TOP1 100

主办方 msup

中间件同城多云高可用平台建设

孟祥勇 云原生资深架构师

讲师简介

孟祥勇

资深云原生架构师

- 目前任网易数帆云原生中间件资深架构师，曾就职京东、同盾，Kubernetes社区贡献者，专注调度、中间件、kubeflow领域。
- 负责网易数帆中间件集群联邦平台建设，已支持Redis、ES、Kafka、Rocketmq、ZK中间件的联邦，在多家银行、证券成功落地使用。
- 负责网易数帆中间件Finops系统建设，已具备成本展示、多种算法资源预测、负载感知调度与重调度能力。

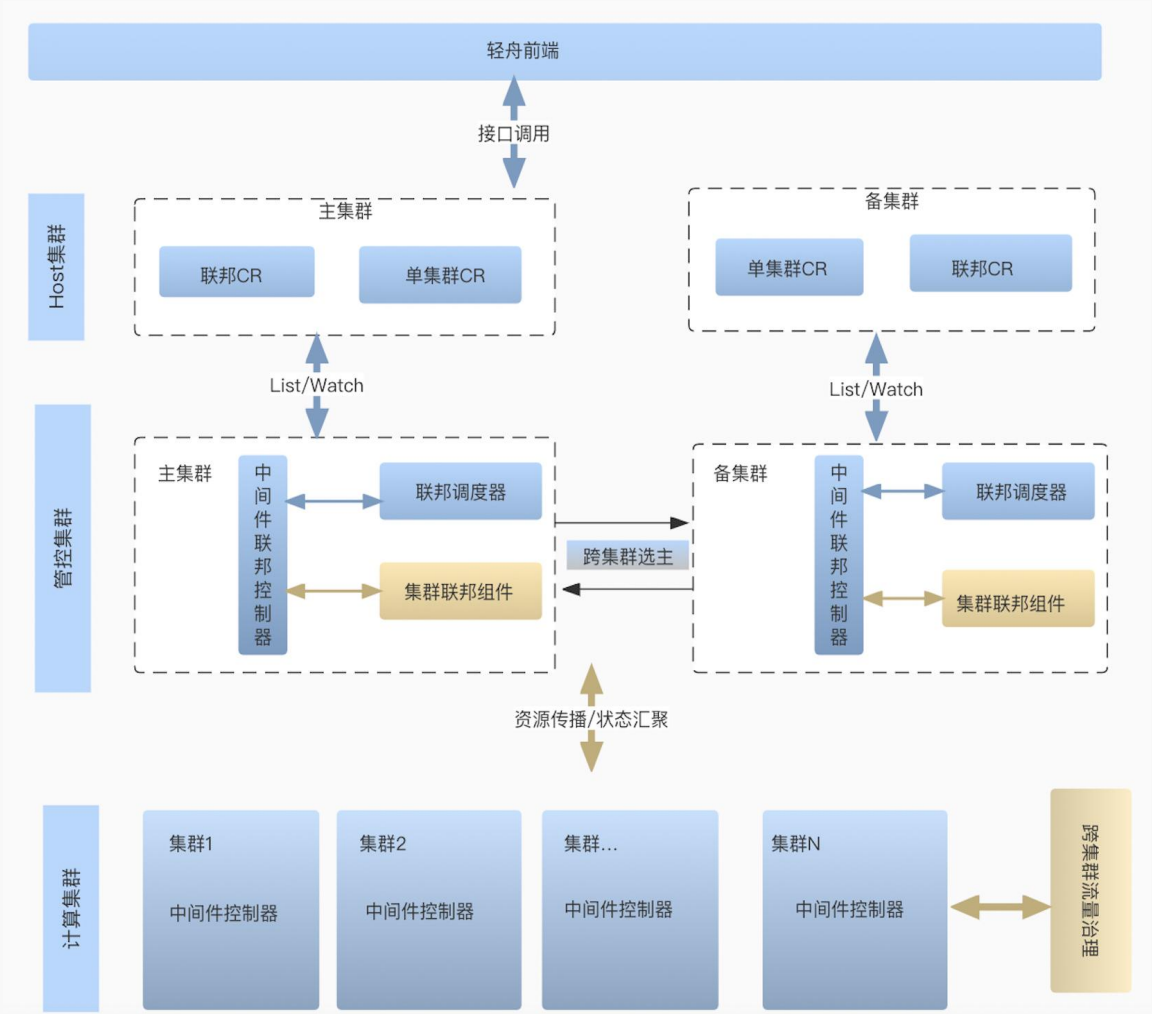
目录

PART 01

亮点介绍

亮点介绍

轻舟中间件集群联邦架构



灵 活

丰富的调度策略

轻 量

同步资源少

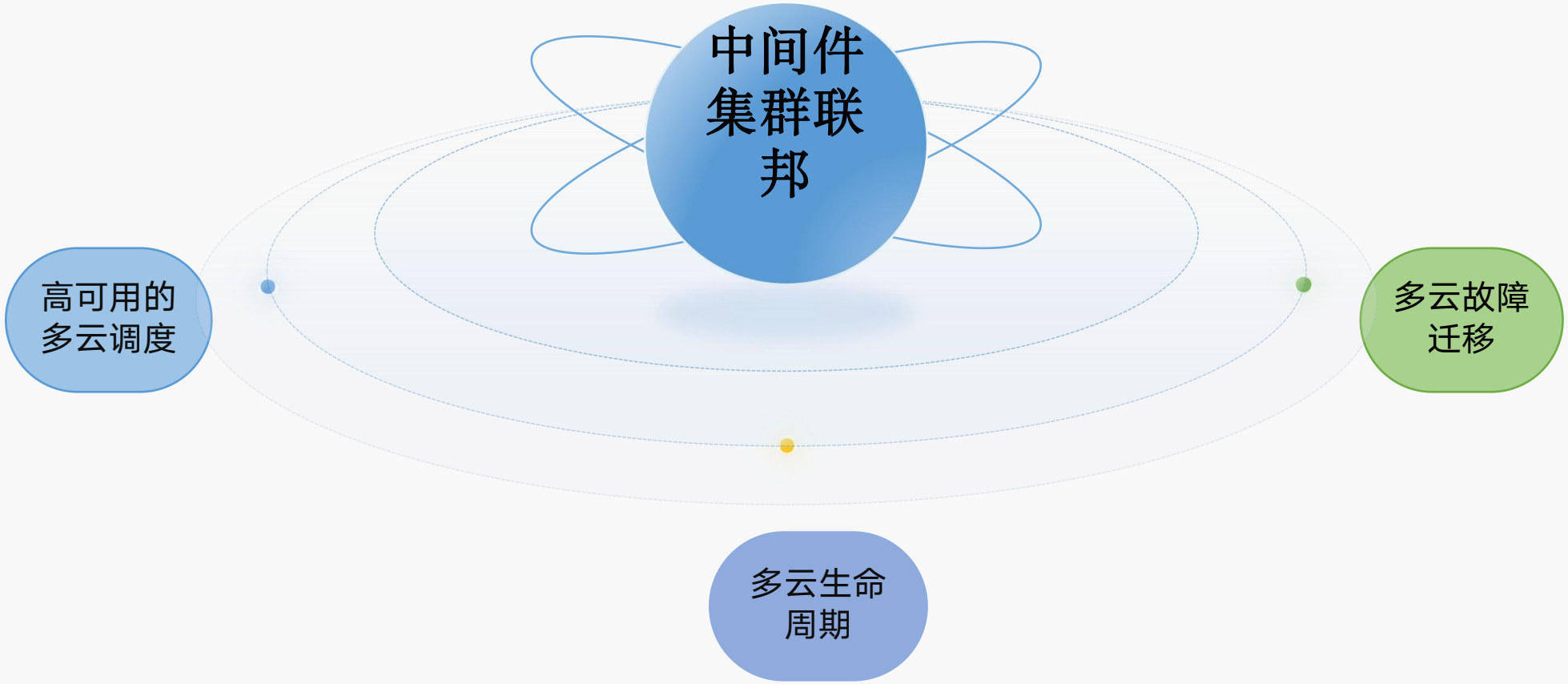
高可用

管控高可用、计算集群自治

侵入小

对管控、监控、日志侵入少

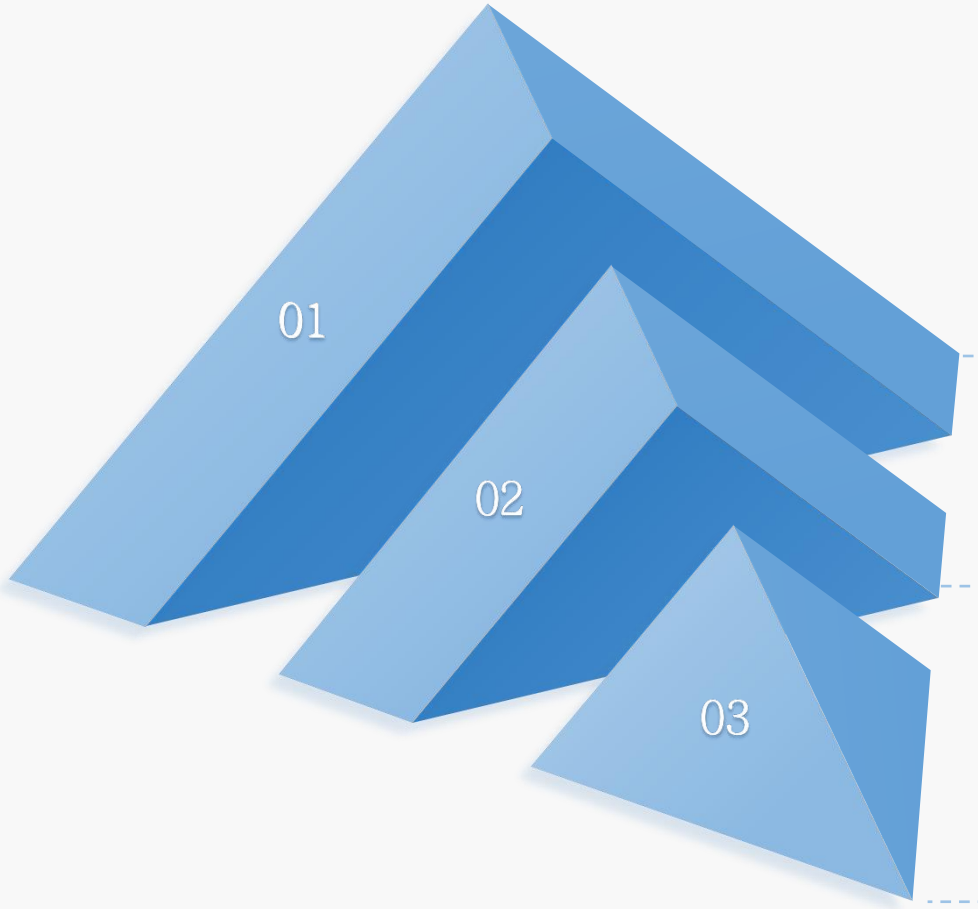
亮点介绍



PART 02

案例背景

案例背景



业务的创新、快速迭代对中间件生命周期管理、高可用提出了更高的要求

云原生技术的广泛的落地，为中间件满足业务发展提供了有力支撑

多云管理技术为中间件集群提供同城多机房高可用。减少因为一个机房故障(网络、存储)后，对中间件集群SLA的影响

