

存储云原生在字节跳动的探索与实践

陈尧

资深架构师



个人简介



陈尧

资深架构师

运维自动化领域超过十年的架构经验，擅长容器、分布式调度、部署系统设计，对有无状态服务等复杂场景的云原生，有丰富的实战经验，目前在字节跳动负责存储服务的云原生工作

目录

Contents

① 定义存储服务

② 是否真的需要云原生

③ 难点及解决方案

④ 最佳实践

/01

定义存储服务

字节的存储服务是什么样的？



说起云原生存储，你会想到什么？

1. K8s concepts

- volume、storage class、CSI
- PV、PVC

2. CNCF 存储项目

- Ceph、GlusterFS
- Soda (openSDS)

...



Kubernetes

+



Storage



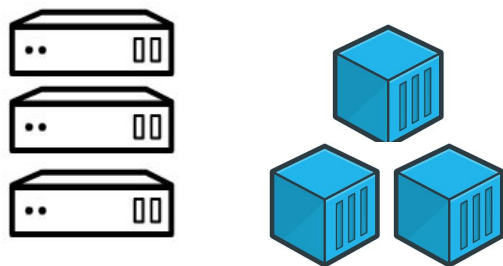
**但是，这不是我们
今天的话题**



字节的底层服务分为计算、存储两类：

- 计算

- IaaS、PaaS、FaaS 等



- 存储

- cache、数据库等





字节的存储服务



/02

是否真的需要云原生？

云原生能解决存储服务的哪些问题？



1.稳定性

- 容错、故障自愈能力
- 规范运维

2.成本

- 资源
 - 调度优化、混部、弹性收缩、超卖
- 运维
 - 抽离资源问题（机器 / 机型 -> cpu core / mem）
 - 能力复用、统一
 - 私有化部署

