



Observability vs. Performance Monitoring: What's the difference and why should I care?

Agenda

- Why do we care about any of this?
- History of monitoring
- Introduction of hard-to-monitor technology/architecture
- Observability: the answer to the ultimate question
- Origin of the term + definition of observability
- How does observability benefit me?
- The three pillars of observability
- Quick Recap
- OpenTelemetry high level intro
- Wrap up

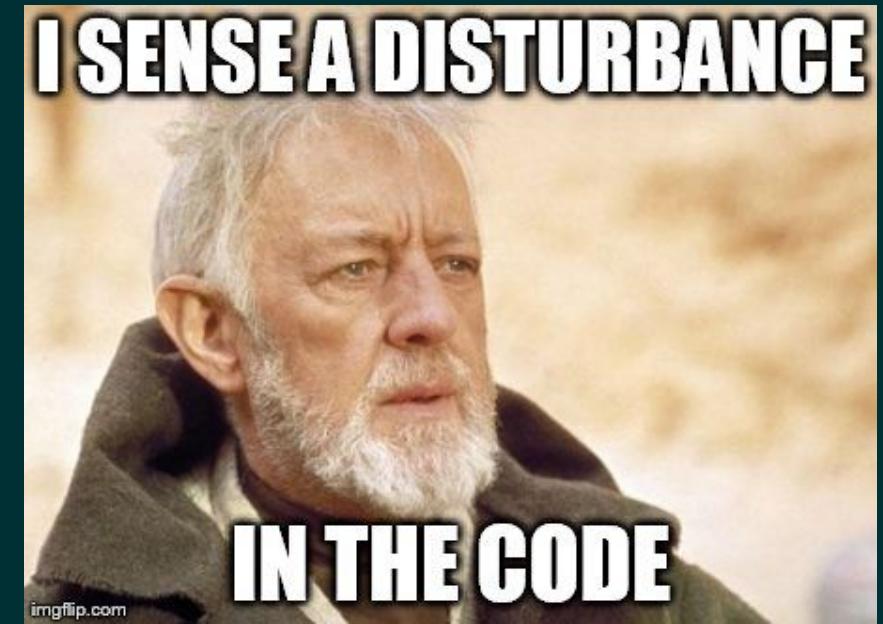
Why do we care?

We want to work on the good stuff

On average, 25% of a developer's time
is spent on debugging

Downtime can cost a company ~
\$5,600 per minute

We don't want to spend our nights and
weekends on call



Let's start at the beginning



Single code base (maybe single process)

Can likely be run locally

Debugger with breakpoints is reasonable

Good at what it's good for

Patrick A. Mackie
https://commons.wikimedia.org/wiki/File:Utah_Desert_Monolith.jpg

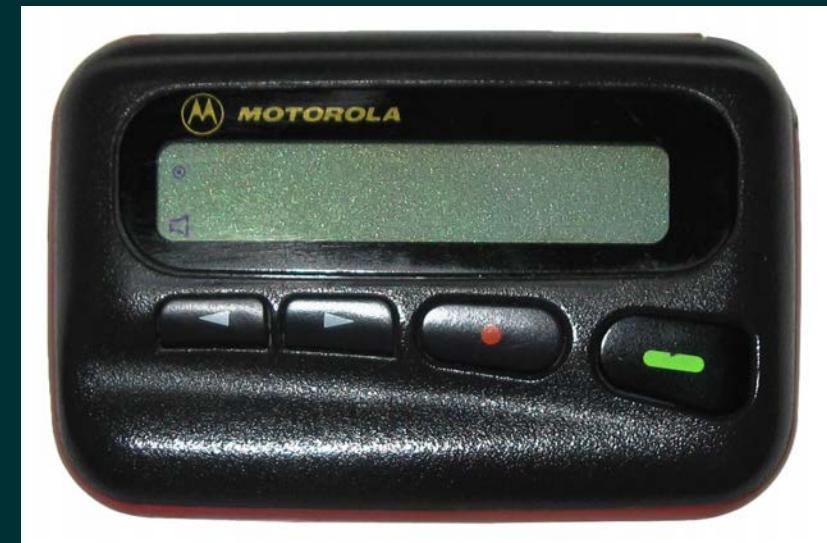
Performance Monitoring - tried and true

Born from the data center

What is the current state of the system?

What do the trends tell us?

When do I need to be alerted?



What else could we need?

Enter Distributed Systems

Improved scalability

More efficient

CI/CD

Modular/easier to understand

Harder to monitor



How do you monitor an ephemeral resource?

Traditional methods no longer apply



OK, so I need to turbo-charge my monitoring.

How do I start?

Observability: The Answer to the Ultimate Question of Life, the Universe, and Everything

42

Observability: The Answer to the Ultimate Question of Life, the Universe, and Everything

Observability is the ability to understand the state of and internal systems by observing the outputs

A large, three-dimensional blue number "42" is centered on a dark background. The number has a metallic texture and is surrounded by a bright, radial glow of light rays, suggesting it is a powerful or significant answer.

Observability: The Answer to the Ultimate Question of Life, the Universe, and Everything

Isn't that monitoring? Yes!

You can only monitor an observable system, so a system with monitoring *is* technically observable.



Confused yet? Just Remember:

- Outcomes are more important than labels
- Observability is never done (just like software)
- It doesn't have to be all or nothing, it doesn't have to be painful
- Observability is a spectrum from opacity to transparency, not binary



It doesn't matter what you call it, what matters is whether or not you can answer any question you need to ask of your system.

(I'm gonna keep talking about "observability" though)

Observability - the origin

“A measure of how well the internal states of a system can be inferred from knowledge of its external outputs.”

“On the General Theory of Control Systems”
August 1960, Science Digest -R. E. Kalman



Easy enough, capture all the data!

Log all the things - then what?

