

5 Reasons Why OpenTelemetry is the Future of Observability

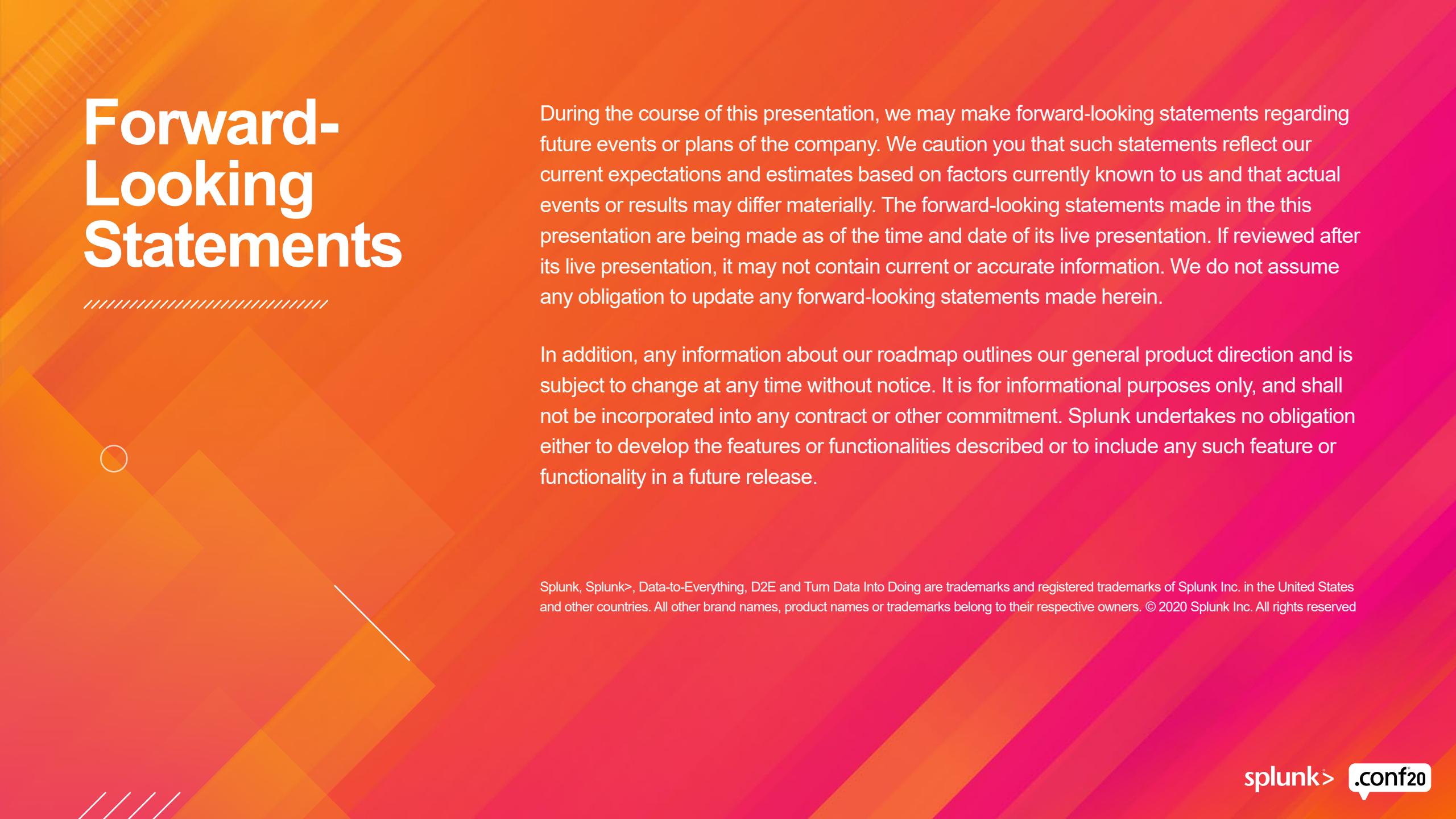
APP1246

Steve Flanders

Splunk



Forward-Looking Statements



During the course of this presentation, we may make forward-looking statements regarding future events or plans of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results may differ materially. The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, it may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements made herein.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only, and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionalities described or to include any such feature or functionality in a future release.

Splunk, Splunk>, Data-to-Everything, D2E and Turn Data Into Doing are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names or trademarks belong to their respective owners. © 2020 Splunk Inc. All rights reserved

Steve Flanders

Director of Engineering | Splunk Observability GDI



New Names, Same Great Technologies

VictorOps  is now **Splunk On-Call**

SignalFx
Infrastructure Monitoring  is now **Splunk Infrastructure Monitoring**

SignalFx
Microservices APM  is now **Splunk APM**

Agenda

1) What is OpenTelemetry?

