From Infrastructure as Code to Environment as Code

Challenges scaling IaC and how to resolve them

Adarsh Shah

Engineering Leader, Coach, Public Speaker

Founder & CEO at CompuZest
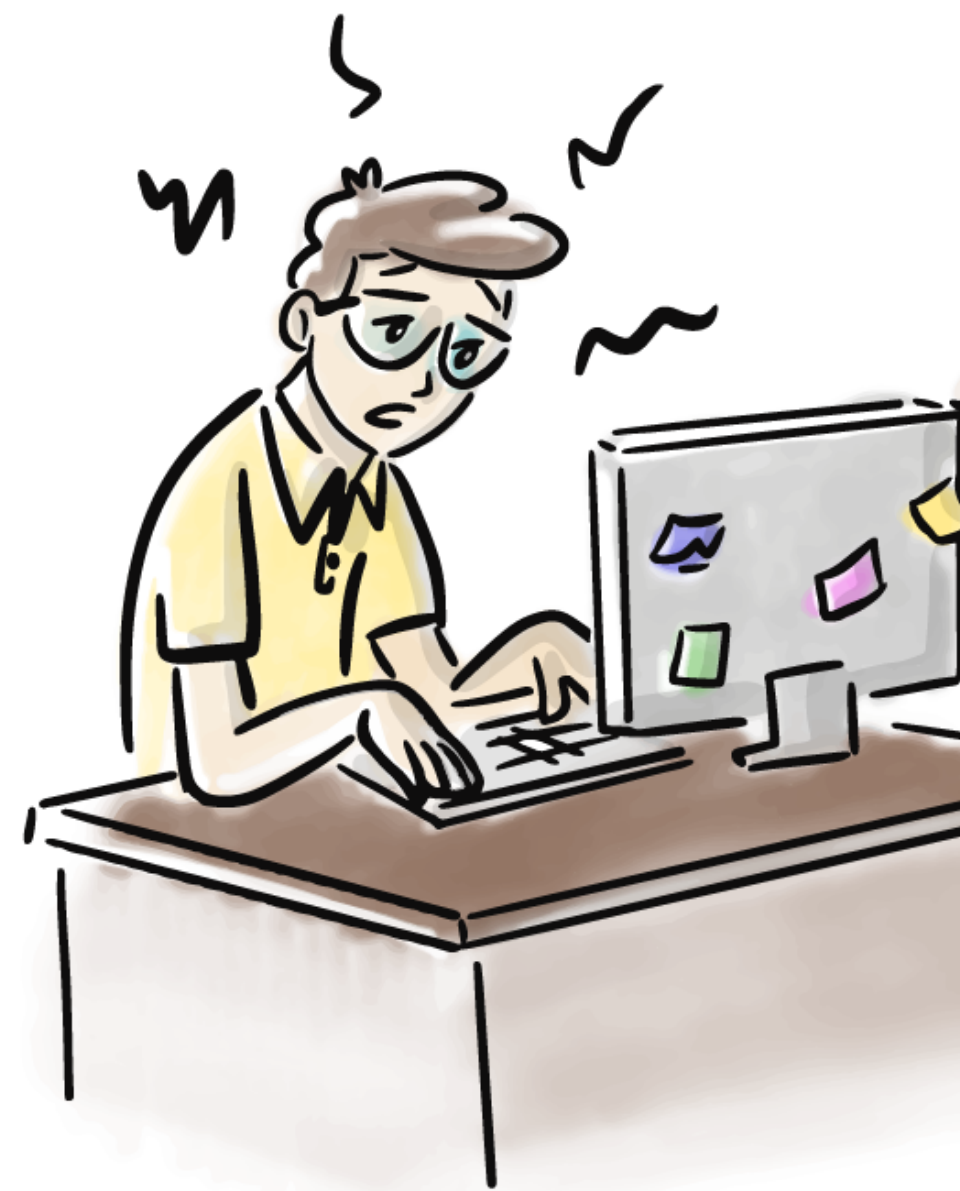
🐦 *@shahadarsh*

