KubeCon | CloudNativeCon
Europe 2020
Virtual

Towards a standardized application definition model for Kubernetes

Phil Prasek, Upbound
Sudhanva Huruli, Microsoft

What’s an app model in k8s?

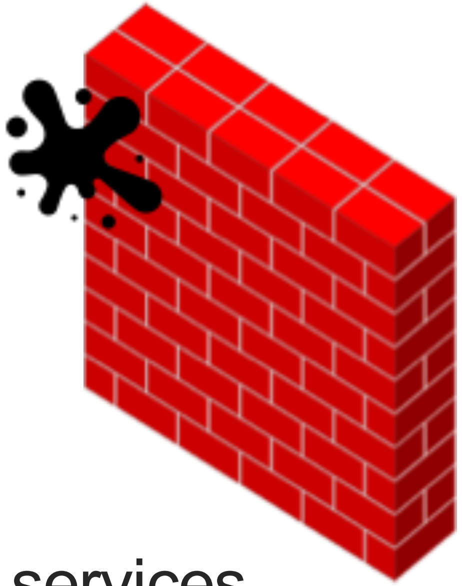
- **Container images**: apps, services
- **Workloads**: pods, deployments, services, …
- **Cloud services**: databases, caches, buckets, …
- **Configuration**: env vars, config maps, secrets, …
- **Ops Behavior**: auto scaling, networking, health, ...

- **Packaging / Customization**: Helm, kustomize, CNAB, …
- **Deployment**: pipelines, GitOps, workload scheduling, …

Path from dev laptop to prod

App Ops

App Dev →



docker compose

w/ in-cluster stateful services (db, cache, …)

Kubernetes + Cloud Services

- Deployments, services, …
- Packages + config
- Imperative pipelines w/ glue
- IaC tooling
- Cloud services: DB, cache, …
- Auto scaling, networking, health, …

