Observability at Scale:

How Open Systems Collects Telemetry from Over 10'000 Edge Devices *Worldwide*

Observability Day

Joel Verezhak

18.04.2023

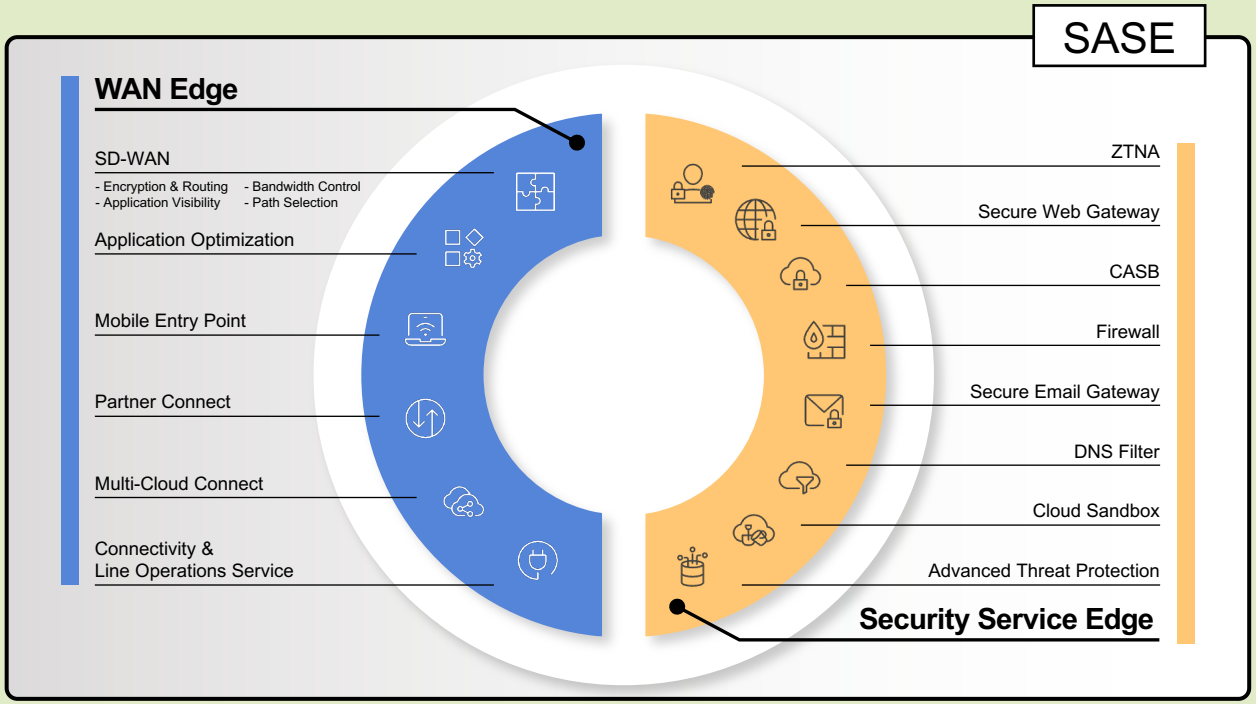
KubeCon | CloudNativeCon Europe 2023

Agenda

- What we do (the *Status Quo*)
- Challenges and Action Plan
- Putting it into motion
- Outlook and Horizon

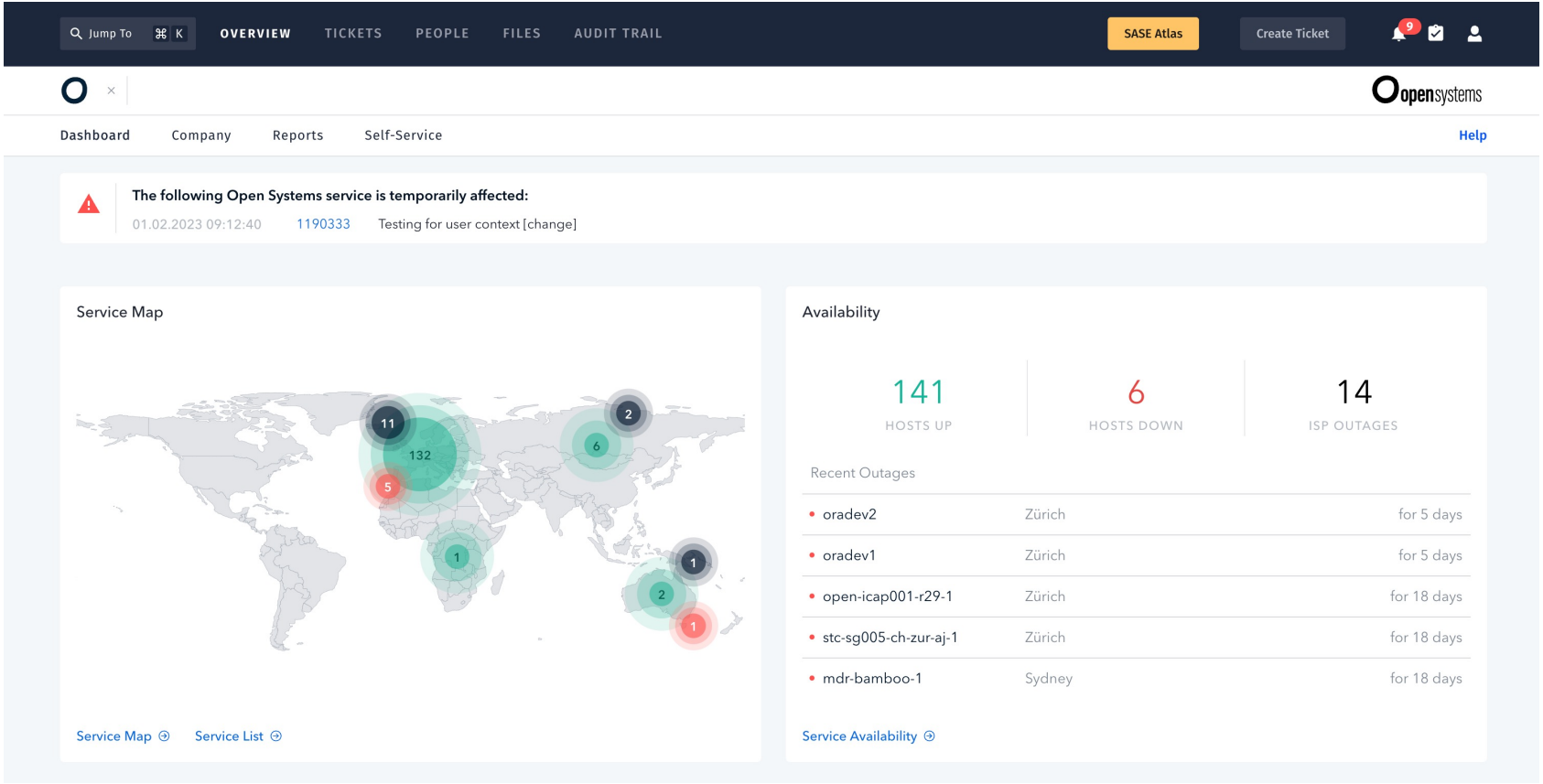
What we do: Open Systems Managed SASE

Managed SASE



- Centralised policy management
- 24/7 Global NOC
- HW and SW lifecycle
- Expert Support
- Data observability/analytics
- Cloud and On-premises Edges
- Unified data platform
- Single pane of glass

