

Chaos Engineering

Inject failures to survive

About me.



Johannes Edmeier
Co-Founder chaosmesh

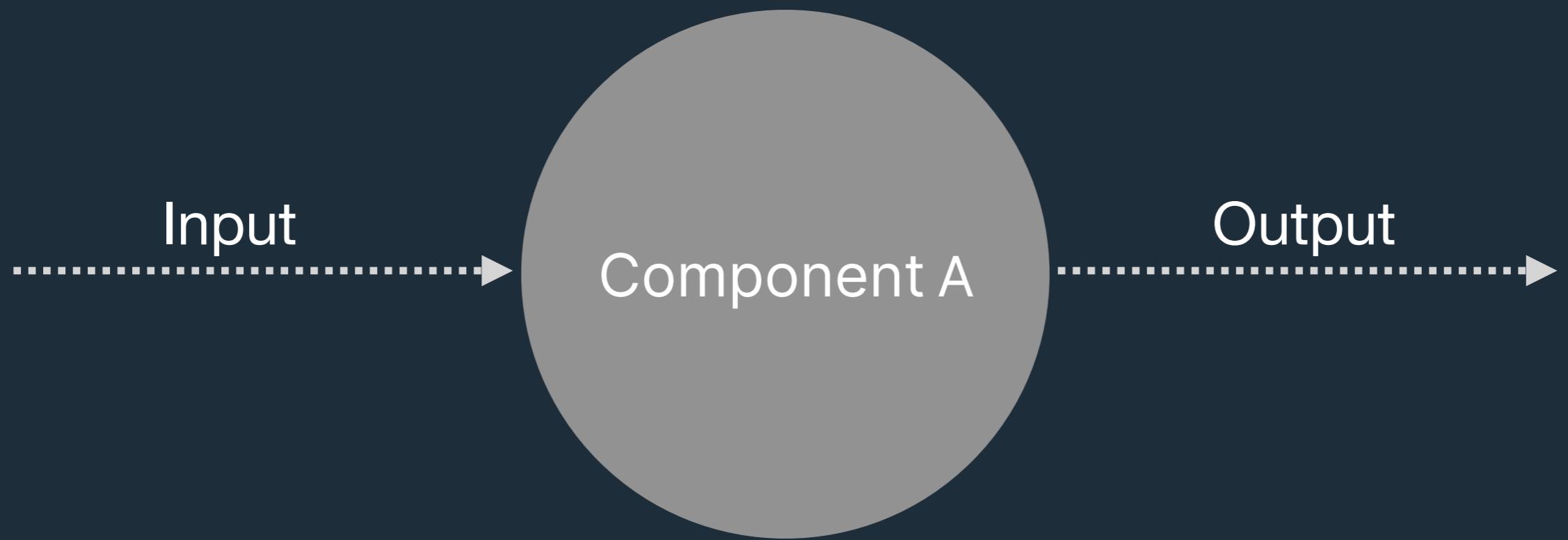
 @joshiste

 chaosmesh.com

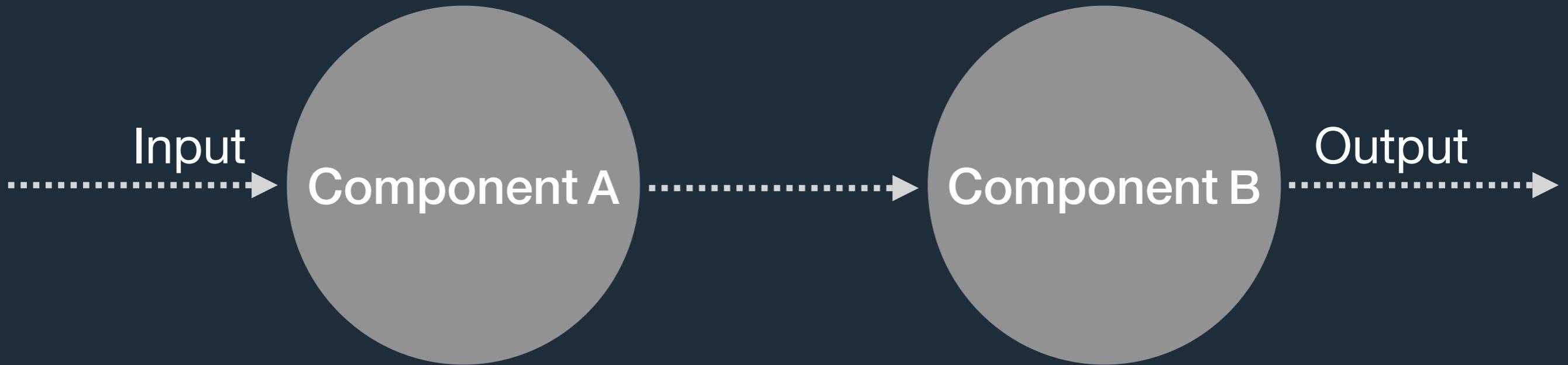
**On the way
to the most
beautiful
place...**