现状 → 预期

收益



目标分析 — 以成本优化为例

1. 是否适合上云原生

- 云原生是否能达成预期收益？
- 改造成本：复杂度、上下游影响
- 是否存在实例独占整机的情况

2. 完全云原生 vs 部分云原生

- 组件所有模块是否都需要云原生化
- 模块是否属于计算型



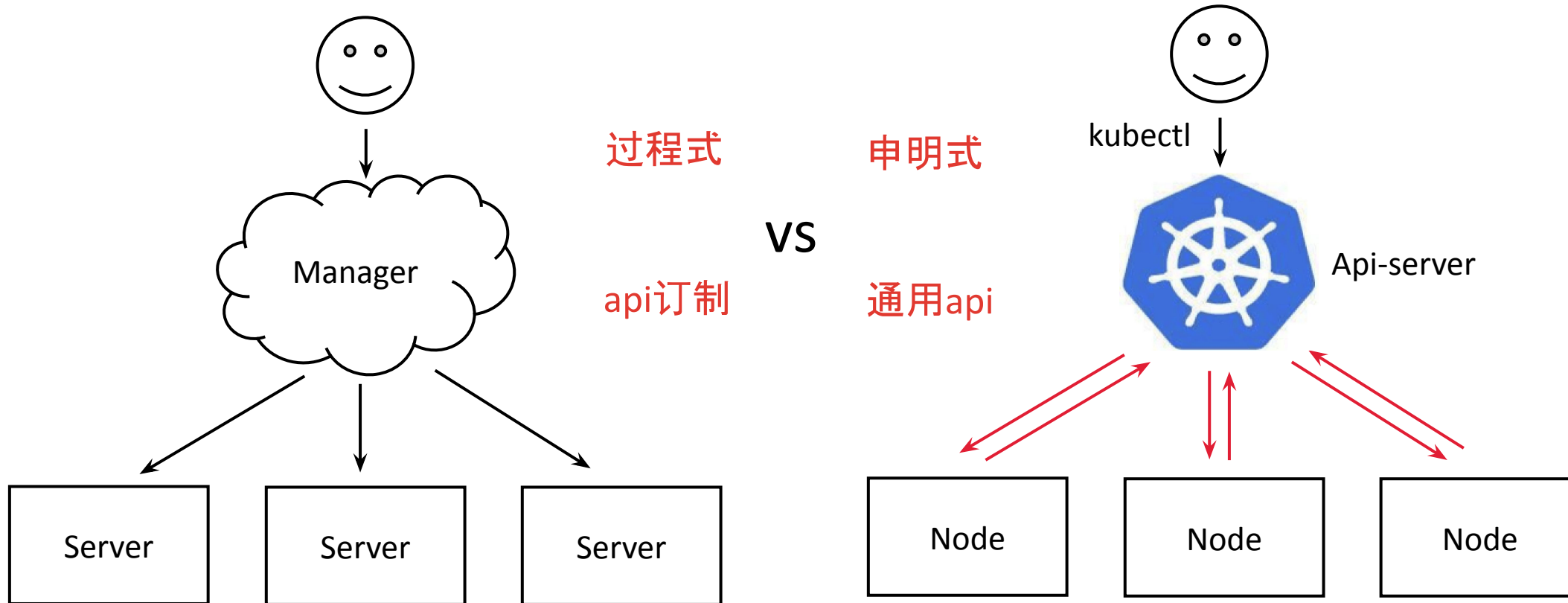
/03

存储云原生的难点

存储云原生面临的问题