Building a haystack can leave you data rich but information poor

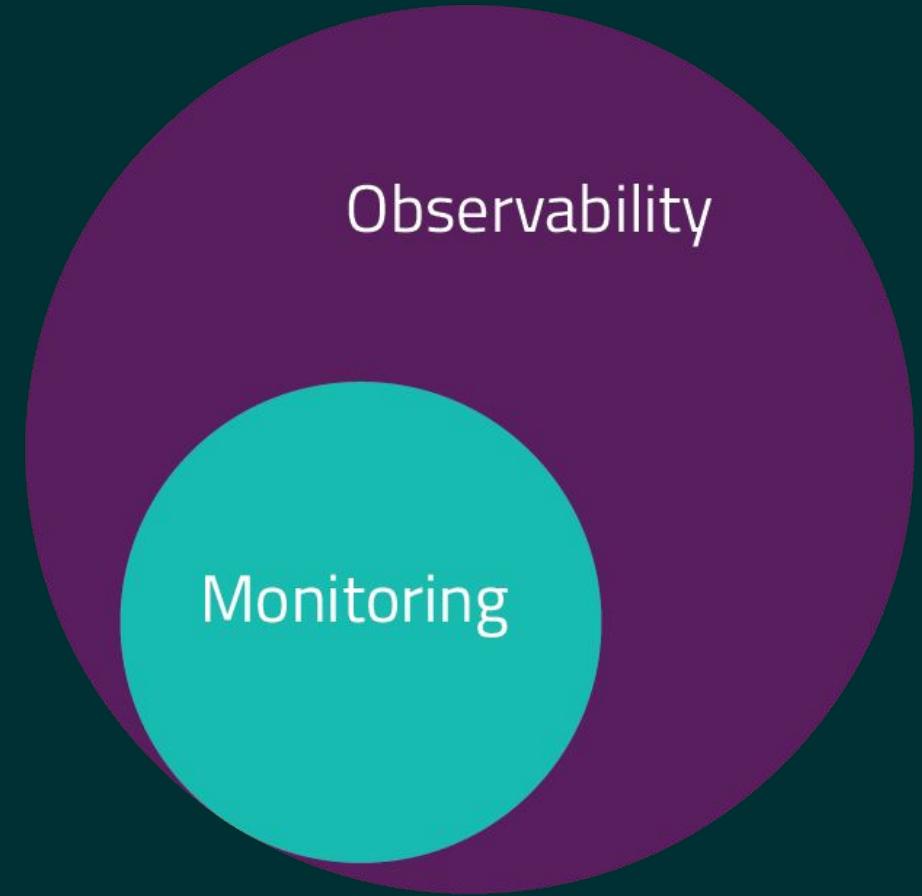
Grepping haystacks of log data is not reasonable



Do You Know the Unknowns?

The real key is:

- Monitoring asks a specific, known, question
 - How healthy is my system?
- Observability lets you ask any question, even one not anticipated
 - Observability can get you closer to a reliable system where user experiences are as expected



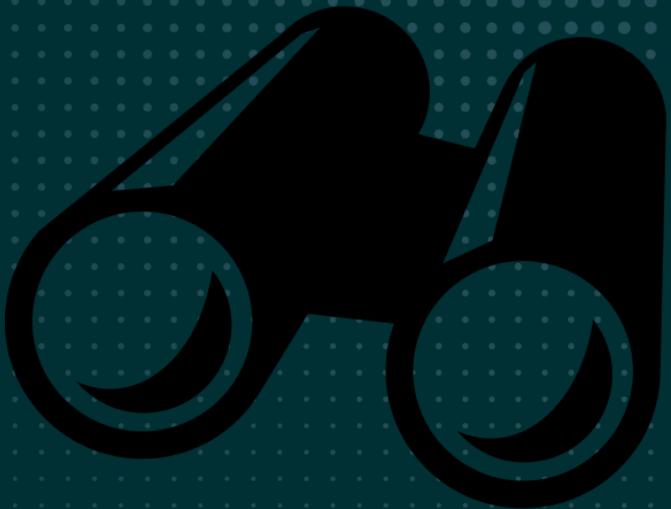
Can't I just buy an observability tool?

You can buy something, but that doesn't make your system observable

The right tool(s) can help (and some are open source!)

Many solutions are labeled as observability tools

The garbage in garbage out rule still applies



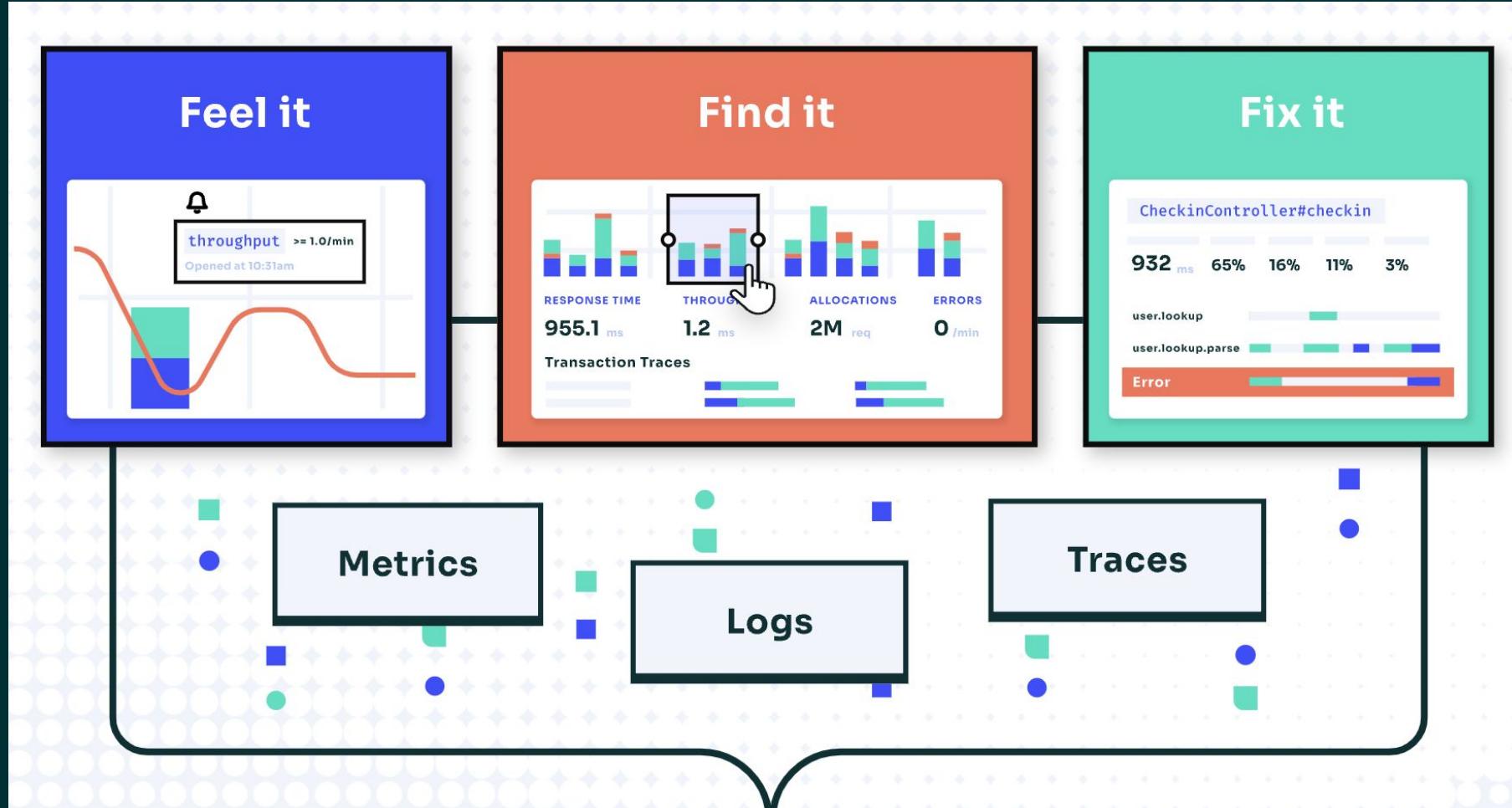
How does any of this help me?

