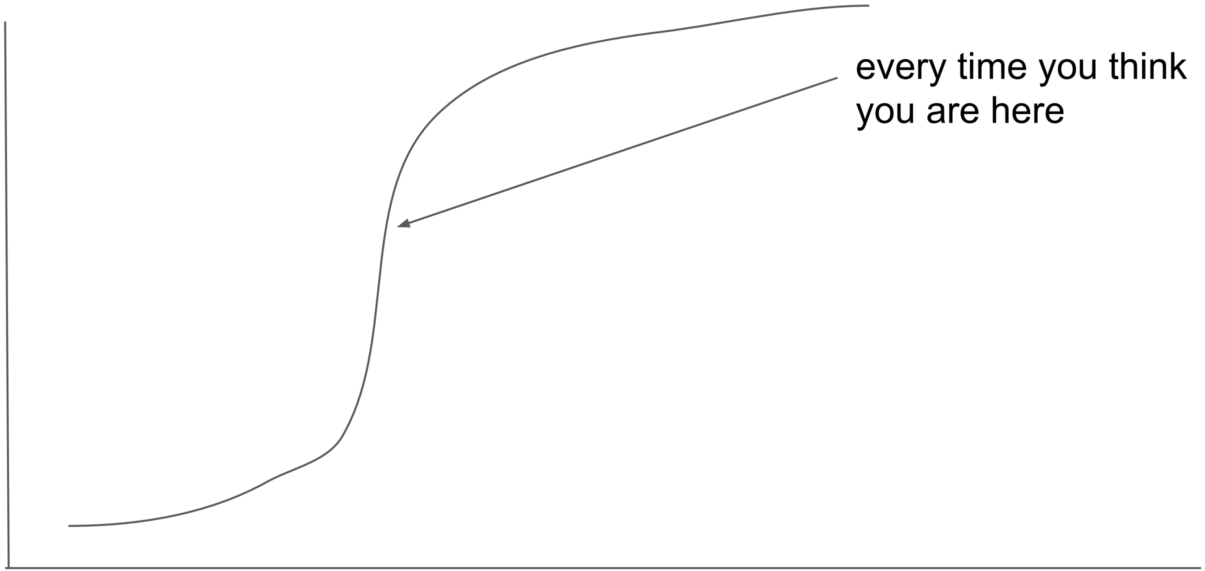
Shaping Reality to Shape Outcomes

Making SRE Work with Uber Growth

Tom Croucher

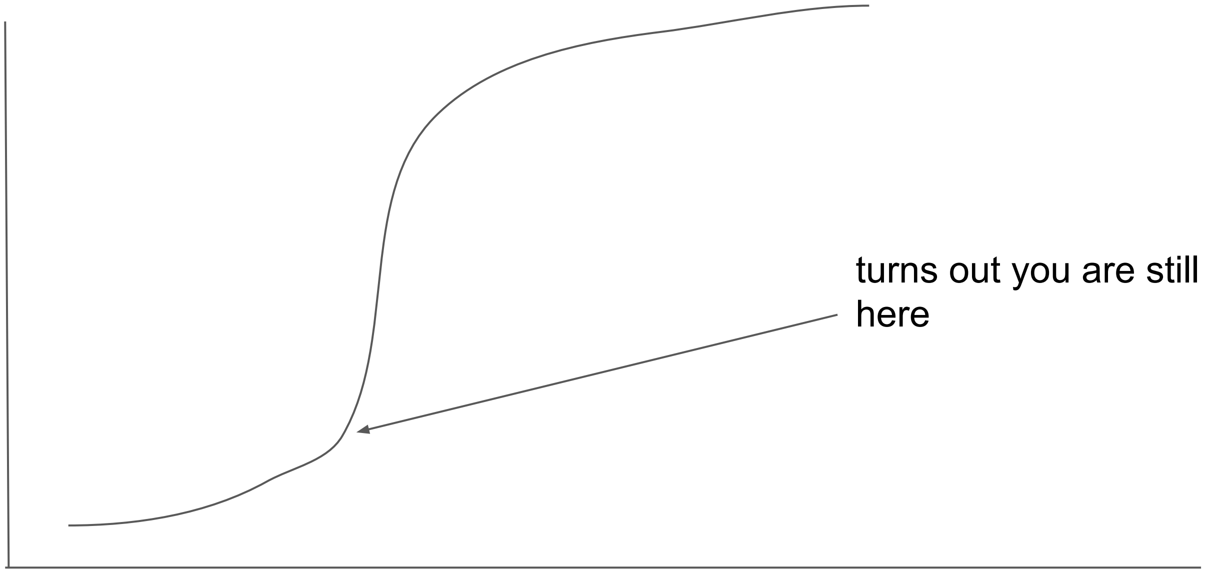
@sh1mmer

Uber Growth (a super scientific graph)



every time you think
you are here

Uber Growth (a super scientific graph)