传统模式 vs K8S 模式





有状态 or 无状态

Stateless



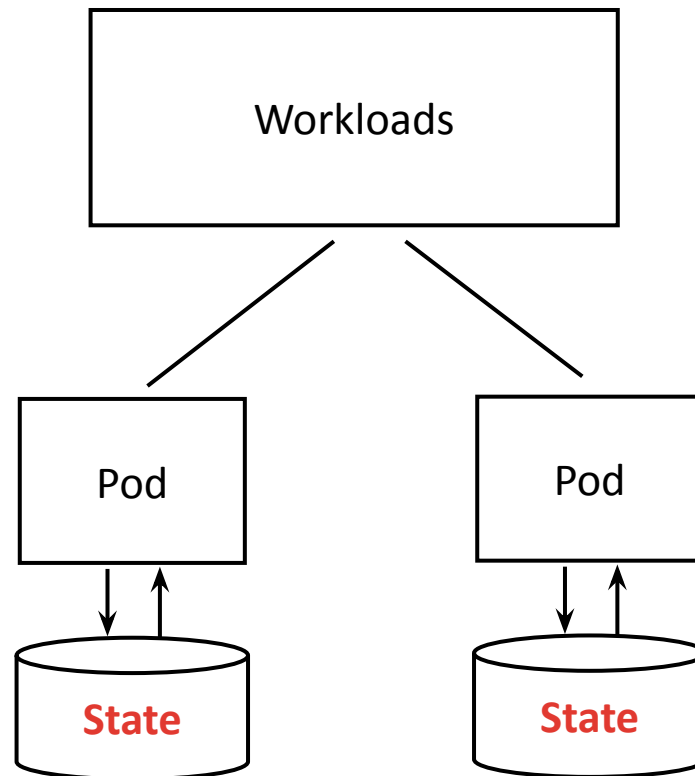
VS

Stateful



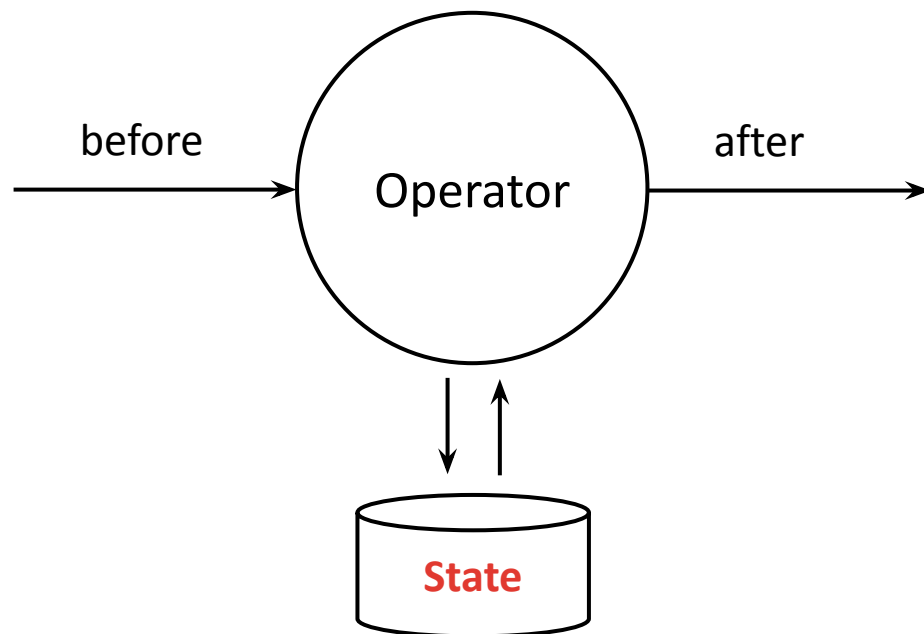


- 持久化存储选型？
- Node 宕机？
- Pod 调度与迁移？





Operator 状态问题



- 实例依赖问题？
- 状态不一致？
- 跨 K8S 集群编排？

