



KubeCon



CloudNativeCon

North America 2017

Would You Like Some Tracing With Your Monitoring?

Yuri Shkuro, Software Engineer, *Uber Technologies*

In This Talk

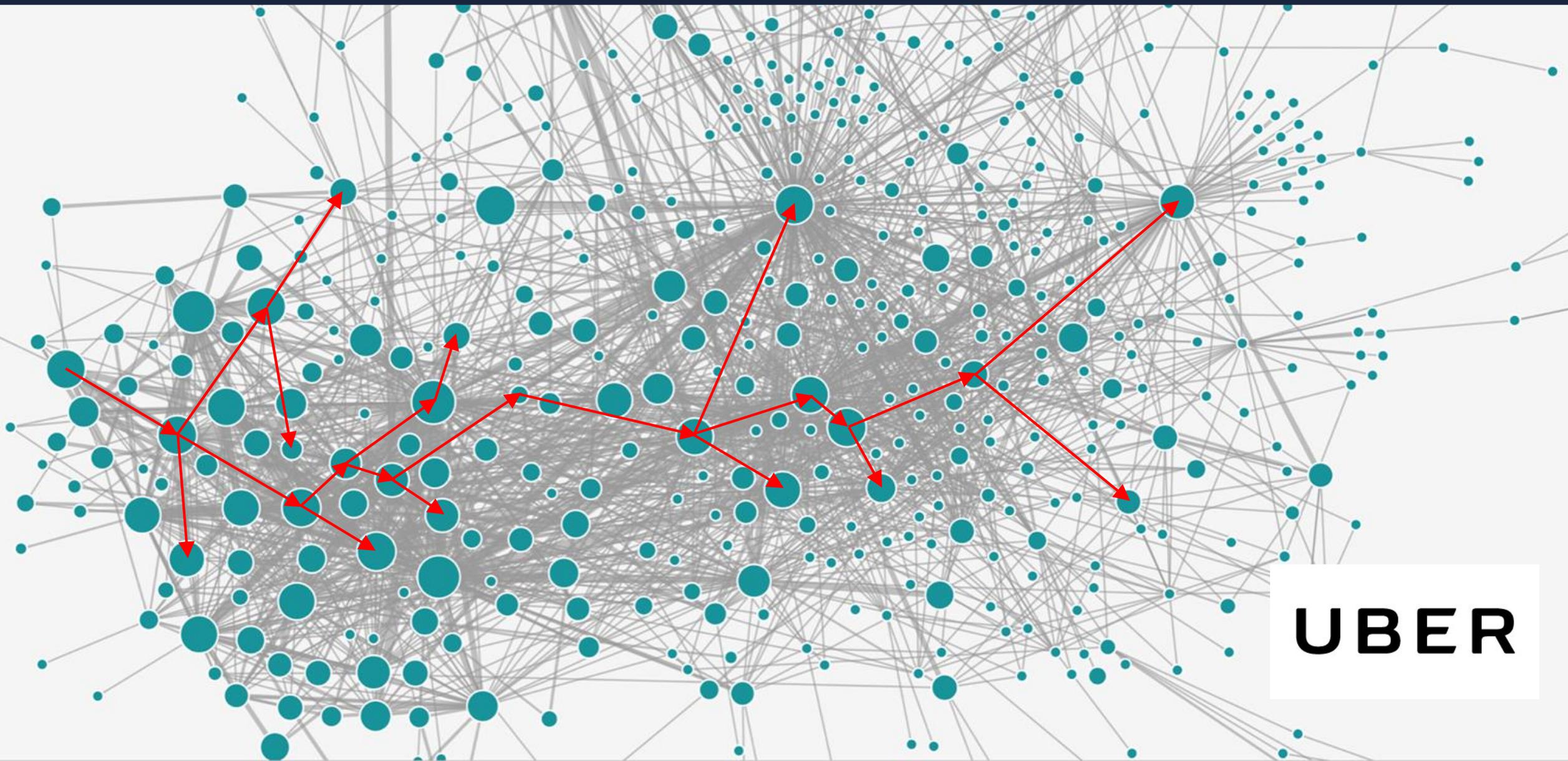
- Why should we care about tracing
- CNCF Jaeger & demo
- The Rollout Challenge
- Lessons Learned

About

- Engineer @ Uber NYC, Observability team
- Founder of Jaeger
- Co-founder of OpenTracing
- Github: yurishkuro
- Twitter: @yurishkuro



BILLIONS times a day!



