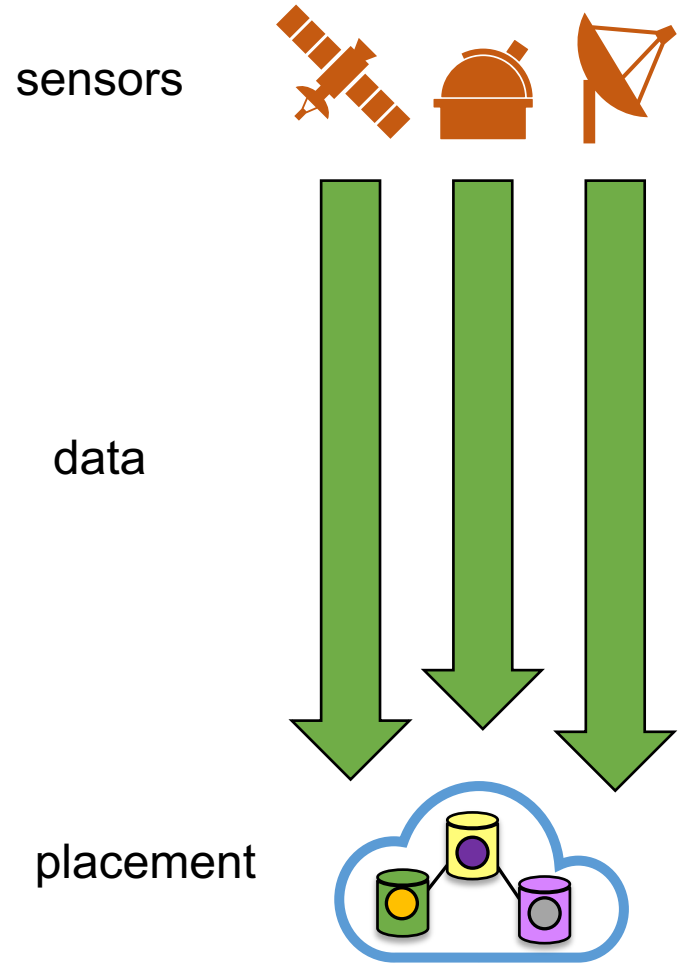
# Notes on avoiding « warrior/berserker mode »

*Pedro Silva*

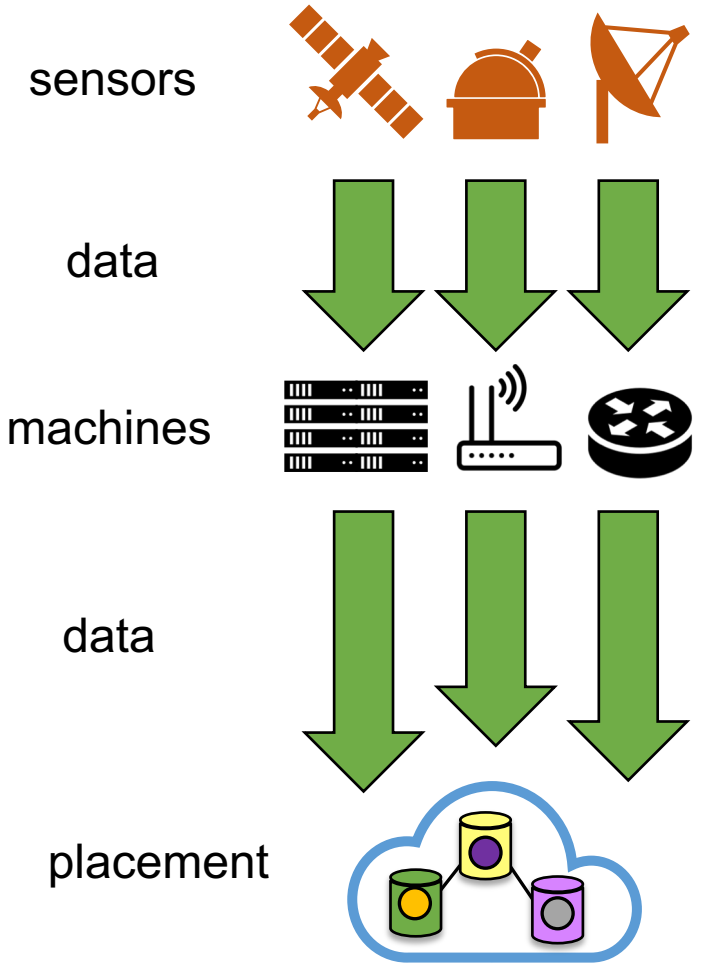
Kerdata Team, Inria

07/11/2019

# Edge/Fog computing

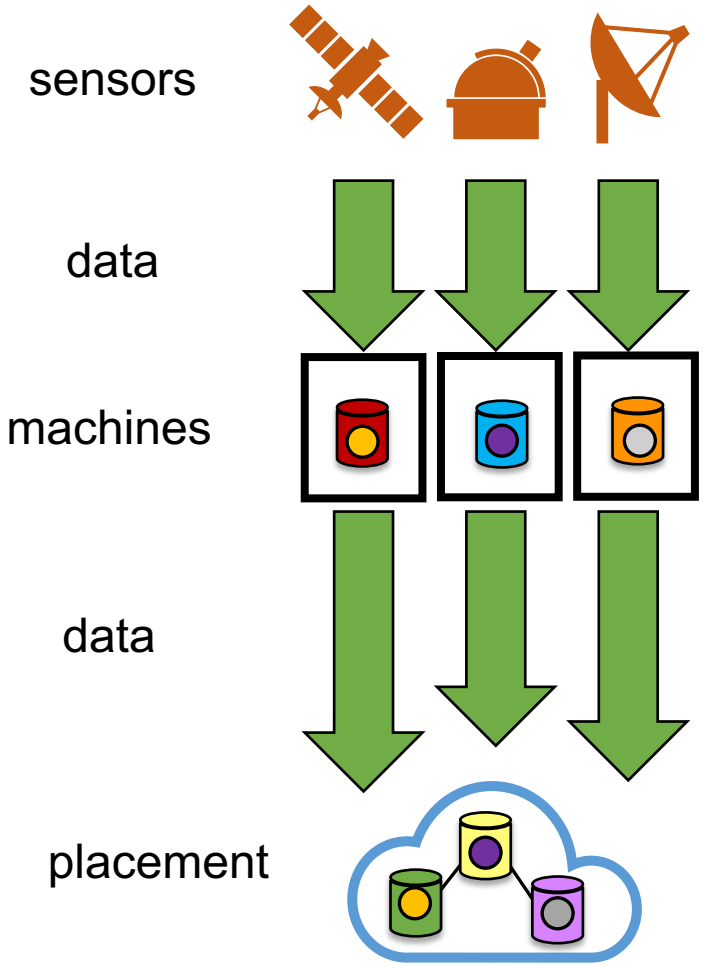


# Edge/Fog computing



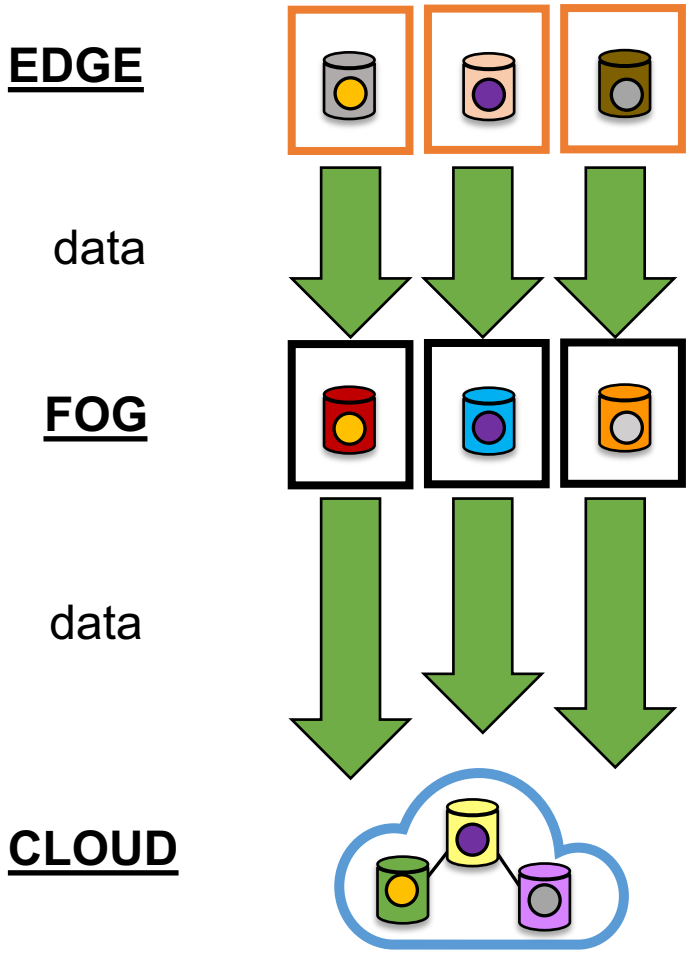
# Edge/Fog computing

- **Objectives**:
  - **Reduce** data transmission costs
  - **Decentralize** the decision making
  - Benefit from the implicit **parallelism**
  - Data **anonymisation**