问题与挑战

组件多

云原生领域多云管理组件多，如何挑选出适合轻舟中间件现有架构的多云管理组件

有状态应用

多云管理组件支持无状态应用，对于中间件这种有状态应用支持不太友好

集群故障

当其中一个集群故障时，如何安全快速的将故障实例迁移到新的集群

PART 03

破题思路与成果

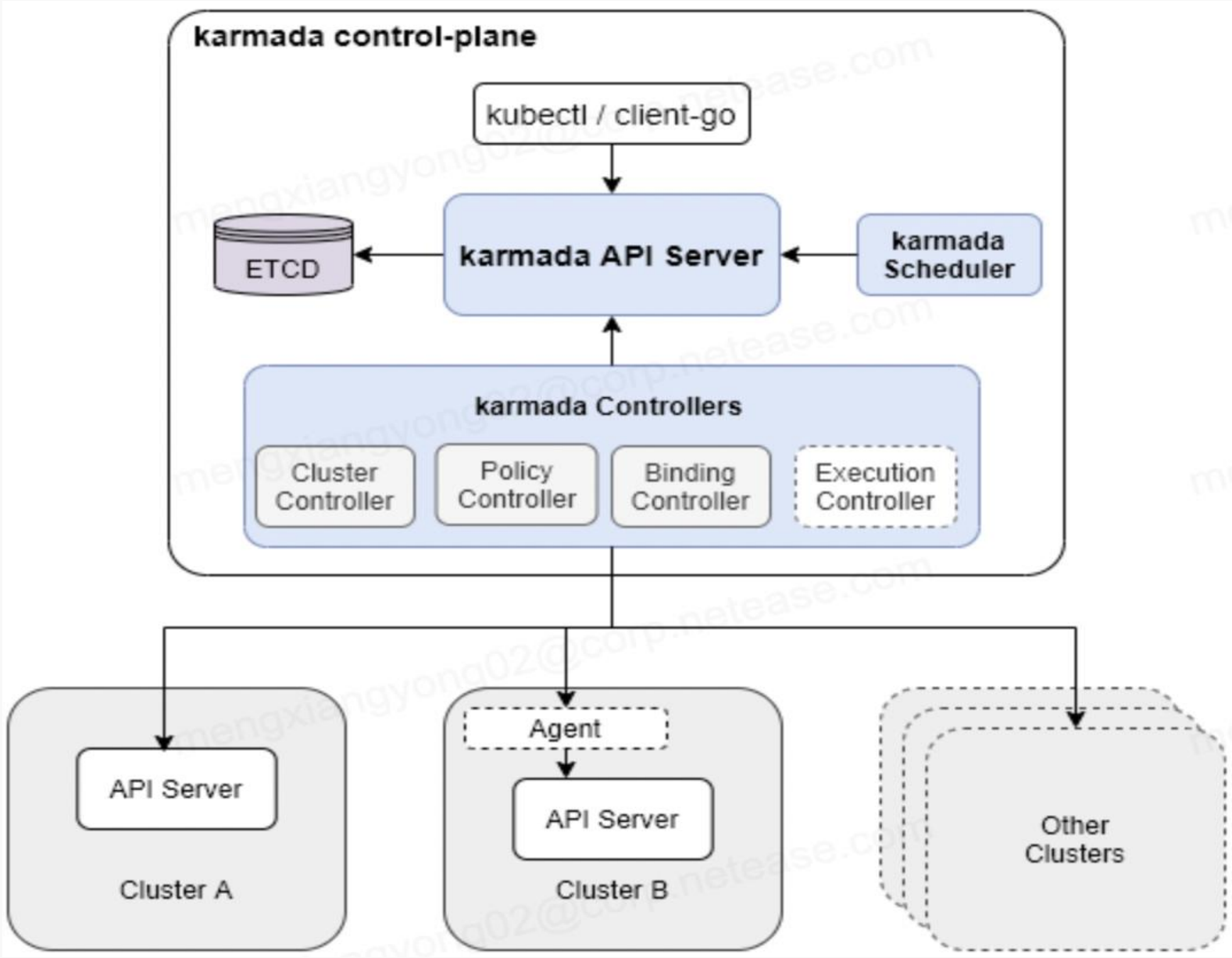
破题思路

- Karmada将资源模版通过覆盖策略、分发策略在集群之间进行差异化传播。这里的资源模版可以是k8s原生资源亦或者自定义资源(CR)。karmada调度功能如下：

集群亲和	实例拓扑	集群拓扑	污点容忍
• 集群名 • label匹配	• 直接复制 • 权重	• 数量约束	• 专用集群

破题思路

Karmada架构如右图：



组件介绍

- 联邦调度器 → 执行创建、扩缩容、故障调度
- 联邦控制器 → 负责跨集群滚动创建、更新、全局视图维护
- 集群联邦组件 → 实现资源的差异化传播与状态收集
- 跨集群治理流量 → 实现高可用、高性能跨集群域名解析

组件介绍-联邦调度器

中间件联邦调度需求

①支持CR级别的调度；

②支持按集群资源余量调度；

③支持redis、elasticsearch、rocketmq跨集群调度需求；

组件介绍-联邦调度器

调度单位：与Hived、Volcano中的组相似，

一个中间CR包含很多组件信息，比如分片、

主从、角色，这些组件需要在一次调度中调度

完成，有一个无法调度就视为调度失败，不再

进行后续的调协逻辑。联邦调度器中的组信息

可以参考右图

```yaml
spec:
  replicaRequirements:
    - replicas: 3
      role: "master"
      maxSkew: 1
      resourceRequest:
        limits:
          cpu: "1"
          memory: 2Gi
        requests:
          cpu: "1"
          memory: 1Gi
    - replicas: 3
      role: "slave"
      shard: 1
      maxSkew: 1
      resourceRequest:
        limits:
          cpu: "4"
          memory: 8Gi
        requests:
          cpu: "2"
          memory: 4Gi
  clusterAffinity:
    labelSelector:
      cluster: compute
    clustersNames:
      - user1
      - user2
```

