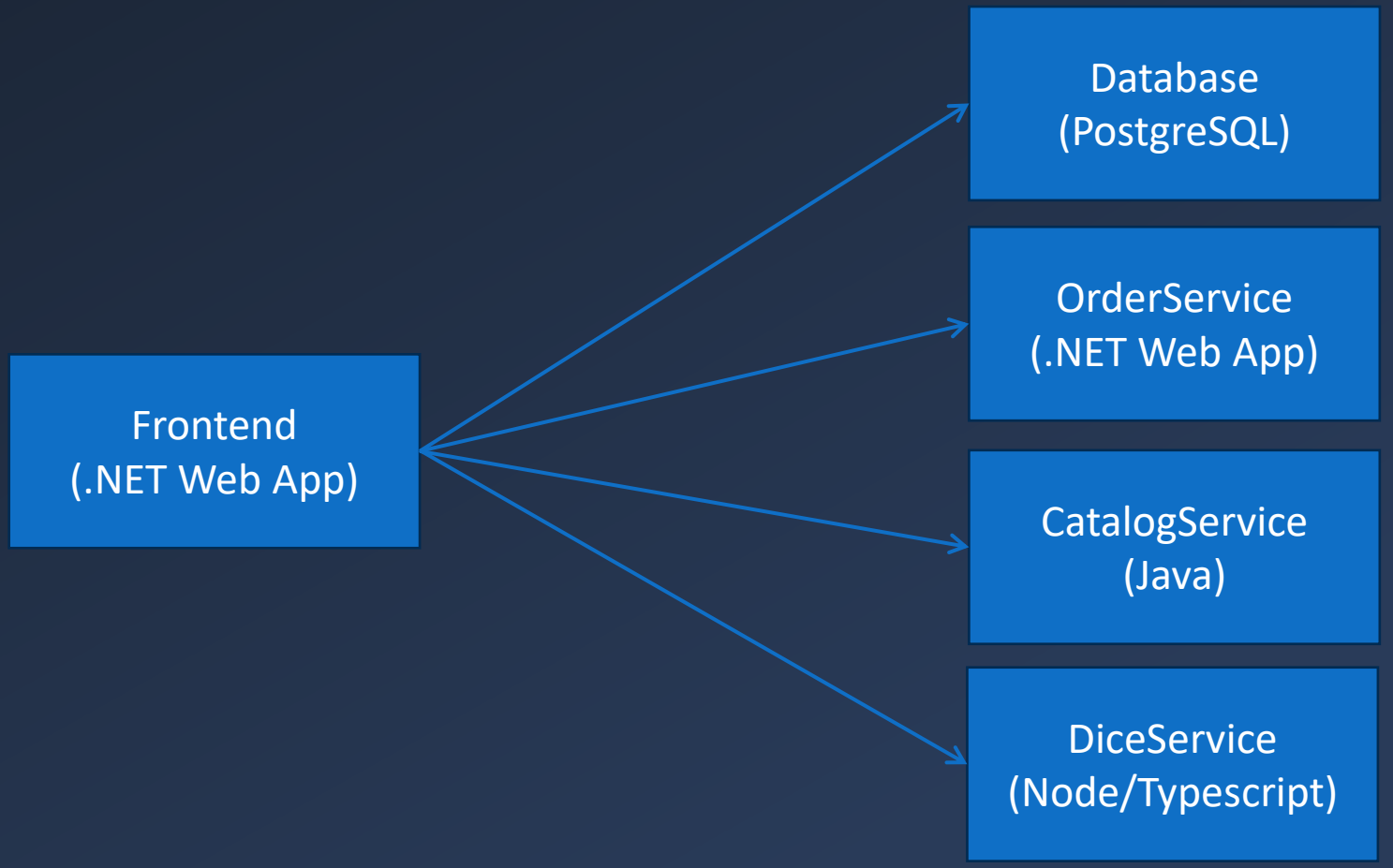
Observability Guide: How to Keep Your Applications Under Control?

Tomáš Jecha

Head of Engineering at cbData

LinkedIn /in/jechtom | X @jechtom

Demo App Architecture



DEMO

Demo App Introduction

What will you do if someone reports an issue in your app?

What value does good observability provide?

