safespring

Anders Bruvik

Infrastructure engineer at Safespring

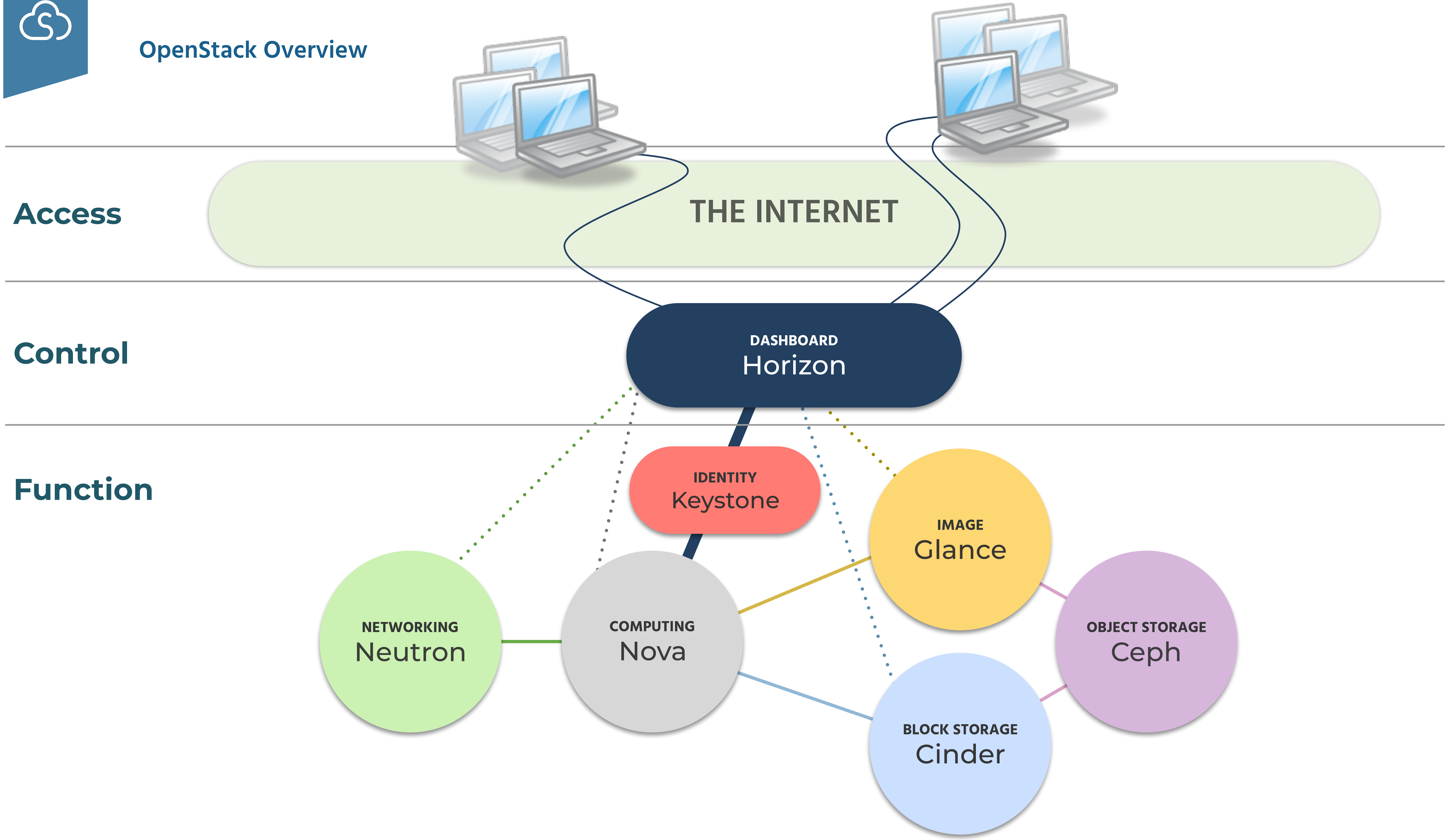
@bruvik

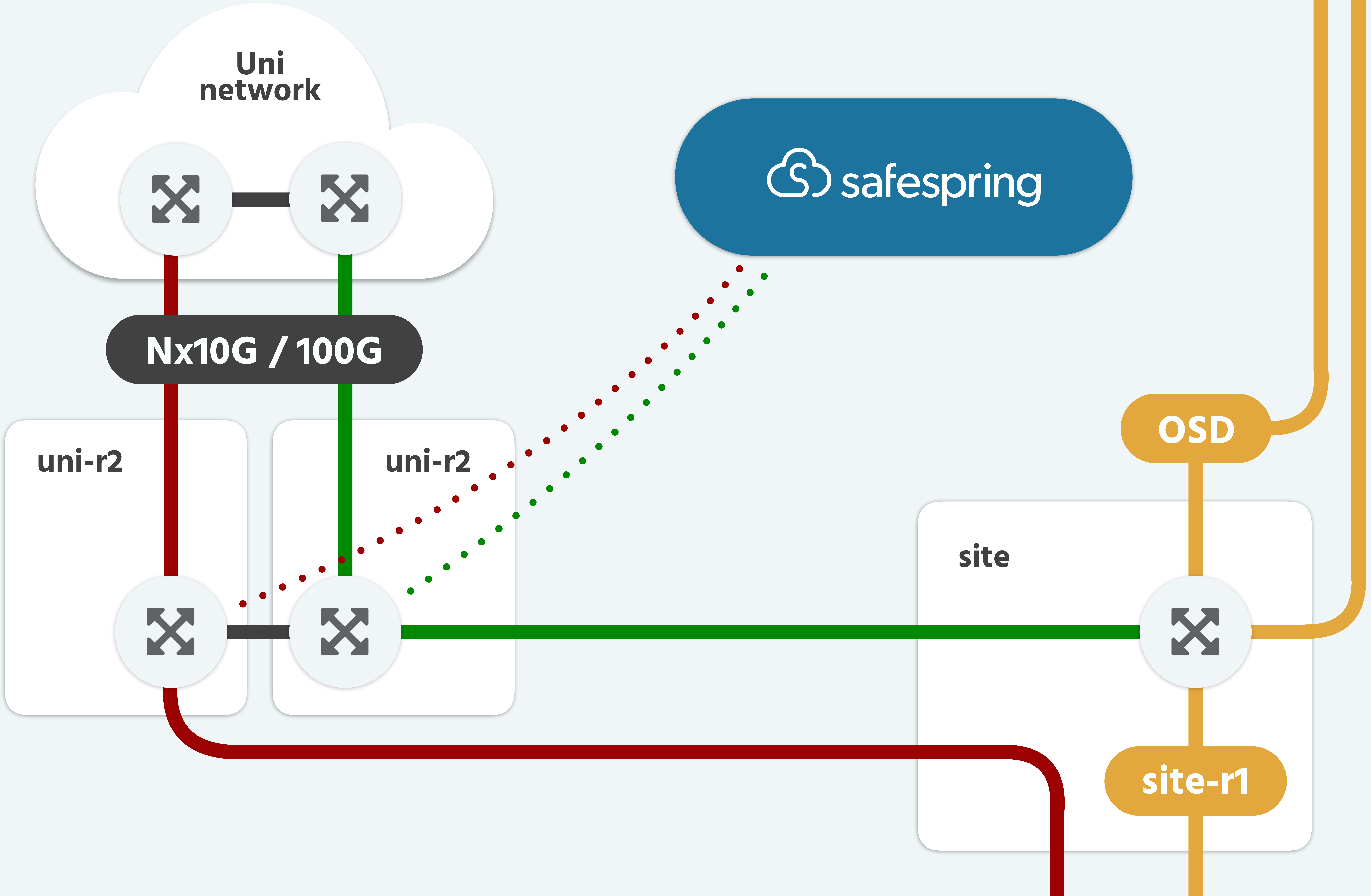
Infrastructure as Code

Infrastructure

The basic physical and organizational structures and facilities (e.g. buildings, roads, power supplies) needed for the operation of a society or enterprise.

OpenStack Overview




Uni network
Nx10G / 100G
uni-r2
uni-r2
safespring
OSD
site
site-r1

DevOps

What is **NOT** DevOps?