组件介绍-联邦调度器

半数约束：部署计算集群数量为N，实例数为M时，则实例分布会有以下几种情况：

M>N时

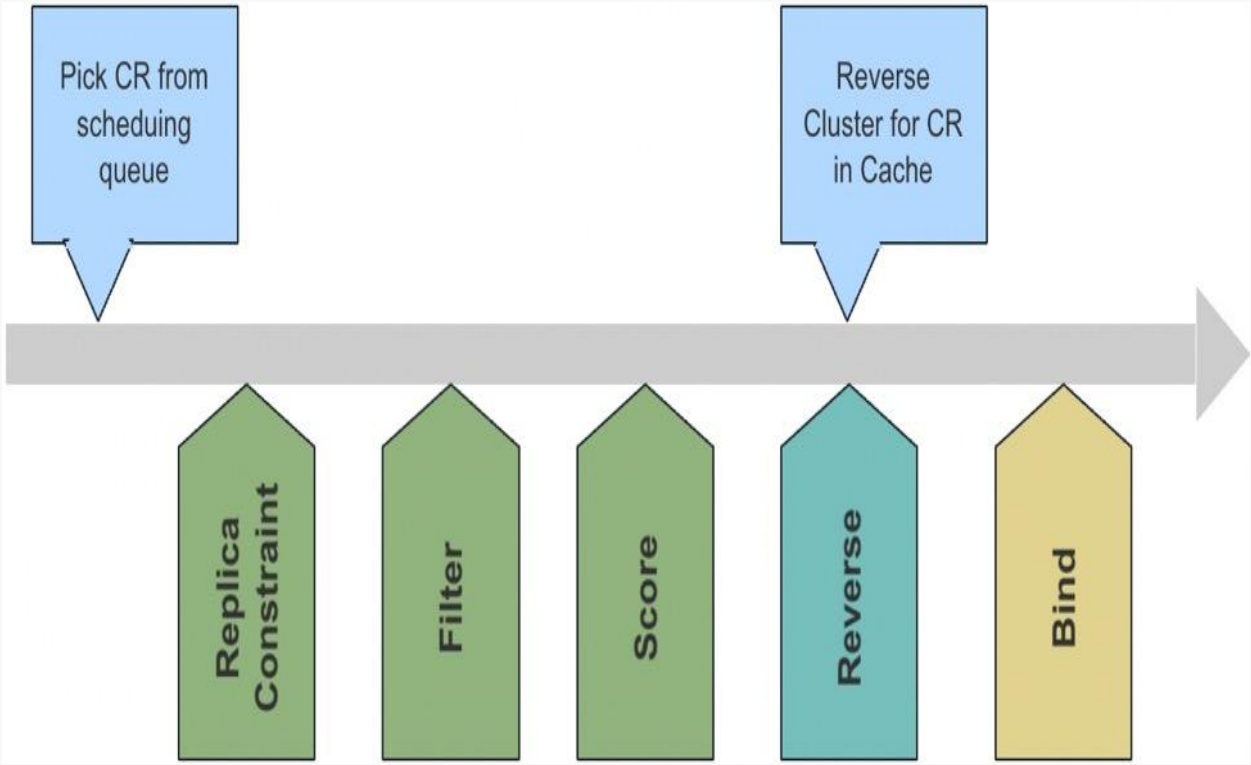
每个计算集群分配M/N个，剩余M%N个实例，再平分到每个计算集群。

M < N时

M个计算集群实例数为1，N-M个计算集群实例数为0

N=1时

所有实例在一个集群，主要是用在lb或者pod网络不满足，又想调试联邦功能。



组件介绍-联邦调度器

扩缩容约束

根据半数约束，计算扩容之后实例约束。根据新的实例分布、以及老的实例分布计算出所有可能扩容组合。以一个副本数3(c1计算集群1、c2计算集群1、c3计算集群1)扩容到7的过程说明一下扩容约束：

根据上述1.1半数约束副本数为7实例分布(2, 2, 3)

根据新老实例分布(c1, c2, c3)(1, 1, 1)，计算扩容所有可能的组合(c1, c2, c3)(1, 1, 2)、(c1, c2, c3)(1, 2, 1)、(c1, c2, c3)(1, 1, 2)