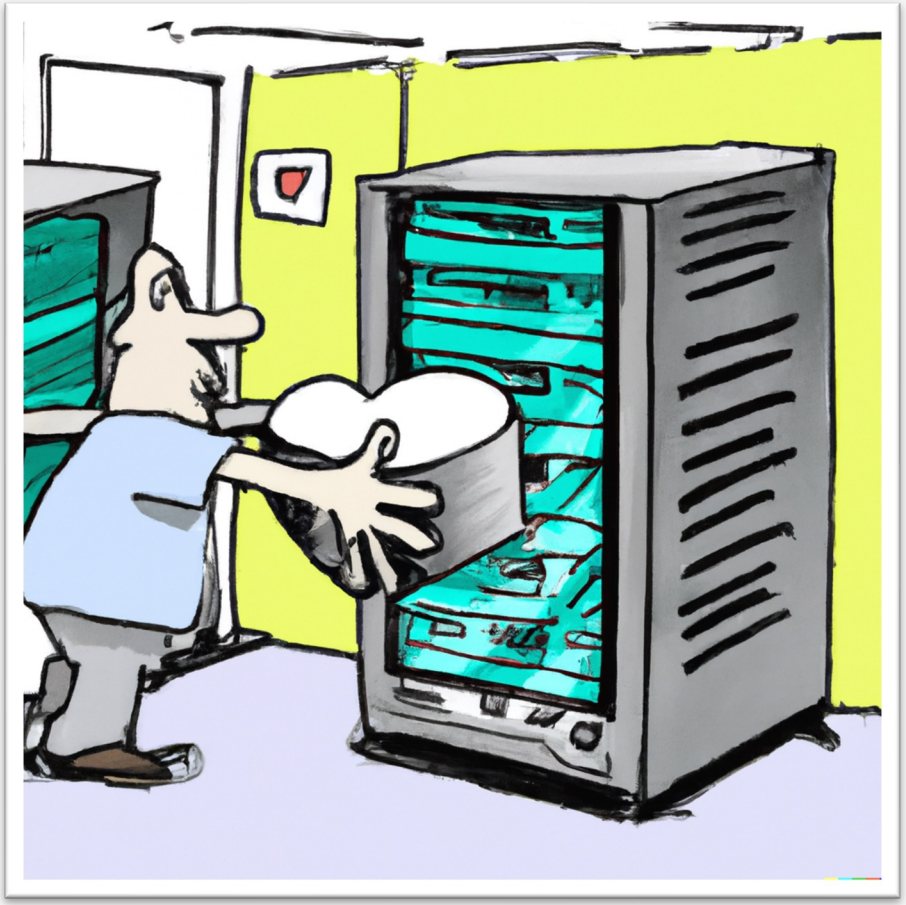
The Portal

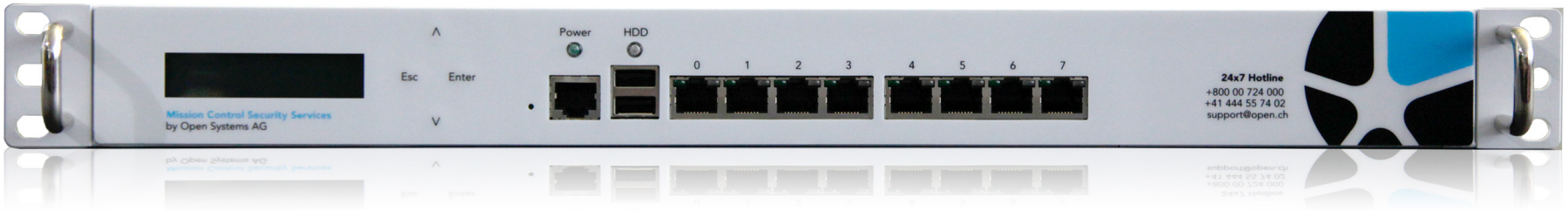


The status quo

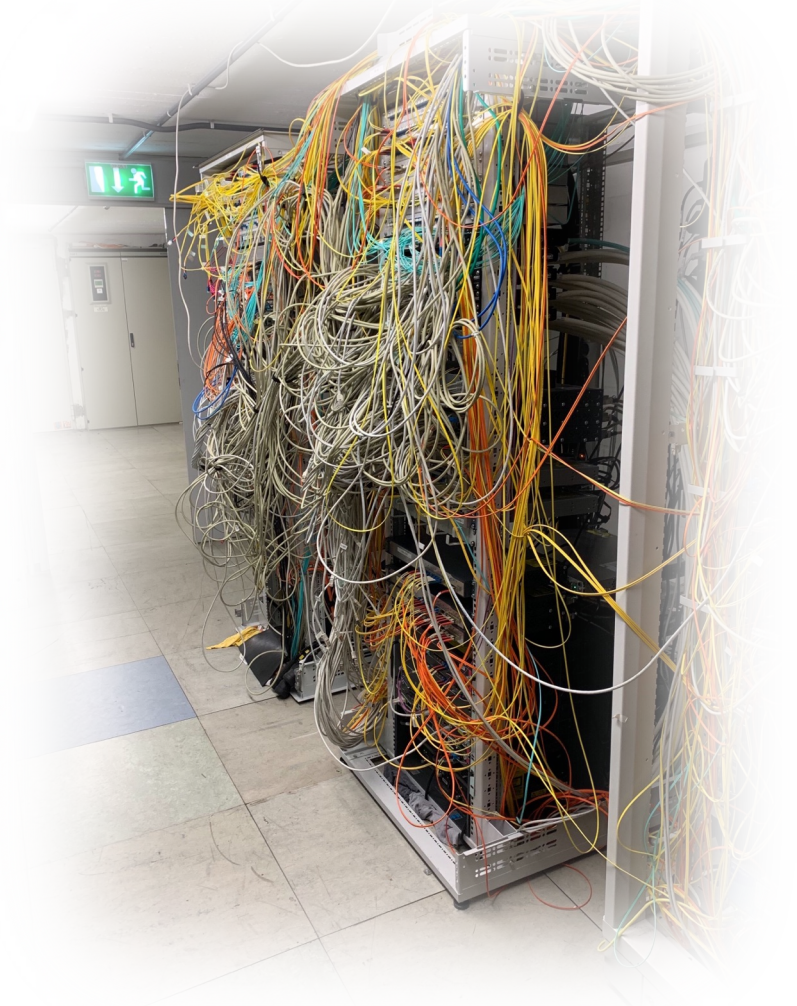
Our fleet of edge devices

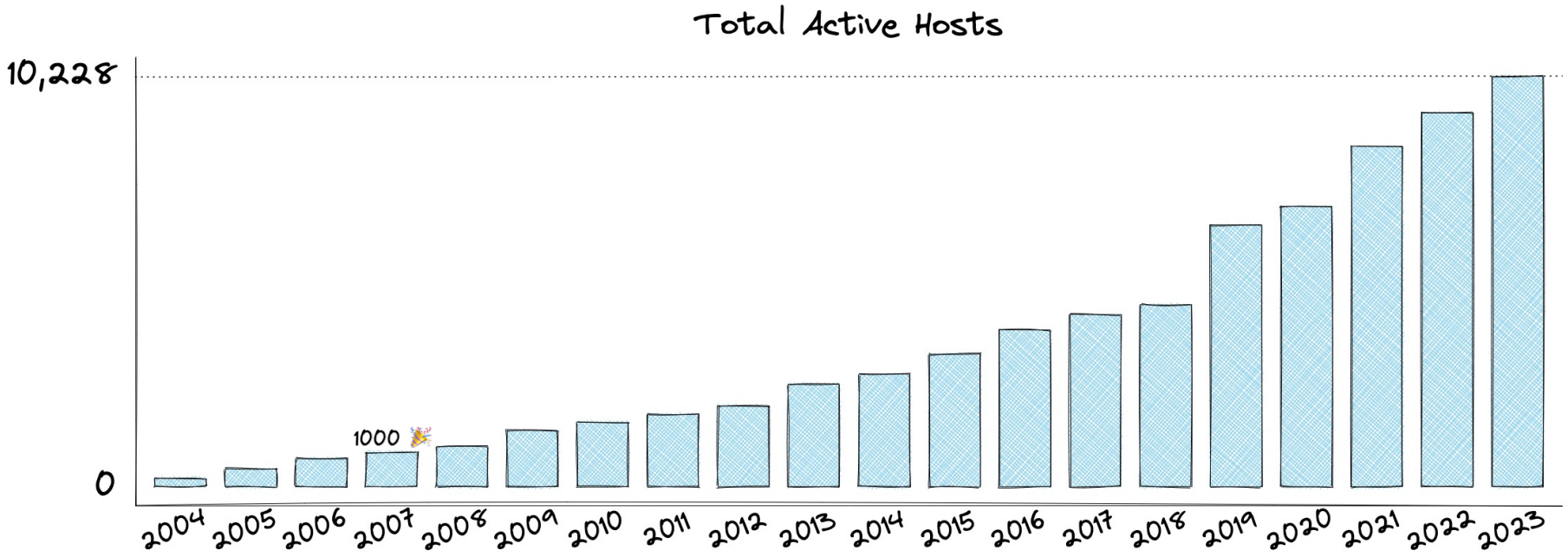
- **Individually named**
- **Lovingly provisioned in-house before shipment to customer**
- **Monitored constantly in our Network Operations Centre**





¯_(ツ)_/¯



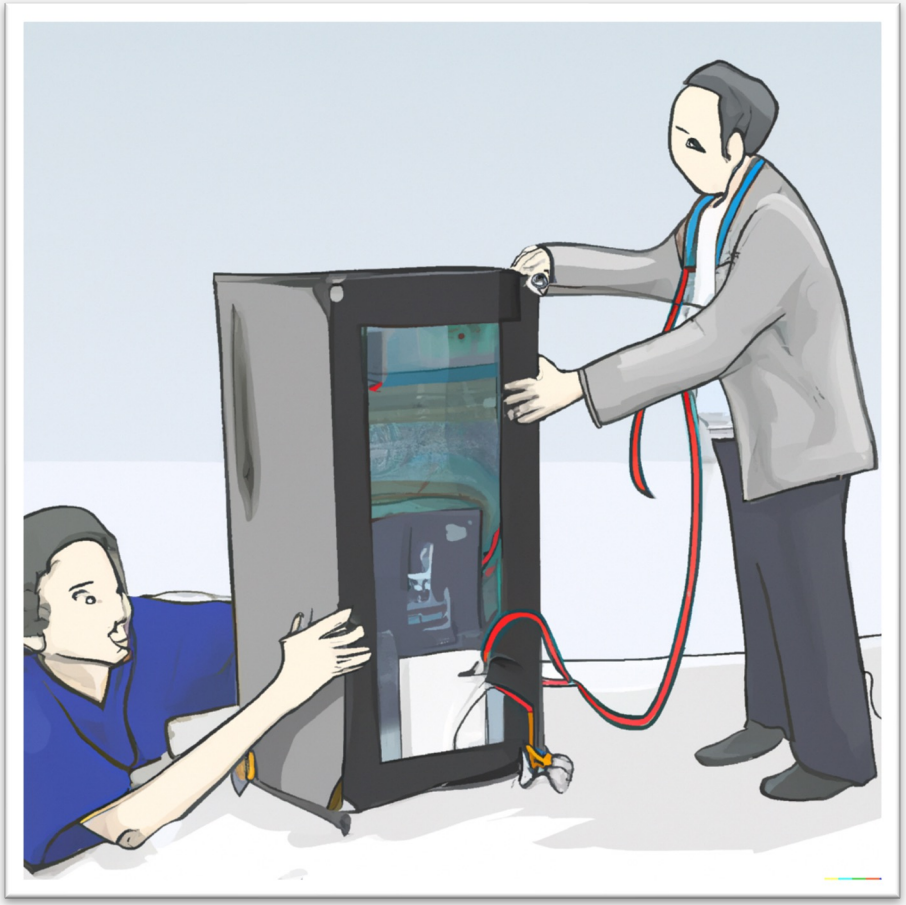


The status quo

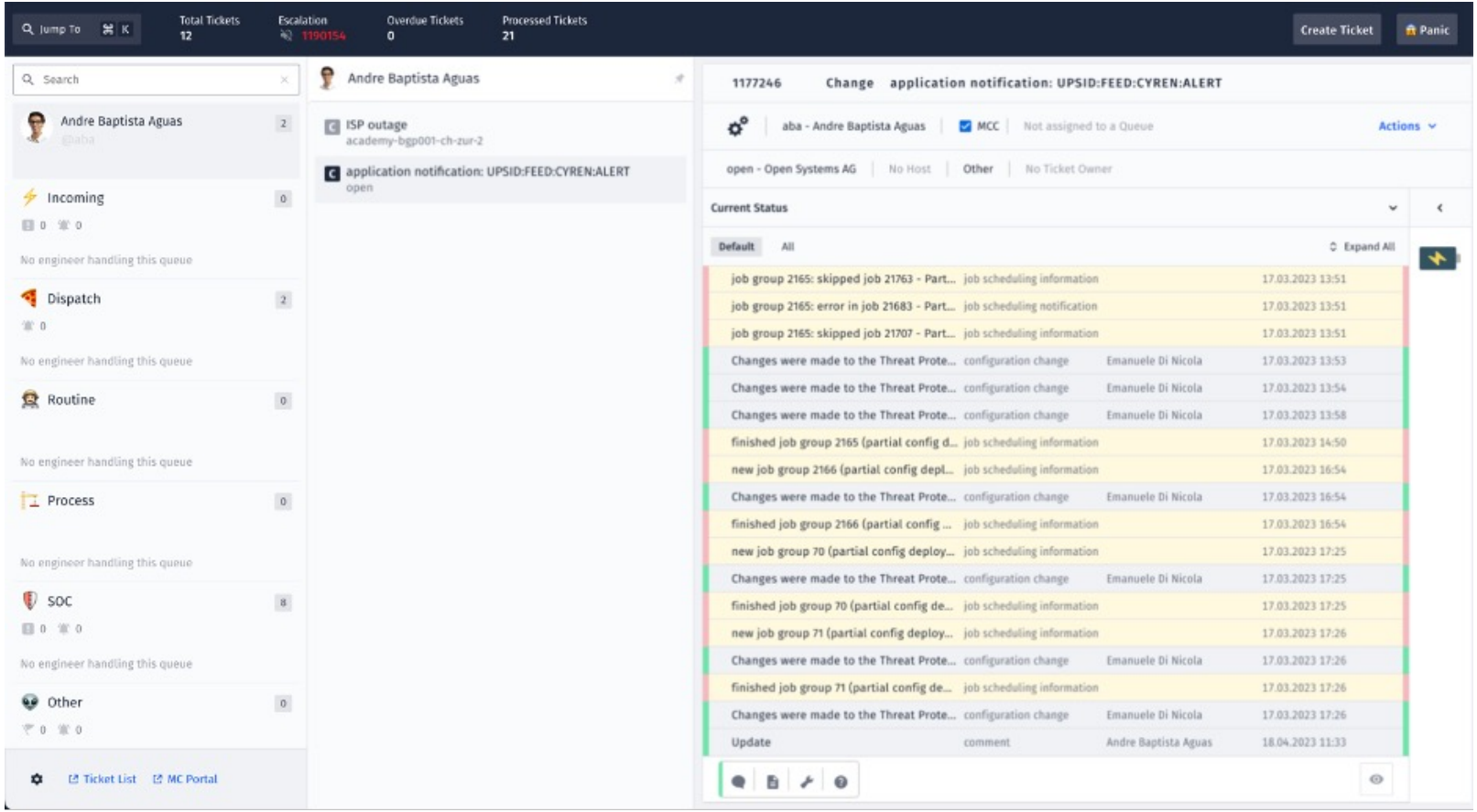
What happens when a device gets sick?