An observable system enables you to fix a problem without prior knowledge or history of system issues, or by testing hypotheses (no sleuthing needed)



Now for a real world ~~nightmare~~ scenario.

You've been paged at 3am, now what?





**Have you tried turning it off
and then on again?**

Can you answer the questions?

- Who is being impacted?
 - Do I care at now or can it wait till tomorrow?
 - What do I have to do to resolve it?
 - Do I have the information available to me without pulling in someone else?
 - Where is the problem?
 - Am I starting a wild goose chase or do I have concrete information?
-
- When did these issues start?
 - Was this sudden or have we been building towards this failure?
 - Why did we end up in this state?
 - Do we know how to keep this from happening again?

Let's get down to the nitty gritty

The Three Pillars of Observability

Traces - Metrics - Logs

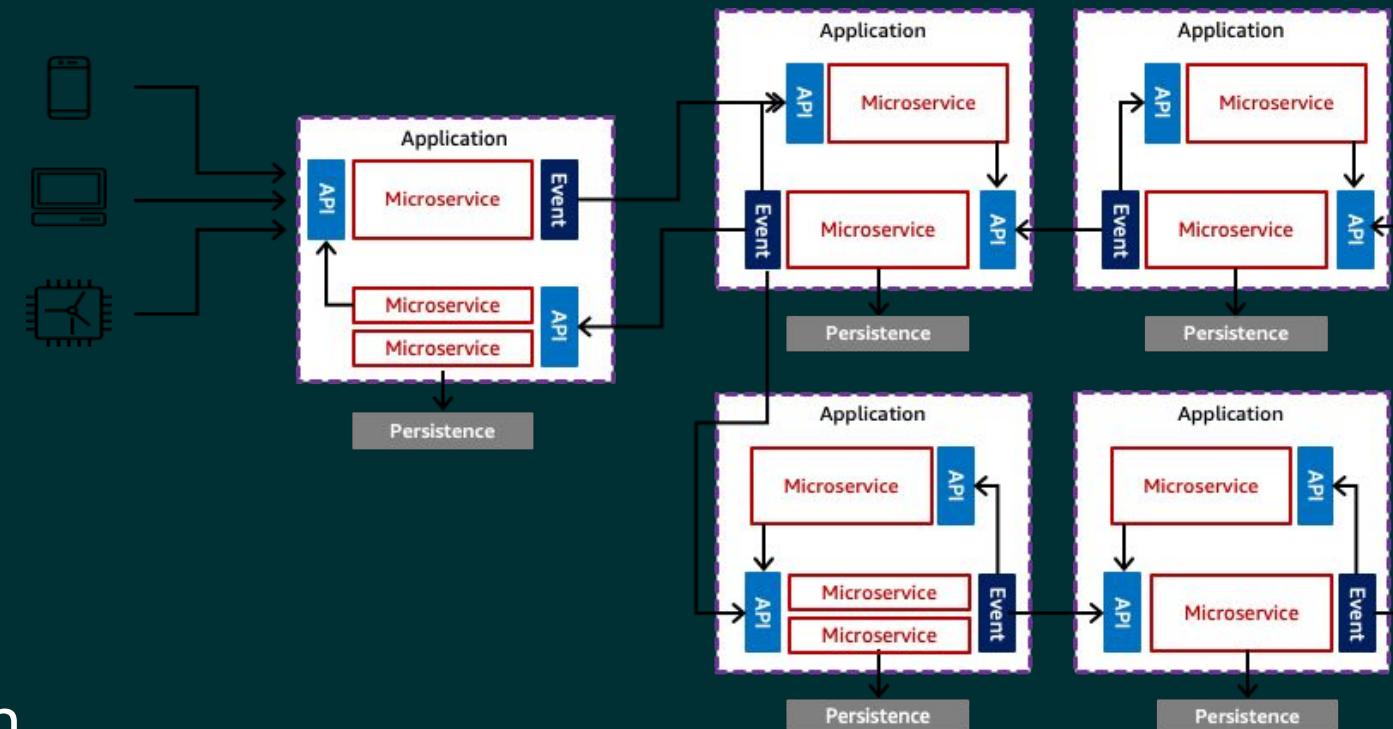
Metrics

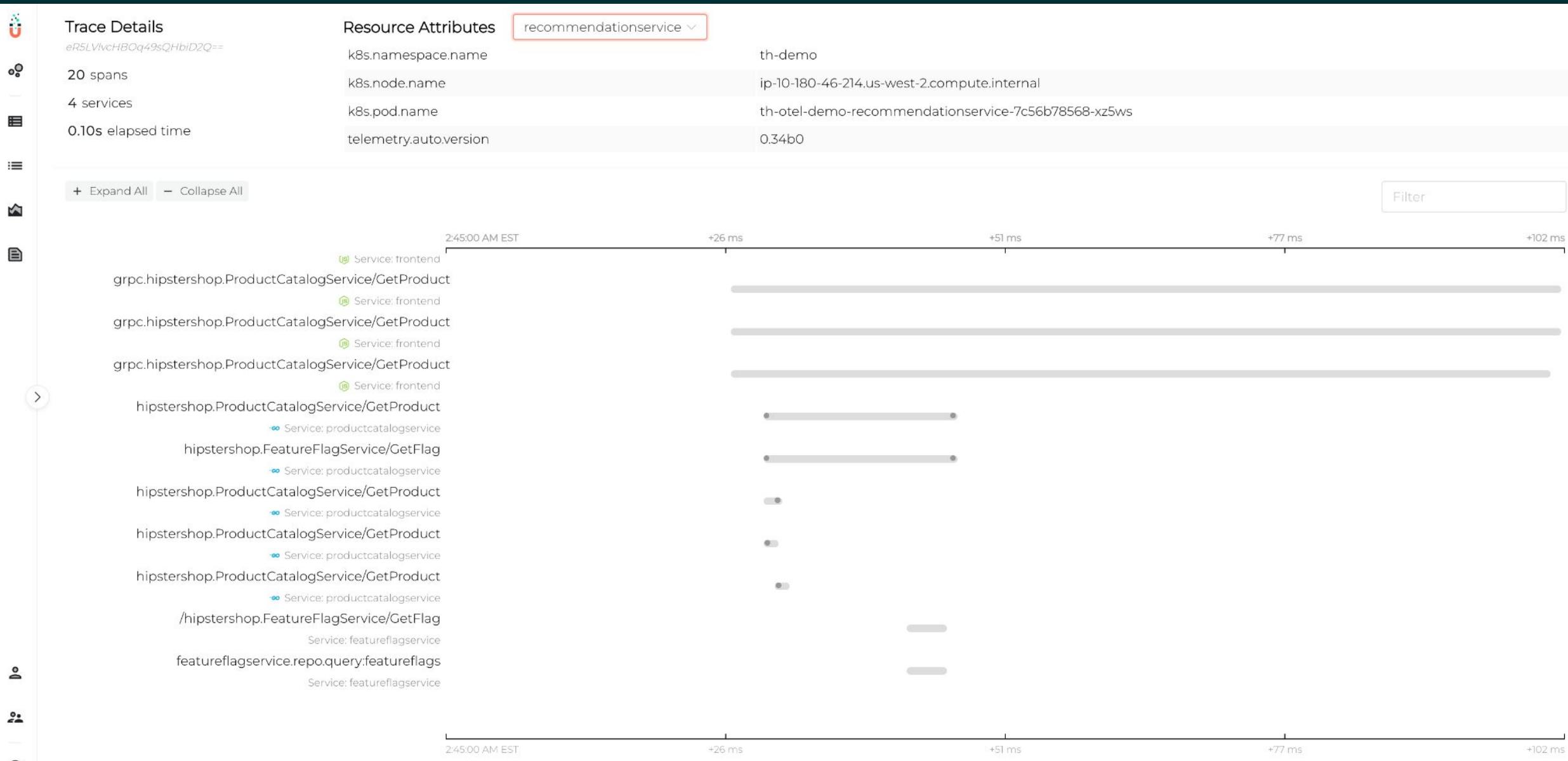
- Metrics are intended to provide statistical information in aggregate
- This can give a good indication of the current state of your system and can be used to trigger events like scaling infrastructure to support increasing load, or alerting in the event of an outage (hello 3am page)
- Metrics are by nature low cardinality data which is good for human consumption and storage of data, but bad for diagnostics which is why alone, they are not enough to understand the state of your system