UBER

How Do We Know What's Going On?

Metrics / Stats

- Counters, timers, gauges, histograms
- Four golden signals
- The USE method
- The RED method
- Statsd, Prometheus, Grafana

Logging

- Application events
- Errors, stack traces
- ELK, Splunk, Fluentd

Monitoring tools must “tell stories” about your system

What's The Story Here?

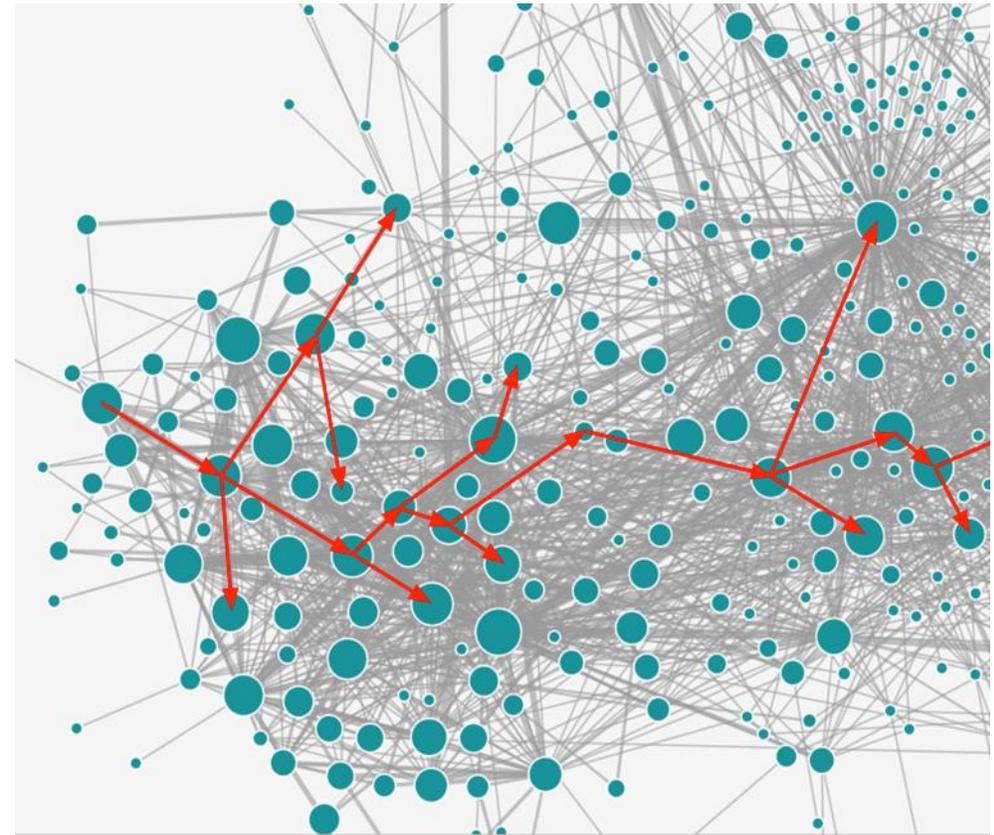
```
2017/12/04 21:30:37 scanning error: bufio.Scanner: token too long
```

