



Break the Toolchain: Rewiring Application Continuous Delivery

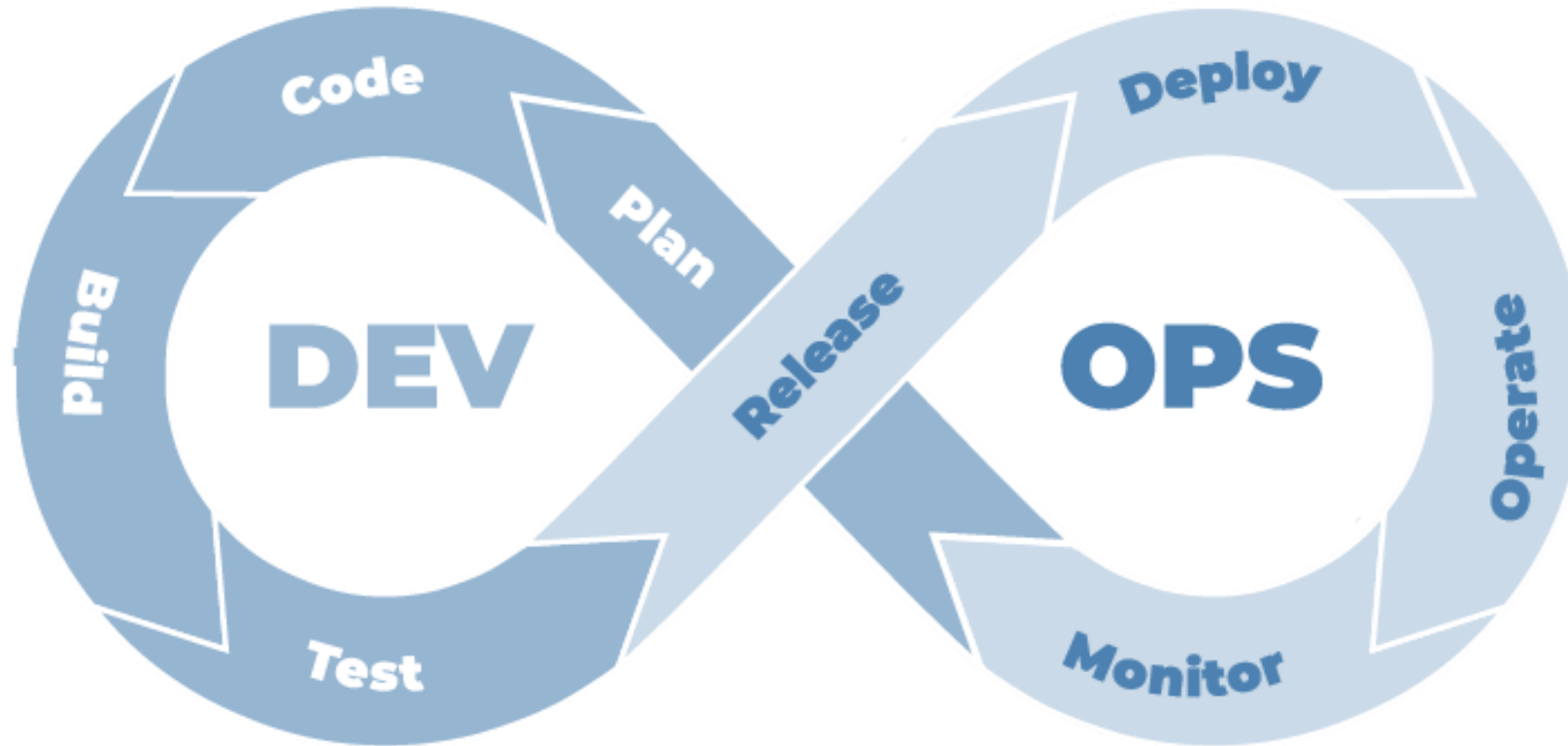
Intro

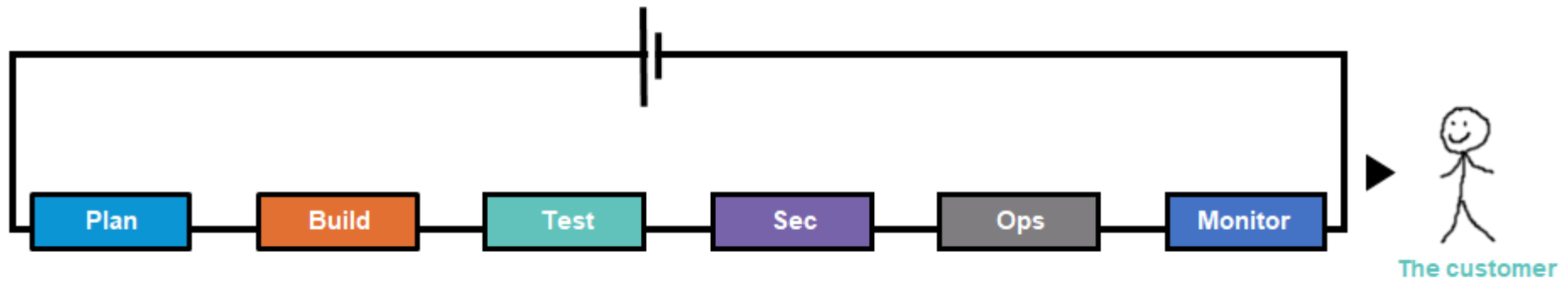
Maya Ber Lerner, CTO
www.linkedin.com/in/mayaber



