

OBSERVABILITY

BEYOND LOGGING, METRICS & TRACING



Photo by [Stephanie McCabe](#) on [Unsplash](#)

@TBOEGHK

Hi there 

I'm a Senior Search- & Operations Engineer based in Hamburg, Germany.

-  Embracing Open Source
-  Maintaining [geoip-api](#), an api for Maxminds geoip database;
-  Running managed Prometheus Exporters in the cloud at [ping7.io](#)
-  Organizing [solr.cool](#) Solr plugin repository;
-  Raving about `arm64` and native Java builds
-  Ping me about **search, solr, observability, infrastructure as code**
-  Reach out to me: twitter.com/tboeghk
-  For details check out [my website](#)
-  I love everything about bikes & biking



You auto**complete** me.

SLIDES BY CHARITY MAJORS

"THE QUEEN OF OBSERVABILITY"



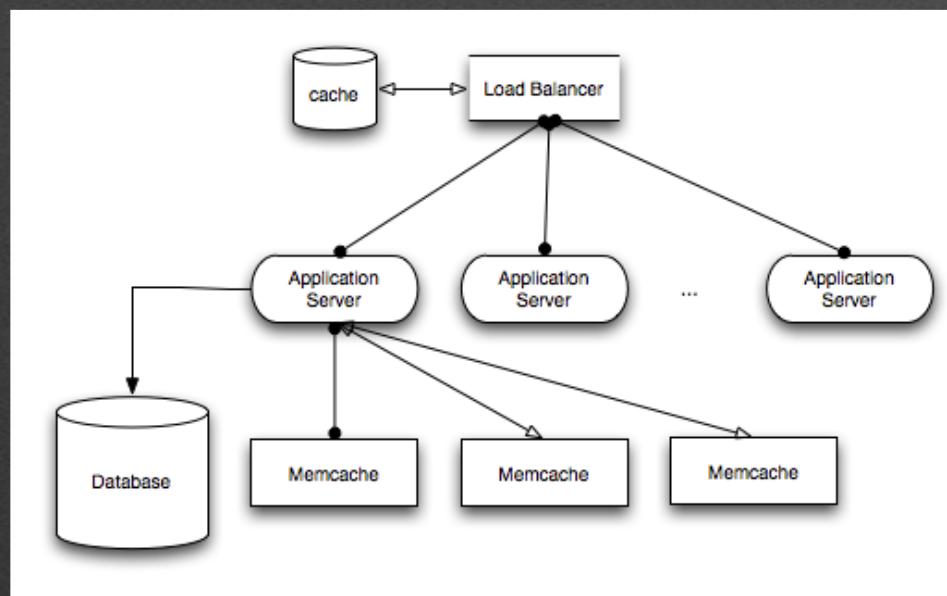
The trifecta:

Monitoring (time series databases, dashboards, 'metric' tools)