Production

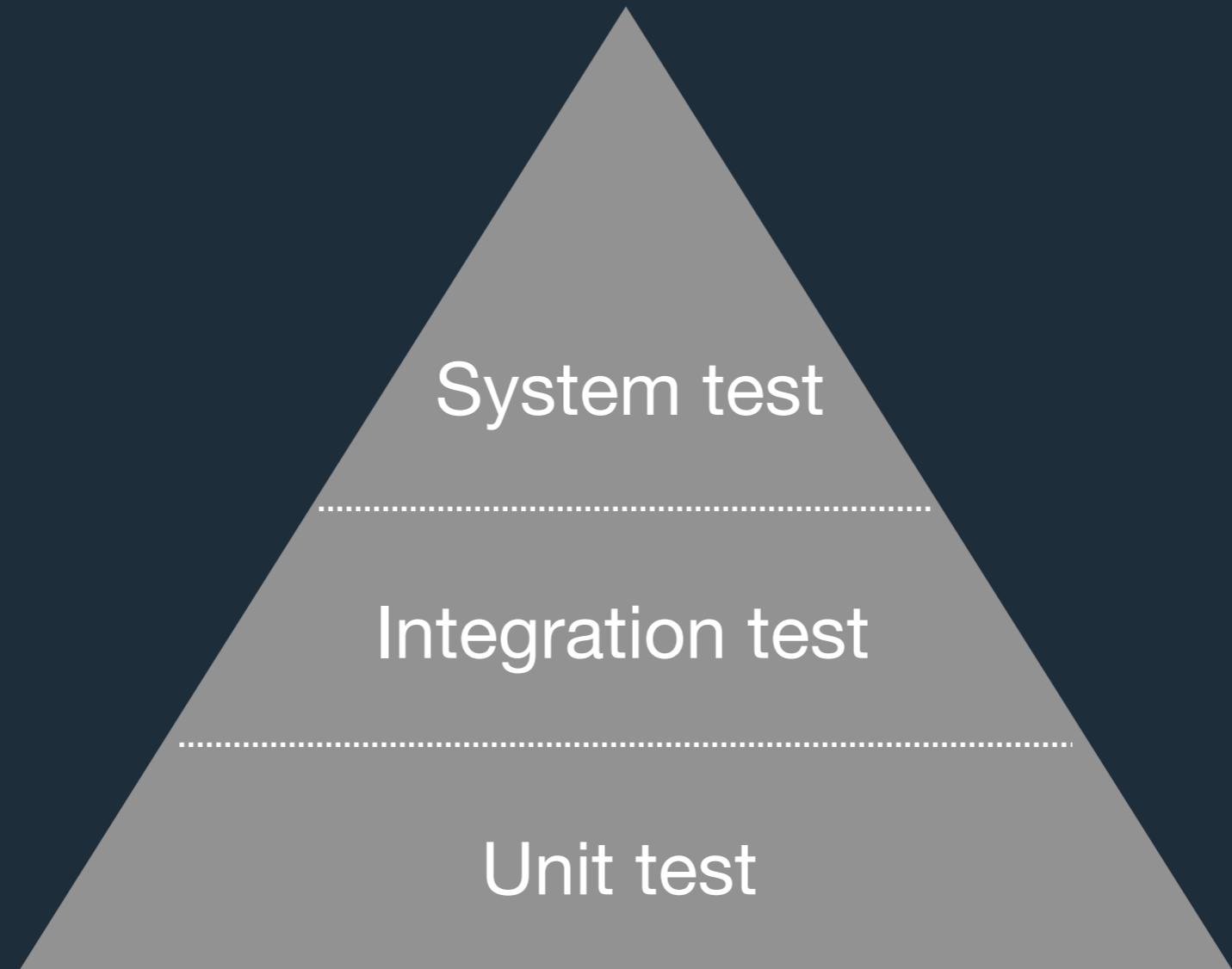
Unit tests



Integration tests



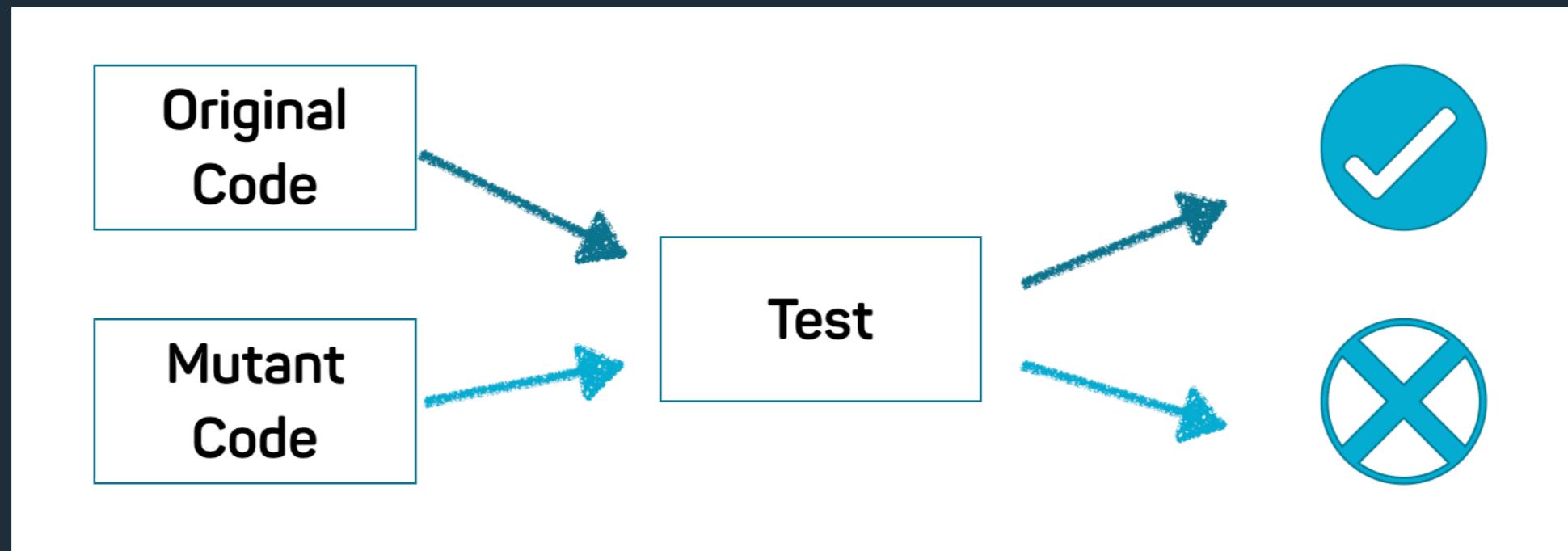
Test pyramid dilemma



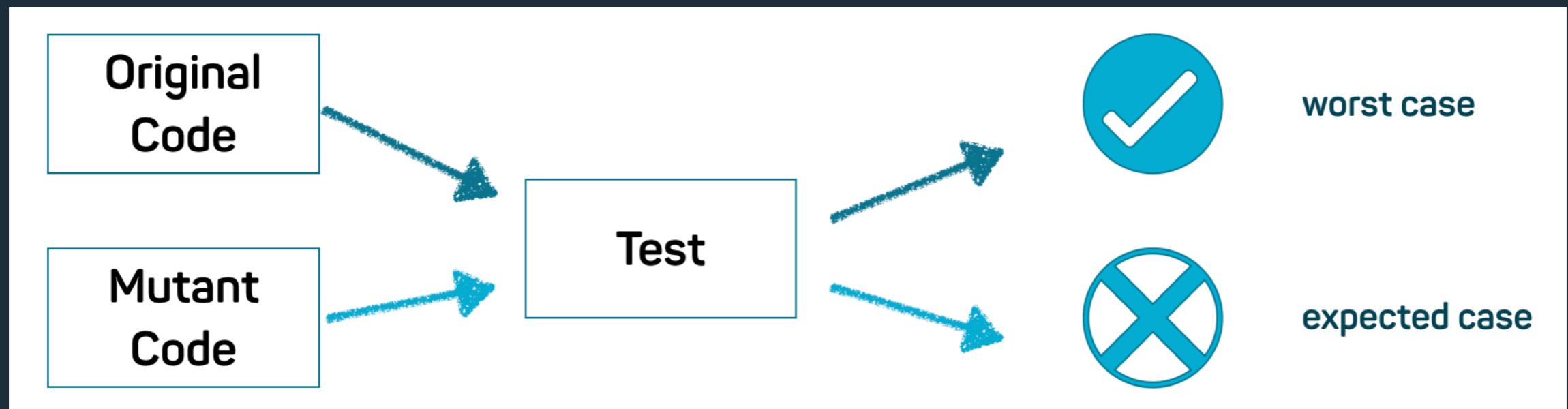
Test coverage

80% - 90%

Mutation testing



Mutation testing



This
unpleasant
feeling
remains...

Is a life in
production
possible at
all?

Why?

Will our fallbacks work?

How does the application behave with network latency?

What if one of our services breaks down?

Service discovery, are up and running?

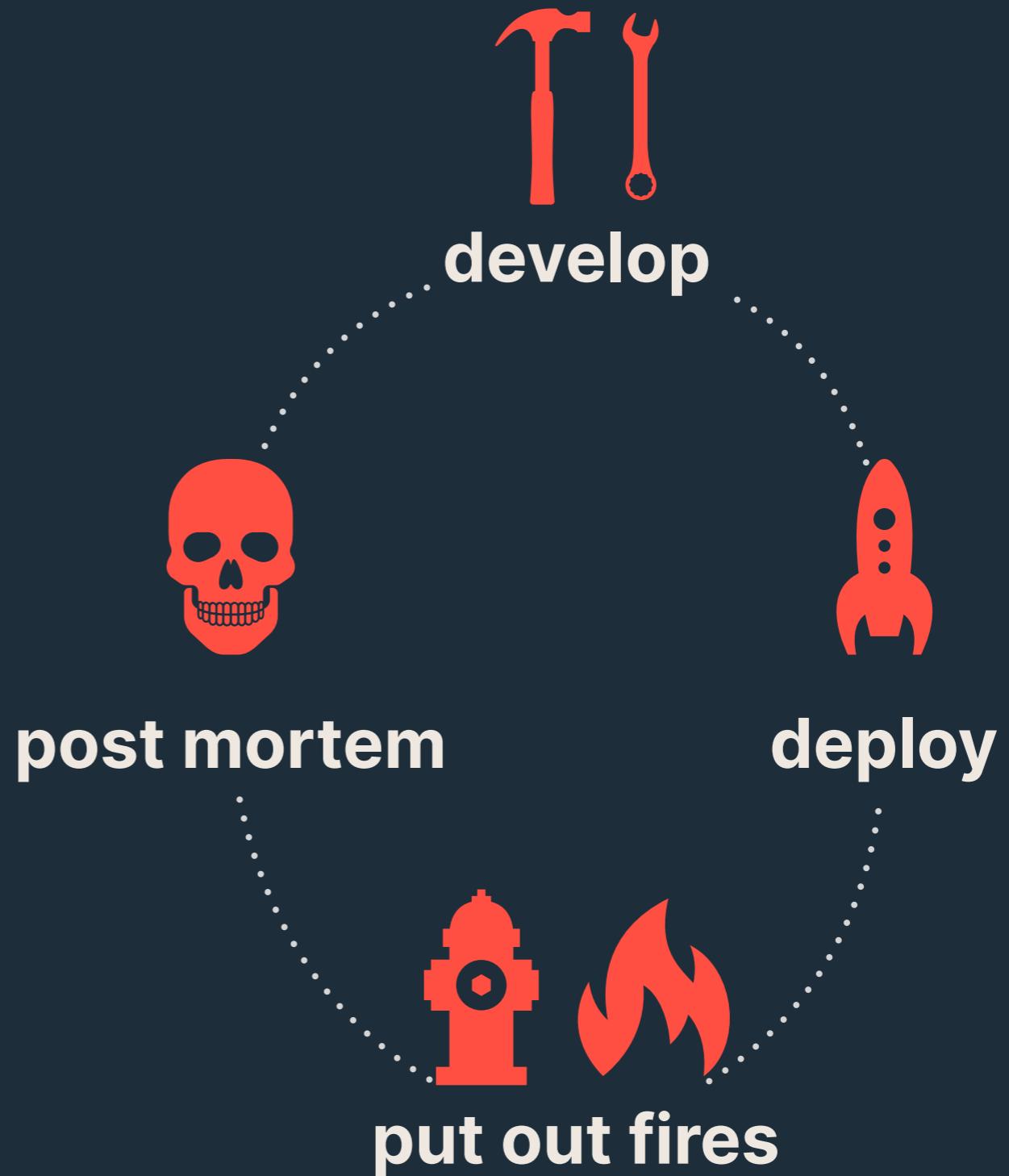
Client-side load balancing, never saw it in action!



