

# Open Telemetry

The Rise of Open Standards in Observability

@Jaykul

April 7-10, 2025



SynEdgy



Jane Street



# Thank you Sponsors

---



PATCH MY PC

# Joel "Jaykul" Bennett

## Principal DevOps Engineer

Solving problems with code

15x Microsoft MVP for PowerShell

-  [github.com/Jaykul](https://github.com/Jaykul) and PoshCode
-  [discord.gg/PowerShell](https://discord.gg/PowerShell)
-  [HuddledMasses.org](https://HuddledMasses.org)
-  [@jaykul.powershell.social](https://@jaykul.powershell.social)
-  [@Jaykul@FOSStodon.org](https://@Jaykul@FOSStodon.org)



# Roadmap:

1. What is Observability?
2. What did we do before?
3. Seriously: What is Observability?
4. Demo: Does it work?
5. Why does it need standards?
6. What are the standards?
7. Wait, there are tools too?
8. Demo: Show us some code!

# What Does Observability Mean to You?

“ having enough information to understand a given situation ”

---

*Henrik Rexed (Dynatrace)*

“being able to look at things coming out of the box and tell what's going on inside”

---

*Amy Tobey (Equinix)*

“actually being able to go out into the unknown and understand how complex systems are performing ,”

---

*Rynn Mancuso (Honeycomb)*

“when you woke up at 2:00 am to go  
fix a problem, you can fix it.”

---

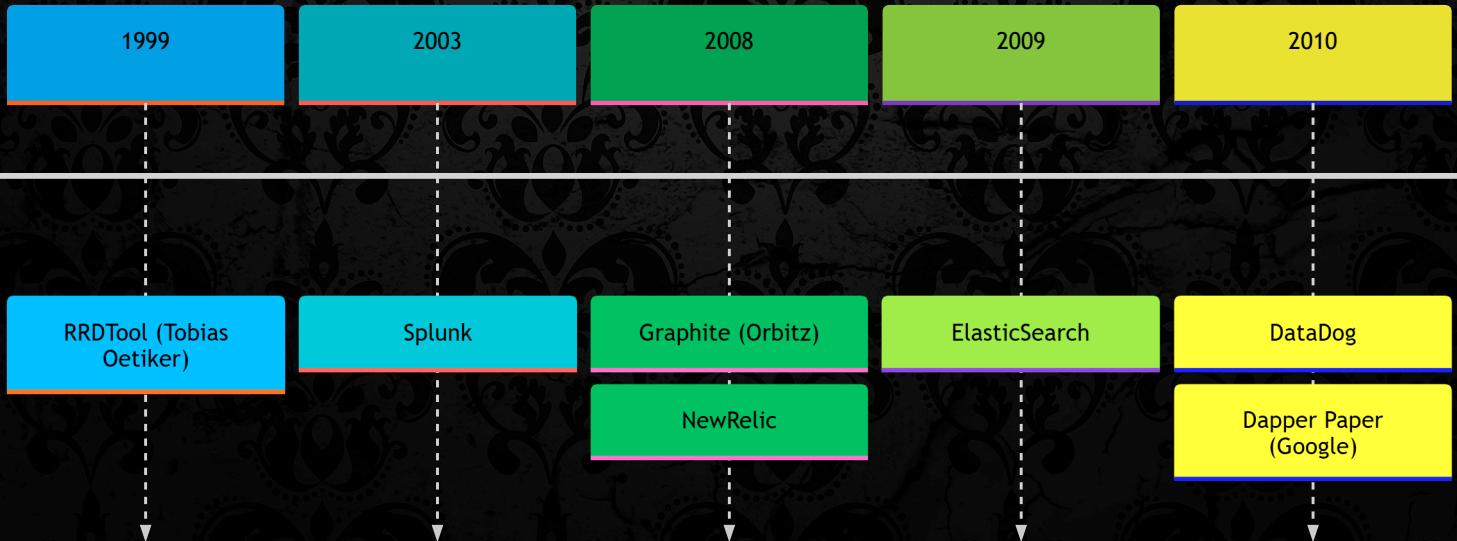
*Tyler Yahn (Splunk)*

“when something goes wrong, I can ask a question about my system and get a sense of what is happening without having to know ahead of time what to expect ”

---

Alex Boten (*Honeycomb*)

# A Brief History



# A Brief History



Zipkin (Twitter)

Kubernetes (Google)

Prometheus  
(SoundCloud)

Jaeger (Uber)  
(CNCF)

OpenTracing (CNCF)

OpenCensus  
(Google)

OpenTelemetry  
(Merger)

# What is Observability?

# What is Observability?

The ability to measure and understand the internal state of a system based on the data it generates

# What is Observability?

The ability to measure and understand the internal state of a system based on the data it generates

The more *observable* a system is, the easier it is to troubleshoot novel problems and get to the root cause

# Telemetry Signals

Data a system generates *in order* for us to be able to understand what is happening inside.

In general, we describe telemetry as a set of system outputs called ***signals***, which require a mechanism for context discovery and propagation.

