

OpenTelemetry and Observability: What, Why, and Why Now?

SREconAPAC 2022
7 December 2022

Greg Leffler
splunk> turn data into doing®



Forward-Looking Statements



This presentation may contain forward-looking statements regarding future events, plans or the expected financial performance of our company, including our expectations regarding our products, technology, strategy, customers, markets, acquisitions and investments. These statements reflect management's current expectations, estimates and assumptions based on the information currently available to us. These forward-looking statements are not guarantees of future performance and involve significant risks, uncertainties and other factors that may cause our actual results, performance or achievements to be materially different from results, performance or achievements expressed or implied by the forward-looking statements contained in this presentation.

For additional information about factors that could cause actual results to differ materially from those described in the forward-looking statements made in this presentation, please refer to our periodic reports and other filings with the SEC, including the risk factors identified in our most recent quarterly reports on Form 10-Q and annual reports on Form 10-K, copies of which may be obtained by visiting the Splunk Investor Relations website at www.investors.splunk.com or the SEC's website at www.sec.gov. The forward-looking statements made in this presentation are made as of the time and date of this presentation. If reviewed after the initial presentation, even if made available by us, on our website or otherwise, it may not contain current or accurate information. We disclaim any obligation to update or revise any forward-looking statement based on new information, future events or otherwise, except as required by applicable law.

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. We undertake no obligation either to develop the features or functionalities described, in beta or in preview (used interchangeably), or to include any such feature or functionality in a future release.

Splunk, Splunk® 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. © 2022 Splunk Inc. All rights reserved.



Greg Leffler

Observability Practitioner @ Splunk

Former:

Sysadmin @ eBay

SRE @ LinkedIn



What is Observability?

Yesterday it worked
Today it is not working
The web is like that

splunk> turn data into doing®

What is Observability?

Depends on who you ask, but . . .

“Metrics, traces, and logs”

“The ability to infer the state of a system by examining its output”

“A great new thing to spend money on”

The real definition:

**Observability is a way to investigate
unknown unknowns by instrumenting everything**

What can you do with it?

Metrics: **Is** there a problem?

Traces: **Where** is the problem?

Logs: **What** caused the problem?

Common tools: Infrastructure monitoring, Application Performance Monitoring, Digital Experience Monitoring (RUM/Synthetics)

Across multiple envs/cloud providers/etc.



Solutions to the Telemetry Problem

Someone without eyes
Cannot see the dense forest
Or anything else

splunk> turn data into doing®

The Telemetry Problem

What were things like before OpenTelemetry/What are they like now?

Multiple tools each with their own special instrumentation process

Each new tool required you to do work on every service

The instrumentation was not compatible across vendors/products

Missing context when solving problems because some services may not be instrumented with every tool

Toil, toil, and more toil!

Solutions to the Telemetry problem

Proprietary agents (“magic”)

OpenTracing (CNCF)

API for sending data to an O11y back-end

OpenCensus (Google)

Language-specific libraries to instrument code

OpenTracing + OpenCensus = OpenTelemetry



What is OpenTelemetry?

Out of a tumult
Arises a clear victor
Both free and open

splunk> turn data into doing®

What is OpenTelemetry?

A framework for collecting observability-related telemetry data

Ships a single collector binary

Vendor-agnostic instrumentation per language

Open-standard semantic conventions and wire format (OTLP)

eBPF networking agent for network telemetry

NOT an observability back-end itself (e.g. not Jaeger, Prometheus)

OT data pipelines



pipestorm.splunk.com



Receivers



Processors



Exporters



Why OpenTelemetry?

Many hands helping
Make the fall harvest faster
And your skills grow too

splunk> turn data into doing®

Data Ownership

Regulatory/compliance requirements

Business flexibility/continuity requirements

Data transferability

Skill set transferability

Data Ownership is a fundamental right.

CNCF Support

Second-busiest CNCF project (52 GH events/hour on average)

Behind Kubernetes

~800 unique contributors per month (representing 150 organizations), thousands of contributions per week

And various other numbers as well...



Why should I care?

The SRE life
Revolve around toil
Decrease the amount

splunk> turn data into doing®

As an SRE, why do I care?

Instrumentation will always require work

- Doing this work is toil
- Do it once!

Open standards win in the end

OpenTelemetry is *the* industry standard (both across vendors and open-source)

OpenTelemetry is still growing and has tons of opportunity for you to participate



What next?

