

OpenTelemetry and Epsagon - a Love Story in Three Acts

learn application-experience full-stack-observability
opentelemetry epsagon



Yosef Arbiv

Wednesday, March 30th, 2022

My name is Yosef Arbiv, and I am an R&D Team Leader at Epsagon - now a part of Cisco. I want to share with you the story of how we became a part of the OpenTelemetry Community. I hope you will be able to learn from it the advantages of being a part of an Open-Source community and the Dos and Don'ts of using Open-Source as a significant part of your product.

Act I: Epsagon in the Pre-OpenTelemetry Era



When we launched the first version of the [Epsagon SDKs](#), there was no standard for tracing collection. [OpenTracing](#) started as a CNCF project in October 2016. In 2018 when we began Epsagon, it was still in an early phase and backed mainly by LightStep, a competitor. Back then, most of the customers we talked with used logging solutions, tracing solutions that didn't correlate across systems, or other early-stage distributed tracing solutions.

We decided to create a proprietary tracing format. By doing so, we were able to add unique features that weren't a part of OpenTracing, (Spoiler: those are still not a part of OpenTelemetry):

1. Payloads collection. Collection of data from the requests, and not only meta-data.

2. Strong IDs. We were targeting making correlations across managed services, mainly AWS. Most of which did not propagate headers. To support this case, we came up with the concept of “strong IDs” to allow matching in the backend. This idea became [Epsagon’s first patent](#) later.

Soon enough, we open-sourced our libraries. We understood this is an industry standard for customer SDKs and will increase our customer’s confidence in the product and maybe even create a community around it.

Act II: Standardization

During 2019, we looked into supporting k8s clusters, in addition to serverless managed services that were supported until then. When we started researching the Java applications that our customers were running, we understood that developing a tracing collection solution on our own would be too complex.

We also figured out that we were too little to create an open-source community on our own, and it made much more sense to join a growing one.

At this point, OpenTracing was more mature and had wider adoption. We decided to create an OpenTracing based Java agent and add our logic on top of OpenTracing.

In mid-2019, [OpenTelemetry](#) was announced, uniting OpenTracing and OpenCensus. We created more libraries that are OpenTelemetry based and did so mainly by using OpenTelemetry code, changing and adapting it to match our needs.

This way, we could create new libraries quickly, but maintaining these became a headache. The distance between our libraries and OpenTelemetry grew as time went by, and keeping them up to date was impossible.

Act III: Joining Cisco and the OpenTelemetry Community



To make our SDKs more sustainable, we decided to try and develop them as OpenTelemetry distributions. We did an experiment with our Java SDK and created an OpenTelemetry based agent that used the OpenTelemetry library and extended it. We needed to add some backend adaptations to support the new format, but overall the change was smooth, and we planned to expand it to other programming languages.

Shortly after the first successful experiment, we had a significant change in plans – [Cisco acquired Epsagon](#). The Epsagon product was to be deprecated gradually, and we started to work on a new Full Stack Observability product. We joined other teams with experience in working with OpenTelemetry and contributing to OpenTelemetry and learned a lot from them.

We decided that our product should support OpenTelemetry natively together with our distributions, which will allow providing more value to our customers.

We started developing such distributions. As part of the process, we also started looking into opportunities to contribute to OpenTelemetry – add new functionality, and fix bugs.

The Future

In the future, we hope to increase our involvement in OpenTelemetry, by contributing any relevant code from our distributions. As a part of Cisco, we hope to become a significant player in the OpenTelemetry community and help build an excellent observability future.

If you want to hear more about our journey with OpenTelemetry, [join my session at Cisco Live 2022](#), in person or virtually. See you there!

