



Distributed Tracing:

The Good, the Bad, and the Ugly

Ricardo Ferreira

Senior Developer Advocate
Amazon Web Services

Before we begin:

Happy hour: it starts at 4:00 PM.

New trailer for Thor: Love and Thunder.

I welcome your questions anytime.

What is distributed tracing anyway?

**It is a type
of logging.**

Logging 101

```
type Log struct {  
    request_path string  
    request_size int64  
    status int32  
    latency_ms float64  
}
```

```
logEntry := Log{  
    "/customers/find",  
    840, 200, 35  
}  
  
fmt.Printf("%+v", logEntry)
```

Stitching additional data is hard

```
type CustomLog struct {  
    request_path string  
    request_size int64  
    status int32  
    latency_ms float64  
    customer_id int64  
    database_id int64  
}
```

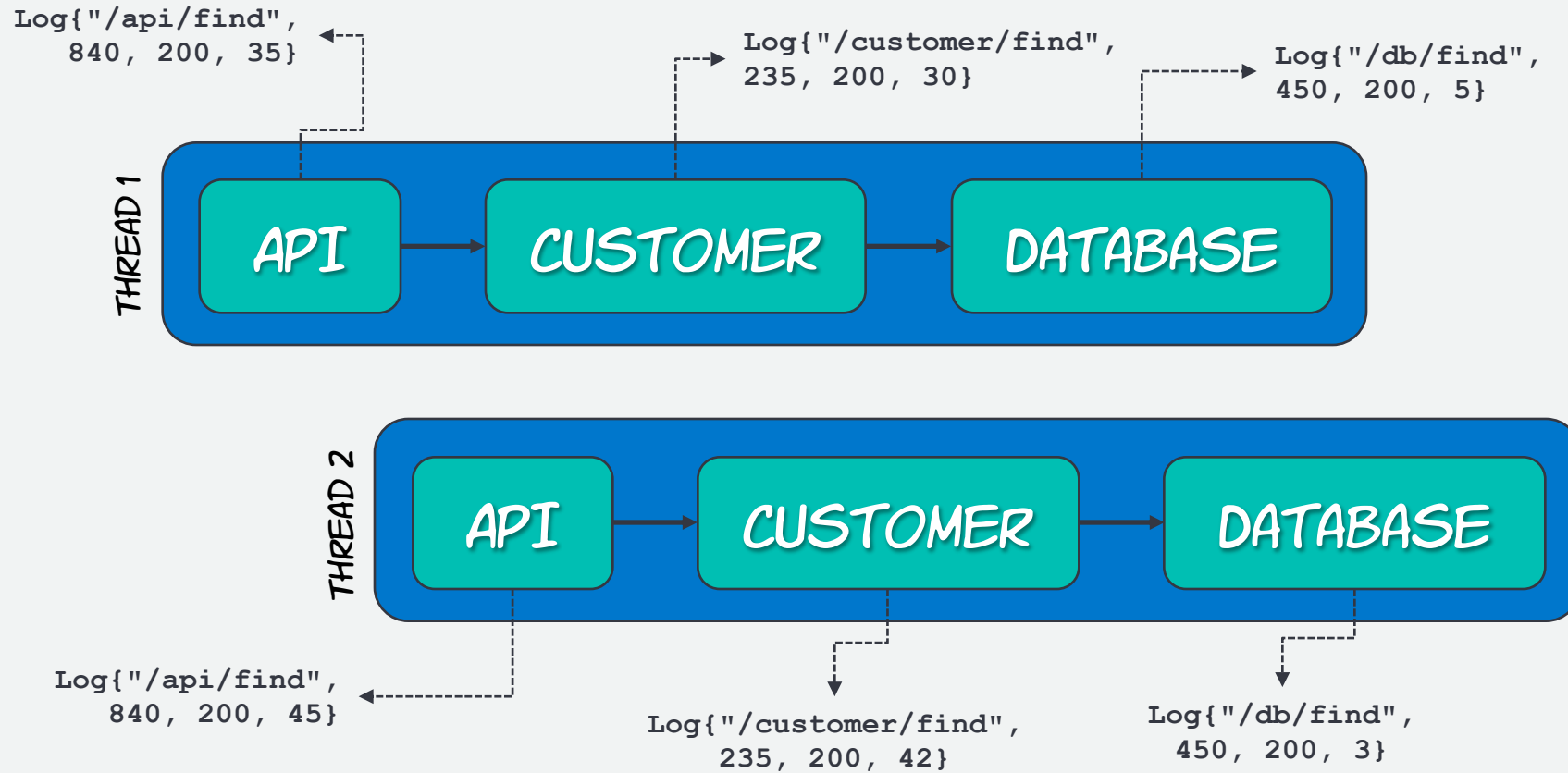
```
logEntry := CustomLog{  
    requestPath(),  
    requestSize(),  
    requestStatus(),  
    latencyMS(),  
    cust.customerID(),  
    db.databaseTenantID()  
}
```



