

Nacos与Istio通过MCPC协议 对接实战

01010101

Nacos Maintainer 朱鹏飞@nkorange

目录

01

Nacos及Istio简介

02

MCP协议介绍

03

MCP协议对接实战

Nacos是什么?

一个更易于构建云原生应用的动态服务发现、配置管理和服务平台



动态配置服务



服务发现与管理



动态DNS服务



Nacos注册中心简介

Nacos注册中心是Nacos中负责服务注册、服务发现、健康检查等功能的组件

- 服务注册与发现

- 支持使用Java、Go、NodeJs等客户端进行服务的注册与发现
- 支持Dubbo服务框架、Spring Cloud Alibaba服务框架
- 支持使用DNS协议进行服务的发现

- 健康检查

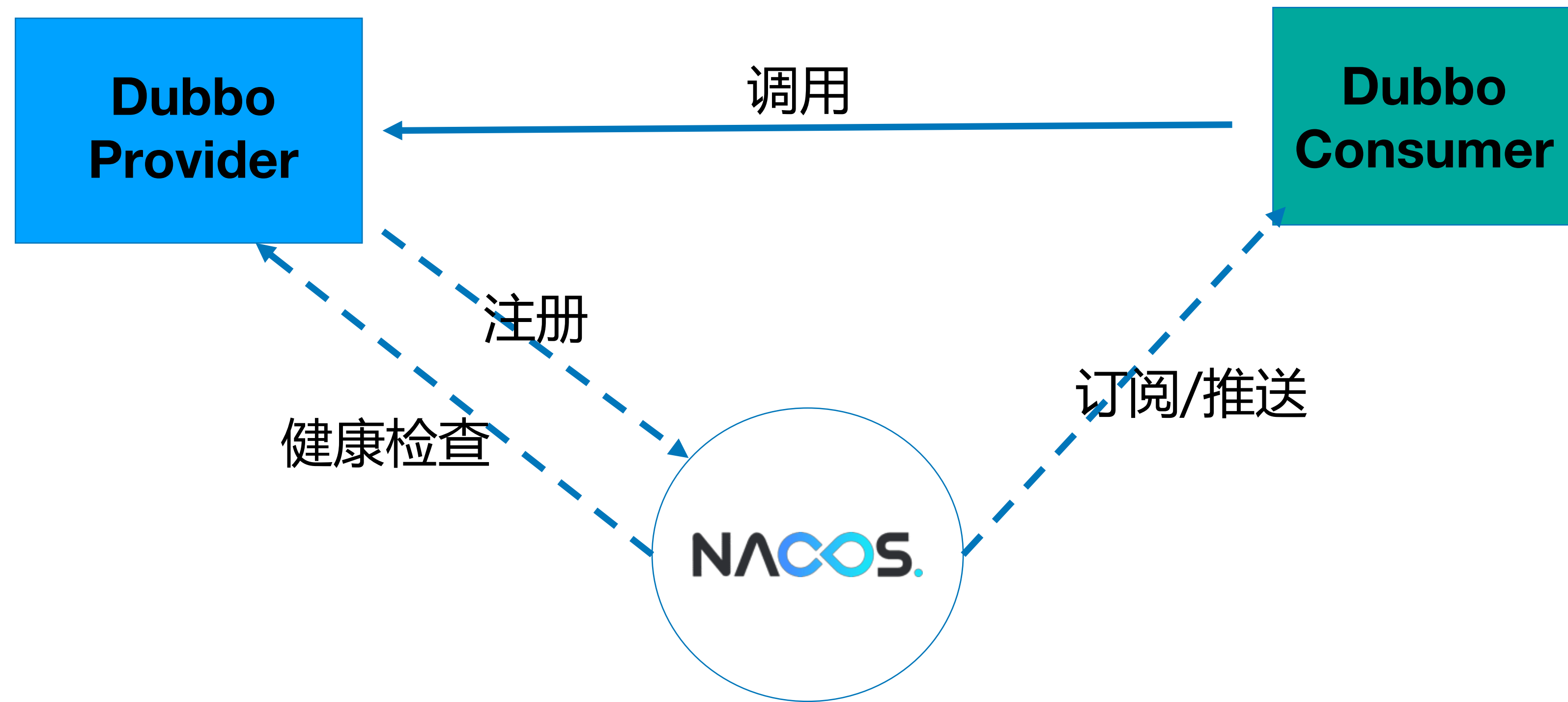
- 支持服务端发送TCP、HTTP、MySQL请求进行服务健康状态的探测
- 支持客户端心跳上报的健康状态更新