- A tool
- A role
- A job description
- A team

CAMS

- **C**ulture
- **A**utomation
- **M**easurement
- **S**haring

DEVOPS

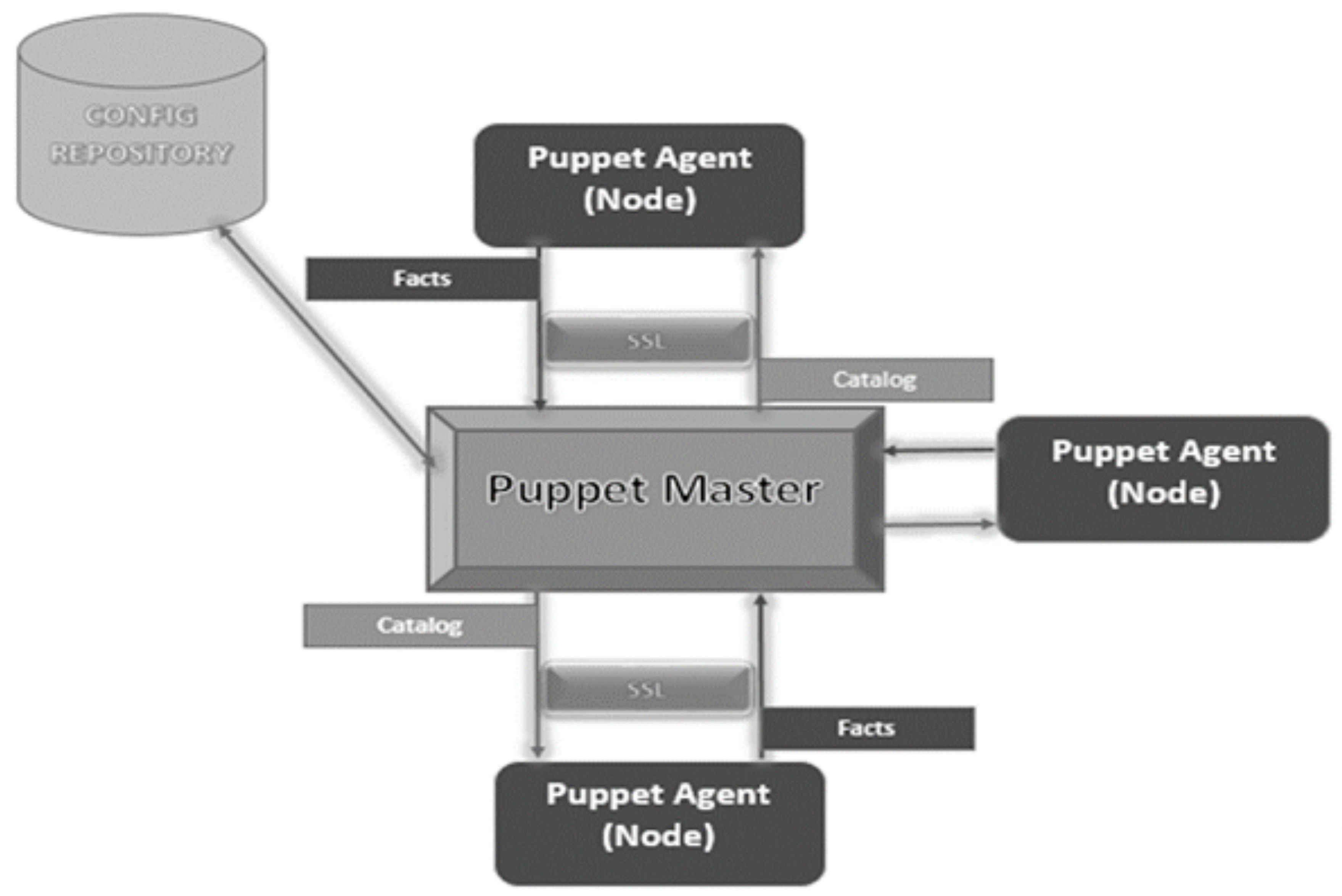
A culture where people, regardless of title or background, work together to imagine, develop, deploy and operate a system – **Ken Mugrage**

Automation

Why?

- Faster to production
- Lower risk of human errors
- Spending more time on valuable tasks
- Support change
- Quicker recovery from failures
- Self documenting
- Continuous improvements

Safespring **DevOps**



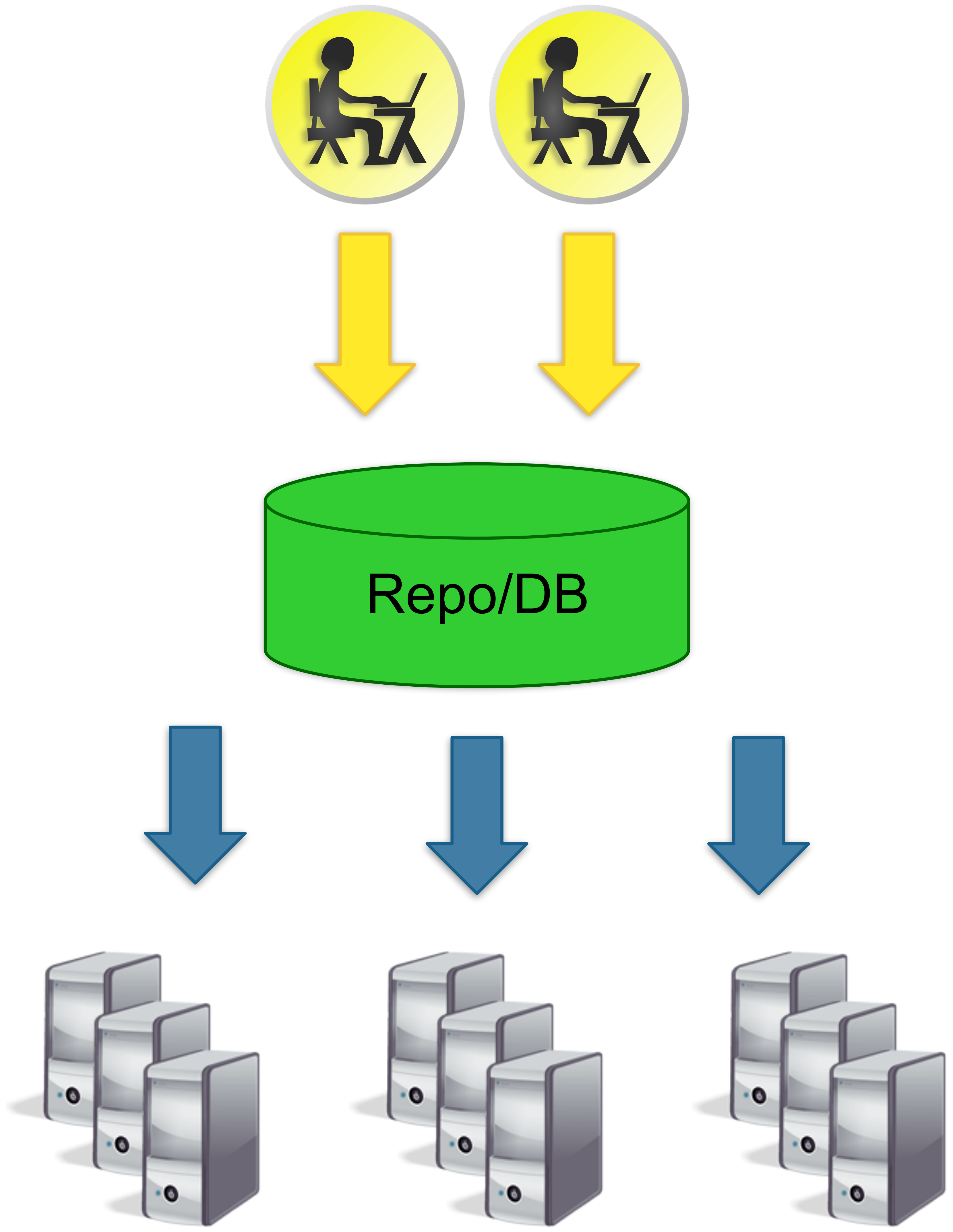
First generation Infrastructure as Code – Puppet

- Puppet Master holds all facts in Puppet DB
- All machines must have Puppet master access
- IPtables, DNS names, certificates generated from Puppet Master
- Facts describing node type (compute, storage or control) in Puppet DB
- Optimized for fleet management (large group homogenous nodes)
- If you do not know exact state of the node Puppet can help you streamline all nodes

Monolithic stacks…

…are not inherently bad — in fact, they are often the best choice for an organization early in a product life cycle.

