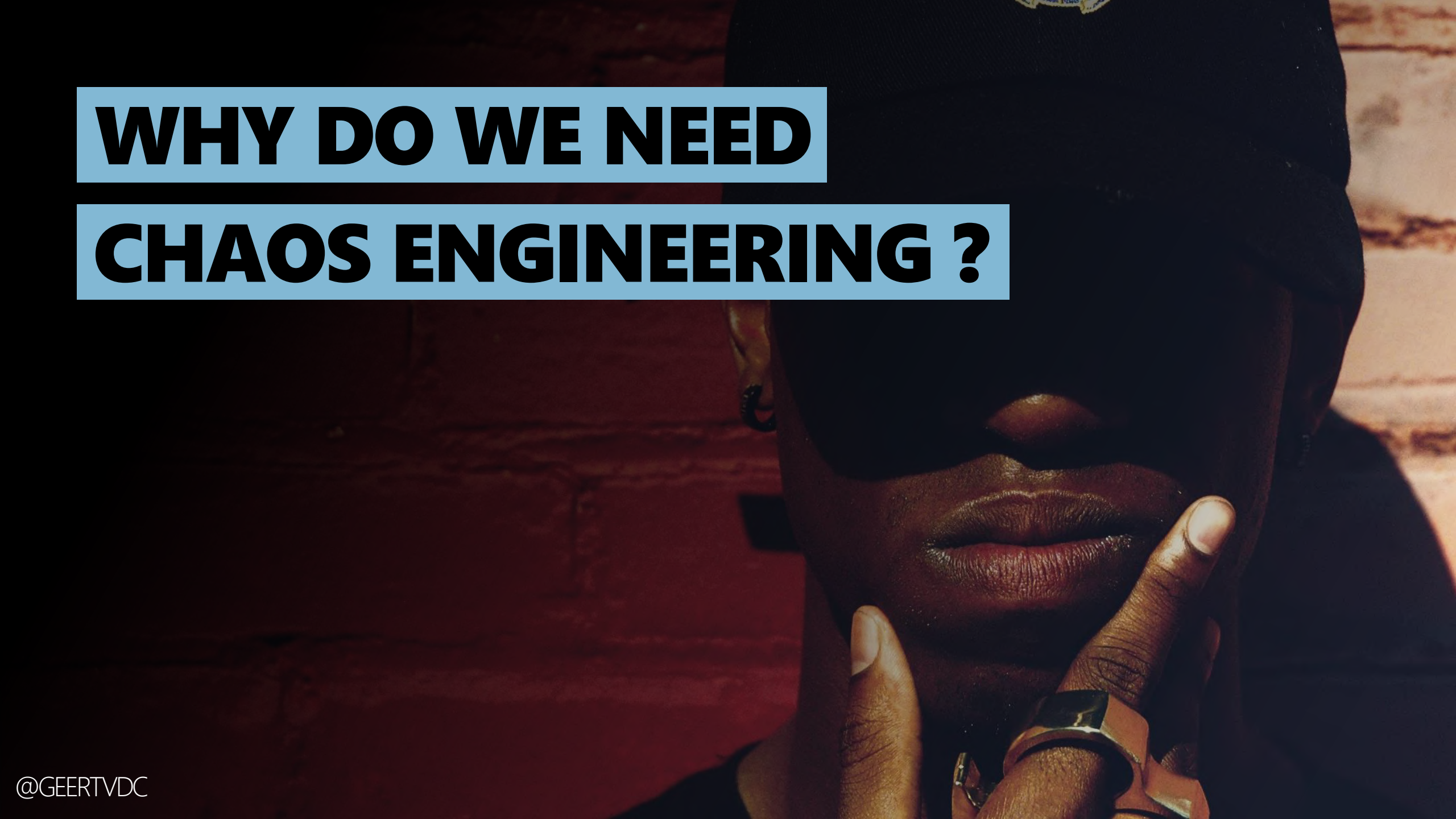
CHAOS ENGINEERING

THE FINE ART OF BREAKING STUFF IN PRODUCTION ON PURPOSE

GEERT VAN DER CRUIJSEN
@GEERTVDC


Xpirit
#DOEPICSHIT
@GEERTVDC
GEERT VAN DER CRUIJSEN
CLOUD NATIVE ARCHITECT
DEVOPS COACH
FULL CYCLE DEVELOPER


WHY DO WE NEED
CHAOS ENGINEERING ?
@GEERTVDC

11:11

businessinsider.com

BUSINESS INSIDER

Amazon's one hour of downtime on Prime Day may have cost it up to $100 million in lost sales

Sean Wolfe Jul 19, 2018, 10:53 AM ET

Sean Gallup/Getty Images**An Amazon warehouse.**

- **Amazon's website had consistent issues during Prime Day, the website's biggest sale event of the year.**
- **Shoppers had issues connecting to the website for over an hour on Monday, causing many to threaten to cancel their Prime subscription.**

08:48

https://www.datacenterknowledge.com/

informa

MENU DataCenter Knowledge SEARCH

RECENT

Travelers wait in line at the Delta check-in counter at LaGuardia Airport August 8, 2016 in the Queens borough of New York City. Delta flights around the globe were grounded and delayed on Monday morning due to a system outage. (Photo by Drew Angerer/Getty Images)

BUSINESS

Delta: Data Center Outage Cost Us $150M

Price tag extraordinarily high in comparison to data center industry average

Yevgeniy Sverdlik | Sep 08, 2016

08:46

https://status.slack.com/2018-06/142ec

slack Status

Dashboard History Twitter

Wednesday June 27 - Wednesday July 11, 2018

Outage

Connectivity issues affecting all workspaces

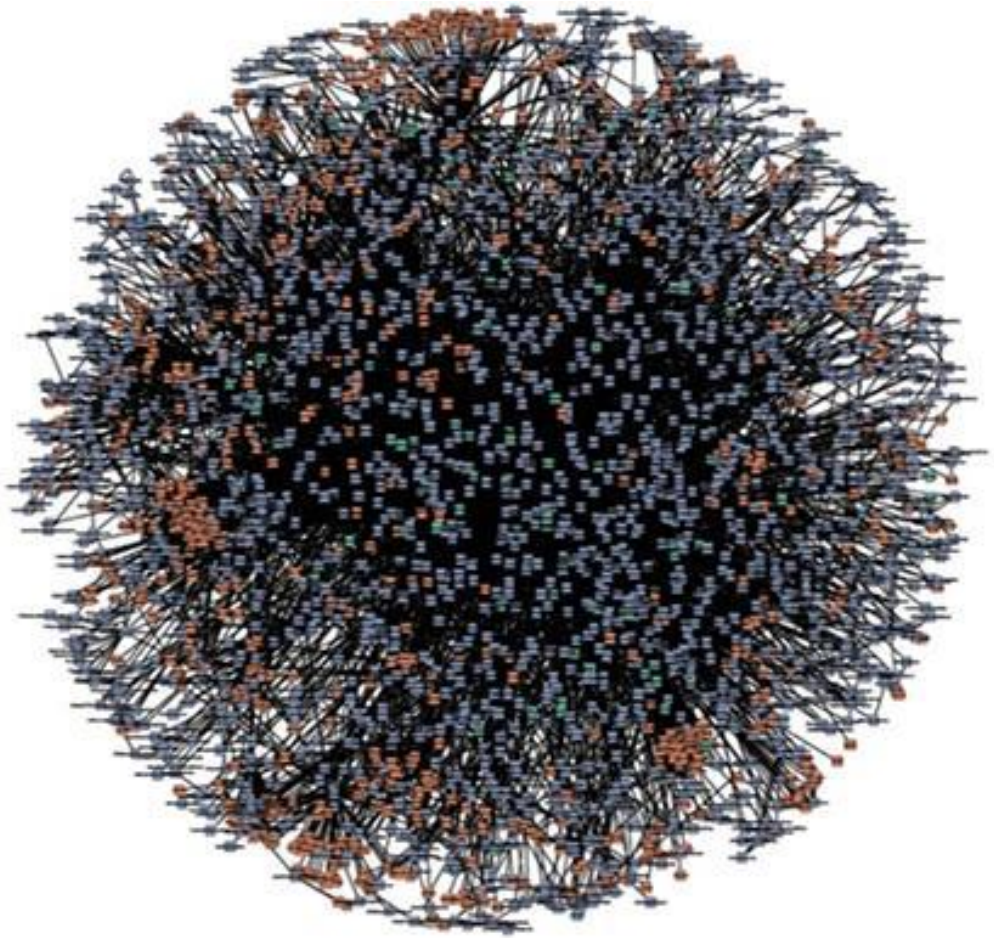
Now that we've conducted our postmortem and root cause analysis, we know the event summarized below started at 6:17 a.m. PDT (7/11/18).

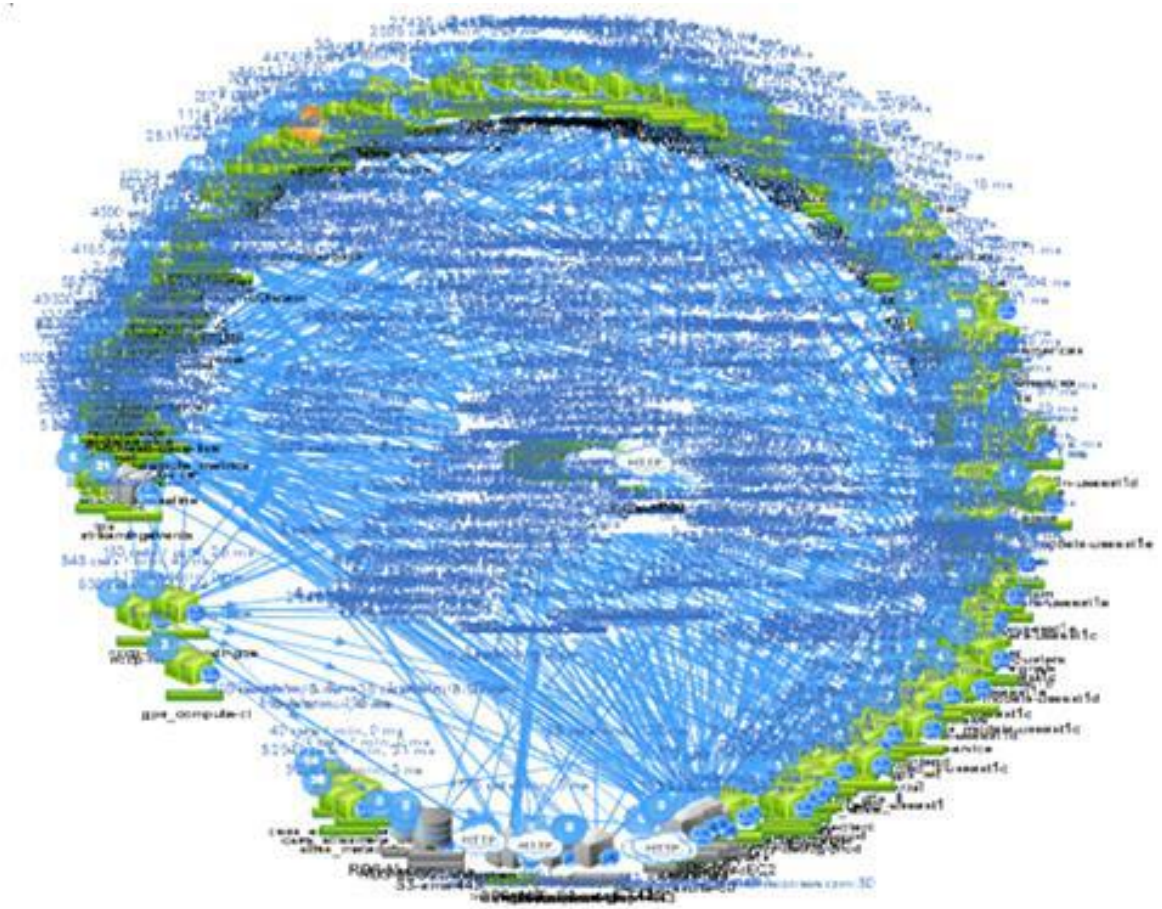
Jul 12, 2:48 AM GMT+3

Thank you for your patience. Everything is back up and running and we wanted to give you a summary of what happened.