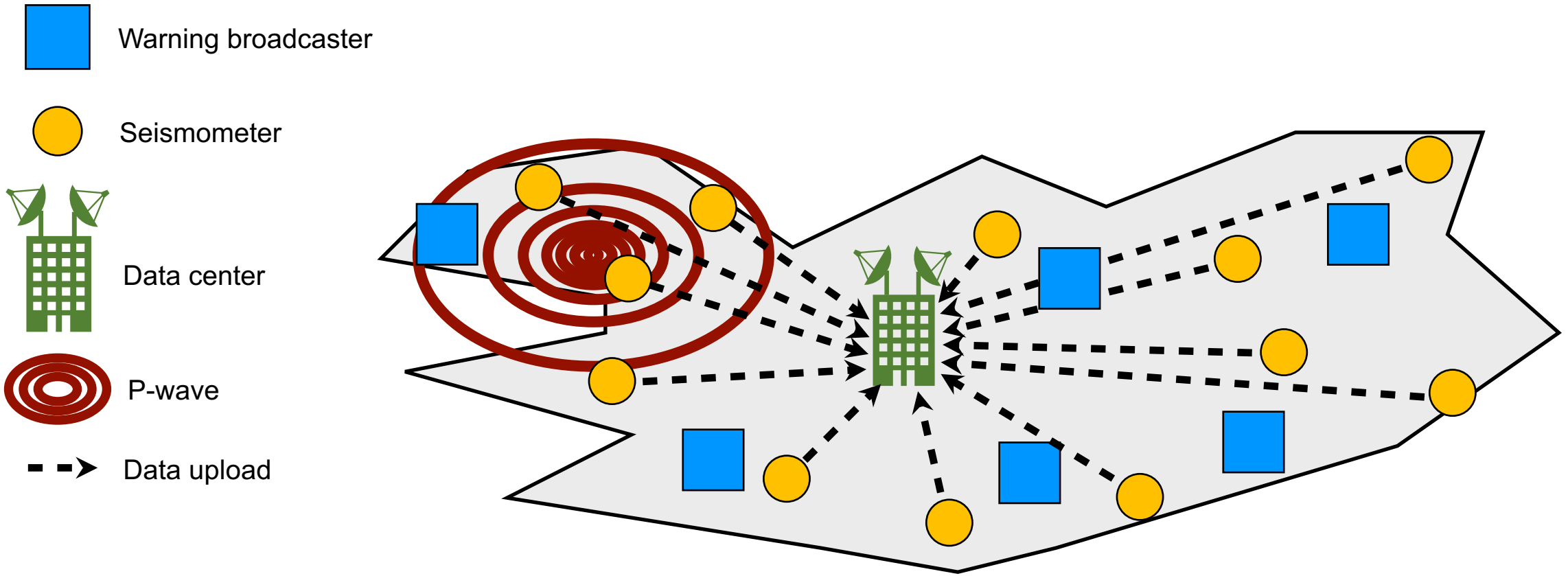


# Edge/Fog computing

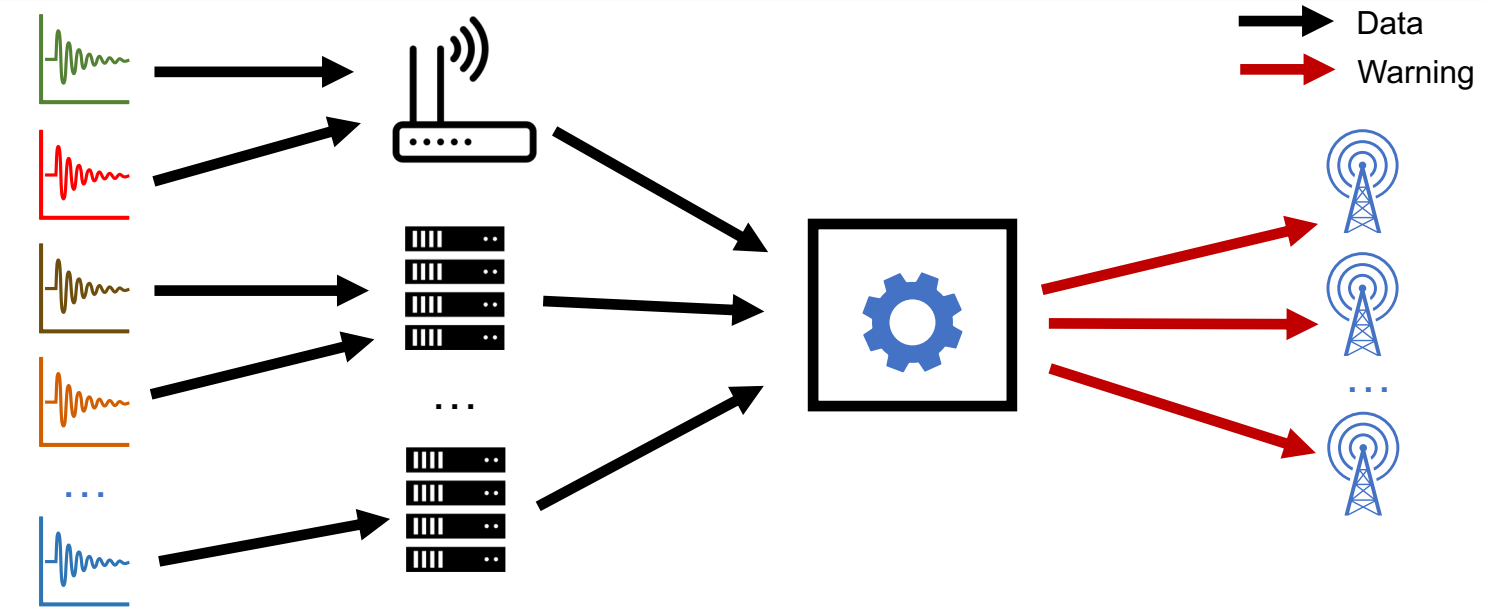
- **Objectives**:
  - **Reduce** data transmission costs
  - **Decentralize** the decision making
  - Benefit from the implicit **parallelism**
  - Data **anonymisation**



# Earthquake Early Warning Systems (EEW)



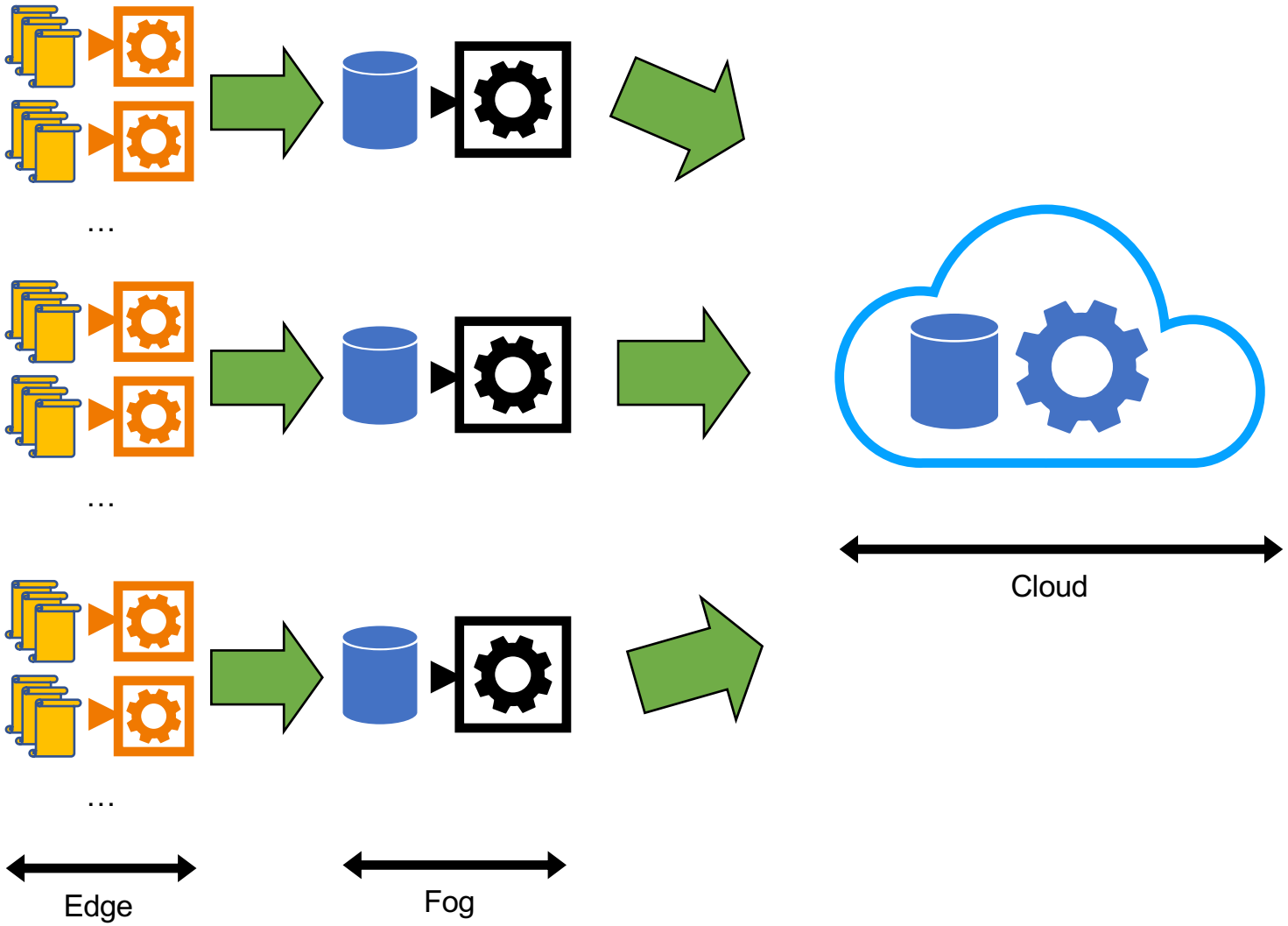
# Earthquake Early Warning Systems (EEW)