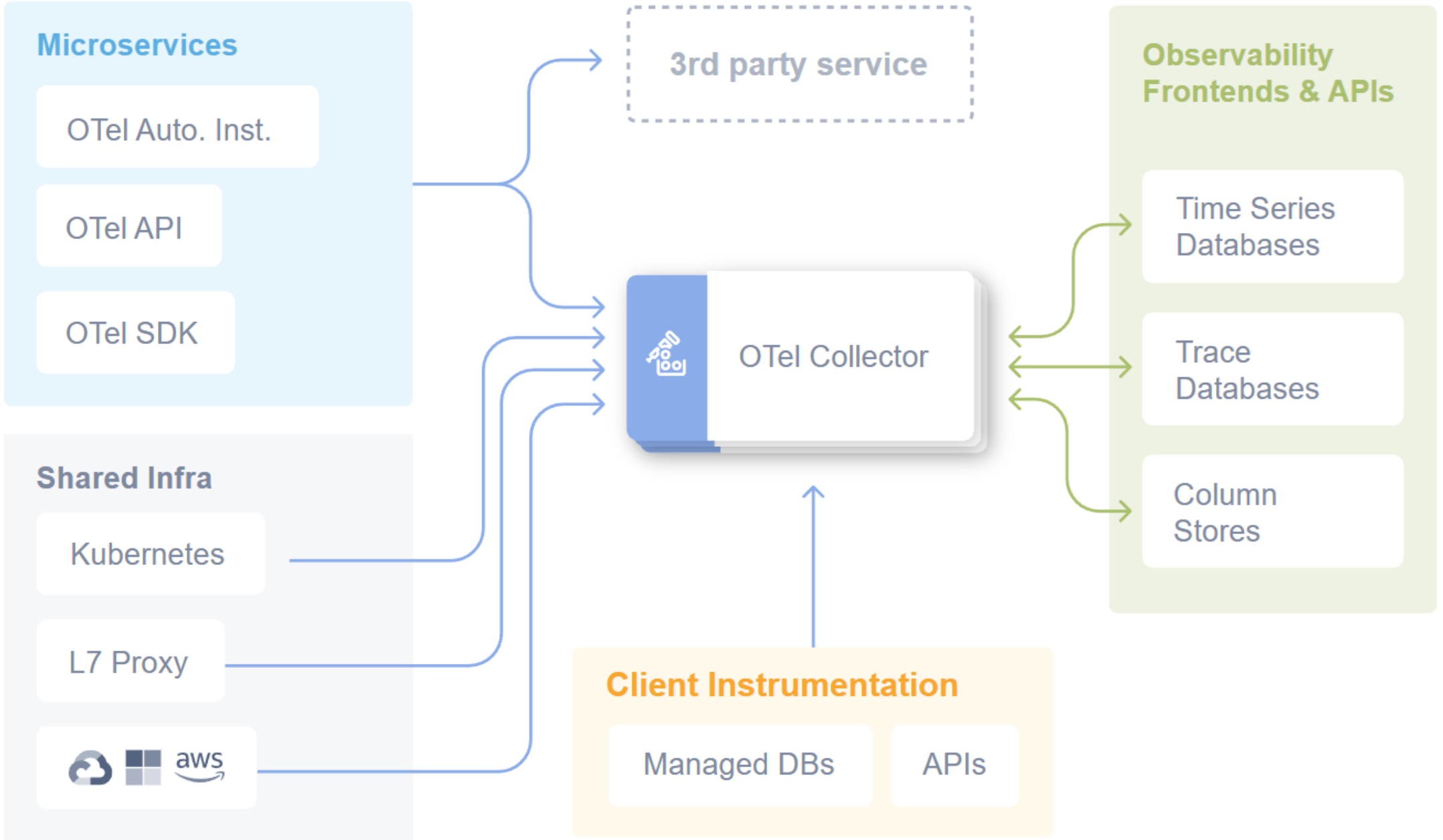
- Easier maintenance of large systems
- Faster onboarding of new team members
- Proactive monitoring and alerting
- Clear assessment and analysis of incidents and their impact
- Measuring performance trends as the system evolves
- Early detection of issues before they become incidents

What’s Wrong with Observability?

- Vendor lock-in
  - Proprietary tools and instrumentation libraries
  - Switching is expensive and painful
  - Every tool requires custom SDKs and configuration
- Tools fragmentation
  - Different tools, formats, protocols → no standardization
- Lack of clear best practices
  - No clear guidelines → Every team does observability differently
  - Inconsistent telemetry data structure

What is OpenTelemetry?

- Semantic rules – naming (log severity, exporters, meters, etc.)
- Protocol – serialization and transport
- APIs & SDKs – C++, .NET, Go, Java, PHP, Python, Rust, Swift, JS, …
- OpenTelemetry Collector – receive, process and export telemetry
- Industry support – https://opentelemetry.io/ecosystem/vendors/


Microservices
OTel Auto. Inst.
OTel API
OTel SDK
Shared Infra
Kubernetes
L7 Proxy
aws
3rd party service
OTel Collector
Client Instrumentation
Managed DBs
APIs
Observability Frontends & APIs
Time Series Databases
Trace Databases
Column Stores

OpenTelemetry Protocol (OTLP)

- OTLP/gRPC (Protobuf) or OTLP/HTTP (Protobuf or JSON)
- Protocol specs and protobuf definitions at https://github.com/open-telemetry/opentelemetry-proto
- Defines services:
  - Logs collector
  - Metrics collector
  - Trace collector

OpenTelemetry Protocol (OTLP) Design Goals

- All signal types over single protocol
- For instrumented apps, telemetry backends and proxies
- Reliable, low CPU and memory usage
- High throughput, backpressure signaling
- Load-balancer friendly

.NET Aspire Dashboard

- Receives and visualizes OTLP telemetry
- Part of the .NET Aspire project
- Has a standalone mode

DEMO

Telemetry Backends

OpenTelemetry Signals

- Logs
- Tracing
- Metrics

Observability Signals – Logs

```
03 Mar 2024 17:21:29.094 OTelDemo.Web Privacy page visited
03 Mar 2024 17:21:28.550 OTelDemo.Web Executed DbCommand (1ms) [Parameters=[], CommandType='Text', CommandTimeout='30'] SEL...
03 Mar 2024 17:21:28.550 OTelDemo.Web received-first-response
03 Mar 2024 17:21:28.548 OTelDemo.Web End processing HTTP request after 64.2179ms - 200
03 Mar 2024 17:21:28.548 OTelDemo.Web Received HTTP response headers after 64.0337ms - 200
03 Mar 2024 17:21:28.537 OtelDemo.Backend Got weather forecast
03 Mar 2024 17:21:28.484 OTelDemo.Web Sending HTTP request GET http://localhost:4006/WeatherForecast
03 Mar 2024 17:21:28.484 OTelDemo.Web Start processing HTTP request GET http://localhost:4006/WeatherForecast
03 Mar 2024 17:21:28.484 OTelDemo.Web Done
03 Mar 2024 17:21:28.477 OtelDemo.Backend Getting weather forecast
03 Mar 2024 17:21:28.437 OTelDemo.Web Part way there
03 Mar 2024 17:21:28.381 OTelDemo.Web Index page visited
```

OpenTelemetry Log Record

- Timespan → When?
- Resource attributes → Who?
- Tracing → Trace Id, Span Id
- Severity → Trace, Debug, Info, Warn, Error, Fatal
- Structured Content → `"Order 32 has been Delivered"`

```
Message = "Order {OrderId} has been {State}"
OrderId = "32"
State   = "Delivered"
```

- Additional Attributes → Scope, Client IP, Identity, …
- See *example-log.json*

Resource represents the entity producing telemetry.