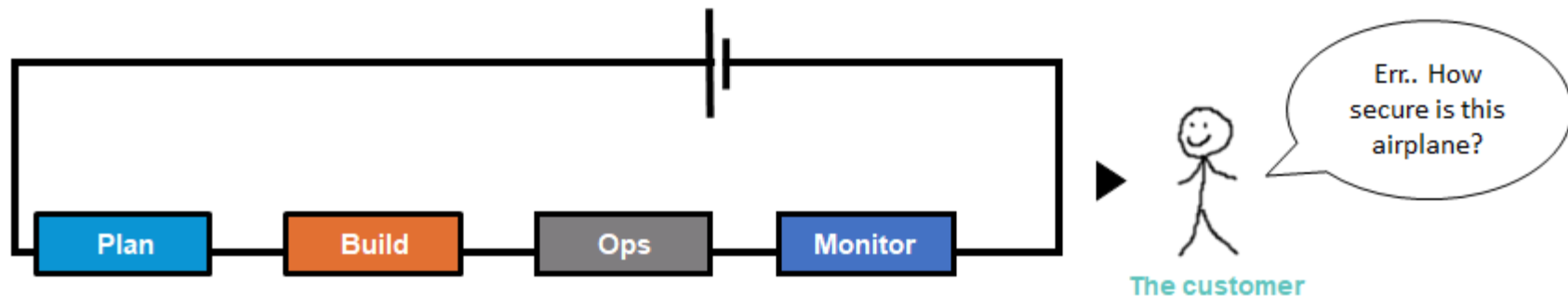
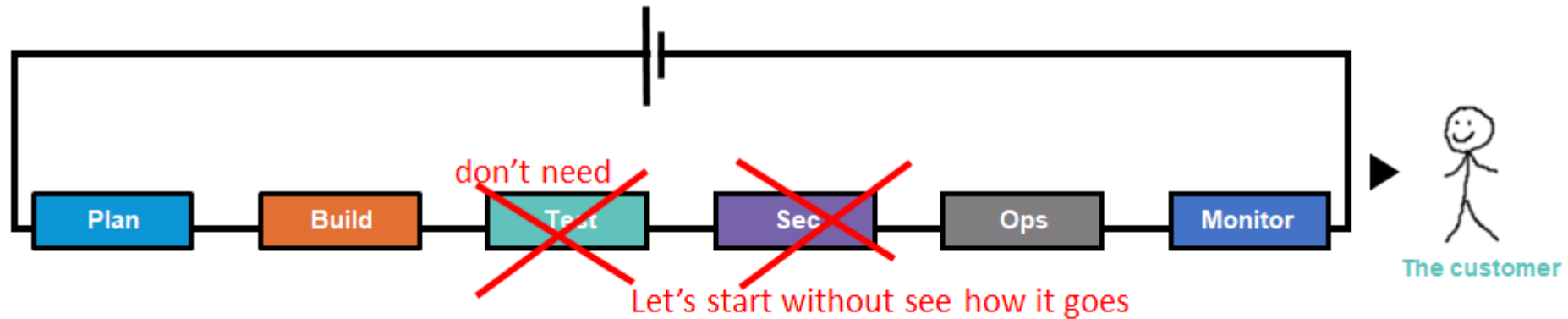
Electric circuits are a good model
for the study of systems

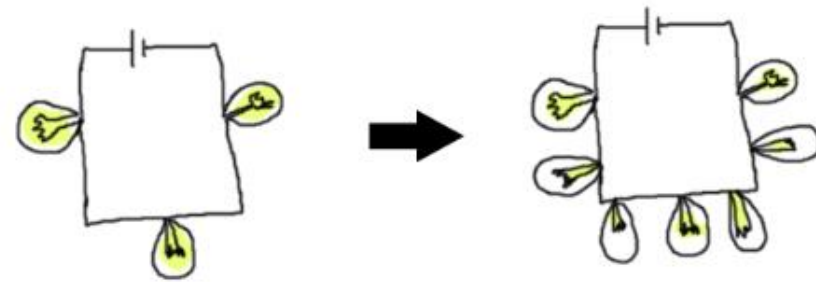
How do you model the DevOps toolchain?

