AWS re:Invent

DEV315-S

Building SRE from scratch at Coinbase during hypergrowth

Niall O'Higgins
coinbase

Daniel Maher
DATADOG

AWS re:Invent

aws

"Our goal is to make Coinbase the **most trusted** and easiest to use digital currency exchange."

Brian Armstrong
Co-founder & CEO of Coinbase

What is SRE?

- New field
- Definitions vary
- Many misconceptions

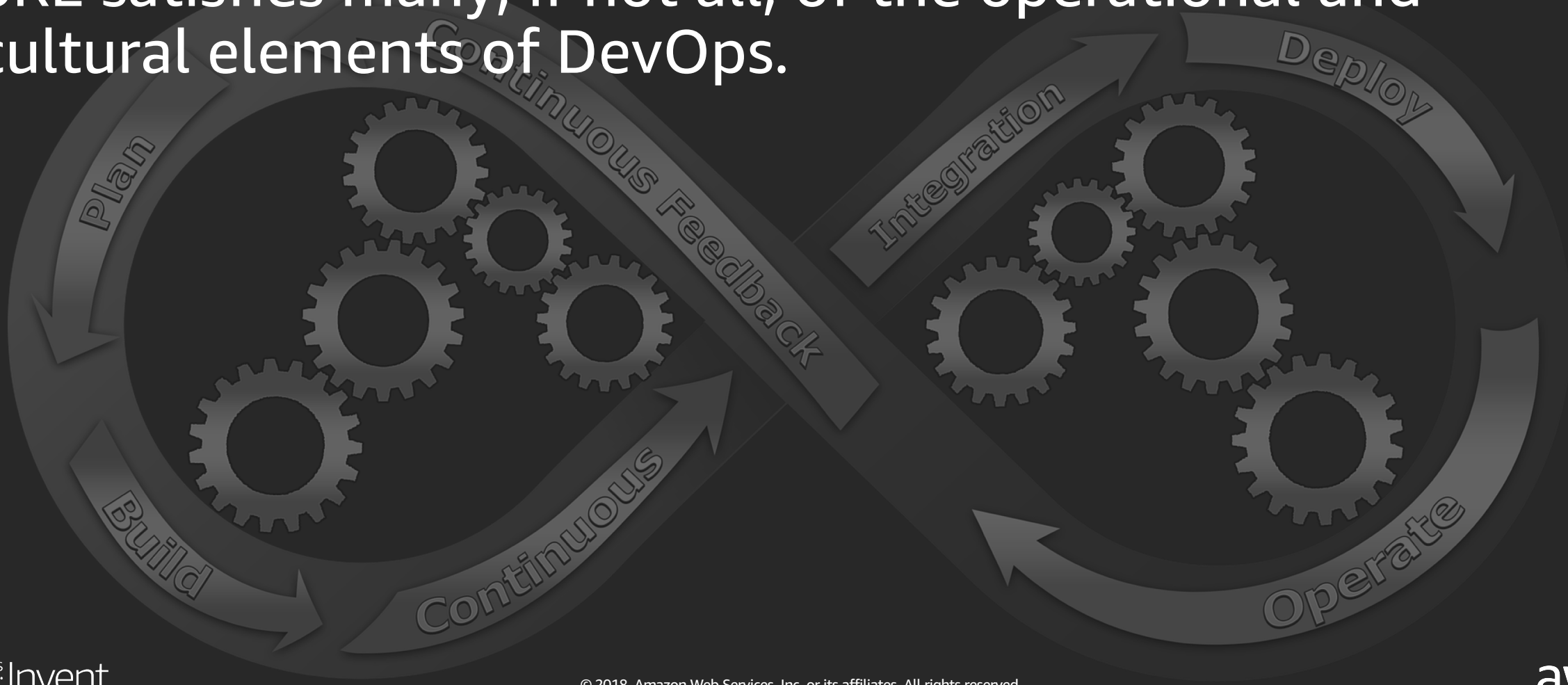
Is this SRE?

- Endless firefighting?
- Being on-call?
- Operational toil?

Those are the **symptoms**; SRE is the **cure**.

What about DevOps?