🌐 *https://compuzest.com*

Infrastructure as Code

Infrastructure as Code (IaC) is an approach that takes proven coding techniques used by software systems and extends it to infrastructure. It is one of the key DevOps practices that **enable teams to deliver infrastructure, and thereby software running on it, rapidly and reliably**.
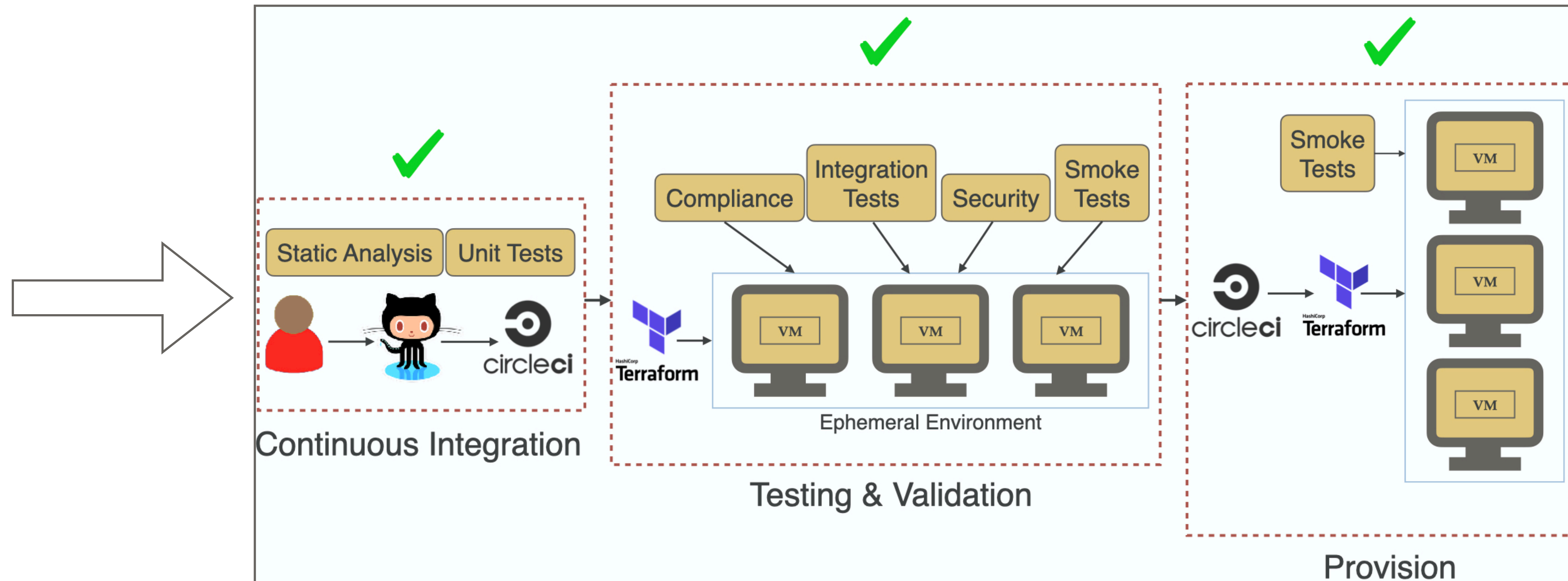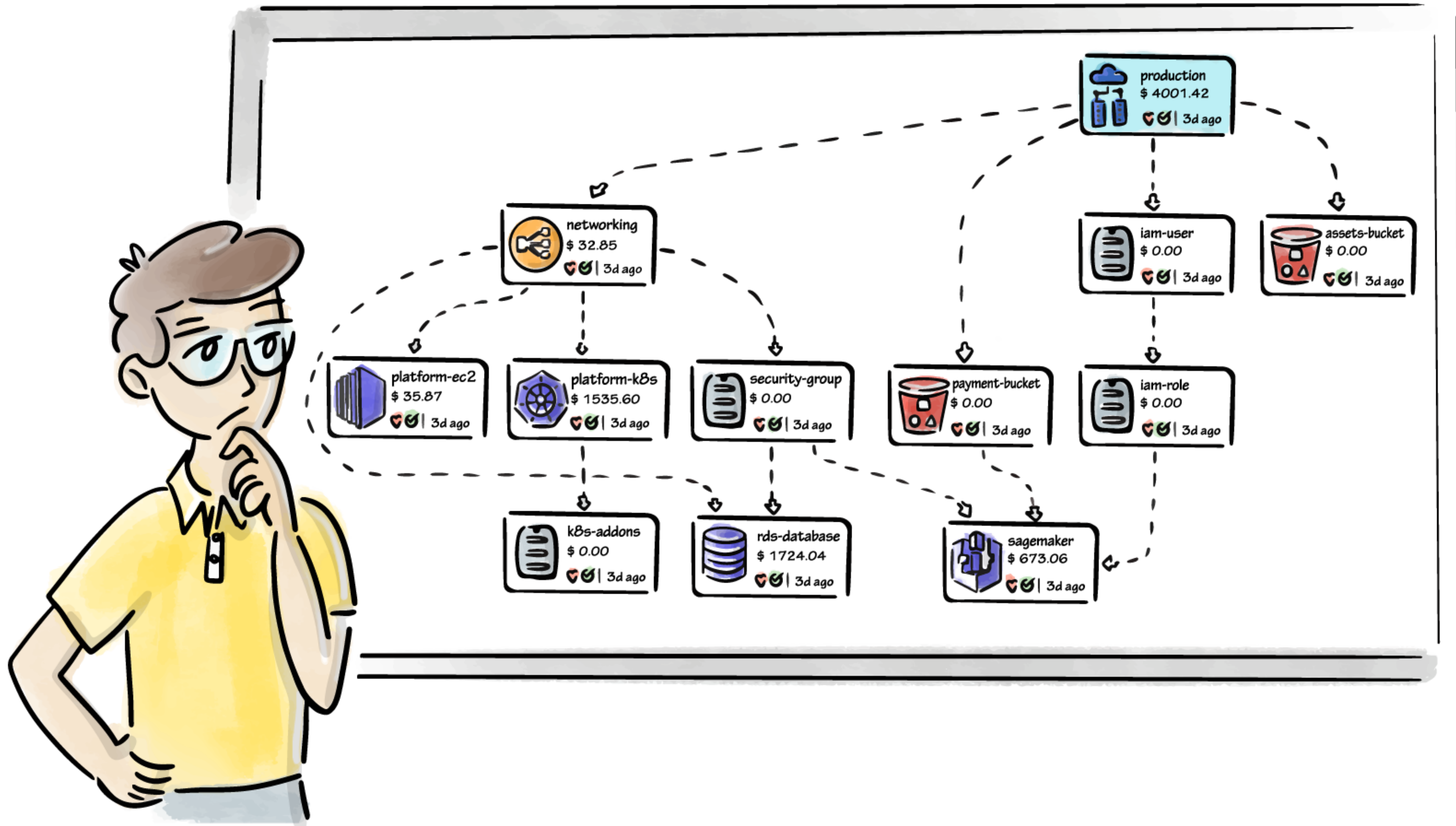
Evolution of IaC Setup

Monolith IaC
with a Single Run

Breaking into Separate & Smaller IaC Runs
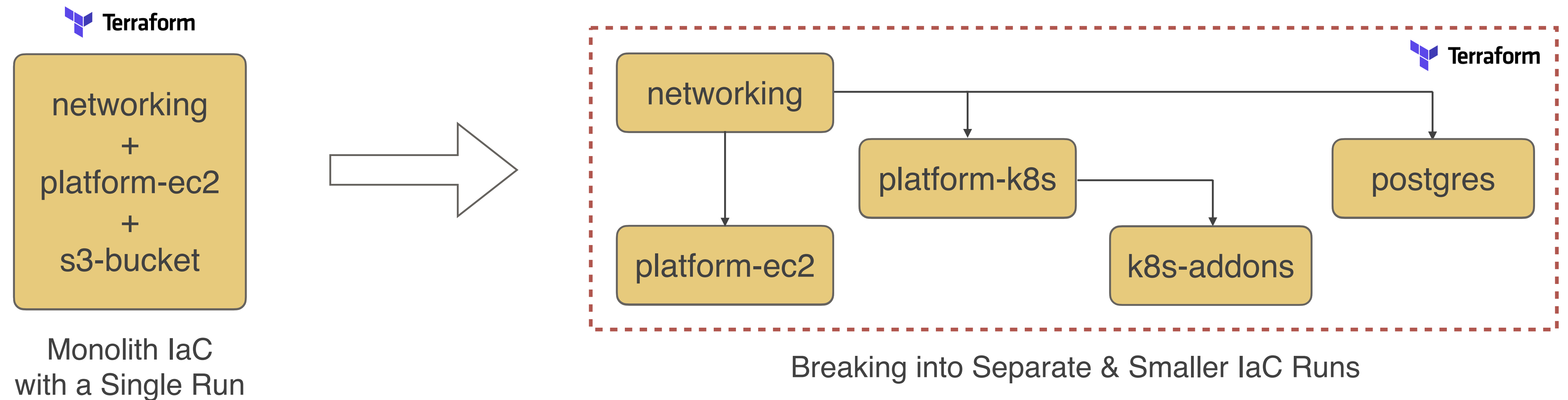
Evolution of IaC Execution

From Engineers machine

IaC Pipeline/GitOps execution from a Shared Environment

Static Analysis | Unit Tests
Continuous Integration

Compliance | Integration Tests | Security | Smoke Tests
Ephemeral Environment
Testing & Validation

Smoke Tests
Provision

Environment

https://bit.ly/iac-to-eac   CompuZest   @shahadarsh

Options for Entire Environment Provisioning

- **Option 1**
  - Create a Monolith IaC
- **Option 2**
  - Hand-roll Pipelines
  - Manage Complex Dependencies

