

Standardizing Cloud Native Application *Virtual* Delivery Across Different Clouds



Hongchao Deng, Andy Shi @Alibaba

Who Are We?



KubeCon



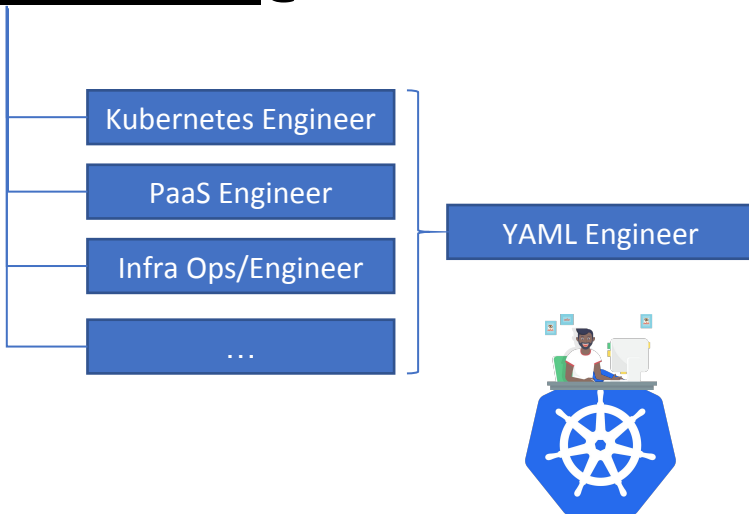
CloudNativeCon

North America 2020

Virtual

We are:

- **Platform Builders** @Alibaba



How Does Alibaba Ensure the Performance of System Components in a 10,000-node Kubernetes Cluster?

Alibaba Developer October 24, 2019 23,653 0

This article looks the problems and challenges that Alibaba Cloud overcame for Kubernetes to function at an ultra-large scale and the specific solutions proposed.

By Zeng Fansong, senior technical expert for the Alibaba Cloud Container Platform, and Chen Jun, systems technology expert at Ant Financial.

This article will take a look at some of the problems and challenges that Alibaba and its ecosystem partner [Ant Financial](#) had to overcome for [Kubernetes](#) to function properly at mass scale, and will cover the solutions proposed to the various problems the Alibaba engineers encountered. Some of these solutions include improvements to the underlying architecture of the Kubernetes deployment, such as enhancements to the performance and stability of etcd, the kube-apiserver, and kube-controller. These were all crucial for Alibaba to ensure the support needed for the 2019 [Tmall 618 Shopping Festival](#) to take full advantage of the 10,000-node Kubernetes cluster deployment. They are also important lessons for any enterprise interested in following Alibaba's footsteps.

What Do We Build?



