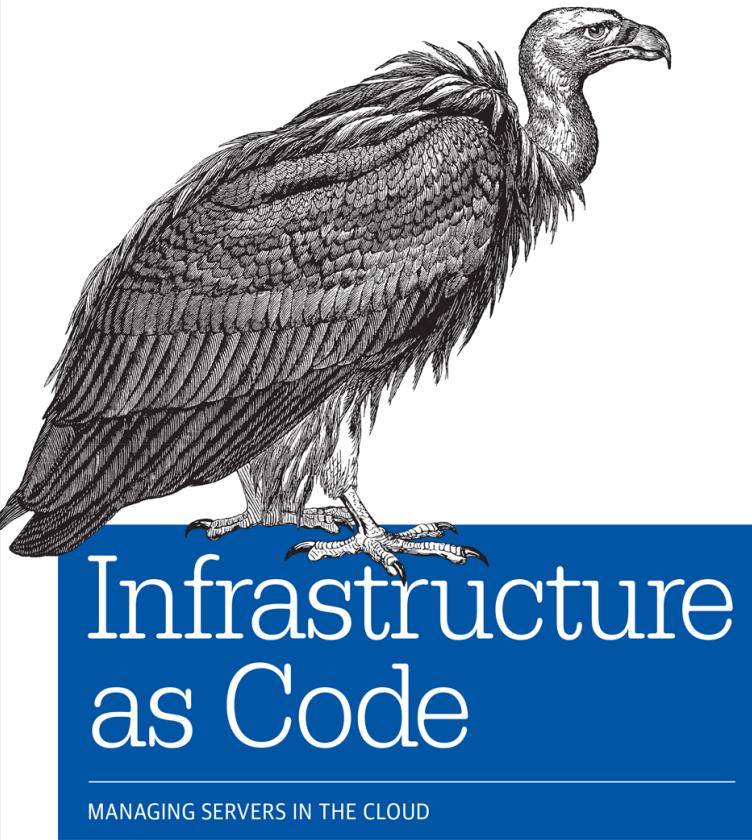




ThoughtWorks®
@cfgmgmtcamp

INFRASTRUCTURE AS CODE



Kief Morris

kief@thoughtworks.com

Cloud Practice Lead (UK)

DevOps, Continuous Delivery, Agile Ops

ThoughtWorks®

Twitter: @kief

Book: <http://oreil.ly/1JKIBVe>

Site: <http://infrastructure-as-code.com>

February 2017



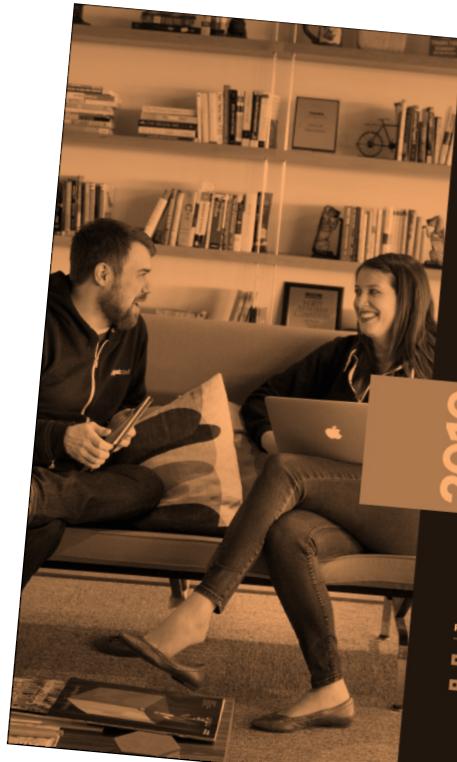
SPEED

RISK



FASTER IS SAFER

<http://bit.ly/2cQQSOk>



2016 State of
DevOps Report

Presented by:

puppet + DORA
DEVS OPS RESEARCH & ASSESSMENT

Sponsored by:

Hewlett Packard Enterprise ThoughtWorks splunk >
Atlassian Automic REVOLUTION

RAPID



RESPONSIBLE

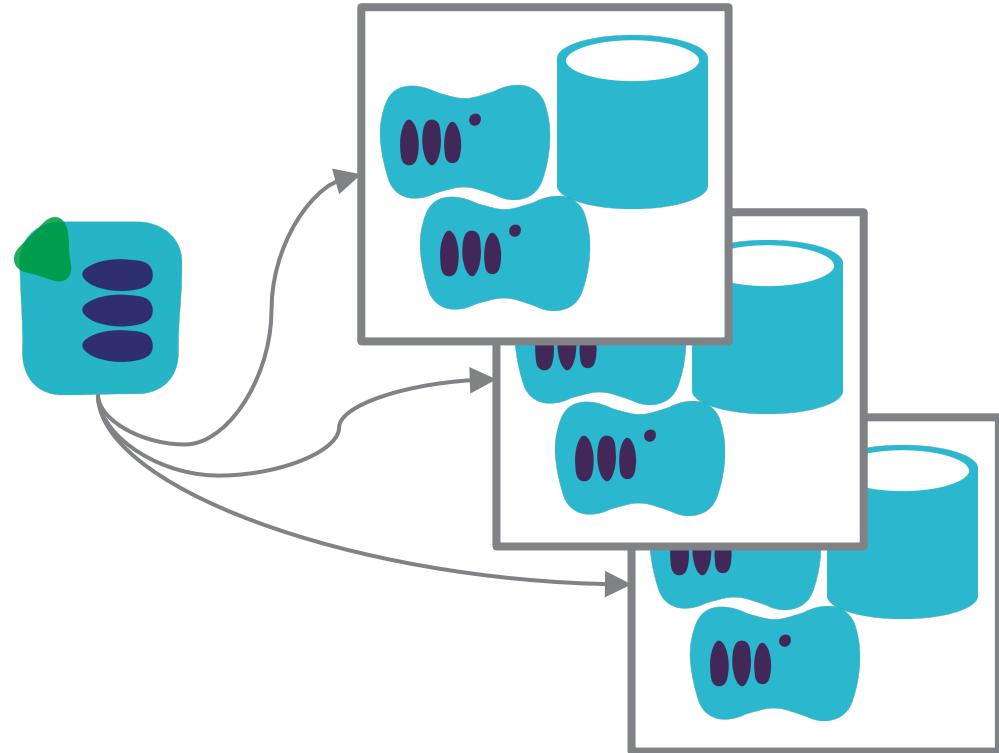
INFRASTRUCTURE AS CODE

Applying tools and practices from
software engineering to managing
infrastructure.

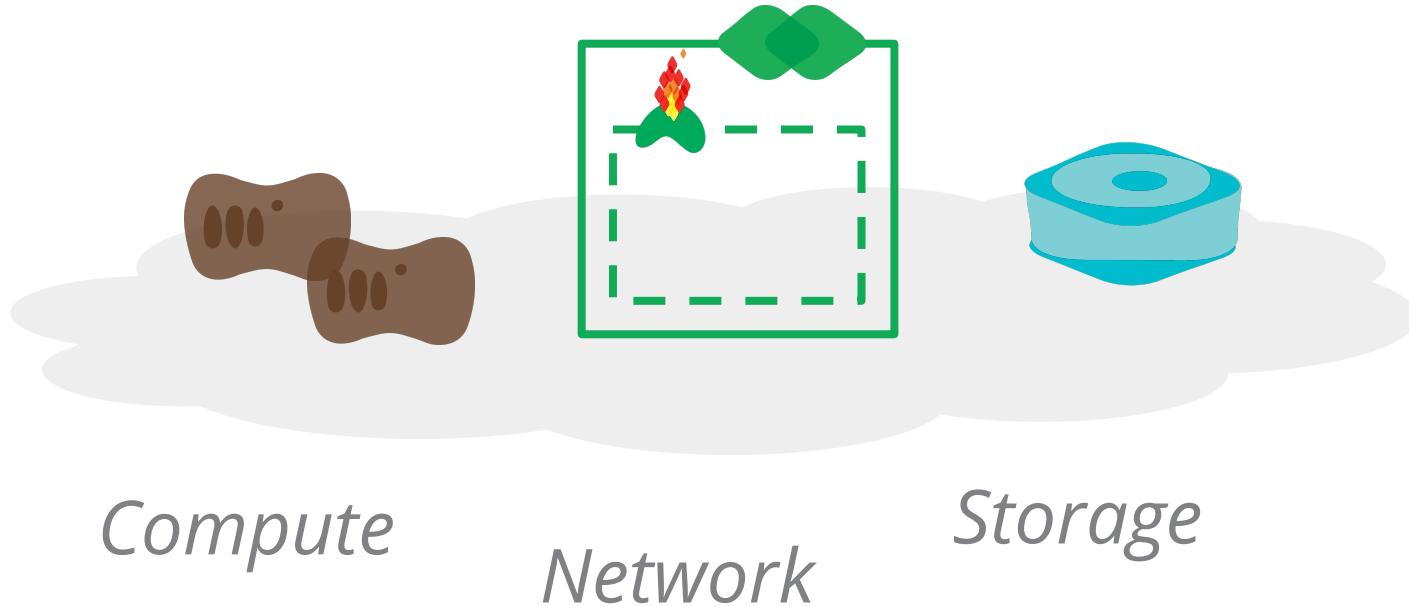
DEFINE SYSTEMS AS CODE

System design is:

- Reusable
- Consistent
- Visible
- Versioned



DYNAMIC INFRASTRUCTURE PLATFORMS



DYNAMIC INFRASTRUCTURE PLATFORMS

Cloud (IaaS)