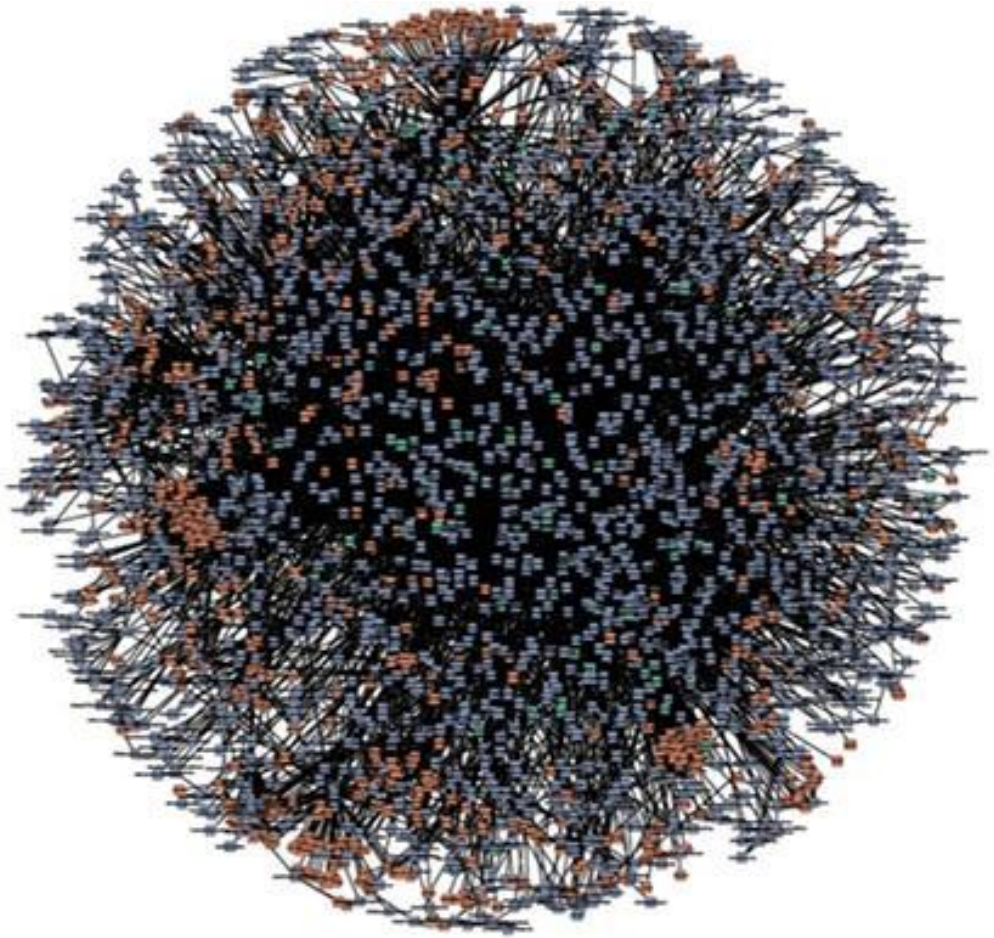
On June 27th (yesterday) between 6:33 a.m. and 9:49 a.m. PDT Slack experienced an outage where people could not connect to their workspaces. The network problems were caused by a bug included in an offline batch process of data, which resulted in unexpected network spikes and led all of our customers to become disconnected and unable to reconnect.

Once we identified the problem, we restricted new connections and provisioned extra capacity. At 9:24 a.m. PST, production was healthy enough to remove restrictions and by 9:44 a.m. PST, all customers had

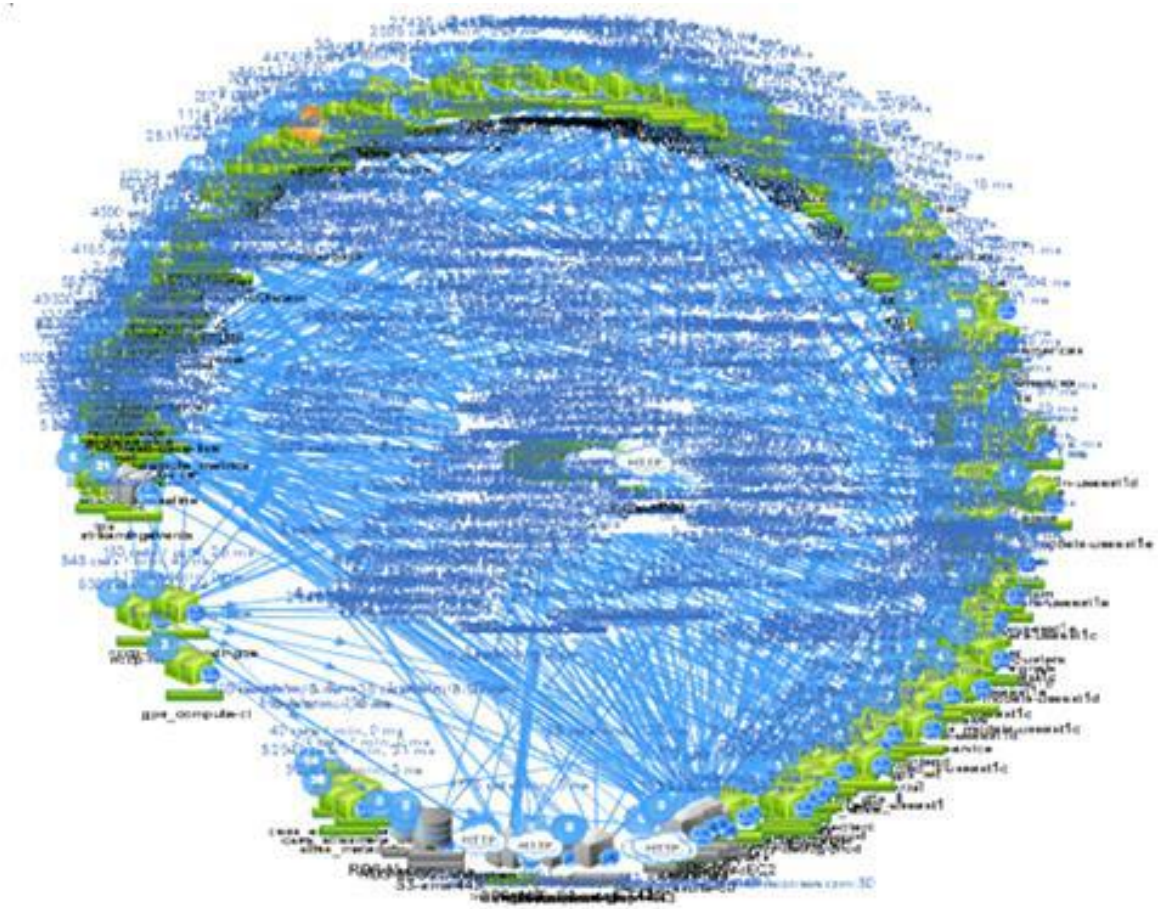




@GEERTVDC



amazon.com



NETFLIX

"IN A COMPLEX LANDSCAPE YOUR APPLICATION IS NEVER FULLY UP"

@GEERTVDC


TRADITIONAL MONITORING
TOOLS ARE DEAD!
@GEERTVDC


MEASURE
USER IMPACT


FRESHNESS
DURABILITY
CORRECTNESS
QUALITY
THROUGHPUT
COVERAGE
AVAILABILITY
LATENCY
RELIABILITY
MEASURE
USER IMPACT


PEOPLE
APPLICATION
NETWORK
INFRASTRUCTURE
RESILIENT APPLICATIONS

@GEERTVDC

FAIL OPEN

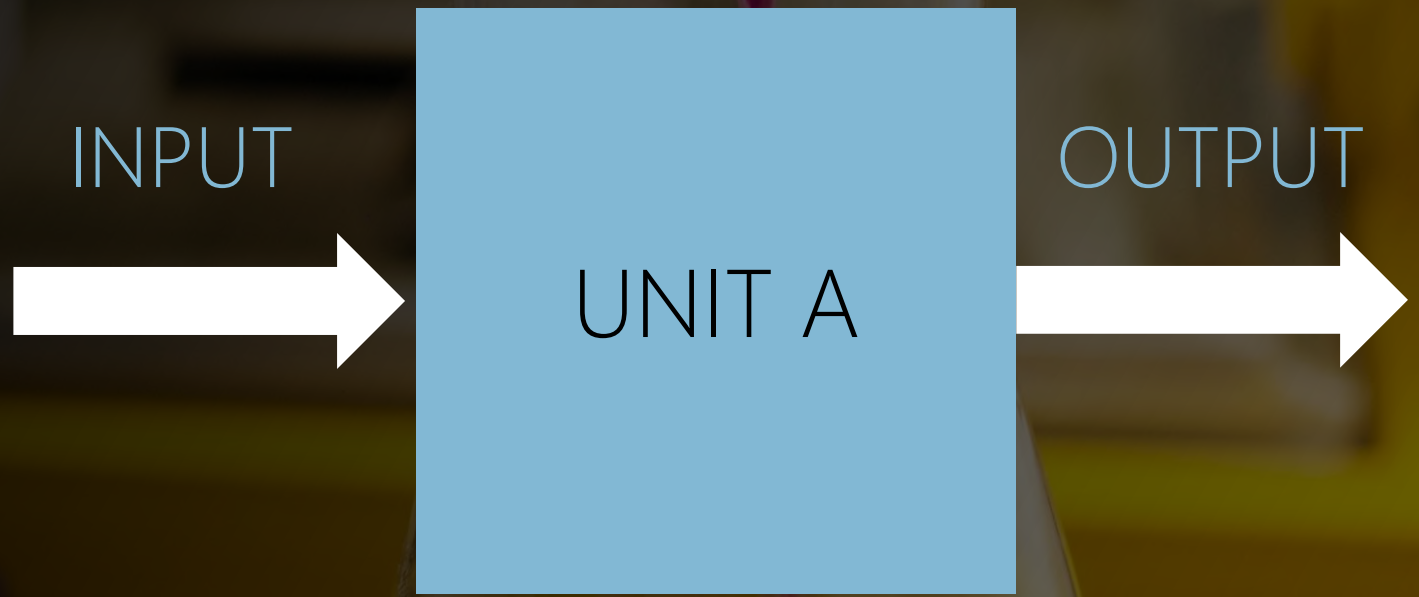
GRACEFUL DEGRADATION

@GEERTVDC


BUT WE DO TESTS?
@GEERTVDC

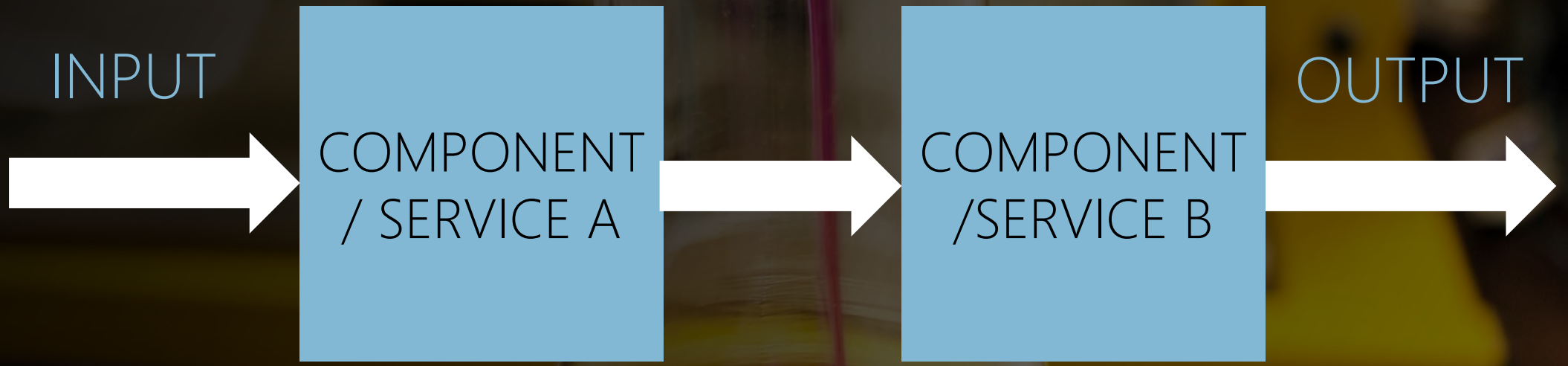
BUT WE DO TESTS?