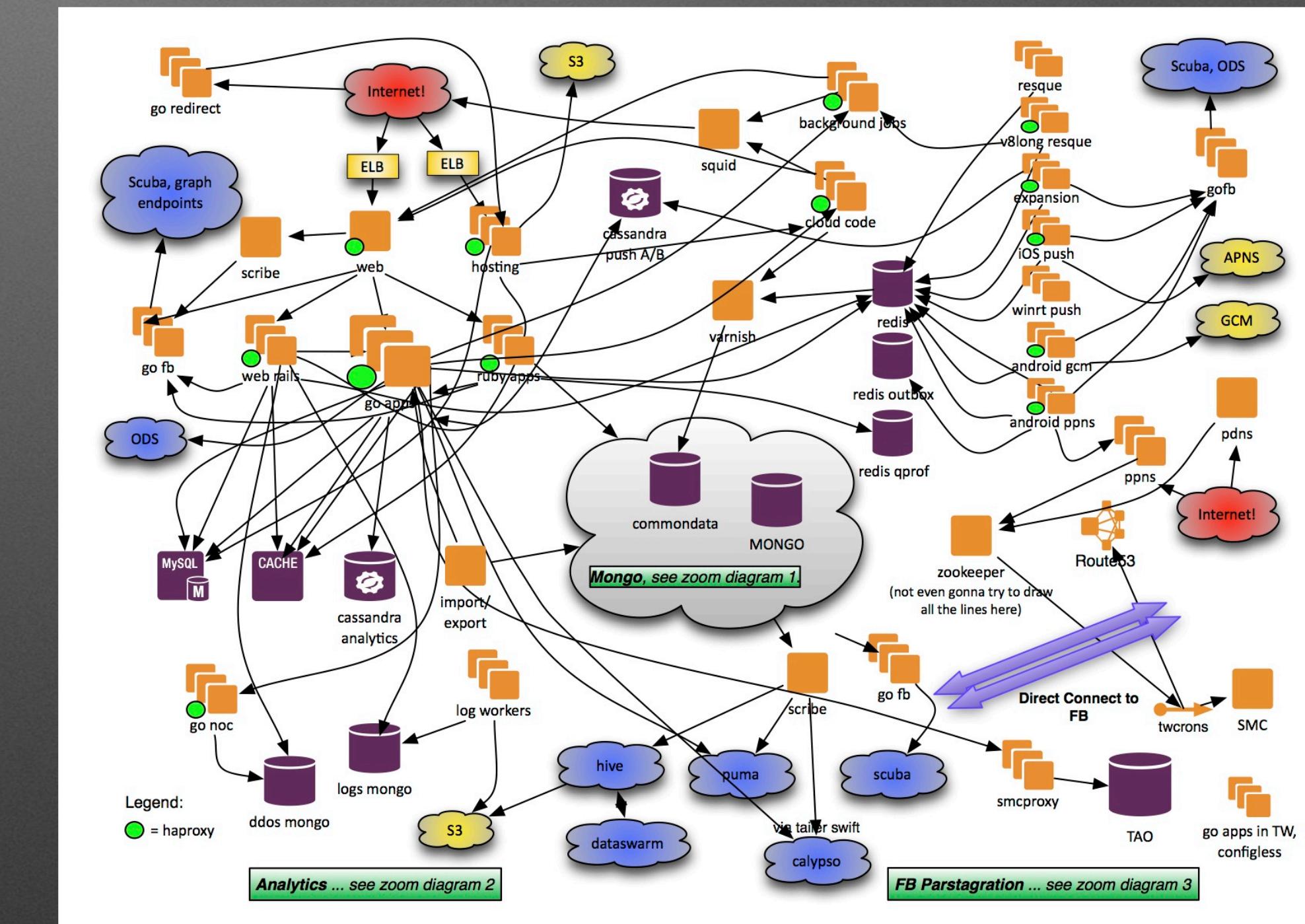
Logs (messy ass strings, really)