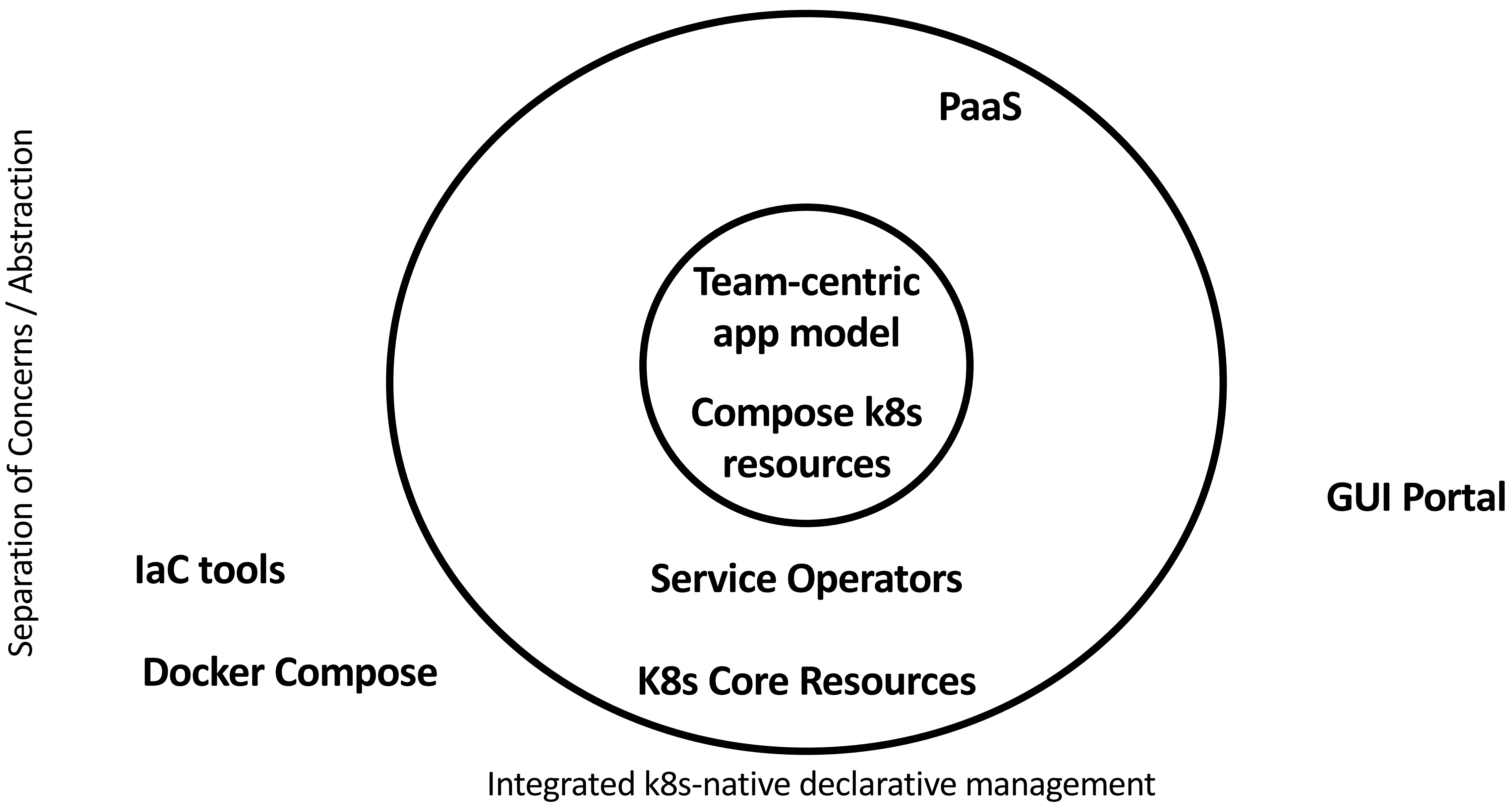
Platform Ops

Dev Laptop → **Dev / Test** → **Staging** → **Production**

Limitations

- Difficult for devs to get apps into prod
- Apps often glued together in imperative pipelines
- Multiple tools & management models
- Inconsistencies across environments
- Hard to master interactions and failure modes
- No clear ownership over various aspects
- Conflicting issues often found at deploy time

What are we looking for?