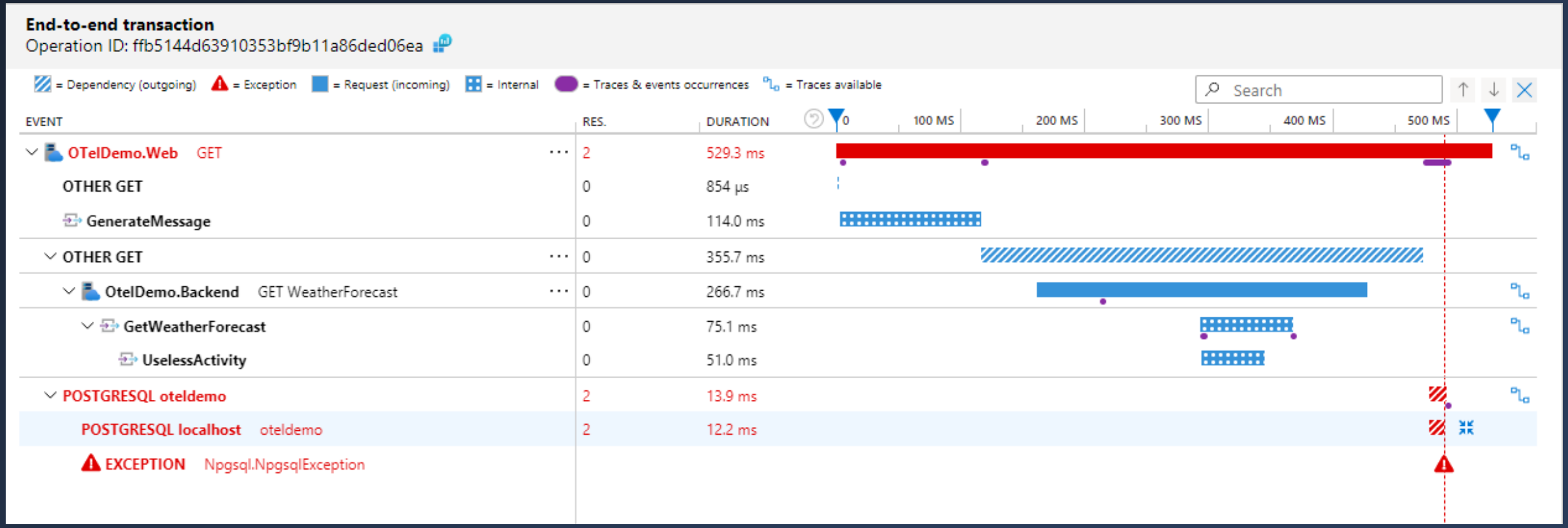
- Windows IIS AppPool
- Process inside Kubernetes pod
- Linux daemon
- …

- Logs
- Metrics
- Tracing

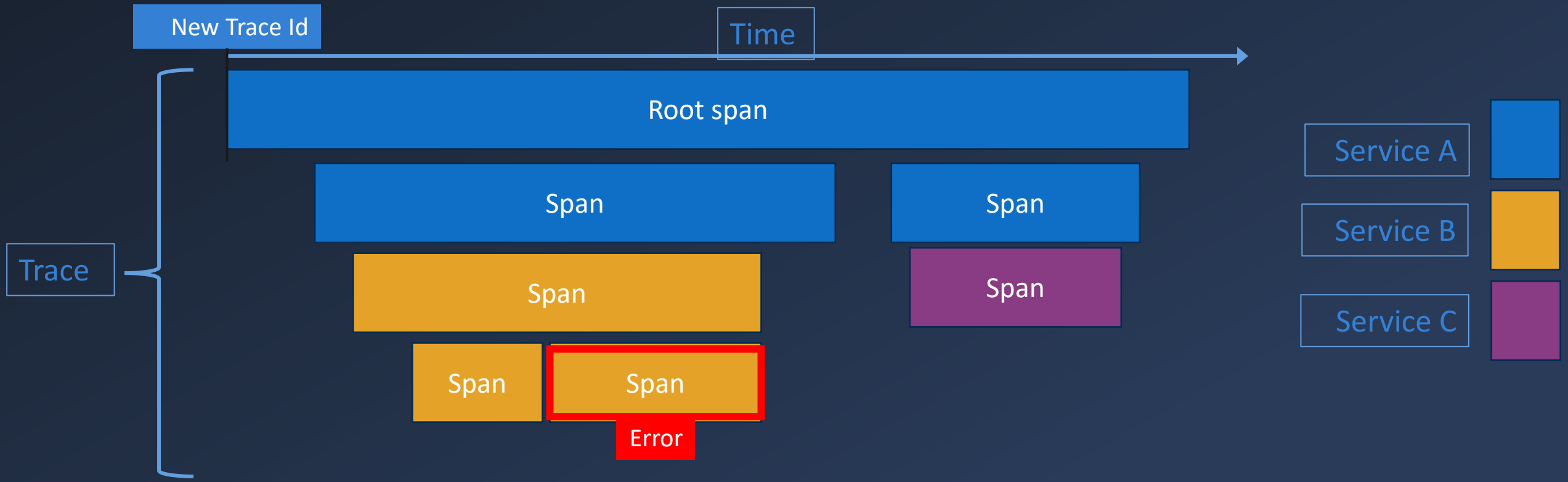
Resource Attributes – Examples

service.name=ShoppingCart
service.instance.id=627cc493-f310-47de-96bd-71410b7dec09
service.version=3.4.5; a01dbef8a
deployment.environment.name=staging
…
telemetry.sdk.language=dotnet
telemetry.sdk.name=opentelemetry
telemetry.sdk.version=1.2.3
…
process.pid=1234
process.executable.path=D:\apps\ShoppingCart\ShoppingCart.exe
os.type=windows
cloud.platform=azure_container_apps
k8s.pod.name=kubernetes-pod-name
…

