plane Microservice

by design Prometheus has got

A Powerfull Query Language

Labels for adding Context

an Efficient Design !

Consolidation and Alarmrules

that can be executed in „Real Time“

by design Prometheus has

No event Processing

No sub second resolution

No Multitenancy

No Receiving

Howto Transport

openmetrics as messaging format over mqtt

```
# HELP http_requests_total The total number of HTTP requests.
# TYPE http_requests_total counter
http_requests_total{method="post",code="200"} 1027 1395066363000
http_requests_total{method="post",code="400"}    3 1395066363000
```

https://github.com/RichiH/OpenMetrics

ingress

RevPI
openmetrics

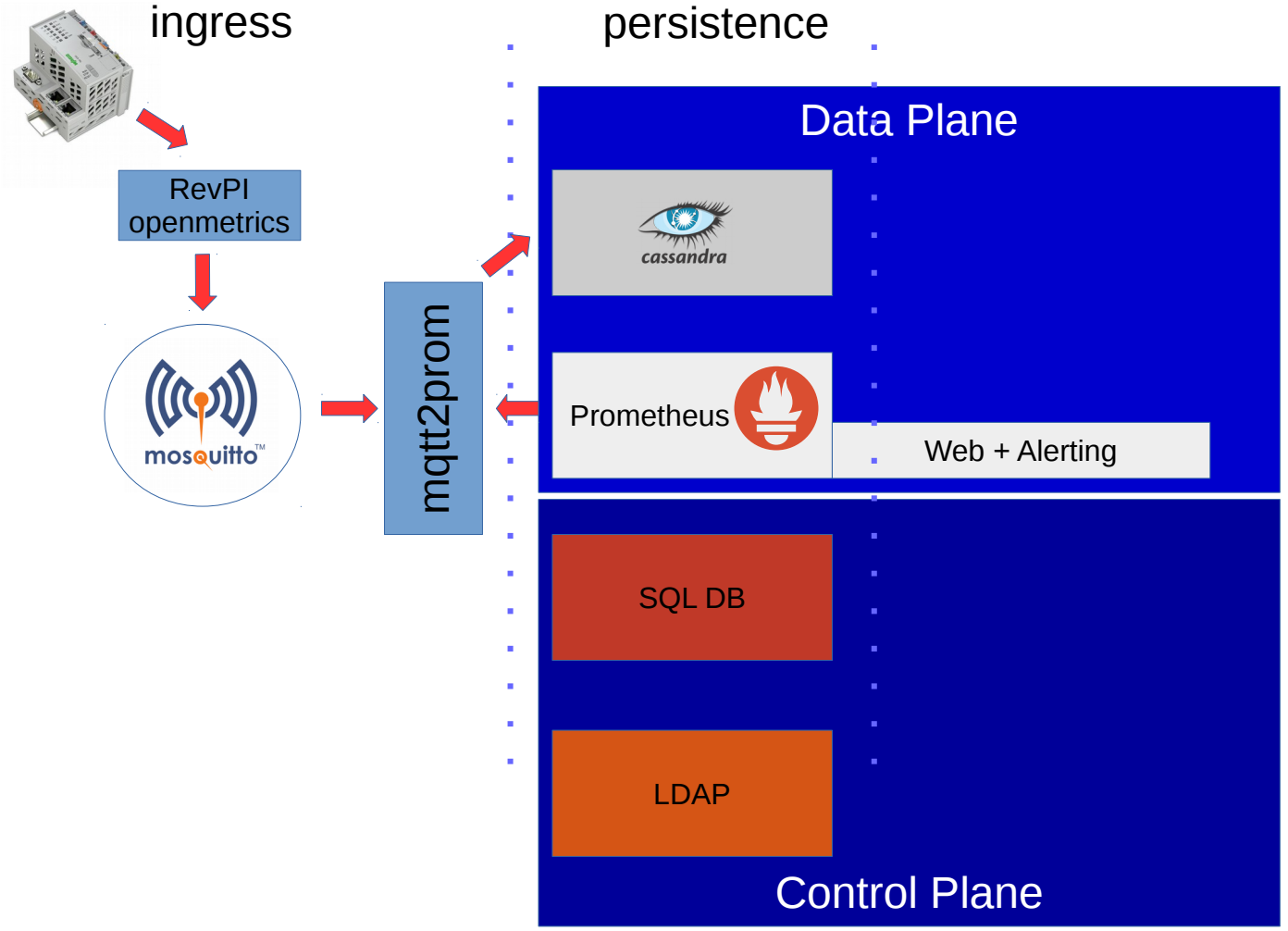
mosquitto™

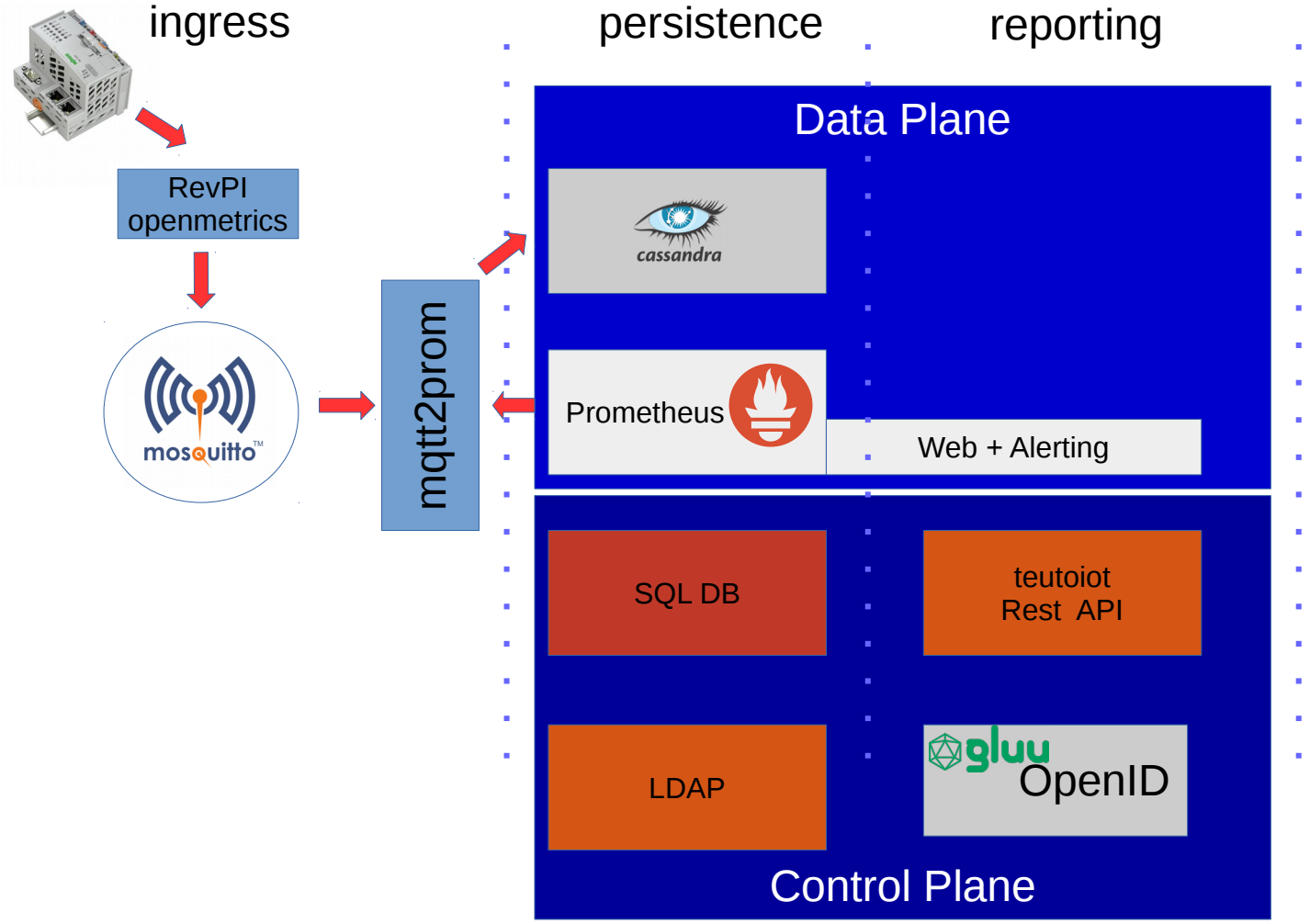
ingress

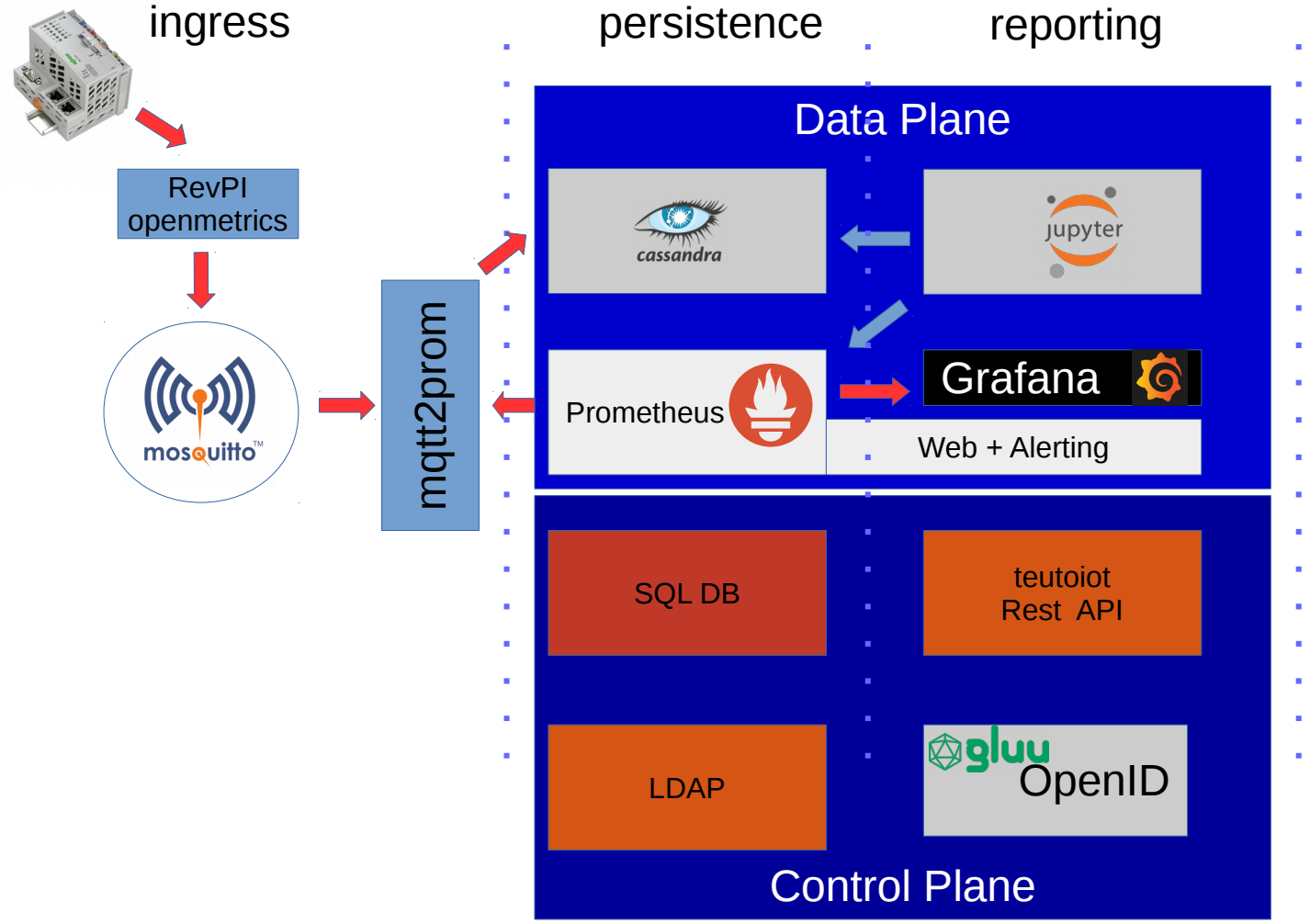
RevPI
openmetrics

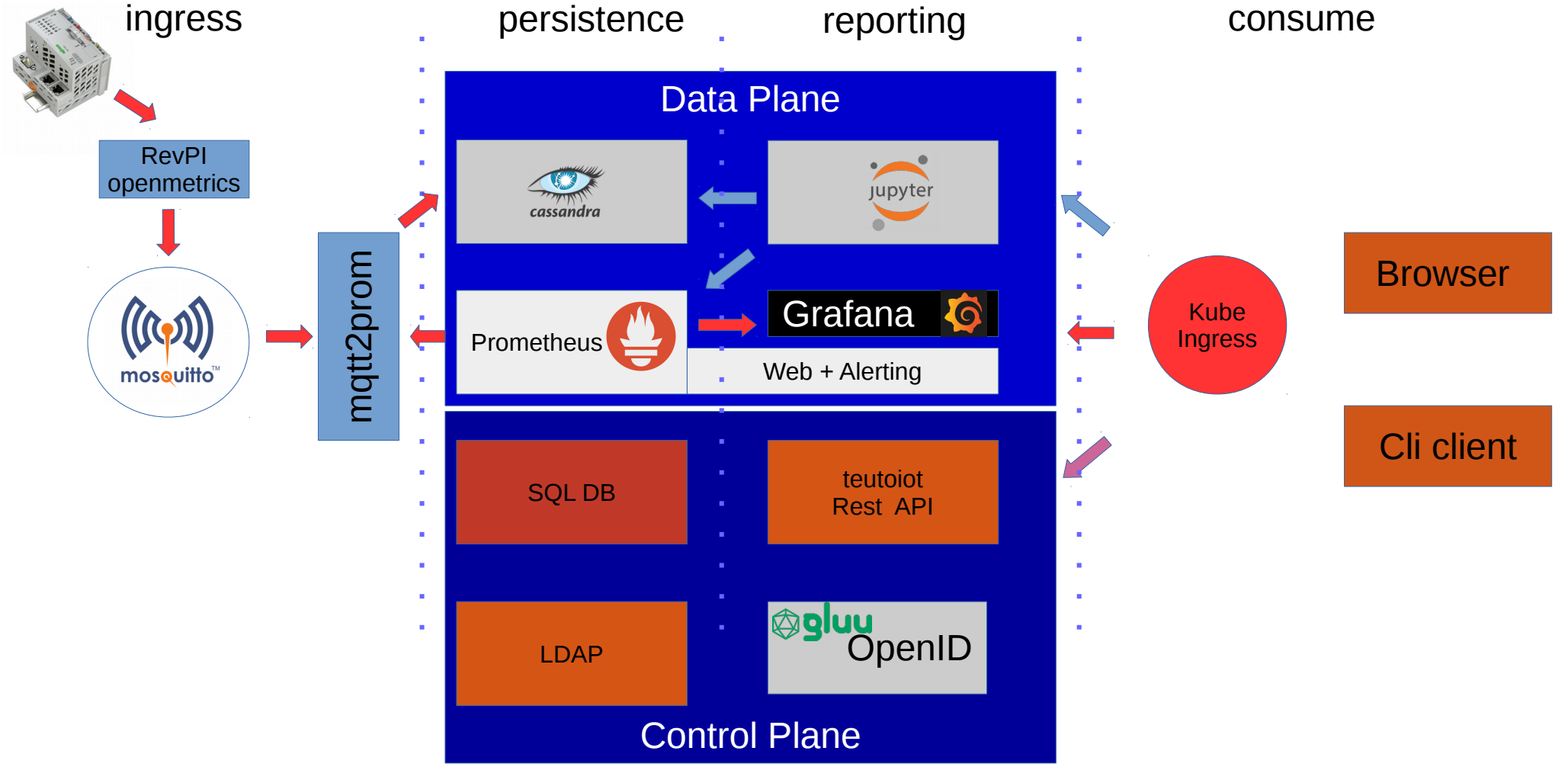
mosquitto™

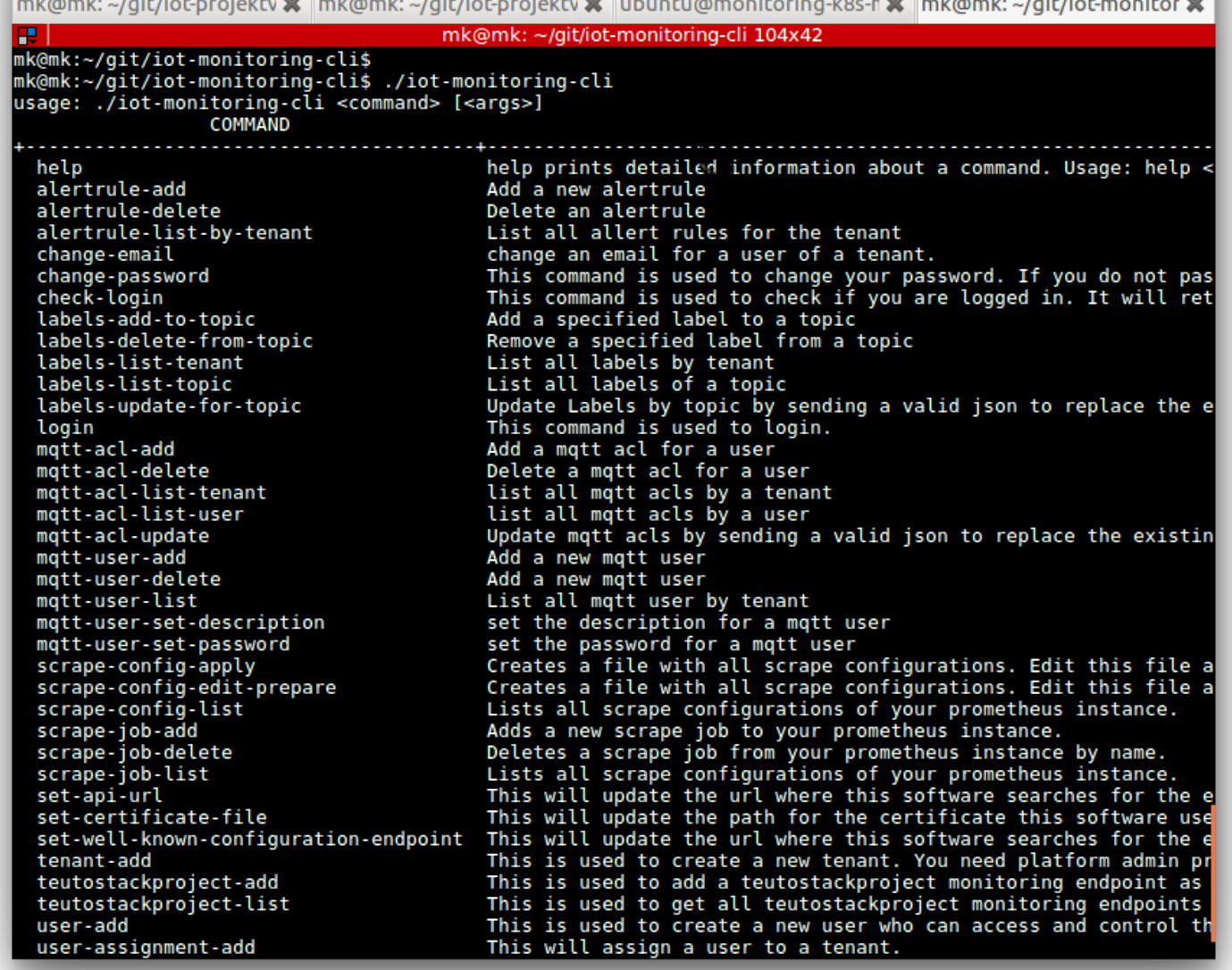
mqtt2prom


ingress
persistence
RevPI
openmetrics
mosquitto
mqtt2prom
Data Plane
cassandra
Prometheus