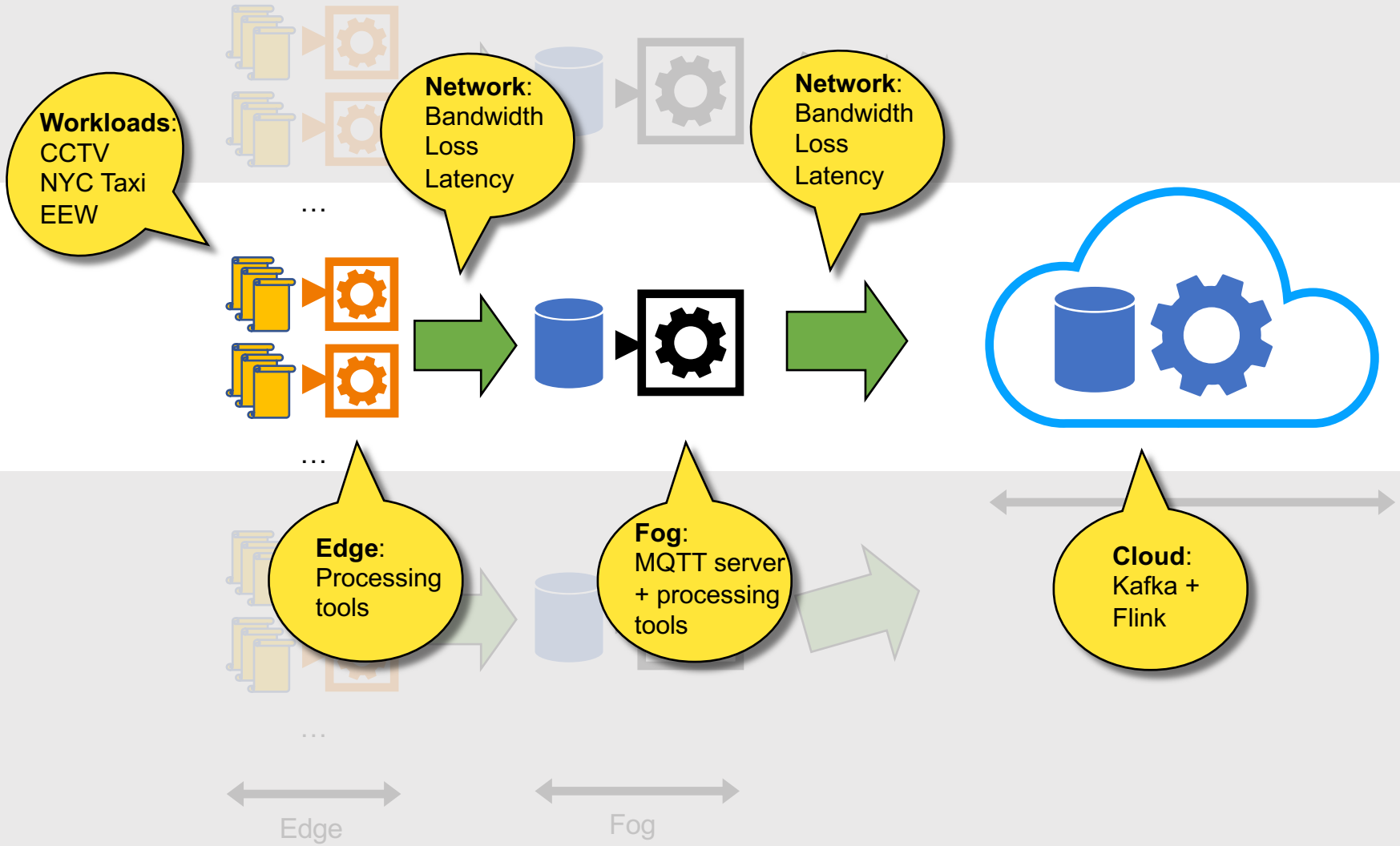

- **Deem: hierarchical** and **distributed** ML algorithm
- Enables the usage of **multiple types** of sensors
- Enables the deployment on **less powerful** networks
- Enables **local** decision making.


- **SILVA, P.**, BALOUEK-THOMERT, D.; FAUVEL, K. ; MELGAR, D. ; SIMONET, A. ; ANTONIU G. ; COSTAN, A ; MASSON, V ; PARASHAR, M. ; RODERO, I. ; TERMIER, A A hybrid Fog and Cloud computing based approach for Earthquake Early Warning Systems. (On-going work.Target: CCGrid 2020)
- FAUVEL, K. ; BALOUEK-THOMERT, D. ; MELGAR, D. ; **SILVA, P.**, SIMONET, A. ; ANTONIU G. ; COSTAN, A ; MASSON, V ; PARASHAR, M. ; RODERO, I. ; TERMIER, A. A Distributed Multi-Sensor Machine Learning Approach to Earthquake Early Warning. Submitted to AAAI 2020.

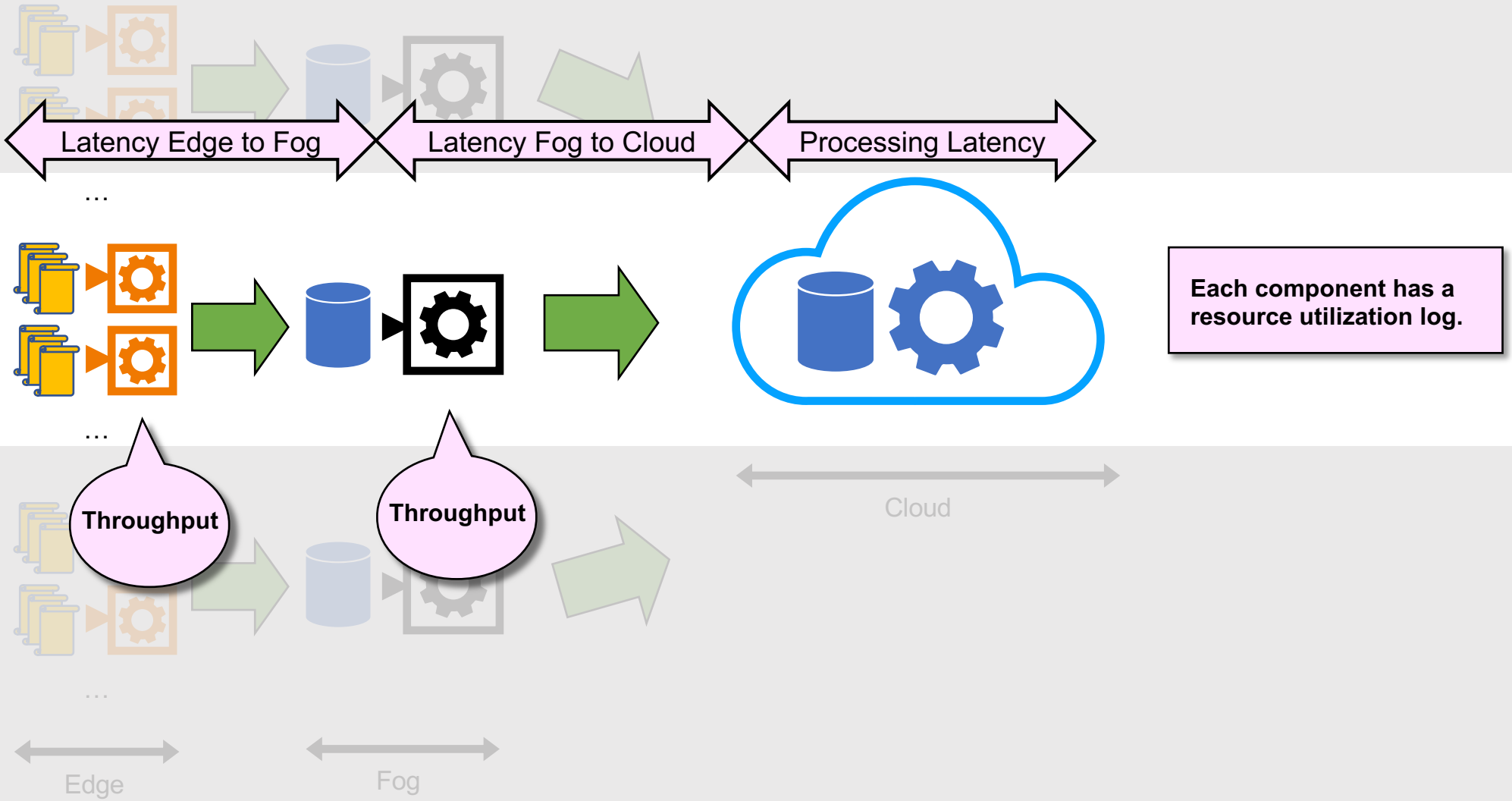
# benchmarking platform: overview & objectives