Observability Signals – Tracing



Observability Signals – Tracing



Simplified span data

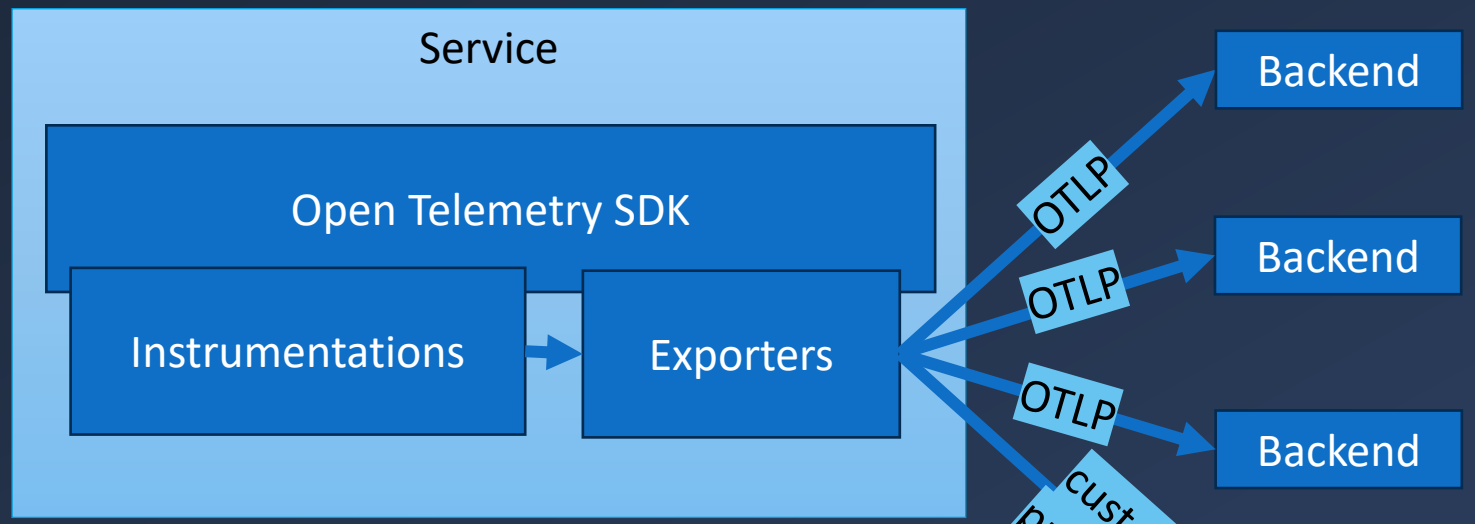

```
{
  "name": "hello",
  "context": {
    "trace_id": "5b8aa5a2d2c872e8321cf37308d69df2",
    "span_id": "051581bf3cb55c13"
  },
  "parent_id": null,
  "start_time": "2022-04-29T18:52:58.114201Z",
  "end_time": "2022-04-29T18:52:58.114687Z",
  "attributes": {
    "http.route": "some_route1"
  },
  "events": [
    {
      "name": "Guten Tag!",
      "timestamp": "2022-04-29T18:52:58.114561Z",
      "attributes": {
        "event_attributes": 1
      }
    }
  ]
}
```

How tracing works?

- W3C Trace Context HTTP headers
- Custom implementation for other protocols (message brokers, etc.)
- Language specific implementation

OpenTelemetry Libraries

- Registry: https://opentelemetry.io/ecosystem/registry
- Instrumentation libraries – generates relevant telemetry data
- Exporter libraries – sends telemetry (via OTLP or other protocols)



DEMO

Collecting Logs & Traces

Observability Signals – Metrics

- A measurement captured at runtime.
- A **Meter** creates **Instruments** that capture **Measurements**.