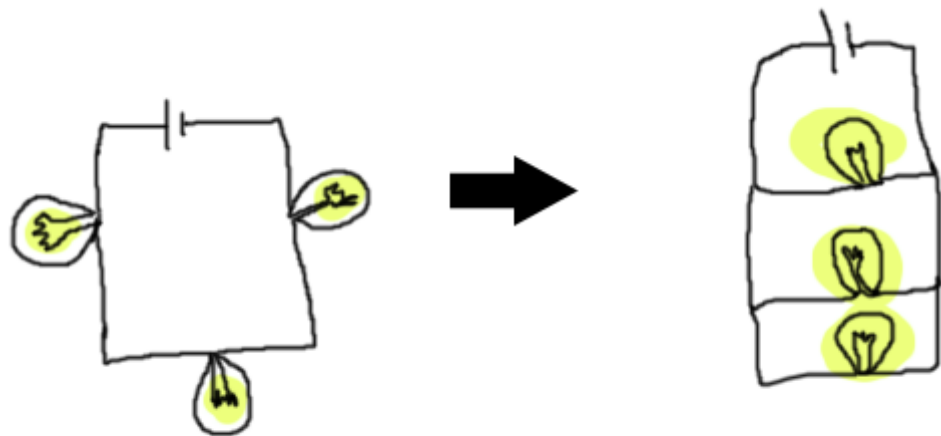


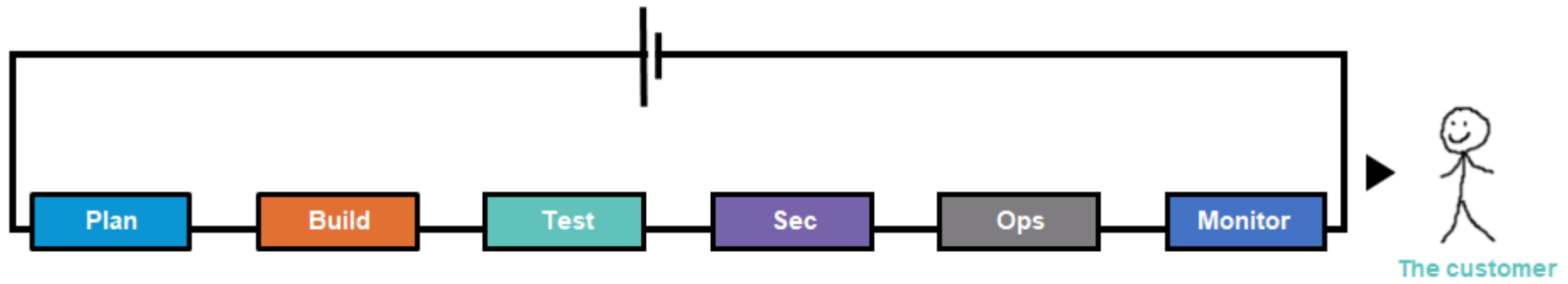


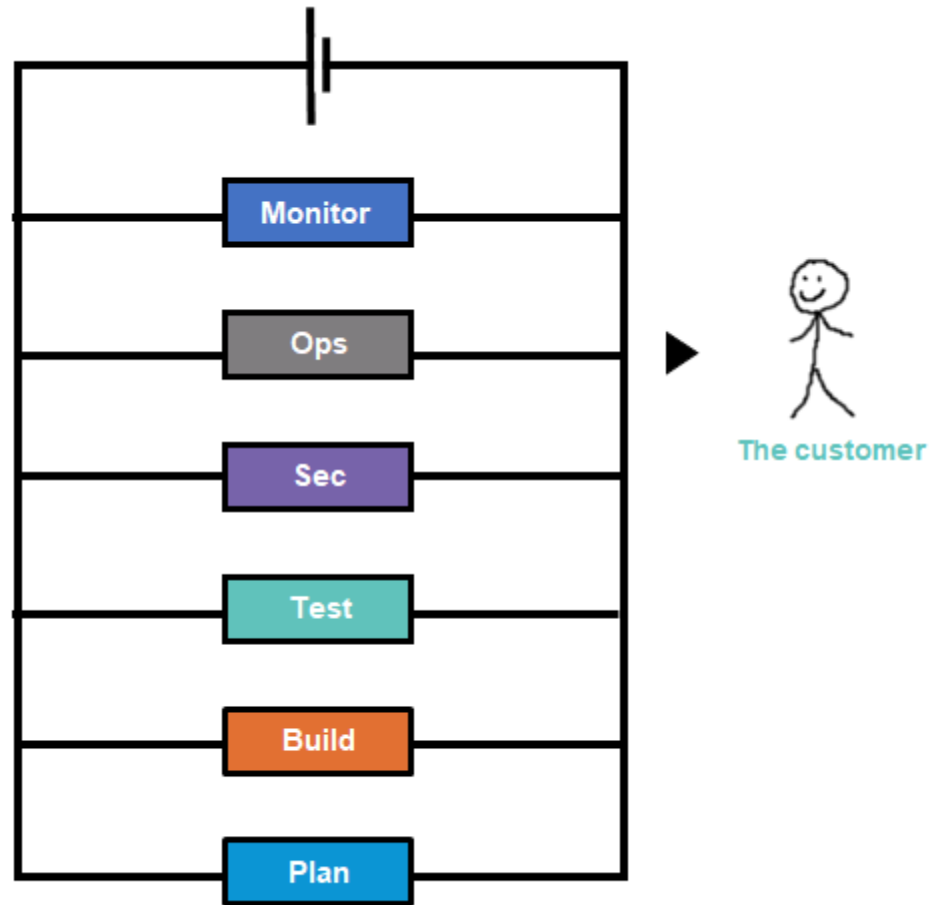
The Series Circuit Works. But it's **Hard to Scale**.