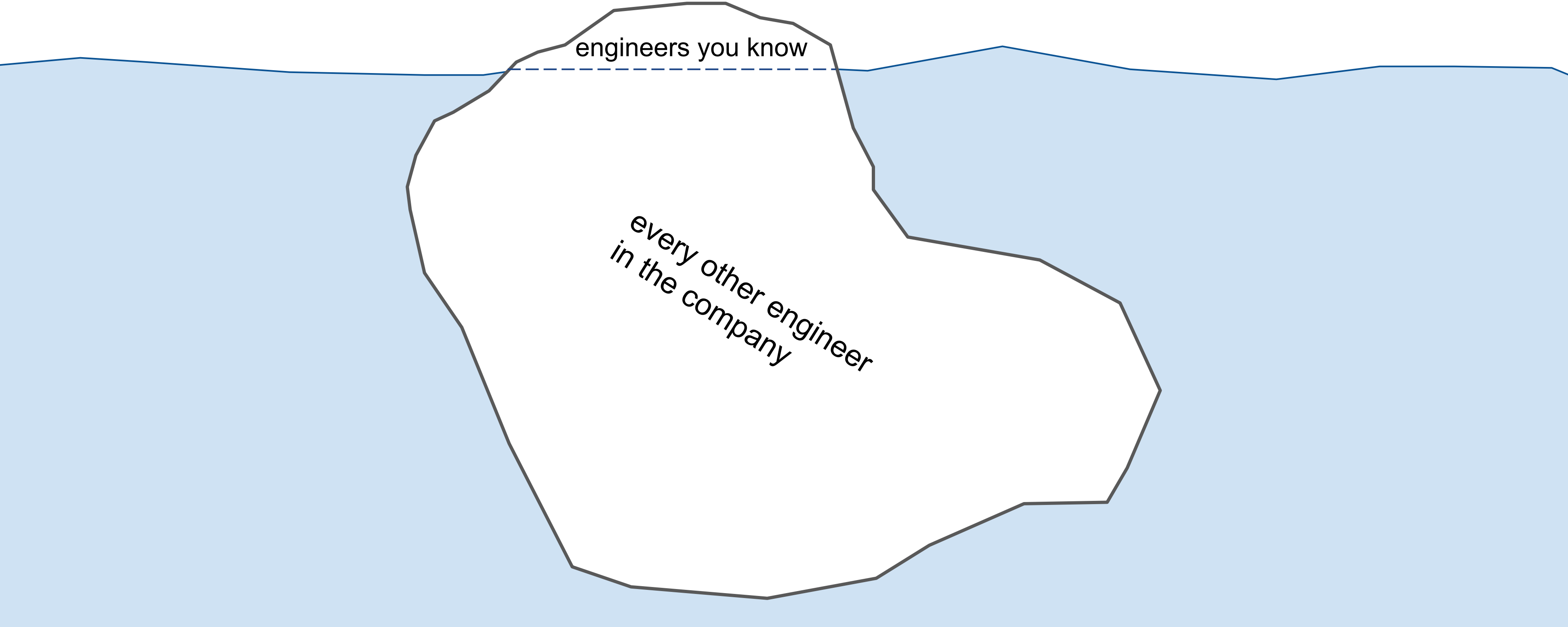
turns out you are still here

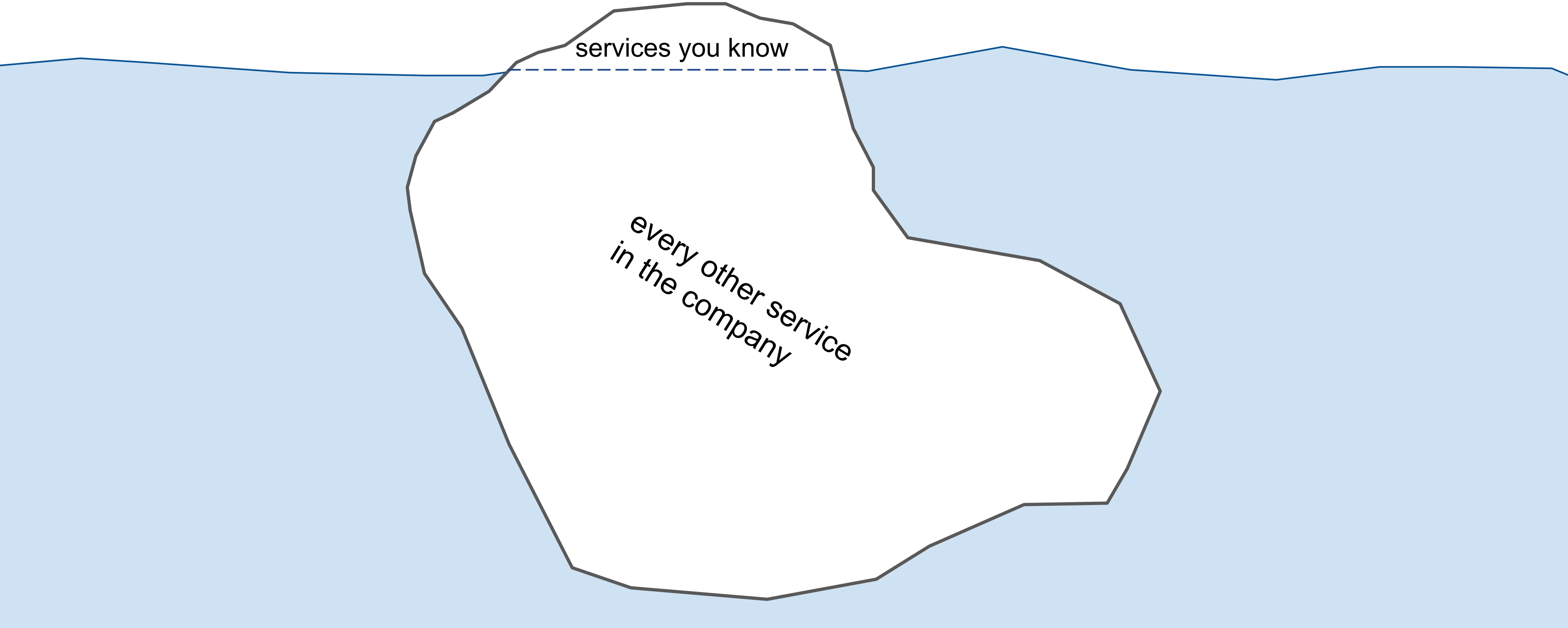
uber is complicated

a bit like my code

How does a rider from **Bangalore**, travelling in **New York**, use Facebook messenger to request a ride via *api.uber.com*?

How does an Uber Engineer build a reliable feature **that** supports this without a building feature specifically **to** support this?