- **Device lovingly nursed back to health by Mission Control**

1. **ssh into the box**
2. **grep the logs**
3. **init.d status**
4. **…**



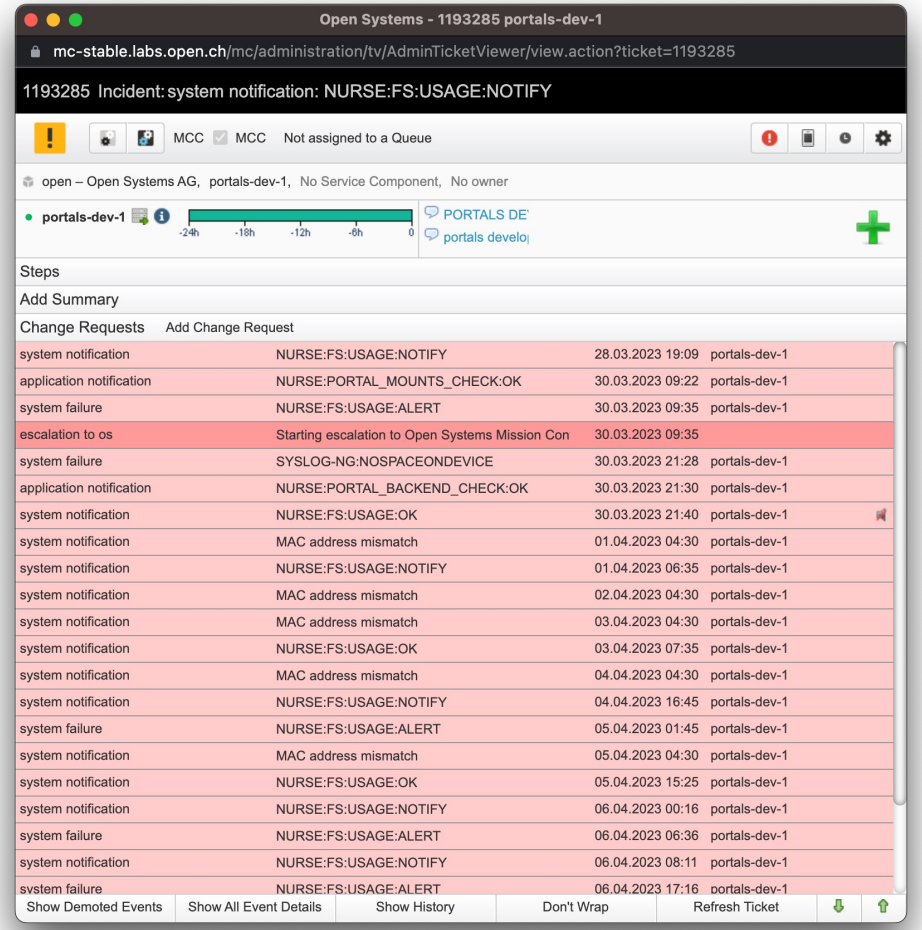
Mission Control



Mission Control

Anatomy of a Ticket

When something goes wrong on a device, it ends up in a ticket



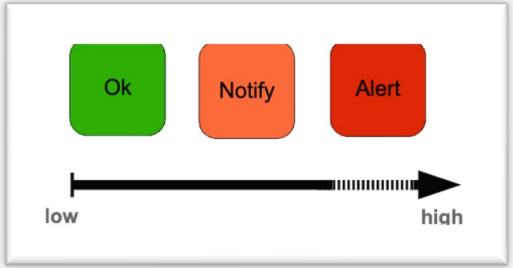
Monitoring the fleet

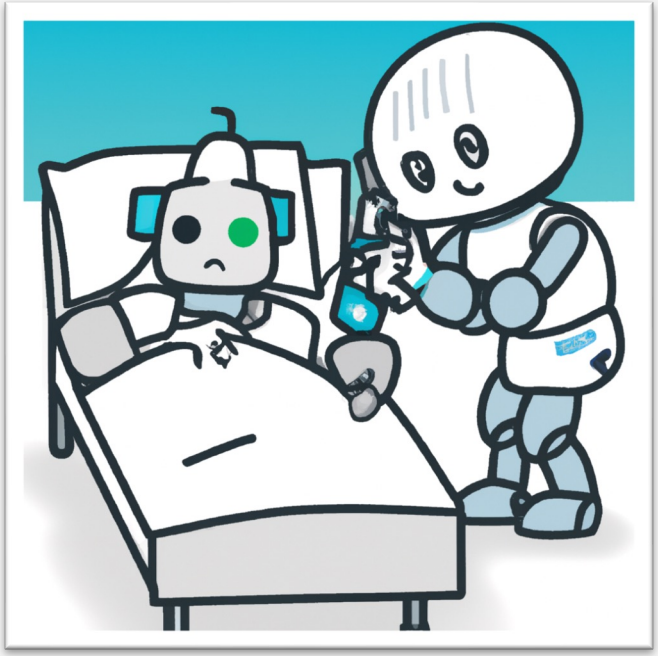
How can we tell when one of our pets is sick?

- **Service Nurses**
- **GUMA (applications)**
- **Metrics**

Service Nurses *Application Monitoring*

- **Small scripts that monitor critical system components**
- **Emit notifications (*log lines*) if predefined threshholds are met**
- **Each nurse is effectively a *state machine:***



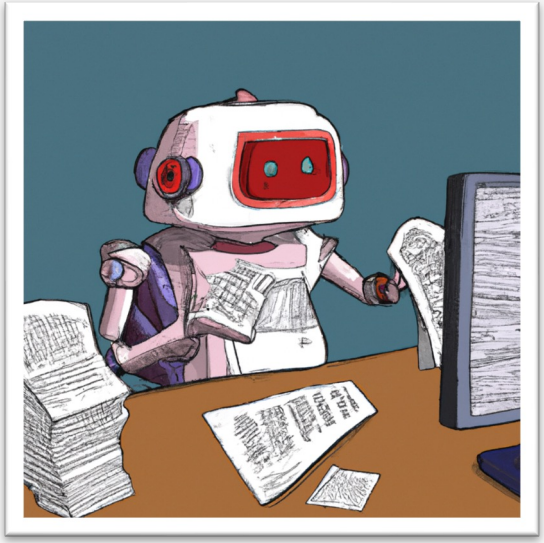


```
NURSE:MEM:USAGE:RAM:NOTIFY 5min average used RAM is 95% (notification limit 95%) [state=Notify, observed=900s]
```

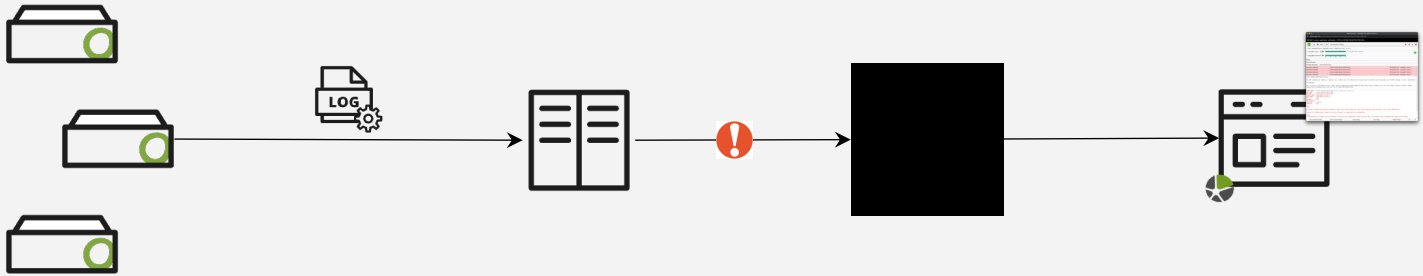
Tag **Reason** **Meta**

GUMA *Application Monitoring*

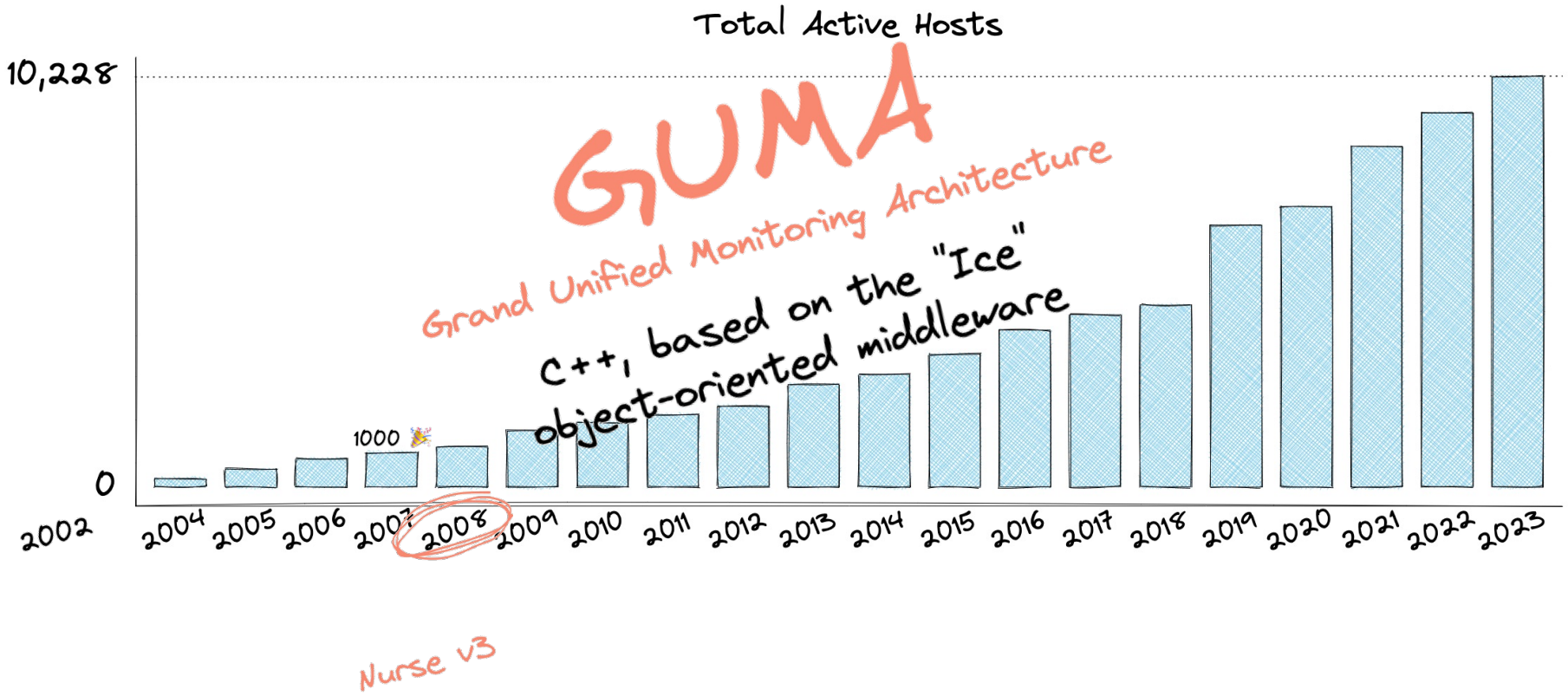
Grand Unified Monitoring Architecture



- **Parse, classify and filter syslog**
- **Match log lines against a list of ~3000 regexes that map to *signatures***
- **Process signatures, decide whether to alert (generate a ticket in Mission Control)**