Other Challenges scaling IaC

Replicating Environments
is a pain

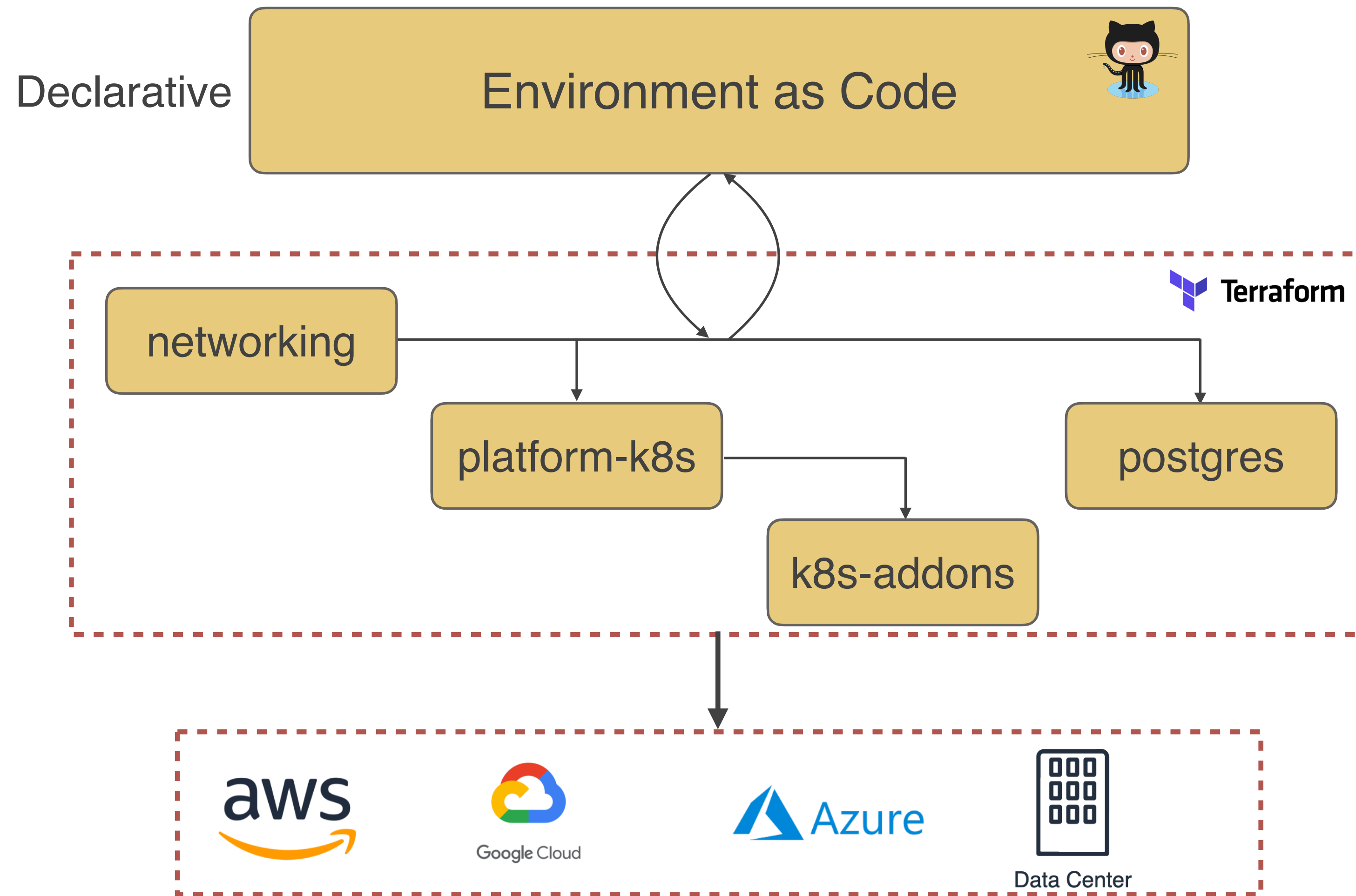Not easy to visualize/understand
Environments

Drift Detection

What is Environment as Code?
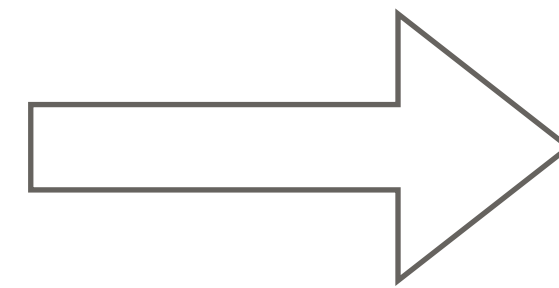
CompuZest

Environment as Code

IaC vs EaC

Infrastructure as Code

↓

*Automates various Lego Pieces
(i.e. Infrastructure Resources)*
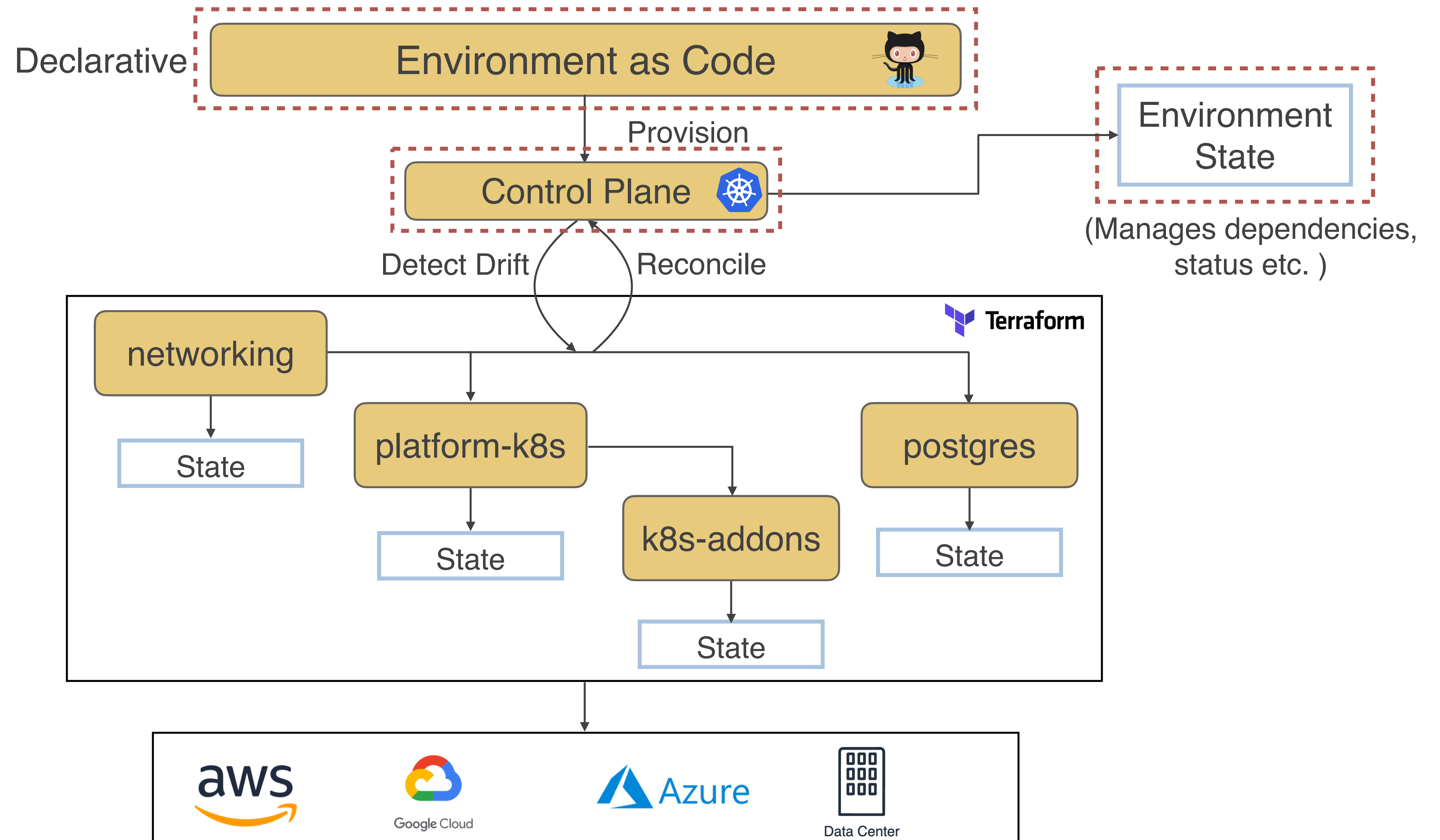

Environment as Code

↓

*Automates how those Lego Pieces
are connected to make up a Lego Toy
(i.e. Entire Environment)*

Environment as Code

***Environment as Code (EaC)** is an abstraction over Infrastructure as Code that provides a **declarative** way of defining an entire Environment. It has a Control Plane that **manages the state of the environment**, including relationships between various resources, Detects Drift as well enables Reconciliation. It also **supports best practices** like Loose Coupling, Idempotency, Immutability, etc. for the entire environment. EaC allows teams to deliver entire environments rapidly and reliably, at scale.*

Provision Environment

Declarative — **Environment as Code**

Provision →

Control Plane → **Environment State**

(Manages dependencies, status etc.)