Web + Alerting
SQL DB
LDAP
Control Plane


ingress
persistence
reporting
RevPI
openmetrics
mosquitto
mqtt2prom
Data Plane
cassandra
Prometheus
Web + Alerting
SQL DB
teutoiot
Rest API
LDAP
gluu
OpenID
Control Plane


ingress
persistence
reporting
RevPI
openmetrics
mosquitto
mqtt2prom
Data Plane
cassandra
jupyter
Prometheus
Grafana
Web + Alerting
SQL DB
teutoiot
Rest API
LDAP
gluu
OpenID
Control Plane

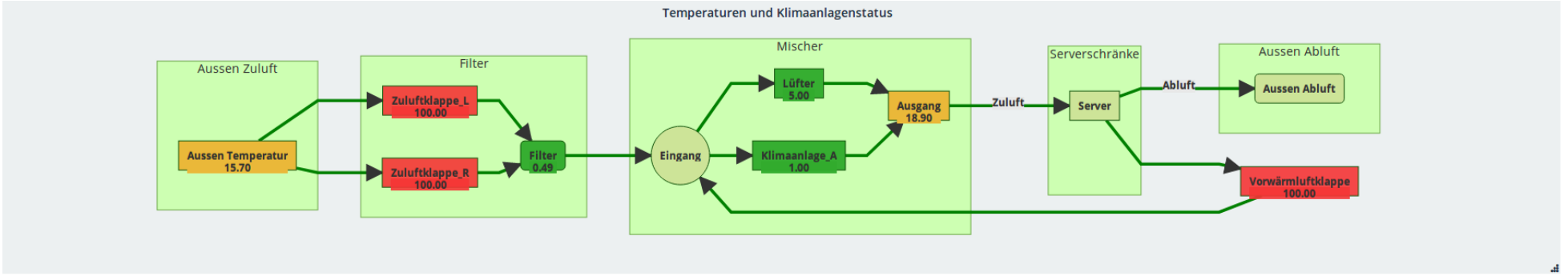

ingress
persistence
reporting
consume
RevPI
openmetrics
mosquitto
mqtt2prom
Data Plane
cassandra
jupyter
Prometheus
Grafana
Web + Alerting
SQL DB
teutoiot
Rest API
LDAP
gluu
OpenID
Control Plane
Kube
Ingress
Browser
Cli client

