

架构实战营模块6

第3课：微服务基础设施选型

一手微信study322 价格更优惠
有正版课找我 高价回收帮回血

李运华

前阿里资深技术专家（P9）

教学目标

1. 掌握微服务基础设施架构
2. 掌握微服务框架三种模式
3. 掌握常见微服务框架选择技巧

一手微信study322 价格更优惠
有正版课找我 高价回收帮回血



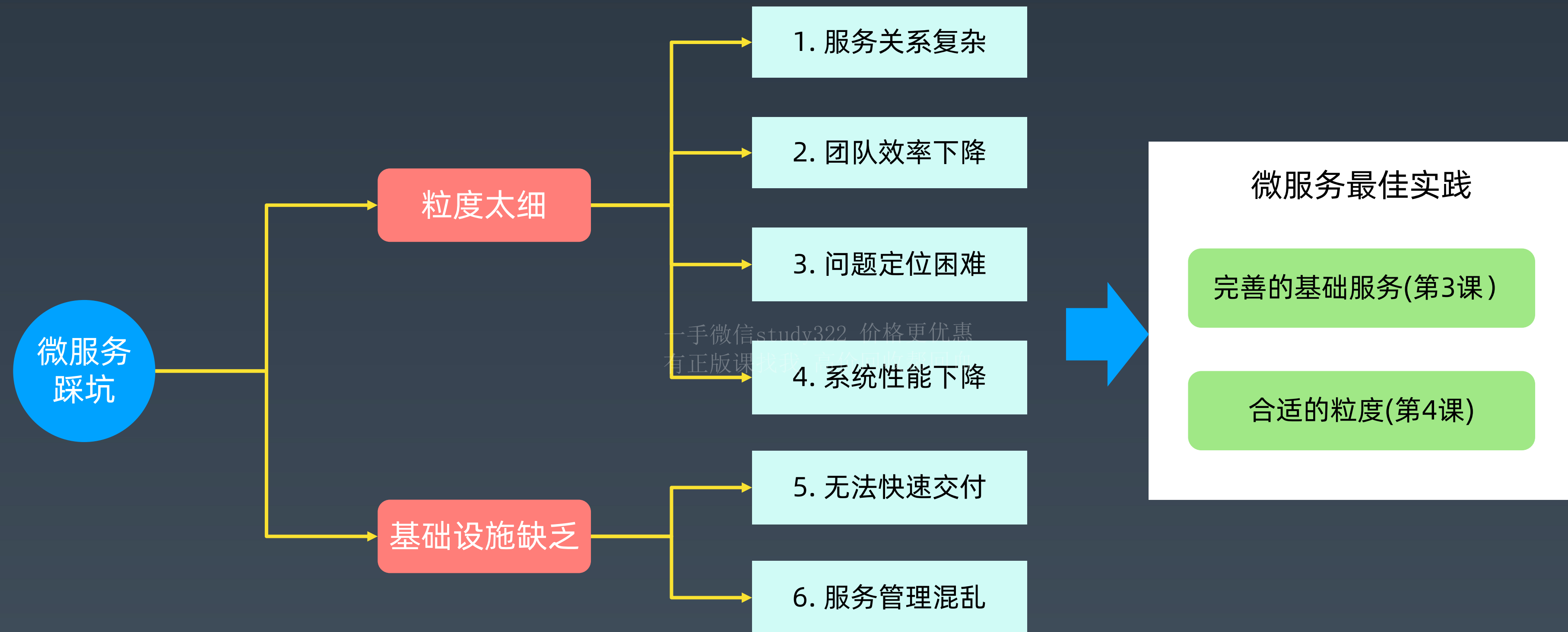
遇事不决 Spring，选择太多 Apache!