Production
HATES YOU

**What
went
wrong?**

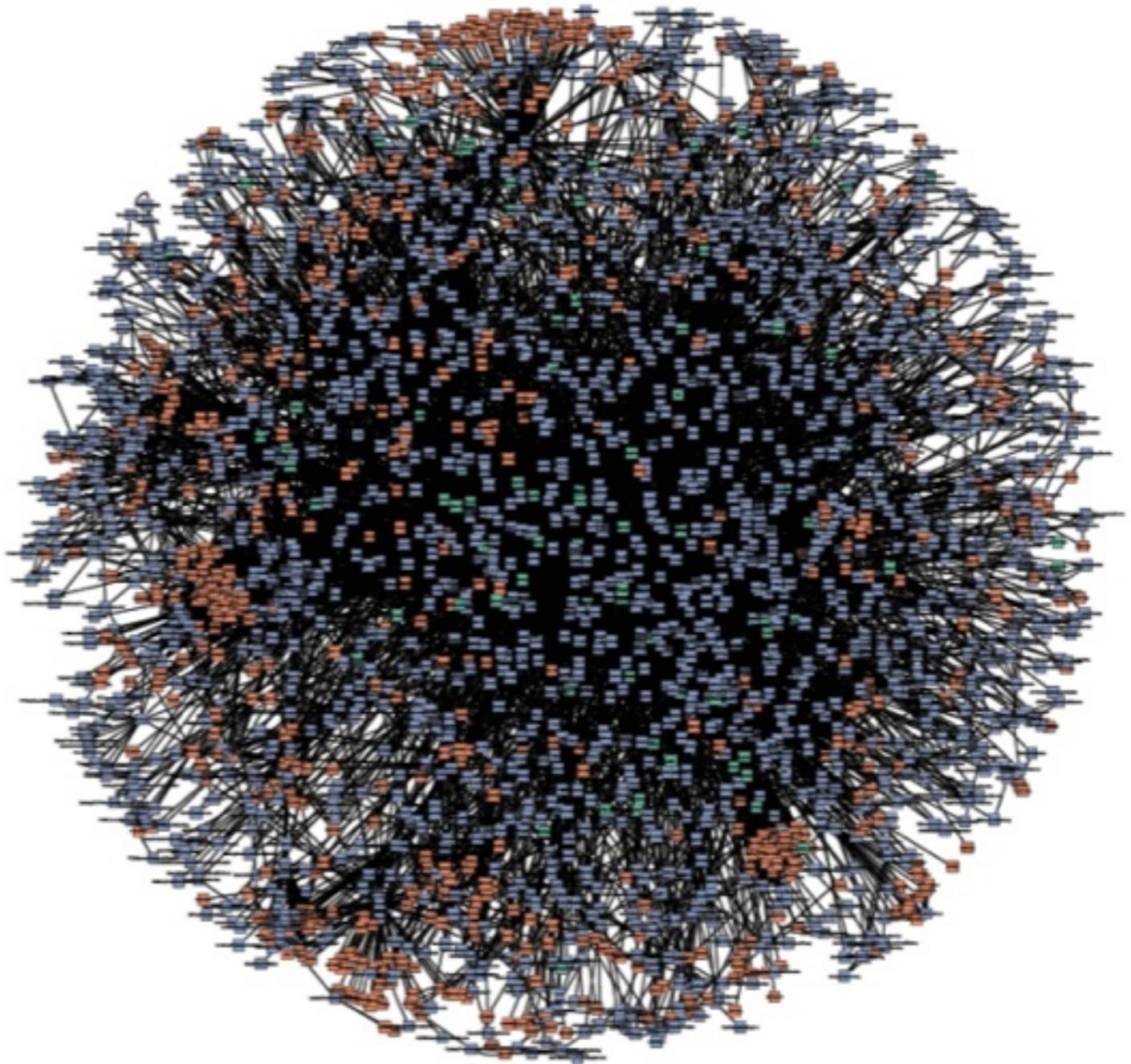
Cycle of Death



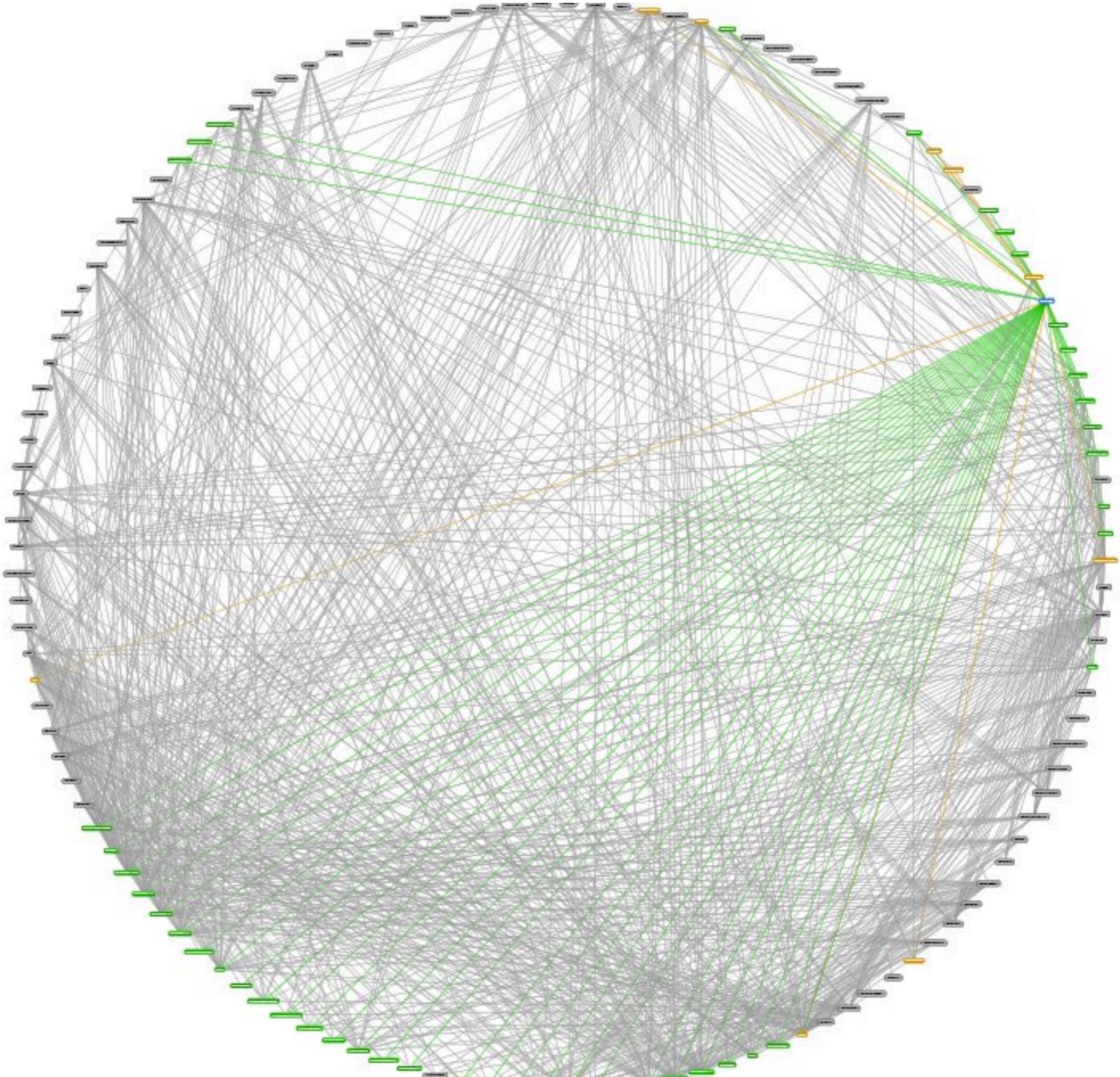
Complex distributed systems



Death Star Amazon



Death Star Twitter



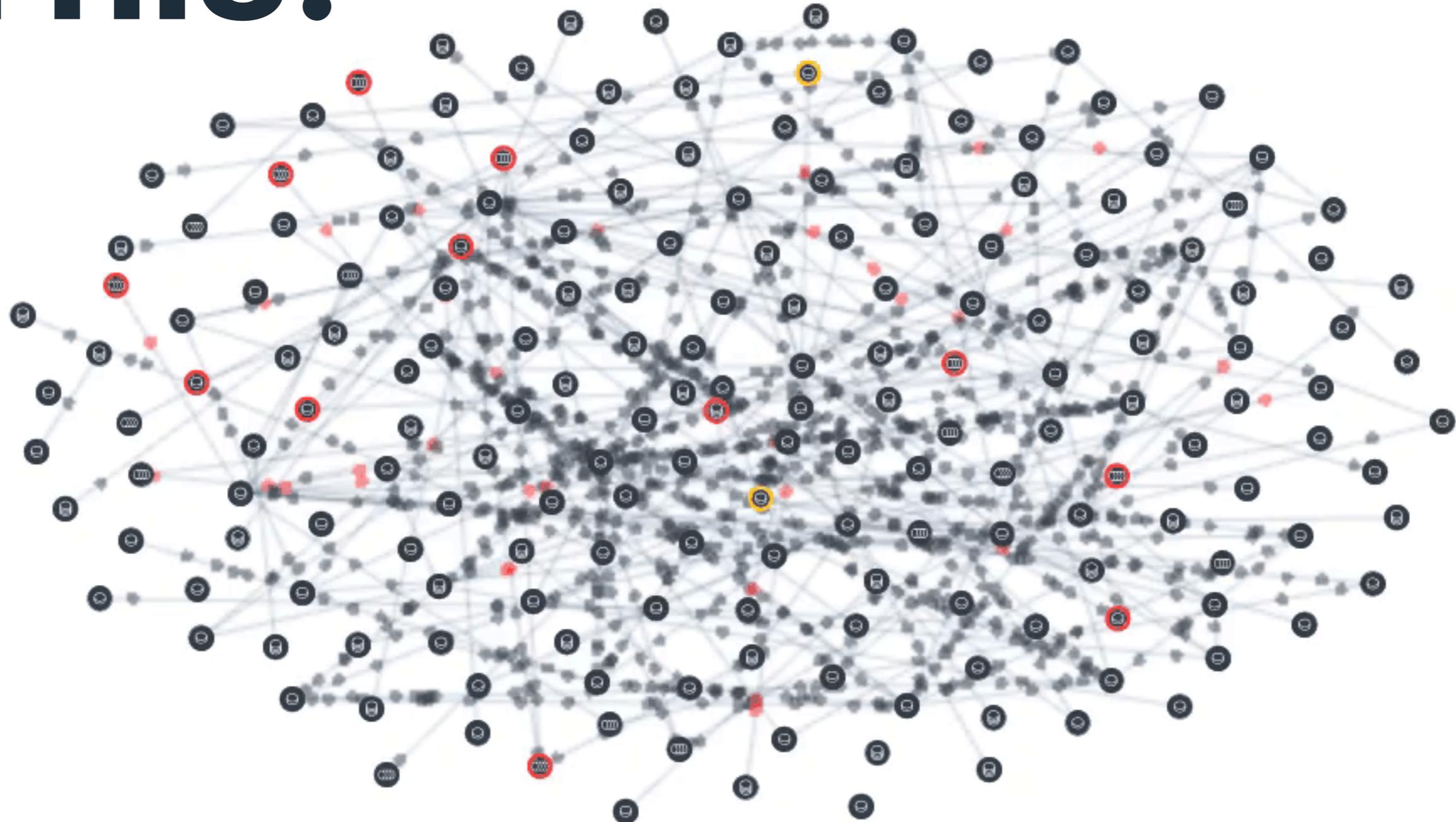
Death Star Netflix



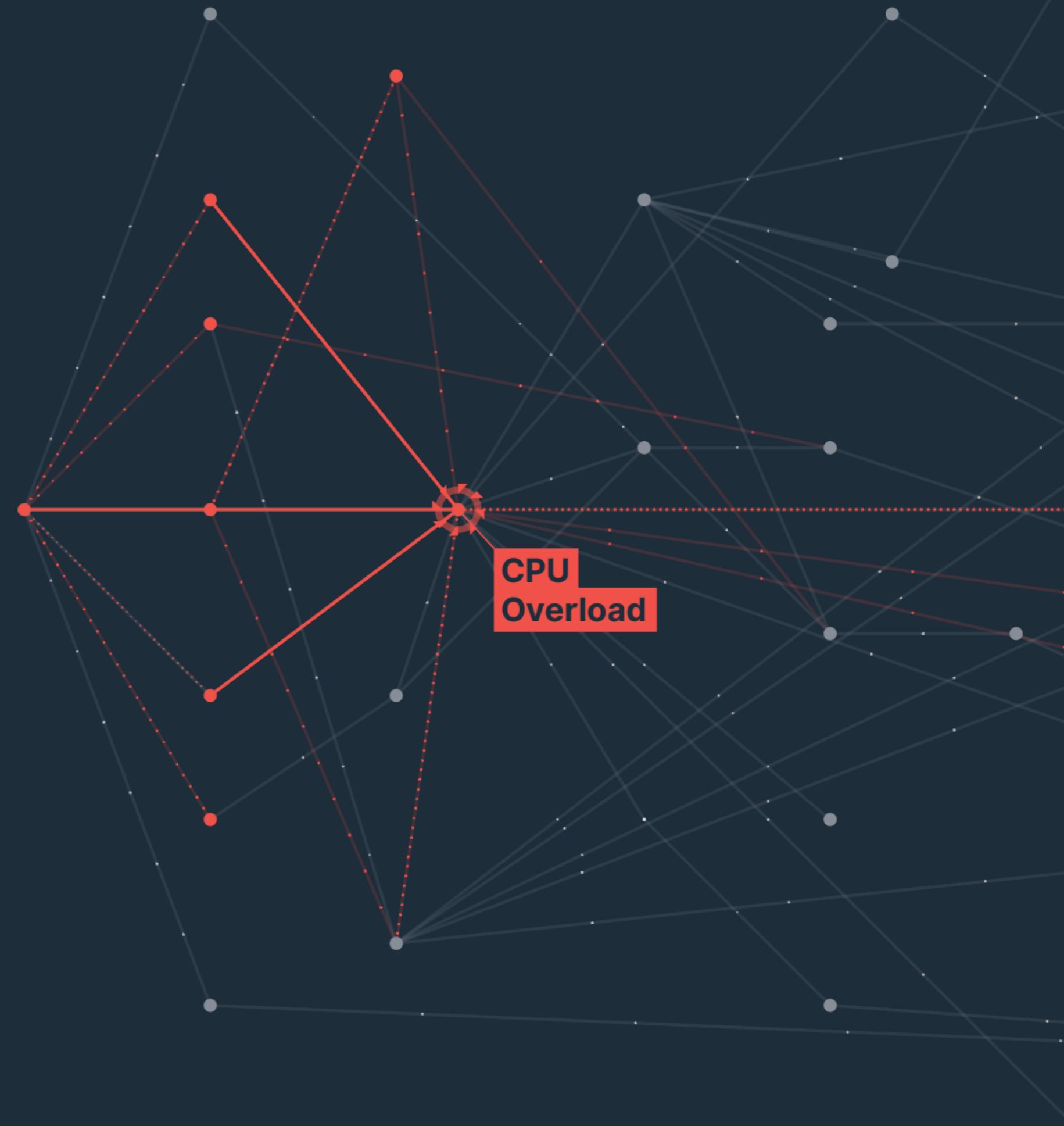
We are not
Amazon,
Twitter or
Netflix...

**We build
and
operate...**

This!



Highly vulnerable: Dark Debt will strike you.



Cost of Downtime

Infrastructure failure

