



# GOTC 2023

## 全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

---

# OPEN SOURCE, INTO THE FUTURE #

---

### 「Cloud Native Summit」专场

本期议题：云原生边缘智能设备管理框架：KubeEdge DMI

赵然 刘琛林 2023年05月28日



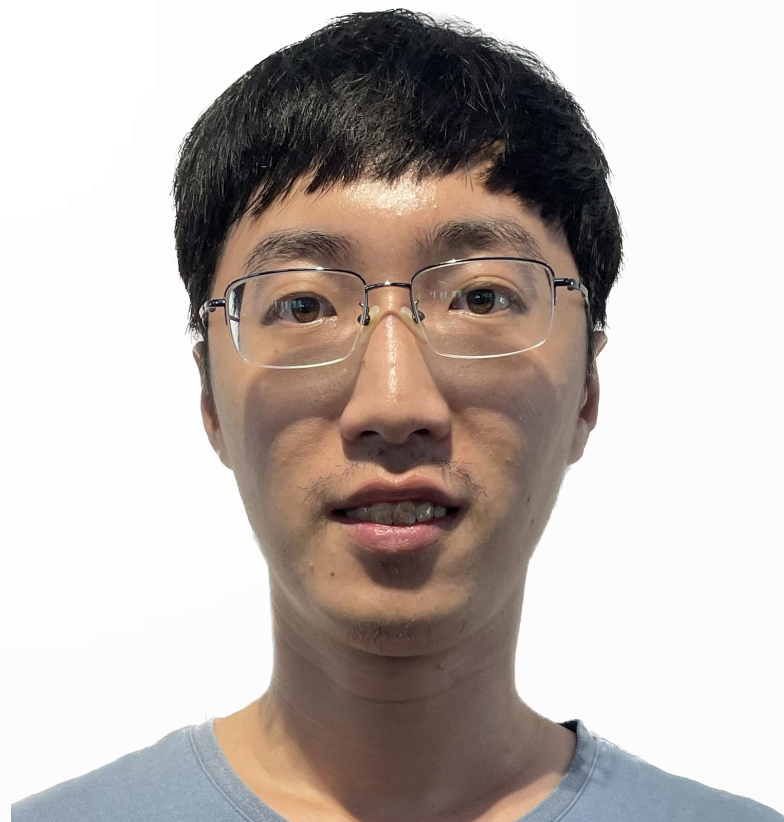
赵然

中国科学院大学 博士  
华为云 高级工程师  
KubeEdge社区 Member



刘琛林

DaoCloud开发工程师  
KubeEdge社区Member



全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

# CONTENTS

01

边缘计算与KubeEdge

02

设备管理接口DMI

03

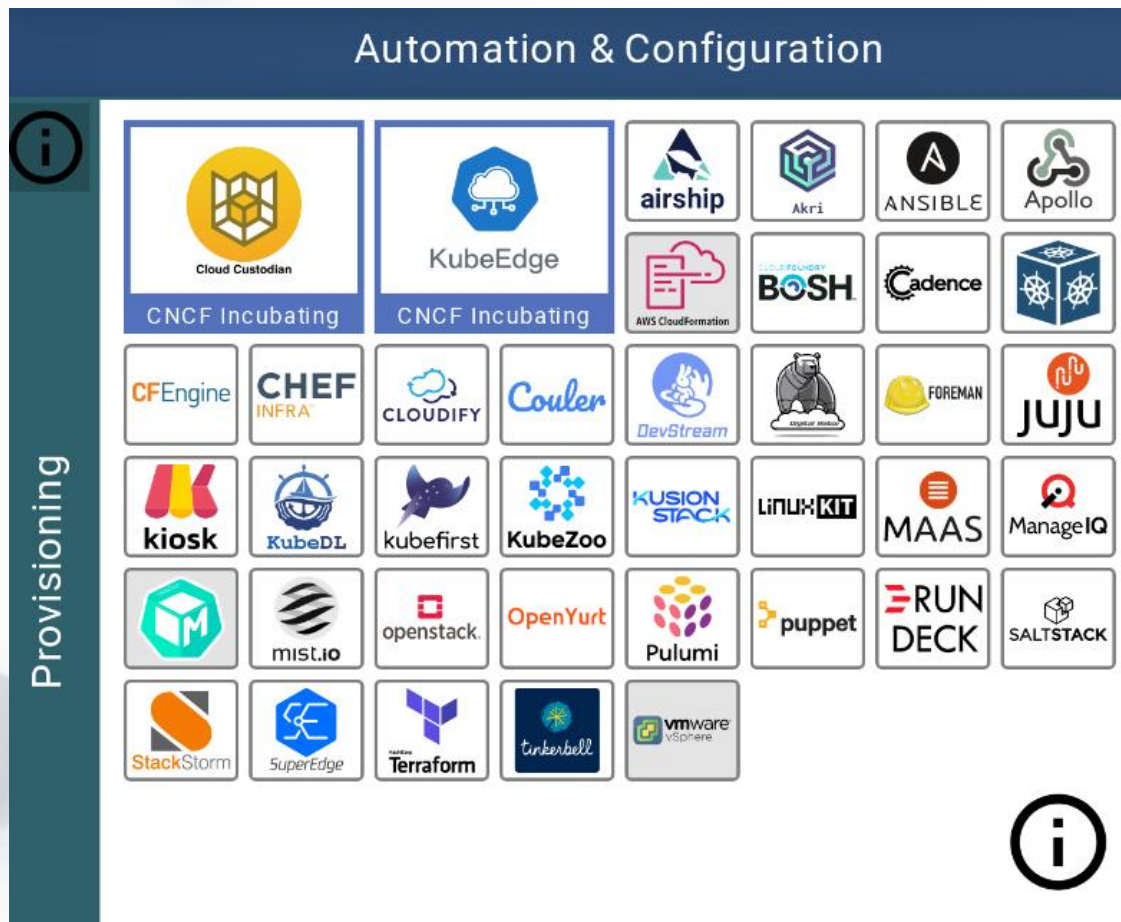
DMI架构设计与实现

04

后续工作规划

# KubeEdge&边缘计算

CNCF 首个云原生边缘计算项目：KubeEdge

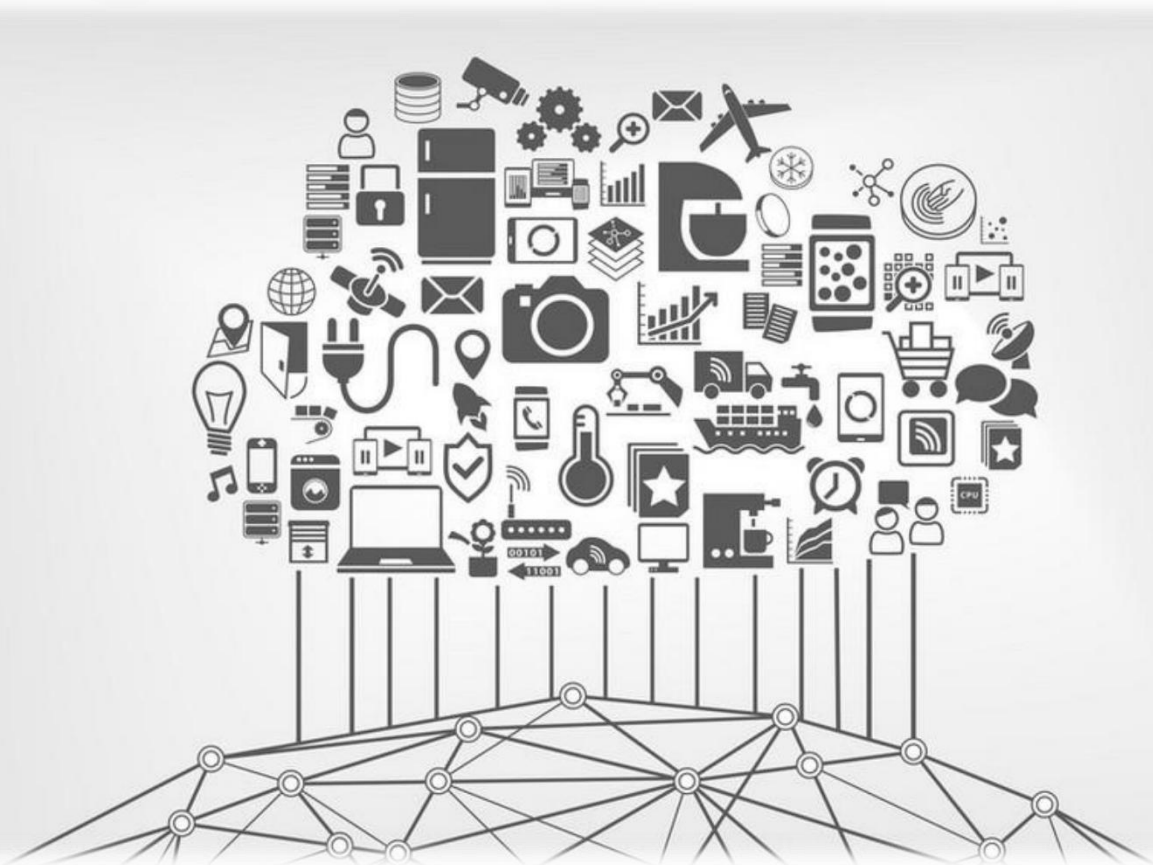


- 业界首个云原生边缘计算框架
- CNCF唯一孵化级边缘计算开源项目
- 160,000+ 全球开发者
- 1200+ 代码贡献者
- 80+ 贡献组织