Separation of Concerns / Abstraction

PaaS

Team-centric app model

Compose k8s resources

GUI Portal

IaC tools

Service Operators

Docker Compose

K8s Core Resources

Integrated k8s-native declarative management

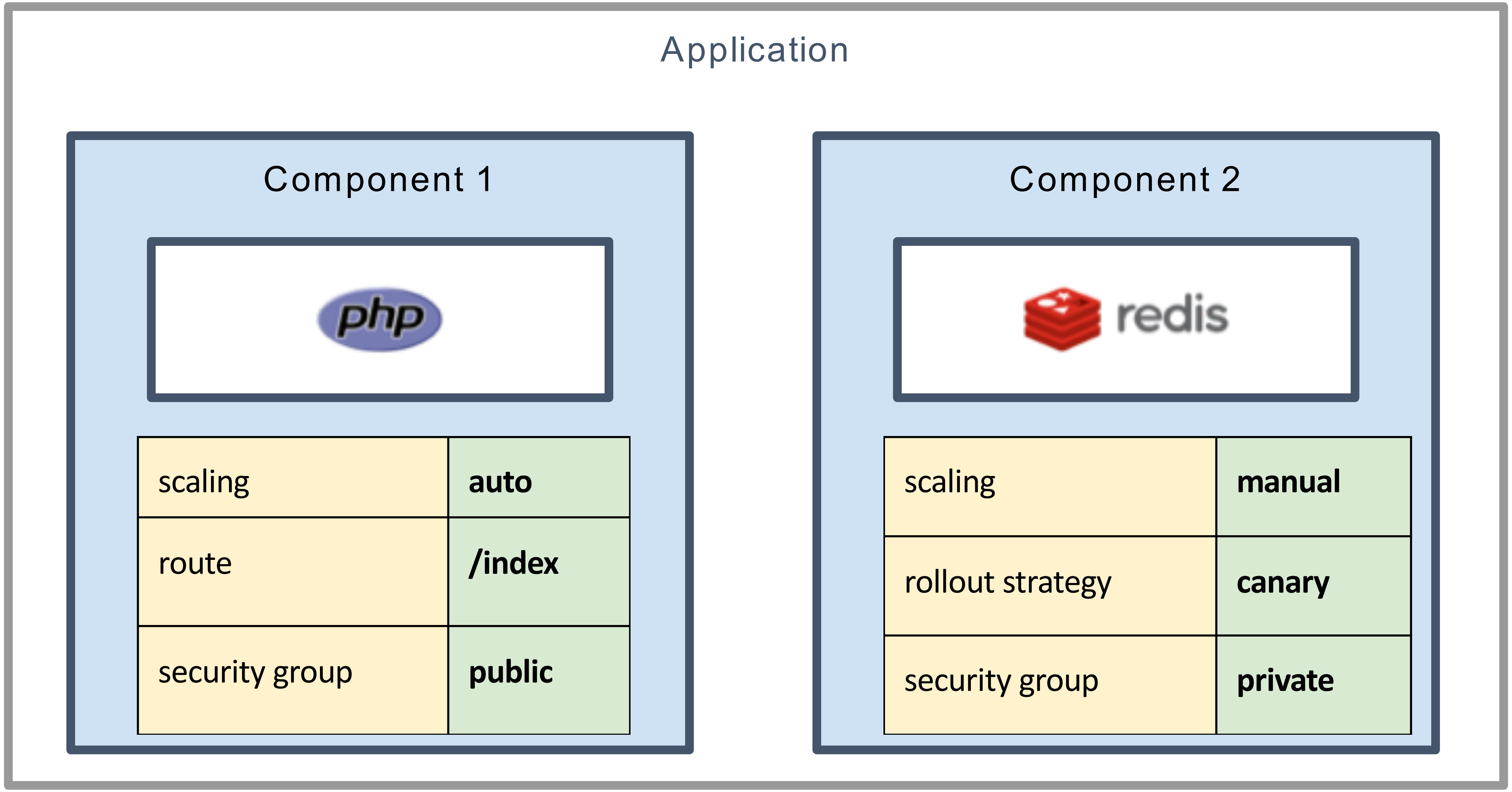
Desired app model properties

- **Models an entire prod application**: app config, infra, ops behavior
- **K8s-style declarative management**: ideally idiomatic Kubernetes
- **Clean separation of concerns**: enables team collaboration
- **Extensible via k8s API**: not a limiting PaaS or GUI Portal
- **Streamlined path from dev laptop to production**

Can use with:

- **Git as a source of truth**: peer review changes, audit trail, …
- **Existing tools**: Helm, kustomize, Flux / ArgoCD, CNAB, TF k8s, …
- **Existing deployment workflows:** pipelines, GitOps, …