SRE satisfies many, if not all, of the operational and cultural elements of DevOps.



Key SRE insight #1

Measure and improve human, organisational, and machine systems.

Key SRE insight #2

Move from reactive to proactive. Go find sources of toil and eliminate them!

Key SRE insight #3

Provide an organisational back-pressure mechanism.

Set early expectations

- New language and fresh set of concepts.
- Takes time to absorb - no instant results!
- The best way to begin, is to begin.

The Coinbase strategy

- Service Level Indicators
- The “Four Golden Signals”

Metrics: the core of SLIs

- Natural tendency to over-engineer.
- Lots of data, none of it actionable or useful.

Optimise for KPIs, or high signal/noise.

The Four Golden Signals

- Latency
- Traffic
- Errors
- Saturation

Latency

- Direct impact on customer experience.
- Where and how you measure it is important.

Traffic

- The amount of work being done - or attempted.
- Direct relationship with business value.

Errors

- A nice, defined target to aim at.
- Direct impact on customer experience.

Saturation

- Real Talk: this is a tricky one.
- Direct relationship to both scaling and capacity planning.

Humans and the Four Golden Signals

- Latency
- Traffic
- Errors
- Saturation

Start, then iterate

- Start with an initial specification - even if it’s not ideal.
- Iterating on feedback is the key to getting it right.
- Keep it simple!

Spreadsheets are simple, right?

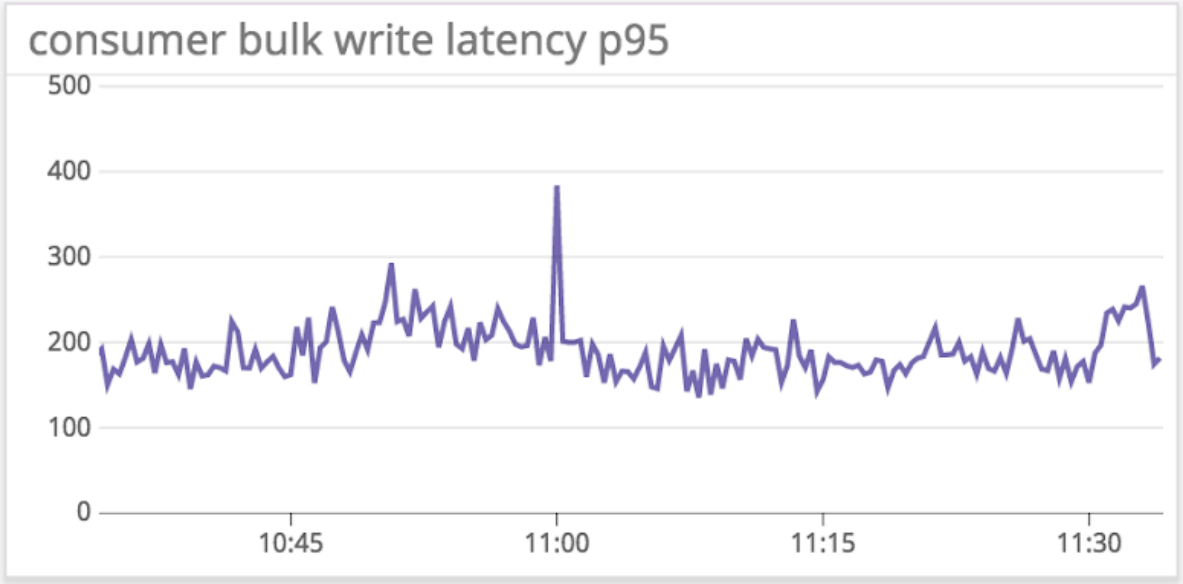
Service	Latency	Errors	Saturation	Traffic
Foo	foo.latency	foo.error_rate	Disc space	TPS
Bar	bar.response_time	bar.error_rate	Memory	TPS

SLIs: Defining “done”

