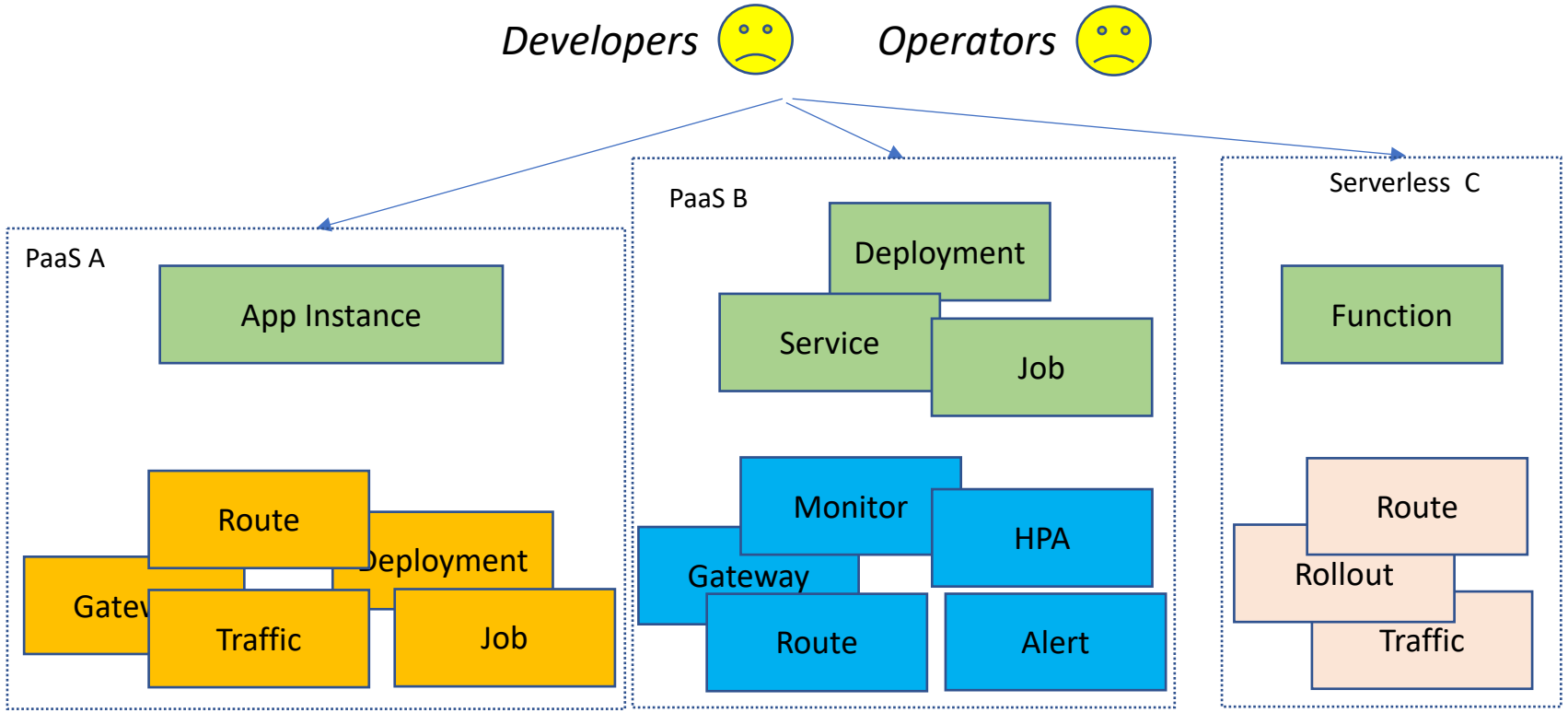
Open Application Model in Alibaba

Xiang Li

Alibaba Cloud

Application Management At Alibaba



Challenges

- **Fragmentation** in app layer, ~11 internal PaaS/Serverless
- **Siloes**, lack reusability and interoperability
- **Infra centric**, low level primitives leaked to end user

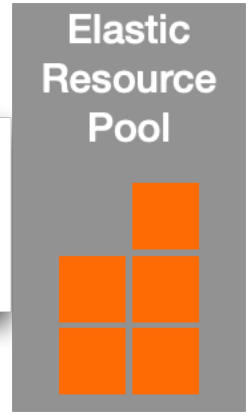
Scale:

- **10,000** nodes/cluster
- **100,000** apps/cluster
- **1,000,000** containers/cluster