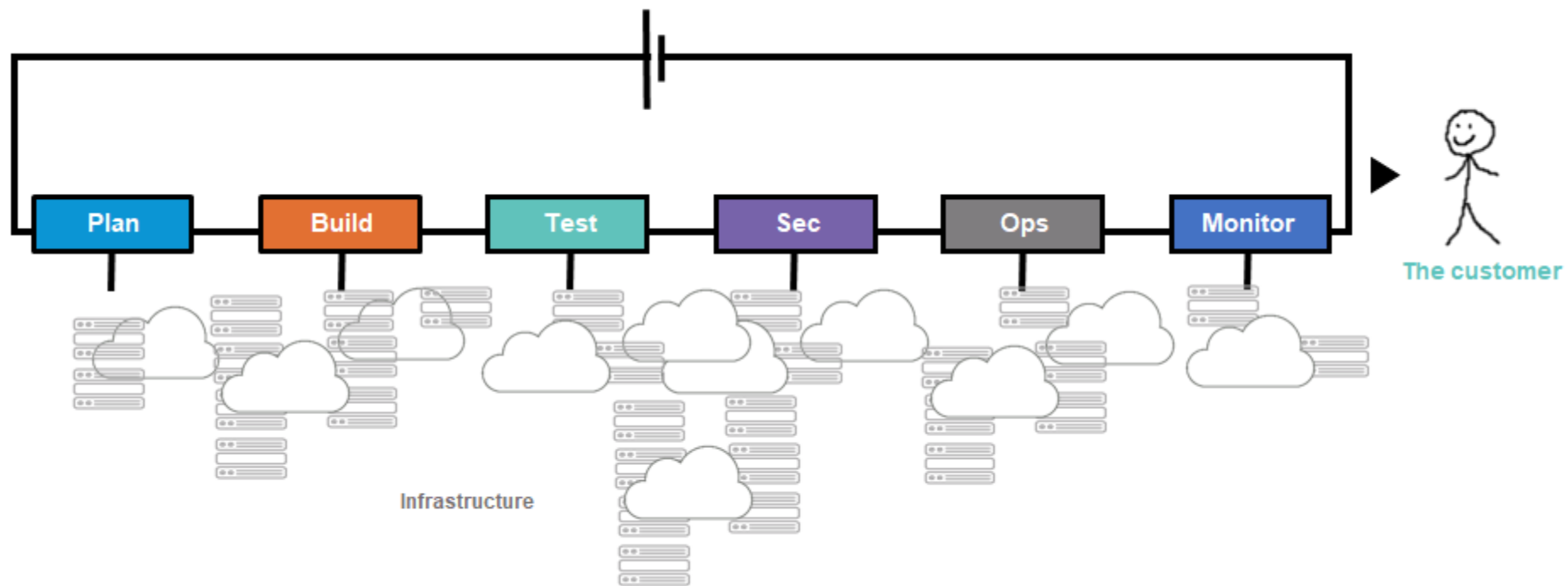


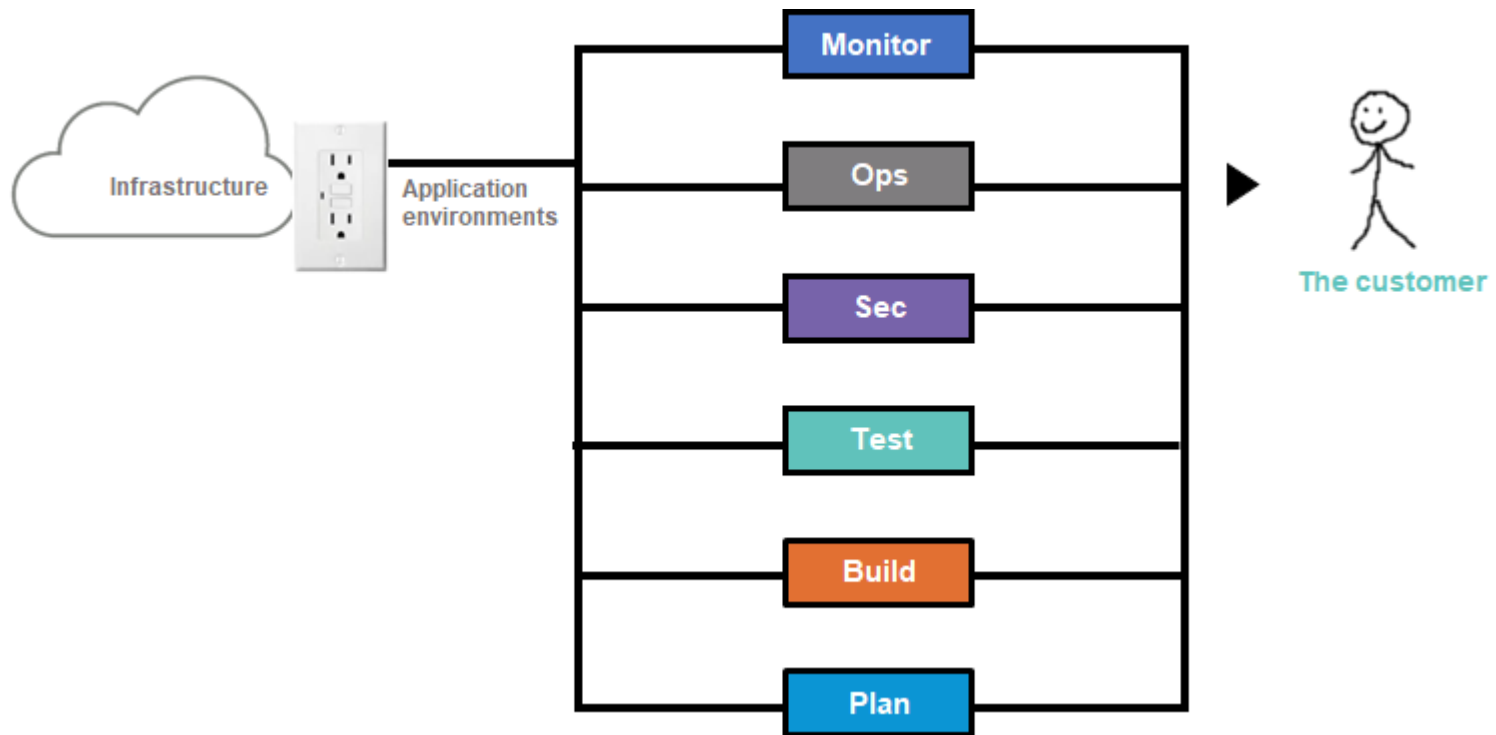






How can the application be in
six different places at the same time?





