



APACHE  
APISIX



# 基于APISIX的全链路灰度方案设计与实践

泮圣伟

阿里云 MSE 微服务引擎研发



APACHE  
APISIX



## CONTENT

- 01 全链路灰度方案简介
- 02 全链路灰度方案产品实践
- 03 客户案例
- 04 总结与展望



APACHE  
APISIX



01

## 全链路灰度方案简介

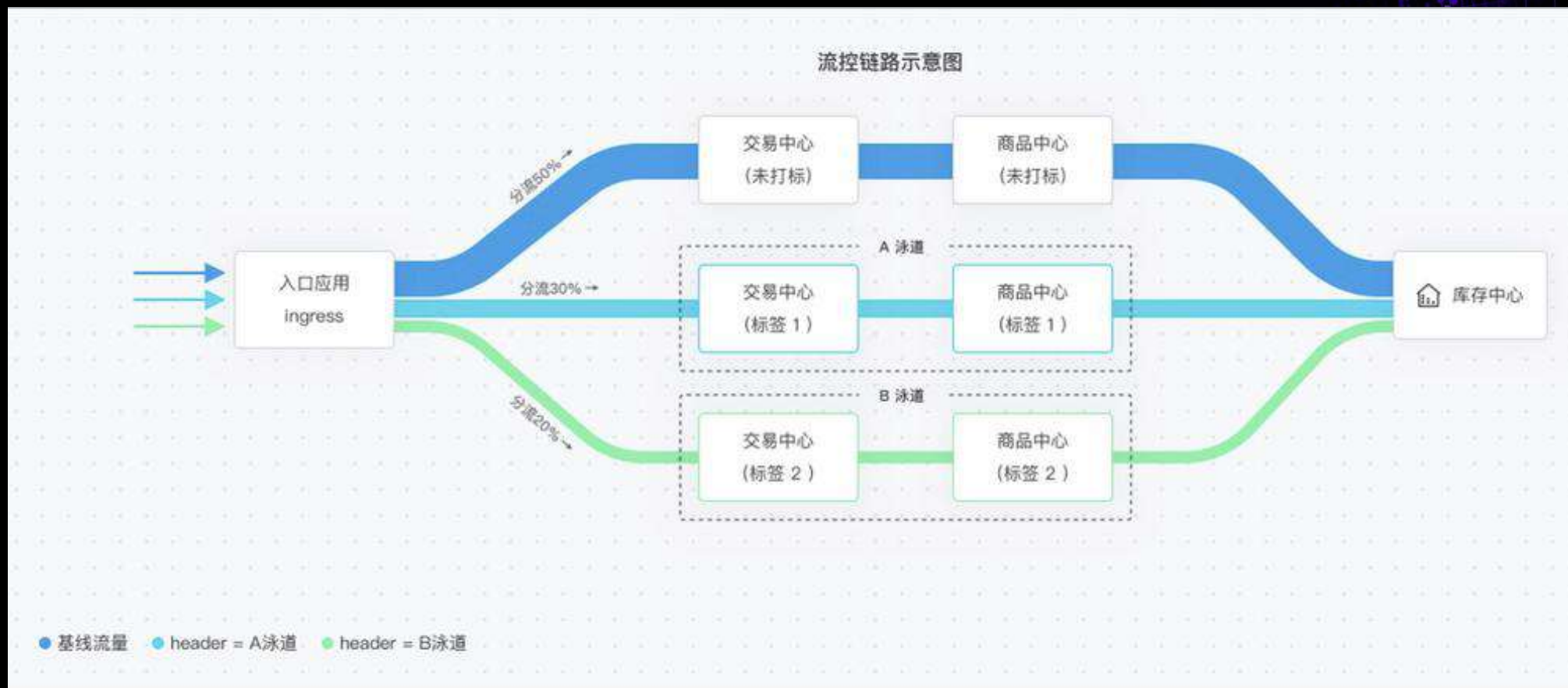
全链路灰度方案涉及到的概念与场景。  
深入剖析全链路灰度方案技术原理。



APACHE  
APISIX

## 概念

- 泳道
- 基线环境
- 流量回退
- 泳道组





APACHE  
APISIX



## 应用场景

### 日常/开发/项目/测试环境隔离

构造日常、开发、项目、测试等多套环境的“泳道”，每个项目环境都有唯一的一个项目标签，流量带上这个项目标签后会路由到该项目环境，否则会去主干环境。如果没有这套机制，开发环境要进行物理隔离，这就需要部署整套微服务架构，成本非常高。

### 全链路灰度发布

线上所有应用部署灰度版本，灰度流量全部进入灰度版本，正常流量进入生产版本。灰度版本只针对灰度流量验证，有效减少风险。云上客户安x就有这样的场景，要灰度发布 N 个应用，想灰度流量在这 N 个应用的灰度版本之间路由。

### 高可用同机房优先路由

业务高可用部署后，服务调用如果跨机房，会带来额外的调用延迟。开启同机房优先路由后，让 Consumer 服务调用优先选择相同机房的 Provider，降低 rt。

### 全链路压测

直接使用线上环境压测，让压测流量的 DB 操作落库到影子表，Redis 落到影子 KEY，MQ 落到影子 TOPIC。如果没有路由能力，需要搭建一套仿真的线上环境用于压测，成本直线上升。用线上环境配合流量控制能力可以实现 0 机器成本，0 维护成本完成全链路压测。



