

“YOU BUILD IT, YOU RUN IT”

—Werner Vogels, Amazon

“...IT ALSO BRINGS THEM INTO DAY-TO-DAY CONTACT WITH THE CUSTOMER. THIS CUSTOMER FEEDBACK LOOP IS ESSENTIAL FOR IMPROVING THE QUALITY OF THE SERVICE.”

—Werner Vogels, Amazon

MEANWHILE,
IN LONDON...

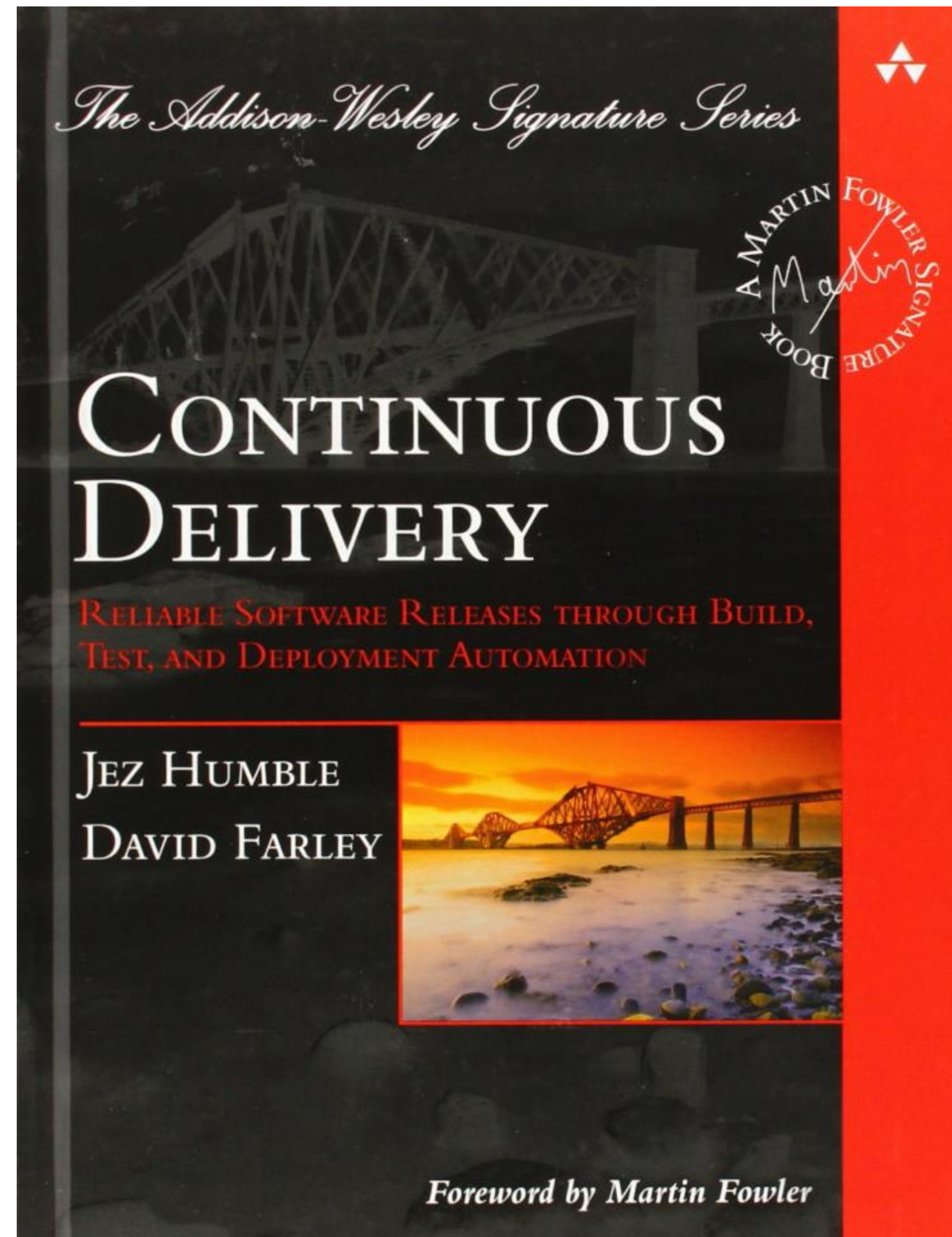
- ▶ Defined Continuous Integration
- ▶ Created the first (or second) CI server
- ▶ Created Selenium

JAVA – WRITE ONCE, RUN ANYWHERE



- ▶ Developed a system on Windows laptops to be deployed to a Solaris cluster
- ▶ Did all of the right Continuous Integration things
- ▶ One small issue...

IT DIDN'T WORK IN
PRODUCTION



**YOU ONLY HAVE TO
CHANGE ONE THING
TO DO THE DEVOPS**

**YOU ONLY HAVE TO
CHANGE ONE THING
TO DO THE DEVOPS**

EVERYTHING

- ▶ (Re)define words for your organization
- ▶ Change your organization to enable DevOps
- ▶ Use modern architectures and technologies
- ▶ Use Continuous Delivery to **safely** deploy on demand

WHAT ISN'T DEVOPS



- ▶ A Toolset
- ▶ A Role
- ▶ A Team

developING and operatING

- ▶ Culture
- ▶ Automation
- ▶ Measurement
- ▶ Sharing

<https://blog.chef.io/2010/07/16/what-devops-means-to-me/>

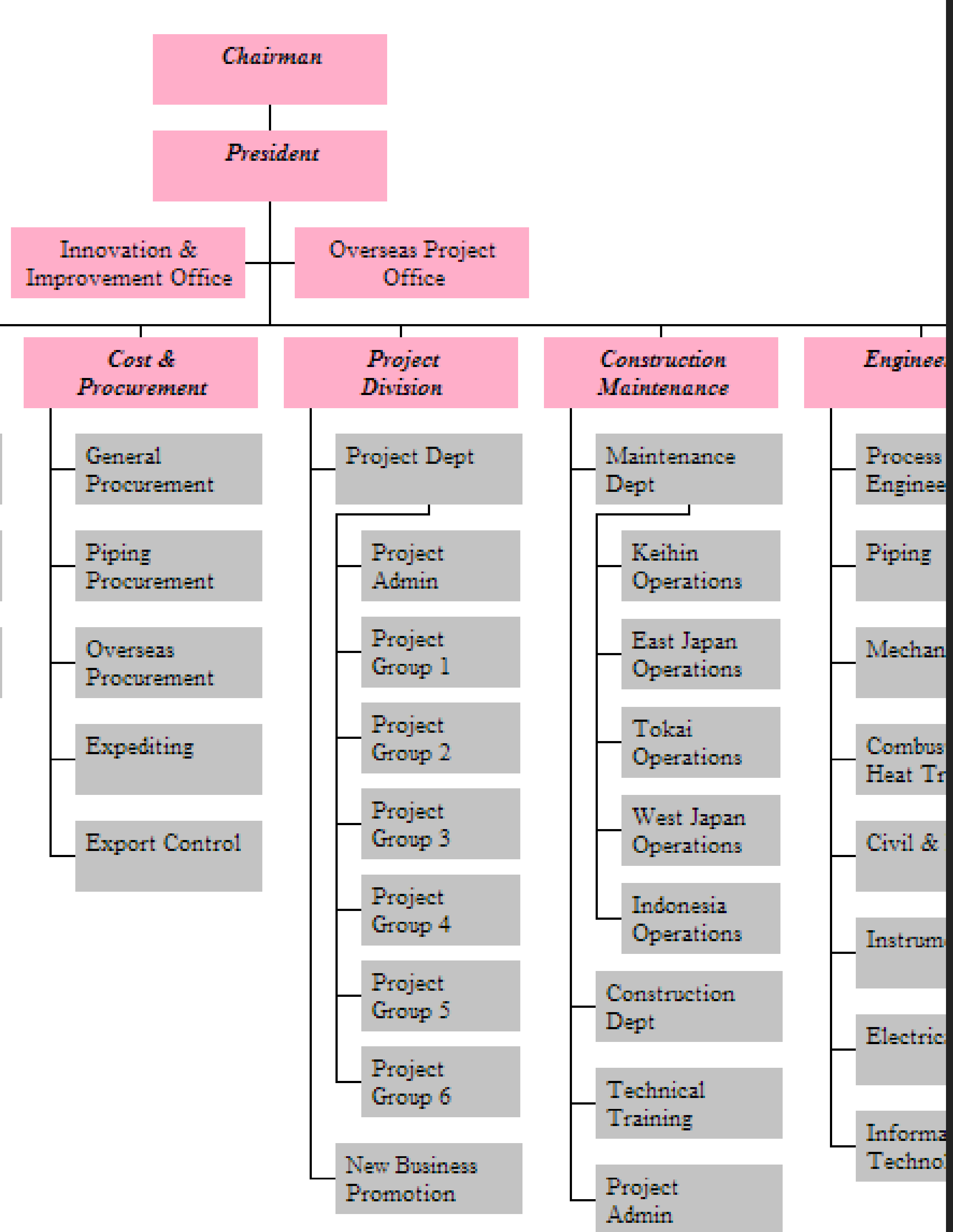
@kmugrage

- ▶ Culture
- ▶ Automation
- ▶ Lean
- ▶ Measurement
- ▶ Sharing

<https://blog.chef.io/2010/07/16/what-devops-means-to-me/>

“DEVOPS: A CULTURE WHERE PEOPLE, REGARDLESS OF TITLE OR BACKGROUND, WORK TOGETHER TO IMAGINE, DEVELOP, DEPLOY AND OPERATE A SYSTEM.”