Puppet - Workflow



Operators working with code

When doing a change the operator must find out how to reach the goal of the operation without unwanted side effects

Target: all servers

First generation Infrastructure as Code – Problems

- Configuration drift – machines out of sync
- Hard to make small changes
- Puppet is declarative and not imperative - in which order will the commands be run?
- Hard to bootstrap new sites since there are some circular dependencies

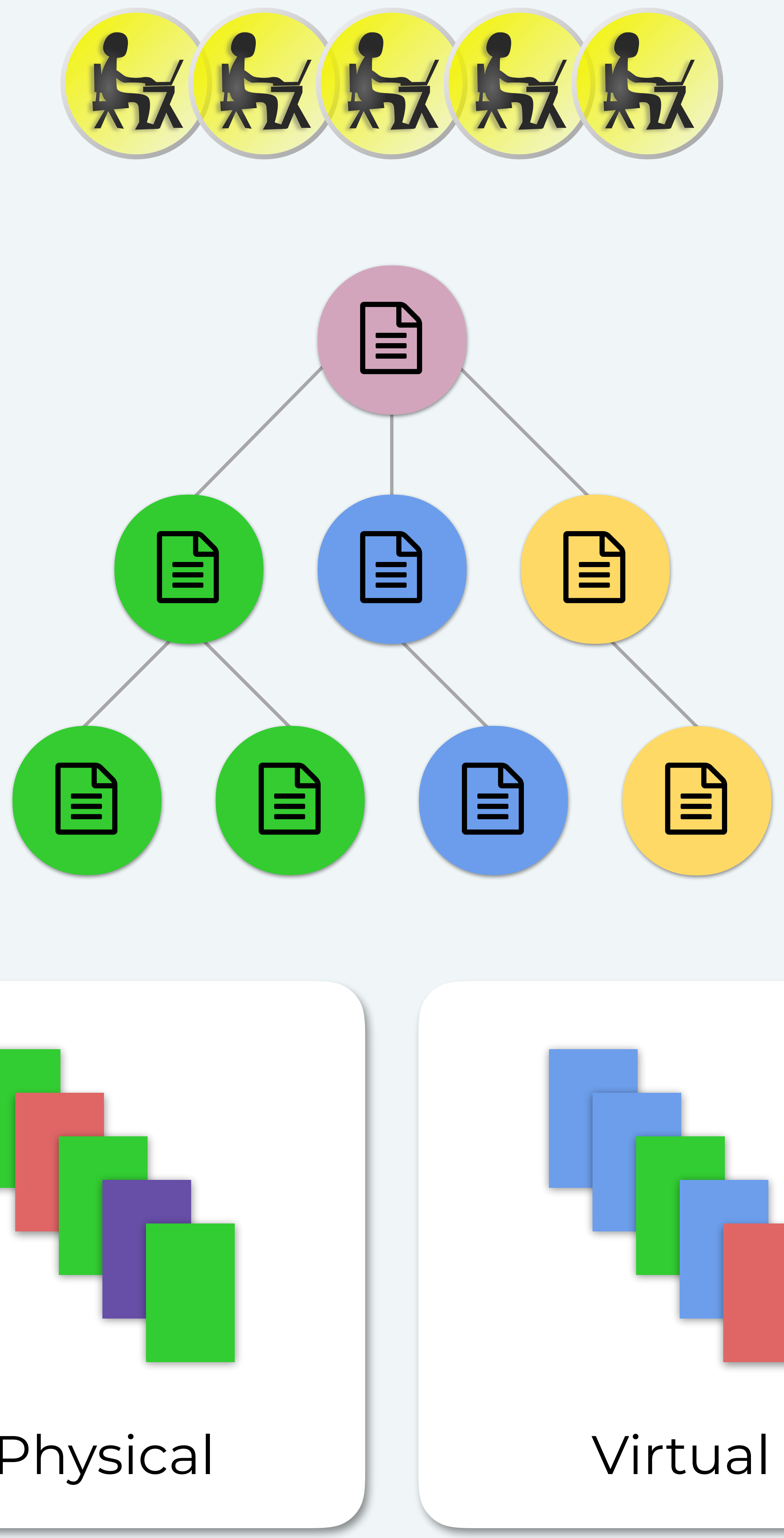
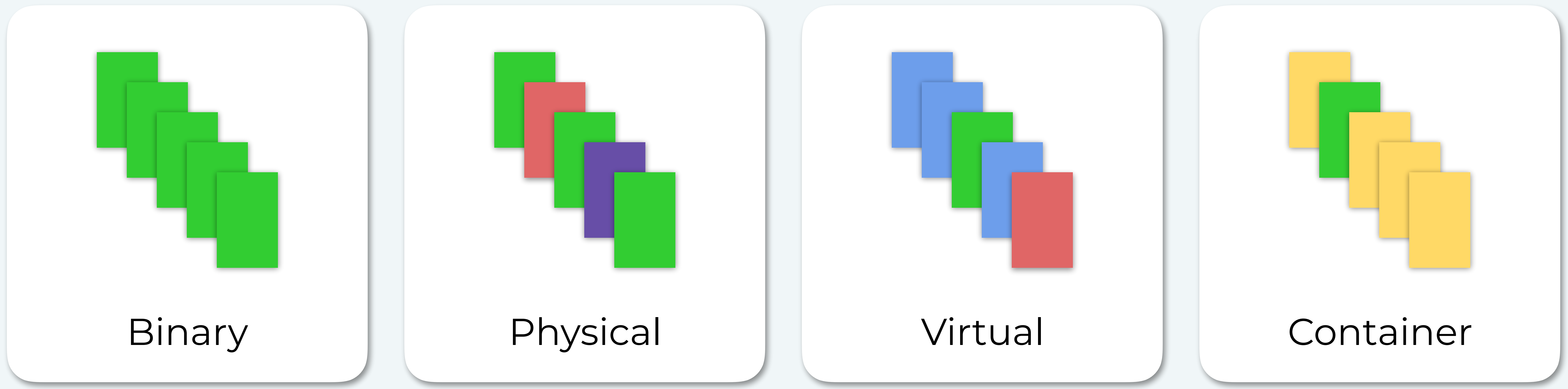
Breaking up the monolith

As systems grow – a monolithic
stack become an antipattern

Antifragile

- Systems that grow stronger during testing.
- The default response to incidents is improvement.
- Minimizing the number of changes will not make a system more robust.

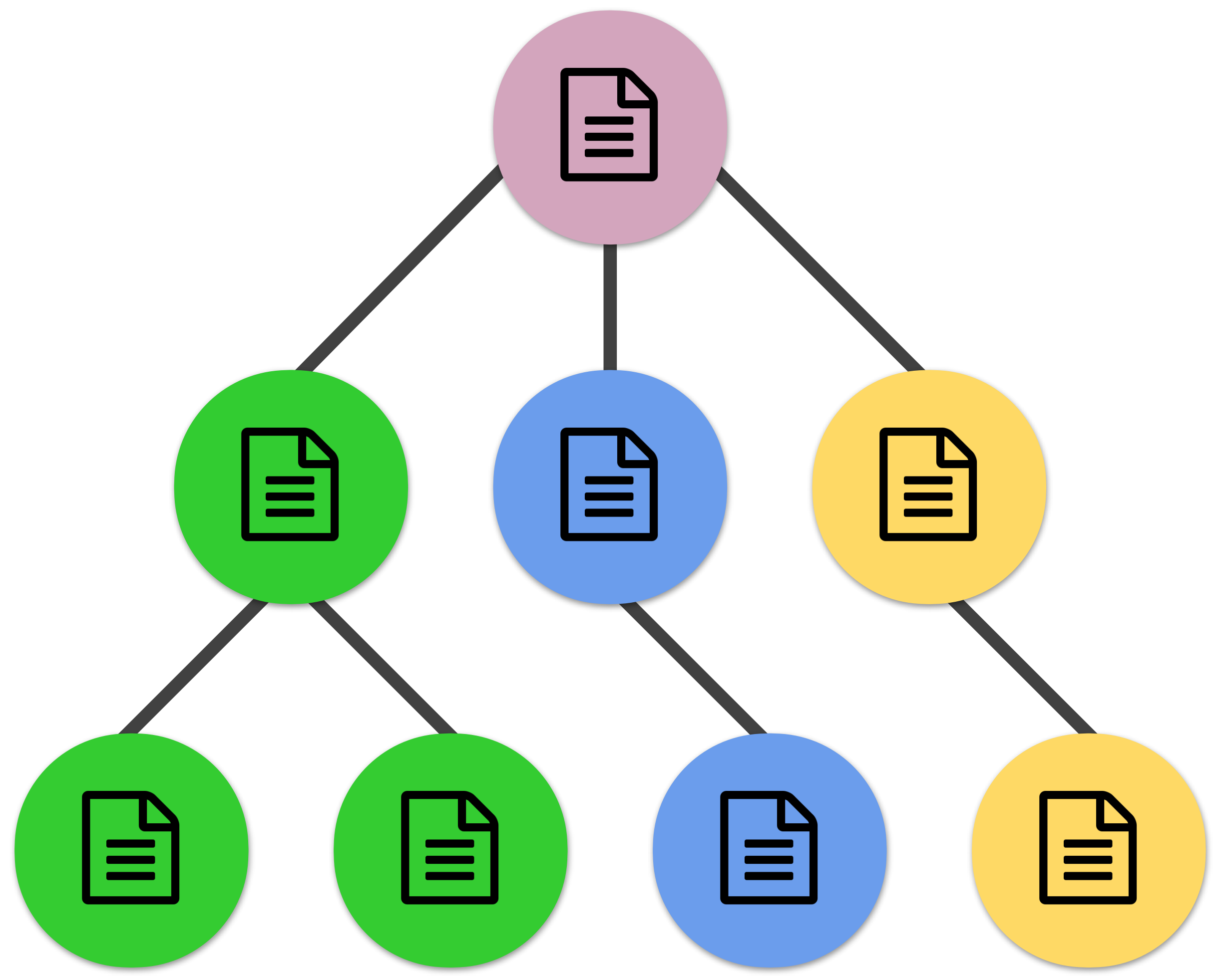
Safespring DevOps- Architecture

Second generation Infrastructure as Code – Workflow

What is needed?

