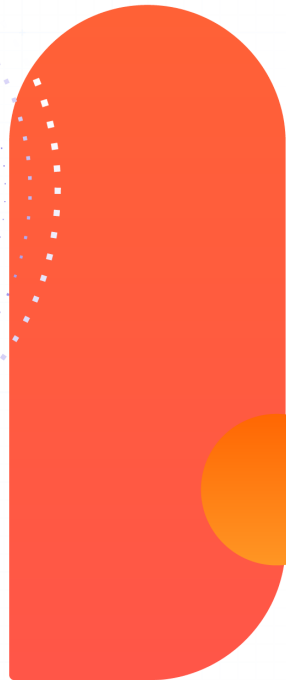
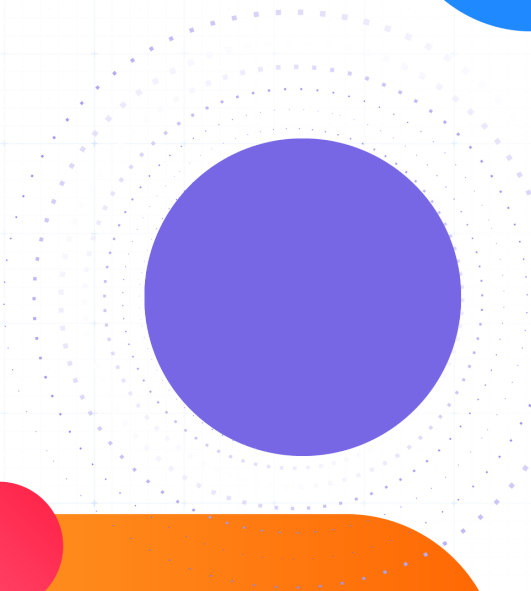
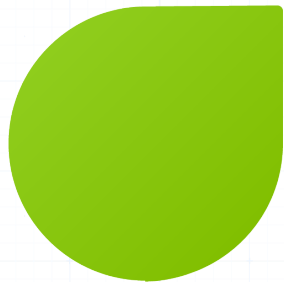