- 多个 SIG 小组
  - ✓ SIG Device-IoT
  - ✓ SIG MEC
  - ✓ SIG AI
  - ✓ SIG Robotics
  - ✓ SIG Node
  - ✓ SIG Security
  - ✓ SIG Scalability

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

- 边云通信网络质量低，时延高，且边缘经常位于私有网络，难以实现双向通信
- 边缘资源受限，需要轻量化的组件管理运行边缘应用
- 边缘离线时，需要具备业务自治和本地故障恢复等能力
- 边缘节点高度分散，如何高效管理，降低运维成本
- 如何对异构资源进行标准化管理和灵活配置



- 物联网设备数量庞大
- 海量设备数据难以处理
- 设备通讯协议、模型、传输格式众多，难以维护
- 设备安全性不完善
- 缺少设备的监控指标
- 上手难度大，需要自写操作设备应用



## Device Management Interface (DMI)

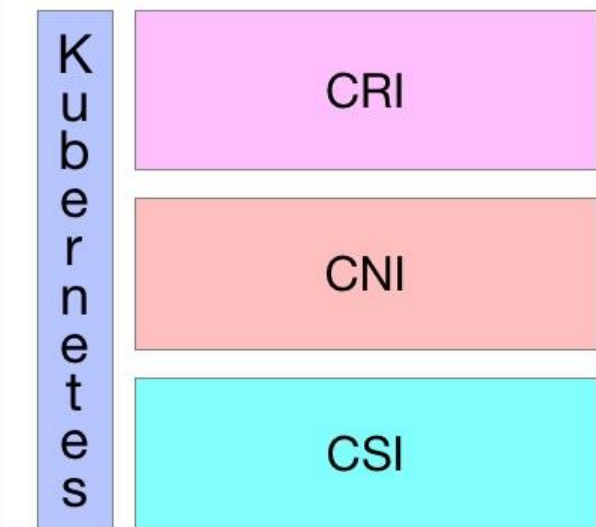
整合设备管理接口，优化 KubeEdge 设备管理能力，打造基于云原生技术的，覆盖设备管理、设备数据的设备数字孪生管理平台

- 云原生：device 作为一个 k8s 资源，生成虚拟化的数字孪生
- 设备管理：像管理 pod 一样管理设备的生命周期
- 设备数据：业务面数据与管理面解耦，支持边缘应用以多种方式获取设备数据