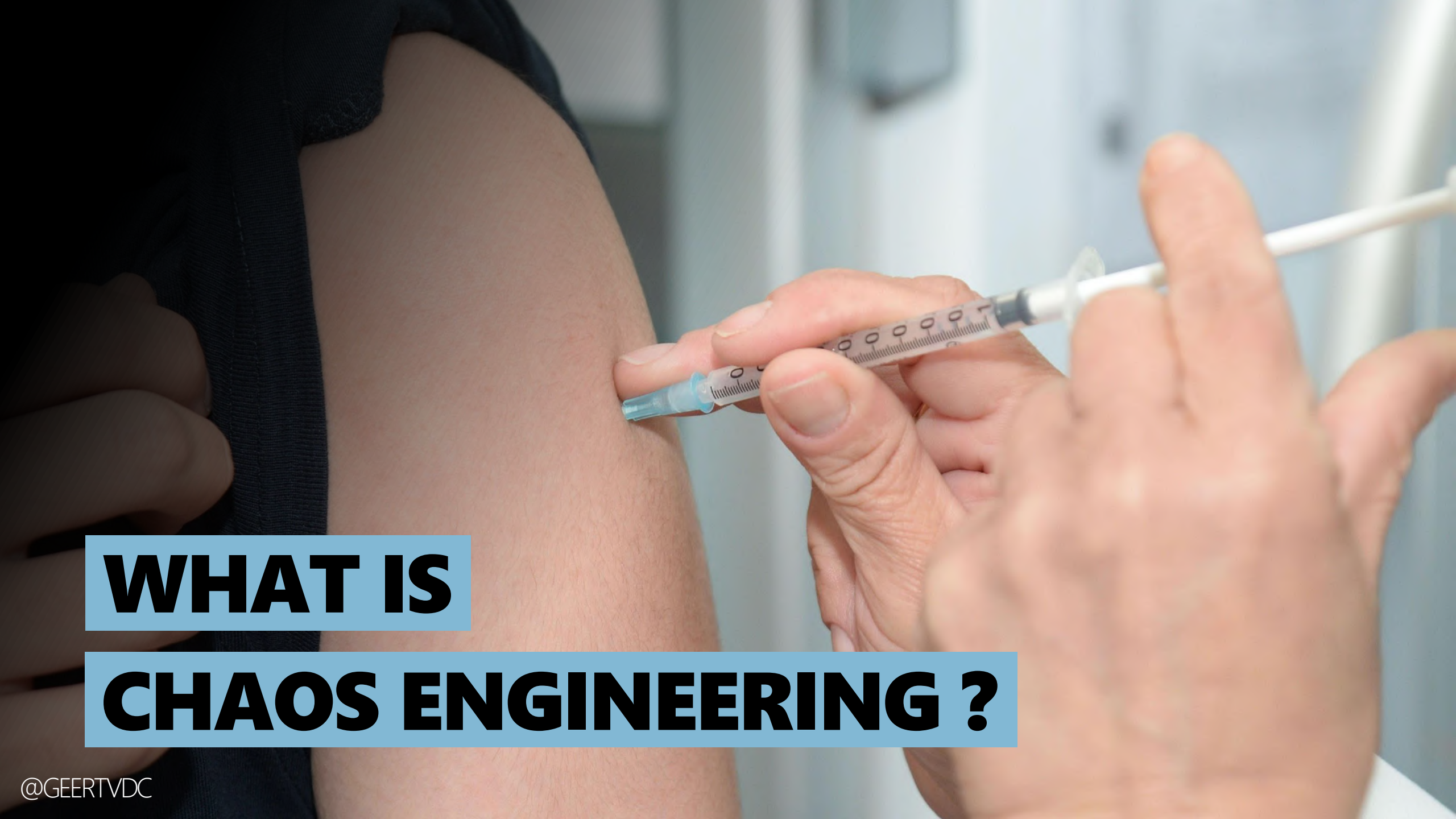
UNIT TESTS



BUT WE DO TESTS?

INTEGRATION TESTS





WHAT IS CHAOS ENGINEERING ?

@GEERTVDC


CHAOS ENGINEERING
IS NOT
RANDOMLY BREAKING
STUFF IN PRODUCTION
@GEERTVDC

CHAOS ENGINEERING

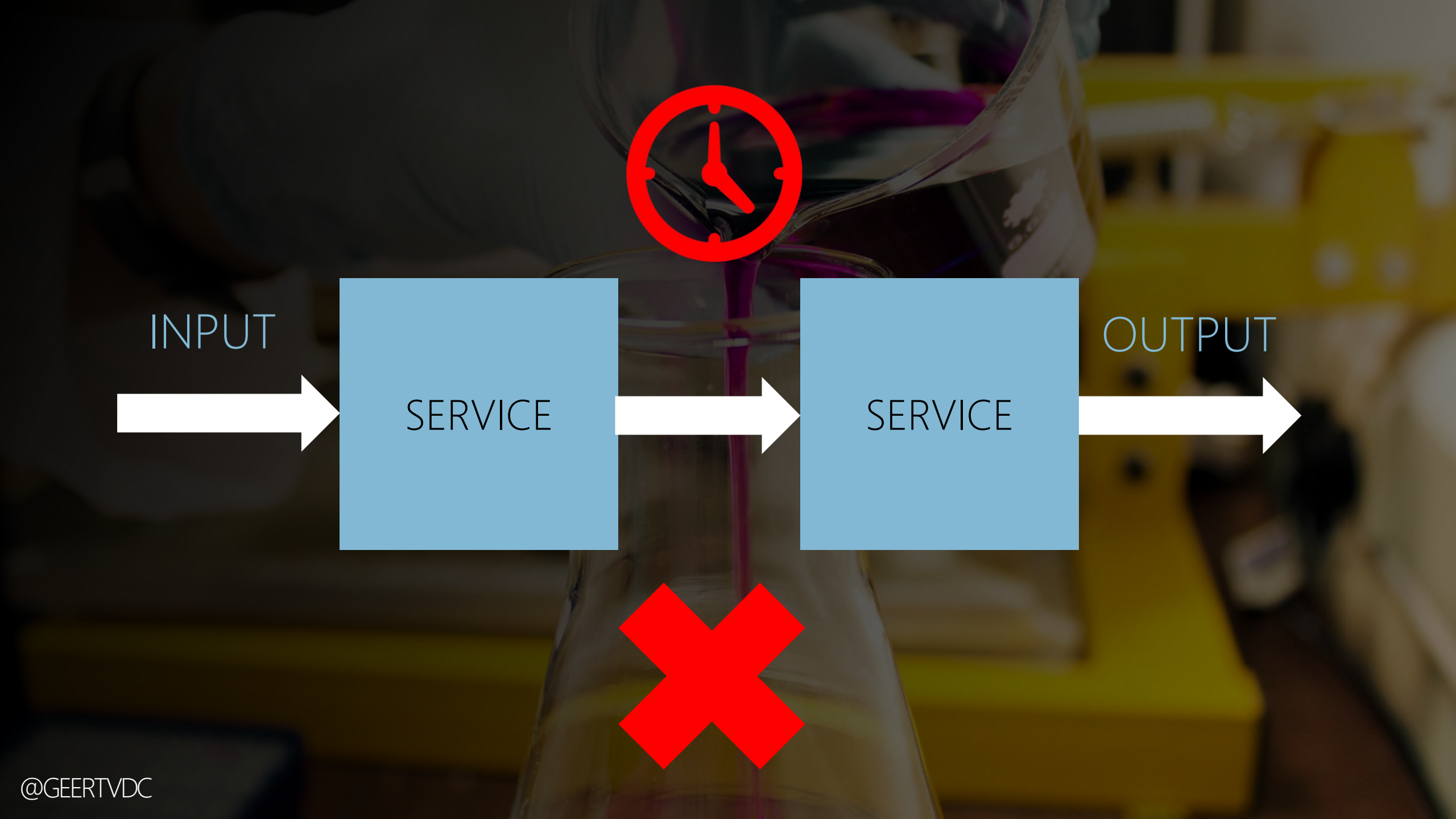
"Chaos Engineering is the discipline of experimenting on a distributed system in order to build confidence in the system's capability to withstand turbulent conditions in production."

