

蚂蚁 SOFAShark Mesh 的大规模落地实践与展望

宋顺（齐天）

蚂蚁集团可信原生技术部



蚂蚁集团高级技术专家
2019年初加入蚂蚁集团，主要负责微服务、Service Mesh 相关产品的研发工作



开源配置中心 Apollo 主要作者
<https://github.com/ctripcorp/apollo>



毕业于复旦大学软件工程系
曾就职于携程、大众点评，负责中间件、后台系统等研发工作

目录

contents

- 一、为什么需要 Service Mesh?
- 二、蚂蚁 SOFAShark Mesh 的大规模落地实践
- 三、展望未来

一、 为什么需要 Service Mesh?

微服务治理与业务逻辑解耦



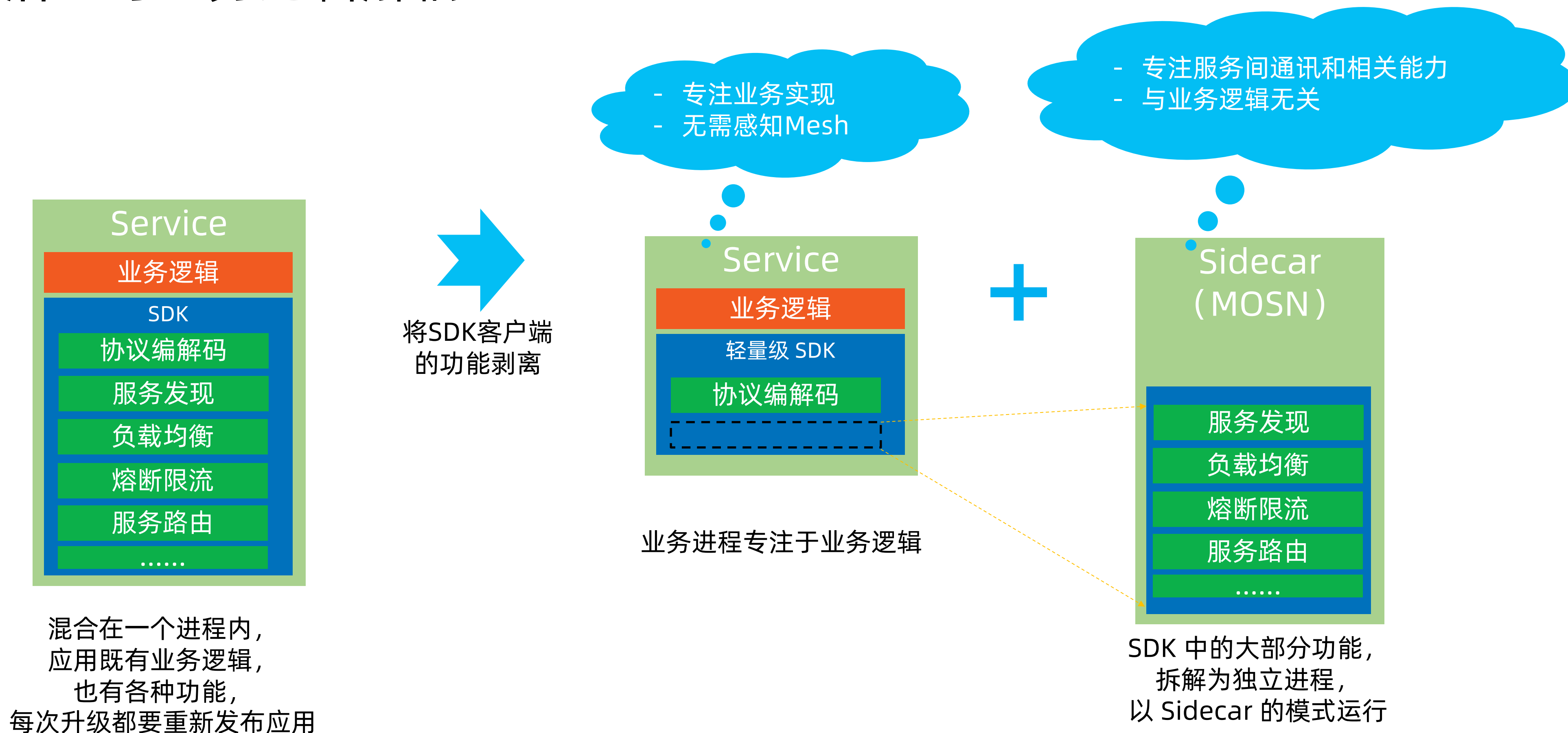
混合在一个进程内，
应用既有业务逻辑，
也有各种功能，
每次升级都要重新发布应用

\$ 升级成本高

版本碎片化严重

中间件演进困难

微服务治理与业务逻辑解耦



将服务治理能力下沉到基础设施, 实现独立演进, 透明升级

异构系统统一治理

 业务百花齐放：前端、搜索推荐、人工智能、安全.....