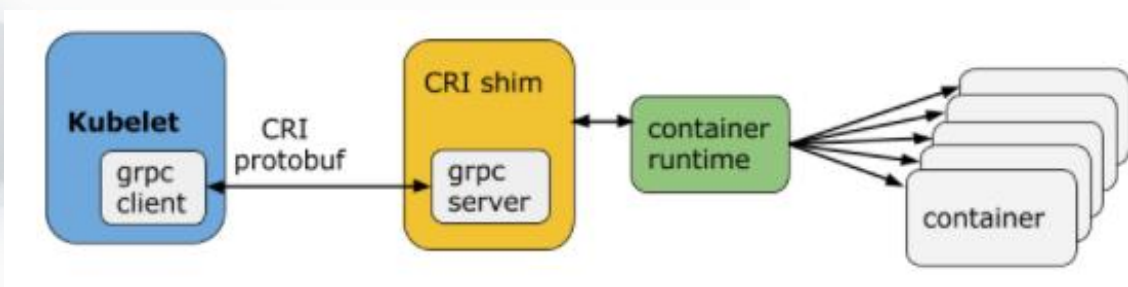
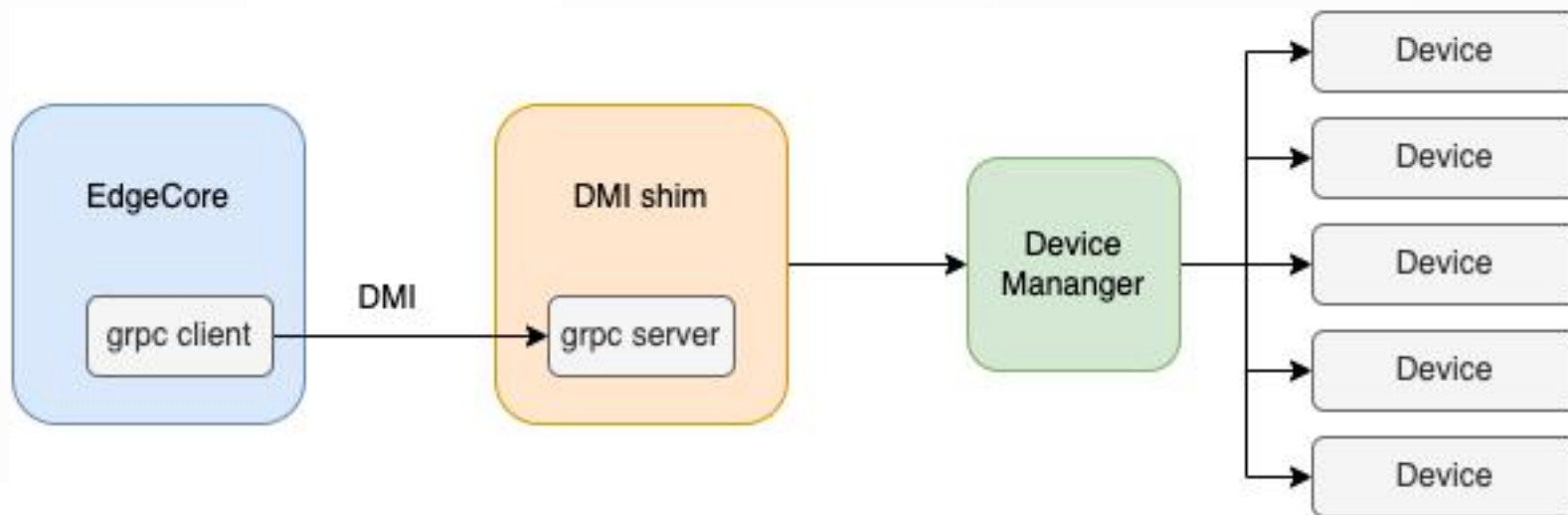