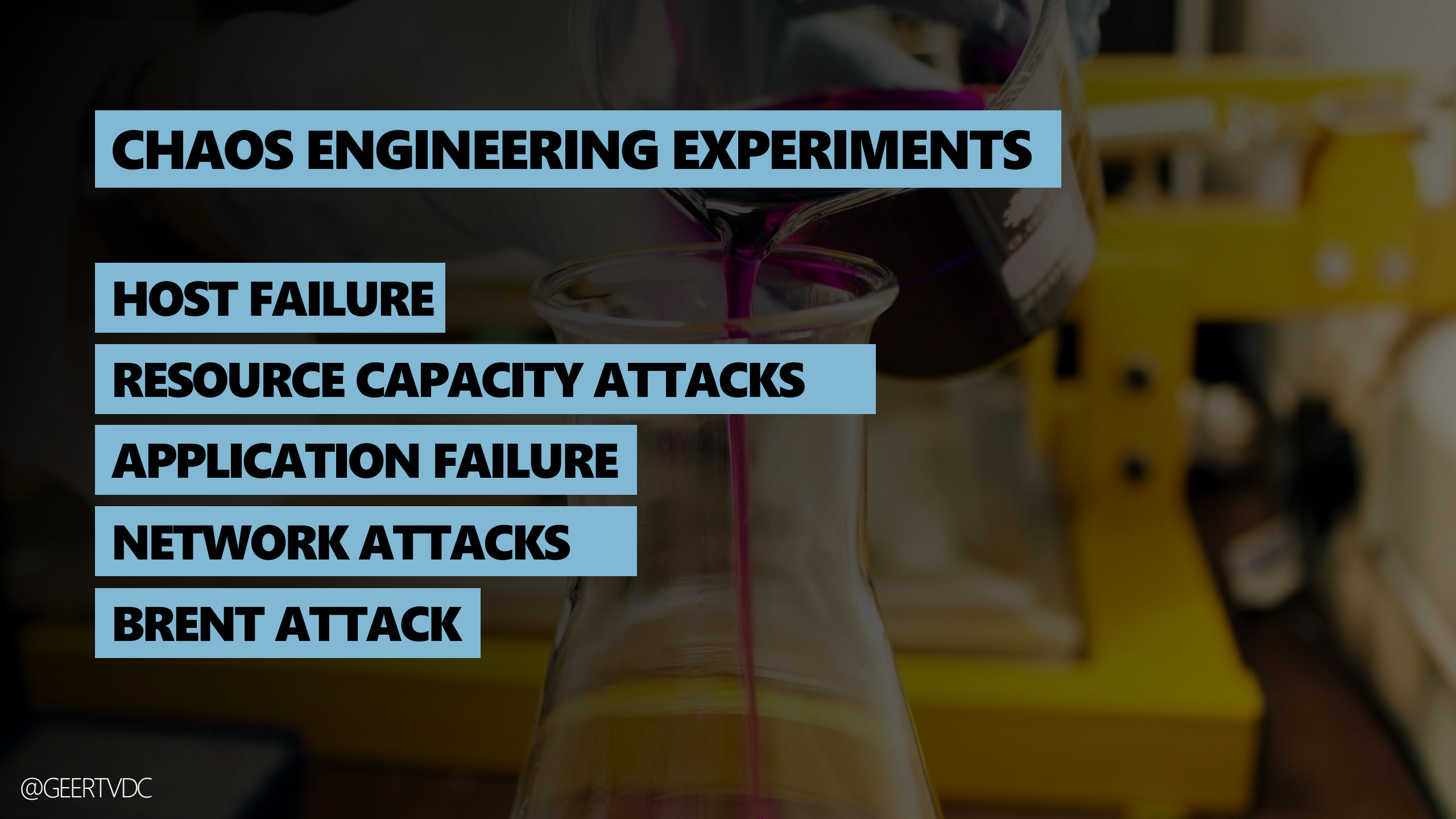
https//principlesofchaos.org

CHAOS ENGINEERING

"Chaos Engineering is the discipline of ***experimenting*** *on a distributed system in order to* ***build confidence*** *in the system's capability to* ***withstand turbulent conditions*** *in production."*

https//principlesofchaos.org


INPUT
SERVICE
SERVICE
OUTPUT

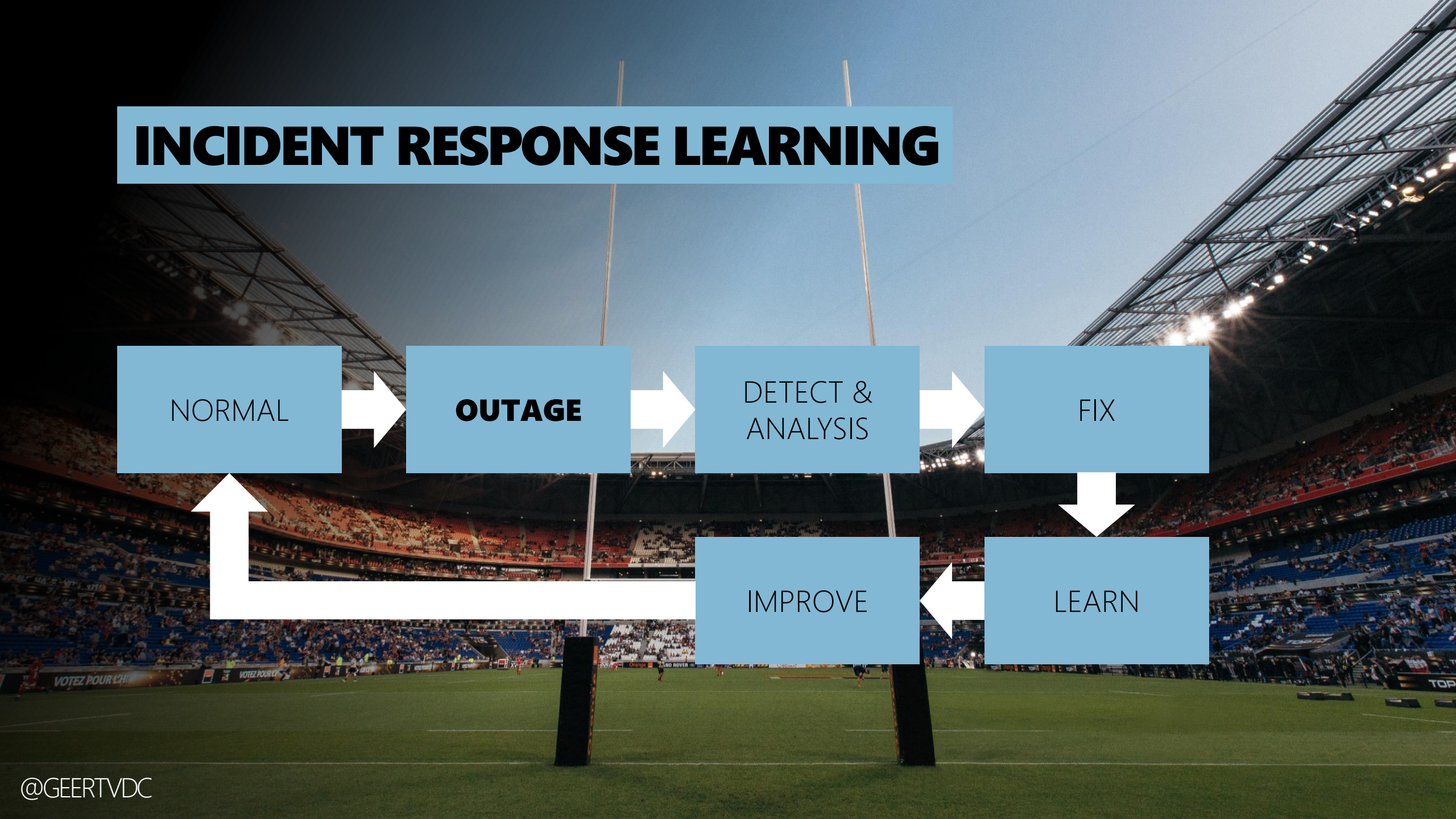

CHAOS ENGINEERING EXPERIMENTS
HOST FAILURE
RESOURCE CAPACITY ATTACKS
APPLICATION FAILURE
NETWORK ATTACKS
BRENT ATTACK
@GEERTVDC