目录

1. 微服务基础设施架构
2. 微服务框架模式
3. 常见微服务框架选择

一手微信study322 价格更优惠
有正版课找我 高价回收帮回血

课前回顾 - 微服务陷阱



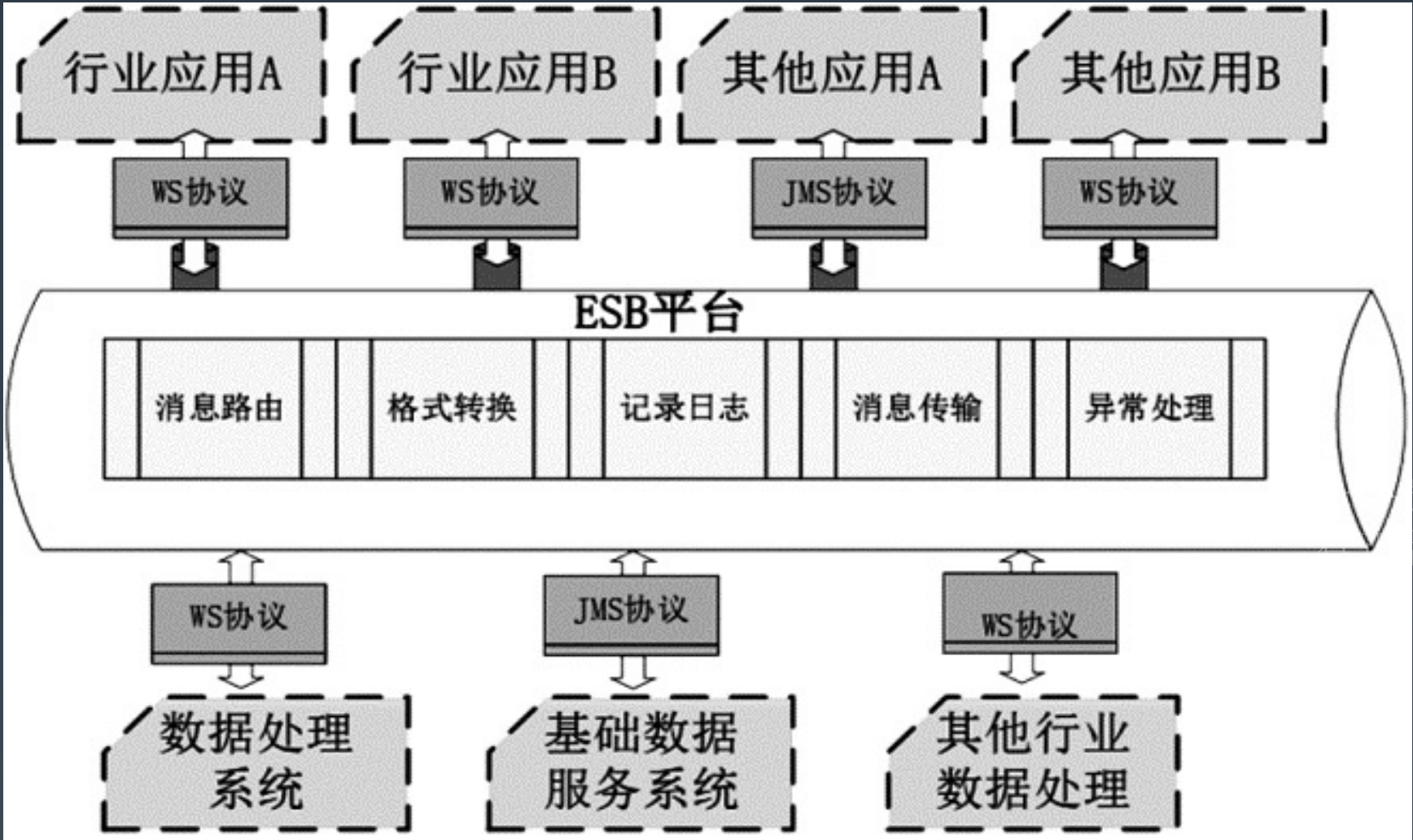
1. 微服务基础设施架构

一手微信study522 价格更优惠
有正版课找我 高价回收帮回血

微服务基础设施架构全貌



微服务 vs SOA (Round 2)



VS

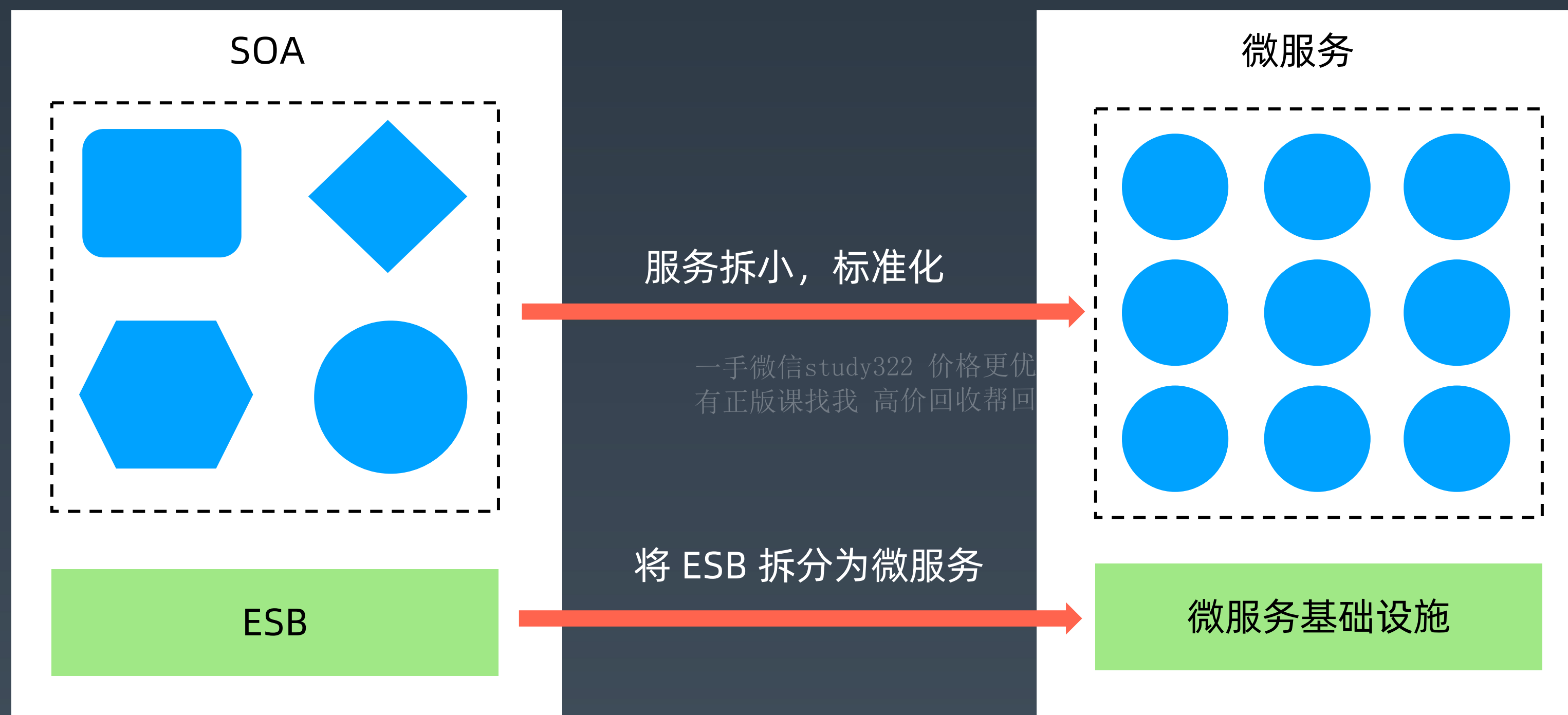
微信study322 价格更优
坂课找我 高价回收帮回

服务接入层	服务网关	服务流控	服务降级	服务安全
服务运行层	服务注册	服务发现	服务路由	服务容错
技术支撑层	接口框架	分布式事务	自动化测试	容器编排
	自动化部署	灰度发布	服务监控	服务跟踪
基础设施层	配置中心	日志中心	分布式锁	消息队列



屠龙少年变超级恶龙，原因是什么？

微服务 vs SOA (Round 3)



已经有公司将自己不那么复杂的业务开始合并了! [参考链接](#)