More recently, **APM and tracing.**

Architectural complexity



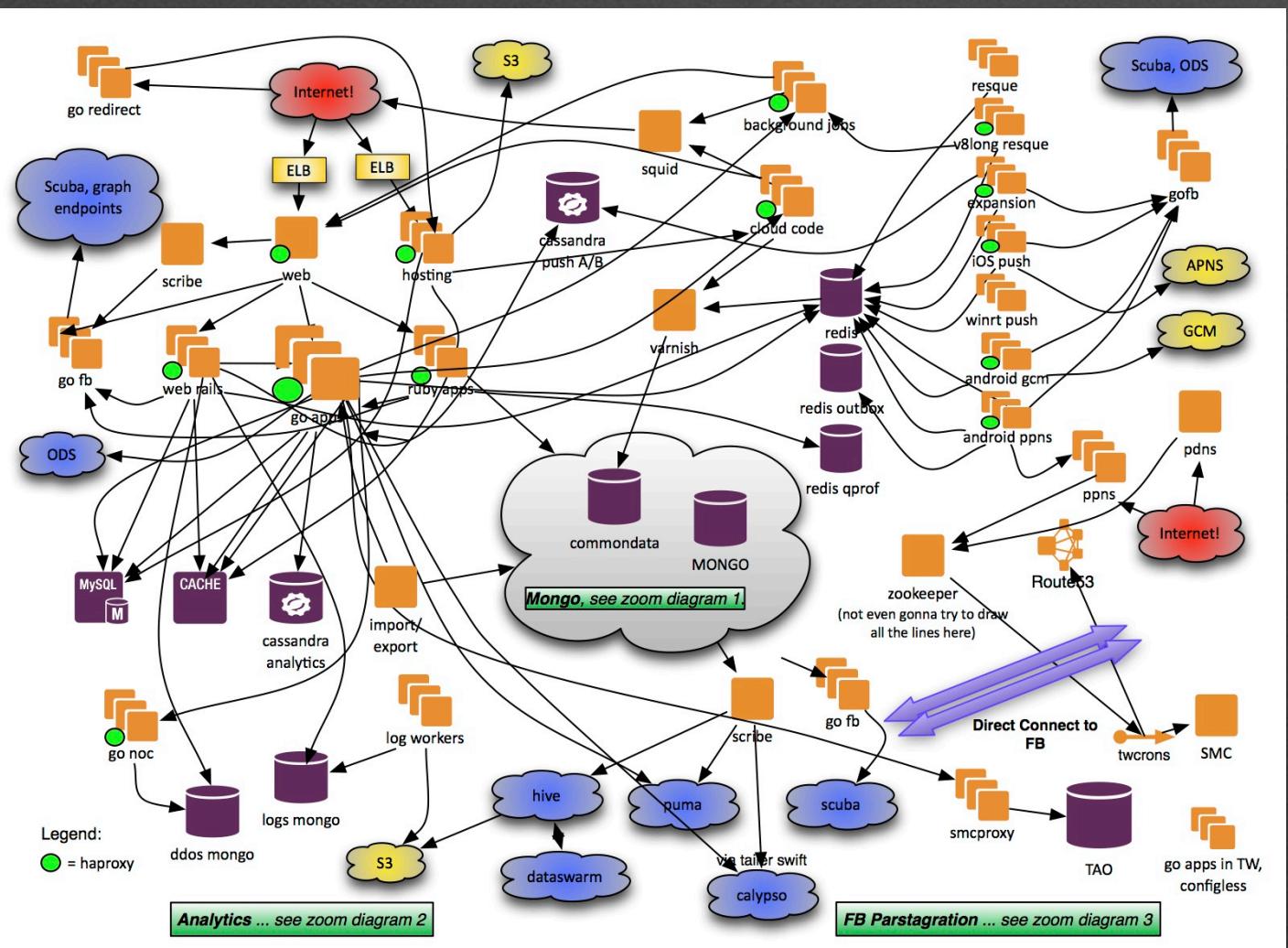
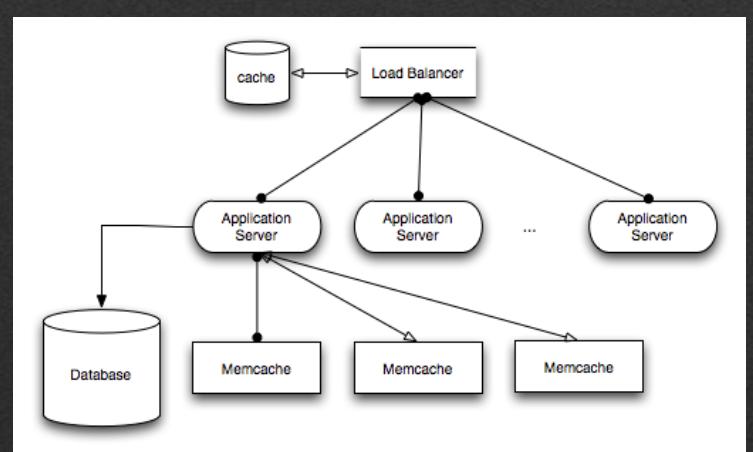
LAMP stack, 2005



Parse, 2015

LAMP stack => distributed systems
monitoring => observability

known unknowns => unknown unknowns





Monitoring is the action of observing and checking the behavior and outputs of a system and its components over time.