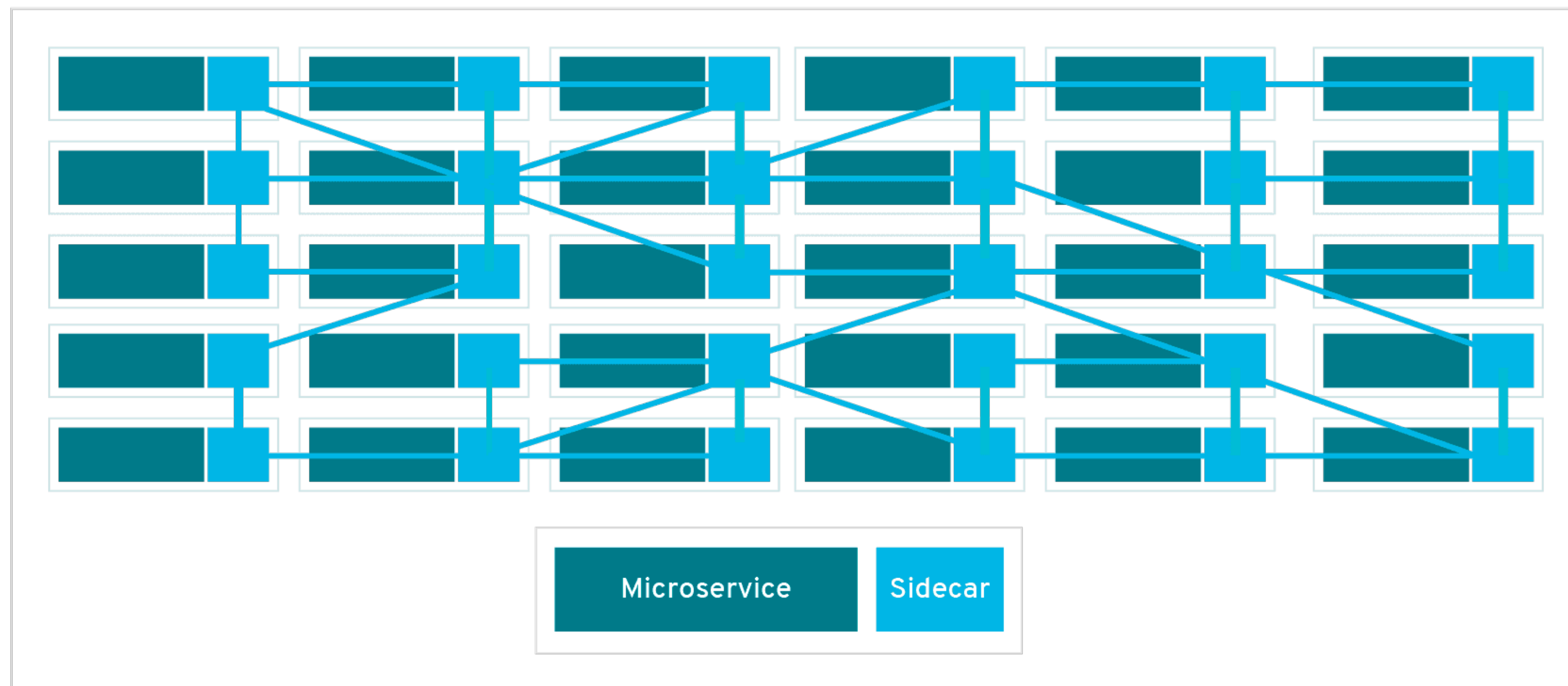
 技术栈多种多样：



异构系统统一治理

🌀 多语言、多协议

🔍 流量控制、监控



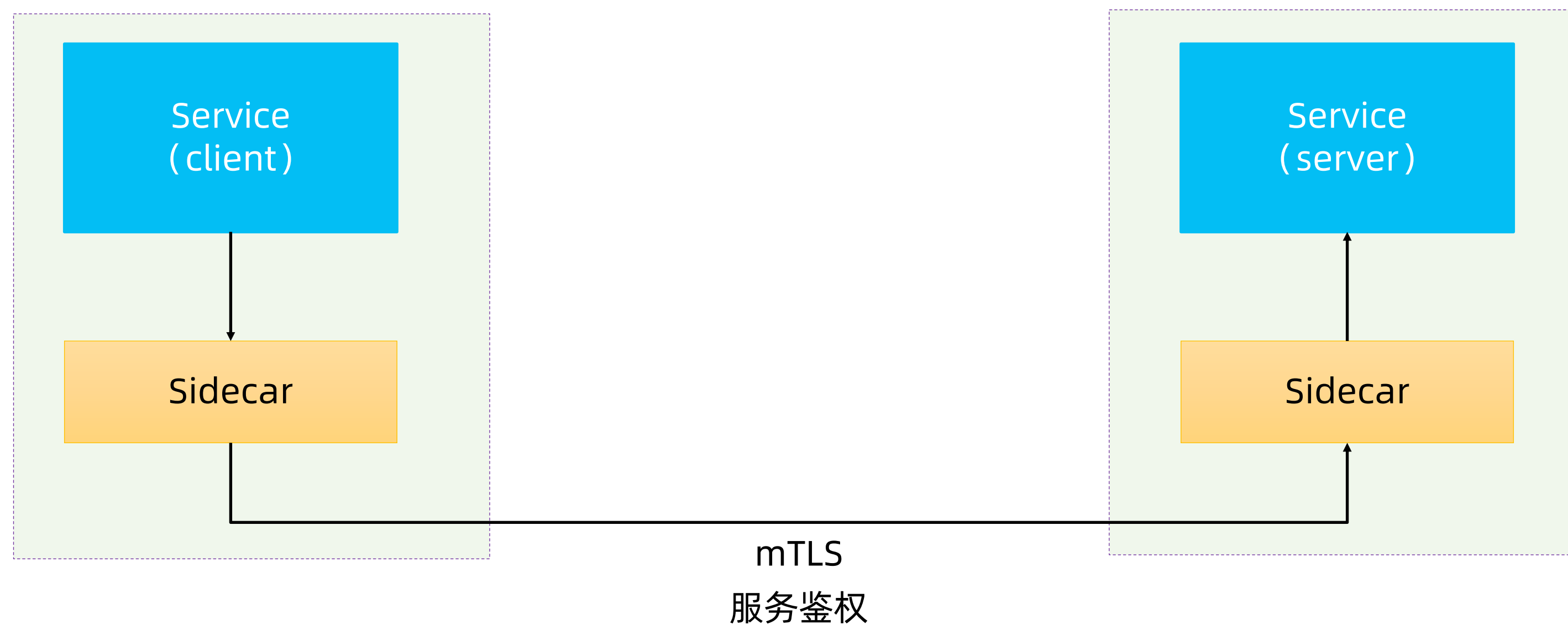
图片来源: <https://www.redhat.com/en/topics/microservices/what-is-a-service-mesh>