微服务基础设施优先级

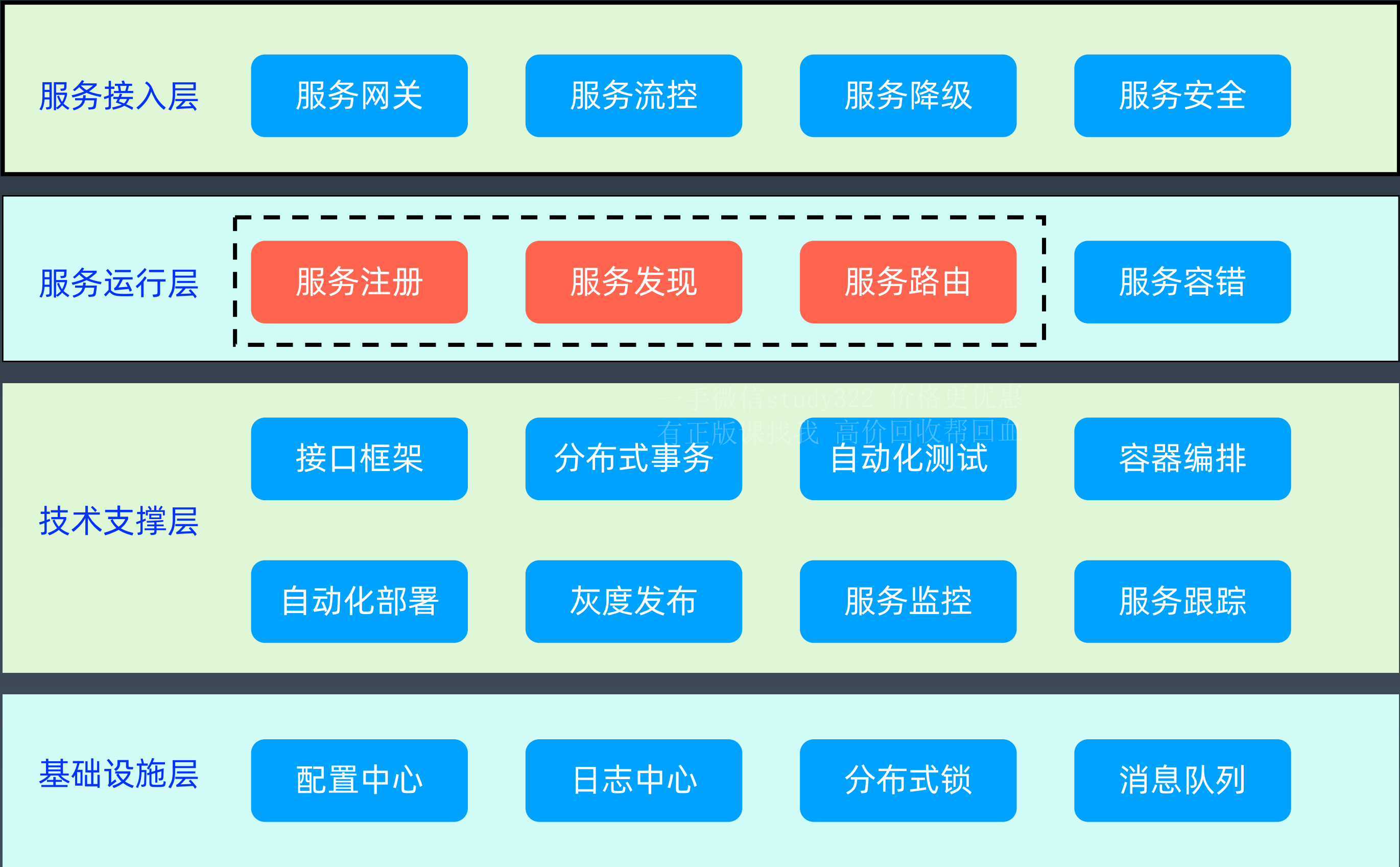


用不同的编程语言来实现微服务，存在什么问题？

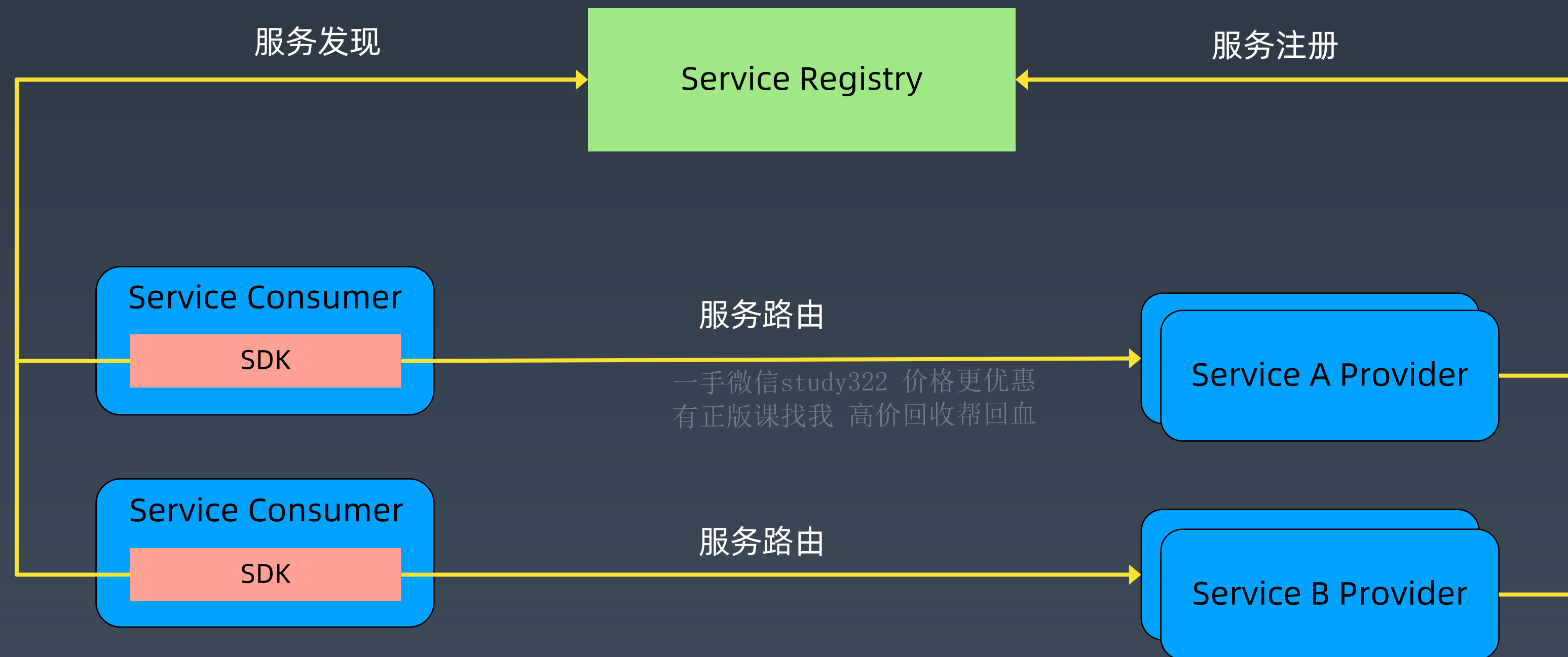
2. 微服务框架模式

手微信study322 价格更优惠
有正版课找我 高价回收帮回血

微服务框架核心



模式1 - 嵌入 SDK 式



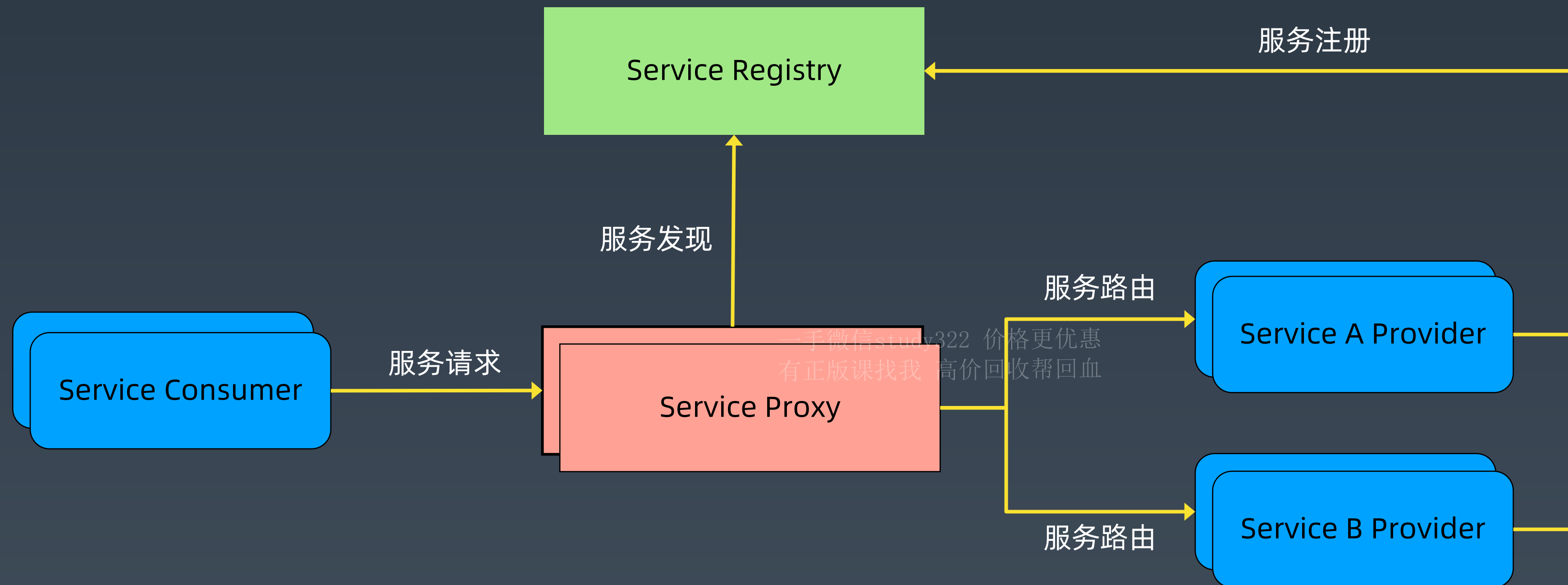
【优点】

1. 架构简单，天然支持高性能高可用；
2. 维护简单，无需维护独立的 Proxy 节点。

【缺点】

1. 应用侵入，需要集成 SDK，并联动升级；
2. 多语言重复开发 SDK。