KubeCon

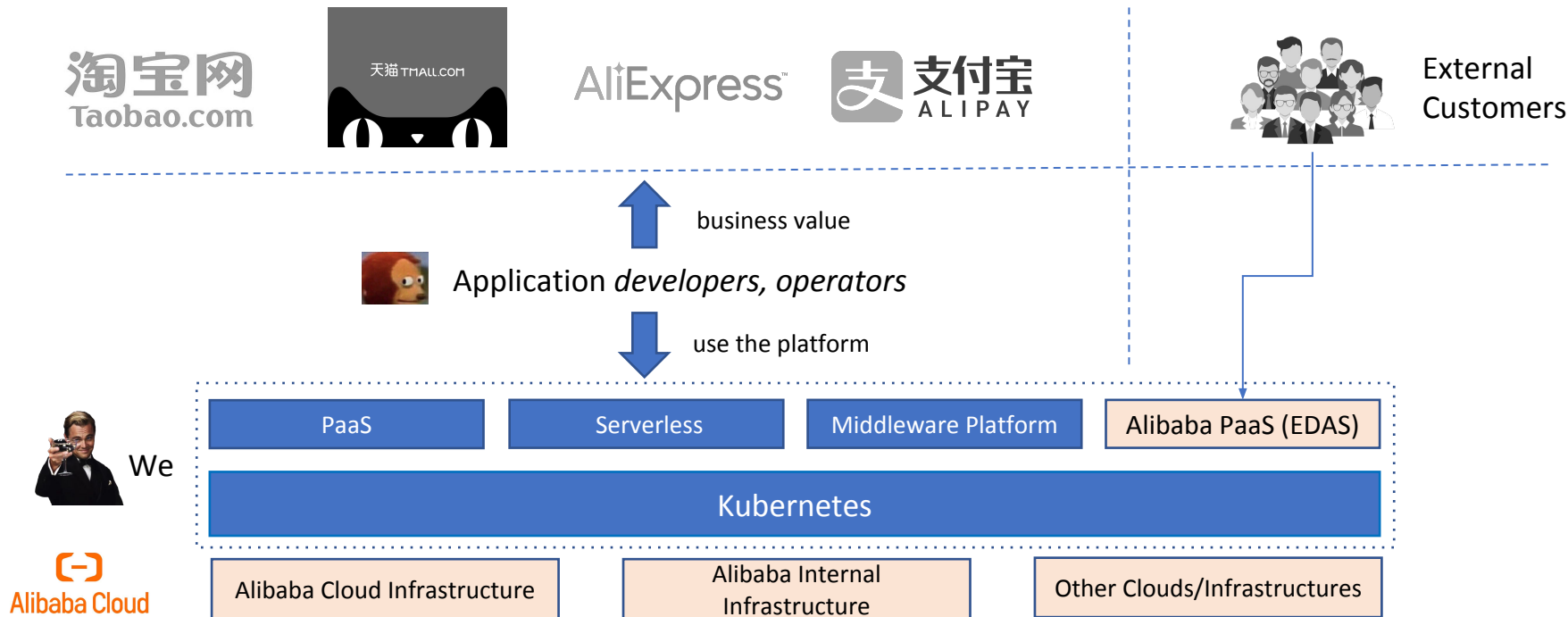


CloudNativeCon

North America 2020

Virtual

- Well ... lots of platforms on top of k8s, in hybrid environments



Why Do We Build Platforms?



KubeCon



CloudNativeCon

North America 2020

Virtual

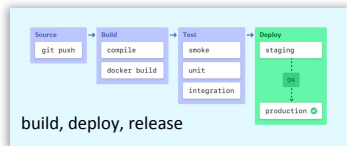
Bring **application** context back to k8s!



Application *developers, operators*



App-Centric API



what our platforms provide

App-Centric Abstractions

scaling

- auto scale +100 instances when latency > 10%

rollout

- promote the canary instance with step of 10%

App-Centric User Interfaces



GUI



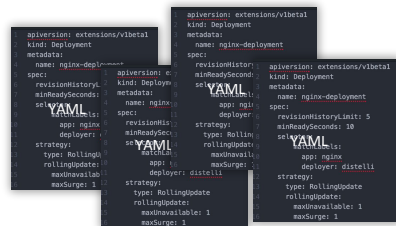
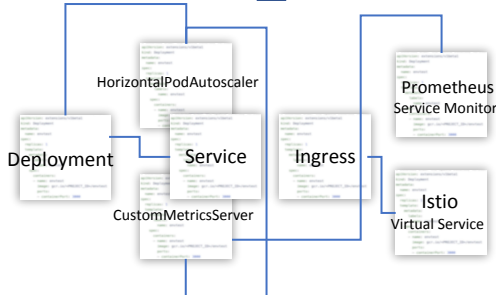
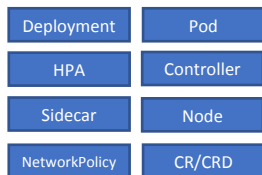
CLI



IaC



what k8s provides



OK, Show Me Your Platform!



KubeCon



CloudNativeCon

North America 2020

Virtual



Demo: the first glance of **KubeVela**

<https://github.com/oam-dev/kubevela>

Tell Me More About Your Platform?



