msup

基于 Apache APISIX 的服务网格方案

讲师简介

张超
API7 技术专家

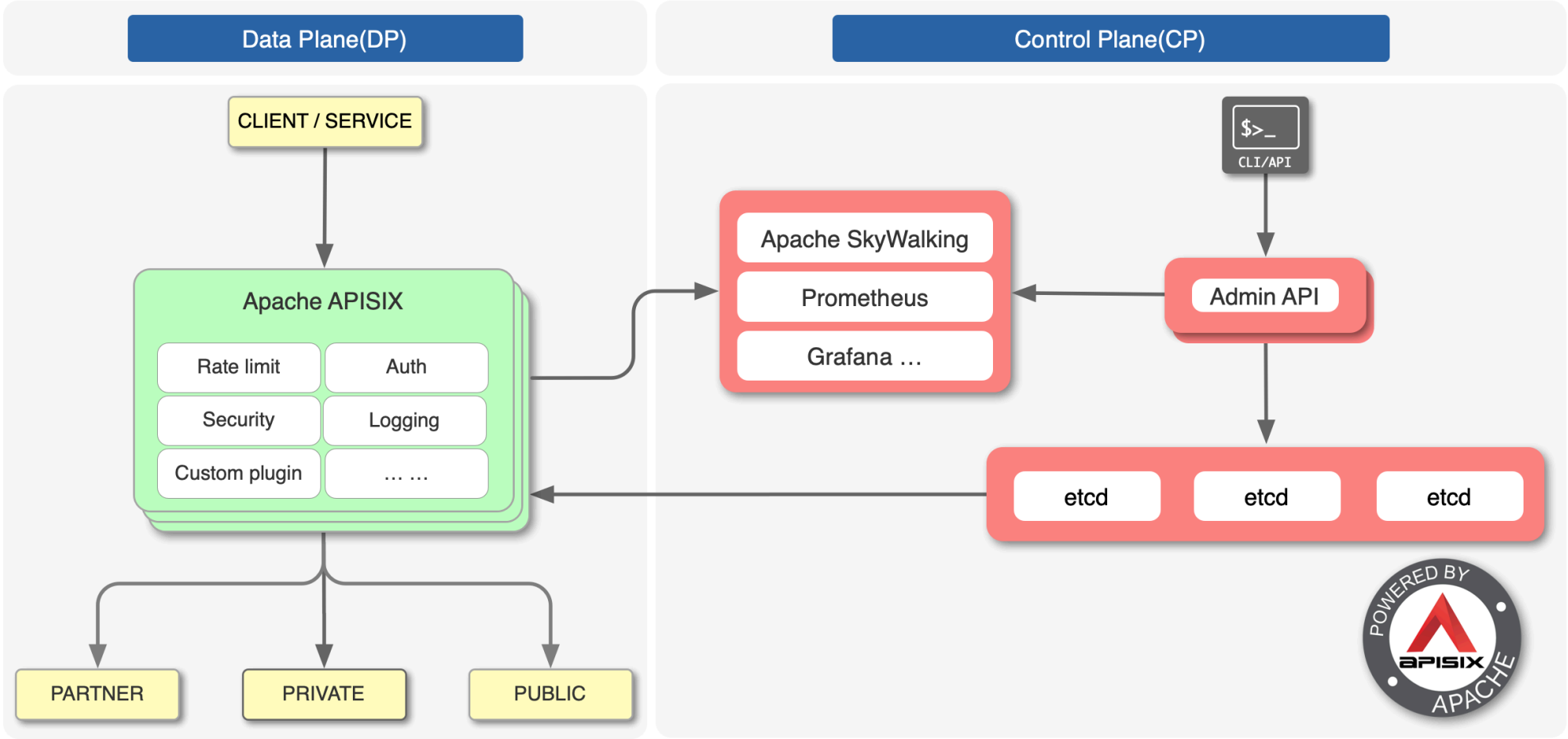
- Apache APISIX PMC 成员
- Tars 基金会大使
- 开源爱好者
- 聚焦于服务网格和 API 网关
- https://github.com/tokers

目录

什么是 Apache APISIX

- 高性能，全动态的云原生 API 网关
- 提供了负载均衡、服务发现、限流限速等在内的诸多功能
- 易于扩展
- 社区充满活力且健康
- https://apisix.apache.org

什么是 Apache APISIX



APISIX Mesh 方案的演进

msup

💡 Apache APISIX 是否可以用于服务网格中?