engineers you know
every other engineer in the company



services you know

every other service in the company

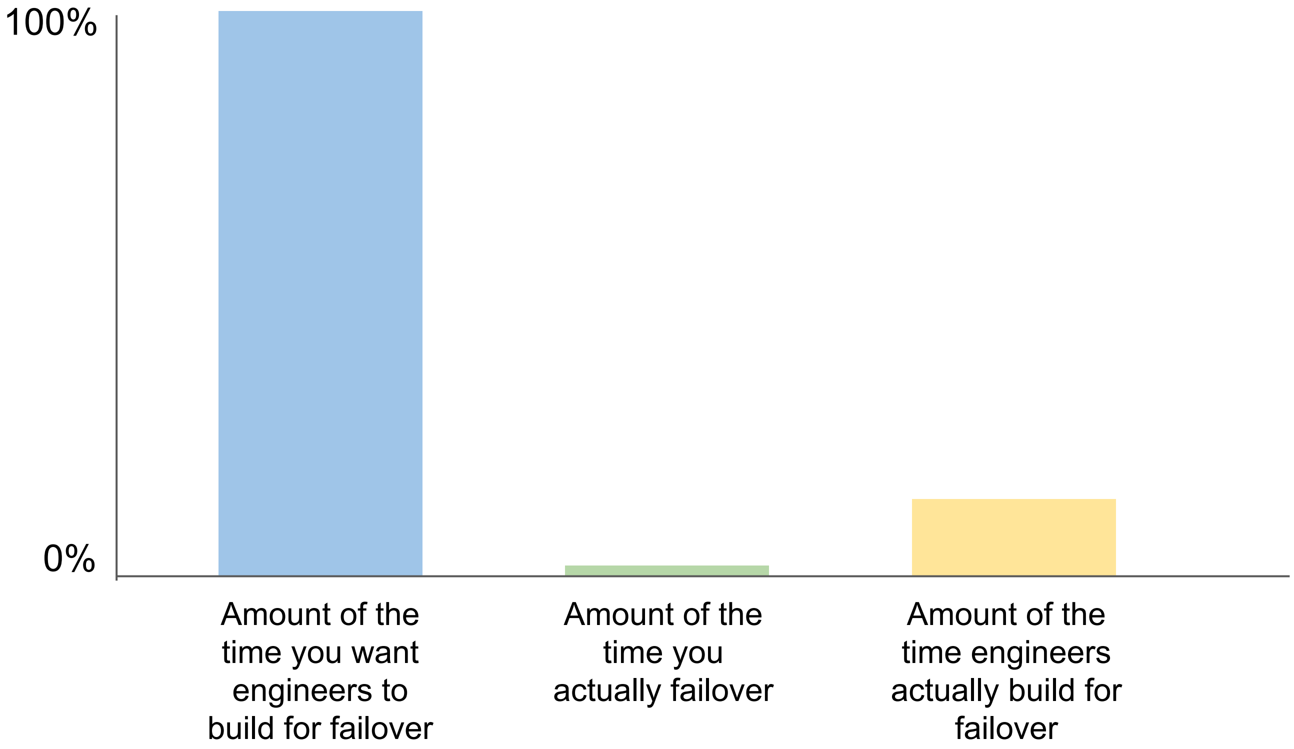
the school of hard knocks

i have 99 problems, and reliability is 1

UBER

creating new realities

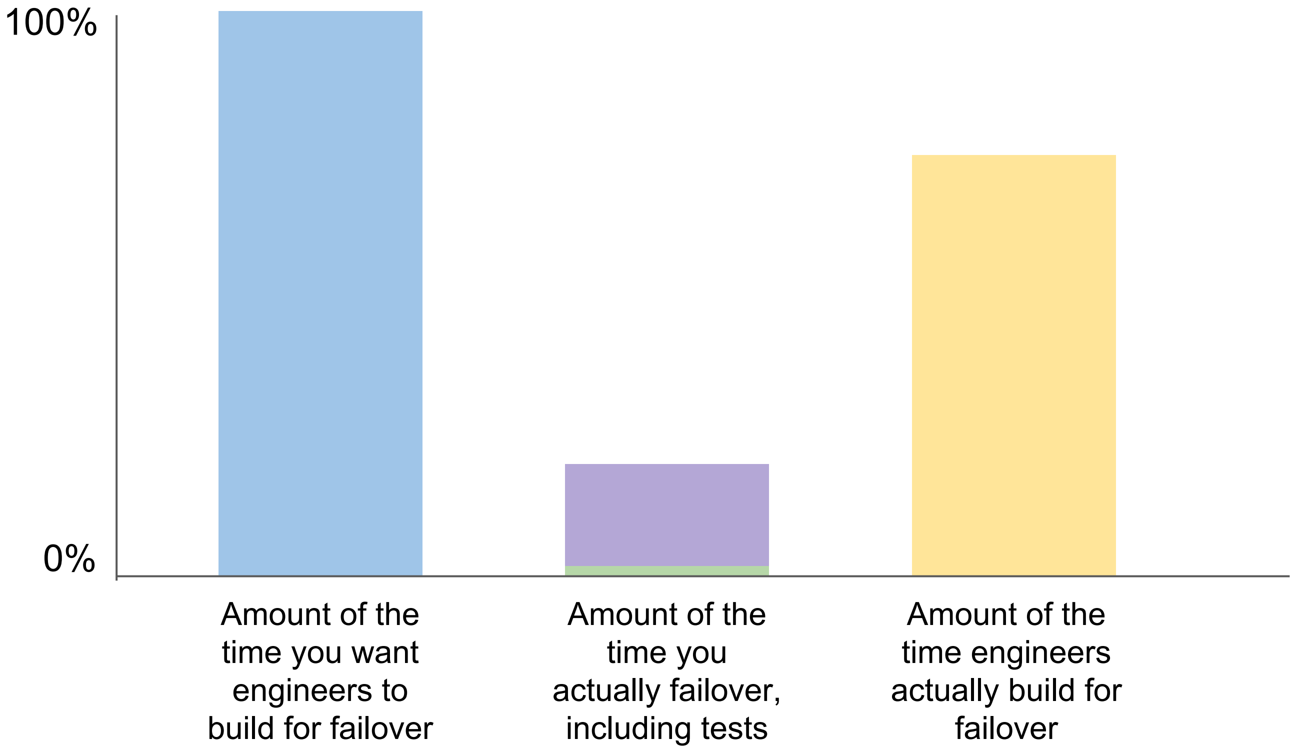
like Steve Jobs but for infrastructure


100%
0%
Amount of the time you want engineers to build for failover
Amount of the time you actually failover
Amount of the time engineers actually build for failover