KubeCon

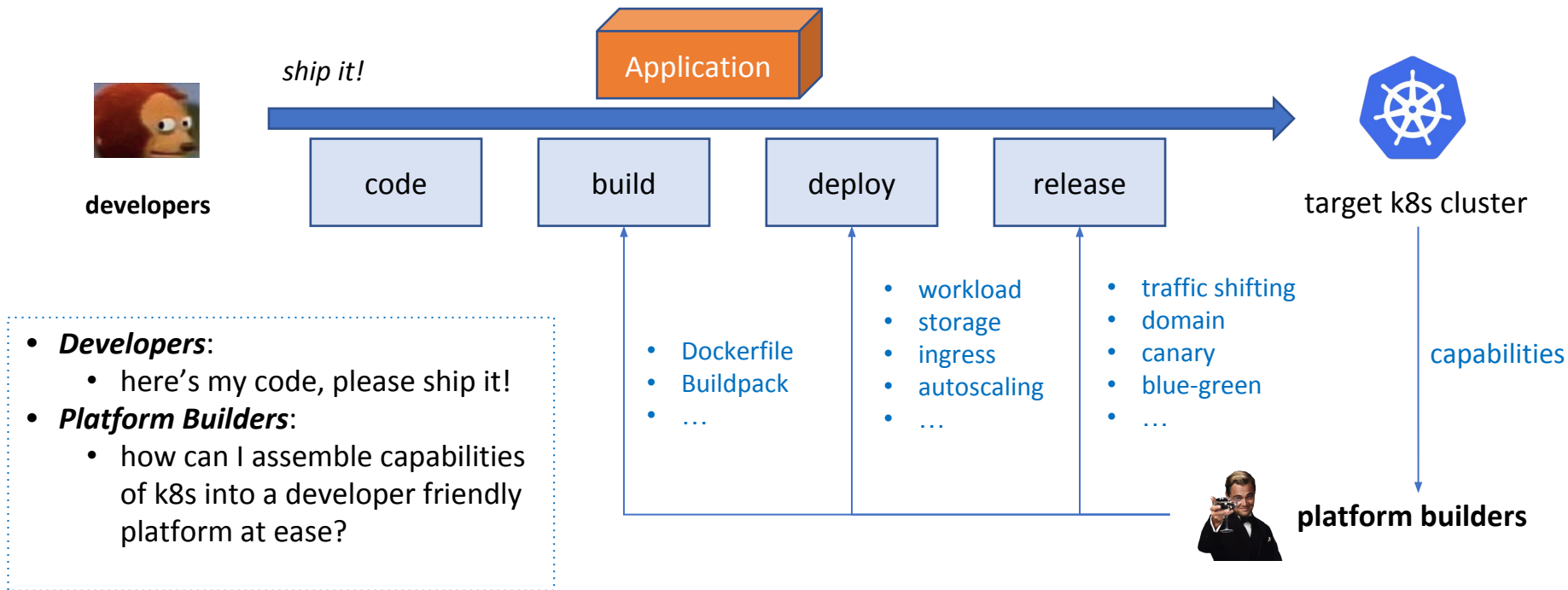


CloudNativeCon

North America 2020

Virtual

KubeVela aims at **both** *developers* and *platform builders*



- **Application-Centric**

- We believe “application” should be the main (maybe the only?) API our platform exposes to users.

