

Observability Day

EUROPE

Real-World Sampling

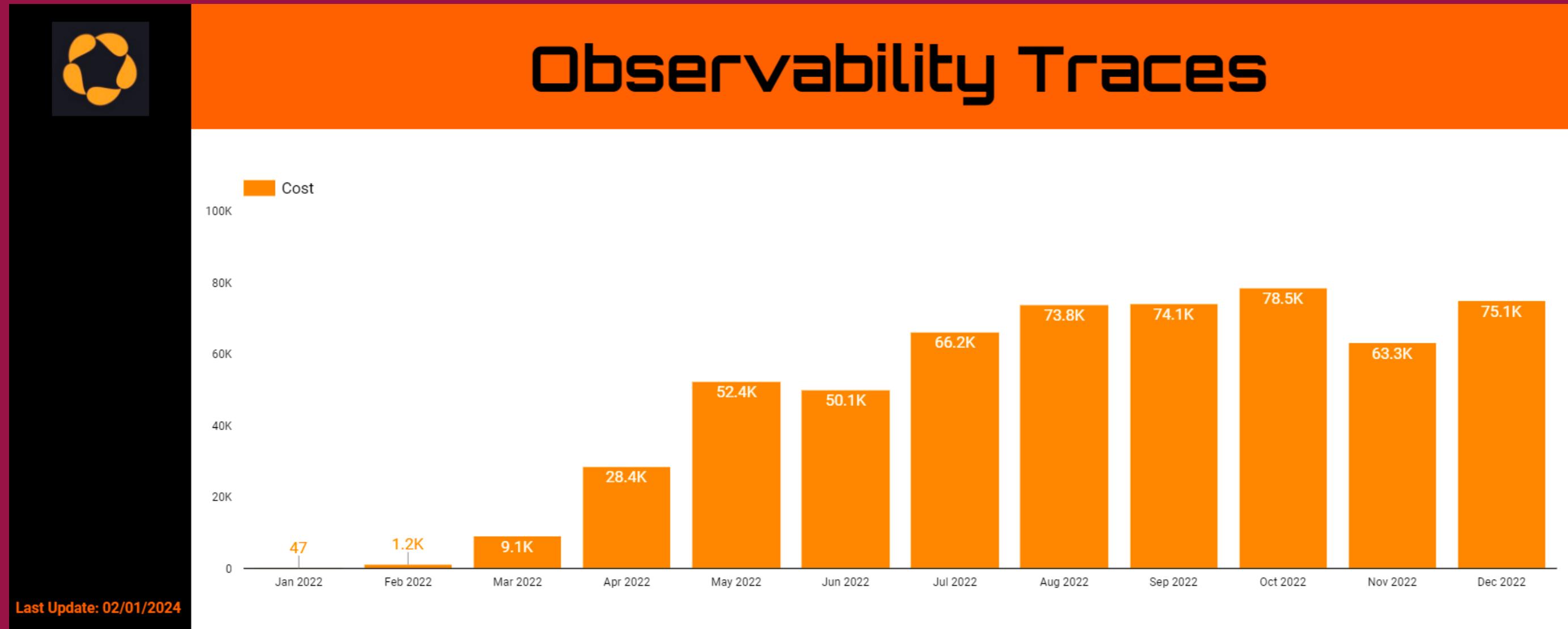
Lessons Learned After Reducing ~80% of Our O11y Costs

*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



Real-World Sampling



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



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)

Real-World Sampling



Traces - Head

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

Real-World Sampling



Traces - Head

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

Real-World Sampling



Traces - Collector

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

Real-World Sampling



Traces - Collector

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

Real-World Sampling



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

Real-World Sampling



Traces - Tail

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

Real-World Sampling



Downsides

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

Real-World Sampling



Why we needed sampling

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

Real-World Sampling

Exponential growth company

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



Real-World Sampling



Pismo in numbers

A central graphic features a light gray rectangular card with a circular icon containing an orange and yellow box. Below the card is a small icon of a credit card next to a dollar sign. Two orange juice cartons are shown; one is positioned directly below the card, and another is connected by a curved white line extending from the bottom left of the card towards the right. The background is dark with abstract horizontal bars and small orange dots scattered across it.

Pismo in numbers

- **\$208B**
in transactions per year
- **33M**
accounts created yearly
- **3B**
transactions per year
- **30M**
accounts migrated until 2024

See the platform overview →

Real-World Sampling

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.