\$100k

per hour

Critical application failure

\$500k

to \$1 Million

per hour

<https://blogs.gartner.com/andrew-lerner/2014/07/16/the-cost-of-downtime/>

<https://www.the20.com/blog/the-cost-of-it-downtime/>

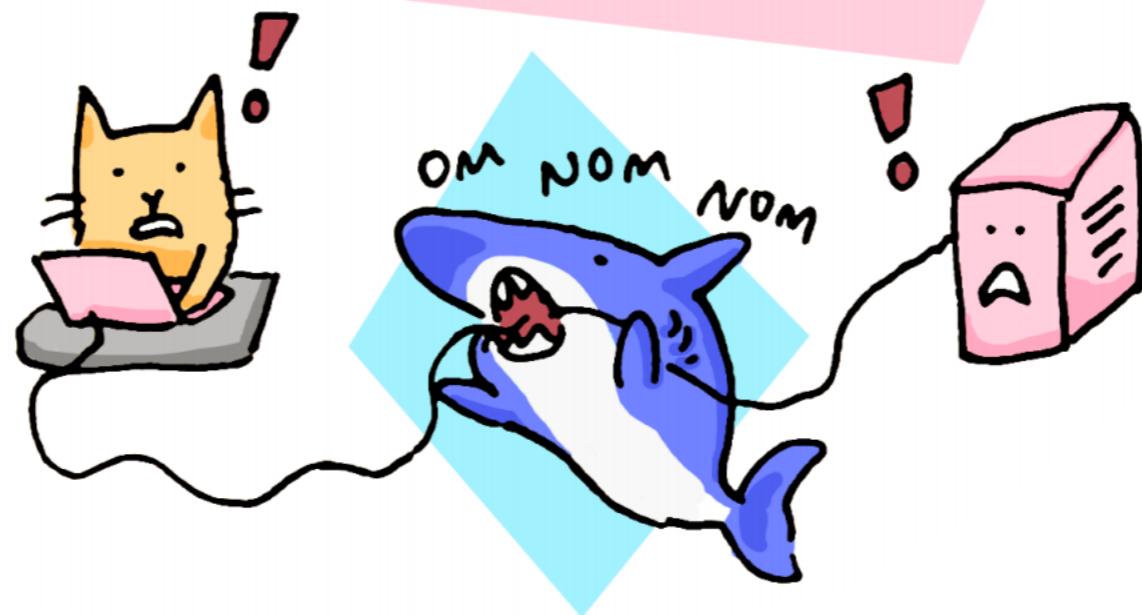
@deniseyu21

the 8 Fallacies of Distributed Computing

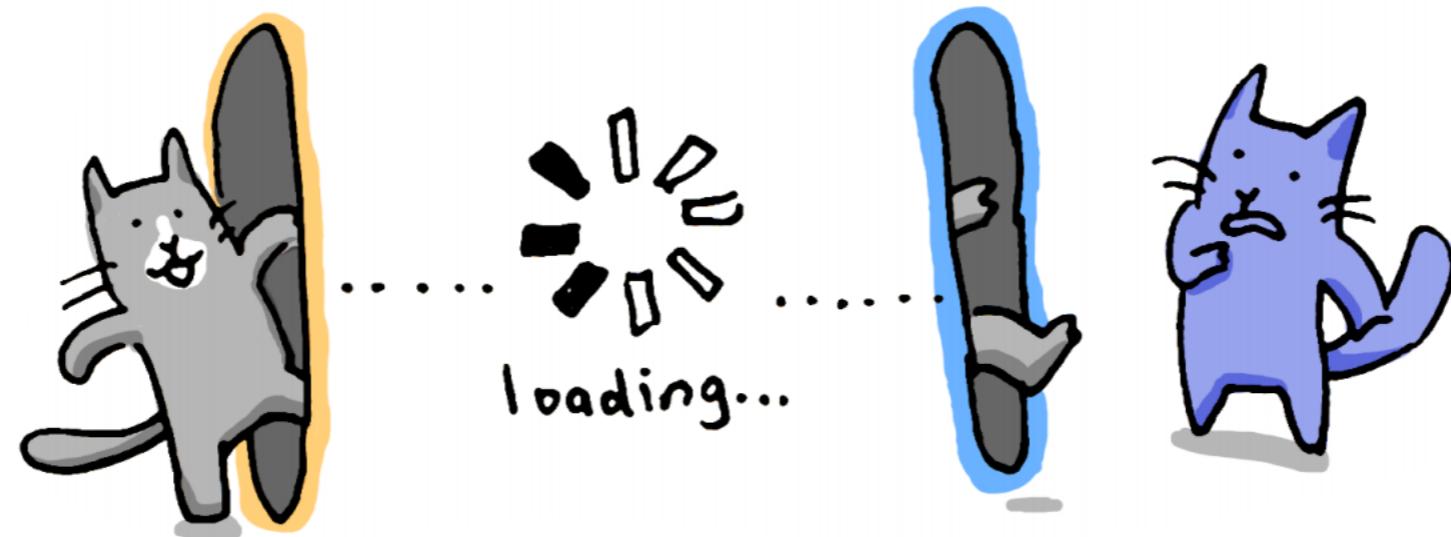


Originally formulated by L. Peter
Deutsch & Colleagues at Sun Microsystems
in 1994; #8 added in 1997 by James Gosling