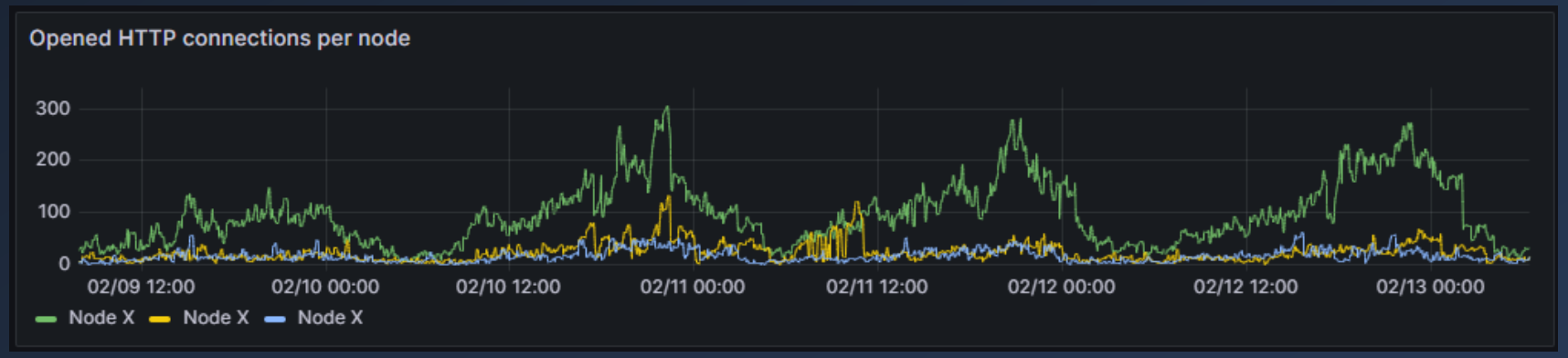
OpenTelemetry Instrument Types

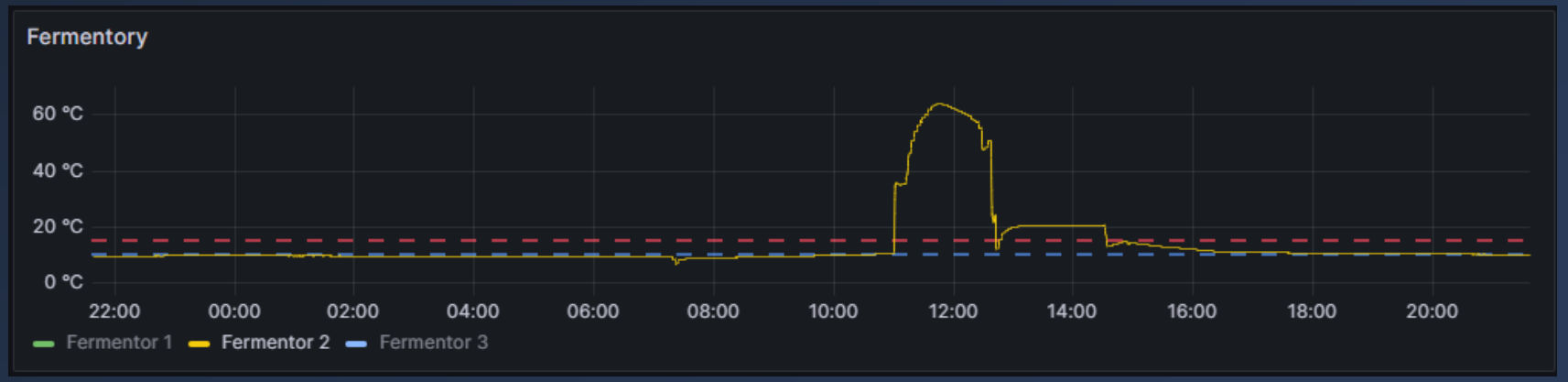
- Gauge
- Counter
- Histogram

Metric Types - Gauge

- Usage: Set instant value
- Examples:
  - CPU usage
  - Allocated threads
  - Open connections
  - Longest running task
  - Timespan of last backup
  - Free disk space
  - Queue length
  - ….

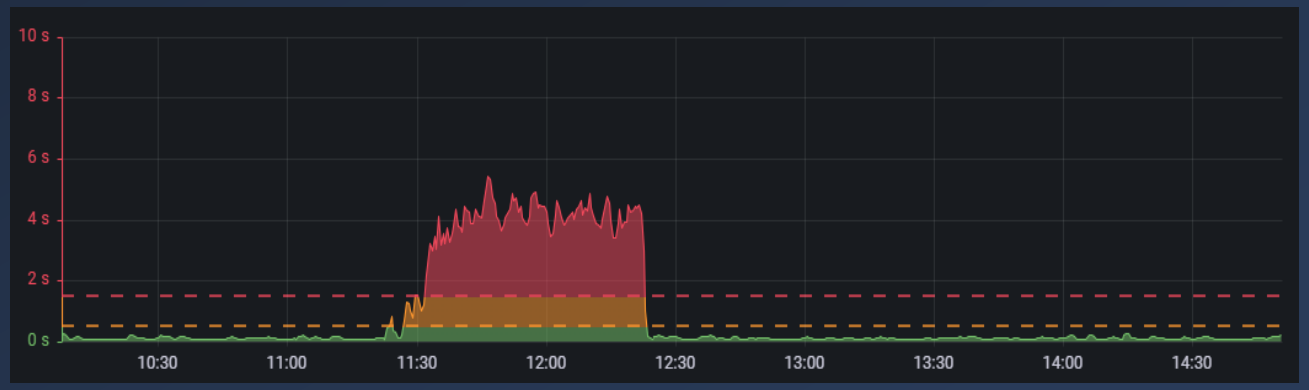
Metric Types - Gauge





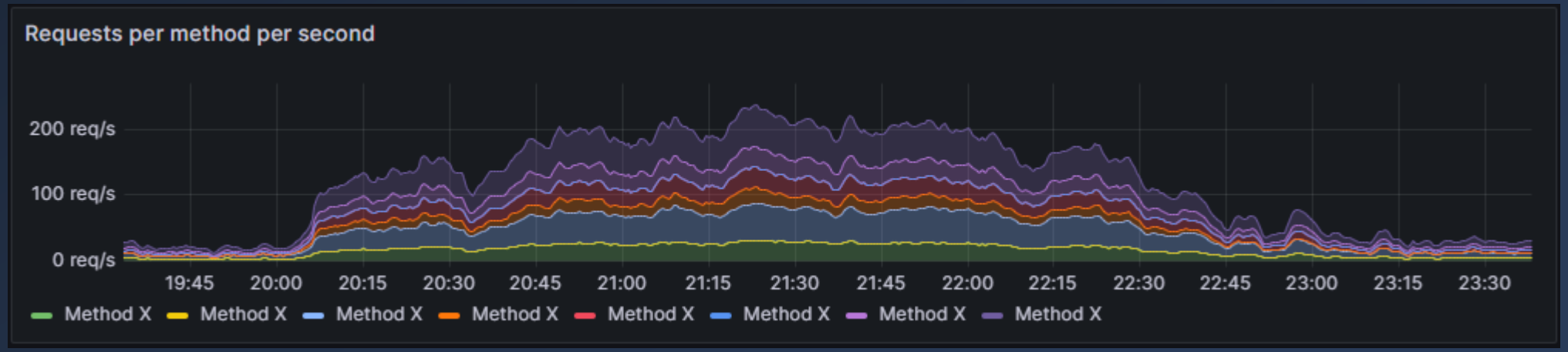
Metric Types – Counter (Sum)

- Usage: Increment +1
  - Examples: Counter of HTTP requests, executions, cache hit/miss, …
- Usage: Add +delta
  - Examples: Bytes transferred, rows processed, request duration, …
- Also: UpDown and Async Counters



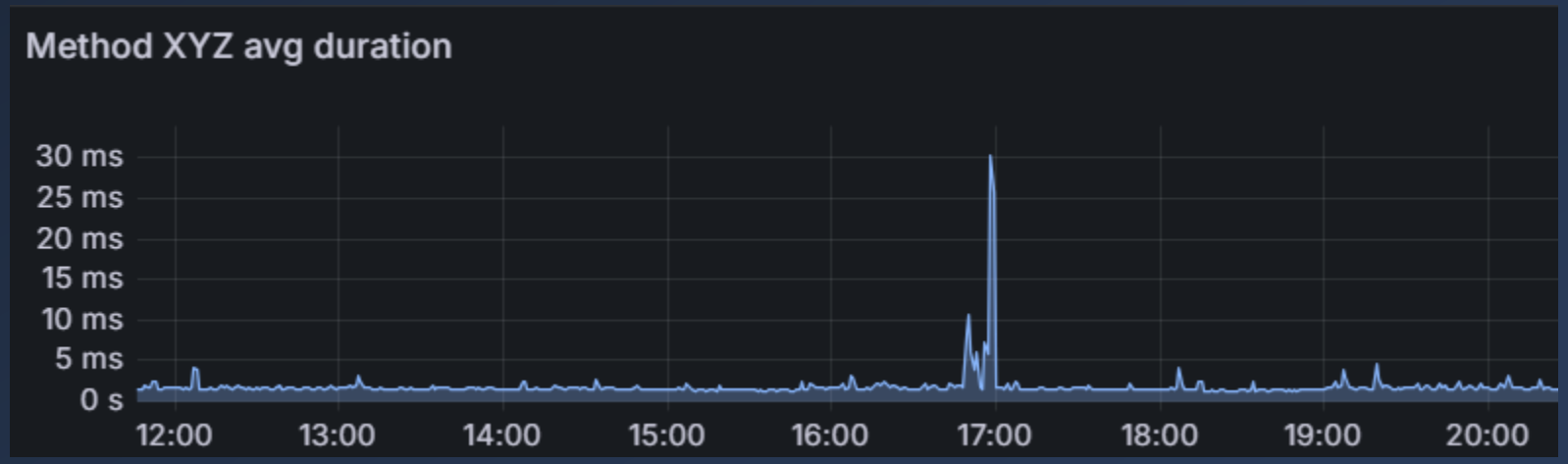
Metric Types – Counter (Sum)