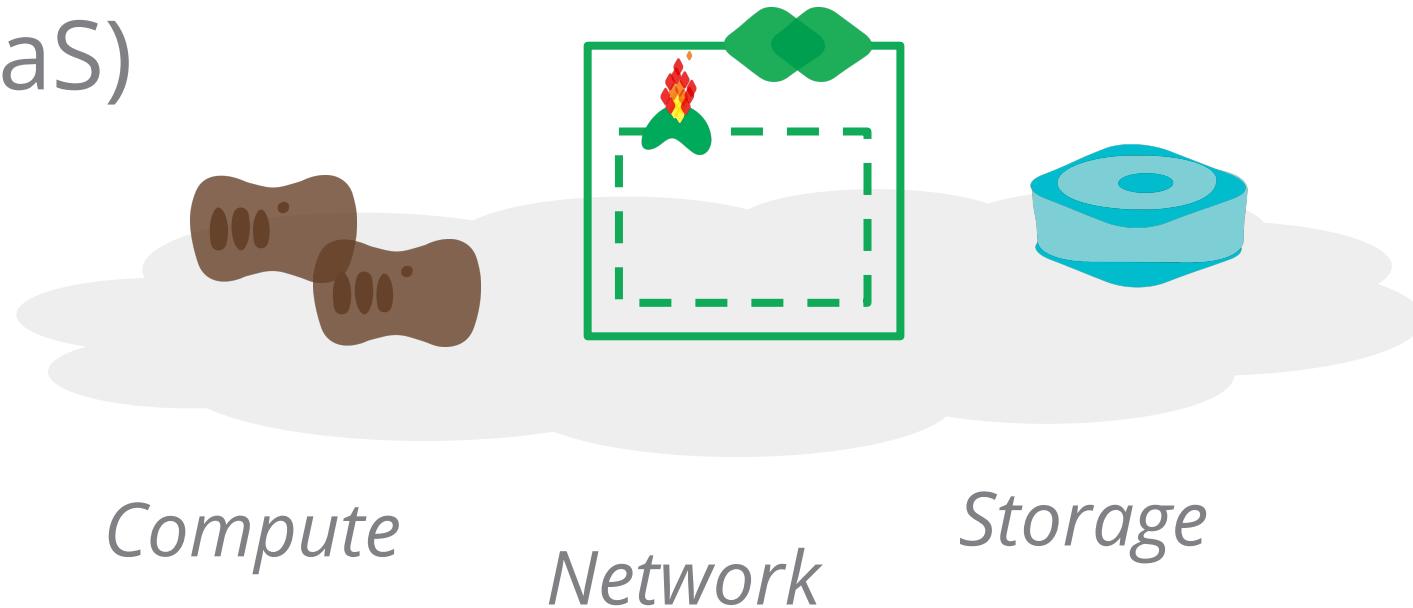
Virtual

Physical

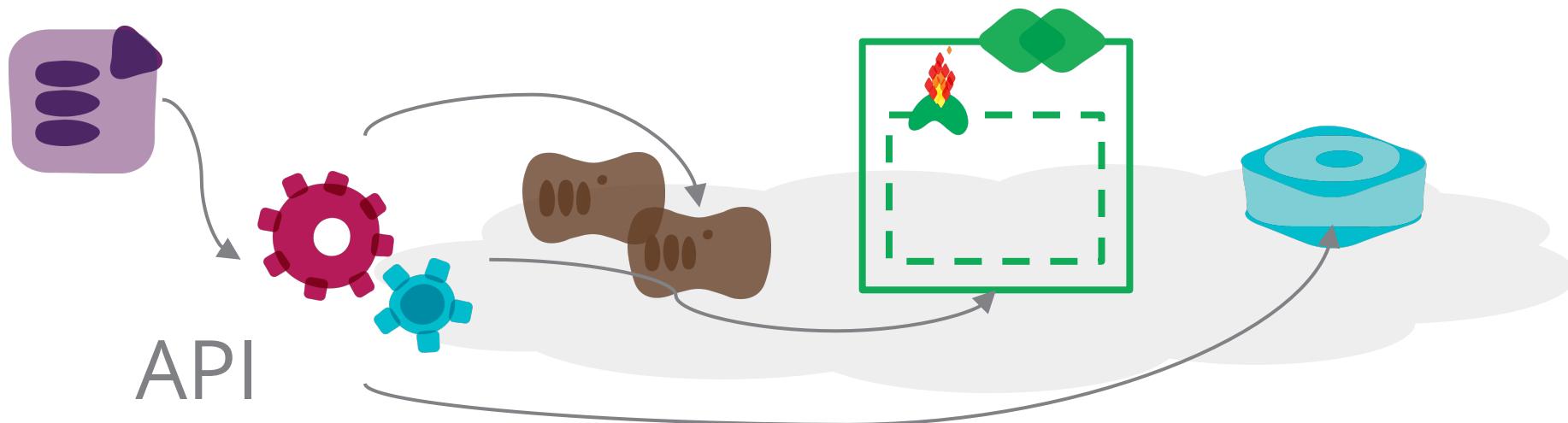
Compute

Network

Storage



PROGRAMMABLE, ON-DEMAND



Cloud and automation

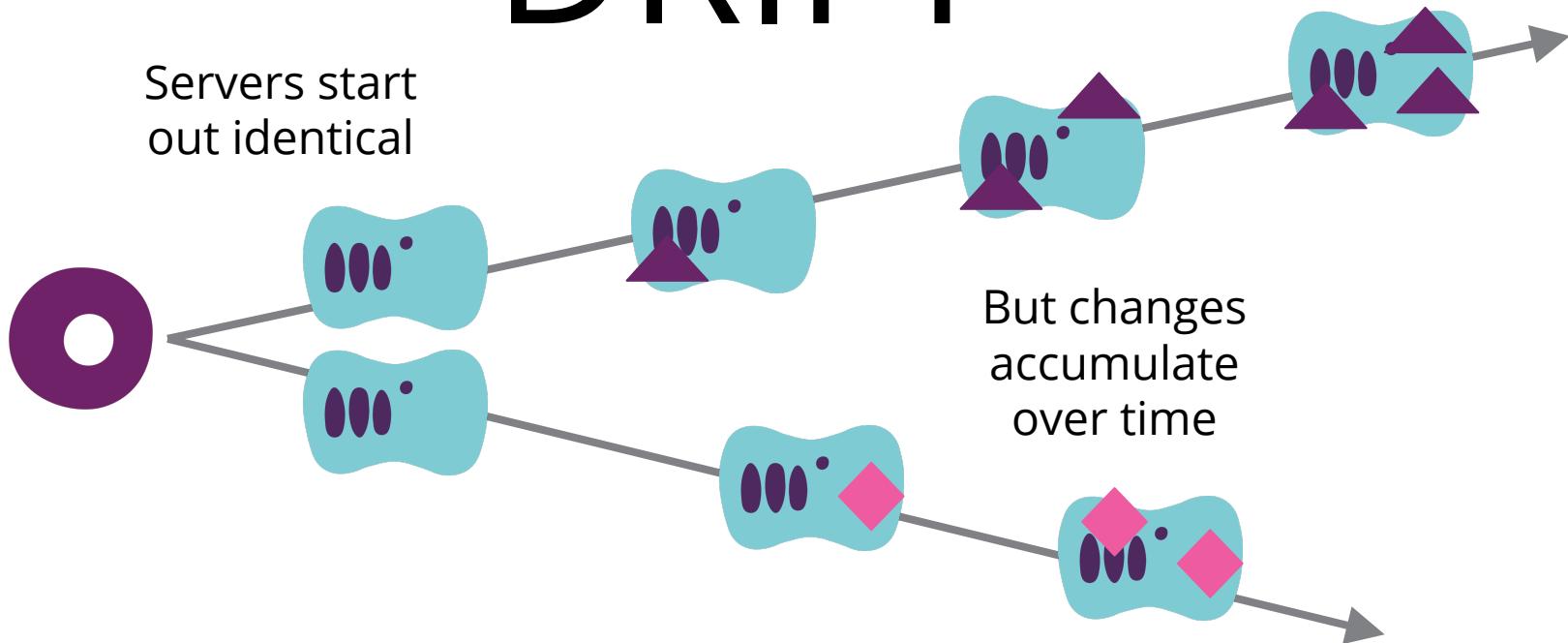


AWESOME!

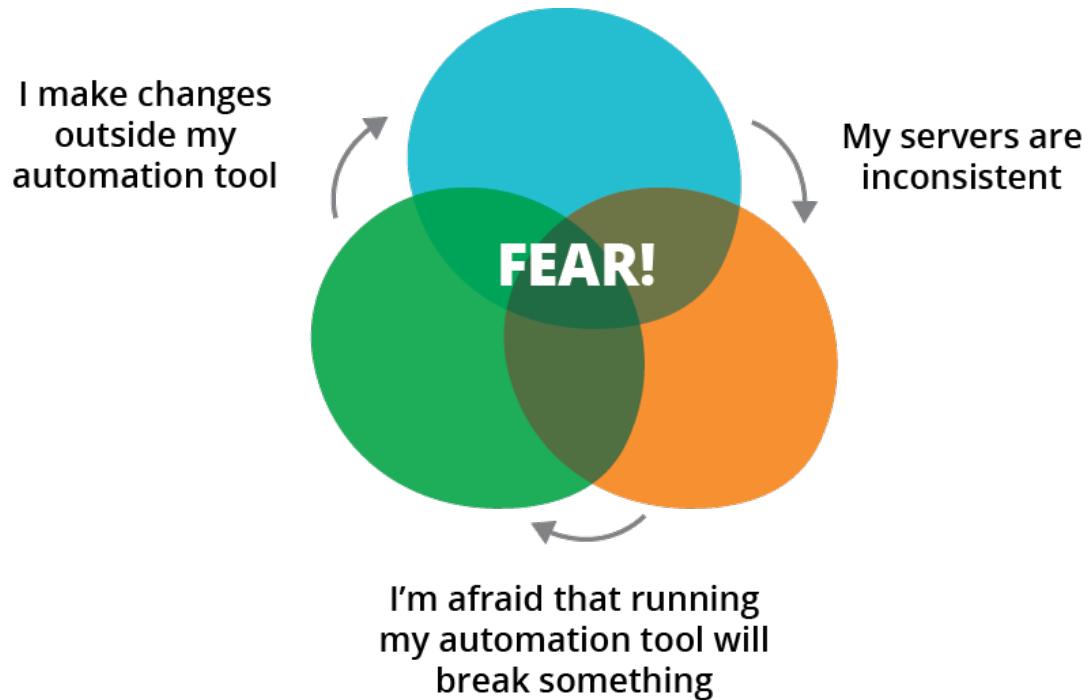
Oh, no!

SERVER SPRAWL!