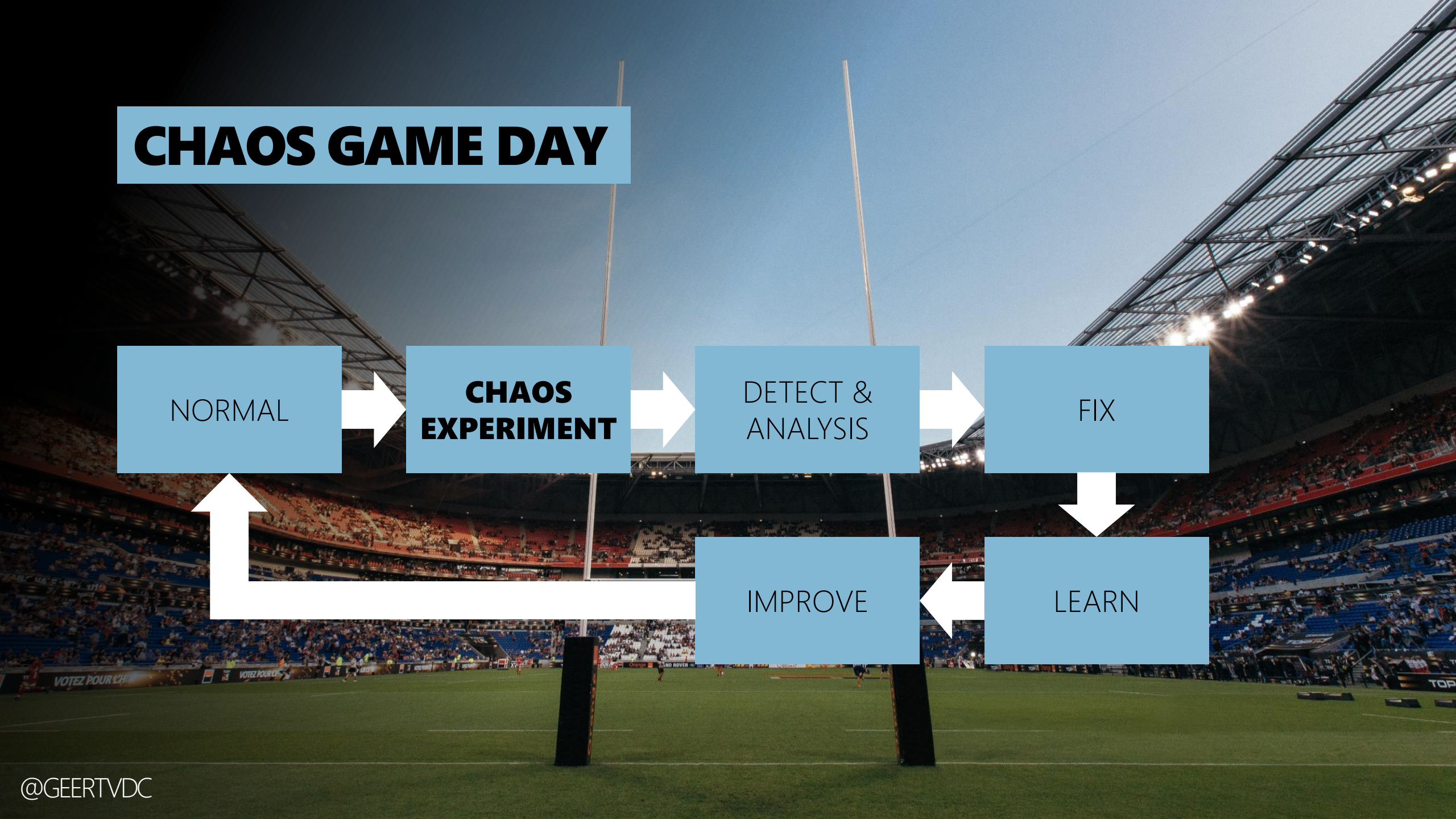
CHAOS ENGINEERING

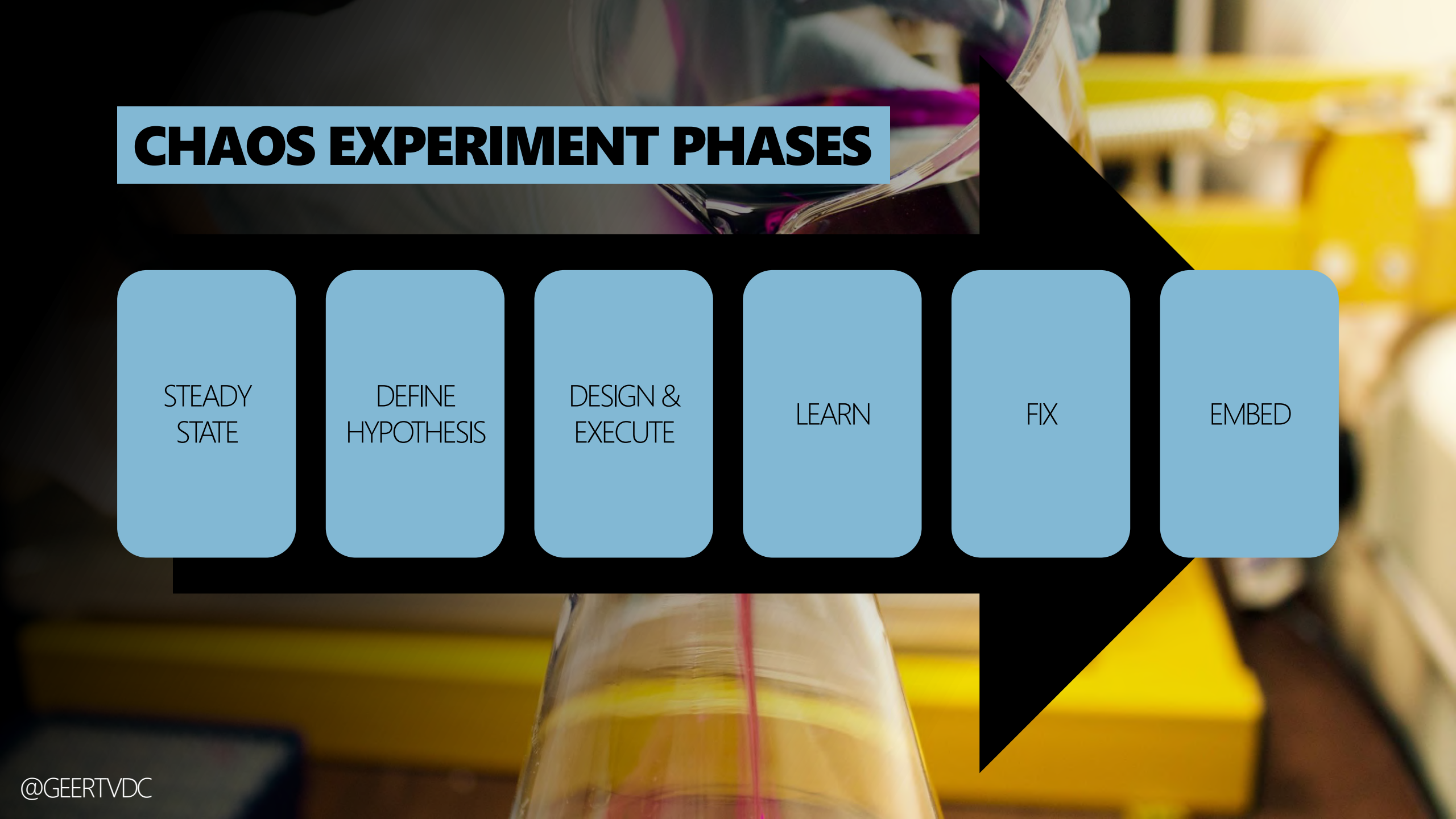
ONLY IN PRODUCTION?