APACHE  
APISIX

## 涉及技术领域

- RPC: 微服务之间的路由
- MQ
- Database
- Redis
- 分布式任务调度
- 前端
- 可观测性





APACHE  
APISIX

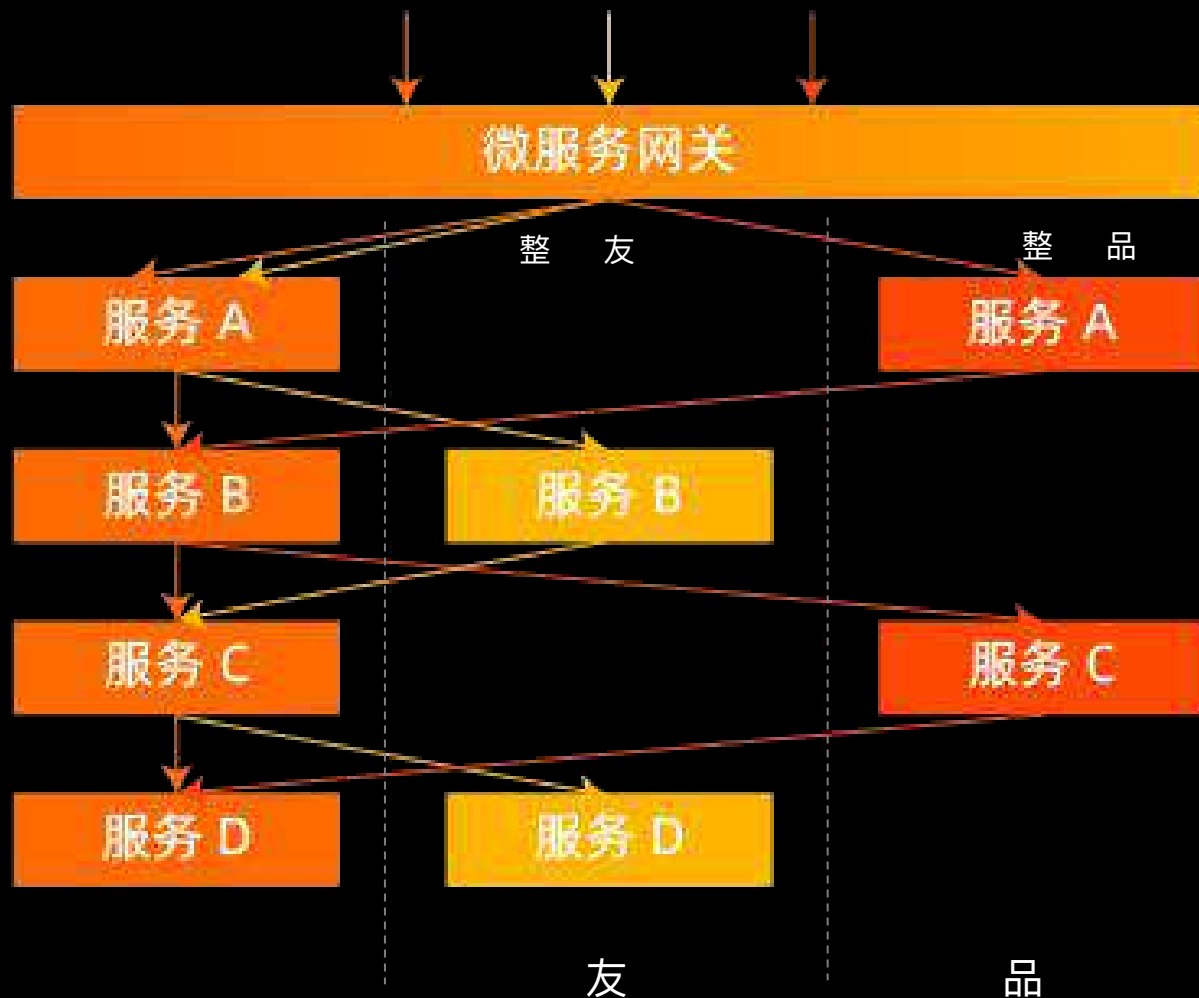
## 方案一：物理隔离





APACHE  
APISIX

## 方案二：逻辑隔离







POWERED BY  
APACHE  
APISIX

## 技术点一：标签路由

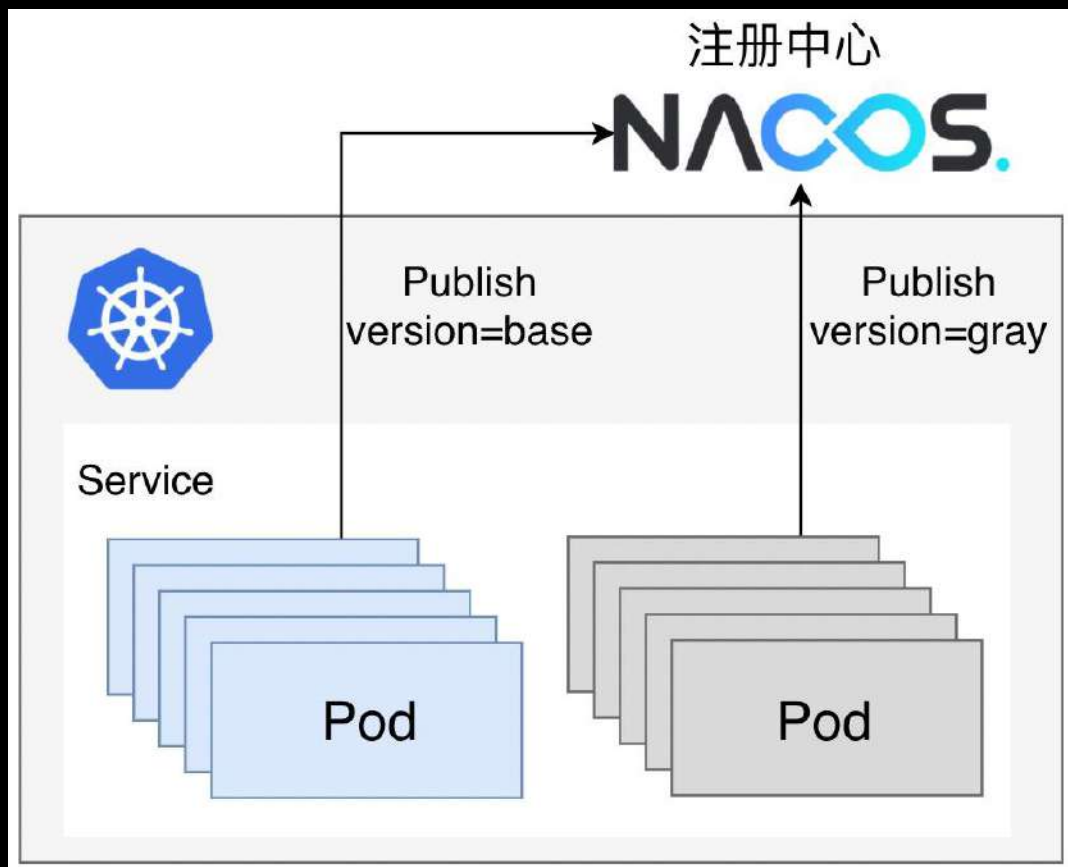




APACHE  
APISIX

## 技术点二：节点打标

以容器化的业务应用举例，服务发现使用Nacos注册中心



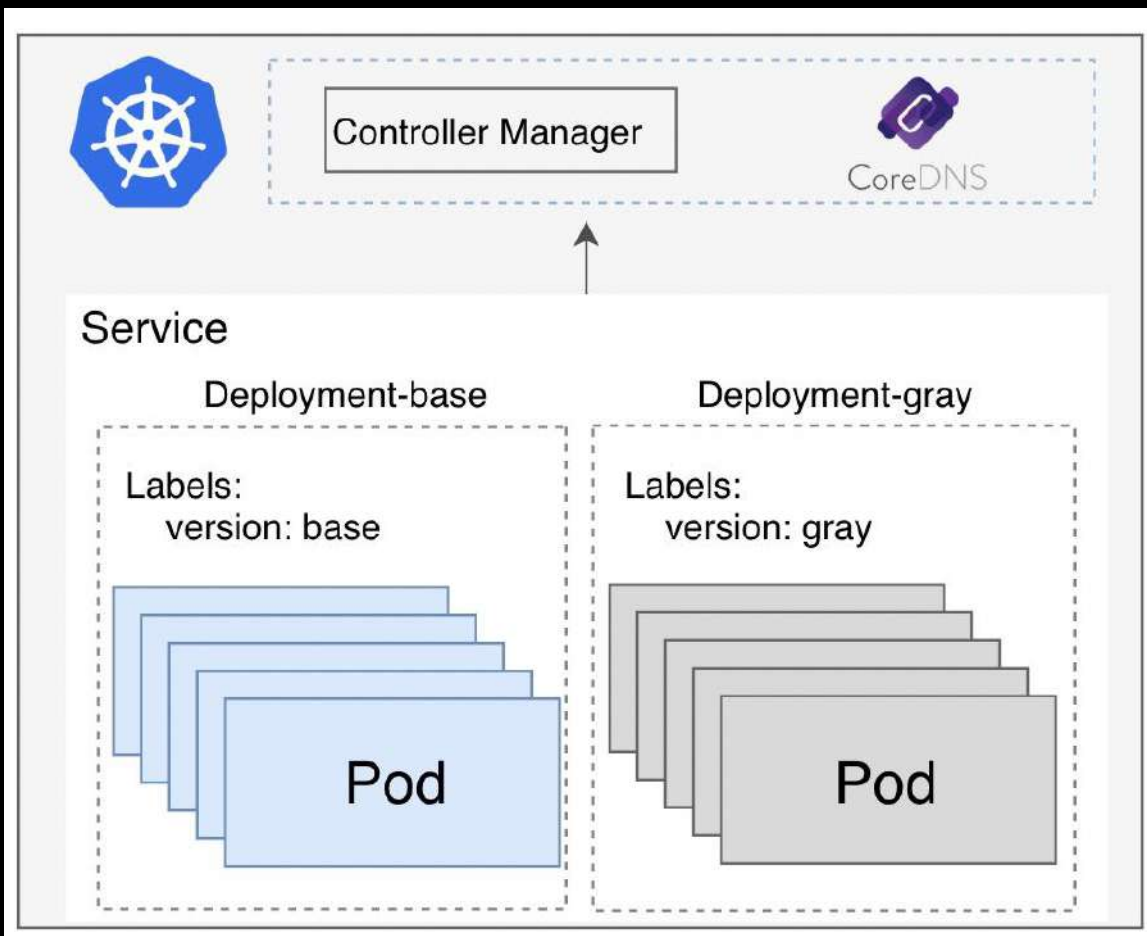
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: httpbin-v1
spec:
  replicas: 3
  selector:
    matchLabels:
      app: httpbin
  template:
    metadata:
      labels:
        app: httpbin
    spec:
      containers:
        - image: xxxx
          name: spring-cloud-httpbin-nacos
          ports:
            - containerPort: 8080
          env:
            - name: spring.cloud.nacos.discovery.metadata.version
              value: gray
```



APACHE  
APISIX

## 技术点二：节点打标

以容器化的业务应用举例，服务发现使用K8s Service



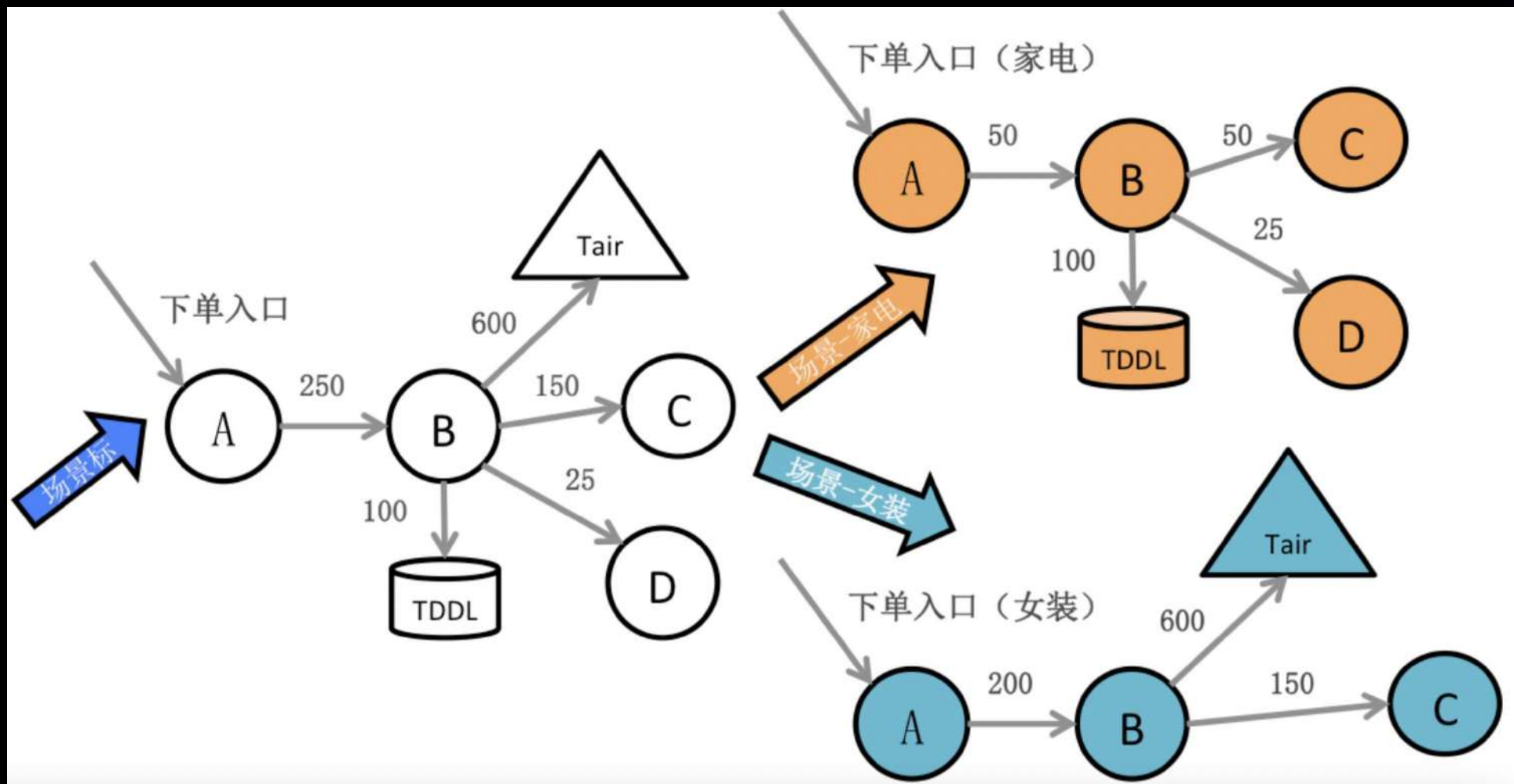
```
apiVersion: v1
kind: Service
metadata:
  name: httpbin
spec:
  ports:
  - port: 8080
    protocol: TCP
  selector:
    app: httpbin
```