# benchmarking platform: parameters



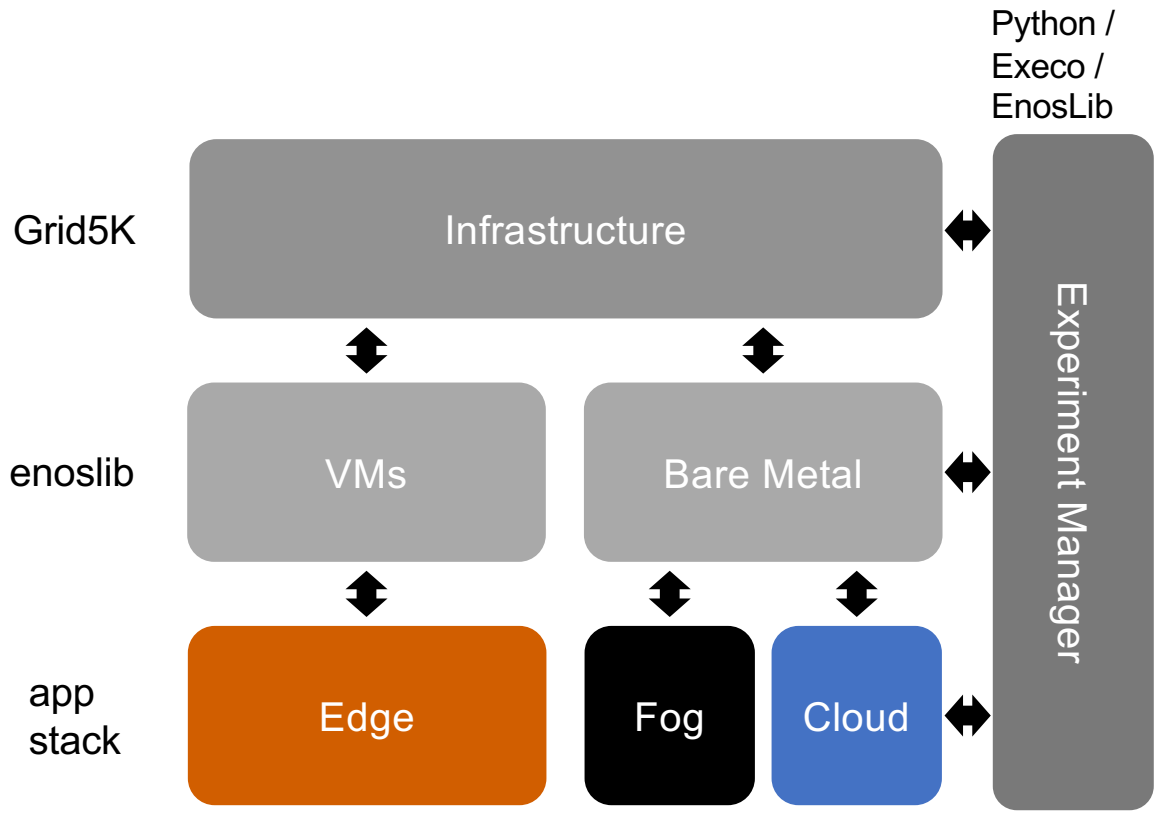
# benchmarking platform: metrics



# benchmarking platform: implementation

- **Experiment manager:**
  - Configures the infrastructure
  - Deploys frameworks/tools
  - Deploys applications and manages their executions
  - Monitors resource usage
  - Gathers metrics and logs

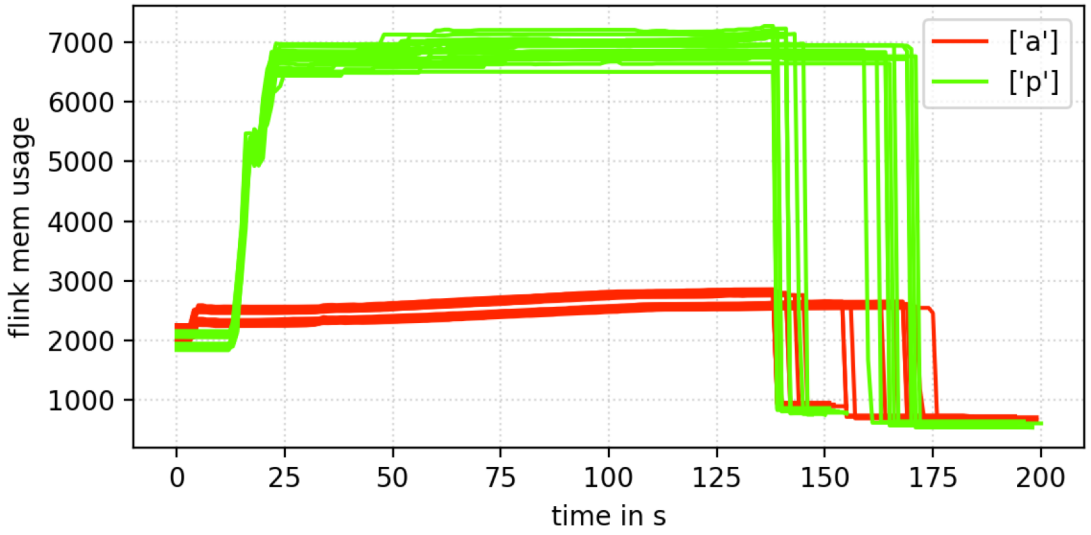
- **Edge+Fog+Cloud processing management:**
  - Wrappers / interfaces (metric generation, configuration, connection)

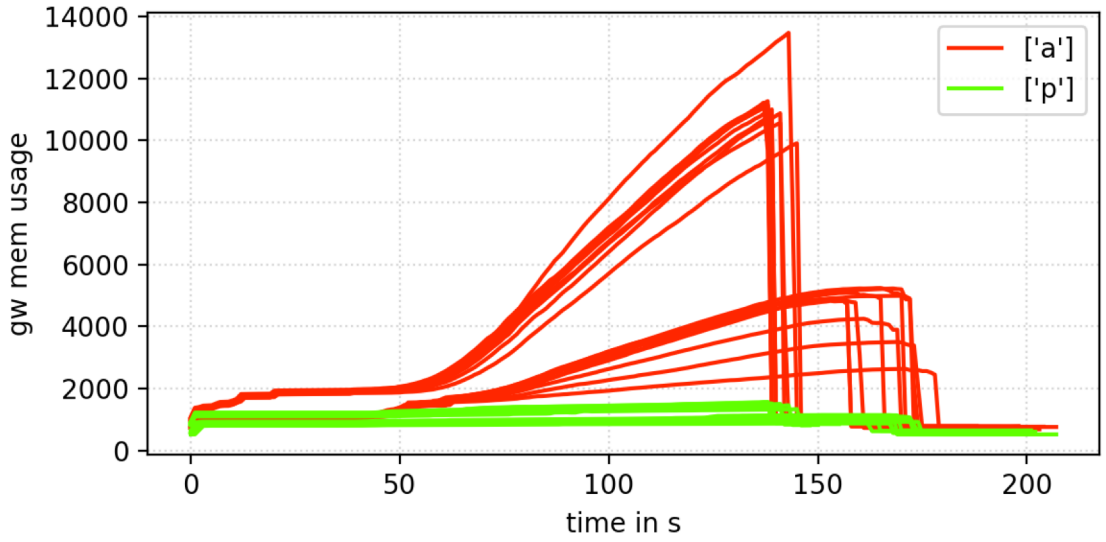


# eew fog based infrastructure

- Thousands of producers
- High load on Fog and Cloud
- Objectives:
  - Reduction of **network costs** (Fiber vs. Laura / 3G / 4G / 5G)
  - Reduction of **Cloud costs**
  - Easier **network reconfiguration** (intelligent fog nodes)