# 云原生在无人驾驶云脑的应用实践

- 驭势科技 · 郭祖龙 -



1. 自动驾驶与云脑

2. 云脑三件事儿

3. 总结与展望

4. 致谢

**UISEE**  
驭势科技



# 全场景 真无人 全天候

-  7x24小时服务
-  远程技术支持
-  现场部署
-  系统维护与升级
-  驻场服务
-  备品备件服务
-  培训服务

UISEE

# 核心技术——U-Drive<sup>®</sup>智能驾驶平台

## U-Drive<sup>®</sup>

为全行业、全场景提供AI驾驶服务

### 业界领先的自动驾驶算法



融合定位



感知识别



决策规划



执行控制



数据驱动

### 功能强大的云端运营管理平台



车辆运营管理



智能驾驶仿真



数据管理服务



高精地图服务



车路云协同

### 全功能车规级智能驾驶控制器



AI  
控制器

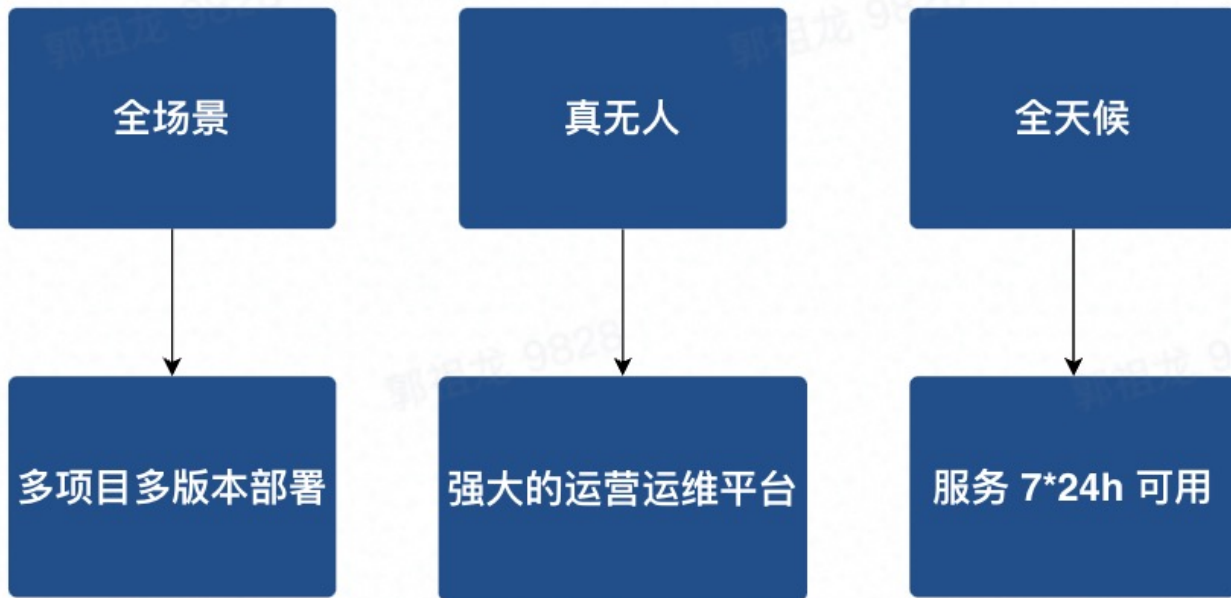
定位  
控制器

黑匣子

线控底盘  
控制器

V2X/远程  
控制器

## 云脑方向



UCloud 提供的云环境给予我们很大帮助

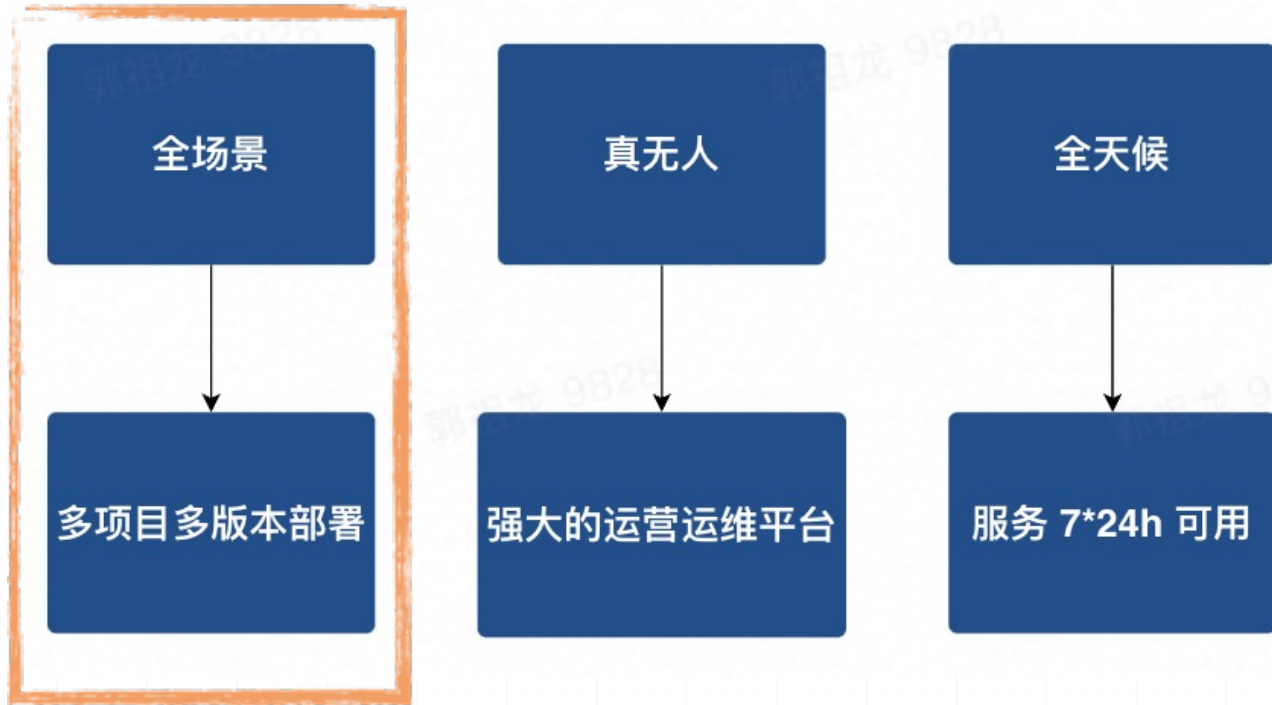
1. 自动驾驶与云脑

2. 云脑三件事儿

3. 未来目标

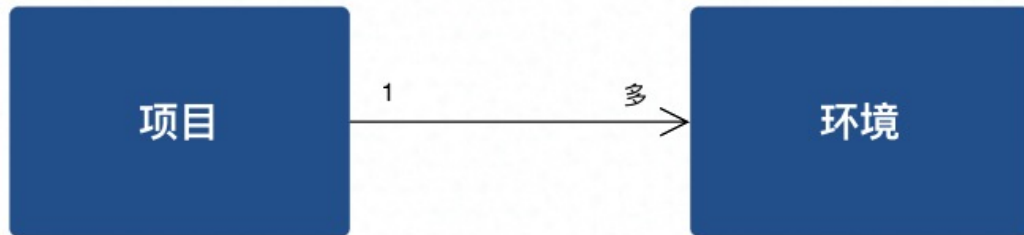
4. 致谢

# 云脑方向





## 支持多项目多版本部署



一个项目至少包括生产环境、预发布环境；

五个项目可以按照此方式部署运维；

五十个项目怎么办？

# SaaS 环境

统一运维

开箱即用

快速上线

按需付费

动态扩展



## 驭势 SaaS 会有哪些问题？

1. 项目功能复杂，甚至有些功能是冲突的，如何解决？

应用端不同；有的项目期望升级以使用新功能，有的项目希望保持稳定

2. 项目需求变化频繁，如何快速响应需求变化？

例如低电量的告警阈值，频繁在变

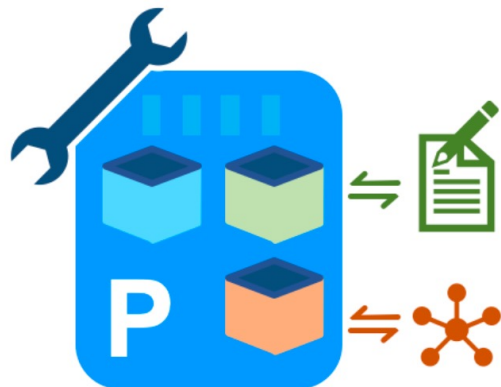
3. 随着项目增多，接入的设备端（车端）越来越多，服务能否支持动态扩展？

# 抓手：微服务 · Sidecar

1. 项目功能复杂，甚至有些功能是冲突的，如何解决？
  - 微服务化，功能模块拆分，可灵活搭配
  - 通过 Sidecar 部署 <https://openkruise.io/en-us/index.html>