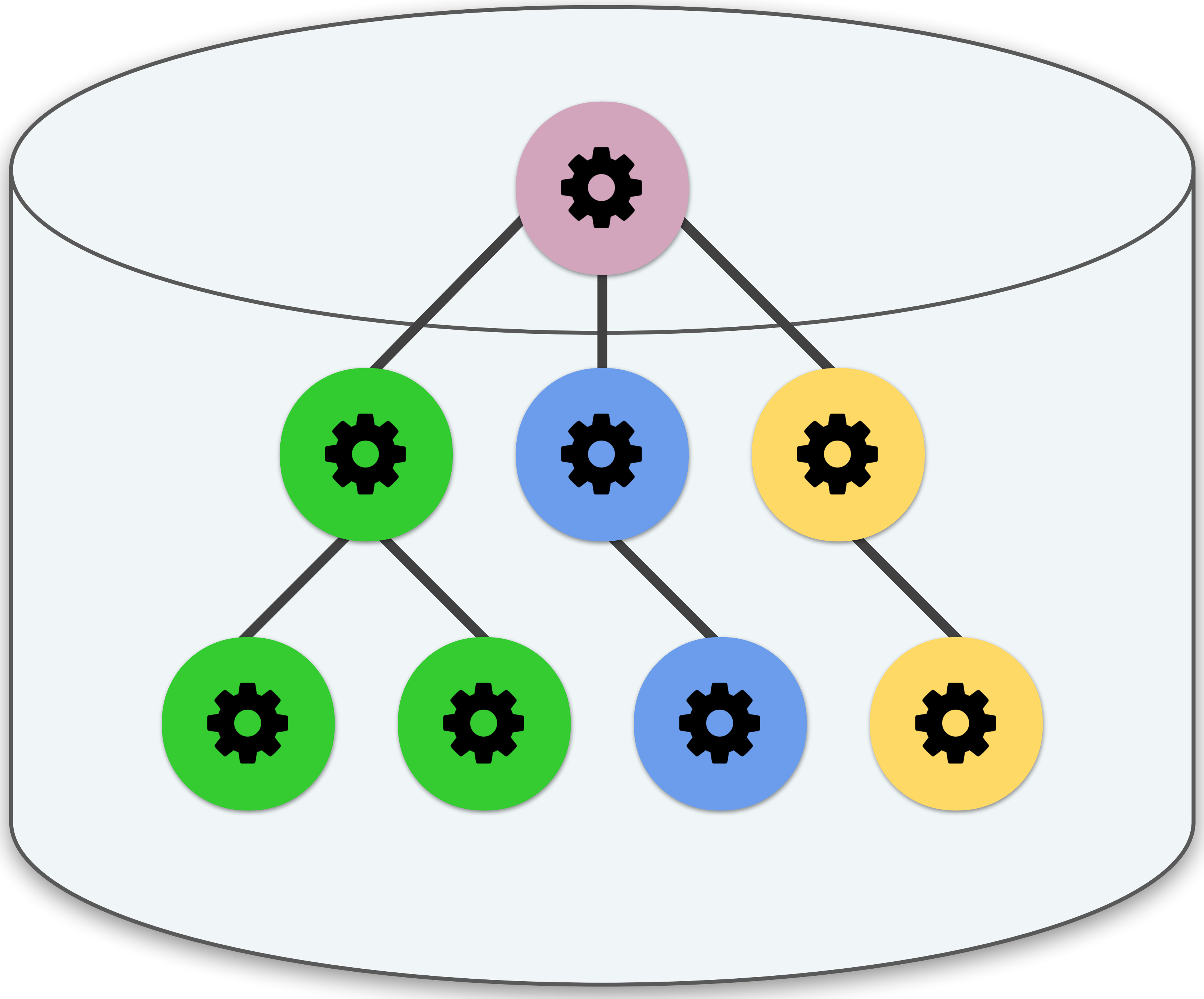
- A mechanism to build (Smie - Forge).
- A place to store artefacts - could be image, container or binary (Naust - boat house)
- Mechanism for deployment (Seter - settlement) that could describe different runtime environments



Second generation Infrastructure as Code – Smie

What is Smie?

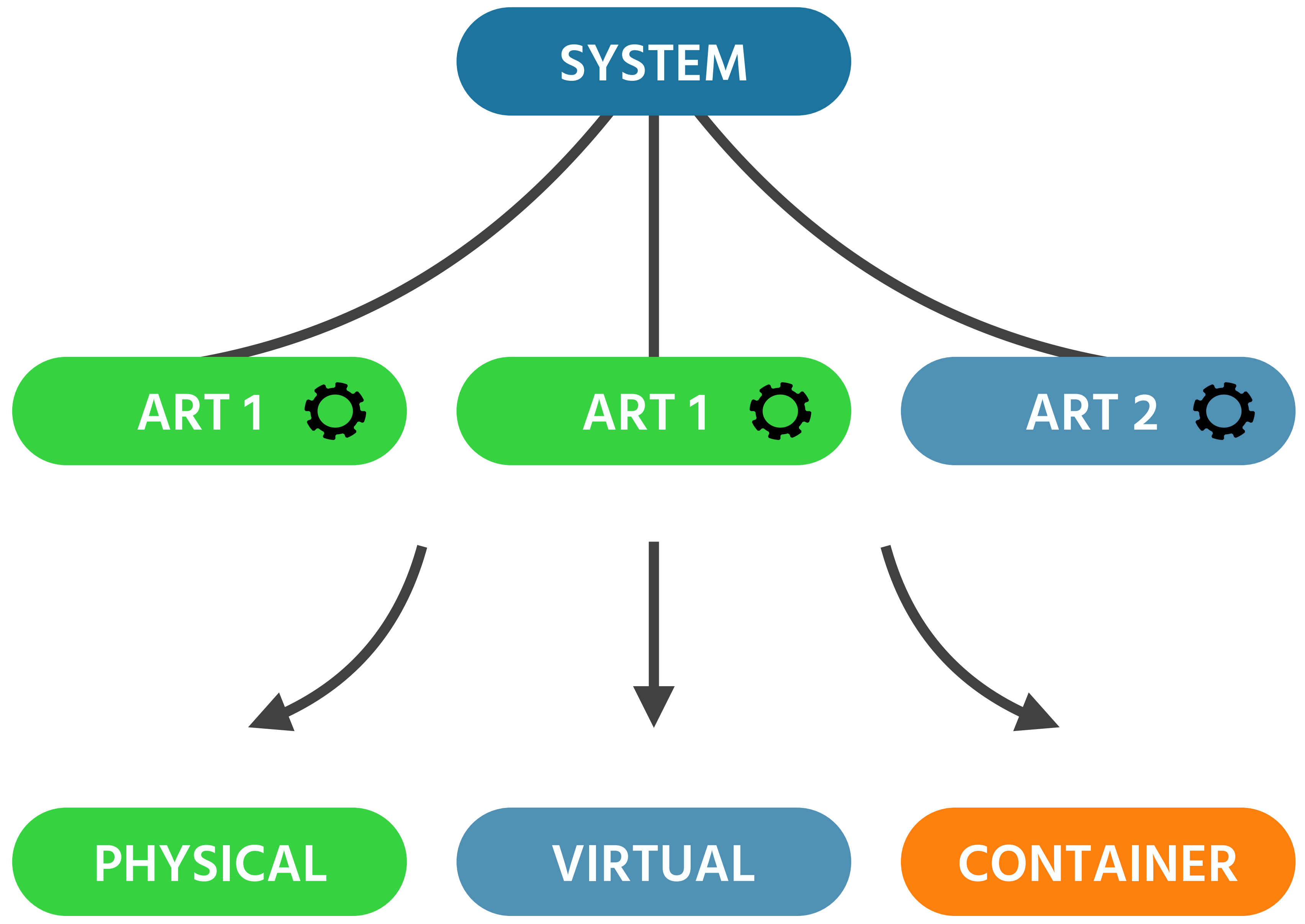
- Wrapper around Packer (Hashicorp)
- Produces artefacts
- All artefacts can be built separately
- Role: service, endpoint or component



Second generation Infrastructure as Code – Naust

What is Naust?

