



Observability Day

EUROPE

Real-World Sampling

Lessons Learned After Reducing ~80% of Our O11y Costs

Alexandre Magno Prado Machado, Pismo
Juraci Paixão Kröhling - Grafana Labs

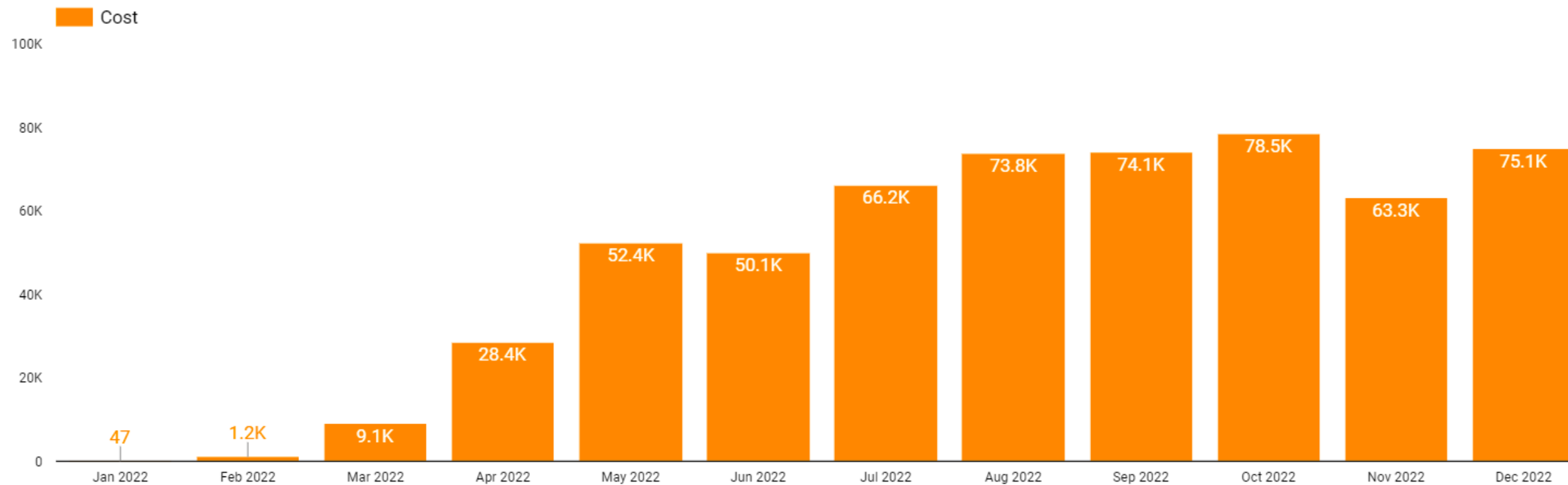
Our company is under exponential growth. Should our observability costs also grow exponentially?



Real-World Sampling



Observability Traces



Last Update: 02/01/2024

Agenda

- What's sampling
- Sampling strategies
- Why we need sampling
- How we did it
- Outcomes
- Next steps

Sampling is the technique used to reduce the amount of telemetry data that is generated or stored.

Real-World Sampling



Observability Day
EUROPE

Sampling for tracing

Limits the number of traces by making a decision and applying it to all spans belonging to the same trace. The decision can be made:

- When the root span is created (head)
- At the collector layer
- After the trace is complete (tail)

Traces - Head

- Typically uses a probabilistic strategy
- Uses context propagation to apply the decision consistently to child spans

Traces - Head

- ↑ No transmission costs for non-sampled data
- ↑ Easy to configure
- ↓ Harder to apply consistently
- ↓ Statistics: probability of 10% $\neq$ 10%

Traces - Collector

- Typically uses a probabilistic strategy
- Uses the trace ID to apply the decision consistently to spans ($\text{hash}(\text{traceID}) \% n$)

Traces - Collector

- ↑ Easy to apply to all services at once
- ↑ Easy to configure
- ↓ Operational costs of a collector
- ↓ Statistics: probability of 10% $\neq$ 10%

