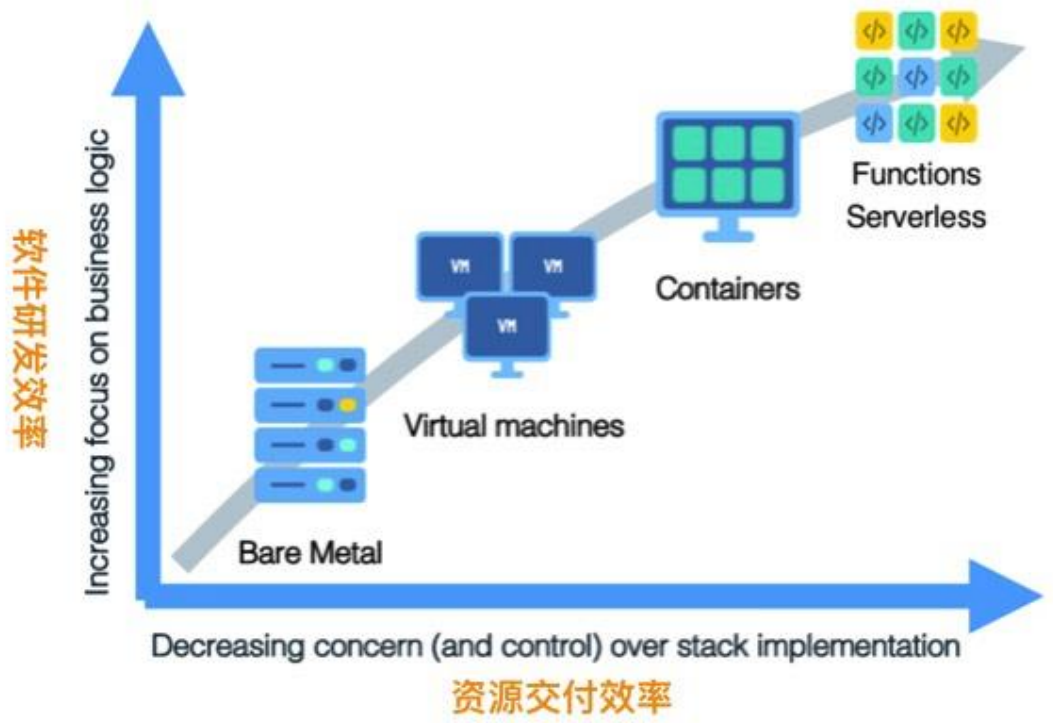
阿里云 | 奥运会全球指定云服务商

阿里云如何基于标准k8s打造边缘计算云原生基础设施

关注“阿里巴巴云原生”公众号
获取第一手技术资料

云原生和边缘计算

云原生概念

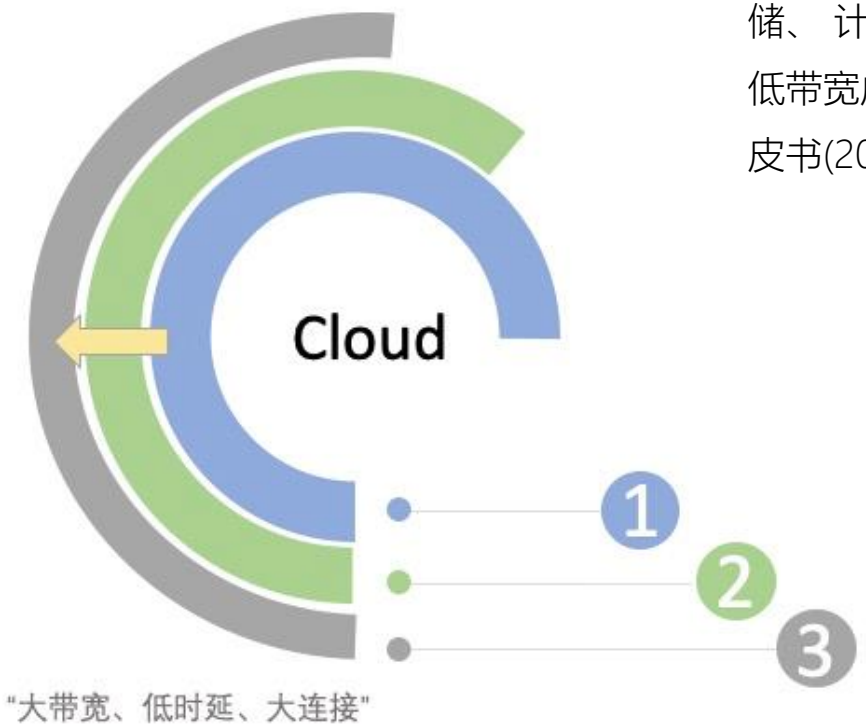


云原生的技术范畴包括了以下几个方面：

- 云应用定义与开发
- 云应用的编排与管理
- 监控与可观测性
- 云原生的底层技术（比如容器运行时、云原生存储技术、云原生网络技术等）
- 云原生工具集
- Serverless

云原生的概念最早是在2013年被提出，经过这几年的发展，尤其是从 2015 年 Google 牵头成立 CNCF 以来，云原生技术开始进入公众的视线并逐渐演变成包括 DevOps、持续交付、微服务、容器、基础设施，Serverless，FaaS等一系列的技术，实践和方法论集合。

边缘云计算-“云边端一体”