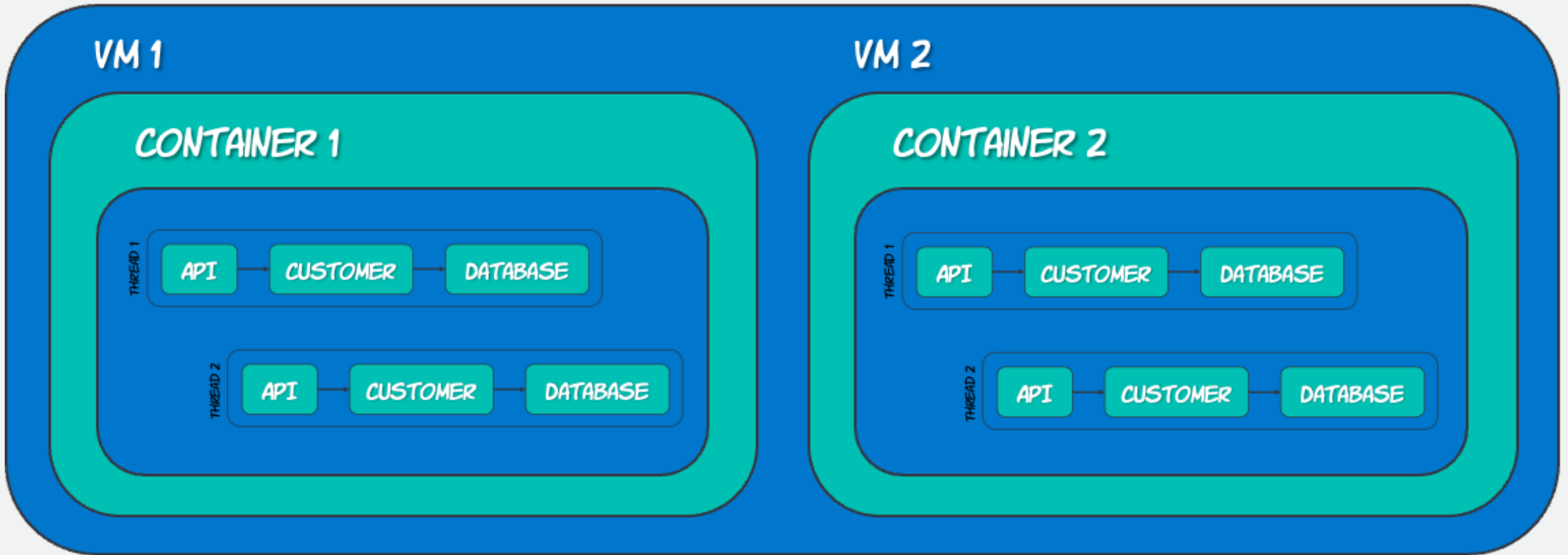
BUT WHY THO?!