# DMI的设计与实现

Why DMI — 统一规范



类似CRI、CNI、CSI，为接入第三方设备管理框架（EdgeX、Akri、IOT）打下基础。

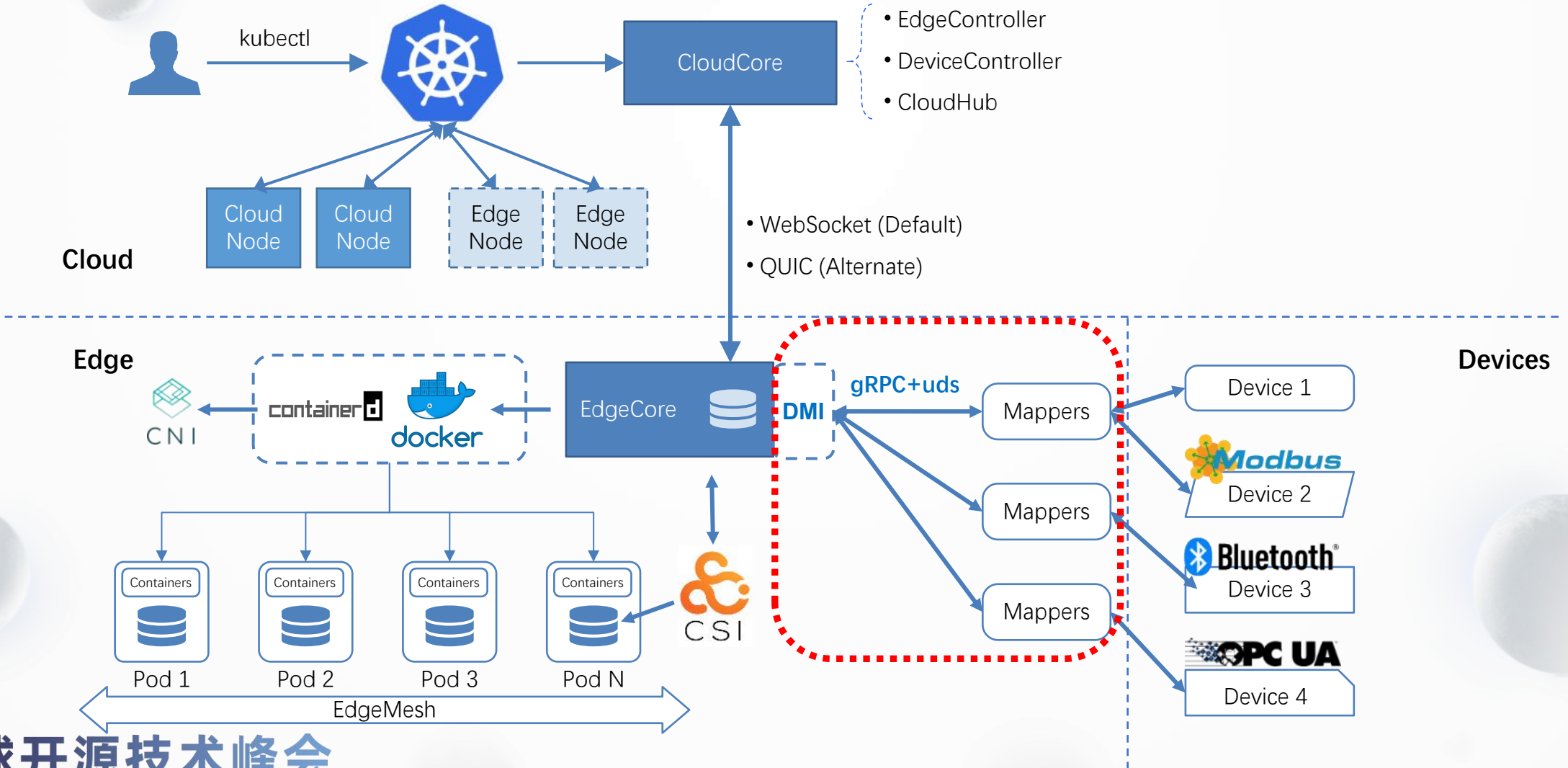


全球开源技术峰会

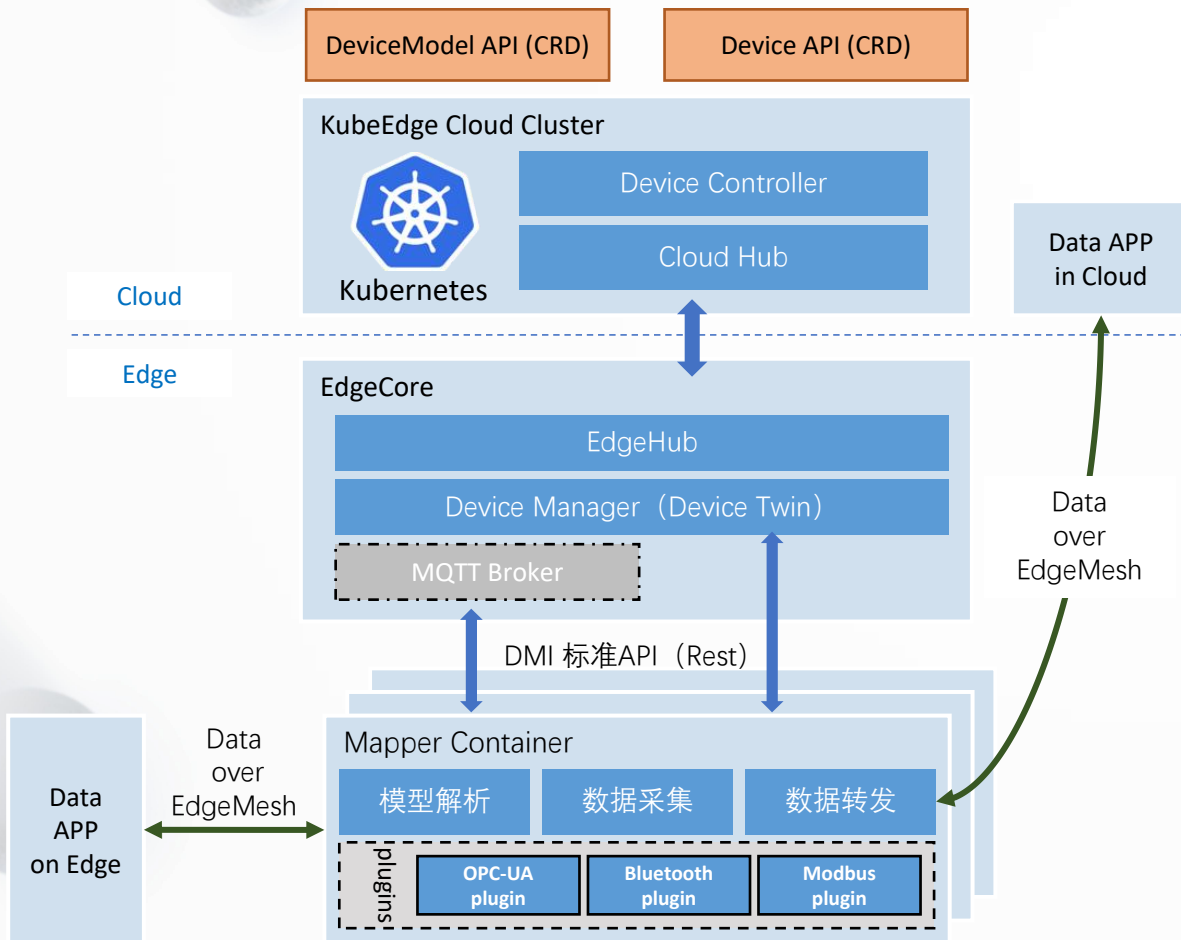
THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

