



Hero

Timeseries Historian

 **BASF**

We create chemistry

inspired by
TUM: Junge Akademie



TUM
TECHNISCHE
UNIVERSITÄT
MÜNCHEN





We are living in the world of Data





“

Without a systematic way to start and keep data clean, bad data will happen.

— Donato Diorio, Principal Consultant, DataZ

94 %

Of companies experience severe data loss do not recover

51 %

Of these companies close within 2 years of the data loss

43%

Of these companies do not reopen again

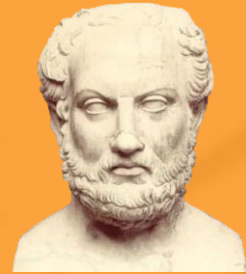
5,000,000\$



HELLO!

We are team Herodotus

We are here to provide the solution of a storage system for time series data.



HERO

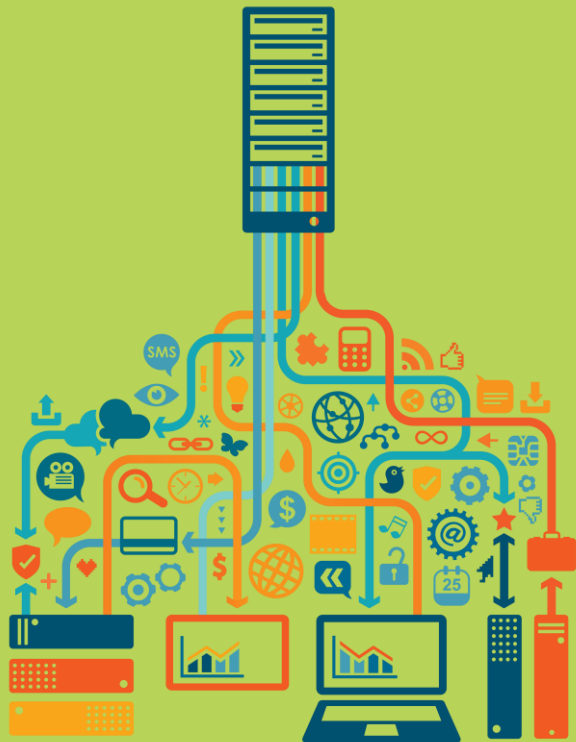
How does it work?

Key feature

- Scalability
- Availability
- Write Throughput / Fast Persistency
- Cloud-Independence
- Efficient Storage Consumption
- High Read Query Performance
- Better User Experience



Collecting data





Collecting data

1 second



Scalability
Availability
Write Throughput / Fast Persistency
Cloud- Independence
Efficient Storage Consumption
High Read Query Performance
Better User Experience

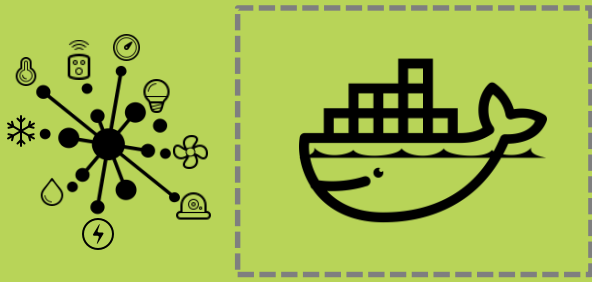
Collecting data

What is happening in a second?

- Millions of datapoints
- Millisecond resolution
- Terabytes of data



- 67% of data loss is caused by hardware or system failure
- 14% of data loss is caused by human error



Docker based containerization

Why?

- Light weight
- Rapid deployment

Scalability

Availability

Write Throughput / Fast Persistency

Cloud- Independence

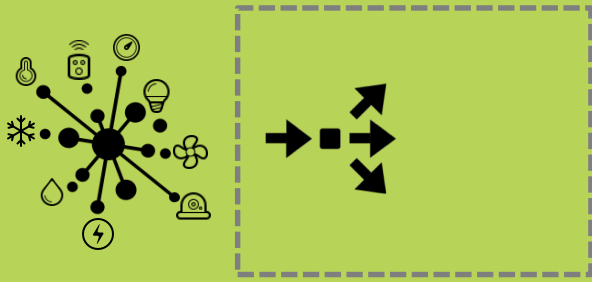
Efficient Storage Consumption

High Read Query Performance

Better User Experience

Containerization





Load-balance with HA Proxy

Why?

- Zero-downtime maintenance
- Event driven architecture

Scalability

Availability

Write Throughput / Fast Persistency

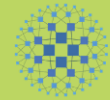
Cloud- Independence

Efficient Storage Consumption

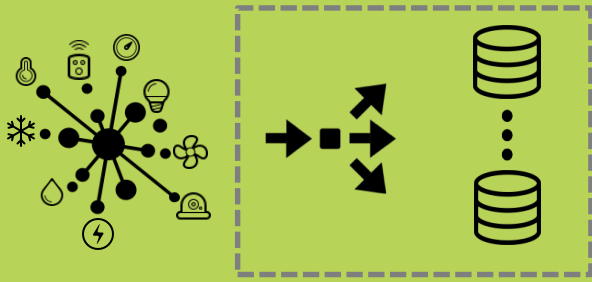
High Read Query Performance

Better User Experience

Load balancing



HAPROXY



Introduce a right database to system

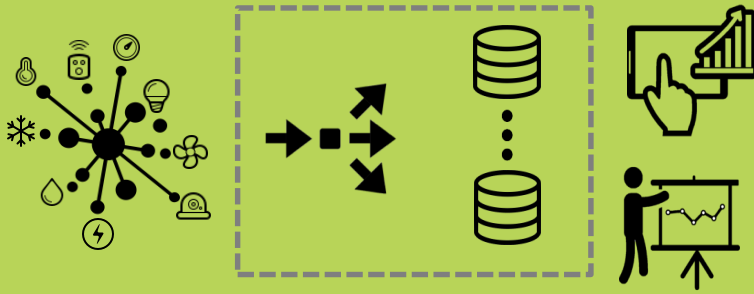
Why?