**Because logging
with distributed
systems is hard.**

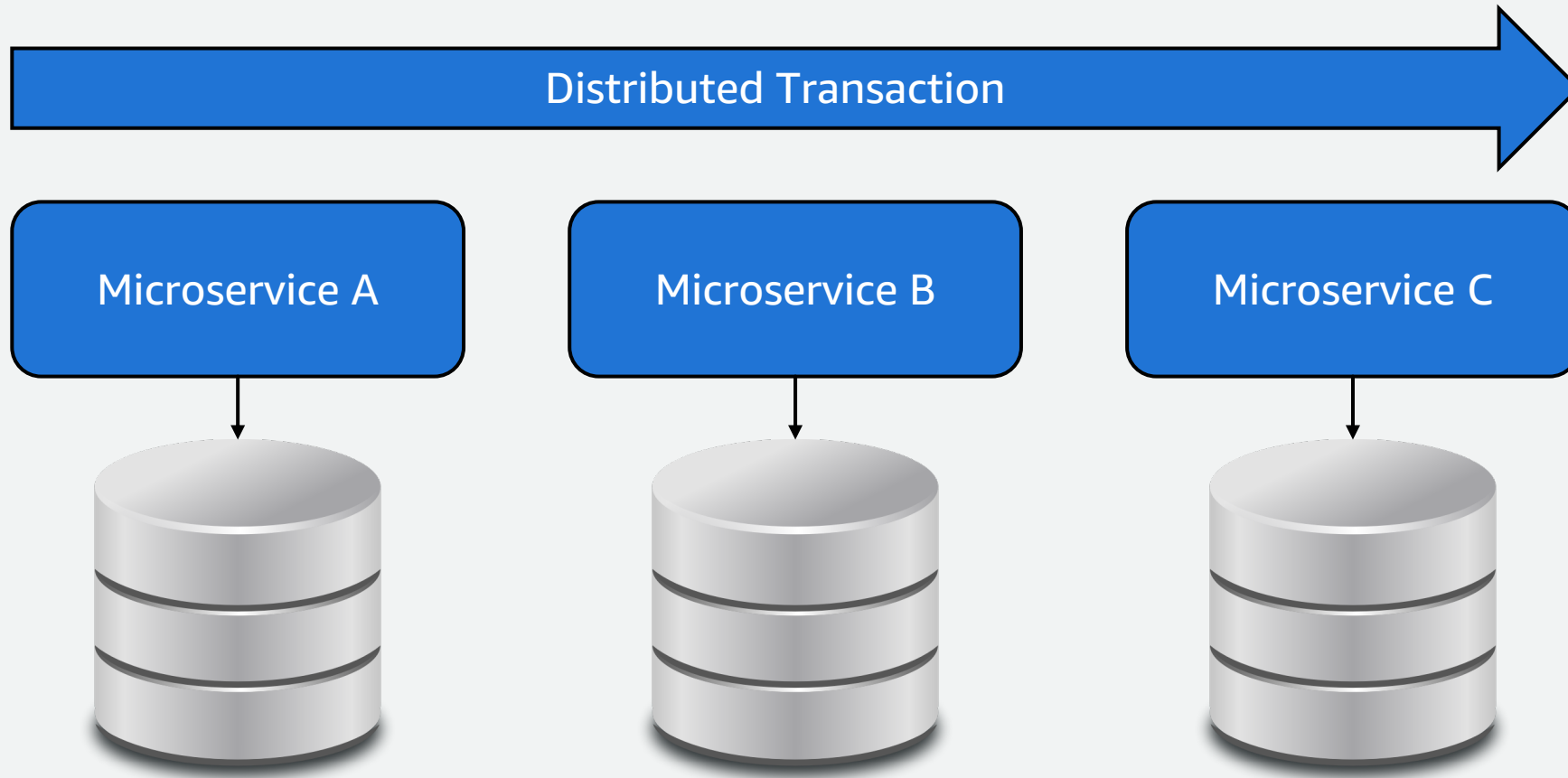
Single-machine threading



Correlating network distributed threads is hard



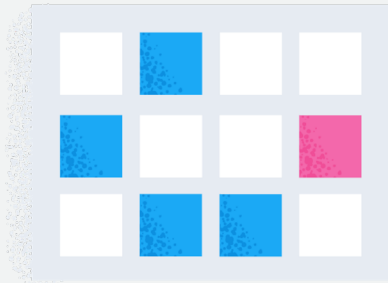
Reason why is called distributed tracing



Black-box versus white-box instrumentation

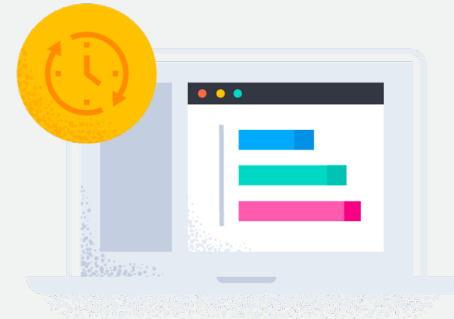
Black-box

- Code is not changed
- Handled by some runtime
- Minimal execution visibility



White-box

- Require code changes
- Handled by the application
- Better execution visibility



**Ideally; you
should use
both of them.**

**But what
about logs
and metrics?**

The famous trio in the SRE world

Metrics: hey, something is not right.