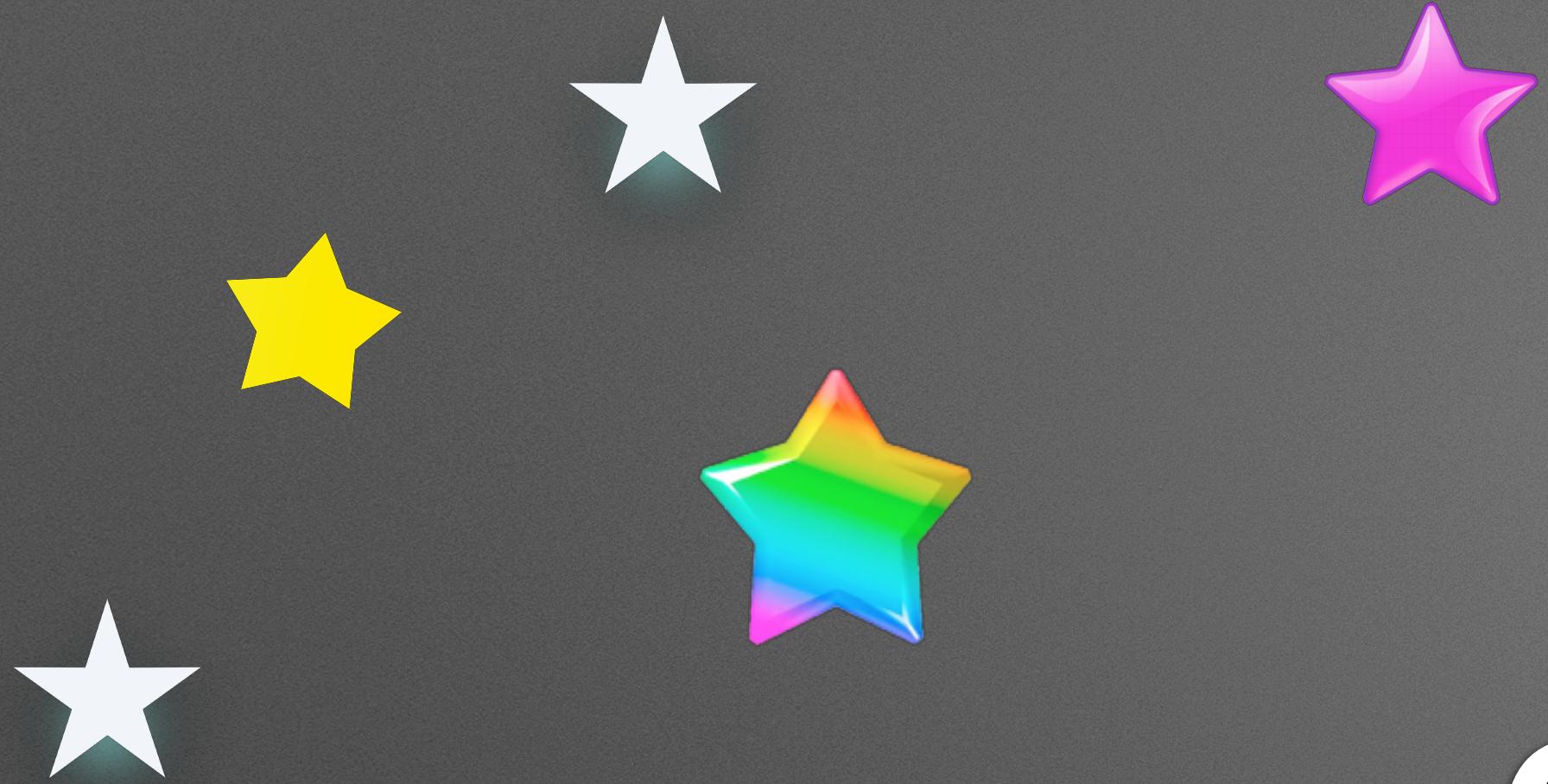
“Monitoring is dead.”

“Monitoring systems have not changed significantly in 20 years and has fallen behind the way we build software. Our software is now large distributed systems made up of many non-uniform interacting components while the core functionality of monitoring systems has stagnated.”

@grepory, Monitorama 2016

Observability is NOT the same as monitoring.





Observability



“In control theory, observability is a measure of how well internal states of a system can be inferred from knowledge of its external outputs. The observability and controllability of a system are mathematical duals.” — wikipedia

... translate??!?



★ Observability... for software engineers:

Can you understand what's happening **inside** your systems, just by asking questions from the **outside**? Can you debug your code and its behavior using its output?

Can you answer new questions **without shipping new code**?



Monitoring

Represents the world from the perspective of a third party, and describes the health of the system and/or its components in aggregate.