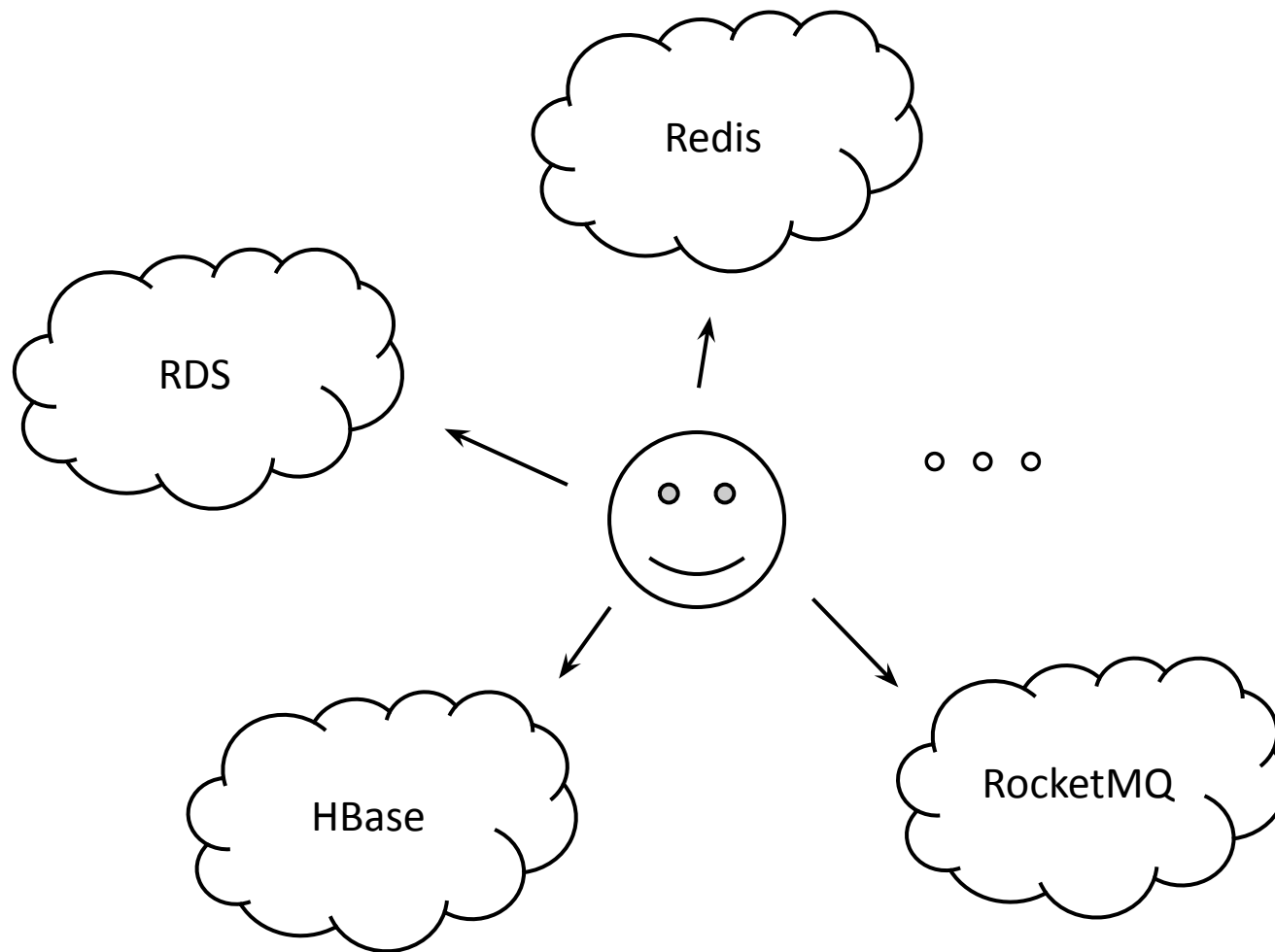
/04

字节的存储云原生实践

从真实场景出发

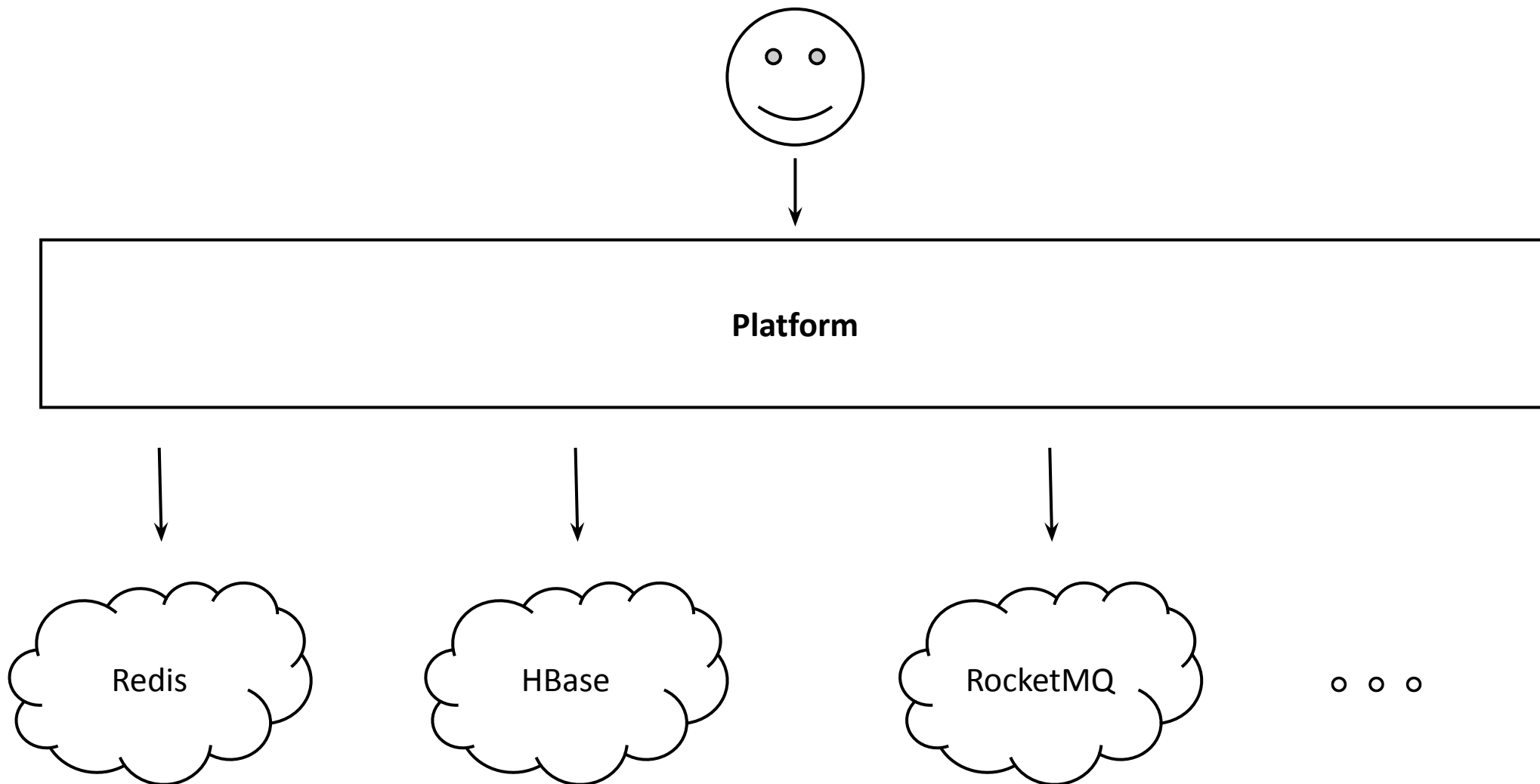


当我们面对各式各样的存储



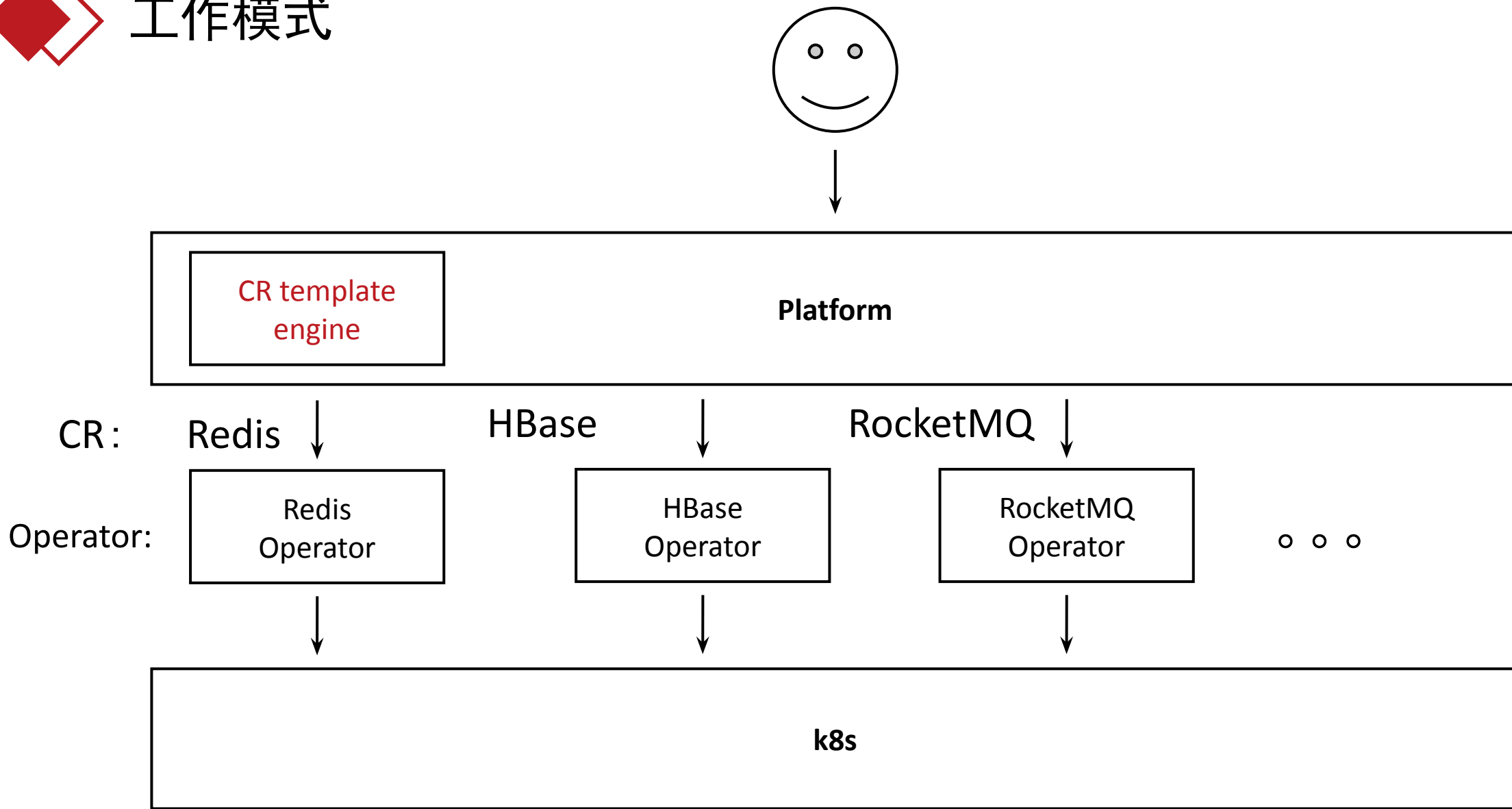


统一的平台建设





工作模式





字节的存储云原生实践

用户层



解释层