—Me

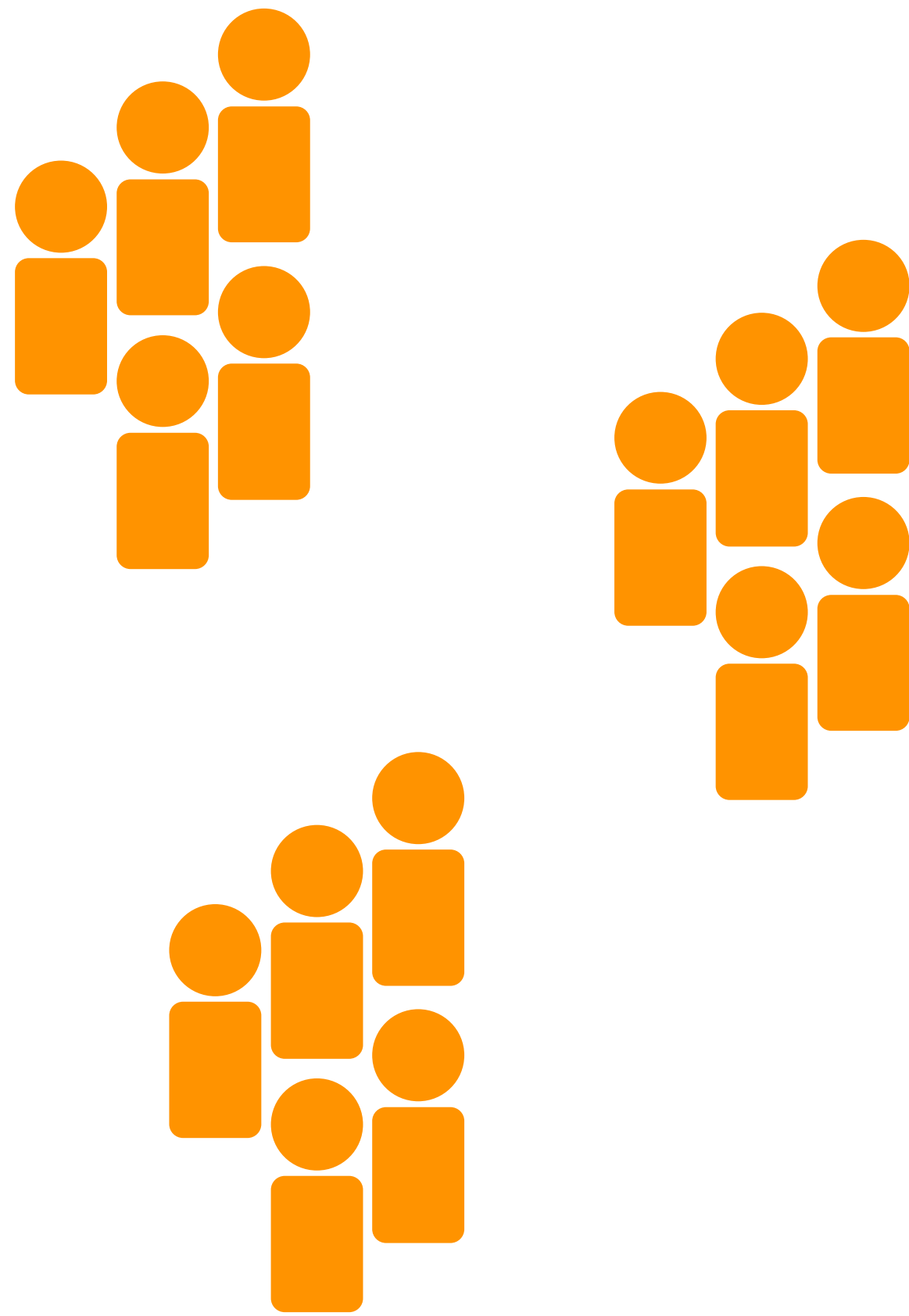


CHANGE YOUR ORG

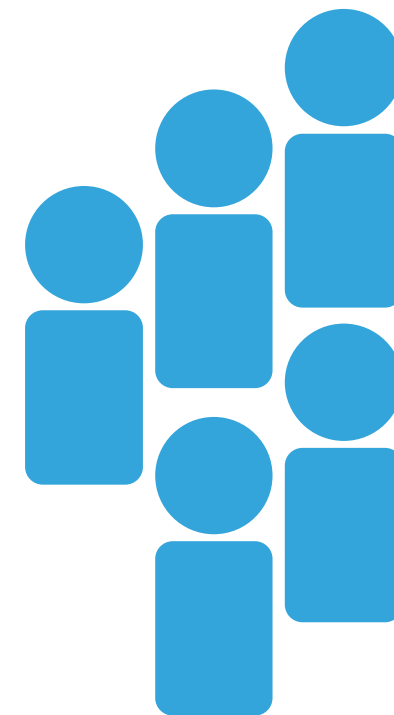
TO ENABLE DEVOPS

TRADITIONAL MODEL

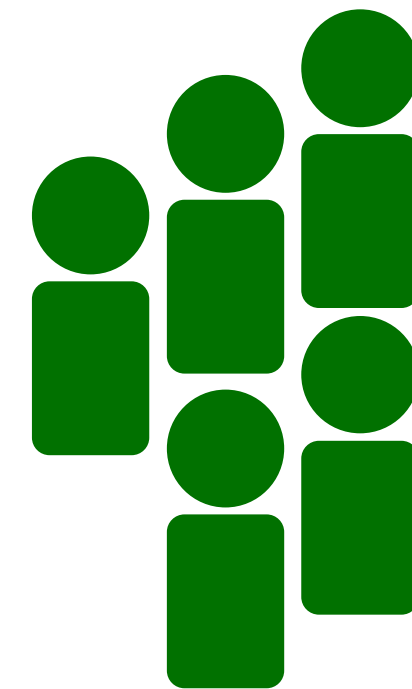
Development Teams



Testing Team



Operations Team

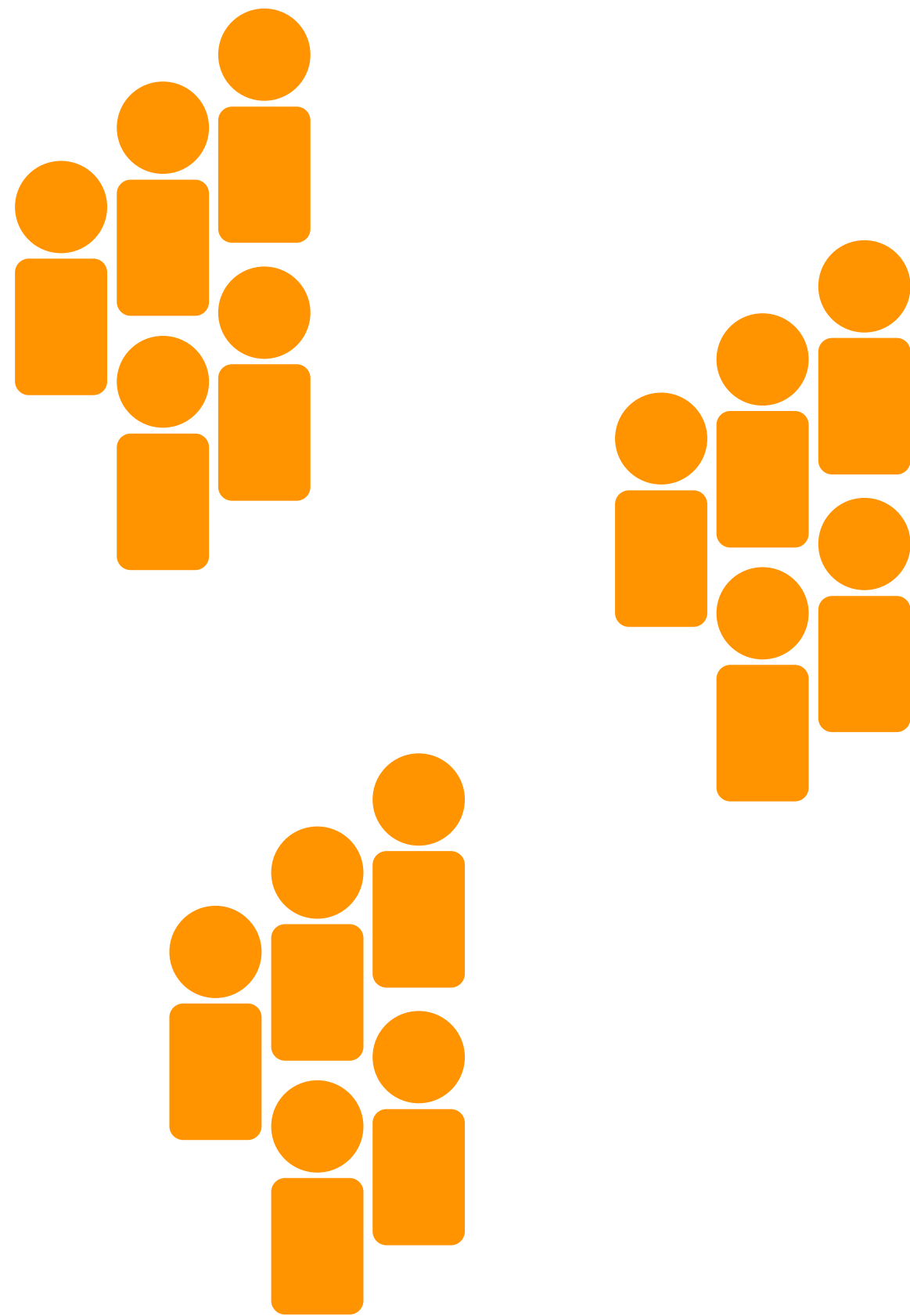


“ANY ORGANIZATION THAT DESIGNS A SYSTEM (DEFINED BROADLY) WILL PRODUCE A DESIGN WHOSE STRUCTURE IS A COPY OF THE ORGANIZATION'S COMMUNICATION STRUCTURE.”