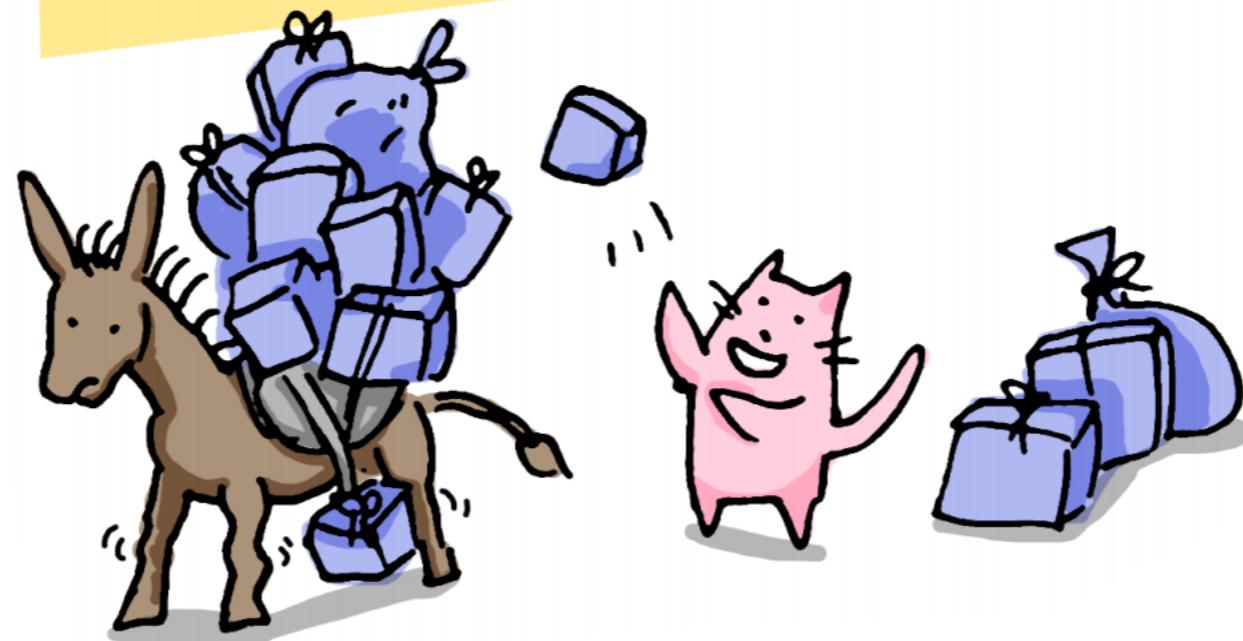
① The network is reliable



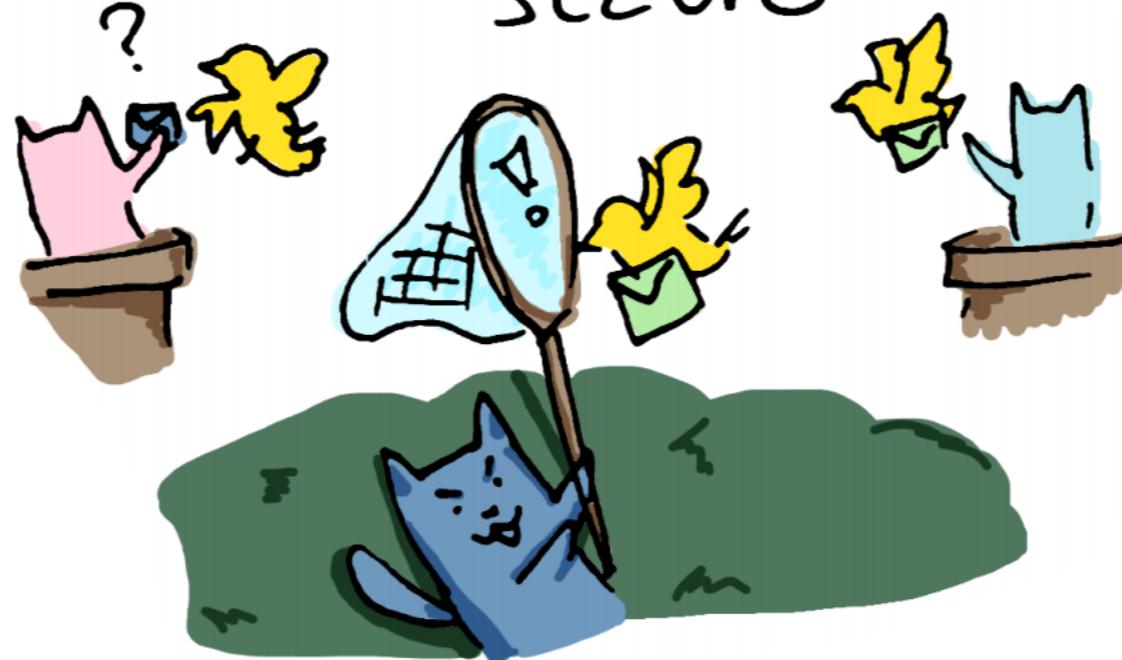
② Latency is ZERO



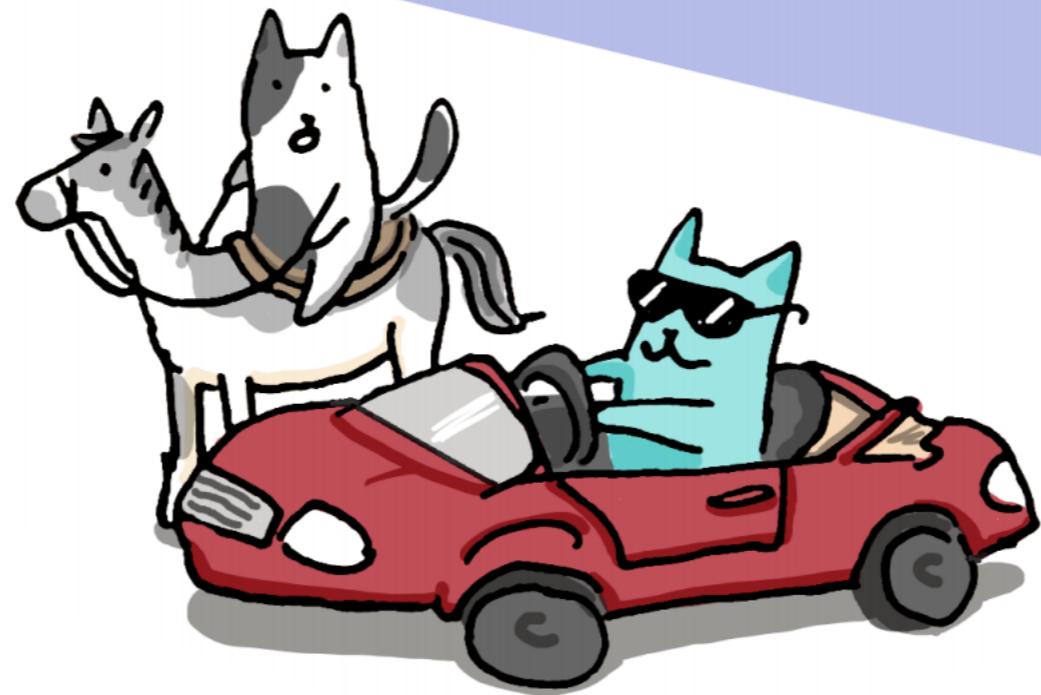
③ Bandwidth is infinite



④ The network is
Secure



(5) Topology doesn't
change



⑥ There is only
one administrator



⑦ Transport costs \$0



⑧ The network is
homogeneous



The background of the image is a dense, multi-layered collage of numerous antique books. The books are stacked haphazardly, creating a textured, almost abstract pattern. They come in a variety of colors, including shades of brown, tan, red, and cream, with some showing signs of wear and discoloration. The spines of many books are visible, displaying titles and names in different fonts and languages.

It's all about availability!

Welcome to Chaos Engineering

Chaos Engineering is
not...

Chaos Engineering is not...
... to cause chaos!

Chaos Engineering is not...

... breaking things just to break them!

Chaos Engineering is not...
... to run a chaos monkey!

Chaos Engineering is not...

... to let the whole simian army out of the cage!

Chaos Engineering is not...

... to use it in production
(from day 1)!

Chaos Engineering is not...
... to replace other kinds of tests!

Chaos Engineering is not...
... to do it alone and without any
arrangement!

**Chaos Engineering
is not just tools**

It's culture!