UBER

failover “testing”

aka you just lost all cooling in your China datacenter


100%
0%
Amount of the time you want engineers to build for failover
Amount of the time you actually failover, including tests
Amount of the time engineers actually build for failover

UBER

1) Humans (Sebastian) inducing failure

2) Baseline for repetition

- Reliable Core Services: RealTime Trip Replication (RTTR)

- Reliable Failover Tools: Zombie Apocalypse Recovery (ZAR)

3) Self-service

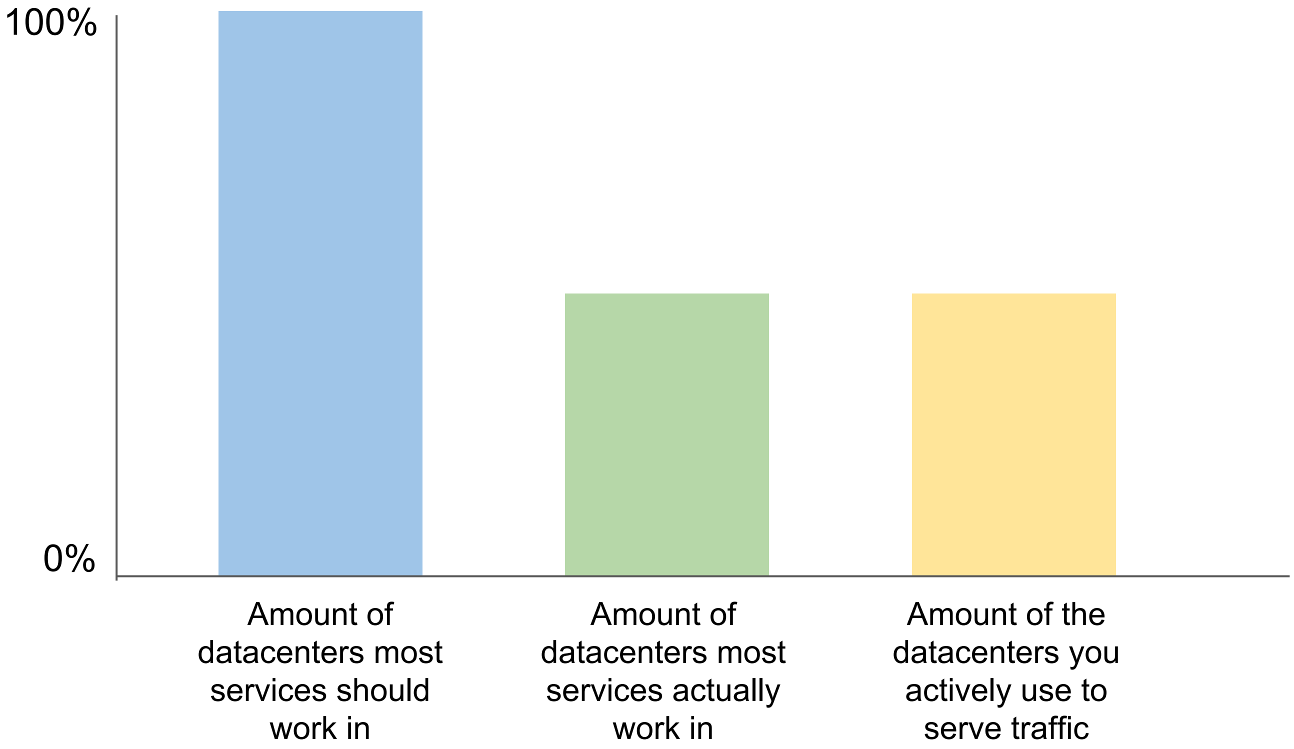
UDESTROY

- Automated destruction

- Randomized tests

ask for what you want

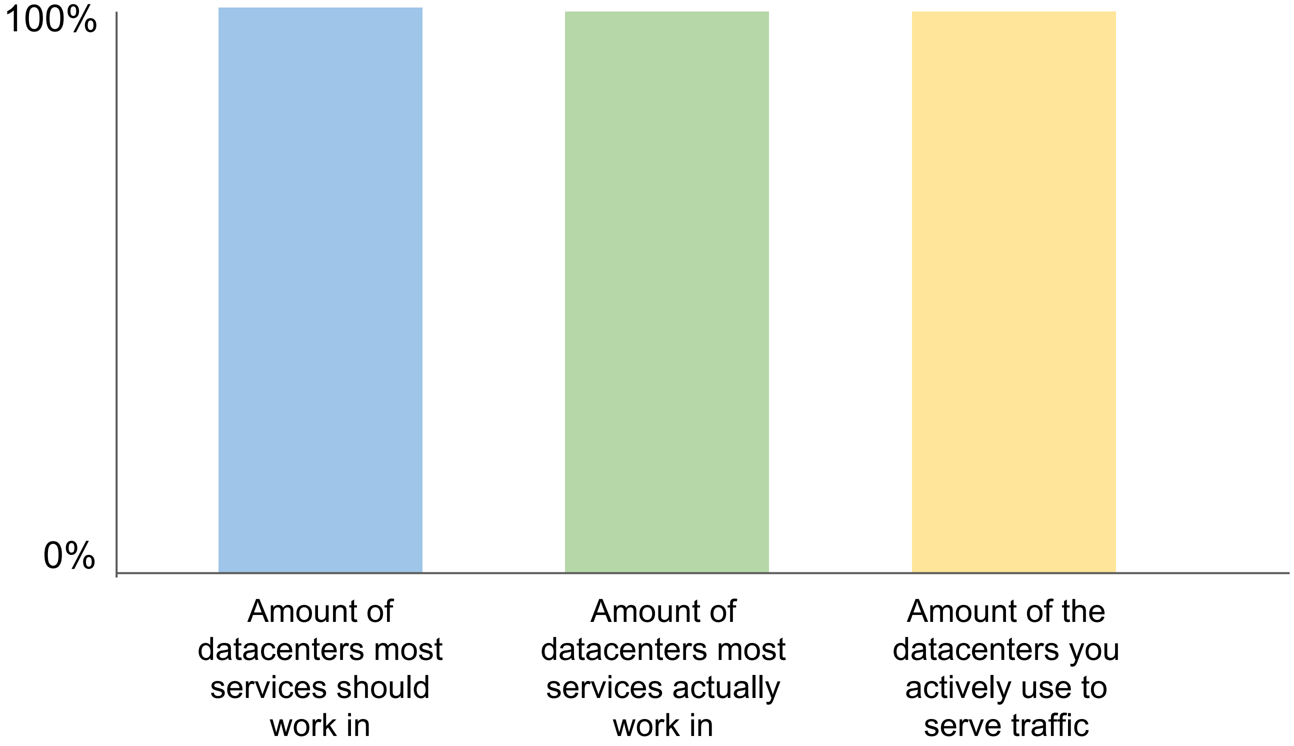
like a pony


100%
0%
Amount of datacenters most services should work in
Amount of datacenters most services actually work in
Amount of the datacenters you actively use to serve traffic