Useful resources

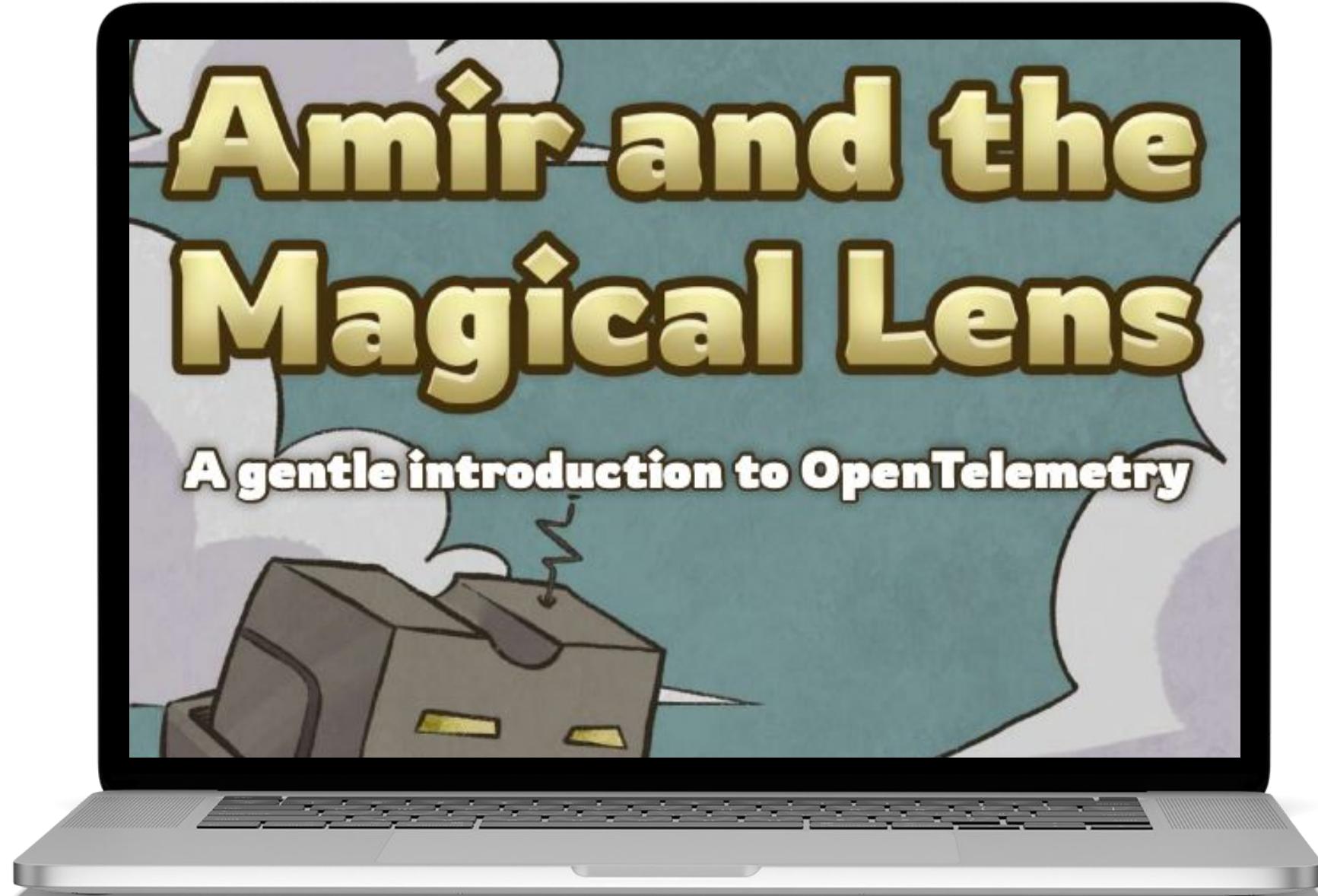
It's not DNS

There's no way it's DNS

It was DNS

splunk> turn data into doing®

Amir and the Magical Lens



<https://splk.it/AmirAndTheMagicalLens>

splunk > turn data into doing

OpenTelemetry Project Resources

<https://opentelemetry.io>

[2022 Project Update](#)



C++



A language-specific implementation of OpenTelemetry in C++.

.NET



A language-specific implementation of OpenTelemetry in .NET.

Erlang/Elixir



A language-specific implementation of OpenTelemetry in Erlang/Elixir.

Go



A language-specific implementation of OpenTelemetry in Go.

Java



A language-specific implementation of OpenTelemetry in Java.

JavaScript



A language-specific implementation of OpenTelemetry in JavaScript (for Node.js & the browser).

PHP



A language-specific implementation of OpenTelemetry in PHP.

Python



A language-specific implementation of OpenTelemetry in Python.

Ruby



A language-specific implementation of OpenTelemetry in Ruby.

Rust



A language-specific implementation of OpenTelemetry in Rust.

Swift



A language-specific implementation of OpenTelemetry in Swift.

Burning questions?

#22apac-day1-track2 on
SRECon Slack

<https://linkedin.com/in/gleffler>



Thank You!