APISIX Mesh 方案的演进

msup

功能一览

- HTTP(s) & gRPC 代理
- TCP & UDP 代理
- Traffic Split (金丝雀发布 & 蓝绿部署)
- 负载均衡（WRR, Consistent Hash, EWMA）
- 主/被动健康检查
- 认证（mTLS、JWT Token 等）
- 可观测性

APISIX Mesh 方案的演进

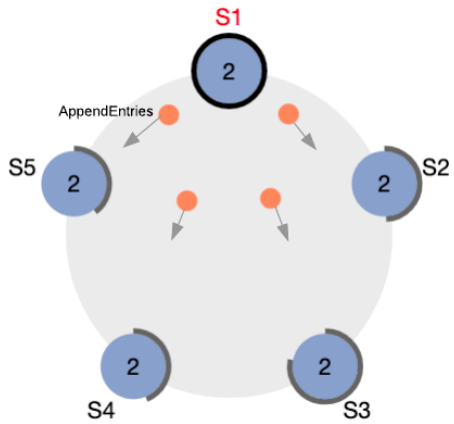
需要克服的问题

- 如何保证配置生效的低时延?
- 控制面选型
- OpenResty & Lua 社区和生态薄弱

APISIX Mesh 方案的演进

保证配置生效的低时延

ETCD 基于 Raft 协议实现分布式共识，无法承载太多的连接数，而在服务网格场景里，通常实例数都不少。因此可能 ETCD 会成为方案的第一个瓶颈。



APISIX Mesh 方案的演进

控制面选型

- 使用开源解决方案，如 Istio、Kuma、Consul Connect
- “再造轮子”

APISIX Mesh 方案的演进

OpenResty & Lua 社区和生态薄弱

- 生态弱小，工具不全
- 无法从社区得到强有力的支持

APISIX Mesh 方案的演进

“All problems in computer science can be solved by another level of indirection”.