```
template:
  metadata:
    labels:
      app: httpbin
      version: v1
  spec:
    containers:
    - image: xxxxx
      imagePullPolicy: Always
      ports:
      - containerPort: 8080
```



APACHE  
APISIX

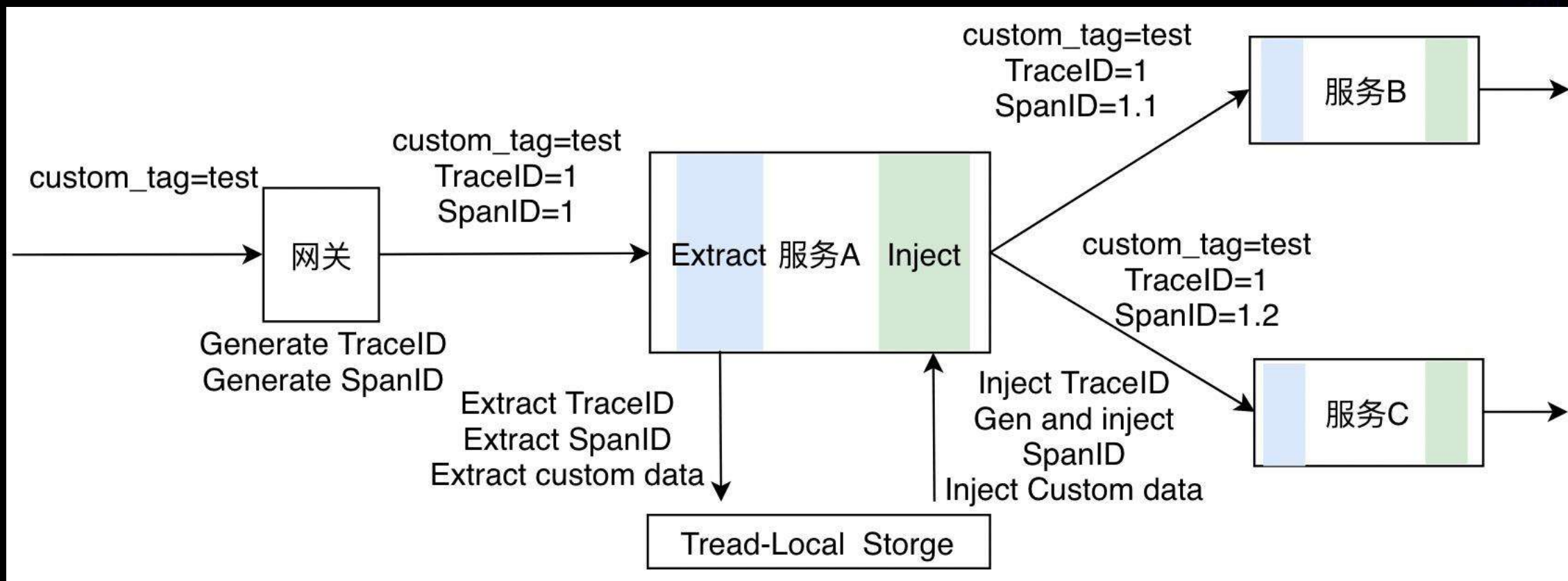
### 技术点三：流量染色





APACHE  
APISIX

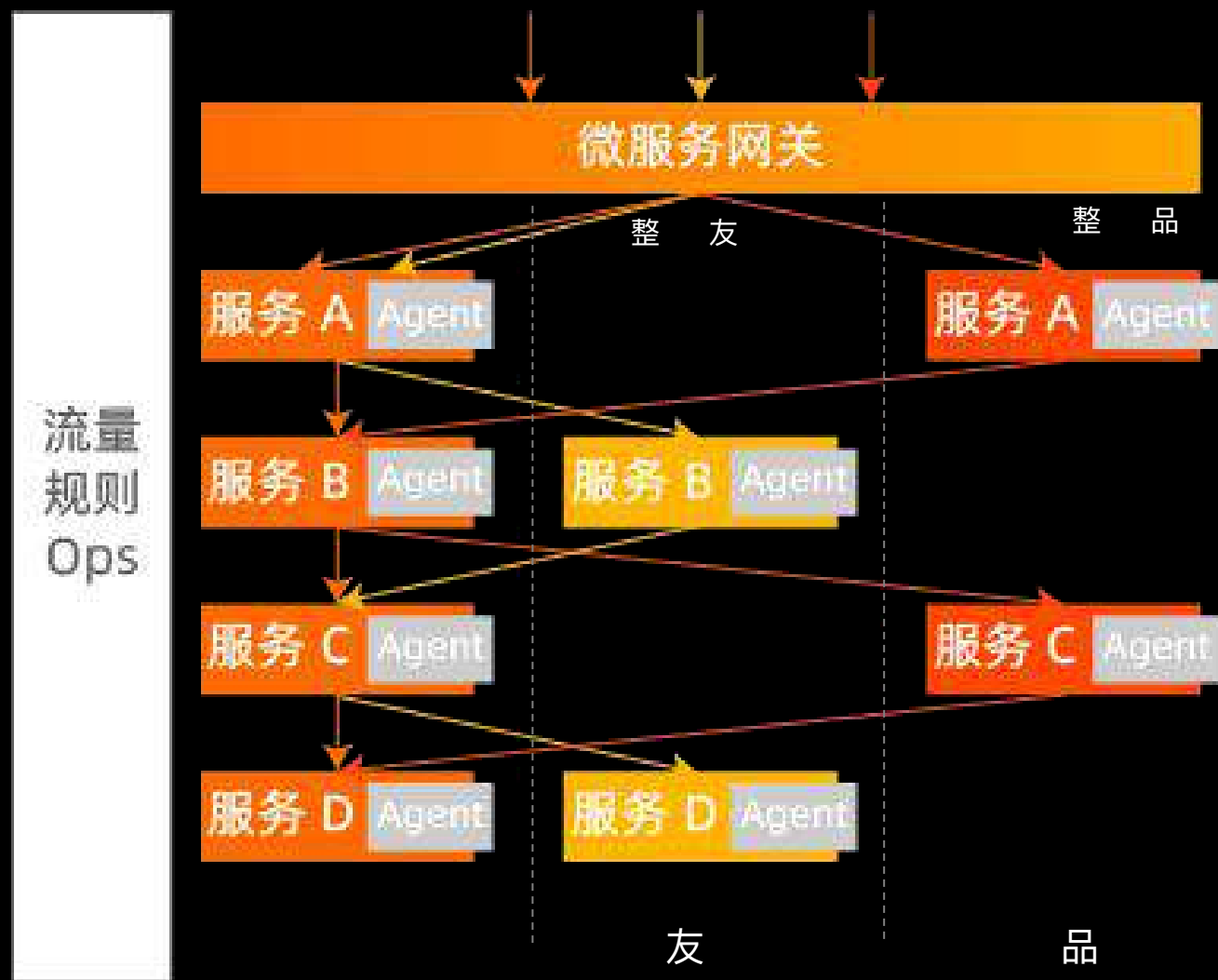
## 技术点三：分布式链路追踪





APACHE  
APISIX

## MSE 全链路灰度的技术实现





APACHE  
