GOTC

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

OPEN SOURCE , OPEN WORLD

开源云原生计算时代论坛

Karmada：开源的云原生多云容器编排平台

王泽锋 Kevin Wang 2021年07月10日

About Me

GOTC

华为云 云原生开源负责人

CNCF中国大使、技术监督委员会贡献者

Kubernetes社区资深Maintainer

KubeEdge、Volcano、Karmada项目创始人

2012年加入华为

2013年起参与容器平台研发

2015年起参与Kubernetes上游社区

成为国内最早的一批Kubernetes Maintainer

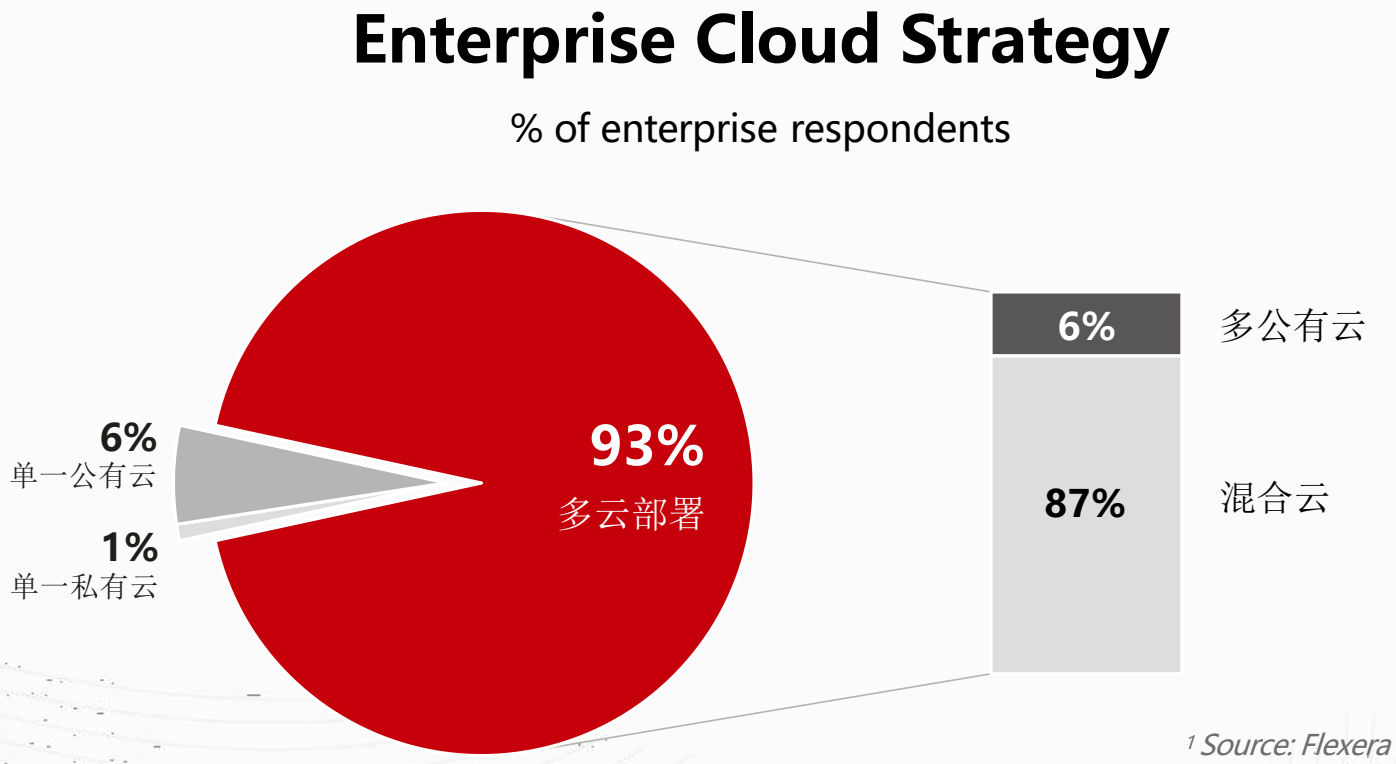
2015~2018期间主导Kubernetes社区多项高级调度特性及多个子项目设计研发