Apache APISIX 自身并不足以成为服务网格的数据面，因此引入了 apisix-mesh-agent。

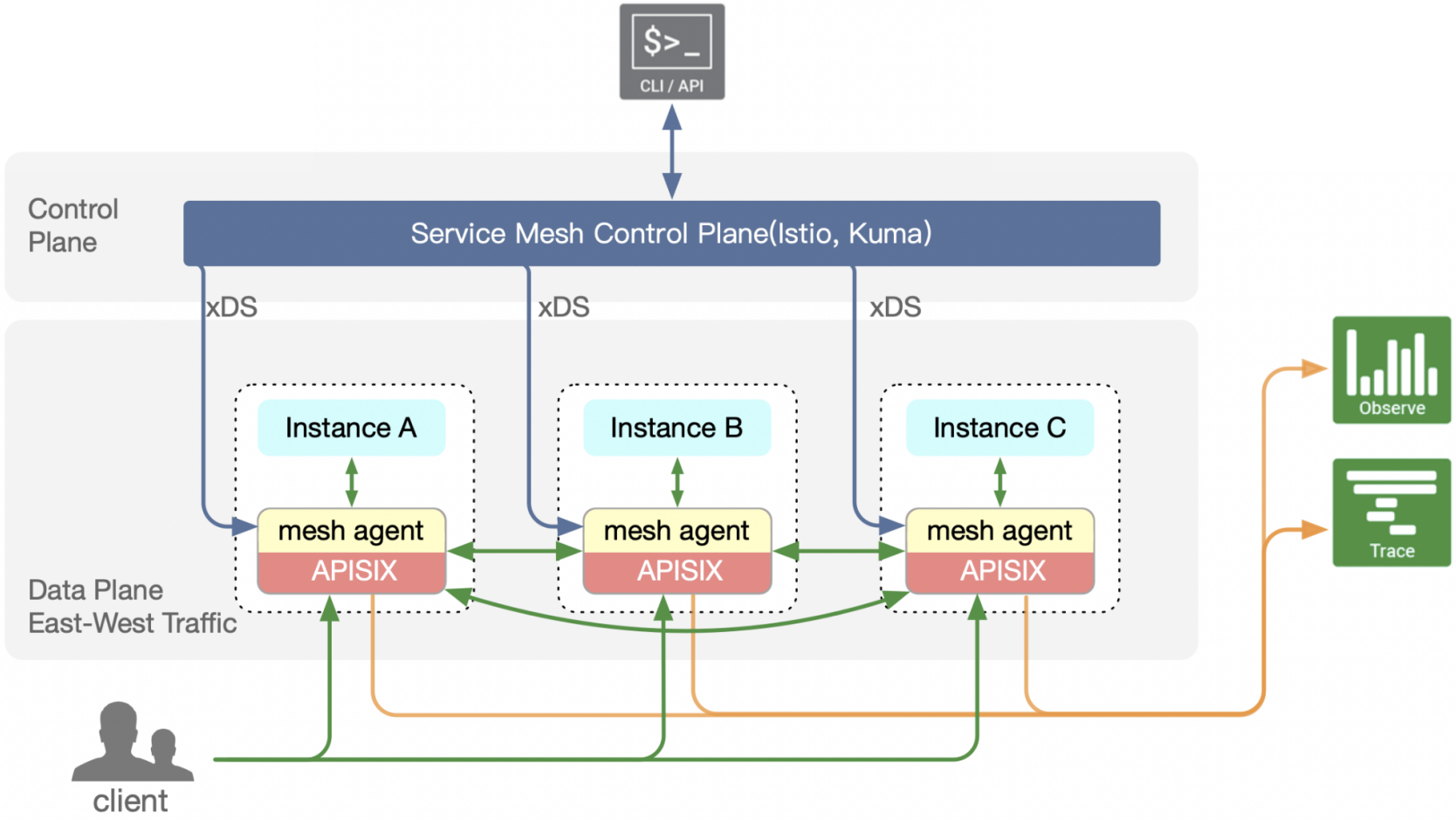
APISIX Mesh 方案的演进

apisix-mesh-agent

- 实现 Envoy xDS 协议，和控制面交互，获取配置变更并转换为 Apache APISIX 可识别的配置
- 模拟 ETCD V3 APIs
- 拦截应用实例的出入流量

