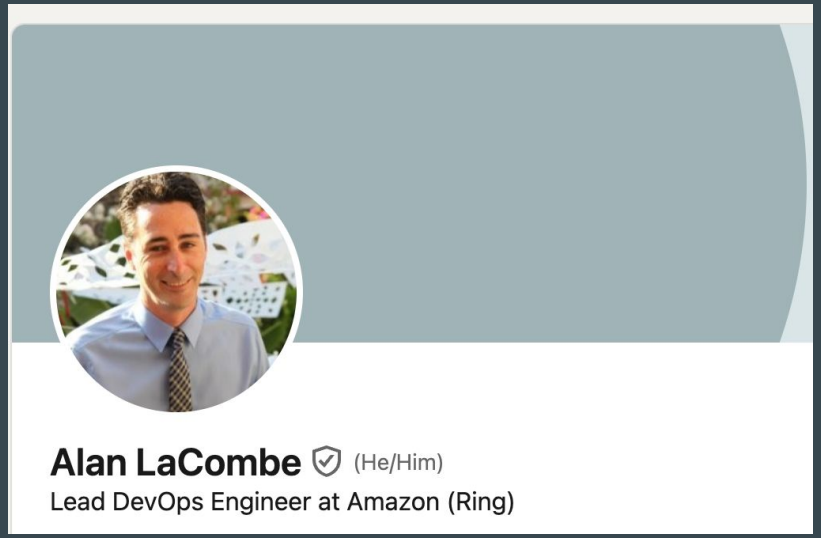
Observability

•••

Path to freedom

Alan LaCombe

Who am I



Experience

Lead DevOps Engineer
Amazon
Jun 2019 - Present · 4 yrs 10 mos
Santa Monica, CA

I do Lead DevOps

AWS Lambda, High Availability (HA) and +18 skills

Lead DevOps Engineer
ReSci (Retention Science) · Full-time
Jun 2017 - Jun 2019 · 2 yrs 1 mo
Santa Monica, CA

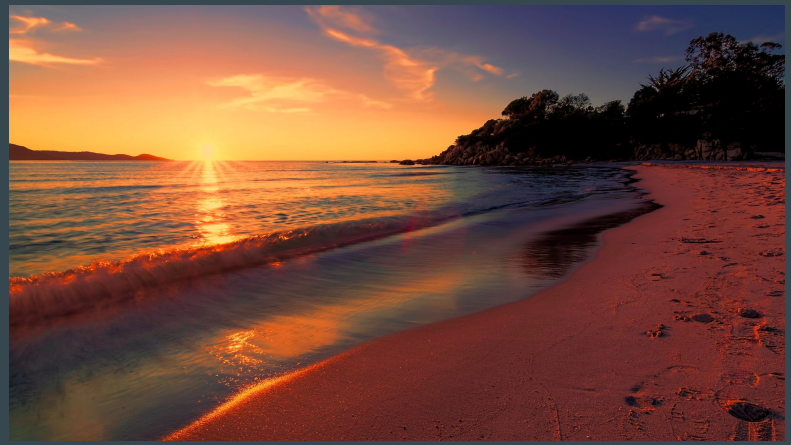
Build and maintain servers for a big data SAAS. Automate all servers and migrat
deployment....

AWS Lambda, High Availability (HA) and +17 skills

What are my biases?

- My entire career in devops is in AWS and my experience and solutions reflect that
- I do not want to reinvent the wheel
- If I have to run it I want widely adopted open source
- I would prefer someone else to run the low level infrastructure
- I do not want to be paged

