OpenTelemetry on Mainframe SIG

OpenTelemetry

OPEN
MAINFRAME
PROJECT

Ruediger Schulze
IBM STSM, Observability

We enable OpenTelemetry on the Mainframe for an improved end-to-end observability.

Agenda

01 What is OpenTelemetry?

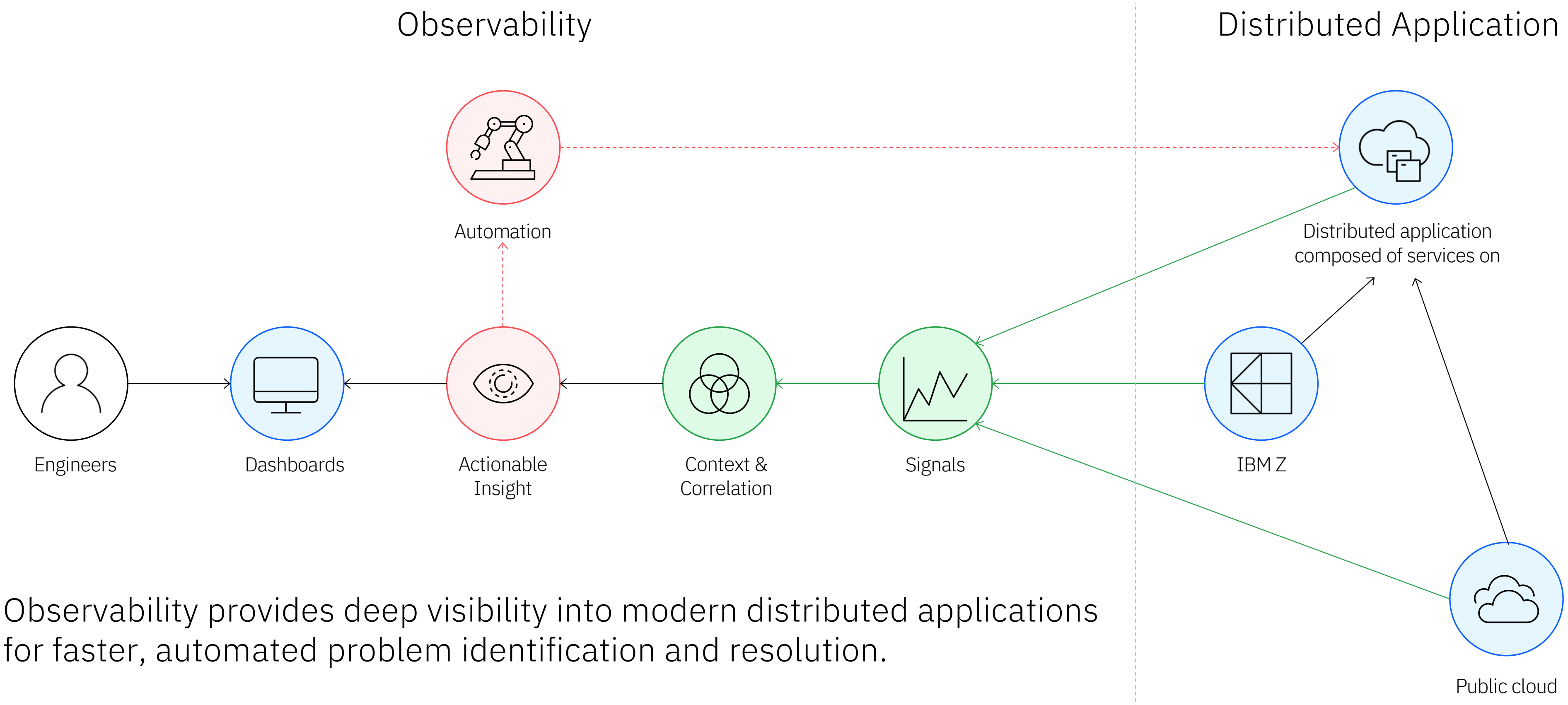
02 Why is OpenTelemetry important?