Traces: I can reveal the culprit for you.

Logs: Let me show you what happened.

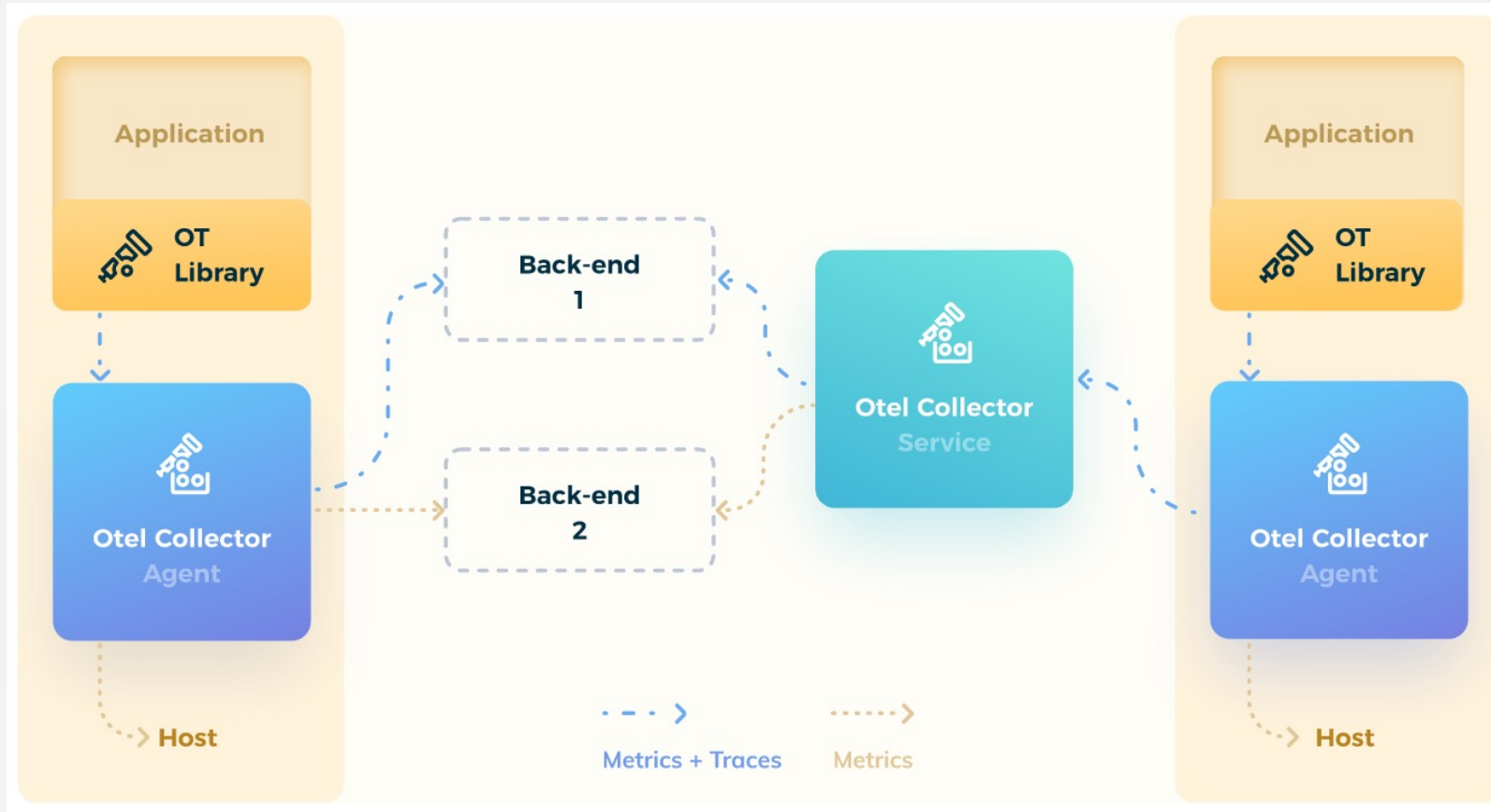
**Best way to
get started
with this?**



**High-quality, ubiquitous, and
portable telemetry to enable
effective observability.**

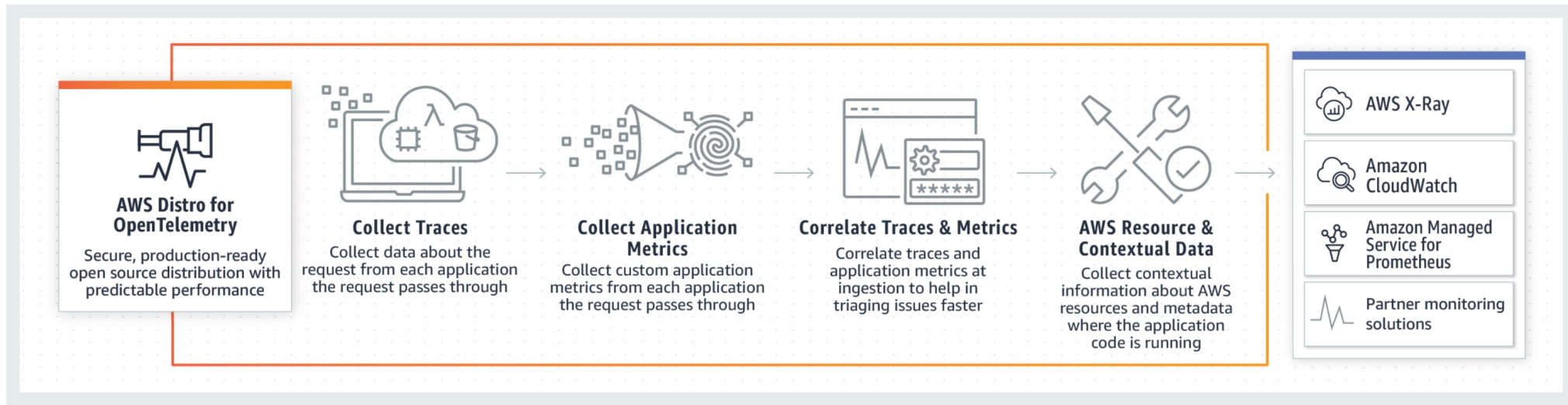
<https://opentelemetry.io>

OpenTelemetry: SDKs, protocols, and integration