HOW TO START
YOUR FIRST EXPERIMENT
@GEERTVDC


GAME DAY
@GEERTVDC

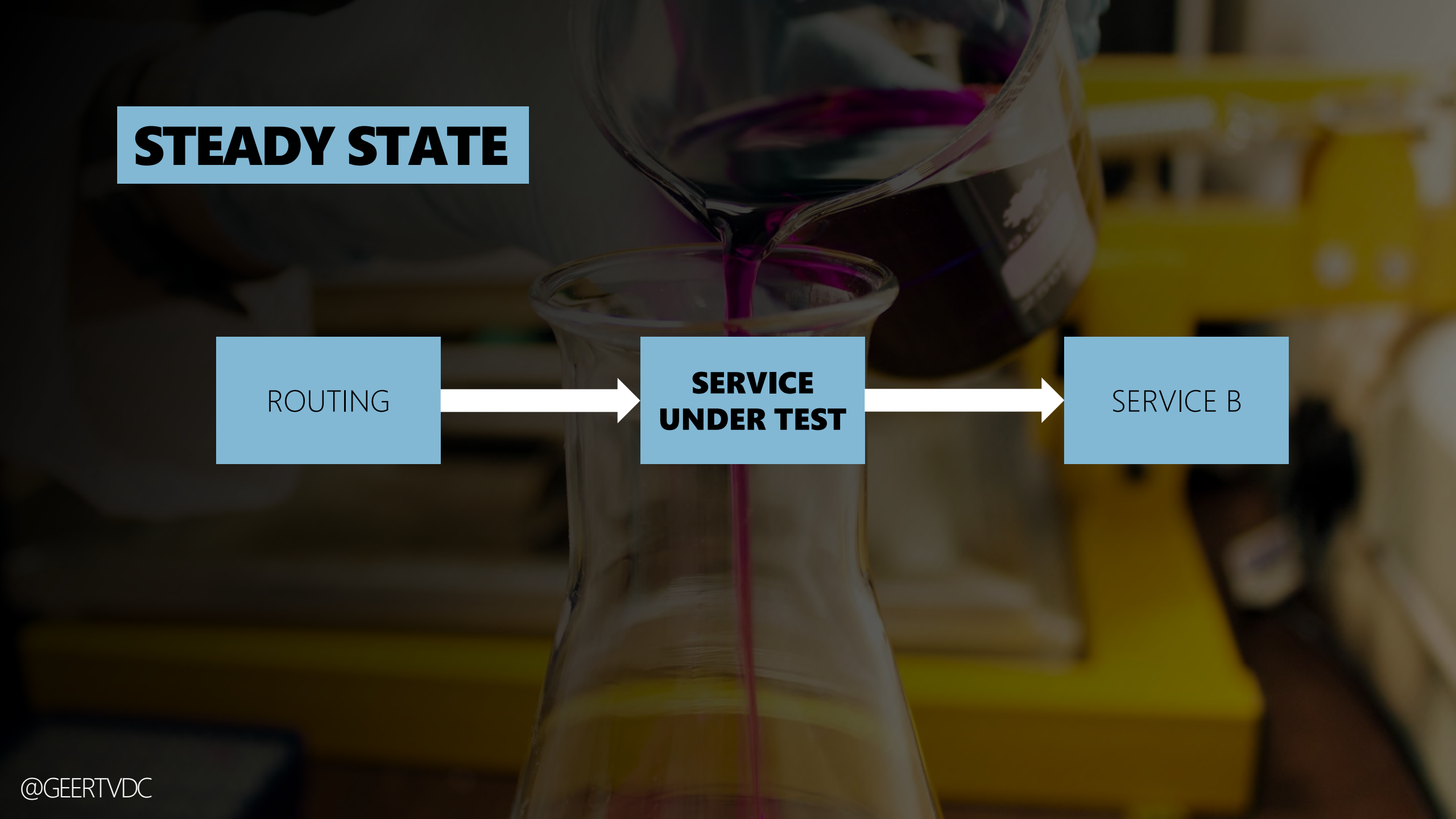

INCIDENT RESPONSE LEARNING
NORMAL
OUTAGE
DETECT & ANALYSIS
FIX
LEARN
IMPROVE
@GEERTVDC


CHAOS GAME DAY
NORMAL
CHAOS EXPERIMENT
DETECT & ANALYSIS
FIX
LEARN
IMPROVE
@GEERTVDC

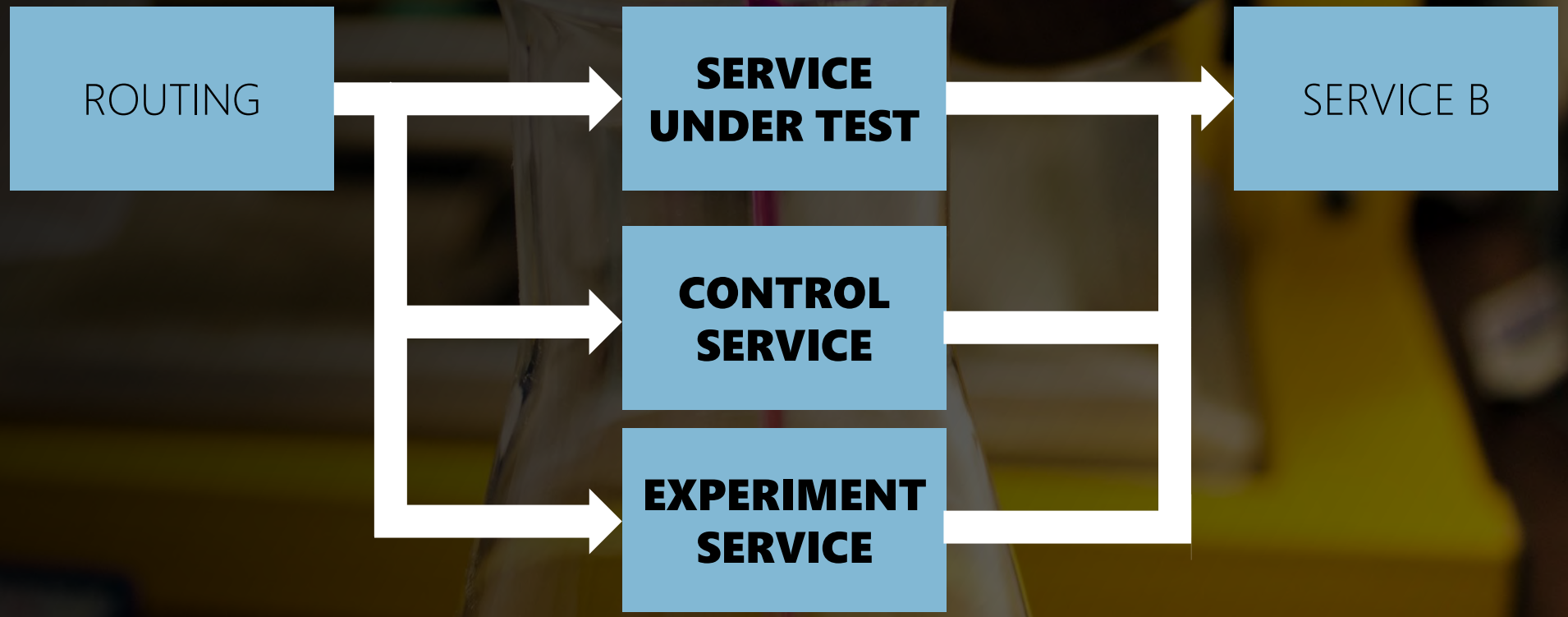

CHAOS EXPERIMENT PHASES
STEADY STATE
DEFINE HYPOTHESIS
DESIGN & EXECUTE
LEARN
FIX
EMBED
@GEERTVDC


STEADY STATE
STEADY STATE
DEFINE HYPOTHESIS
DESIGN & EXECUTE
LEARN
FIX
EMBED
@GEERTVDC

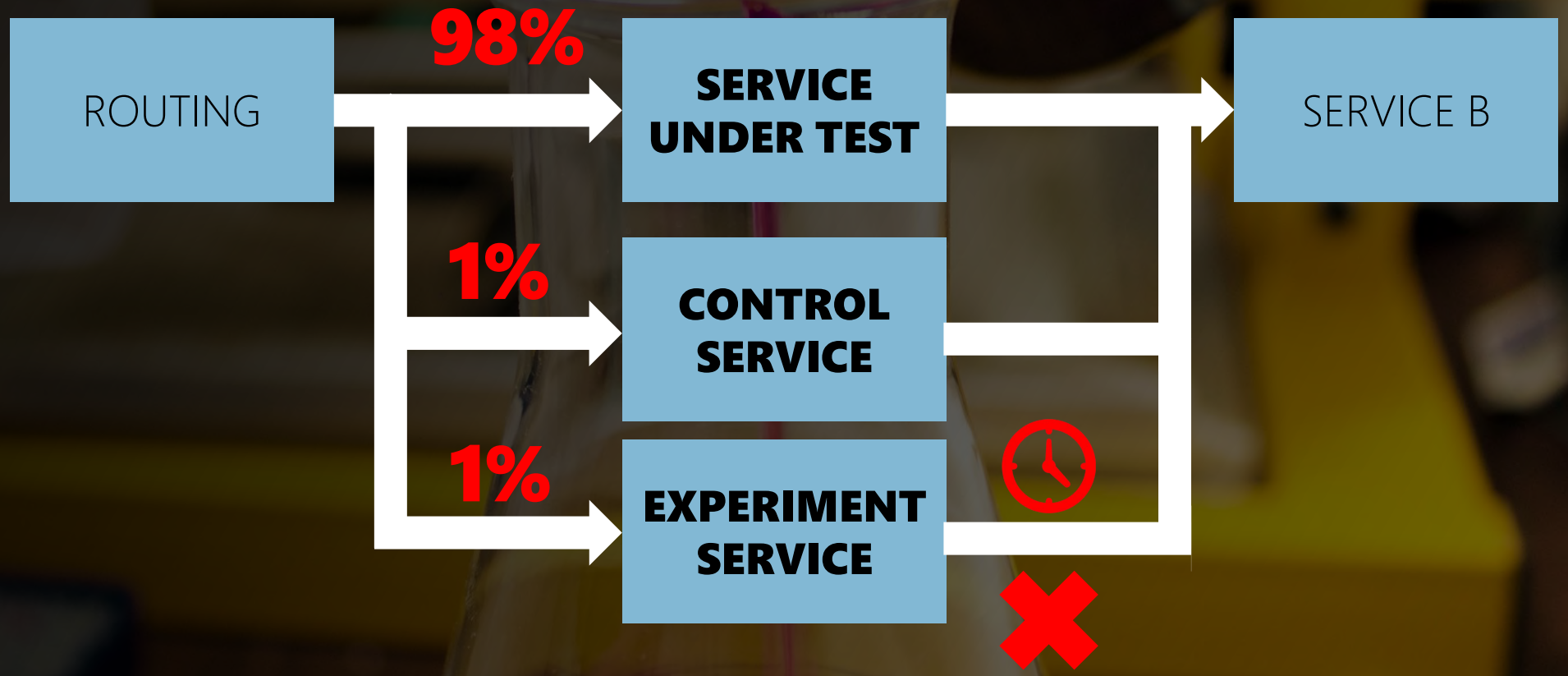

STEADY STATE
MEASURE BUSINESS METRICS
100ms extra load time drop Amazon's sale by 1%
@GEERTVDC