This is bob



He is responsible for service A



These are bob's teammates



They depend on bob's service



They blame bob for the bad performance of service A



Chaos Engineering 101

Be social and communicate

Share your experiences and thoughts

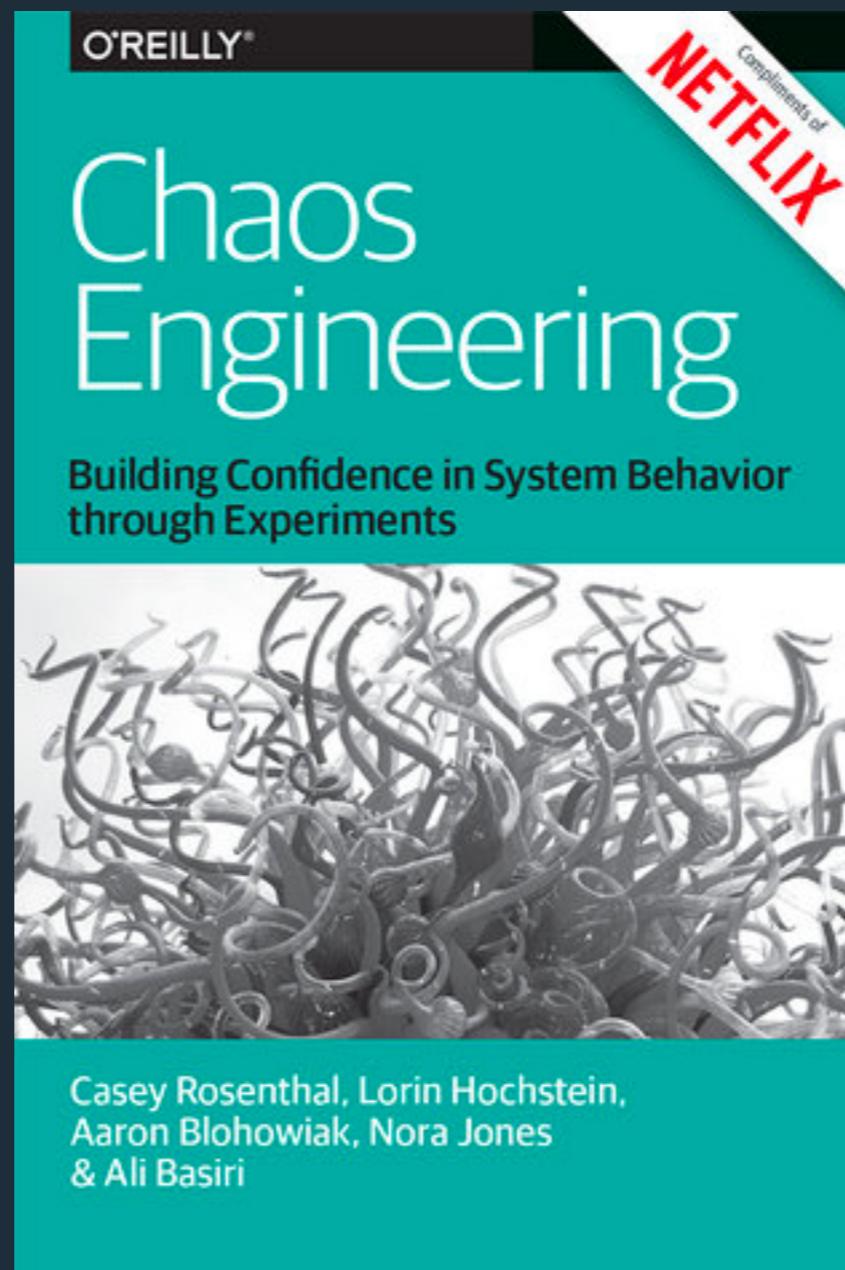
Stop blaming each other

Work together

What is Chaos Engineering?



Chaos Engineering book

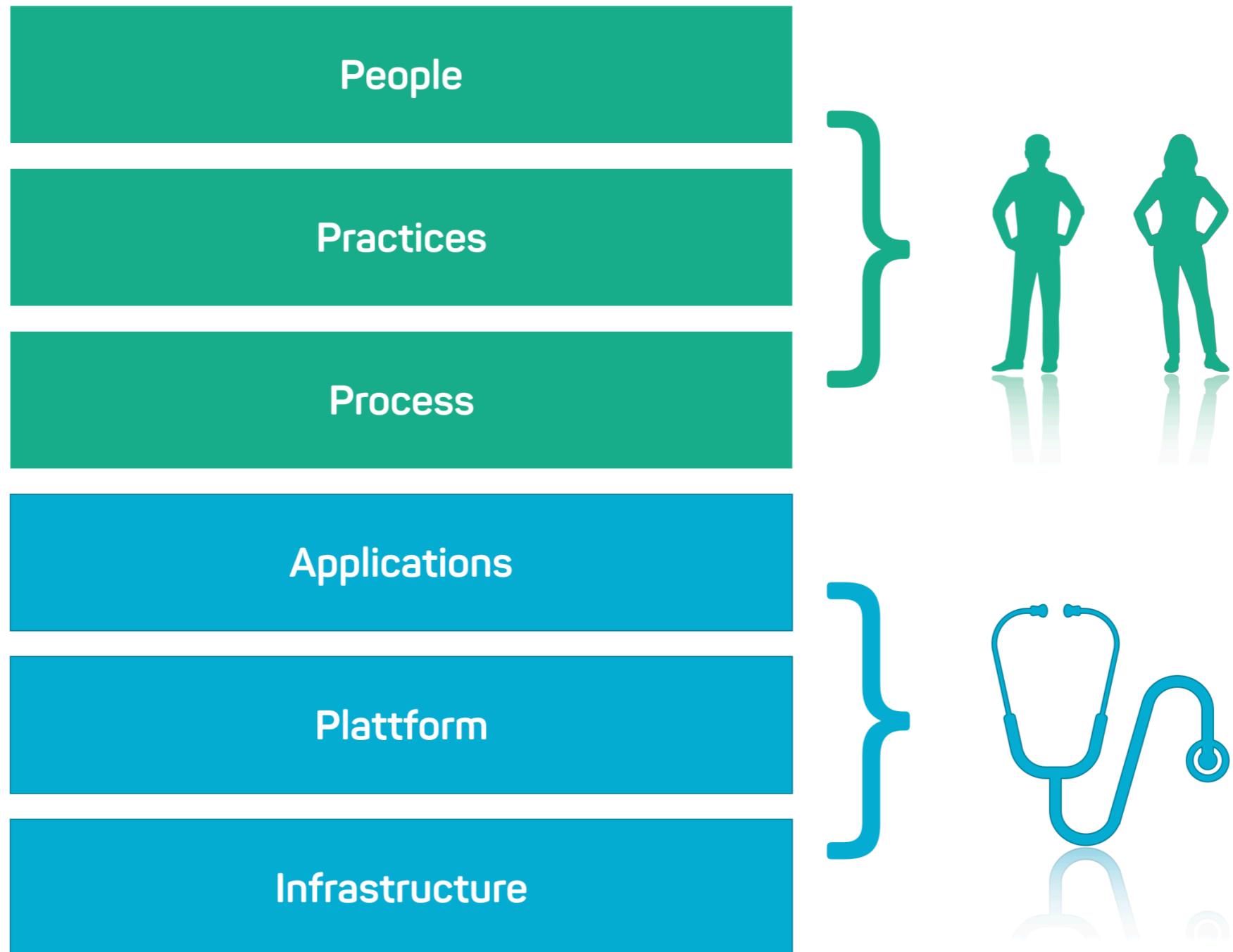


<https://www.oreilly.com/library/view/chaos-engineering/9781491988459/>

„Chaos Engineering is the discipline of experimenting on a distributed system in order to build confidence in the system's capabilities to withstand turbulent conditions in production!“