APISIX



# 02

## 全链路灰度方案产品实践

MSE 微服务治理全链路灰度方案介绍

基于 Apache APISIX 全链路灰度方案实践



APACHE  
APISIX

## MSE 微服务治理全链路灰度

- 全链路隔离流量泳道
- 端到端的稳定基线环境
- 流量一键动态切流
- 可观测能力
- 低成本接入，基于Java Agent 技术实现无需修改一行业务代码
- 具备无损上下线能力，使得发布更加丝滑





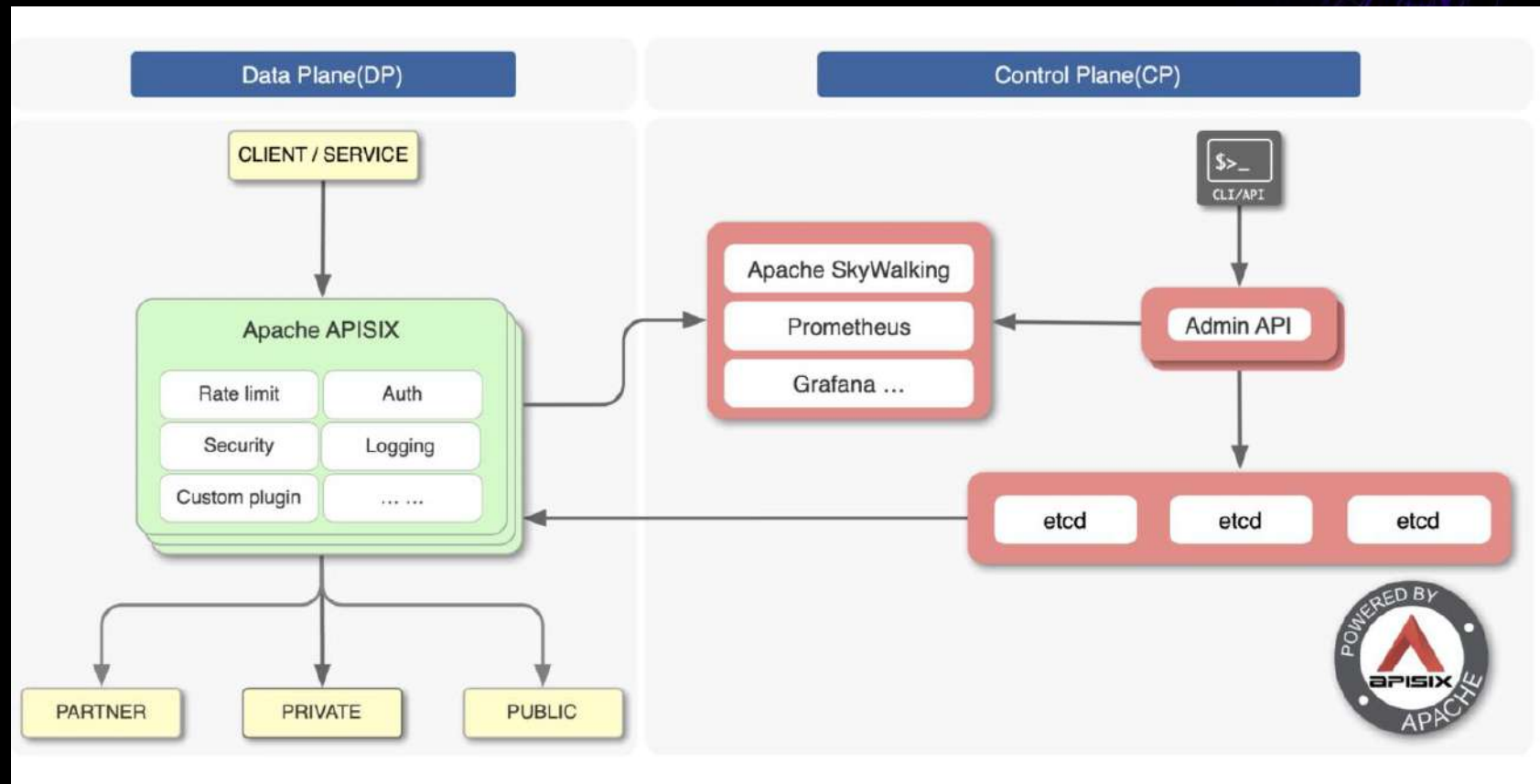


APACHE  
APISIX

# Apache APISIX

Apache APISIX 是一个云原生、高性能、可扩展的微服务API 开源网关。

- 
- Dashboard
- 