What is the story I wanted to tell



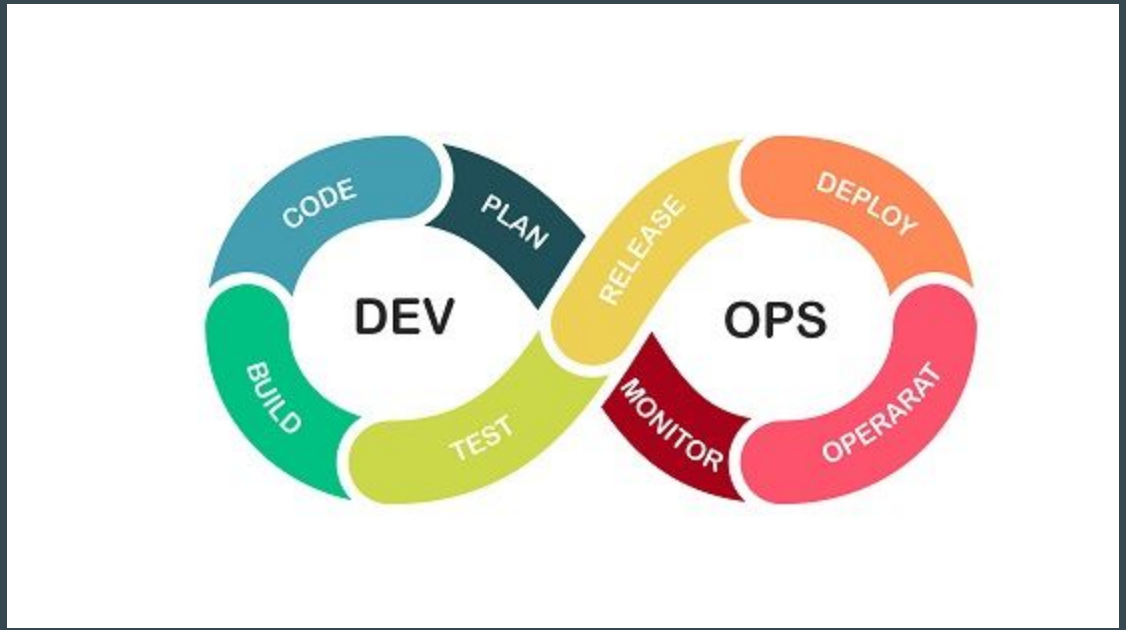
What is the real story

IT DEPENDS

EVERYTHING SUCKS

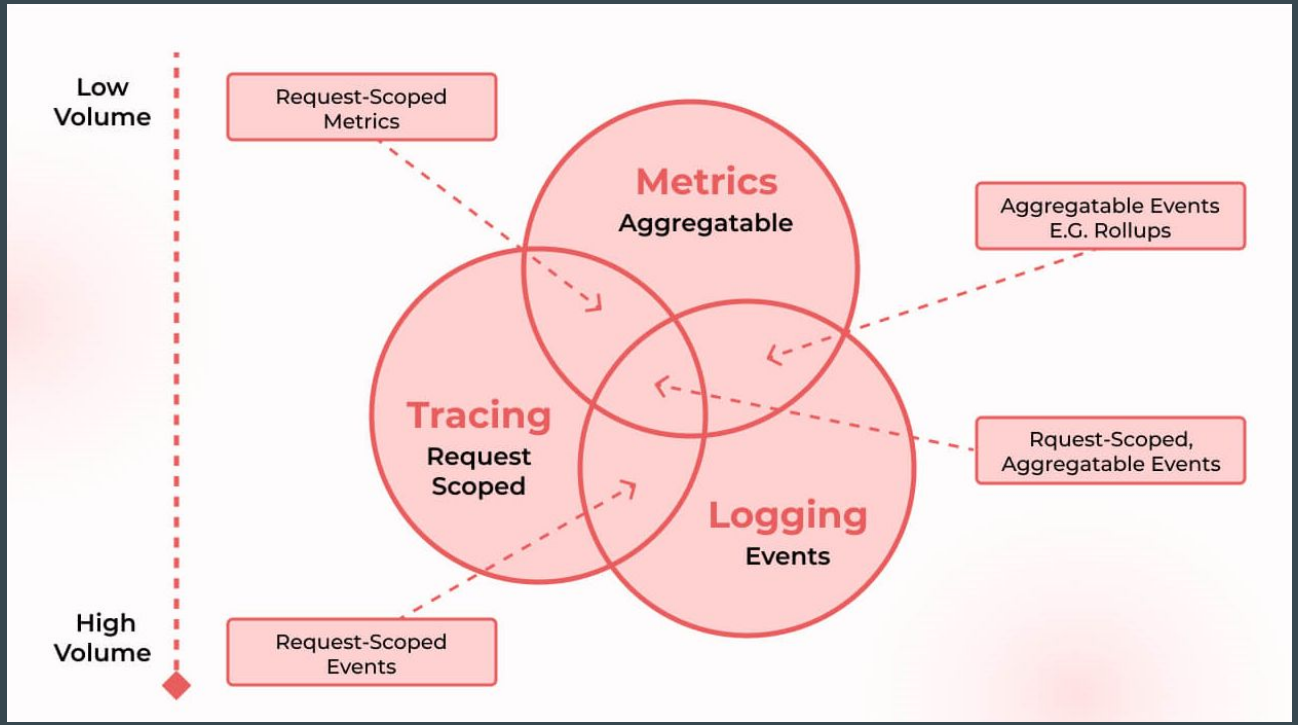
Different audiences

- C level
  - Costs
  - Risk to company
  - Resources to implement and maintain
- Devops
  - Building and deploying
  - Operations
  - Troubleshooting
- Developers
  - Visibility
  - Reducing risk to service