Logging

Tracing

Metrics

Context

Propagation

# The USE Method

Popularized by Brendan Gregg (shouting guy), An Intel Fellow, and internationally renowned expert in computing performance. While at Netflix, pioneered the use of eBPF as an observability technology.

- Utilization (% time that the resource was busy)
- Saturation (amount of work resource has to do, e.g. queue length)
- Errors (count of error events)

# The RED Method

Popularized by Tom Wilkie, VP at Grafana, co-founder of Kausal, while he was at Weaveworks...

- *Rate* (the number of requests per second)
- *Errors* (the number of those requests that are failing)
- *Duration* (the amount of time those requests take)

# Four Golden Signals (LETS?)

Popularized by the Google SRE Book

- *Latency* (time taken to serve a request)
- *Errors* (rate of requests that are failing)
- *Traffic* (how much demand is placed on your system)
- *Saturation* (how "full" your service is)

# Demo Time

# Telemetry Layers

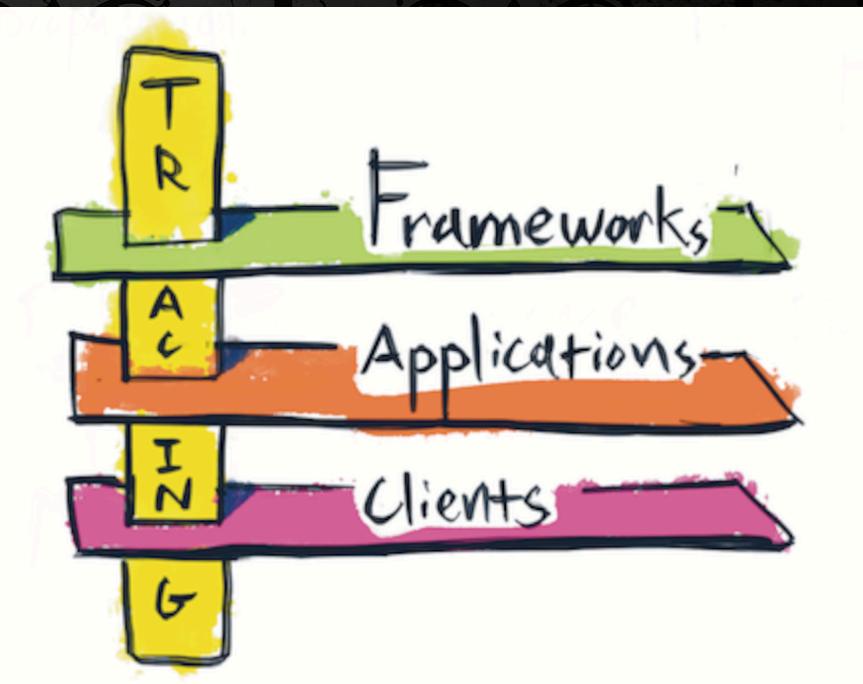
Each signal works as a cross-cutting concern, involving each part of the system.

Tracing      Metrics      Logs

Frameworks      APIs for each Metric Type

Applications      Libraries for each Language

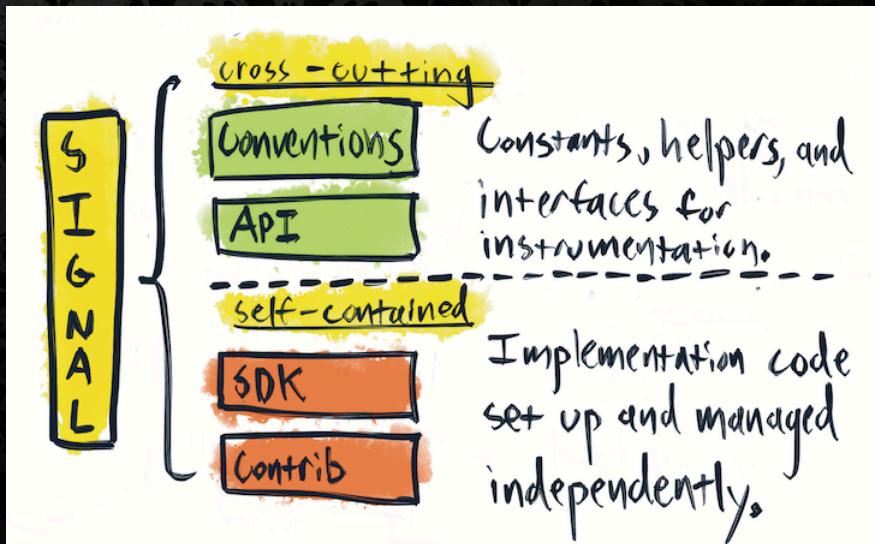
Collectors      Sidecars and Agents Galore



# OpenTelemetry Architecture

Specifications, Protocols, Standards and conventions.

But also APIs, SDKs, zero-code instrumentation agents, collectors, forwarders, and more.



# Open Standards and Specifications

## Open Telemetry Protocol (OTLP)

A gRPC protocol for encoding telemetry data and sending it from a client (any source of telemetry data) to a server, with special attention to "agents" and "forwarders" (which are both client and server).