Metrics and Logs Don't Cut It Anymore

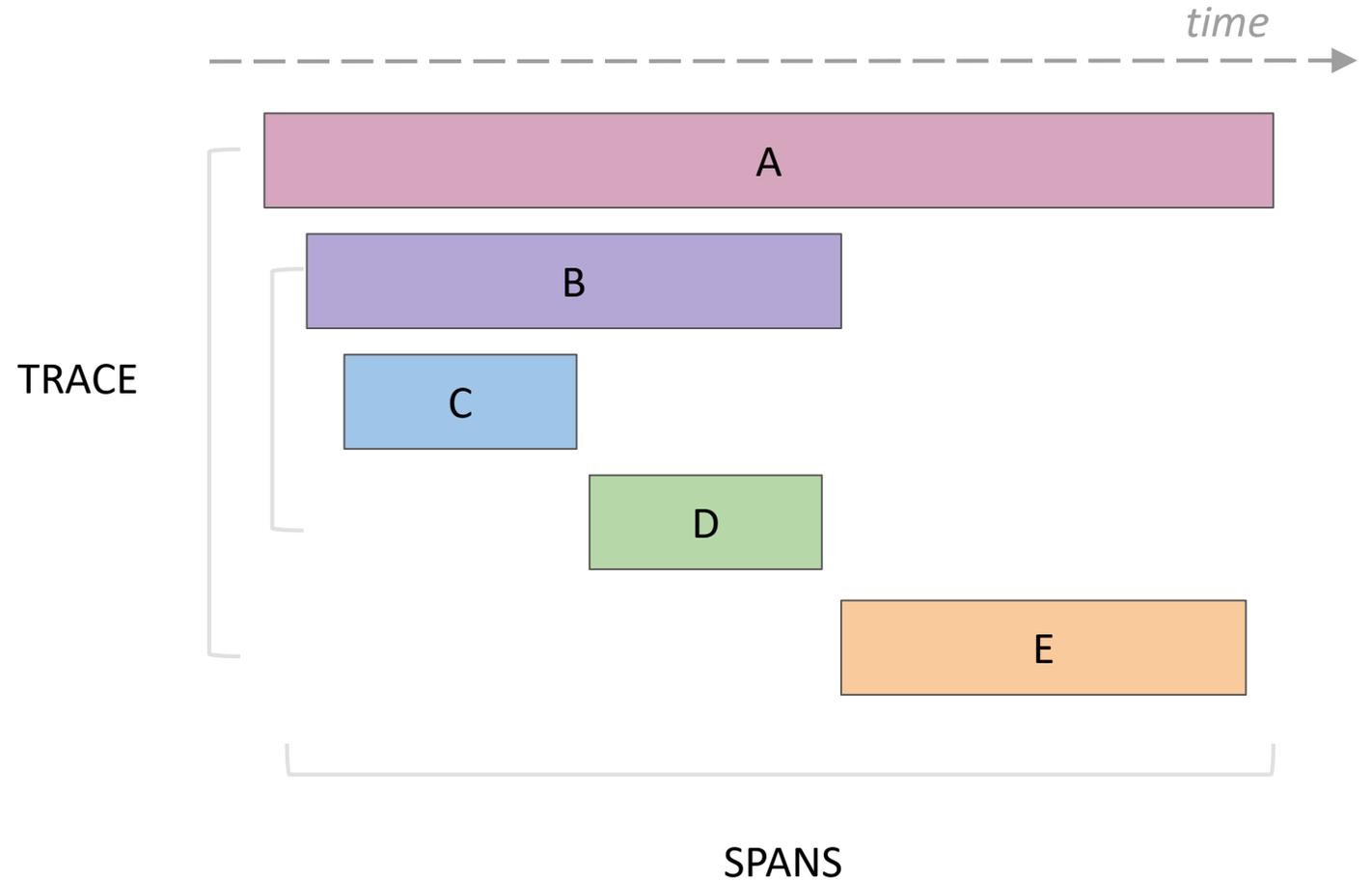
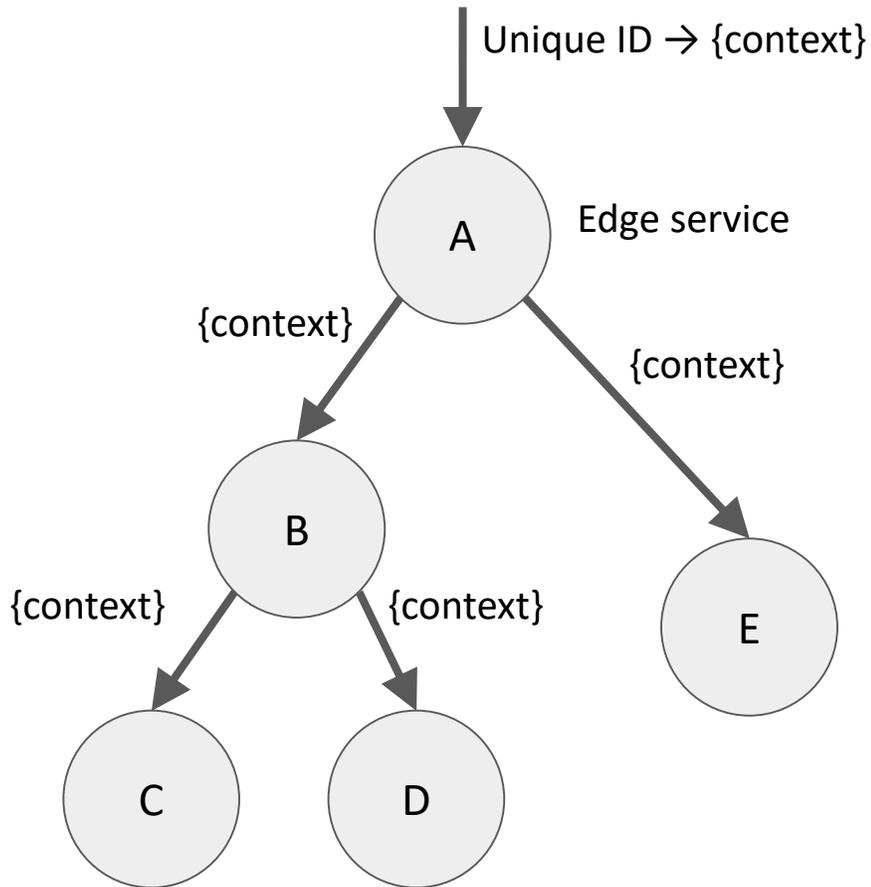
Metrics and logs are per-instance.