概念

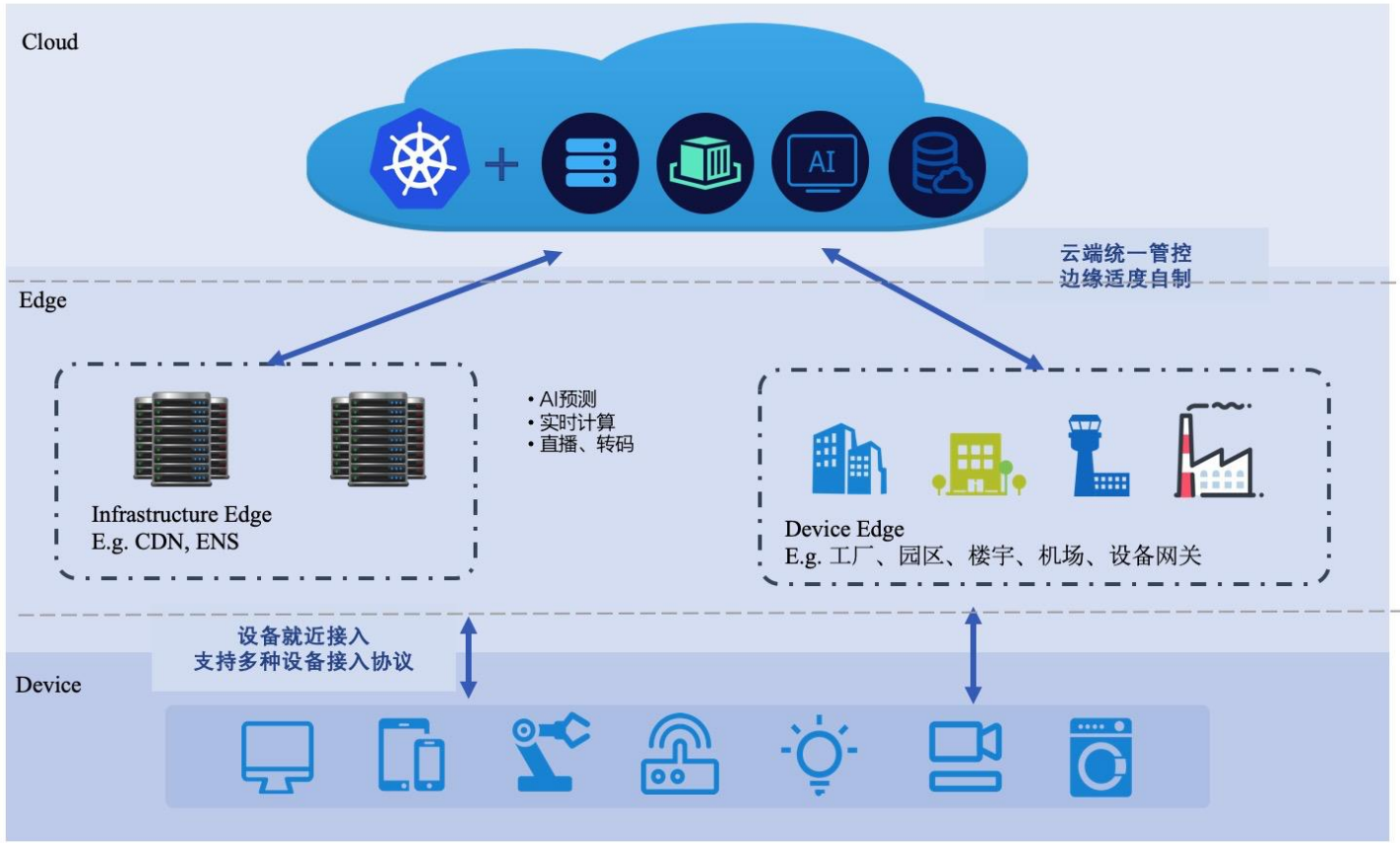
边缘云计算是基于云计算技术的核心和边缘计算的能力，构筑在边缘基础设施之上的云计算平台。形成边缘位置的计算、网络、存储、安全等能力全面的弹性云平台，并与中心云和物联网 终端形成“云边端三体协同” 的端到端的技术架构，通过将网络转发、存储、 计算，智能化数据分析等工作放在边缘处理，降低响应时延、减轻云端压力、降低带宽成本，并提供全网调度、算力分发等云服务。(引自: 边缘云计算技术及标准化 白皮书(2018))