AWS Distro for OpenTelemetry

How it works



<https://aws-otel.github.io/download>

Distributed tracing: the good parts 👍

Decrease time spent with Incident management

10%

- Monitoring metric values
- Catching up with alerts
- Evaluating trends and changes

60%

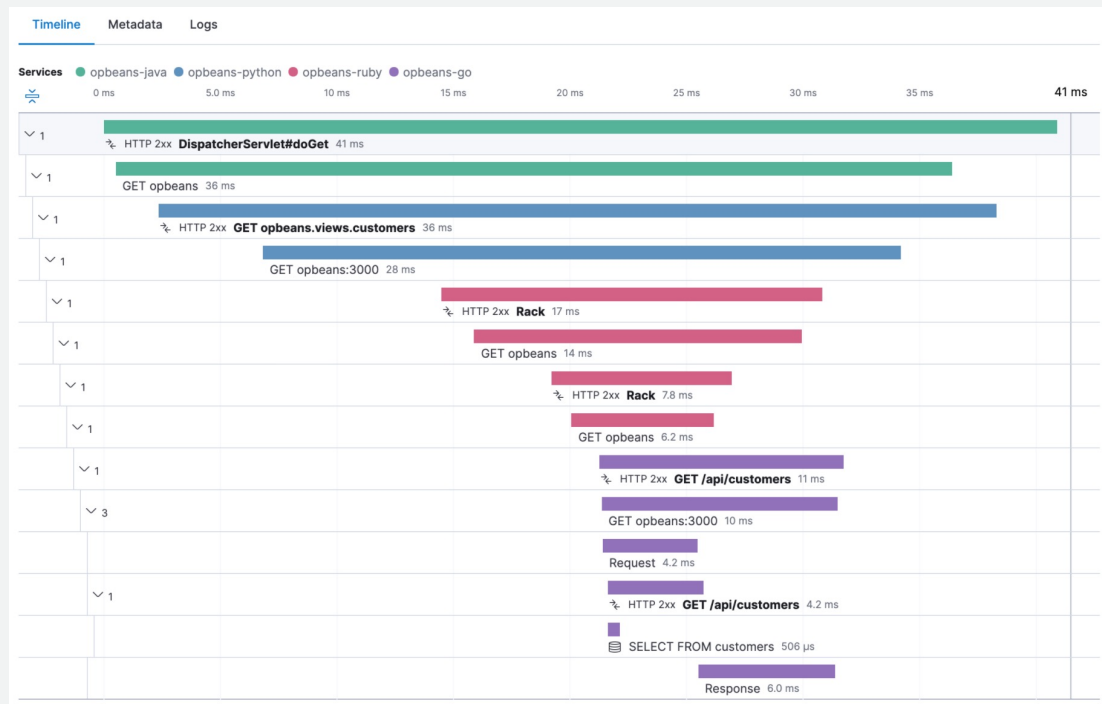
- Understanding topologies
- Isolating the anomalies
- Collecting context data