In order to make this happen, we need:



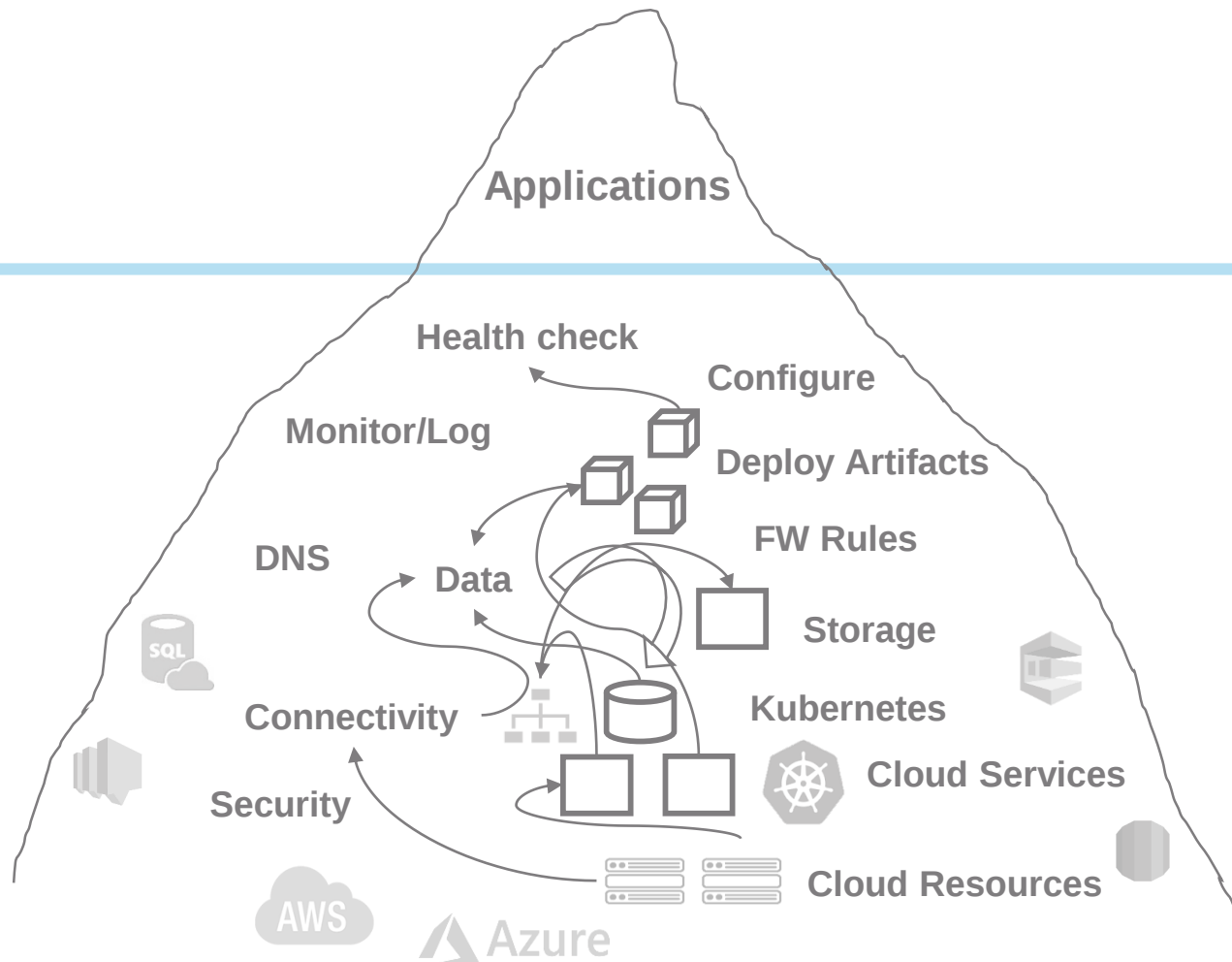
A good way to **define** (blueprint) our application environment, as code. Not just applications. Not just infrastructure. The entire environment.



A good way to **serve** these application environments to the teams involved in value stream tasks—development, functional testing, performance testing, security, production.