The modern application



Application

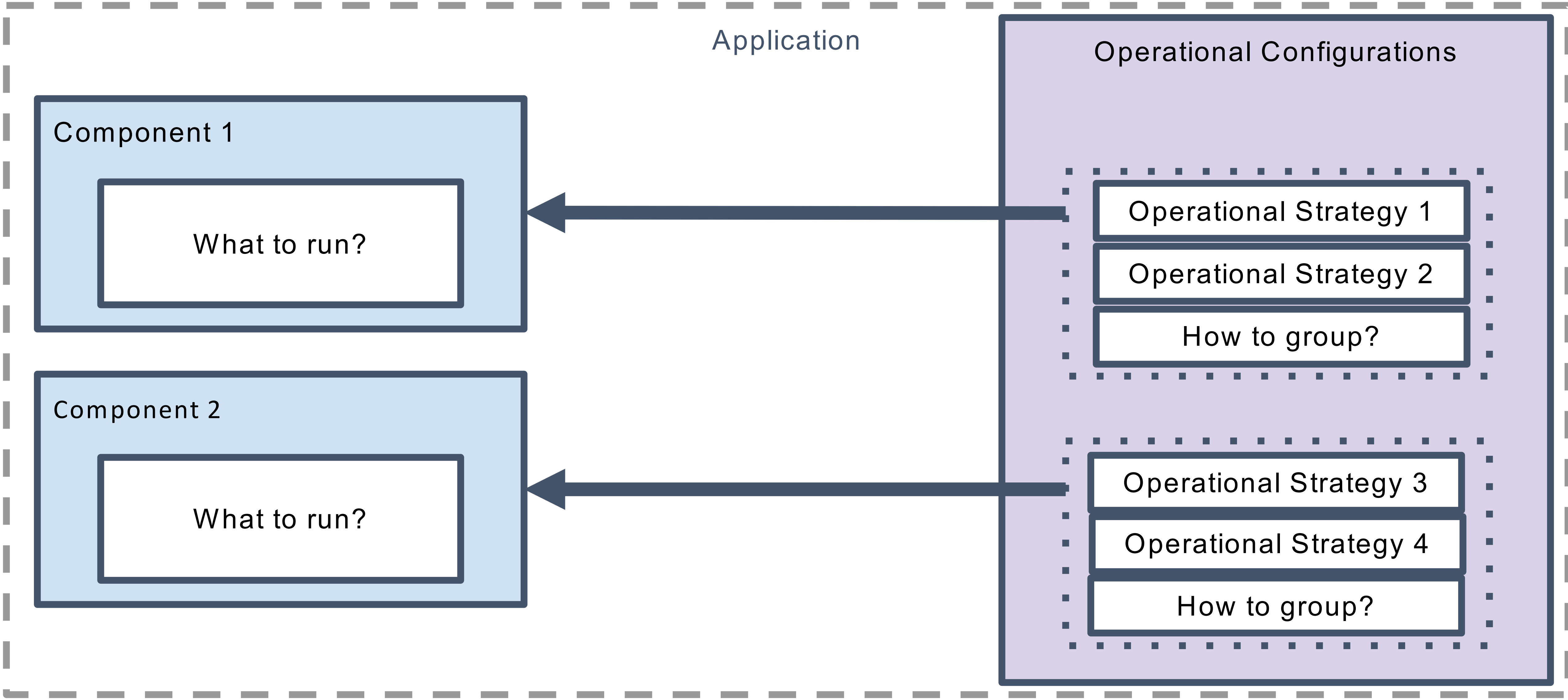
Component 1

scaling	**auto**
route	**/index**
security group	**public**

Component 2

scaling	**manual**
rollout strategy	**canary**
security group	**private**

Pattern for a modern app



Application

Component 1

What to run?

Component 2

What to run?

Operational Configurations

Operational Strategy 1

Operational Strategy 2

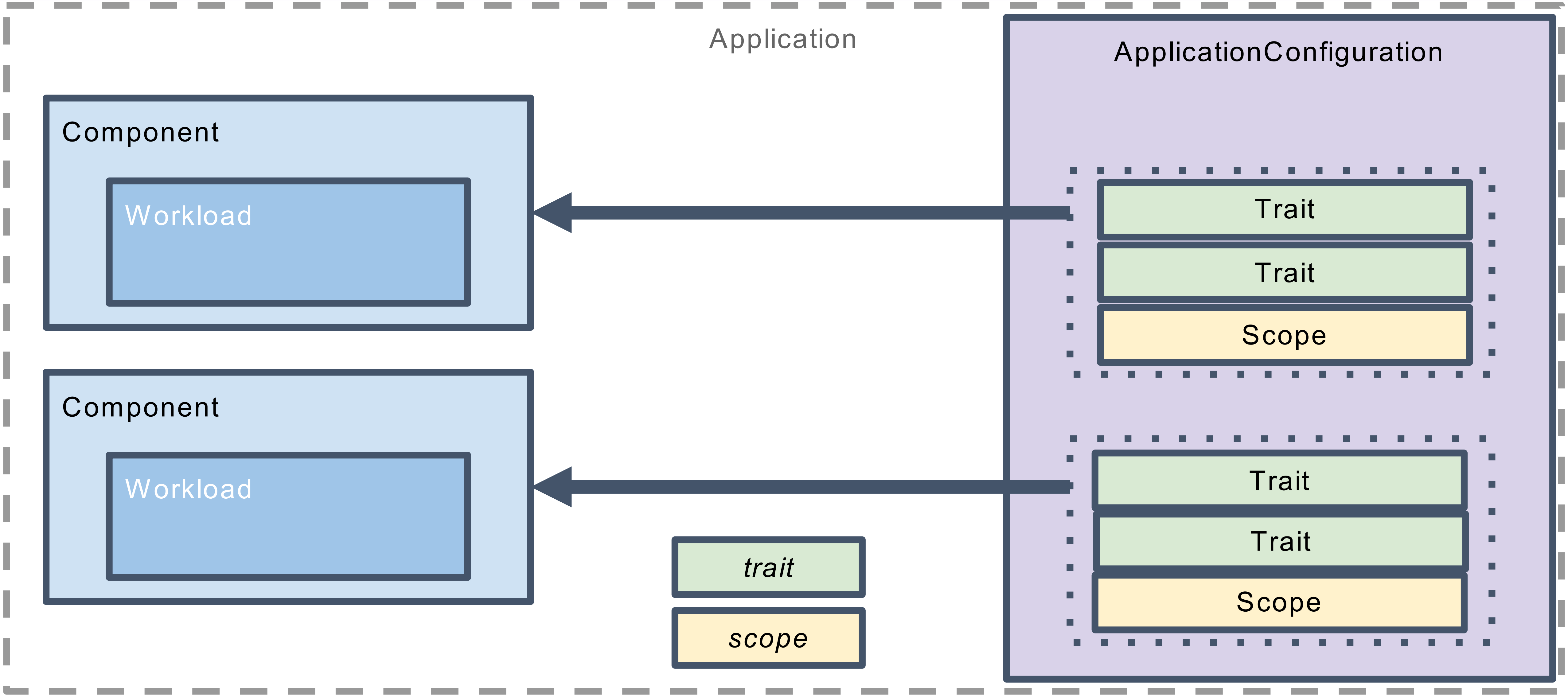
How to group?

