



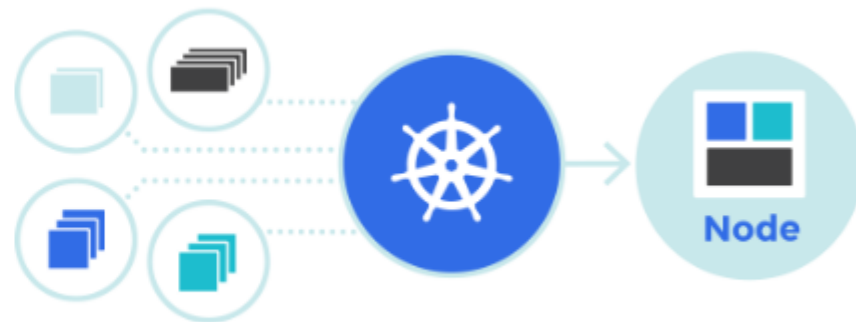
K8S 是什么



什么是 K8S

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

It groups containers that make up an application into logical units for easy management and discovery. Kubernetes builds upon [15 years of experience of running production workloads at Google](#), combined with best-of-breed ideas and practices from the community.



- 古希腊语：舵手，领航员，船长
- 官方站点： <https://kubernetes.io/>

K8S 主要特性

- 自动化部署
- 系统自愈
- 水平扩展
- 服务发现和负载均衡
- 自动更新和回滚
- 密钥和配置管理
- 存储挂载
- 批量作业执行



Thank You

石志国 jasonshi@yunify.com



K8S 与 KubeSphere 的关系



Kubernetes 发行版

底层基于原生

Kubernetes, 上层封装多

种服务组件, 能在

Kubernetes 之上提供更

多 value-add 的分布式容

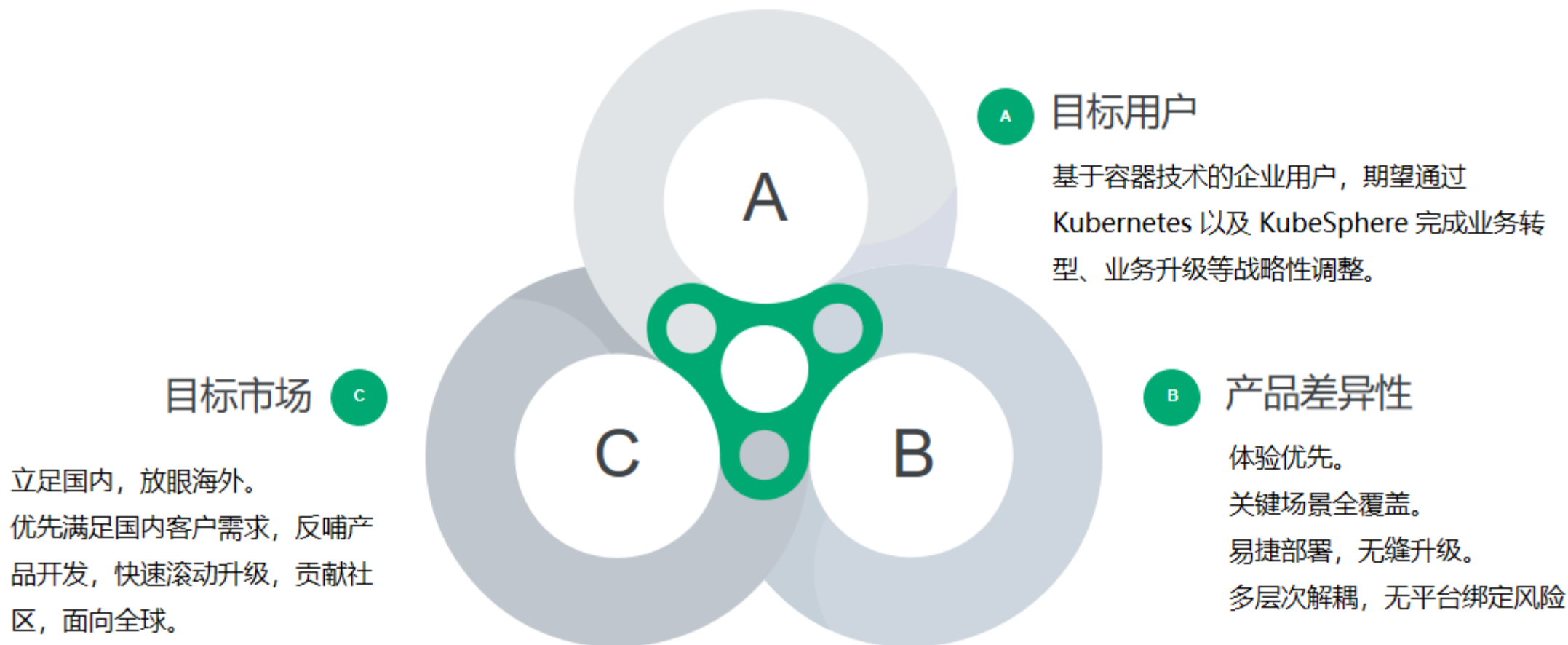