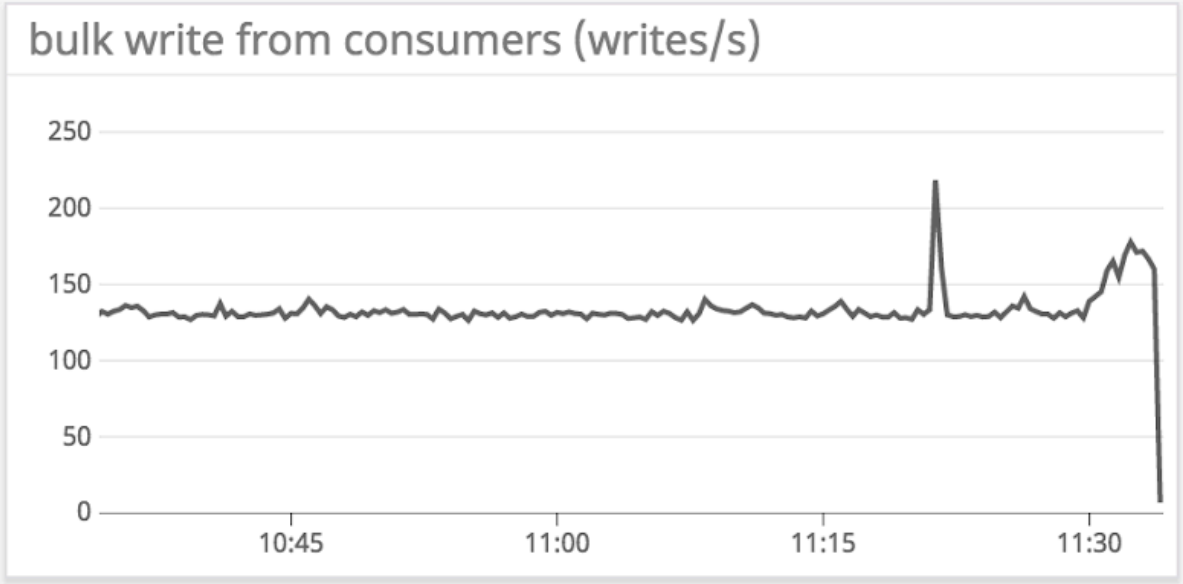
- Per-service dashboard in Datadog with timeseries chart for each indicator.
- Document describing the indicators and why they are important.