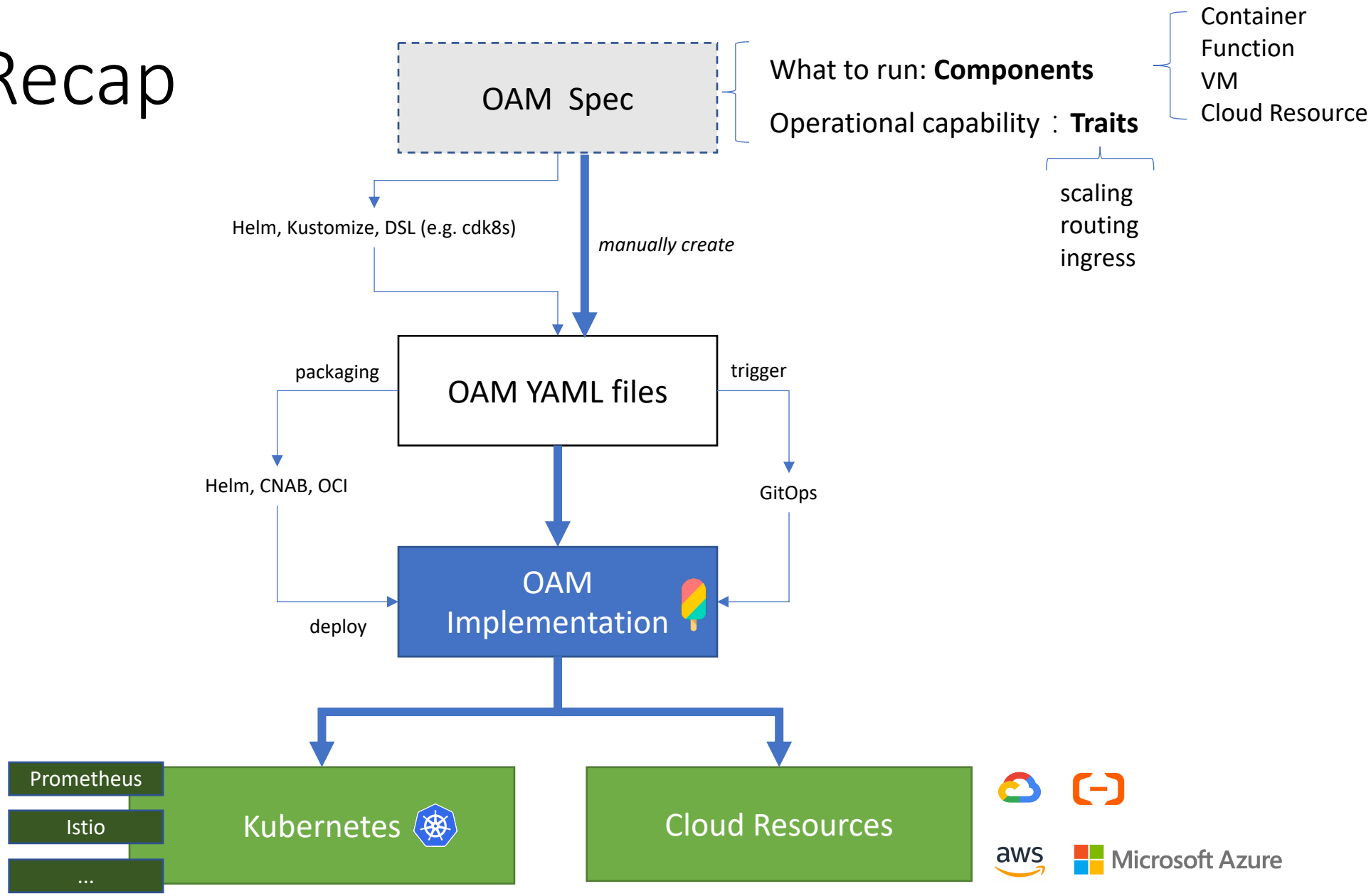
Velocity:

- **100,000** deploys/day
- **500~1000** replicas/app

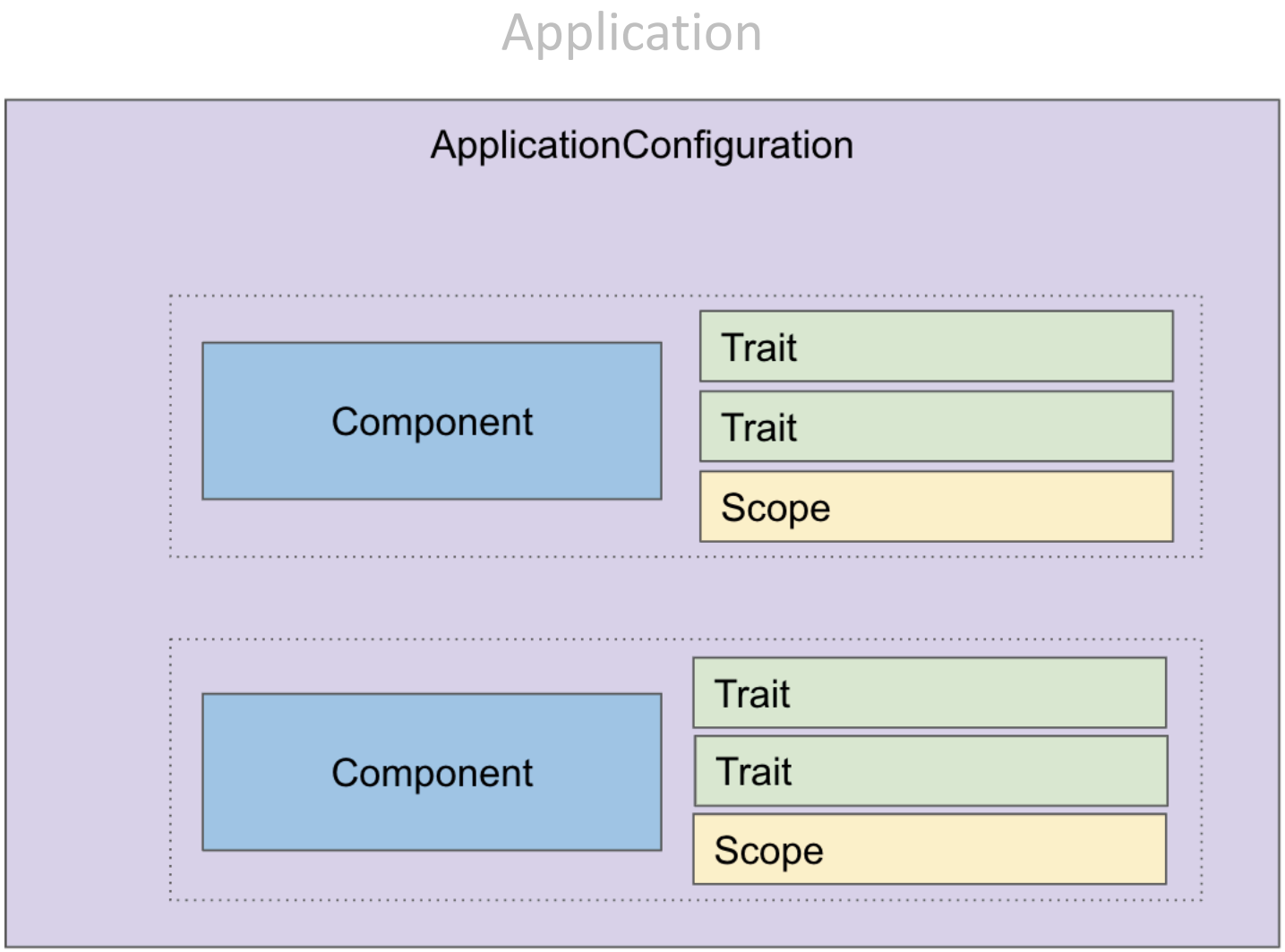
Infra Ops



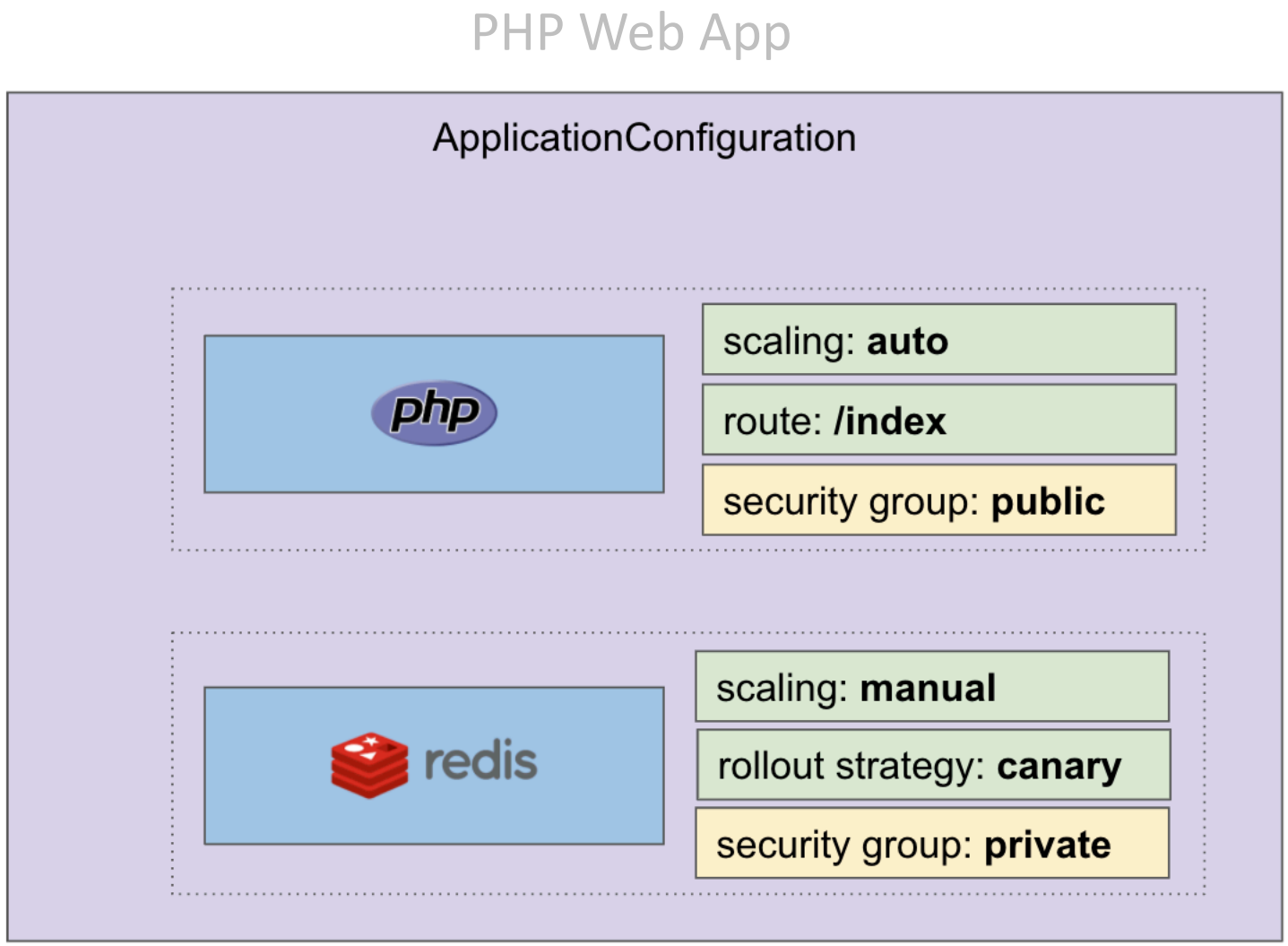
OAM Recap


OAM Spec
What to run: **Components**
Container
Function
VM
Cloud Resource
Operational capability：**Traits**
scaling
routing
ingress
Helm, Kustomize, DSL (e.g. cdk8s)
manually create
packaging
OAM YAML files
trigger
Helm, CNAB, OCI
GitOps
deploy
OAM Implementation
Prometheus
Istio
...
Kubernetes
Cloud Resources
aws
Microsoft Azure