CONFIGURATION DRIFT



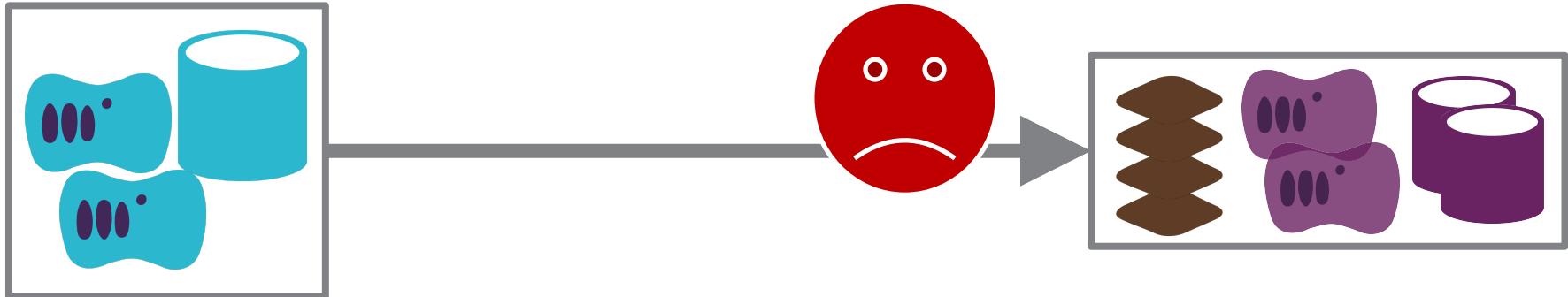
AUTOMATION FEAR CYCLE



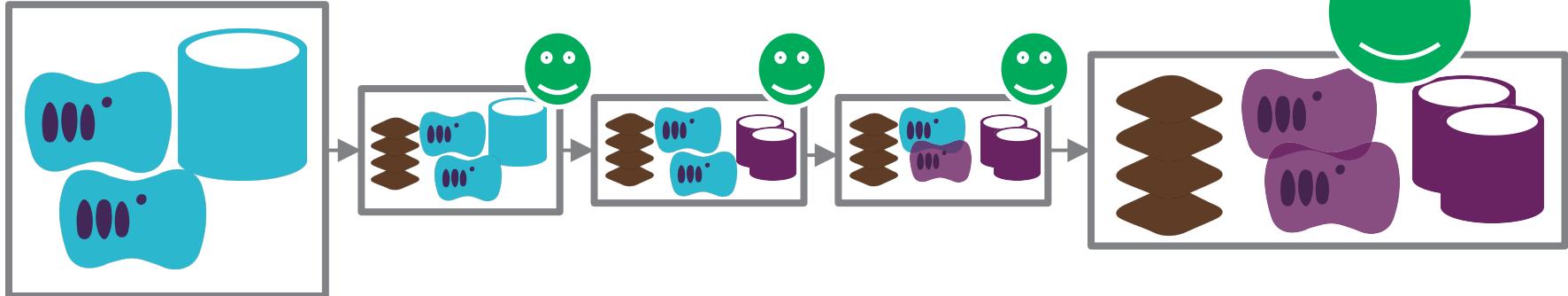
AUTOMATION LAG

The longer it's been since an automated process has run in the same context ...

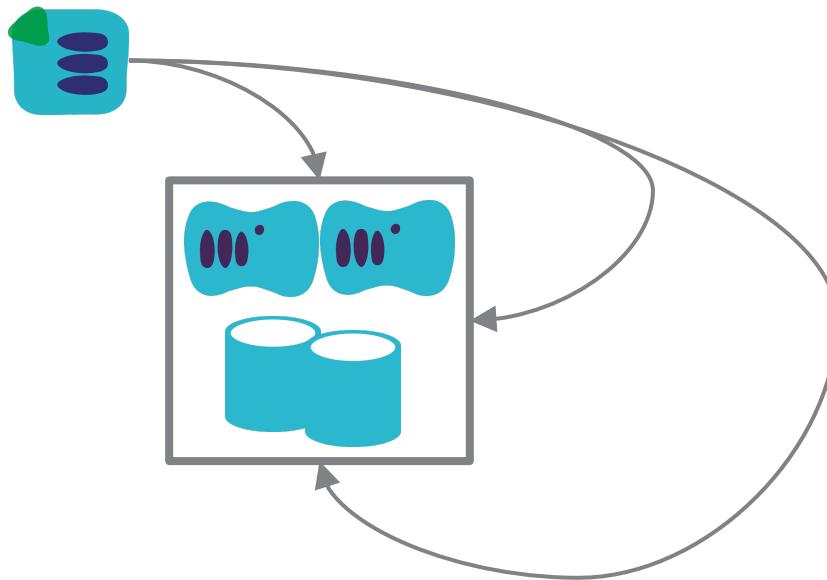
... the more work is needed to run it again



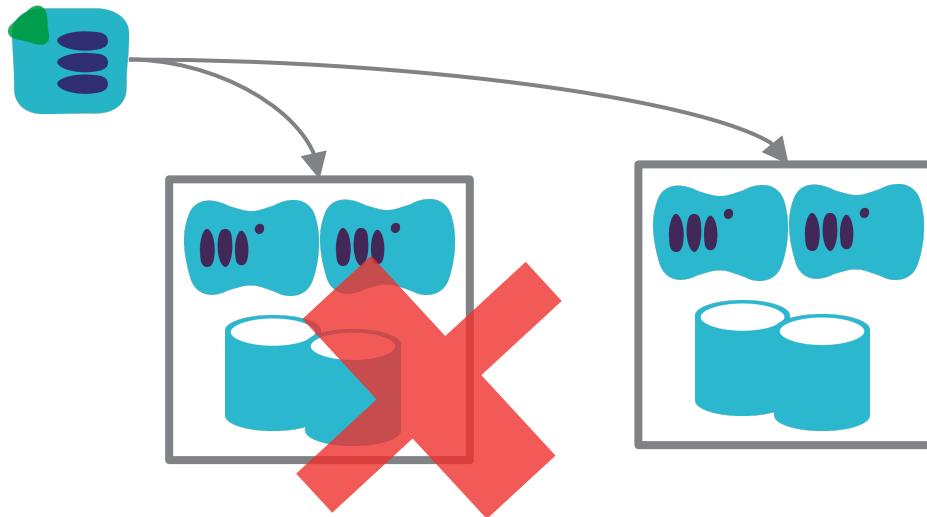
Apply small changes frequently
rather than large batches infrequently