- **Capability Oriented Architecture (COA)**

- **Every** feature in KubeVela is a independent plugin (either a k8s built-in resource or your own CRD controller).
 - e.g. Alibaba use KubeVela adopts Flagger as *rollout* trait, KEDA as *autoscaling* trait

- **Highly extensible, even for its user interface**

- When a new capability is installed, it should immediately consumable by end users without re-compiling or re-installing KubeVela.
 - e.g. KubeVela's [Appfile](#)

So, KubeVela in Nutshell



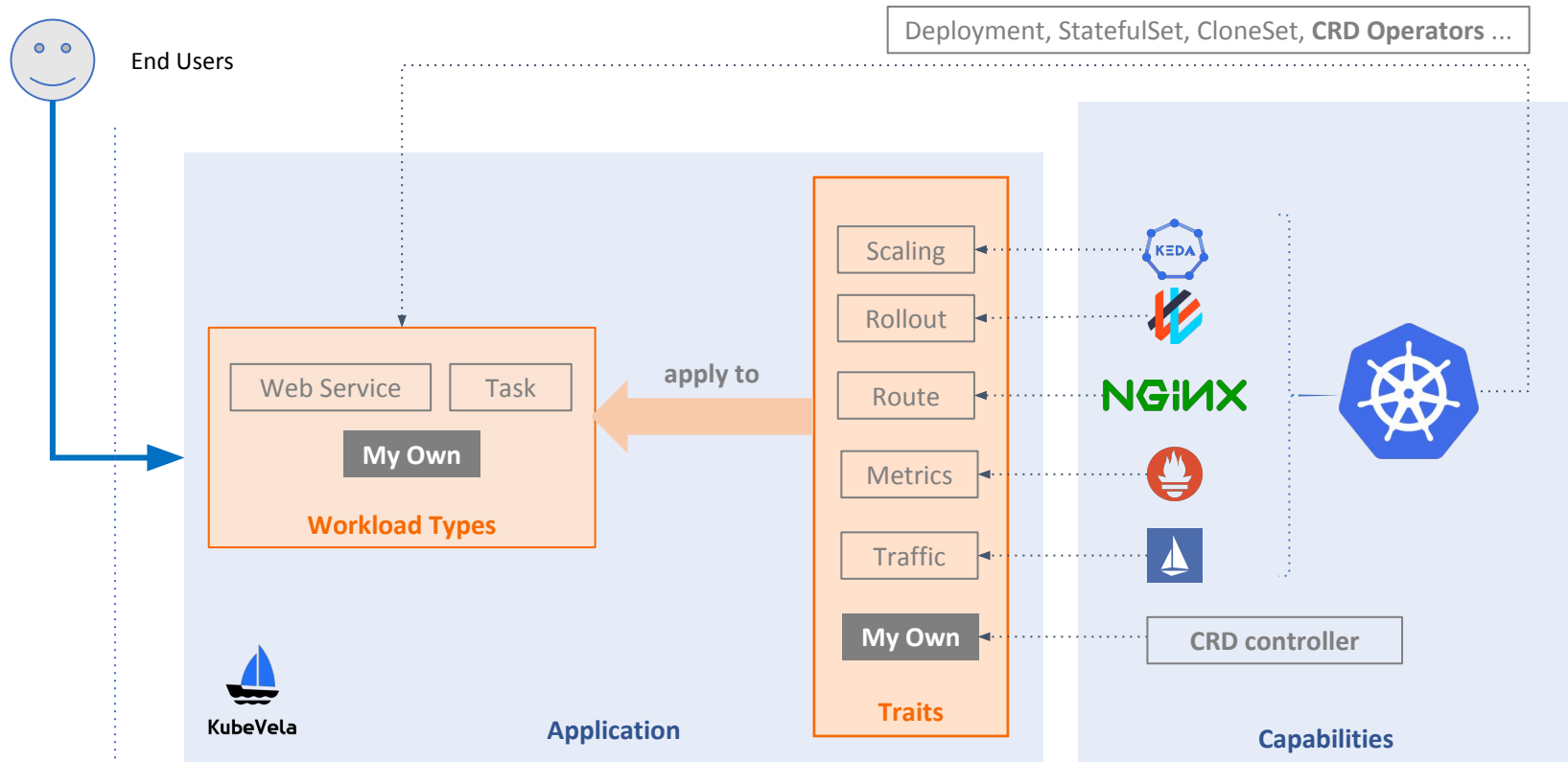
KubeCon



CloudNativeCon

North America 2020

Virtual



Extensible UI

"Application" as the main API

Capability discovery and management

Introducing Appfile



KubeCon



CloudNativeCon

North America 2020

Virtual

```
services:
  express-server:
    build:
      image: oamdev/testapp:v1
    docker:
      file: Dockerfile
      context: .

    cmd: ["node", "server.js"]

  route:
    domain: example.com
    http: # match the longest prefix
      "/": 8080

env:
- F00=bar
- F002=sec:my-secret # map the key sam
- F003=sec:my-secret:key # map specifi
- sec:my-secret # map all KV pairs fro

files: # Mount secret as a file
- /mnt/path=sec:my-secret

scale:
  replica: 2
  auto: # automatic scale up and down ba
  range: "1-10"
  cpu: 80 # if cpu utilization is abov
  qps: 1000 # if qps is higher than 1k

canary: # Auto-create canary deployment.
  replica: 1 # canary deployment size
  headers:
  - "foo:bar.*"
```

Capability Definition 1

```
apiVersion: core.oam.dev/v1alpha2
kind: WorkloadDefinition
metadata:
  name: webservice
spec:
  definitionRef:
    name: deployments.apps
  extension:
    template: |
      parameter: #webservice
      #webservice: {