模式2 - 反向代理式



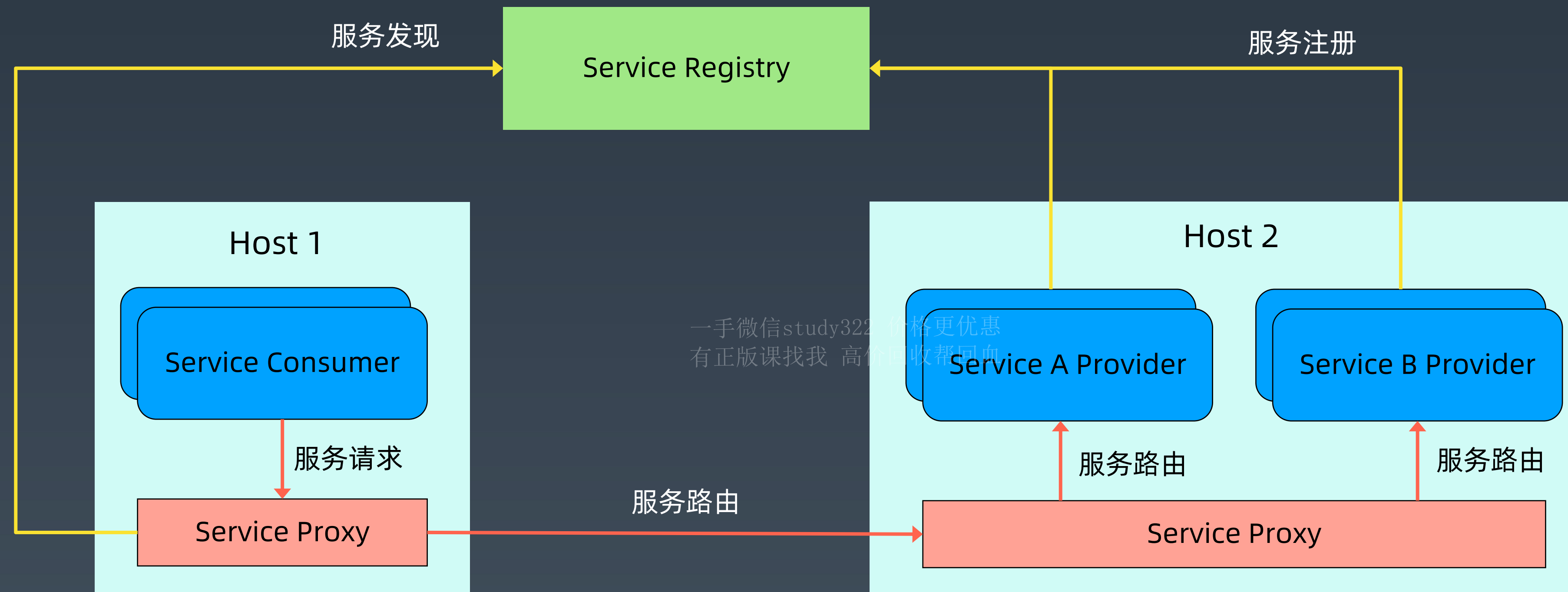
【优点】

1. 应用无侵入;
2. 天然支持多语言。

【缺点】

1. Service Proxy 需要通过集群来做高性能高可用;
2. 维护复杂, 需要维护 Service Proxy 集群。

模式3 - 网络代理式(Service Mesh)



【优点】

1. 应用无侵入;
2. 天然支持多语言;
3. 天然支持架构高性能、高可用。

【缺点】

1. 维护复杂, 需要维护每台服务器上的 Service Proxy;
2. 单台服务器的 Service Proxy 是单点;
3. 全链路请求性能会下降。