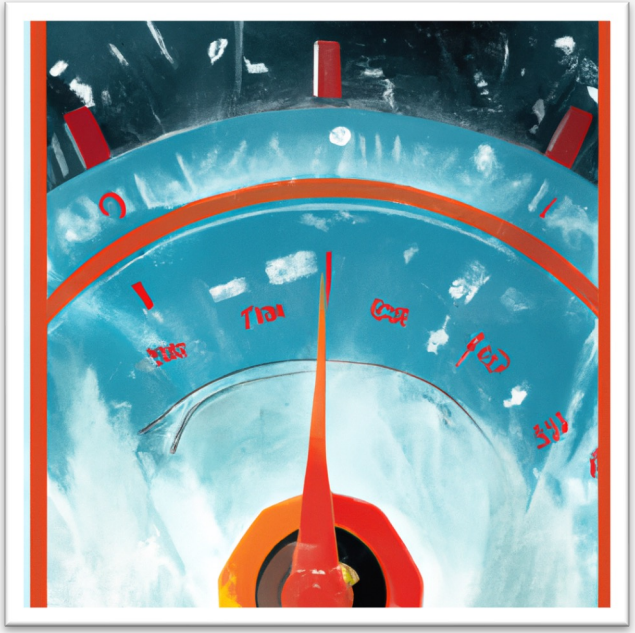


Customer Fleet | GUMA Server | Implication Engine | Mission Control

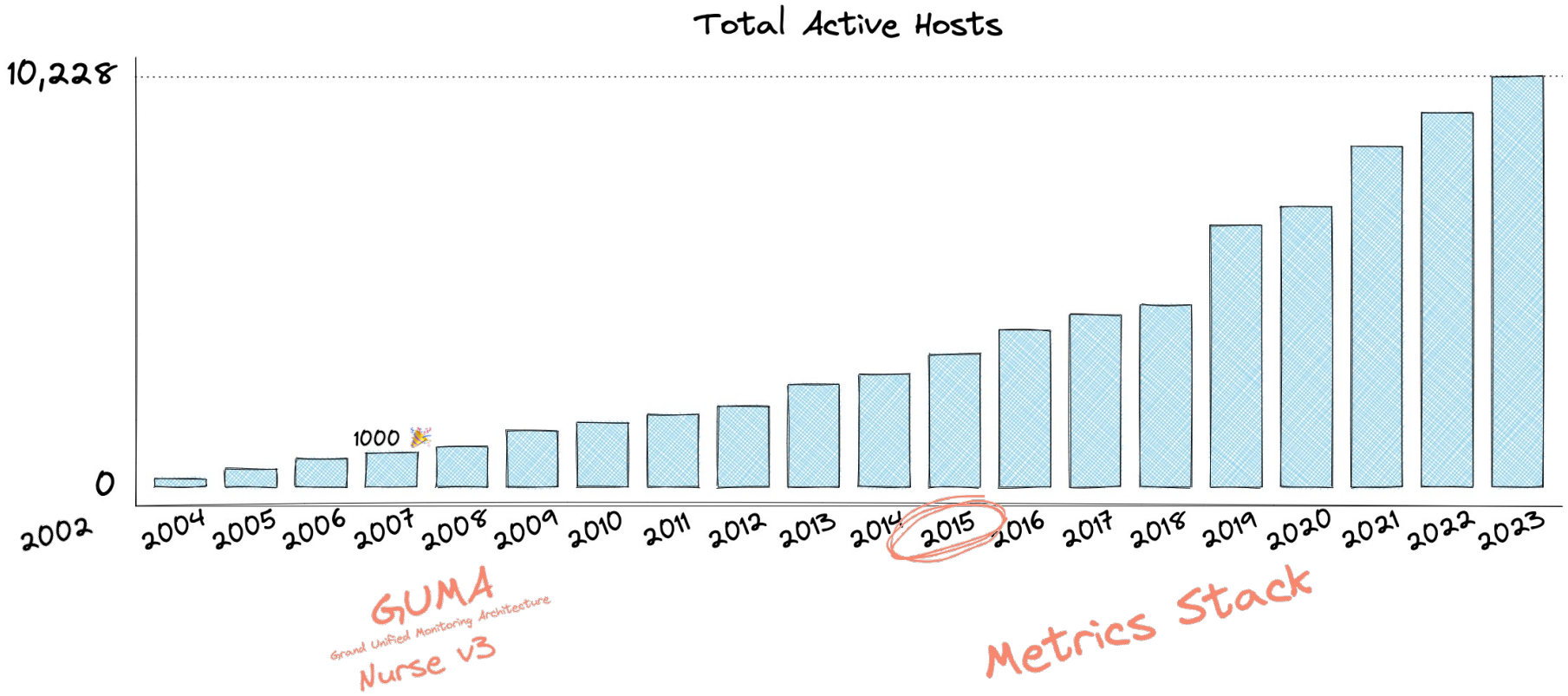


Metrics

Dashboards for the portal



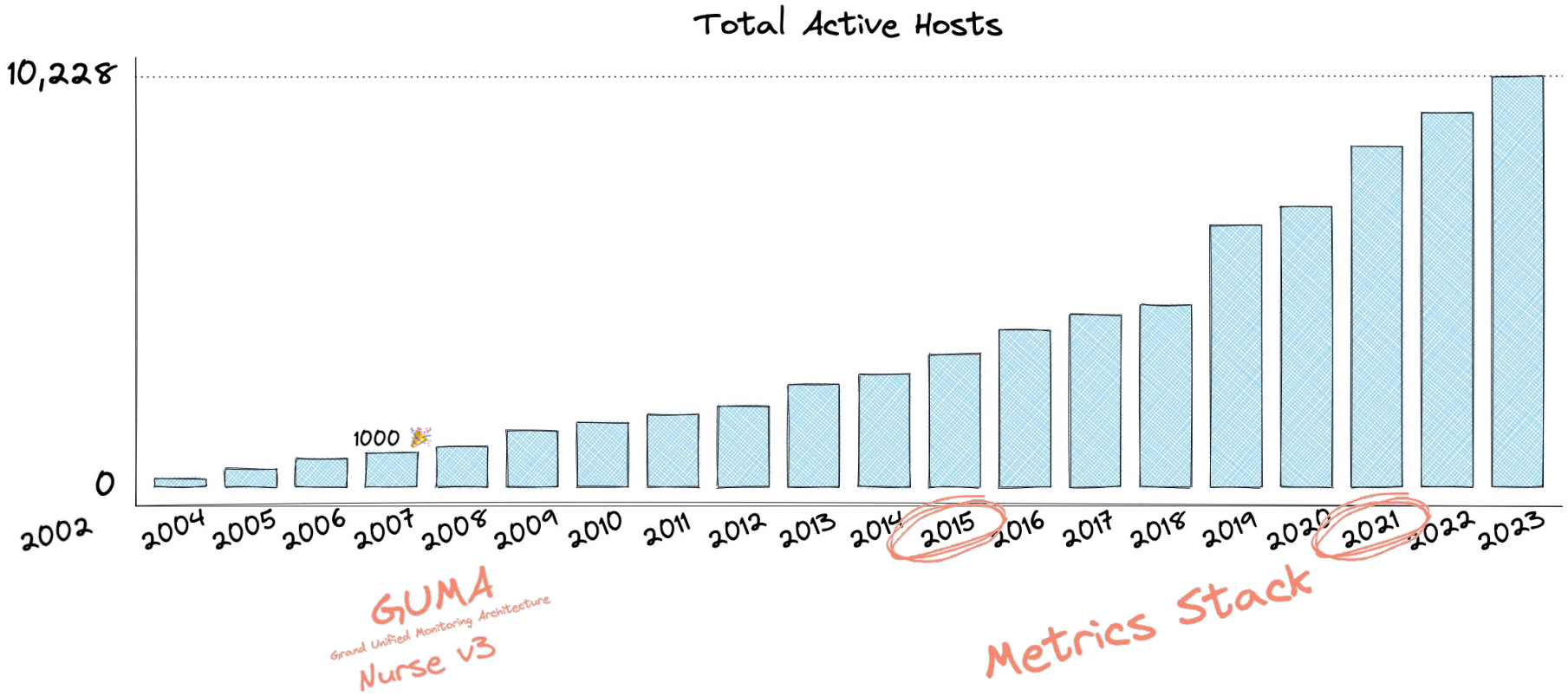
- **Scrape locally on the host with Prometheus**
- **Publish to Kafka topic with Telegraf**
- **Consume into InfluxDB**
- **Continuous queries yield drill-down statistics for the Portal**