- Logs
- Metrics
- Traces
- Profiling (in development)

# What is in the Protocol?

How deep shall we go?

# What is in the Protocol?

## How deep shall we go?

HTTP/2 changed the "on the wire" representation of HTTP for performance: a *binary encoding* of HTTP supports multiplexing over a single connection, making it faster and more efficient.

# What is in the Protocol?

## How deep shall we go?

HTTP/2 changed the "on the wire" representation of HTTP for performance: a *binary encoding* of HTTP supports multiplexing over a single connection, making it faster and more efficient.

gRPC is a general Remote Procedure Call framework built on HTTP/2 and Protocol Buffers. It features bi-directional streaming, load-balancing, service mesh, authentication...

# What is in the Protocol?

## How deep shall we go?

HTTP/2 changed the "on the wire" representation of HTTP for performance: a *binary encoding* of HTTP supports multiplexing over a single connection, making it faster and more efficient.

gRPC is a general Remote Procedure Call framework built on HTTP/2 and Protocol Buffers. It features bi-directional streaming, load-balancing, service mesh, authentication...

OTLP is *therefore* a high-throughput binary request/response protocol, supporting interleaved, concurrent requests and responses.



# Semantic Conventions

A common set of attributes which provide meaning to data

**Attribute:** a standardized name and data type

**Metric:** a standardized name, type, and unit of measure

For Resources, Events, Metrics, Logs, Trace Spans, performance Profiles, etc.

# A few examples

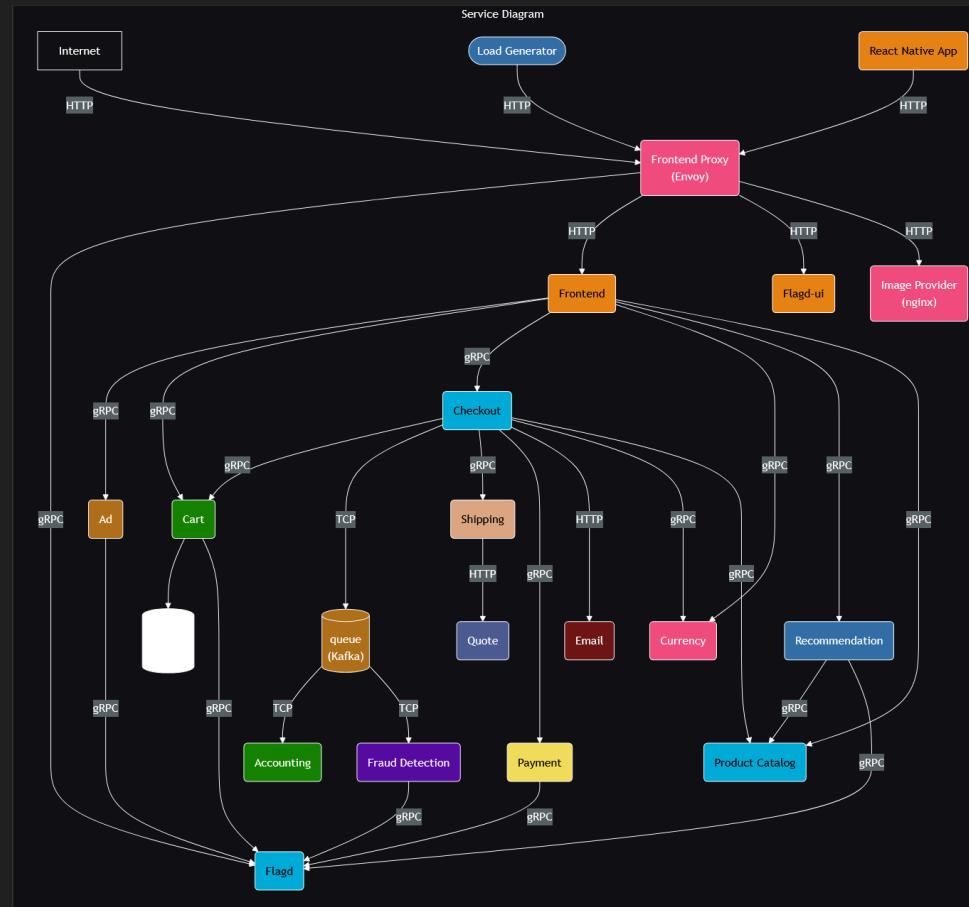
- Http Metrics
- CICD & VCS Metrics
- GenAI Metrics
- GraphQL Spans
- Code Attributes
- Azure Events
- Messaging Spans

# Let's see it

There is an [OpenTelemetry Demo](#) on GitHub with great docs, which you can run locally to see how OpenTelemetry works in practice.

Right now, I'm running it on [otel.poshcode.com](https://otel.poshcode.com) and you can even access [grafana](#) and [Jaeger](#) to see the data.

Please don't mess with the feature flags while I'm running the demo 😊



# THANK YOU!

Feedback is a  
gift. Please  
review this  
session!