Detect Drift / Reconcile

Terraform

networking → State

platform-k8s → State

k8s-addons → State

postgres → State

aws · Google Cloud · Azure · Data Center
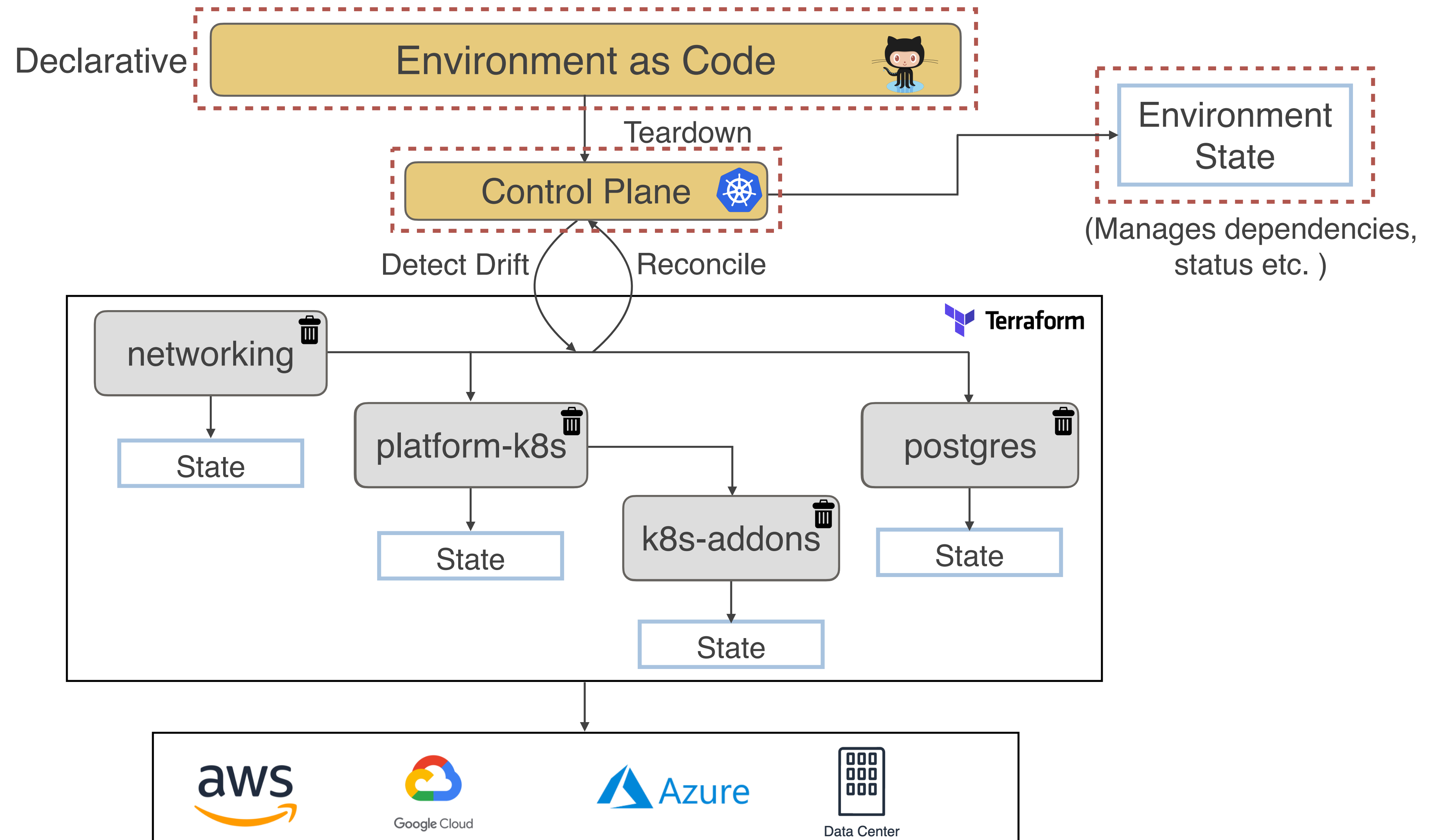
Teardown Environment
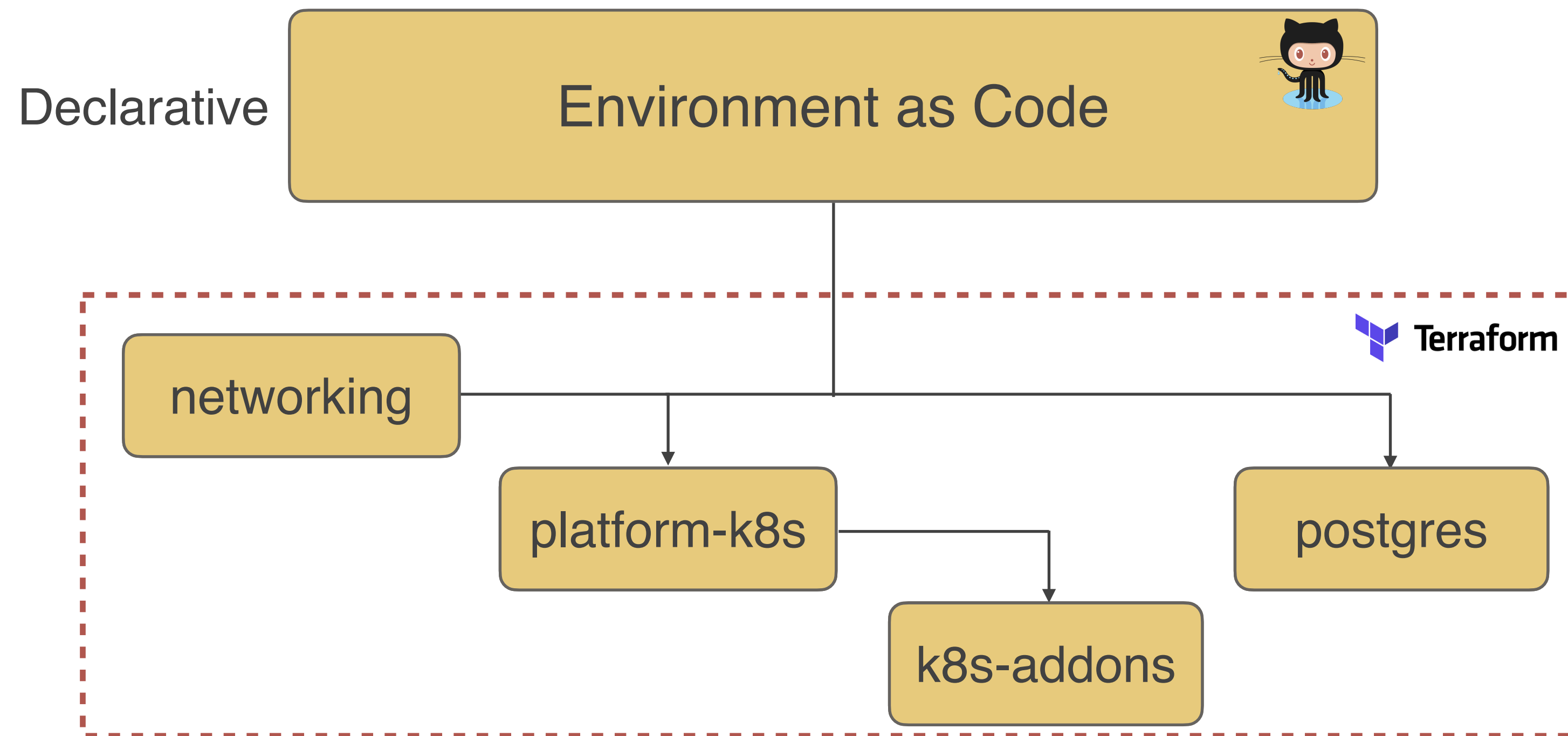
Attributes of Environment as Code