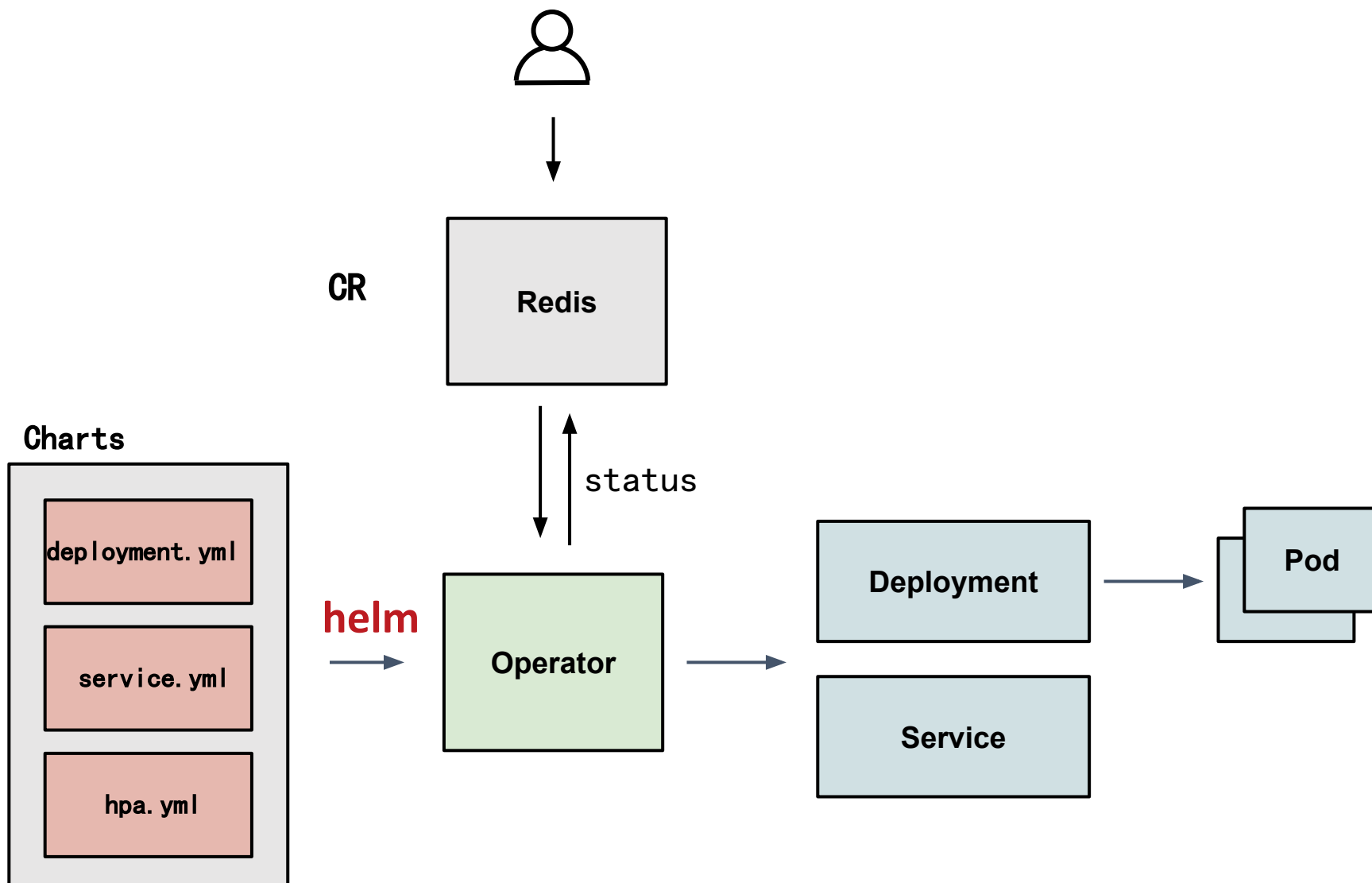
基础设施



权限管理

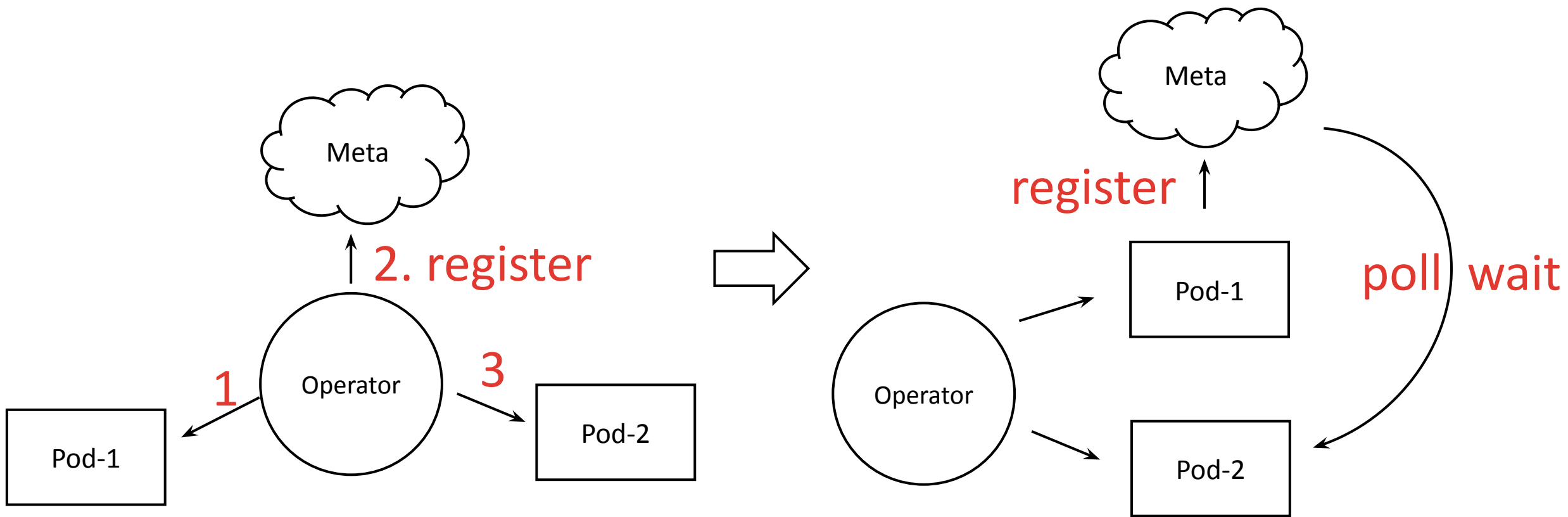


Operator 设计





状态解除

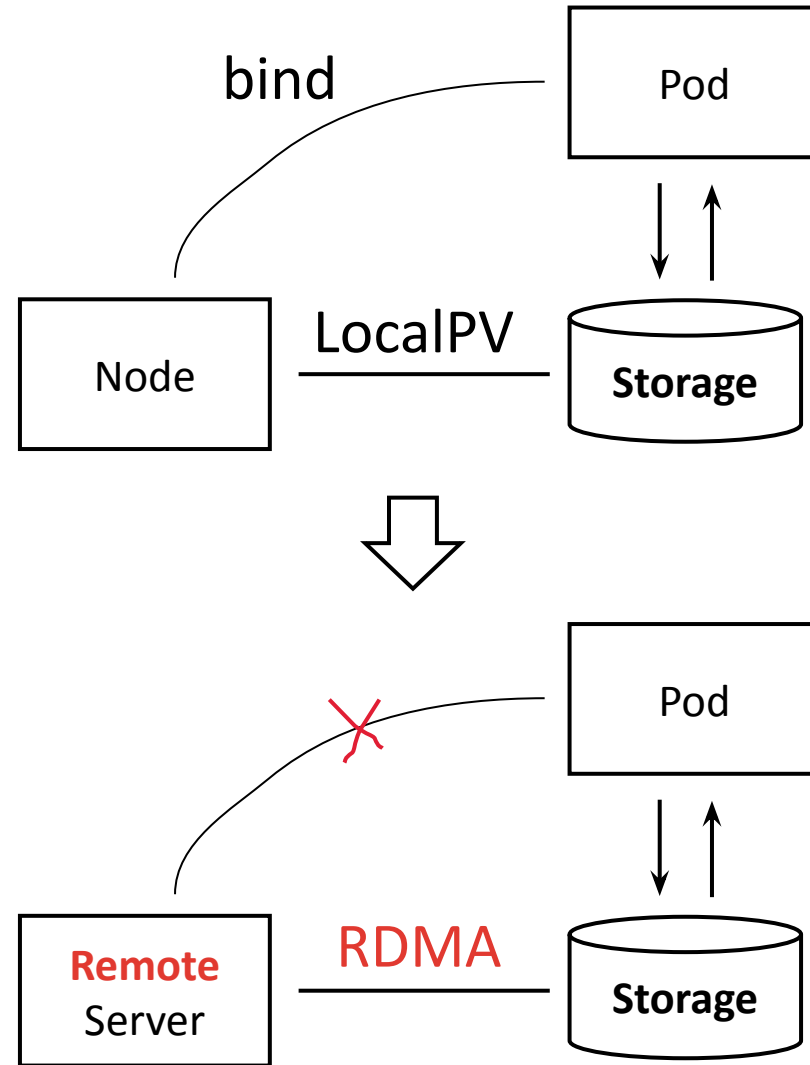


1. 方案

- 本地存储 (LocalPV)
 - io 敏感场景
- 远程存储 (存储拉远、HDFS 等)