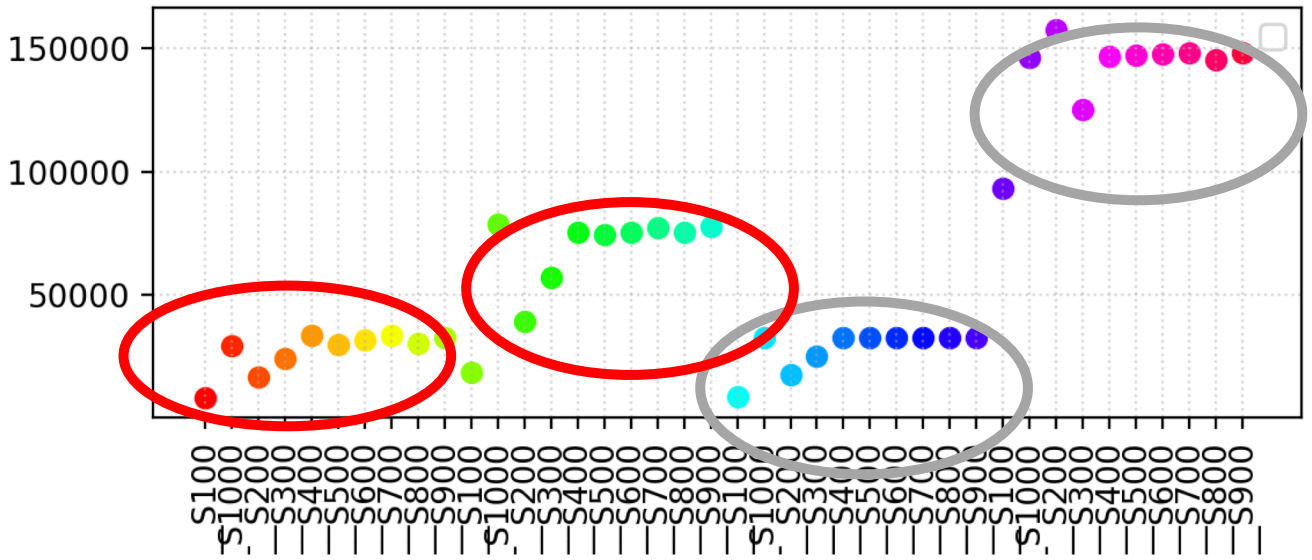
# eew fog based infrastructure: gateway overhead

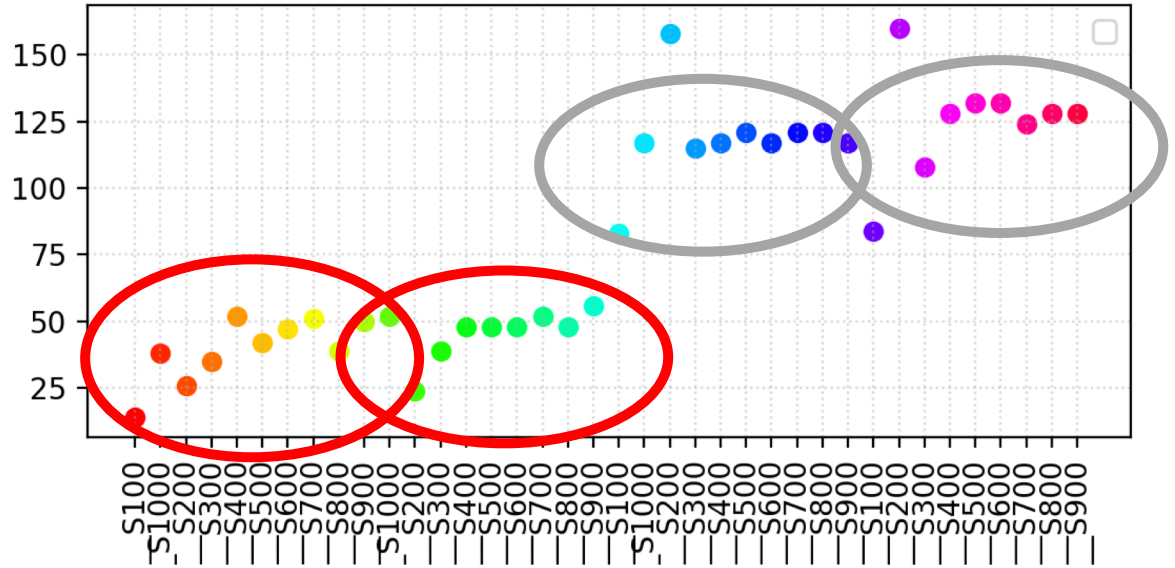



- ❑ reduction of res consumption in the Cloud.
- ❑ augmentation of res consumption in the Fog

# eew fog based infrastructure: cost reduction




- less data on the network (left), similar output (right)

# know your objectives

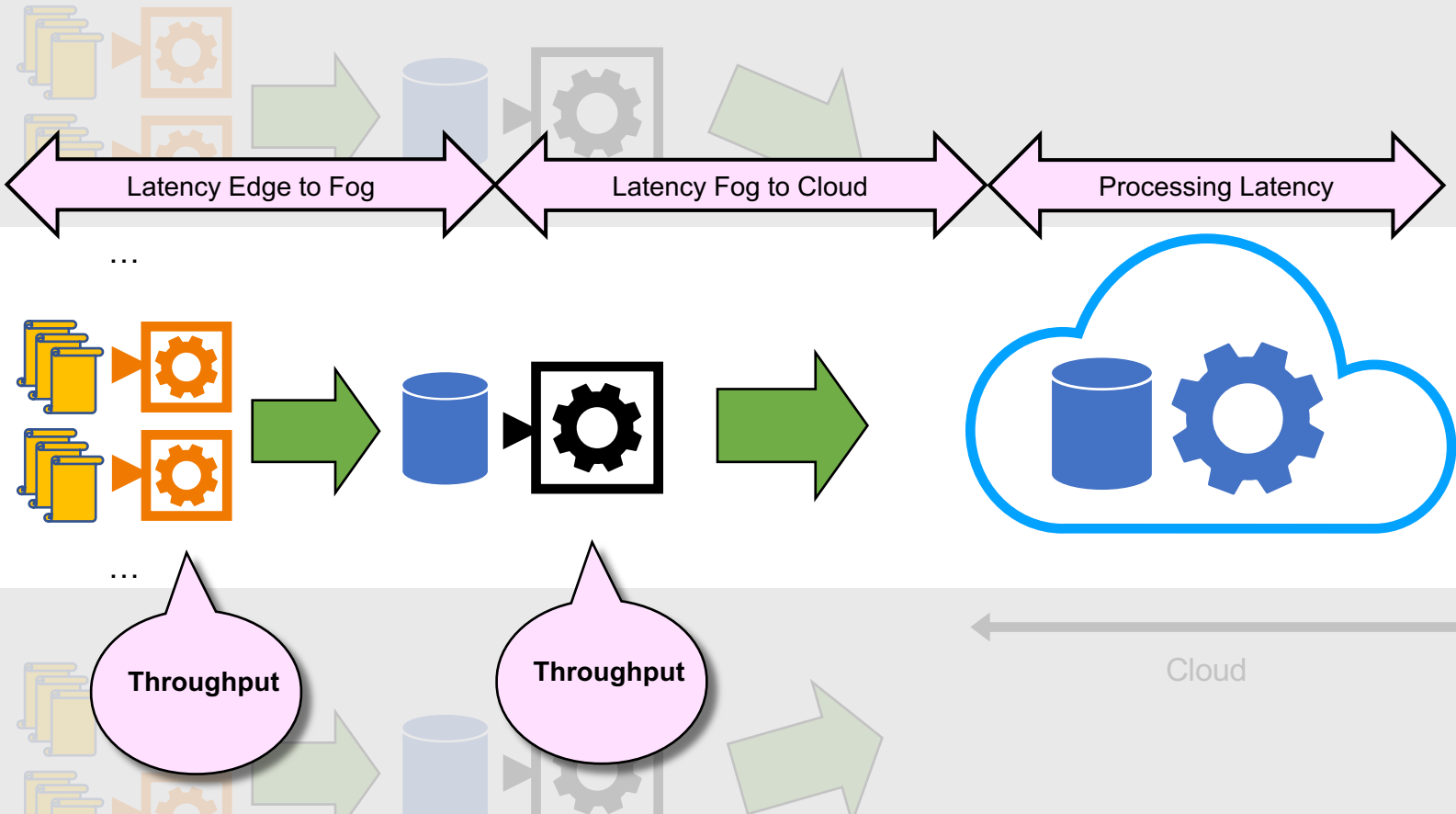
- define high level objectives:
  - "I want to show that my algorithm reduces data transmission between Fog and Cloud."
  - "I want to indicate that approach A causes the machines to consume more energy than approach B."
- define your assumptions:
  - "the stream of data is continuous"
  - "there are no failures of edge machines"
  - "fixed number of gateways"
- define the simplest possible metrics and infra