03 What is the OpenTelemetry on Mainframe SIG?

04 How to engage?

01 What is OpenTelemetry?

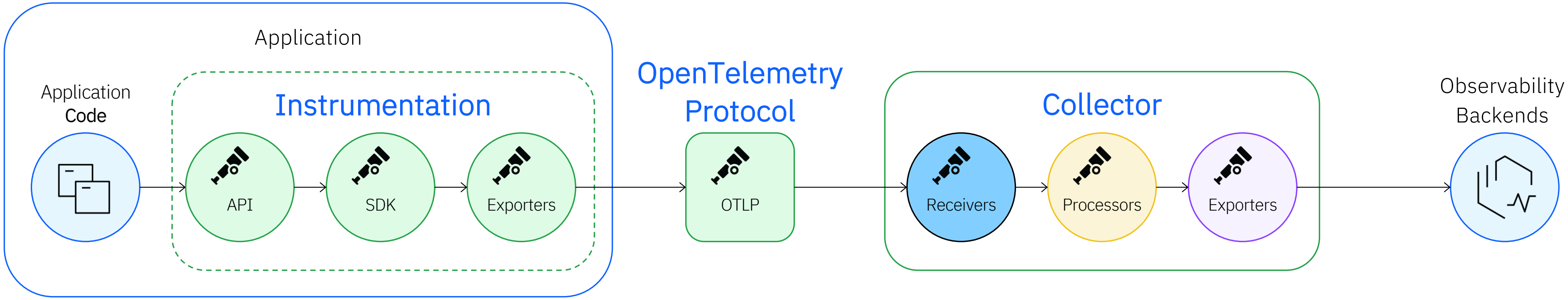
Observability recap



Observability provides deep visibility into modern distributed applications for faster, automated problem identification and resolution.

Source: IBM – What is observability?

What is OpenTelemetry?



OpenTelemetry is a vendor-agnostic observability framework that assists in generating, processing, and distributing telemetry data such as metrics, traces, and logs.

OpenTelemetry's Mission: to enable effective observability by making high-quality, portable telemetry ubiquitous.

Use the OpenTelemetry Operator for the deployment and management of the OpenTelemetry Collector in Kubernetes, together with auto-instrumentation.

Source: opentelemetry.io

OpenTelemetry Instrumentation

OpenTelemetry code instrumentation is supported for several languages.

Depending on the language, support includes:

- Manual instrumentation
- Automatic instrumentation

On Kubernetes, use the OpenTelemetry Operator for Kubernetes to inject auto-instrumentation libraries for Java, Node.js, Python, .NET or GO into your application