        // +vela:cli:enabled=true
        // +vela:cli:usage=specify commands to run in container
        // +vela:cli:shortc
        cmd: [...string]

        env: [...string]

        files: [...string]
      }

    output: {
      apiVersion: "apps/v1"
      kind: "Deployment"
      metadata:
        name: context.name
      spec: {
        selector: {
          matchLabels:
            app: context.name
        }
        template: {
          metadata:
            labels:
              app: context.name
          spec: {
            containers: [{
              name: context.name
              image: context.image
              command: parameter.cmd
            }]
          }
        }
      }
    }
```

Capability Definition 2

```
apiVersion: core.oam.dev/v1alpha2
kind: TraitDefinition
metadata:
  name: route
spec:
  definitionRef:
    name: routes.standard.oam.dev
  extension:
    template: |
      parameter: #route
      #route: {
        domain: string
        http: {string}: int
      }

    // trait template can have multiple outputs and they are all traits
    outputs: service: {
      apiVersion: "v1"
      kind: "Service"
      metadata:
        name: context.name
      spec: {
        selector:
          app: context.name
        ports: [
          for k, v in parameter.http {
            port: v
            targetPort: v
          }
        ]
      }
    }
```

• Simple

- Think about docker-compose but for Kubernetes.
- Designed to ship (build -> release) cloud native app by one click.

• Extensible

- Every section in Appfile references a independent capability definition

• CUE based

- The schema of each section is enforced by CUE template defines in capability definition.