- Visualize **rate**
- Examples: Requests/second, MB/s bandwidth, orders per hour, …



Metric Types – Counter (Sum)

- Visualize **average**
- Examples: average duration/size
  - From Requests counter / Requests total duration

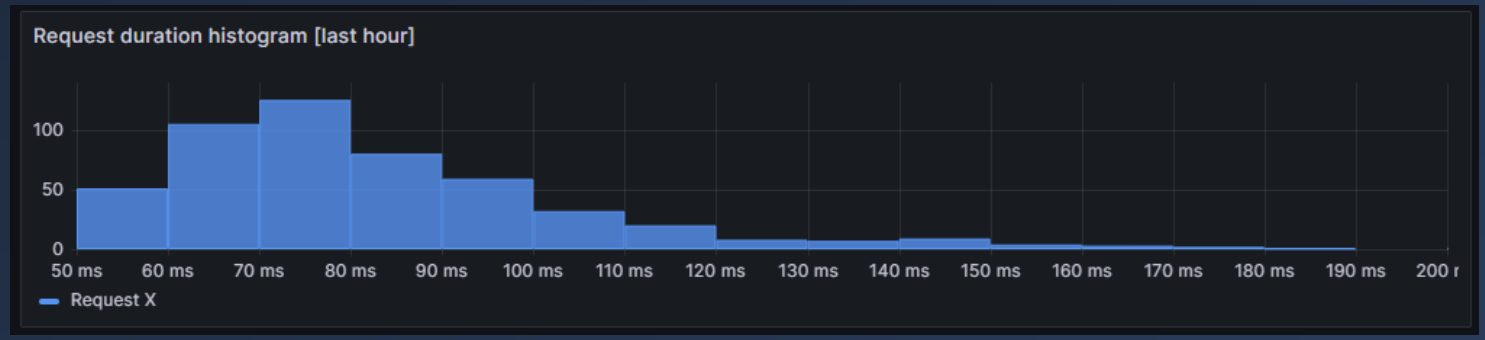


Metric Types – Counter (Sum)

- Visualize **ratios**
- Examples: cache hit/miss ratio, success/failure rate, …

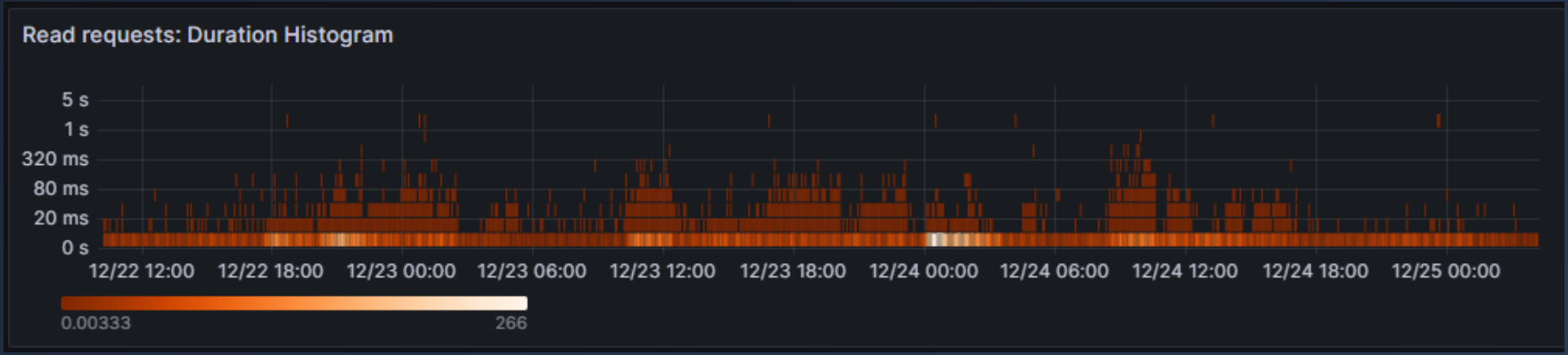
Metric Types – Histogram

- Usage: Record frequency of value (buckets)
- Examples: Request duration, message size, quantity per order
- Visualized as: Histogram, heatmap, percentile, average
- Default buckets for OpenTelemetry:
  [0, 5, 10, 25, 50, 75, 100, 250, 500, 750, 1000, 2500, 5000, 7500, 10000]



Metric Types – Histogram

- **Heatmap** visualization example (histogram over time)