CONTINUOUSLY SYNCHRONIZE



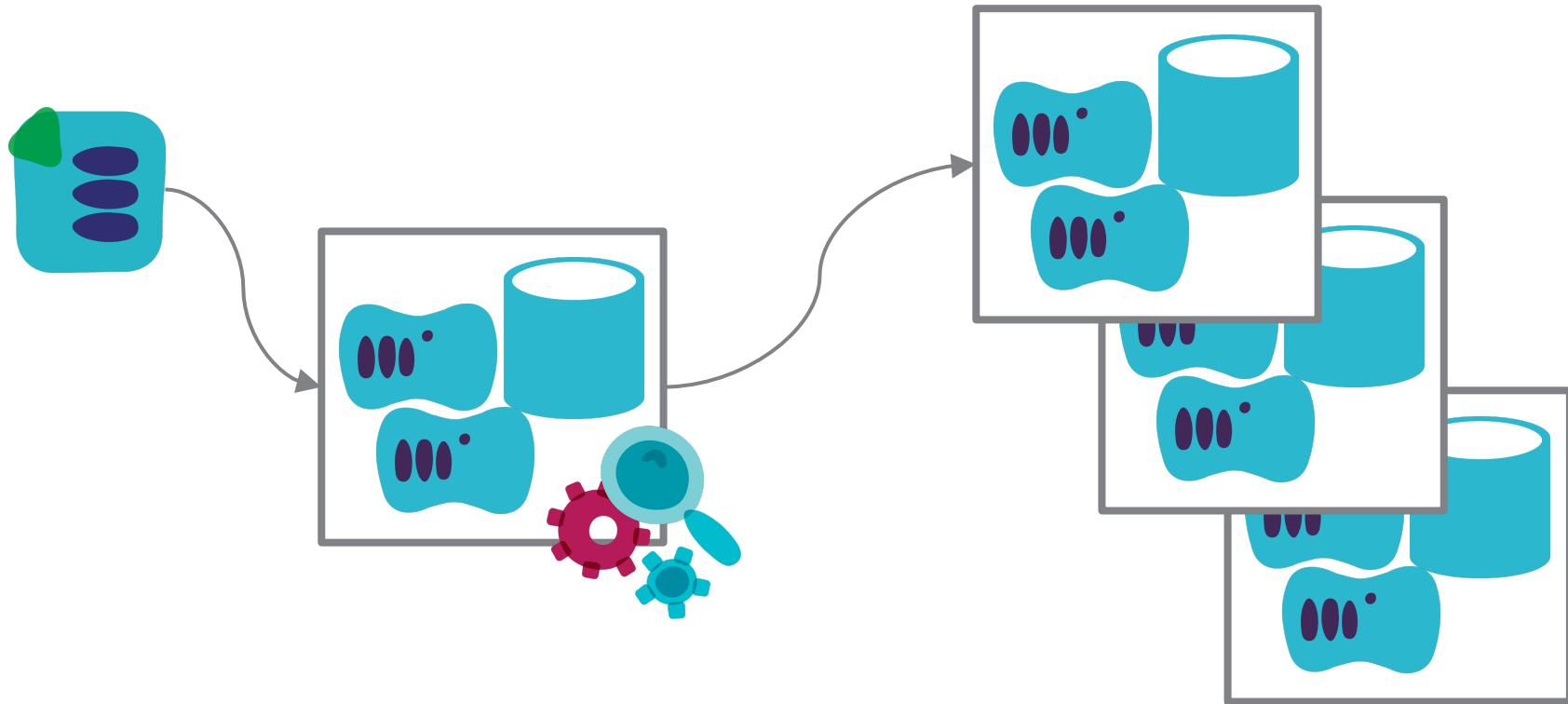
OR CONTINUOUSLY REBUILD



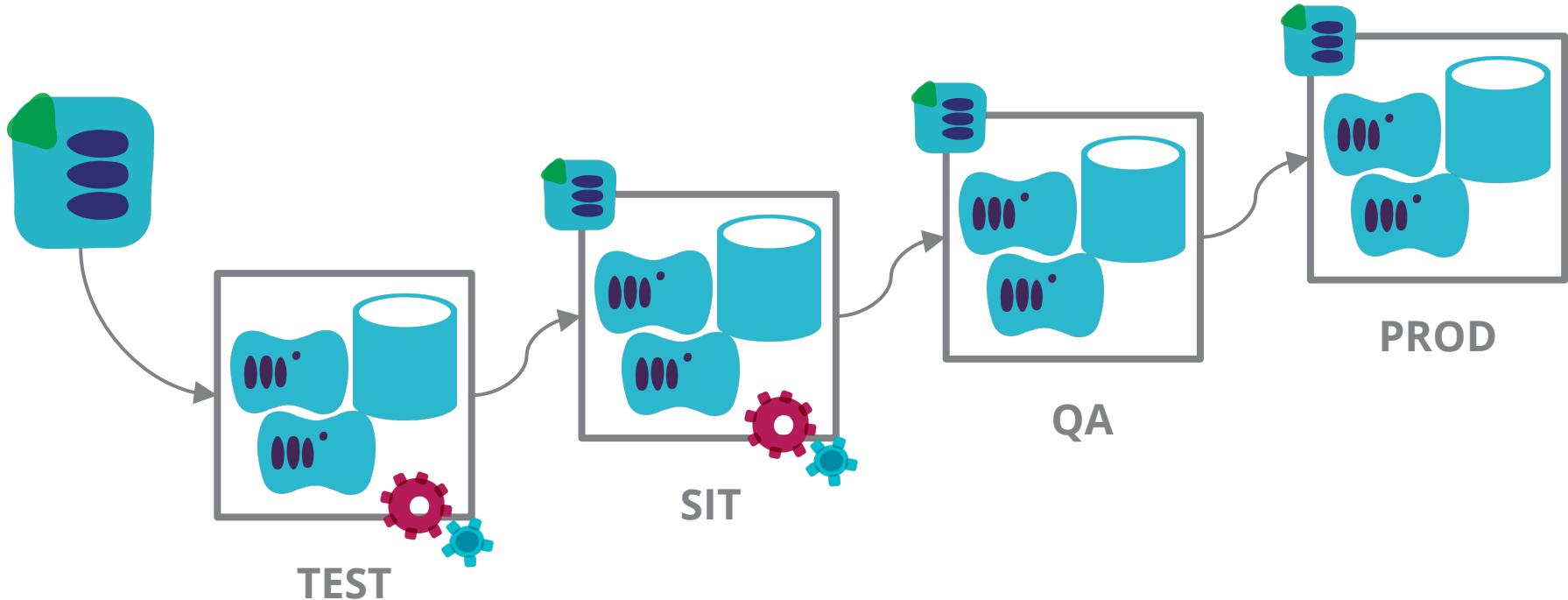
DEVOPS

How can we avoid damage from
automated mistakes?

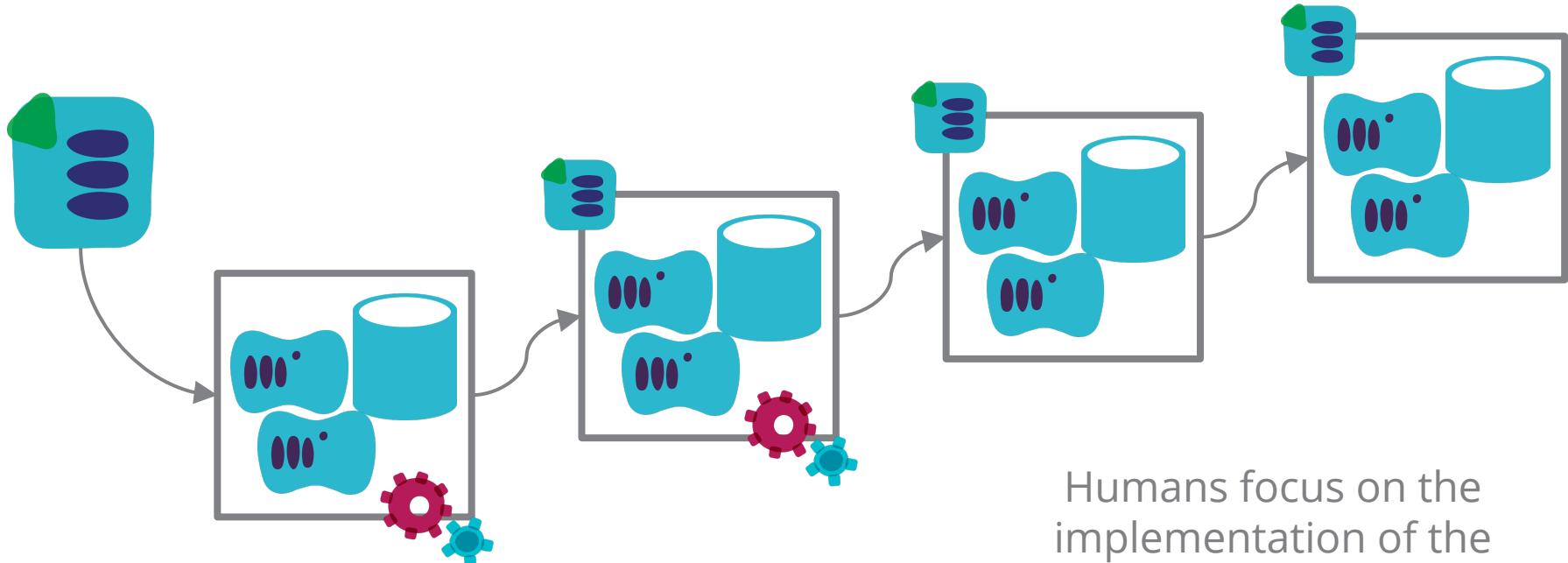
AUTOMATICALLY TEST EVERY CHANGE



PROMOTE CHANGES



BUILD COMPLIANCE INTO THE PIPELINE



Use the pipeline to **continuously validate** operational requirements and compliance, and to implement **controls**

Humans focus on the implementation of the **pipeline** and **audit trails**

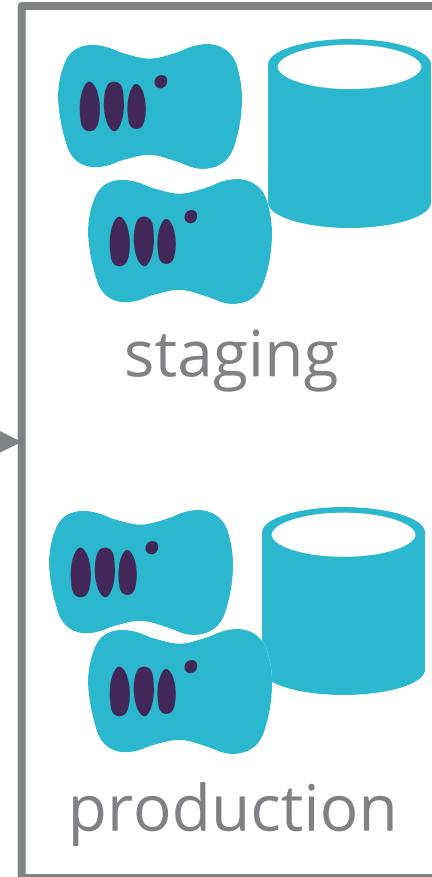
INFRASTRUCTURE DESIGN PATTERNS

Organizing code for dynamic
infrastructure

`our-project/main.tf`



One definition,
*multiple
environments*

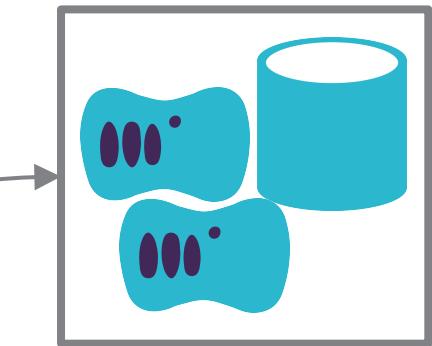


INFRASTRUCTURE DESIGN PRINCIPLE

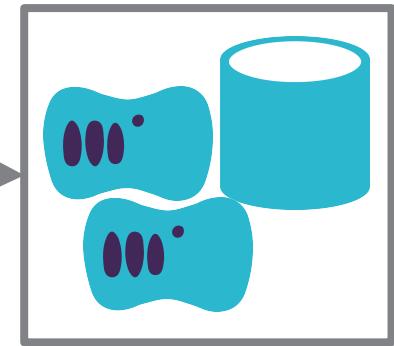
Structure your code to minimize risks for
making (small, frequent) changes