- Both source and destination for Smie (cut dependencies to Internet repos)
- Full control over everything built for production
- Protocols:
  HTTPS/file, S3, Docker Registry
- Protocols depend on target systems
- Everything built get an URI with metadata (type, version, date)



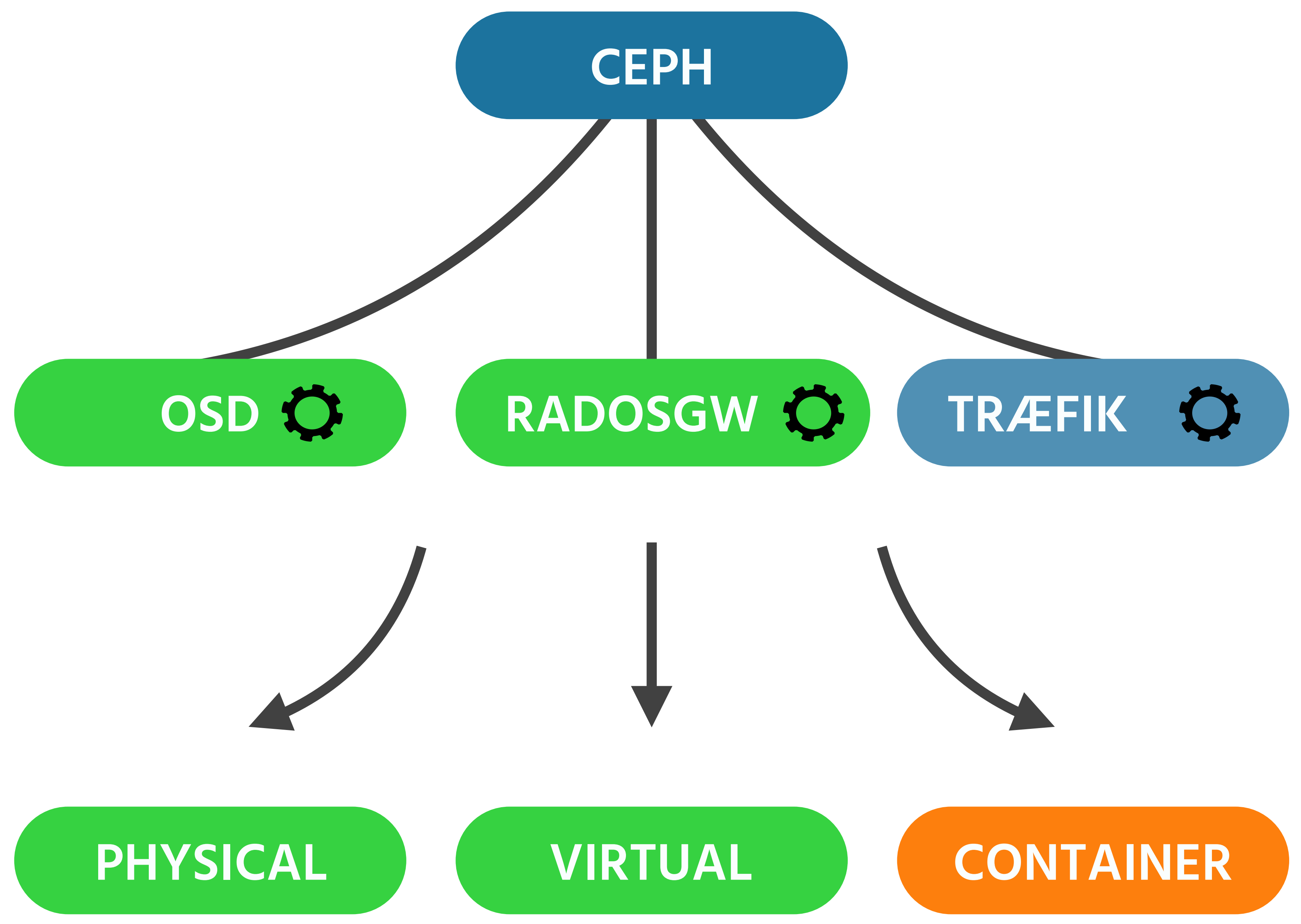
Second generation Infrastructure as Code – Seter

What is Seter?

- Provisioning mechanism
- Wrapper around Ansible and Terraform
- Describes a set of artefacts needed to get a component running
- Also describes target: Physical node, virtual node or container

Immutable infrastructure

- Changes done at templating stage
- New deployment preferred over change at host
- Easier to implement testing
- Simpler configuration management tooling



Example

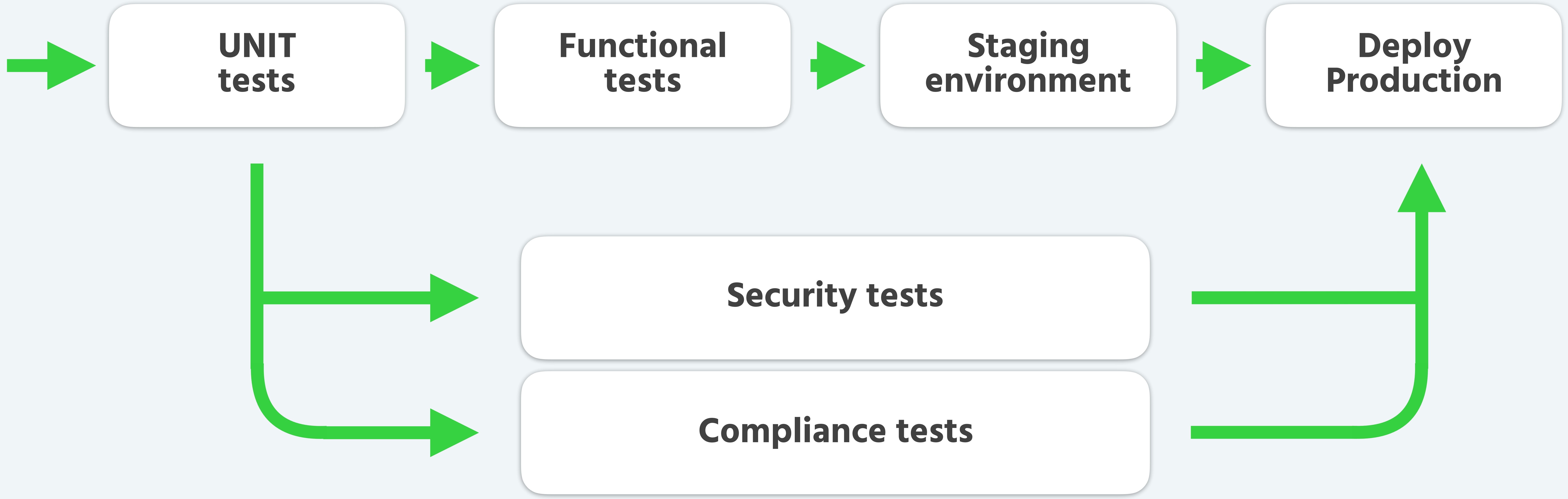
- Ceph Object Storage backend needs a set of OSD and RadosGW role images
- Træfik as load balancer – role reused across different stacks