器编排管理平台。

Certified Kubernetes - Distribution



QINGCLOUD

KubeSphere 定位





Thank You

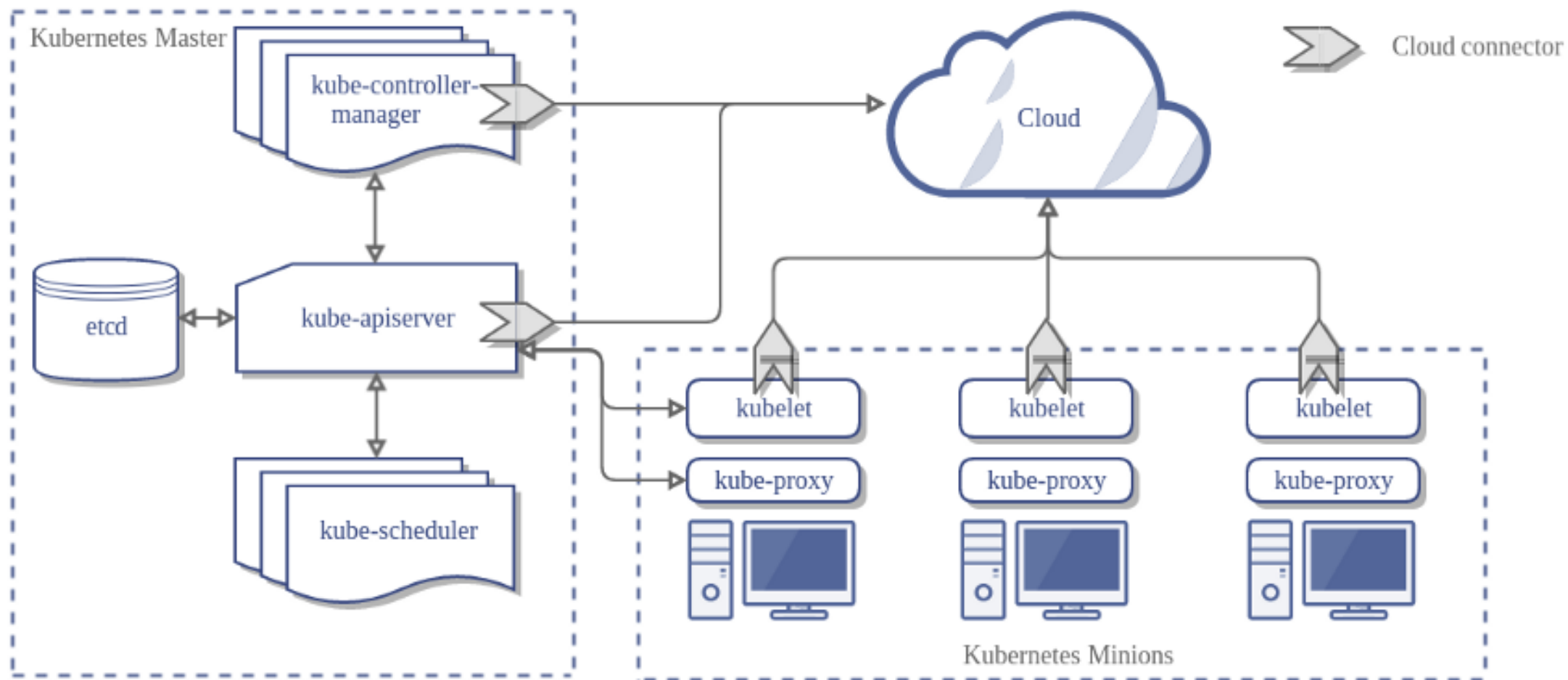
石志国 jasonshi@yunify.com



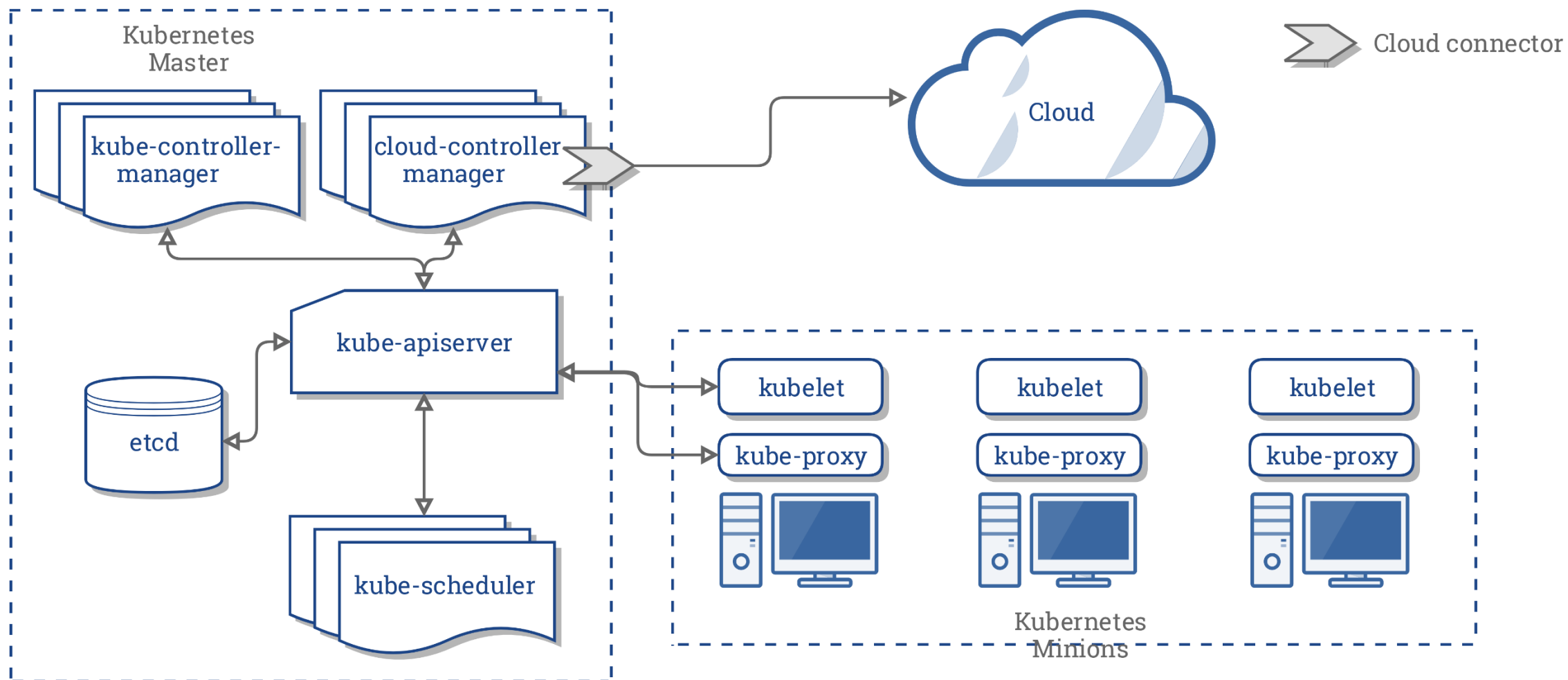
Kubernetes 架构



Kubernetes Architecture



Kubernetes Architecture with CCM



Kubernetes 分层架构思想

- 核心层
- 应用层
- 管理层
- 接口层
- 生态系统
 - Kubernetes 外部
 - Kubernetes 内部



Thank You

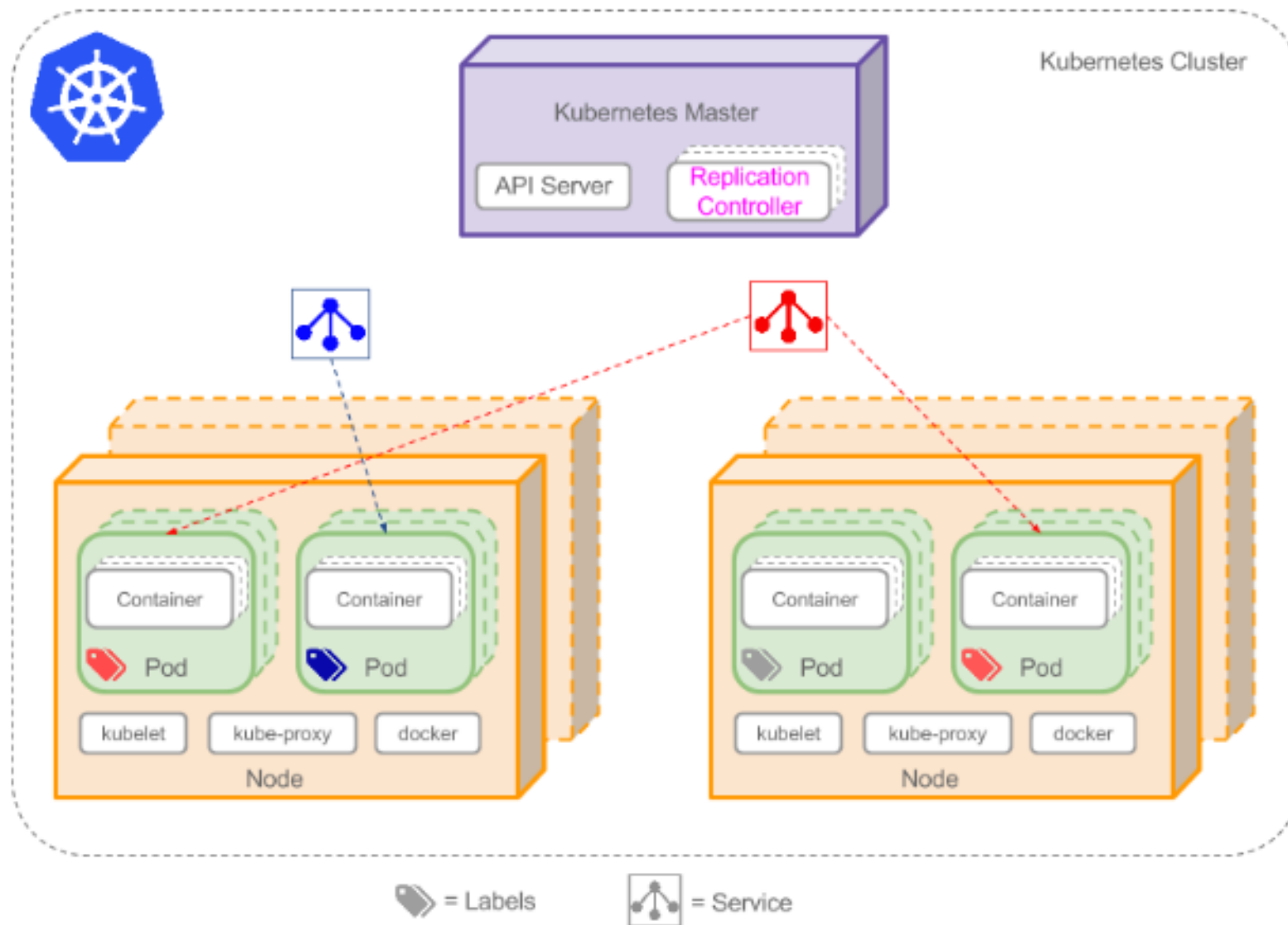
石志国 jasonshi@yunify.com



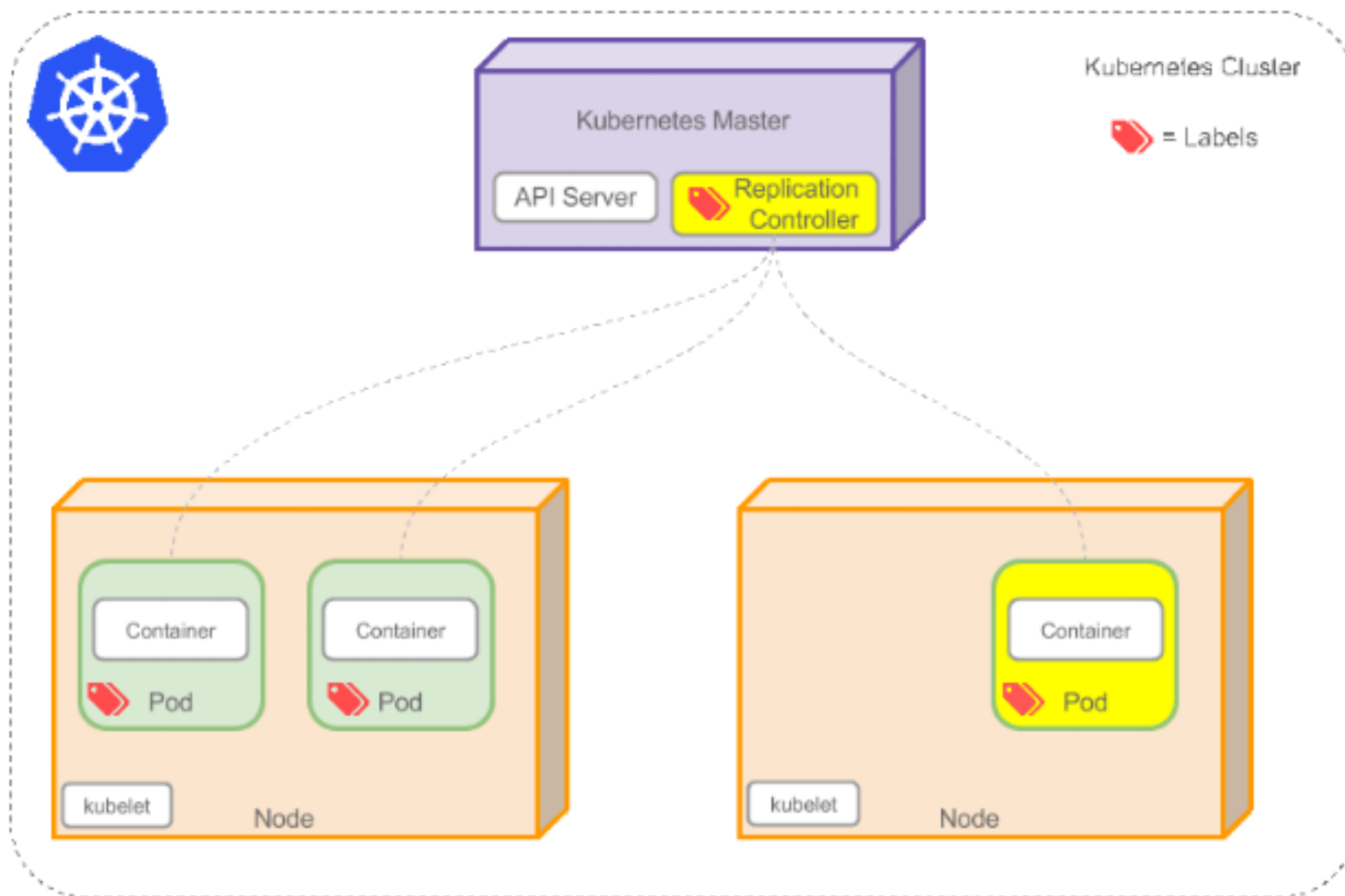
Kubernetes 基本概念 - 1



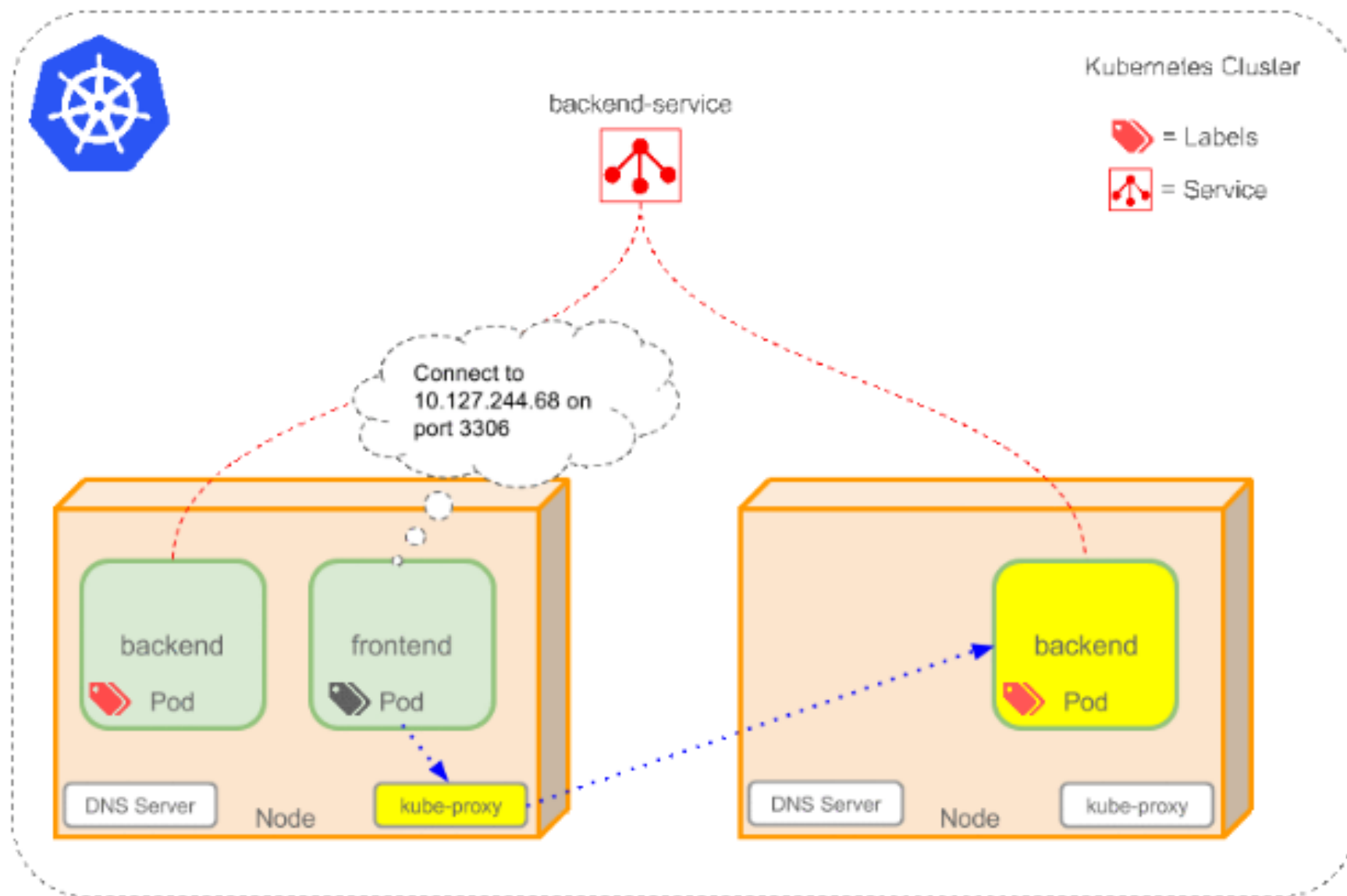
Kubernetes 基本概念 – 集群/Pod/Label



Kubernetes 基本概念 – Deployment / RC



Kubernetes 基本概念 – Service





Thank You

石志国 jasonshi@yunify.com