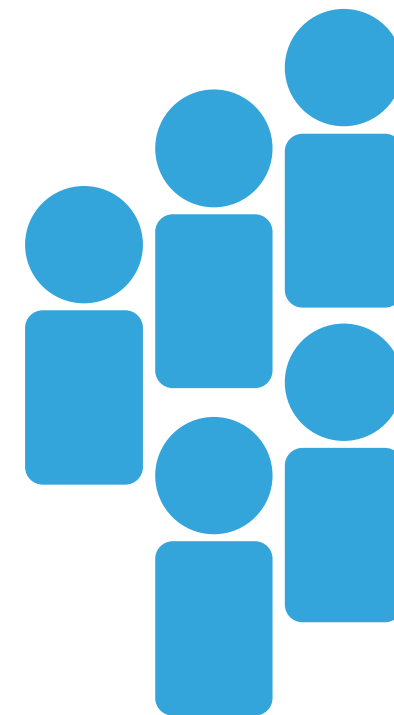
—Mel Conway

TRADITIONAL MODEL

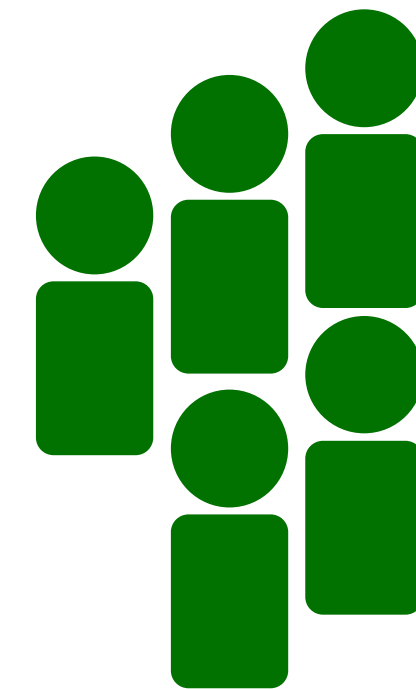
Development Teams



QA Team

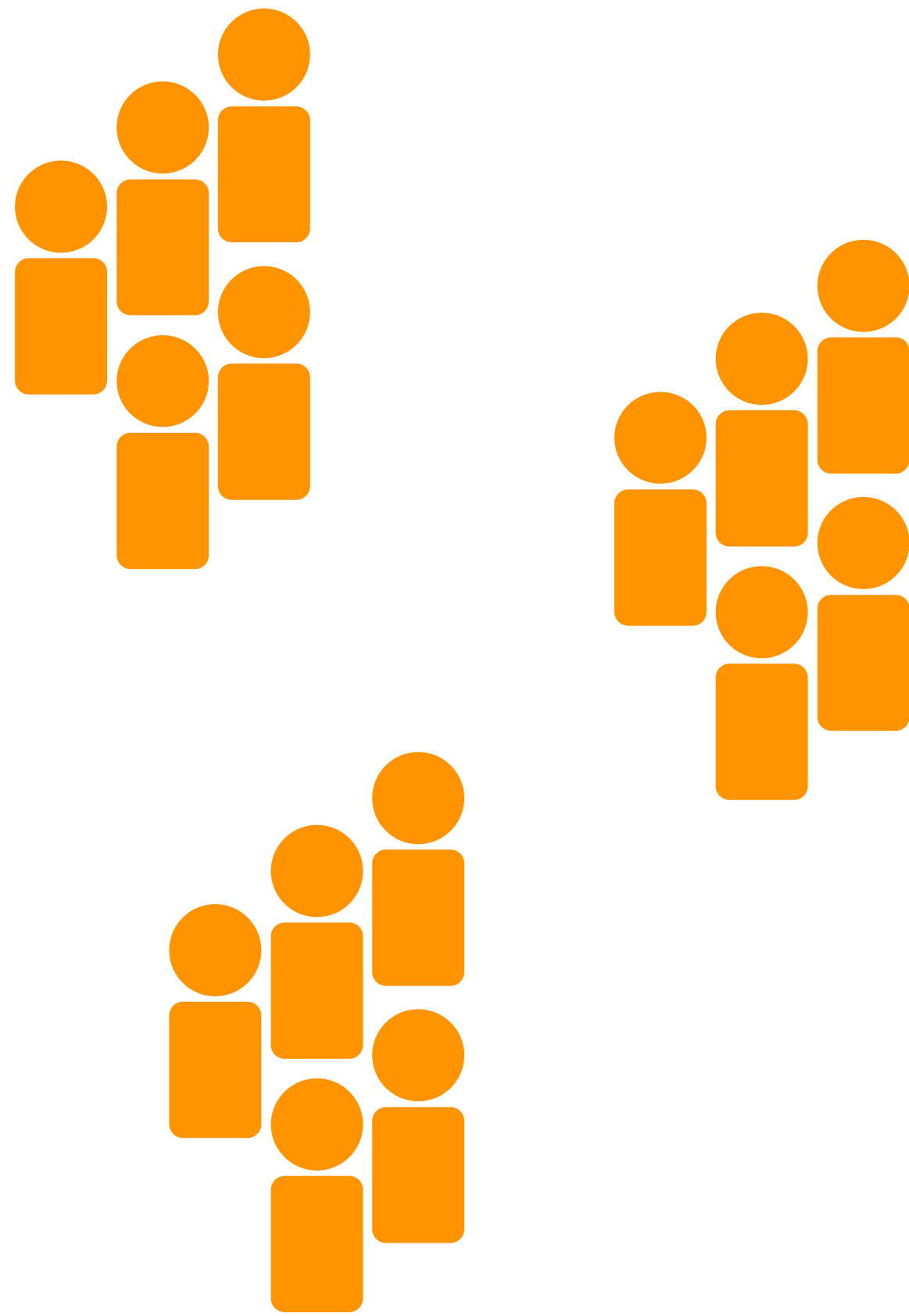


Operations Team

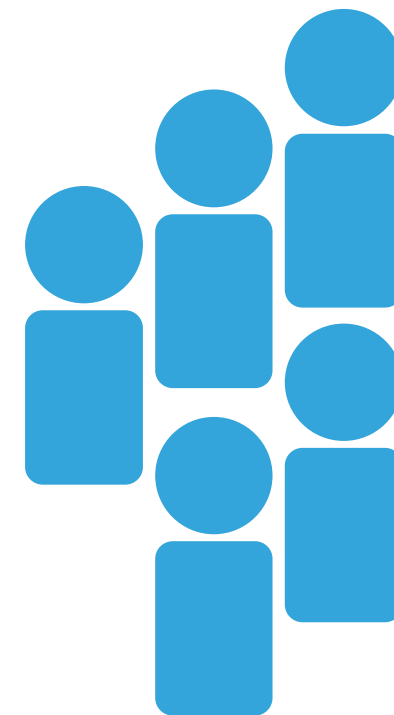


RENAMING OPS WON'T SOLVE THE ISSUE

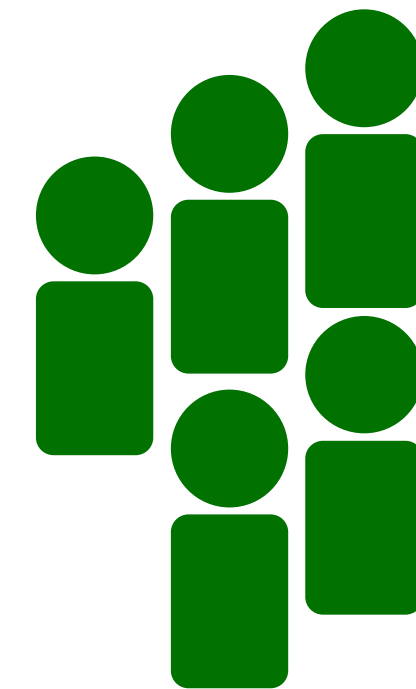
Development Teams



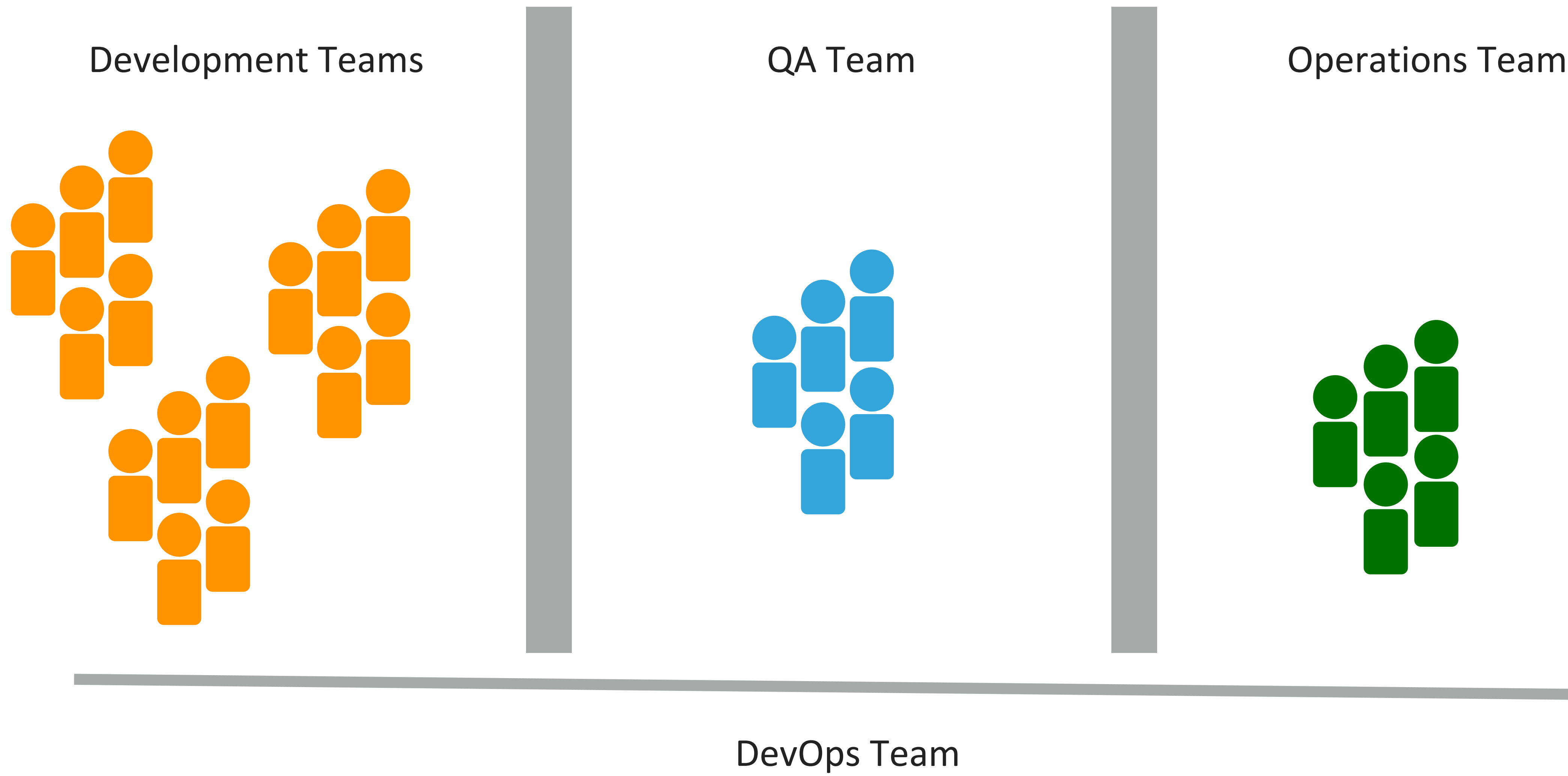
QA Team



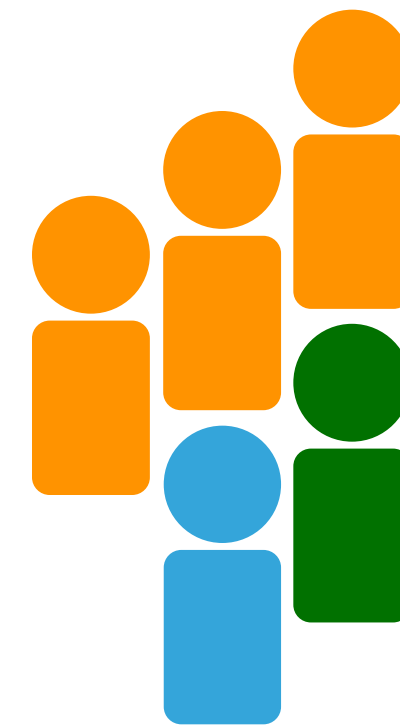
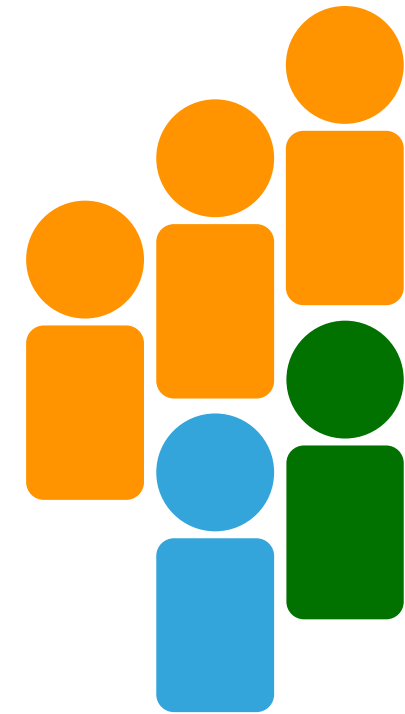
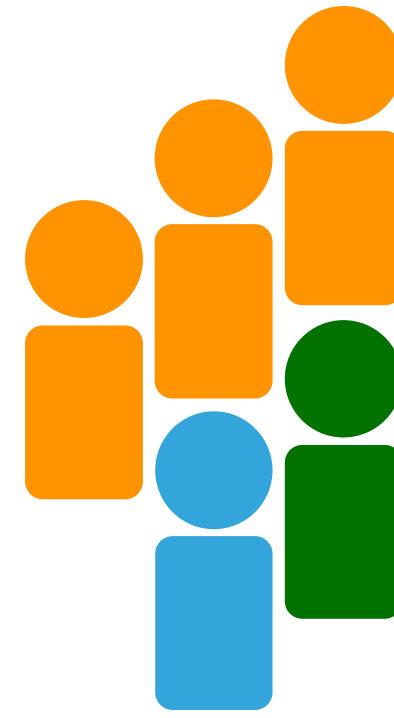
DevOps Team



CREATING ANOTHER SILO DOESN'T SOLVE SILOS



PRODUCT TEAMS





USE MODERN

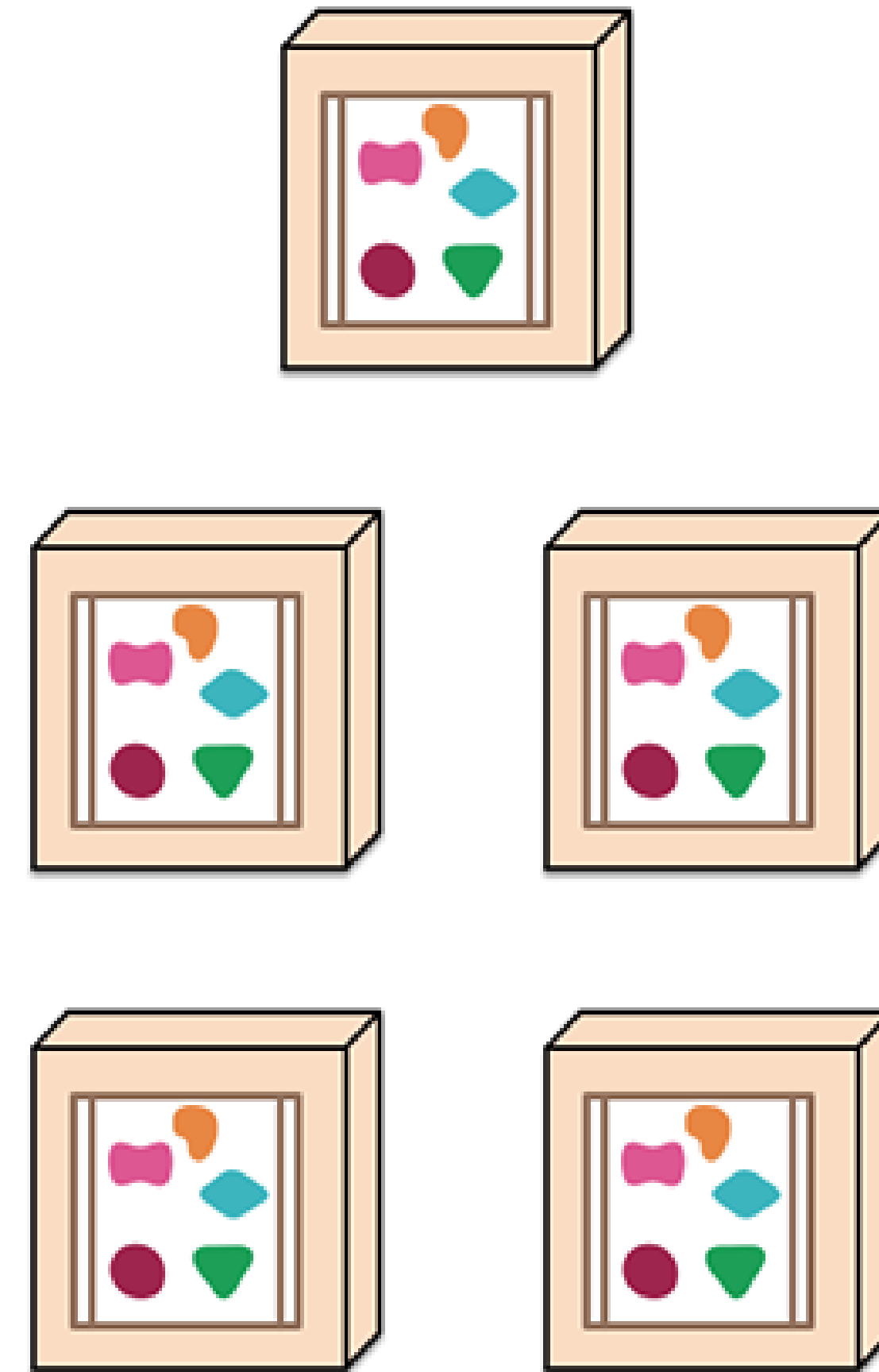
ARCHITECTURES
AND
TECHNOLOGIES

THERE'S A PHRASE THAT DEFINES
LEGACY SOFTWARE. "THE STUFF
THAT RUNS OUR BUSINESS"

Me, and a lot of other people

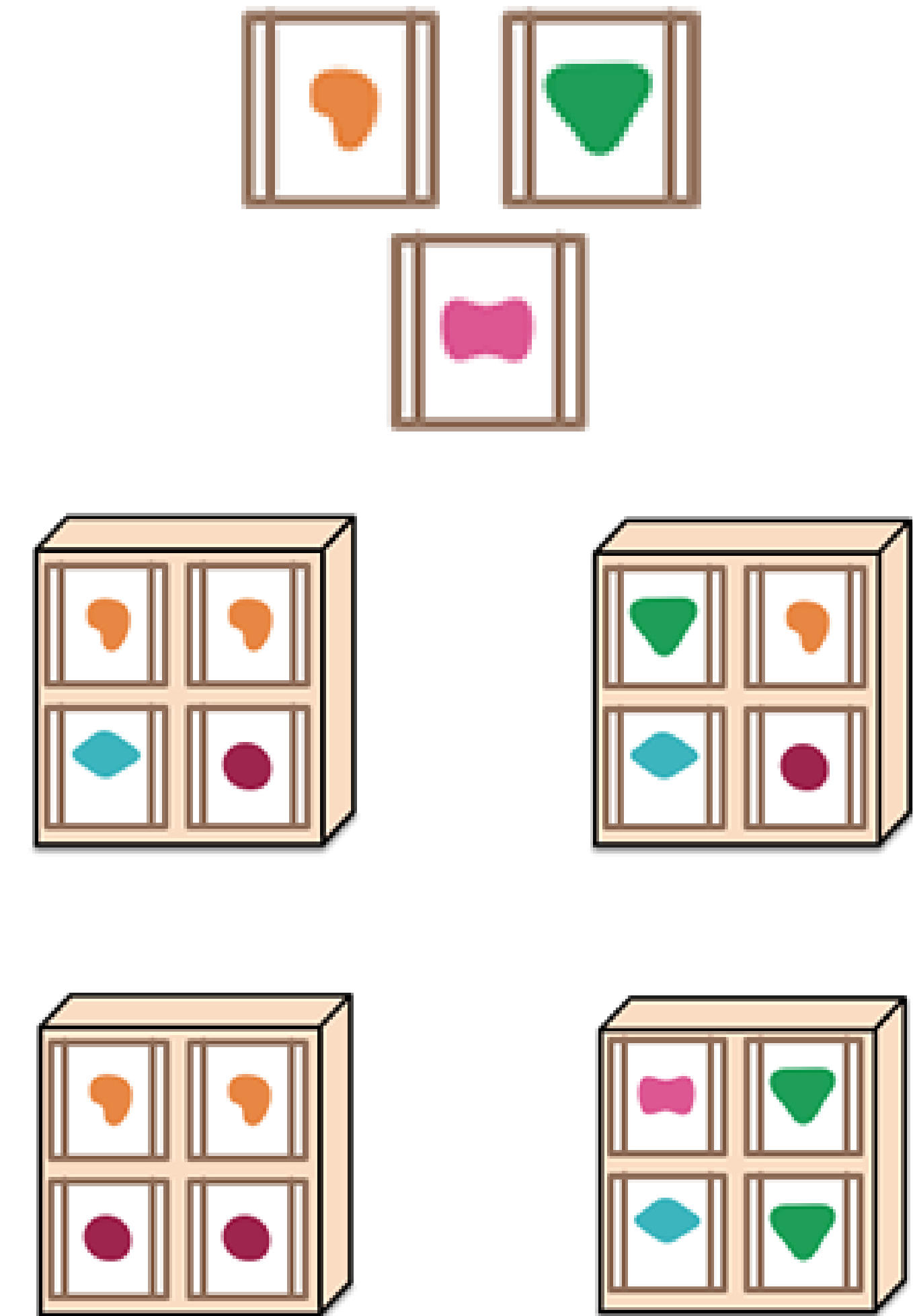
MONOLITHS CAN BE HARD

- ▶ All functionality is in one process
- ▶ Scale by replicating the monolith on multiple servers



MICROSERVICE ARCHITECTURE

- ▶ Each element of functionality is in a separate service
- ▶ Scale by distributing these services across servers, replicating as needed



- ▶ Need to deploy specific versions at the same time? Not a microservice.
- ▶ Have a shared data source? Not a microservice.

DON'T REBUILD THE WHOLE THING!



- ▶ Build services **only** when you understand the boundaries
- ▶ Build services where you need to react faster
- ▶ Build services where you need scalability

DON'T REBUILD THE WHOLE THING!



Silvia Botros @dbsmasher · Mar 19



Literally the more I learn about the current fancy new thing in infra the more I just wanna use bash

Am I principal engineering this right?