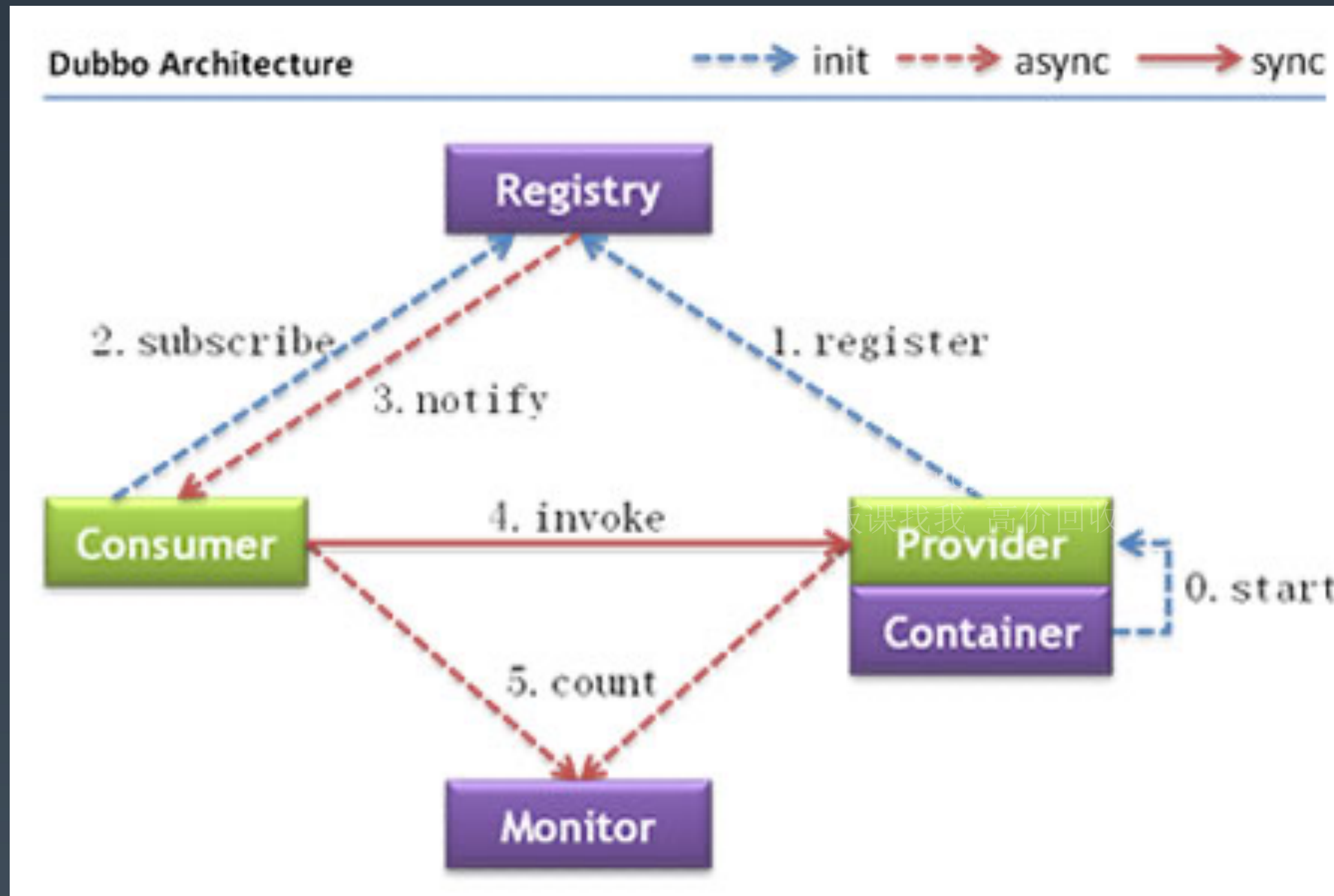
微服务框架模式对比

	嵌入 SDK	反向代理	网络代理
多语言	每个语言都需要开发 SDK	天然支持语言	天然支持语言
应用侵入	需要集成 SDK	无	无
可维护性	无需额外维护工作	维护 Proxy 服务器集群	维护每台机器上的 Proxy 节点，数量很多。
高性能	天然支持	需要通过 Proxy 集群来实现	整体架构是高性能的，但全链路请求性能会下降。
高可用	天然支持	需要通过 Proxy 集群来实现	整体架构高可用，单台服务器上的 Proxy 是单点。
适应场景	团队技术栈统一为某个单语言	团队技术栈多语言，服务集群规模不大（1000 服务器以内）。	团队技术栈多语言，服务集群规模超大（1000 服务器以上）。