.NET

C++

Erlang/Elixir

Go

Java

PHP

JavaScript

Ruby

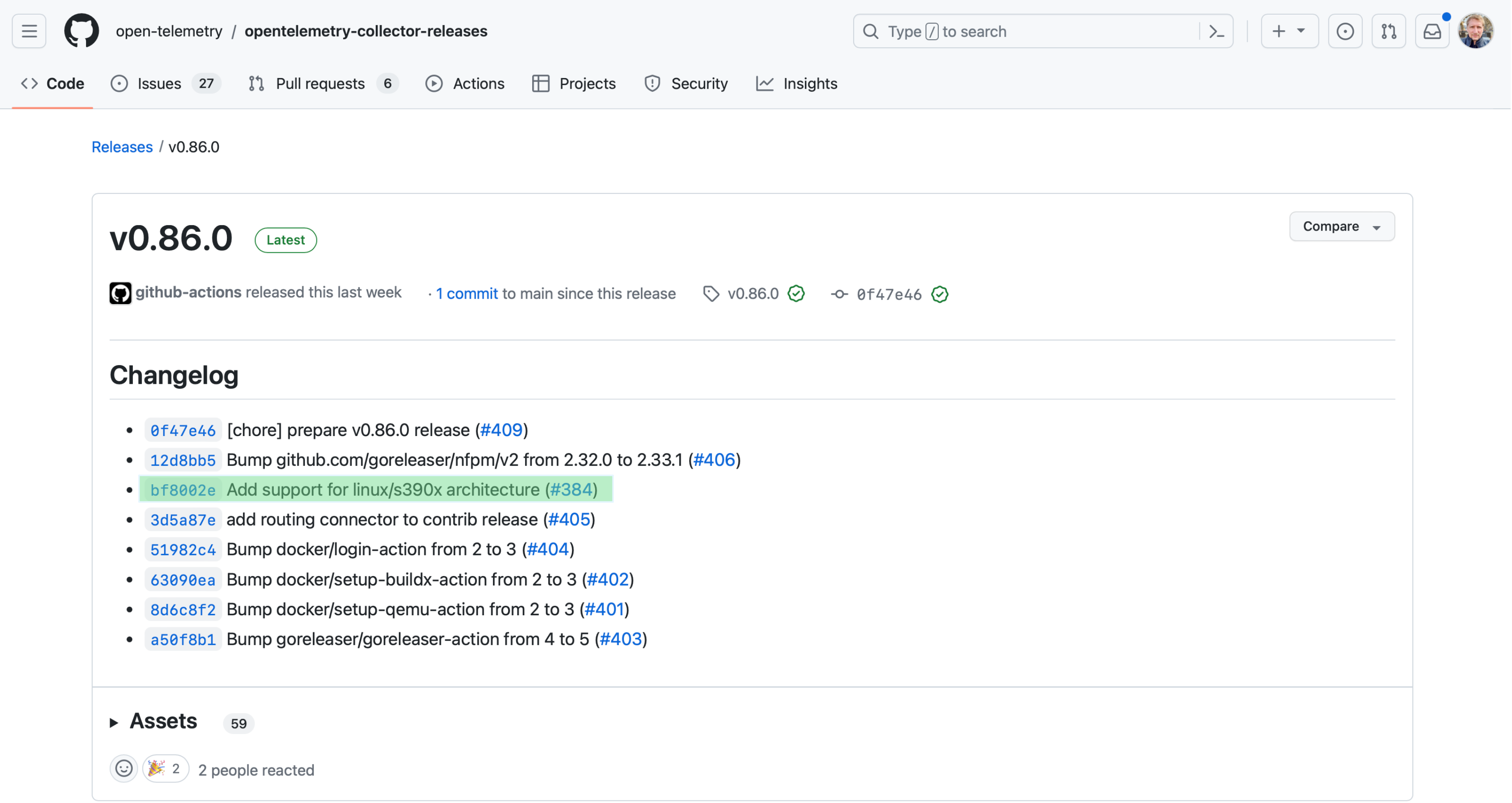
Python

Rust

Swift

Source: https://opentelemetry.io/docs/instrumentation/

OpenTelemetry Collector now available for linux/s390x



Releases / v0.86.0

v0.86.0

github-actions released this last week · 1 commit to main since this release v0.86.0 0f47e46

Changelog

- 0f47e46 [chore] prepare v0.86.0 release (#409)
- 12d8bb5 Bump github.com/goreleaser/nfpm/v2 from 2.32.0 to 2.33.1 (#406)
- bf8002e Add support for linux/s390x architecture (#384)
- 3d5a87e add routing connector to contrib release (#405)
- 51982c4 Bump docker/login-action from 2 to 3 (#404)
- 63090ea Bump docker/setup-buildx-action from 2 to 3 (#402)
- 8d6c8f2 Bump docker/setup-qemu-action from 2 to 3 (#401)
- a50f8b1 Bump goreleaser/goreleaser-action from 4 to 5 (#403)