2) 5 Reasons

3) Demo

What is OpenTelemetry?



OpenTelemetry makes robust, portable telemetry a built-in feature of cloud-native software.

OpenTelemetry provides a single set of APIs, libraries, agents, and collector services to capture distributed traces and metrics from your application. You can analyze them using Prometheus, Jaeger, and other observability tools.

OpenTelemetry Components

Specification

API: Baggage, tracing,
metrics

SDK: Tracing, metrics,
resource, configuration

Data: Semantic
conventions, protocol

Collector

Receive, process, and
export data

Default way to collect
from instrumented apps

Can be deployed as
an agent or service

Client Libraries

Application
instrumentation

Support for traces and
metrics today

Log support coming soon



Beta
March

GA
Autumn



**is the second most active
project in CNCF today!**



(per CNCF DevStats)

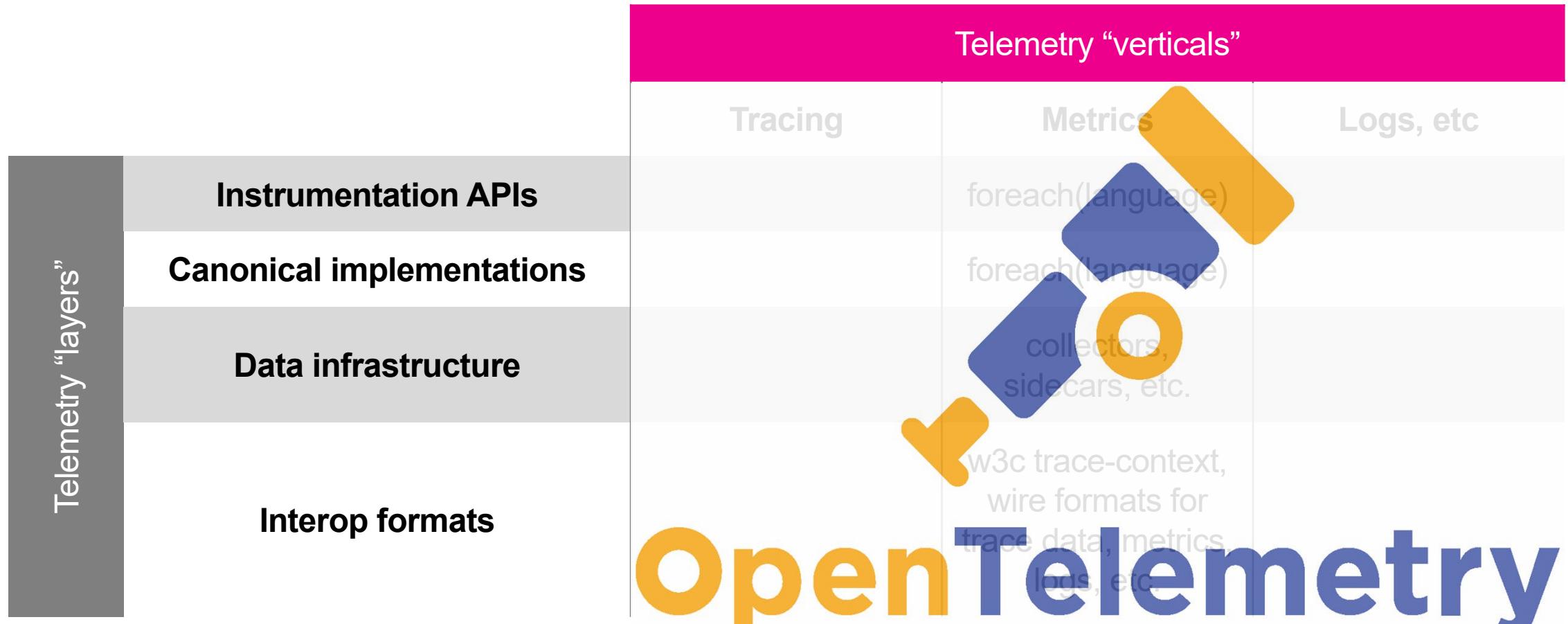