It's like debugging without stack traces.

We need to monitor
distributed transactions.



Context Propagation and Distributed Tracing



Let's look at some traces

- CNCF Jaeger, a distributed tracing system
- Created at Uber in Aug 2015
- Open sourced in Apr 2017
- <http://jaegertracing.io>
- Demo: <http://bit.do/jaeger-hotrod>



Distributed Tracing Supports:

distributed
transaction
monitoring

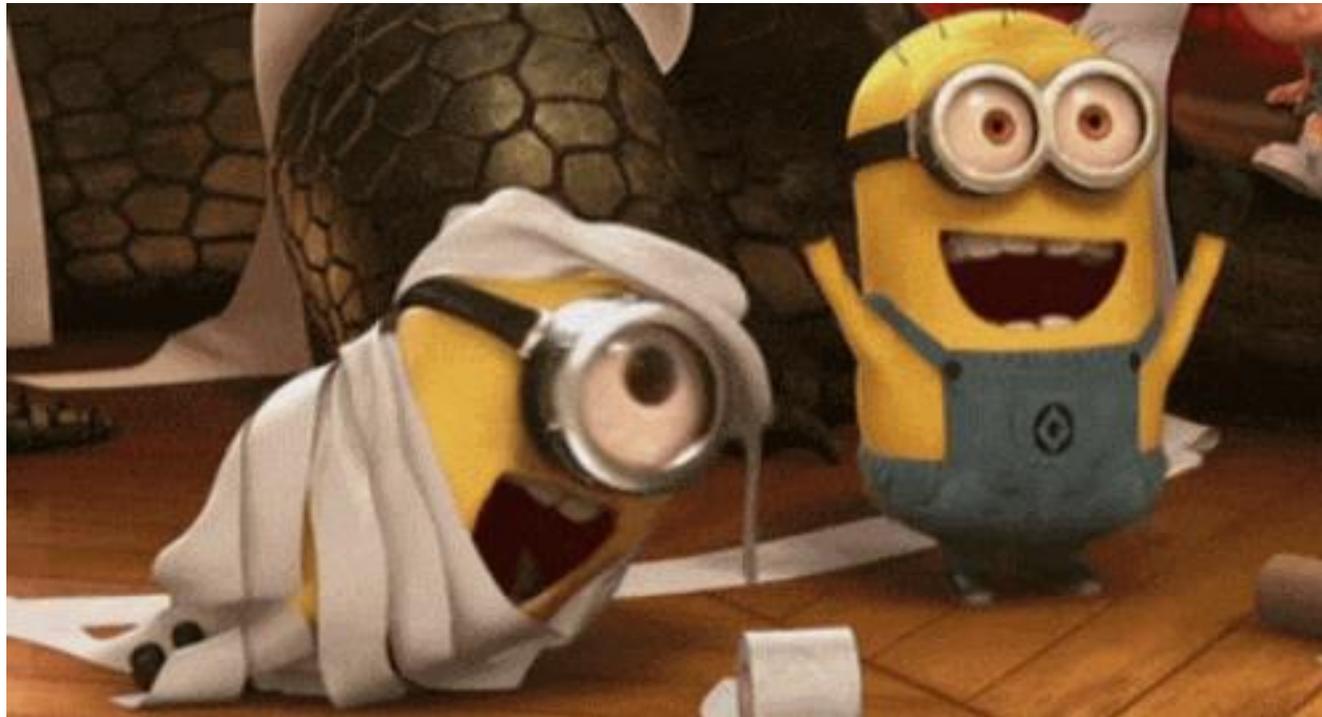
performance
and latency
optimization

root
cause
analysis

service
dependency
analysis

distributed context propagation

Who Thinks Tracing is Awesome?