Kubernetes 基本概念 - 2



Controllers

Kubernetes contains a number of higher-level abstractions called Controllers. Controllers build upon the basic objects, and provide additional functionality and convenience features. They include:

- ReplicaSet
- Deployment
- StatefulSet
- DaemonSet
- Job



Thank You

石志国 jasonshi@yunify.com



Kubernetes 基本概念 - 3



What is Ingress ?

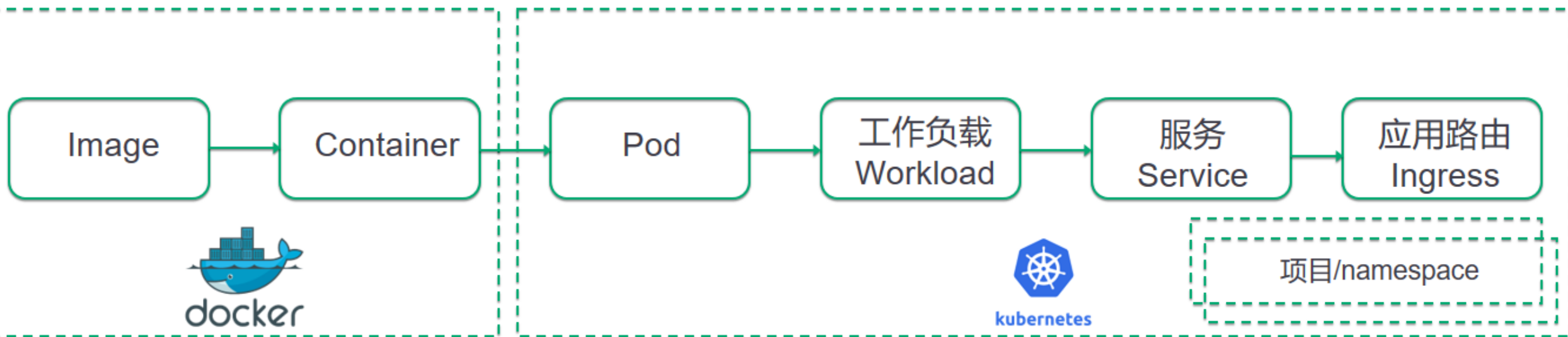
Typically, services and pods have IPs only routable by the cluster network. All traffic that ends up at an edge router is either dropped or forwarded elsewhere. Conceptually, this might look like:



An Ingress is a collection of rules that allow inbound connections to reach the cluster services.