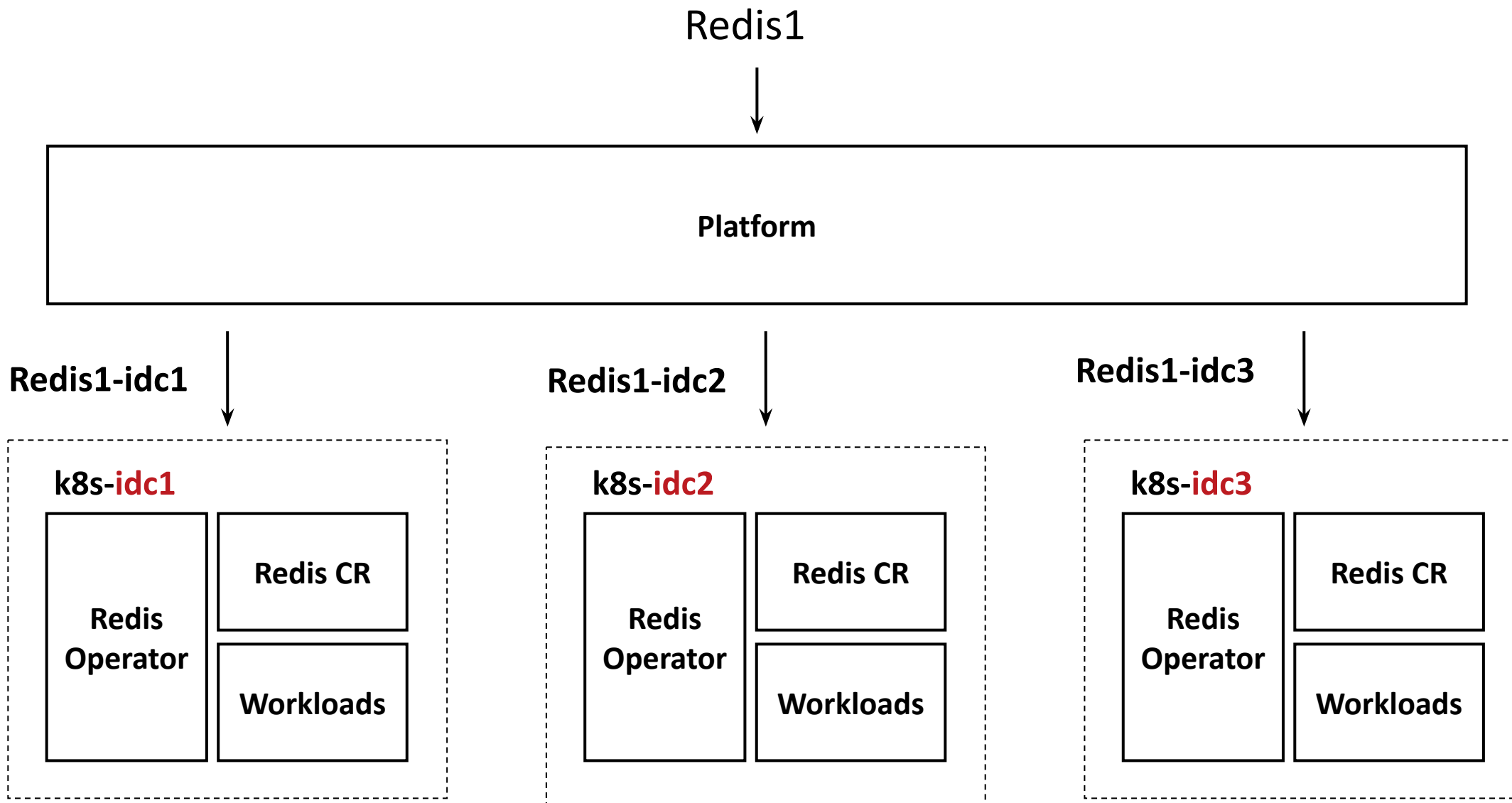
2. 选型

- 计算存储分离利于调度和资源优化
- 拉远 (RDMA) 是本地存储的替代方案



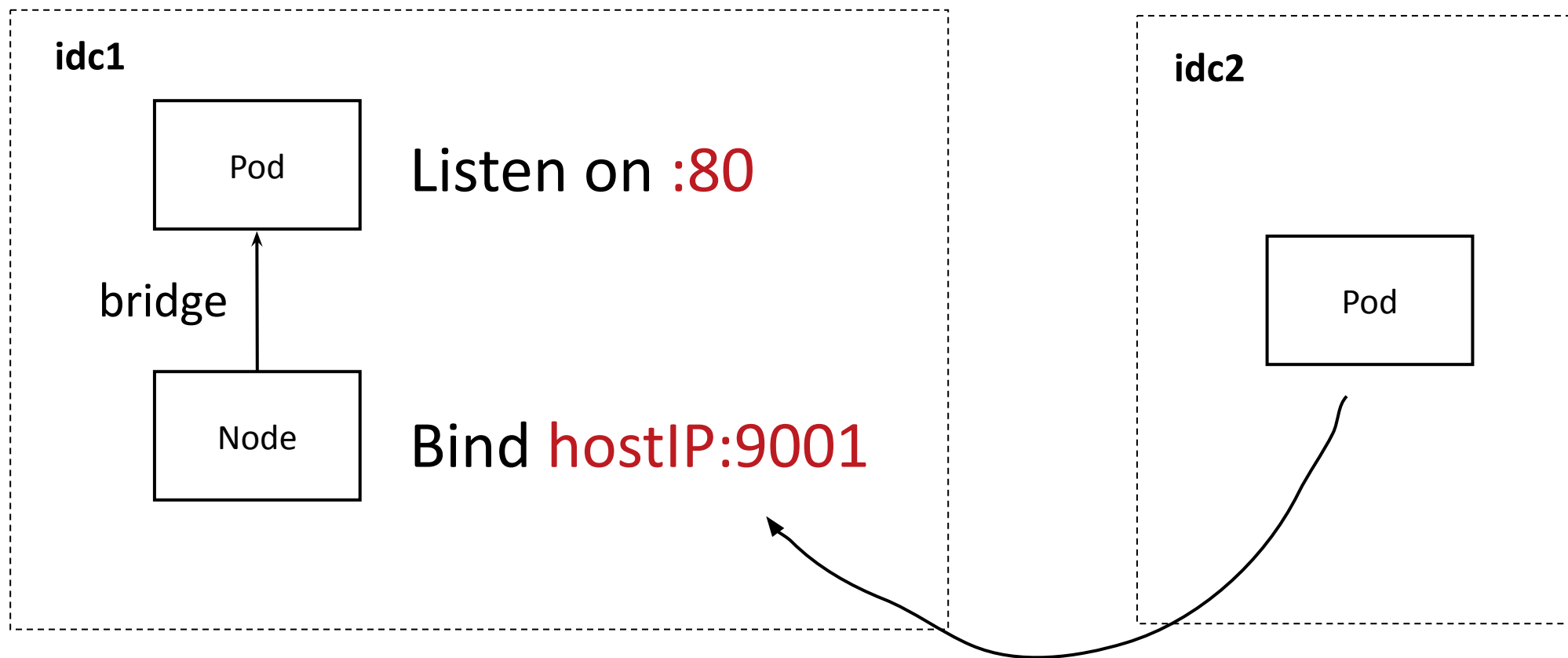


多机房架构



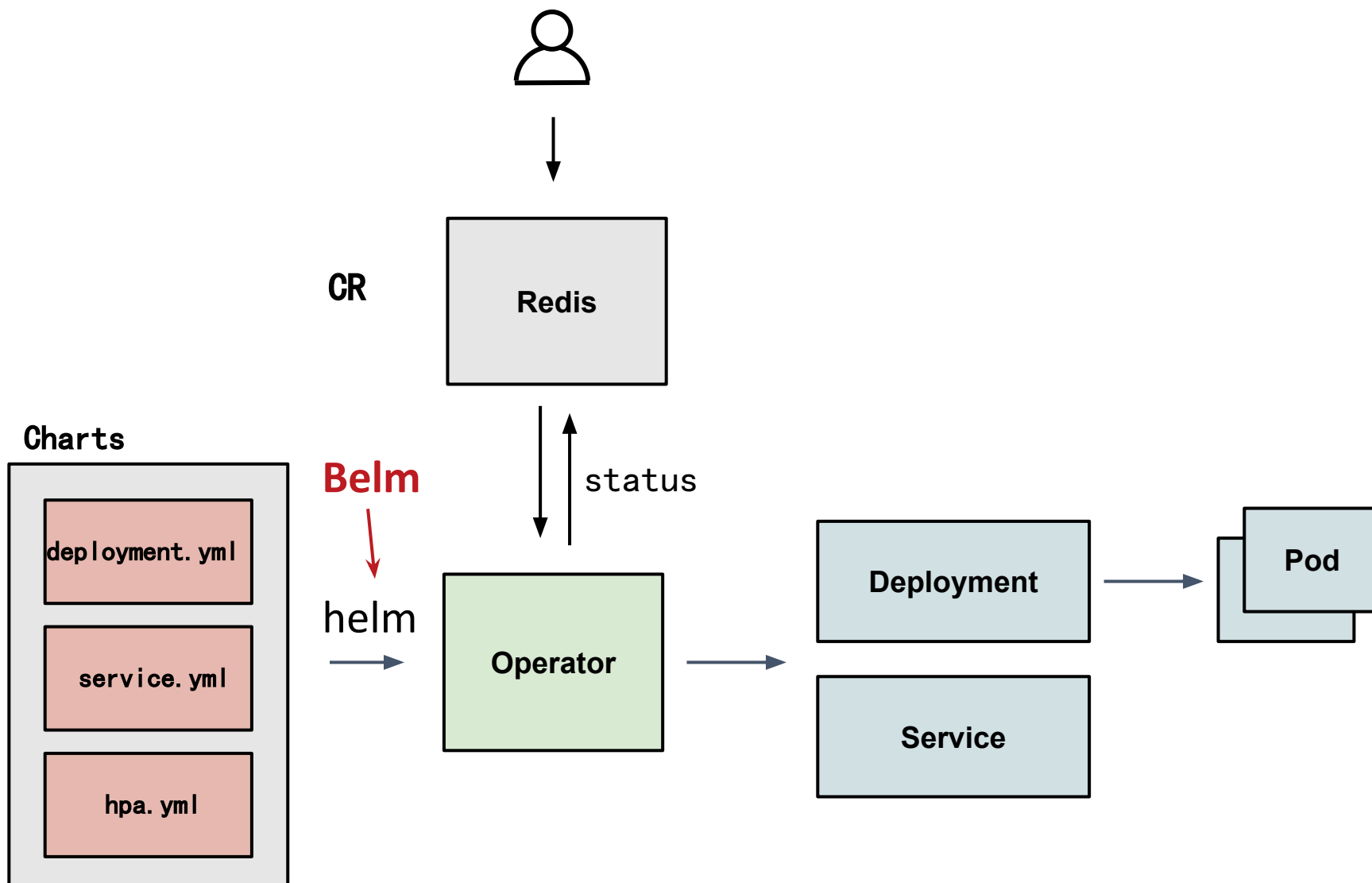