## Sidecar 容器管理

支持在一个单独的 CR 中定义 Sidecar 容器，Kruise 控制器将在创建 Pod 时把它们注入到所有符合条件的 Pod 中。除此之外，通过更新 CR 中的 container Spec，也可以在原地升级 Sidecar 容器镜像。主容器管理和 Sidecar 容器管理的分离，极大地简化了开发者的合作。



# 抓手：网关

1. 项目功能复杂，甚至有些功能是冲突的，如何解决？

- 南北向网关

- 应用端通过 API 网关配置不同项目使用不同的应用

- 车云通信基于设备网关桥接转发，根据设备端项目信息路由至不同服务

- 东西向网关

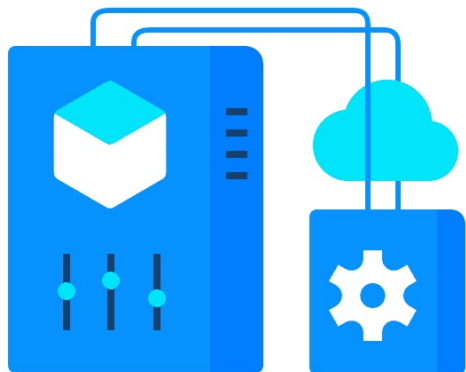
- 服务网格 Istio 等

- 服务及中间件尽可能合并

# 抓手：配置中心

2. 项目需求变化频繁，如何快速响应需求变化？

- 配置中心，服务支持动态配置加载 <https://nacos.io/zh-cn/>
- 设备端配置更新，车云双向同步机制



## 动态配置服务

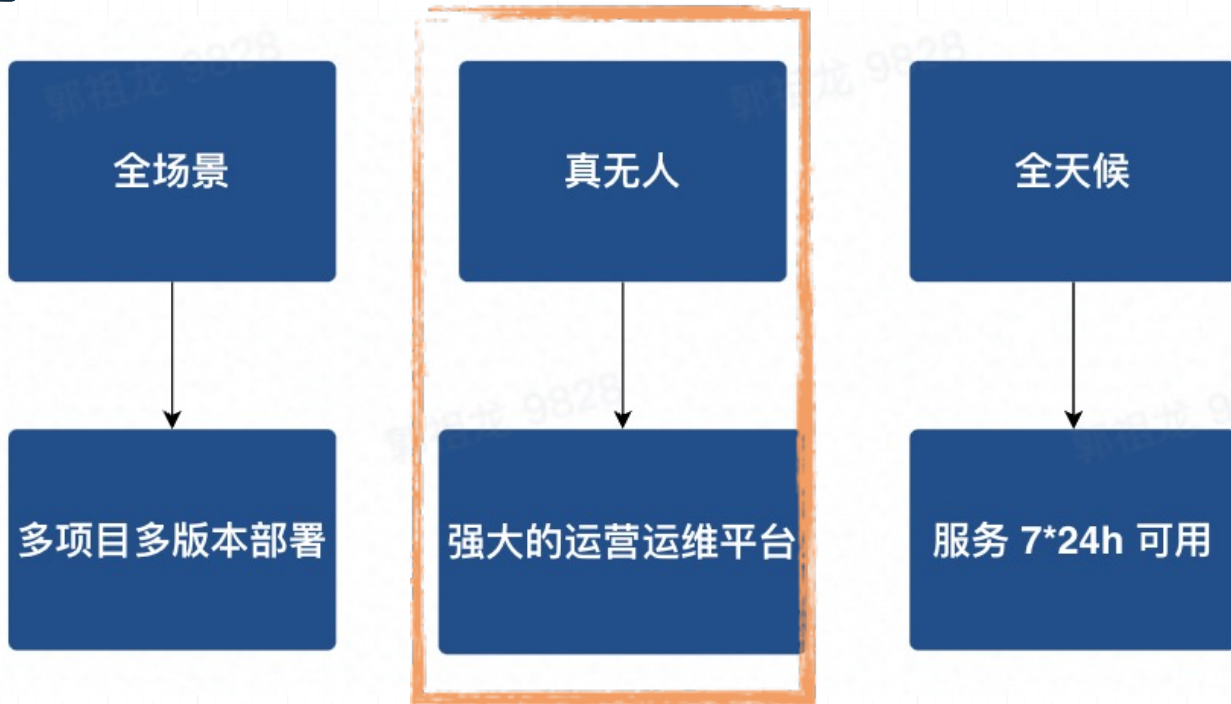
动态配置服务让您能够以中心化、外部化和动态化的方式管理所有环境的配置。动态配置消除了配置变更时重新部署应用和服务的需要。配置中心化管理让实现无状态服务更简单，也让按需弹性扩展服务更容易。

## 抓手：无状态·有状态服务

3. 随着项目增多，接入的设备端（车端）越来越多，服务能否支持动态扩展？

- 多开发无状态服务
- 选用「云原生」的有状态服务
- 如果要开发有状态服务，充分结合「云原生」思想

# 云脑方向





# 提供强大的运营运维平台

类比云商 IoT 平台，异同：

1. 自动驾驶车端是一个「庞然大物」
2. 自动驾驶系统中的数据量极为可观
3. 自动驾驶车端需要充分考虑弱网络场景
4. 自动驾驶车端软件包更复杂
5. 运营运维平台场景更加灵活