① **“云”**中心云，又称传统云计算。管控端，全网算力统一管理与调度，与边缘协同。

② **“边”** “边”位于边缘云计算中的边侧。靠近设备和数据源的计算资源，用于部署边缘侧应用。可能是云厂商也可能是用户自己的边缘节点。

③ **“端”**“端”位于边缘云计算中的端侧，即设备端。跟据 Gartner 的报告，到2020年全球连接到网络的设备将达到约208亿台。

云-边-端 一体化协同基础设施



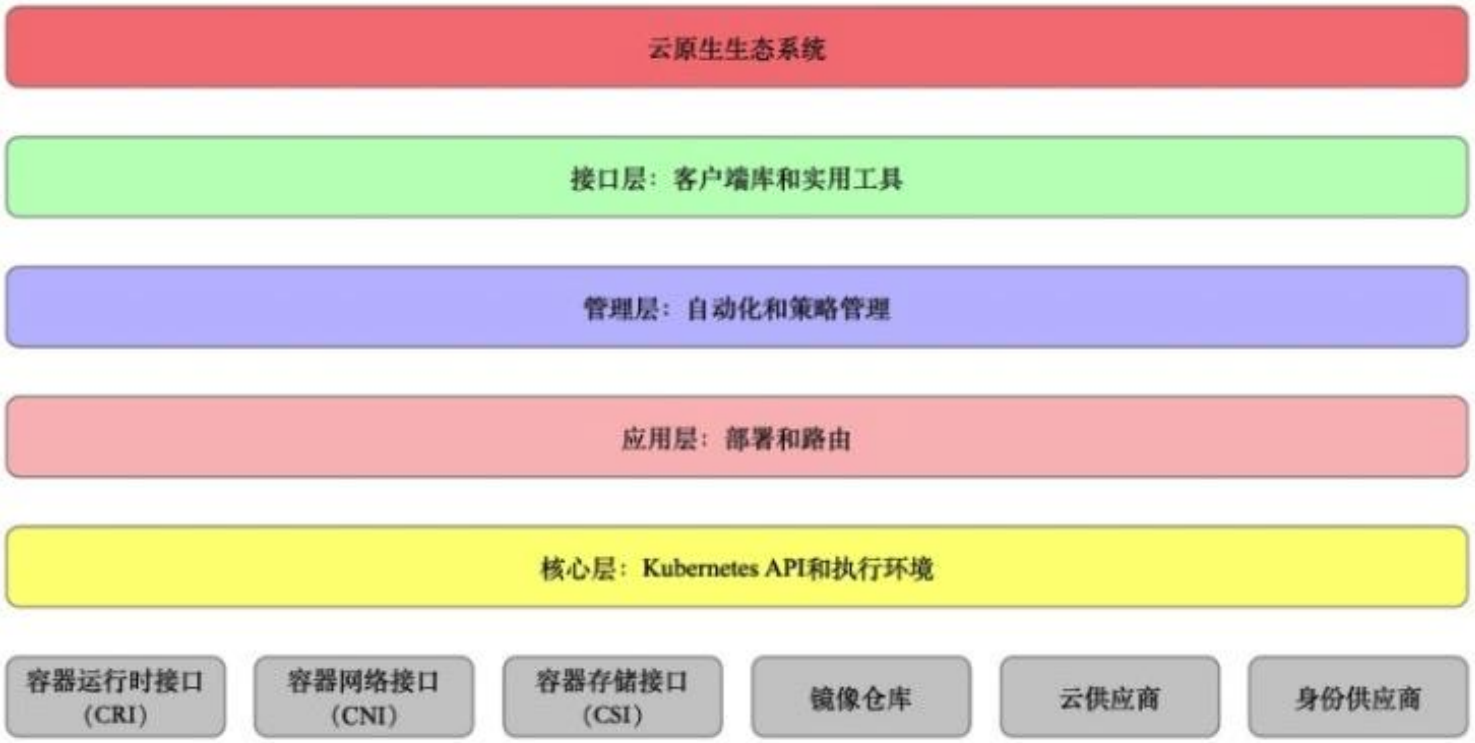
云·边多区域统一的容器托管服务

PaaS应用的能力下沉通道

边缘自治、兼顾安全、效率、稳定

- 边缘计算可能面临的一些挑战：
- 云边端协同：缺少统一的交付、运维、管控标准。
- 安全：边缘服务和边缘数据的安全风险控制难度较高。
- 网络：边缘网络的可靠性和带宽限制。
- 异构资源：对不同硬件架构、硬件规格、通信协议的支持，以及基于异构资源、网络、规模等差异化提供标准统一的服务能力的挑战

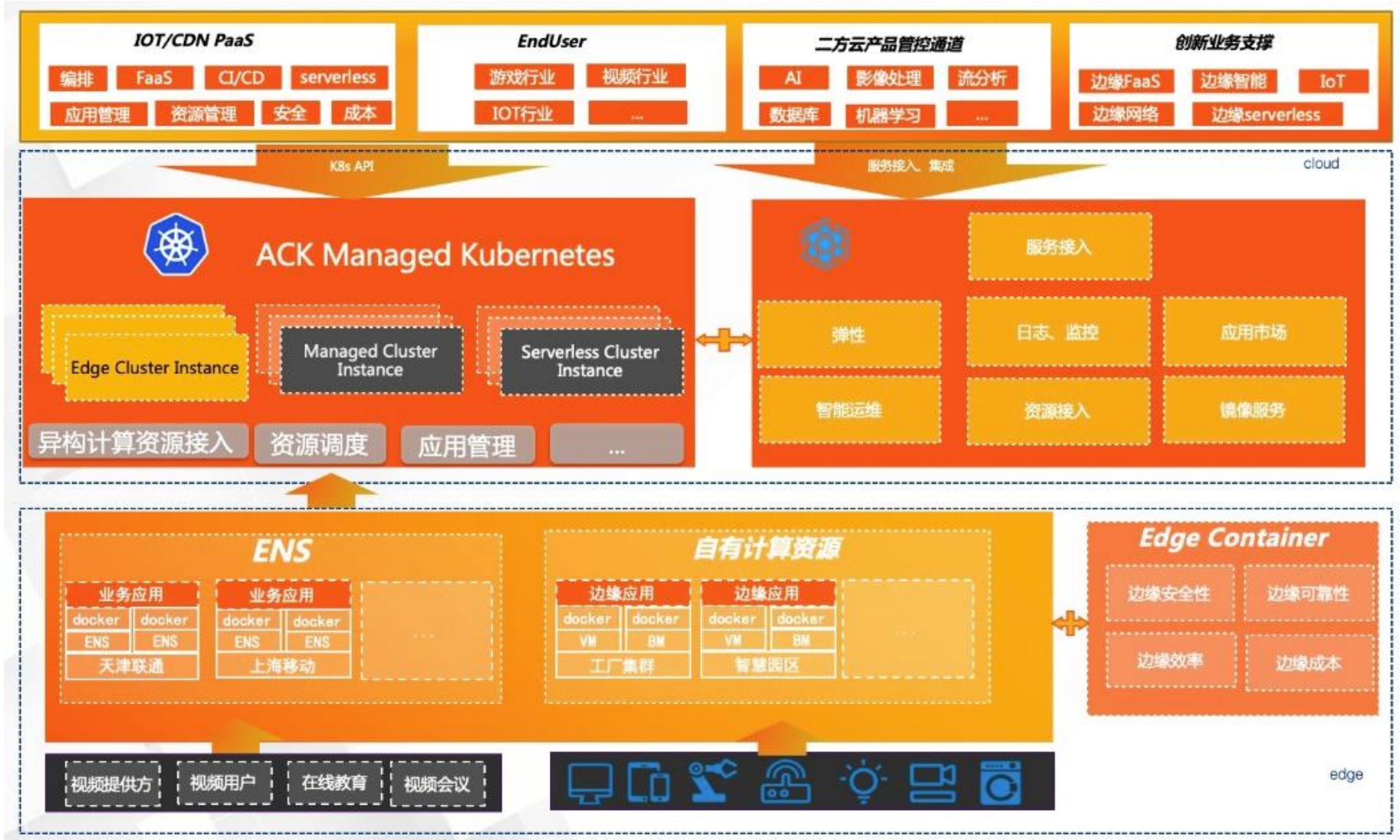
Kubernetes基本概念



- 核心层：Kubernetes最核心的功能，对外提供API构建高层的应用，对内提供插件式应用执行环境
- 应用层：部署（无状态应用、有状态应用、批处理任务、集群应用等）和路由（服务发现、DNS解析等）
- 管理层：系统度量（如基础设施、容器和网络的度量），自动化（如自动扩展、动态Provision等）以及策略管理（RBAC、Quota、PSP、NetworkPolicy等）
- 接口层：kubectl命令行工具、客户端SDK以及集群联邦
- 生态系统：在接口层之上的庞大容器集群管理调度的生态系统，可以划分为两个范畴
- Kubernetes外部：日志、监控、配置管理、CI、CD、Workflow、FaaS、OTS应用、ChatOps等
- Kubernetes内部：CRI、CNI、CVI、镜像仓库、Cloud Provider、集群自身的配置和管理等