- 访问策略

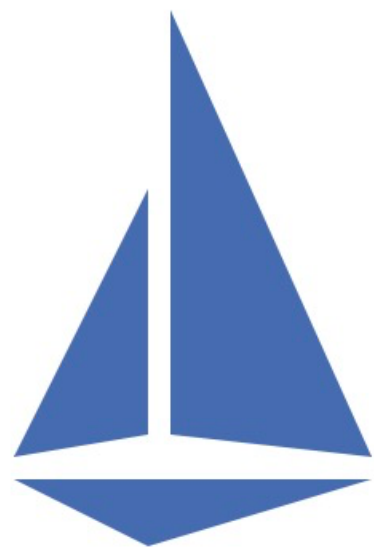
- 支持基于权重的负载均衡策略
- 支持保护阈值，解决服务雪崩问题
- 支持服务上下线
- 支持基于机房等环境信息设置访问策略



传统微服务模式下的服务发现



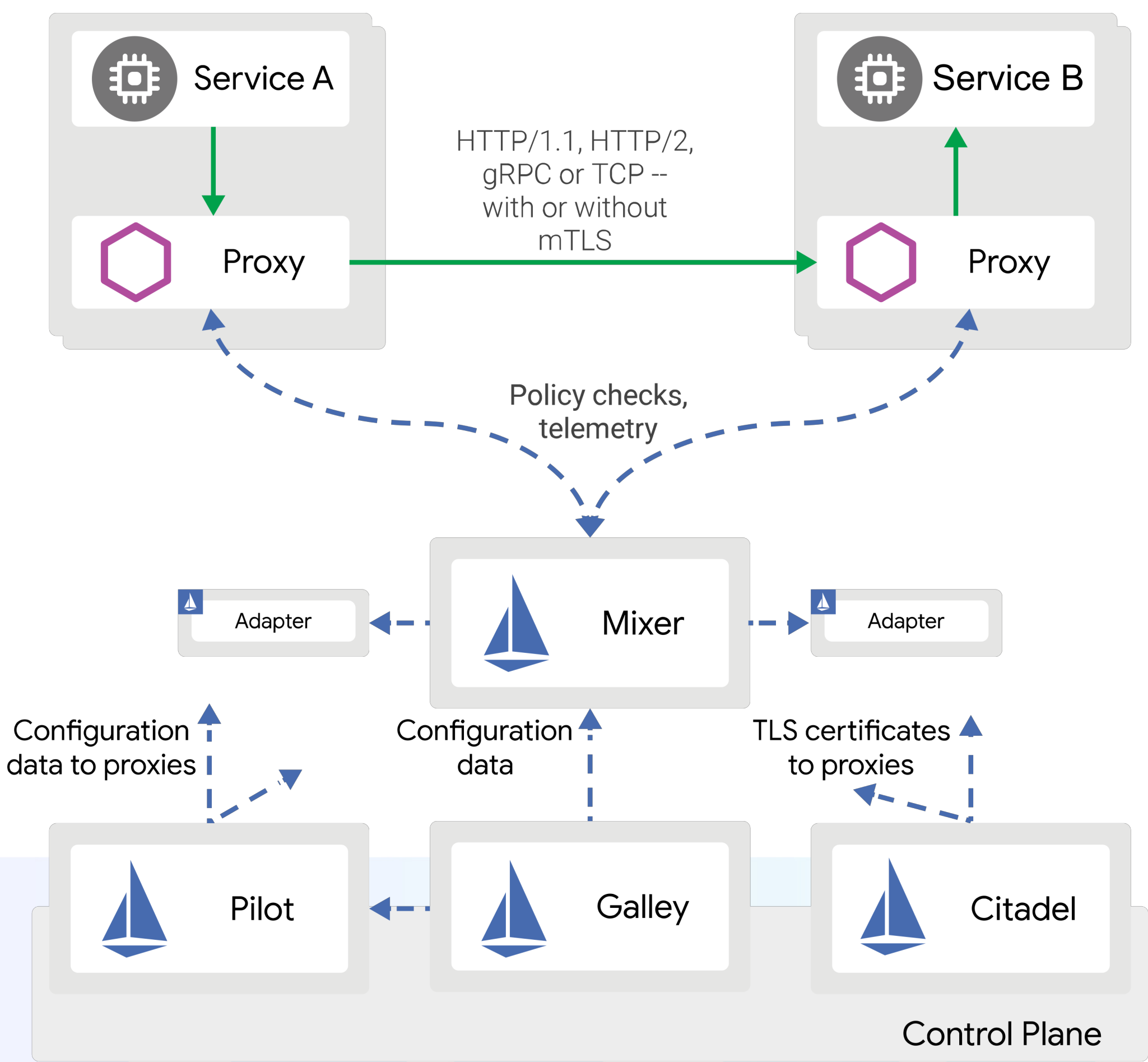
Istio介绍



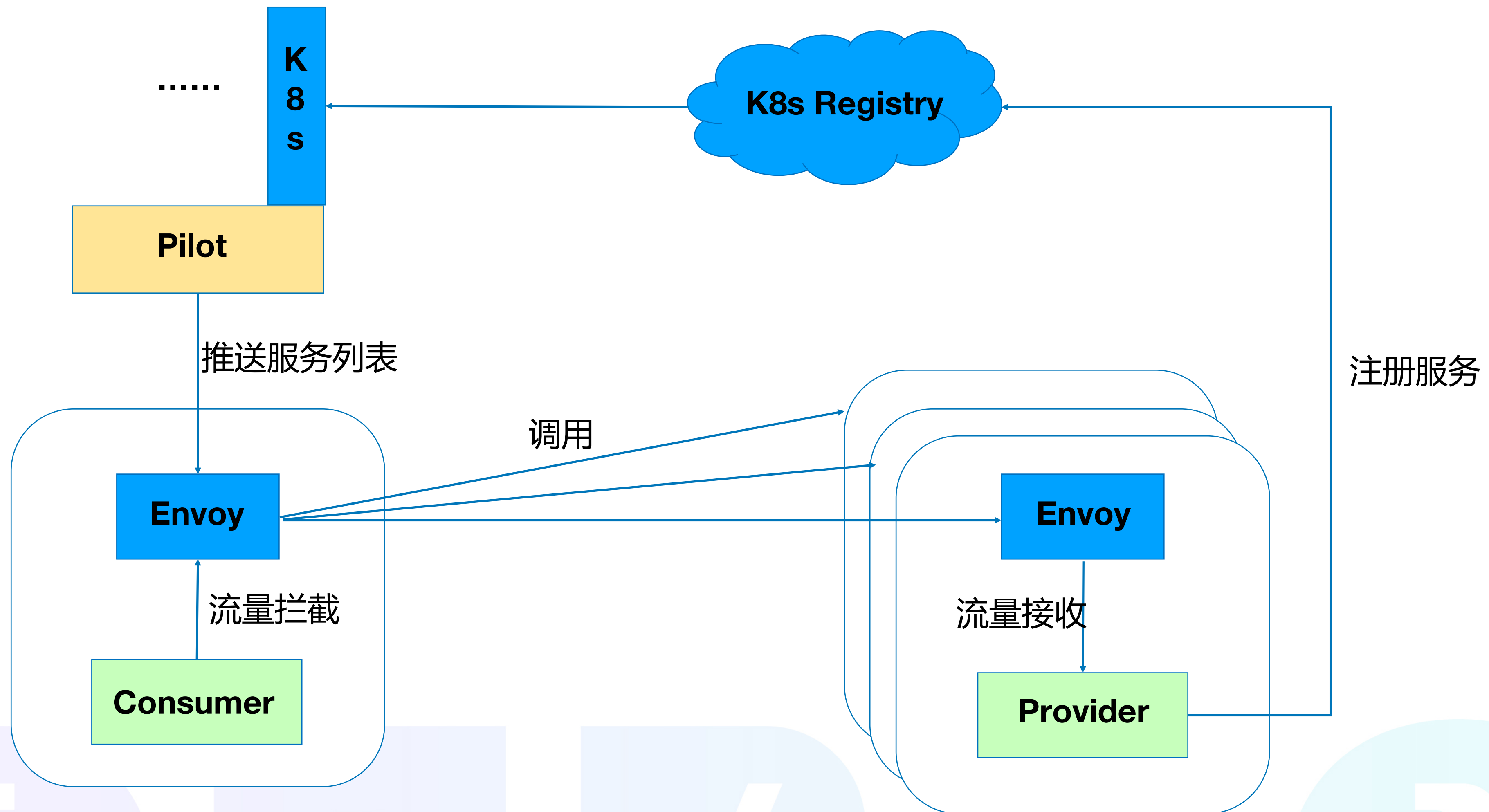