针对上述所有可能的组合走一次过滤、优选算法，主要就是资源余量算法

组件介绍-联邦控制器

全局视图

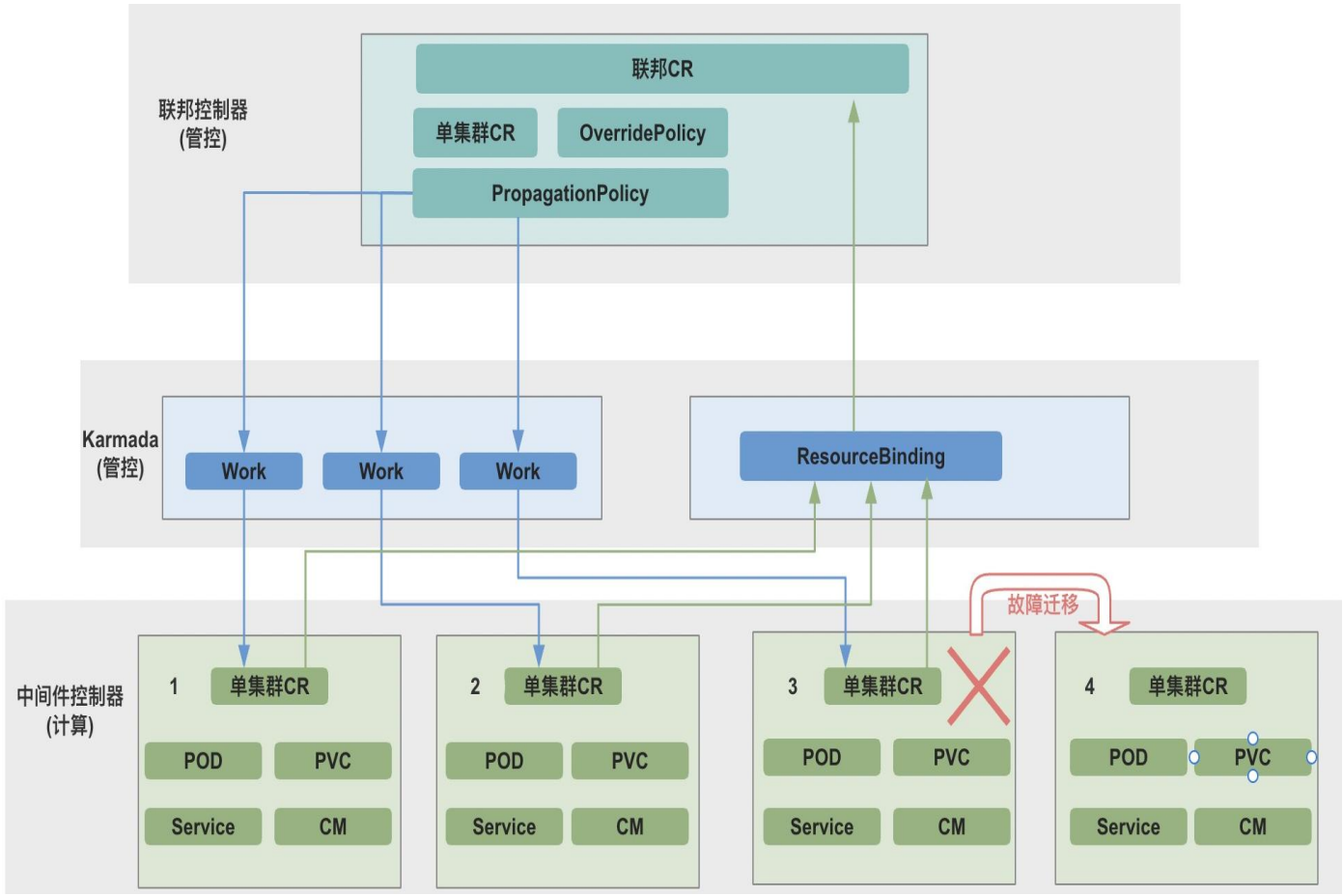
所有实例信息汇聚下发

滚动操作

按集群滚动创建、更新

故障迁移

根据故障调度结果，在新的集群创建中间件实例



组件介绍-联邦控制器

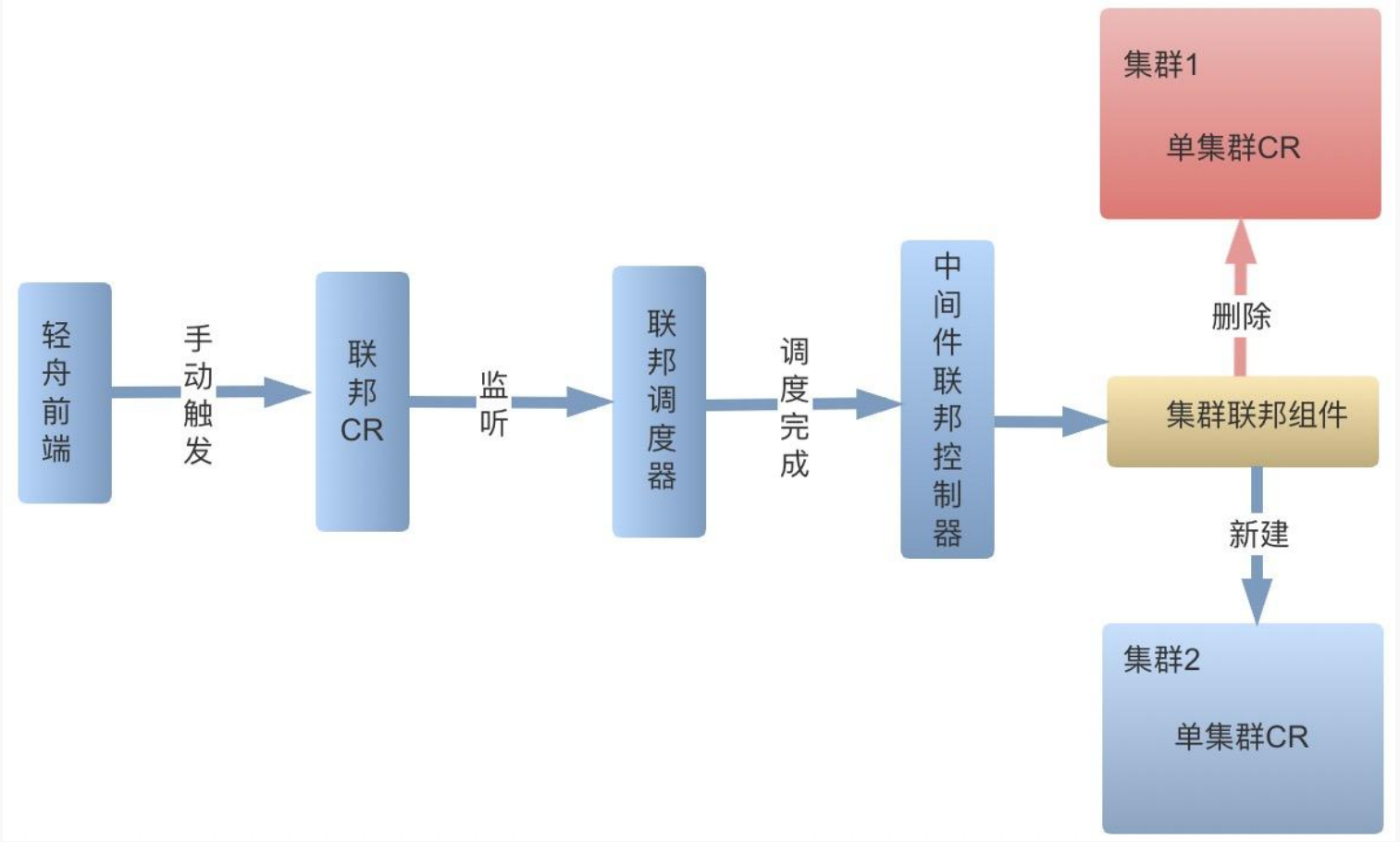
故障迁移

1、用户前端手动变更部署位置