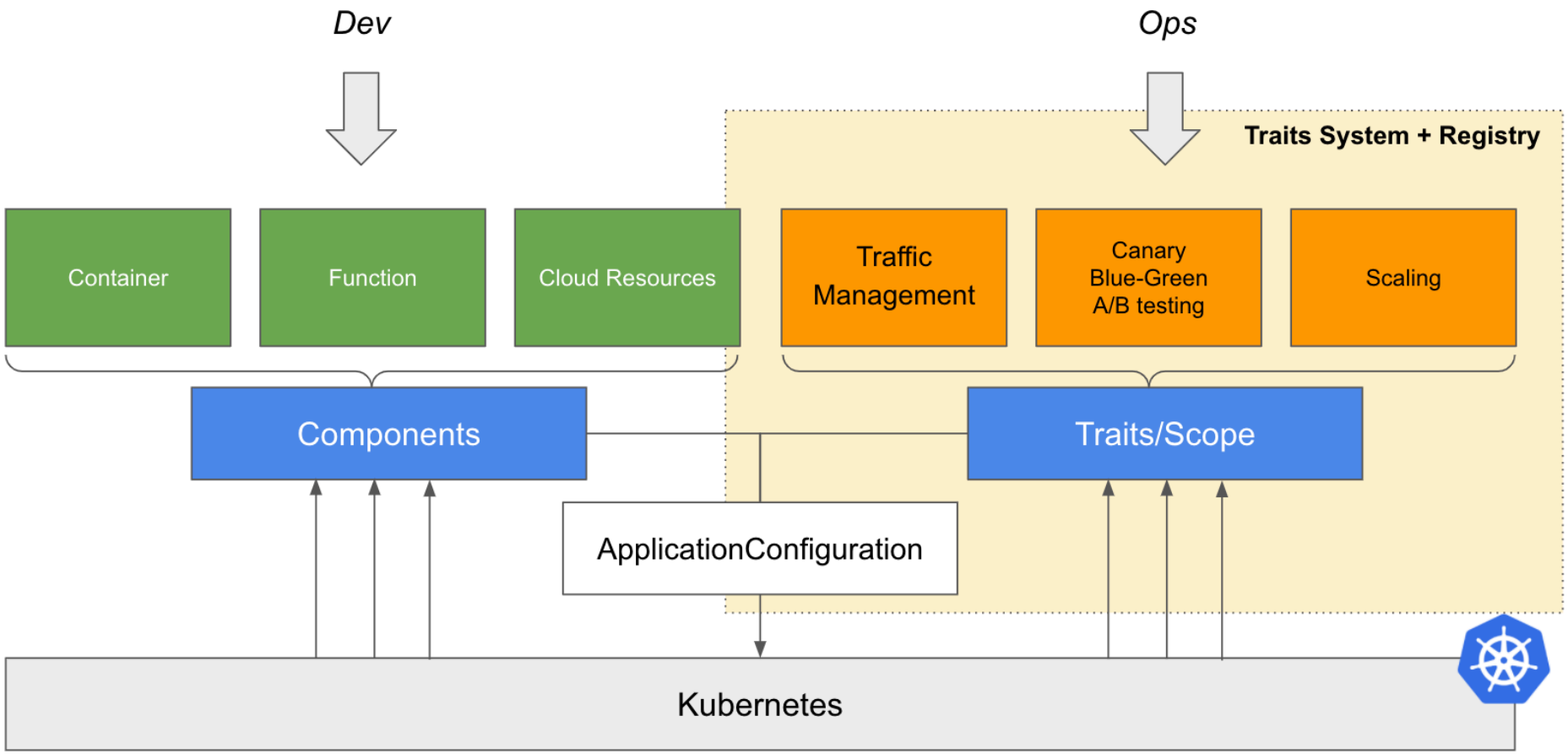
Open Application Model



For example


PHP Web App
ApplicationConfiguration
php
scaling: auto
route: /index
security group: public
redis
scaling: manual
rollout strategy: canary
security group: private

A typical OAM based app platform



How K8s capabilities are managed today?

```yaml
apiVersion: autoscaling/v2beta2
kind: CronHorizontalPodAutoscaler
metadata:
  name: nginx-scaler
spec:
  # NOTE: CronHPA can not be used with HPA at the same time
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nginx-deployment
  jobs:
  - name: "scale-down"
    schedule: "30 */1 * * * *"
    targetSize: 1
  - name: "scale-up"
    schedule: "0 */1 * * * *"
    targetSize: 3
```

a real world example in Alibaba PaaS

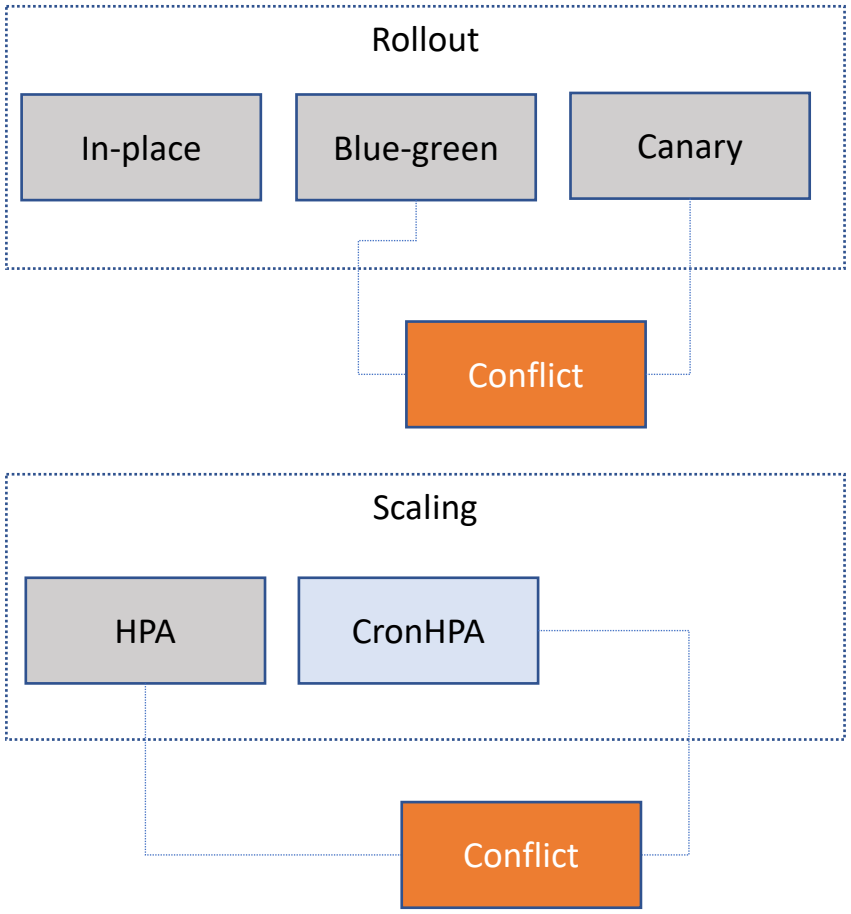
Current state

- ~100 CRD + controllers installed in one cluster
- use annotation/webhook to discover and manage