Datadog Dashboards

Latency

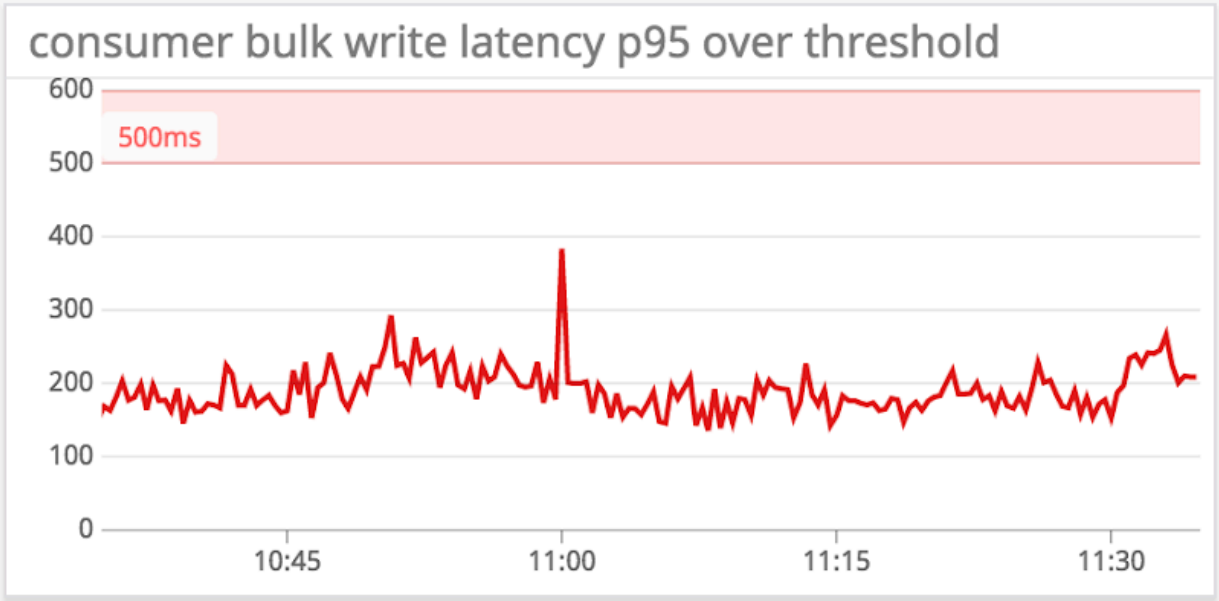


Traffic

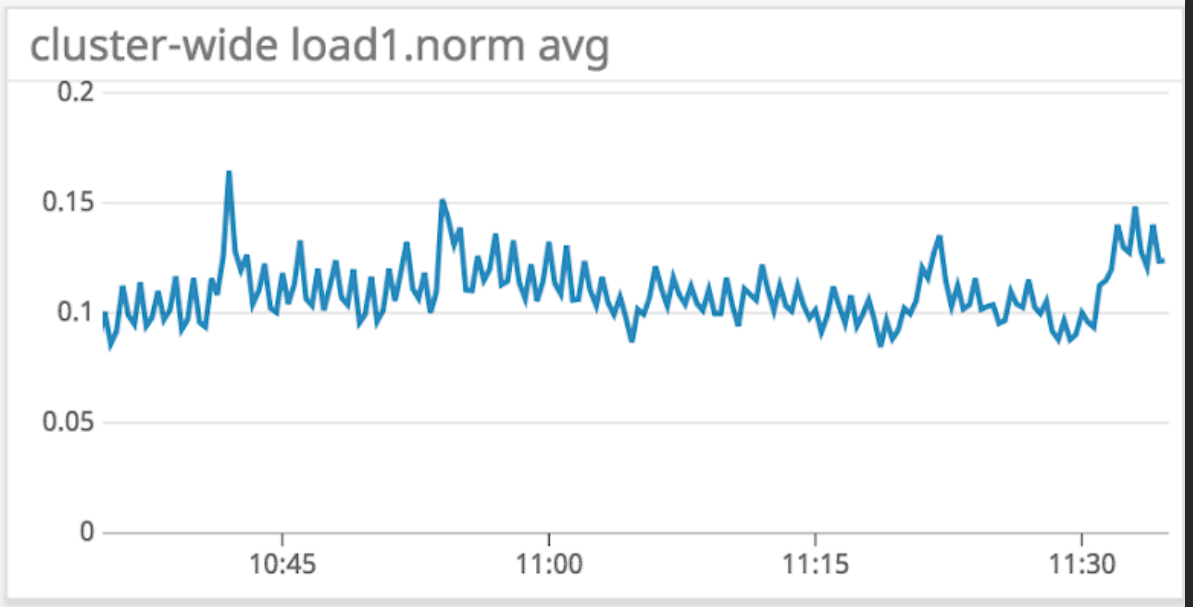


Datadog Dashboards

Errors