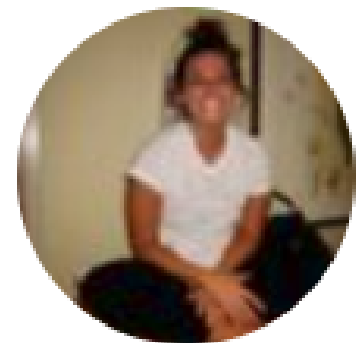
67



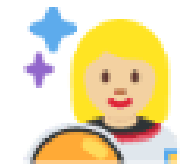
61



928



jessie frazelle



@jessfraz



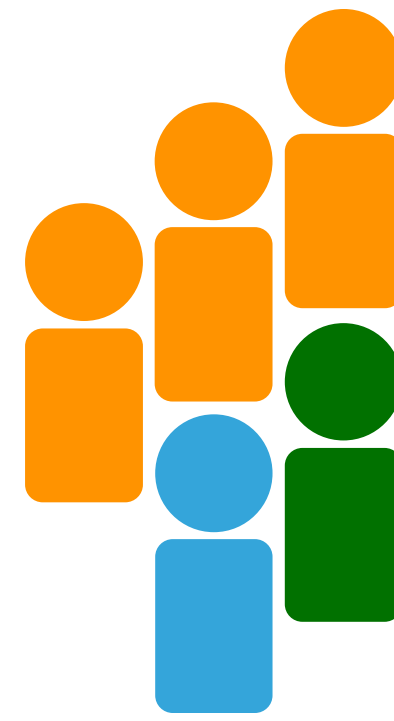
Replying to @dbsmasher

Yes

@kmugrage



Car Rental



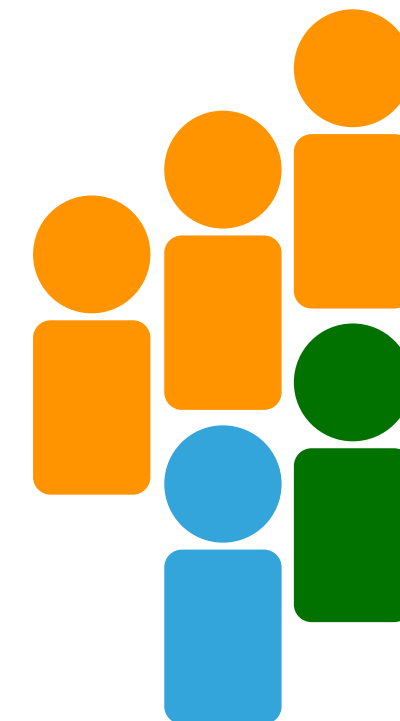
Tax
Payments



Consumer
Business to Business

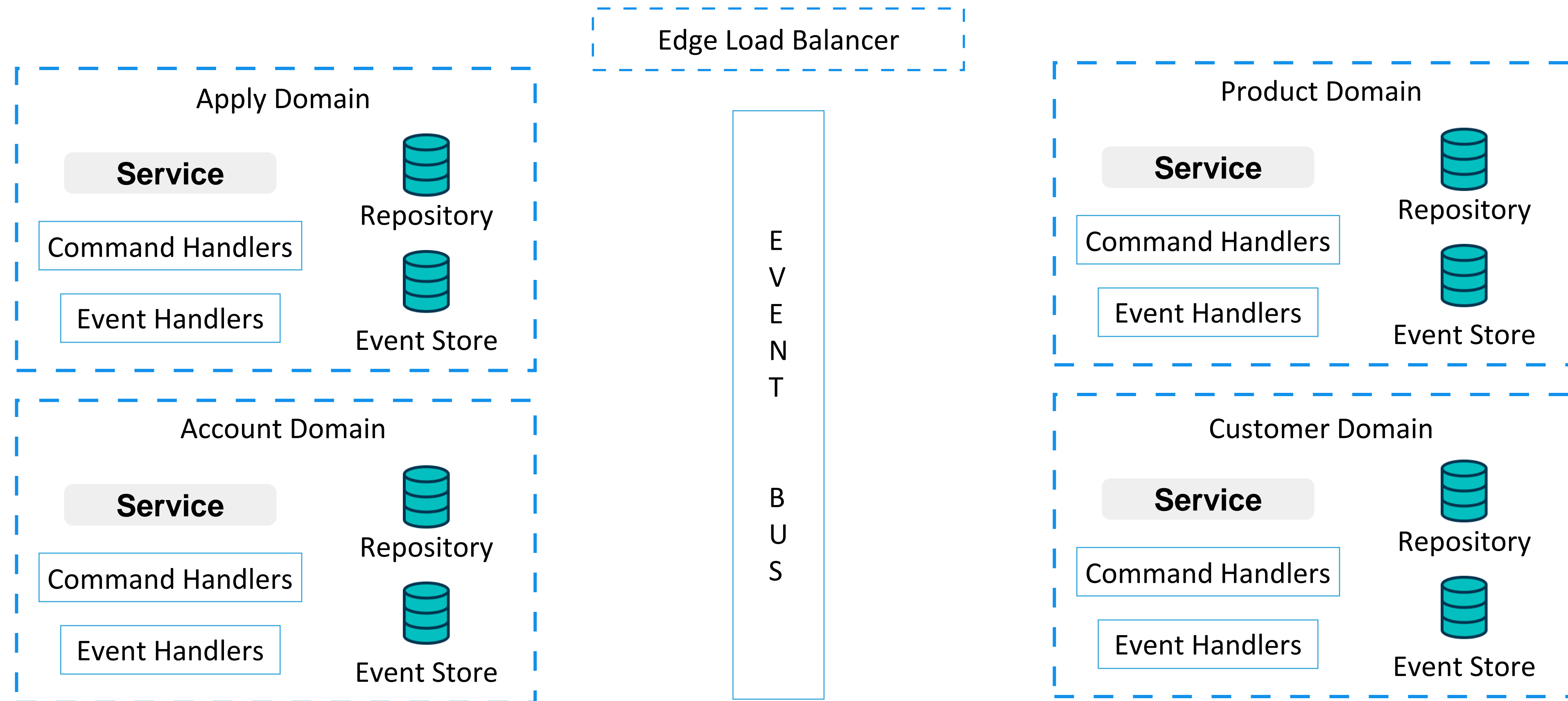


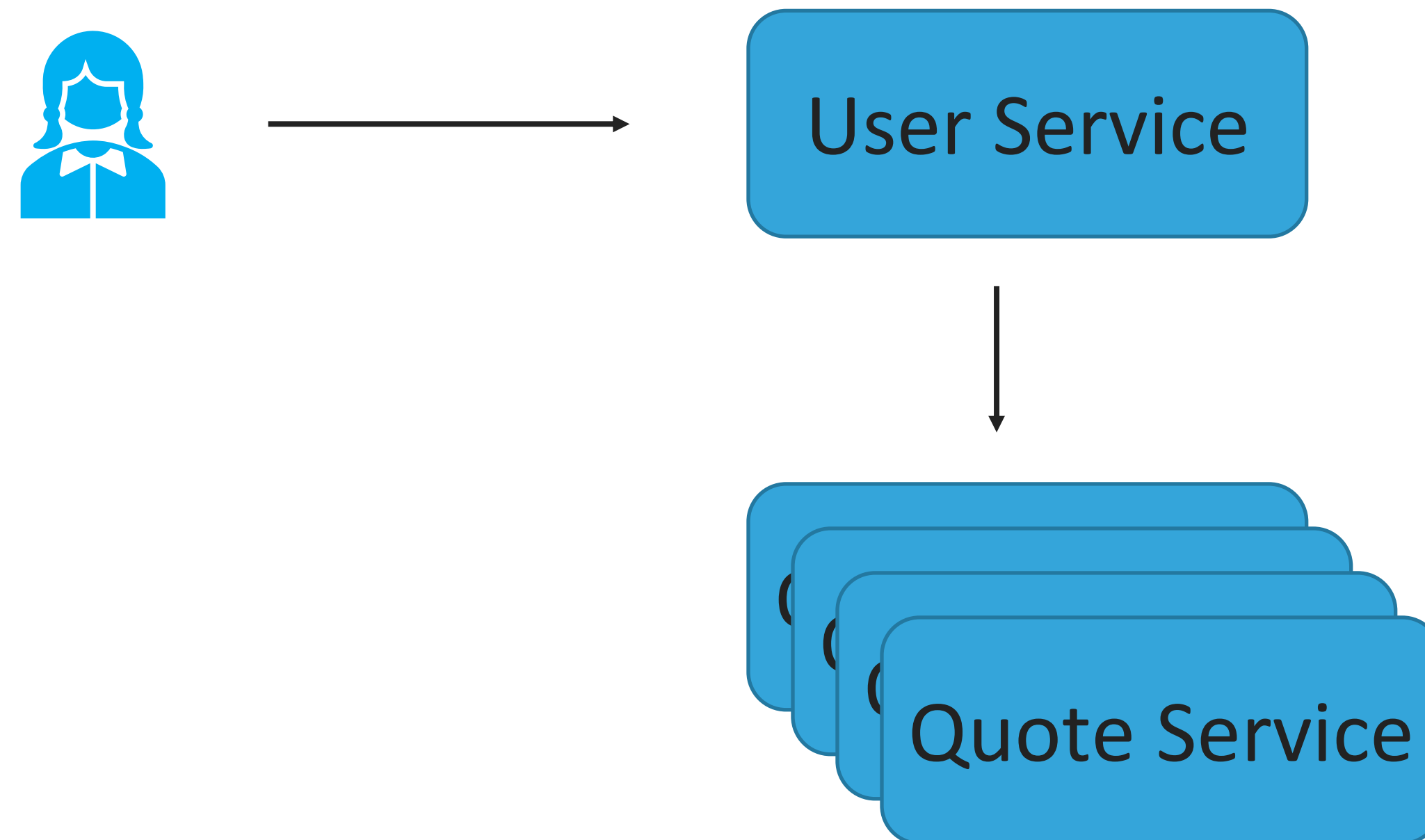
Hotels

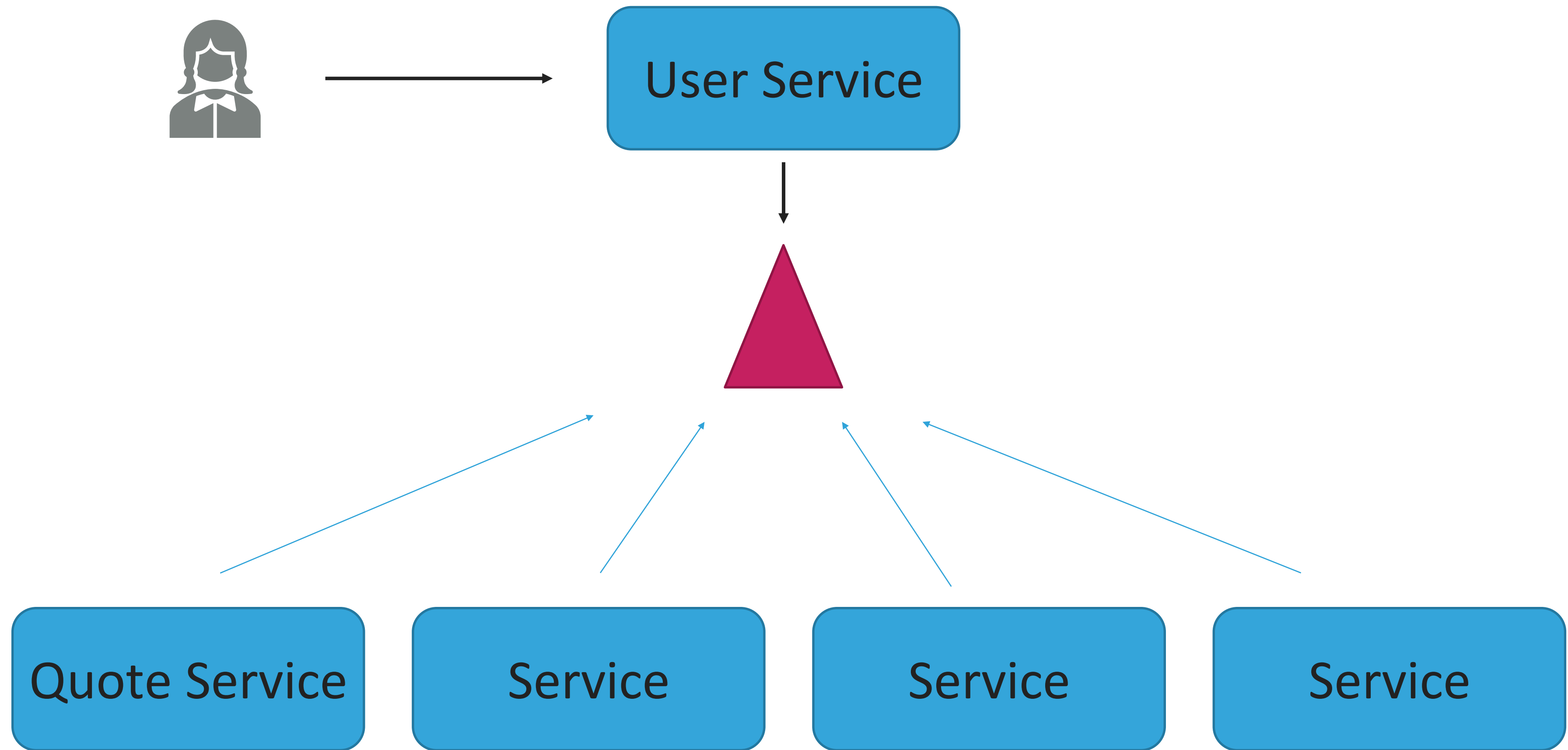


Airlines

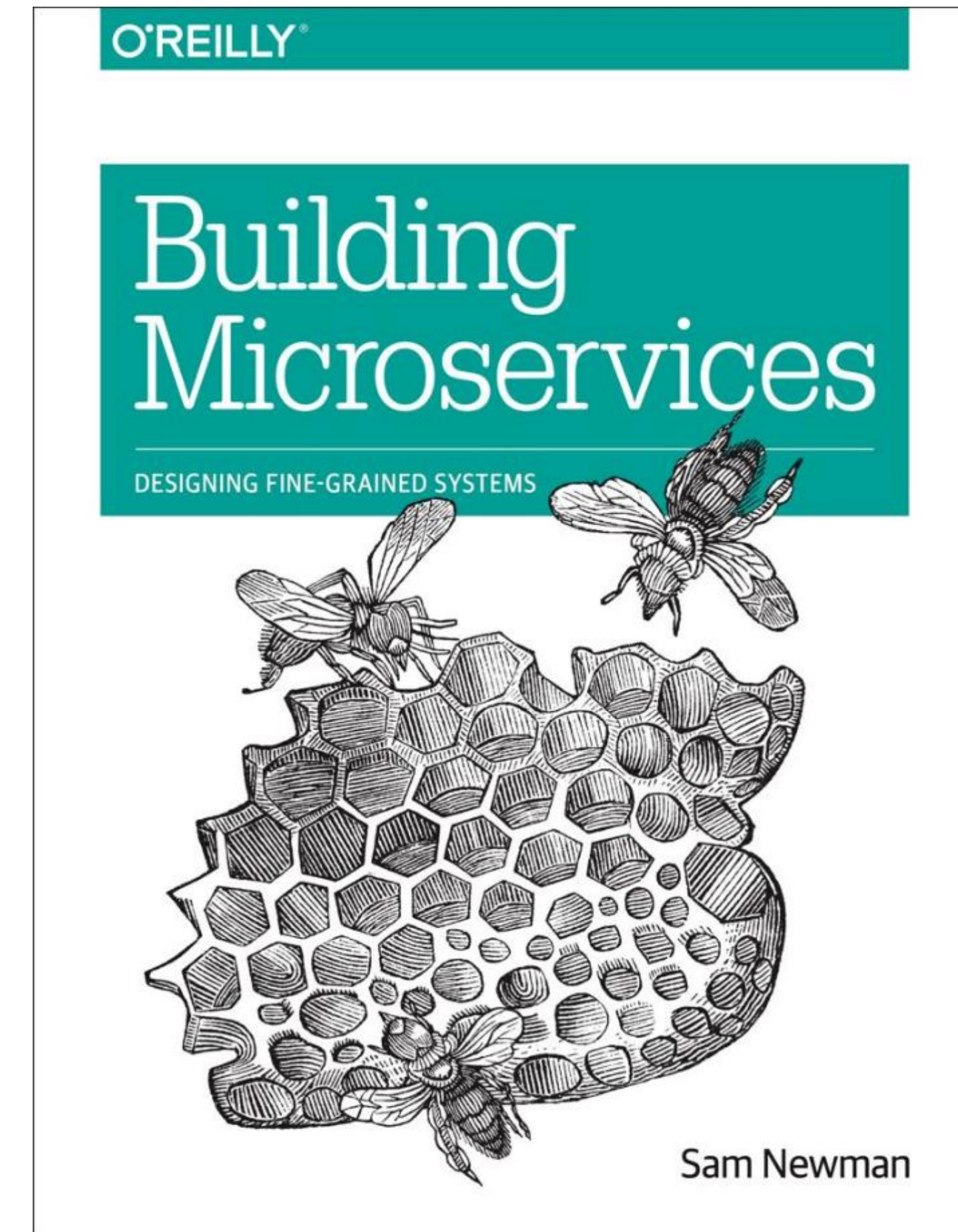
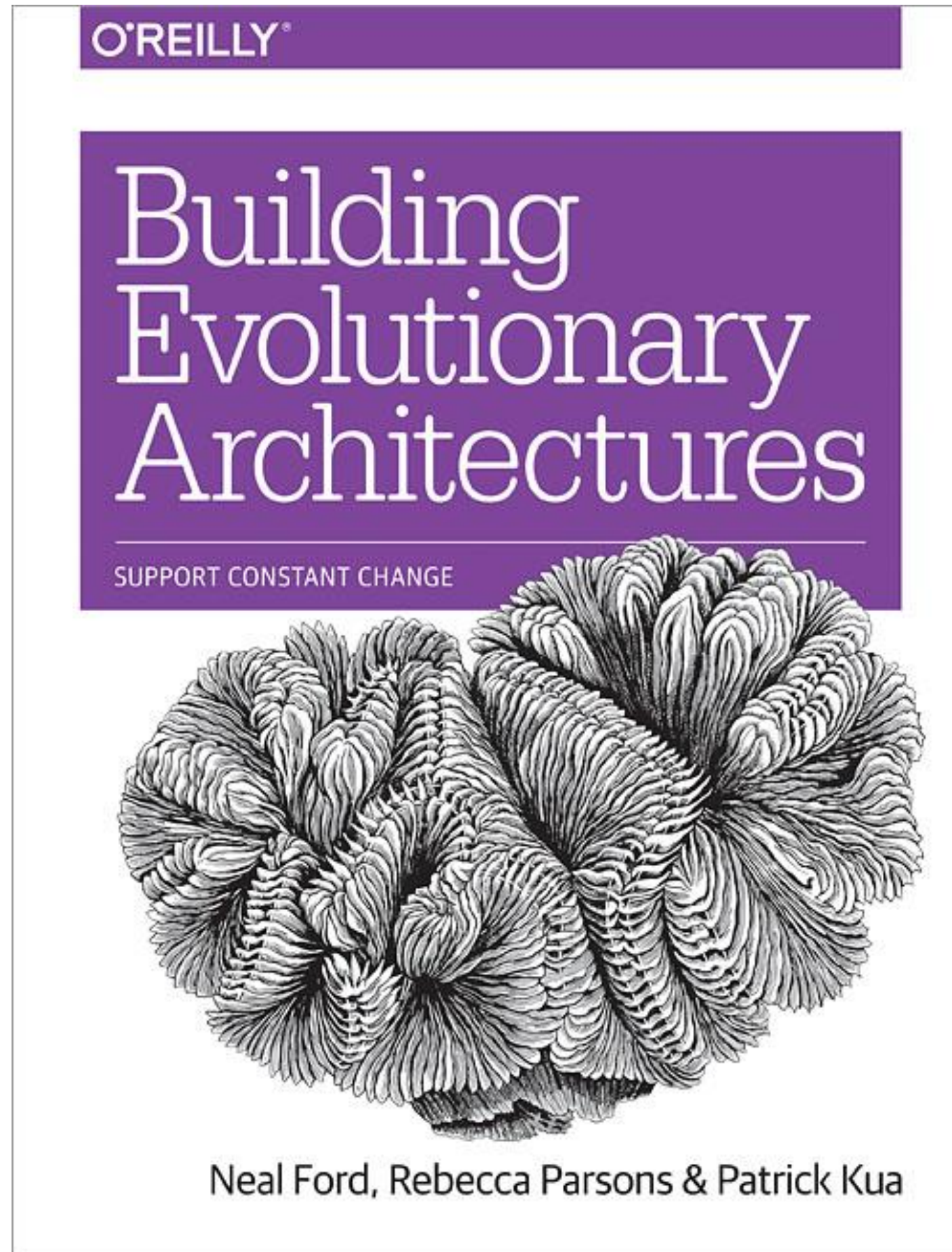
FINANCIAL SERVICES PLATFORM