Quick Poll

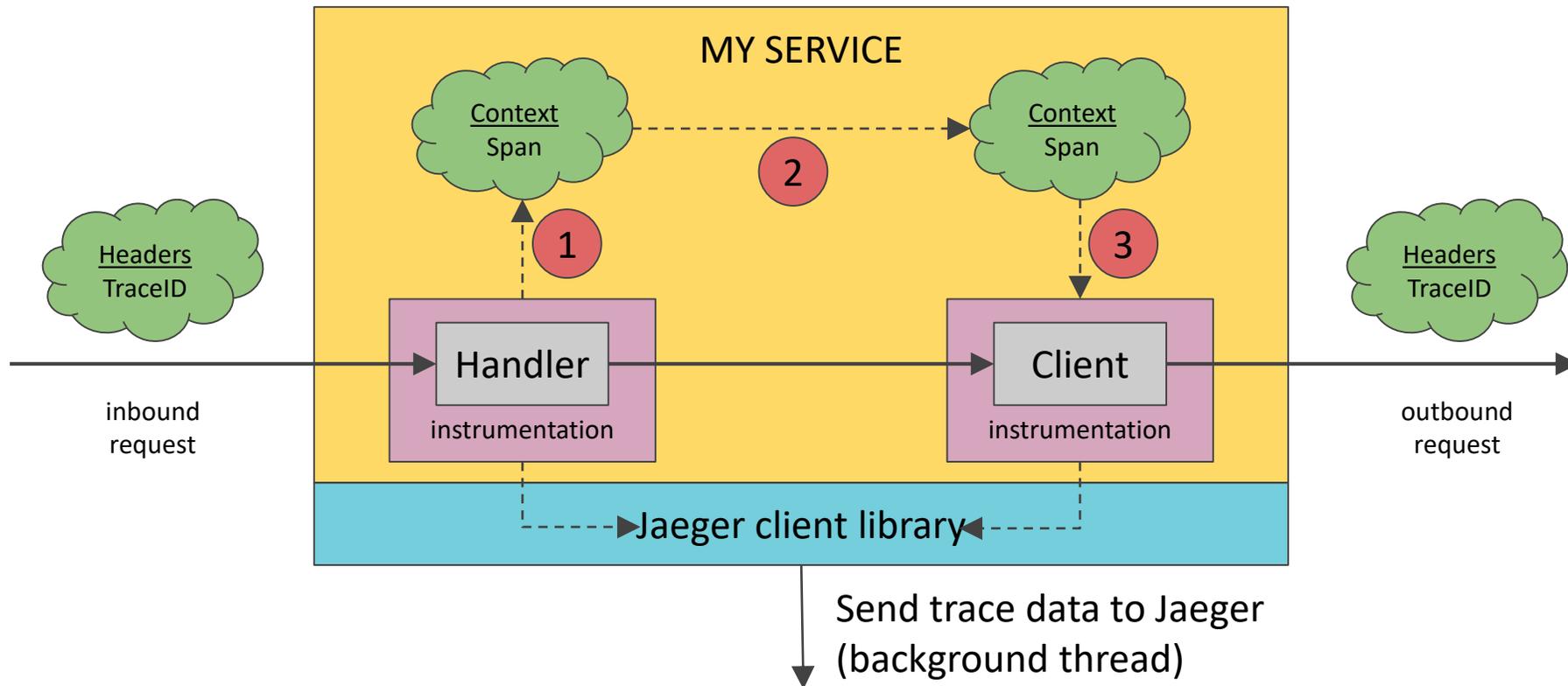
Does your company / organization use distributed tracing technology anywhere in their stack?

Why doesn't everyone do tracing?