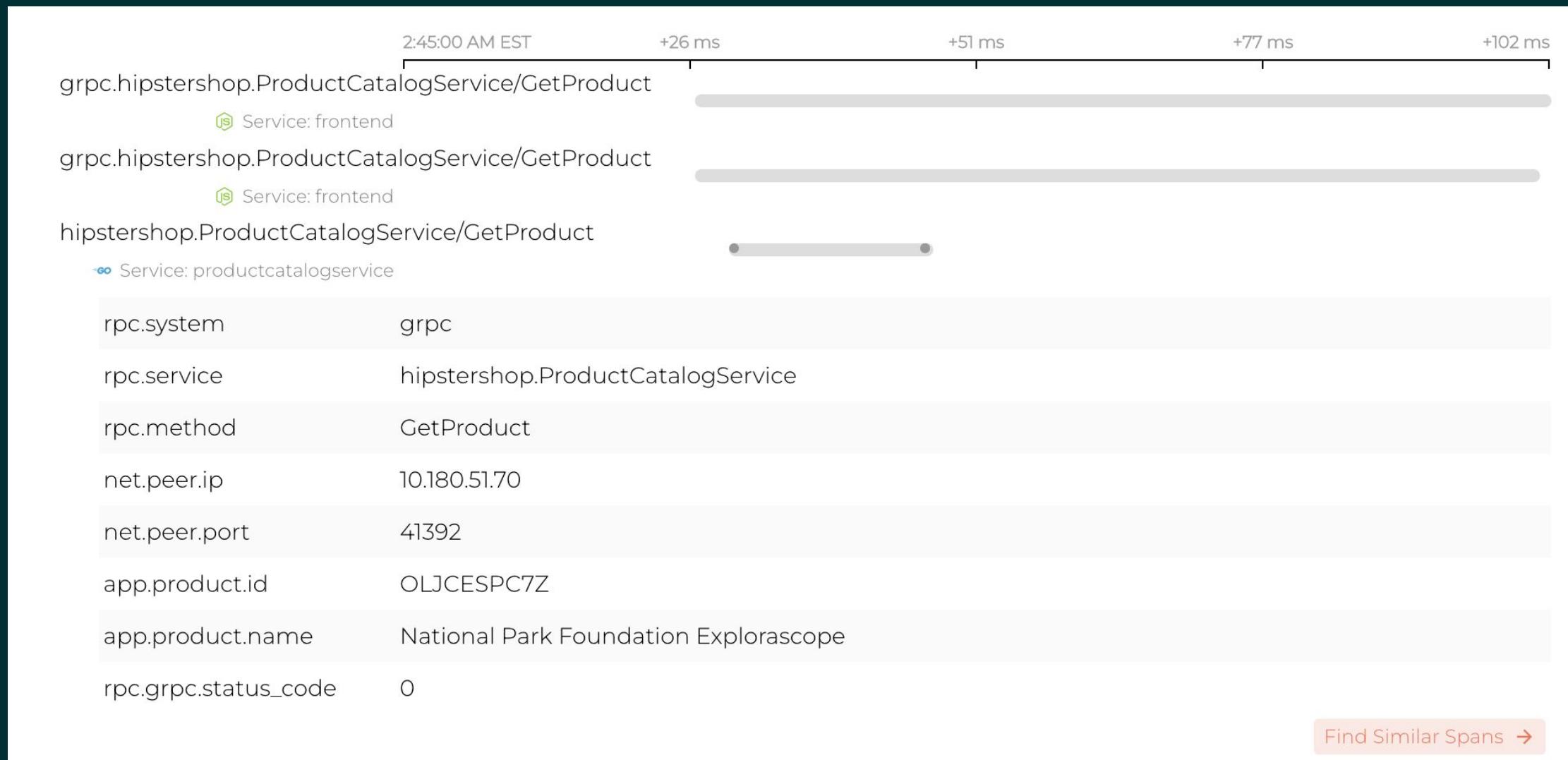


Distributed Traces

- Tracing across systems to understand the entire path a request took
- This requires trace and span ids to be propagated across the lifecycle of the request so that all of the players can be accounted for (context propagation)
- With these trace and span IDs in your logs, you have effectively correlated your telemetry signals







Logs - the opposite of an aggregate

Herein lies the answer to your question, if you can find it.

Contain timestamp, level, metadata - information about the state of the system

High cardinality data aids diagnosis, makes storage and exploration tricky



Quick Recap

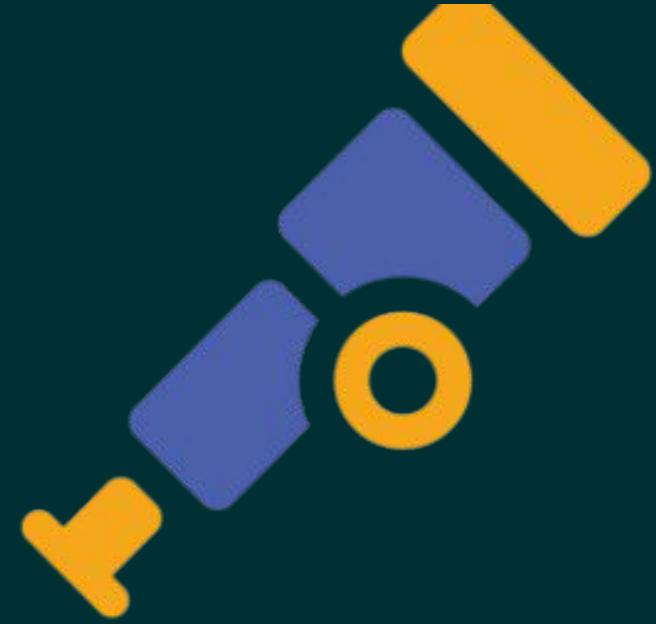
Collecting data does not make a system observable

The value is the ability to answer questions

You can amass huge amounts of data, but without being able to effectively navigate it to realize value, it's just wasting space (and \$\$)

You may experience tooling (and maybe invoice) fatigue

Your needs and team will dictate the solutions needed
(You don't have to buy a Ferrari if a Smart car will do the trick)



OpenTelemetry

Enter OpenTelemetry

www.opentelemetry.io

- 2nd most active Cloud Native Computing Foundation (CNCF) project behind Kubernetes
- Backed by 130+ companies, including AWS, Google, Microsoft, Facebook, Splunk & Scout
- Integrated directly into open source projects and cloud-native stacks
- Simple to use, with extensive ability to customize instrumentation specific to each environment



OpenTelemetry - A Shared Standard

- Provides a shared concept of observability “Signals” - metrics, traces, logs
- Provides a shared protocol for sending and receiving these Signals
- Provides open-source SDKs in popular languages to instrument code
- Everyone can contribute!