“Kubernetes——让容器应用进入大规模工业生产，已然成了容器编排系统的事实标准；”（图文引用自https://jimmysong.io/kubernetes-handbook/cloud-native/from-kubernetes-to-cloud-native.html）

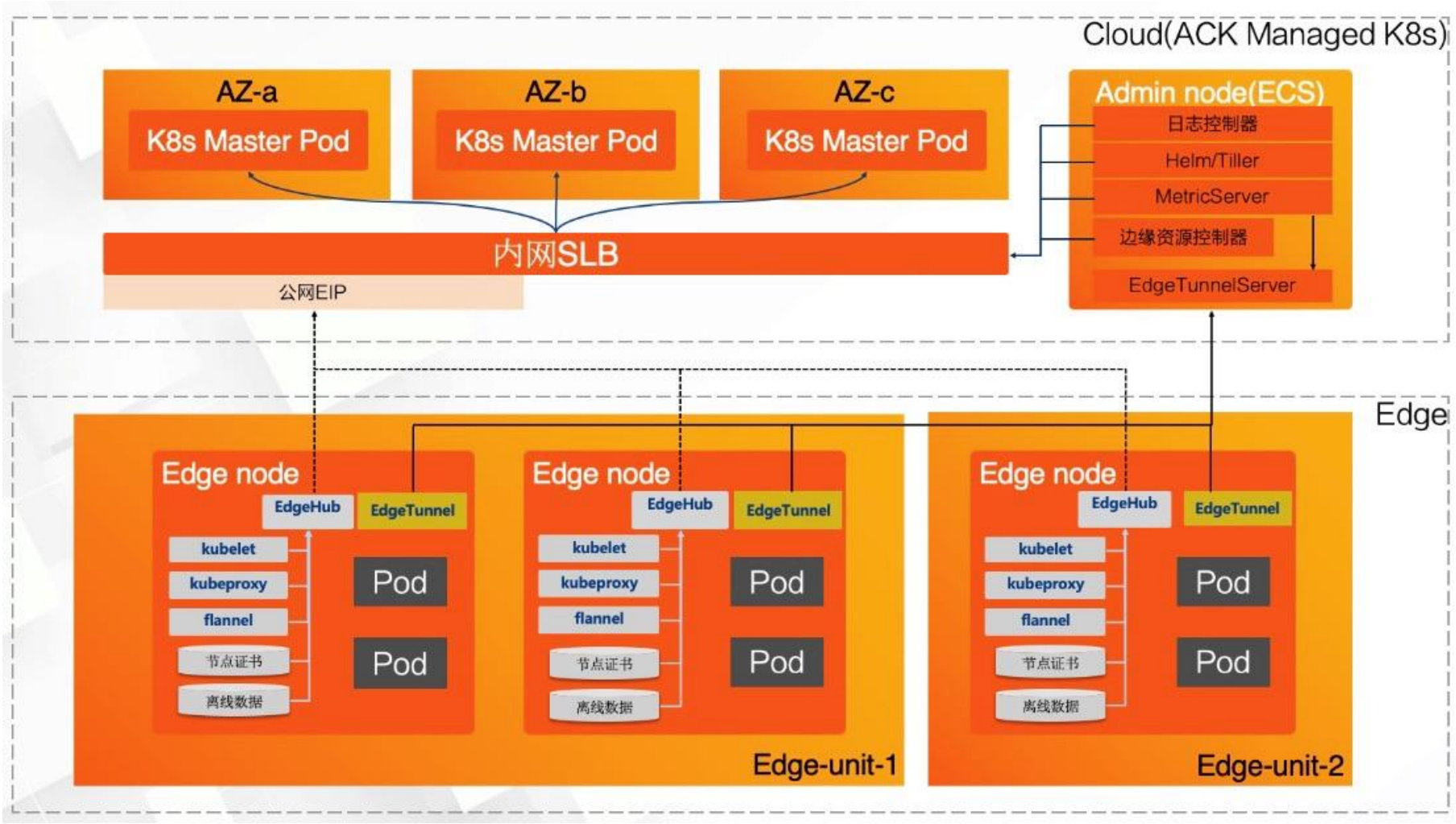
云端托管原生k8s



- 在云边一体的设计理念中，将原生的标准k8s托管在公共云上
- k8s免运维上面已经谈到，原生的k8s便于被上层业务系统集成却常常被忽视，
- 通过云厂商将kubernetes和其他云能力（弹性、日志监控、应用市场，镜像服务等）打通
- 无论是终端用户直接使用，还是做新业务创新，复杂度都大大降低
- 将云管控作为中心式服务，通过提供统一管控能力，反而非常适合管理边缘场景中零散分布的计算资源和应用，比如CDN服务、IoT业务等
- 最后，云原生的方式又可以让用户以往K8s经验用于新的边缘计算业务。