Pod/Service/Ingress 的关系



Deployment 示例

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.7.9
          ports:
            - containerPort: 80
```

kubectl create -f nginx-deployment.yaml



Thank You

石志国 jasonshi@yunify.com

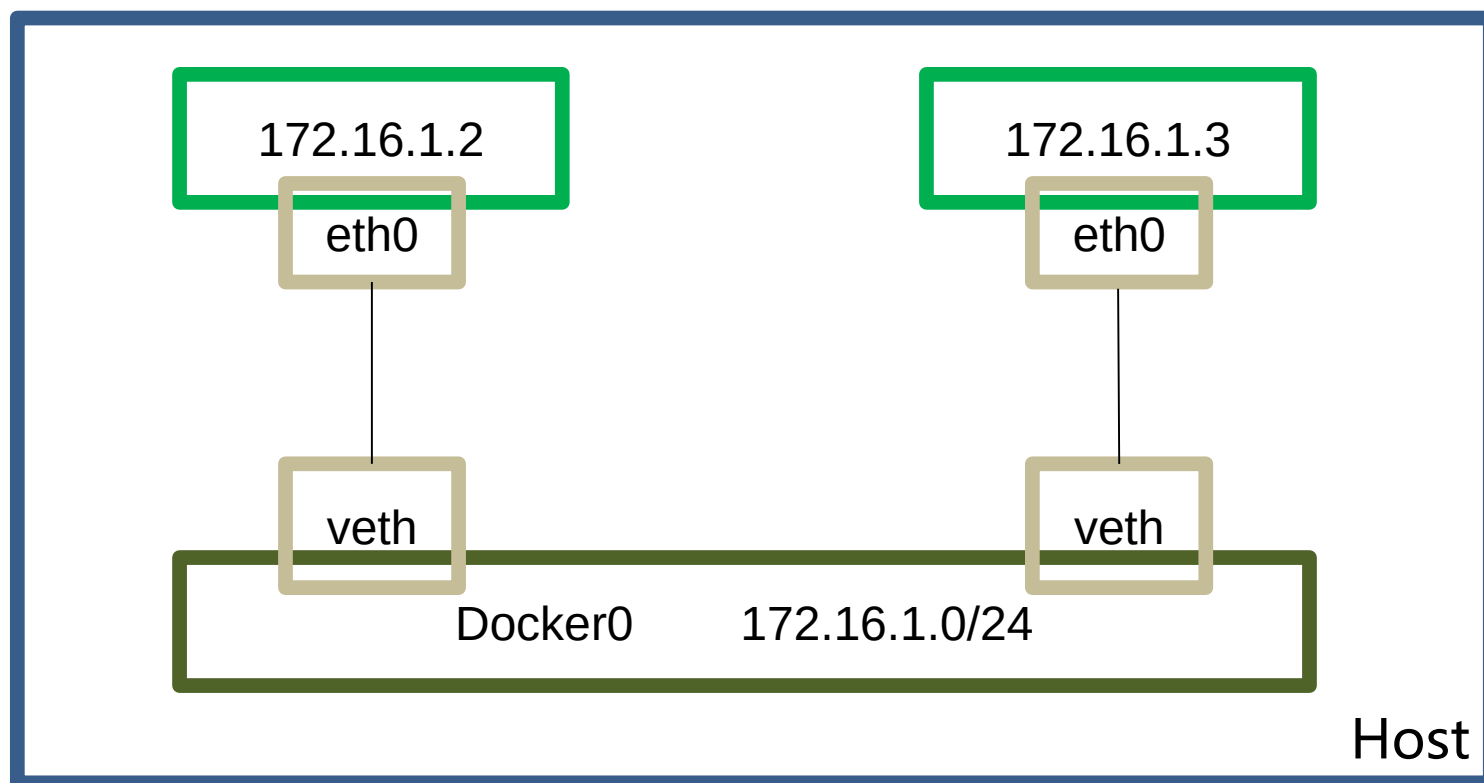


Kubernetes Networking



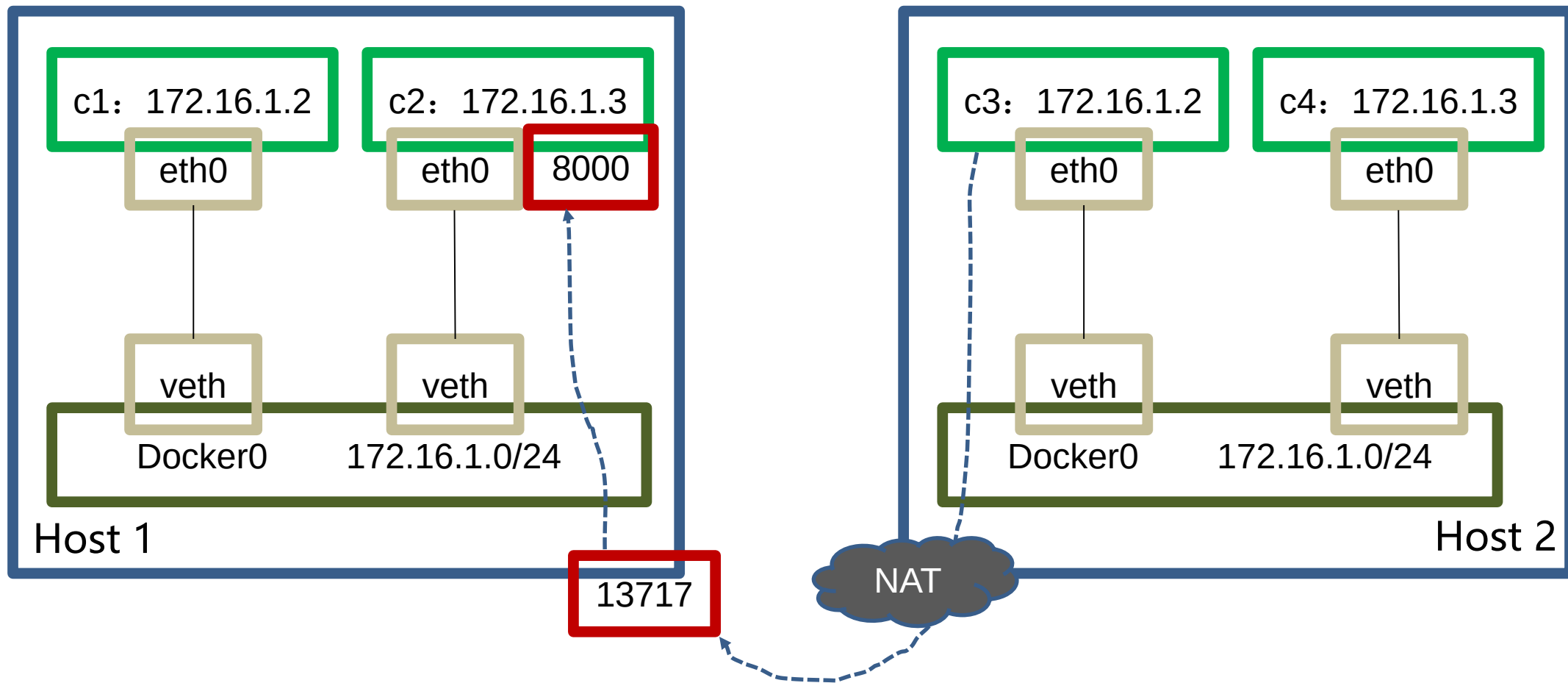
Docker 网络工作流程

Bridge 模式



Docker 网络工作流程

跨主机通信



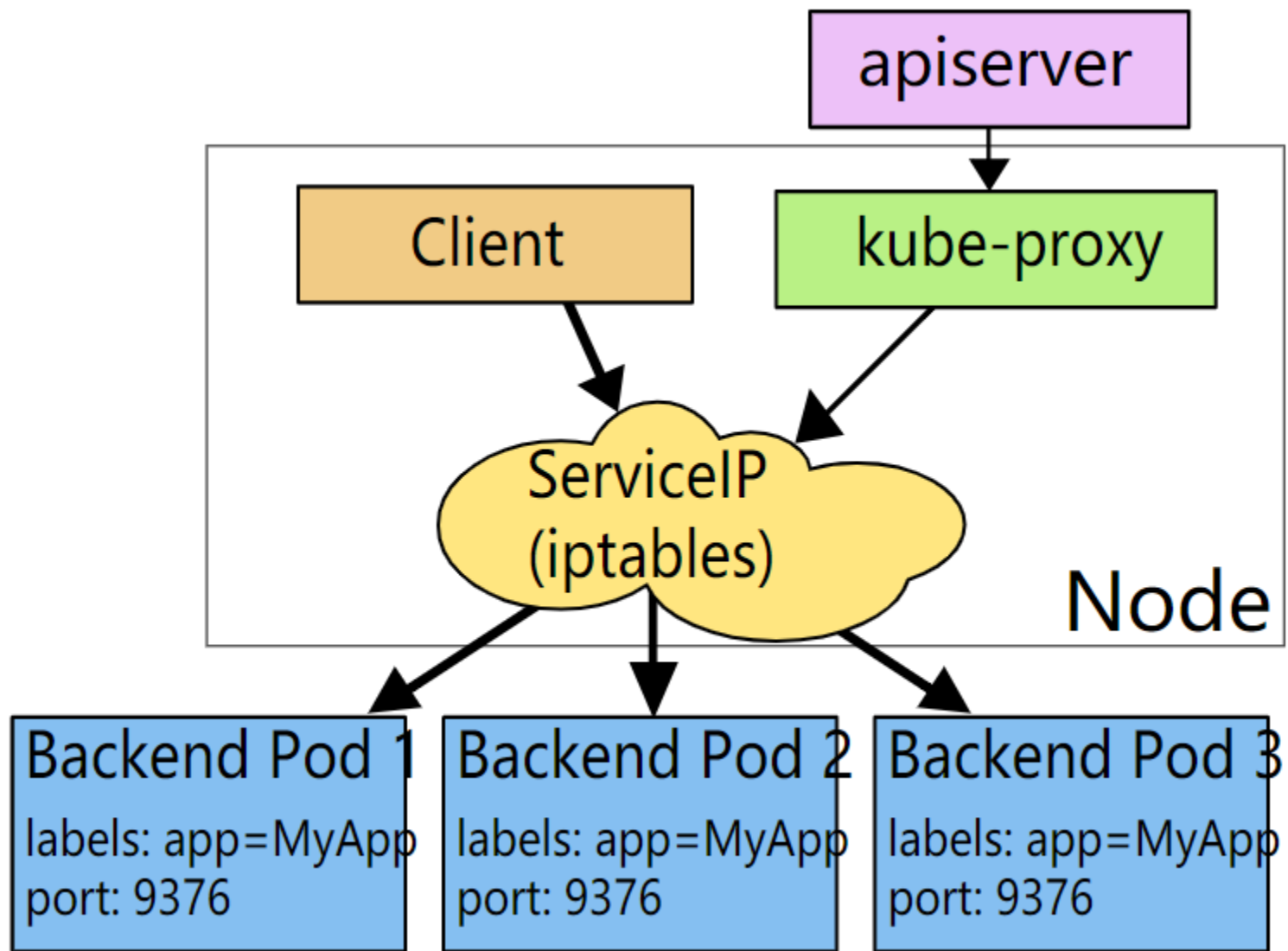
Docker 网络的问题

- 利用 host 端口进行访问，需要协调所有开发人员，大规模应用不现实
- 动态端口分配会给系统增加复杂度
- API Server 必须知道如何将动态分配的端口随时写入配置
- 服务发现困难

Kubernetes Networking 基本要求

- Pod 网络
 - Pod 内容容器间直接互通，不做 NAT
 - Pod 与 Node 直接互通，不做 NAT
 - Pod 与 Pod 直接互通，不做 NAT