Real-World Sampling



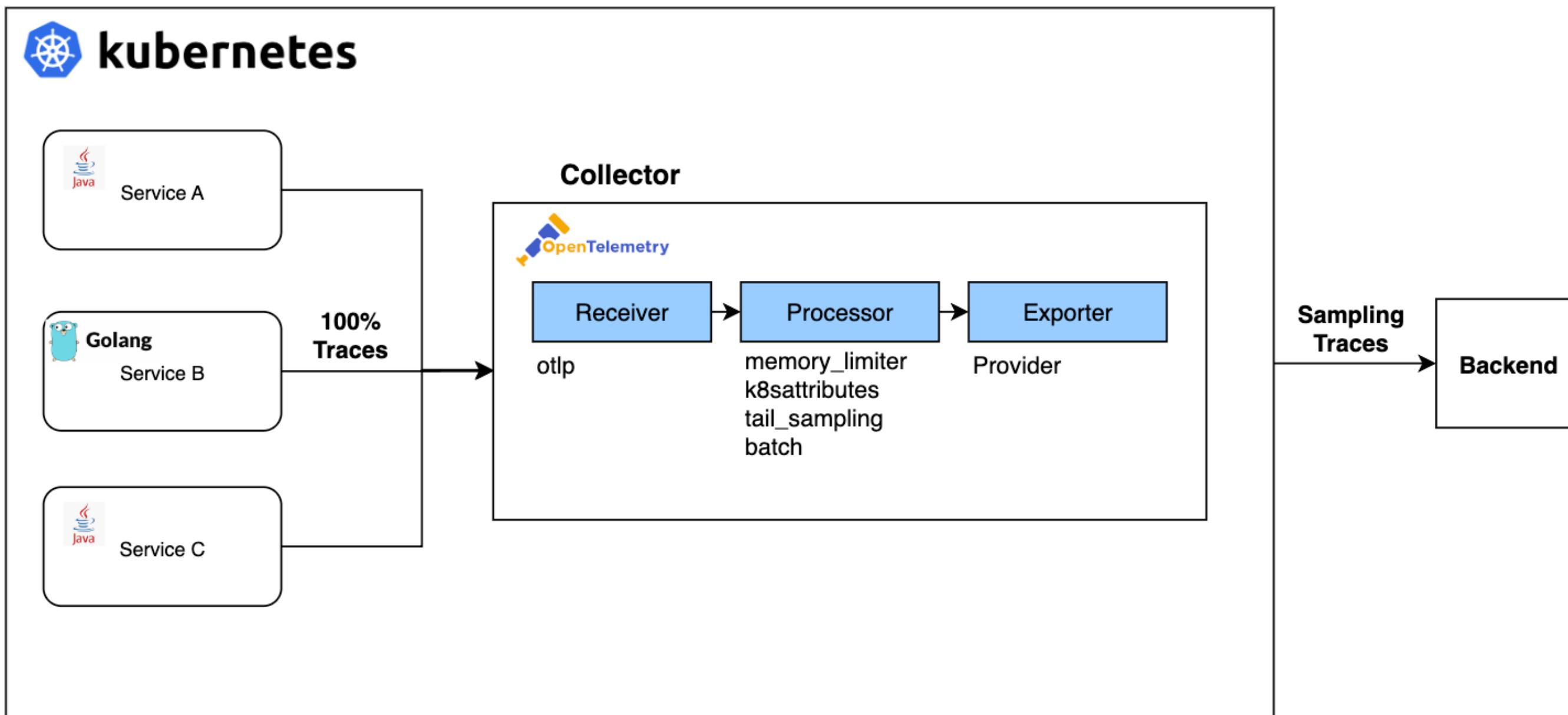
How we did it

- 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

Real-World Sampling



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 },  
  },  
]
```



Real-World Sampling

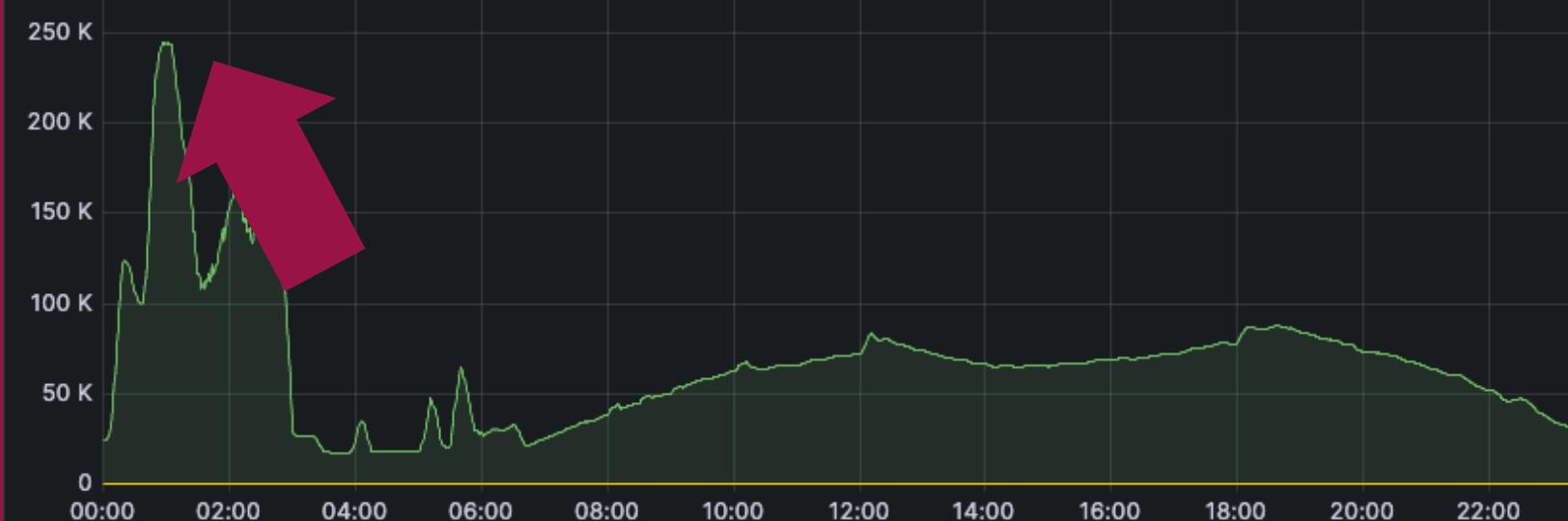
```
tail_sampling:  
decision_wait: 5s  
num_traces: 120000
```



Real-World Sampling