2、联邦调度器进行故障调度

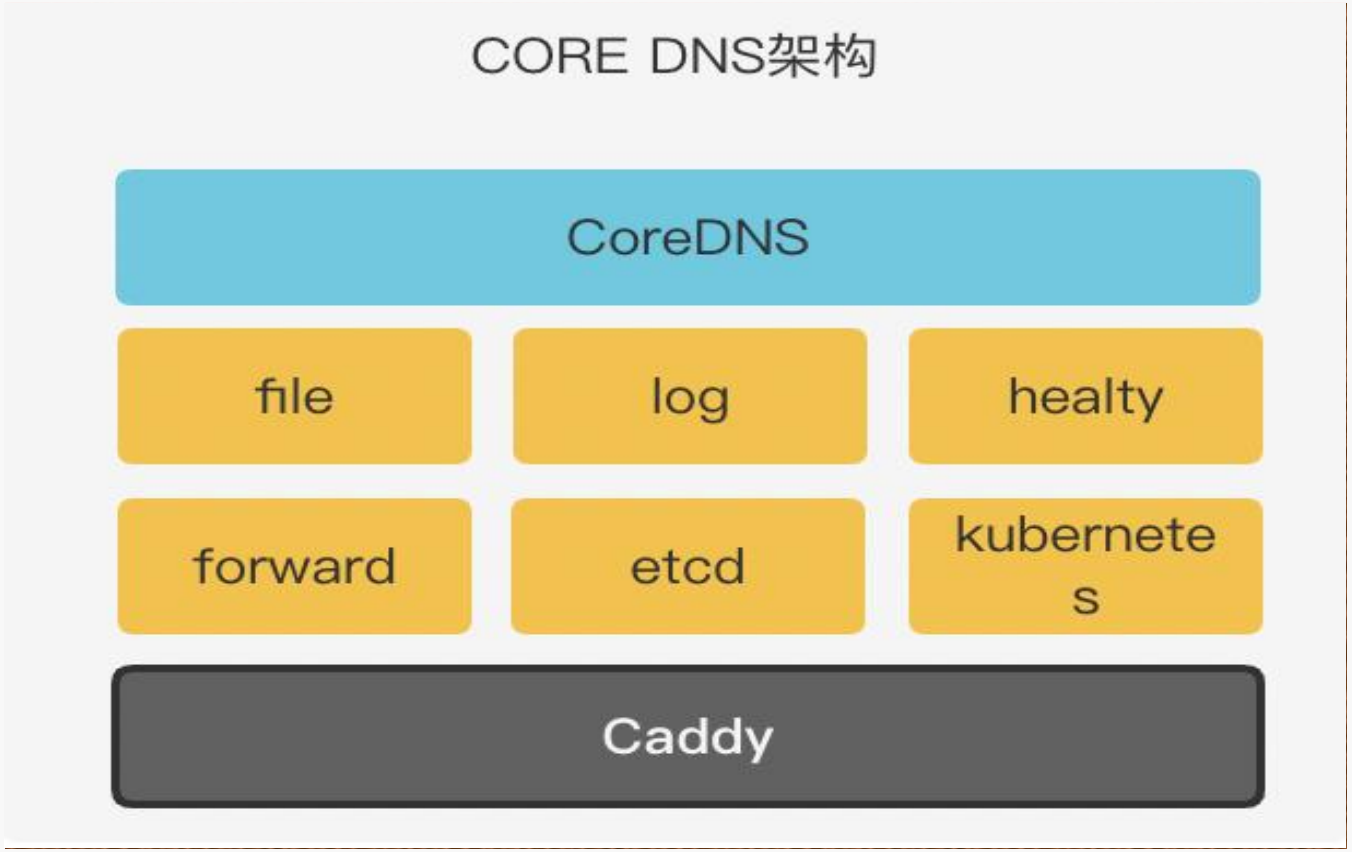
3、联邦控制器进行故障迁移逻辑

- 重新生成Baseid起始ID
- 生成新计算集群对应覆盖策略
- 修改分发策略计算集群信息



组件介绍-跨集群流量治理

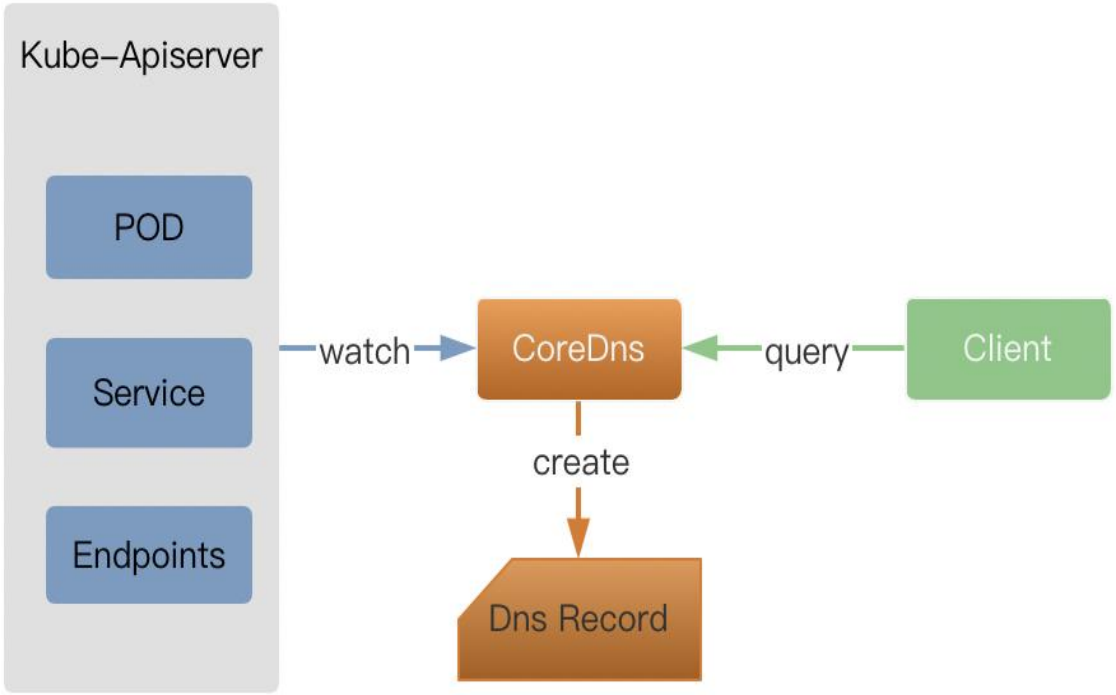
跨集群部署后导致原有的单集群域名解析和服务暴露的方式发生变化，从服务发现、域名解析两个角度进行跨集群流量治理说明。