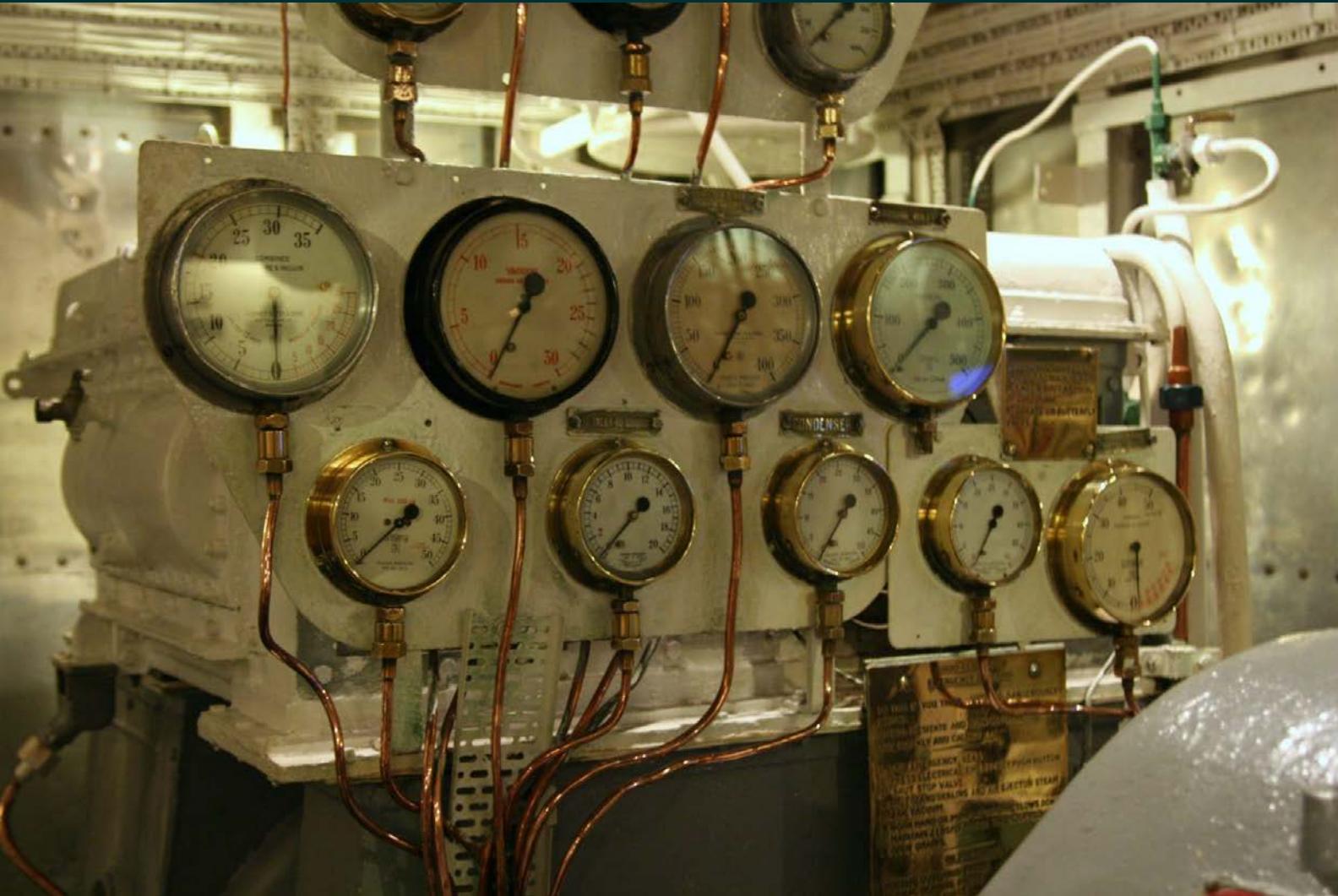
OpenTelemetry

Components of OpenTelemetry

- Cross-language specification
- Tools to collect, transform, and export telemetry data
- Per-language SDKs
- Automatic instrumentation and contrib packages



I thought open source meant DIY



This does not have to be hard

The TL;DR Version

Dependencies you'll need to install get started:

```
pip install opentelemetry-api, opentelemetry-sdk, opentelemetry-exporter-otlp
```

Imports you'll need to get started:

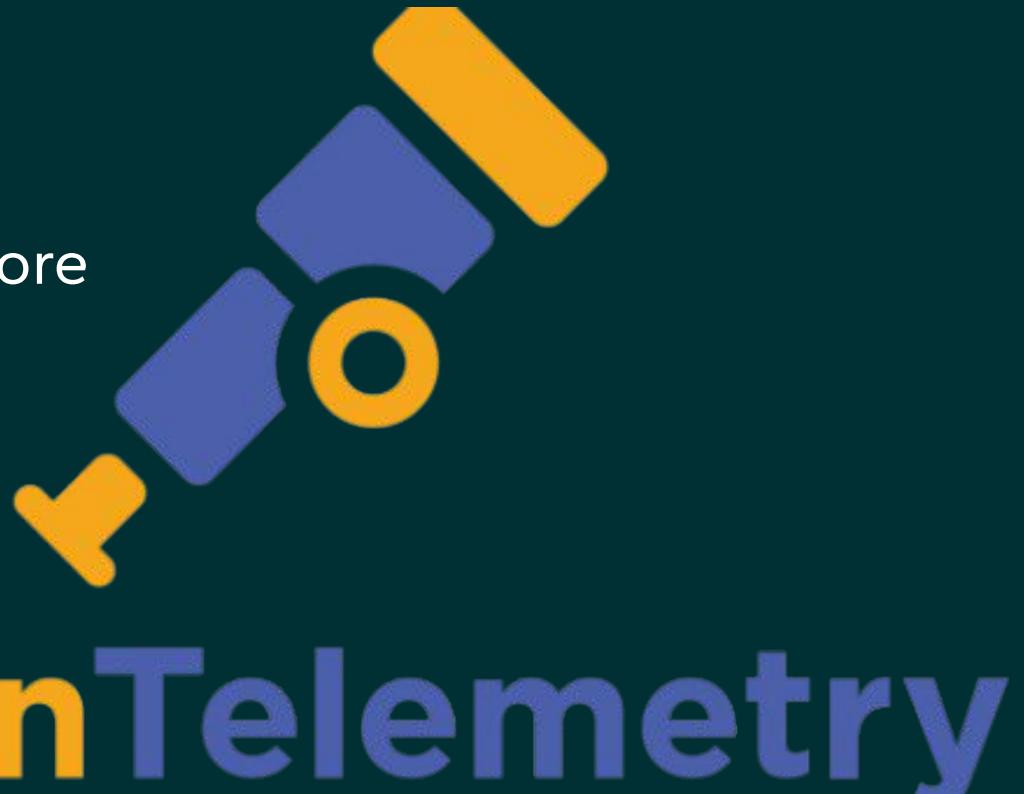
```
from opentelemetry import trace
from opentelemetry.exporter.otlp.proto.grpc.trace_exporter import OTLPSpanExporter
from opentelemetry.sdk import resources
from opentelemetry.sdk.trace import TracerProvider
from opentelemetry.sdk.trace import export as trace_export
```