3. 常见微服务框架选择

一手微信study322 价格更优惠
有正版课找我 高价回收帮回血

嵌入式 SDK 样例 - Dubbo



[Apache Dubbo](#) 是一款高性能、轻量级的开源 **Java** 服务框架，提供了六大核心能力：

1. 面向接口代理的高性能 RPC 调用；
2. 智能容错和负载均衡；
3. 服务自动注册和发现；
4. 高度可扩展能力；
5. 运行期流量调度；
6. 可视化的服务治理与运维。

【出品方】

阿里出品，2017年停更过，现在是 Apache 项目。

嵌入式 SDK 样例 - Spring Cloud

What is Spring Cloud?



www.educba.com

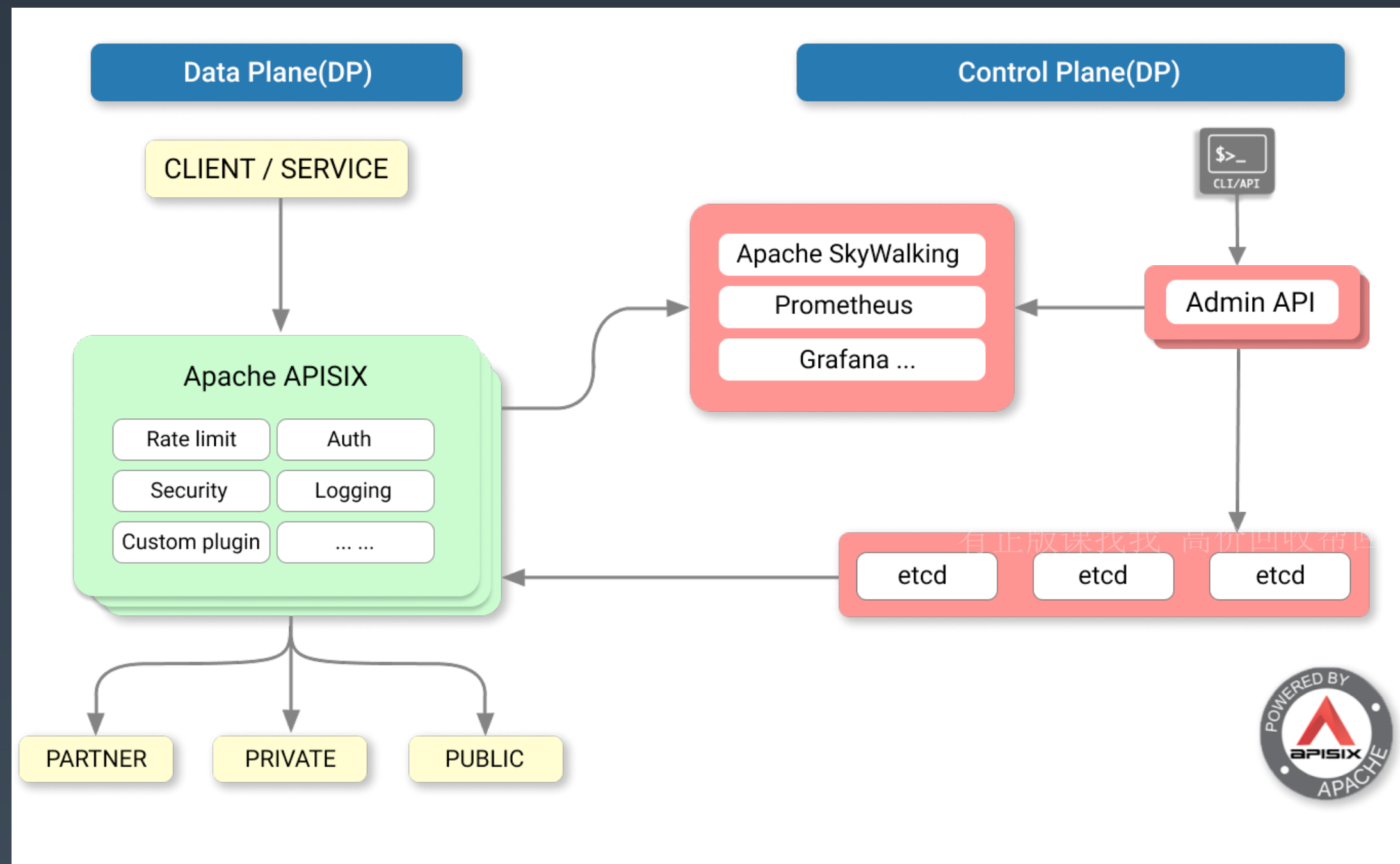
[Spring Cloud](#) provides tools for (Java) developers to quickly build some of the common patterns in distributed systems:

1. Distributed/versioned configuration.
2. Service registration and discovery.
3. Routing.
4. Service-to-service calls.
5. Load balancing.
6. Circuit Breakers.
7. Global locks.
8. Leadership election and cluster state.
9. Distributed messaging.

【出品方】

Spring 基金会, VMWare 公司。

反向代理式案例 - APISIX



[Apache APISIX](#) is a dynamic, real-time, high-performance **API gateway**. Based on the **Nginx library** and **etcd**.