30%

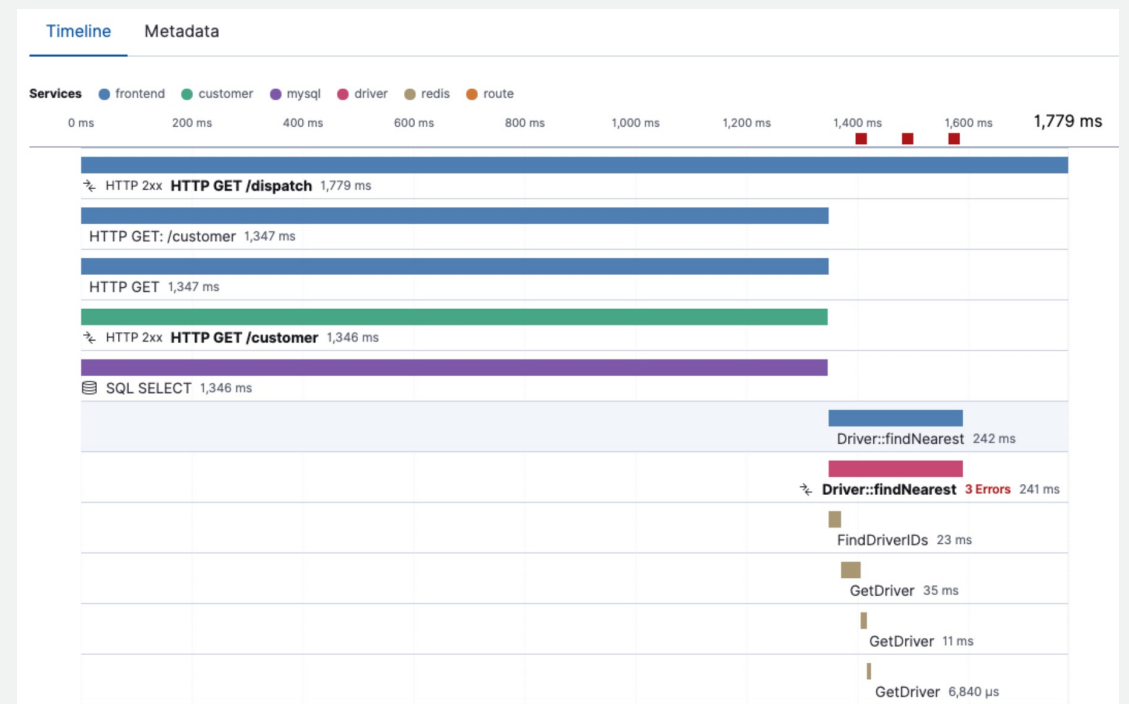
- Reading logs and events
- Reproducing exceptions
- Creating code patches

Finding bugs between releases

V1.12



V1.13



**Understand
what the code
(really) does.**

Distributed tracing: the bad parts 👎

**My language.
Your language.
Their language.**

**UI and frontends
give the vendors
goosebumps.**

**Asynchronous
programming has
its challenges.**

**To correlate, or
not correlate, that
is the question.**

**eBPF is the
future; but not
the present.**

Distributed tracing: the ugly stuff 🤮

A close-up shot of Jon Snow from the TV series Game of Thrones. He is looking off-camera with a serious expression. He has dark, curly hair and a light beard. He is wearing a dark, fur-lined cloak over a dark tunic. The background is a snowy, mountainous landscape with some wooden structures visible in the distance.

BRACE YOURSELVES

**Native binaries
may not come in
handy for you.**

**Black-box is
great; but not
for everybody.**

**Sampling is a
nightmare to
get it right.**

**Mixing open
standards with
proprietary stuff.**

**Be prepared
to become a
data engineer.**



Thank you!

Ricardo Ferreira

 @riferrei