2018年~：作为联合发起人主导KubeEdge开源项目

2019年~：作为联合发起人主导Volcano开源项目

2021年~：作为发起人主导Karmada开源项目

多云、多集群部署已经成为常态



[1] Source: Flexera 2021 State of the Cloud Report
[2] N=750

调查显示，超过93%的企业正同时使用多个云厂商的服务。

云原生技术和云市场不断成熟，未来将是编程式多云管理服务的时代。

云原生多云多集群的典型阶段

GOTC

一群孤岛

- 一致的集群运维
- 一致的应用交付
- 业务割裂，互不感知
- 数据孤岛、资源孤岛、流量孤岛

We are here

威尼斯水城

- 统一应用交付（部署运维）
- 统一应用访问（流量分发）
- 统一资源分配（编排调度）
- 少量、小压力的跨集群业务访问

大航海时代

实例、数据、流量：

- 自动调度
- 自由伸缩
- 自由迁移

云原生的多云仍然充满挑战

GOTC

多云容器集群管理的挑战

集群繁多

繁琐重复的集群配置

云厂商的集群管理差异

碎片化的API访问入口

业务分散

应用在各集群的差异化配置

业务跨云访问

集群间的应用同步

集群的边界限制

资源调度受限于集群

应用可用性受限于集群

弹性伸缩受限于集群

厂商绑定

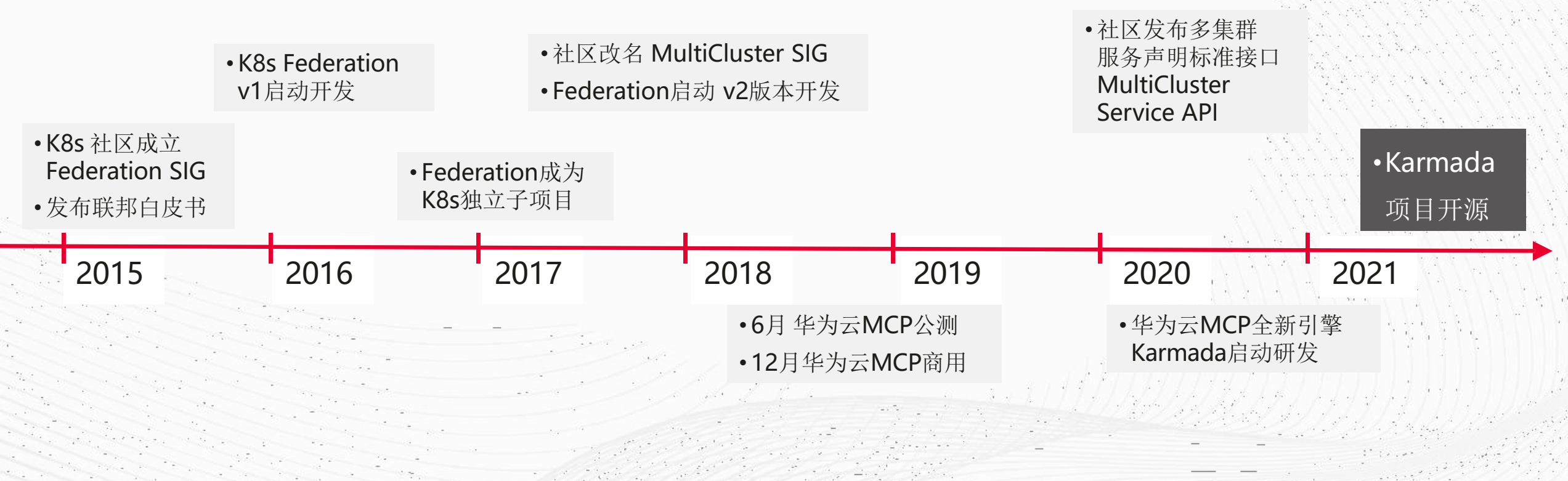
业务部署的“黏性”