适度定制适配边缘场景

阿里云 | 奥运会全球指定云服务商

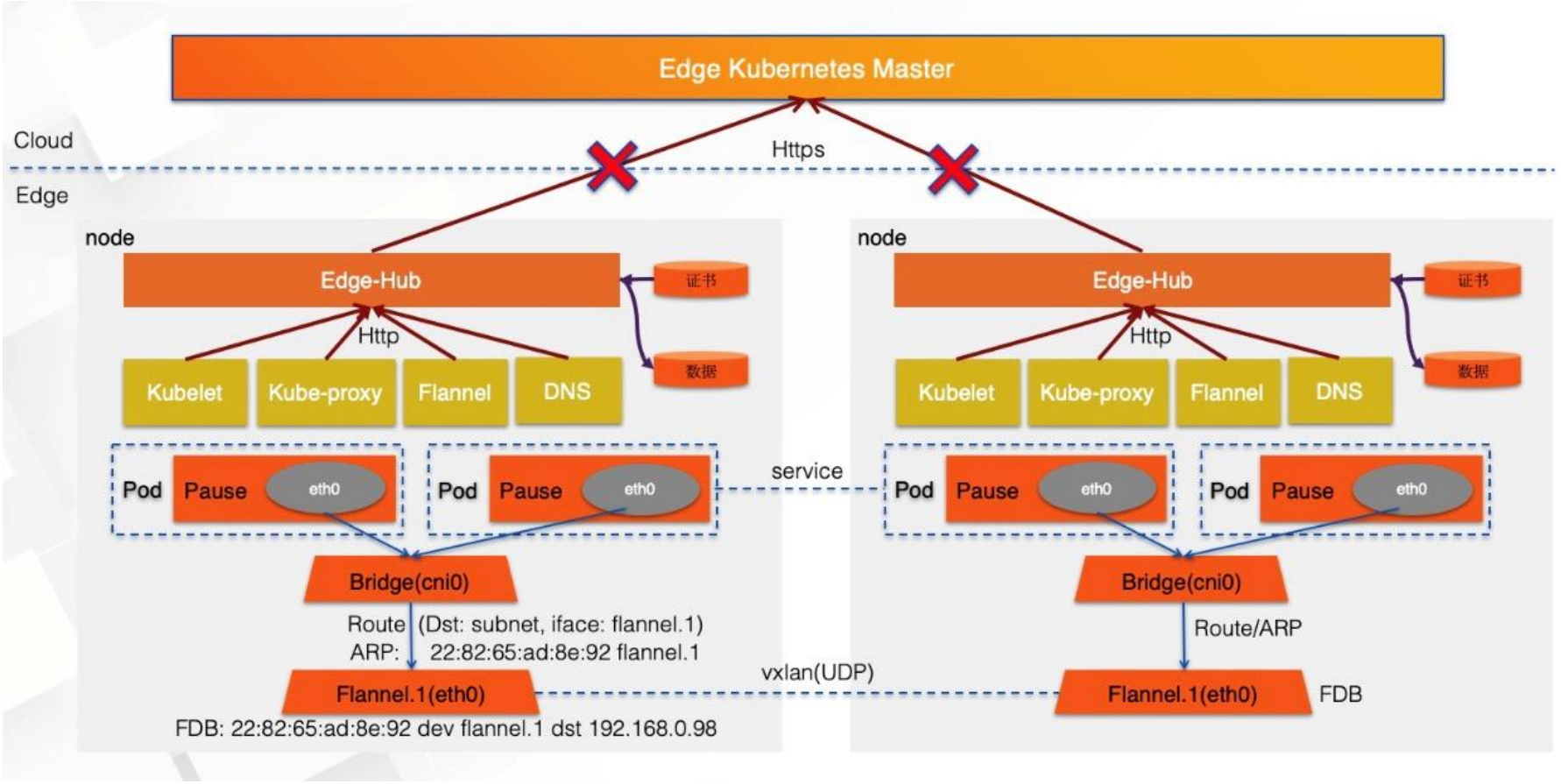


- **设计原则：**
- 标准化
- No overhead
- 边缘高可用
- 一致体验

- **方案要点：**
- 云管边架构；
- 边缘自治；
- Edge unit；
- 云边双向通道；
- 异构边缘节点
- 独占集群，解决多租问题；

技术细节和设计要点

边缘节点自治



目标：

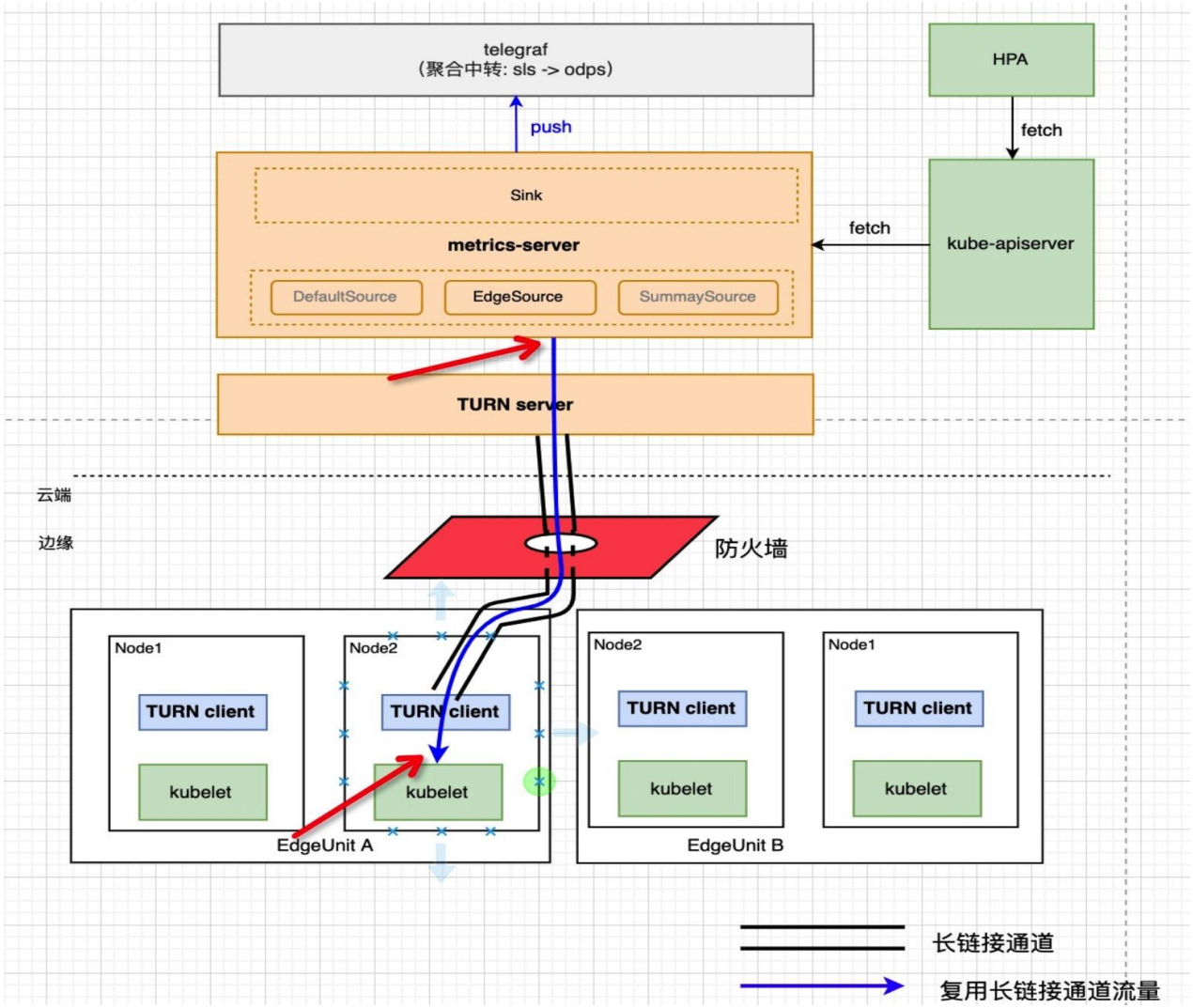
- 云边断网时，保障边缘业务连续性

方案：

- Edge-hub缓存云端的数据，所有系统组件均从Edge-hub获取数据
- 业务容器重启，Pod IP保持不变
- 宿主机重启时，flannel vtep的MAC地址保持不变
- EdgeHub上实现了诸多k8s api，除了给agent用，节点上承载的业务pod也可以使用，轻量又便捷；