InfluxDB

a13

i33



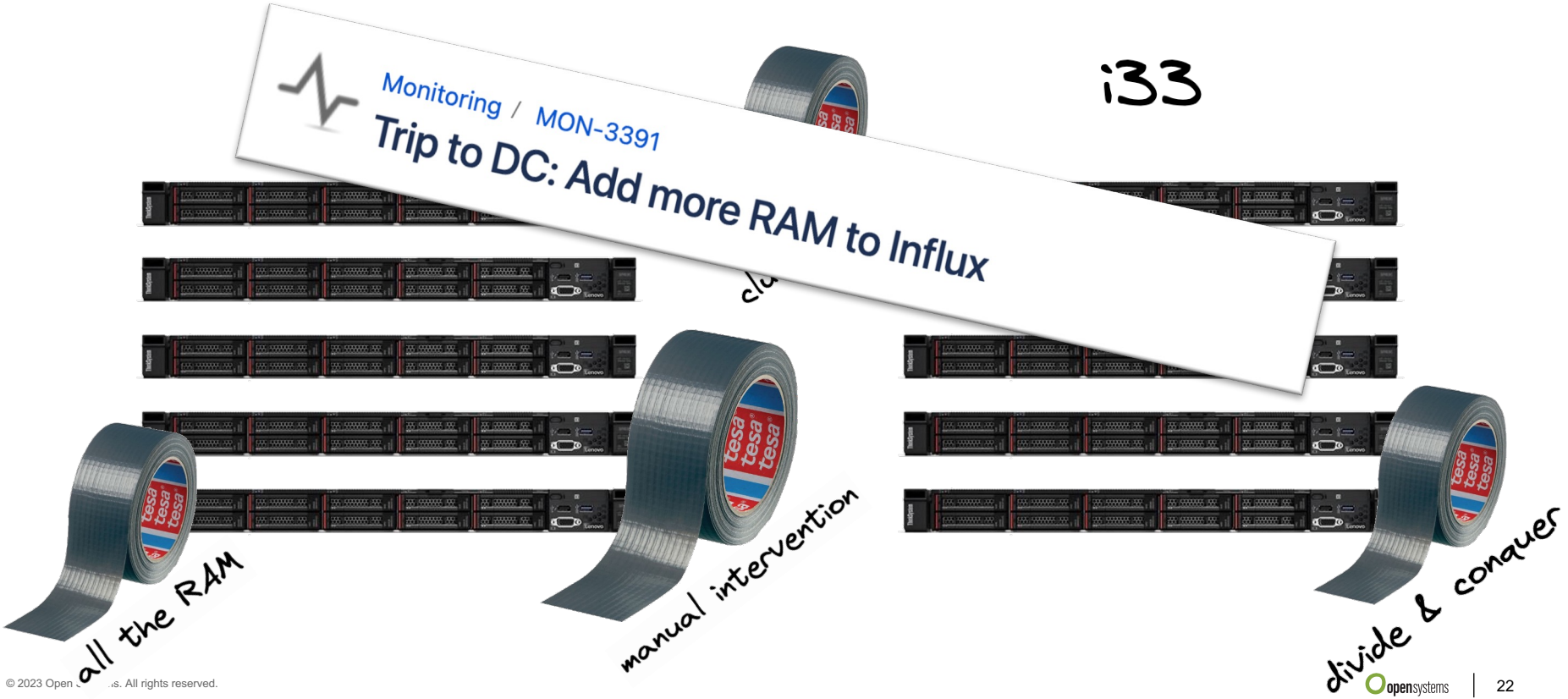
InfluxDB

a13

*actual number of hosts deployed

i33

InfluxDB

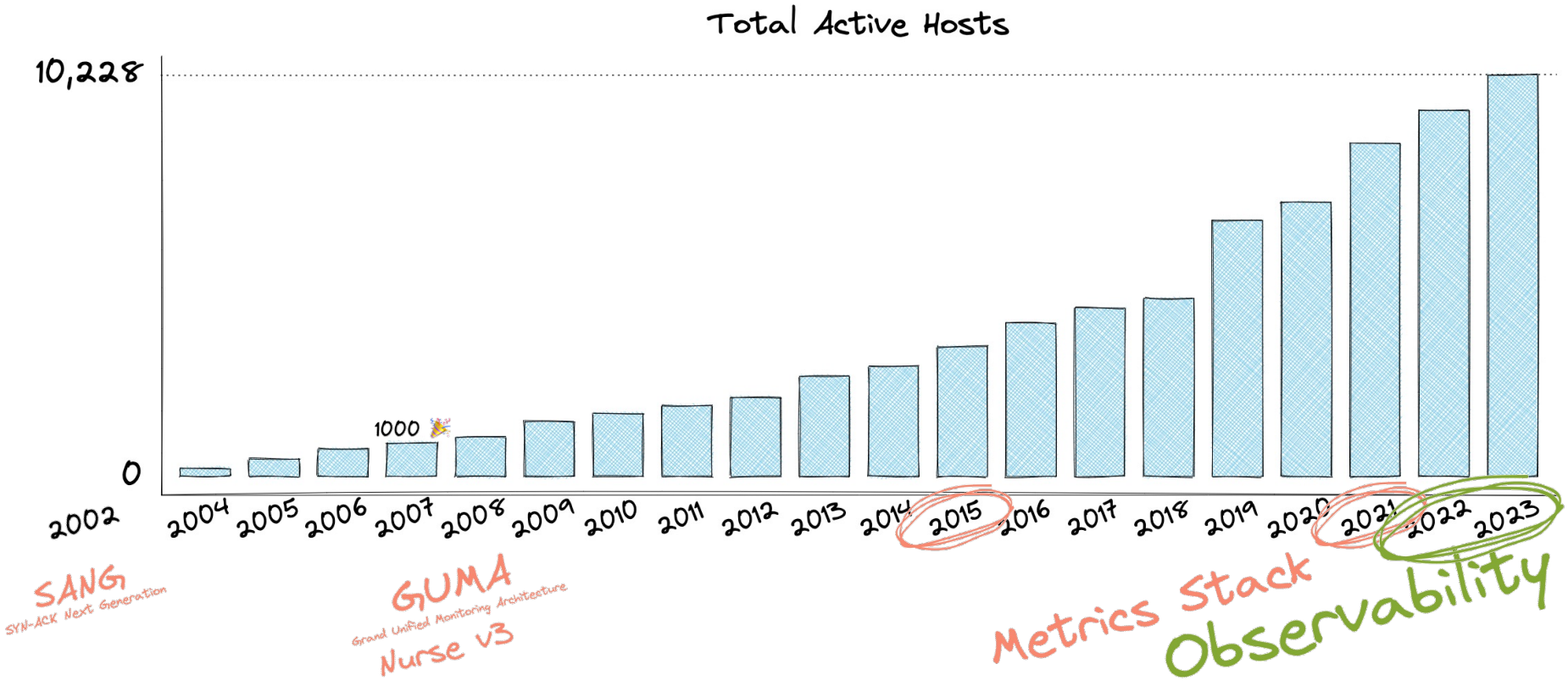


The biggest challenges we need to solve

New Environments

Log-centric Alerting

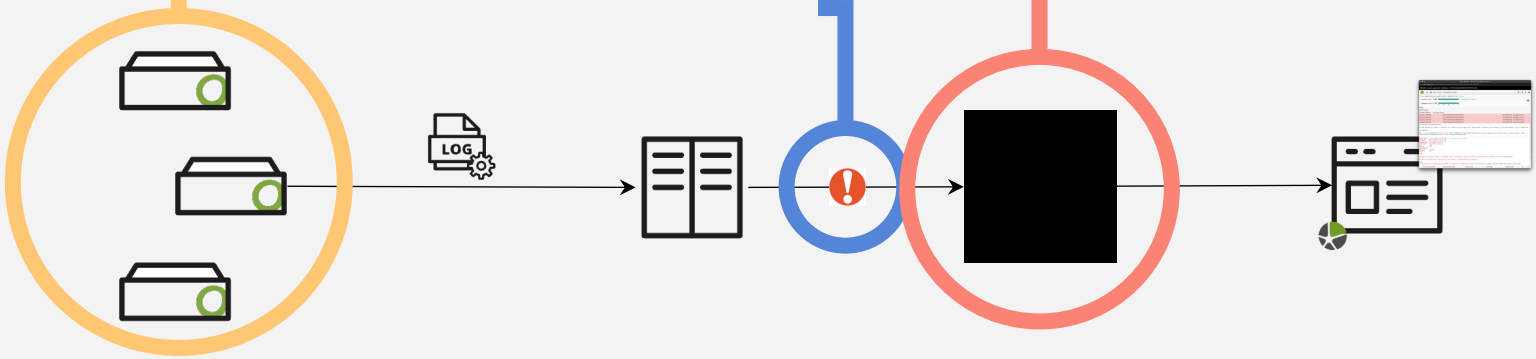
Scalability



Environment agnostic Observability Platform

Alerting on Metrics

Smarter Event Handling



Customer Fleet

GUMA Server

Implication Engine

Mission Control

Our Telemetry Stacks at a Glance

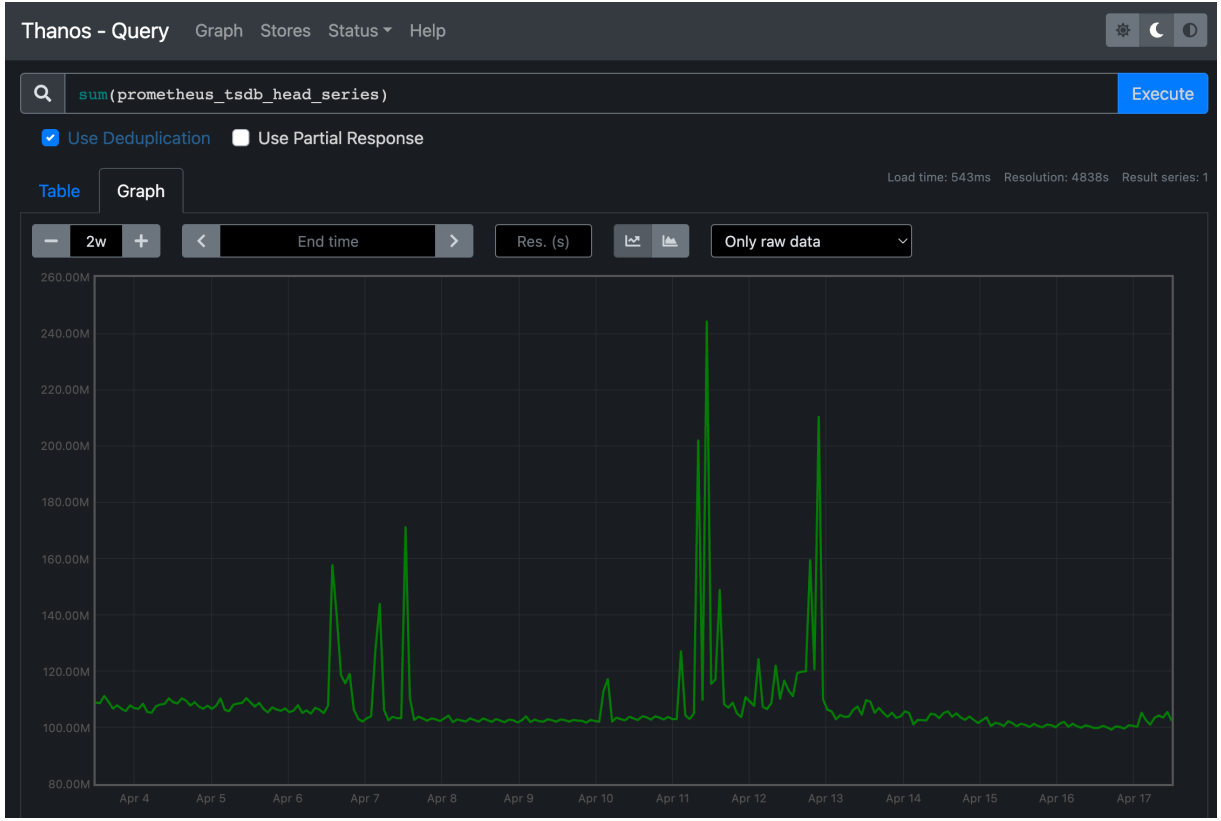
Metrics

- 50 cores
- 800 GiB RAM
- ~370 pods
- 110 million metrics

Logs

- 55 cores
- 700 GiB RAM
- ~400 pods
- 3 TiB / day

Thanos

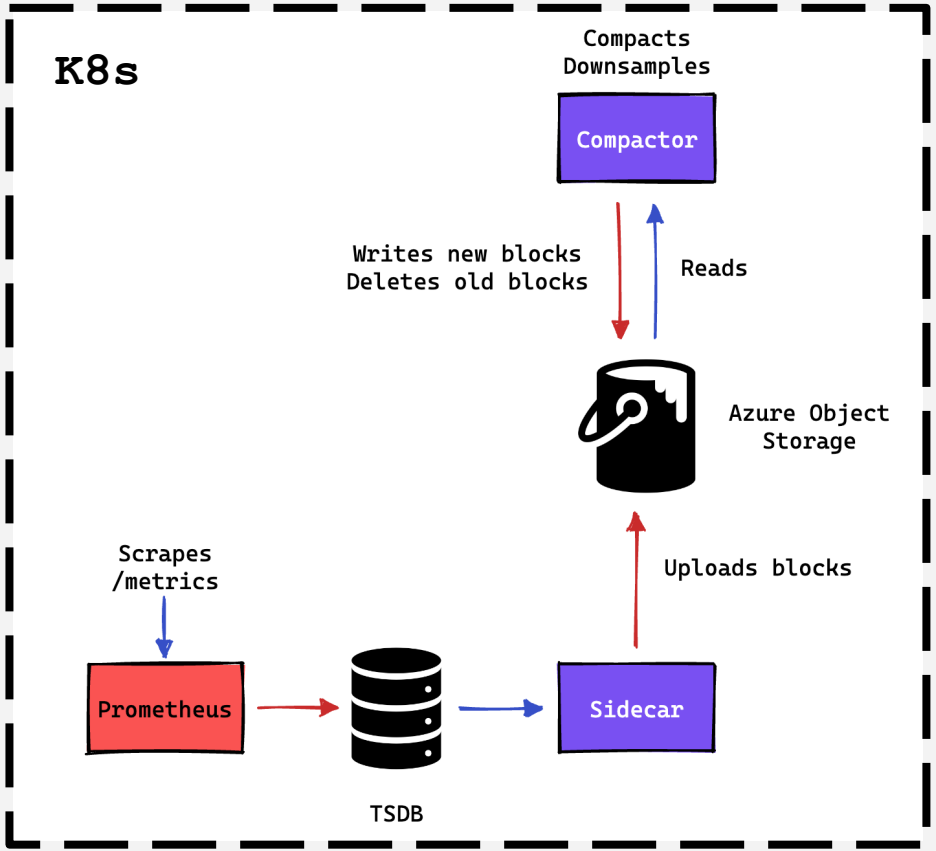


~110 million metrics, with bursts as high as 240m metrics

Thanos on the cluster

Thanos Classic architecture

***Thanos Sidecar*: plugs Prometheus into the StoreAPI, periodically flushes blocks to object storage**

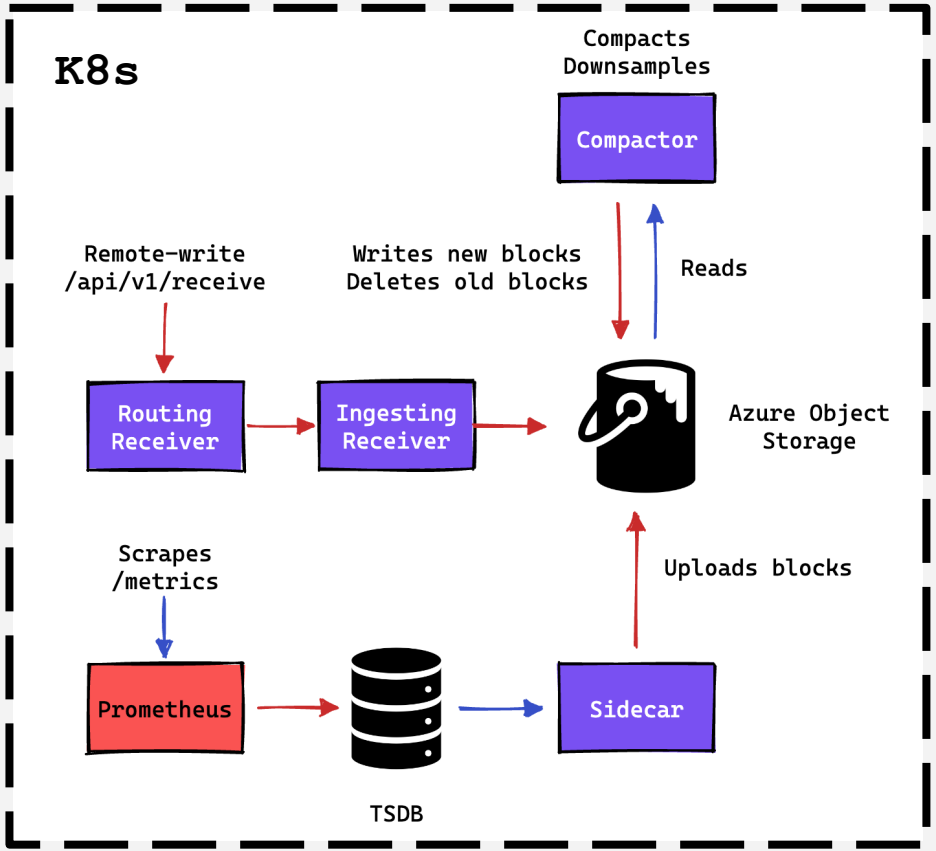


Thanos on the cluster

Remote-write Architecture

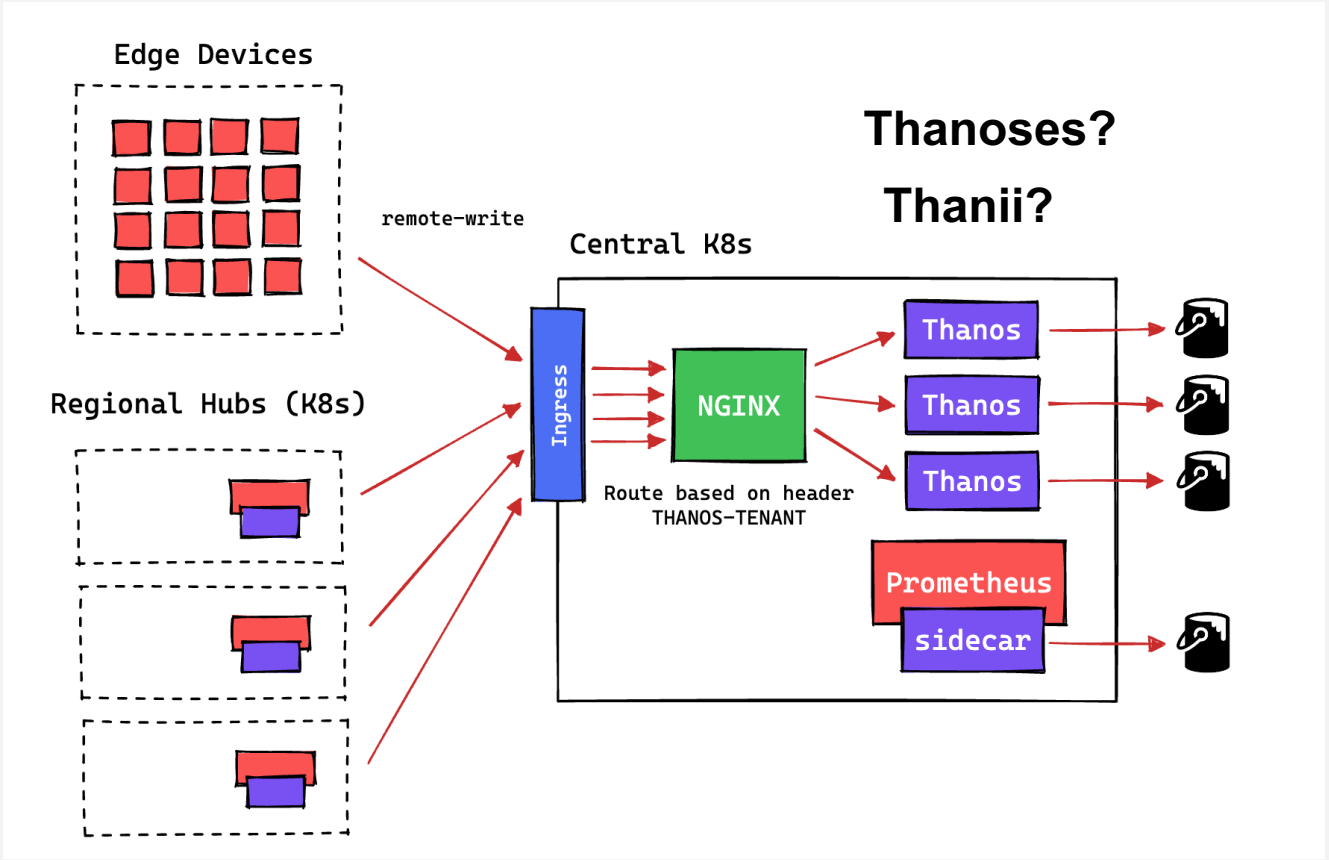