Ability to define Entire Environment

Example Environment as Code

```
1    apiVersion: stable.compuzest.com/v1
2    kind: Environment
3  > metadata: …
6    spec:
7      teamName: payment
8      envName: demo
9      teardown: true
10     autoApprove: true
11     components:
12 >     - name: static-assets …
20 >     - name: payment-receipt …
28 >     - name: networking …
41 >     - name: platform-ec2 …
53      - name: platform-eks
54        type: terraform
55        dependsOn: [networking]
56        module:
57          source: aws
58          name: eks
59        variables:
60          - name: vpc_id
61            valueFrom: networking.vpc_id
62          - name: subnets
63            valueFrom: networking.public_subnets
64        variablesFile:
65          path: "demo/vars/platform-eks.tfvars"
66 >     - name: eks-addons …
75 >     - name: sg-payment-rds …
88 >     - name: payment-rds …
105 >    - name: sagemaker-user …
122 >    - name: sagemaker-role …
137 >    - name: sagemaker …
```

Loosely Coupled

Declarative

Environment as Code

networking

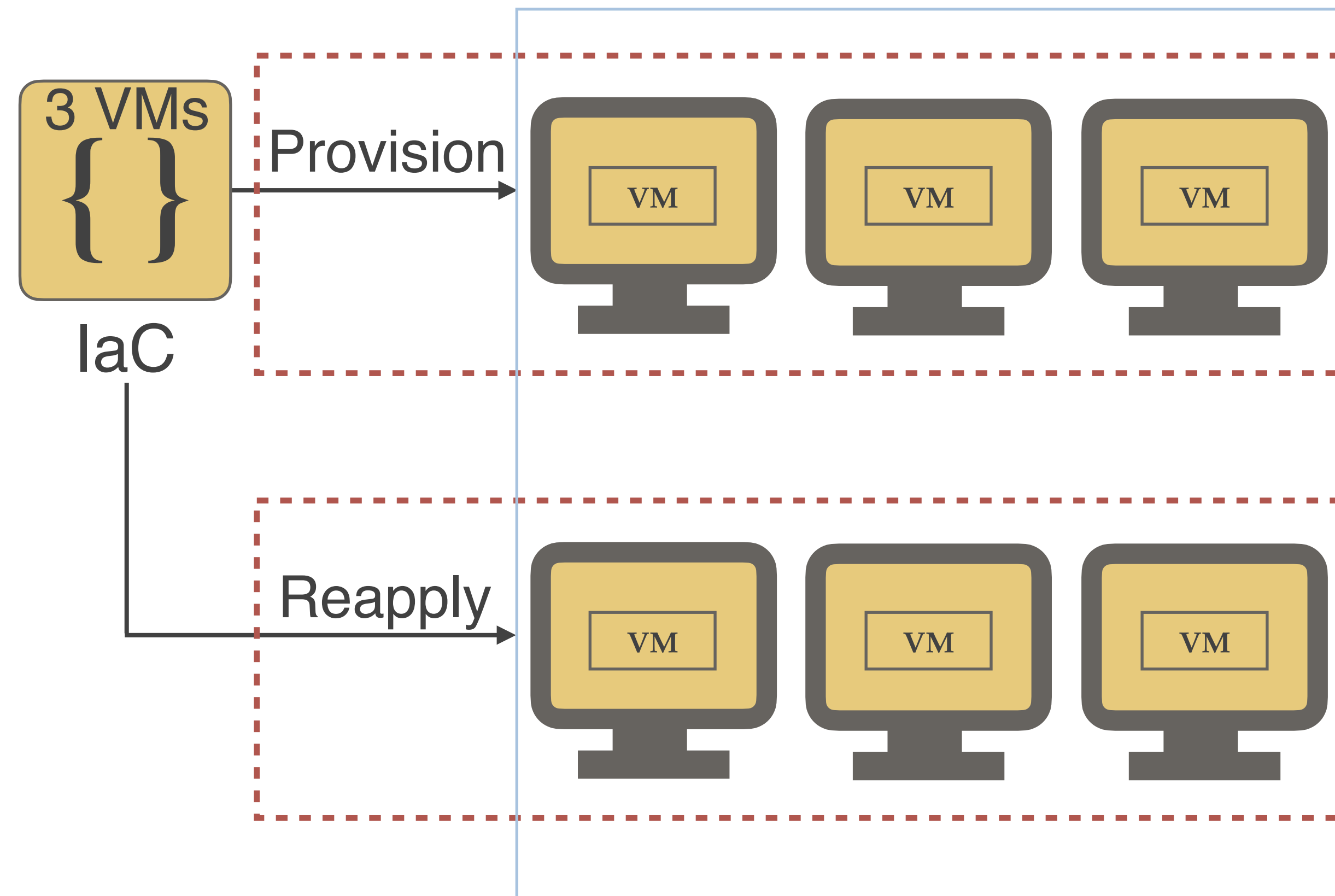**platform-k8s**

k8s-addons

postgres

Terraform

Manage State for the entire Environment

Example Environment State File

```
1   {
2     "version": 4,
3     "id": "ec98b888-f193-c6cc-9530-0b7409d6d229",
4     "kind": "Environment",
5     "spec": {
6       "operation": "provision",
7       "status": "success",
8       "teamName": "payment",
9       "envName": "demo",
10      "components": [
11  >       {⋯
22      },
23      {
24        "name": "platform-eks",
25        "type": "terraform",
26        "operation": "provision",
27        "status": "success",
28        "dependsOn": [
29          "networking"
30        ],
31        "module": {
32          "source": "aws",
33          "name": "eks"
34        },
35  >     "variables": [⋯
44        ],
45  >     "variablesFile": {⋯
47        }
48      },
49  >     {⋯
70      }
71    ]
72    }
73  }
```