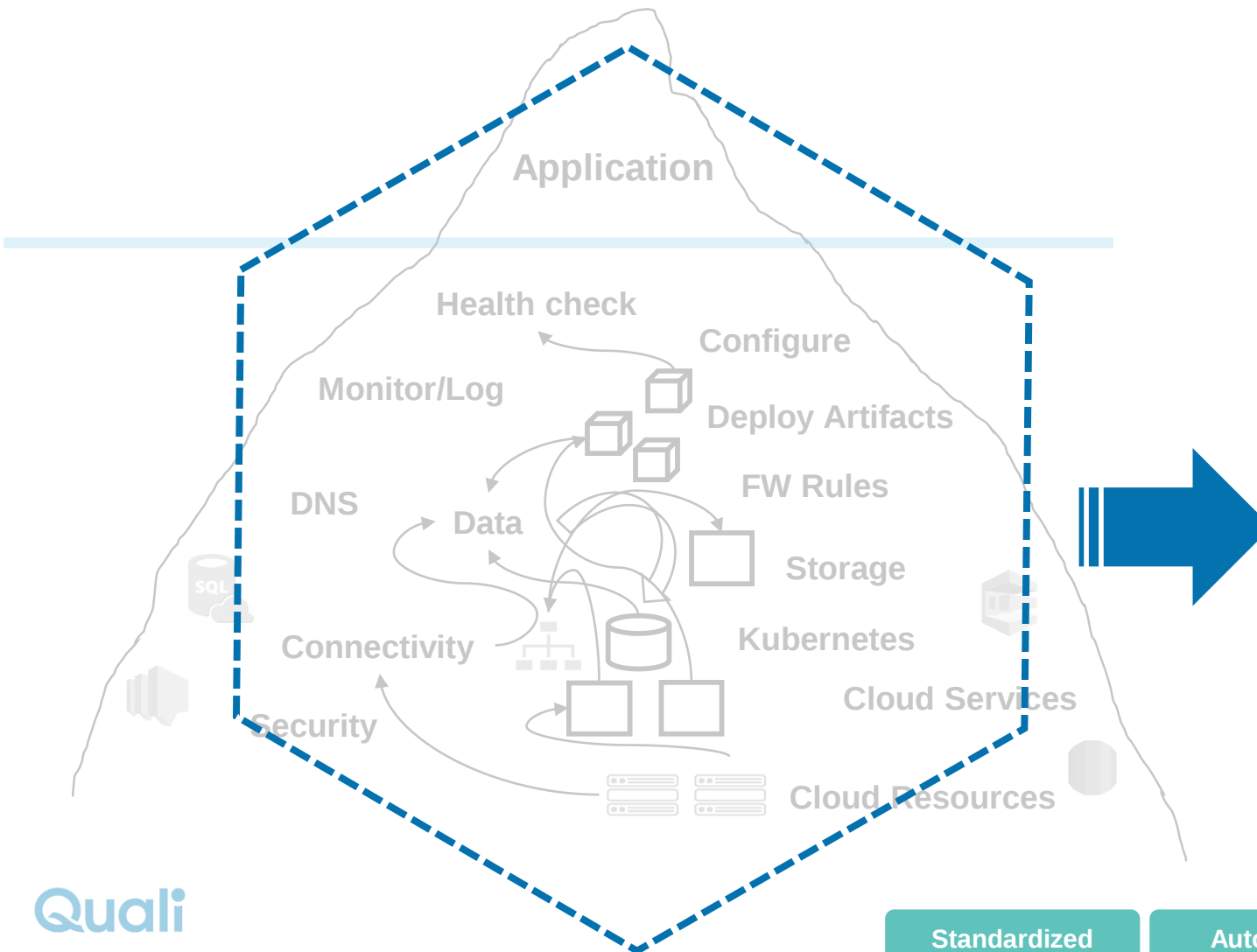
The Environment as the **Business Unit**

The business need



The Environment as the Business Unit

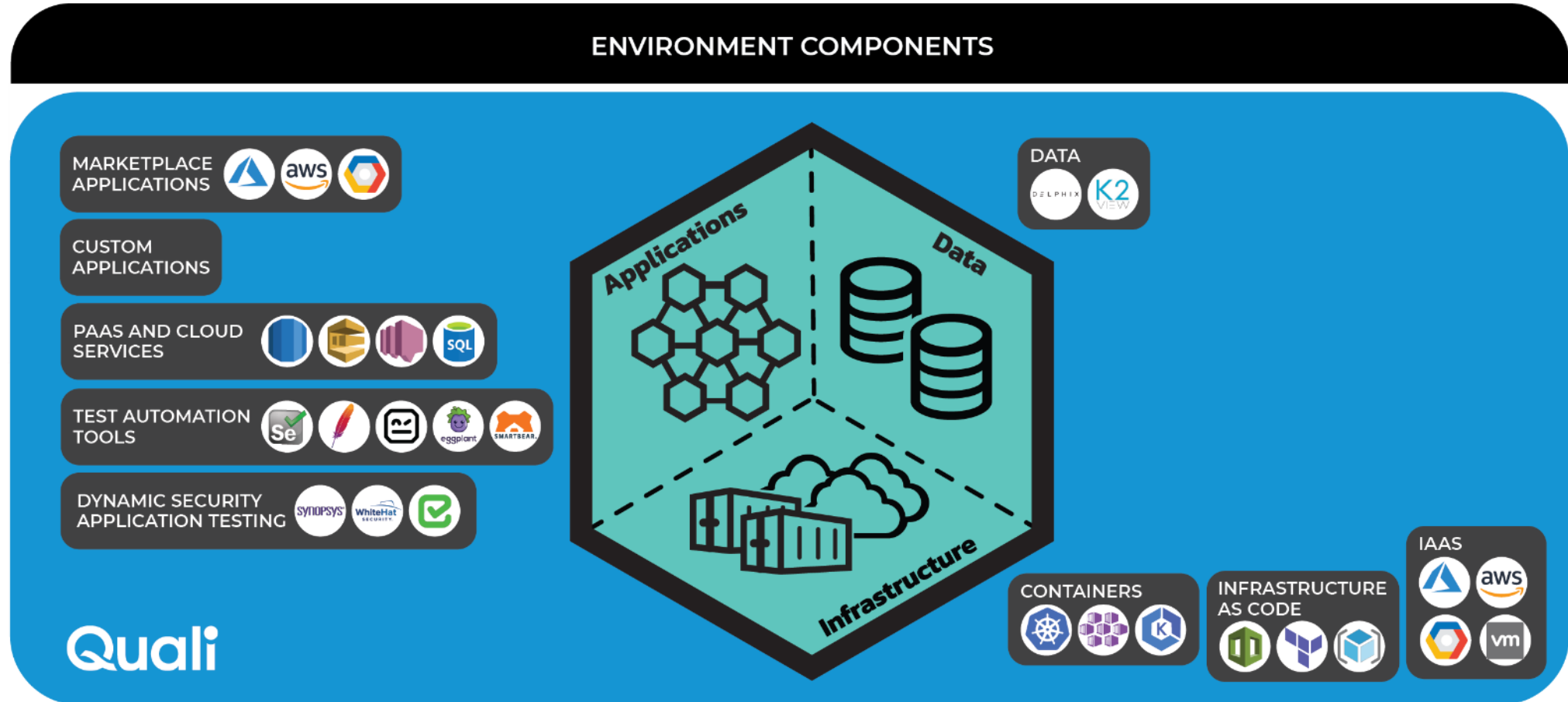
Blueprints enable repeatable environments