服务网格

SideCar模式，负责服务之间的连接、路由、限流、熔断、加密

Istio



Istio是怎么服务发现的



目录

01

Nacos及Istio简介

02

MCP协议介绍

03

MCP协议对接实战

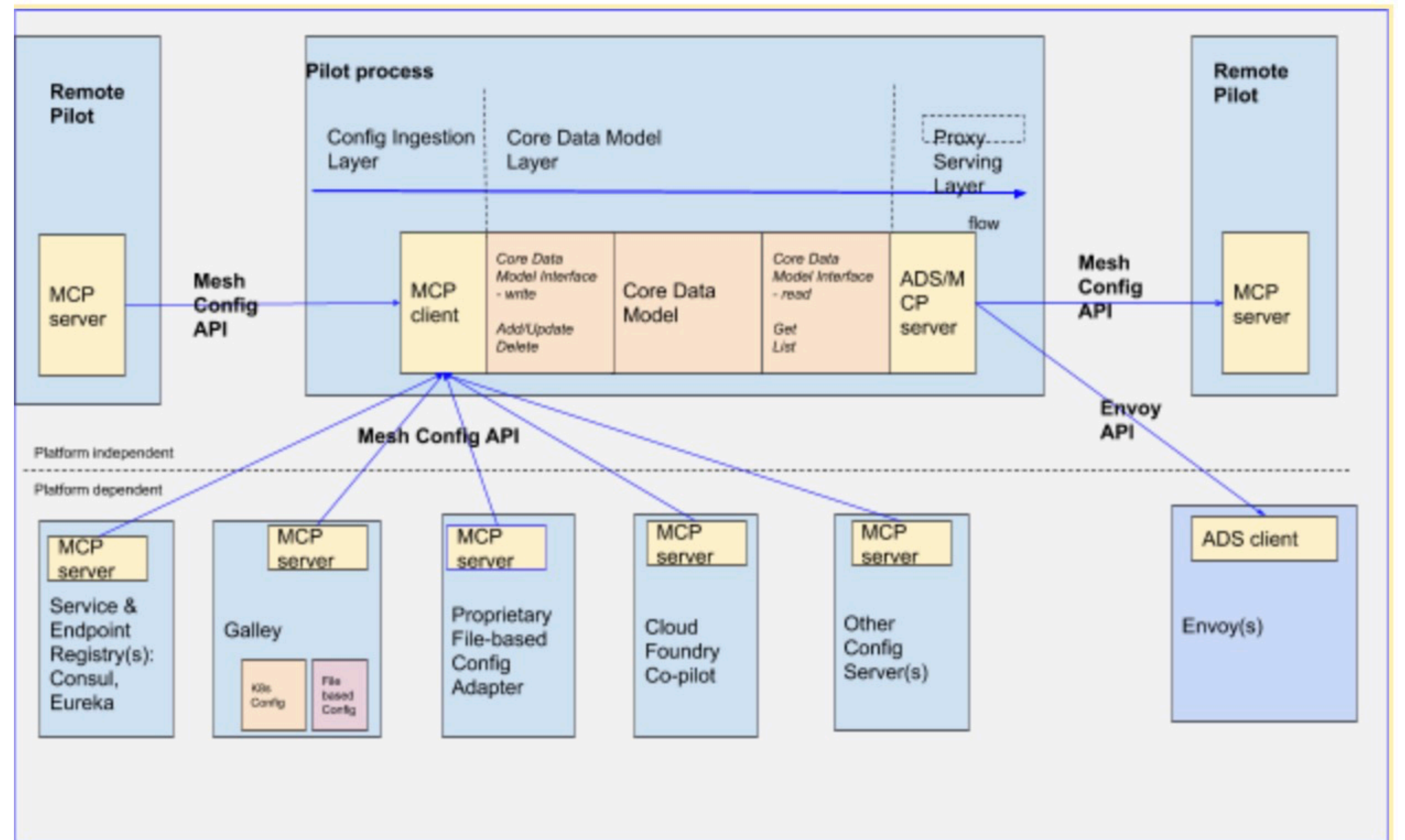
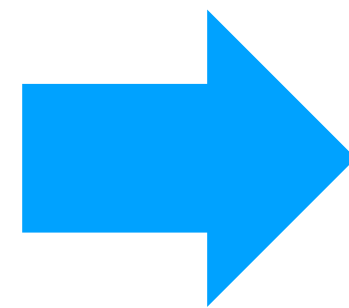
Pilot的重新设计