Traces - Tail

- Typically uses complex strategies
- Stores traces in memory and makes a decision based on the trace as a whole
- Requires all spans from the same trace to be sent to the same collector instance

Traces - Tail

- ↑ Allows selecting highly relevant traces
- ↑ Multiple, complex conditions
- ↓ Higher operational costs (more memory!)
- ↓ Stateful, harder to scale

Downsides

- It is harder to extract statistics
- Potentially misses important telemetry
- More complex to operate

Why we needed sampling

- Exponential growth company
- Hopefully without exponential observability costs
- While keeping the visibility of the services

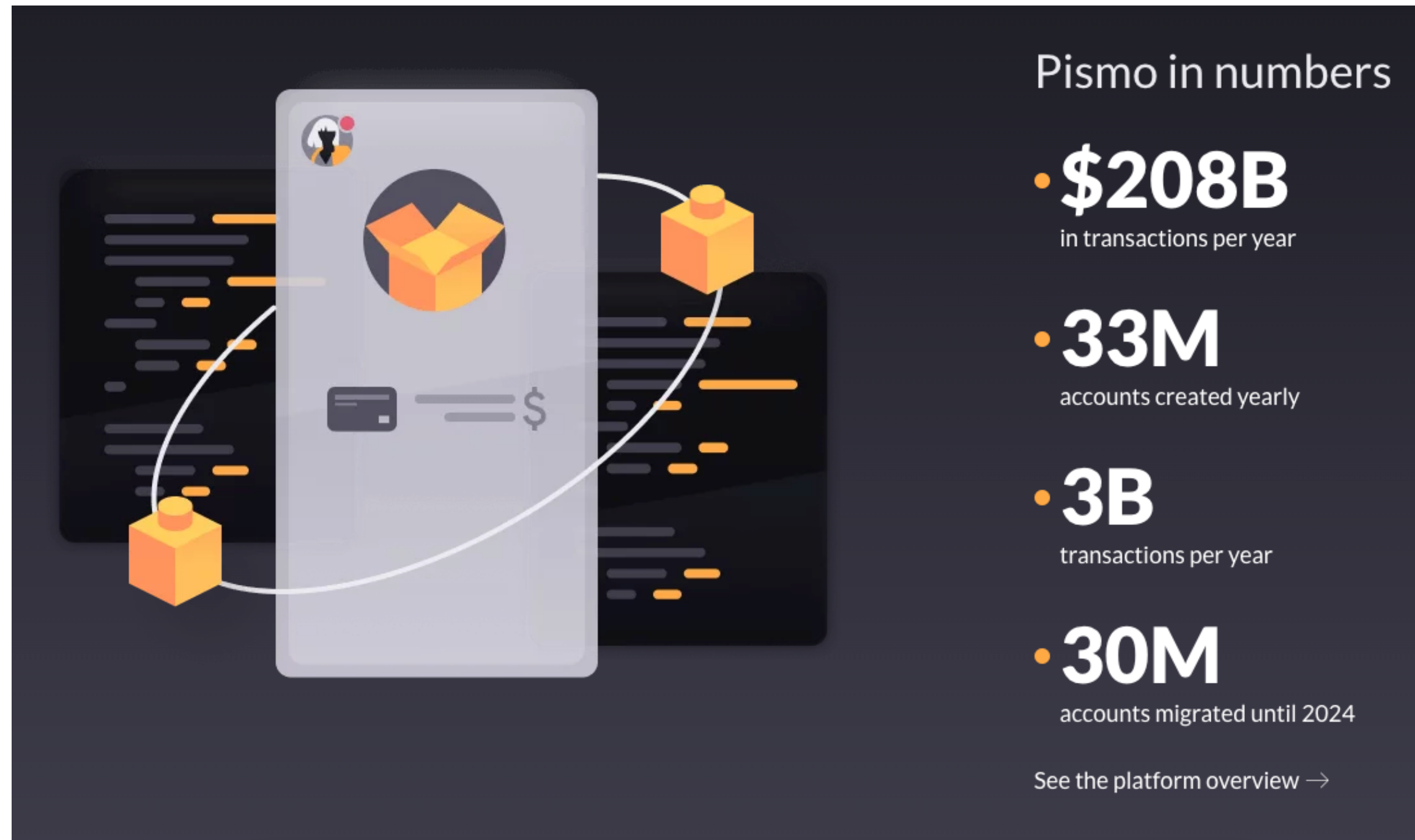


Exponential growth company

- Pismo is a company providing banking as a service
- Large customers around the world
- Critical operation