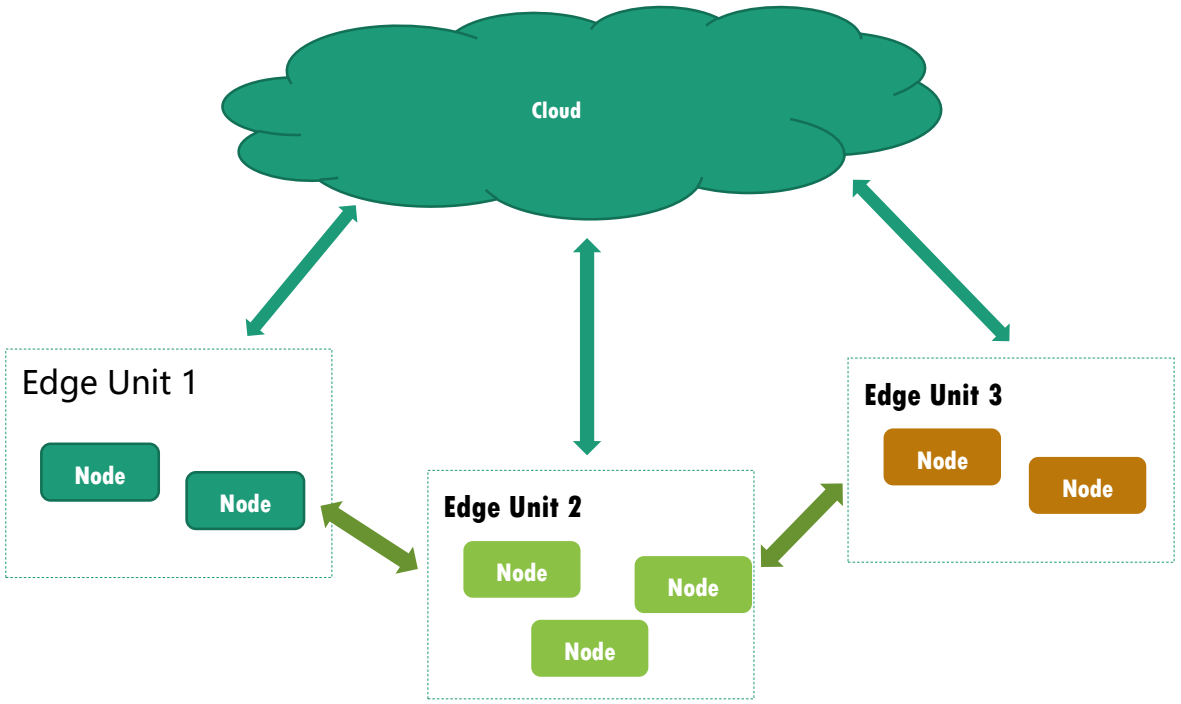
Kubernetes IoT/Edge WG组织的一个调查显示，30%的用户希望在边缘部署完整的Kubernetes集群，而70%的用户希望在云端部署Kubernetes的管理面并且在边缘节点上只部署Kubernetes的agent

原生运维支持



- 边缘场景，由于大多数边缘节点没有暴露在公网之上，主动的云到边的网络建链突然变成不可能，所有的原生运维api黯然失效
- 通过在管控与边缘worker节点之间建立反向通道，并和worker节点的生命周期完整联动，支持证书配置，支持websocktet等等。我们看到最终apiserver通过它按需中转，metricserver（k8s原生运维工具）通过它按需中转

边缘单元



Edge Unit：

- ENS：CDN机房（北京移动、上海电信...）
- IoT：工厂、园区、楼宇...