Pilot目前存在的问题

- 强依赖K8S
- 各插件代码复杂，协议不统一，代码难维护
- 进程内引用其他组件，稳定性差

Pilot重新设计

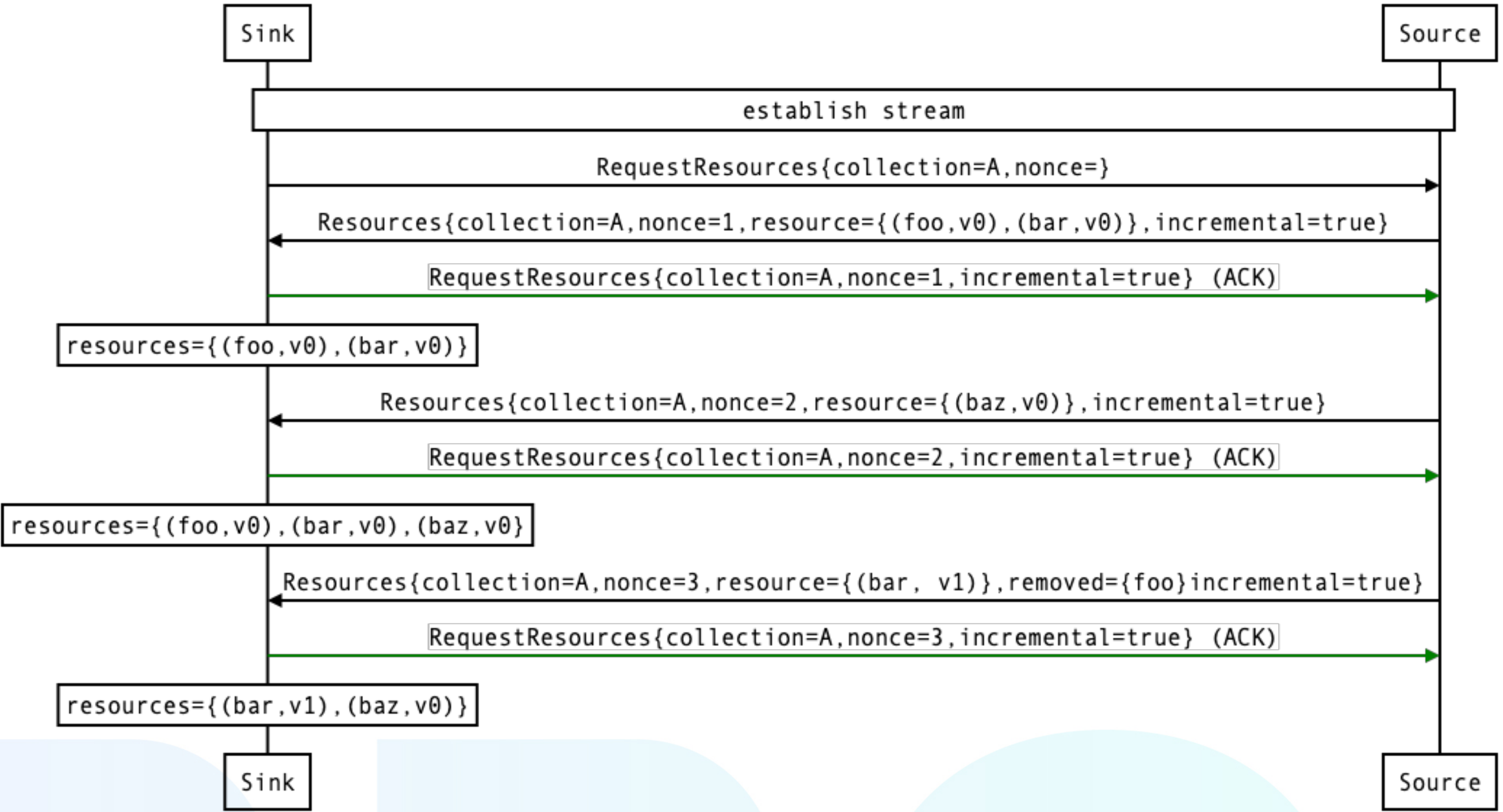