Challenges

- Nobody knows **how many capabilities** the system supports exactly, even some of them are duplicated/deprecated
- Nobody knows **how many capabilities are applied to a given application**
- Nobody knows if **one capability is conflict with the other** until the deployment is fail

Trait Registry *(WIP)*



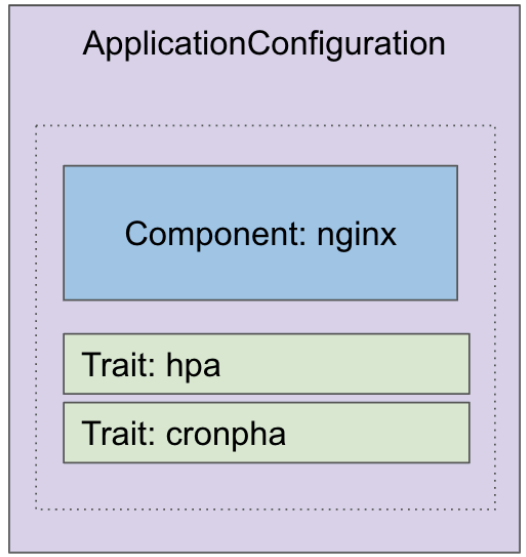
1. Register

$ kubectl oam install cronhpa-def.yaml category=scaling conflict=hpa
SUCCESS – CRD ***cronhpa.core.oam.dev*** is registered as ***scaling*** trait!