金融级网络安全

 身份认证/访问控制

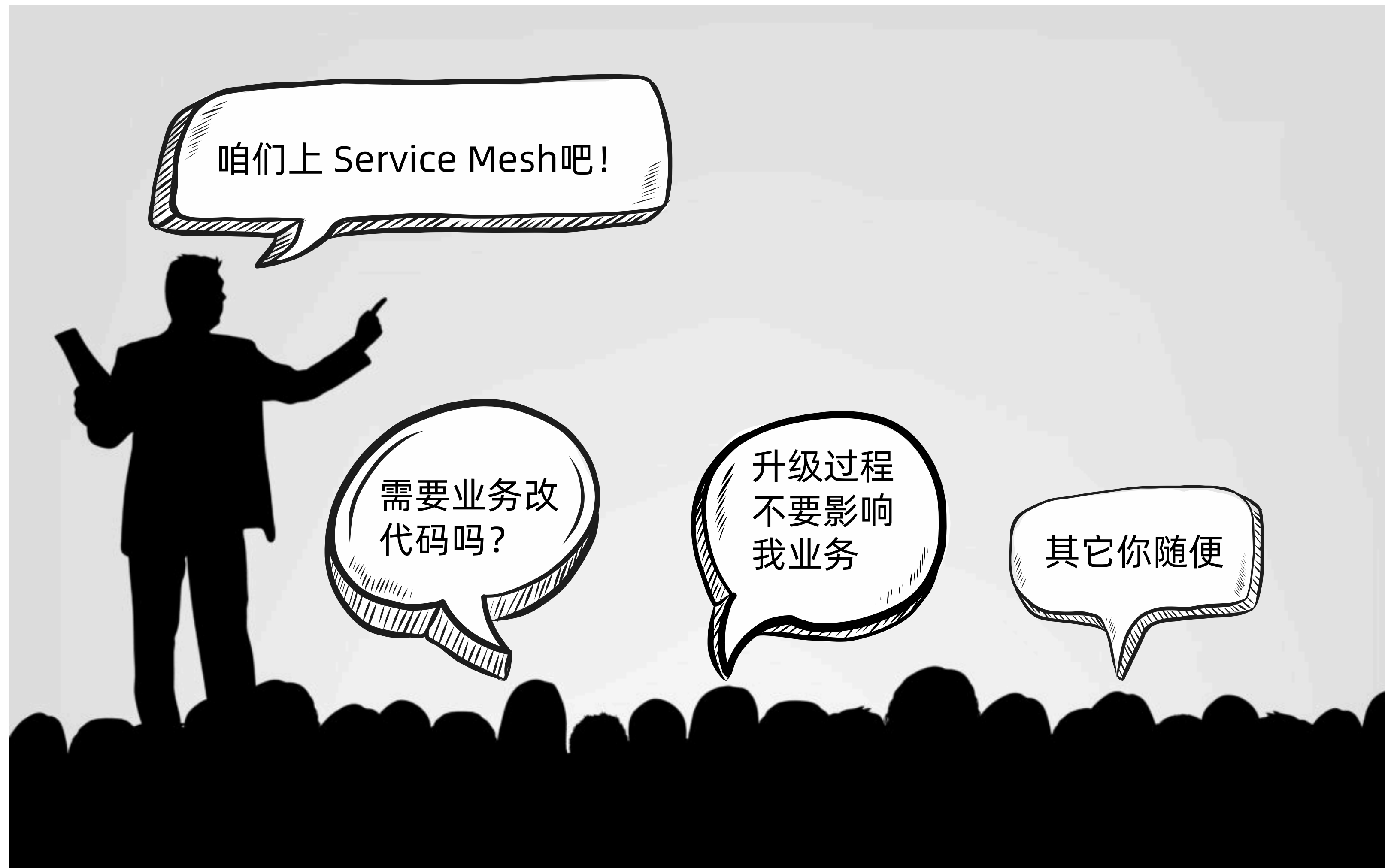
 全链路可信、加密

 零信任网络



二、蚂蚁 SOFAShark Mesh 的大规模落地实践

灵魂拷问



图片来源: <https://www.skipprichard.com/audience-know-thy-speaker-7-tips-audience/>

灵魂拷问

Service Mesh 带来的好处

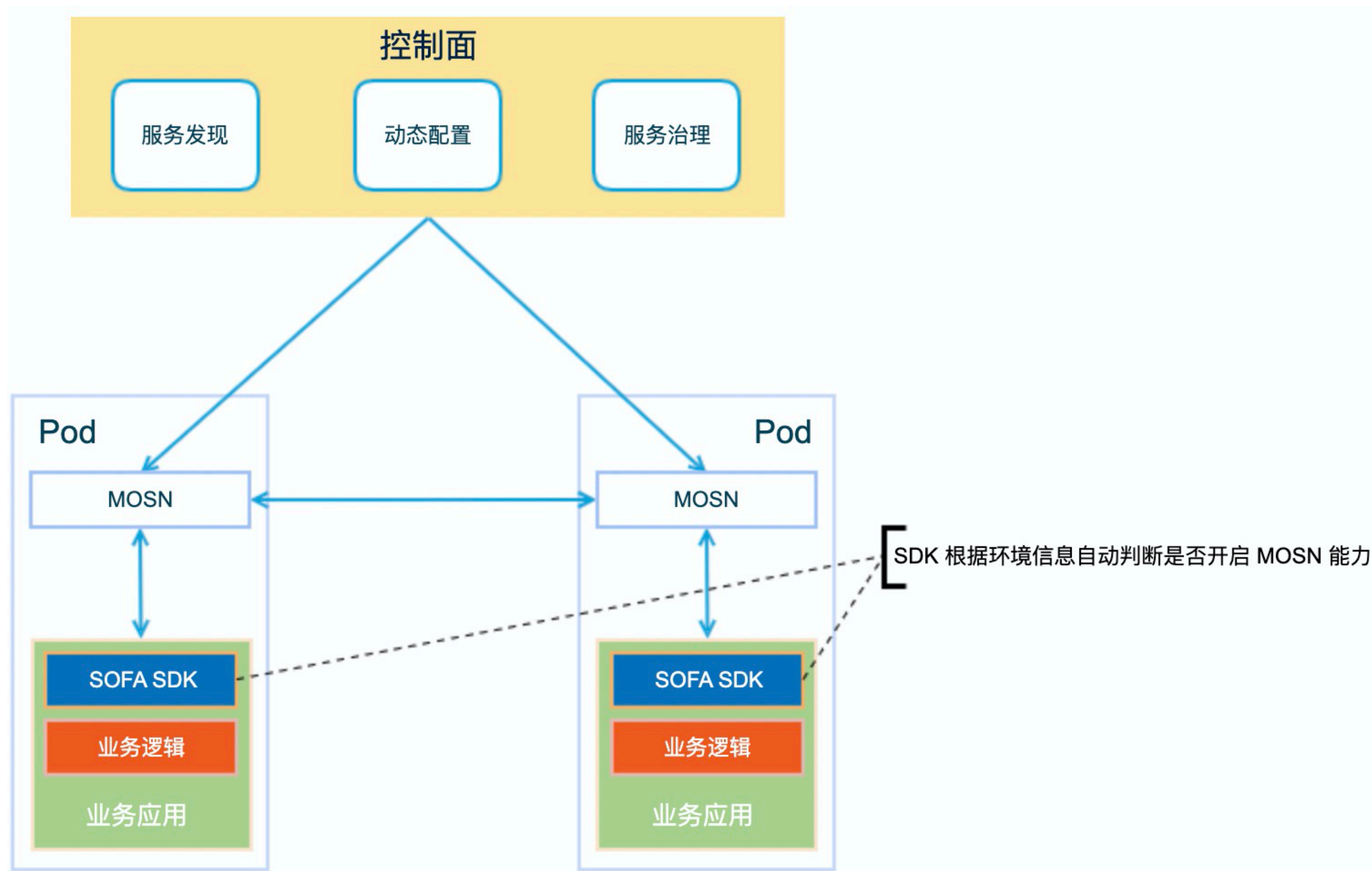
$$\text{产品价值} = (\text{新体验} - \text{旧体验}) - \text{迁移成本}$$