Instrumentation has been
TOO HARD



Tracing Instrumentation



In-Process Context Propagation



Implicit, via thread-locals

Explicit



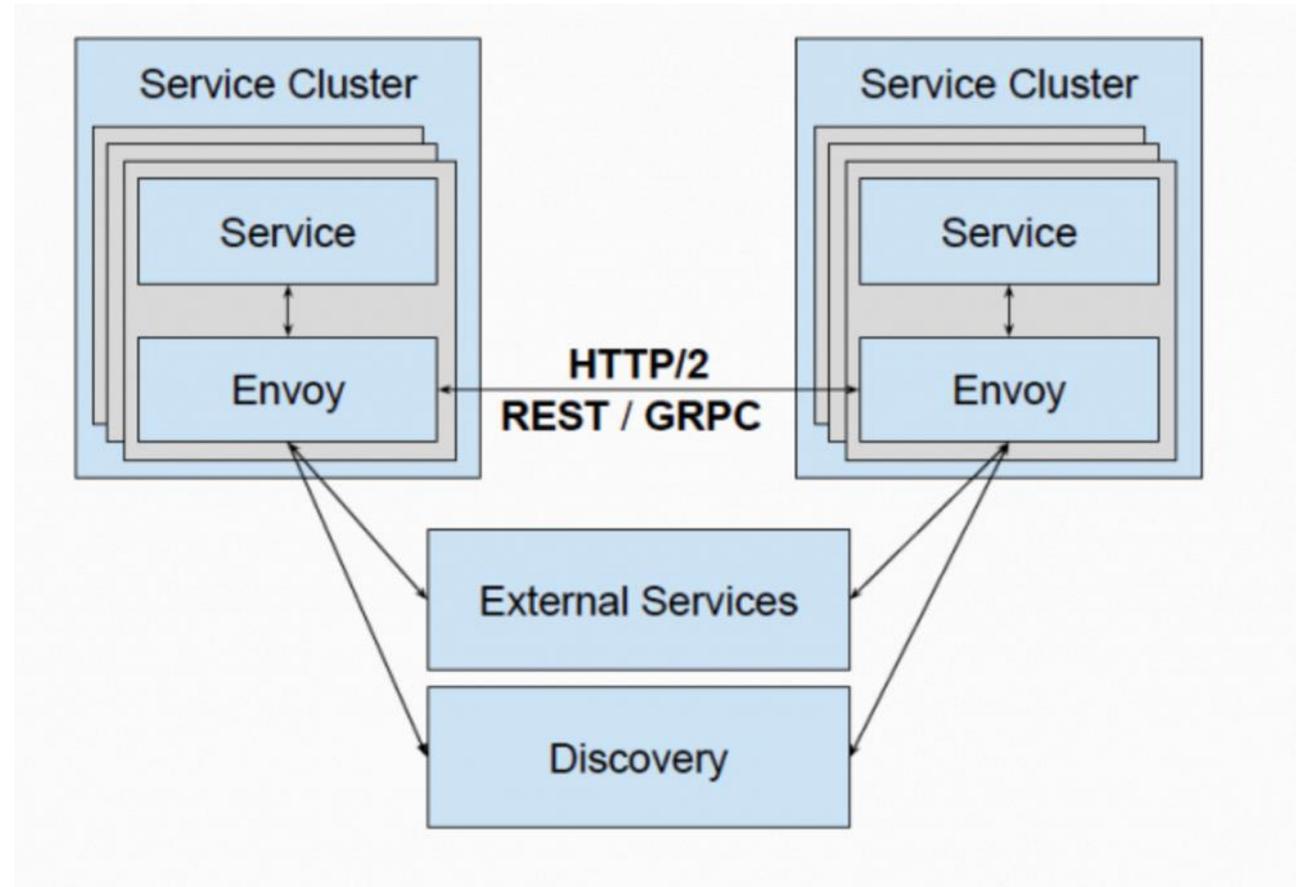
But: thread pools, futures, etc.

Zero-Touch Tracing Instrumentation?

- Fundamentally impossible in some languages
 - Otherwise not hard with explicitly passed Context
- Double-edge sword in languages with thread-locals
 - Easy in request-per-thread frameworks
 - Possible in async frameworks
 - Difficult with adhoc threading models

What About Service Meshes?

- Envoy, Linkerd, Istio
- Move RPC logic to a side car
 - Discovery, routing, health checking, load balancing, **monitoring (!!!)**
- To enable tracing, “just pass through this header”
- **It's the same in-process context propagation problem**