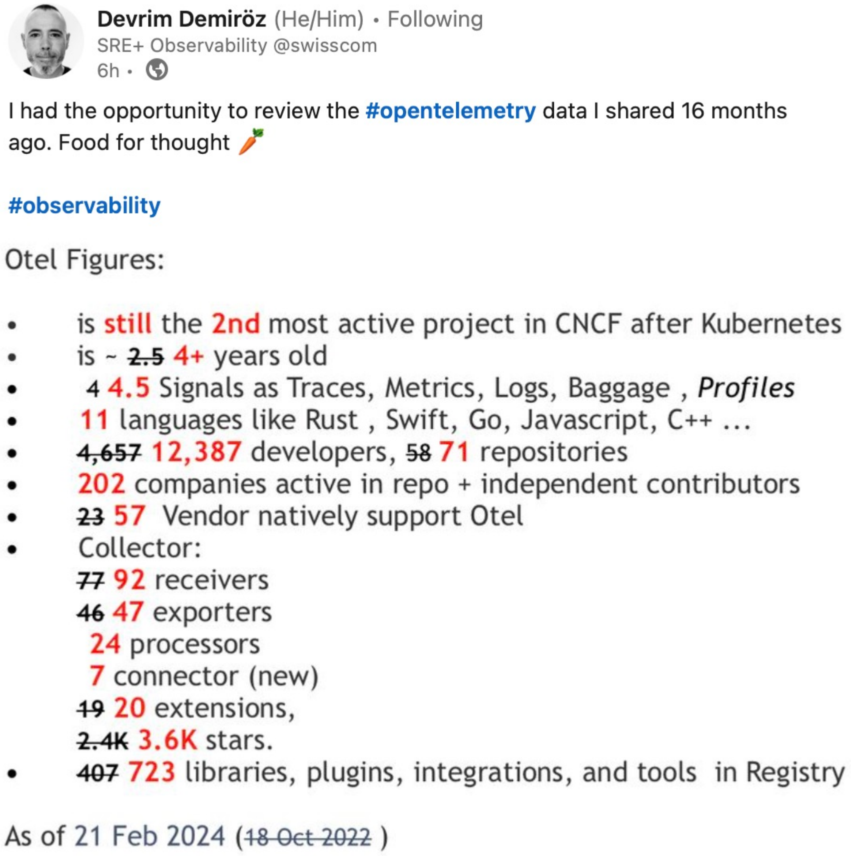
▸ Assets 59

🎉 2 2 people reacted

Source: OpenTelemetry Collector Releases

02 Why is OpenTelemetry important?

Why OpenTelemetry is important?



Devrim Demiröz (He/Him) • Following
SRE+ Observability @swisscom
6h •

I had the opportunity to review the **#opentelemetry** data I shared 16 months ago. Food for thought 🥕

#observability

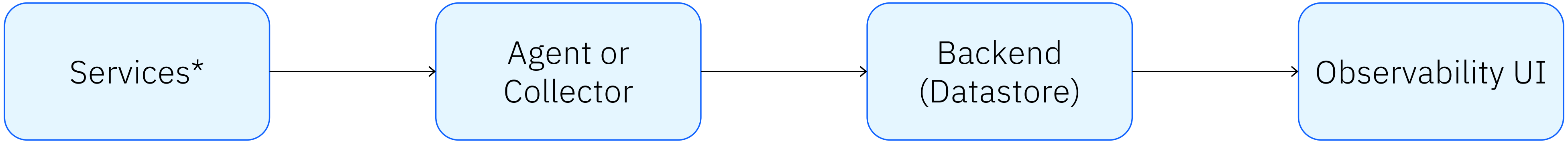
Otel Figures:

- is **still** the **2nd** most active project in CNCF after Kubernetes
- is ~ ~~2.5~~ 4+ years old
- 4 4.5 Signals as Traces, Metrics, Logs, Baggage , *Profiles*
- 11 languages like Rust , Swift, Go, Javascript, C++ ...
- ~~4,657~~ 12,387 developers, ~~58~~ 71 repositories
- 202 companies active in repo + independent contributors
- ~~23~~ 57 Vendor natively support Otel
- Collector:
  ~~77~~ 92 receivers
  ~~46~~ 47 exporters
  24 processors
  7 connector (new)
  ~~19~~ 20 extensions,
  ~~2.4K~~ 3.6K stars.
- ~~407~~ 723 libraries, plugins, integrations, and tools in Registry

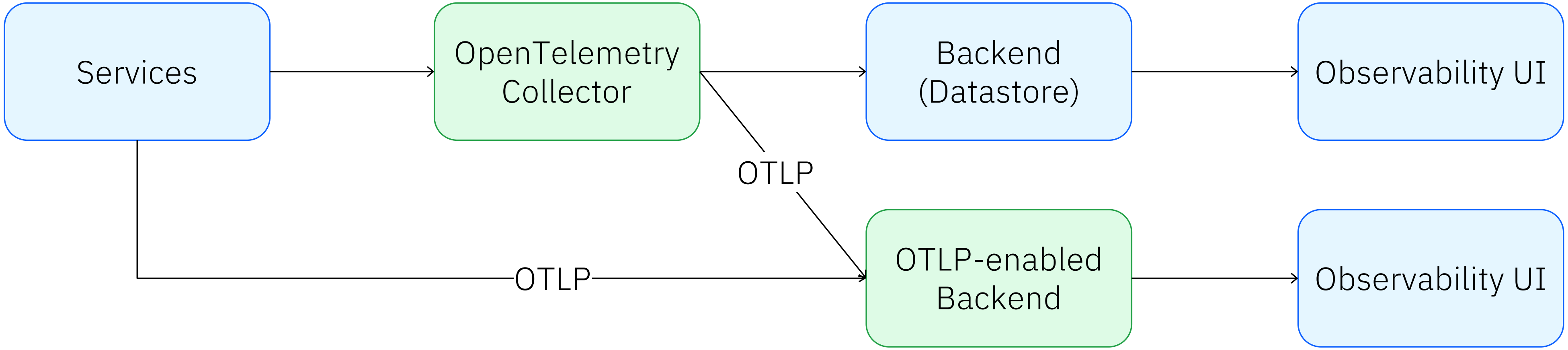
As of 21 Feb 2024 (~~18 Oct 2022~~)