- 1. 接入成本
- 2. 是否平滑迁移
- 3. 稳定性

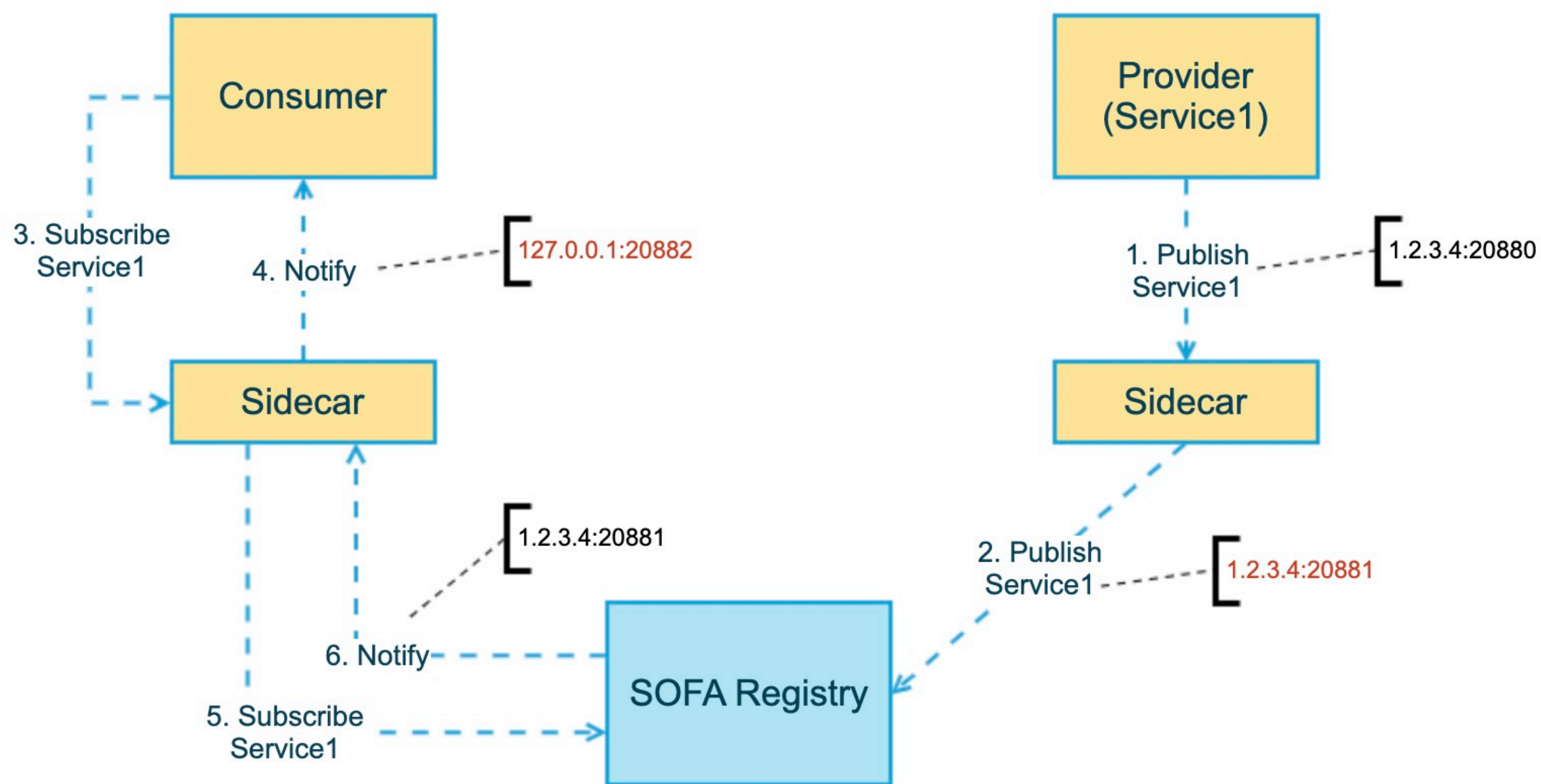
接入成本

不需要改业务代码

升级 SDK，自动识别运行模式

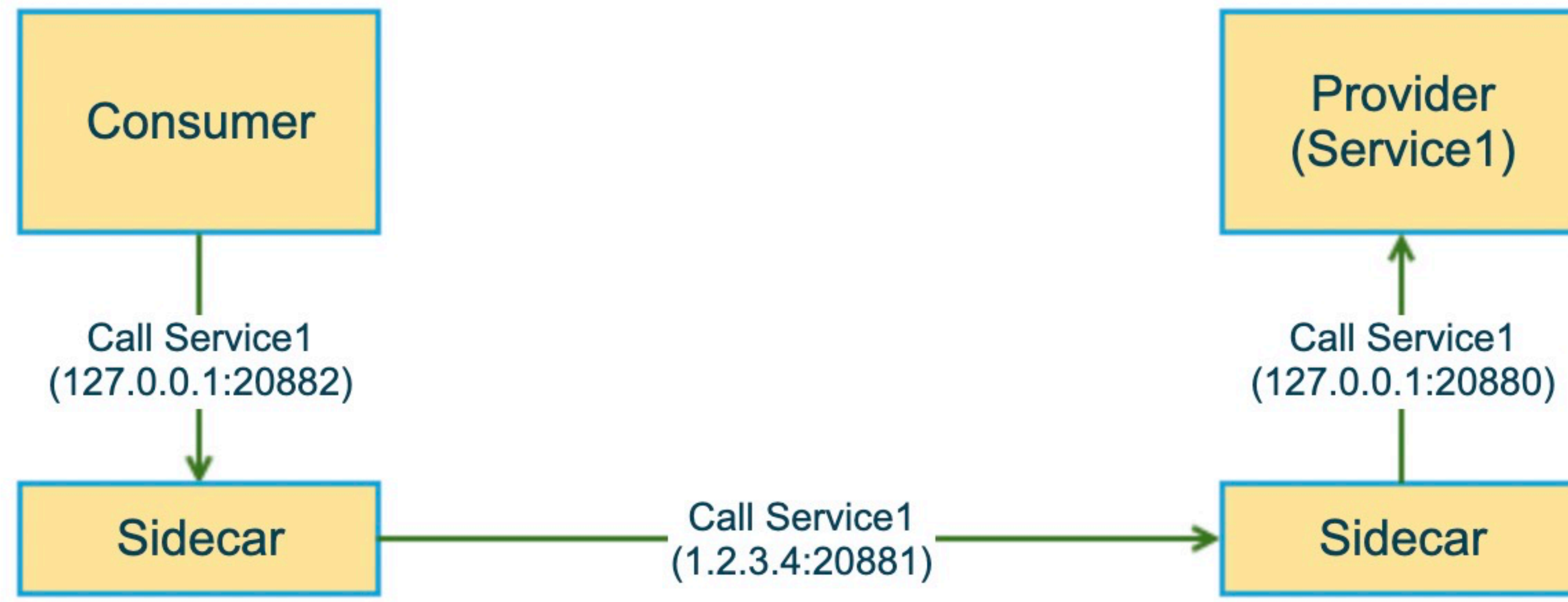


SDK $\rightleftharpoons$ Sidecar



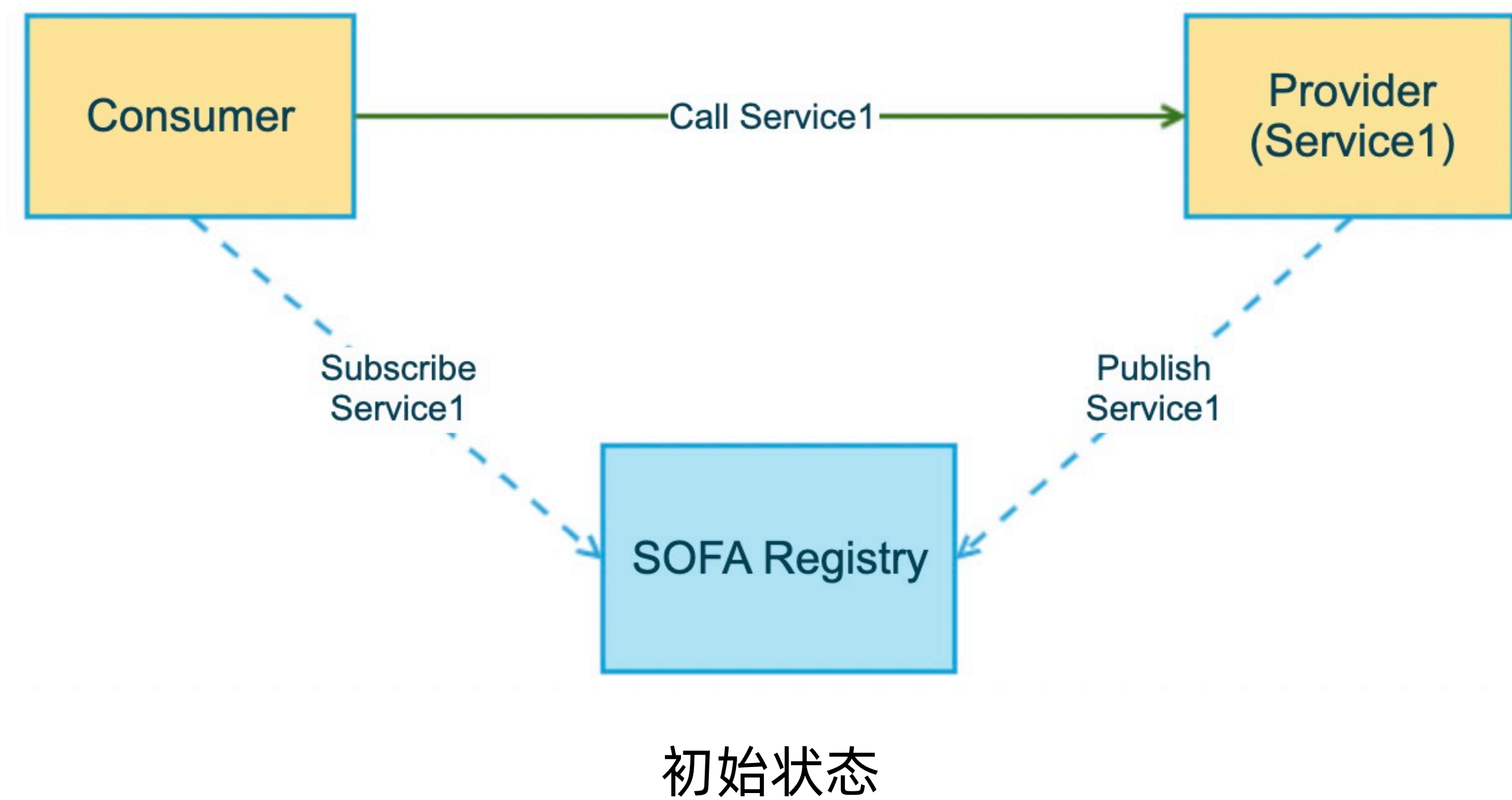