Pismo in numbers



Control observability costs

- Company growth can impact Observability costs
- Eventually impacts the financial margins of products



Real-World Sampling

Keep the services
observable

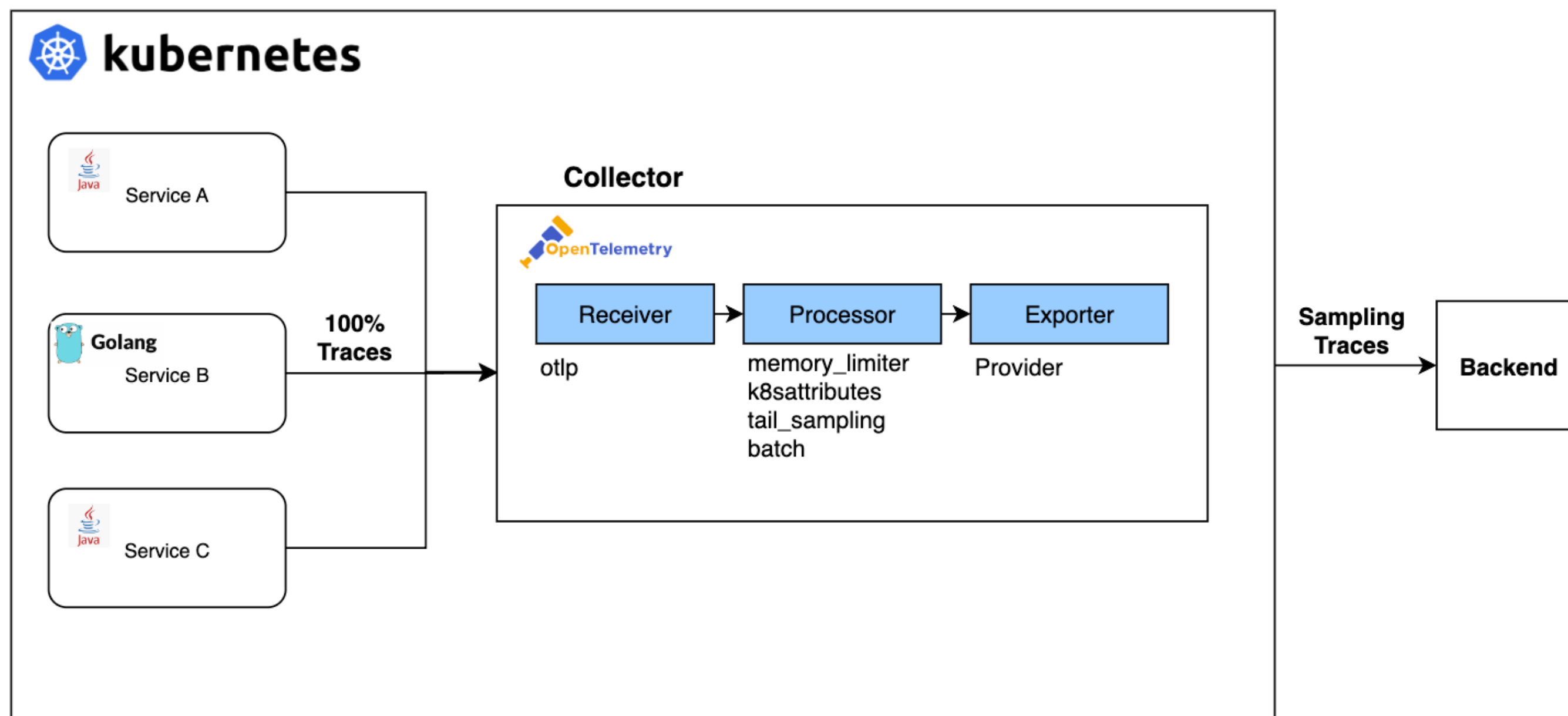
Cost control cannot impact the
ability to observe our services.