This charts show the rate of Spans received by collector (Success or Failed)

Spans Accepted vs Refused /Second



▼ Exporters

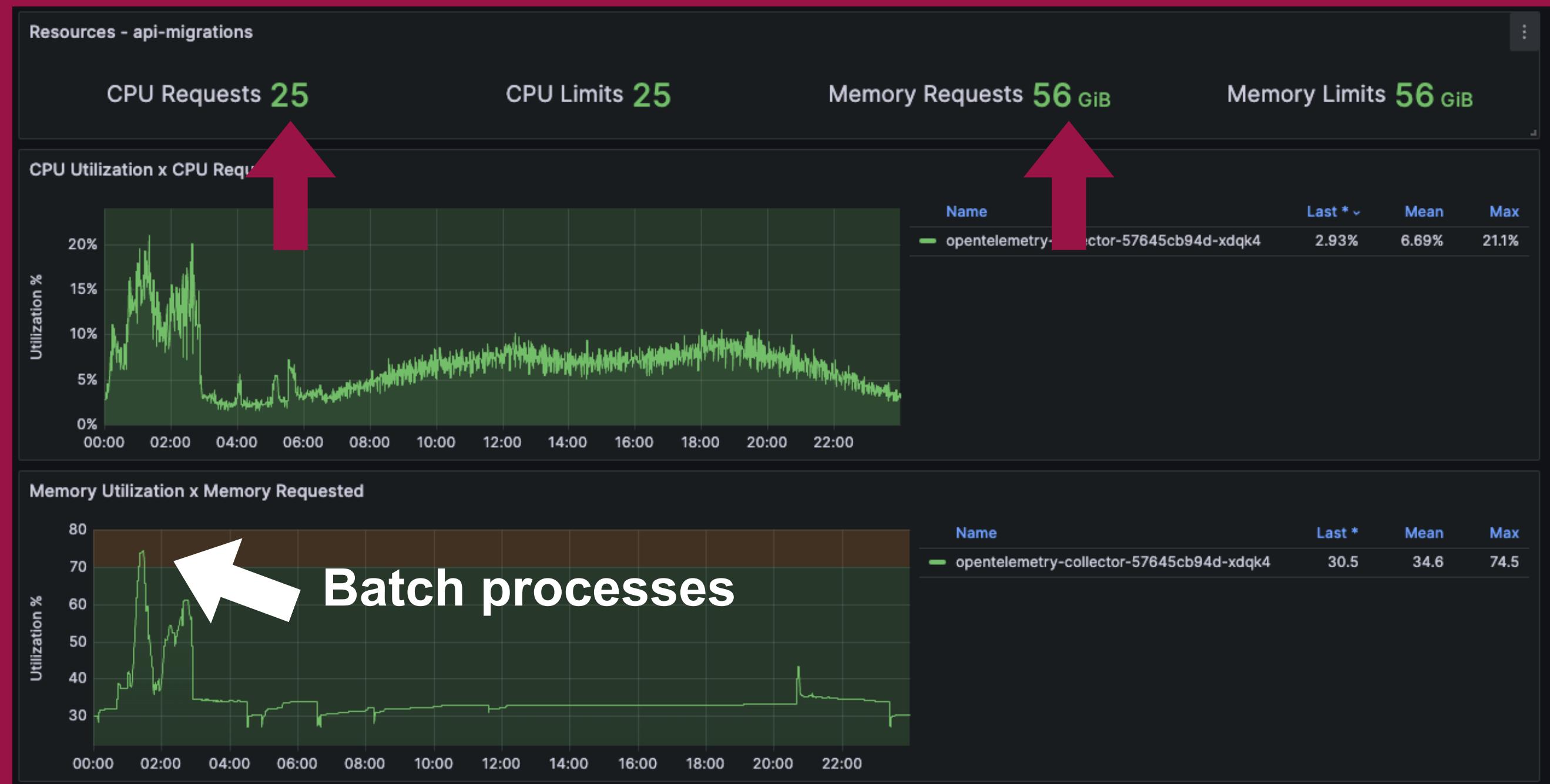
This charts show the rate of Spans sent to [REDACTED] Success or Failed)

Spans Exported vs Failed / Second



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.



Real-World Sampling

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.



Real-World Sampling