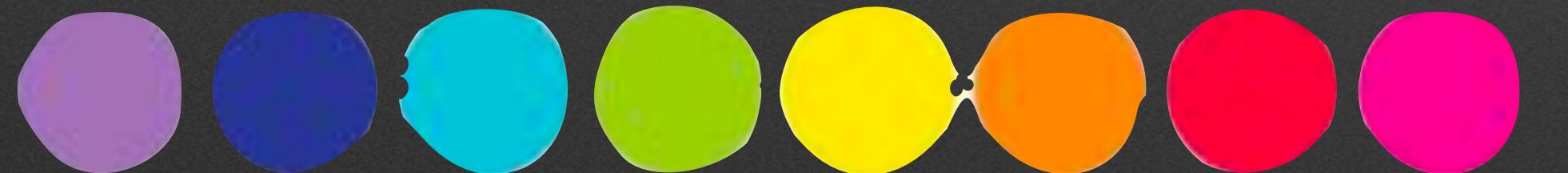
Observability

Describes the world from the first-person perspective of the software, executing each request. Software explaining itself from the inside out.



Let's try some examples!

Can you quickly and reliably track down problems like these?



“Photos are loading slowly for some people. Why?”

(old-school LAMP stack)

The app tier capacity is exceeded. Maybe we rolled out a build with a perf regression, or maybe some app instances are down.

monitor these things

Errors or latency are high. We will look at several dashboards that reflect common root causes, and one of them will show us why.

DB queries are slower than normal. Maybe we deployed a bad new query, or there is lock contention.

Monitoring

“Photos are loading slowly for some people. Why?”

(microservices)

Any microservices running on c2.4xlarge instances and PIOPS storage in us-east-1b has a 1/20 chance of running on degraded hardware, and will take 20x longer to complete for requests that hit the disk with a blocking call. This disproportionately impacts people looking at older archives due to our fanout model.

wtf do i ‘monitor’ for?!

Canadian users who are using the French language pack on the iPad running iOS 9, are hitting a firmware condition which makes it fail saving to local cache ... which is why it FEELS like photos are loading slowly

Our newest SDK makes db queries sequentially if the developer has enabled an optional feature flag. Working as intended; the reporters all had debug mode enabled. But flag should be renamed for clarity sake.

Monitoring?!?

Welcome to distributed systems.



it's probably fine.

Is it?
How do you
know?

WHERE DO WE START?

HOW DO WE IMPLEMENT OBSERVABILITY?





OBSERVABILITY IS A MINDSET

IT'S NEITHER A TOOL NOR A DEPARTMENT

SAAS SOLUTIONS

Fast setup & Easy insights

High Lock-In-Effect

Transaction-Data stored in the Cloud

Various pricing models

One size fits all

New Relic, Honeycomb, Elastic APM, Instana



Photo by Ignacio Amenábar on [Unsplash](#)

OPENSOURCE SOLUTIONS

Highly customizable

Understand your data (leads to better data)

Run on your own infrastructure

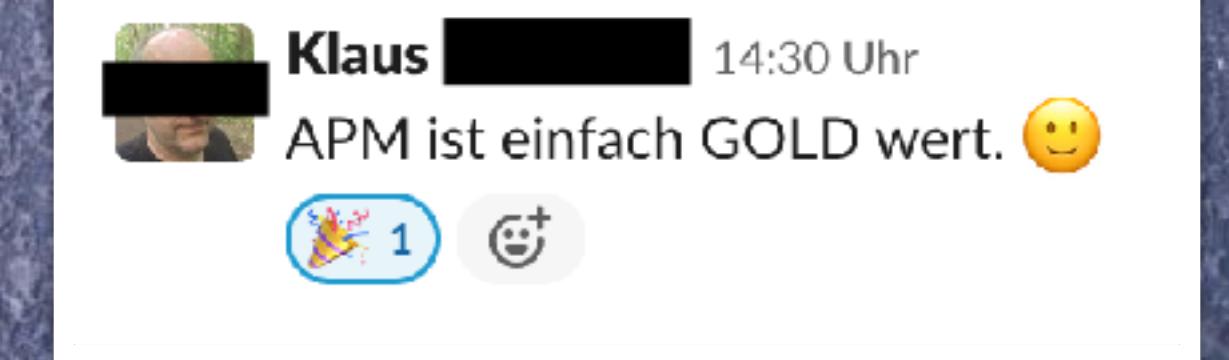
Available as SaaS offering

Prometeheus, Grafana, Elastic



TOOLS DON'T MATTER. START!

START
HERE.



FURTHER READINGS



O'REILLY®

Observability Engineering

Achieving Production Excellence



Charity Majors,
Liz Fong-Jones
& George Miranda