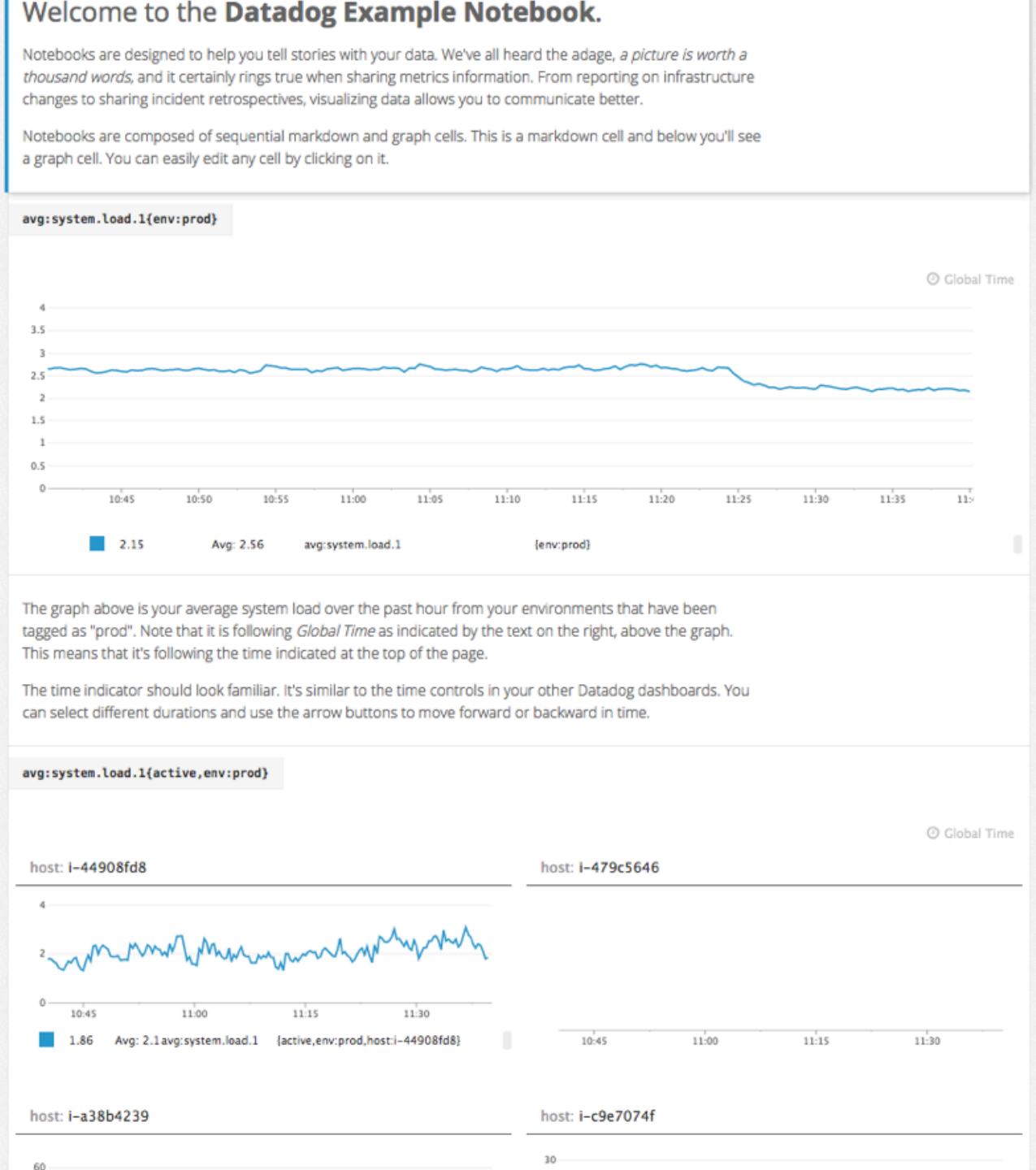


Saturation



Specification Documentation

- Spec vs. implementation
- Where do you want to instrument?
- Where is it easy to instrument?