- ▶ You don't write to the data store, you create an event which writes to the store
- ▶ The test: You could completely blow away the store and recreate it from the event stream
- ▶ You use this model every day (I hope)



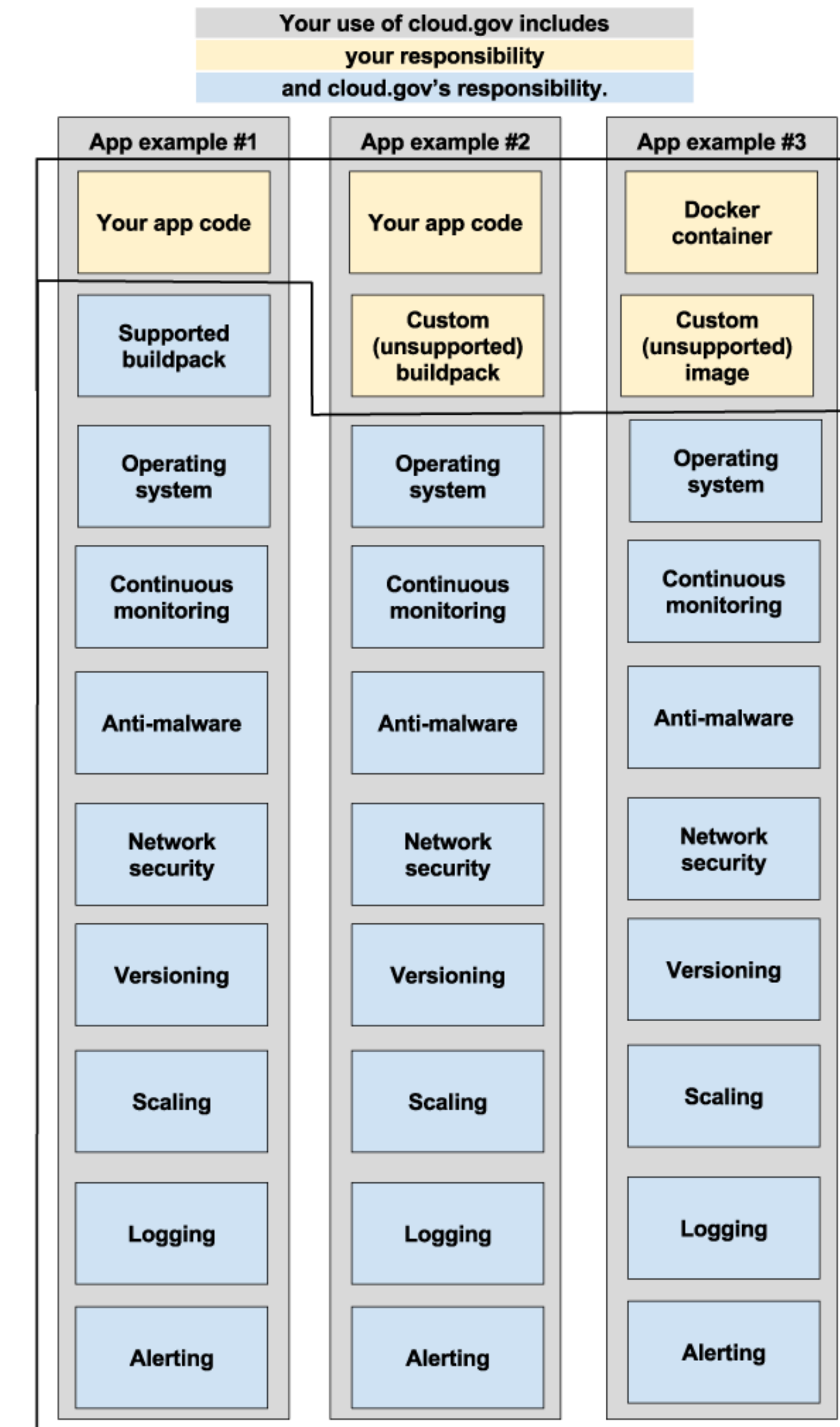
WE STILL HAVE TO DEPLOY
SOMEWHERE

AN EXAMPLE OF PLATFORM AS A SERVICE



CLOUD.GOV

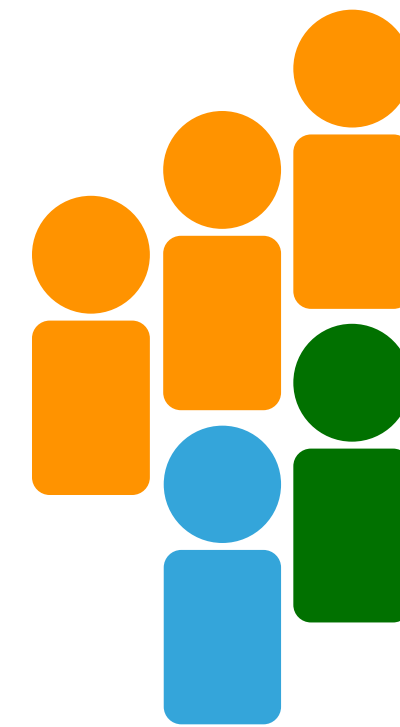
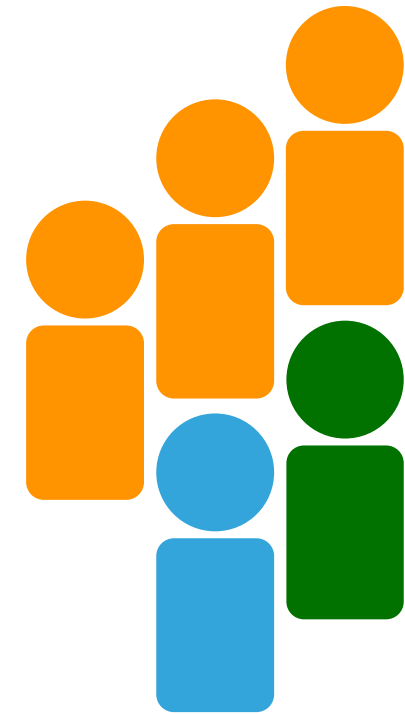
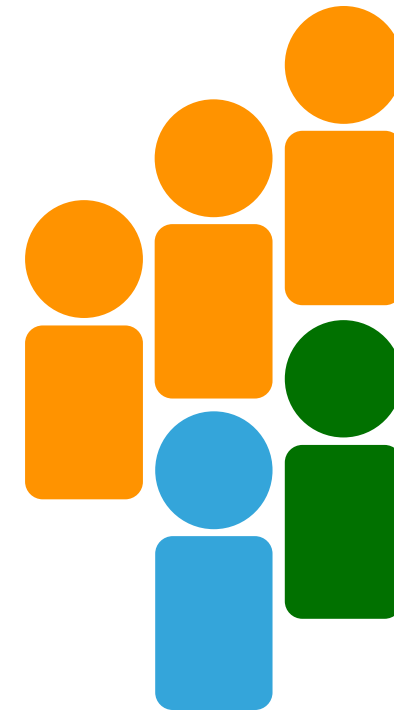
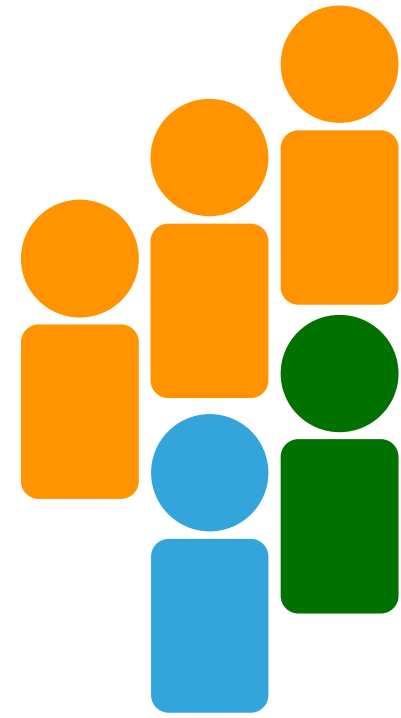
- ▶ Official service of the US Government
- ▶ 325 required security controls
 - ▶ 269 handled by cloud.gov
 - ▶ 41 shared
 - ▶ 15 handled by customer



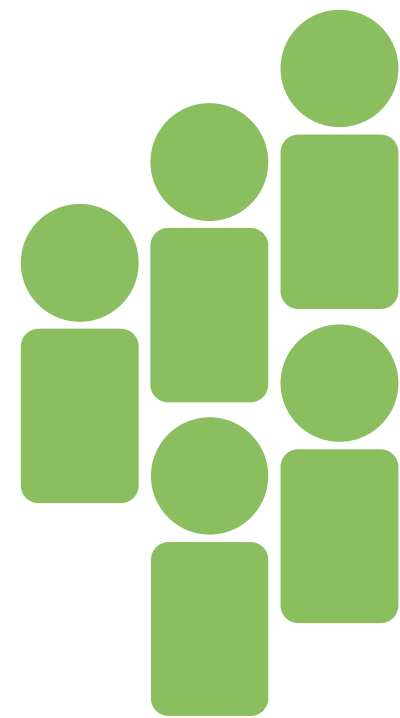
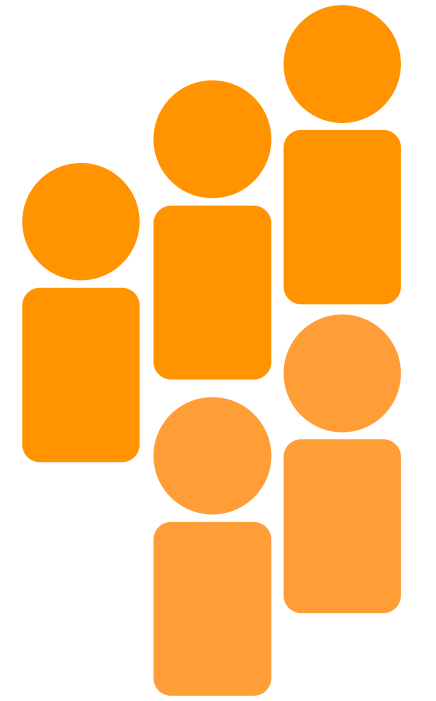
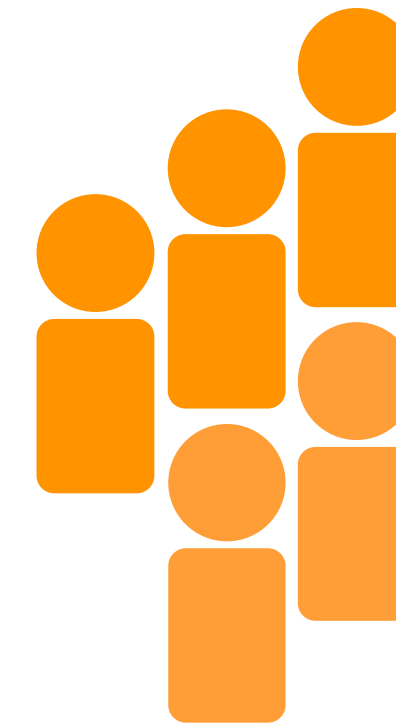
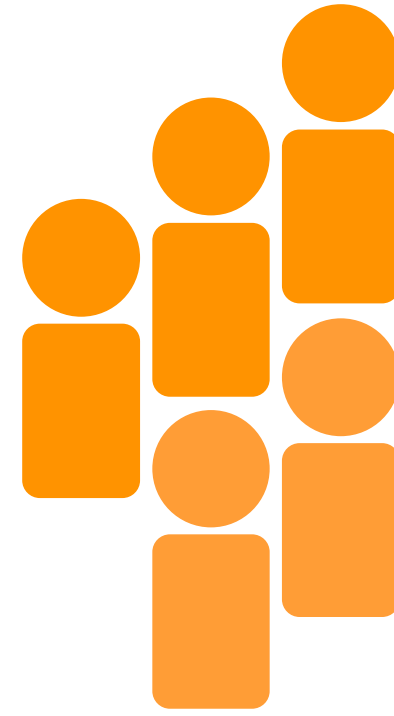
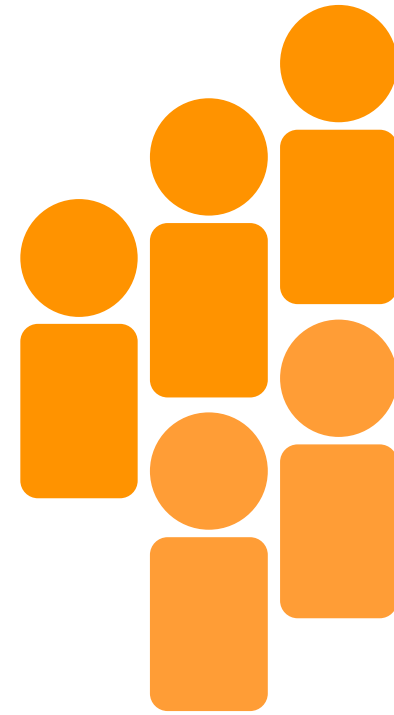
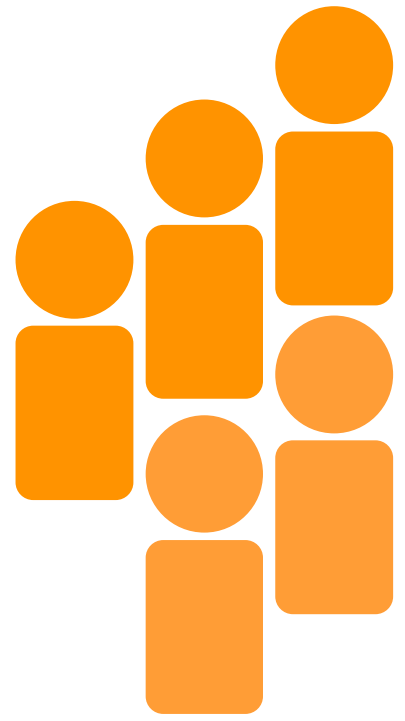
THE CLOUD IS JUST
SOMEBODY ELSE'S
COMPUTER