Edge网络：

- 方案一：集群内所有Node通过公网Vxlan构建Pod网络
- 方案二：EU内构建内网vxlan，EU间无vxlan(Pod网络不通)

Edge调度

- 单元化部署：亲和/互斥

Best practice：

- 同步数据流在EU内闭环，云-边、边-边通道主要是控制面或异步数据流
- EU内东西流量可直接使用Pod网络
- EU间调用使用Service和Ingress通过公网流转
- 单个Service/Deployment下的pod不跨EU

轻量化

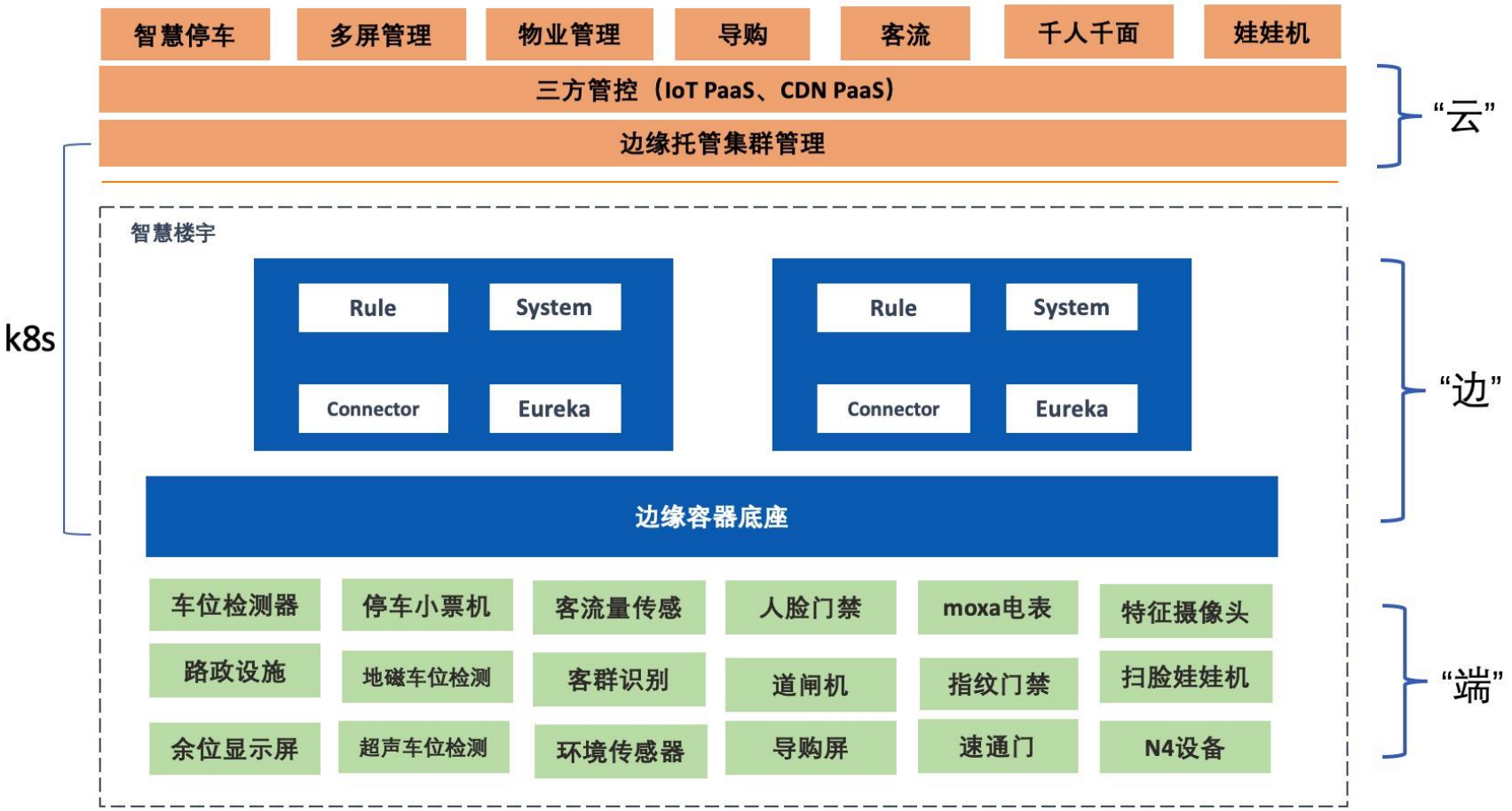
- 边缘算力的涵义颇广，包括：规模较大的CDN资源，通常被称之为“另一朵云”的边缘基础设施；也有浩如烟海的IoT的边缘设备，算力规模小但是数量庞大。在处理IoT场景云原生转型问题上，轻量化是绕不开的一环。我们知道，IoT业务场景充斥着大量的异构、小规格的边缘算力，像各种智能终端、设备，它们对资源的约束是极致的，难于接受额外的过多资源占用。所以，首当其冲的必然是管控层面的轻量化，简单介绍下目前常见的技术尝试：
- 管控组件的轻量化替代和压缩，如containerd替代docker，以及减少额外node sidecar的部署和开销的方案等等；
- 管控组件的裁剪，在k8s体系下对kubelet相关功能的裁剪和模块化，社区也有类似方案；