服务发现过程

SDK $\rightleftharpoons$ Sidecar

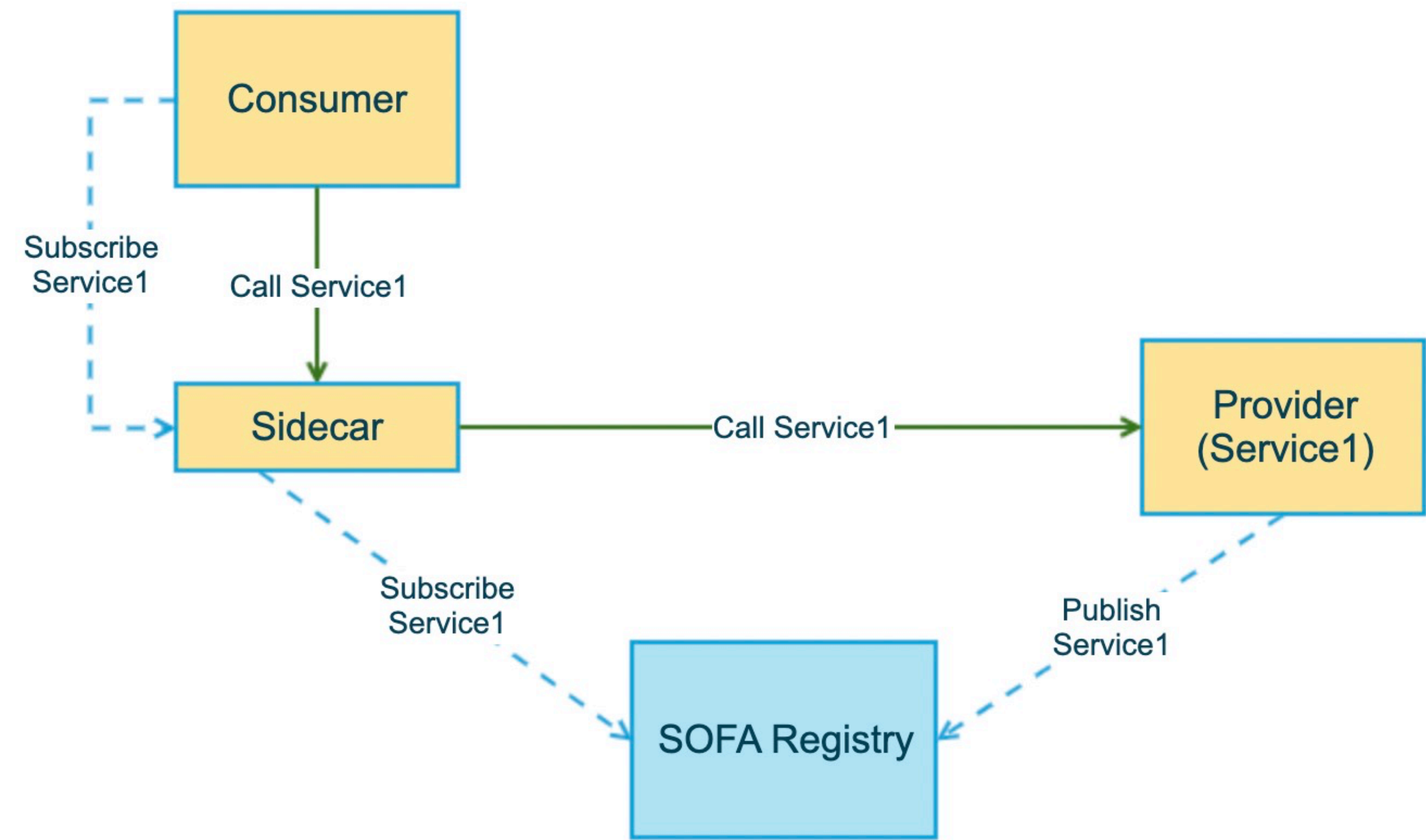


服务通信过程

平滑迁移

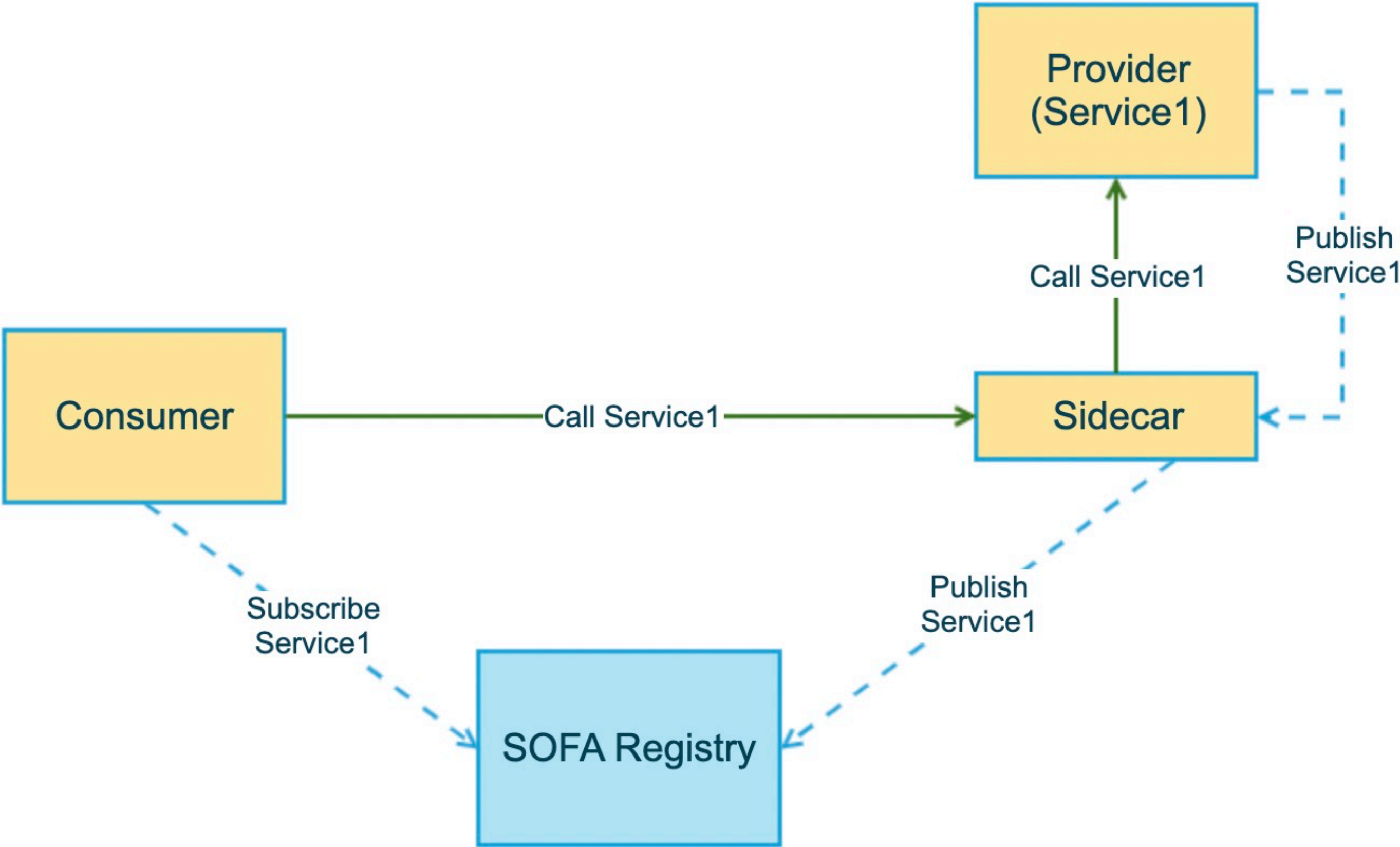


平滑迁移



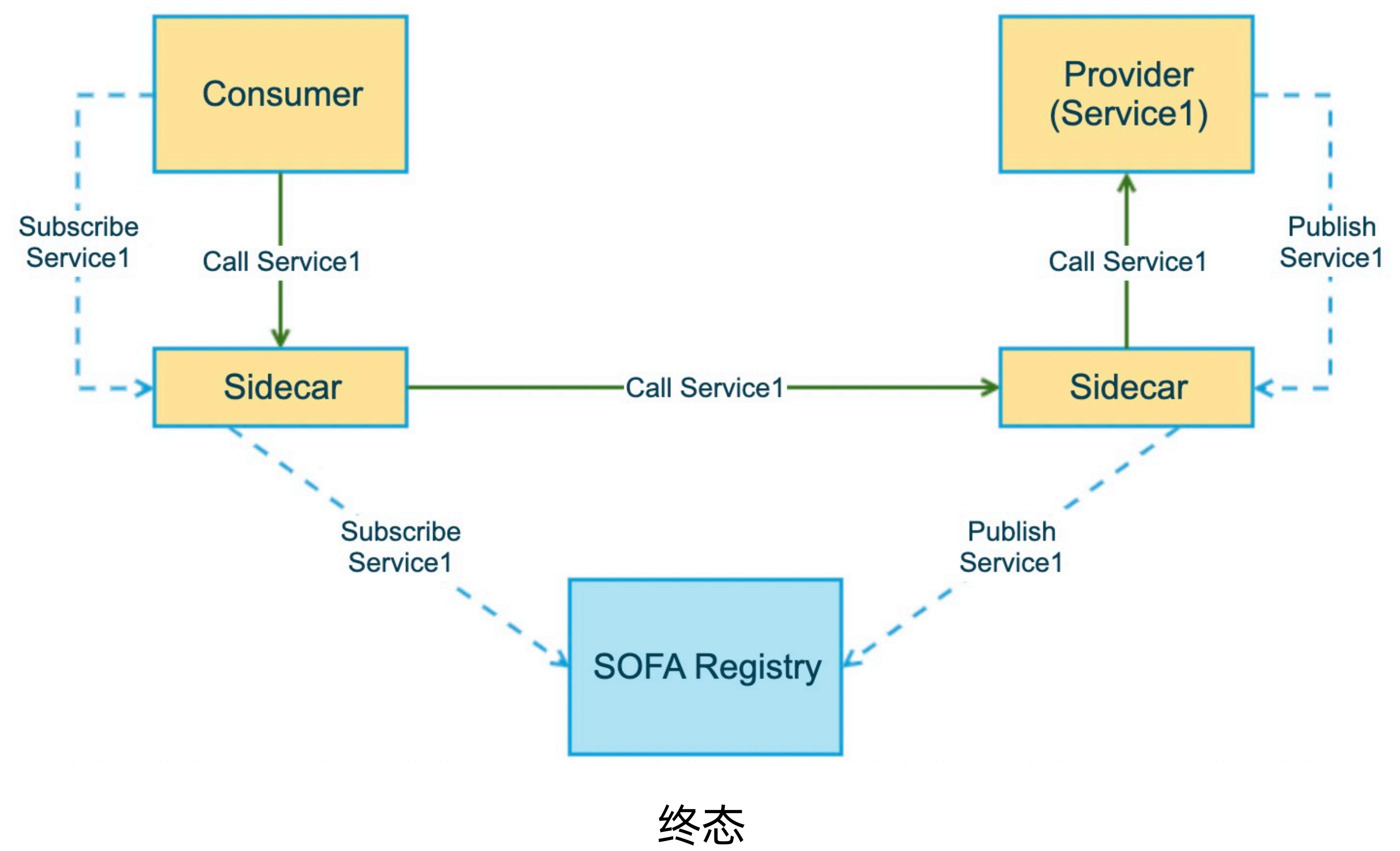
透明迁移调用方

平滑迁移



透明迁移服务方

平滑迁移



稳定性

Service Mesh => 加快基础设施的迭代速度 => 稳定性?

稳定性

无人值守变更

无人值守变更



图片来源: <https://medium.com/@TMTPOST/baidu-driverless-cars-run-in-wuzhen-powered-by-four-leading-technologies-tmtpost-53c0b3072cec>