TABLE IV
EXPERIMENT CONFIGURATION FOR THE TLC SCENARIO.

| **Parameters** | **Class** $S$ | **Class** $L$ |
|---|---|---|
| T | $\{Edge, Cloud\}$ | $\{Edge, Cloud\}$ |
| P | $\{10, 20, 30\}$ | $\{20, 30, 40\}$ |
| GW | $\{10\}$ | $\{30\}$ |
| BW (Mb) | $\{0.5, 1, 10\}$ | $\{10, 100, 1000\}$ |
| FR (%) | $\{30, 50\}$ | $\{30, 50\}$ |
| Timeout (s) | 300 | 420 |
| Total experiments | 36 | 36 |

# know your objectives

- define the simplest possible infrastructure necessary

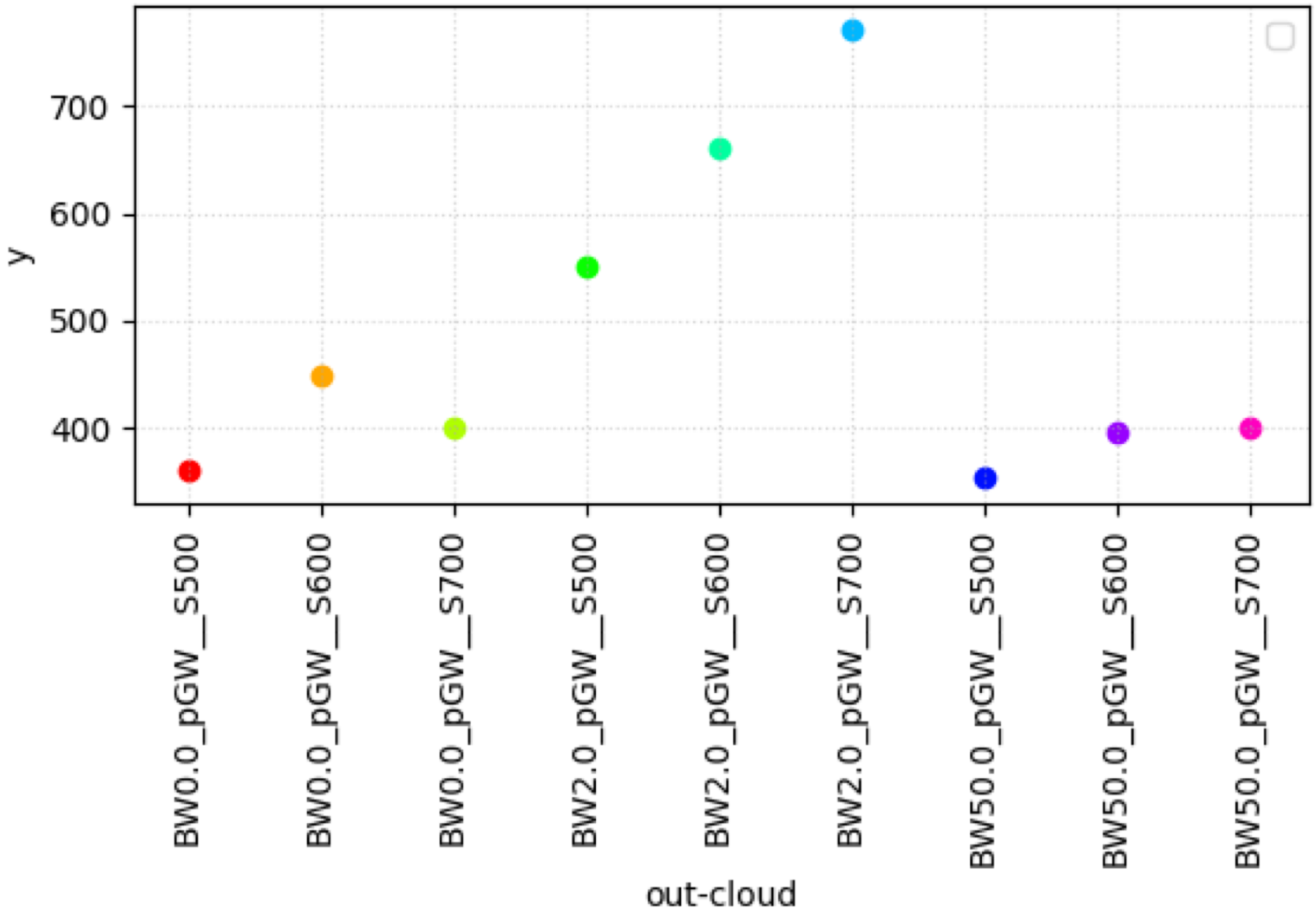


# know your software stack

- be sure you know the frameworks, libraries, etc. you're using!
  - worst: time.time() v.s. System.currentTimeMillis()
  - framework features: suggesting sliding windows in Apache Edgent.
  - Hoping that Eclipse Mosquitto had partition management options
  - Late events in Apache Flink
  - Performance parameters from Apache Kafka and Apache Flink

# know your software stack

- going berserk:
  - when the bandwidth was limited, I had a better throughput than with unlimited network!
  - ML algorithm was too slow when windows had too many messages to process.

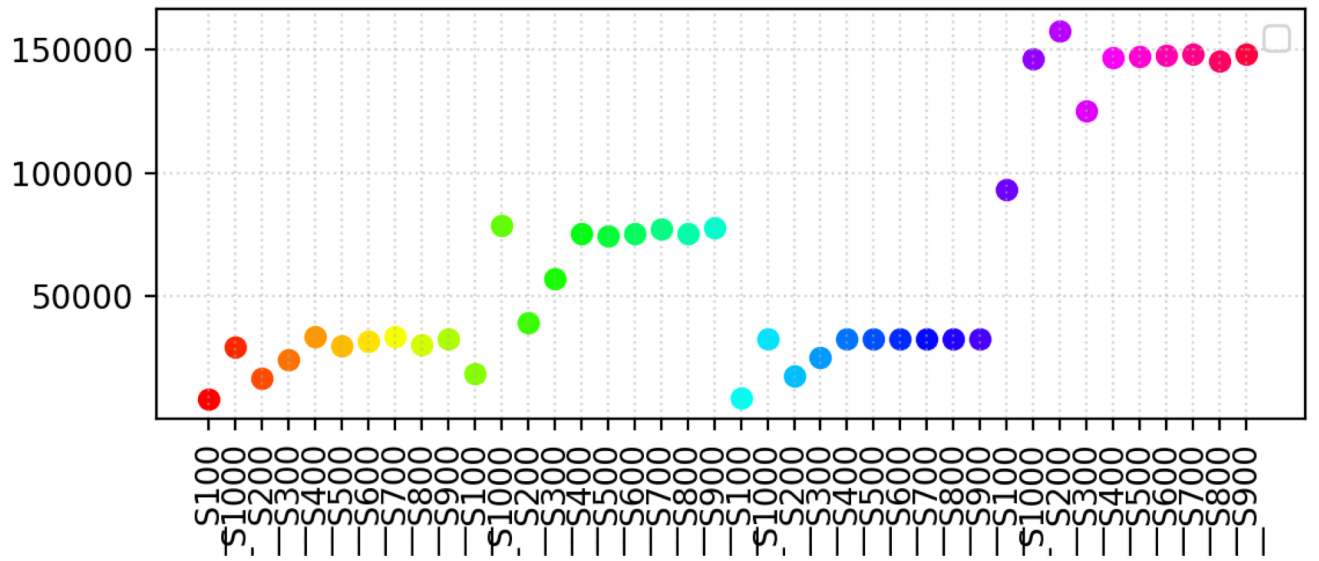


# know your data

- data can have a huge impact on the performance and results.
  - zero results on evaluating some event time windows on Apache Flink
  - amount of keys and its impact on Apache Flink performance
  - repeated keys

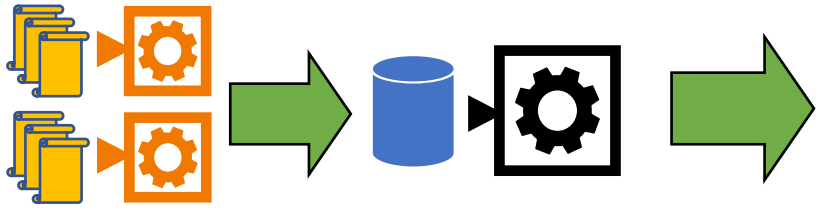
# know your result expectations

- Plan ahead what the results should be.
  - ex.: what's the theoretical throughput and latency on each point?