***Routing receivers*: process remote-write requests**

***Ingesting receivers*: write local TSDB blocks with periodic flush to object store**



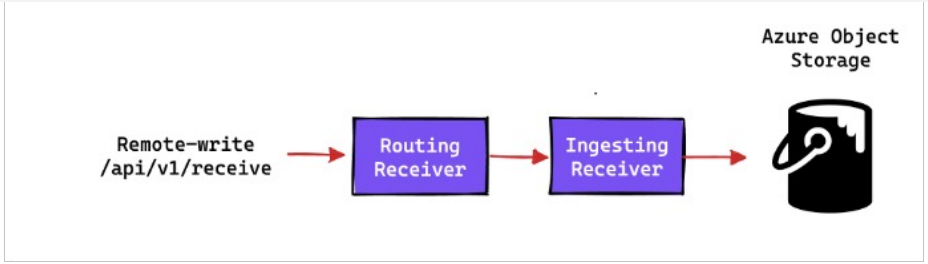
Thanos: Write-path

- **Everything written centrally (compliance)**
- ***Client certificate* authentication for external requests**
- **NGINX gateway maps Thanos tenants to the correct ingestion pipeline**



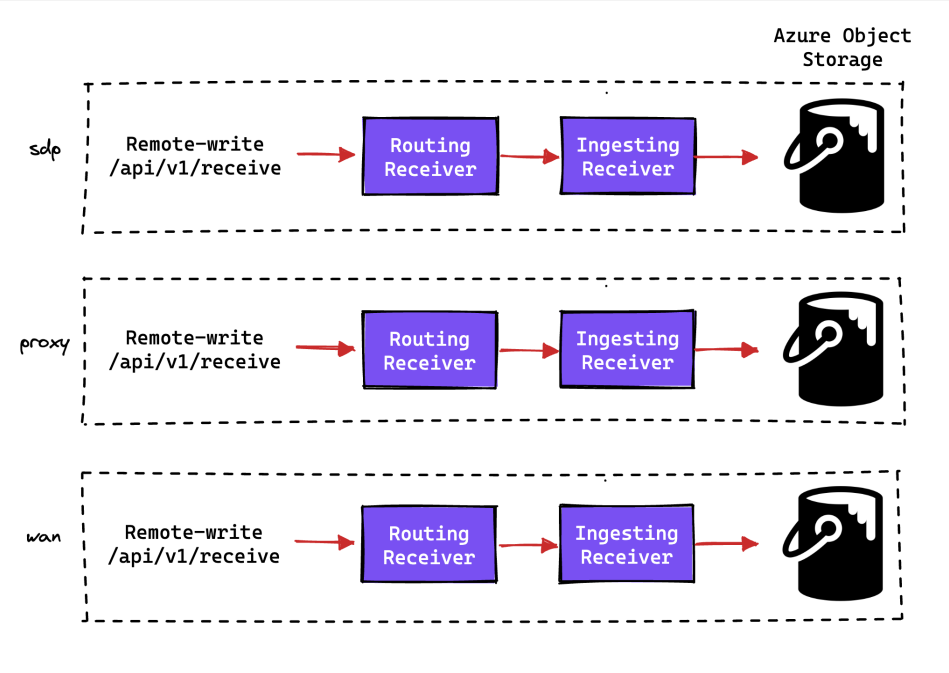
Thanos: Tenancy

What do we mean by a Tenant?



Thanos: Tenancy

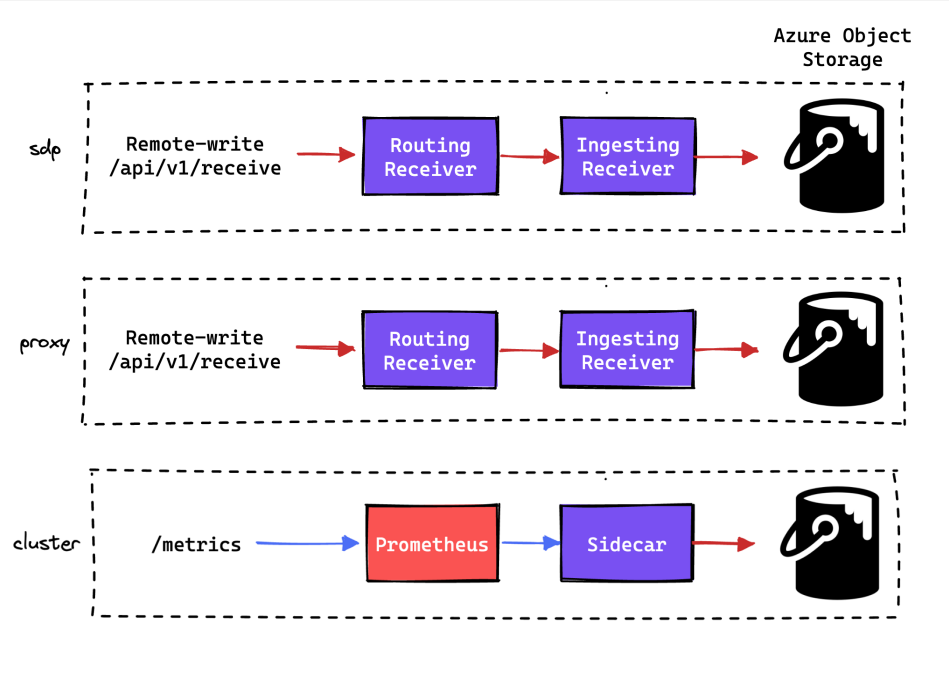
What do we mean by a Tenant?



- **Hard tenancy => one dedicated pipeline**
- **Each tenant maps to one service team**
- **Why? Mitigate the effects of unexpected *cardinality explosion***

Thanos: Tenancy

Concept extends to account for on-cluster metrics

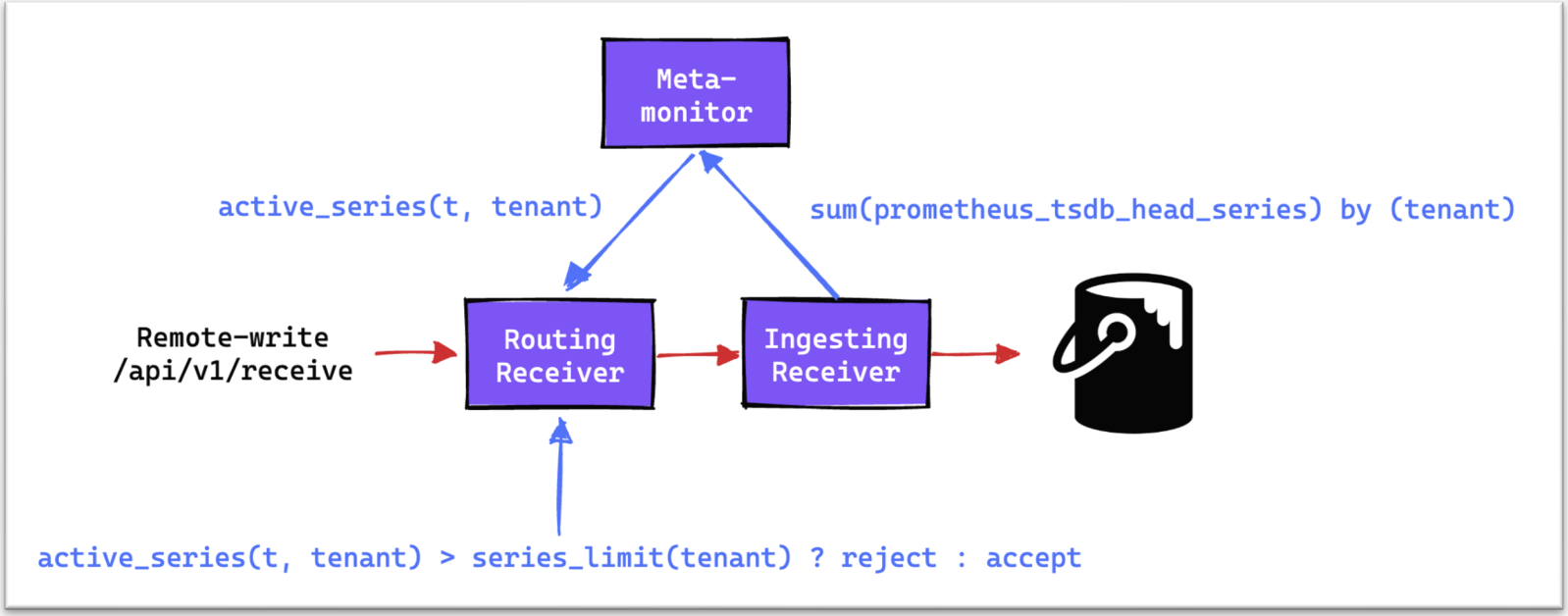


- **Cluster metrics belong to their own tenant with a dedicated storage bucket**

Cardinality Explosion: mitigation

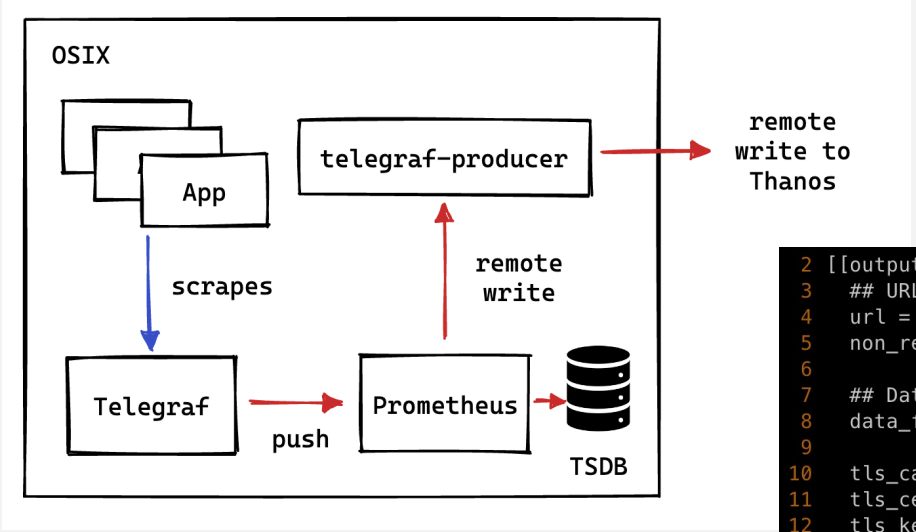
Active series limiting (available since v0.29.0)