Operational Strategy 3

Operational Strategy 4

How to group?

Open Application Model (OAM)



Application

Component

Workload

Component

Workload

ApplicationConfiguration

Trait

Trait

Scope

Trait

Trait

Scope

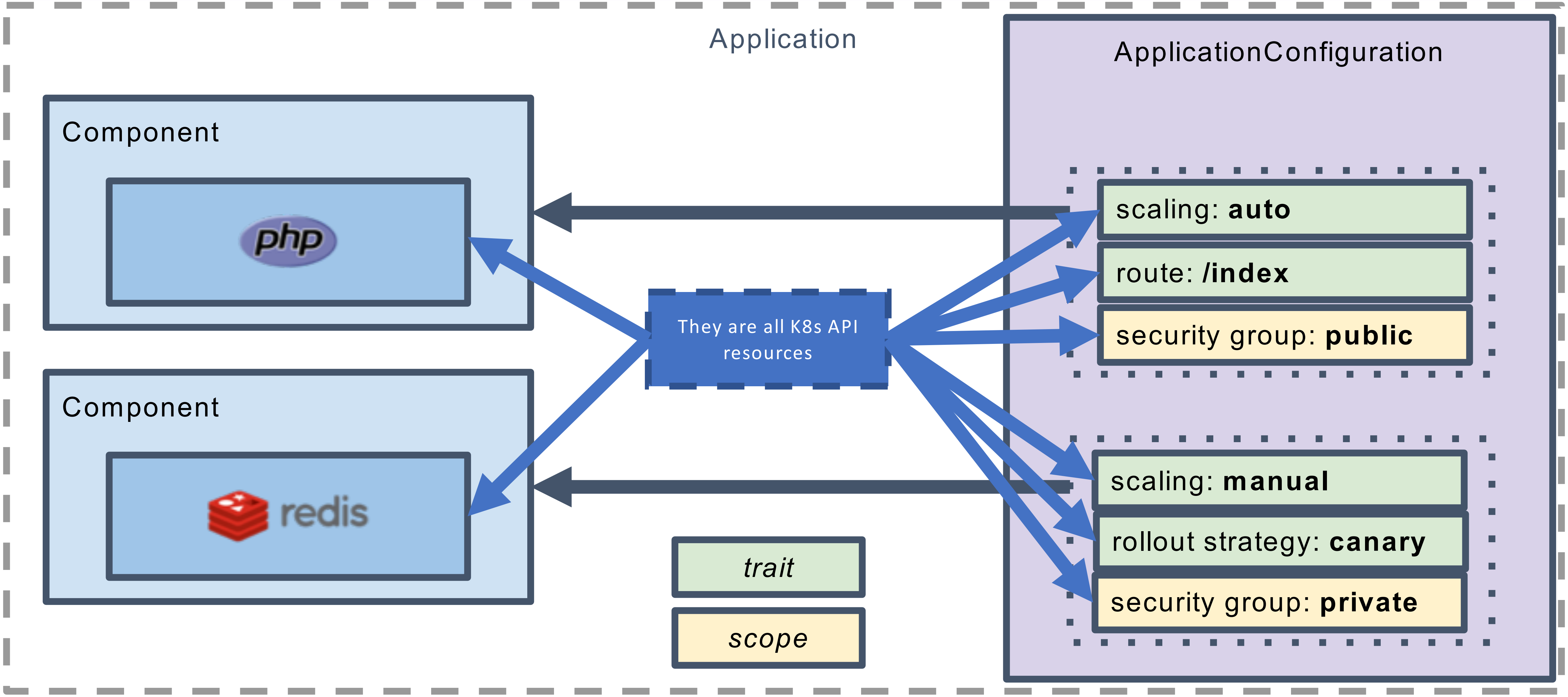
trait

scope

What to run?

How to operate?