缺少自动的故障迁移

缺少中立的开源多集群编排项目

多集群容器编排的前世今生

GOTC



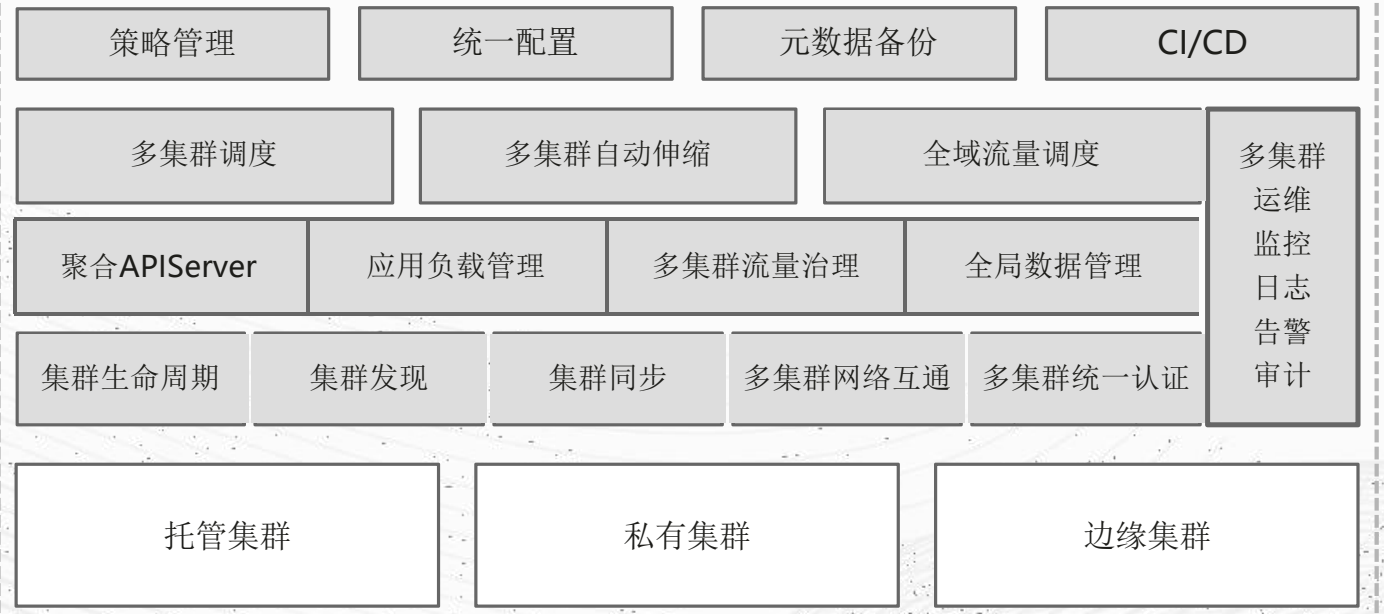
Karmada：开源的云原生多云容器编排引擎

GOTC

KARMADA

使用Karmada构建无限可扩展的容器资源池

让开发者像使用单个K8s集群一样使用多云



联合发起单位

HUAWEI　ICBC　浦发银行 SPD BANK　小红书

VIPKID　趣　中国一汽　T3出行

兼容K8s API

0代码改造升级多云架构

全网统一管理

公有云、混合云统一管理

能力开箱即用

内置10+基于行业场景的调度能力插件

https://github.com/karmada-io/karmada

GOTC

多集群应用部署

零改造 — 使用K8s原生API部署一个多集群应用

GOTC

可复用的分发策略

```yaml
apiVersion: policy.karmada.io/v1alpha1
kind: PropagationPolicy
metadata:
  name: multi-zone-replication
spec:
  resourceSelectors:
    - apiVersion: apps/v1
      kind: Deployment
      labelSelector:
        matchLabels:
          ha-mode: multi-zone-replication
  placement:
    spreadConstraints:
      - spreadByField: zone
        maxGroups: 3
        minGroups: 3
```