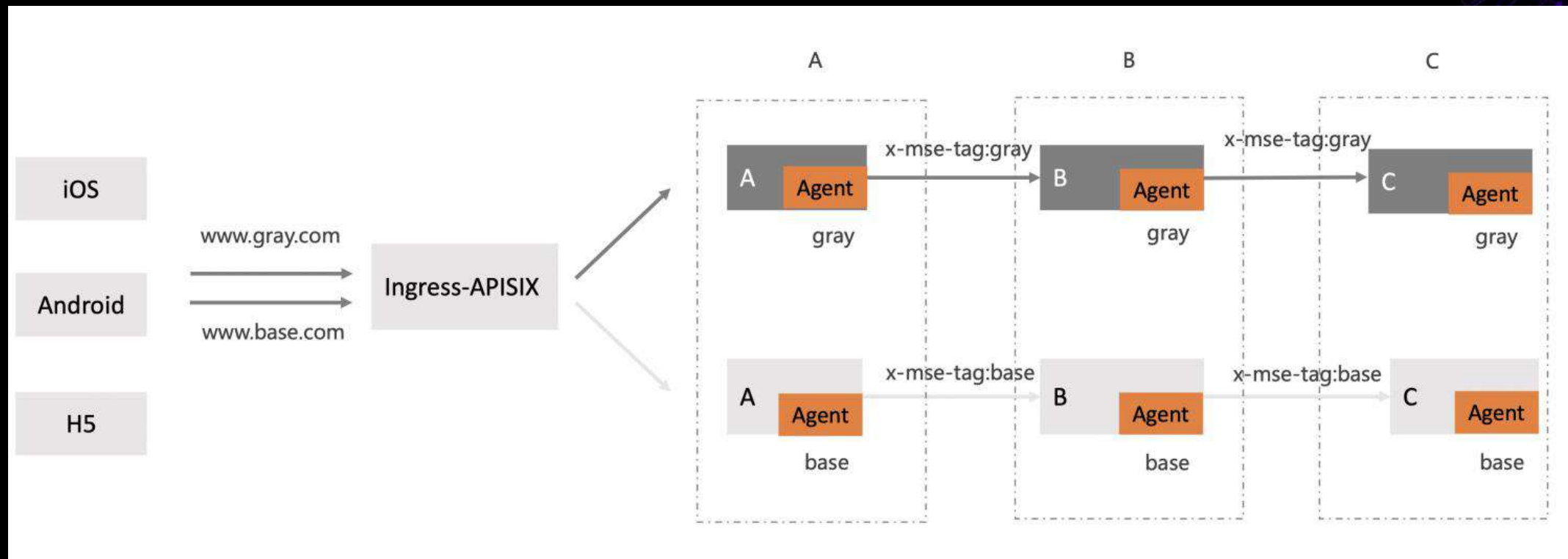




APACHE  
APISIX

# Apache APISIX

场景一：按照域名路由，实现全链路灰度



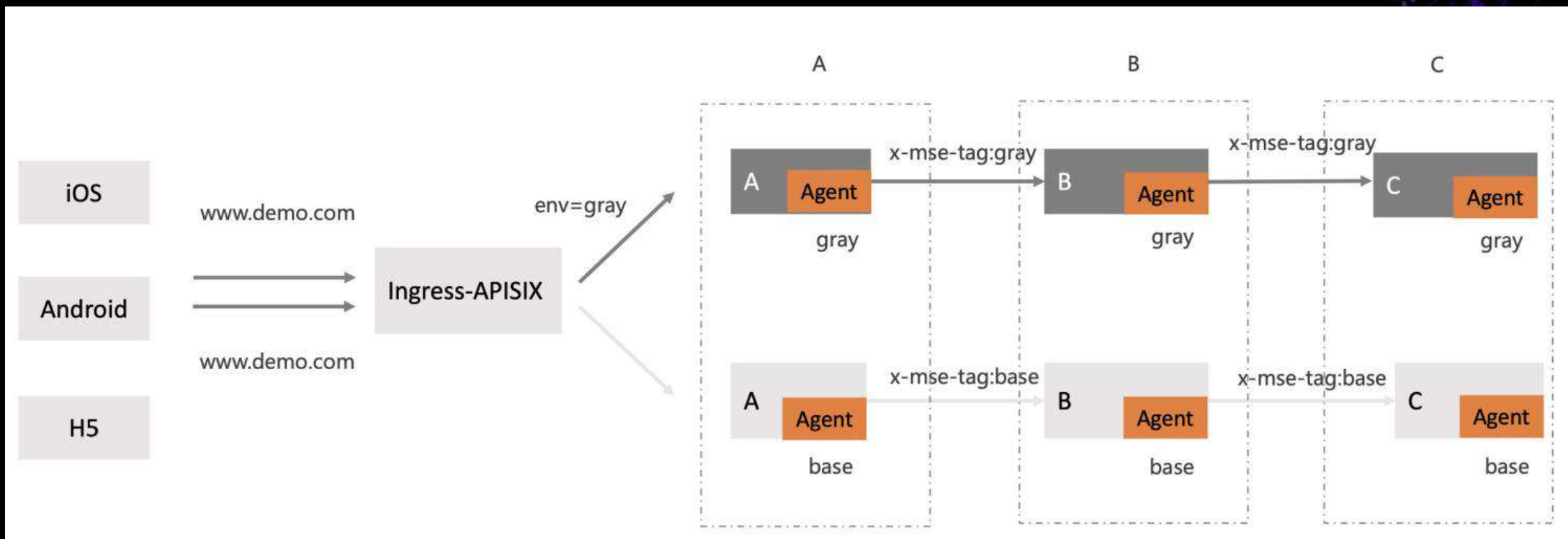
调用链路 Ingress-APISIX -> A -> B -> C，其中 A 可以是一个 spring-boot 的应用。



APACHE  
APISIX

# Apache APISIX

场景二：按照指定请求参数进行路由，实现全链路灰度



调用链路 Ingress-APISIX -> A -> B -> C，其中 A 可以是一个 spring-boot 的应用。



APACHE  
APISIX



# 03

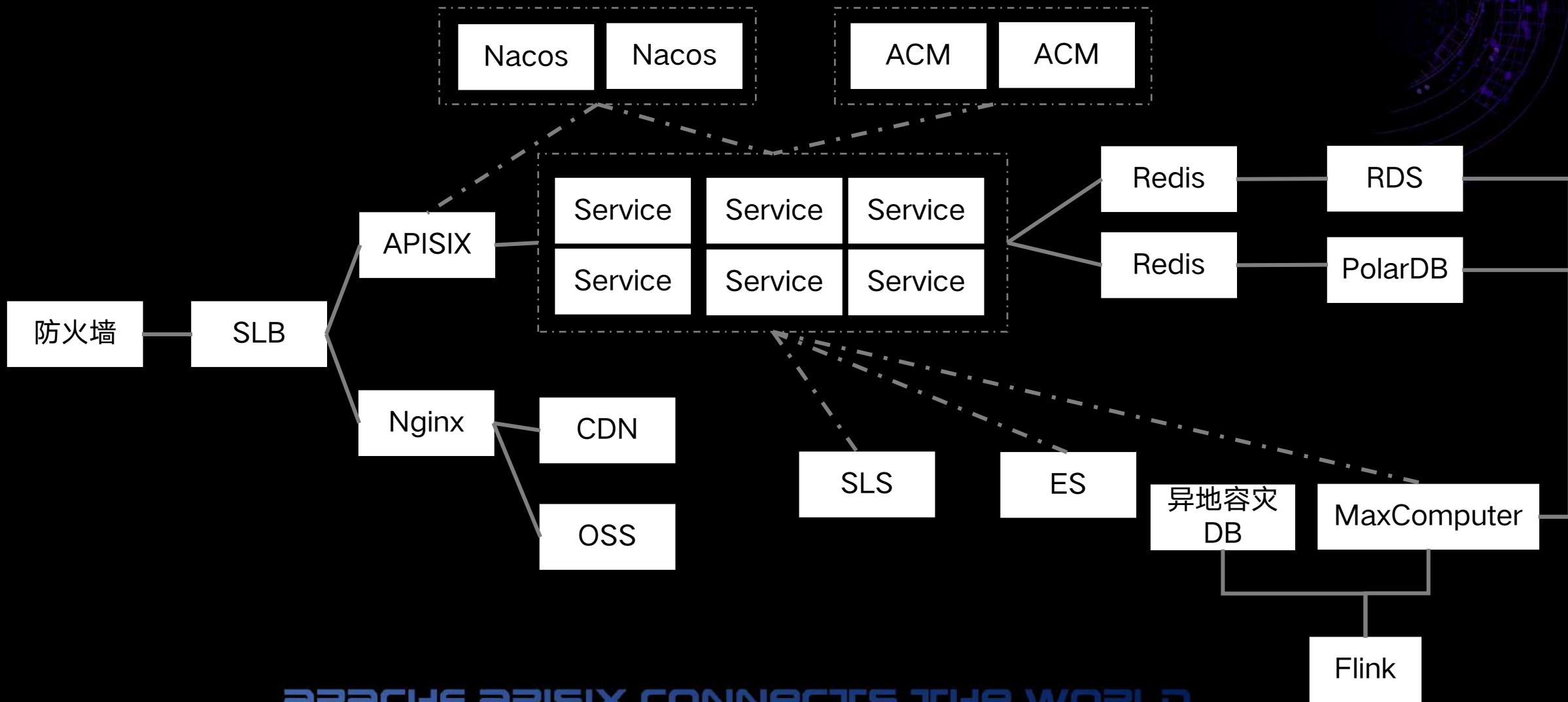