CODE
PLAN
DEV
BUILD
TEST
RELEASE
DEPLOY
OPS
MONITOR
OPERARAT

What is observability vs monitoring

- Monitoring is based on gathering predefined sets of metrics and logs
- Observability is based on exploring properties and patterns not defined in advance.



Error handling
Continuous profiling

What do we really care about?

- We want to know when something is broken
  - Before the customer
- Where is it broken?
- What caused it to break?
- Give me some tools

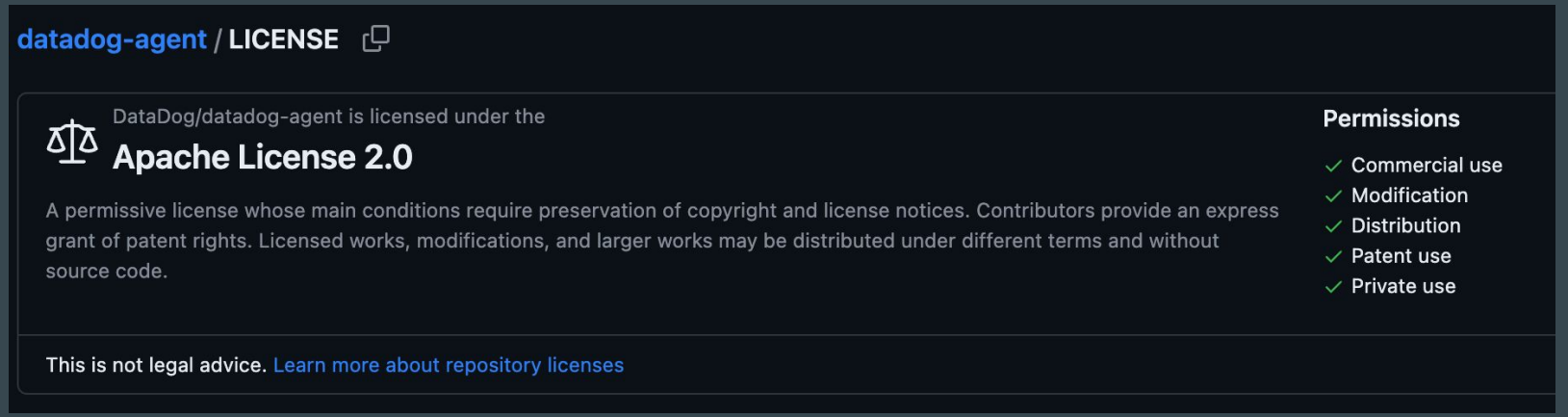
Current list of technologies

https://landscape.cncf.io/

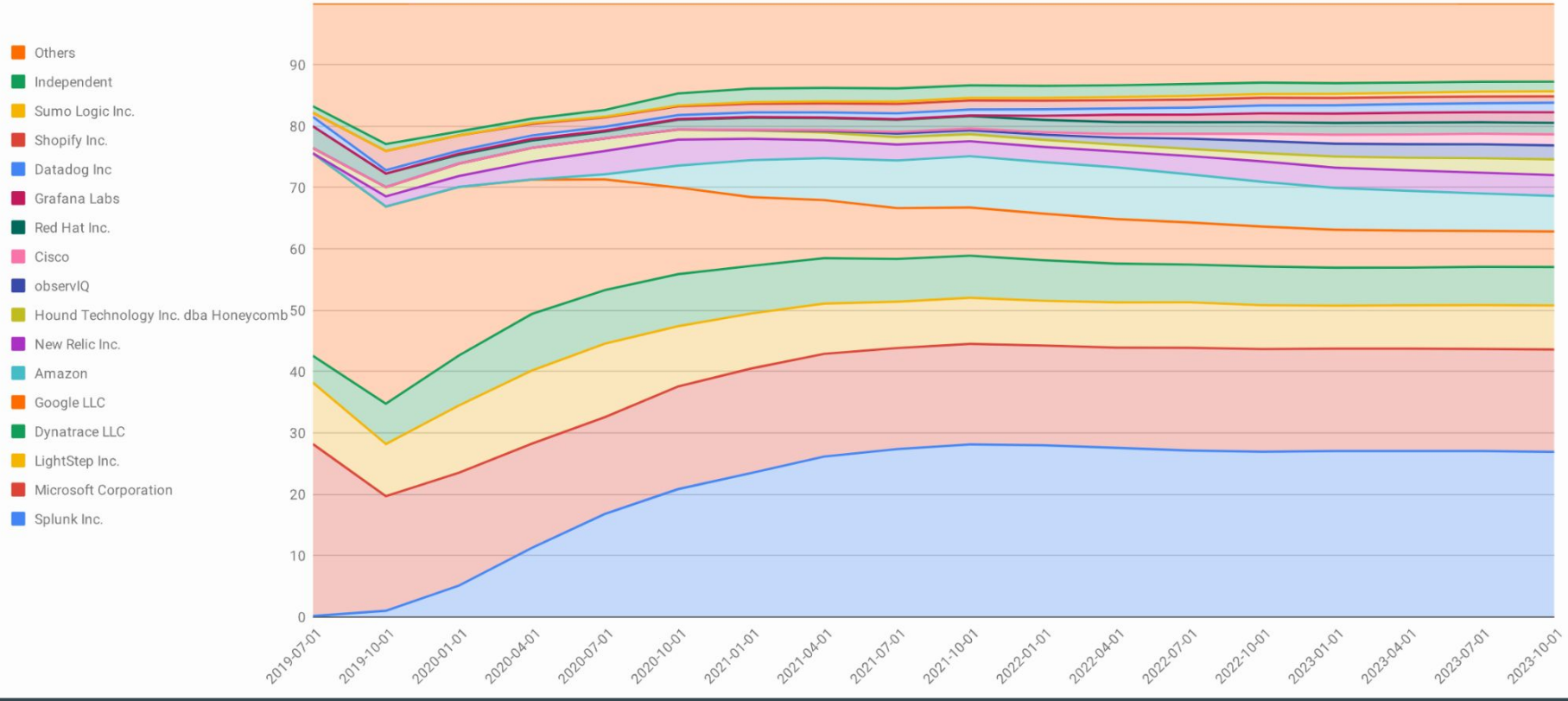
What is open source vs “closed” source

The software library, the collector and the backend are open sourced. Minimal amount of features gated by enterprise licence.

In this talk I am going to focus on opentelemetry and prometheus as open source.


datadog-agent / LICENSE
DataDog/datadog-agent is licensed under the
Apache License 2.0
A permissive license whose main conditions require preservation of copyright and license notices. Contributors provide an express grant of patent rights. Licensed works, modifications, and larger works may be distributed under different terms and without source code.
Permissions
Commercial use
Modification
Distribution
Patent use
Private use
This is not legal advice. Learn more about repository licenses

Percentage breakdown of OpenTelemetry contributions by company from Q2 2019-Q2 2023



What are the advantages of paid closed solution

You extend your engineering team with experts in the field of observability.

Not having to take tickets, solve 4am outages or be on call

Opportunity cost of devops engineer building and migrating to new unknown service

We are in alignment, they want us to send all the data and we want them to monitor all the data

Lots of options and tooling available at the push of a button

What are disadvantages of paid closed source?

High costs.

Lack of portability.

All or nothing. It is hard to piecemeal solutions to reduce costs.

Advantages of open source

Lower costs.

You can pick and choose your technology stack for the best performance to cost ratio.

Need faster queries? Keep data in memory or build aggregated metrics. Want long term cheap as possible with minimal querying? Put it in S3 or just expire the data or drop fidelity.

Huge community support for adding new features and you can just submit fixes to problems.

Lots of options and flexibility to try different backends and plugins

Industry support behind otel ensuring it will be the standard going forward.

What are the disadvantages of open source?

You are bound to community for fixes or features that are missing from product. If the community and your org do not align then you may never see the feature you need.

Lack of support.

You have to host the data yourself with high SLAs.

For otel in particular this is a project that is still in works. There are many features just being rolled out that I have wanted for years.

Each individual language has a completely different community supporting it.

So why did I ever move?