First SLIs, then Promises

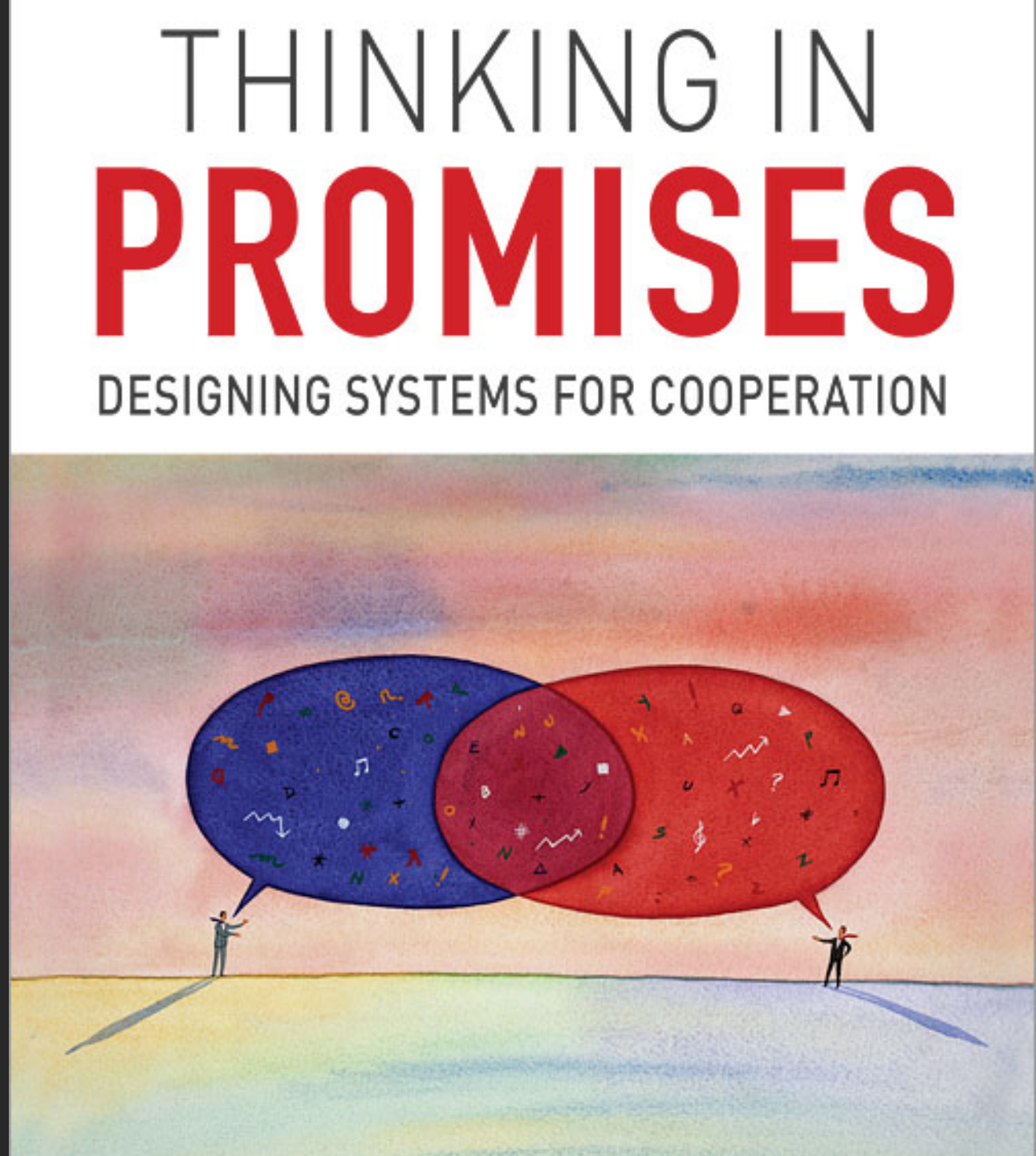
- Plain-language statements; easy to parse, easy to understand.
- Plenty of potential stakeholders.
- Start simple!

"You can rely on us to buy or sell crypto whenever you want."

Coinbase's "Prime Promise"

Essential Reading

Thinking in Promises
by Mark Burgess



Concerning Promises

- Promises have two parties.
- Promises can be human to machine, human to human, or machine to machine.