## 客户案例

云上某客户全链路灰度方案实践.



APACHE  
APISIX

云上



APACHE APISIX CONNECTS THE WORLD



APACHE  
APISIX



# 04 总结与展望



APACHE  
APISIX

## 总结

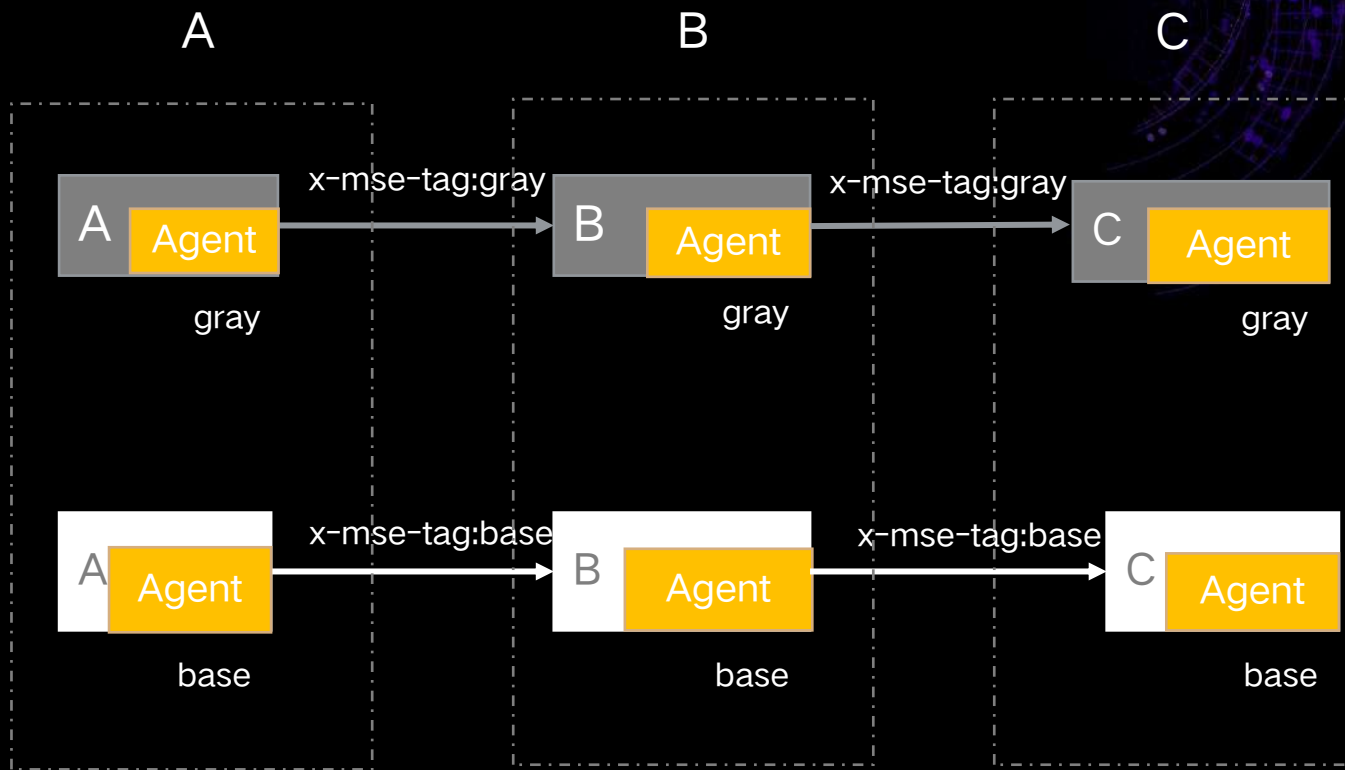
iOS

Android

H5

APISIX

- Header
- Cookie
- Params
- 域名





APACHE  
APISIX



- 前端
- 网关
- RPC
- MQ
- Database
- Redis
- 分布式任务调度
- 基于 Ingress-nginx 网关实现全链路灰度
- 基于Apache APISIX网关实现全链路灰度