Example



Application

Component

php

Component

redis

They are all K8s API resources

trait

scope

ApplicationConfiguration

scaling: **auto**

route: **/index**

security group: **public**

scaling: **manual**

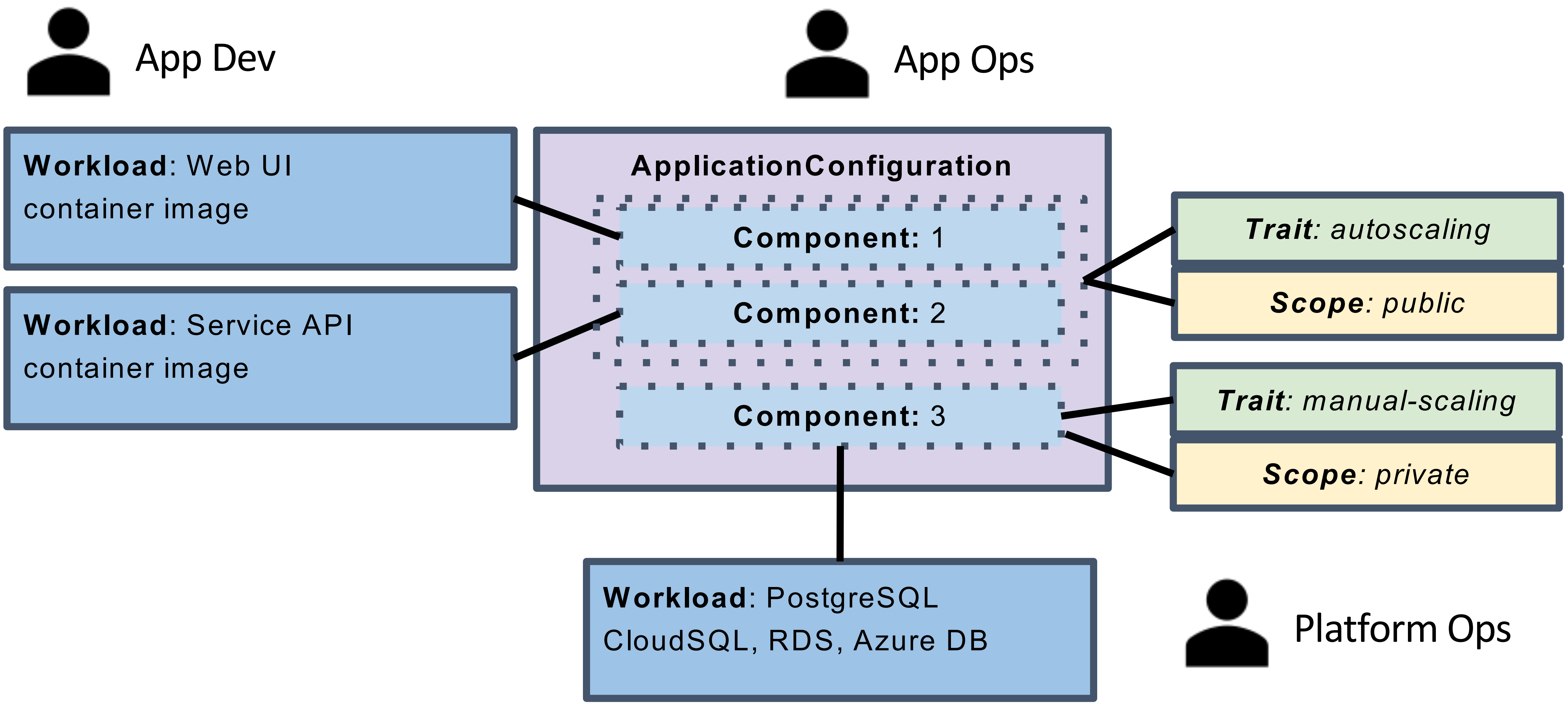
rollout strategy: **canary**

security group: **private**

What to run?

How to operate?