Someone at every conference

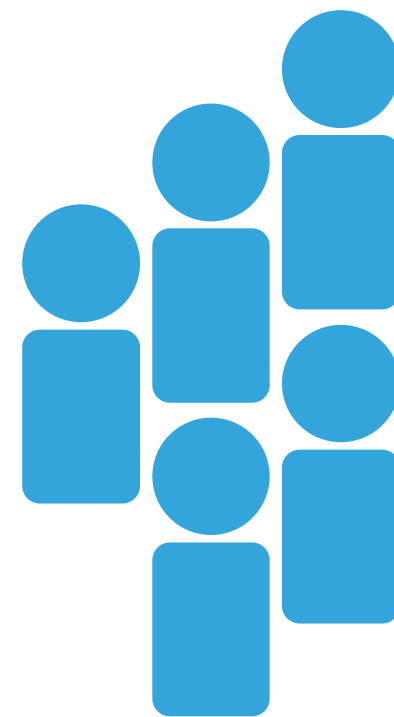
PRODUCT TEAMS



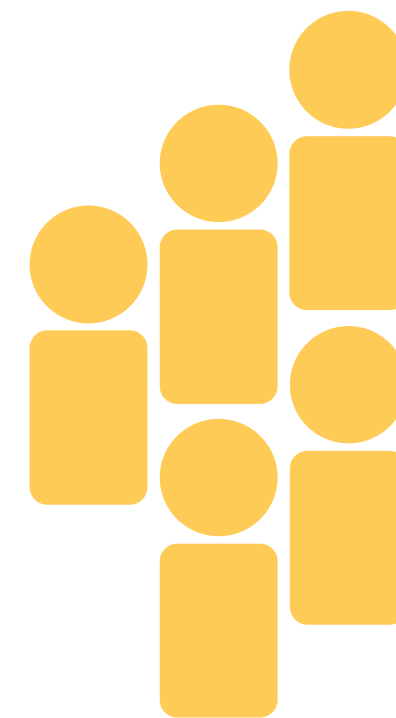
PRODUCT TEAMS



Platform Team

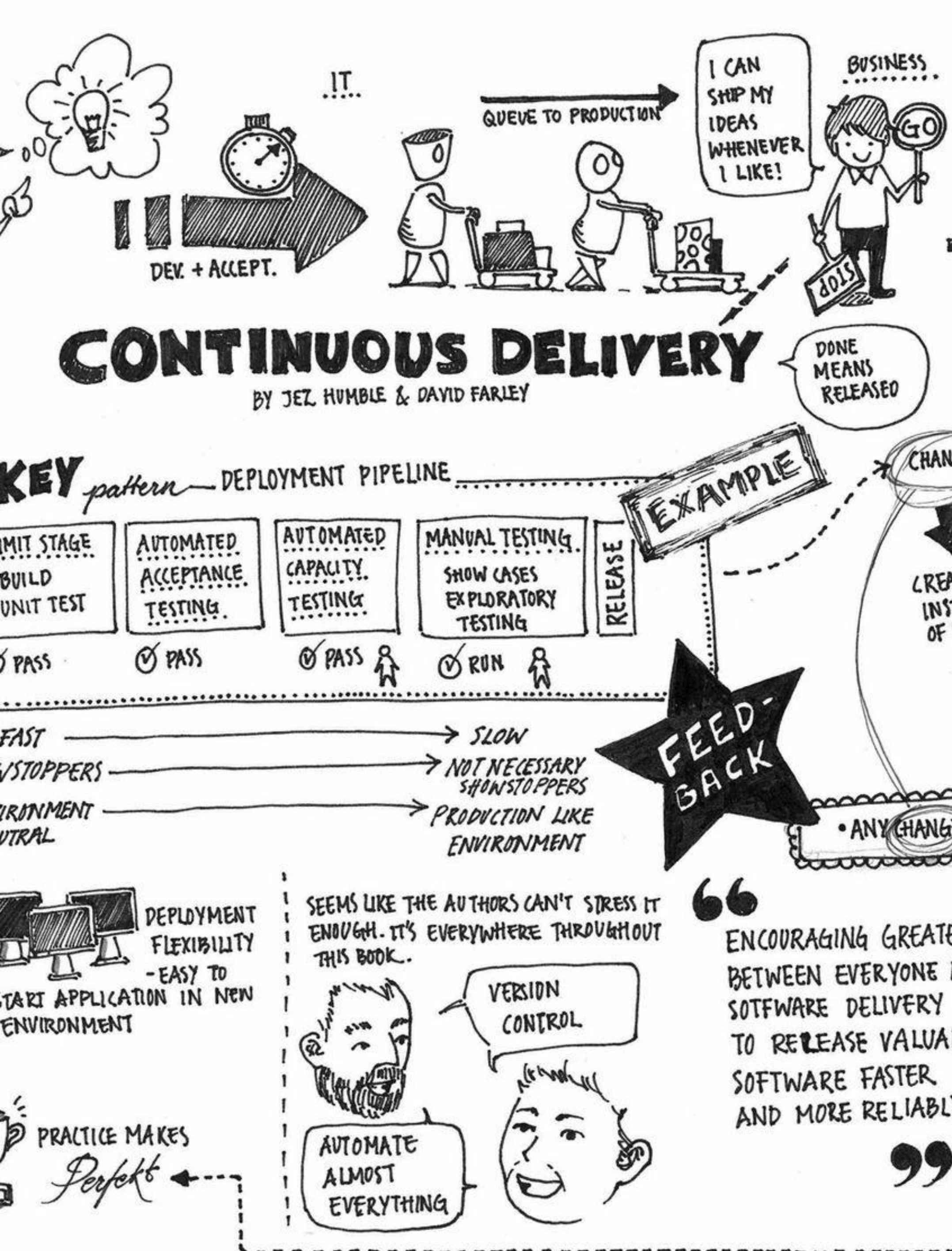


Compliance Team



Security Team

https://commons.wikimedia.org/wiki/File:Continuous_Delivery_by_Jez_Humble_and_David_Farley.jpg

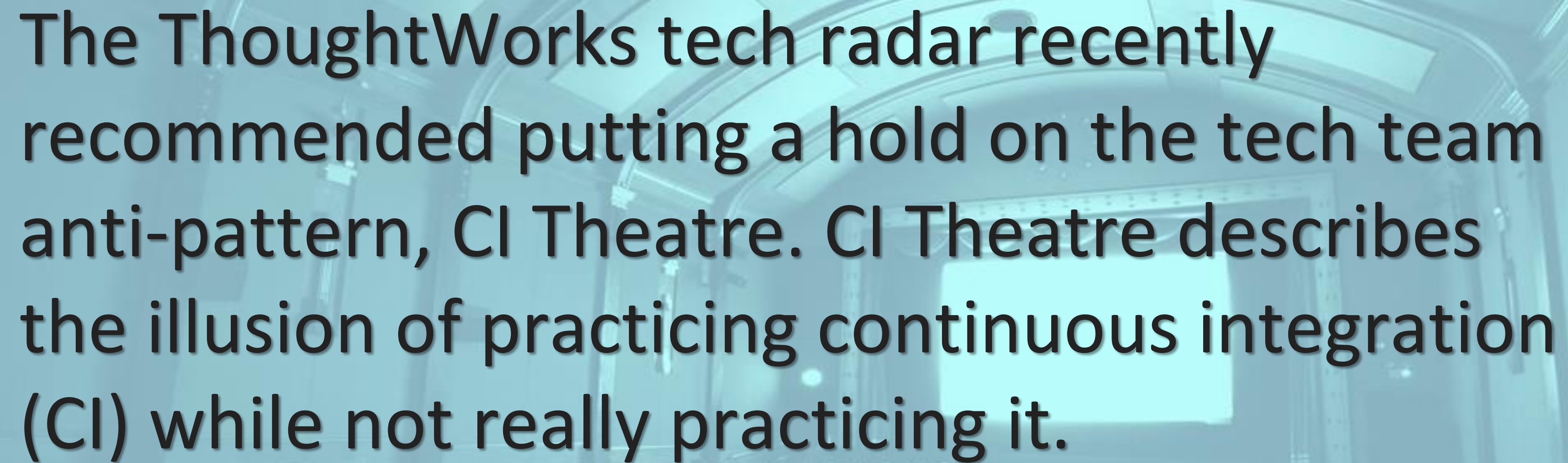


USE CONTINUOUS DELIVERY

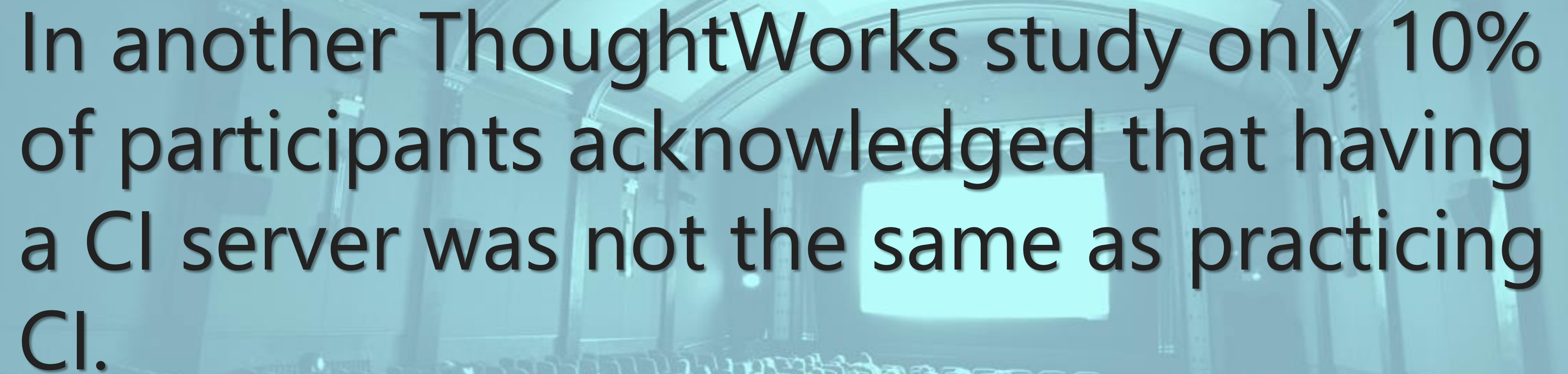
TO SAFELY
DEPLOY MORE
OFTEN

A PREREQUISITE TO CD

CONTINUOUS INTEGRATION



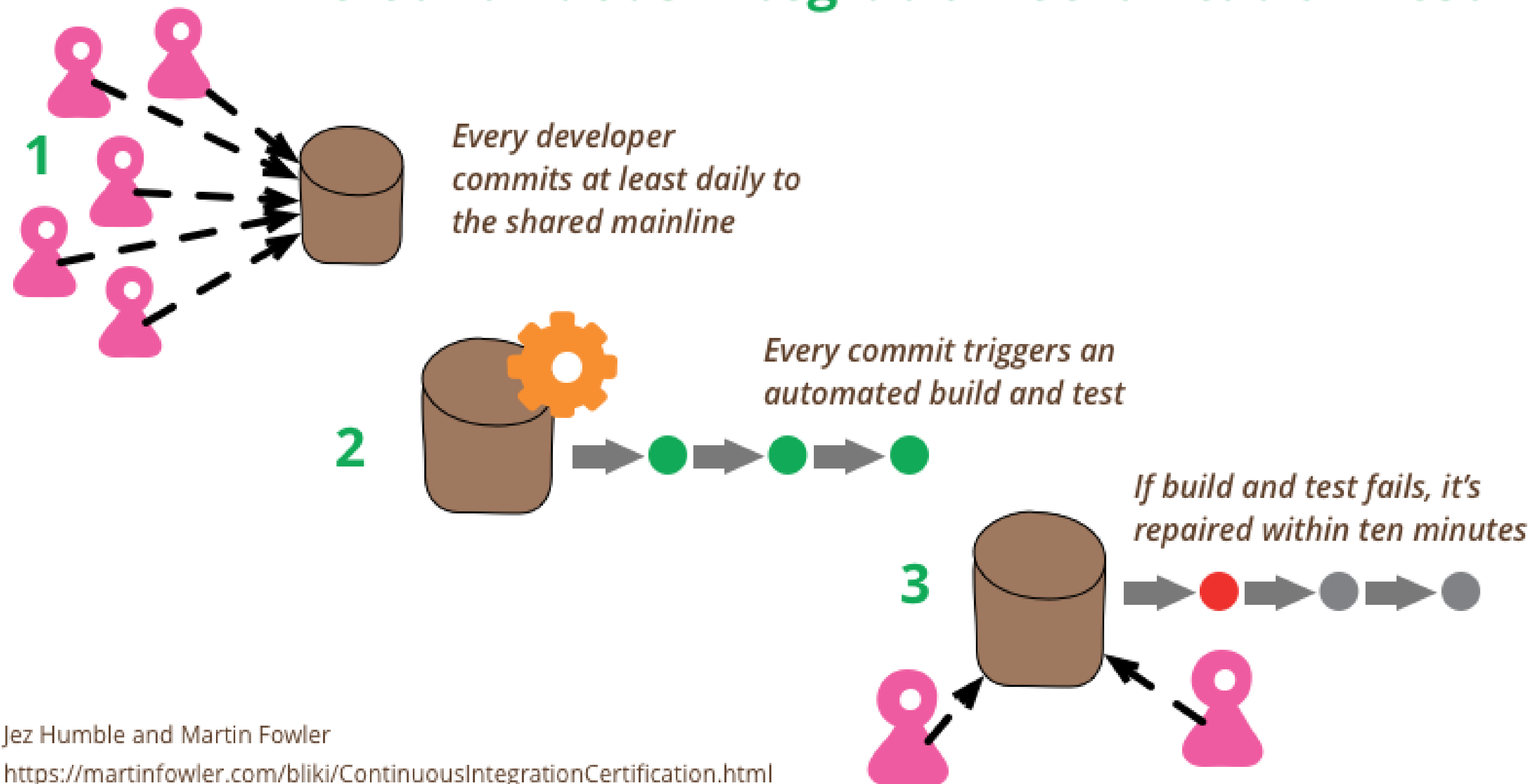
The ThoughtWorks tech radar recently recommended putting a hold on the tech team anti-pattern, CI Theatre. CI Theatre describes the illusion of practicing continuous integration (CI) while not really practicing it.