One definition
per environment

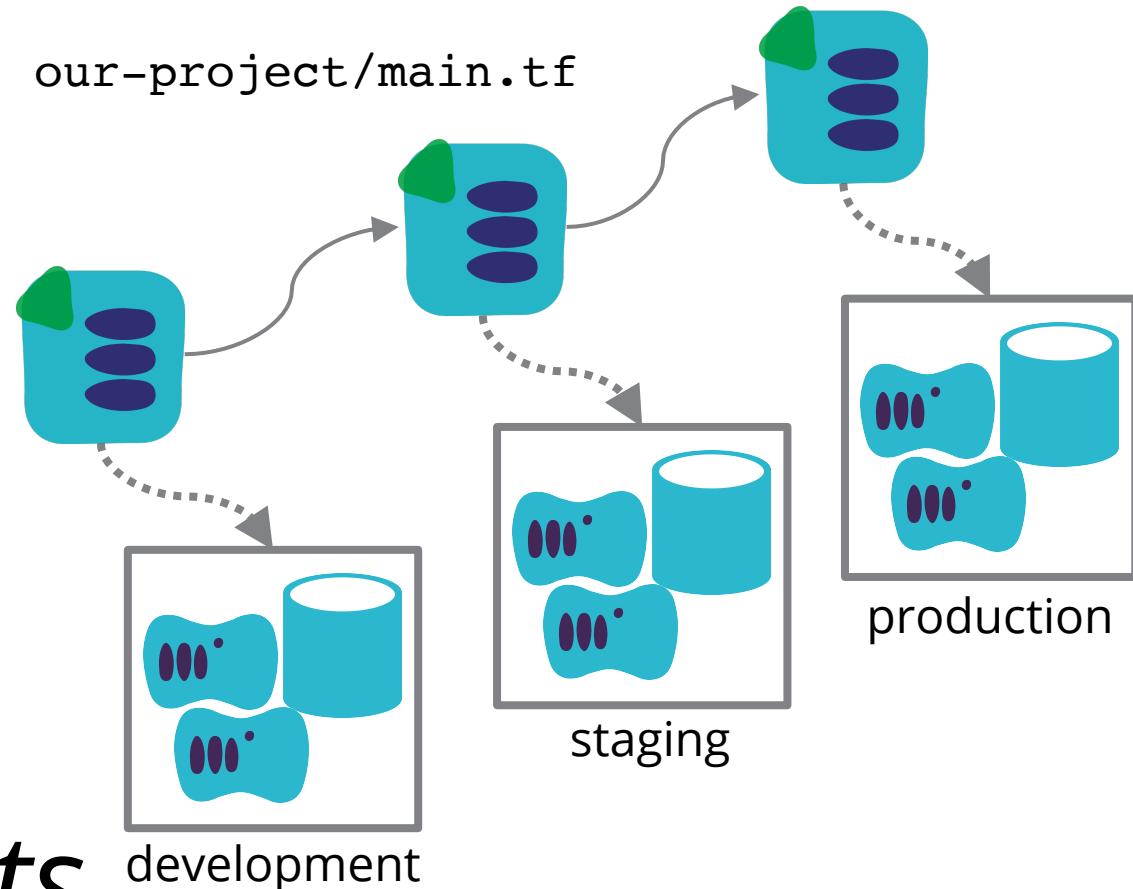
our-project/staging/main.tf



our-project/production/main.tf



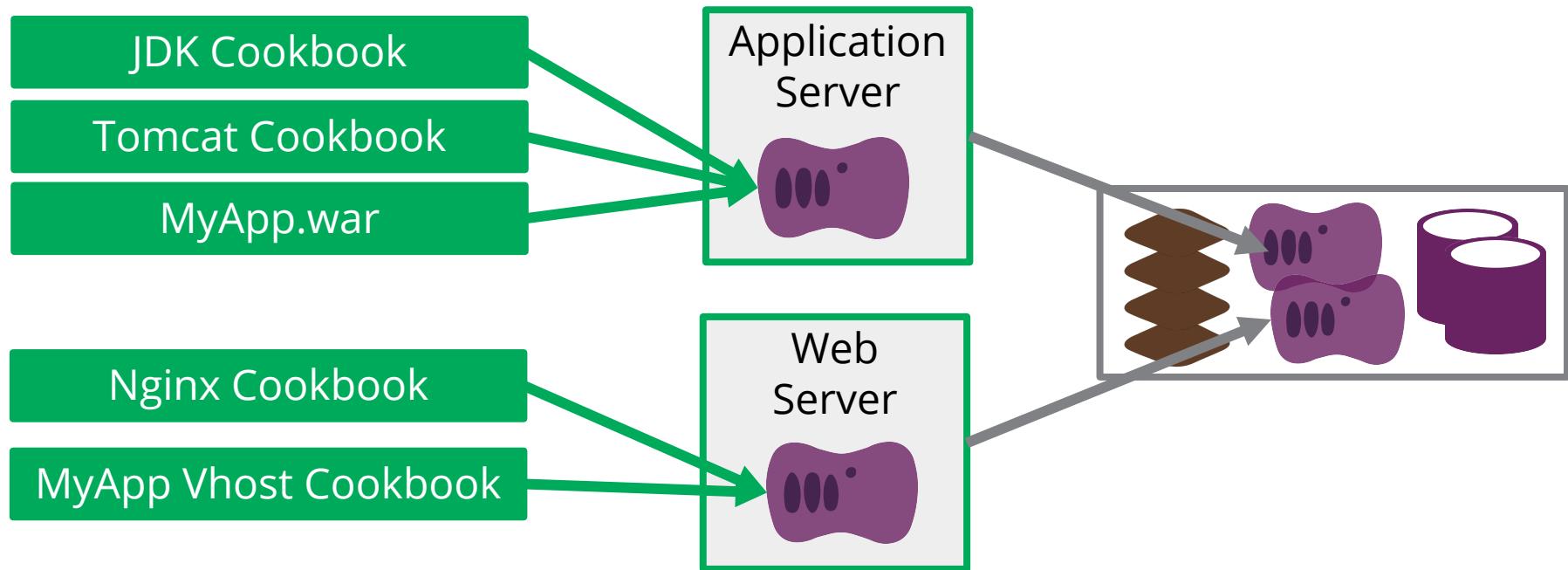
**Single
definition
template,
promoted
across
*environments***



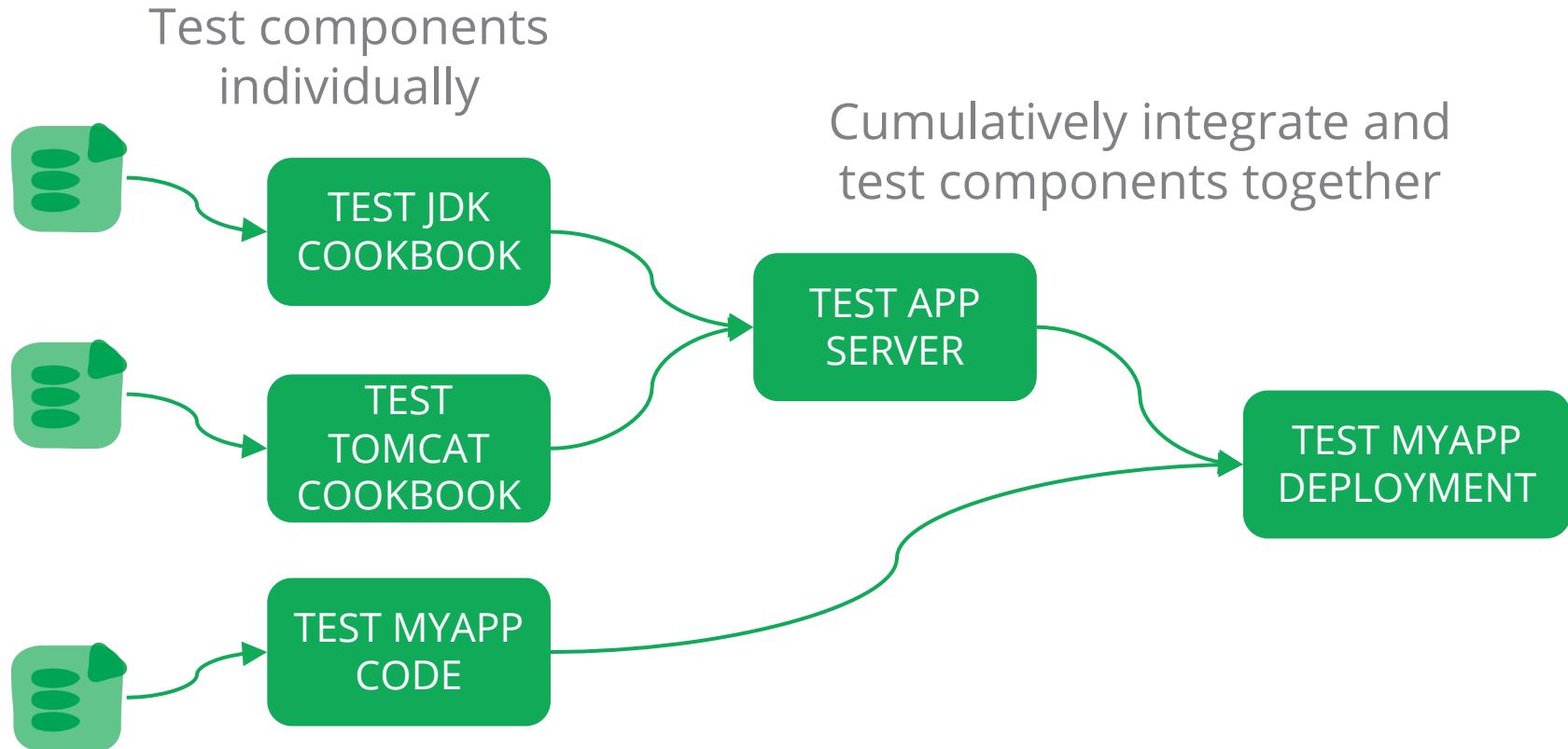
How do we build and test this stuff?