Source: [LinkedIn post from 2/22/2024](#)

Classical observability pipeline



Observability pipeline with OpenTelemetry



Services

OpenTelemetry Collector

Backend (Datastore)

Observability UI

OTLP

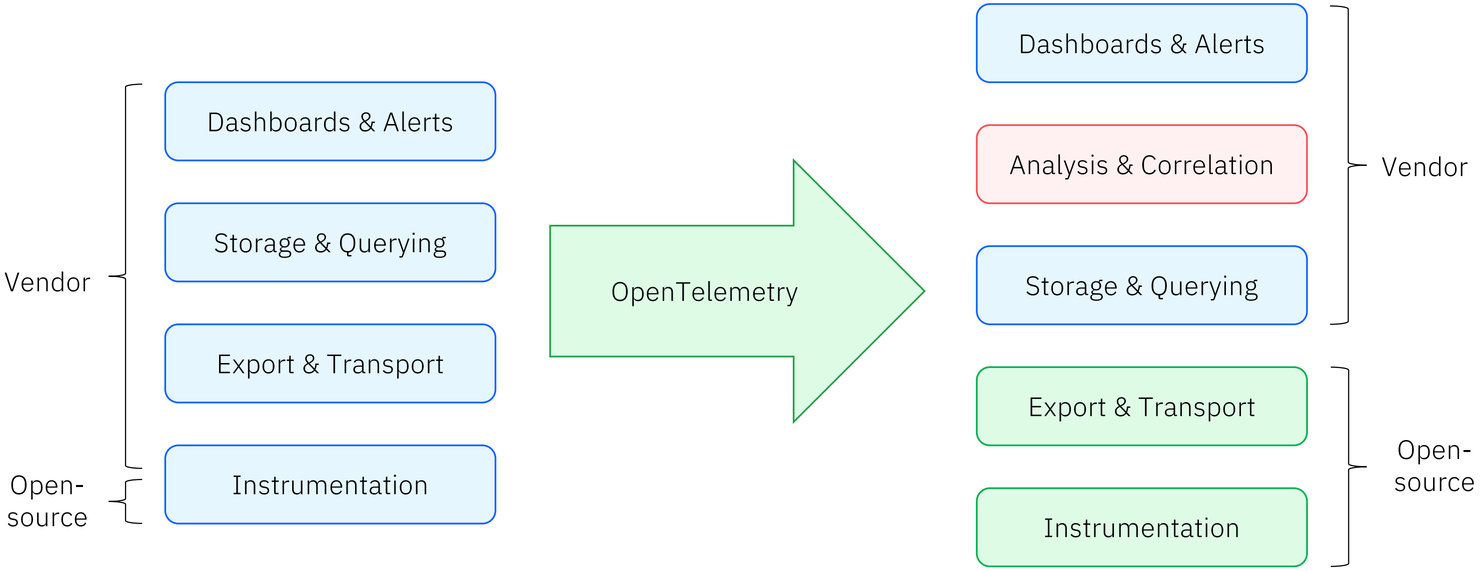
OTLP-enabled Backend

OTLP

Observability UI

The impact of OpenTelemetry

- Shift in vendor vs. open-source boundaries
- Observability products must achieve differentiation through advanced analysis and correlation capabilities