Additional protection against 🔥 in the ingesting receivers



Collecting metrics at the Edge

Metrics mapped to tenants on edge device



Prometheus Config

```
16      metric_relabel_configs:
17      - source_labels: ['__name__']
18        regex: '(?:tunnel|tunnel_interface|tmon2).*'
19        replacement: 'wan'
20        target_label: 'service_component'
```

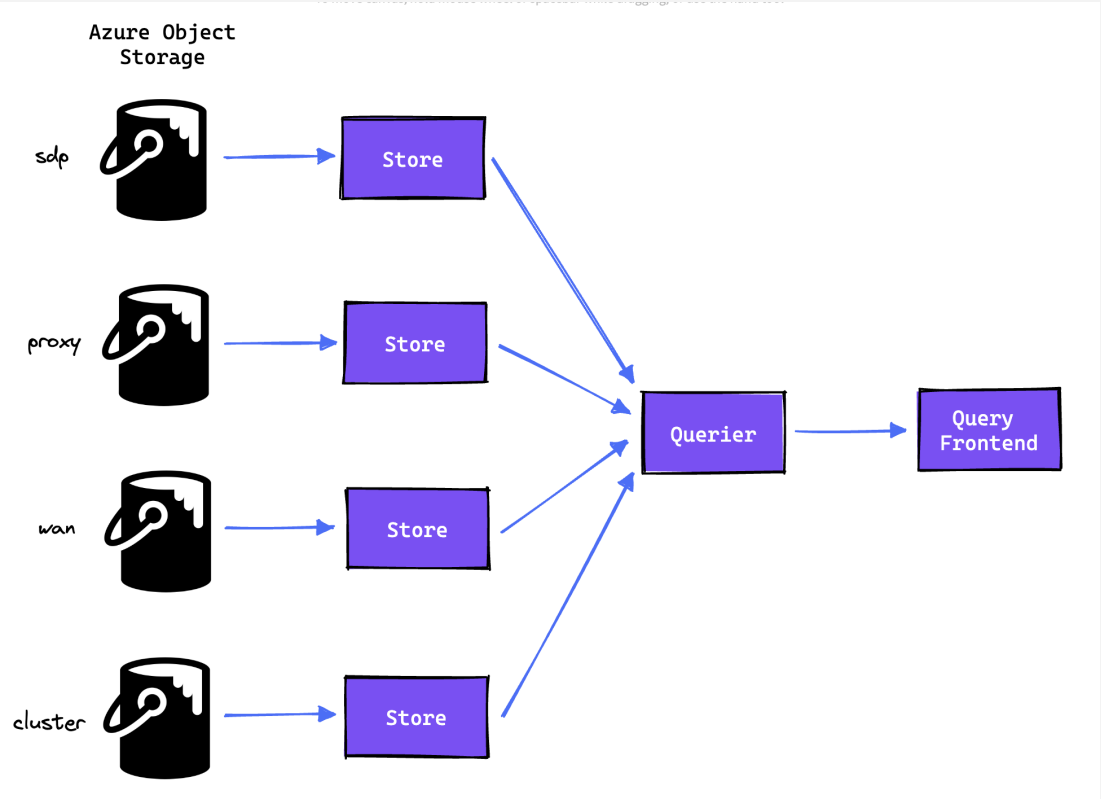
Telegraf Remote-write Config

```
2  [[outputs.http]]
3    ## URL is the address to send metrics to
4    url = "https://thanos.test.osdp.open.ch/api/v1/receive"
5    non_retryable_statuscodes = [409]
6
7    ## Data format to output.
8    data_format = "prometheusremotewrite"
9
10   tls_ca = "/opt/OSAGtelegraf/etc/telegraf-kafka-producer.d/ca_server.pem"
11   tls_cert = "/opt/OSAGtelegraf/etc/telegraf-kafka-producer.d/cert.pem"
12   tls_key = "/opt/OSAGtelegraf/etc/telegraf-kafka-producer.d/key.pem"
13
14   [outputs.http.headers]
15     Content-Type = "application/x-protobuf"
16     Content-Encoding = "snappy"
17     X-Prometheus-Remote-Write-Version = "0.1.0"
18     THANOS-TENANT = "proxy"
19
20   [outputs.http.tagpass]
21     service_component = ["proxy"]
```

Match metrics with this label

Thanos: Read-path

Goal: global view of metrics, regardless of the source

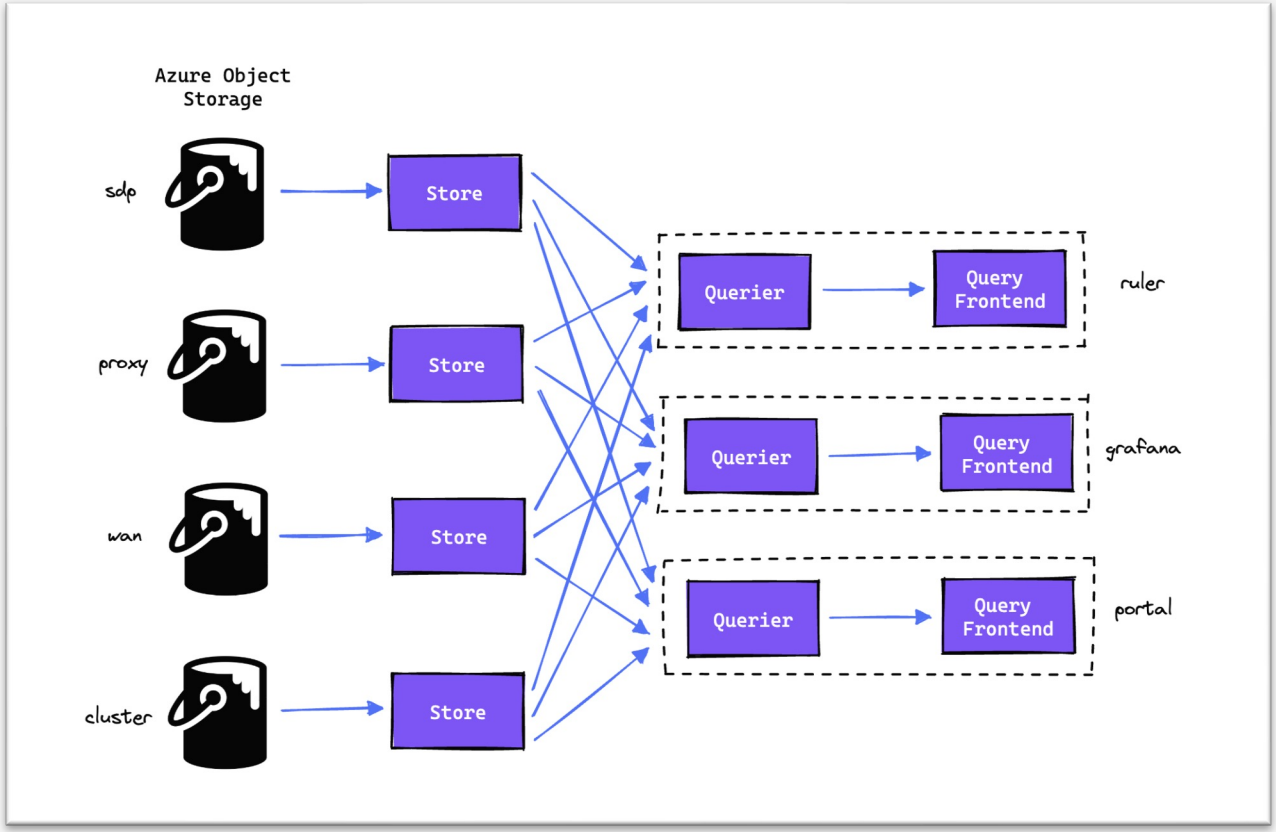


- **Store component interface to fetch data from object storage**
- **Naïve approach: plug everything into the same querier!**
- **We have many different users, with different priorities**
- **Can we prioritize query traffic?**

Thanos: Read-path

Querier quality of service

- **In this architecture, all query users have the same priority**
- **What if we want to prioritize portal queries?**

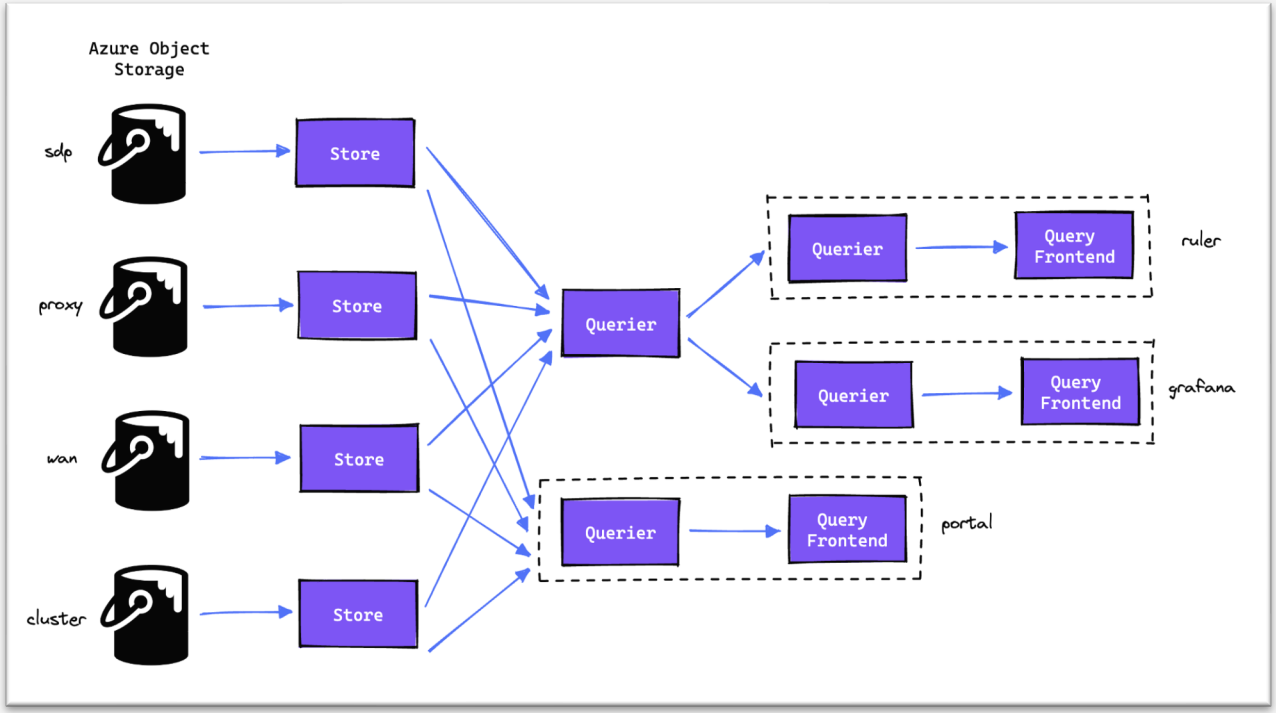


[1] https://grafana.com/docs/loki/next/lids/0003-queryfairnessinscheduler

Thanos: Read-path

Querier quality of service

- **Queriers at the same *level* have equal priorities**
- **By shifting the `grafana` and `ruler` users up a level, `portal` queries now have higher priority [1]**



[1] https://grafana.com/docs/loki/next/lids/0003-queryfairnessinscheduler

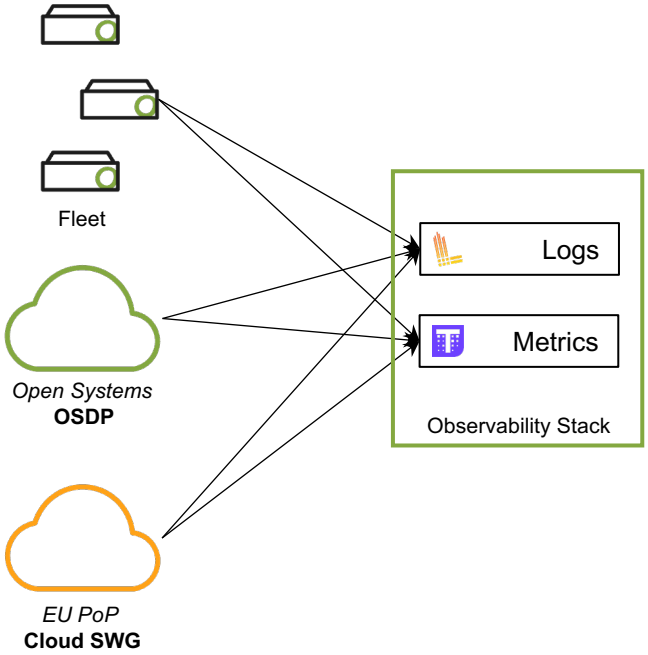
Reflect

What has the new metrics pipeline given us?