5 Reasons Why OpenTelemetry is the Future of Observability

1. Single Way to Capture Data

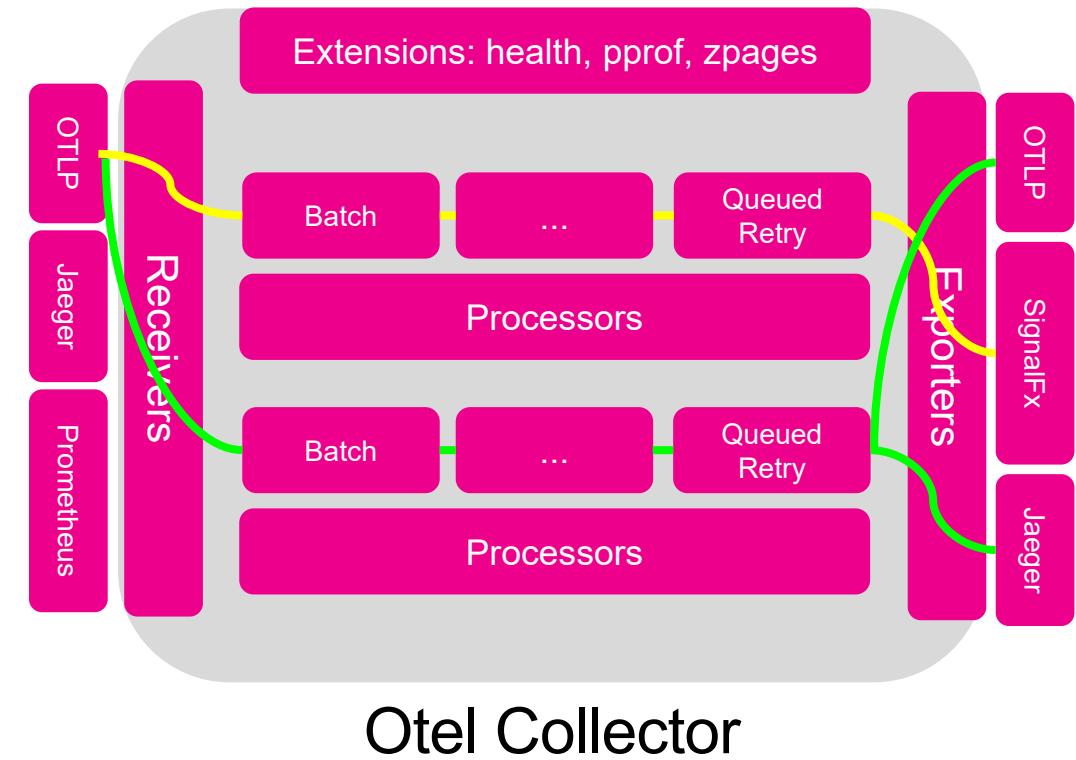
		Telemetry “verticals”		
		Tracing	Metrics	Logs, etc
Telemetry “layers”	Instrumentation APIs	foreach(language)		
	Canonical implementations	foreach(language)		
	Data infrastructure	collectors, sidecars, etc.		
	Interop formats	w3c trace-context, wire formats for trace data, metrics, logs, etc.		

1. Single Way to Capture Data



2. You Have Control of Your Data

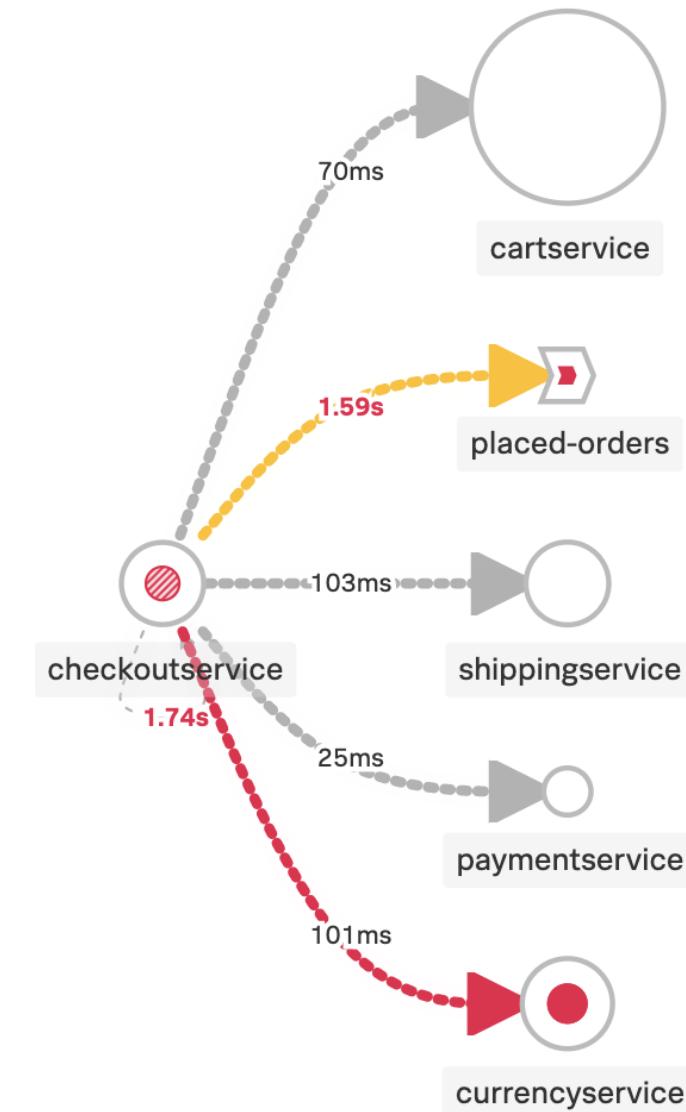
```
receivers:  
  otlp:  
    protocols:  
      grpc:  
      zipkin:  
processors:  
  batch:  
exporters:  
  sapm:  
    access_token: YOUR_TOKEN  
    endpoint: https://ingest.us0.signalfx.com/v2/trace  
service:  
  pipelines:  
    traces:  
      receivers: [otlp, zipkin]  
      processors: [batch]  
      exporters: [sapm]
```