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

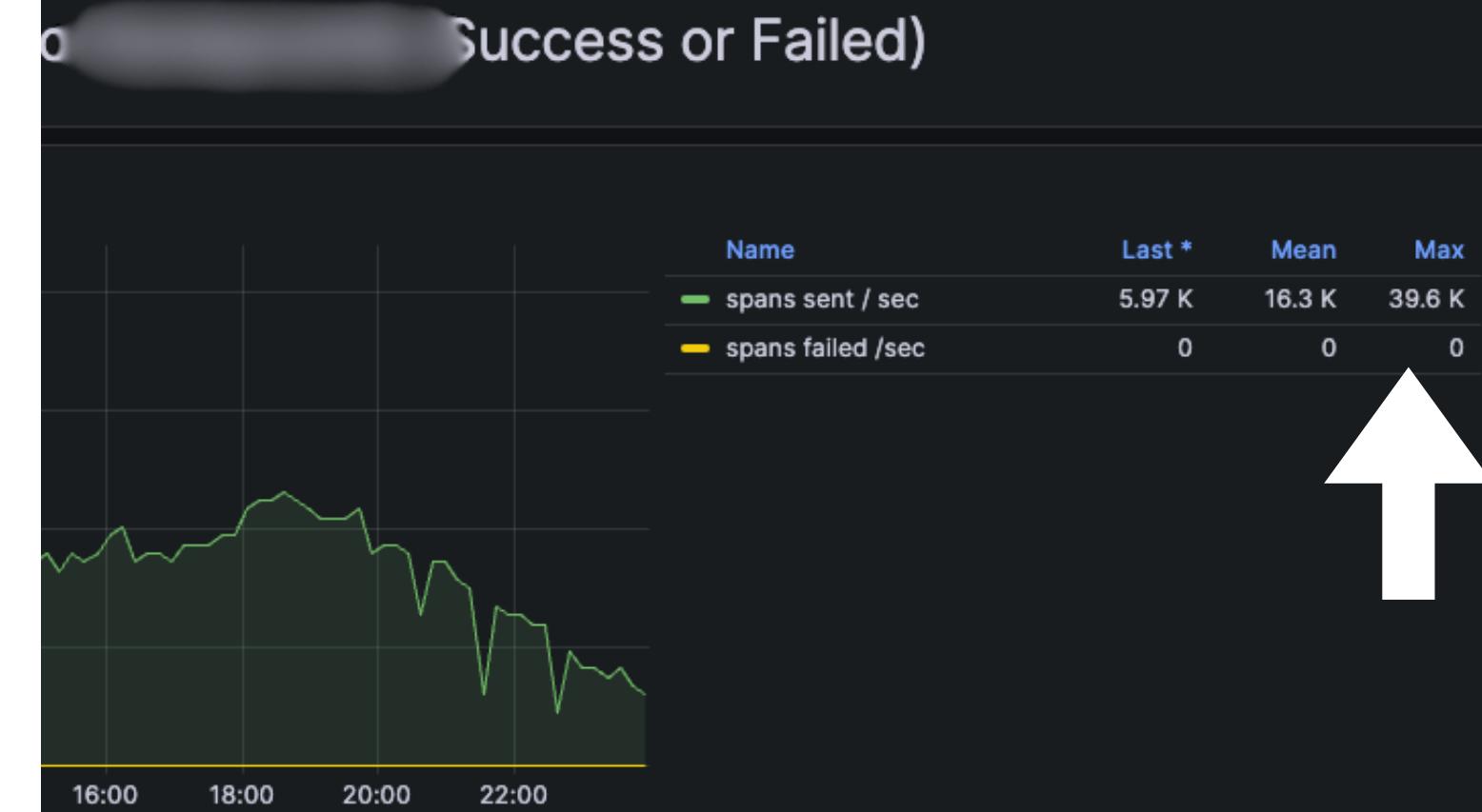
Lesson #4
Probability of 10% \neq 10%

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

Spans received by collector (Success or Failed)



Spans sent by collector (Success or Failed)



Real-World Sampling



Lesson #5 Last resort for error detection

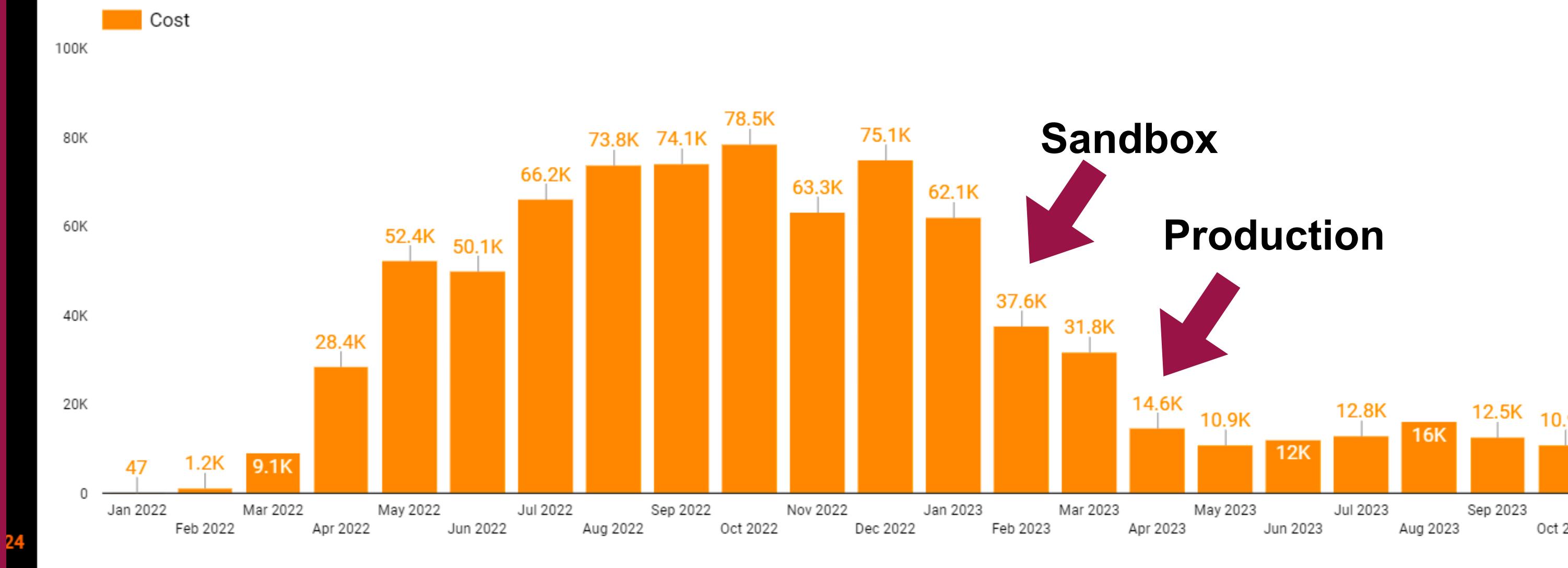
With our current policy, we can easily tell when errors are happening, even if they are not being caught in earlier layers.



Real-World Sampling



Observability Traces



Real-World Sampling



Next Steps

- Implement Load Balancer
- Implement HPA



Thank you!



19 March 2024 | Paris, France



**Alexandre Magno
Prado Machado**
Tech Lead Observability
Pismo



Juraci Paixão Kröhling
Software Engineer
Grafana Labs