- Environment agnostic ingestion pipeline ✅
- Global metrics query view ✅

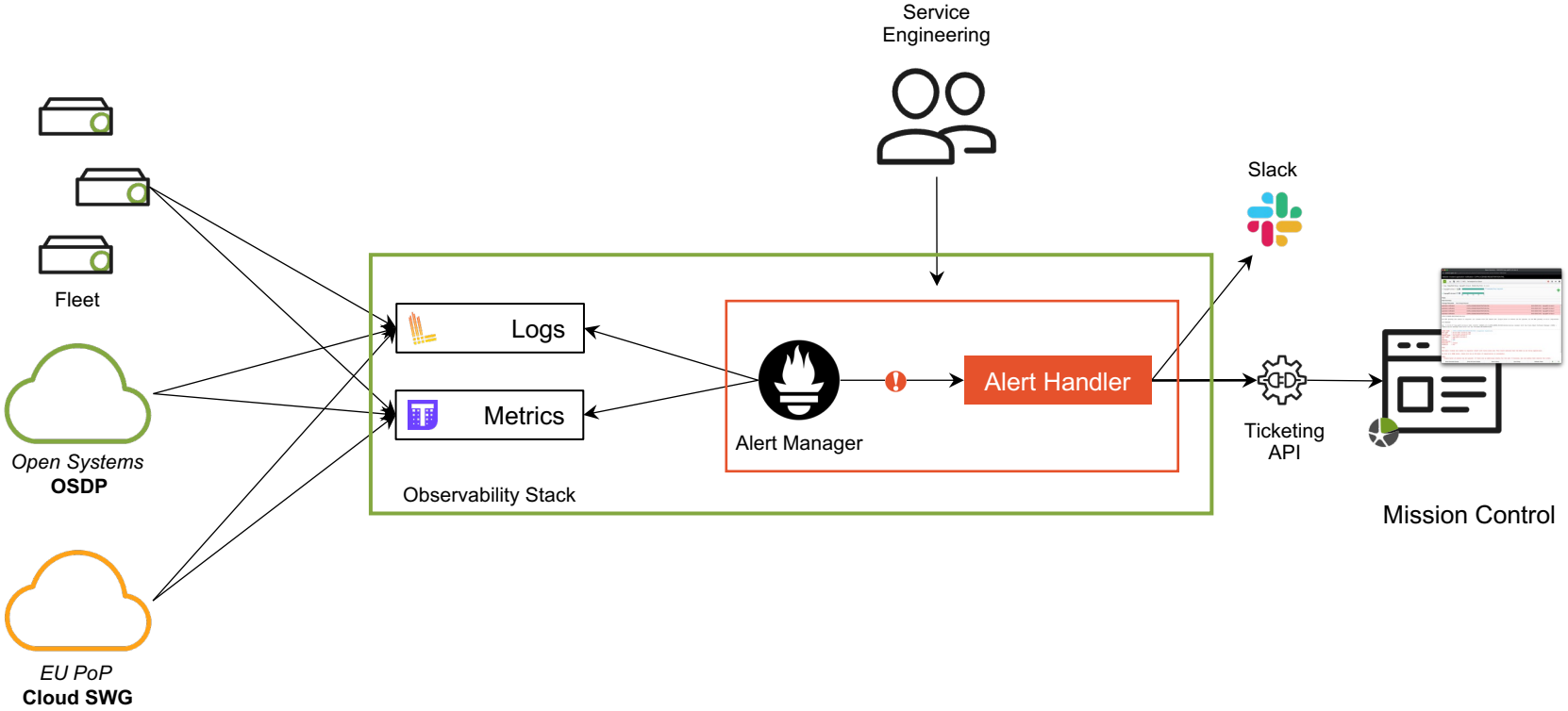
- How can we put this to use to achieve our other goals?
  - Smarter alerts (metrics, not logs)
  - Smarter alert *handling*

How do we get the relevant alerts into Mission Control?



Implication Engine?

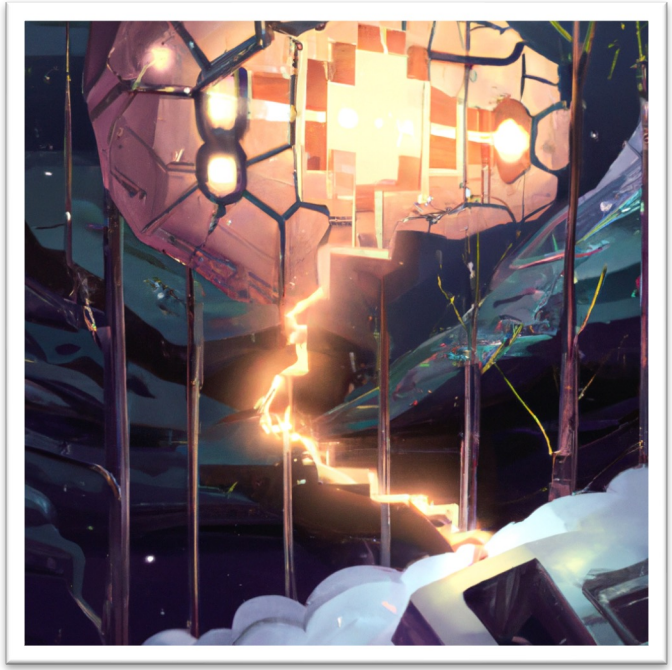
The unified alerting pipeline



The unified alerting pipeline

The Alert Handler

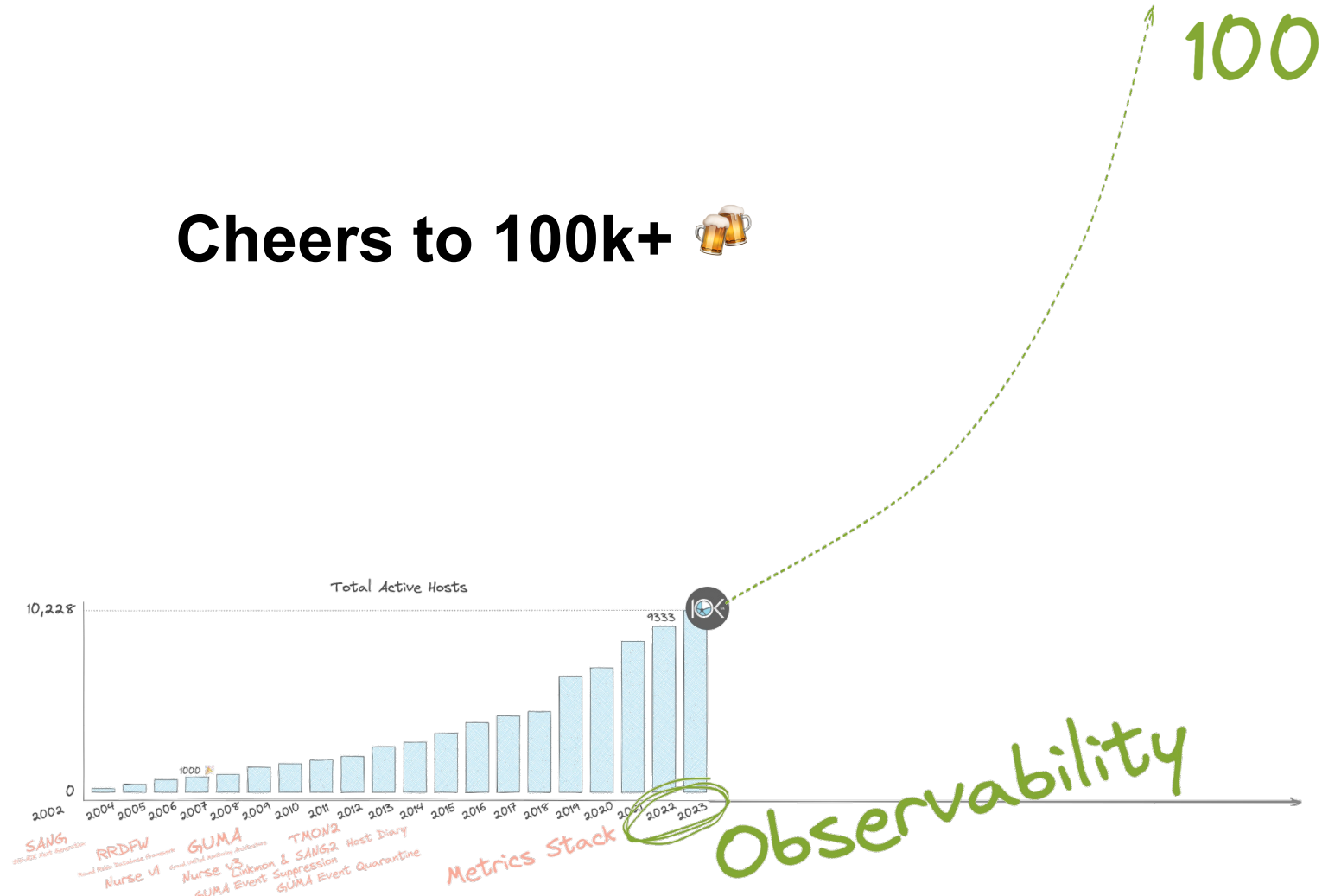
- **Replacement for the black box “implication engine”**
- **Goal is to group alerts, correlate related alerts, and enrich alerts with additional data before creating MC tickets**
- **Designed as a K8s operator that service teams can configure independently**



Conclusion & Outlook

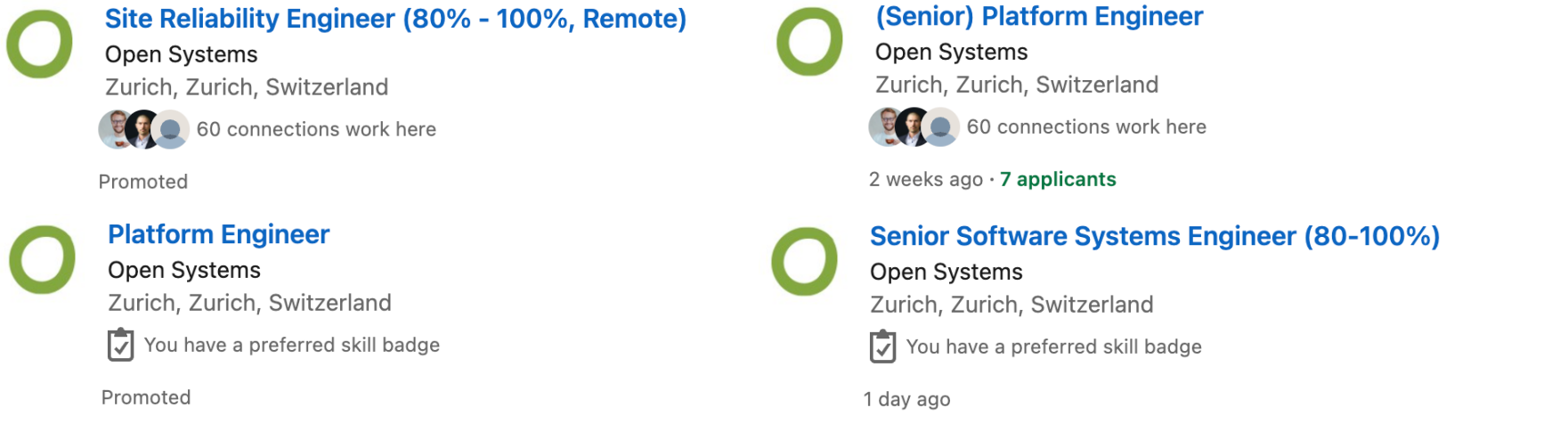
- **The scalability dragon is at the gates, but we are working hard on implementing our Action Plan**
- **Core telemetry (logs/metrics) backends are in production and ready to roll**
- **We now go full steam ahead with making use of these data effectively, to provide real value**

Cheers to 100k+ 🍻



We are hiring!

Interested? Come say hi during the conference 🙂



More information and job specs on LinkedIn

Thanks for your attention!

Questions?

https://verejoel.com

https://github.com/verejoel