Jerakia

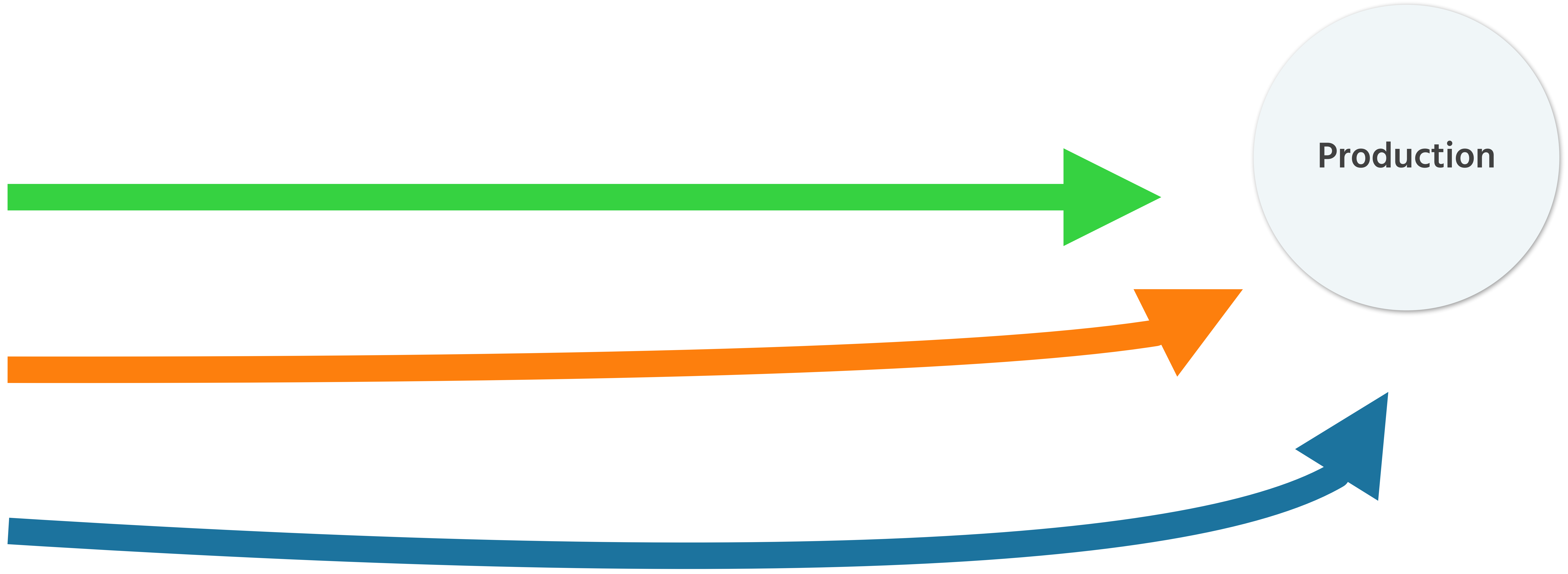
- Key/Value lookups against pluggable data stores
- Allows defining global values, and override at different levels of a hierarchy
- Open source project – Contributions from Safespring

Pipelines

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**


UNIT tests
Functional tests
Staging environment
Deploy Production
Security tests
Compliance tests

Multiple pipelines



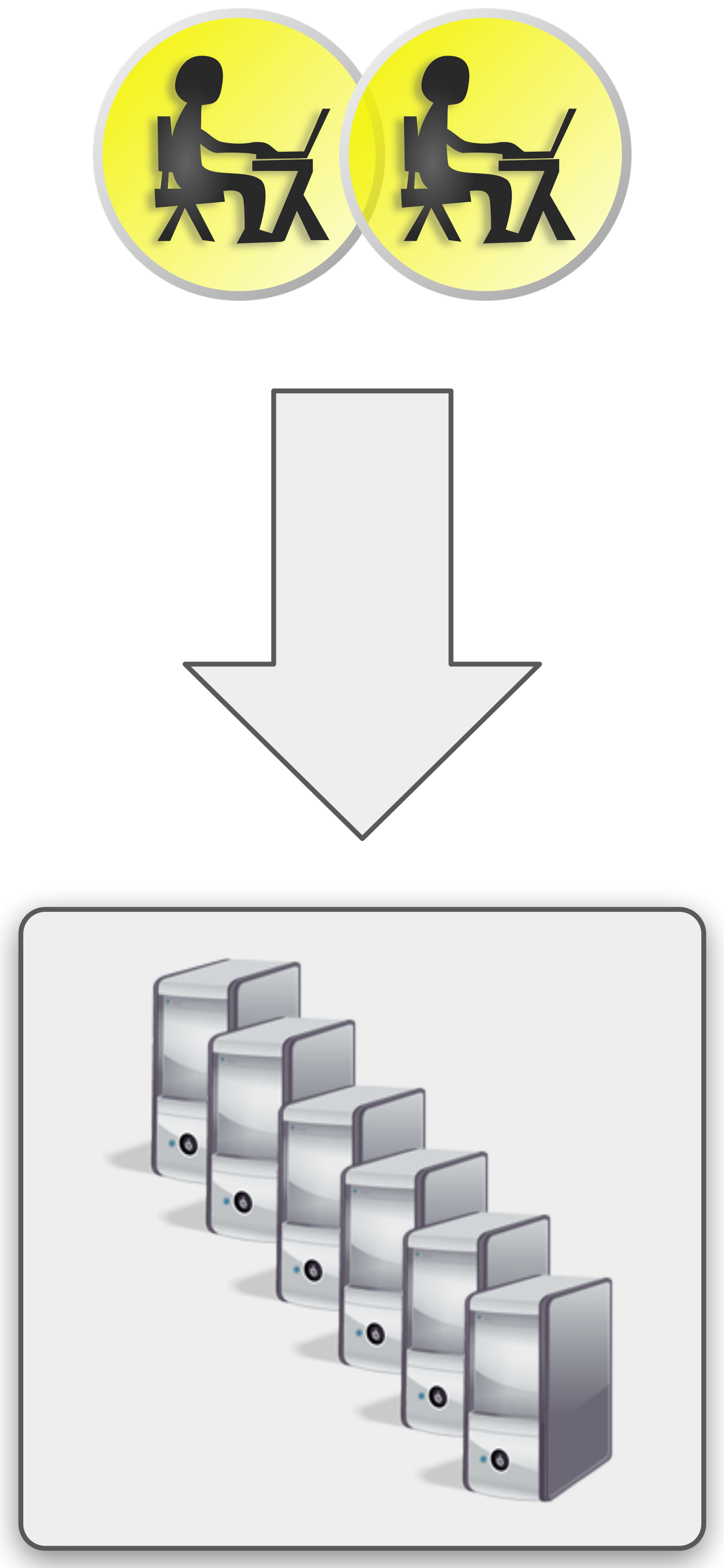
Second generation Infrastructure as Code – Advantages

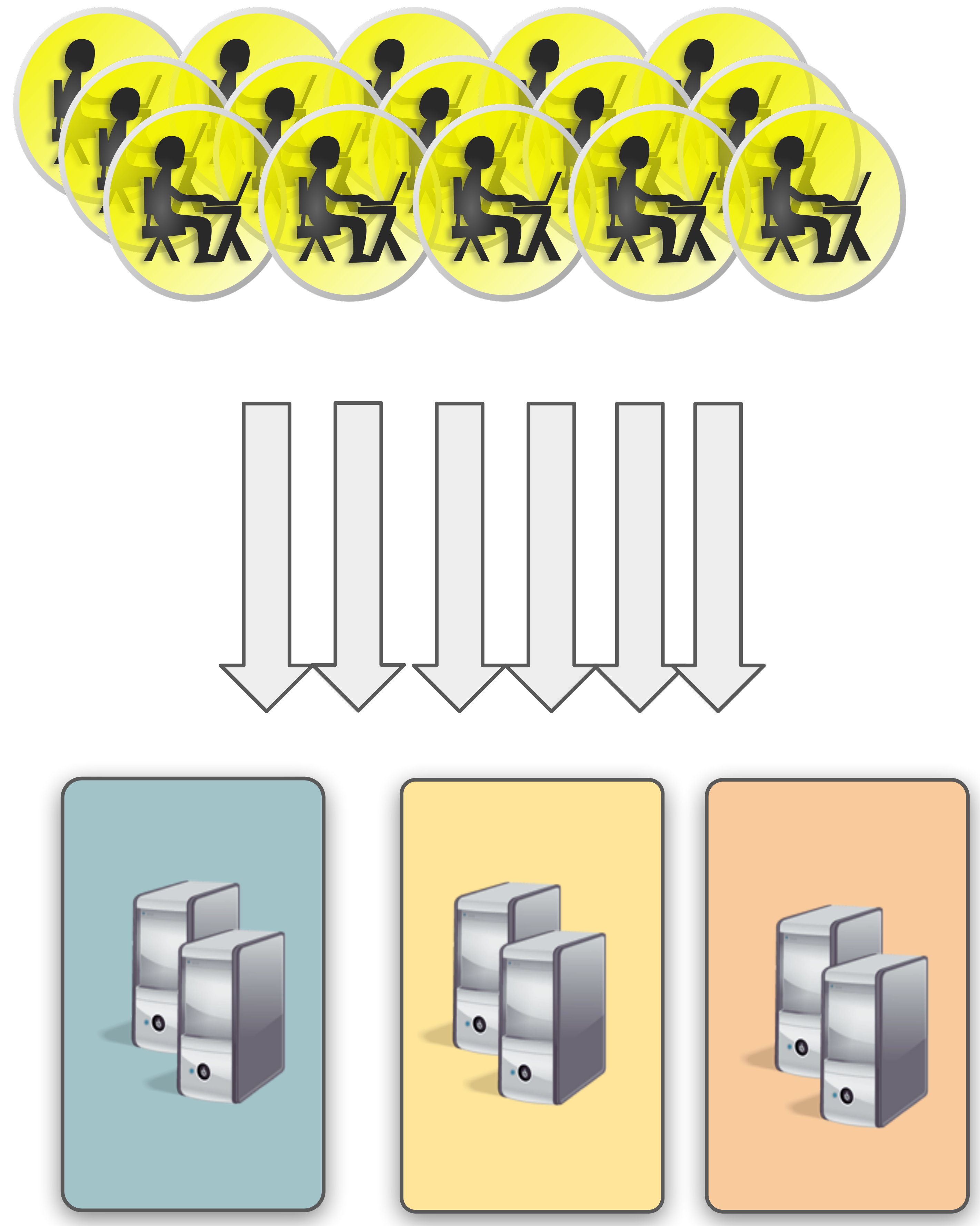
- Update systems faster
- Lower barrier to changes
- Reproduce systems as needed
- Build everything with as few dependencies as possible
- Add or change easily
- Target the affected nodes easily
- Verify that software works as intended
- Scales better with many operators

Testing

- It’s difficult to write automated tests for an existing, legacy system
- Restructuring a systems design in a way that facilitates independently testing components
- Test in production!