Lessons From Rolling Out Tracing

Out of ~3000 microservices,
about half are instrumented
for tracing



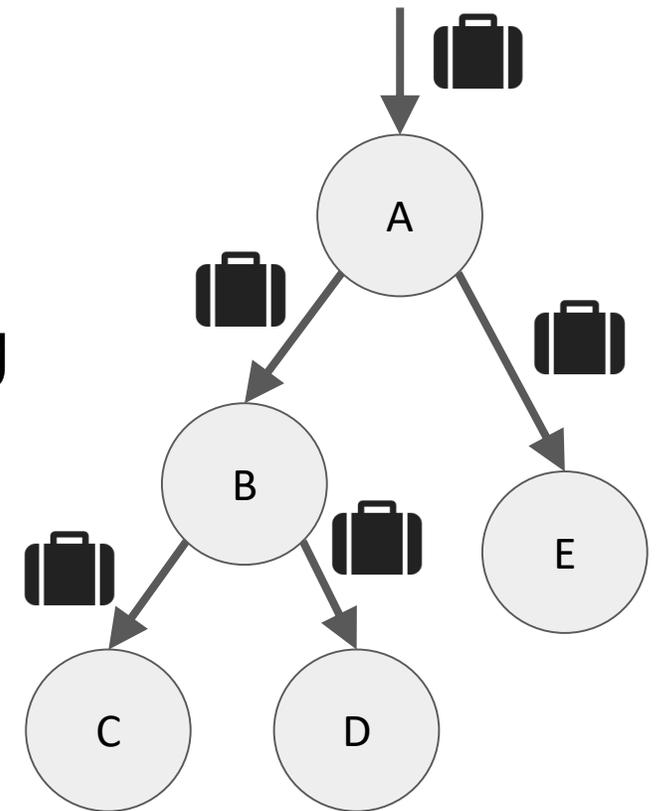
Aim for Zero-Touch Experience

- Use OpenTracing
- Instrument frequently used frameworks
- Many of them may be already instrumented with OpenTracing
- Enable tracing by default



Educate

- Distributed context propagation is still new to many people
- Context Propagation is Built-in in OpenTracing
- Baggage is a general purpose in-band key-value store
 - `span.SetBaggageItem("Bender", "Rodriguez")`



Context Propagation Use Cases

- Identifying synthetic traffic
 - Can use as a dimension for metrics
- Tenancy
 - E.g. at Google the top-level product (Docs, Gmail) is propagated
- Chaos engineering
 - Random killings must stop!

Measure Adoption and Quality

We show tracing quality metrics
as part of “service health” dashboards

Clear instructions how to improve



Trace Quality Metrics by Service

Click on pass or fail numbers to see example traces that exhibit that behavior

Metric	Pass %	Num Passes	Num Failures	Last Failure	Description
HasClientAddress	100	16	0		The server span emitted by this server had a good Client Address annotation saying where the request was coming from
HasClientAnnotations	100	128	0		The service emitted a client-side span with good client annotations
HasClientVersion	100	16	0		This service emitted a span that has a client version
HasSamplerType	100	16	0		This service initiated the trace and emitted the `sampler.type` annotation
HasServerAddress	0	0	128	6 hours ago	The client span emitted by this server had a good Server Address annotation saying where the request was going
HasServerAnnotations	100	16	0		The service emitted a server-side span with good server annotations
IsValidSamplerParam	100	16	0		This service initiated the trace and emitted a valid `sampler.Param` annotation
MeaningfulEndpointName	88	128	16	6 hours ago	The name of the endpoint being called had a meaningful name, e.g. not GET or POST
ParentSpanExists	100	128	0		The service (the Parent) that called this service emitted a span
TracedRootService	100	16	0		This service was the root service (i.e. it initiated the trace) and it correctly emitted a span
UniqueServerSpanID	100	16	0		Multiple server spans in this trace share the same span ID

Integrate With Other Tools

- Black box testing
 - External probes exercising the backend APIs
 - Low traffic allows 100% sampling
 - Incident reports include links to specific traces
- Developer Studio
 - Internal Web tool to simulate trip workflows
 - Makes a lot of API calls capturing all payloads
 - All requests are traces and traces are available in the same Web UI