Otel Collector

3. Context Reduces MTTR

- Tracing supports the W3C trace-context format as well as Zipkin's B3 headers (even both in parallel!)
- Correlation context makes it possible to share metadata regardless of data source
- Context is shared via Propagators

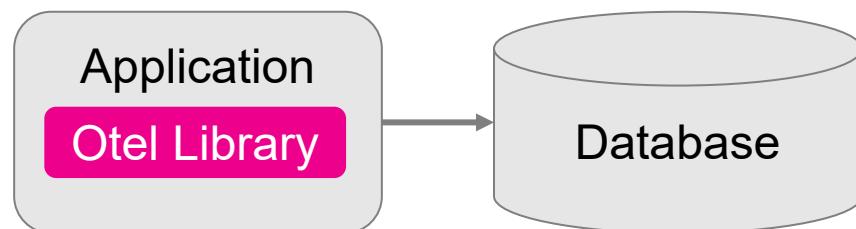


<https://github.com/open-telemetry/opentelemetry-specification/blob/master/specification/overview.md#spancontext>
<https://github.com/open-telemetry/opentelemetry-specification/blob/master/specification/context/context.md>

4. Open Standards = Data Portability

Semantic conventions normalize data.

- **HTTP:** http.method, http.status_code
- **Database:** db.instance, db.statement
- **Messaging:** messaging.system
- **FaaS:** faas.trigger



Resources capture environment metadata.

- **Application:** Service, Telemetry
- **Compute Unit:** Container, FaaS, Process
- **Compute Instance:** Host
- **Environment:** OS, Cloud, Deployment

region	...	us-west-1	us-west-1
k8s.io/pod/name	...	currencyervice-239jsdn5hl-	
k8s.io/cluster/name	...	us-west-1-cluster	

https://github.com/open-telemetry/opentelemetry-specification/blob/master/specification/trace/semantic_conventions/README.md
https://github.com/open-telemetry/opentelemetry-specification/tree/master/specification/resource/semantic_conventions

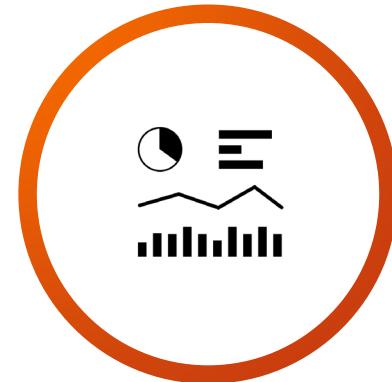
5. Everyone is Contributing and Adopting

Cloud Providers



AWS | Azure | GCP

Vendors



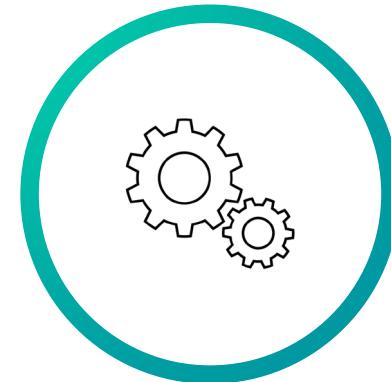
Every major vendor!
Splunk #1 contributor

End-users



Mailchimp (PHP)
Postmates (Erlang)
Shopify (Ruby)