组件介绍-跨集群流量治理

服务发现：CoreDNS中的kubernetes plugin中的工作原理如右图所示，实现了[k8s规定的域名解析规范]。针对需要跨集群访问service，创建时不要带上selector，根据全局拓扑构建endpoints，在构建endpoints时指定hostname如下：

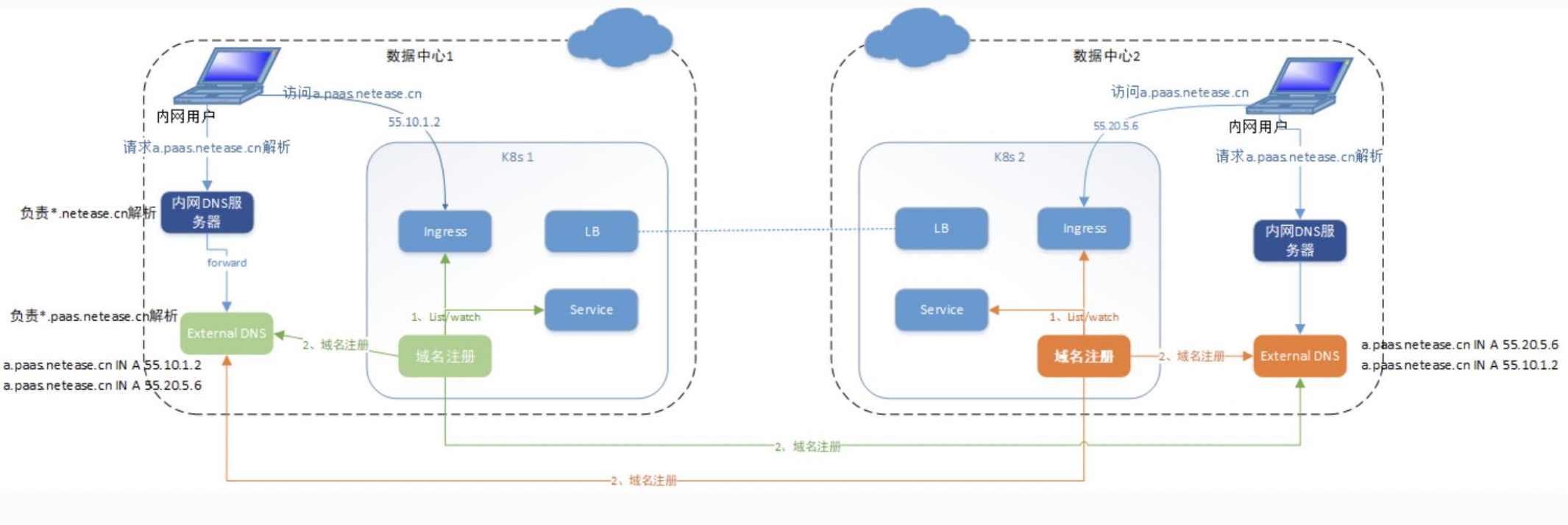
```
apiVersion: v1
kind: Endpoints
metadata:
  name: zookeeper
subsets:
- addresses:
  - hostname: dummy-hostname
    ip: 8.8.8.8
  - hostname: zookeeper-cluster-54b7cf7dbf-statusfulset-0
    ip: 10.160.228.246
  - hostname: zookeeper-cluster-6dc949d566-statusfulset-0
    ip: 10.160.225.50
  - hostname: zookeeper-cluster-7855bb8699-statusfulset-0
    ip: 10.160.225.189
```



组件介绍-跨集群流量治理

域名解析：Domain-Register重写coredns的etcd plugin。主要有一下两个组件：

- 域名注册：监听单个集群中service、ingress资源，向多个集群的external DNS自动注册/更新域
- External DNS：支持按机房就近解析，在中间件集群联邦场景下暂时不需要该功能。