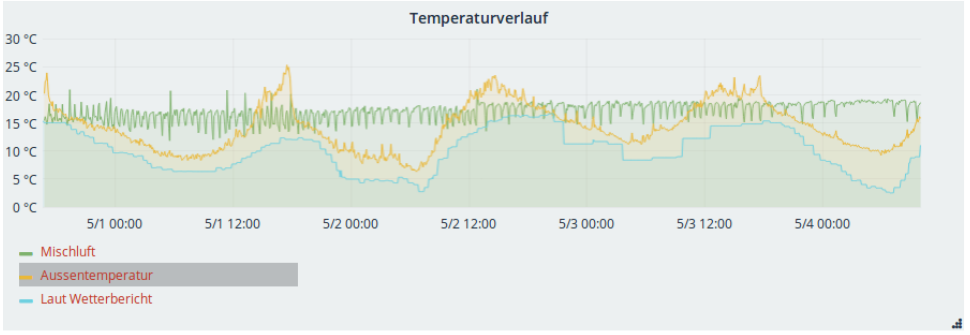


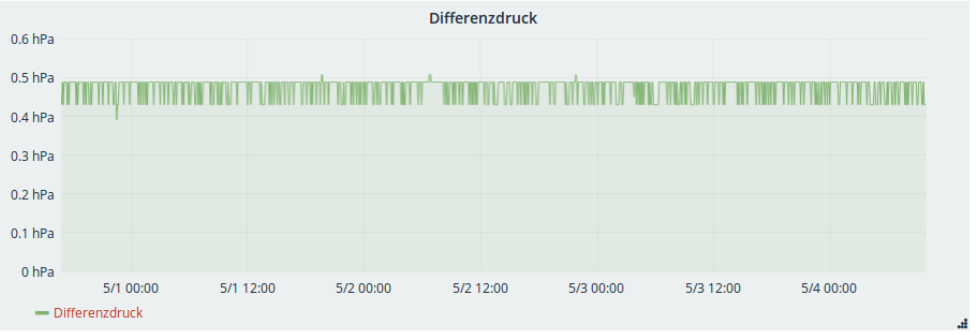
Golang CLI for Linux/Windows

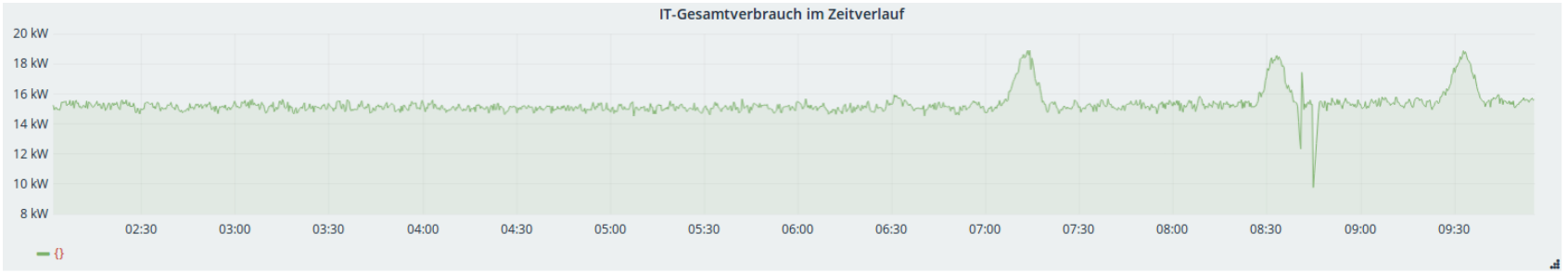
Workflow at factory Floor

- Generate API User with SensorPath
- Configure Edge Converter with a Certain Machine konfiguration e.g. Modbus Register Mapping
- Connecting with Internet
- Configure Grafana Dashboard

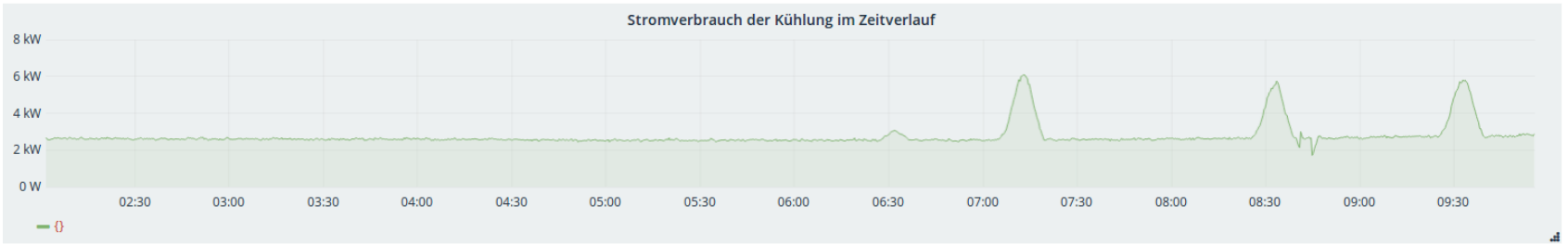

Temperaturen und Klimaanlagenstatus
Aussen Zuluft
Aussen Temperatur
15.70
Filter
Zuluftklappe_L
100.00
Zuluftklappe_R
100.00
Filter
0.49
Mischer
Eingang
Lüfter
5.00
Klimaanlage_A
1.00
Ausgang
18.90
Zuluft
Serverschränke
Server
Abluft
Aussen Abluft
Aussen Abluft
Vorwärmluftklappe
100.00