相同：设备影子、设备管理、安全认证、物模型等



### 提供企业级管理工具

提供一站式的物联网设备和数据管理

#### 物理设备数字化建模

可对物理世界的设备数字化建模，做到协议标准化和数据结构化，加速应用开发流程

#### 低延时远程控制设备

支持远程实时控制设备，结果同步返回，延时在50ms内

#### 设备归属可灵活转移

设备分发服务支持设备跨地域、跨账号、跨实例自由分发

#### 设备数据可实时存储

企业版实例自带时序存储，可实时写入和查询设备上报的结构化数据

### 多种低成本远程运维手段

减少已出货物联网设备的人力运维成本

#### 实例级监控报警

提供实例级别的指标监控和报警，可实时感知业务异常变化

#### 全链路日志分析

设备和平台全链路日志记录，可通过关键字、TraceID、时间等维度快速检索分析

#### 设备OTA升级

支持设备固件、软件等模块OTA升级，可指定差分、灰度、全量等多种升级策略

#### 设备智能诊断

根据设备状态、消息、报警等行为数据，快速分析问题设备



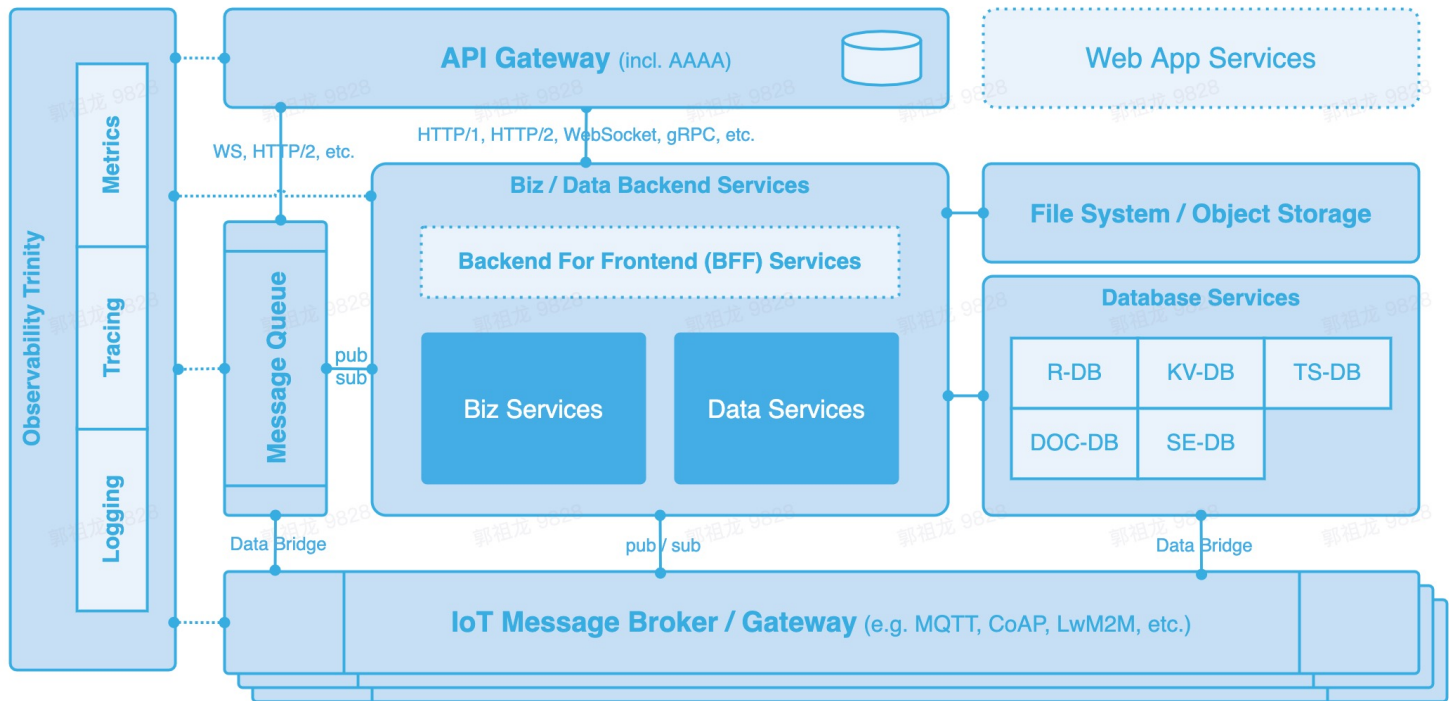
## 运营运维业务中台

相对于「前台」（应用类服务）  
和「后台」（核心业务逻辑）；  
相对于「数据」中台，主要关注  
「数据流」和「控制流」管理

## 泛物联网

面向多终端；  
在一定时间内有高频数据采集；  
可以实时的进行「端到端」的操  
控类指令下发