成果展示

1、已在银行、证券、工业等行业落地，越来越多的S级应用使用联邦中间件集群

2、定期对机房进行故障演练，减少因为机房级别故障导致的业务不可用时长

PART 04

案例复盘与总结

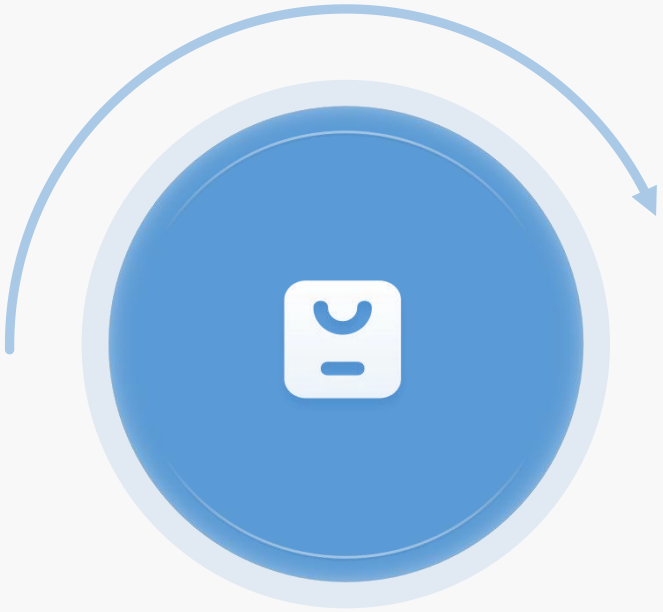
案例复盘与总结

在跨集群调度上熟悉karmada调度能力、kubernetes原生调度能力、结合中间件特有属性自研跨集群调度器

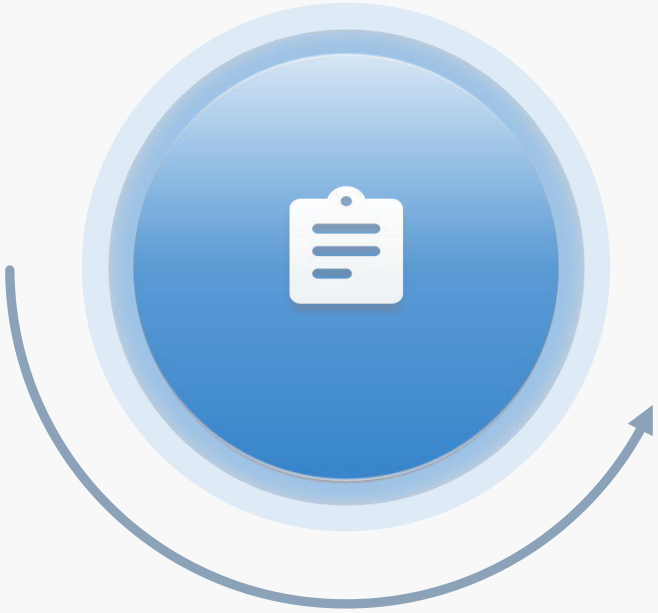
在联邦组件选型上深入对比karmada、Virtual Kubelet等开源组件

在对中间件熟悉的基础上、提出了便捷的、系统的跨集群流量治理

案例启示



基于云原生的业务创新，除了要懂业务模型、也要对云原生相关技术有深度、广度积累

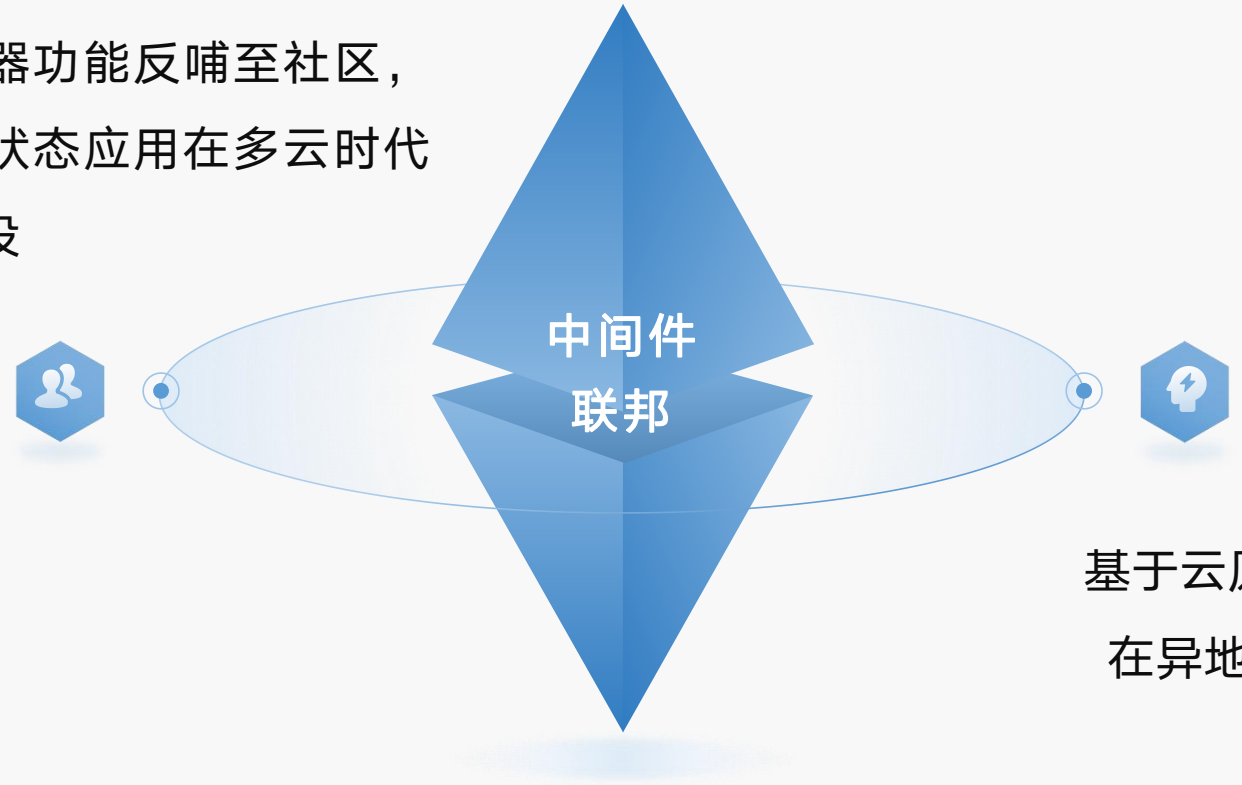


当组件发展成熟，在进行业务架构设计时要试着在组件上层做抽象，不要涉及对组件本身进行改造

PART 05
下一步计划

下一步计划

将自研调度器功能反哺至社区，
一起推动有状态应用在多云时代
的高可用建设



基于云原生将联邦集群应用
在异地多活、单元化场景

TOP100

主办方 msup

微信官方公众号：壹佰案例
关注查看更多年度实践案例