Code to get you started sending telemetry data:

```
resource = resources.Resource(
    attributes={
        resources.SERVICE_NAME: "MY_SERVICE_NAME",
        resources.SERVICE_NAMESPACE: "MY_SERVICE_NAMESPACE",
    }
)
span_exporter = OTLPSpanExporter(
    headers={"x-telemetryhub-key": "$YOUR_INGEST_KEY"},
    endpoint="https://otlp.telemetryhub.com:4317"
)
trace_processor = trace_export.BatchSpanProcessor(span_exporter)
trace_provider = TracerProvider(resource=resource)
trace.set_tracer_provider(trace_provider)
trace_provider.add_span_processor(trace_processor)
```

The OTel Collector

- Receive - supports receiving telemetry data in many formats
- Process - process and filter data before sending to a backend
- Export - send telemetry data to one or more backends



Utilizing the Outputs?

Where can you send your OpenTelemetry data?

Anywhere you want, OpenTelemetry is vendor agnostic!

Dip your toe in the water with some free tools and level up to larger vendors as needed.

Use the OTEL Collector to stream your data to one or more backends and manage exactly what goes where.

Take advantage of that propagated context for the “single pane of glass” view of your systems.

A quick analogy, and we're outta here!

Home Cooking

- Just you and your Cuisinart
- Your kitchen is totally observable
- Every dish is easily tracked



Restaurant Operations

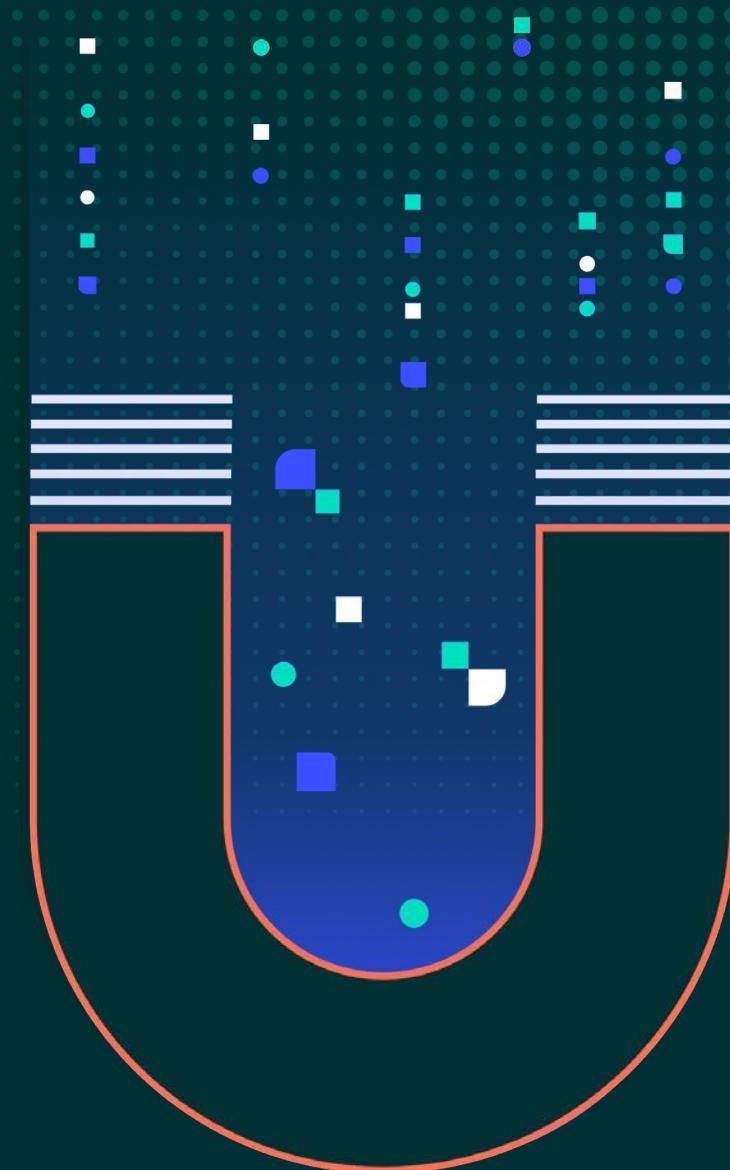
- Multiple stations
- One order takes input from many locations
- How do we know who oversalted the soup?





Thank you!

sarah@telemetryhub.com





scoutapm.com/opentelemetry



CI/CD has shortened the gap between IDE<>Prod

No longer promote code through a series of environments testing along the way

It's now very difficult to replicate a production environment for development or staging

Understanding the impact of a release immediately and being able to pinpoint the genesis of an issue and tie it to a commit is key to managing release problems

Title and content slide

Phasellus hendrerit dui in congue bibendum

Integer lorem nulla, varius fringilla rutrum imperdiet, ullamcorper non elit. Phasellus molestie pharetra congue. Phasellus eu eleifend turpis, ut volutpat est.

- Nam at cursus dolor. Fusce a dapibus lectus, a tristique quam.
- Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.
- Sed sit amet efficitur libero, et imperdiet orci.

Quisque sit amet tincidunt velit. Maecenas nibh dolor, vehicula sodales urna nec, pellentesque finibus tellus. Praesent convallis fringilla lorem sed suscipit.

Title only slide - light

Phasellus hendrerit dui in congue bibendum

Title only slide - dark

Phasellus hendrerit dui in congue bibendum

Title Only Slide – Example use

Fusce a dapibus lectus, a tristique quam



Aenean a euismod



Fusce semper dolor quis pulvinar

Quisque nec accumsan nibh. Fusce semper dolor quis ex pulvinar vel dignissim dolor egestas.



Fusce semper dolor quis pulvinar

Quisque nec accumsan nibh. Fusce semper dolor quis ex pulvinar vel dignissim dolor egestas.