# 基于云原生的架构方案



## 设备数据怎么上传下达 🤔

### 【策略】中间人 (Broker)

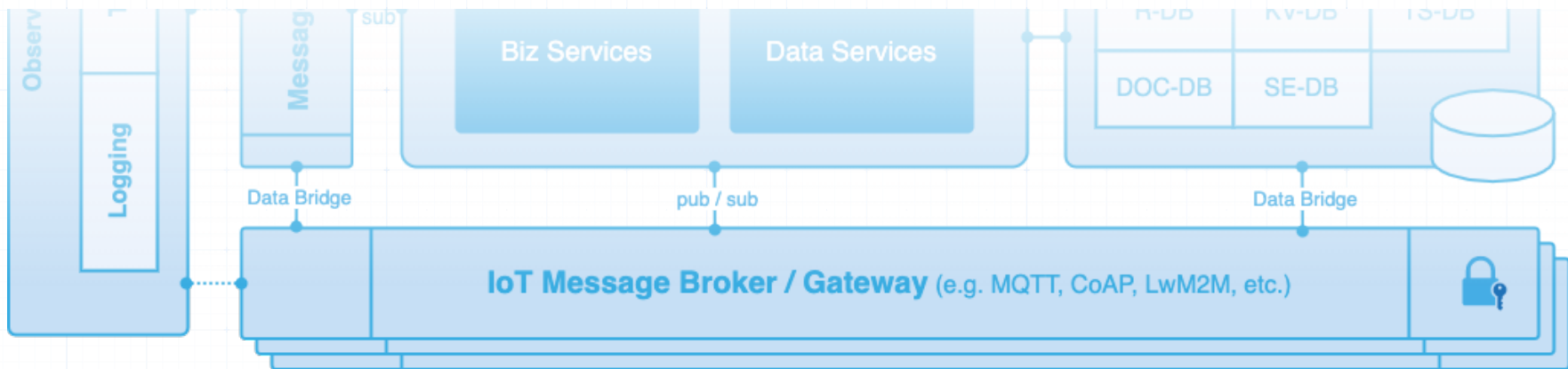
- 👍 部署灵活，私有云/公有云皆宜 (也可以直接使用云商的服务)
- 👍 连接方便，受防火墙干扰小
- 👍 发布/订阅模式便于双向数据传递
- 👍 若有网关能力，便于多协议串联
- 👍 若有数据桥接能力，便于读写分离

### 【协议】MQTT / CoAP

- 👍 最常用的工业级物联网通信协议
- 👍 轻量、高效 (相比 HTTP/1)、安全 (TLS)
- 👍 MQTT 具有 3 个 QoS 等级，可结合业务选用不同的服务等级 (常用 QoS 1 保证至少一次)
- 👍 MQTT 基于 TCP，弱网络环境可结合业务考虑引入基于 UDP 的 CoAP 协议进行补充

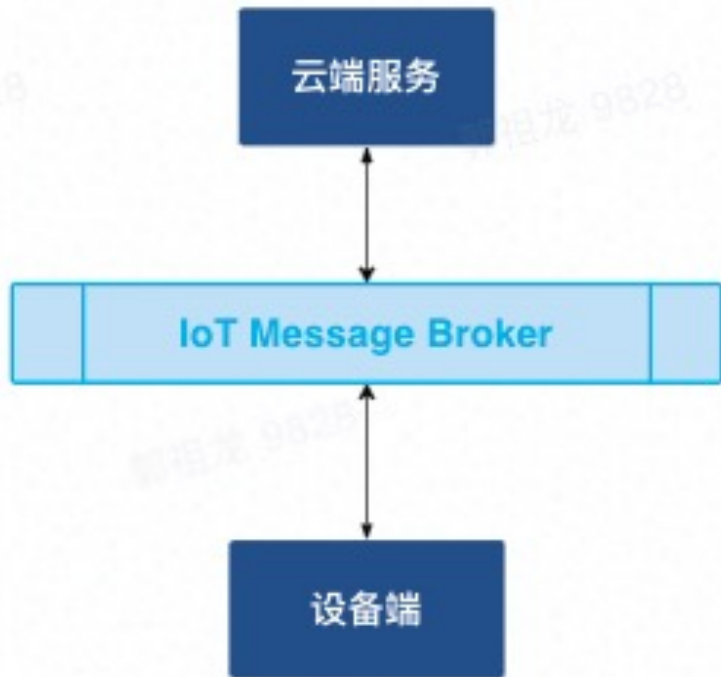
### 【推荐组件】EMQ X Broker

- 👍 基于 Erlang 的开源项目，可用性良好
- 👍 安装部署包形态丰富，支持 Docker
- 👍 支持海量设备接入，低延时消息路由
- 👍 具备网关能力，支持 MQTT 3.1.1 / 5.0，同时支持接入 CoAP / LwM2M 协议
- 👍 多种认证模式，商用版数据桥接能力丰富

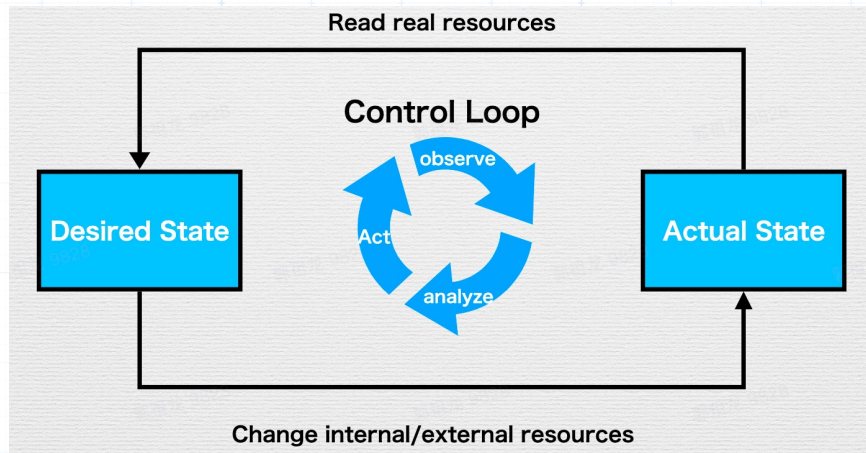


# 设备数据定义

1. 数据流向：数据上行与数据下行
2. 数据类型：状态与事件，举例子
3. 状态数据是一定频率产生的，可以容忍丢失；需要特殊的机制保证事件不能丢失
4. 通过学习 k8s 核心机制