Separation of concerns



App Dev

App Ops

Workload: Web UI
container image

Workload: Service API
container image

ApplicationConfiguration

Component: 1

Component: 2

Component: 3

Trait: *autoscaling*

Scope: *public*

Trait: *manual-scaling*

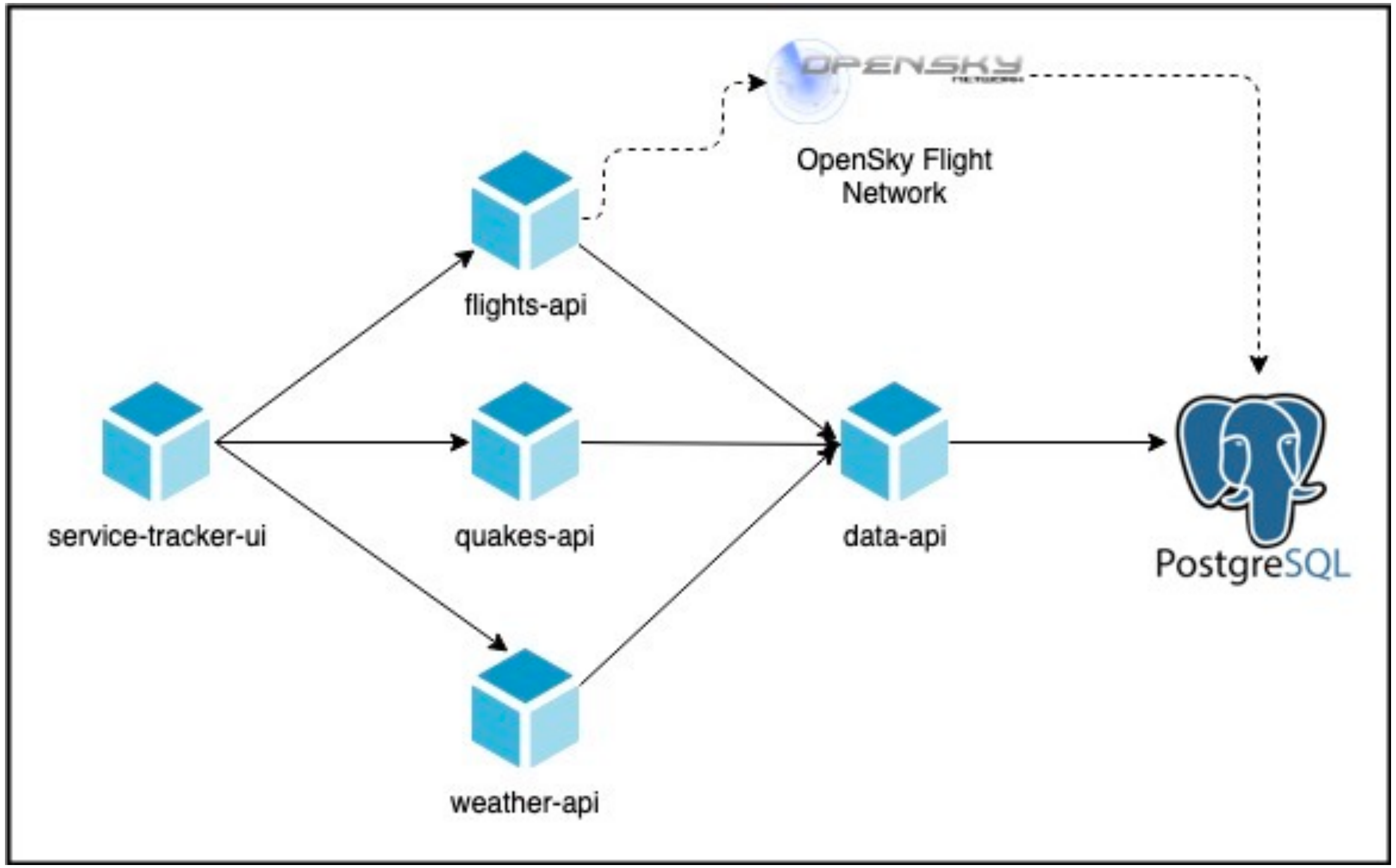
Scope: *private*

Workload: PostgreSQL
CloudSQL, RDS, Azure DB

Platform Ops

Demo

Deploy k8s app with cloud services – entirely from k8s API


OpenSky Flight Network
flights-api
service-tracker-ui
quakes-api
data-api
PostgreSQL
weather-api

Crossplane (Platform Ops):