# DMI的设计与实现

## Kubeedge+DMI

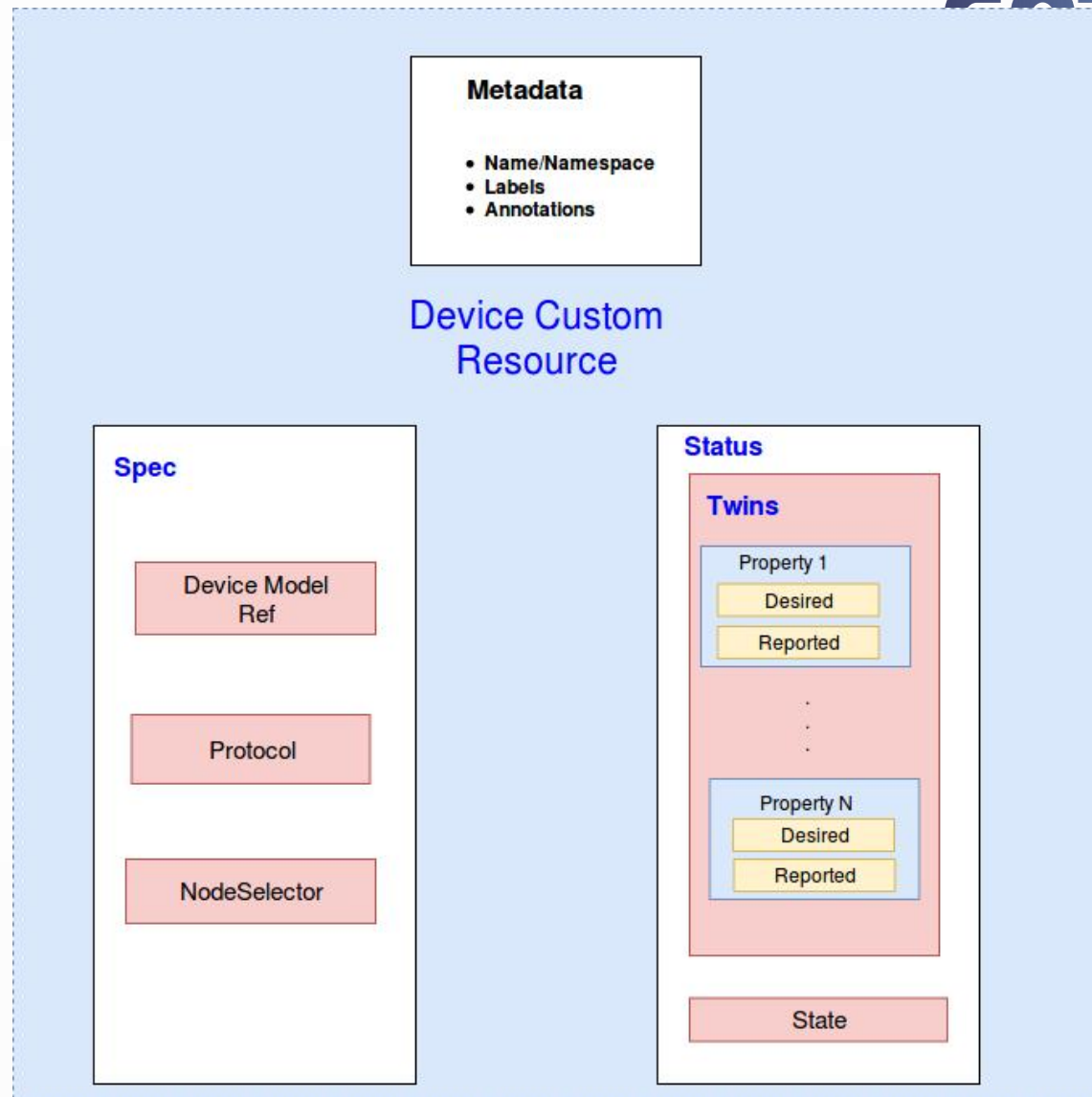
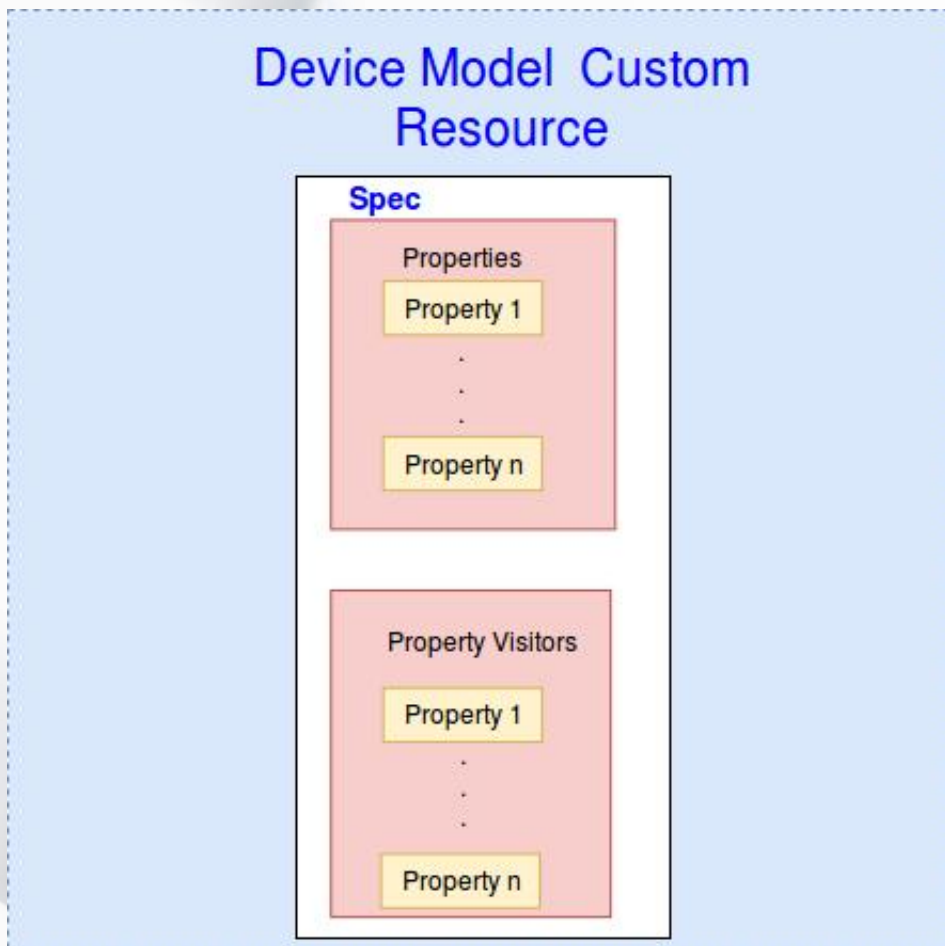


# DMI的设计与实现：架构



- ✓ 解耦设备管理面与设备业务面数据
- ✓ 设备数据微服务化，Device as a Service
- ✓ 帮助开发者专注于业务应用开发本身
- ✓ 降低云边通道拥塞可能，提高系统可用性
- ✓ 提供更灵活、更统一的标准化设备管理方式