# 声明式命令



1. 通过 API Server 控制资源 vs 通过设备影子操控设备

资源 <-> 设备 , API Server <-> 影子

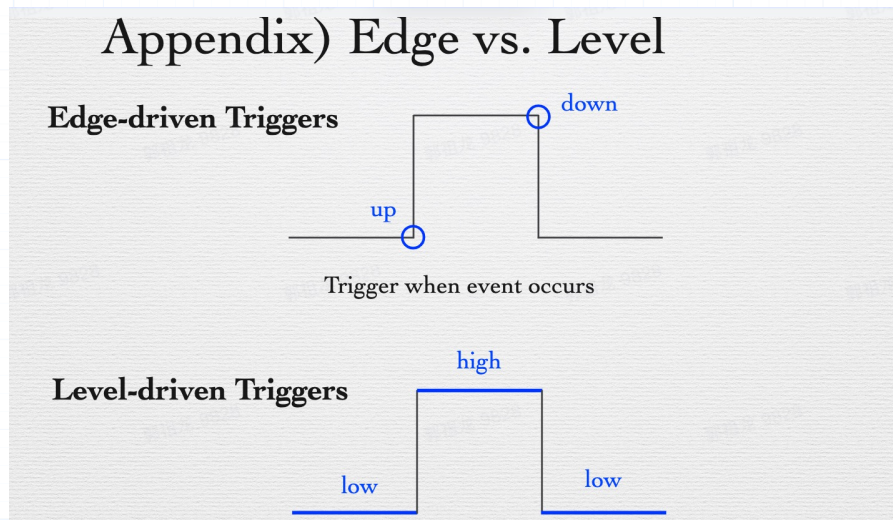
2. 实际状态与期望状态 vs 设备上报状态 , 判断状态是否符合预期

实际状态逼近期望状态 , 判断上报的状态是否等于预期状态 , 超时重试

# 事件上报机制

k8s Controller 事件触发机制：

1. 边缘触发
2. 水平触发



边缘触发是根据状态的变化产生，是真正的事件。水平触发类似于状态上报，是一种对边缘触发的弥补方式，通过此方式保持数据一致性，避免因为边缘触发产生的事件丢失导致数据不一致