 - 容器自己看到的 IP 地址和其他容器看到的一样
- Service Cluster IP

Service Cluster IP





Thank You

石志国 jasonshi@yunify.com



Kubernetes Networking - 2



Kubernetes Networking - CNI 与集群网络方案

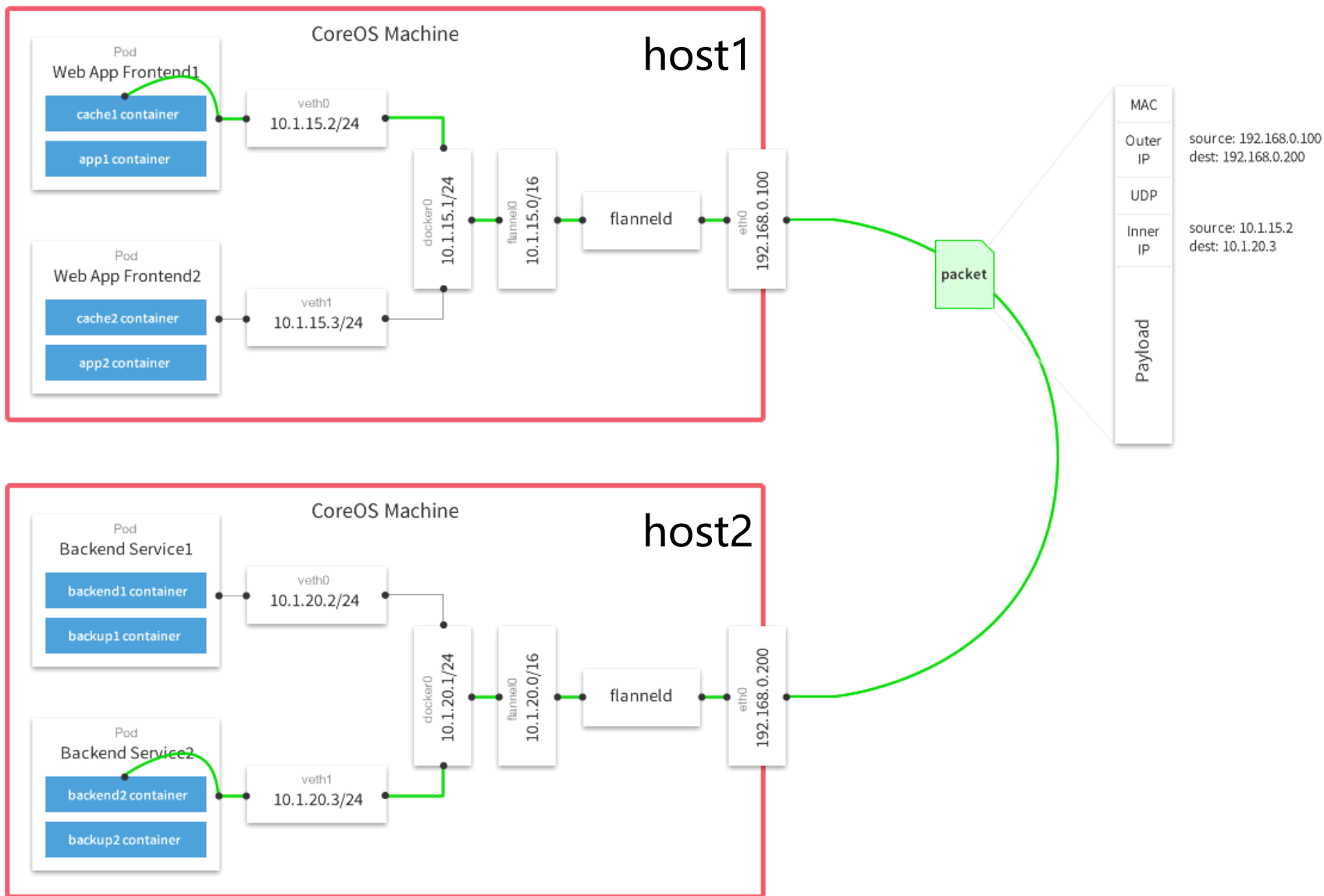
- 隧道方案 (Overlay Networking)

- Weave
- Open vSwitch (OVS)
- Flannel

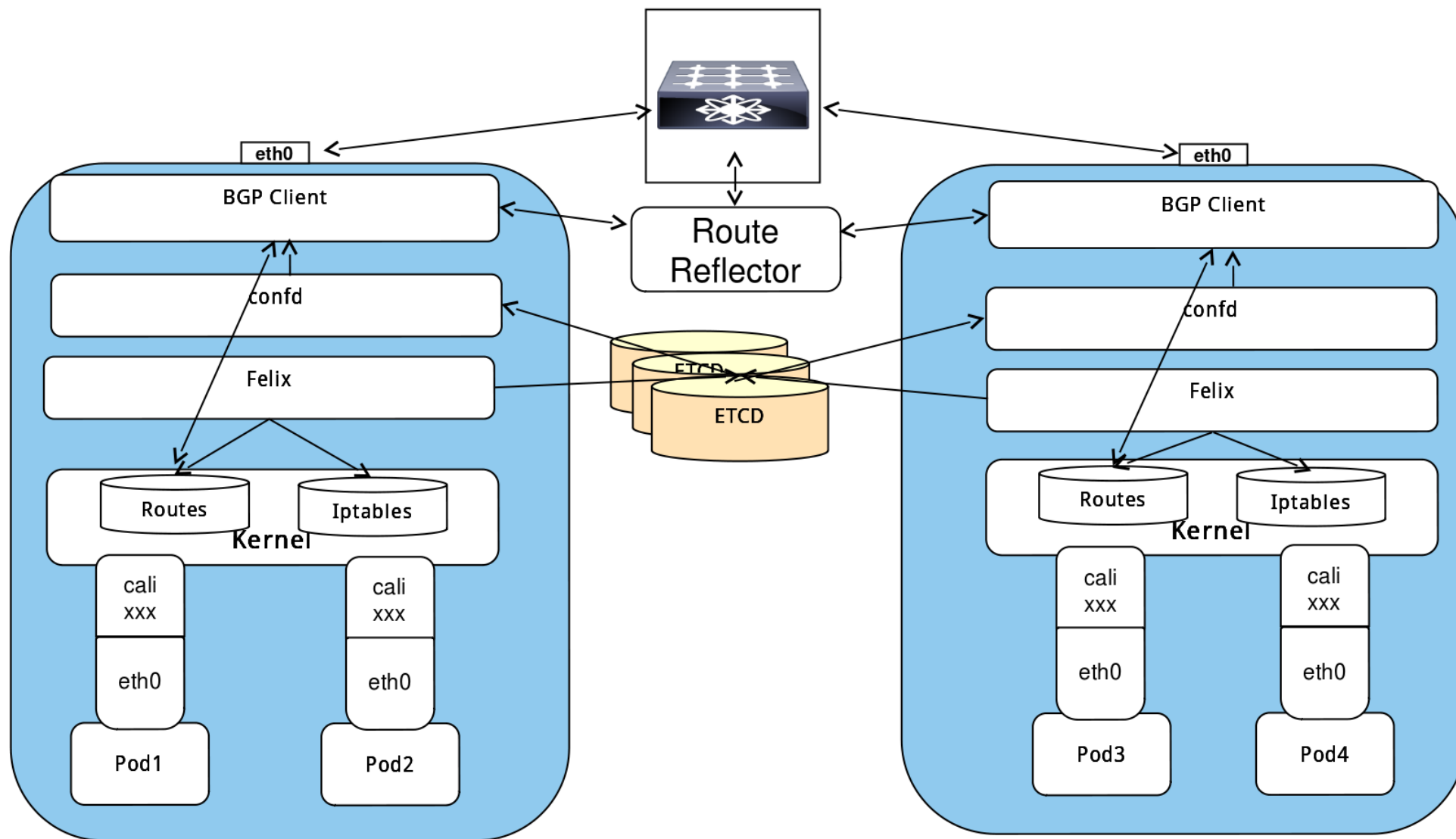
- 路由方案

- Calico
- Macvlan

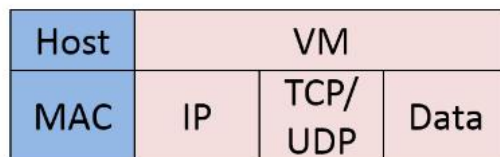
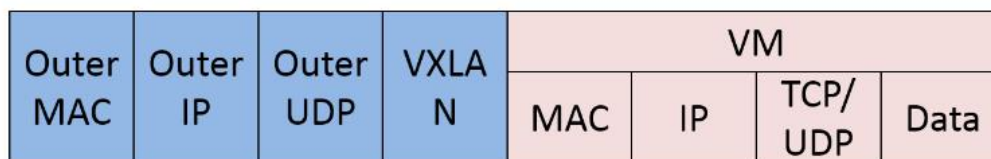
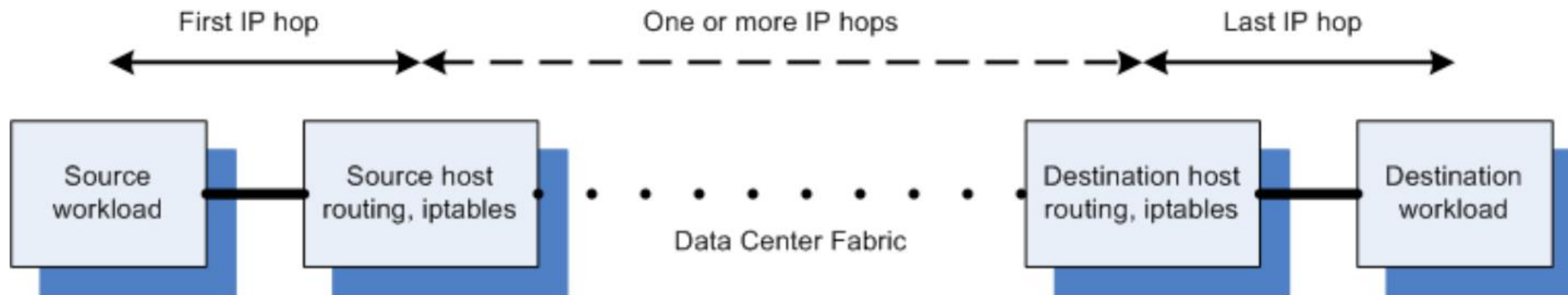
Kubernetes Networking - 集群网络方案 - Flannel



Kubernetes Networking - 集群网络方案 - Calico



Kubernetes Networking - 集群网络方案 - Calico





Thank You

石志国 jasonshi@yunify.com



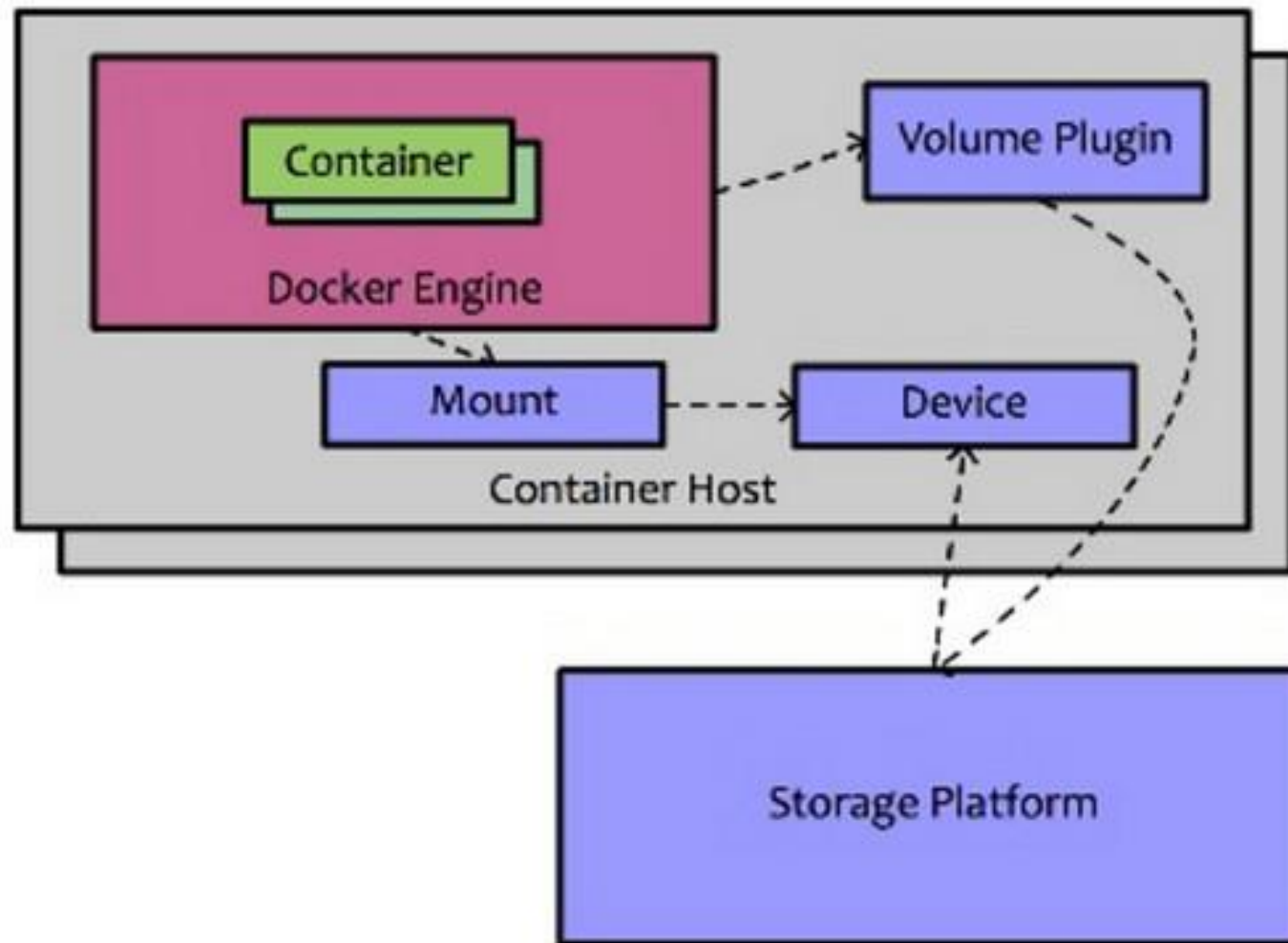