In another ThoughtWorks study only 10% of participants acknowledged that having a CI server was not the same as practicing CI.

ARE YOU PRACTICING CI?

The Continuous Integration Certification Test



Jez Humble and Martin Fowler

<https://martinfowler.com/bliki/ContinuousIntegrationCertification.html>



Ken Mugrage @ DevOpsDays India

@kmugrage



1/2 - "Our research shows that effective trunk-based development is characterized by fewer than three active branches and branches and forks having lifetimes of less than a day before being merged to master." -
Accelerate State of DevOps



Ken Mugrage @ DevOpsDays India

@kmugrage



2/2 When most "trunk based development" zealots talk about it (including me), this is what we mean. Especially in a Git world, we aren't saying you never branch at all, just that it goes to master every day.

CONTINUOUS DELIVERY

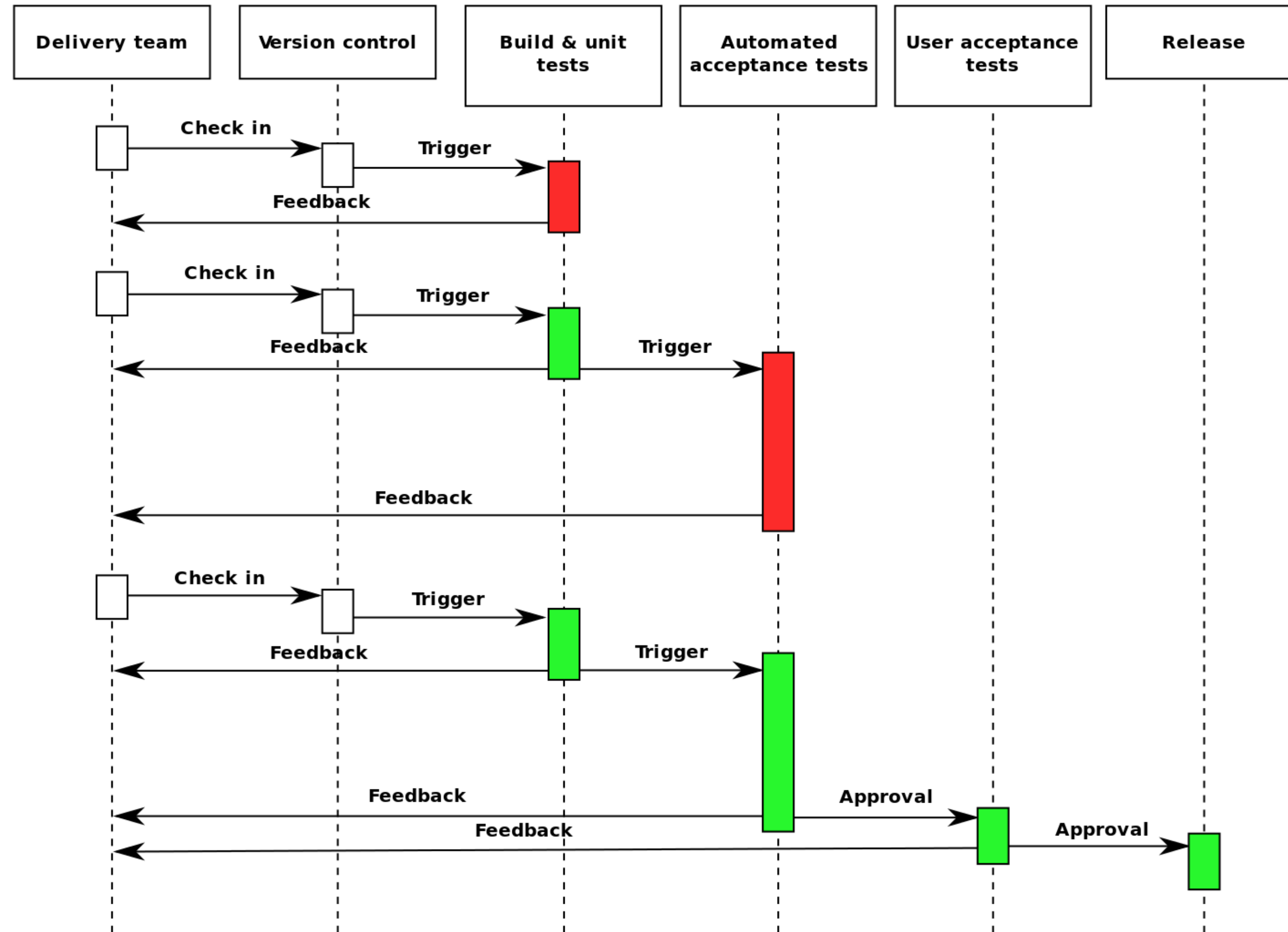
CONTINUOUS DELIVERY IS THE ABILITY TO GET CHANGES OF ALL TYPES—INCLUDING NEW FEATURES, CONFIGURATION CHANGES, BUG FIXES AND EXPERIMENTS—INTO PRODUCTION, OR INTO THE HANDS OF USERS, SAFELY AND QUICKLY IN A SUSTAINABLE WAY.

