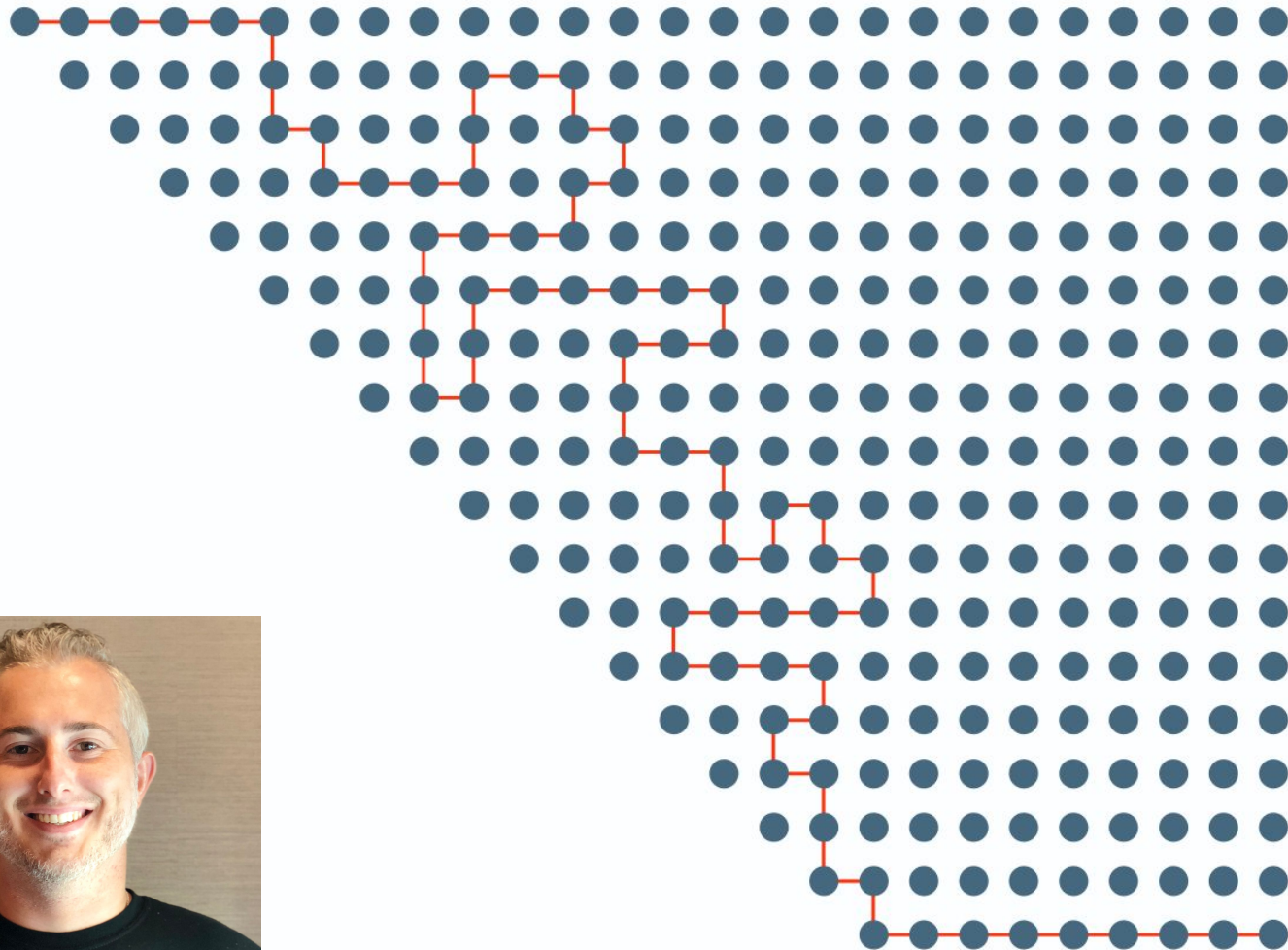


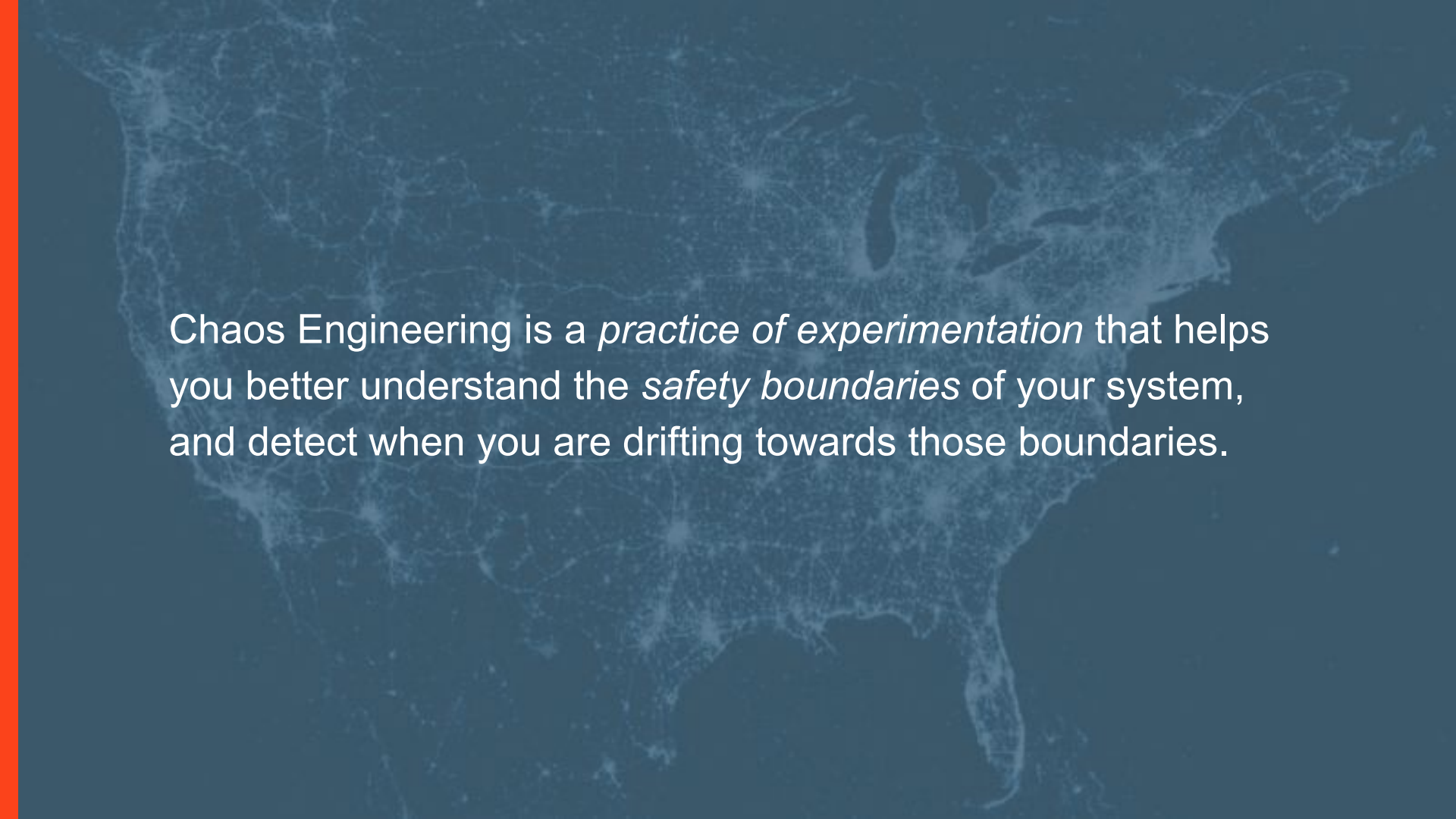


Courtney Nash
Internet Incident Librarian
Verica



Troy Koss
Director of SRE
Capital One





Chaos Engineering is a *practice of experimentation* that helps you better understand the *safety boundaries* of your system, and detect when you are drifting towards those boundaries.

Myth: Chaos Engineering is an "advanced capability."

You have to be at a certain level of sophistication with your systems in order to consider undertaking CE.