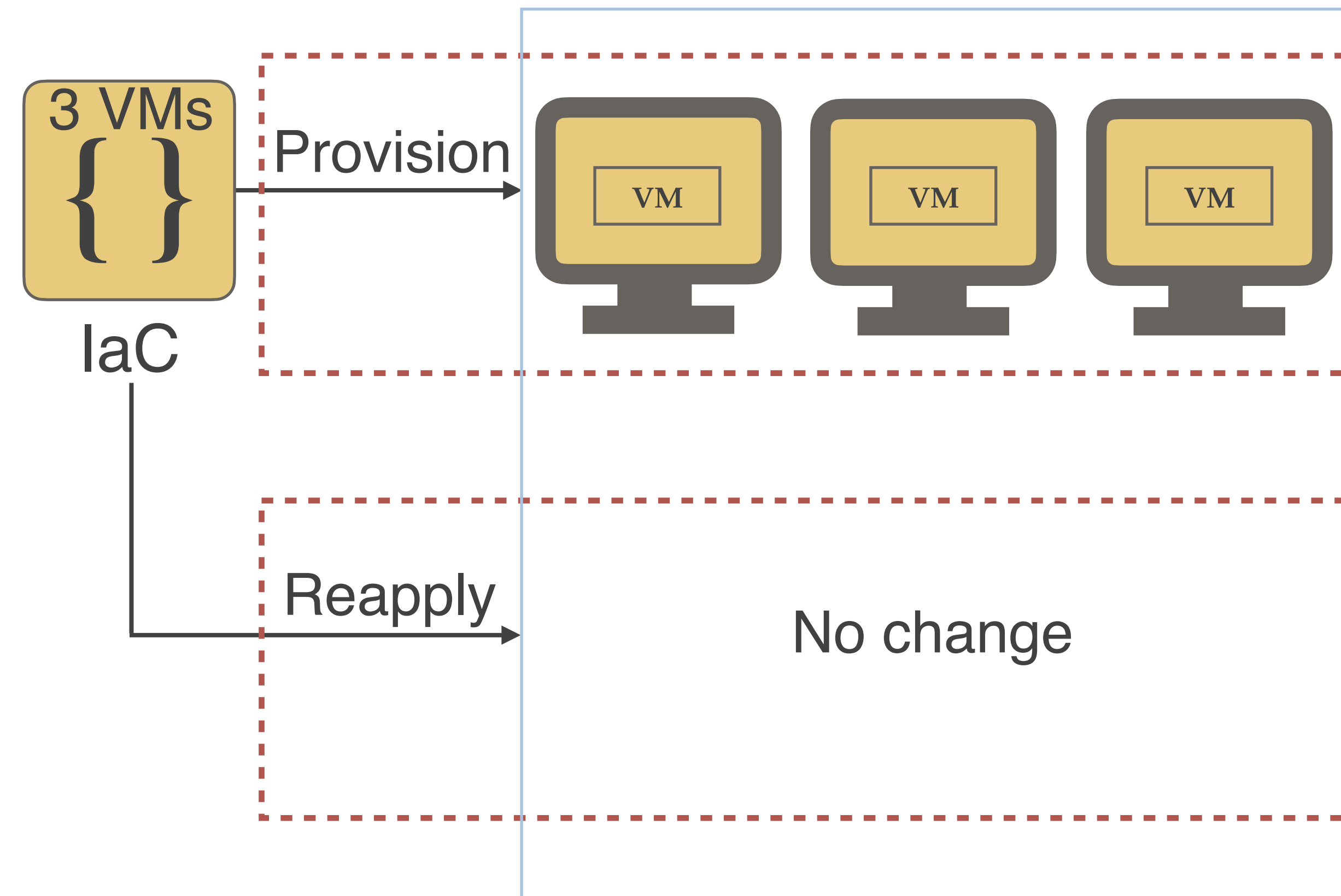CompuZest

Idempotent and Immutable for entire Environment

CompuZest

Non-Idempotent

3 VMs
{ }
IaC

Provision → VM VM VM

Reapply → VM VM VM

End State
Expected = 3
Actual = 6 ❌

Idempotent

3 VMs
{ }
IaC

Provision → VM VM VM

Reapply → No change

End State
Expected = 3
Actual = 3 ✅

https://bit.ly/iac-to-eac

CompuZest

@shahadarsh

Mutable Infrastructure

{}
IaC

Provision

{}
IaC

Change

Apply changes

v2 v2 v2

👤 User
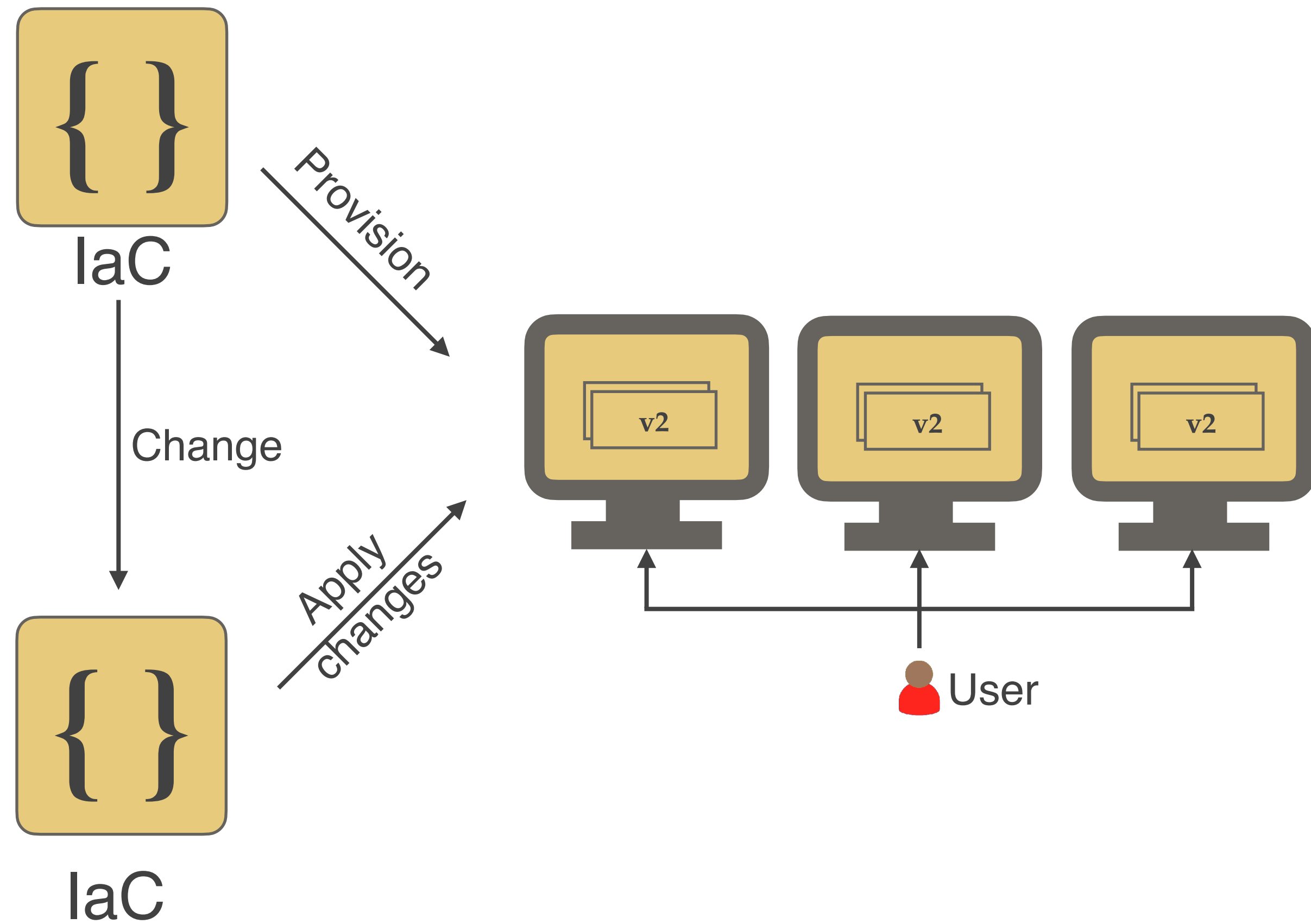
Deploys v2 to same Infrastructure