Kubernetes Storage - 1



Docker 插件机制与类型

Docker 插件类型

- 存储 Volume
- 网络 Network
- 认证 Authorization
- IP地址管理 IPAM



Kubernetes Storage 概述

- 普通 Volume
- 持久化 Volume - PV & PVC
- 动态存储供应 - Storage Class

在 Pod 中使用存储卷

- 来源 `spec.volumes`: 提供指定的存储卷
- 挂载点 `spec.containers.volumeMounts`: 将存储卷挂载到容器中



Thank You

石志国 jasonshi@yunify.com





Kubernetes Storage - 2

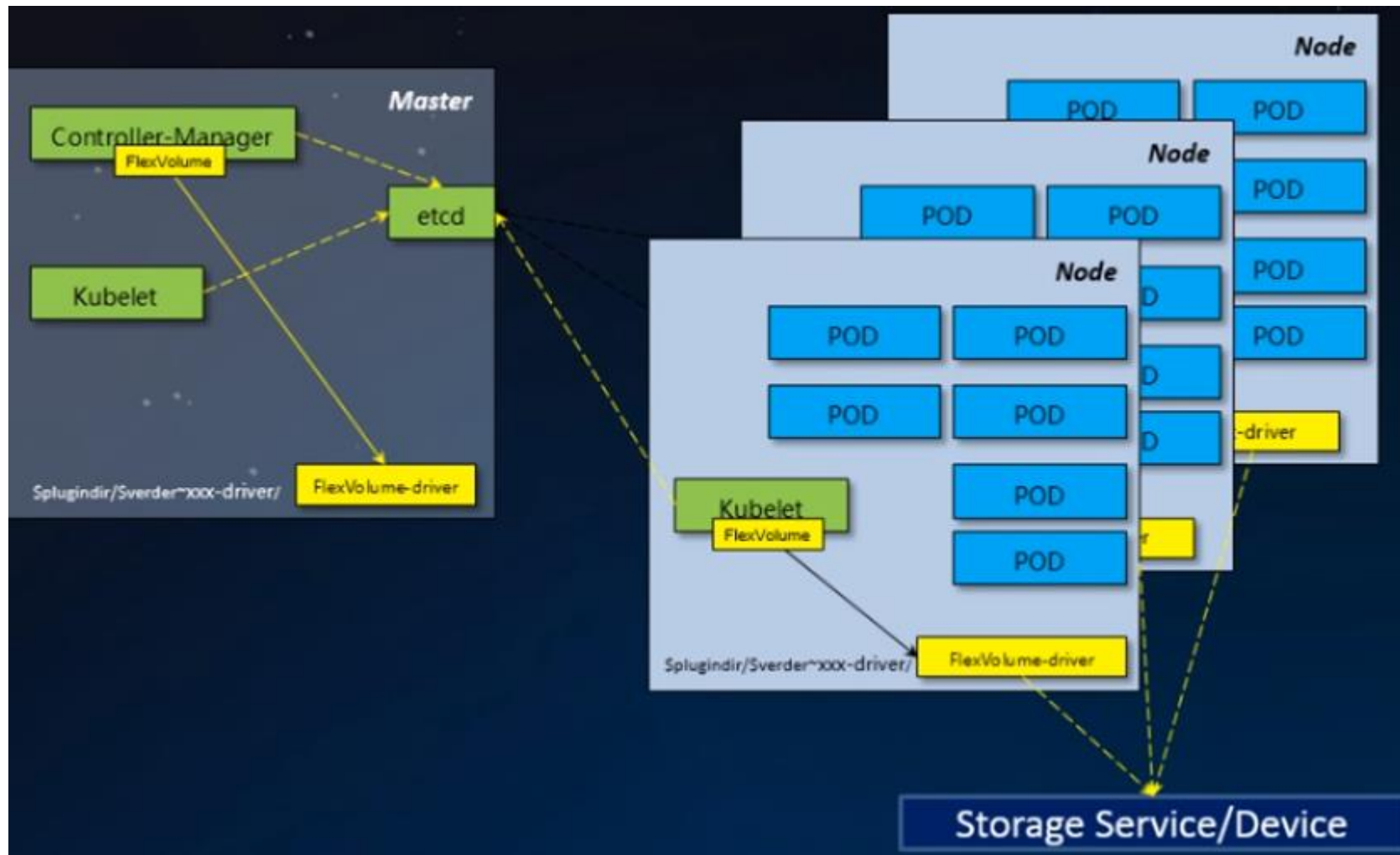
Flex Volume 与 CSI 架构



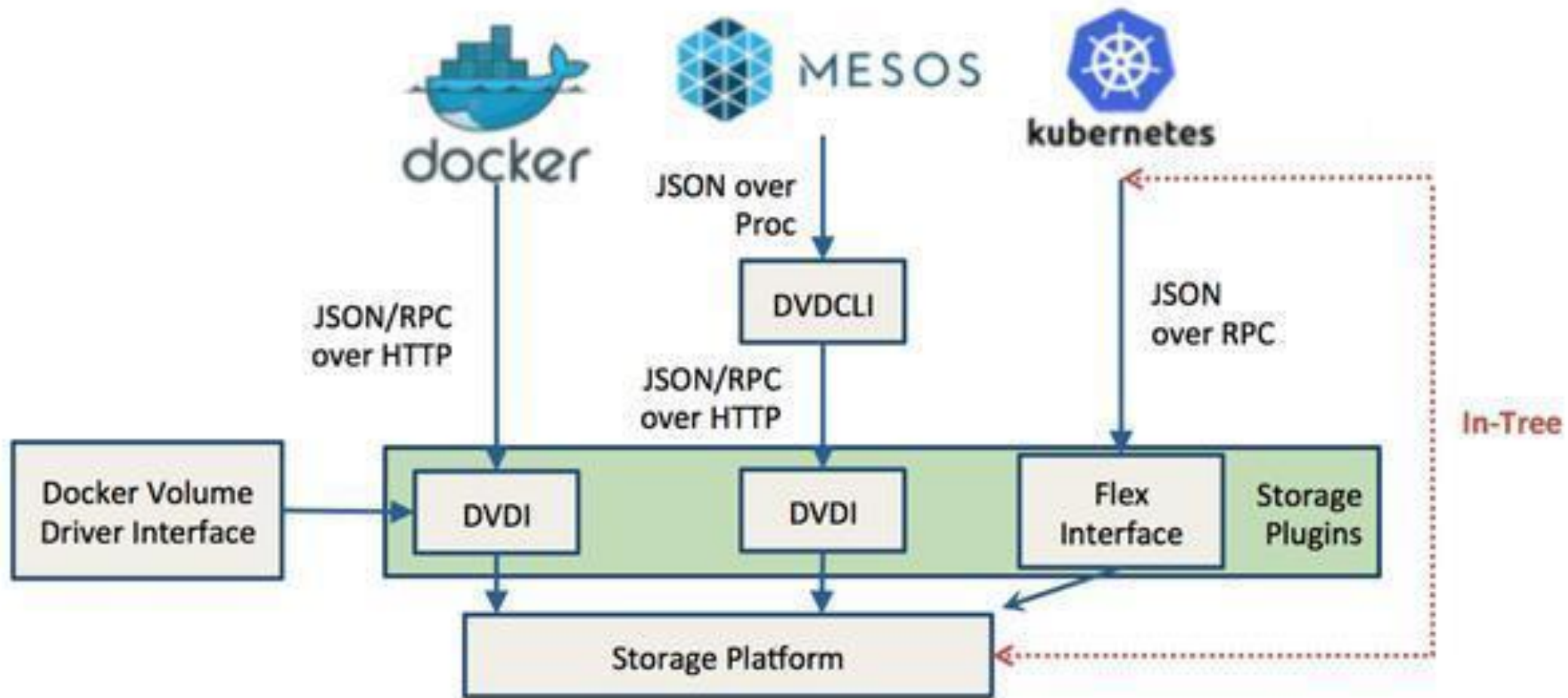
Kubernetes volume plugin 类型

- 内置插件 (in-tree)
- 外置插件 (out-of-tree)
 - Flex Volume
 - CSI

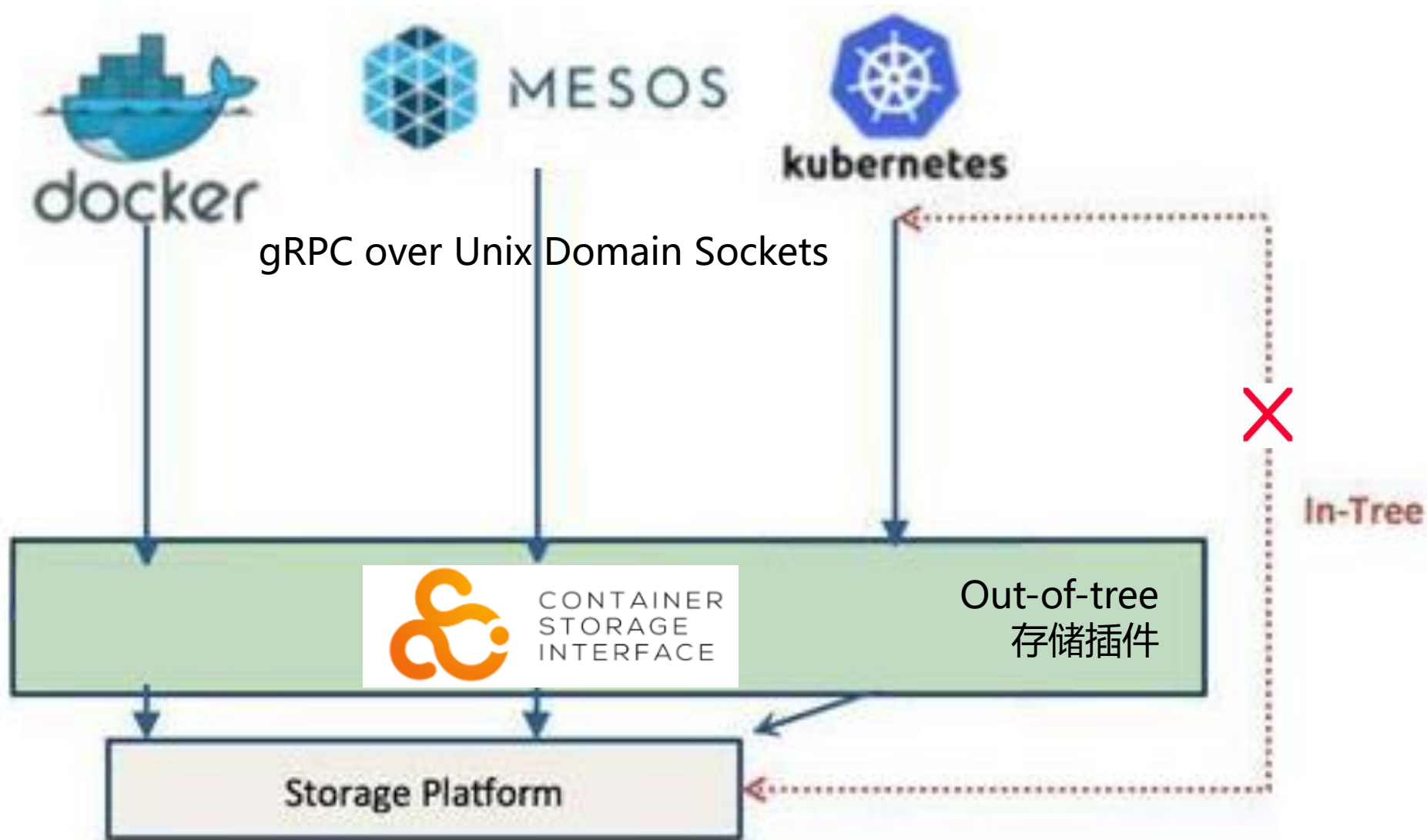
Flex Volume 架构



Why CSI ?



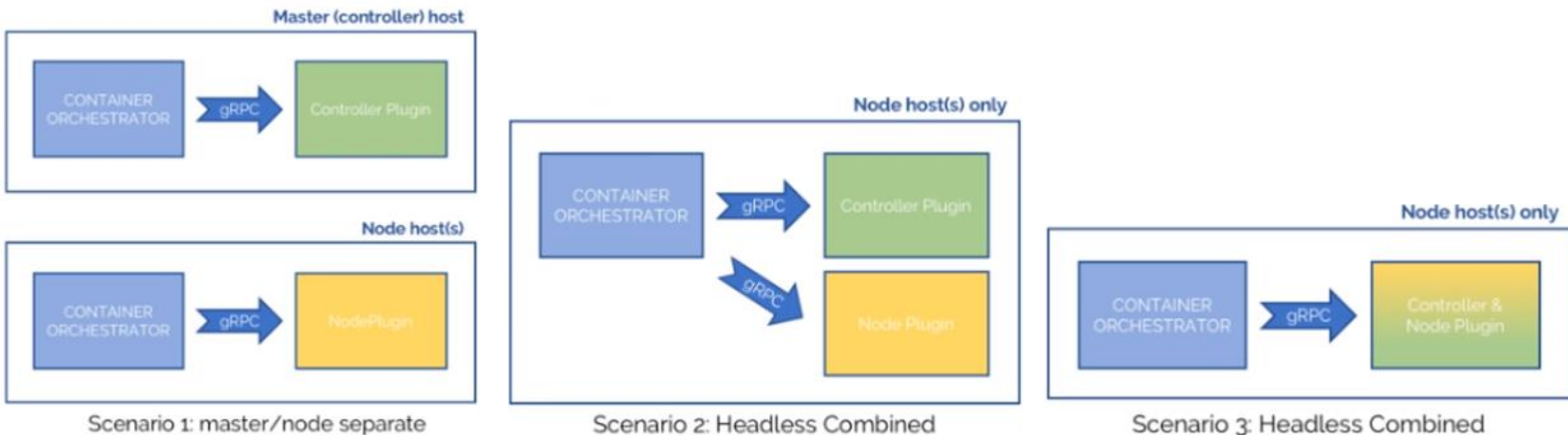