示例策略：为所有deployment配置多AZ的HA部署方案

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  app: nginx
  ha-mode: multi-zone-replication
spec:
  replicas: 3
  selector:
  matchLabels:
    app: nginx
  template:
  metadata:
    labels:
      app: nginx
  spec:
  containers:
    -name: nginx
    image: nginx
    ports:
    - containerPort: 80
```

原生的单集群API

使用标准的K8s API定义部署应用

```
kubectl create -f nginx-deployment.yaml
```


https://github.com/karmada-io/karmada

Propagation Policy: 可重用的应用多集群调度策略

GOTC

```yaml
apiVersion: policy.karmada.io/v1alpha1
kind: PropagationPolicy
metadata:
  name: example-policy
spec:
  resourceSelectors:
    - apiVersion: apps/v1
      kind: Deployment
      name: deployment-1
      labelSelector:          # standard labelSelector
  propagateDependensies: false
  placement:
    clusterAffinity:
    clusterNames:
        - cluster1
        - cluster3
    clusterTolerations:       # like pod tolerations
    spreadConstraints:
      - spreadByLabel: failuredomain.kubernetes.io/zone
        maxGroups: 3
        minGroups: 3
  schedulerName: default
```

resourceSelector

- 支持关联多种资源类型
- 支持使用 *name* 或 *labelSelector* 进行对象筛选

placement

- *clusterAffinity*:
  - 定义倾向调度的目标集群
  - 支持通过 names 或 labelselector 筛选
- *clusterTolerations*:
  - 类似单集群中Pod tolerations和 node taints
- *spreadConstraints*:
  - 定义应用分发的HA策略
  - 支持对集群动态分组：按Region、AZ、特性label分组，实现不同层级的HA

Override Policy: 跨集群可重用的差异化配置策略

GOTC

```yaml
apiVersion: policy.karmada.io/v1alpha1
kind: OverridePolicy
metadata:
  name: example-override
namespace: default
spec:
  resourceSelectors:
    - apiVersion: apps/v1
      kind: Deployment
  targetCluster:
    labelSelector:
      matchLabels:
        failuredomain.kubernetes.io/region: dc1
  overriders:
    imageOverrider:
    - component: prefix
      operator: replace
      value: "dc-1.registry.io"
```

resourceSelector

- 支持使用 *name* 或 *labelSelector* 进行对象筛选

overriders

- 支持多种override插件类型
- *plainTextOverrider* :
  - 基础插件，纯文本操作替换
- *imageOverrider*:
  - 针对容器镜像的差异化配置插件

Clusters API：用户自助可查的资源池基本单元

GOTC