Immutable Infrastructure

{}
IaC

Provision

v1 v1 v1

👤 User

{}
IaC

Change

Provision

v2 v2 v2
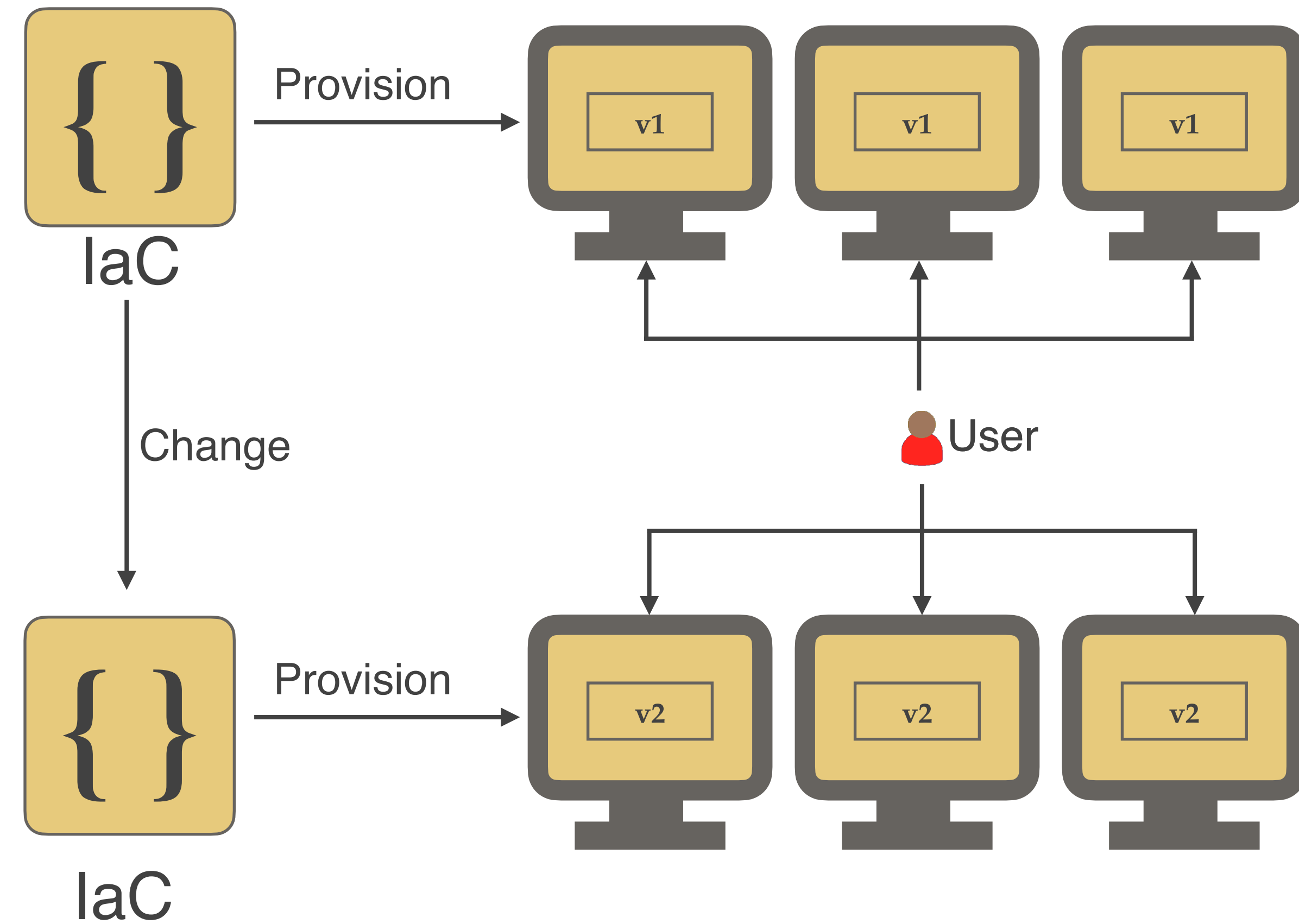
Provisions new Infrastructure with v2

https://bit.ly/iac-to-eac

CompuZest

🐦 @shahadarsh

Visualize and Understand Environments
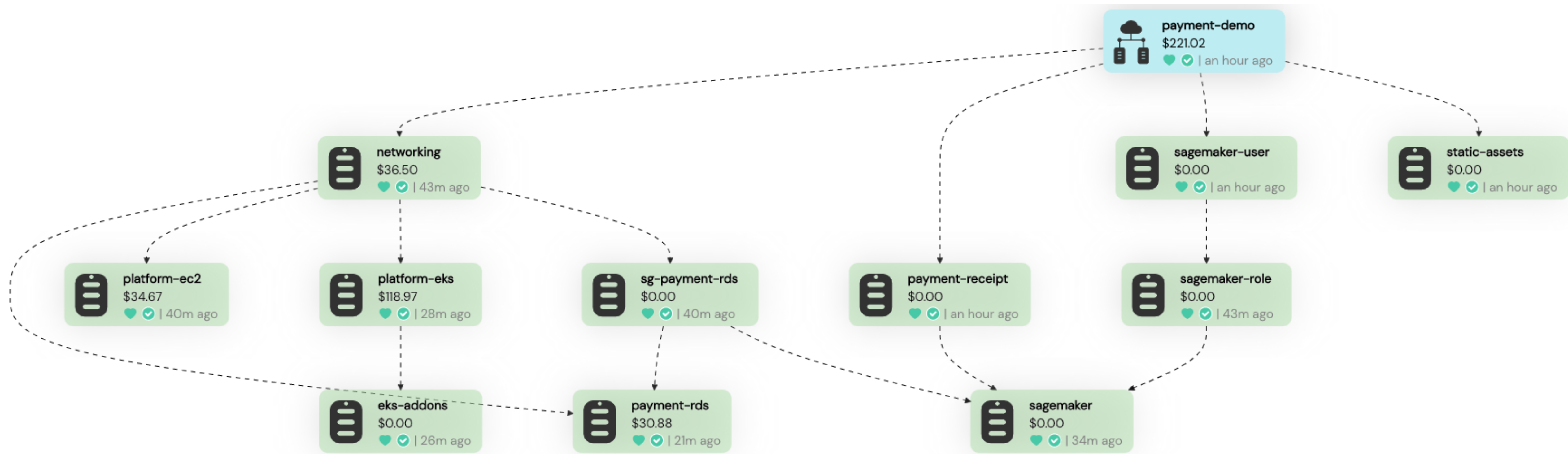
Drift Detection and Reconciliation

Observe/Detect Drift

Desired State

Control Loop

Current State

Reconcile
(Preferably with Approval step
that shows the plan)