Temperaturverlauf
30 °C
25 °C
20 °C
15 °C
10 °C
5 °C
0 °C
5/1 00:00
5/1 12:00
5/2 00:00
5/2 12:00
5/3 00:00
5/3 12:00
5/4 00:00
Mischluft
Aussentemperatur
Laut Wetterbericht

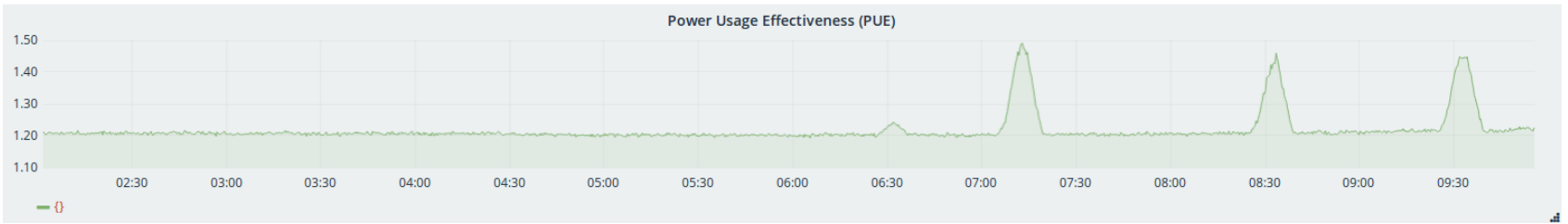

Differenzdruck
0.6 hPa
0.5 hPa
0.4 hPa
0.3 hPa
0.2 hPa
0.1 hPa
0 hPa
5/1 00:00
5/1 12:00
5/2 00:00
5/2 12:00
5/3 00:00
5/3 12:00
5/4 00:00
Differenzdruck

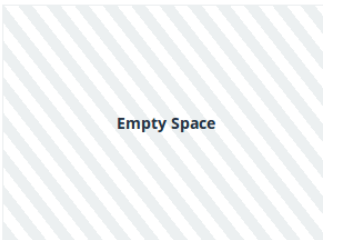

IT-Gesamtverbrauch im Zeitverlauf
20 kW
18 kW
16 kW
14 kW
12 kW
10 kW
8 kW
02:30
03:00
03:30
04:00
04:30
05:00
05:30
06:00
06:30
07:00
07:30
08:00
08:30
09:00
09:30
{}


IT-Gesamtverbrauch
15.565 kW


Stromverbrauch der Kühlung im Zeitverlauf
8 kW
6 kW
4 kW
2 kW
0 W
02:30
03:00
03:30
04:00
04:30
05:00
05:30
06:00
06:30
07:00
07:30
08:00
08:30
09:00
09:30
{}


Stromverbrauch der Kühlung
2.797 kW


Power Usage Effectiveness (PUE)
1.50
1.40
1.30
1.20
1.10
02:30
03:00
03:30
04:00
04:30
05:00
05:30
06:00
06:30
07:00
07:30
08:00
08:30
09:00
09:30
{}


Empty Space

Tensorflow Prototype

- Multi Dimension Metrics from a (virtual) Machine
- Splitting them into a Training Set, Validation Set
- Setup a LTSM Network and train it