```yaml
apiVersion: cluster.karmada.io/v1alpha1
kind: Cluster
metadata:
  name: member-cluster-1
spec:
  syncMode: Push
  apiEndpoint: https://172.17.0.5:6443
  secretRef:
    name: member-cluster-1
  namespace: karmada-cluster
  provider: huaweicloud
  region: ap-southeast-1
  zone: az-1
  taints:                # just like node taints
status:
  conditions:
  - message: "/healthz responded with  ok"
    reason: ClusterReady
    status: "True"
    type: ClusterReady
  kubernetesVersion: v1.17.0
  apiEnablements: []
  nodeSummary: {}
  resourceSummary: {}
```

syncMode

- 支持使用 *Push* 或 *Pull* 模式与集群进行同步

secretRef

- 分离*Push*模式下集群访问凭据，便于开放 clusters API 供用户自助查询

taints

- 集群级别taint - toleration机制，支持集群级资源预留及驱逐

kubernetesVersion, apiEnablements

- K8s版本，集群开启的API列表，支持基于API依赖的调度

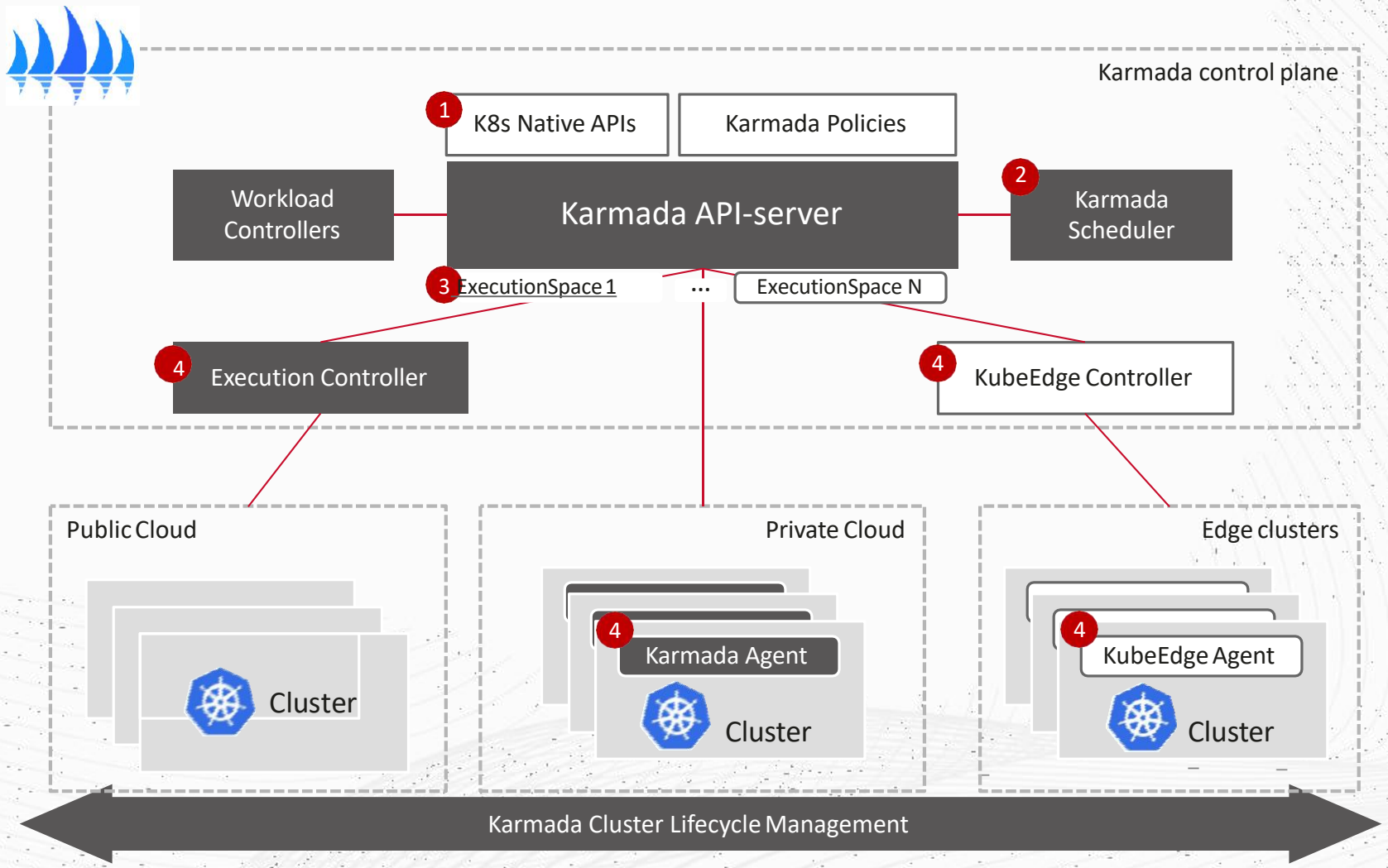
resourceSummary

- 集群资源信息（容量、使用量、调度中），

GOTC

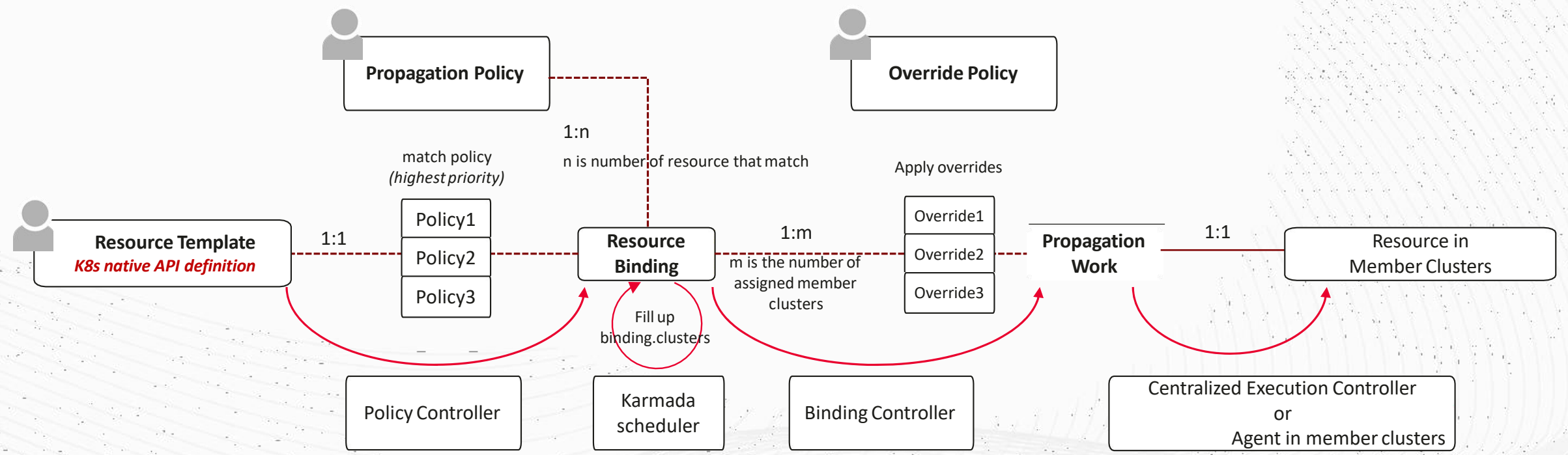
Karmada工作原理