Show Value

- Tracing is a product
- Engineers are your customers

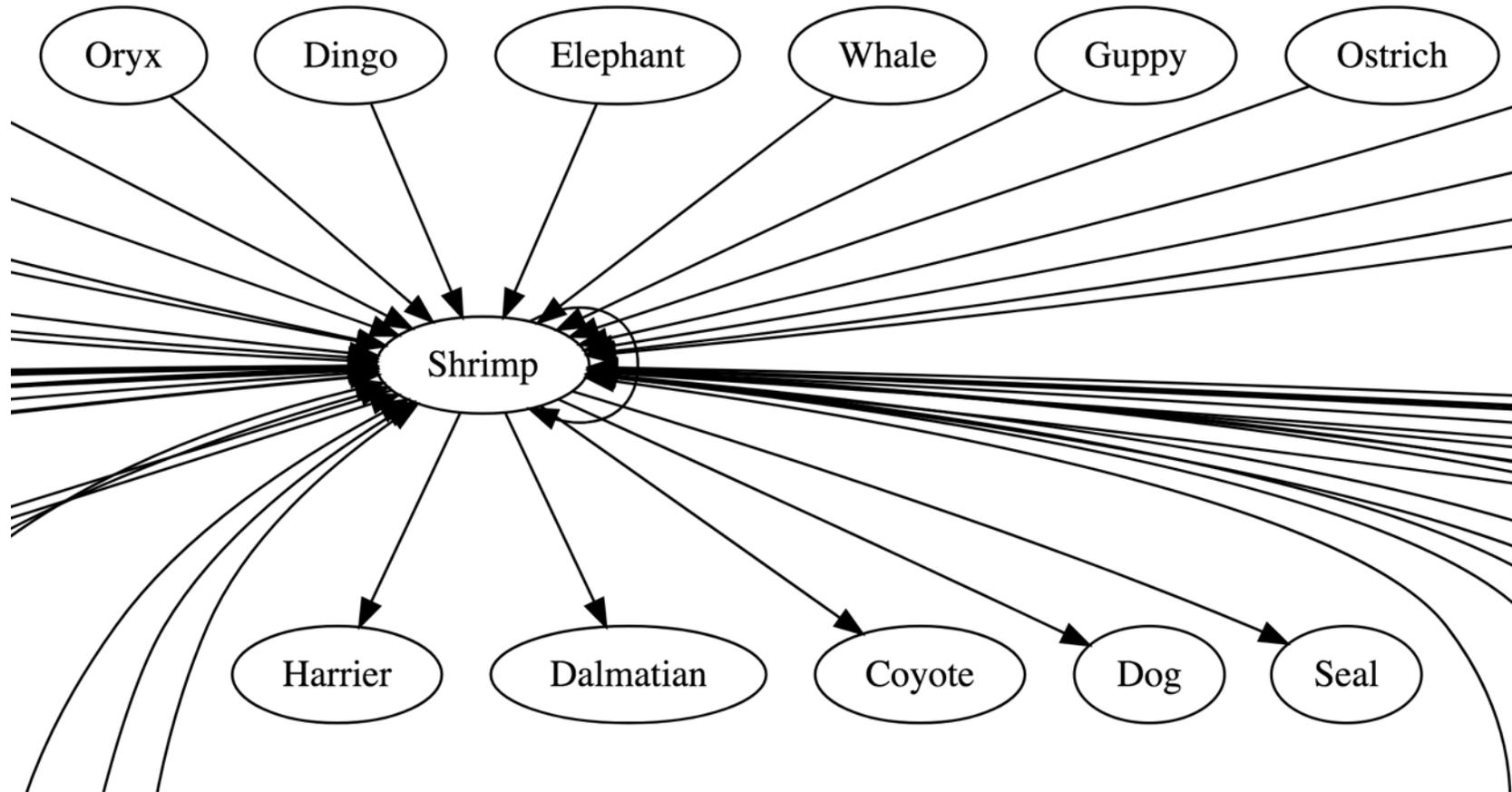


Service Dependency Analysis

- Who are my upstream and downstream dependencies?
- How many different workflows depend on my service?
- Is my service a critical (tier 1) service for core business flows?
- How do my SLIs affect other services?
- Will my service survive Halloween?

Tough questions when ~3000 microservices are working together

Does Dingo Depends on Dog?



From Firefighting to Fire Prevention

Use Distributed Tracing to

- Understand your system
- Optimize performance
- Increase efficiency
- Improve reliability



For More Information on Tracing

- SIG Jaeger Update, Thursday, December 7 • 11:10am - 11:45am
- SIG Jaeger Deep Dive, Thursday, December 7 • 2:00pm - 3:20pm
- OpenTracing Salon, Thursday, December 7 • 3:50pm - 4:50pm
- Jaeger Salon, Friday, December 8 • 2:00pm - 3:20pm

- Also don't miss the keynote by Ben Sigelman
 - Service Meshes and Observability
 - Wednesday, December 6 • 5:10pm - 5:30pm

Thank You

- Jaeger: <http://jaegertracing.io>
 - Twitter: <https://twitter.com/jaegertracing>
 - Gitter chat: <https://gitter.im/jaegertracing/>
 - Demo walkthrough: <http://bit.do/jaeger-hotrod>
- Contributors are welcome