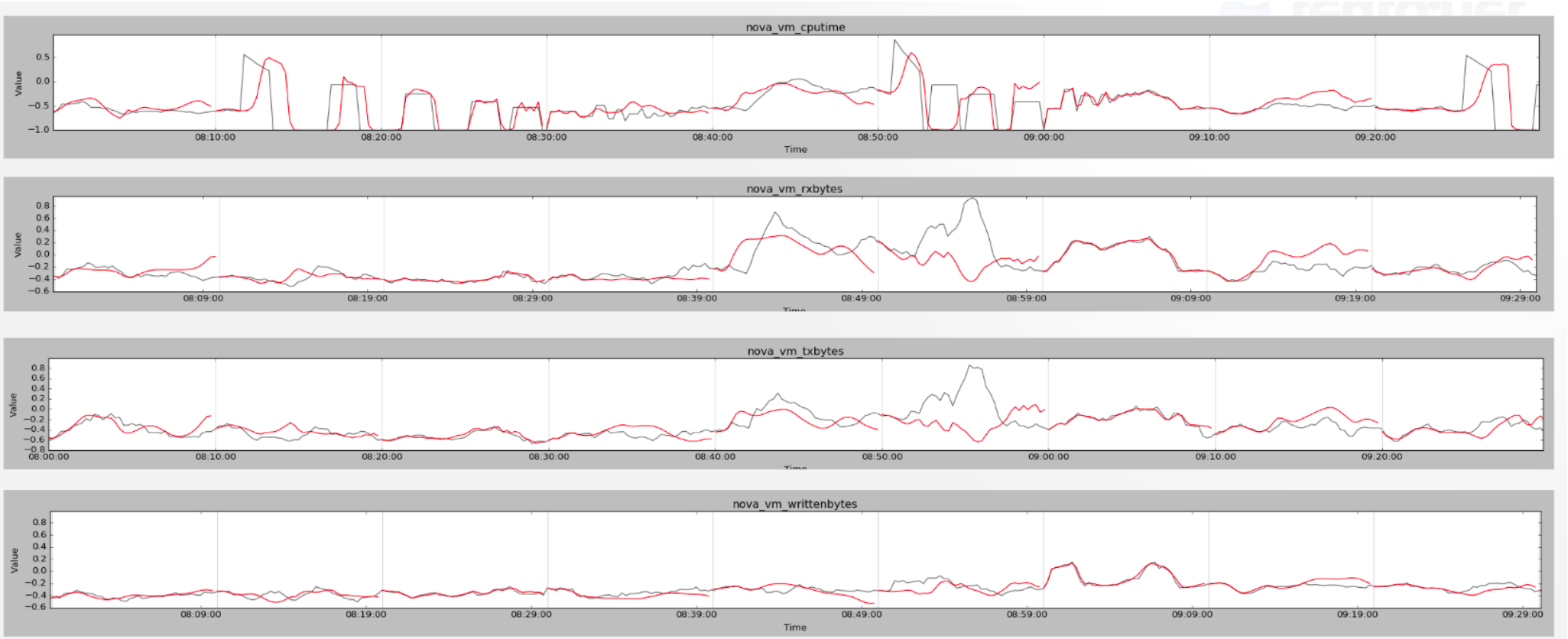
Dimension	Prometheus Query
1	`rate(nova_vm_cputime{instance_name="db3",cpu_no="0"}[2m])`
2	`rate(nova_vm_writtenbytes{instance_name="db3"}[2m])`
3	`rate(nova_vm_rxbytes{instance_name="db3"}[2m])`
4	`rate(nova_vm_txbytes{instance_name="db3"}[2m])`

Getting Data out of Prometheus

prometheus.teuto.net:9090/api/v1/query_range?query=rate(nova_vm_cputime{instance_name="db3", cpu_no="0"}[2m])&start=1488355200&end=1488358800&step=30m


```
{
   status: "success",
 ▾ data: {
      resultType: "matrix",
    ▾ result: [
       ▾ {
          ▸ metric: {…},
          ▾ values: [
             ▾ [
                  1488355200,
                  "0"
               ],
             ▾ [
                  1488357000,
                  "1864952380.952381"
               ],
             ▾ [
                  1488358800,
                  "3197428571.428571"
               ]
            ]
         }
      ]
   }
}
```

Prediction


nova_vm_cputime
Value
0.5
0.0
−0.5
−1.0
08:10:00
08:20:00
08:30:00
08:40:00
08:50:00
09:00:00
09:10:00
09:20:00
Time
nova_vm_rxbytes
Value
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
08:09:00
08:19:00
08:29:00
08:39:00
08:49:00
08:59:00
09:09:00
09:19:00
09:29:00
nova_vm_txbytes
Value
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
08:00:00
08:10:00
08:20:00
08:30:00
08:40:00
08:50:00
09:00:00
09:10:00
09:20:00
nova_vm_writtenbytes
Value
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
08:09:00
08:19:00
08:29:00
08:39:00
08:49:00
08:59:00
09:09:00
09:19:00
09:29:00
Time

What we accomplished

- Confidence in the Team the Microservices are doable
- CI Testing of Microservice is hard
- Ci Build of x86 and arm Container in same CI run
- Initial Tensorflow Prototype is doable
- Getting valid predictions is very very hard.

References

- https://github.com/gonium/gosdm630 (electricity meter)
- https://www.gluu.org/
- https://github.com/multiarch/alpine
- https://opcfoundation.org/about/what-is-opc/
- https://github.com/riptideio/pymodbus
- http://python-opcua.readthedocs.io/en/latest/index.html
- https://github.com/ernoaapa/eliot (device management)