STEADY STATE
ROUTING
SERVICE UNDER TEST
SERVICE B

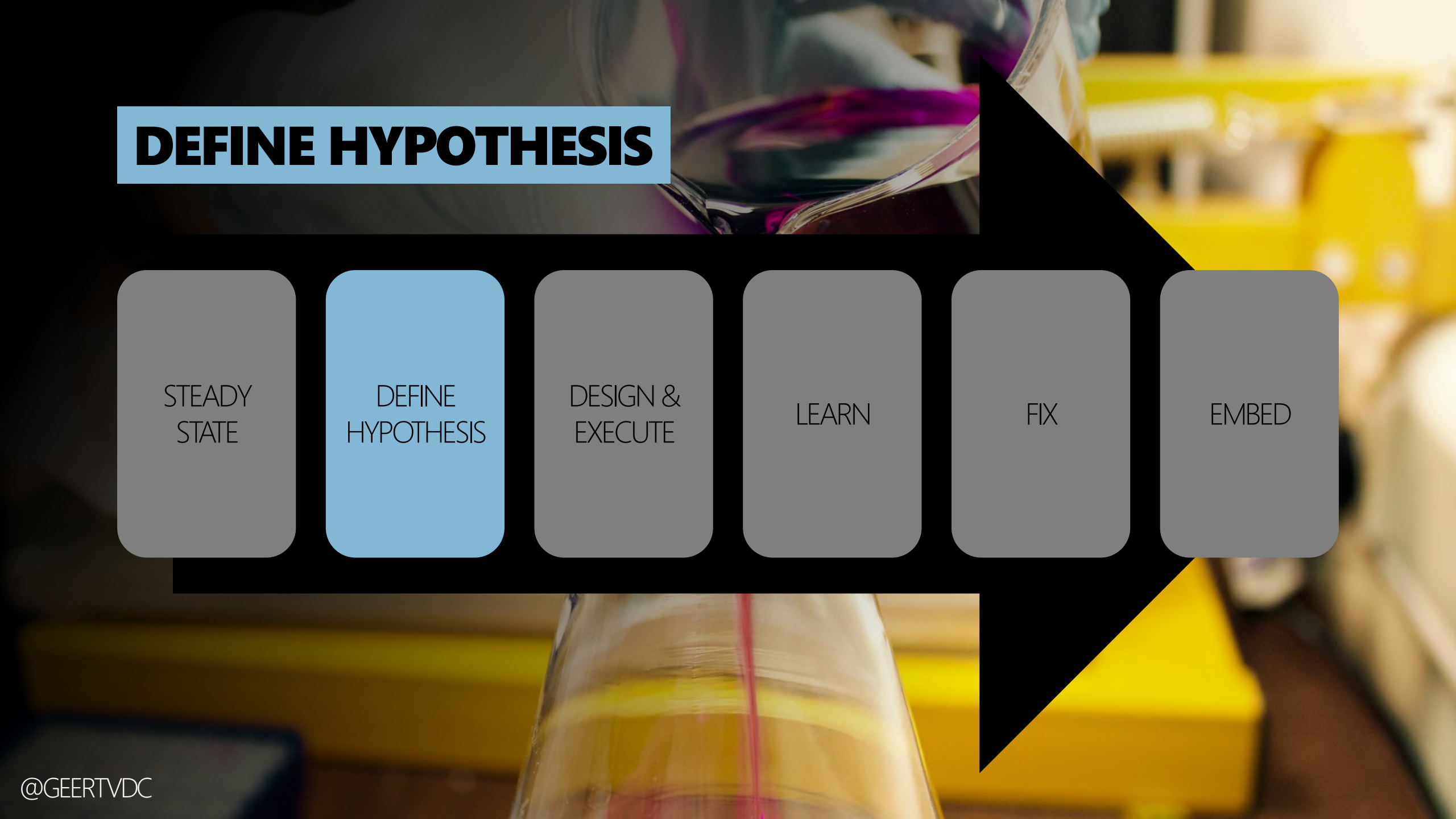
STEADY STATE

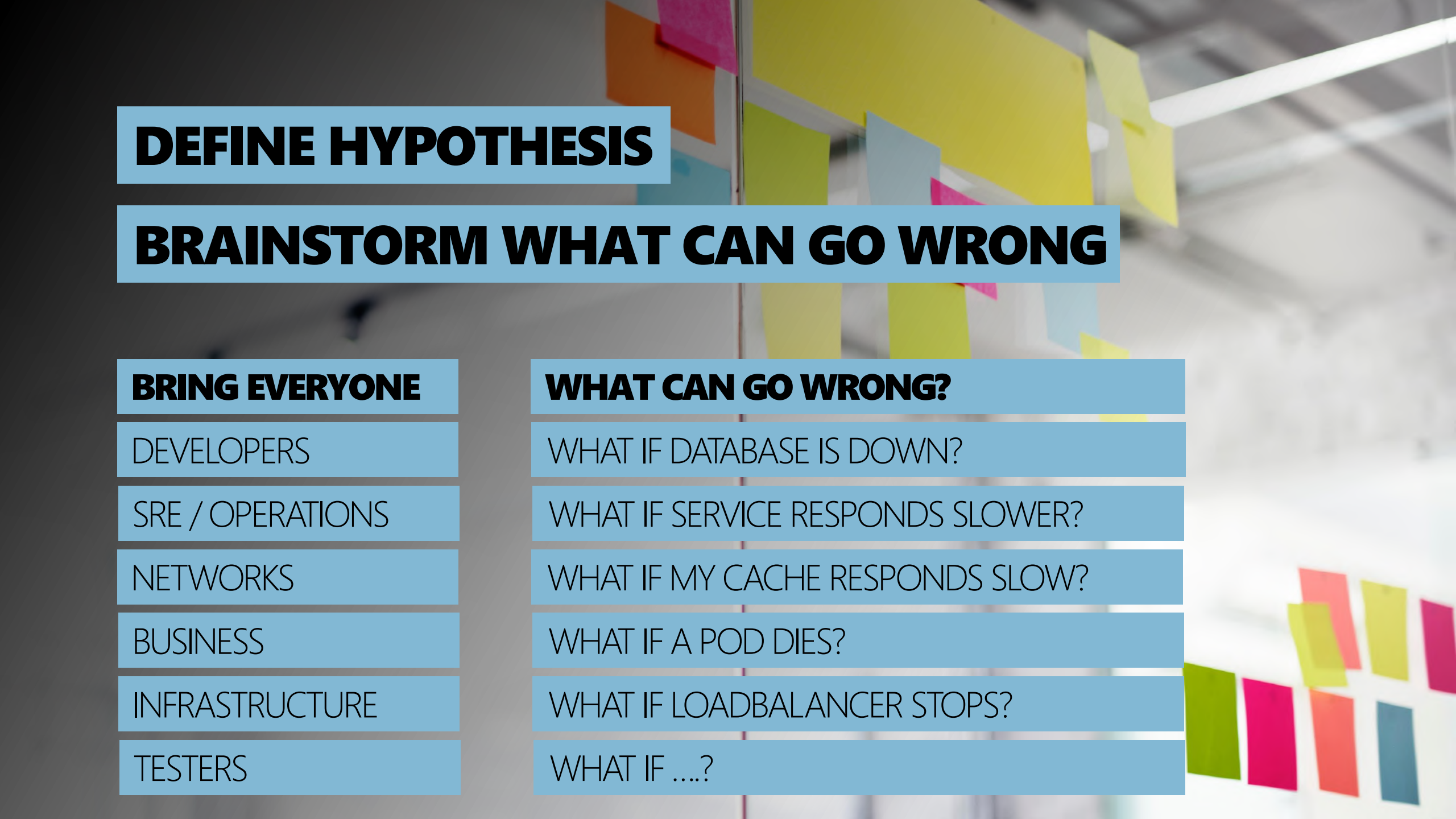


STEADY STATE




ALWAYS BE ABLE TO ABORT
@GEERTVDC


DEFINE HYPOTHESIS
STEADY STATE
DEFINE HYPOTHESIS
DESIGN & EXECUTE
LEARN
FIX
EMBED



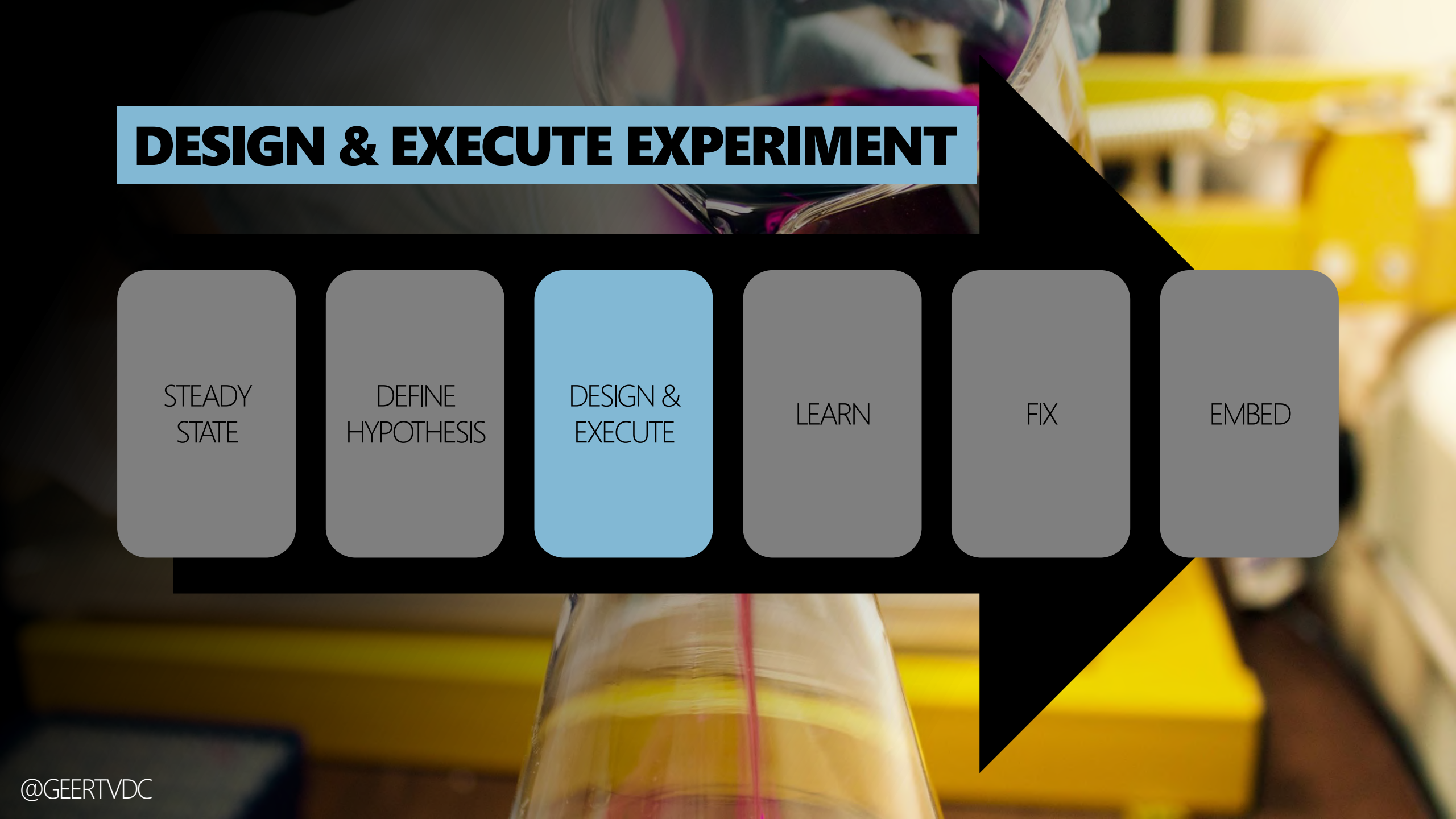
DEFINE HYPOTHESIS

BRAINSTORM WHAT CAN GO WRONG

BRING EVERYONE
DEVELOPERS
SRE / OPERATIONS
NETWORKS
BUSINESS
INFRASTRUCTURE
TESTERS

WHAT CAN GO WRONG?
WHAT IF DATABASE IS DOWN?
WHAT IF SERVICE RESPONDS SLOWER?
WHAT IF MY CACHE RESPONDS SLOW?
WHAT IF A POD DIES?
WHAT IF LOADBALANCER STOPS?
WHAT IF ….?


STOP
STOP IF YOU KNOW THE EXPERIMENT WILL BREAK
@GEERTVDC


DESIGN & EXECUTE EXPERIMENT
STEADY STATE
DEFINE HYPOTHESIS
DESIGN & EXECUTE
LEARN
FIX
EMBED

DESIGN & EXECUTE EXPERIMENT

START SMALL

NOTIFY PEOPLE INVOLVED

SLOWLY INCREASE BLAST RADIUS

TOOLS:

GREMLIN.COM
CHAOSTOOLKIT.ORG
GITHUB.COM/NETFLIX/SIMIANARMY
GITHUB.COM/ASOBTI/KUBE-MONKEY

GREMLIN

ChaosToolkit


LEARN
STEADY STATE
DEFINE HYPOTHESIS
DESIGN & EXECUTE
LEARN
FIX
EMBED

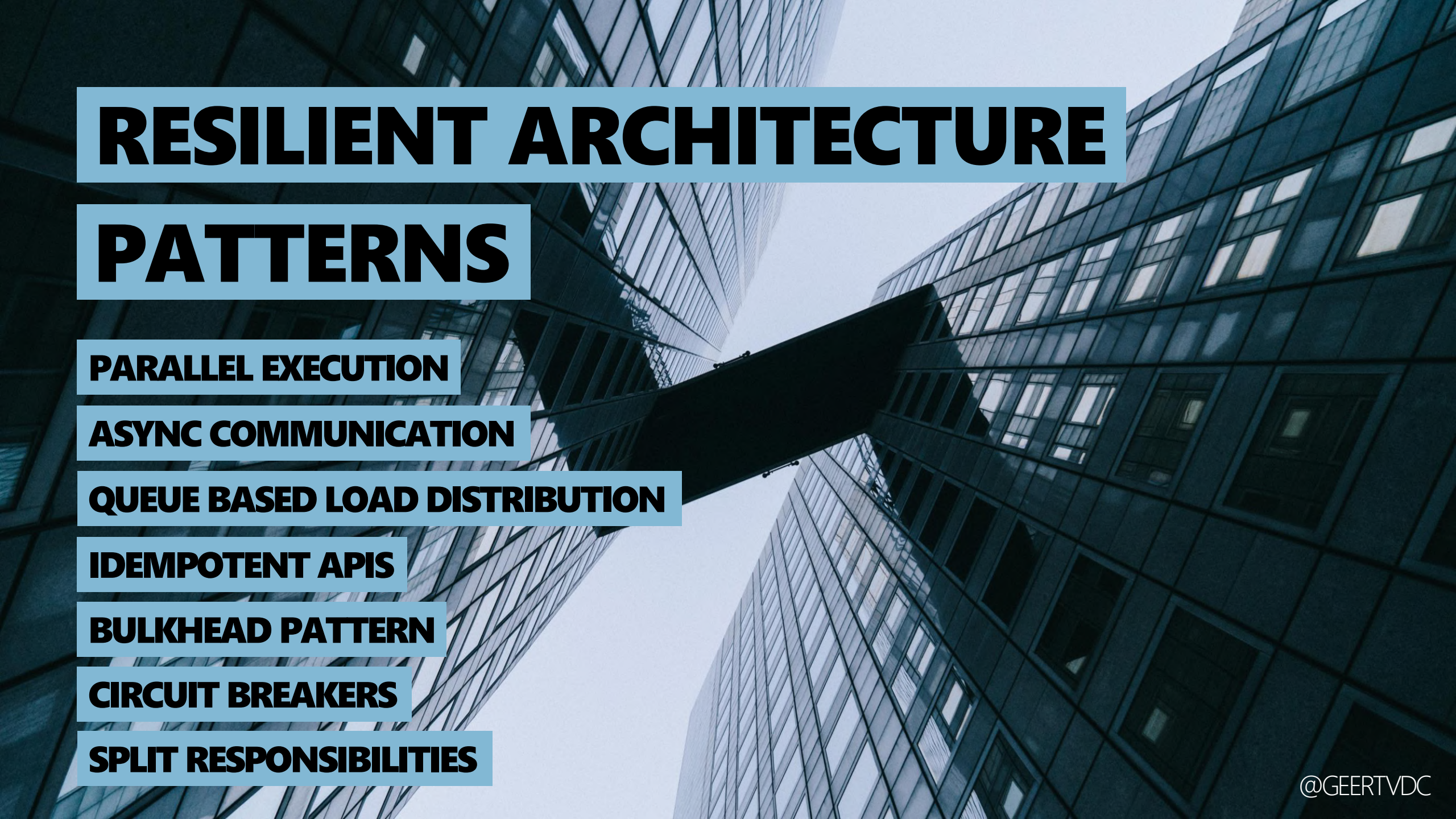

LEARN
HOW FAST DID WE DETECT?
HOW FAST DID WE RECOVER?
DO NOT BLAME!
@GEERTVDC


FIX
STEADY STATE
DEFINE HYPOTHESIS
DESIGN & EXECUTE
LEARN
FIX
EMBED


FIX
IMPLEMENT FIX
RERUN EXPERIMENT
SATA 95111
@GEERTVDC


EMBED
STEADY STATE
DEFINE HYPOTHESIS
DESIGN & EXECUTE
LEARN
FIX
EMBED


EMBED
EMBED IN CULTURE
ONBOARDING
CONTINUOUS CHAOS
@GEERTVDC


RESILIENT ARCHITECTURE
PATTERNS
PARALLEL EXECUTION
ASYNC COMMUNICATION
QUEUE BASED LOAD DISTRIBUTION
IDEMPOTENT APIS
BULKHEAD PATTERN
CIRCUIT BREAKERS
SPLIT RESPONSIBILITIES
@GEERTVDC

MULTI PARALELLISM

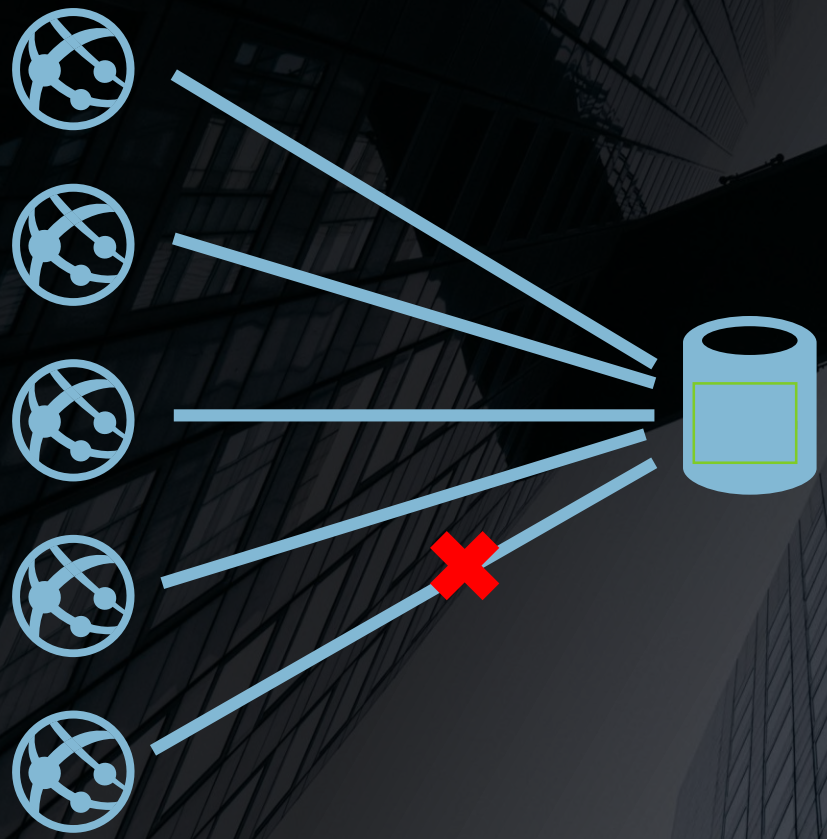
PARALLELISM	AVAILABILITY	DOWNTIME PER YEAR
1	99%	3 DAYS 16 HOURS
2	99,99%	53 MINUTES
3	99,9999%	32 SECONDS

HOW PARALEL IS YOUR CLOUD COMPONENT ?