Fusce semper dolor quis pulvinar

Quisque nec accumsan nibh. Fusce semper dolor quis ex pulvinar vel dignissim dolor egestas.



Fusce semper dolor quis pulvinar

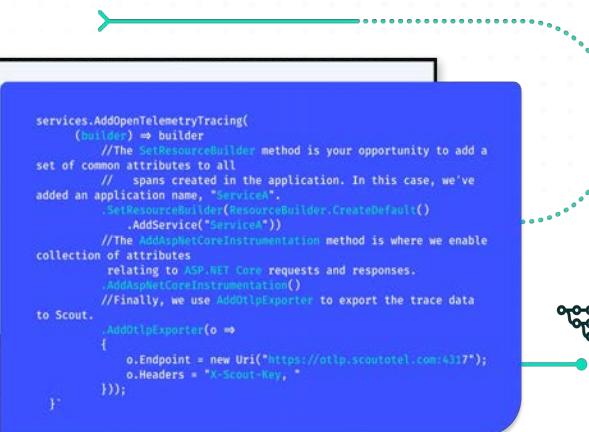
Quisque nec accumsan nibh. Fusce semper dolor quis ex pulvinar vel dignissim dolor egestas.

Two content slide

Phasellus hendrerit dui in congue bibendum

Two content & images slide

Lorem ipsum elit



```
services.AddOpenTelemetryTracing()
    .Builder()
    //The SetResourceBuilder method is your opportunity to add a
    // set of common attributes to all
    // spans created in the application. In this case, we've
    // added an application name, "ServiceA".
    .SetResourceBuilder(ResourceBuilder.CreateDefault()
        .AddService("ServiceA"))
    //The AddAspNetCoreInstrumentation method is where we enable
    // collection of attributes
    // relating to ASP.NET Core requests and responses.
    .AddAspNetCoreInstrumentation()
    //Finally, we use AddOtlpExporter to export the trace data
    to Scout.
    .AddOtlpExporter(o =>
    {
        o.Endpoint = new Uri("https://otlp.scoutotel.com:4317");
        o.Headers = "X-Scout-Key";
    }));
}
```

Lorem ipsum elit

```
const OTLPOptions = {
  url: "undefined",
  headers: {
    X-Scout-Key: "undefined"
  },
};
const otlpExporter = new OTLPTraceExporter(OTLPOptions);
const sdk = new opentelemetry.NodeSDK({
  traceExporter: otlpExporter,
  instrumentations: [getNodeAutoInstrumentations()],
});
};
```



Features slide – One row – Light



**Fusce semper dolor
quis pulvinar**



**Dignissim dolor
egestas**



**Euismod lorem
consequat quis**

Comparison slide with icons – color



Title lorem ipsum elit

Integer posuere, diam nec suscipit
premium, diam ante tristique augue, vel
dapibus orci nisi sed sem. Nulla facilisi.
Aliquam eu bibendum magna.



Title lorem ipsum elit

Integer posuere, diam nec suscipit
premium, diam ante tristique augue, vel
dapibus orci nisi sed sem. Nulla facilisi.
Aliquam eu bibendum magna.

Title & image slide

 Lorem ipsum dolor sit amet,
 consectetur adipiscing elit.
 Suspendisse lobortis molestie
 velit. Vestibulum mollis orci in
 efficitur vestibulum. Praesent
 vehicula aliquam magna sed
 ornare.

```
// this will be needed to get a tracer
from opentelemetry import trace
// and an exporter with span processor
from
opentelemetry.exporter.otlp.proto.grpc.trace_exporter
import (
OTLPSpanExporter,
)

// Connect to Scout Otel by configuring the exporter
// with your endpoint and access token.
headers = {"X-Scout-Key": " "}
options = {"headers": headers}
span_exporter = OTLPSpanExporter(
    **options, endpoint="https://otlp.scoutotel.com:4317"
)
```

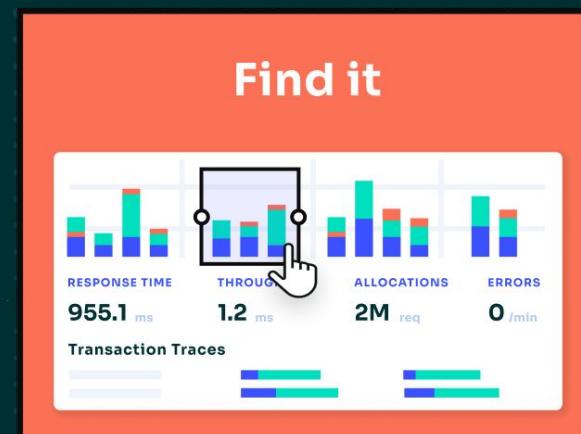
Title & image slide

 Lorem ipsum dolor sit amet,
 consectetur adipiscing elit.
 Suspendisse lobortis molestie
 velit. Vestibulum mollis orci in
 efficitur vestibulum. Praesent
 vehicula aliquam magna sed
 ornare.

```
// this will be needed to get a tracer
from opentelemetry import trace
// and an exporter with span processor
from opentelemetry.exporter.otlp.proto.grpc.trace_exporter
import (
    OTLPSpanExporter,
)

// Connect to Scout Otel by configuring the exporter with
// your endpoint and access token.
headers = {"X-Scout-Key": ""}
options = {"headers": headers}
span_exporter = OTLPSpanExporter(
    **options, endpoint="https://otlp.scoutotel.com:4317"
)
```

Blank slide - any content



Metrics

Logs

Traces

TelemetryHub

Comparison slide

Title lorem ipsum elit

Integer posuere, diam nec suscipit
pretium, diam ante tristique augue, vel
dapibus orci nisi sed sem. Nulla facilisi.
Aliquam eu bibendum magna.

- Nulla facilisi aliquam eu bibendum
- Diam ante tristique augue
- Dapibus orci nisi sed sem

Title lorem ipsum elit

Integer posuere, diam nec suscipit
pretium, diam ante tristique augue, vel
dapibus orci nisi sed sem. Nulla facilisi.
Aliquam eu bibendum magna.

- Nulla facilisi aliquam eu bibendum
- Diam ante tristique augue
- Dapibus orci nisi sed sem

Comparison slide with icons – light



Title **lorem ipsum elit**

Integer posuere, diam nec suscipit
pretium, diam ante tristique augue, vel
dapibus orci nisi sed sem. Nulla facilisi.
Aliquam eu bibendum magna.



Title **lorem ipsum elit**

Integer posuere, diam nec suscipit
pretium, diam ante tristique augue, vel
dapibus orci nisi sed sem. Nulla facilisi.
Aliquam eu bibendum magna.

Features slide - One row - Dark



Fusce semper dolor quis pulvinar

Quisque nec accumsan nibh. Fusce semper dolor quis ex pulvinar vel dignissim dolor egestas. In tristique ligula nec leo fermentum sit amet scelerisque.



Dignissim dolor egestas

Quisque nec accumsan nibh. Fusce semper dolor quis ex pulvinar vel dignissim dolor egestas. In tristique ligula nec leo fermentum sit amet scelerisque.