Other

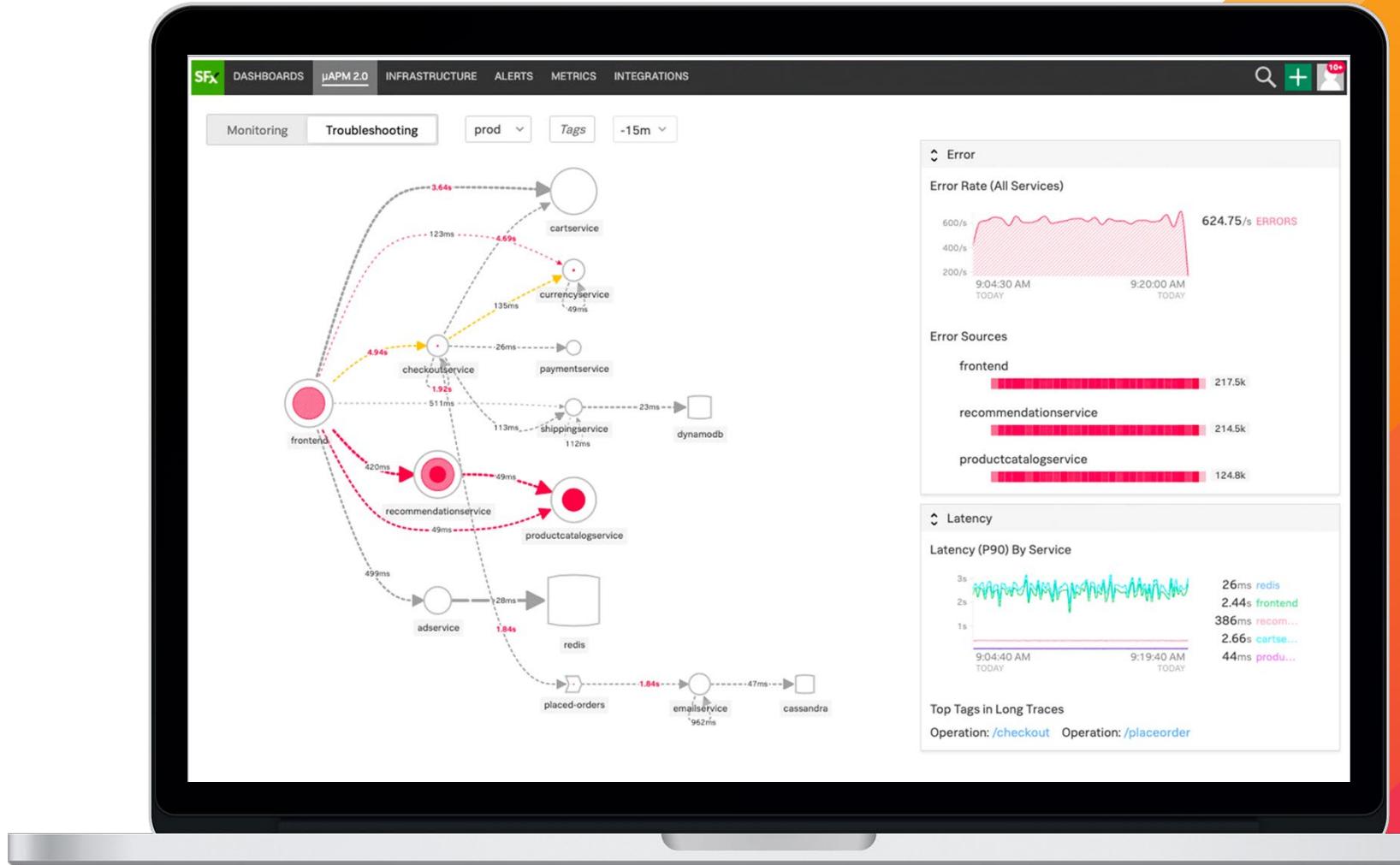


Jaeger > OtelCol
Fluent-bit <3 log SIG
Envoy roadmap
OpenMetrics roadmap
Spring roadmap

<https://github.com/open-telemetry/community/blob/master/ADOPTERS.md>
<https://medium.com/jaegertracing/jaeger-embraces-opentelemetry-collector-90a545cbc24>

Demo

OpenTelemetry



Join Us!

Conversation

Gitter: open-telemetry/
community

Gitter: open-telemetry/
getting-started

SIG

GitHub: open-telemetry/
community

#special-interest-groups
#related-groups

Submit a PR

consider
good-first-issue and
help-wanted labels



Thank You

Please provide feedback via the
SESSION SURVEY