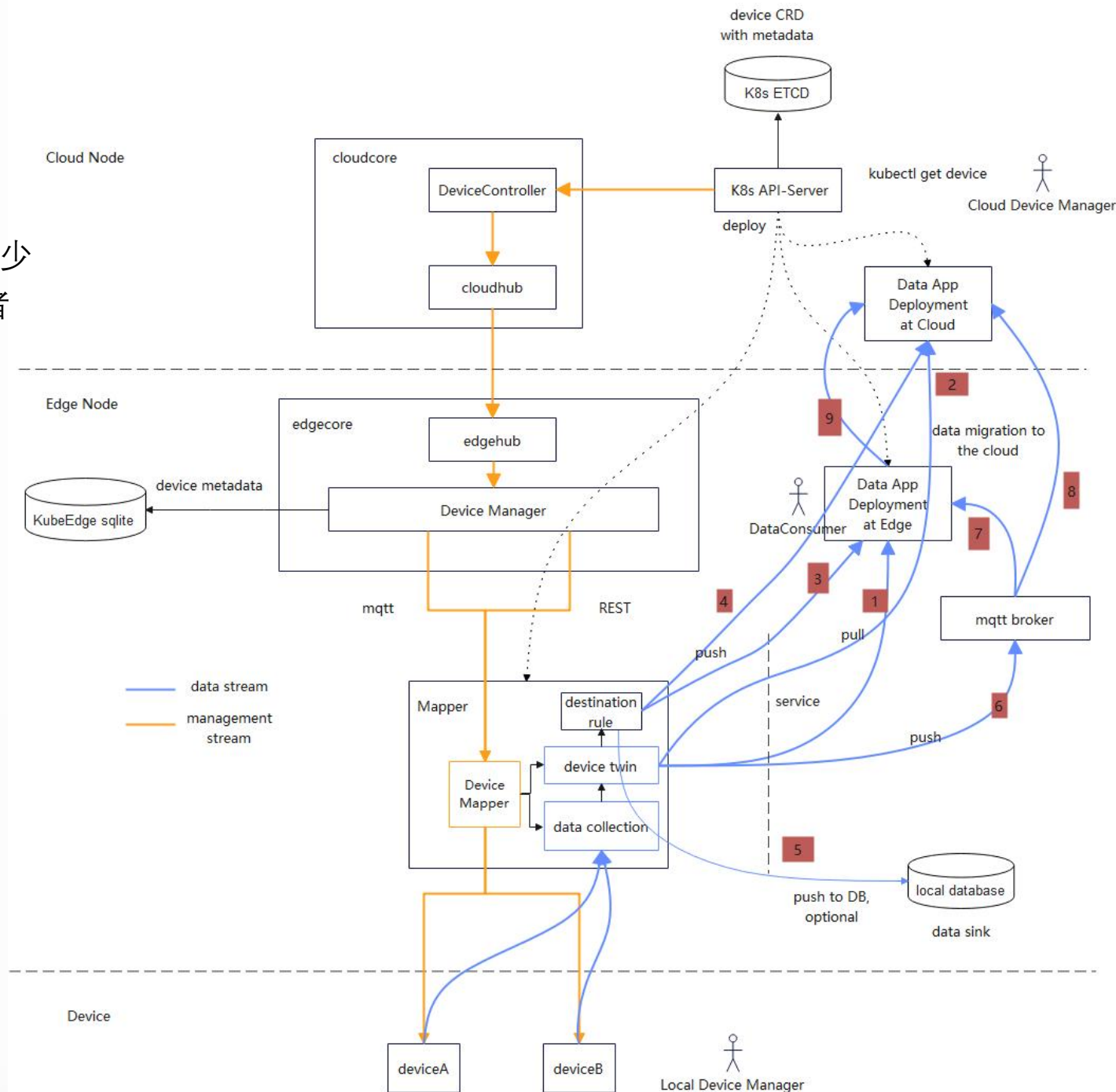
# DMI的设计与实现：设备模型定义





# DMI的设计与实现：管理面

- 管理面数据与业务面数据分离
  - 管理面数据保存在KubeEdge的ETCD，变化少
  - 业务面数据直接走数据面导出到数据处理者
- 管理面数据包括：
  - 元数据
  - 属性
  - 配置
  - 状态
  - 生命周期
- Device信息管理：
  - 缓存在KubeEdge sqlite中
  - 通过node+protocol指定Mapper
  - 通过Mapper注册接口返回值进行初始化
  - 通过添加、删除设备接口下发到Mapper
- Mapper通过grpc+UDS的方式实现接入