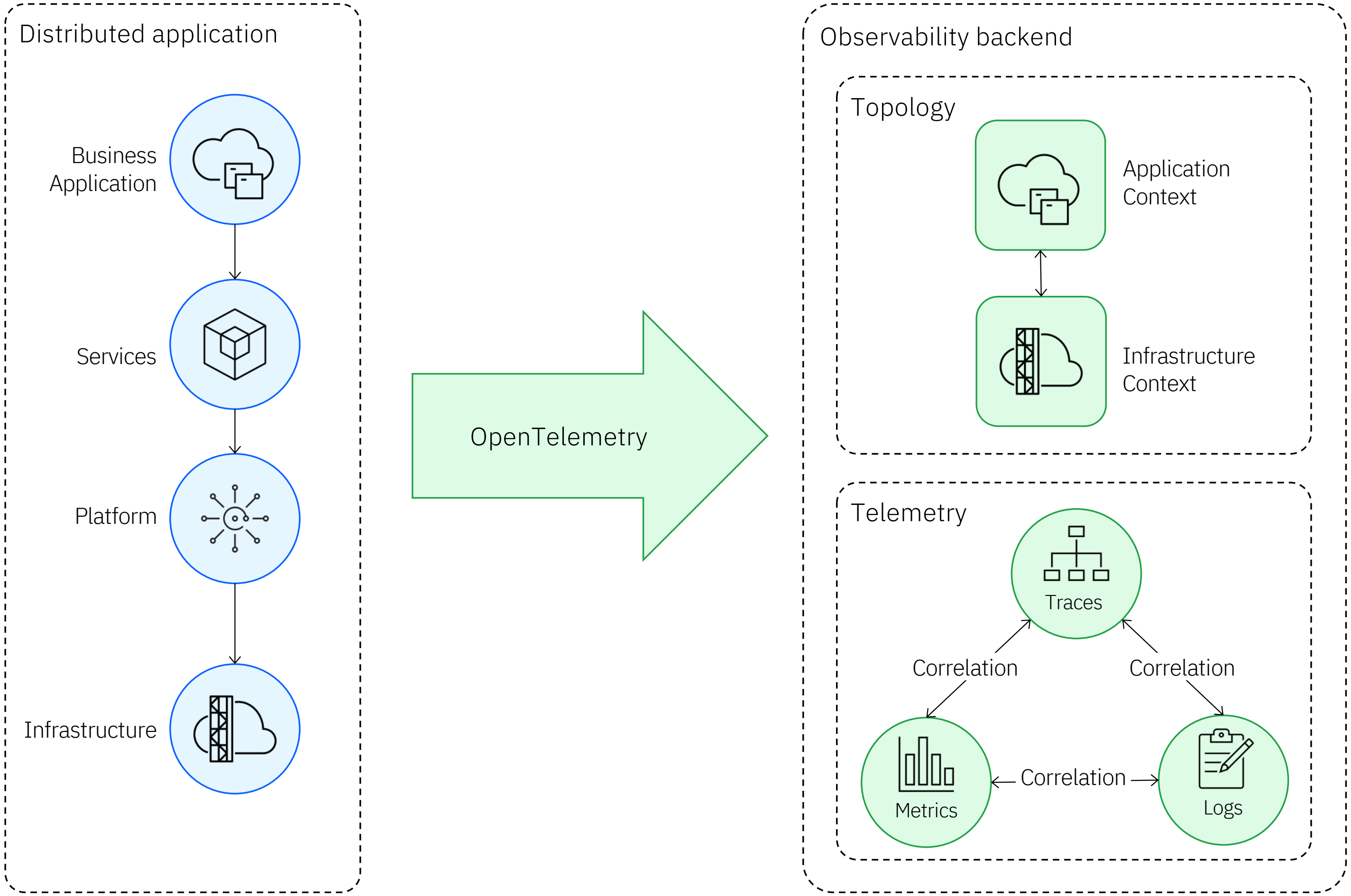
Source: Practical OpenTelemetry

The value of OpenTelemetry

- Correlation of signals based on time of execution, execution context and origin of telemetry

- Telemetry annotation and context propagation

- Enables new ways to analyse application and infrastructure performance issues in context



03 What is the OpenTelemetry on Mainframe SIG?

Goals of the OpenTelemetry on Mainframe SIG

End-to-end observability

Mainframe users can capture application spans (traces), metrics, logs, and (eventually) profiles from their applications with OpenTelemetry and can export these to the tools of their choice.