CSI



CSI 通用架构

Container Orchestrator 通过 gRPC 与插件交互，每个 Storage Provider 必须实现以下两个 plugin:

- Node Plugin - 需运行在使用 volume 的 Node 节点上，主要负责 volume mount/unmount 等操作
- Controller Plugin - 可运行在任何节点上，主要负责 volume creation/deletion、attach/detach 等操作





Thank You

石志国 jasonshi@yunify.com

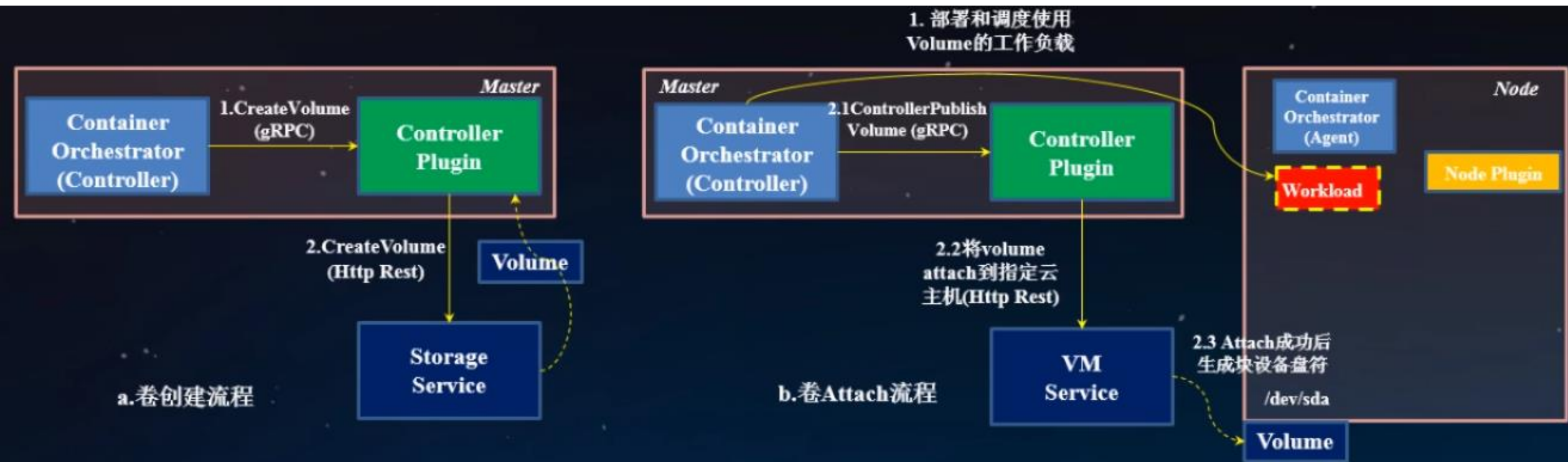


Kubernetes Storage - 3

CO 与 Plugin 的交互

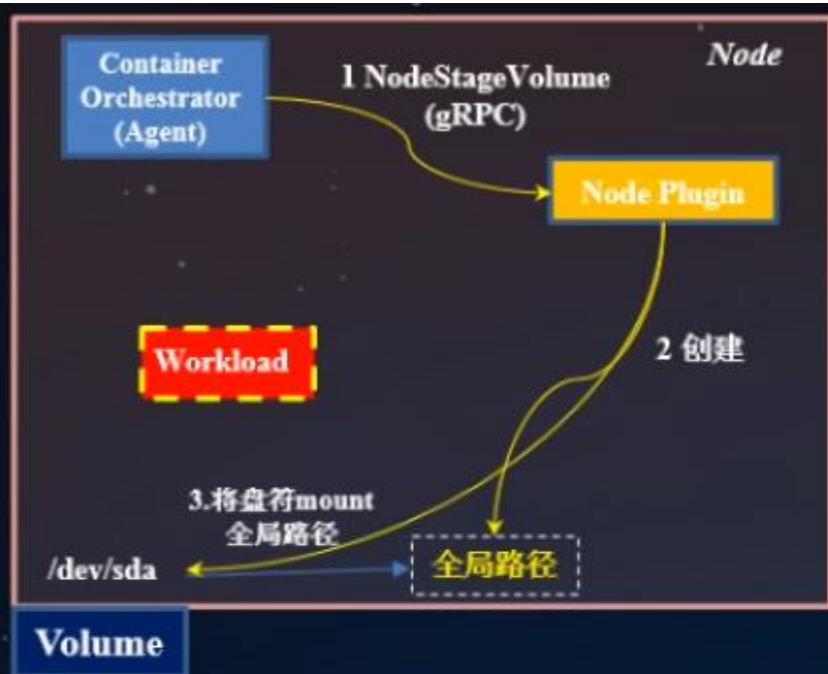


CO 与 Plugin 的交互 - 1. 卷的创建和 Attach



卷的Attach是由Workload的调度和启动所触发的

CO 与 Plugin 的交互 - 2. 将卷 mount 到全局路径、workload 挂载路径



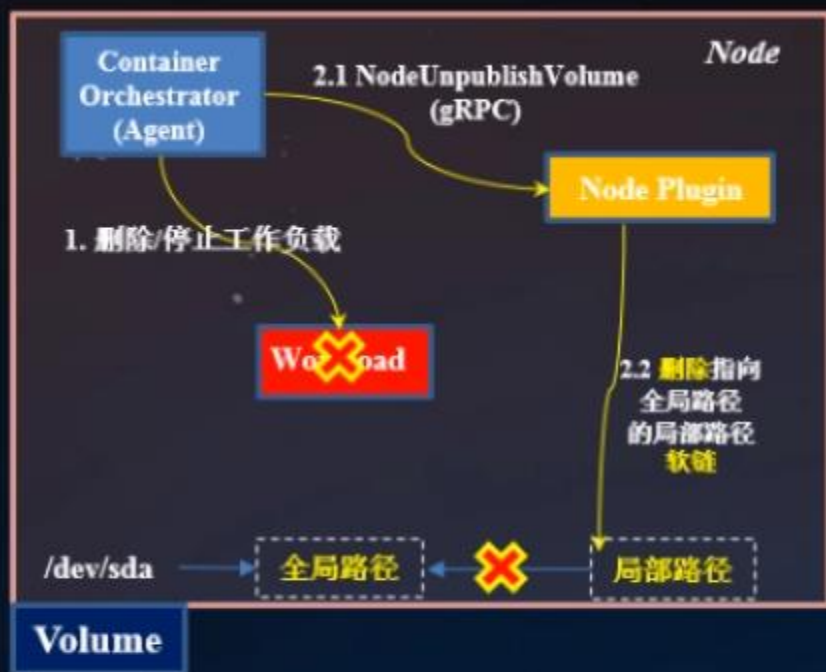
c. 卷mount到全局路径流程



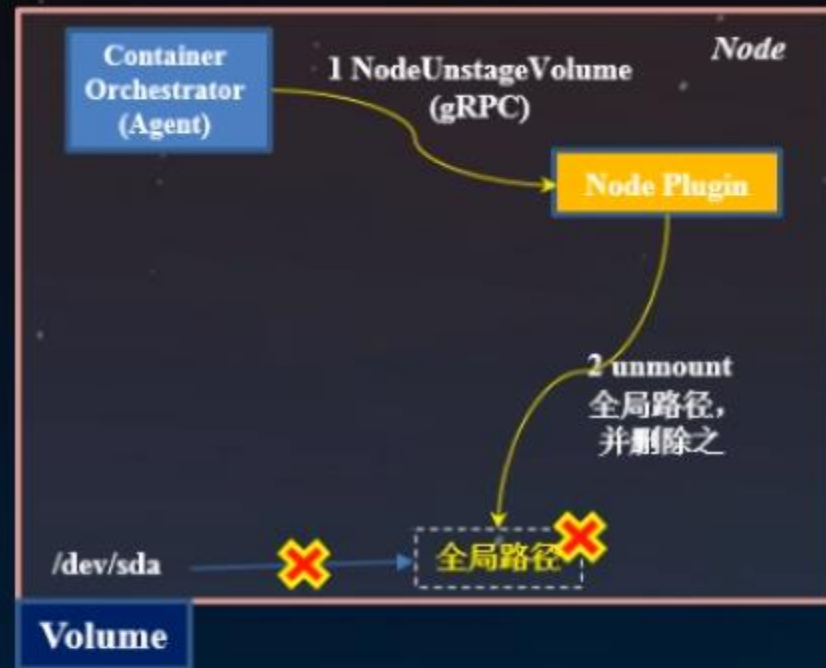
d. 卷mount到workload挂载路径流程