How It Started



How It's Going

Enterprise Complexity



Reactive

100% on or 100% off

Proactive

SLOs and Error Budgets

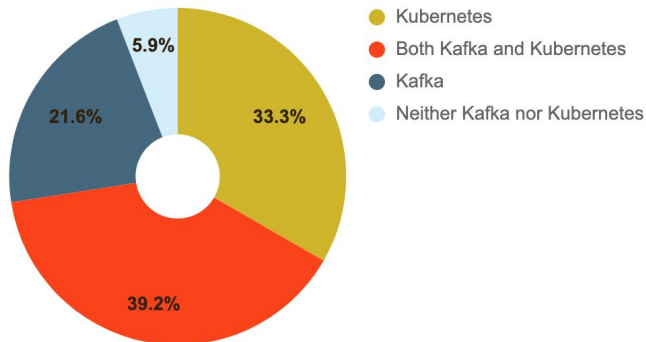
SRE

Clarity Through Chaos

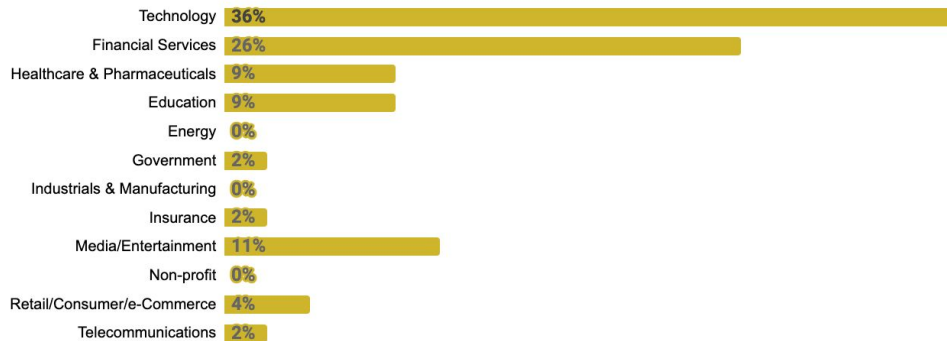


Who Is Ready for/Interested in CE?

What Are They Using?



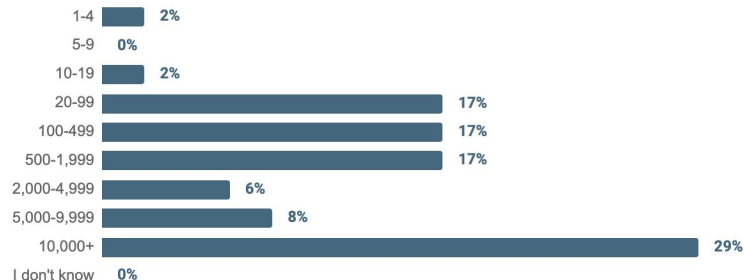
Industry



Role

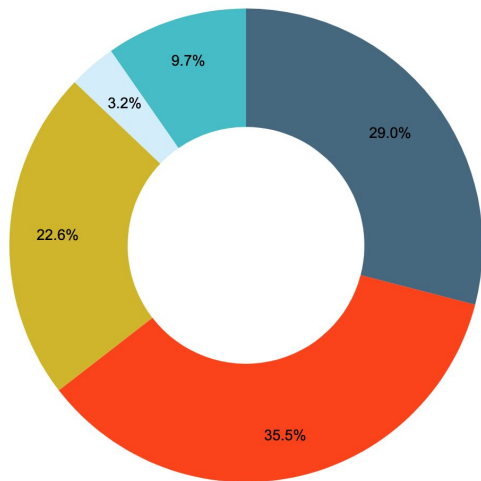


Organization Size



Operator Experience with Kubernetes and Kafka

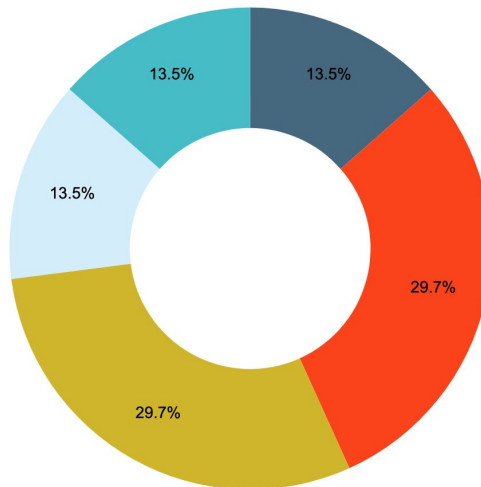
Kafka Experience Levels



64% Novice/Beginner

Kubernetes Experience Levels

● Novice
● Beginner
● Competent
● Proficient
● Expert



● Novice
● Beginner
● Competent
● Proficient
● Expert

43% Novice/Beginner

Myth: Chaos Engineering *introduces* (more) chaos into your system

"Our systems are chaotic enough as it is, why make it worse?"