AVAILABILITY ZONES | REGIONS


ASYNC COMMUNICATION
SYNC REQUIRES A CONNECTION PER REQUEST
FOCUS ON MESSAGE BASED COMMUNICATION
DECOUPLING
PUB SUB
LISTENER
@GEERTVDC

QUEUE BASED LOAD DISTRIBUTION



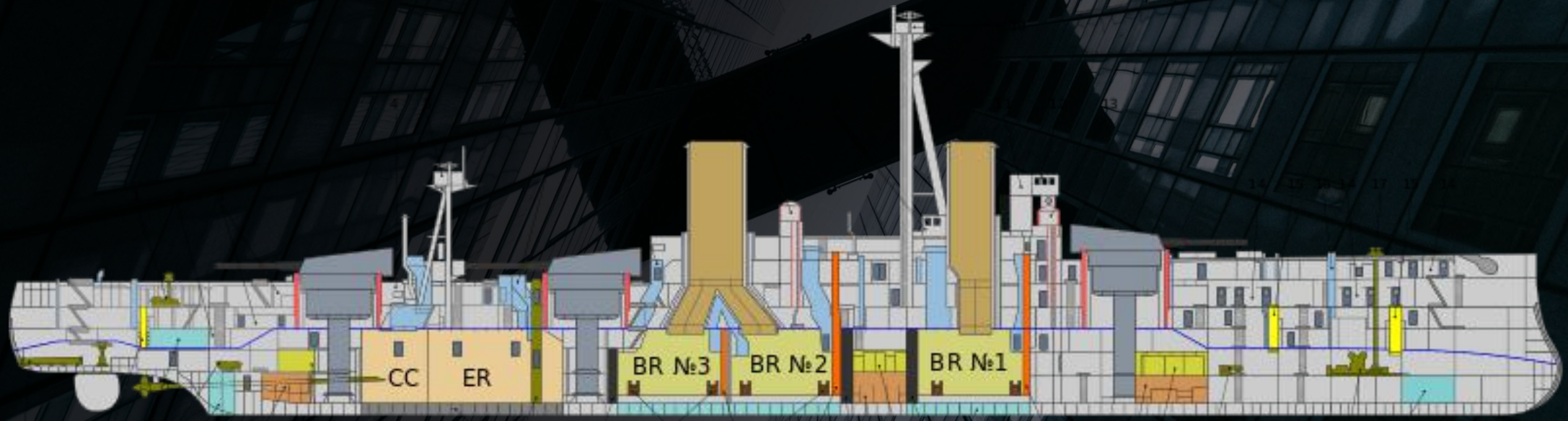
QUEUE BASED LOAD DISTRIBUTION

IDEMPOTENT APIS

HTTP METHOD	IDEMPOTENCE	SAFETY
GET	YES	YES
HEAD	YES	YES
PUT	YES	NO
DELETE	YES	NO
POST	NO	NO
PATCH	NO	NO

BULKHEAD PATTERN

ISOLATE WORKLOADS LIKE THE HULL OF A SHIP



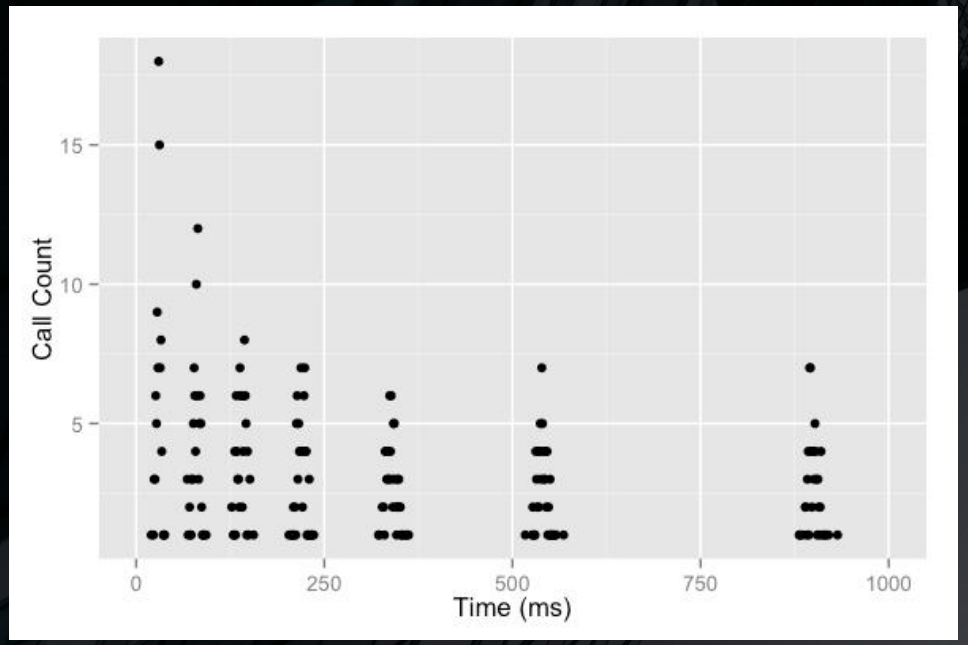
@GEERTVDC

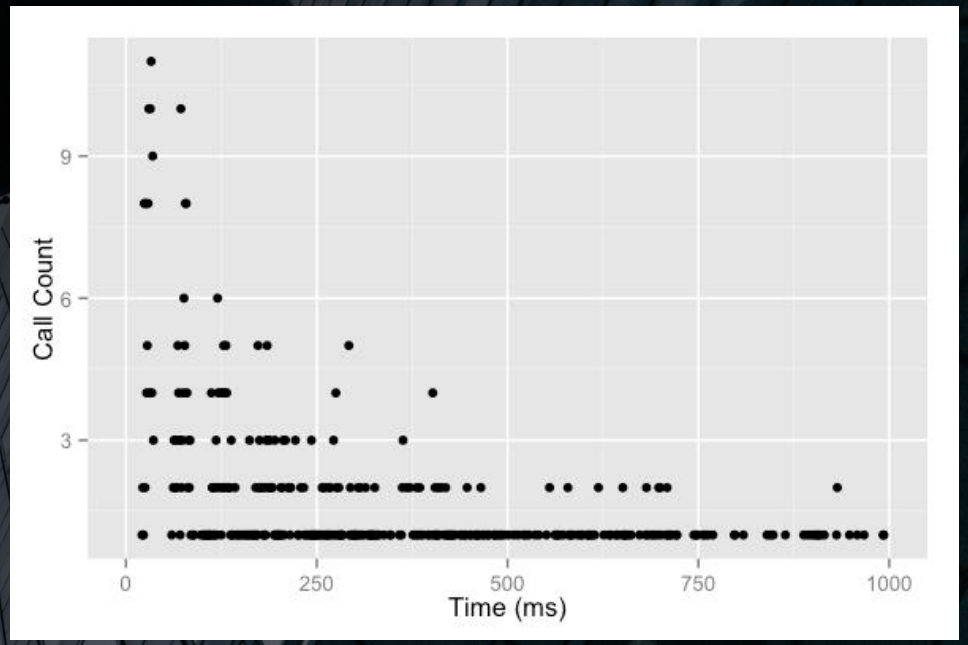
CIRCUIT BREAKER

@GEERTVDC

CIRCUIT BREAKER

ADD JITTER TO RETRIES






SPLIT RESPONSIBILITIES
READ / WRITE SHARDING
CQRS
@GEERTVDC

BIG CULTURE CHANGE

- PRODUCTION ACCESS
- FULL CYCLE DEVELOPERS

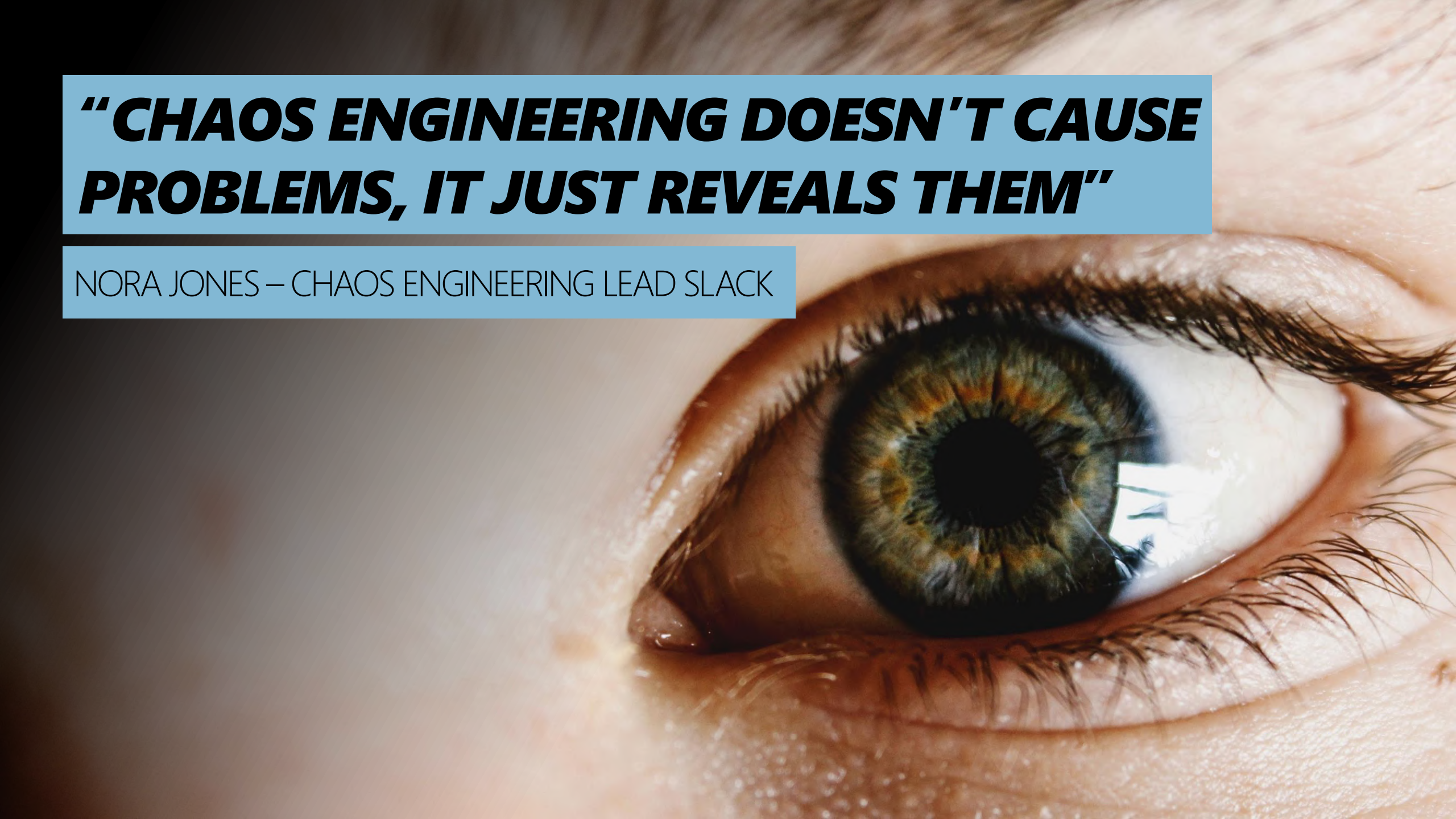
START EXPERIMENTING

- OBSERVABILITY
- START SMALL
- CHECK OUT TOOLS

WRAP UP

"CHAOS ENGINEERING DOESN'T CAUSE PROBLEMS, IT JUST REVEALS THEM"

NORA JONES – CHAOS ENGINEERING LEAD SLACK



RESOURCES

BOOKS:
Chaos engineering - O'Reilly
Chaos engineering observability - O'Reilly

TOOLS:
chaostoolkit.org
gremlin.com
github.com/netflix/simianarmy
github.com/asobti/kube-monkey

RESOURCES:
principlesofchaos.org
github.com/dastergon/awesome-chaos-engineering
docs.microsoft.com/en-us/azure/architecture/patterns/category/resiliency

ALL PICTURES USED ARE FROM UNSPLASHED.COM

@GEERTVDC

GEERT VAN DER CRUIJSEN

THANK YOU!