Karmada 架构



https://github.com/karmada-io/karmada

Karmada 工作原理



Karmada 社区路标

GOTC

2021 Q1

- K8s原生API部署多集群应用
- 多集群应用状态聚合
- 多集群部署时，应用配置差异化
- 针对集群的亲和性调度
- 多集群高可用调度策略
- 纳管集群
- 集群目录

2021 Q2

- 多集群的应用故障迁移
- 根据多种策略动态拆分应用实例数
- 针对集群的Taint Toleration
- 多种集群同步模式：Push，Pull
- 集群生命周期管理

2021 Q3

- 多集群服务发现
- 多集群外部流量接入
- 多集群的应用自动伸缩
- 多集群的资源重平衡
- 更多高可用调度策略：多Region/多AZ/多厂商
- 聚合API访问

2021 Q4

- 多形态下多集群容器网络
- 多集群服务治理
- 多集群监控
- 多集群日志收集
- GitOps支持

Take Away

- 多云已成必然
  - 云原生技术与多云诉求相互促进
- 多云的三个阶段
  - 标准技术栈，互操作的多个孤岛
  - 统一平台，云间统一调度，统一弹性
  - 多云无缝合一
- 云原生多云的典型挑战
  - 集群繁多
  - 业务分散、碎片化
  - K8s集群造成的边界
  - 厂商绑定

- Karmada项目核心价值
  - K8s原生API兼容，丰富云原生生态
  - 内嵌策略，开箱即用
  - 丰富的多集群调度支持
  - 集群资源空间隔离
  - 多种模式集群同步，屏蔽地域、网络限制
- Karmada后续计划
  - 整体技术栈Q4成型

加入社区

GOTC

https://github.com/karmada-io/karmada

https://karmada-io.slack.com

https://space.bilibili.com/1626207655

容器魔方公众号

每日推送图文

社区最新动态

直播课程、技术干货

扫码添加小助手

发送“karmada”加群

社区专家入驻

技术问题随时答疑