- 基于 MSE 云原生网关实现全链路灰度
- 基于 Java 网关实现全链路灰度
- 基于 RocketMQ 实现全链路灰度
- 通过Jenkins构建CI/CD实现金丝雀发布
- 微服务敏捷开发最佳实践
- 完整的产品控制台
- 秒级可观测能力
- 流量大盘

APACHE APISIX CONNECTS THE WORLD



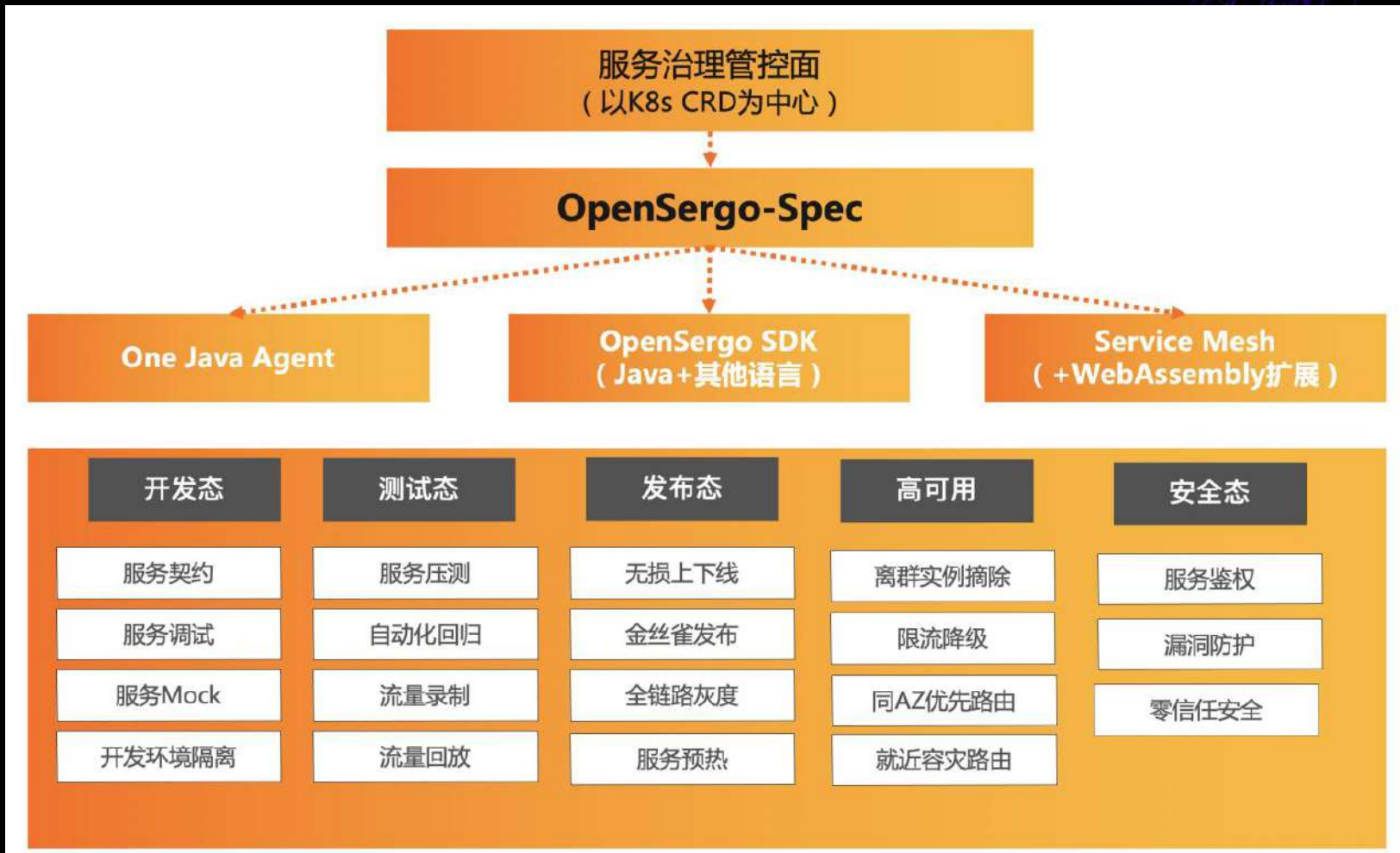


APACHE APISIX

# 展望

服务治理是微服务改造深入到一定阶段之后的必经之路

- 除了全链路灰度，服务治理还有没有其他能力？
- 多语言场景下，有无全链路的最佳实践或者标准？
- 异构微服务如何可以统一治理？





APACHE  
APISIX



感谢聆听  
THANKS

APACHE APISIX CONNECTS THE WORLD