APISIX Mesh 方案的演进

架构



案例简介

- APISEVEN Cloud 是一款在多云环境下连接用户 API 和微服务的云产品
- 通过 引入 APISIX Mesh 实现了 mTLS 和金丝雀发布

案例背景

APISEVEN Cloud 基于公有云搭建，运行在 Kubernetes 集群中，同时使用了部分中间件的云产品。

考虑到安全因素，服务间需要启用 mTLS。考虑到发布的风险，需要引入金丝雀发布。

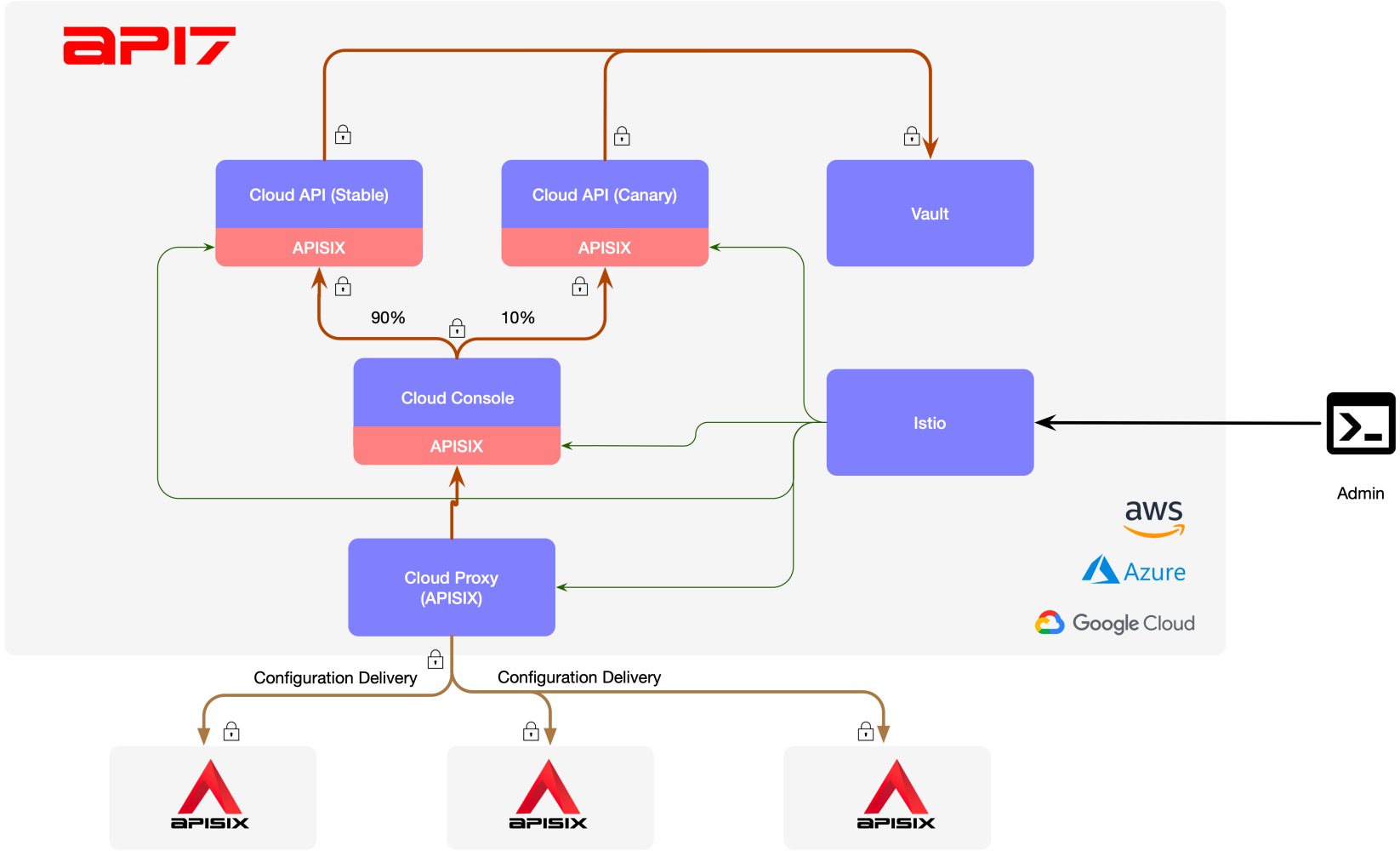
同时为了加快落地节奏，避免业务代码过于复杂，引入 APISIX Mesh 来完成服务通信时的双向认证和证书卸载。

此外，通过 APISIX Mesh 来实现金丝雀发布，降低发布风险

案例实践

- 通过使用 Istio 作为 APISIX Mesh 控制面，同时引入 Cert Manager 进行证书管理
- 数据面 APISIX Sidecar 帮助应用进行双向认证和证书卸载，降低了业务代码的复杂度
- 通过 Istio 作为 APISIX Mesh 控制面，实现了金丝雀发布

案例实践


API7
Cloud API (Stable)
APISIX
Cloud API (Canary)
APISIX
Vault
90%
10%
Cloud Console
APISIX
Istio
Admin
aws
Azure
Google Cloud
Cloud Proxy
(APISIX)
Configuration Delivery
Configuration Delivery
APISIX
APISIX
APISIX

案例总结

引入 APISIX Mesh 解决了一部分问题，但也引入了其他的开销，比如：

- Istio 的维护成本
- 更多的资源开销（Sidecar container）

需要权衡一个方案的利和弊，才能更好地进行抉择。

使用 APISIX 作为 Sidecar 的优势

- 高性能（单路由性能是 Envoy 的 120%）
- 插件丰富，可被复用
- 低扩展成本（Lua 语言简单，支持其他主流语言进行插件开发）
- 统一网关和服务网格数据面基础设施，降低运维成本

APISIX Mesh 的未来

- 自研的控制面 - 更好的适配性
- 可观测性能力的加强
- 同时接管南北向流量
- 更趋近标准（Service Mesh Interface 和 Gateway APIs）

msup®

Thanks!