Example Promises

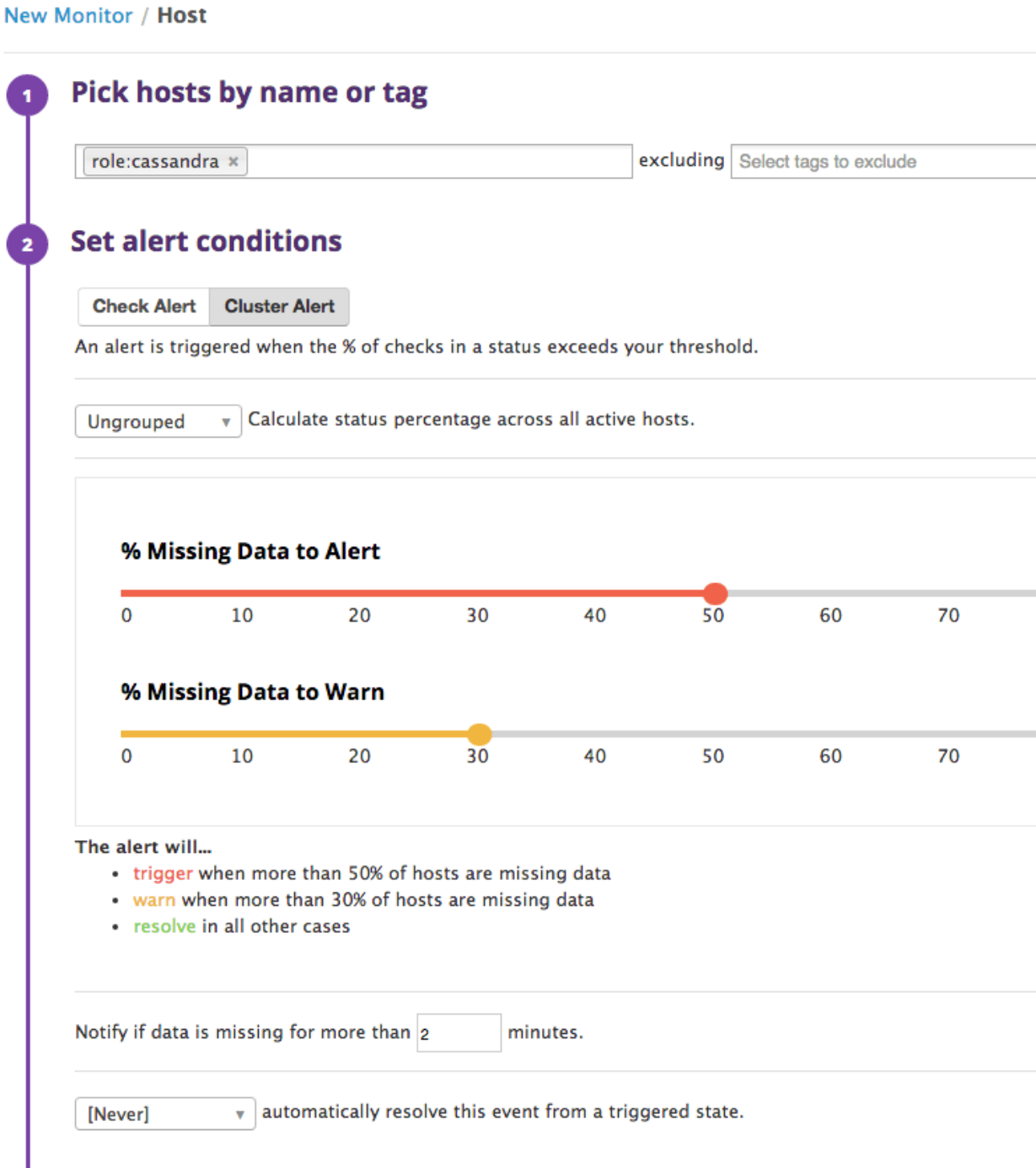
- You service promises to respond to clients within 50ms.
- A service you depend on promises that its error rate will be < 1%.
- On-call promises they will engage an incident within 15 minutes.

Promise enumeration

- Each team must formalise the promises they are willing to keep.
- They must also understand the promises they rely upon to function.
- When a promise you rely upon is broken, what should you do? Who should you contact?

Promises: defining “done”

Promises are done when they have a Datadog monitor (alert).



When Promises are broken…

- It is inevitable that a promises will be broken at some point.
- What to do when that happens?