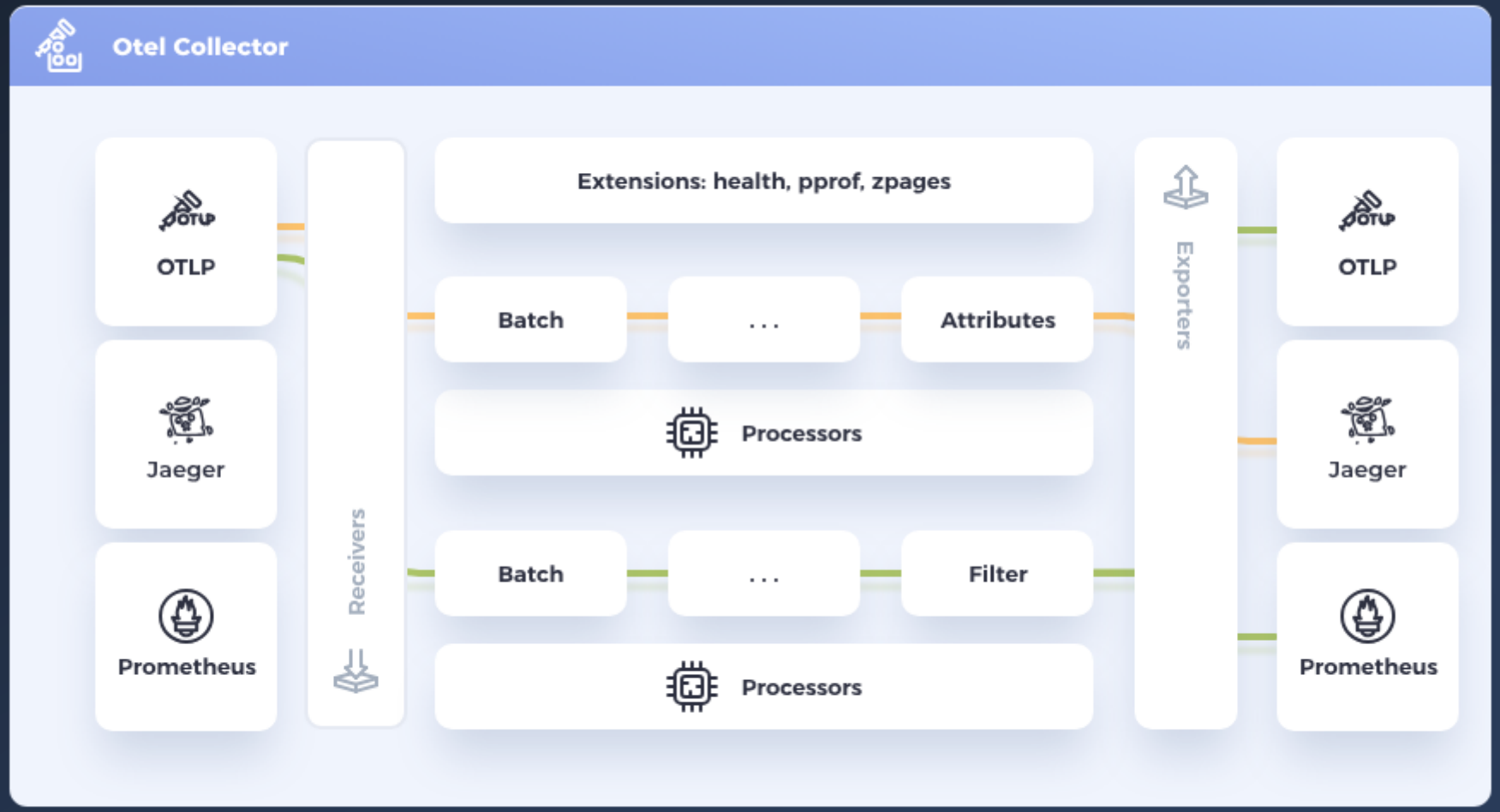


DEMO

Metrics

OpenTelemetry Collector

- Receive, process and export telemetry data
- https://opentelemetry.io/docs/collector/
- Similar tools:
  Alloy (metrics), Logstash (logs), Fluentd (logs), Telegraf (InfluxDB), …
- Registry:
  https://opentelemetry.io/ecosystem/registry/?language=collector


Otel Collector
OTLP
Jaeger
Prometheus
Receivers
Extensions: health, pprof, zpages
Batch
. . .
Attributes
Processors
Batch
. . .
Filter
Processors
Exporters
OTLP
Jaeger
Prometheus

DEMO

OpenTelemetry Collector

Zero-code Instrumentation

- Tools to inject observability to apps without having to edit the source.
- https://opentelemetry.io/docs/zero-code/
- Works (-isch) with: Go, .NET, PHP, Python, Java, JavaScript
- *Magic** 🦄 🌈