1. generation IaC versus 2. generation IaC

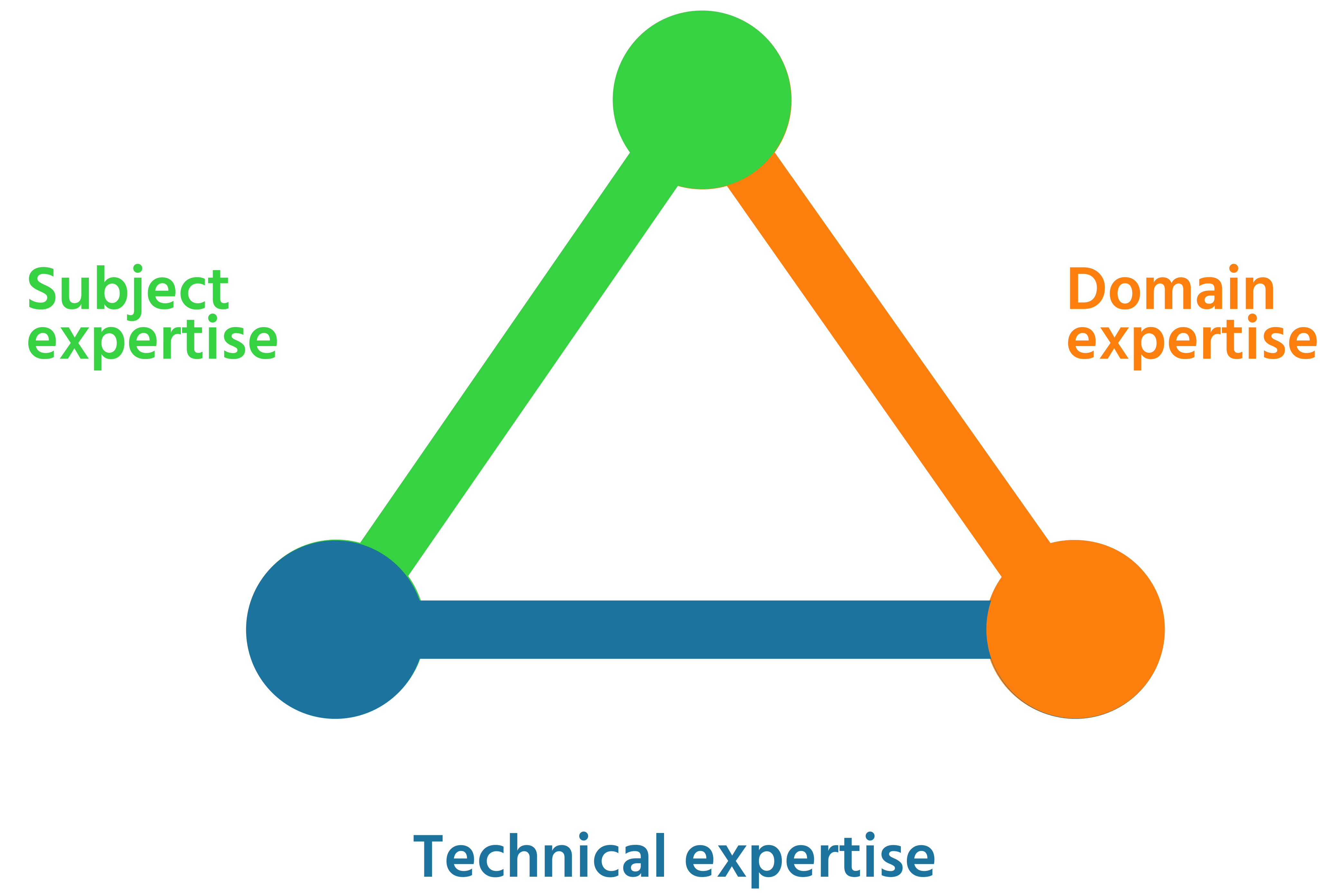

Homogenous targets

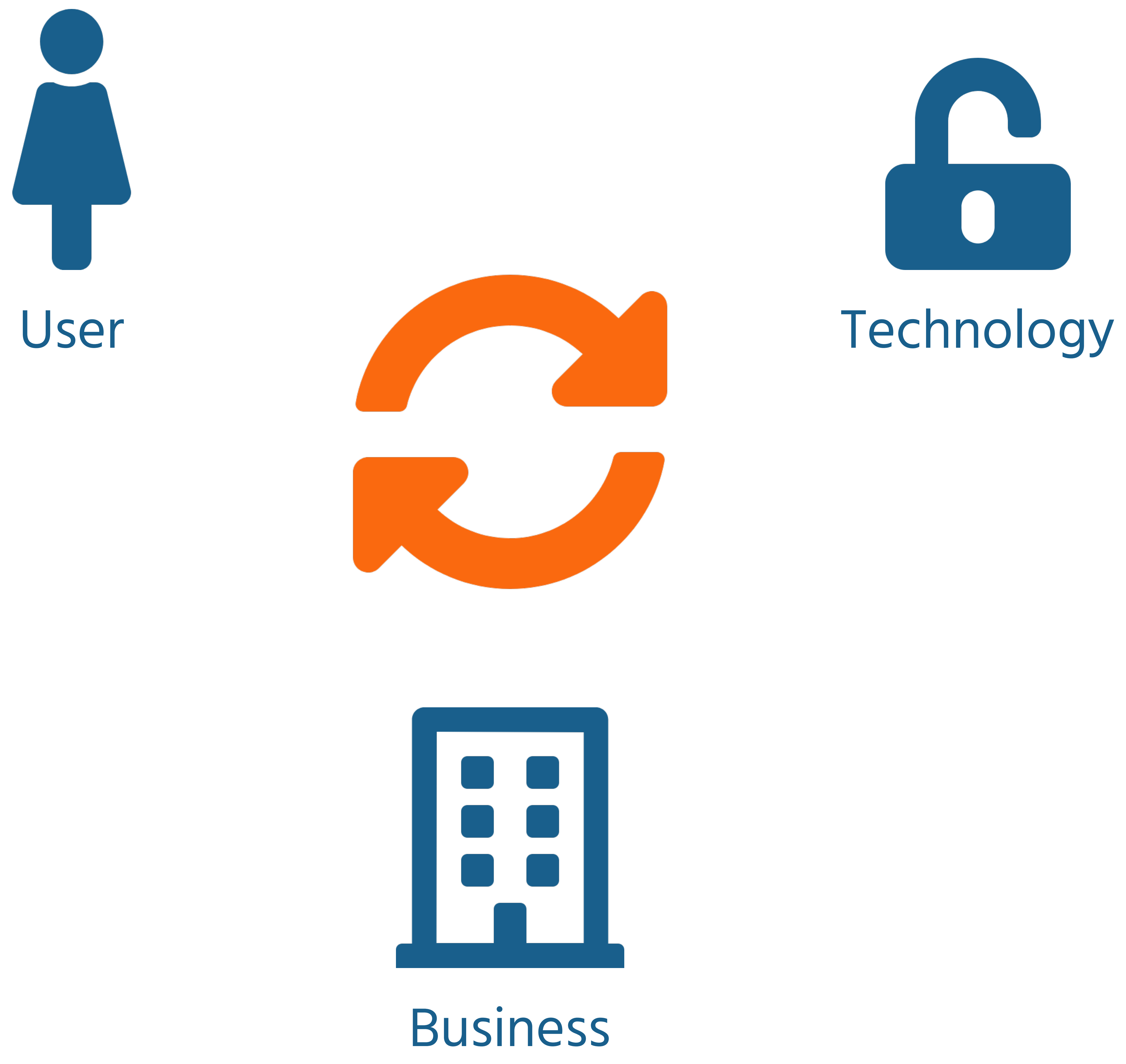

Heterogenous targets

Does it work?