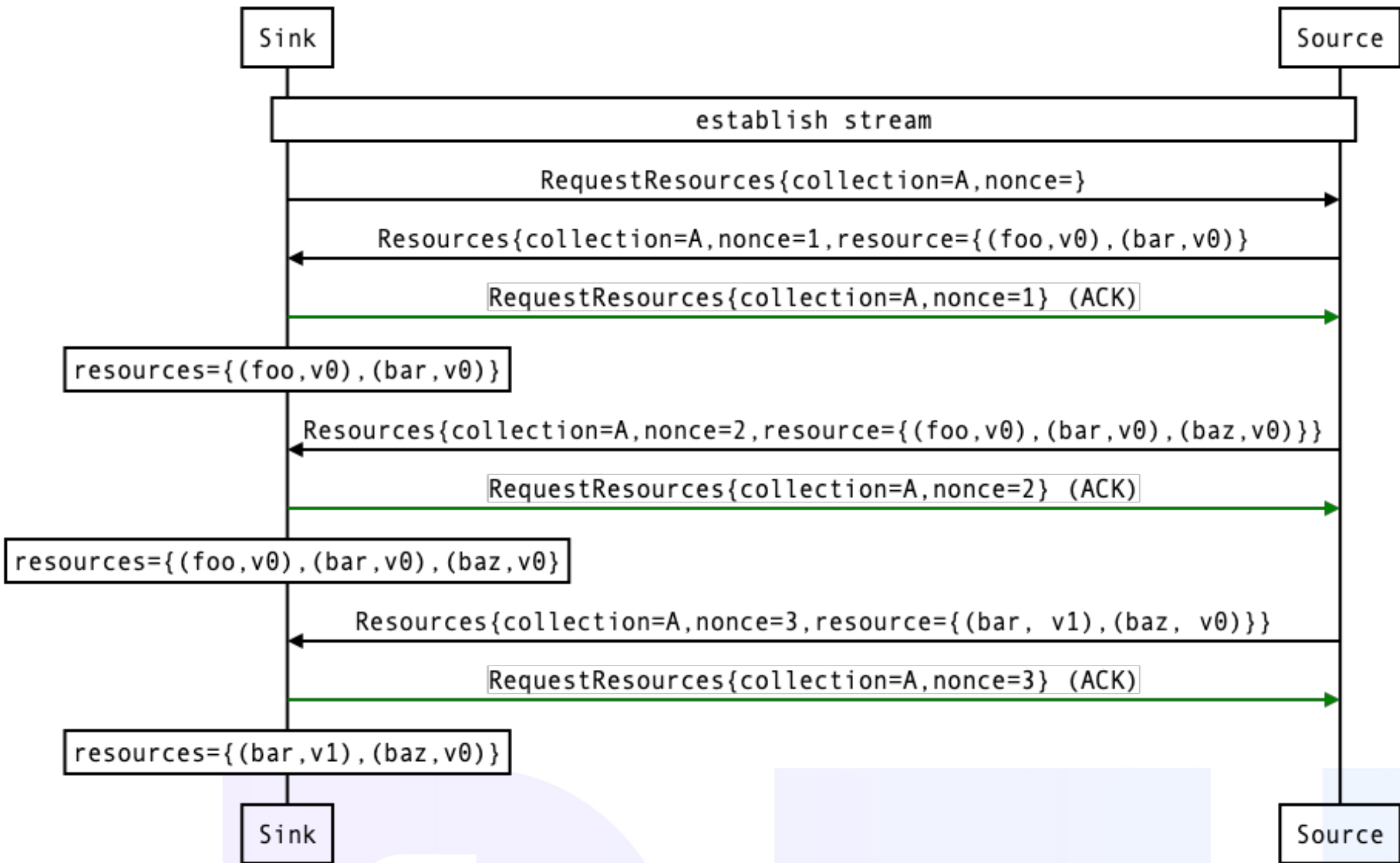
- 去K8S依赖，将逻辑放到Galley
- 上游组件通过基于MCP (Mesh Configuration Protocol) 协议的Mesh Config API进行通讯