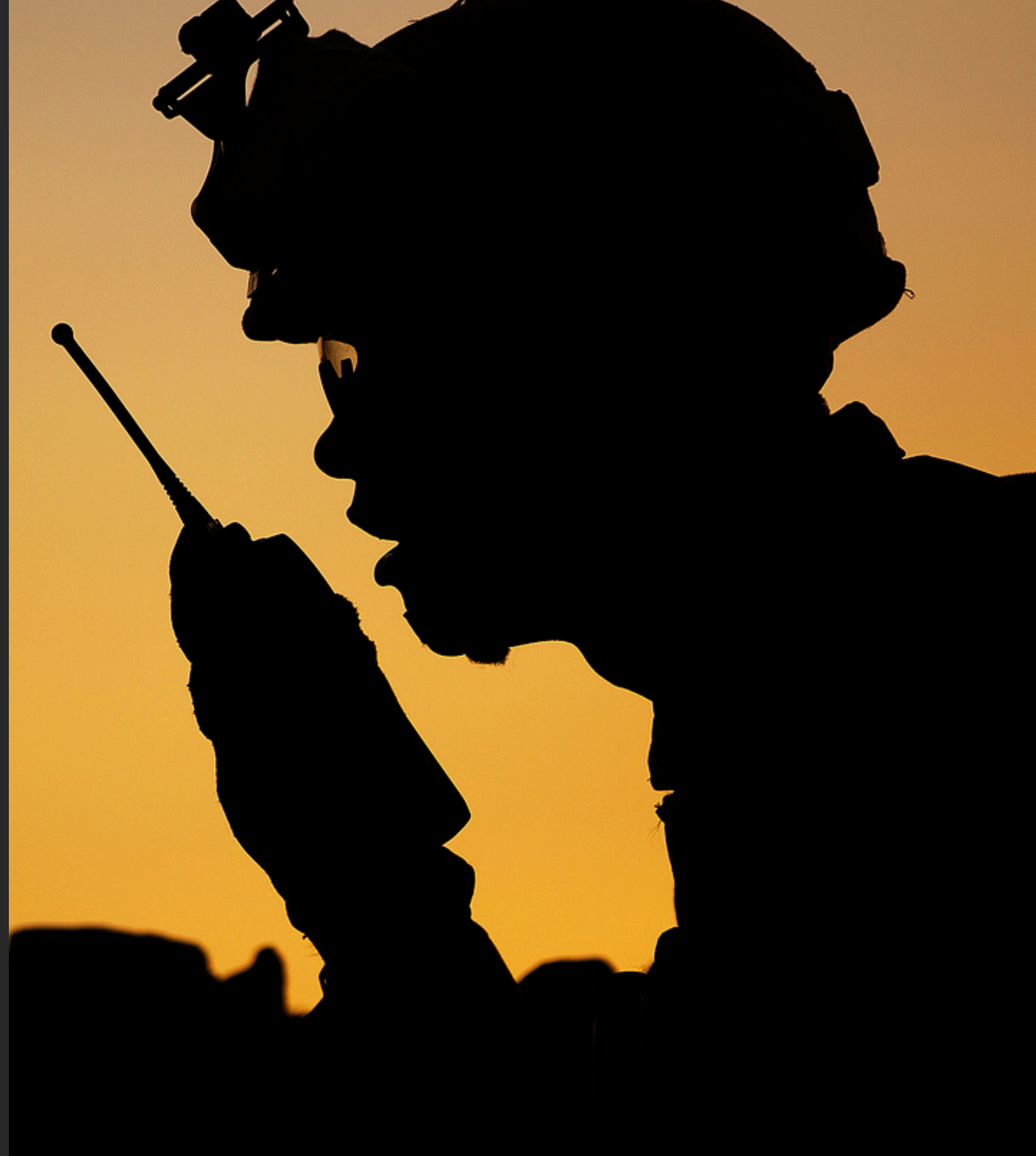
Blameless post-mortems

- Is “blameless post-mortem” a real thing?
- What about data-driven post-mortems?
- How does this relate to promises - specifically broken ones?

- `https://v.gd/jpr_post_mortems`
- `https://v.gd/jyee_datadriven`

Interpreting incidents

- Build a shared language.
- Practise communicating.
- Understand that incidents and outages are broken promises.



Measuring incident response

- Quantify and measure the quality of your incident responses.
- Quantitative: Time to detect, time to engage, time to fix.
- Qualitative: Quality of communication.

The end-game

- Have a clear answer for “why SRE?”
- Start with instrumentation - keep it simple to start.
- Enumerate your promises.
- Measure your response when promises are broken.

- Transparency
- Understanding
- Confidence

Thank you!

Niall O'Higgins
coinbase

Daniel Maher
DATADOG

AWS re:Invent

aws

Please complete the session survey in the mobile app.