How we did it

- Choosing the sampling strategy
 - Head or tail sampling
- Why tail sampling
 - Keep visibility in case of the crash of apps
 - Fine control with policies
 - Tuning

OpenTelemetry Diagram



Real-World Sampling

```
policies:
[
  {
    name: errors-policy,
    type: status_code,
    status_code: { status_codes: [ERROR] },
  },
  {
    name: filter-by-service-name-policy,
    type: string_attribute,
    string_attribute: { key: service.name, values: [app - name] },
  },
  { name: latency-policy, type: latency, latency: { threshold_ms: 5000 } },
  {
    name: randomized-policy,
    type: probabilistic,
    probabilistic: { sampling_percentage: 25 },
  },
]
```

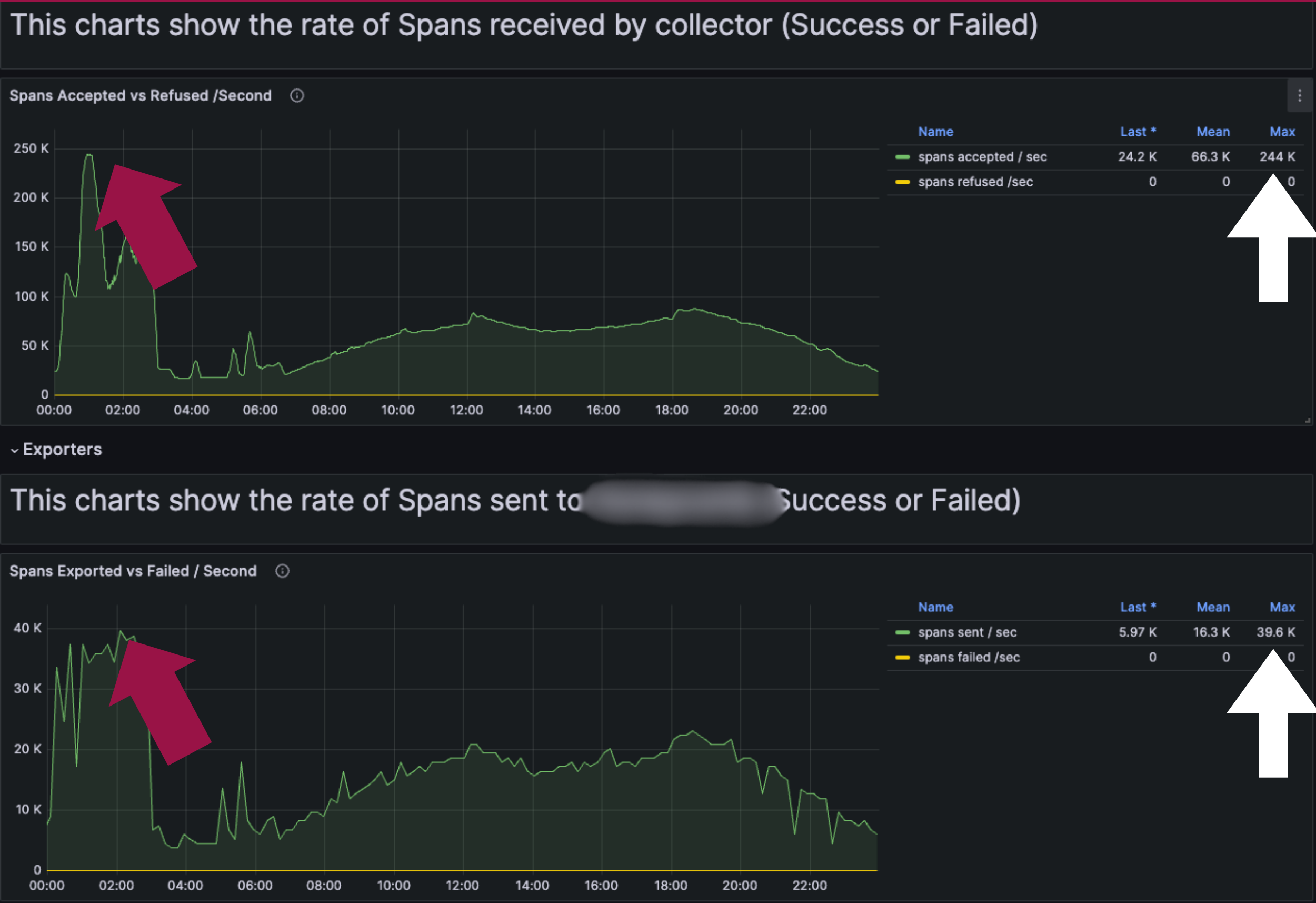
Observability Day
EUROPE

Real-World Sampling

```
tail_sampling:  
  decision_wait: 5s  
  num_traces: 120000
```