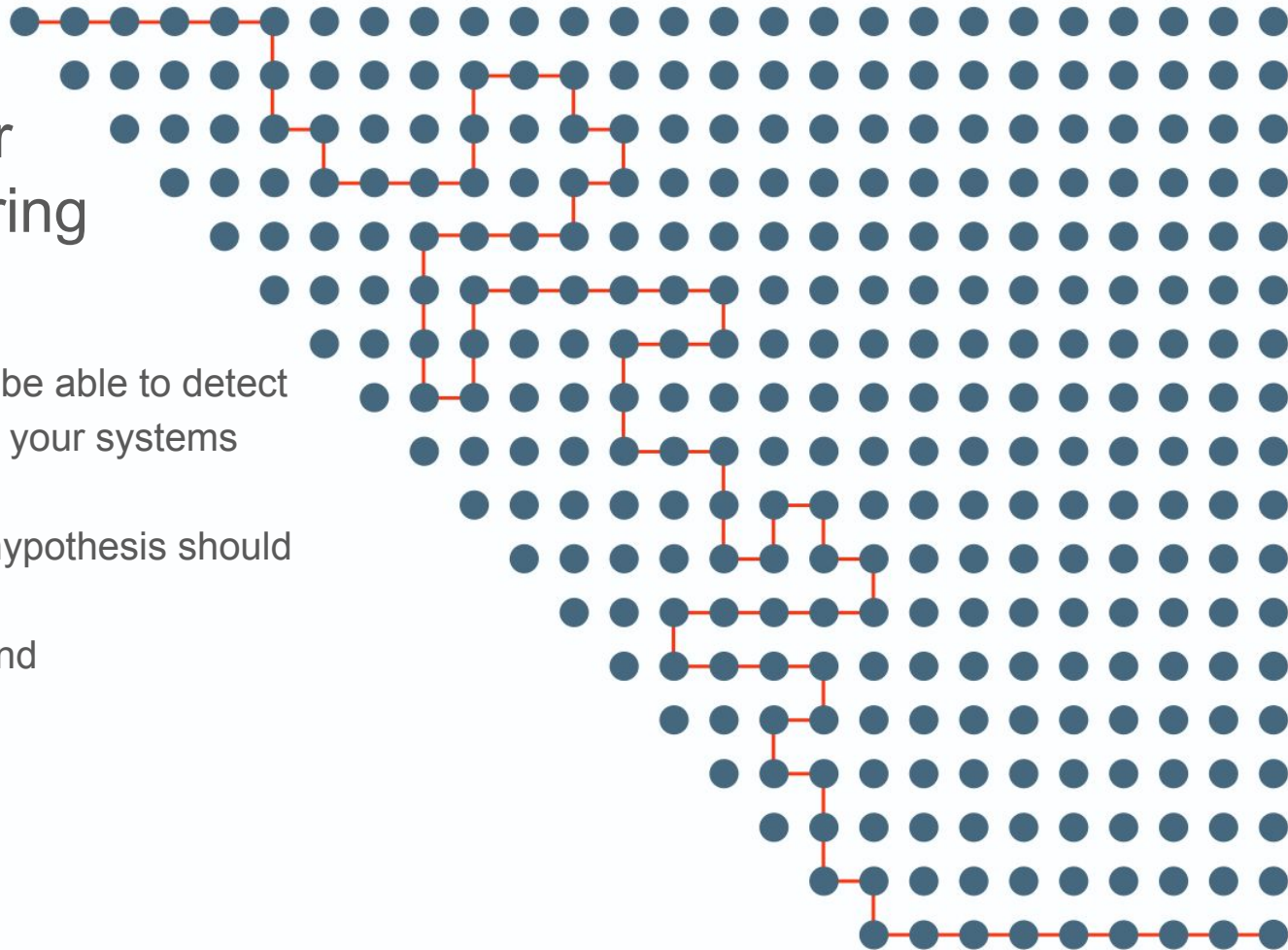


SLOs to the Rescue!



Prerequisites for Chaos Engineering

1. **Instrumentation** to be able to detect degraded state(s) in your systems
2. Social **awareness**
3. **Expectations** that hypothesis should be upheld
4. **Alignment** to respond



Social Awareness

You'll want buy in from anyone who touches or relies on any system you'd include in an experiment.

Running experiments without telling people who'd be involved/impacted will only create animosity and resistance to future CE efforts.



Hypotheses to Uphold

New knowledge is not generated if an experiment only confirms that a suspected broken component is indeed broken.

Fix what is *known* to be broken in a system you plan to experiment with before getting started with Chaos Engineering.



Basics get it done



Alignment to Respond

Your team or organization should be prepared to *do something* when a vulnerability or issue is found.



To properly support Chaos Engineering, your *cultural infrastructure* is as important as your technical infrastructure.



Here's the help we're looking for



Along with Chaos Engineering, I study failure in complex systems. I just launched something called the VOID, which is a database of public-facing incident reports. The goal is to facilitate sharing and learning from incidents, so we can all improve and help make software safer and more reliable. We're asking everyone to contribute their incident reports to it, so we can do more research, please join us! <https://www.thevoid.community>



Excited to see you get involved with Chaos engineering on your own projects!

Looking for Software and Site Reliability Engineers to work with teams across the organization to build and maintain auditable, secure, performant, reliable software! Reachout on LinkedIn! <https://www.linkedin.com/in/troykoss/>