CPU
Overload

<https://principlesofchaos.com>

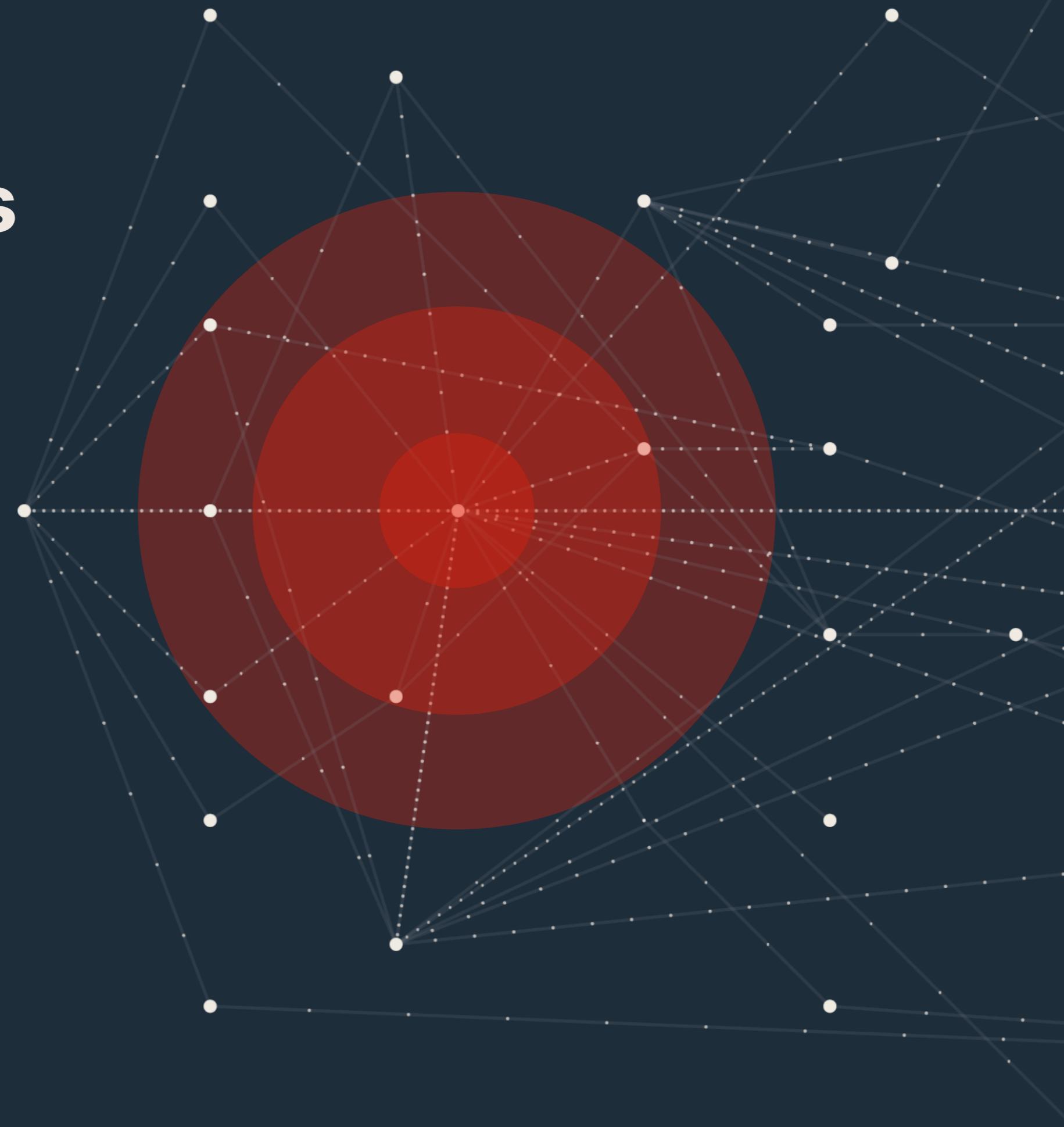


Hypothesis

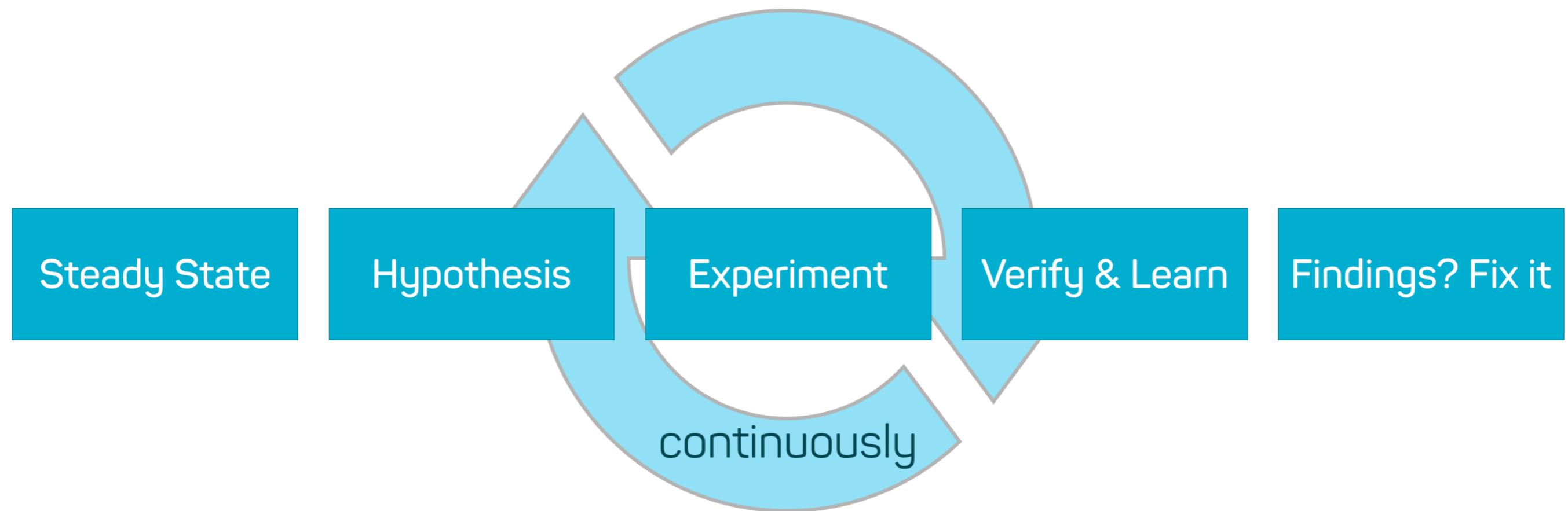
What should happen when...

- Latency
- Exception
- Outage of service A

Blast Radius



Chaos engineering cycle

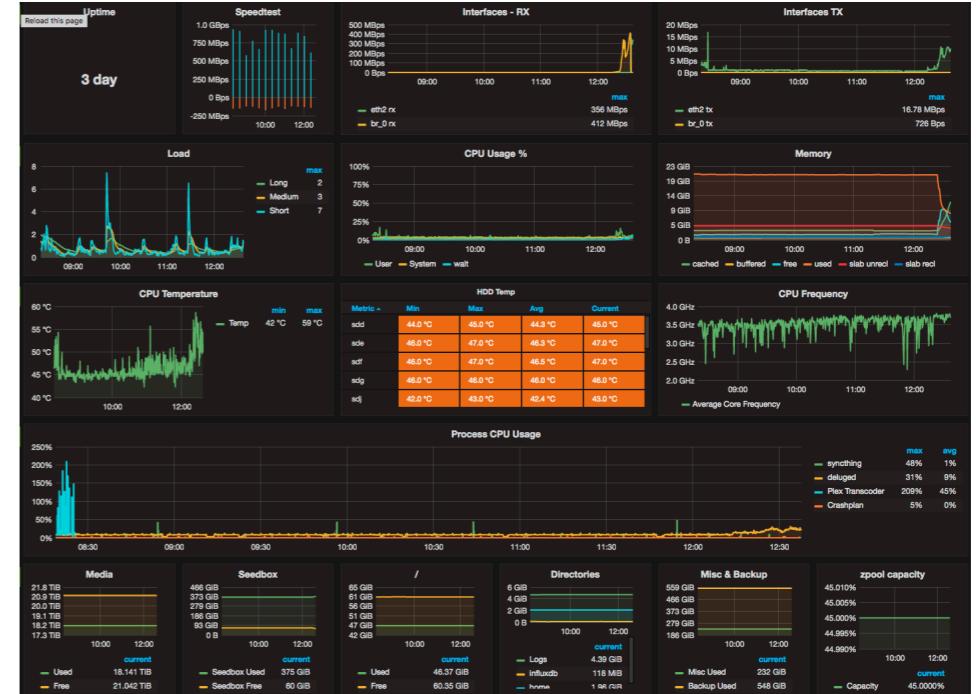


If you know your
experiment will fail...

...don't do it!

What is normal?

Do you see it now?

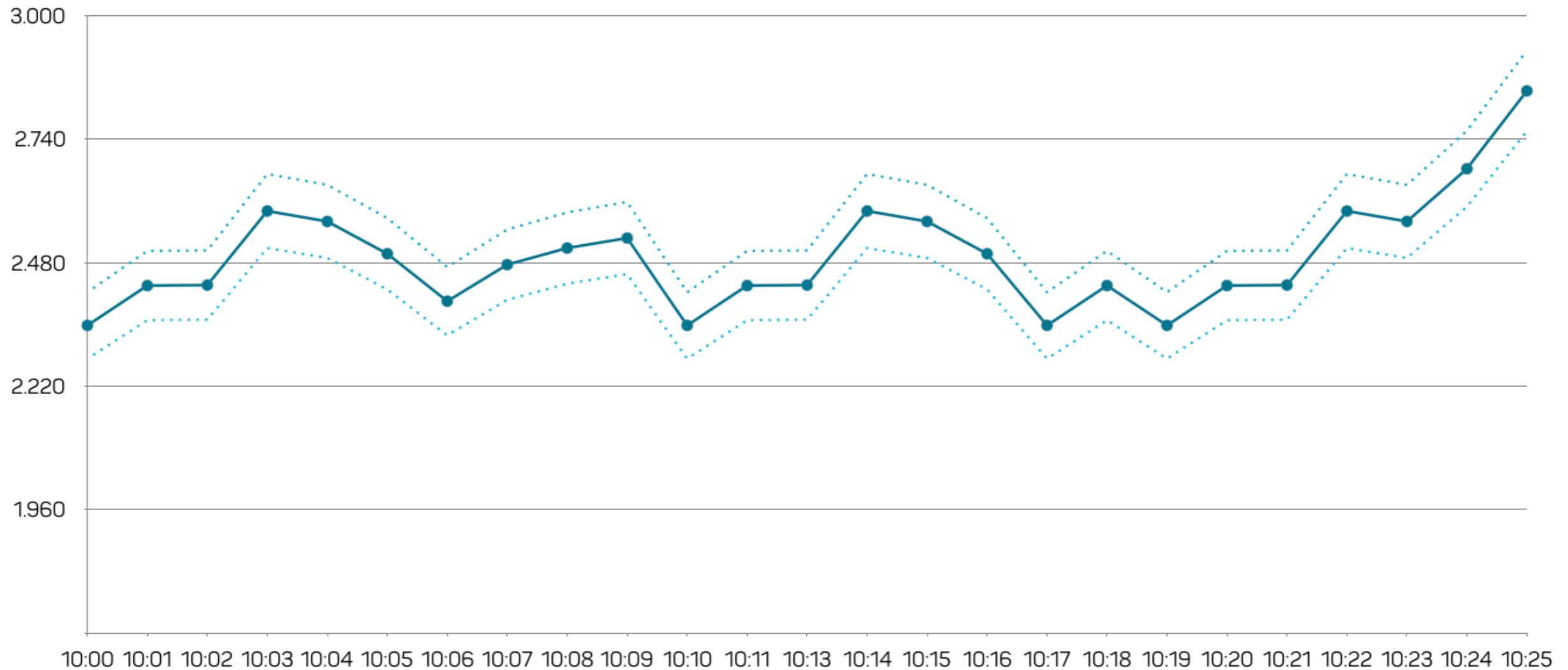


Steady State

- Define metrics about the overall state of your system
- Monitor all your metrics
- Technical metrics like: CPU, Memory, etc.
- Business metrics
- Business metrics outweigh technical metrics