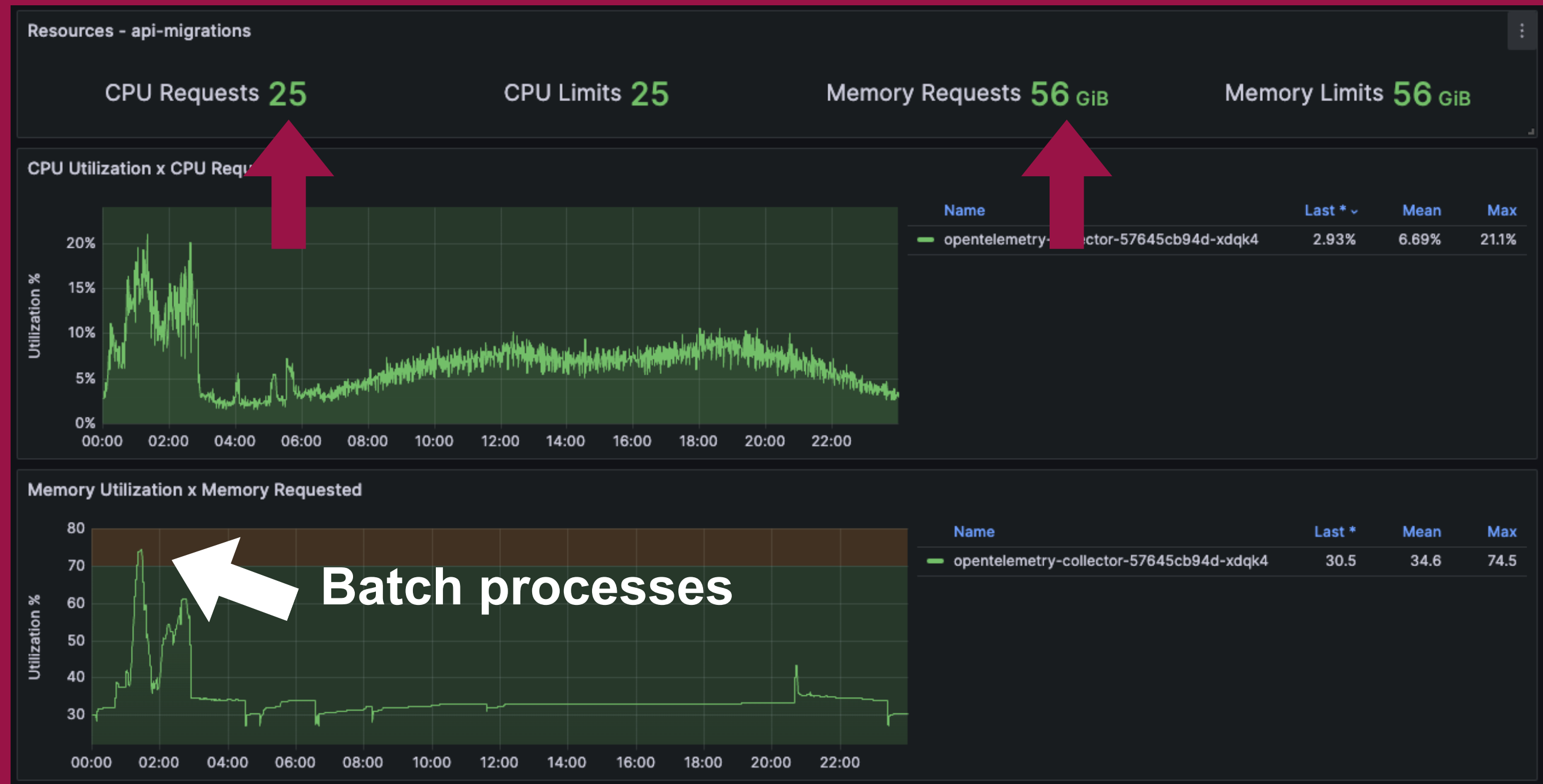
Observability Day
EUROPE

Real-World Sampling



Real-World Sampling

Pod Resources



Real-World Sampling

Lesson #1 The Collector can handle it

We observed that we can grow the Collector instance vertically and it will handle our load, avoiding operational complexity.



Lesson #2 Up to a certain level...

But there are limits to vertical scaling! After a couple of years, we finally hit the limit and we **NEED** to have a load-balancing layer now.



Lesson #3 Collector crashes, it's fine

Instead of trying to have a highly available Collector, we accepted that it might crash from time to time. Within 4s, it's back to business.