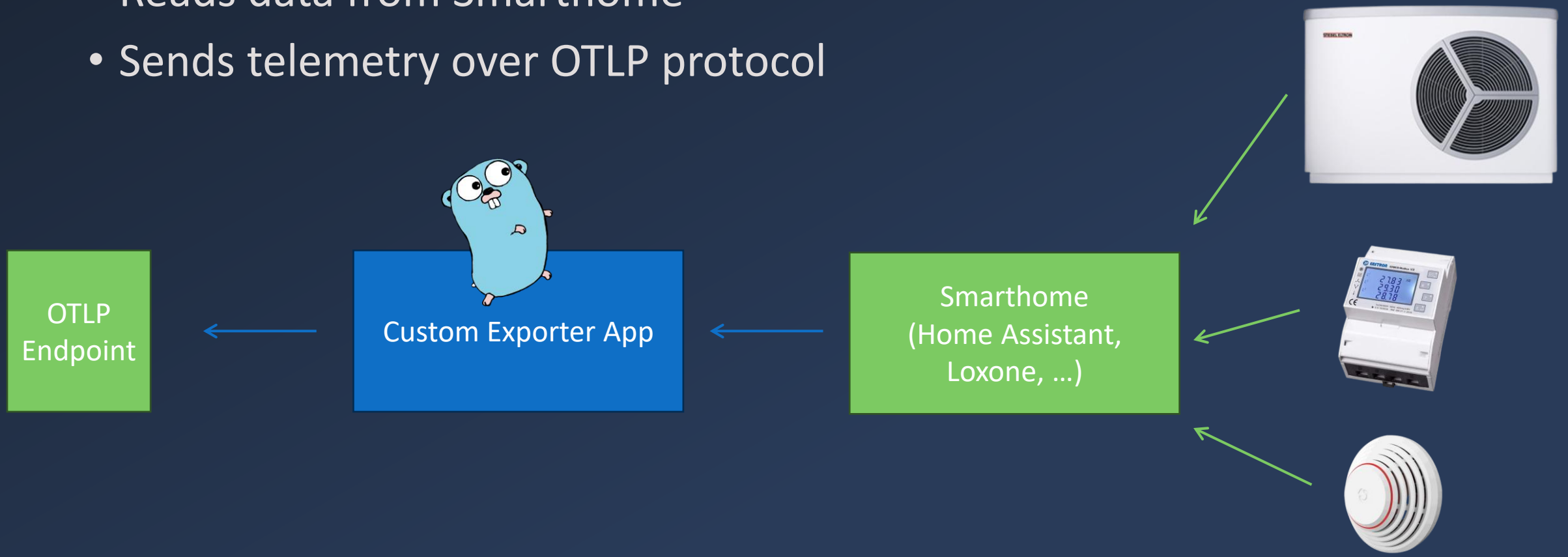
* *magic* is limited

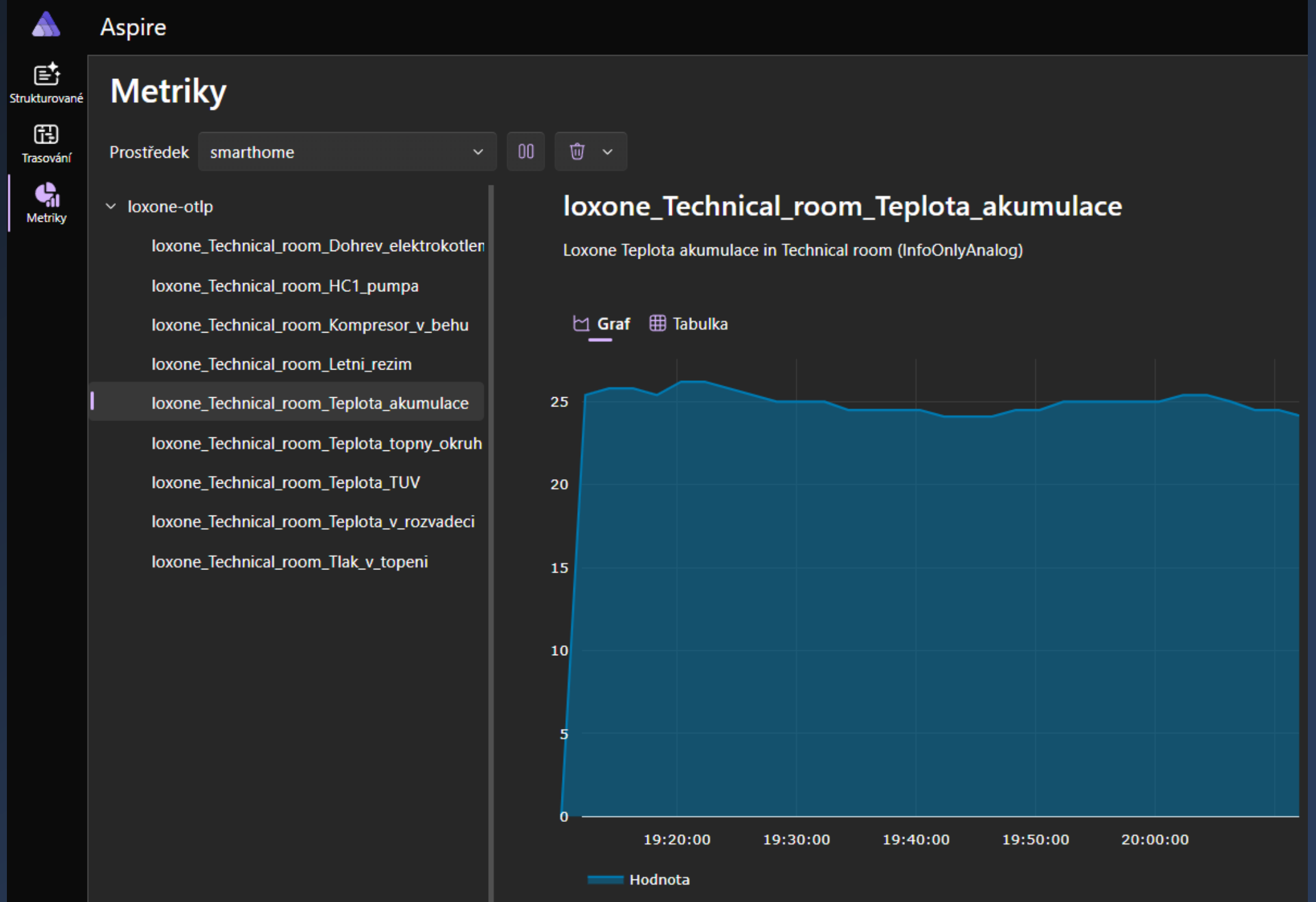
DEMO

Zero-code Instrumentation

OpenTelemetry Experiment: Custom OTLP Exporter

- Written in Go
- Reads data from Smarthome
- Sends telemetry over OTLP protocol




Aspire
Strukturované
Trasování
Metriky
Metriky
Prostředek
smarthome
loxone-otlp
loxone_Technical_room_Dohrev_elektrokotlen
loxone_Technical_room_HC1_pumpa
loxone_Technical_room_Kompresor_v_behu
loxone_Technical_room_Letni_rezim
loxone_Technical_room_Teplota_akumulace
loxone_Technical_room_Teplota_topny_okruh
loxone_Technical_room_Teplota_TUV
loxone_Technical_room_Teplota_v_rozvadeci
loxone_Technical_room_Tlak_v_topeni
loxone_Technical_room_Teplota_akumulace
Loxone Teplota akumulace in Technical room (InfoOnlyAnalog)
Graf
Tabulka
25
20
15
10
5
0
19:20:00
19:30:00
19:40:00
19:50:00
20:00:00
Hodnota


Rozvaděč
30.6 °C
Akumulace
25 °C
Tlak
2.10 bar
TUV
48.7 °C
Historie běhu
Pumpa okruh 1
Kompresor
Dohřev
20:40
20:45
20:50
20:55
21:00
21:05
21:10
21:15
21:20
21:25
21:30
21:35
21:40
21:45
21:50
21:55
New panel
30
28
26
24
22
20
On
Off
Akumulace
Topny okruh
Kompresor

https://github.com/jechtom/demo-open-telemetry

Tomáš Jecha

LinkedIn /in/jechtom | X @jechtom