MCP协议详情

- Sink：数据的接收端
- Source：数据的发送端即数据源
- Sink端可以主动发送RequestResources请求来向Source端要求数据，同时Source端也可以主动将Resources数据发送给Sink端
- Sink端在收到数据后会返回一个ACK确认

- 增量推送：可以在RequestResources请求中增加incremental=true字段，这样Sink在收到数据后，会根据增量的形式进行数据的更新
- 多源：Pilot可以配置多个Source，每个Source之间的数据是隔离的



目录

01

Nacos及Istio简介

02

MCP协议介绍

03

MCP协议对接实战



Nacos对接Istio背景

Nacos与Istio完全可以共存：作为Pilot的MCP Server，下发服务信息和配置信息

云原生时代已经来临：K8s、Service Mesh、Serverless是目前最火的三块云原生领域

75%容器化

40%线上运行K8s

38%使用serverless

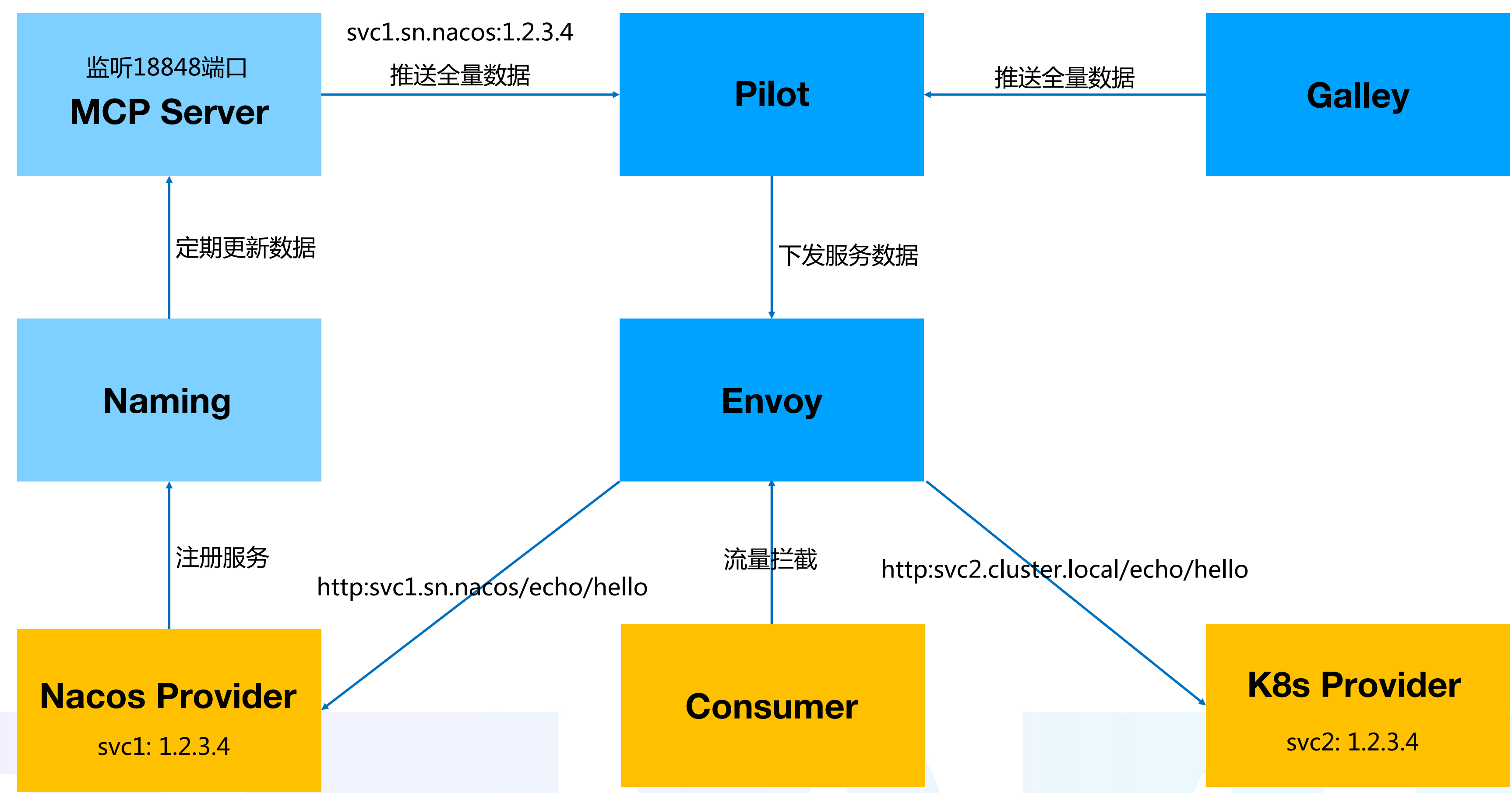
17%线上运行Istio

(来源：CNCF 2018年度报告)

阿里巴巴大规模场景落地：基于Nacos的服务发现、配置管理与Service Mesh融合



Nacos实现MCP协议设计



Demo时间

Nacos社区群

NACOS

4113人



扫一扫群二维码，立刻加入该群。



1

Thank you !

01010101