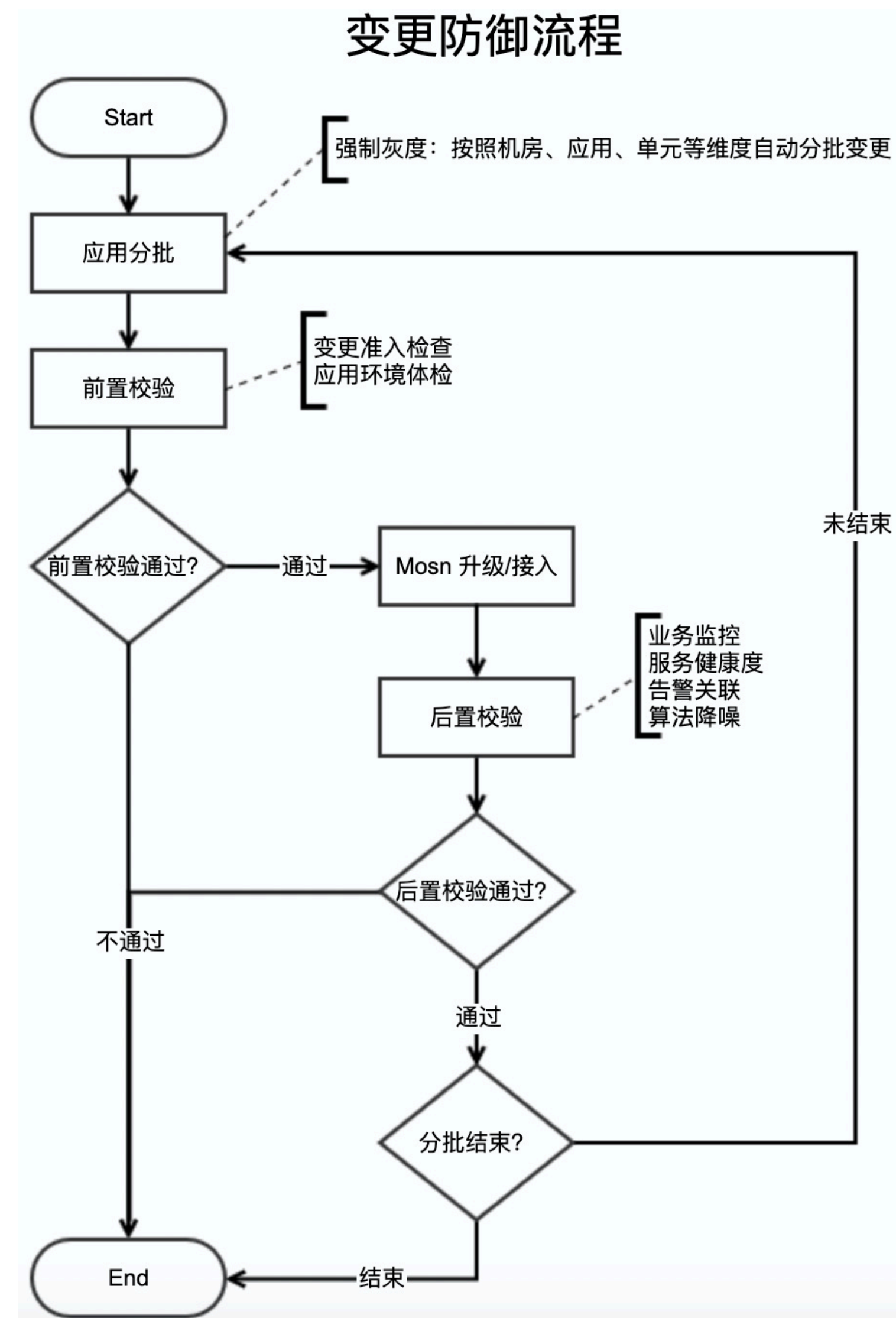
无人值守变更



自动编排分批策略，强制灰度



前置、后置校验，阻断问题变更





自动编排分批策略，强制灰度



前置、后置校验，阻断问题变更



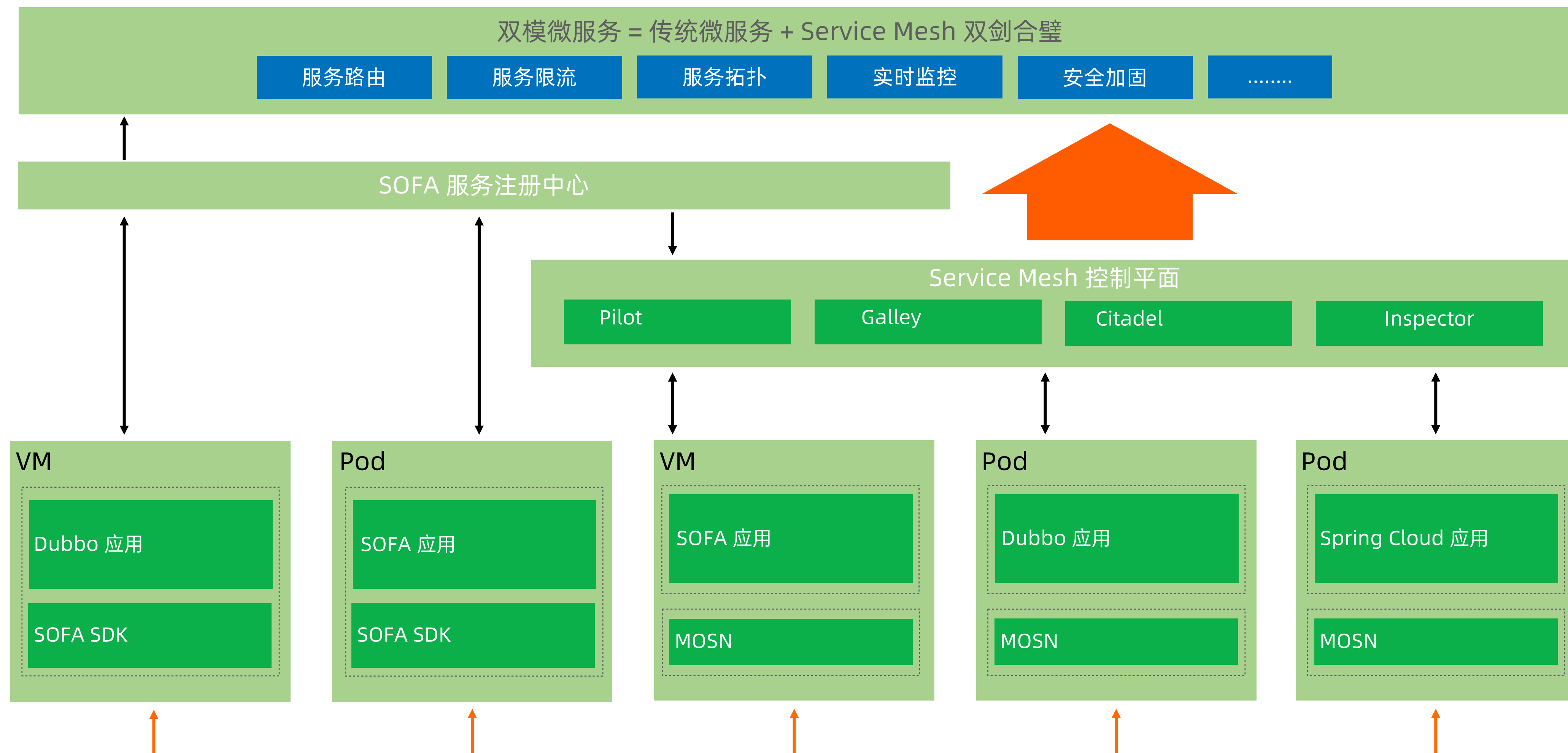
无人值守变更



17 天全蚂蚁无业务介入、
无故障完成全站升级

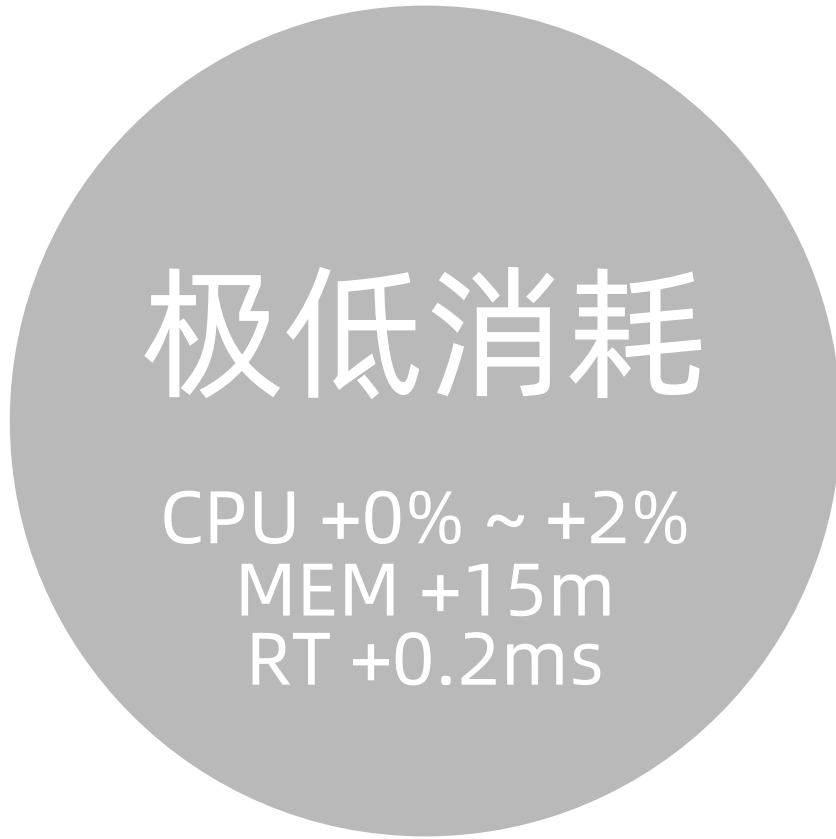
变更日	预发	灰度	生产分批 1	生产分批 2	生产分批 3	生产分批 4	生产分批 5	生产分批 6	生产分批 7	生产分批 8
WEEK1-D1	100%应用	10%应用								
WEEK1-D2	100%	100%应用								
WEEK1-D3	100%	100%	1%应用	1%应用						
WEEK1-D4	100%	100%	10%应用	10%应用						
WEEK2-D5	100%	100%	25%应用	25%应用						
WEEK2-D6	100%	100%	50%应用	50%应用						
WEEK2-D7	100%	100%	100%应用	100%应用						
WEEK2-D8	100%	100%	100%	100%	1%应用	1%应用	1%应用			
WEEK3-D9	100%	100%	100%	100%	10%应用	10%应用	10%应用			
WEEK3-D10	100%	100%	100%	100%	25%应用	25%应用	25%应用			
WEEK3-D11	100%	100%	100%	100%	50%应用	50%应用	50%应用			
WEEK3-D12	100%	100%	100%	100%	100%应用	100%应用	100%应用			
WEEK4-D13	100%	100%	100%	100%	100%	100%	100%	1%应用	1%应用	1%应用
WEEK4-D14	100%	100%	100%	100%	100%	100%	100%	10%应用	10%应用	10%应用
WEEK4-D15	100%	100%	100%	100%	100%	100%	100%	25%应用	25%应用	25%应用
WEEK4-D16	100%	100%	100%	100%	100%	100%	100%	50%应用	50%应用	50%应用
WEEK4-D17	100%	100%	100%	100%	100%	100%	100%	100%应用	100%应用	100%应用