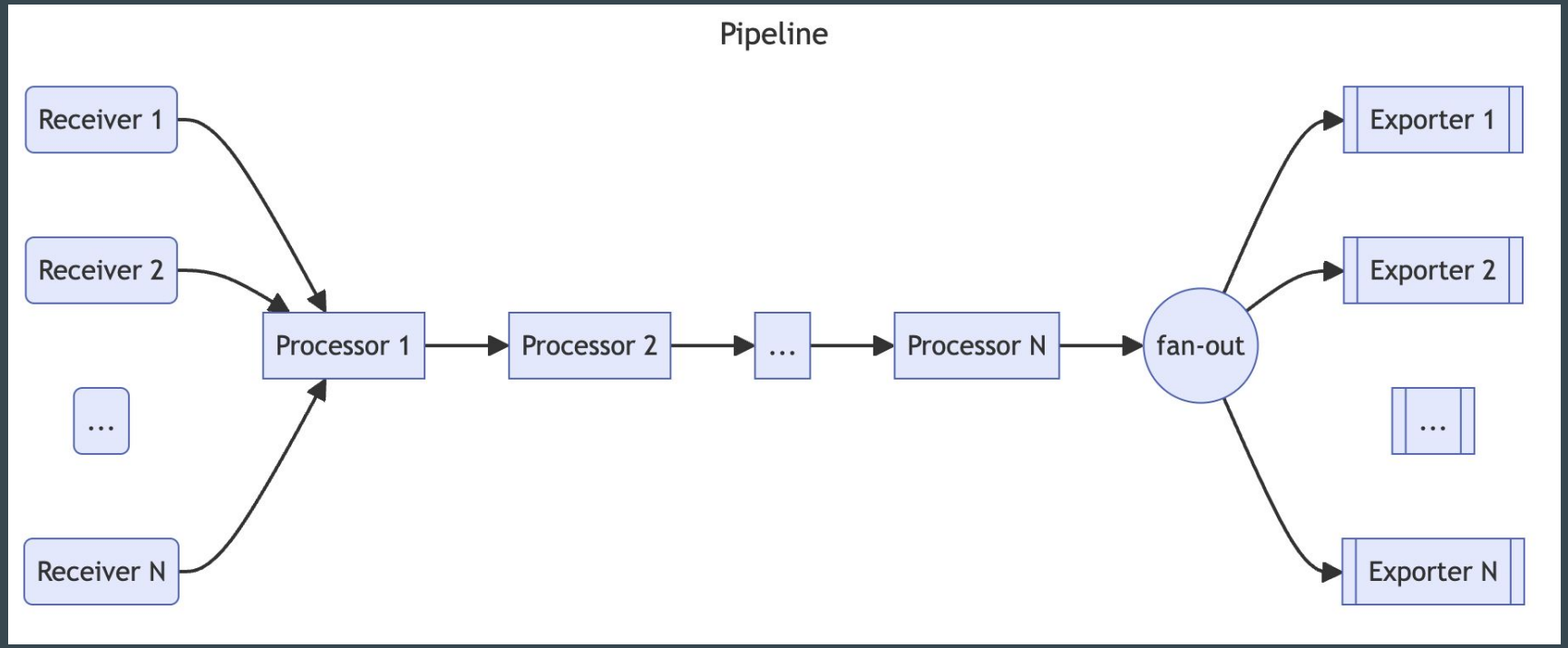
What did we decide to do

Migrate to opentelemetry collector for metrics

This future proofed us and left us flexible in the future

Kept our existing logging

Planned phase two tracing


Pipeline
Receiver 1
Receiver 2
...
Receiver N
Processor 1
Processor 2
...
Processor N
fan-out
Exporter 1
Exporter 2
...
Exporter N

What went well?

Metrics were easy to migrate

We used AWS AMP to reduce risk and not have to maintain low level infra

Lots of great options for dashboarding

Adoption was straight forward across the teams

What went bad

Prometheus queries required re-training teams and rebuilding graphs

Opentelemetry metrics were not done yet within libraries

Quantiles are different way to do storage and require some understanding and knowledge from teams

Initially alert manager was not installed on our prometheus and we had to create alerting on monitoring plane

Migrating dashboards was manual and tedious

Lack of automation around dashboard creation and modification

What went really bad

Logs just went GA 6 months ago

Lost some AMP features until we could implement phase two.

Phase two completely stalled

Each language is its own open source project and different features are implemented per the community

How do you get there?

Add open telemetry today to every new project

Scope if you can do it for older services. Review older codebases and validate they are supported.

Swap current collector for open telemetry collector

Start playing with pipelines and outputs to send data to other open source backends

Thank you