UBER

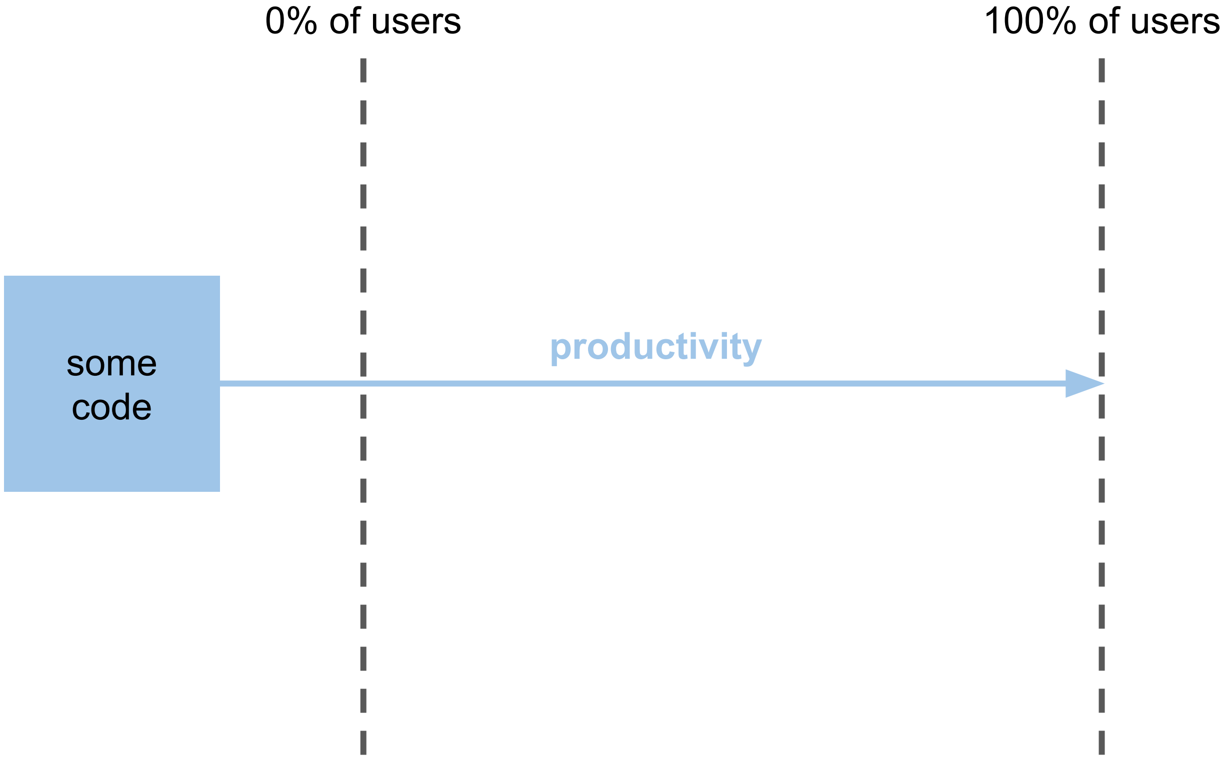
all datacenter reality

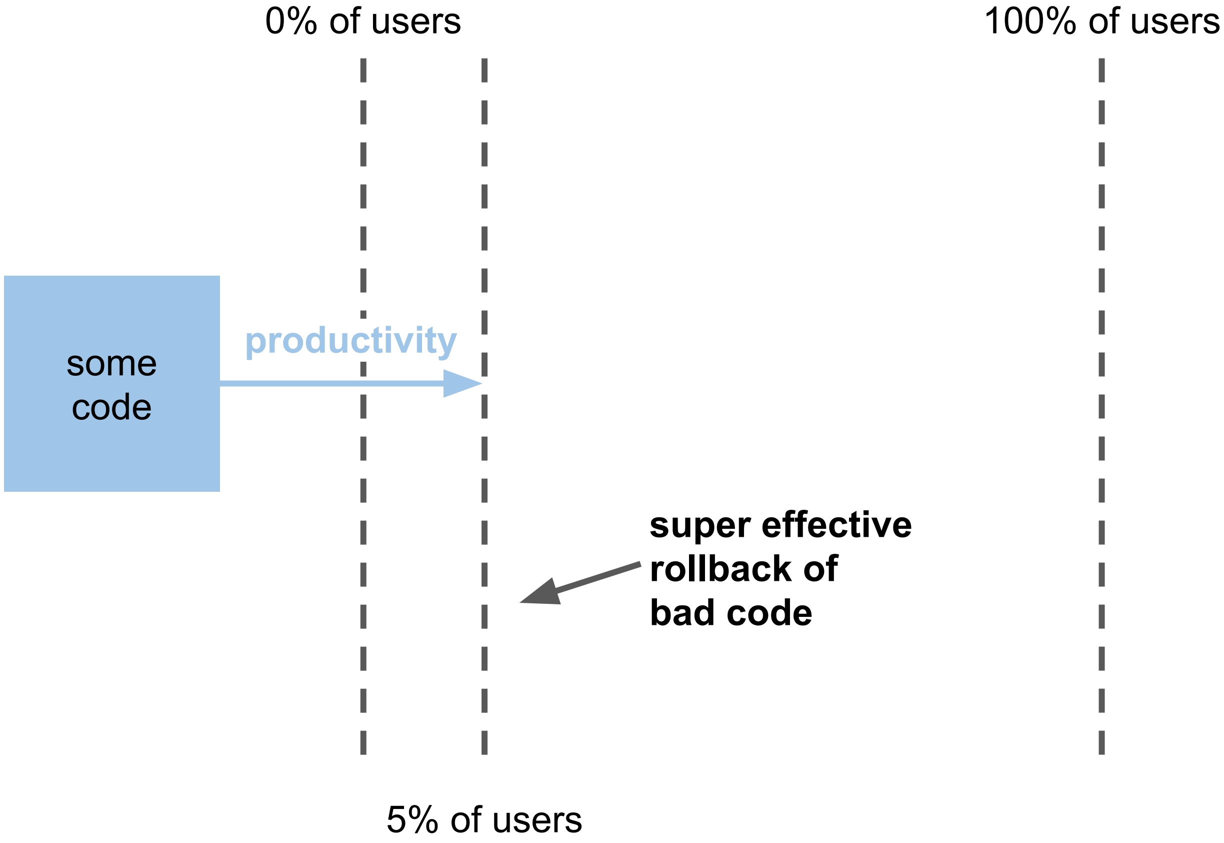
all requirements all the time


100%
0%
Amount of datacenters most services should work in
Amount of datacenters most services actually work in
Amount of the datacenters you actively use to serve traffic