It runs sooooo sloooooow! 😞

ORGANIZE INFRASTRUCTURE INTO SEPARATELY TESTABLE PIECES



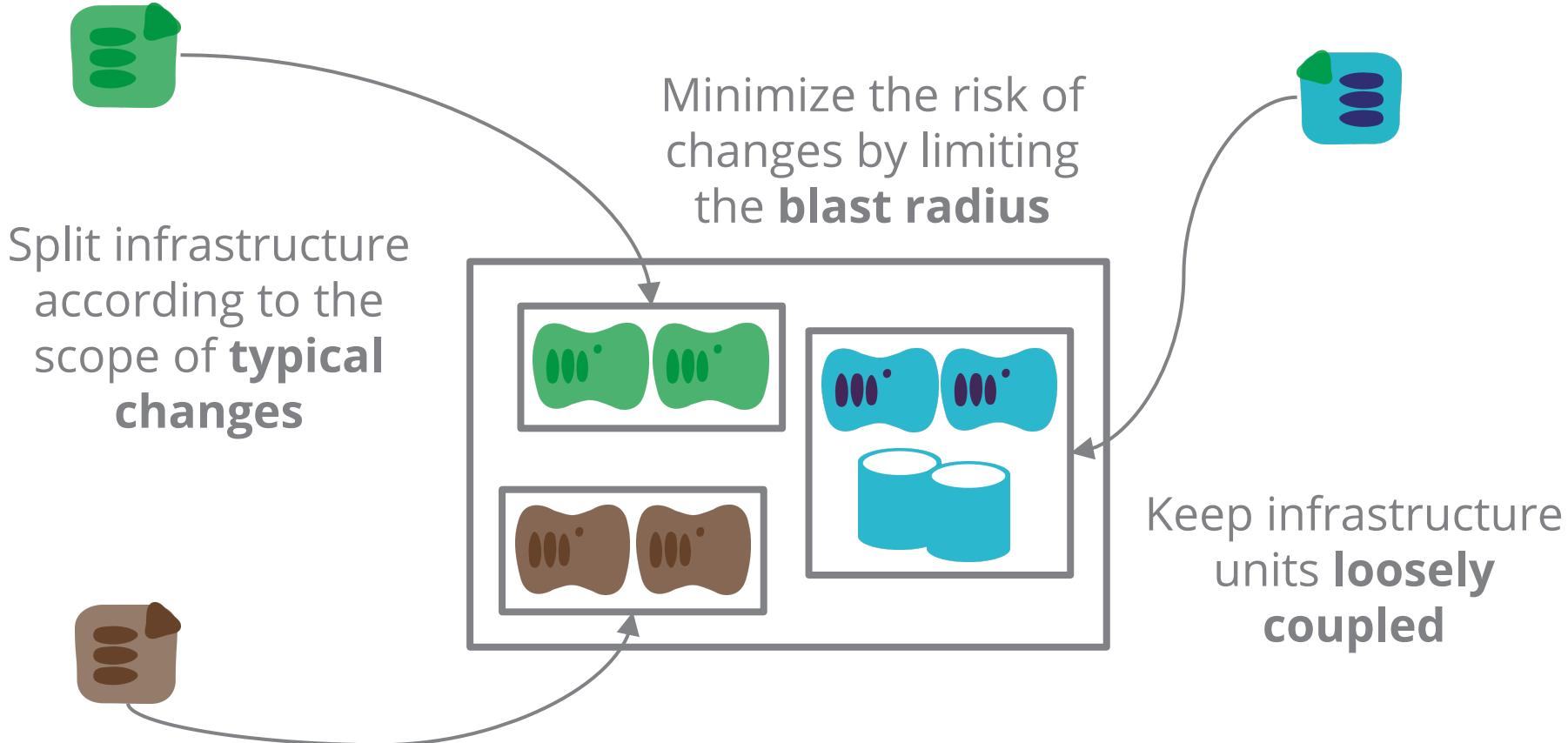
FAN-IN PIPELINES



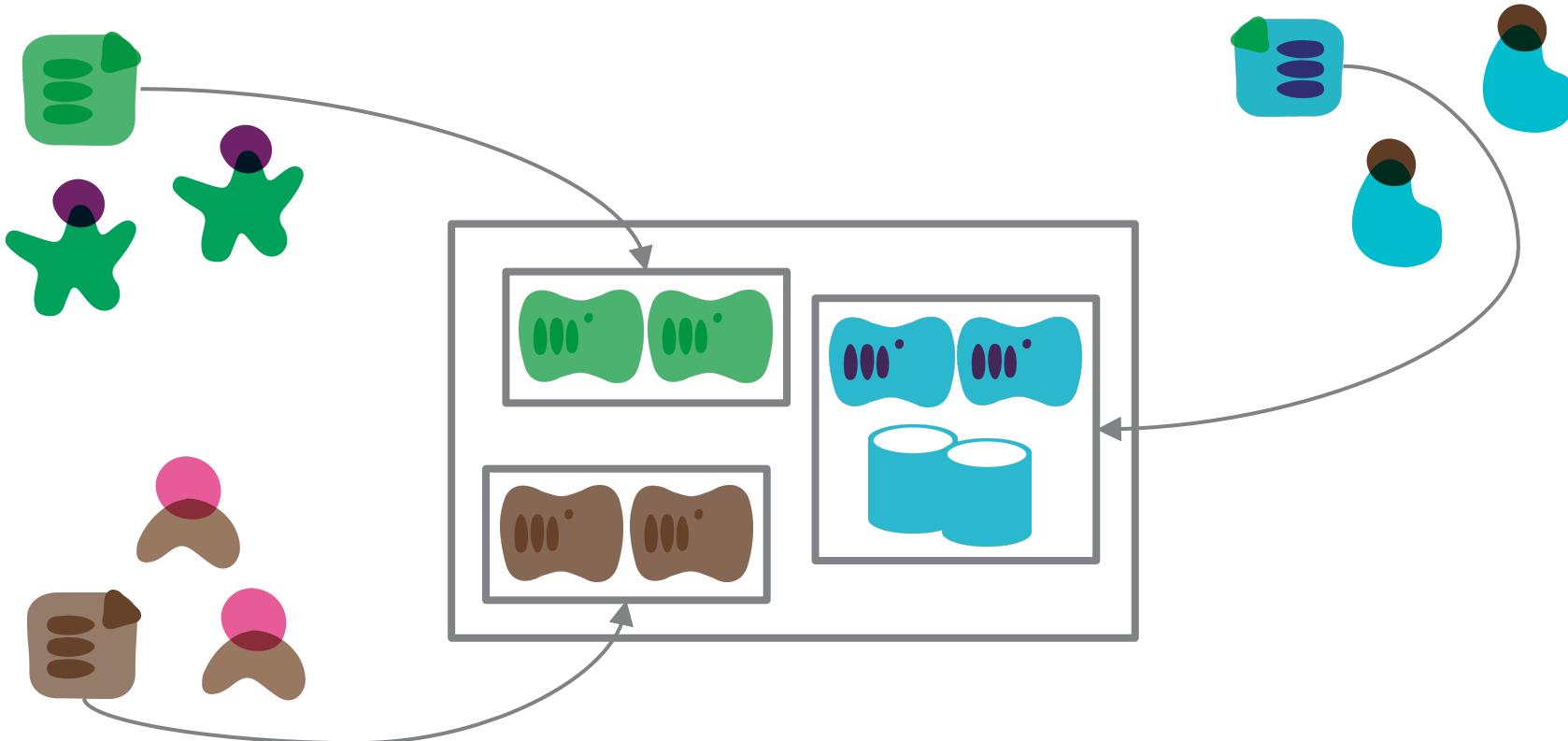
DESIGNING FOR CHANGE

Cope with growth, evolving
requirements, expanding teams

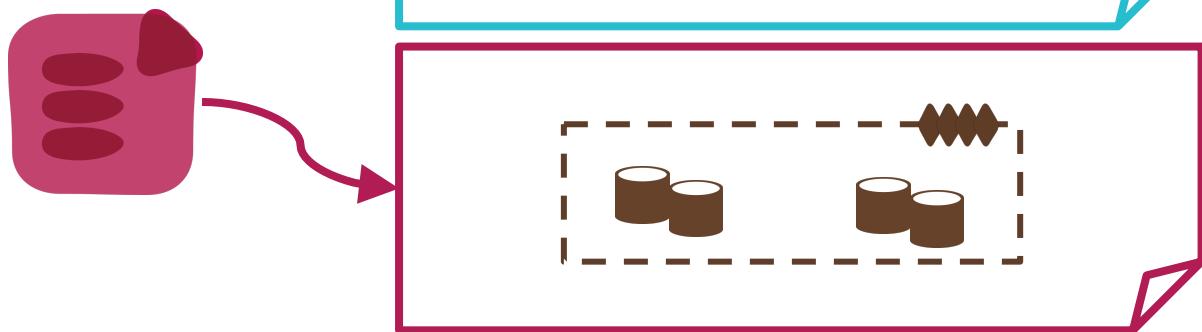
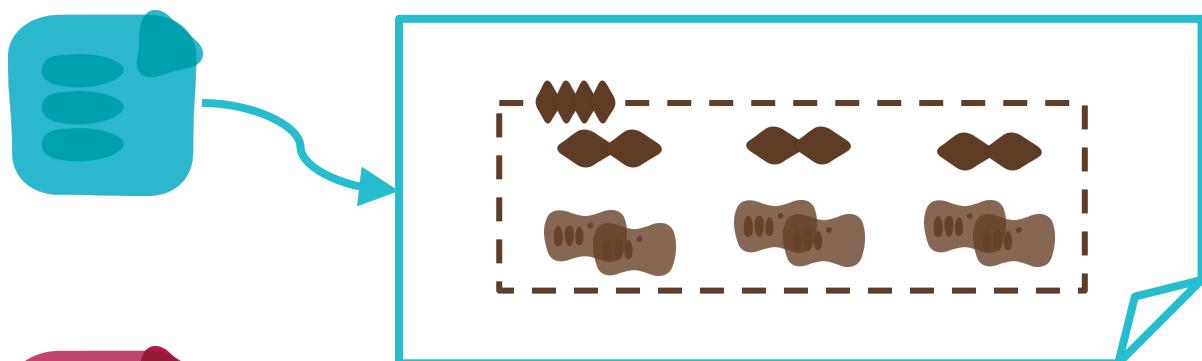
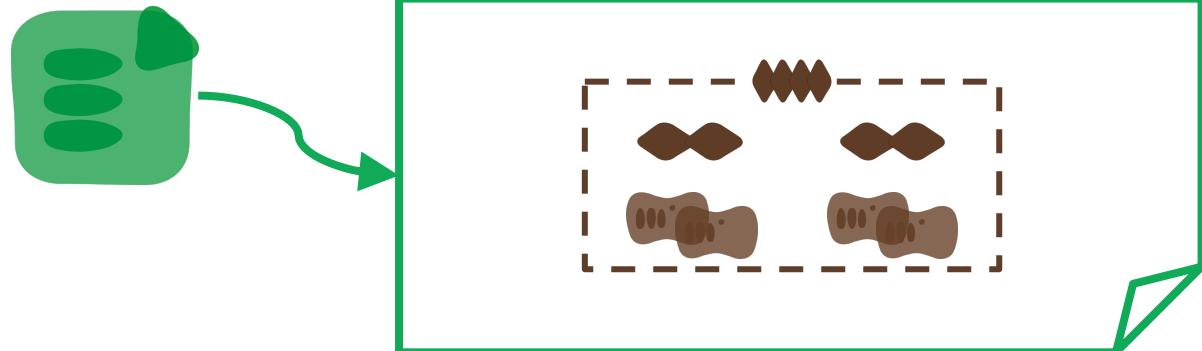
DESIGN TO ENABLE FREQUENT CHANGES