Infrastructure Observability

Mainframe users can capture infrastructure metrics and logs with OpenTelemetry and can export these to the tools of their choice.

Semantic Conventions

All captured telemetry has appropriate resource and interaction metadata attached to it, and this metadata conforms to the OpenTelemetry semantic conventions.

Low footprint

Capturing infrastructure and application telemetry from a mainframe incurs a minimal performance impact and allows for different deployment scenarios of the OpenTelemetry Collector to minimize both mainframe CPU costs and mainframe network egress costs.

Source: SIG proposal

Tracks for enabling OpenTelemetry on the Mainframe

Track: Semantic Conventions

Map the mainframe concepts and metrics on the OpenTelemetry semantic conventions

Track: Code Instrumentation

Support instrumentation of mainframe-specific programming languages

- COBOL, PL/1, …

Enable platform support of the code instrumentation for the languages supported by OpenTelemetry today.

Track: Collector Enhancements

Enable platform support of the OpenTelemetry Collector within mainframe environments and support resource detection.

Capture resource metrics from mainframes for z/OS and Linux on IBM Z.

Capture logs from mainframes for z/OS and Linux on IBM Z.

Source: SIG proposal

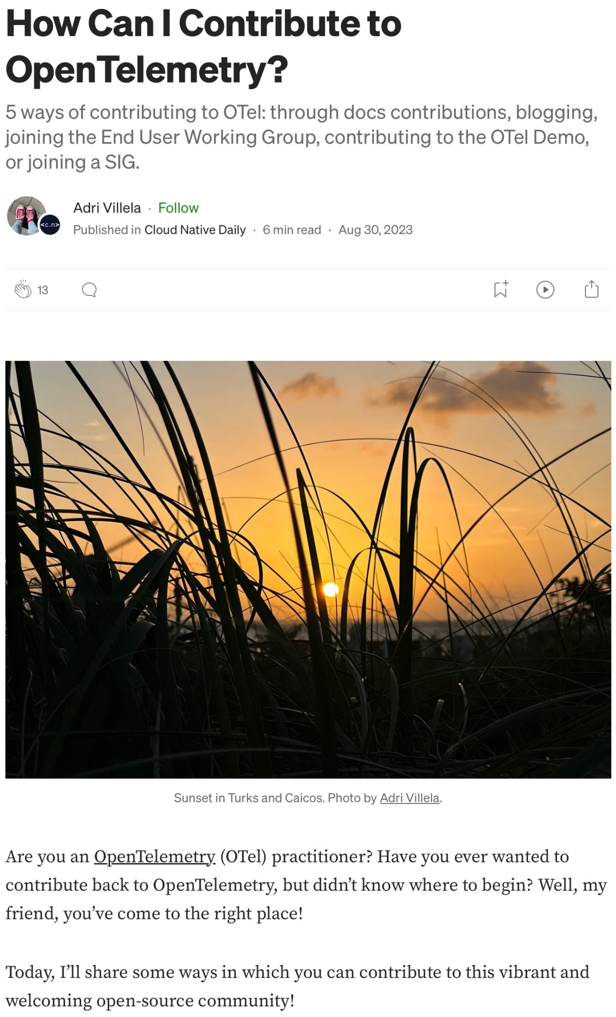
04 How to engage?

How to engage?

1. Get approval from your organization to contribute OpenTelemetry as a CNCF project (EasyCLA)

2. Join the SIG meeting on Tuesdays, 10:00 am PT
   - Zoom meeting
   - Meeting Notes

3. Join the Slack channel #otel-mainframes



How Can I Contribute to OpenTelemetry?

5 ways of contributing to OTel: through docs contributions, blogging, joining the End User Working Group, contributing to the OTel Demo, or joining a SIG.