Creating **blueprints** of the infrastructure, applications, and data components allows **repeatable** orchestration of environments, serving multiple teams

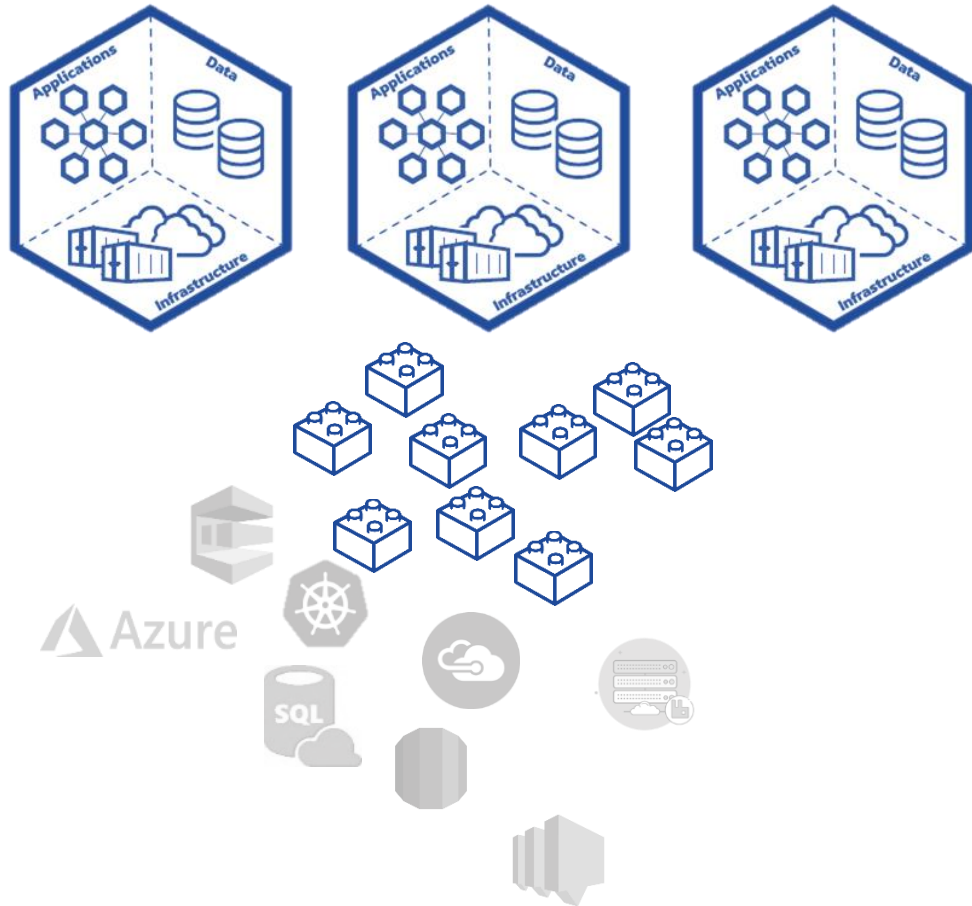


The Environment as the **Business Unit**



The Environment as the **Business Unit**

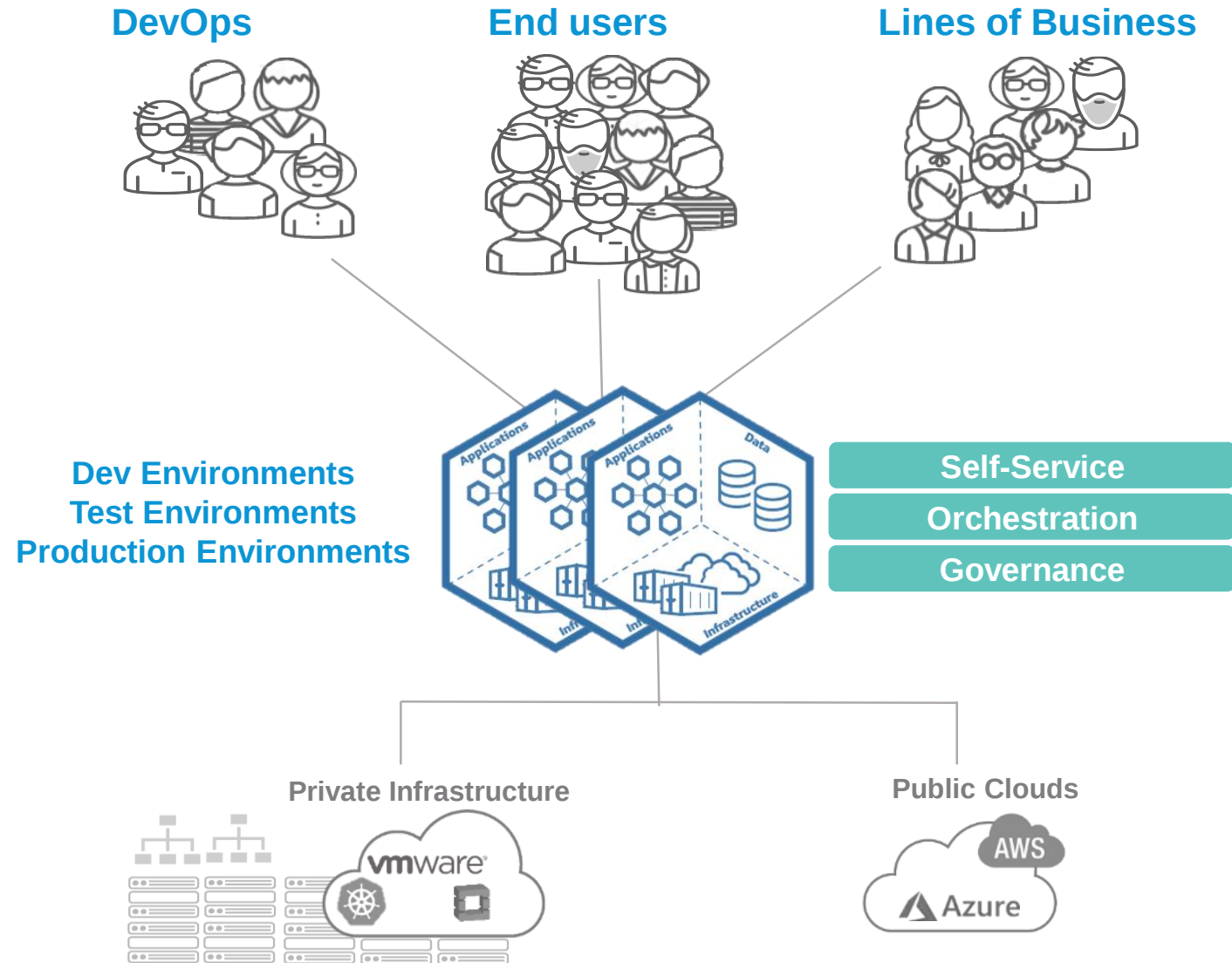
Application Centric Building Blocks



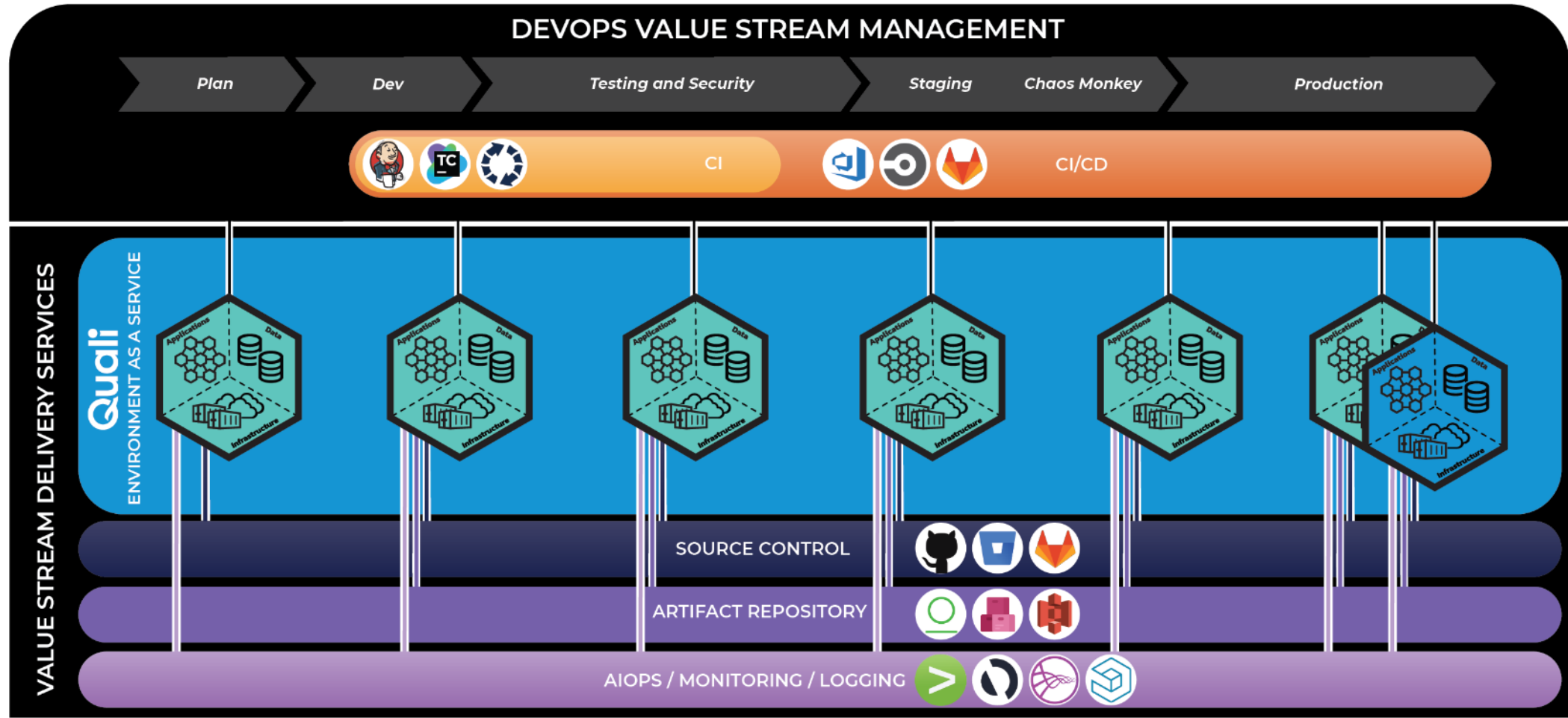
When environment blueprints are **modular**, it is possible to reuse building blocks for creating standardized blueprints faster.

It makes it easier to expand the service to **new technologies** and use automation to capture knowledge, without everyone having to acquire specialized skills in infrastructure and coding.

Self-Service Environments



Self-Service



Takeaways

- Blueprint **entire environments**
- **Self-service**
- **Don't wait** for K8S to save you

Q&A

Thank You