Yes!

Know-how

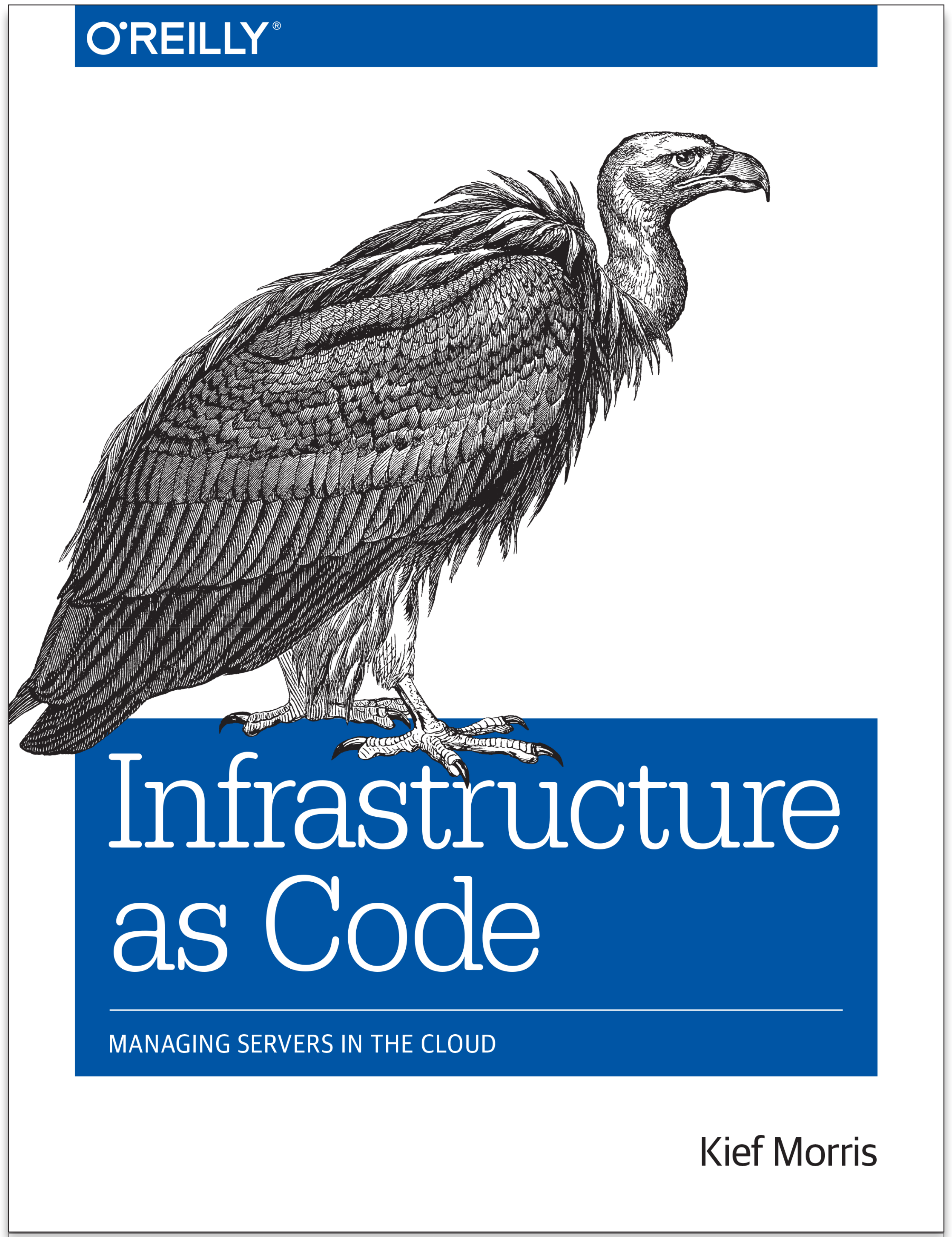




Technical Expertise
Safespring core competency

Closing words

1. Safespring builds its products on open source
2. Safespring has moved from central CM solution to a distributed image based solution
3. Safespring offers Private Cloud solution for best practices solution in-house

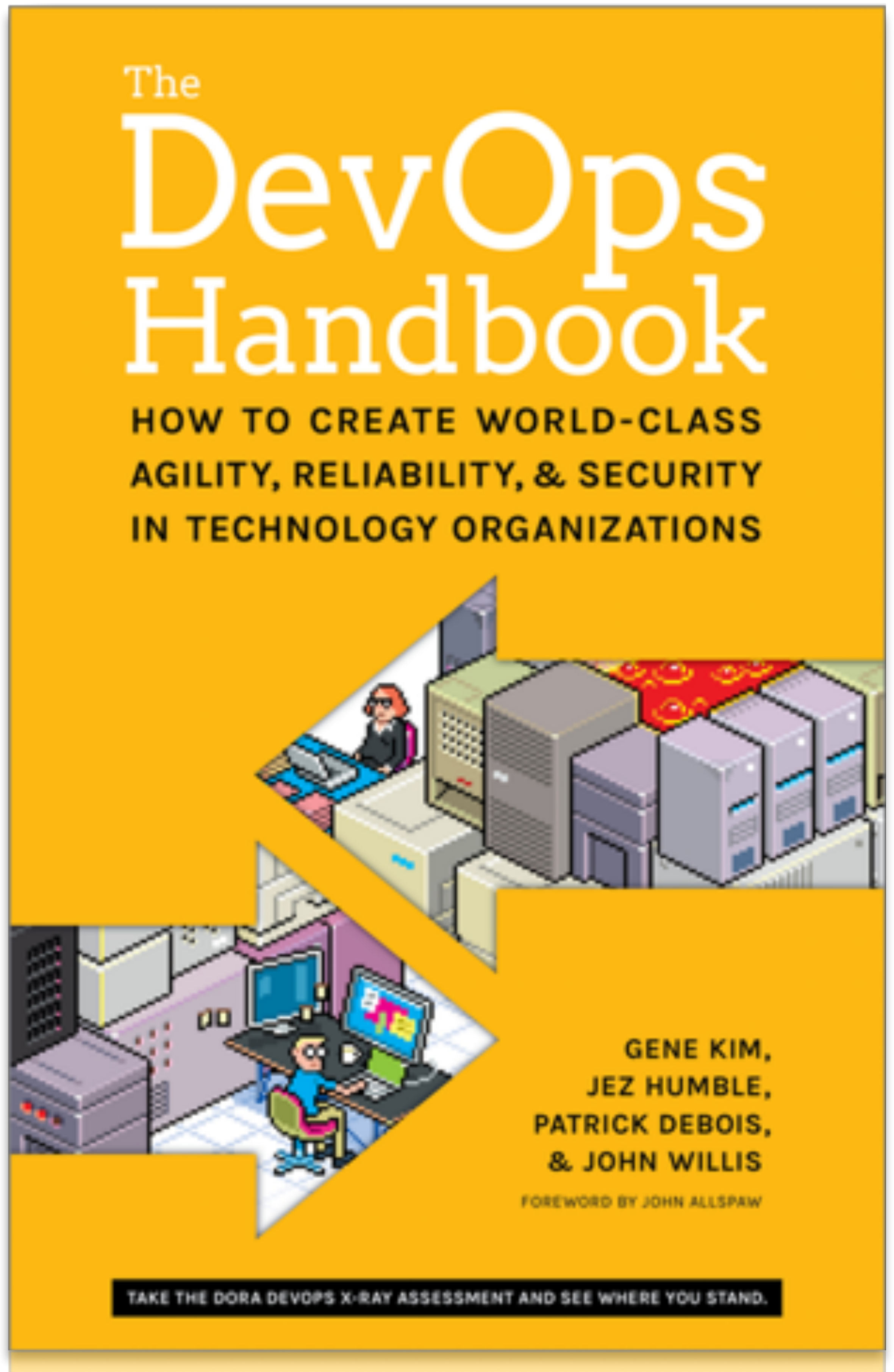


O'REILLY®

Infrastructure as Code

MANAGING SERVERS IN THE CLOUD

Kief Morris



The DevOps Handbook

HOW TO CREATE WORLD-CLASS AGILITY, RELIABILITY, & SECURITY IN TECHNOLOGY ORGANIZATIONS

GENE KIM, JEZ HUMBLE, PATRICK DEBOIS, & JOHN WILLIS

FOREWORD BY JOHN ALLSPAW

TAKE THE DORA DEVOPS X-RAY ASSESSMENT AND SEE WHERE YOU STAND.

QA

@bruvik

2019-04-04

www.safespring.com