Adri Villela · Follow
Published in Cloud Native Daily · 6 min read · Aug 30, 2023

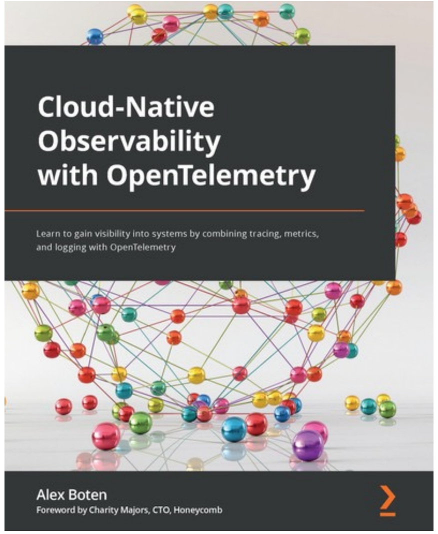
Sunset in Turks and Caicos. Photo by Adri Villela.

Are you an OpenTelemetry (OTel) practitioner? Have you ever wanted to contribute back to OpenTelemetry, but didn't know where to begin? Well, my friend, you've come to the right place!

Today, I'll share some ways in which you can contribute to this vibrant and welcoming open-source community!

Source: medium.com

Where to learn more about OpenTelemetry?

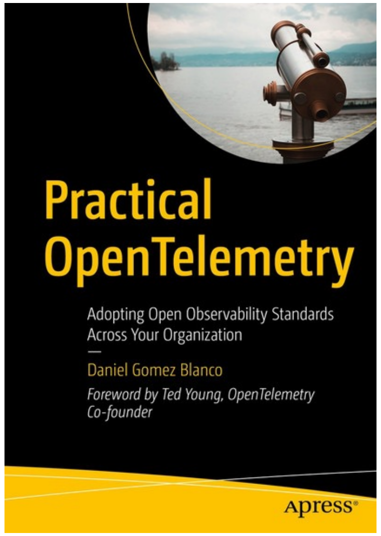


Cloud-Native Observability with OpenTelemetry

Learn to gain visibility into systems by combining tracing, metrics, and logging with OpenTelemetry

Alex Boten

Foreword by Charity Majors, CTO, Honeycomb



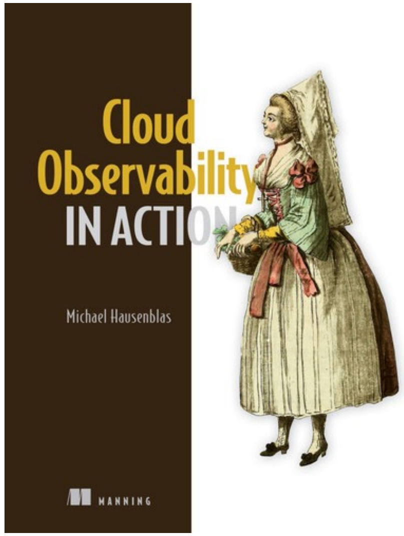
Practical OpenTelemetry

Adopting Open Observability Standards Across Your Organization

Daniel Gomez Blanco

Foreword by Ted Young, OpenTelemetry Co-founder

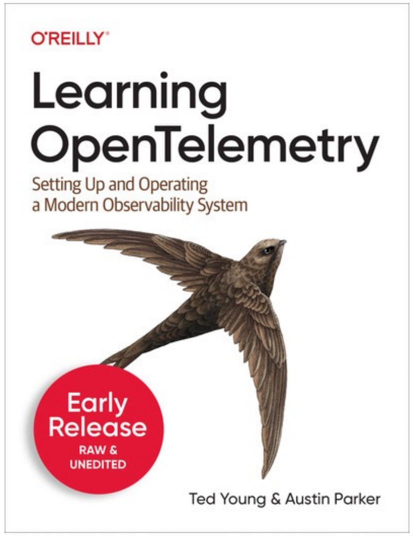
Apress®



Cloud Observability IN ACTION

Michael Hausenblas

MANNING



O'REILLY®

Learning OpenTelemetry

Setting Up and Operating a Modern Observability System

Early Release RAW & UNEDITED

Ted Young & Austin Parker

How does OpenTelemetry improve the observability of the mainframe?

OpenTelemetry will simplifiy the distribution of telemetry data generated on mainframe.

OpenTelemetry can be used for well-selected use cases on the mainframe already today.

OpenTelemetry will fill “white spaces” in the platform instrumentation of established observability products.

OpenTelemetry will enable observability of the mainframe for products that historically do not or only partially support the platform.

Come and join the
OpenTelemetry on Mainframe SIG!

We welcome long-time mainframers and those who are new to the platform and want to learn about mainframe technology.