经典案例

边缘容器案例一

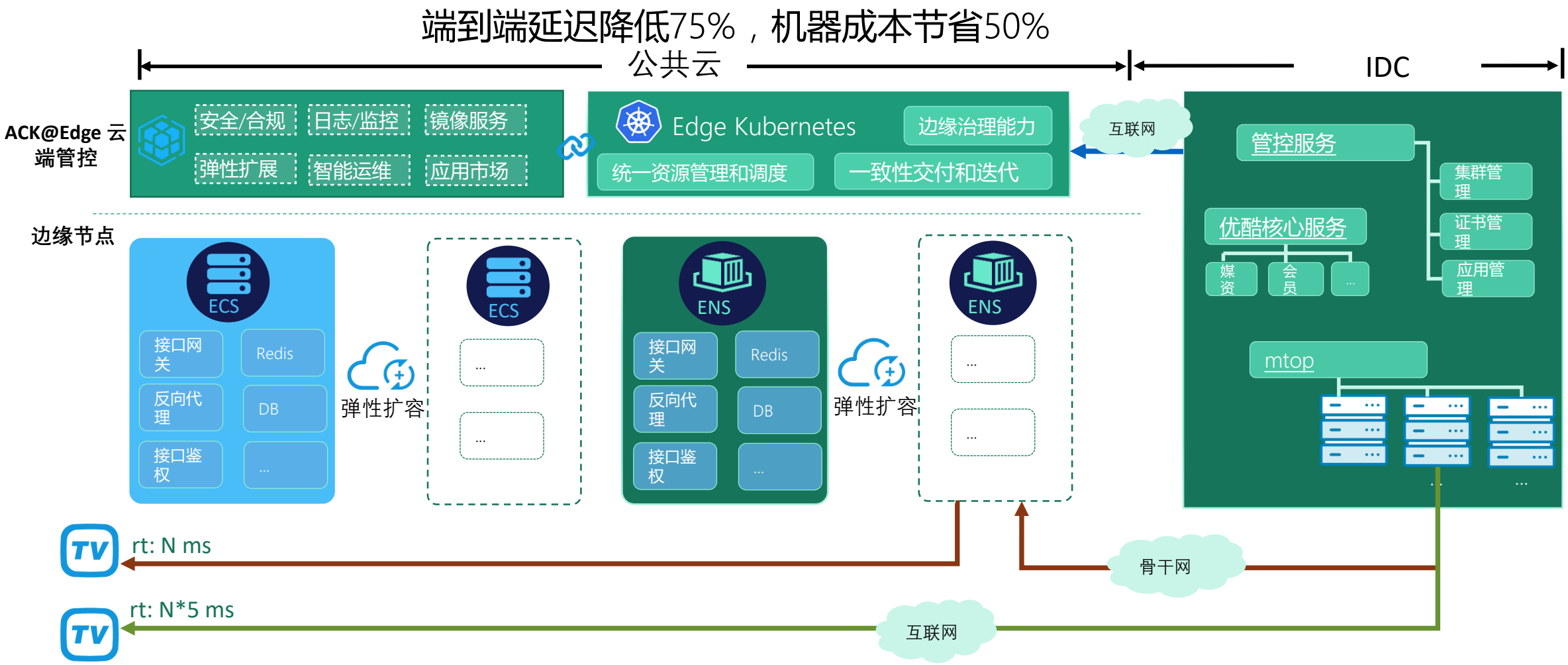
边缘容器支持IoT智慧楼宇

- 本地高可用服务
- 远程部署和运维
- 共享云端集群，优化成本
- 边缘节点弱连接



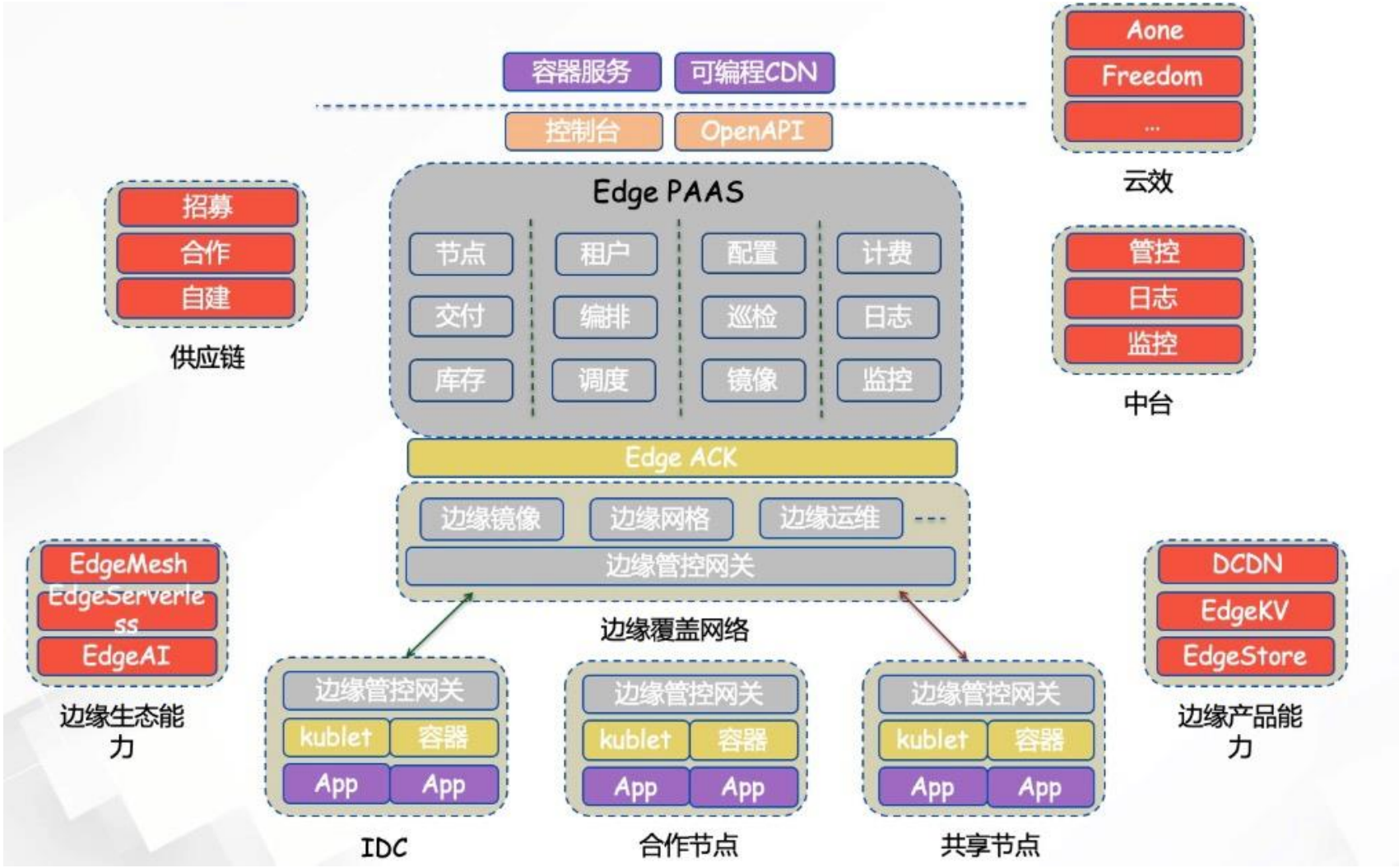
边缘容器案例二

基于ACK@Edge的优酷筋斗云



边缘容器案例三

基于ACK@Edge的CDN PaaS



钉钉扫码进群交流

关注“阿里巴巴云原生”公众号
获取第一手技术资料