- Time-series data platform
- High availability
- Data compression

Scalability
Availability
Write Throughput / Fast Persistency
Cloud- Independence
Efficient Storage Consumption
High Read Query Performance
Better User Experience

Database

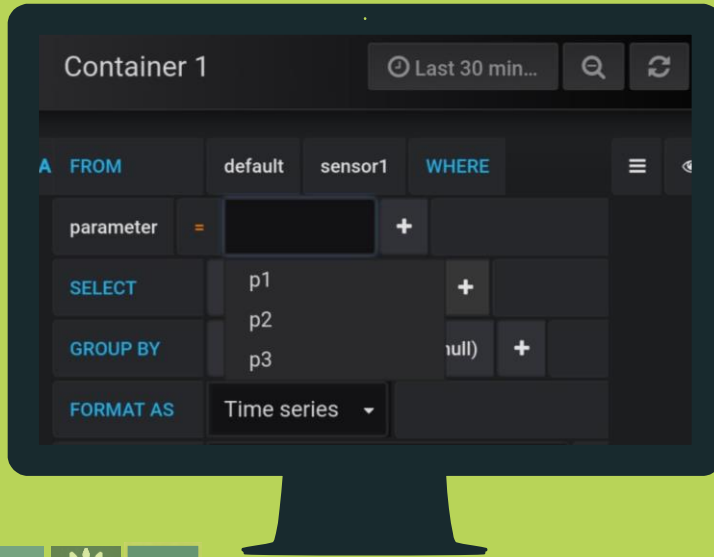




Read and visualize data



- Scalability
- Availability
- Write Throughput / Fast Persistency
- Cloud- Independence
- Efficient Storage Consumption
- High Read Query Performance
- Better User Experience



Why?

- Direct plugin to InfluxDB

Visualization

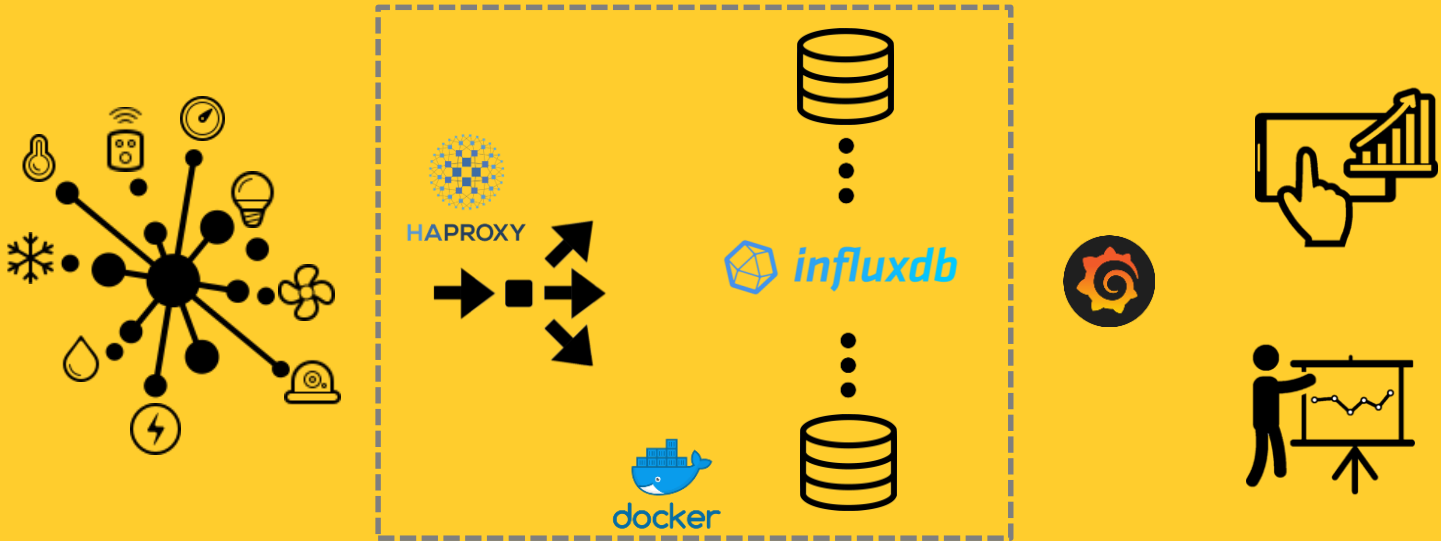


Grafana



plotly

Architecture



DEM

O

```
hackmachine@hackmachine:~/data/sciencehack$ python post.py
Traceback (most recent call last):
  File "post.py", line 95, in <module>
    client.write_points(json_body)
  File "/usr/local/lib/python2.7/dist-packages/influxdb/client.py", line 490, in write_points
    tags=tags, protocol=protocol)
  File "/usr/local/lib/python2.7/dist-packages/influxdb/client.py", line 551, in write
    protocol=protocol)
  File "/usr/local/lib/python2.7/dist-packages/influxdb/client.py", line 327, in write
    headers=headers)
  File "/usr/local/lib/python2.7/dist-packages/influxdb/client.py", line 262, in _write
    raise InfluxDBServerError(response.content)
influxdb.exceptions.InfluxDBServerError: <html><body><h1>503 Service Unavailable</h1>
No server is available to handle this request.
</body></html>

hackmachine@hackmachine:~/data/sciencehack$ sudo python post.py
```