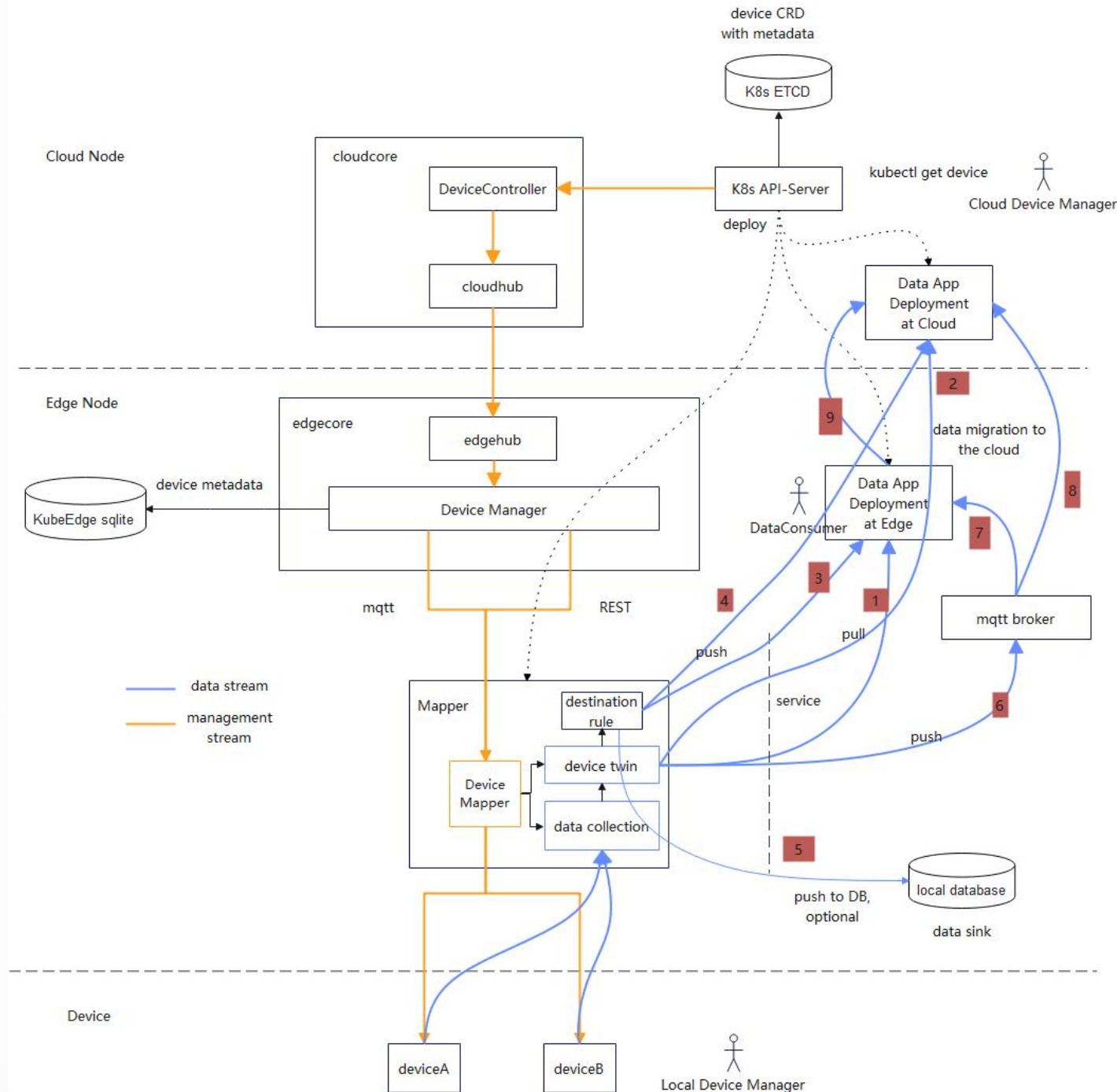


# DMI的设计与实现：数据面

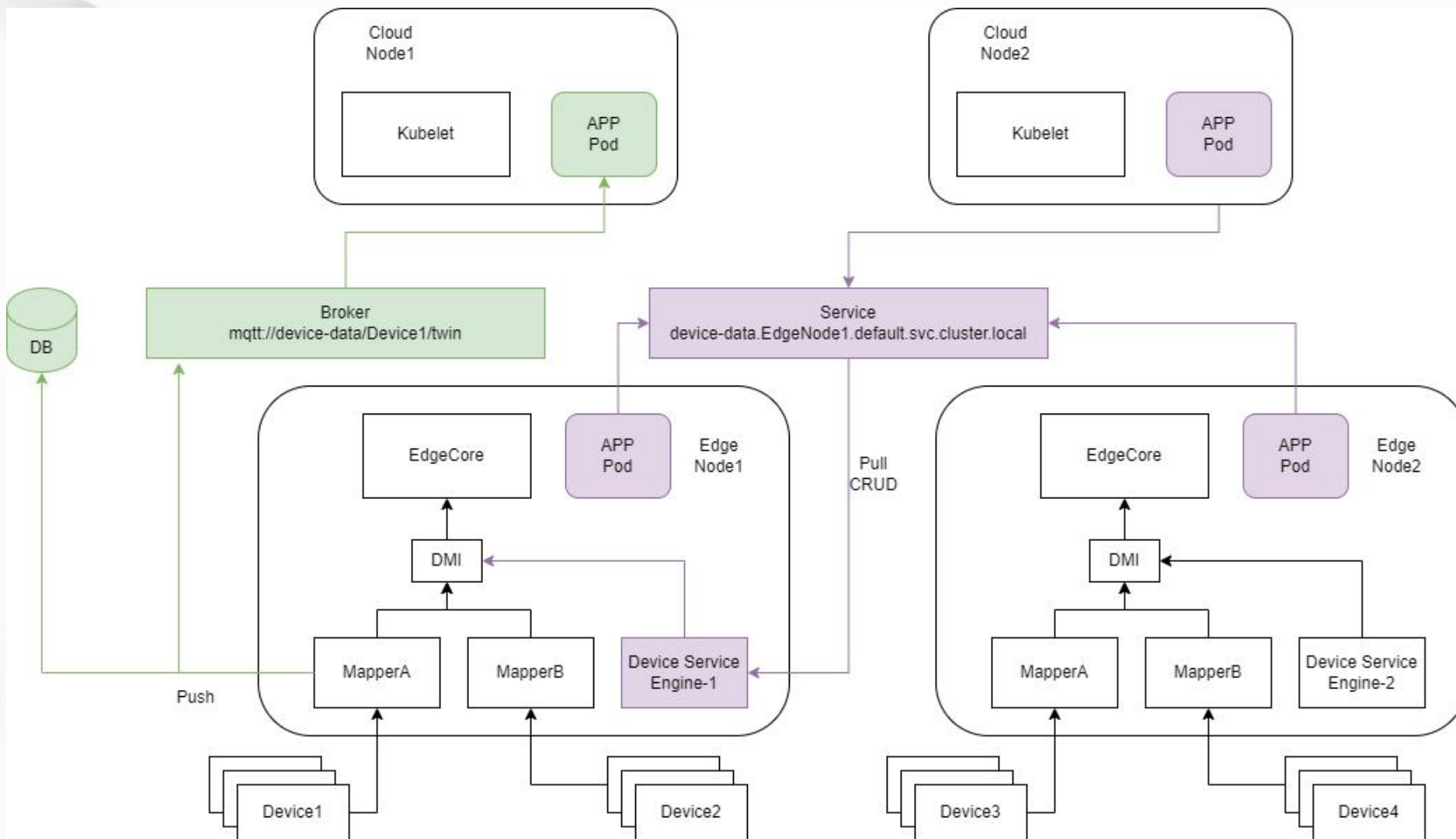
1. 边缘侧应用通过REST Service访问设备数据
2. 云侧应用通过REST Service访问设备数据
3. mapper通过配置REST目的地址，将数据推送到边缘侧应用
4. mapper通过配置REST目的地址，将数据推送到云侧应用
5. mapper通过配置目的地址，将数据推送到边缘侧数据库
6. mapper通过配置目的地址，将数据推送到mqtt broker
7. 边缘侧应用通过mqtt broker topic订阅设备数据
8. 云侧应用通过mqtt broker topic 订阅设备数据
9. 边缘侧应用处理数据后将处理结果传上云