自动映射容器端口到物理机端口，使用物理机 IP+映射后的端口访问容器（修改内核实现）





Operator 演进





模板引擎 — Belm (Bytedance Helm Engine)

- 兼容 helm chart 主体功能
- 轻量、性能数量级提升
- 版本控制

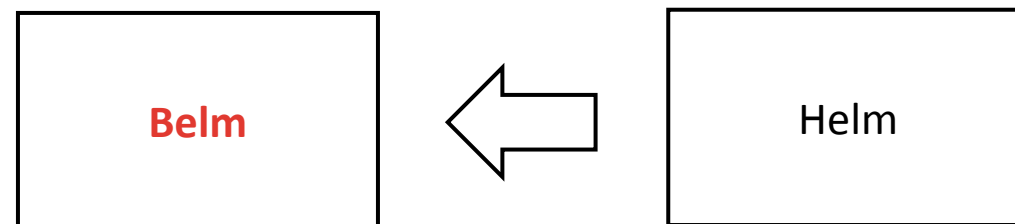
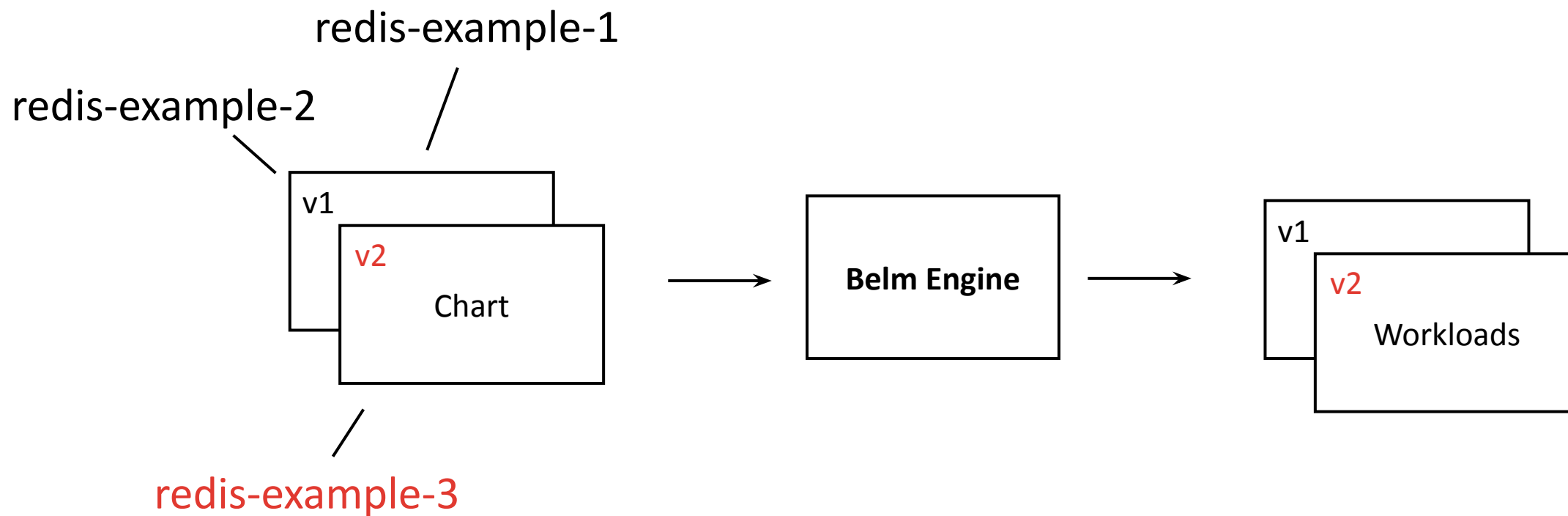