what even is productivity

it’s probably measured in $%#! / minute


0% of users
100% of users
some code
productivity



0% of users

100% of users

some code

productivity

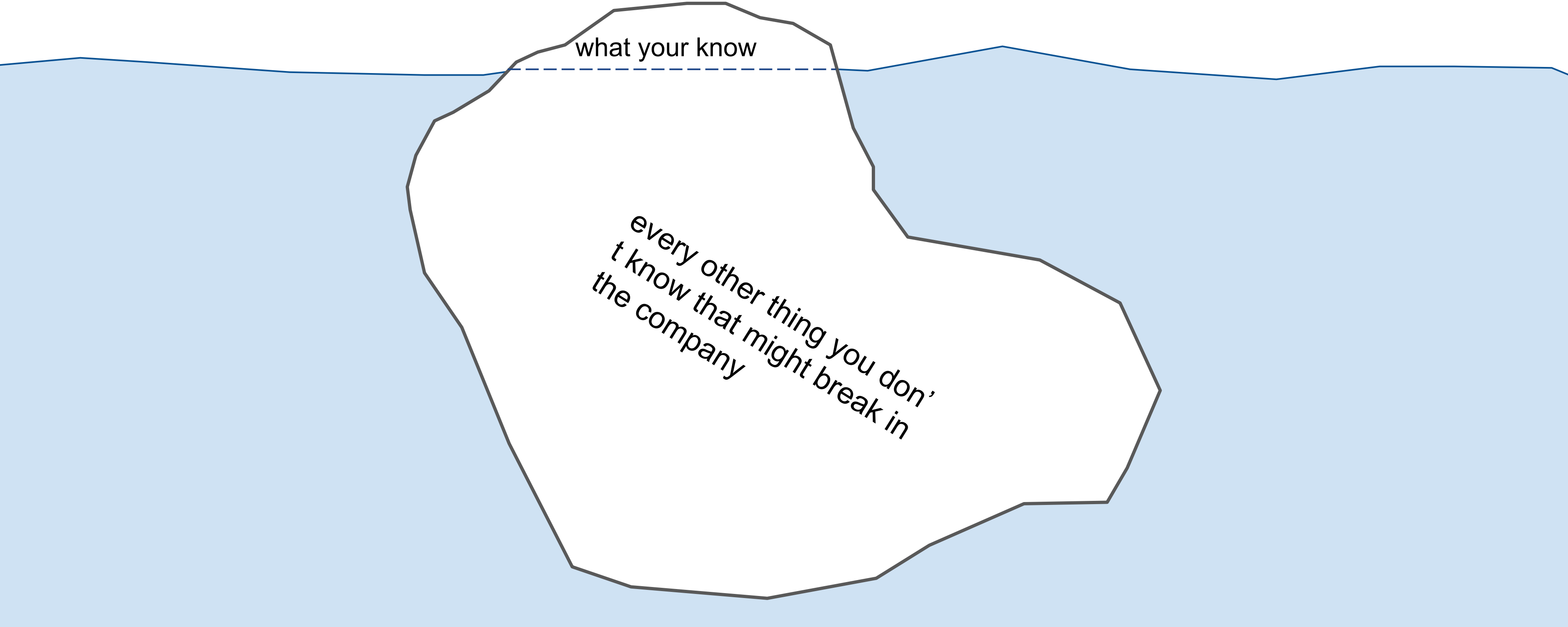
super effective rollback of bad code

5% of users

you don’t know anything

and you never will

UBER



what your know

every other thing you don’
t know that might break in
the company

Universal Guide to Stuff Just Works Now™

1) Make sure you have a place to move traffic that is not broken (most of the time)

2) Get good at knowing that users are having a bad time (why is less important)

3) Get really good at moving traffic to a safe place fast (ideally automatically)

3b) Fix whatever is broken, I guess

Questions?

tomc@uber.com

@sh1mmer

We’re hiring!

http://t.uber.com/srejobs