全球开源技术峰会

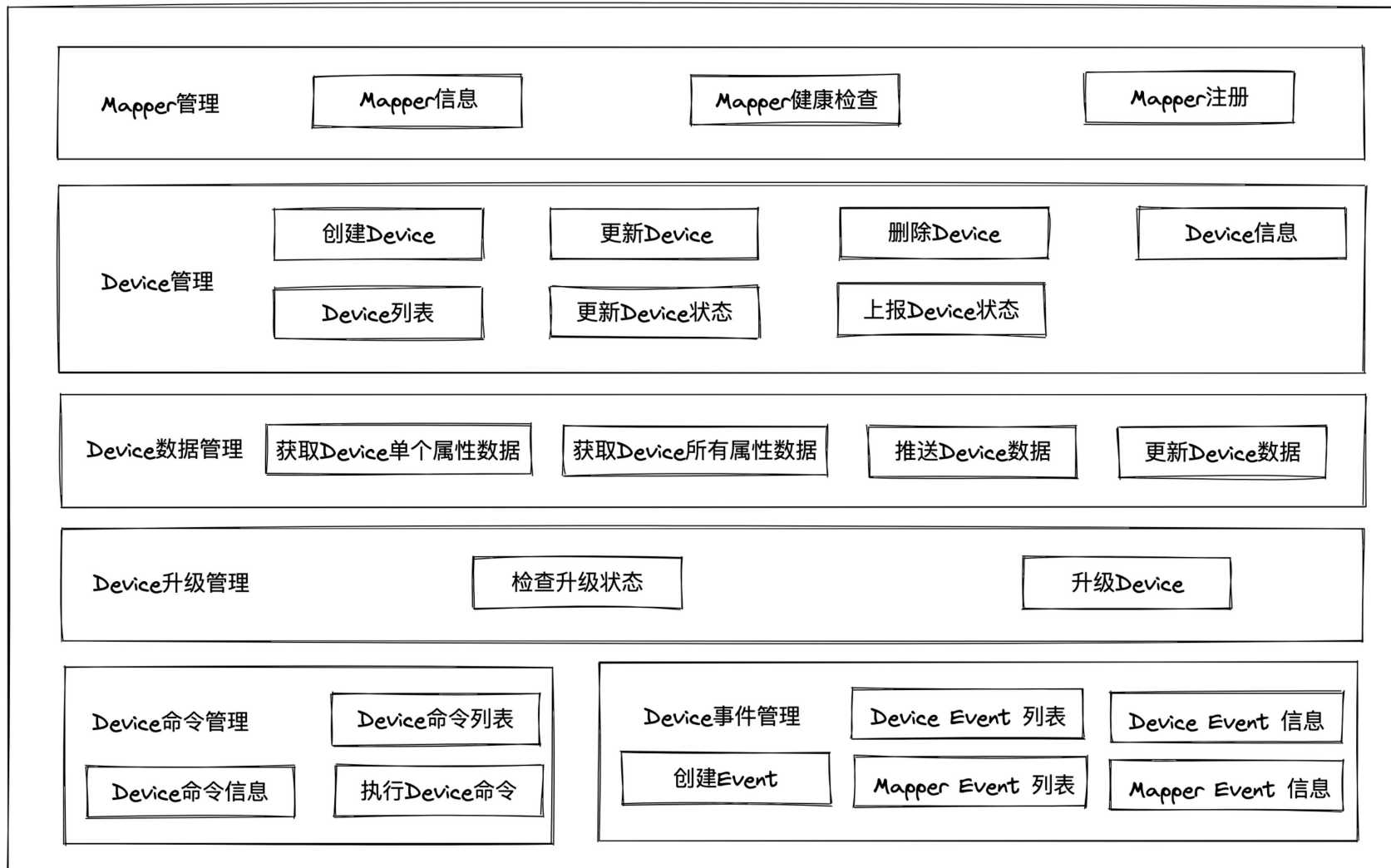
THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE



# DMI的设计与实现：数据面



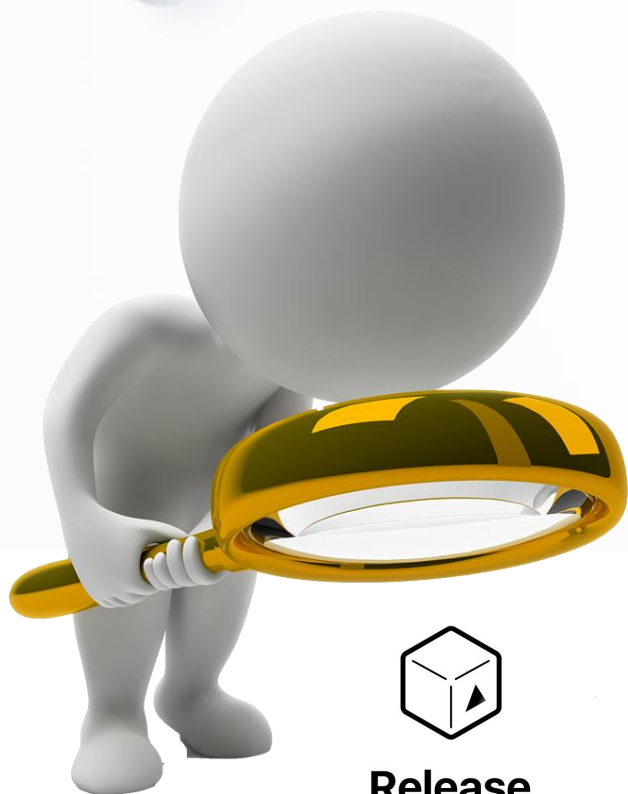
# DMI的设计与实现：接口定义





```
{} grpcserver package
  (*Server).CreateDevice function
  (*Server).RemoveDevice function
  (*Server).UpdateDevice function
  (*Server).CreateDeviceModel function
  (*Server).UpdateDeviceModel function
  (*Server).RemoveDeviceModel function
  (*Server).UpdateDeviceStatus function
  (*Server).GetDevice function
```

- 分为上行和下行两个独立的 interface
- 上行接口为Mapper访问 Edgecore, 注册Mapper以及上报设备状态, 使用公共UDS /etc/kubeedge/dmi.sock
- 下行接口为Edgecore访问 Mapper, 对设备的CRUD操作, 使用每个Mapper各自的UDS



## ➤ Alpha 版本

- ✓ 设备生命周期管理
- ✓ DMI Mapper Demo

## ➤ Beta 版本（进行中）

- DMI开发框架
- 设备命令操作管理
- 设备数据管理

## ➤ GA版本

- 多平台对接
- 设备升级管理
- 设备事件管理

# 加入社区

GOTC



<https://github.com/kubeedge>



<https://kubeedge.slack.com>



<https://space.bilibili.com/448816706>



公众号



交流群

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

THANKS