CompuZest
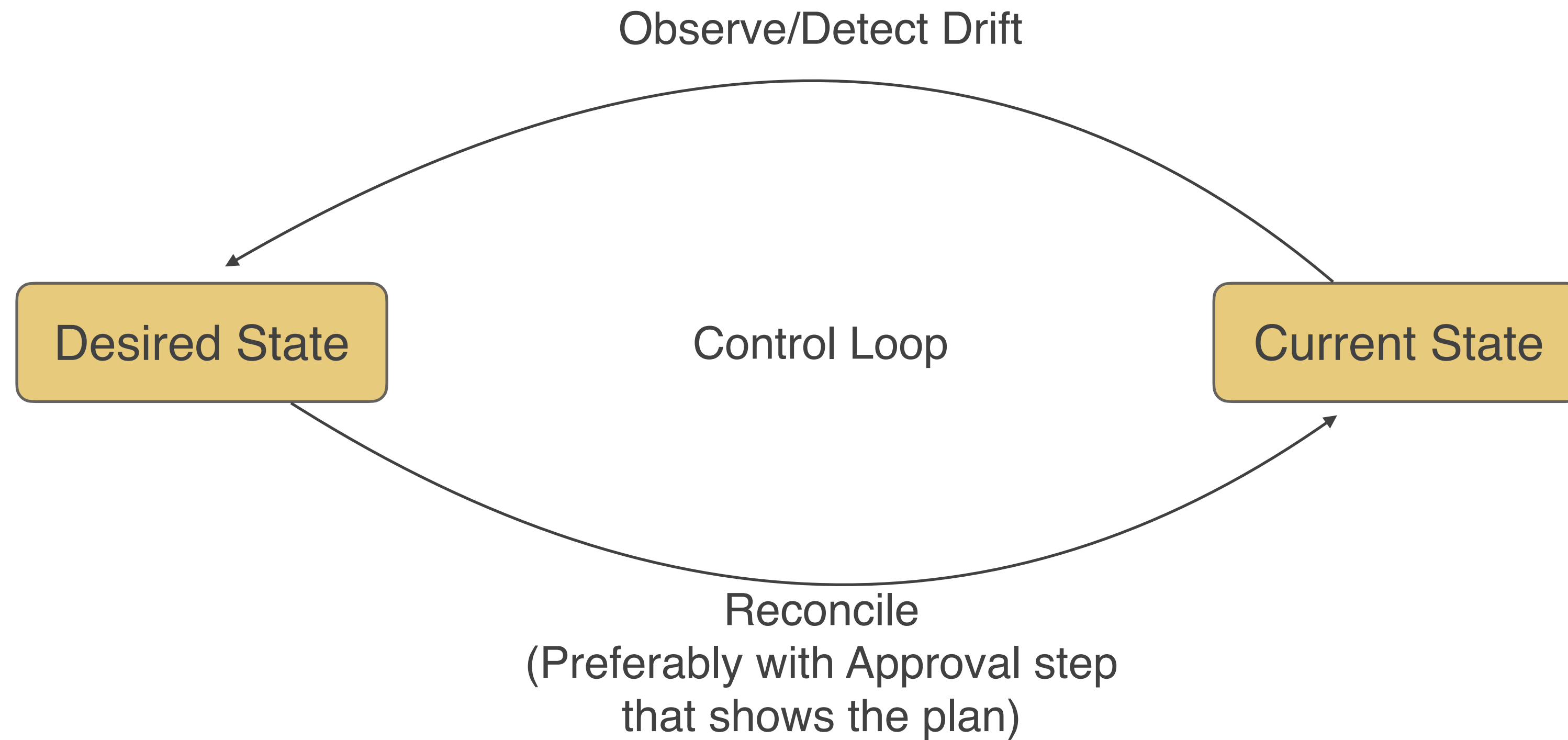
Compare and Promote Changes between Environments

Compare & promote changes
across environments

GitOps for Entire Environment

GitOps for Environment

Control Plane

Observe — Reconcile

Validate

Branch

Trunk

EaC

Create Pull Request,
EaC Validate

Approve &
Merge PR

networking

platform-k8s

k8s-addons

postgres
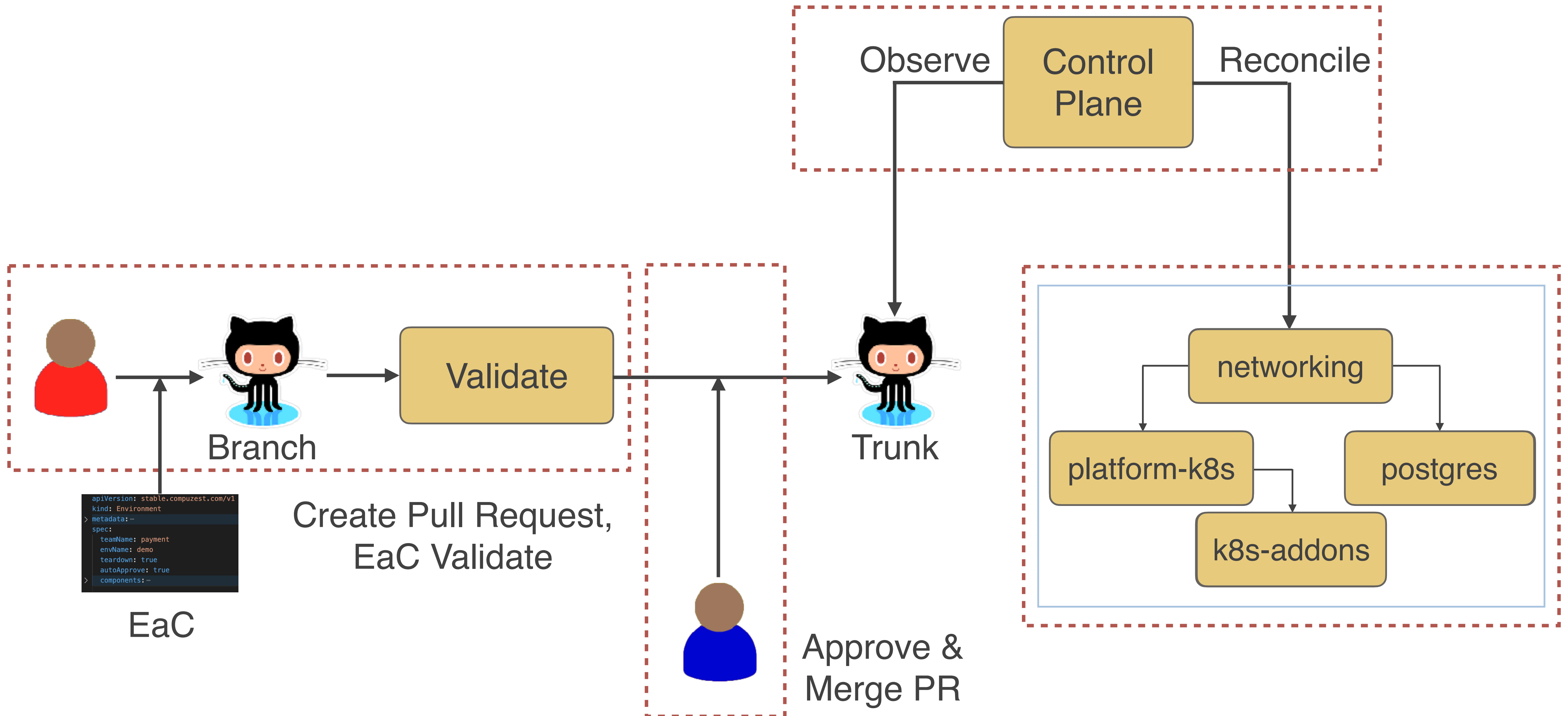
Thank You

Adarsh Shah

Engineering Leader, Coach, Public Speaker
Founder & CEO at CompuZest
🐦 *@shahadarsh*
🌐 *https://compuzest.com*

🐦 @shahadarsh