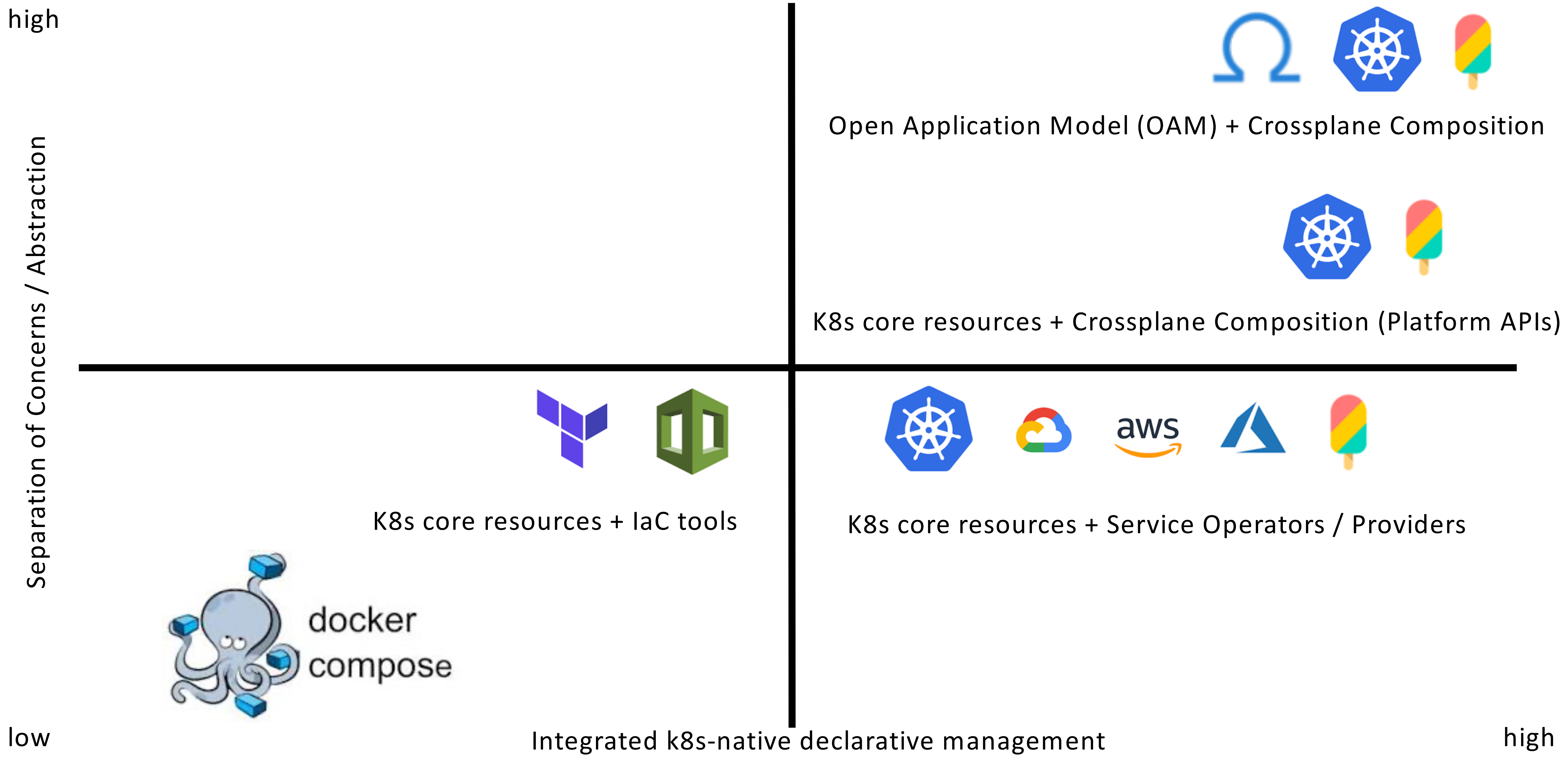
- Connect cloud provider
- Compose Infra Resource(s)
- Publish Infra CRDs for app teams to use with no code (PostgresSQL)

OAM (App Dev / App Ops)

- Define Workloads & Traits
- Define App Components
- Define App Configuration
- Run Application

https://crossplane.github.io/docs/v0.12/getting-started/install-configure.html

Towards a standard k8s app model



high

Separation of Concerns / Abstraction

Open Application Model (OAM) + Crossplane Composition

K8s core resources + Crossplane Composition (Platform APIs)

K8s core resources + IaC tools

K8s core resources + Service Operators / Providers

docker compose

low

Integrated k8s-native declarative management

high

What’s next?

Today:

Alibaba Cloud Microsoft upbound Crossplane ❤ Open Application Model

- Alibaba Cloud and Microsoft have joined the Crossplane Community
- OAM spec: now at v1alpha2
- Crossplane: the Kubernetes implementation of OAM
- Crossplane: reviewed / approved in CNCF sig-app-delivery
- Crossplane: CNCF Sandbox Project

Next:

- Active OAM development in Crossplane to implement the full spec
- Moving towards v1beta1 as part of the upcoming Crossplane 1.0 release

Get Involved

https://oam.dev/ - OAM spec

https://crossplane.io/ - the Kubernetes impl. of OAM

https://slack.crossplane.io/ - #oam channel

https://github.com/crossplane/crossplane

https://github.com/crossplane/oam-kubernetes-runtime

More KubeCon EU Talks

Tuesday, August 18

18:30 Panel: App Management on K8s: The Good, the Bad and the Ugly

- Matt Butcher & MacKenzie Olson, Microsoft; & Jian He, Alibaba

Wednesday, August 19

16:55 Standardizing Applications For the Cloud at a Global Scale

- Jared Watts, Upbound & Lei Zhang, Alibaba

Q & A

KubeCon | CloudNativeCon

Europe 2020

Virtual

KEEP CLOUD NATIVE
CONNECTED