- 卷的Mount系列操作是由Workload的启动所触发的
- K8S中volume的**全局挂载路径**（存储挂载点）格式样例：
`/var/lib/kubelet/plugins/kubernetes.io/$volume_plugin/mounts/$volume_name`
- K8S中volume的**workload挂载路径**（软链）格式样例：
`/var/lib/kubelet/pods/$pod_id/volumes/$volume_plugin/$volume_name`

CO 与 Plugin 的交互 - 3. 将卷从 workload 挂载路径、全局路径 unmount



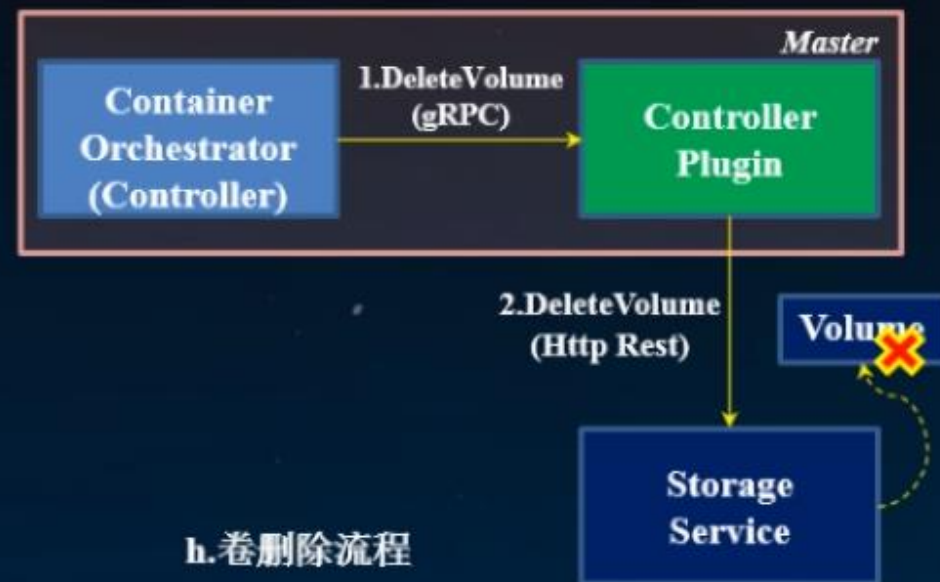
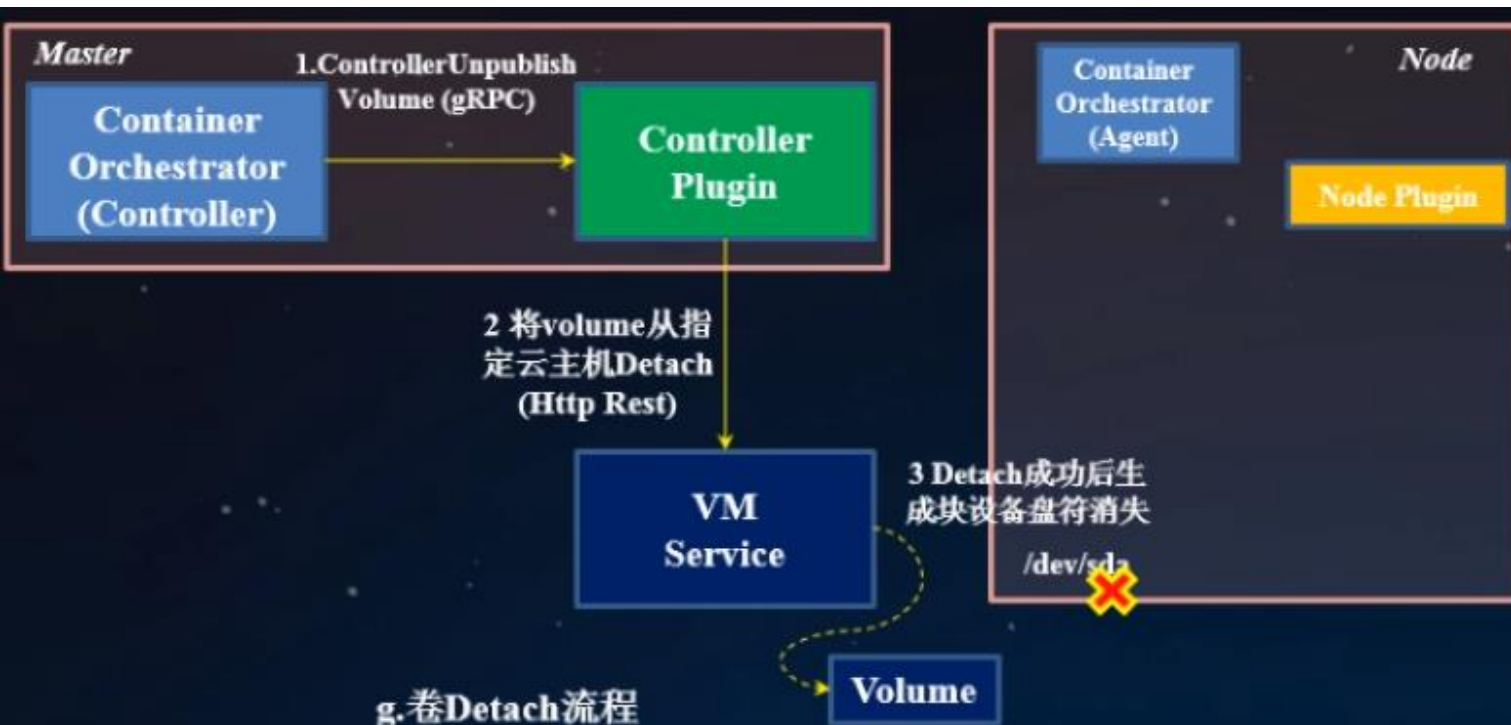
e. workload停止, 将卷从局部路径unmount(删除指向全局路径的局部路径软链)流程



f. 将卷从全局路径unmount流程

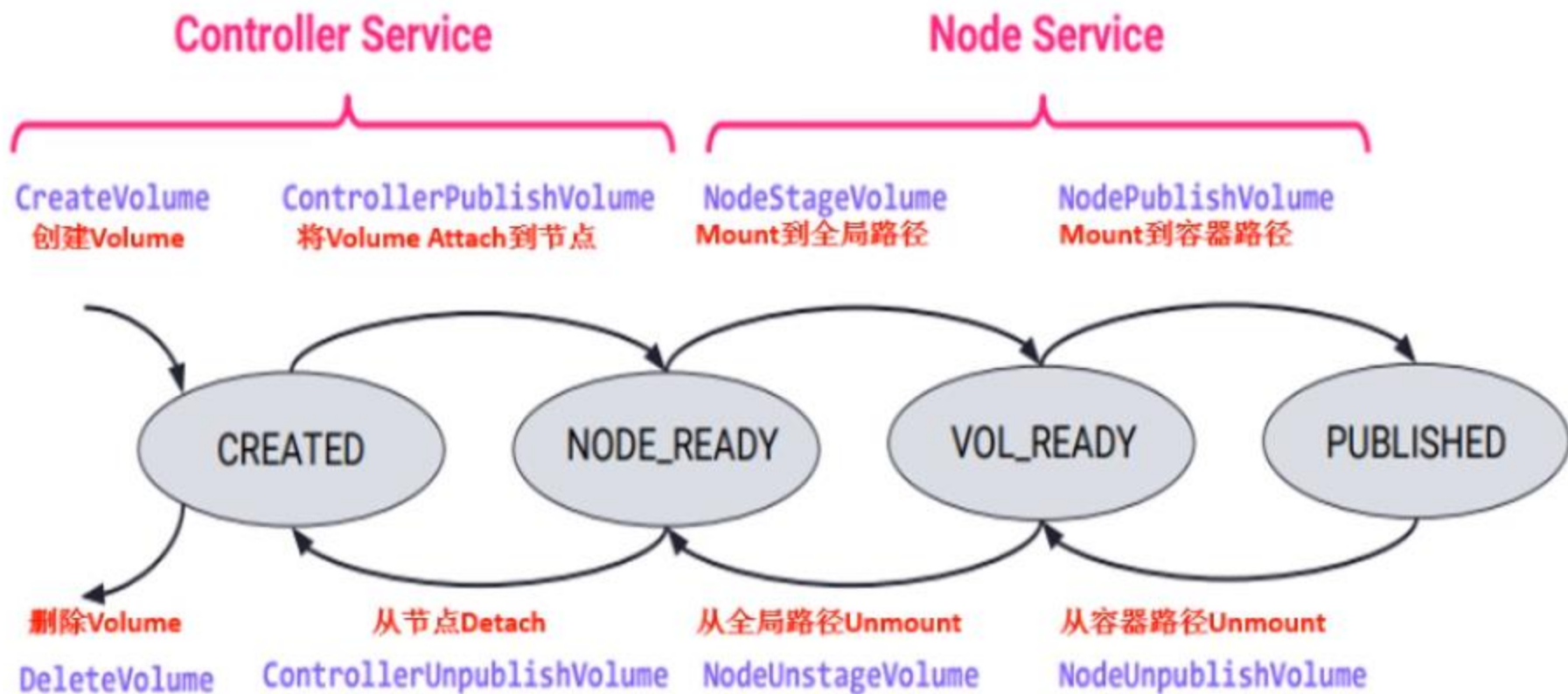
卷的unmount系列操作是由Workload的重启、迁移、删除所触发的

CO 与 Plugin 的交互 - 4. 卷的 Detach 和删除



卷的Detach是由Workload的重启、迁移、删除所触发的

卷的生命周期





Thank You

石志国 jasonshi@yunify.com