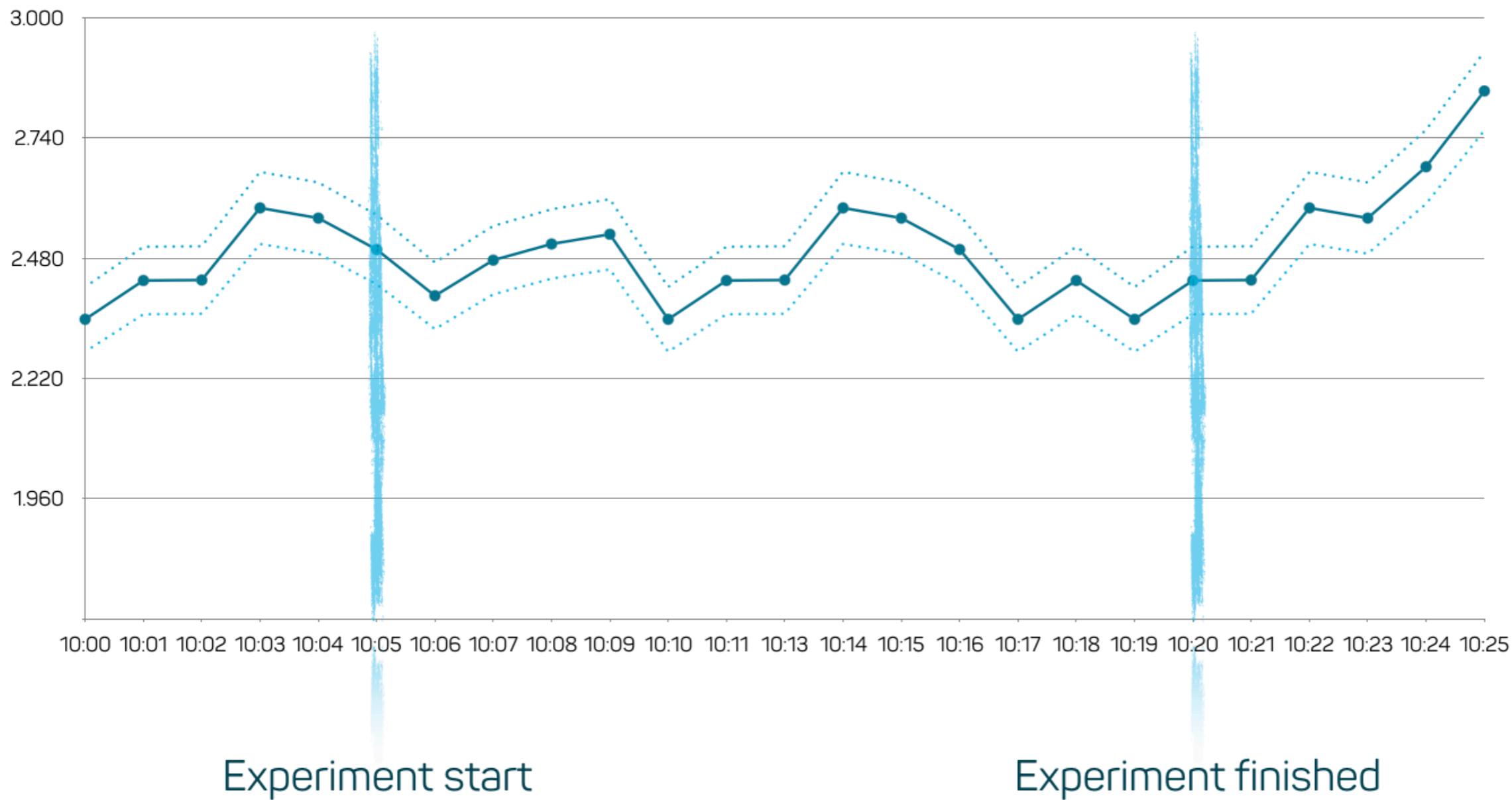
Steady state

Orders per minute - on a typical monday morning



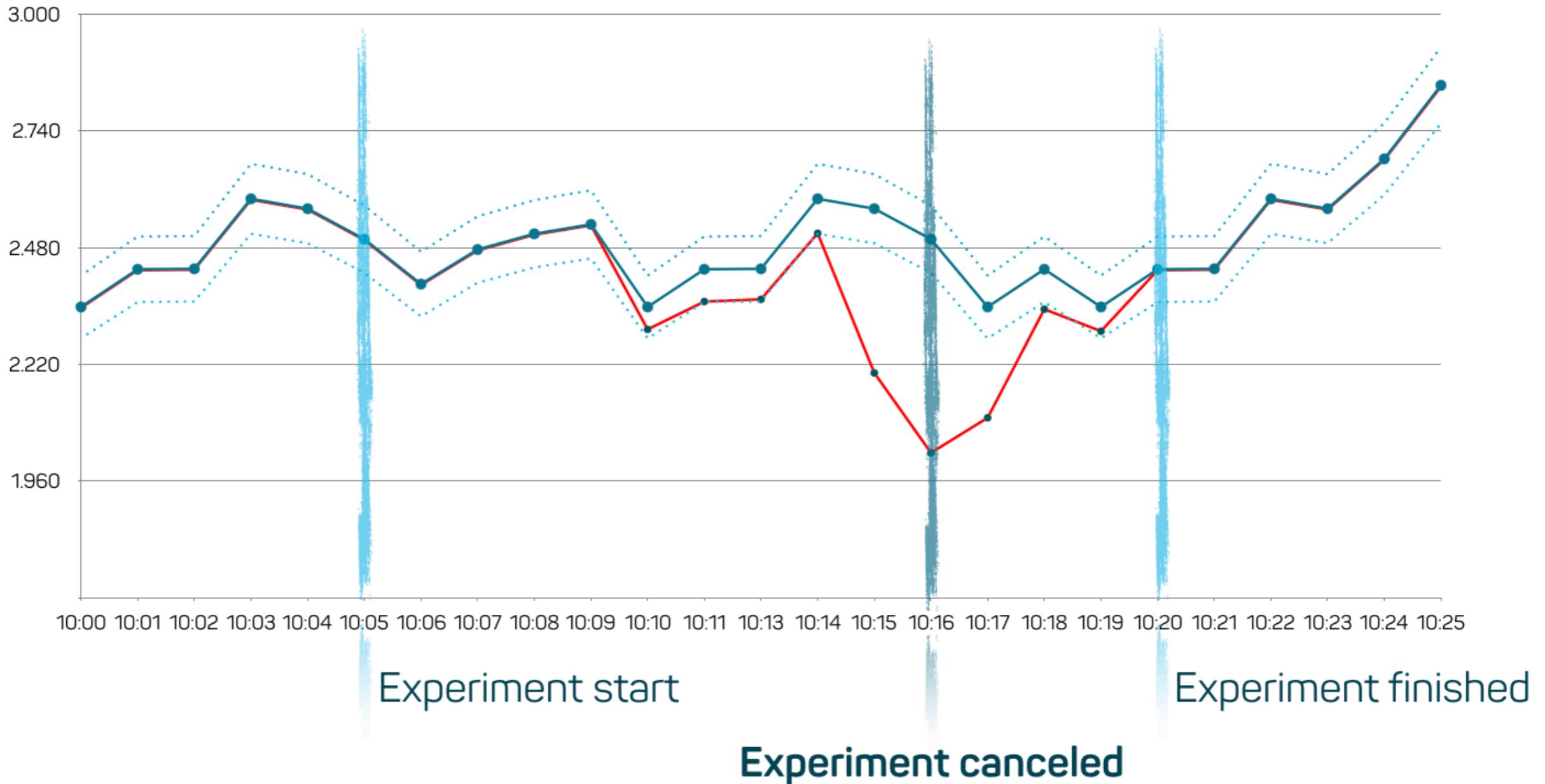
Steady state

Orders per minute



Steady state

Experiment was cancelled



Chaos Engineering Tools

Linux basic attacks

Shutdown / Reboot

```
# shutdown in 5s  
shutdown -t 5 -h
```

```
# reboot in 5s  
shutdown -t 5 -r
```

Linux basic attacks

CPU burning - inspired by Tammy Button

With openssl speed test the performance of cryptographic algorithms

```
# burn.sh
while true;
    Do openssl speed;
done
```

```
# cpu_burning.sh
for i in {1..32}
do
    nohup burn.sh &
done
```

Linux basic attacks

Stress CPU

A load average of four is imposed on the system by specifying two CPU-bound processes, one I/O-bound process, and one more memory allocator process.

```
stress --cpu 2 --io 1 --vm 1 --vm-bytes 128M --timeout 10s --verbose
```

Linux basic attacks

Traffic control (tc) - package iproute2

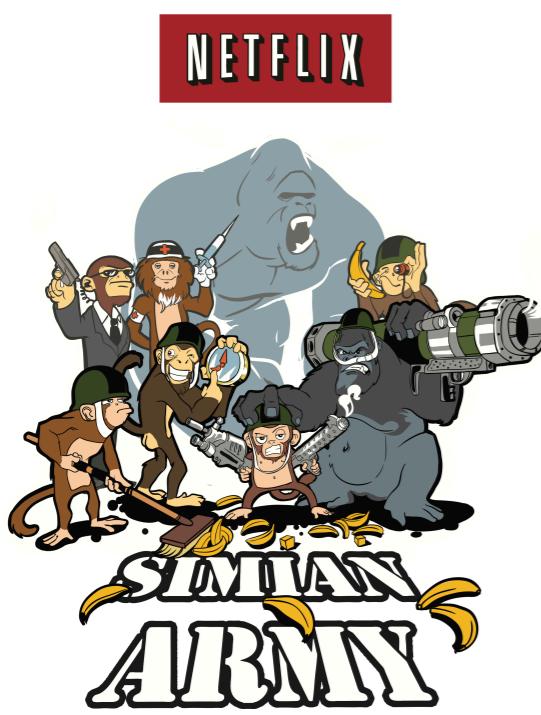
```
# delay  
tc qdisc add dev eth0 root netem delay 100ms 20ms
```

```
# packet loss  
tc qdisc add dev eth0 root netem loss 0.1%
```

```
# packet duplication  
tc qdisc add dev eth0 root netem duplicate 1%
```

```
# packet corruption  
tc qdisc add dev eth0 root netem corrupt 0.1%
```

Chaos Engineering Tools: OSS



... and many more:

<https://github.com/dastergon/awesome-chaos-engineering#notable-tools>

**Thank
you.**