Architecture of KubeVela



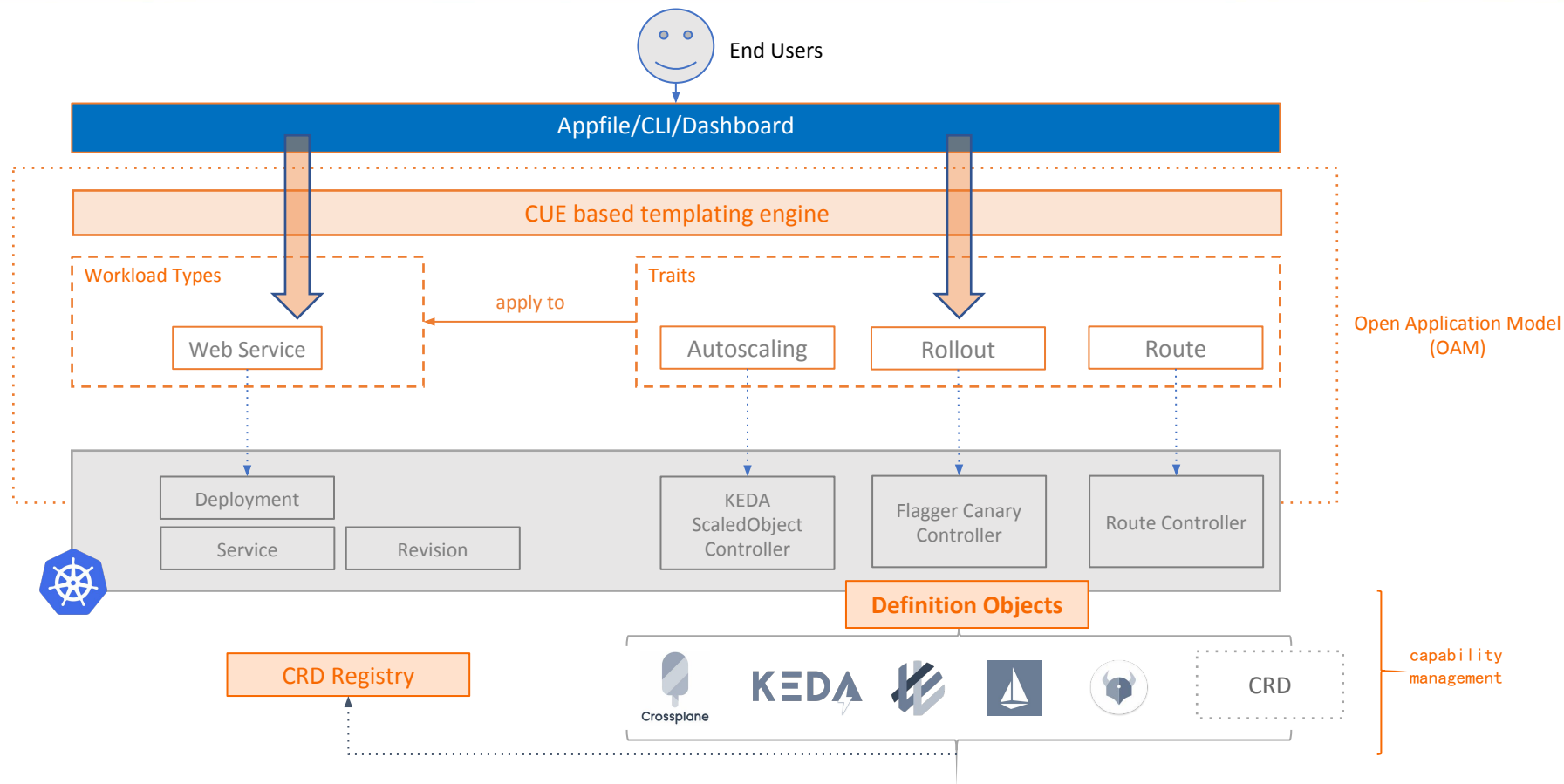
KubeCon



CloudNativeCon

North America 2020

Virtual



- **KubeVela**

- **For developers:** a app-centric platform (micro-PaaS?) to ship application to k8s cluster on any cloud or infrastructure.
- **For platform builders:** a highly extensible engine to build such app-centric platforms in k8s native approach.

- **Project Status**

- *Developer Preview* stage with features still WIP, **NOT** ready for production.
- Roadmap: <https://github.com/oam-dev/kubevela/projects/1>
 - v1.0.0 release targets at Dec. 2020
- Current feature set:
 - *Appfile, CLI, dashboard (preview)*
 - **Web Service** & **Task** workload types, **Route, Rollout** (*Flagger*) & **Autoscaling** (*KEDA*)

- **Community**

- Gitter: <https://gitter.im/oam-dev/>
- Slack: <https://cloud-native.slack.com/messages/kubevela/>

Thank you!



- **KubeVela** is initialized by open source community since day 0 with [bootstrapping contributors from 8+ different organizations](#).
- We intend to donate KubeVela to neutral foundation at early stage.