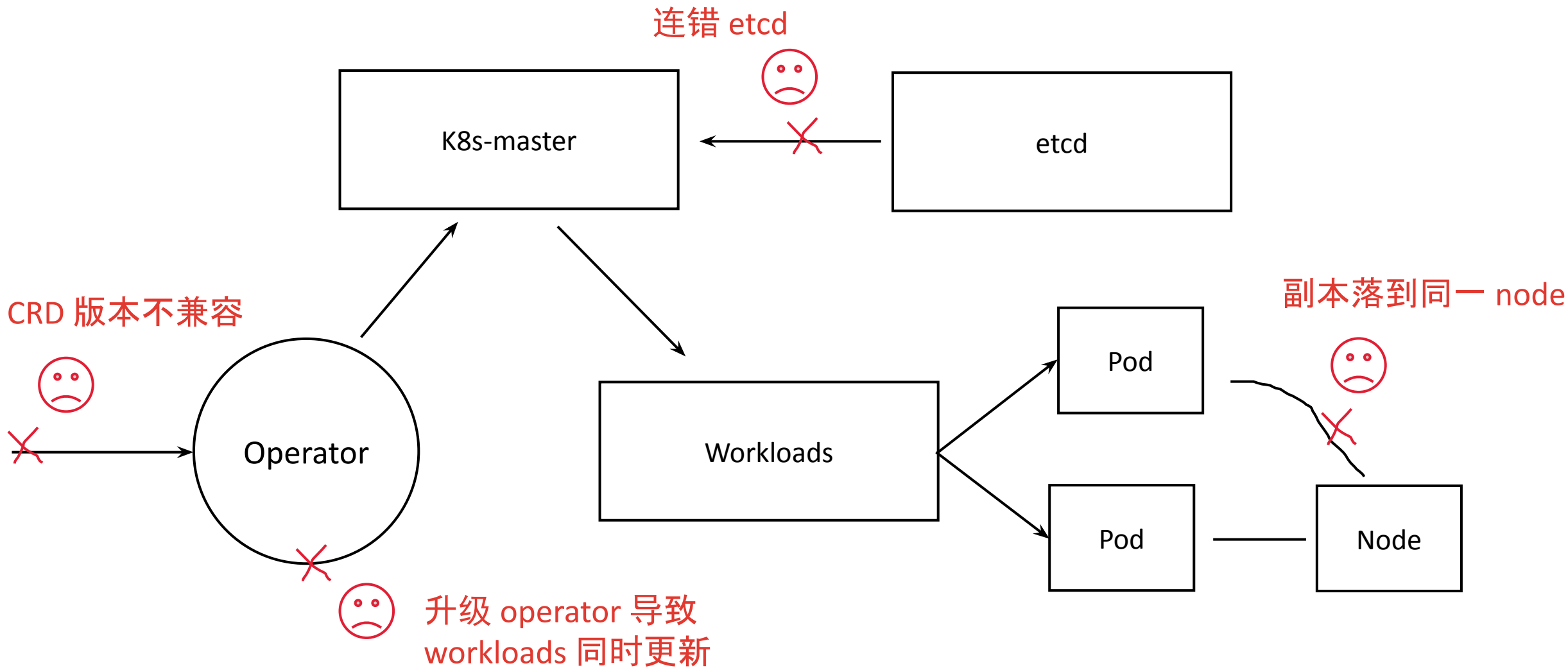


Chart 版本控制





安全及稳定性问题





kubectl delete
namespace

=

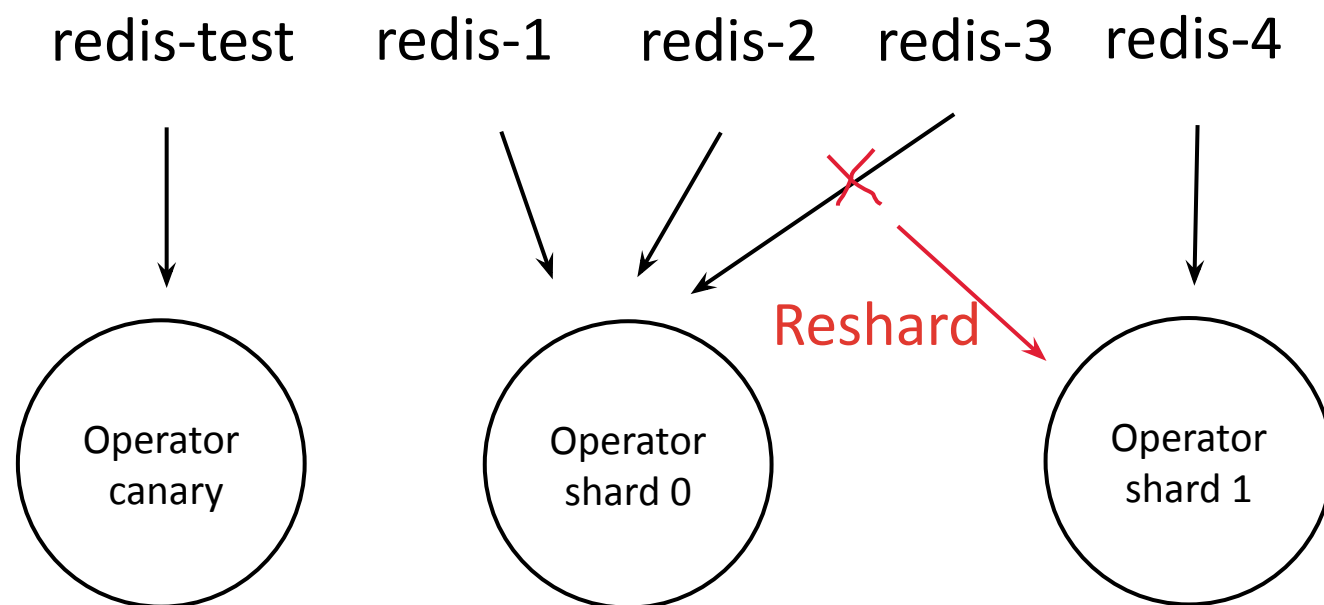


sudo rm -rf /*



- 控制面灰度发布流程
- NS、CRD 等删除阻断
- 安全策略 / 危险操作熔断

operator 灰度机制：





我们在存储云原生中做了什么：

- 存储组件云原生方案
 - 存储
 - 网络
 - 多机房架构
 - operator 框架
- 平台
 - 存储运维平台统一
 - 运维规范、交付标准



收益及展望

- Resource & Cost
- Efficiency
- Stability & Reliability





Thanks

开放运维联盟

高效运维社区

DevOps 时代

荣誉出品



想第一时间看到高效运维社区的
最新动态吗？