# know your scale

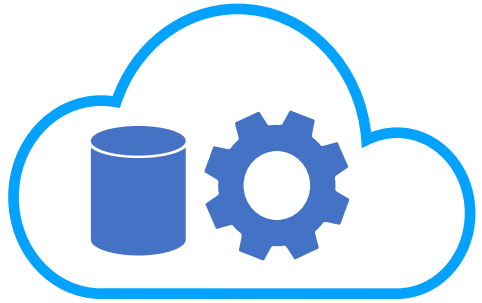
- ❑ project what could be happen when you scale!
- ❑ isolate each part of your experiments in order to find eventual performance leaks



...

- message producer (100Hz)

- timestamp adding
- retransmission

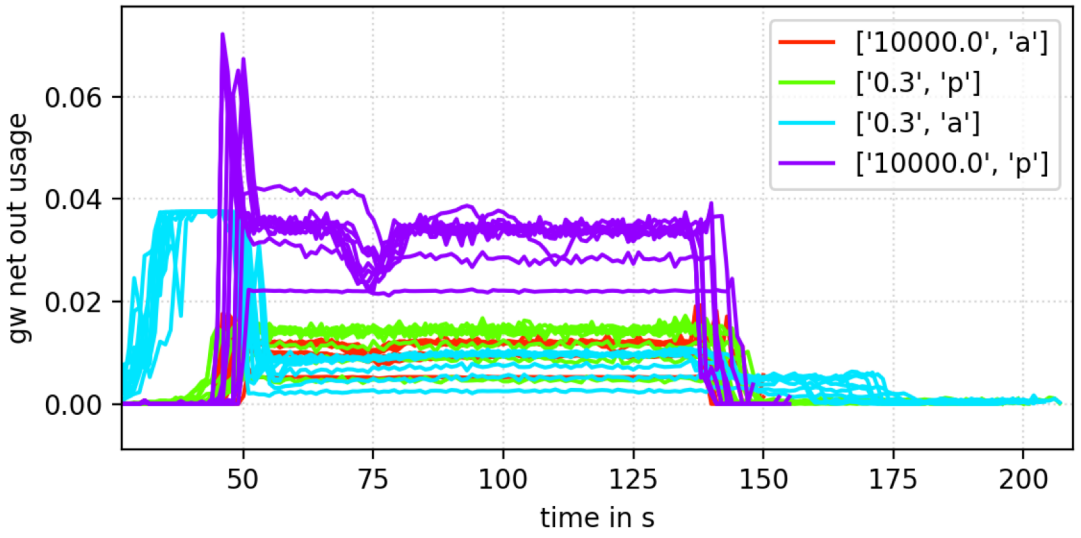


- sliding window per sensor (30 s / 1 s)

- tumbling window per seismic region (every 1 second)

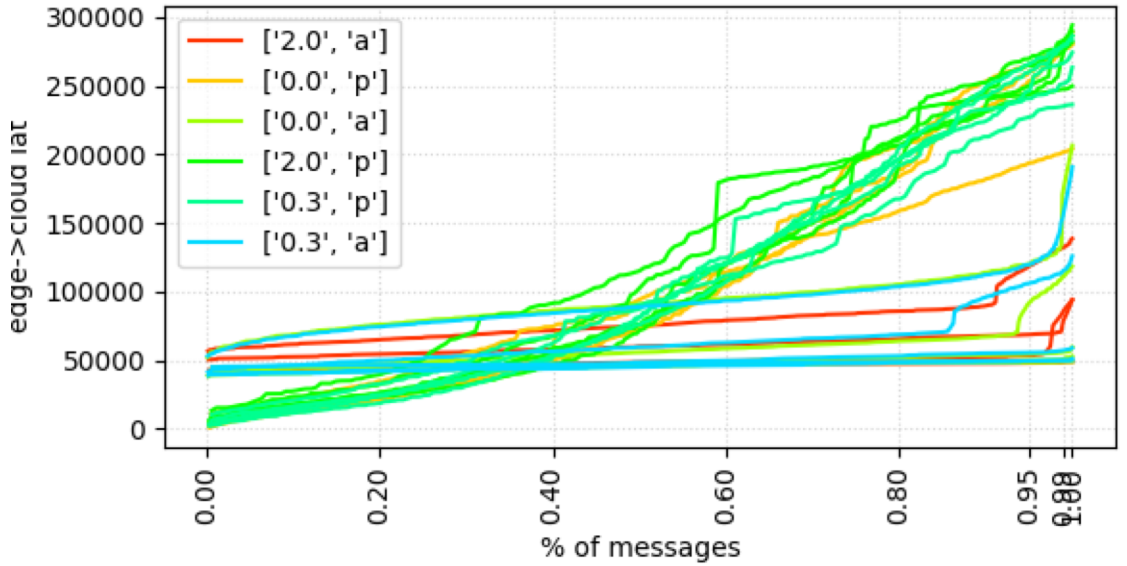
# know your scale

- ❑ project what could be happen when you scale!
- ❑ isolate each part of your experiments in order to find eventual performance leaks



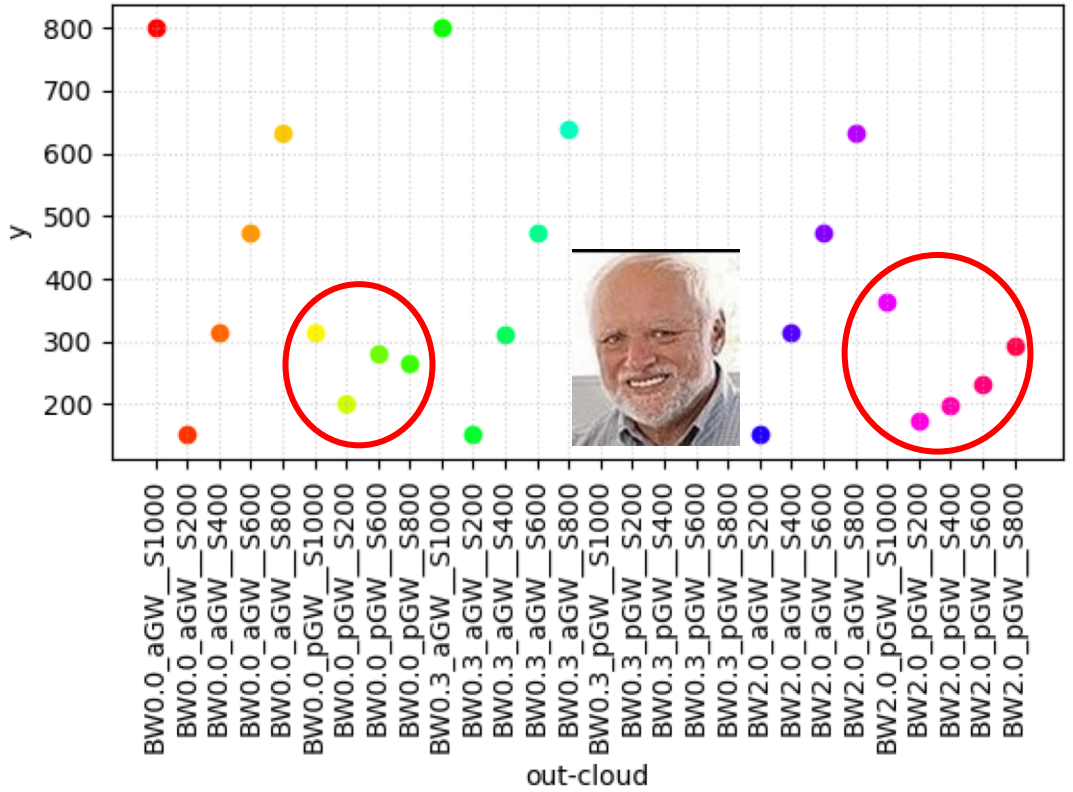
# know your scale

- project what could be happen when you scale!
- isolate each part of your experiments in order to find eventual performance leaks