Euismod lorem consequat quis

Quisque nec accumsan nibh. Fusce semper dolor quis ex pulvinar vel dignissim dolor egestas. In tristique ligula nec leo fermentum sit amet scelerisque.

Features slide – Two rows – Light



Fusce semper dolor quis pulvinar



Fusce semper dolor quis pulvinar



Fusce semper dolor quis pulvinar



Fusce semper dolor quis pulvinar



Fusce semper dolor quis pulvinar



Fusce semper dolor quis pulvinar

Features slide - Two rows - Dark



Fusce semper dolor quis pulvinar

Quisque nec accumsan nibh. Fusce semper dolor quis ex pulvinar vel dignissim dolor egestas. In tristique ligula nec leo fermentum.



Fusce semper dolor quis pulvinar

Quisque nec accumsan nibh. Fusce semper dolor quis ex pulvinar vel dignissim dolor egestas. In tristique ligula nec leo fermentum.



Fusce semper dolor quis pulvinar

Quisque nec accumsan nibh. Fusce semper dolor quis ex pulvinar vel dignissim dolor egestas. In tristique ligula nec leo fermentum.



Fusce semper dolor quis pulvinar

Quisque nec accumsan nibh. Fusce semper dolor quis ex pulvinar vel dignissim dolor egestas. In tristique ligula nec leo fermentum.



Fusce semper dolor quis pulvinar

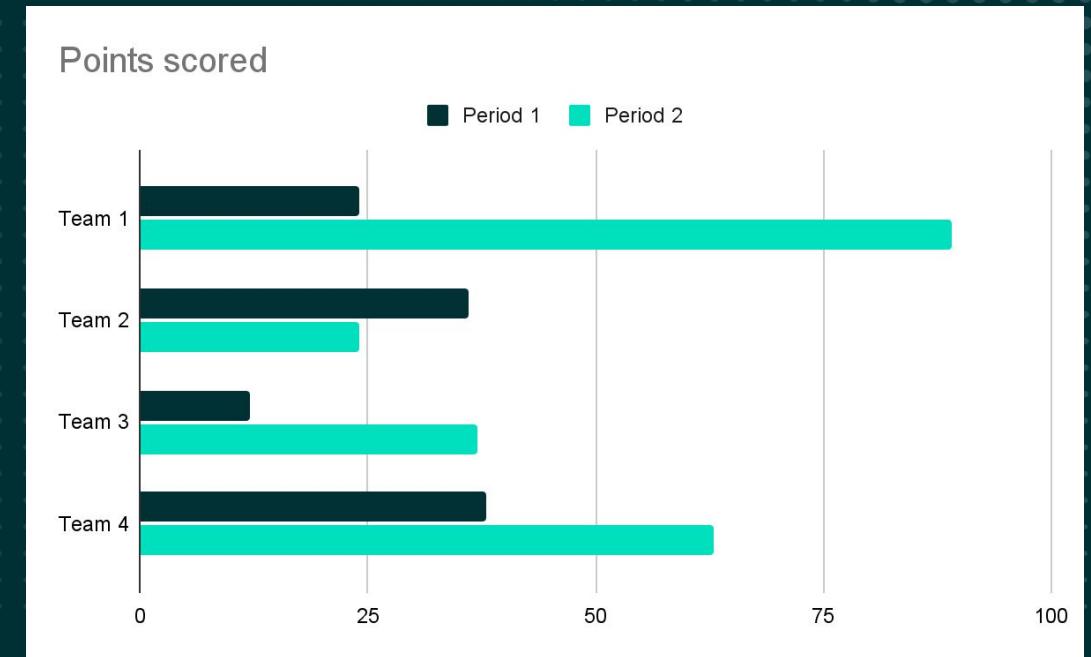
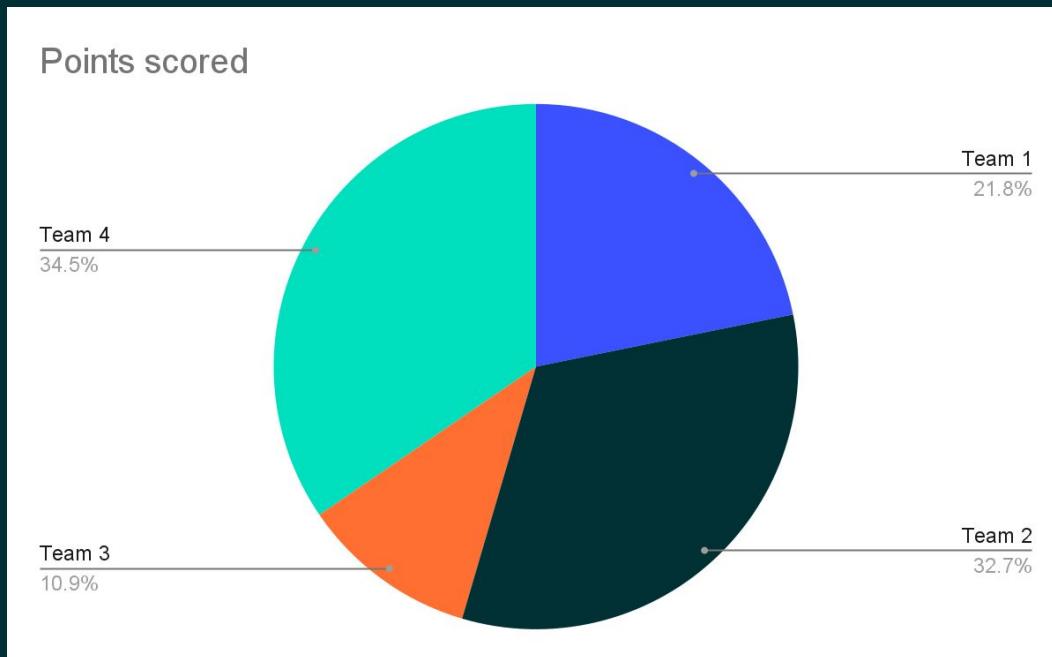
Quisque nec accumsan nibh. Fusce semper dolor quis ex pulvinar vel dignissim dolor egestas. In tristique ligula nec leo fermentum.



Fusce semper dolor quis pulvinar

Quisque nec accumsan nibh. Fusce semper dolor quis ex pulvinar vel dignissim dolor egestas. In tristique ligula nec leo fermentum.

Charts & Graphs (Two image slide)



Section Divider

Vestibulum eu venenatis ex dictum molestie erat

Section Divider

Vestibulum eu venenatis ex dictum molestie erat

Section Divider – Custom Image

Vestibulum eu venenatis ex dictum molestie erat

Section Divider - Custom Image

Vestibulum eu venenatis ex dictum molestie erat

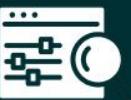


Thank you.

Fusce a dapibus lectus a tristique quam

Aliquam erat volutpat. Integer quis erat tellus.

Icon Library - On Light



Icon Library - On Color



Icon Library - On Dark