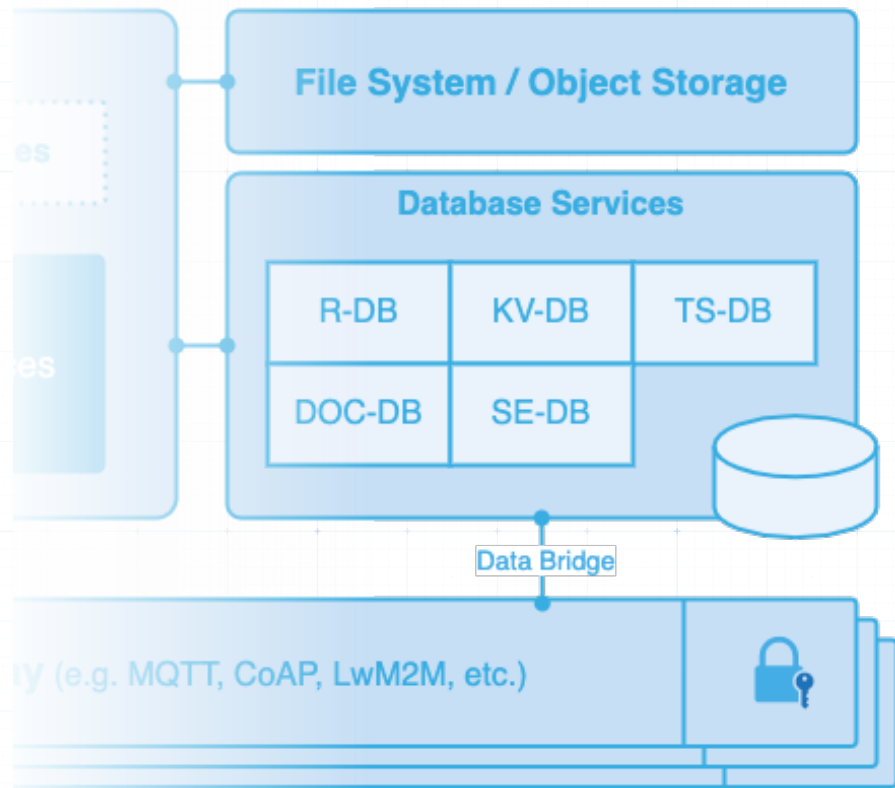
## 云端数据怎么存储 🤔

### 【数据库】一定要结合业务做选型

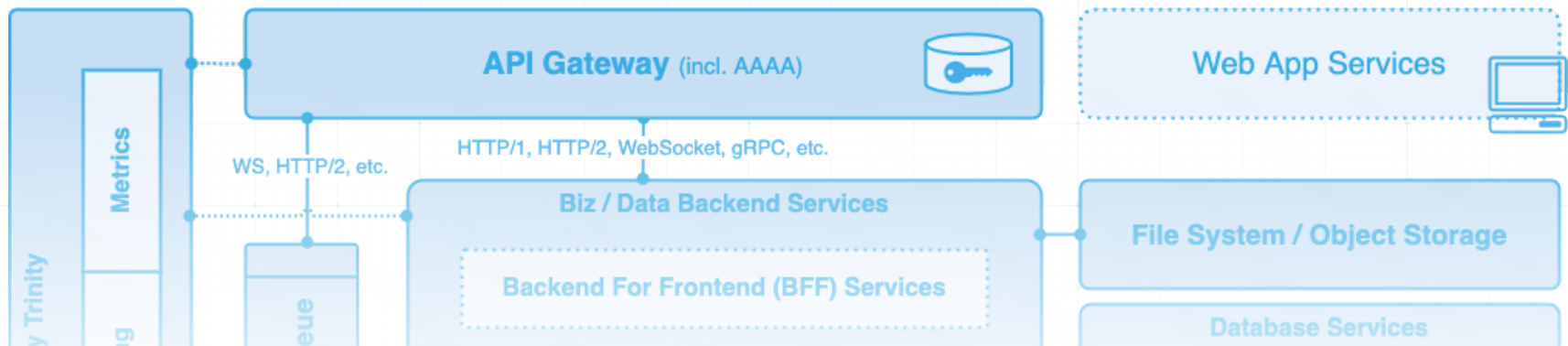
- 👉 关系型数据库 (RDB) : 适合存储结构性、逻辑关联性 (一对多/多对多) 较强的数据, 代表: Oracle, MySQL, PostgreSQL (适合存储地理信息), TiDB (云原生)
- 👉 键值型数据库 (KVDB) : 适合存储各类键值对, 键值的类型支持丰富, 常用作于缓存, 代表: Redis, TiKV
- 👉 时序型数据库 (TSDB) : 适合存储按时间顺序产生的数据, 如设备的状态采样数据及日志, 也常用作于事件日志存储 (Event Log), 代表: InfluxDB, OpenTSDB
- 👉 文档型数据库 (DocDB) : 适合存储弱关联、或有版本管理诉求的数据, 常用于配置、文档数据存储, 代表: MongoDB
- 👉 搜索引擎型数据库 (SEDB) : 适合有全文检索需求、对数据索引能力需求较大的场景, 代表: Elasticsearch

### 【文件存储】利用好云平台/云商能力

- 👉 文件存储: 利用好 Kubernetes 自身丰富的持久化存储能力
- 👉 对象存储: 可直接使用云商存储, 自建推荐使用 MinIO







## 怎么操控好业务服务接口 🤔

### 【策略】统一的入口管控

- 👍 便于统一控制和管理各业务模块（包括外部服务）的外部访问（南北向）
- 👍 便于统一实施安全相关策略、机制
- 👍 便于后续对接服务网格（东西向）
- 👎 容易形成性能瓶颈，需要强高可用

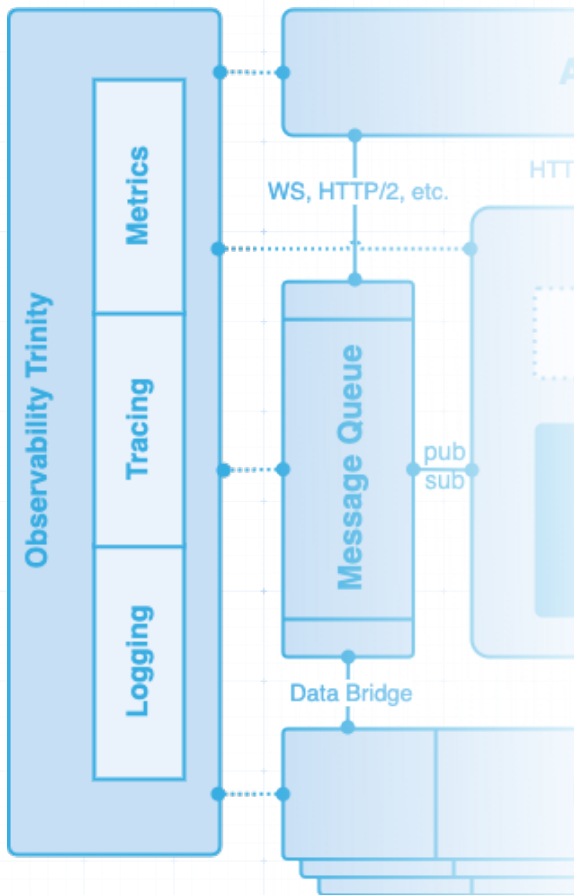
### 【方案】API Gateway

- 👍 统一的 API 路由前/中/后阶段管理
- 👍 便于应用 4A（鉴权/授权/记账/审计）
- 👍 便于统一桥接业务服务的各种接口协议
- 👍 便于实施南北向流量控制（限流/熔断等）
- 👍 配合 BFF 代理类服务食用更佳
- 👎 网关的数据需要单独存储、维护和管理

### 【推荐组件】Apache APISIX

Apache APISIX 是一个动态、实时、高性能的 API 网关，基于 Nginx 网络库和 etcd 实现，提供负载均衡、动态上游、灰度发布、服务熔断、身份认证、可观测性等丰富的流量管理功能。👉 [apisix.apache.org](https://apisix.apache.org)

- 👍 社区每月更新一个版本，需求响应快
- 👎 插件开发是基于 Lua / OpenResty 体系



## 怎么观测业务服务的使用 🤔

### 【日志】了解服务的历史行为

- 👉 目前比较成熟的体系是 ELK / EFK，即 Elasticsearch + Kibana + Logstash / Fluentd / Fluentbit，EFK 偏云原生
- 👉 通常是通过落盘文件或 stdout 标准输出来采集，也可以考虑入侵的方式：通过定制的日志库向采集器推送日志数据
- 👉 日志系统是可观察性三元组中最容易实现的，各企业/云商都有自己的方案，自建的维护成本较高，建议考虑云商日志服务