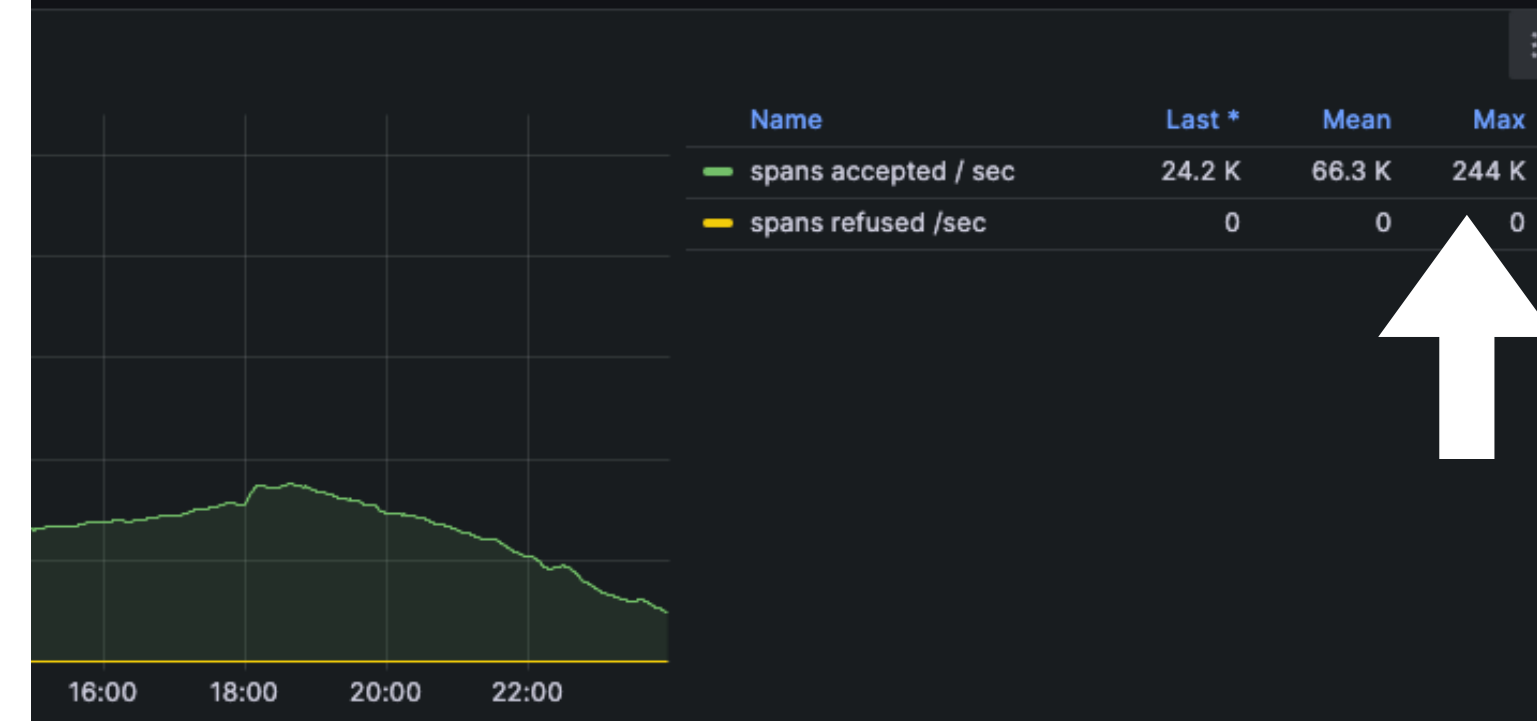


Real-World Sampling

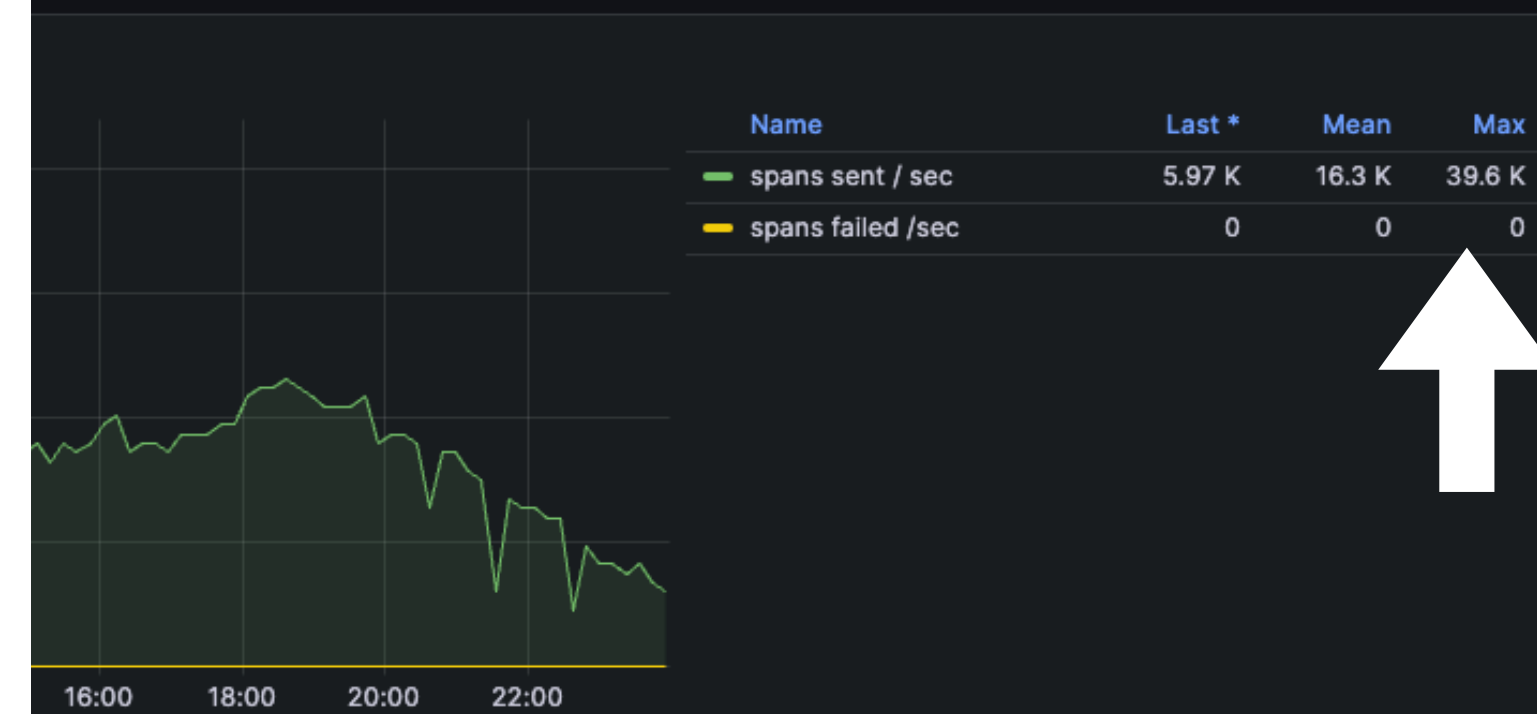
Lesson #4 Probability of 10% $\neq$ 10%

39.6k isn't 25% of 244k, it's ~16%

ed by collector (Success or Failed)



o Success or Failed)