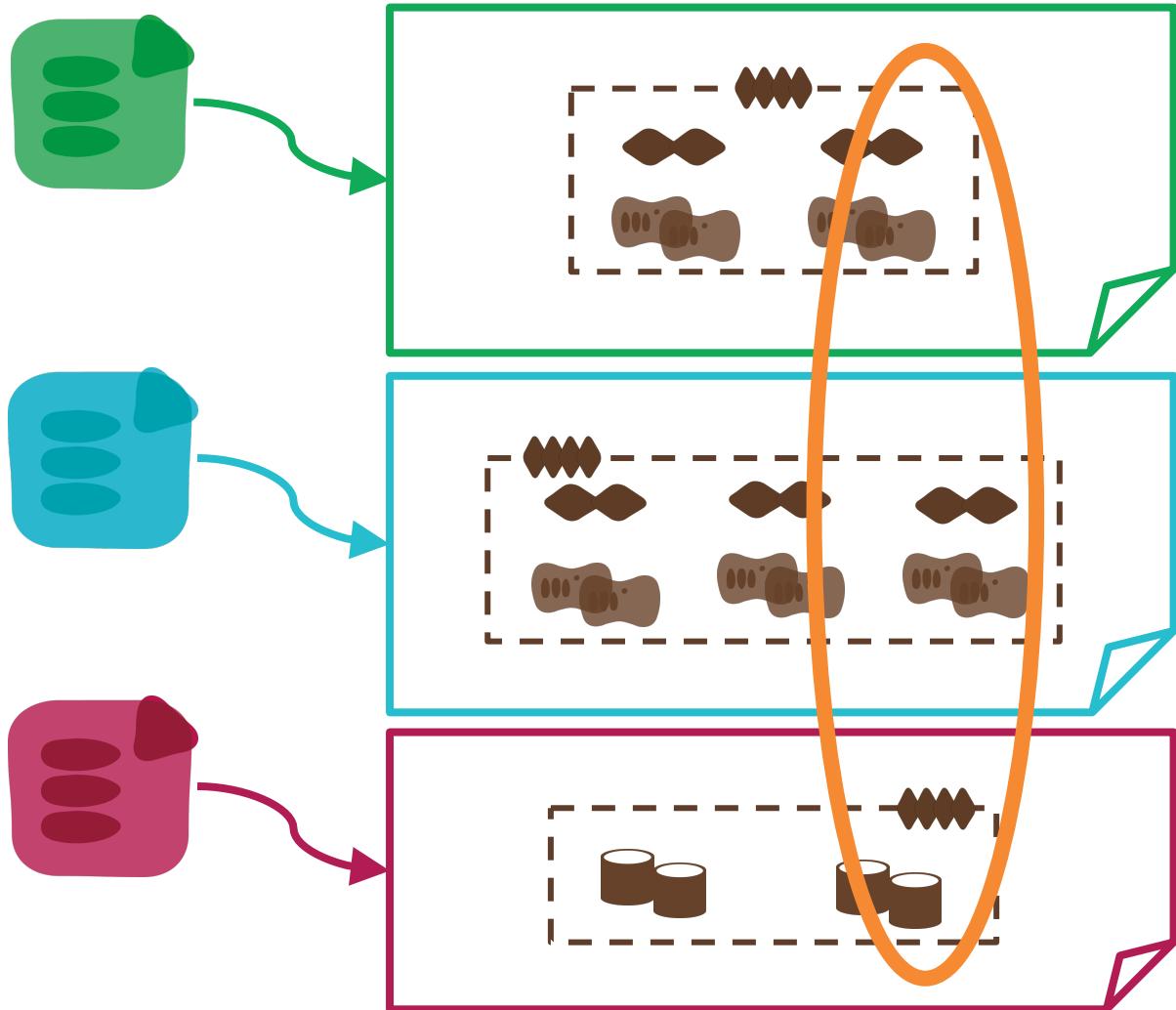
ALIGN INFRASTRUCTURE DESIGN WITH ORGANIZATION STRUCTURE



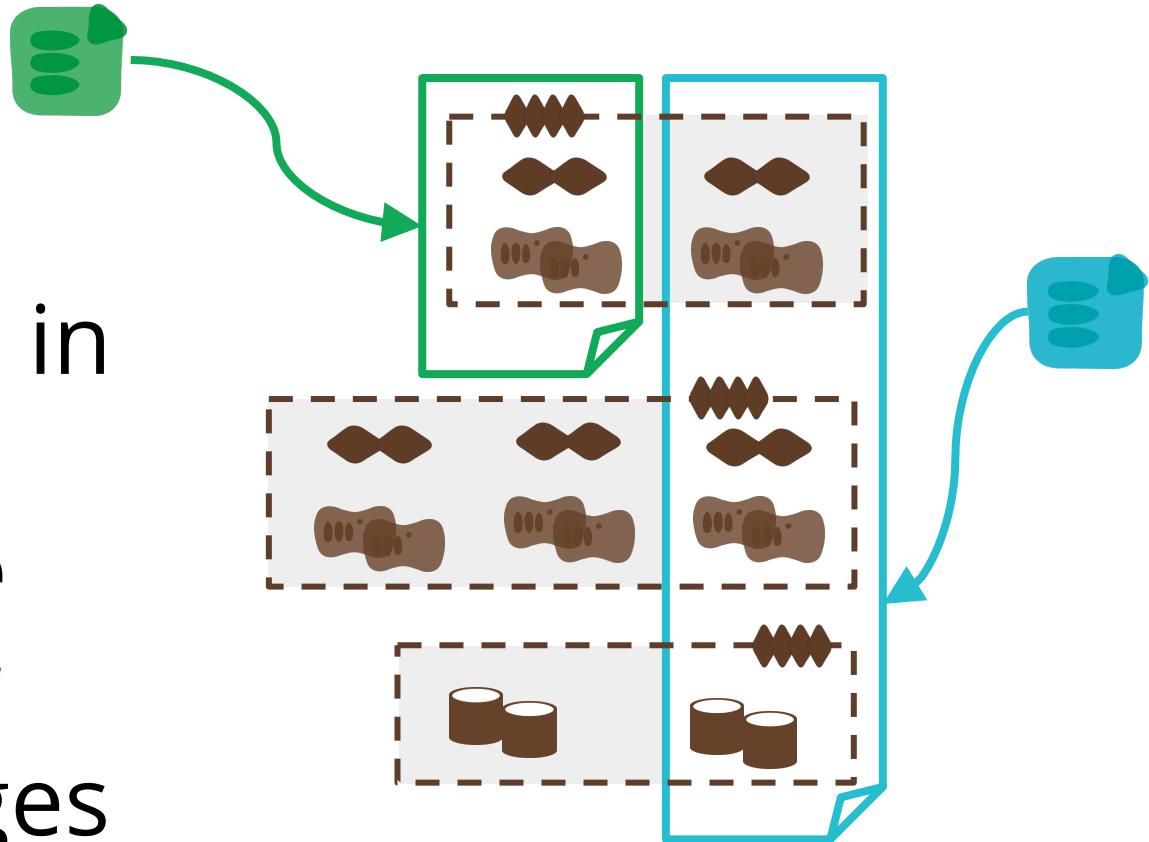
WHERE SHOULD WE SPLIT?



WHERE SHOULD WE SPLIT?



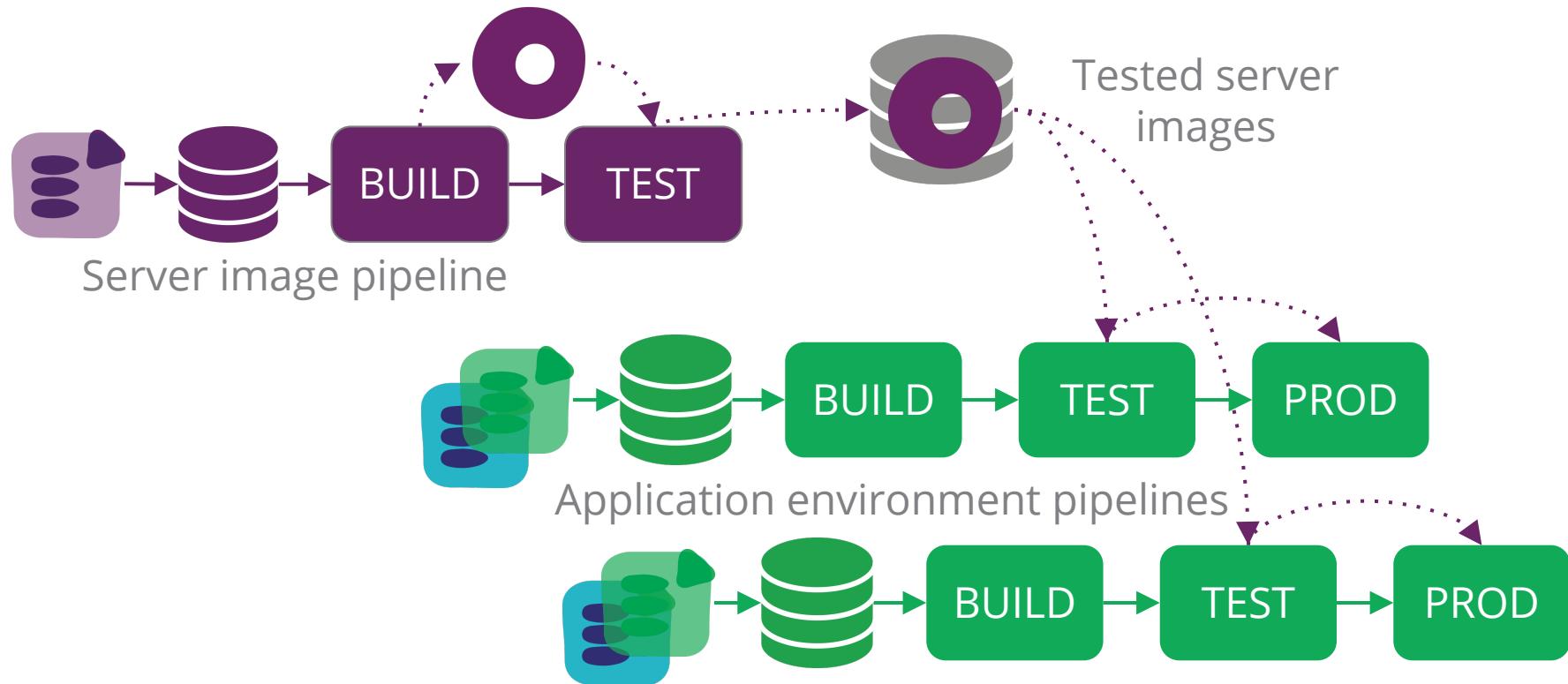
Split
infrastructure in
ways that
reduce the
friction for
making changes