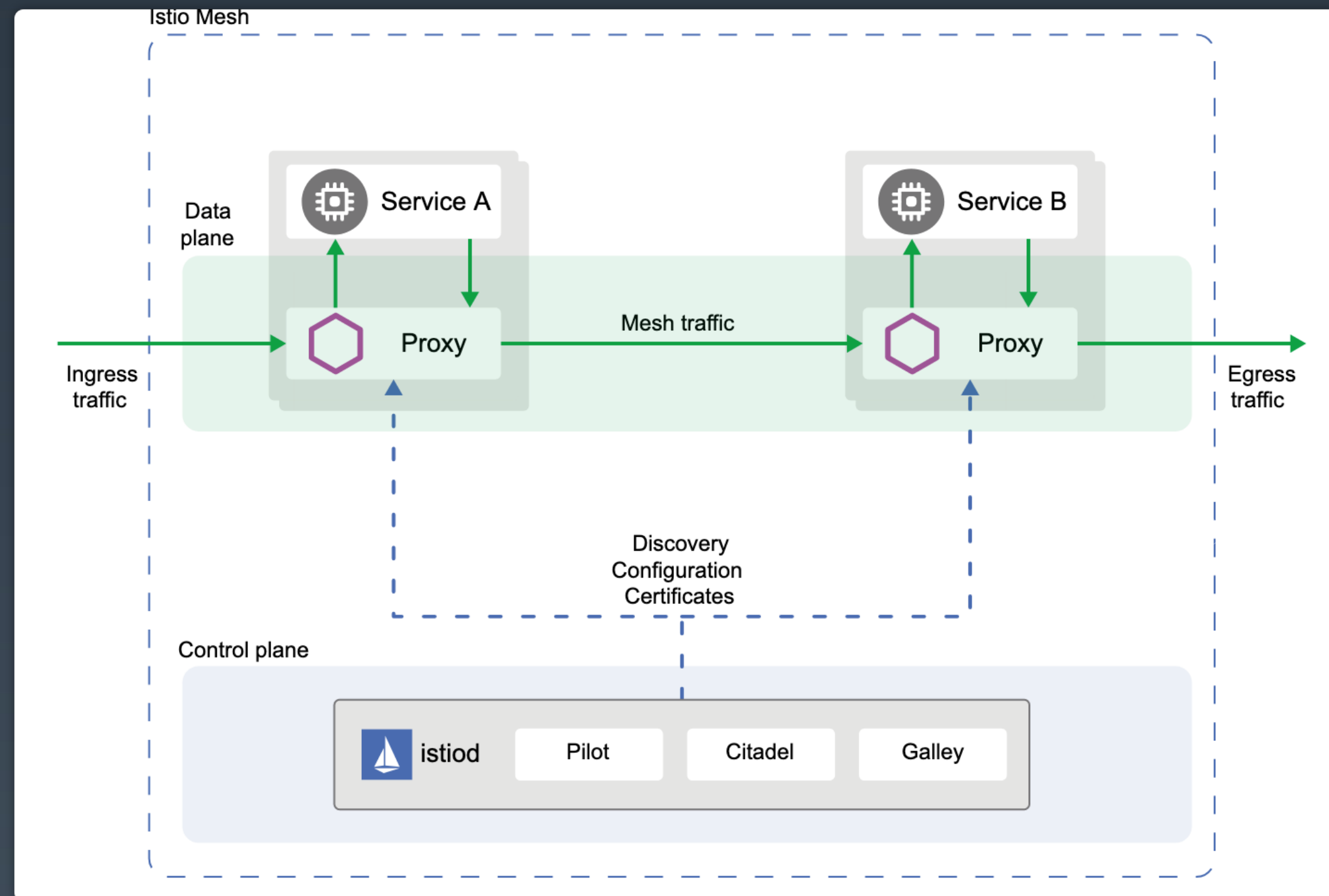
APISIX provides rich traffic management features such as load balancing, dynamic upstream, canary release, circuit breaking, authentication, observability, and more.

You can use Apache APISIX to handle traditional **north-south traffic**, as well as **east-west traffic** between services. It can also be used as a k8s ingress controller.

【出品方】

支流科技(国产), 现在是 Apache 基金会维护。

Service Mesh 案例 - istio



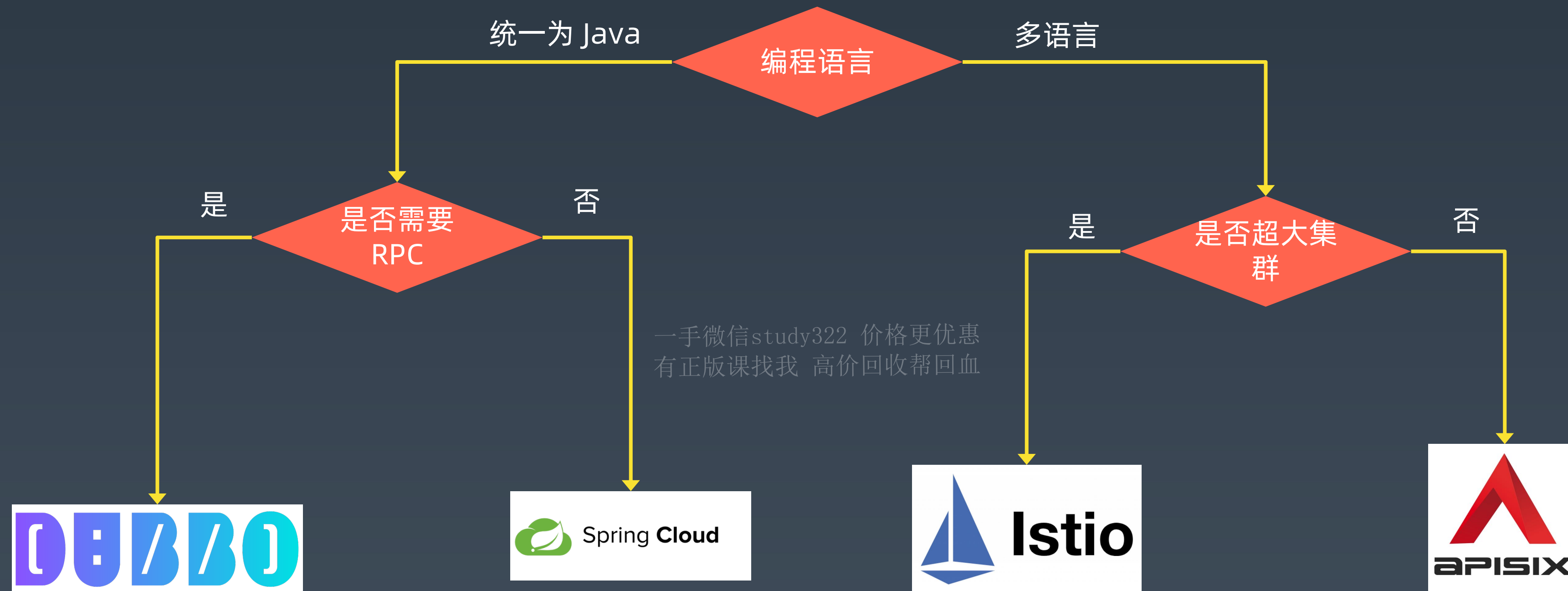
[istio](#) makes it easy to create a network of deployed services with load balancing, service-to-service authentication, monitoring, and more, with few or no code changes in service code.