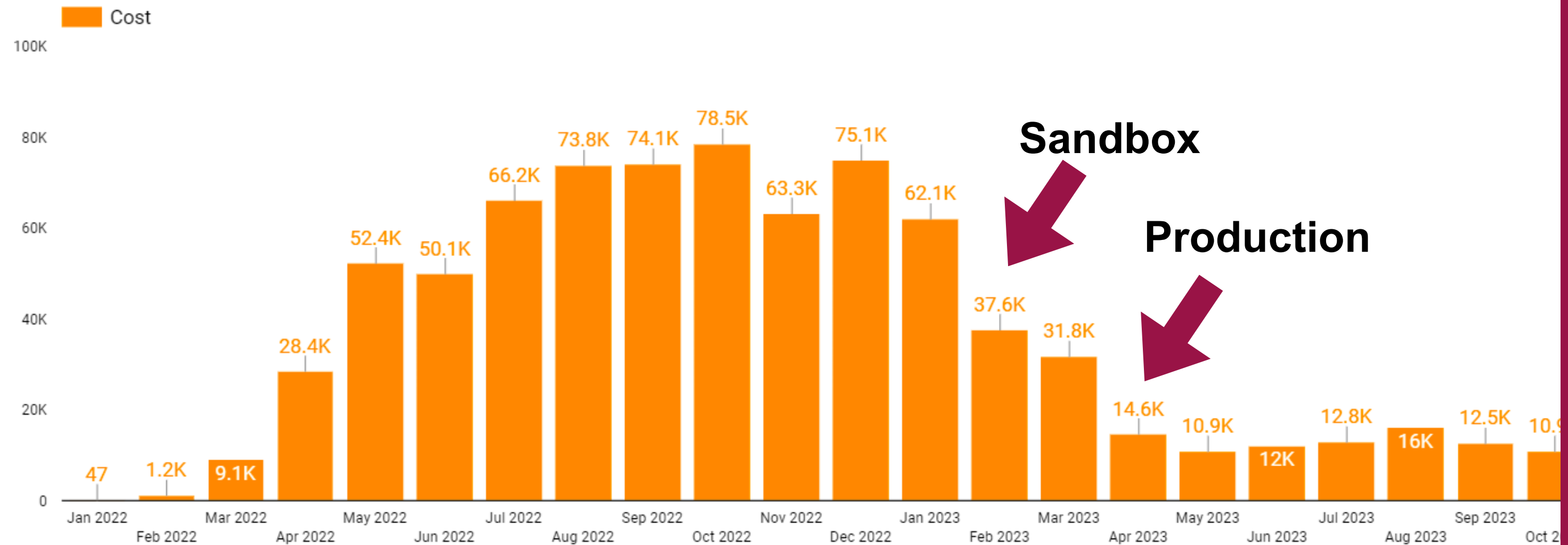


Lesson #5 Last resort for error detection

With our current policy, we can easily tell when errors are happening, even if they are not being caught in earlier layers.



Observability Traces



Next Steps

- Implement Load Balancer
- Implement HPA



Thank you!

Observability Day
EUROPE

19 March 2024 | Paris, France



**Alexandre Magno
Prado Machado**

Tech Lead Observability
Pismo



Juraci Paixão Kröhling

Software Engineer
Grafana Labs