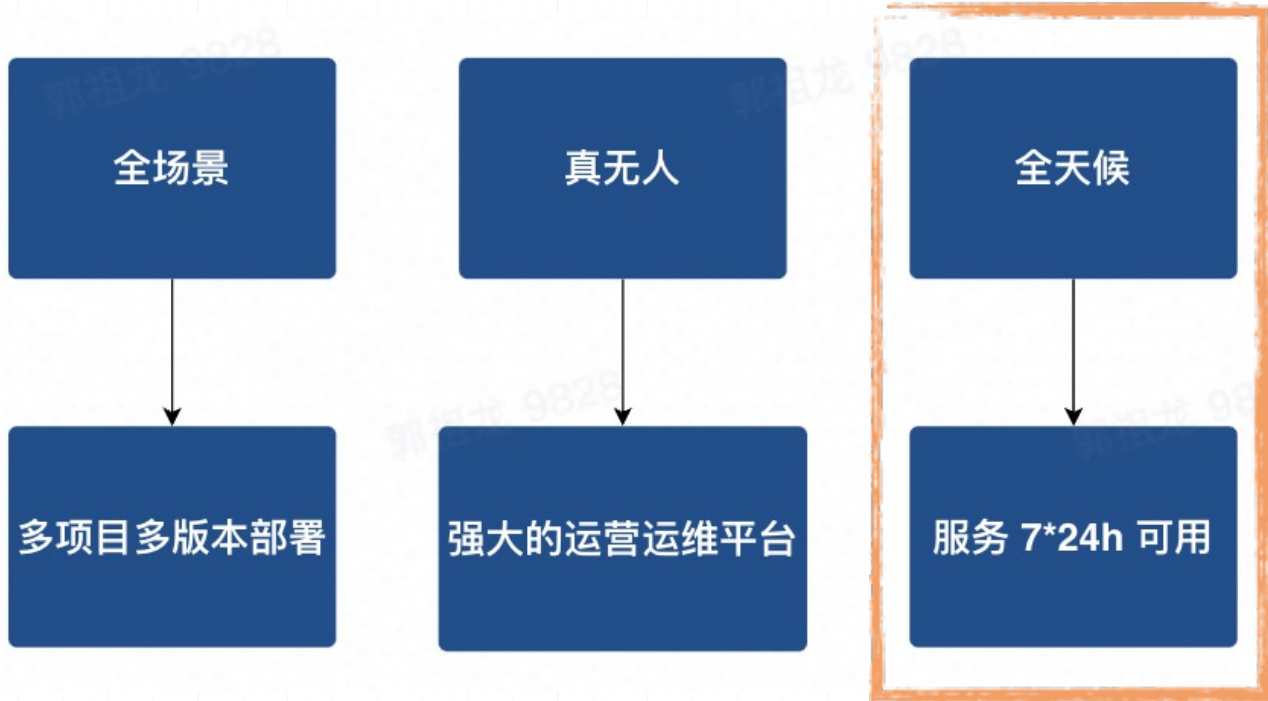
### 【追踪】侦测服务间的拓扑和链路

- 👉 比较老牌的链路追踪服务有 Jaeger, Zipkin
- 👉 目前社区比较热门的新星是 Apache SkyWalking，对各语言各中间件支持都相对比较丰富，最新 8.x 版本各方面持续增强
- 👉 通常可以直接考虑通过类库入侵服务来进行数据采集，也可以选择通过 Istio / Envoy 网格的非入侵方式接入

### 【指标】实时检测服务的运行状态

- 👉 业内基本一致使用 Prometheus + Grafana 的方案
- 👉 Prometheus 有很好的云原生基础，集群版 Thanos 持续演进中
- 👉 服务需要提供自己的 Exporter 指标数据接口供 Prometheus 拉取

# 云脑方向



# 保证服务 7\*24h 可用

## 高可用目标

高可用 \*3  
尝试对服务可用性在于小数点后面  
多加几个 9  
尝试减少故障恢复时间

## 热备与冷备

热备：负载均衡或分片，空间换  
时间  
冷备：具备不确定性，故障恢复  
时间无法估量

# 保证服务 7\*24h 可用

## 无状态服务高可用

开启多副本机制，平台能力

## 有状态服务高可用

套路多多，例如 Raft 选举机制等  
学习了解知名中间件实现方式  
例如 Apache Pulsar 等

# 高可用方案改造

如果有服务当前是单节点的方式，如何实现高可用方案？

1. 有状态服务改造为无状态服务
2. 有状态服务支持高可用，例如通过 ETCD 选举机制
3. 基于 k8s 实现工作节点切换的托底方案，自主实现 k8s operator 监控及切换

1. 自动驾驶与云脑

2. 云脑三件事儿

3. 总结与展望

4. 致谢

## 总结

1. 支持多项目多版本部署
2. 提供强大的运营运维平台
3. 保证服务 7\*24h 可用
4. 充分拥抱云原生

## 展望

1. SaaS 降本增效，服务网格落地
2. 探索私有云和公有云使用不同的解决方案
3. 服务平台化，以实现和其他厂商对接
4. 车端引入边缘云技术方案



1. 自动驾驶与云脑

2. 云脑三件事儿

3. 总结与展望

4. 致谢

# 致谢

1. 感谢 UCloud 的邀请
2. 感谢云脑团队的小伙伴给予的帮助支持
3. 感谢我太太在百忙之中对我的细心照顾

# THANKS!

驭势科技 · 郭祖龙

[zulong.guo@uisee.com](mailto:zulong.guo@uisee.com)