1. Automatic load.
2. Fine-grained control of traffic behavior with rich routing rules, retries, failovers, and fault injection.
3. A pluggable policy layer and configuration API supporting access controls, rate limits and quotas.
4. Automatic metrics, logs, and traces for all traffic within a cluster, including cluster ingress and egress.
5. Secure service-to-service communication in a cluster with strong identity-based authentication and authorization.

【出品方】

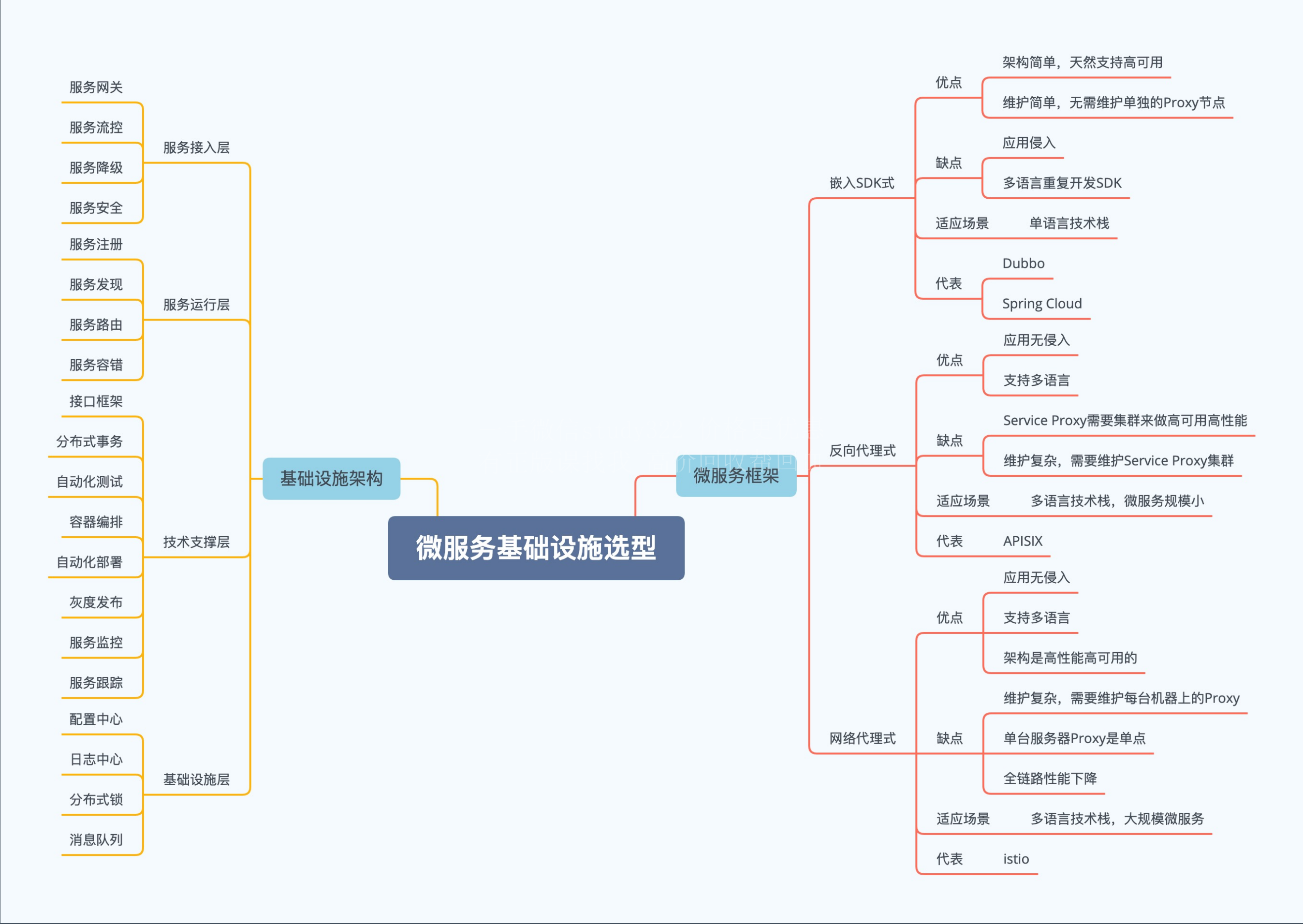
Google、IBM 和 Lyft 联合开源

如何选择开源微服务框架



遇事不决 Spring，选择太多 Apache！

本节思维导图



随堂测验

【判断题】

1. 微服务的基础设施比 SOA 的 ESB 实际上要更庞大和复杂。
2. 微服务基础设施都是必不可少的，需要全部都准备好后才开始微服务拆分。
3. 微服务框架的核心是服务注册发现。
4. Service Mesh 无需侵入应用，支持多语言，应该作为第一选择方案。
5. PHP、Java、Python 三种语言混合开发的微服务系统，可以用 APISIX，也可以用 istio。

一手微信study322 价格更优惠
有正版课找我 高价回收帮回血

【思考题】

既然微服务的基础设施比 ESB 还要复杂了，微服务还有什么存在的必要？



茶歇时间



八卦，趣闻，内幕.....

THANKS

一手微信study322 价格更优惠
有正版课找我 高价回收帮回血