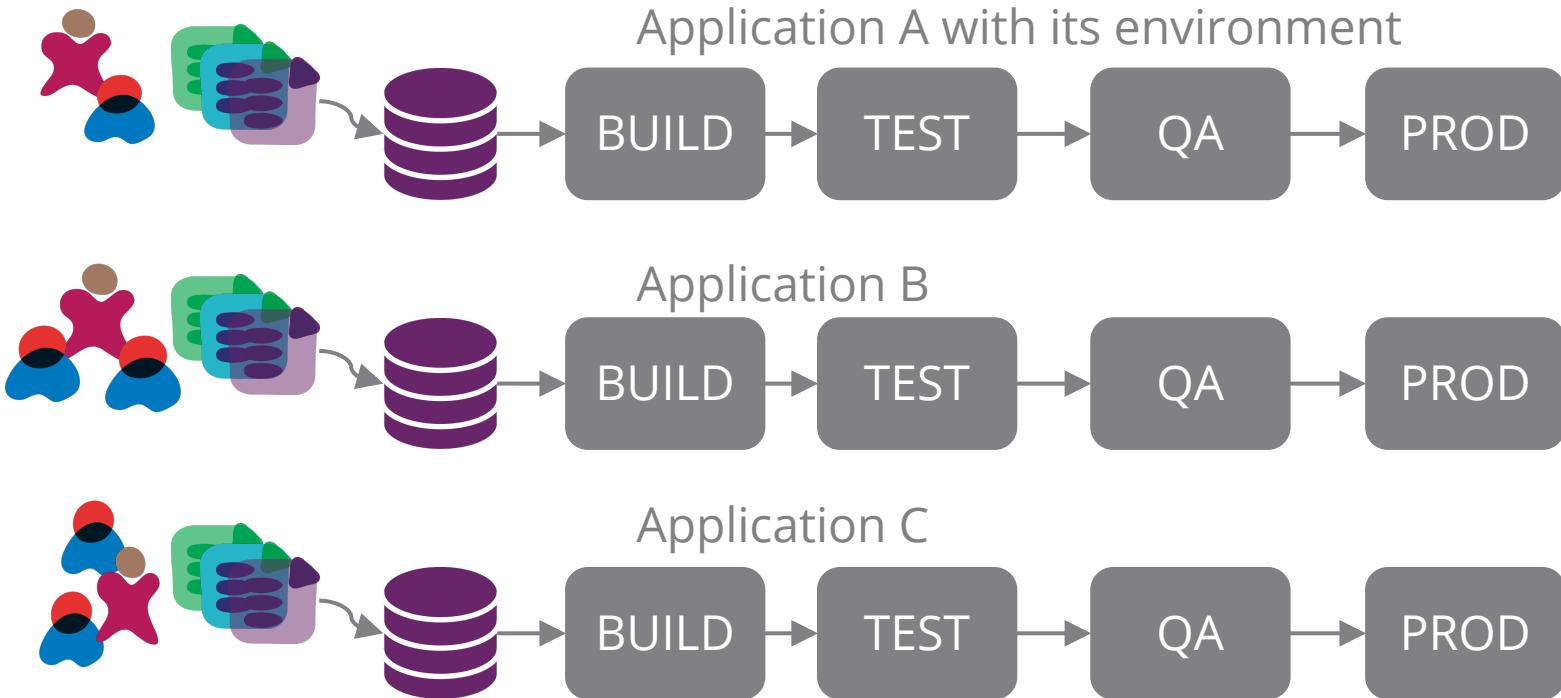
SHARING

How can teams share their stuff?

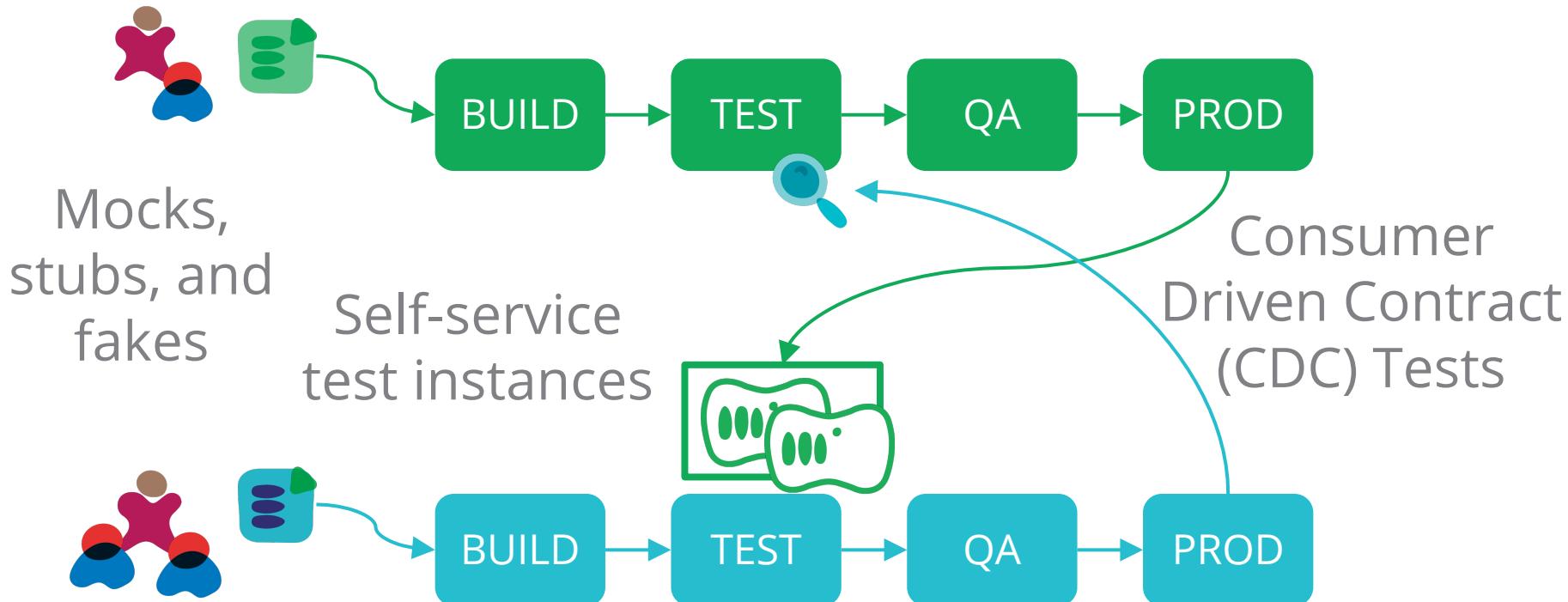
LIBRARY PATTERN FOR INFRA



DECOUPLED CHANGE PIPELINES



HANDLING DEPENDENCIES



CYCLE TIME

Measure and optimize the elapsed time from identifying a need to satisfying it

SOME ACTIVITIES TO MEASURE

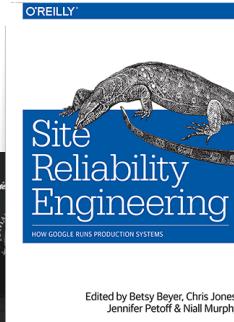
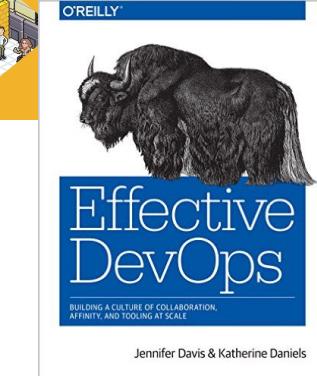
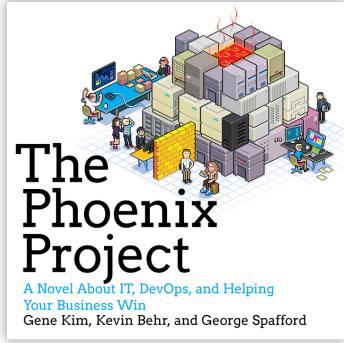
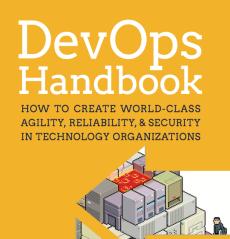
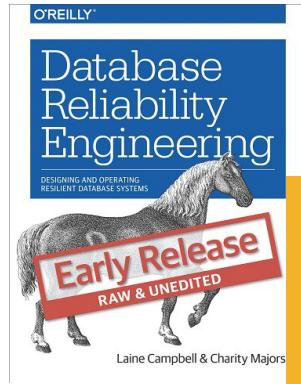
Rebuild (Recover)

New environment

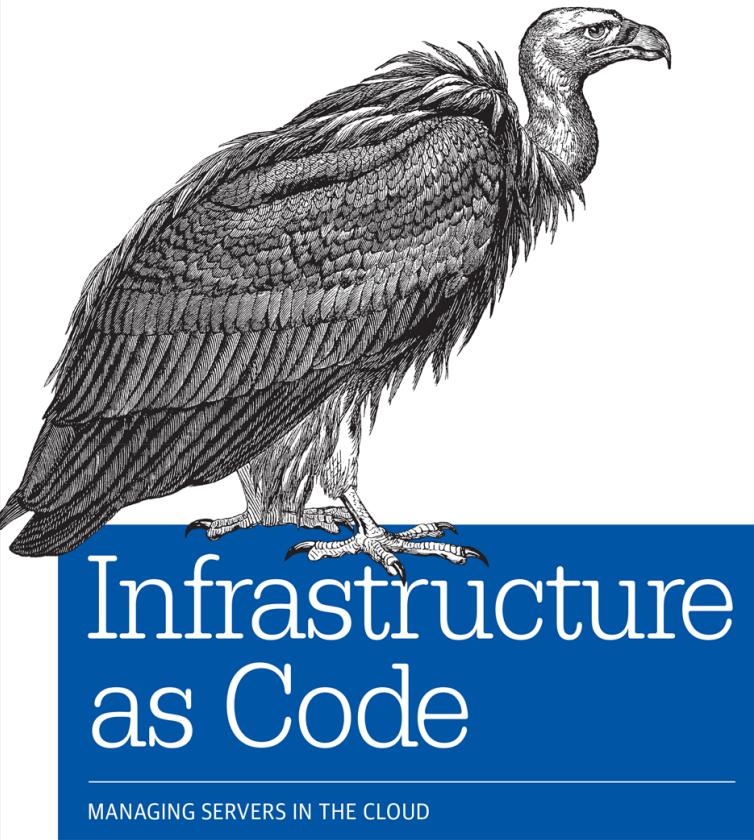
Update existing environments

Introducing a new tech stack

SOME OTHER BOOKS



ThoughtWorks®



Kief Morris

kief@thoughtworks.com

Cloud Practice Lead (UK)

DevOps, Continuous Delivery, Agile Ops

Book: <http://oreil.ly/1JKIBVe>

Site: <http://infrastructure-as-code.com>

Twitter: @kief

ThoughtWorks®