Jez Humble

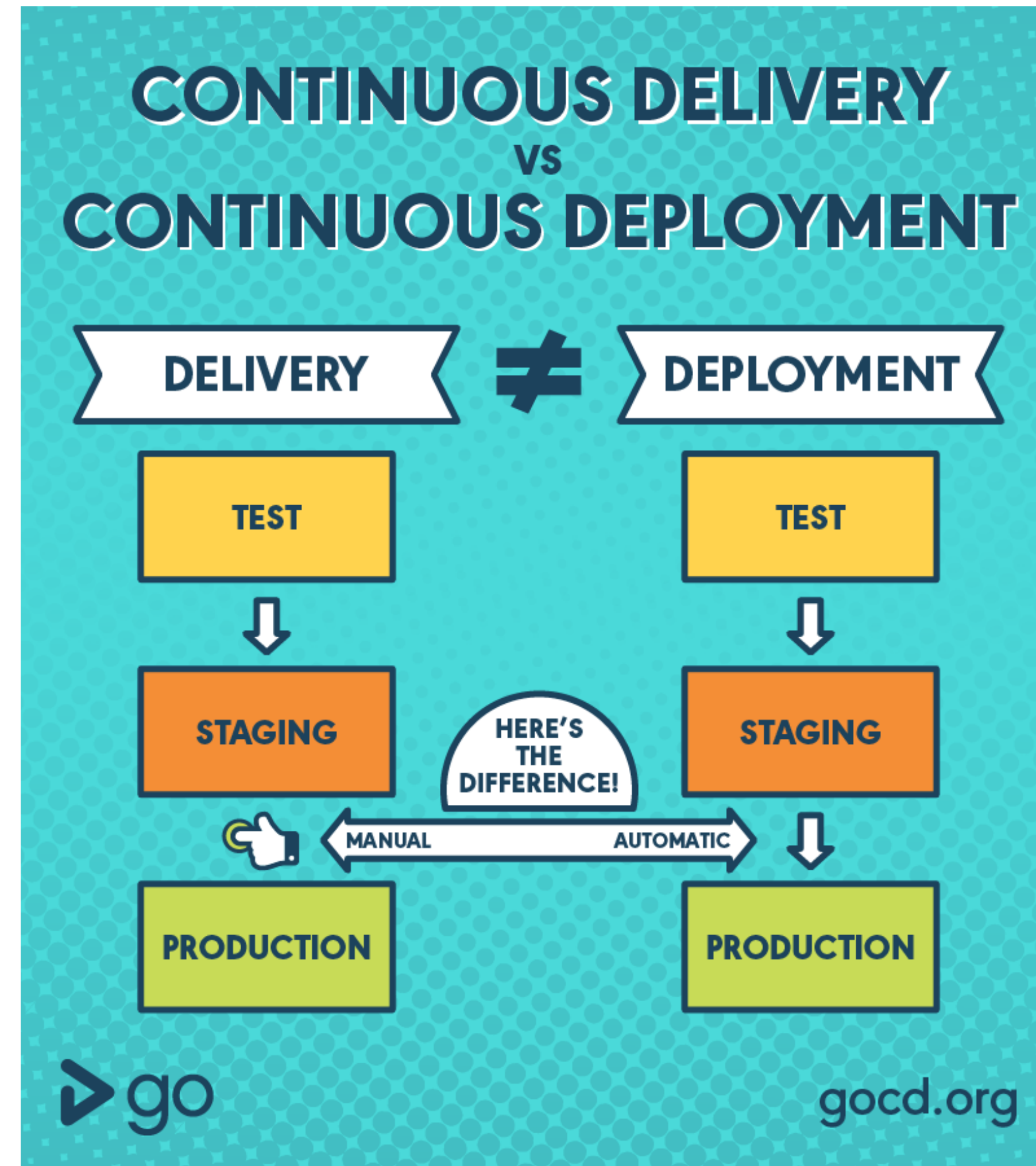
CONTINUOUS DELIVERY IS
WHAT YOU “DO” AS PART OF
A DEVOPS CULTURE

Me

CONTINUOUS DELIVERY PIPELINE

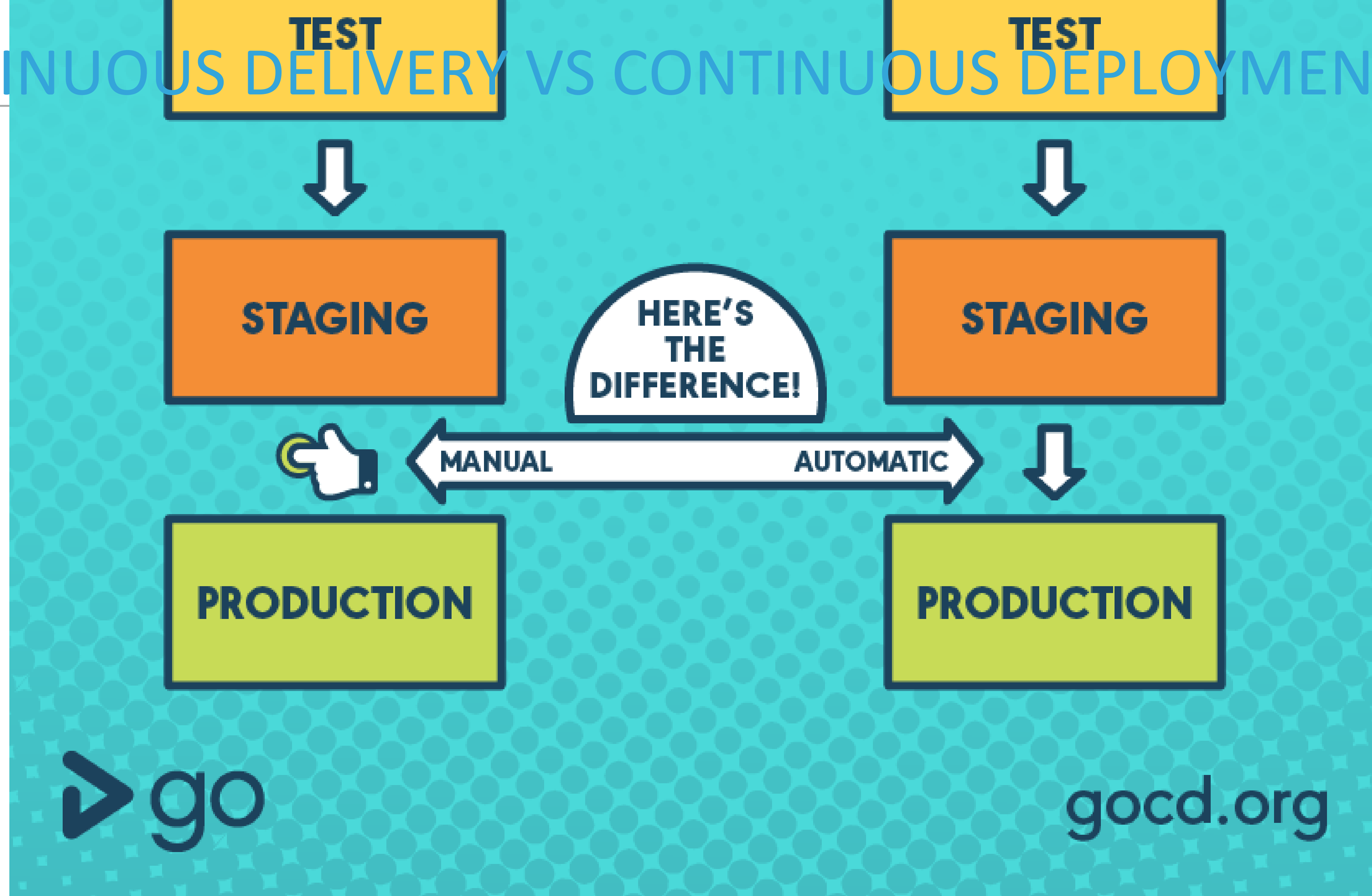


CONTINUOUS DELIVERY VS CONTINUOUS DEPLOYMENT

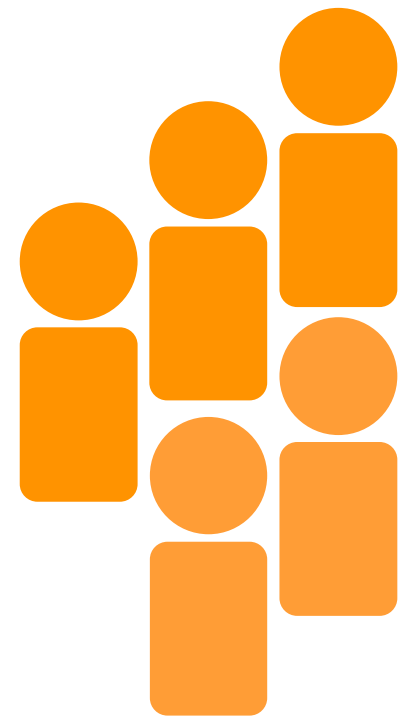


CONTINUOUS DELIVERY VS CONTINUOUS DEPLOYMENT

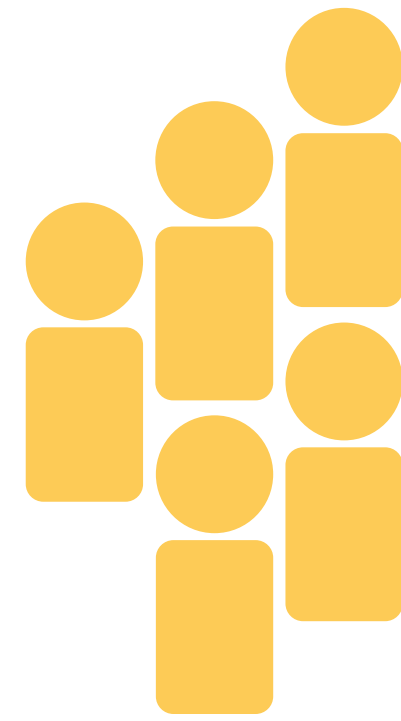
➤ go



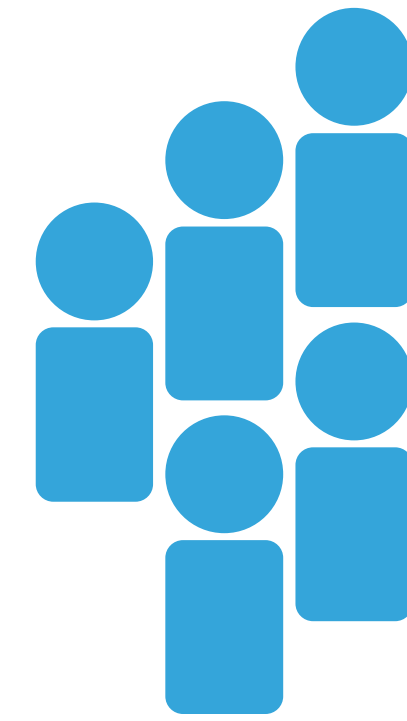
OUR TEAMS



Product Team

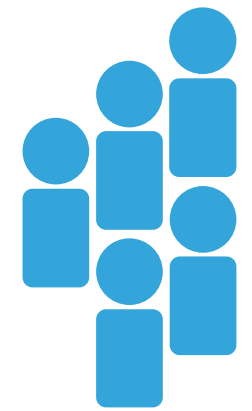
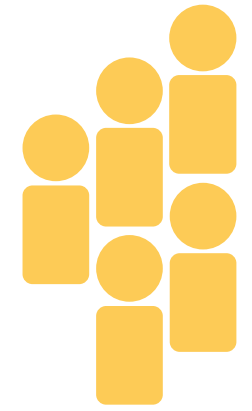


Security Team



Compliance Team

OUR CONTINUOUS DELIVERY PIPELINE



UNIT
TESTS



FUNCTIONAL
TESTS



DEPLOY
STAGING



DEPLOY
PRODUCTION

CONTINUOUS DELIVERY IS NOT
JUST CONTINUOUS INTEGRATION
WITH A PLUGIN AND A COUPLE
EXTRA STEPS

Me

EXAMPLES OF THINGS WHICH ARE BAD

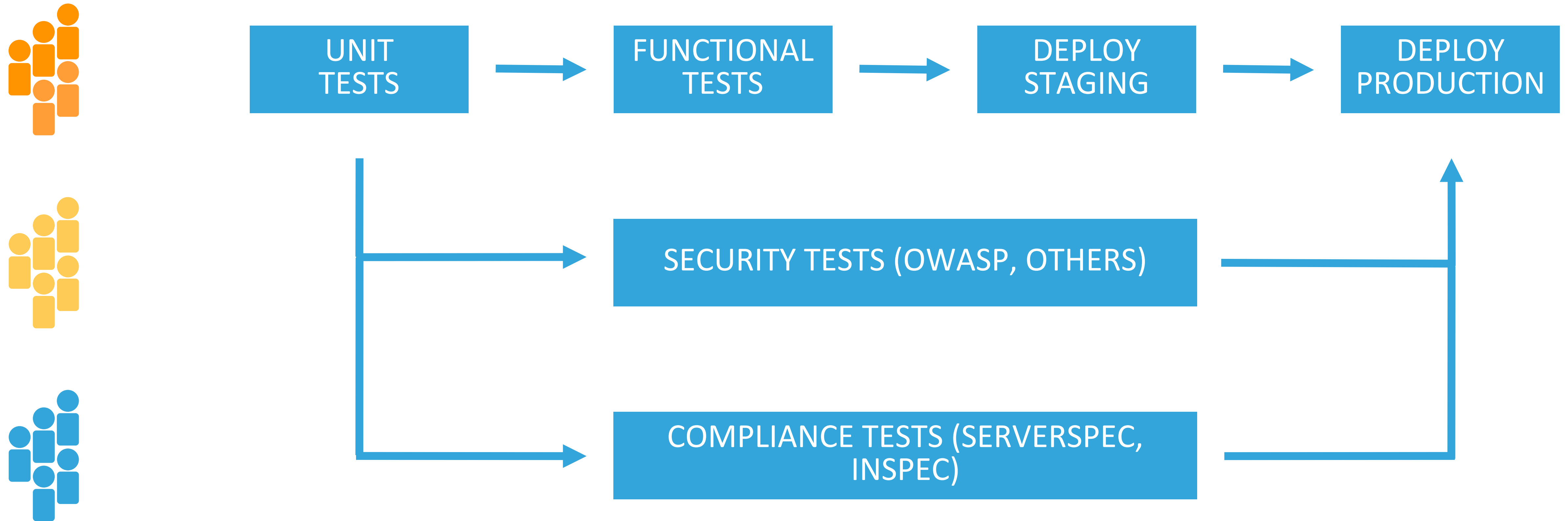


- ▶ Deploying insecure software
- ▶ Deploying non-performant software
- ▶ Deploying non-complying software
- ▶ Deploying ineffective software

THE PURPOSE OF A CONTINUOUS
DELIVERY PIPELINE IS TO KILL A
RELEASE CANDIDATE

Me, and a lot of other people

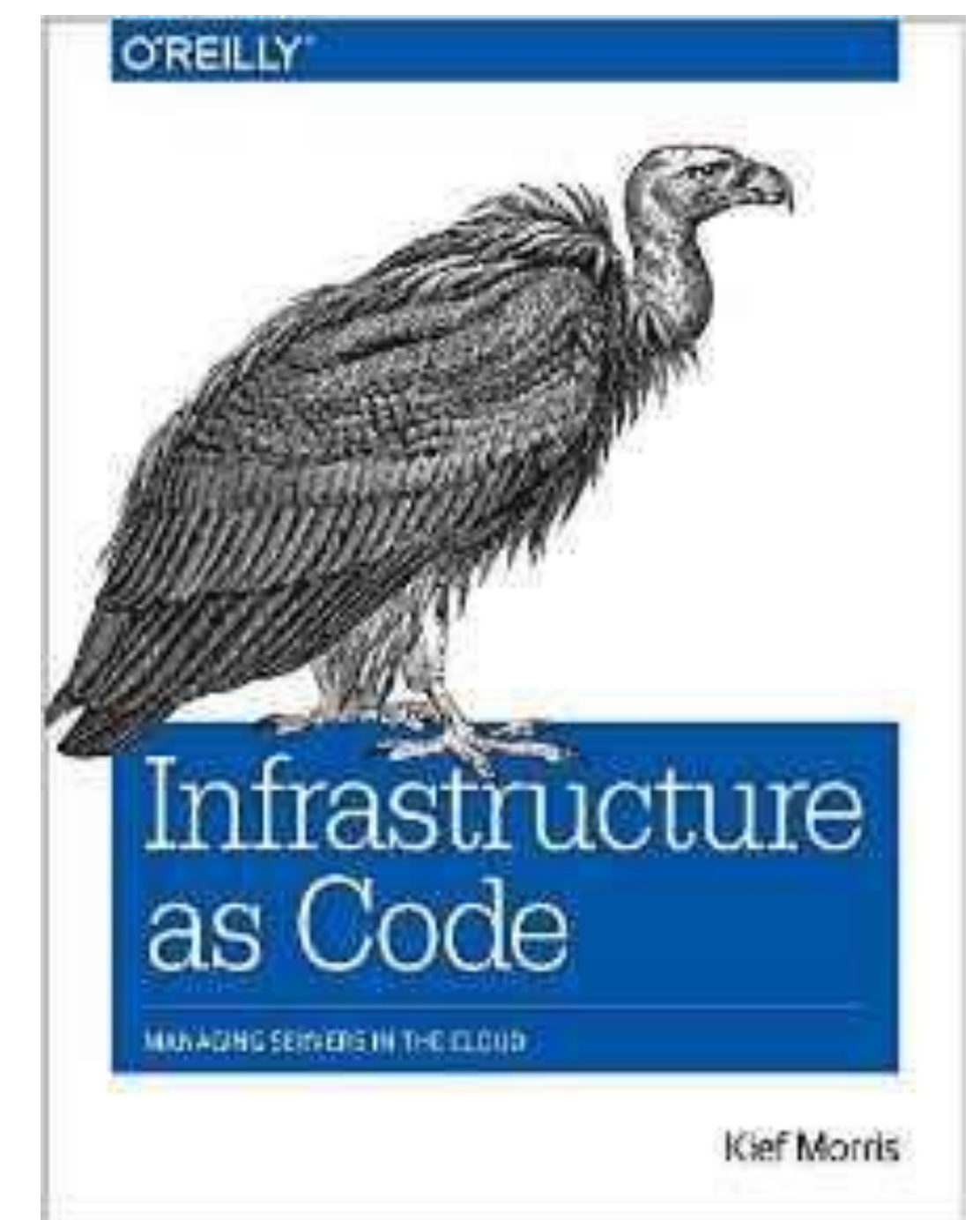
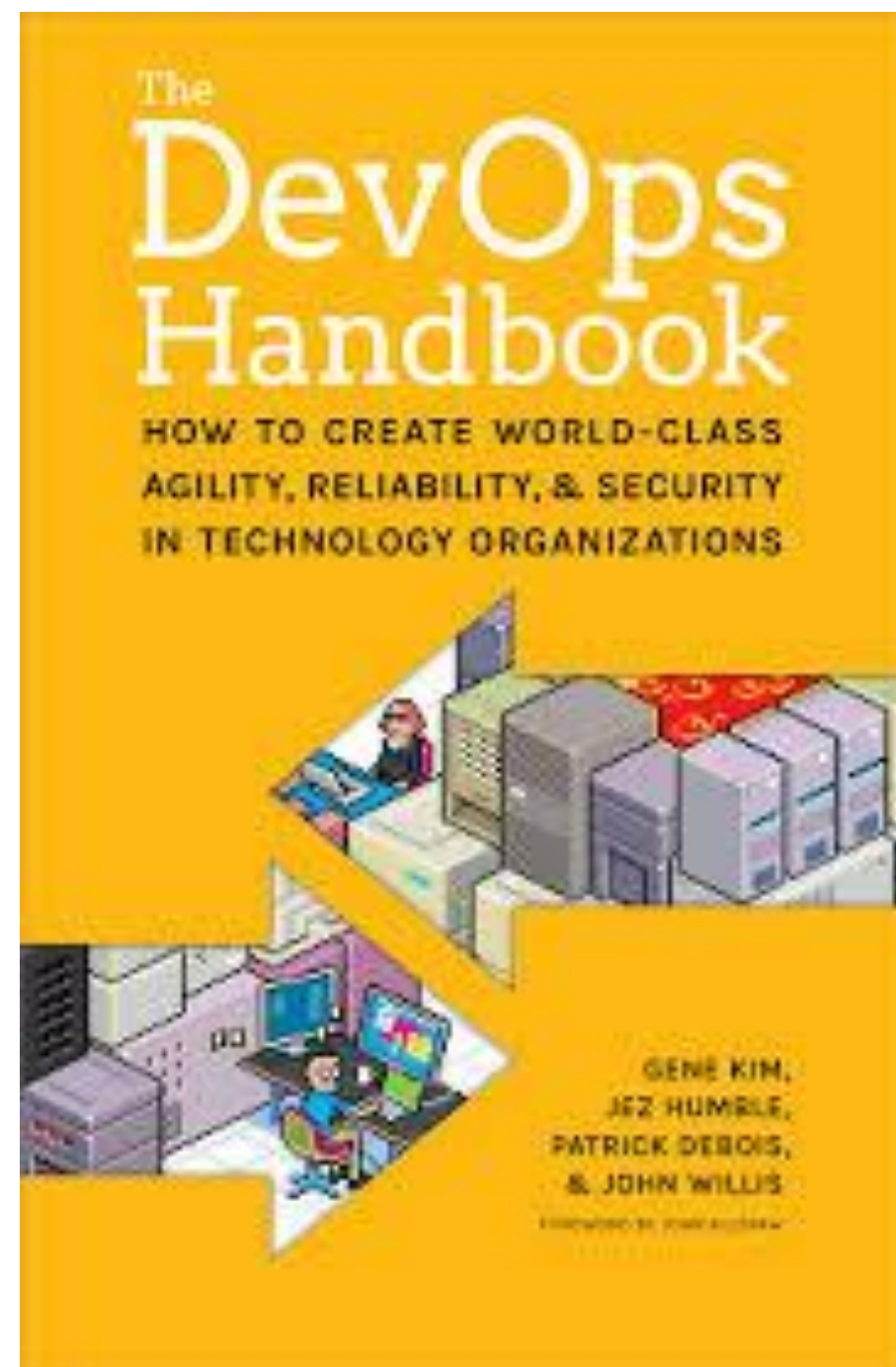
OUR CONTINUOUS DELIVERY PIPELINE



SORRY, THERE'S NO
SILVER BULLET

SORRY, THERE'S NO SILVER BULLET

BUT THESE ARE SOLVED PROBLEMS
AND THERE IS A LOT OF HELP



- ▶ Redefine words for your organization
- ▶ Change your organization to enable DevOps
- ▶ Use modern architectures and technologies
- ▶ Use Continuous Delivery to safely deploy more often

THANK YOU!

@kmugrage

@goforcd

<https://www.gocd.org/>