2. Discover

$ kubectl get traits

NAME	CATEGORY
blue-green	rollout
in-place	rollout
hpa	scaling
cronhpa	scaling



app-config.yaml

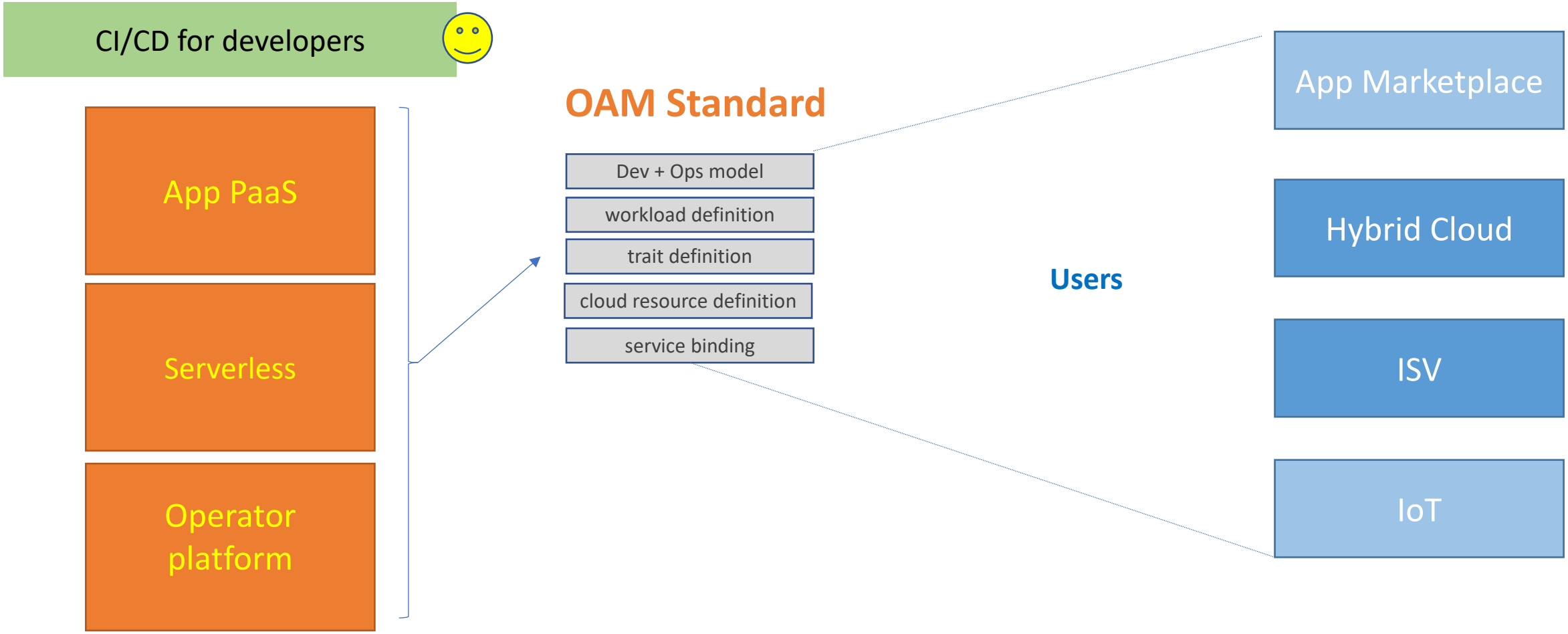
3. Auto conflict detect at deploy time

$ kubectl apply –f app-config.yaml
ERROR – traits ***hpa*** and ***cronhpa*** MUST not be applied to same component!

Why?

- ***Right level of abstraction*:** developer level abstraction to consolidate the application management experience,
- ***Team-centric***: separate concerns for *developers*, *operators*, and *infra engineers.*
- ***Platform agnostic:*** deploy with same YAML to serverless platform, k8s on prems, or any cloud.
- ***Manageability and discoverability:*** detect conflicted operational strategies at deploy

OAM in Alibaba at highest level



App/Serverless platforms in Alibaba

- Alibaba Application Platform (EDAS)
  - distributed Java application platform
  - Public product
- Alibaba Operator Platform
  - Stateful application platform
  - Internal platform
- Alibaba Serverless Platform
  - Serverless (event driven) application platform
  - Both public product and internal platform

OAM + Alibaba Application PaaS (EDAS)

- Before: traditional PaaS with close ecosystem
- After: OAM as core application model, provide simplified end user experience but still keep the extensibility of k8s, make EDAS become into an open ecosystem

Feature	OAM Category	Description
• ContainerizedWorkload	• component	• Stateless app
• Virtual Machine	• component	• Legacy app
• traffic rollout	• trait	• Traffic shifting
• monitoring	• trait	• Prometheus configuration
• logging	• trait	• SLS log collecting policy
• auto-scaler	• trait	• HPA
• manual-scaler	• trait	• Manual scaling
• rollout	• trait	• Blue-green deployment
• ingress	• trait	• Routing policy

Share with other platforms

Balance extensibility of k8s with best end-user experience by OAM based abstractions

Reusable components and traits to save engineering time

Discoverability for capabilities in the system, detect strategy conflict at deploy

OAM + Alibaba Operator Platform

- Before: in-house app definition, lack of portability and interoperability across clusters
- After: OAM as core application model to manage stateful applications, e.g. Kafka, Zookeeper and Database on any cluster

feature	OAM Category	Description
OpenKruise	component	Alternative to StatefulSet
Operators	component	Workloads based on operator
In-place rollout	trait	In-place rollout
canary	trait	Canary deployment
khpa	Trait	HPA to scale Operators
pvc	trait	Claim persistent storage
backup	trait	Data backup
restore	trait	Data restore
ingress	trait	Routing policy
monitoring	trait	Integrated internal monitoring policy

Share with other platforms

All speak OAM API, easy to integrate with other OAM systems

OAM to decouple application from runtime, easy to upgrade infrastructure

Reusable components and traits to save engineering time

OAM + Alibaba Serverless Platform

- Before: in-house app definition, reinvent every wheel for serverless
- After: OAM as unified definition of various workload types, and reuse traits from may other systems like EDAS.

feature	OAM Category	Description
• ContainerizedWorkload	• component	• Serverless container
• Function	• component	• FaaS workload
• Knative service (WIP)	• component	• Knative serving workload
• traffic-mgmt	• trait	• Istio based Traffic shifting
• cron-scaler	• trait	• Cron HPA
• manual-scaler	• trait	• Manual scale
• auto-scaler	• trait	• HPA
• sls-project, sls-logstore, sls-index	• component	• Alibaba Cloud Resources as BaaS