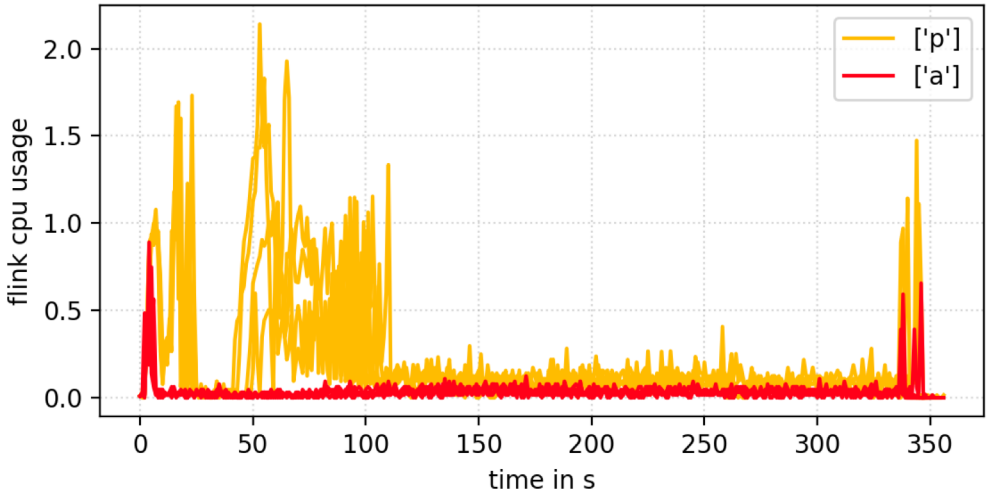
# know your scale

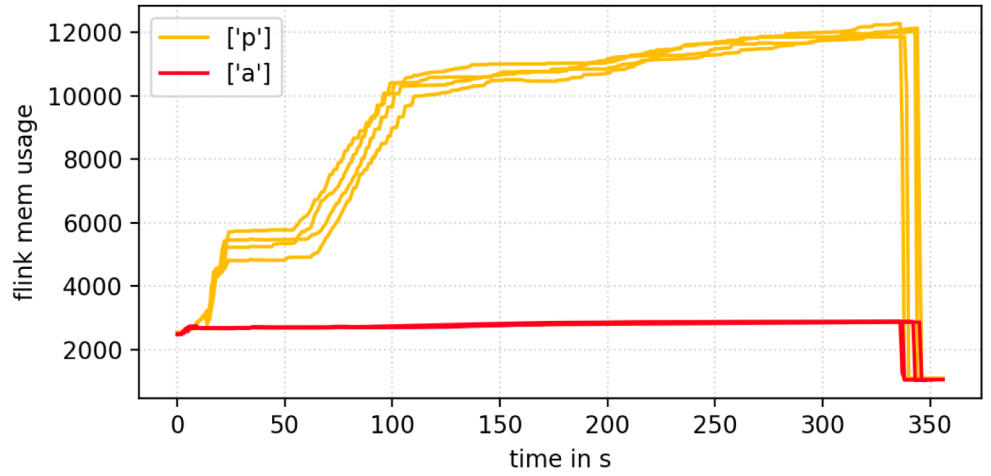
- ❑ project what could be happen when you scale!
- ❑ isolate each part of your experiments in order to find eventual performance leaks



# know your scale

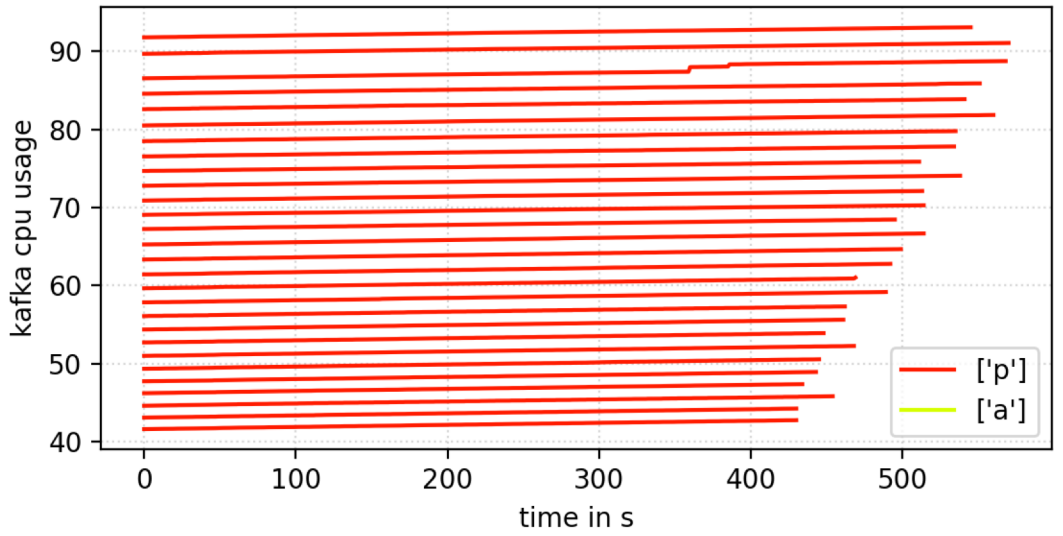
- ❑ project what could be happen when you scale!
- ❑ isolate each part of your experiments in order to find eventual performance leaks



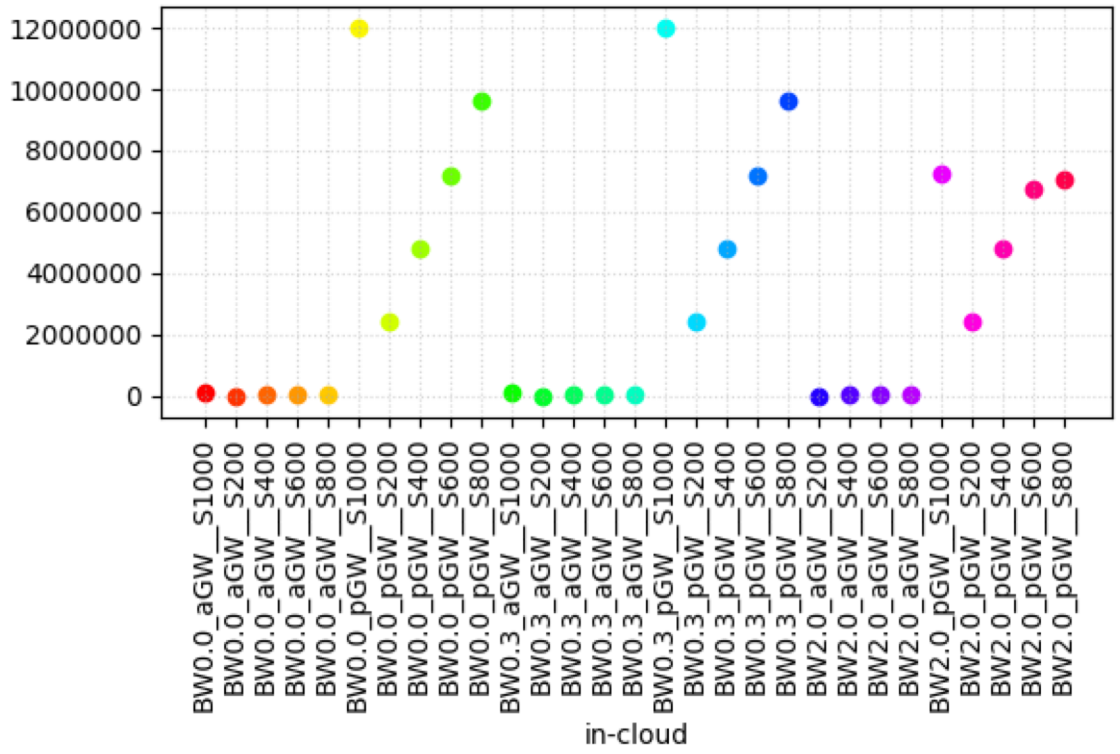
# know your scale

- project what could be happen when you scale!
- isolate each part of your experiments in order to find eventual performance leaks



# know your scale

- ❑ project what could be happen when you scale!
- ❑ isolate each part of your experiments in order to find eventual performance leaks



# überalles: do not forget what you once knew

- an **experiment journal** is the most important tool and your best friend (ok, maybe I don't have enough friends..)
- prevents that **"what was I doing?"** feeling.
- prevents re-fixing bugs from scratch.
- organizes your todo lists