Share with other platforms

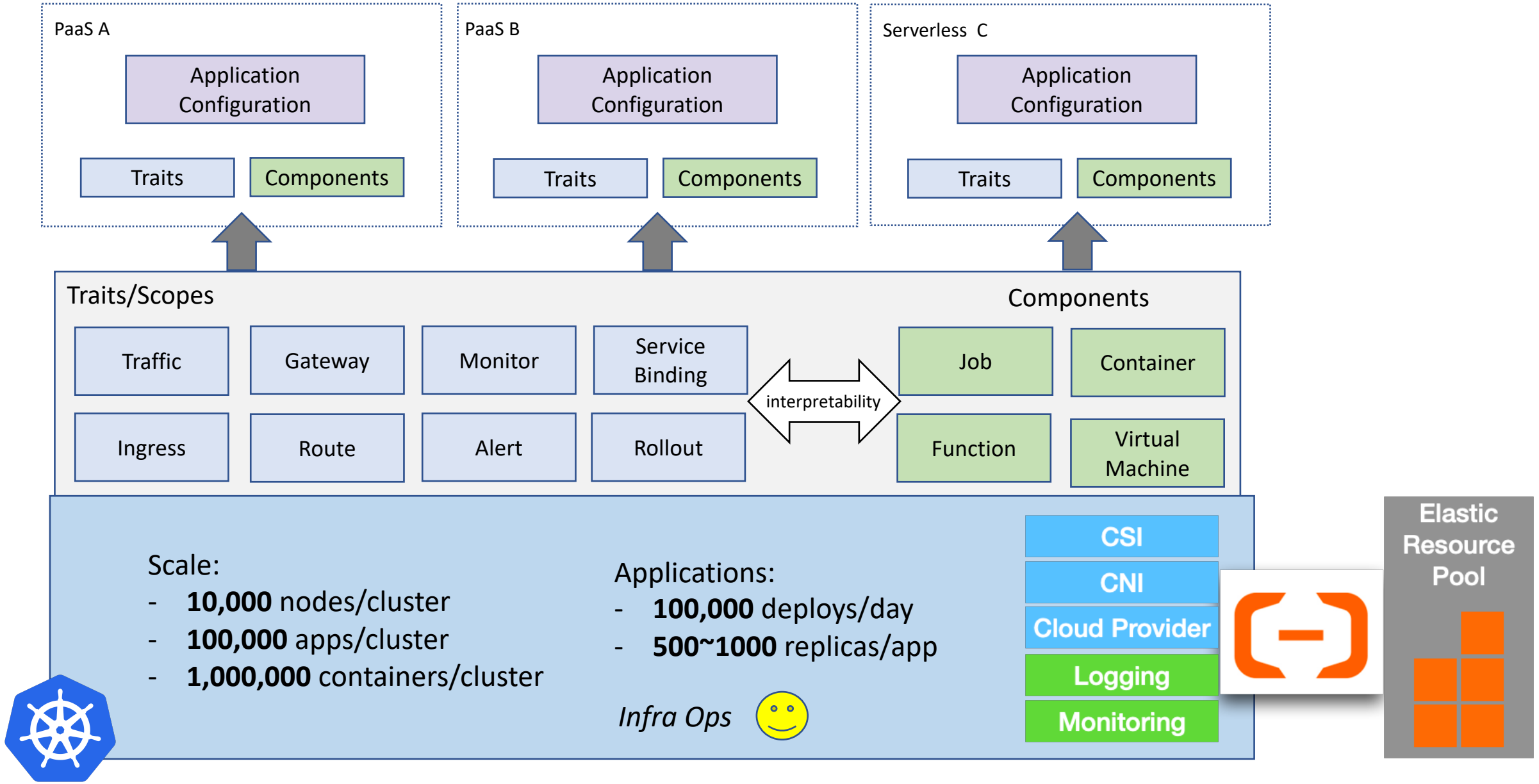
Separate concerns make developers accept serverless model way easier

Migrate from K8s/PaaS to Serverless only need to update its trait

Reusable components and traits to save engineering time

Connect and consume cloud resources thru OAM based BaaS

Future


PaaS A
Application Configuration
Traits
Components
PaaS B
Application Configuration
Traits
Components
Serverless C
Application Configuration
Traits
Components
Traits/Scopes
Traffic
Gateway
Monitor
Service Binding
Ingress
Route
Alert
Rollout
interpretability
Components
Job
Container
Function
Virtual Machine
Scale:
- 10,000 nodes/cluster
- 100,000 apps/cluster
- 1,000,000 containers/cluster
Applications:
- 100,000 deploys/day
- 500~1000 replicas/app
Infra Ops
CSI
CNI
Cloud Provider
Logging
Monitoring
Elastic Resource Pool

Kubernetes in Alibaba with OAM

Modular, reusable, standardized, and manageable platform capabilities, best engineering efficiency

Clear separate of concerns in the workflow of app management, best communication and collaboration efficiency

Minimal migration effort from PaaS/K8s to OAM platform

The simplest and unified approach to integrate and **consume Alibaba Cloud Resources**

For Platform

- **Simple, team-centric and unified experience**
- Enjoy “unlimited” capabilities provided by k8s ecosystem - **the “open” PaaS**

For end users (Dev and Ops)

Demo - FlightTracker

Deploy and run the app on different clouds with exactly same YAML file