整体架构

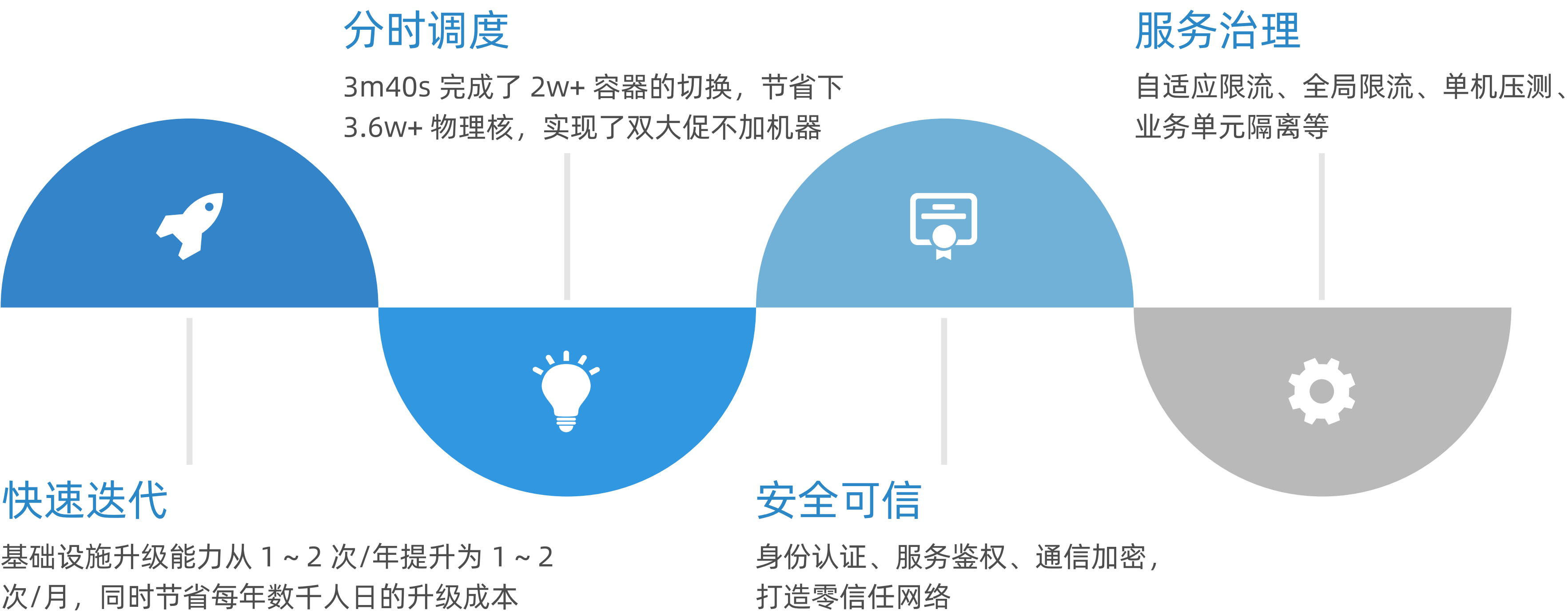


基于 SDK 的传统微服务 ↔ 互联互通，平滑迁移，灵活演进 ↔ 基于 Sidecar 的 Service Mesh 微服务

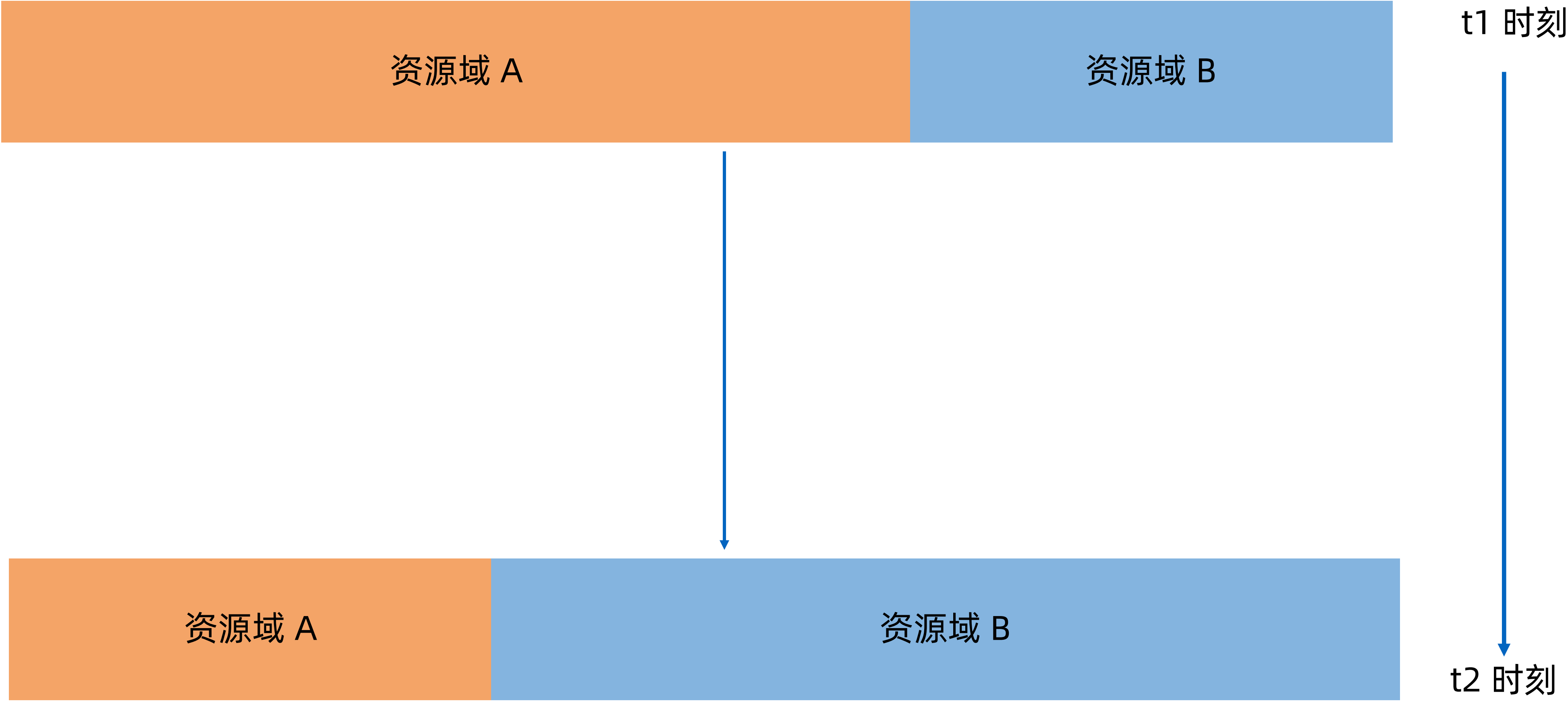
落地规模



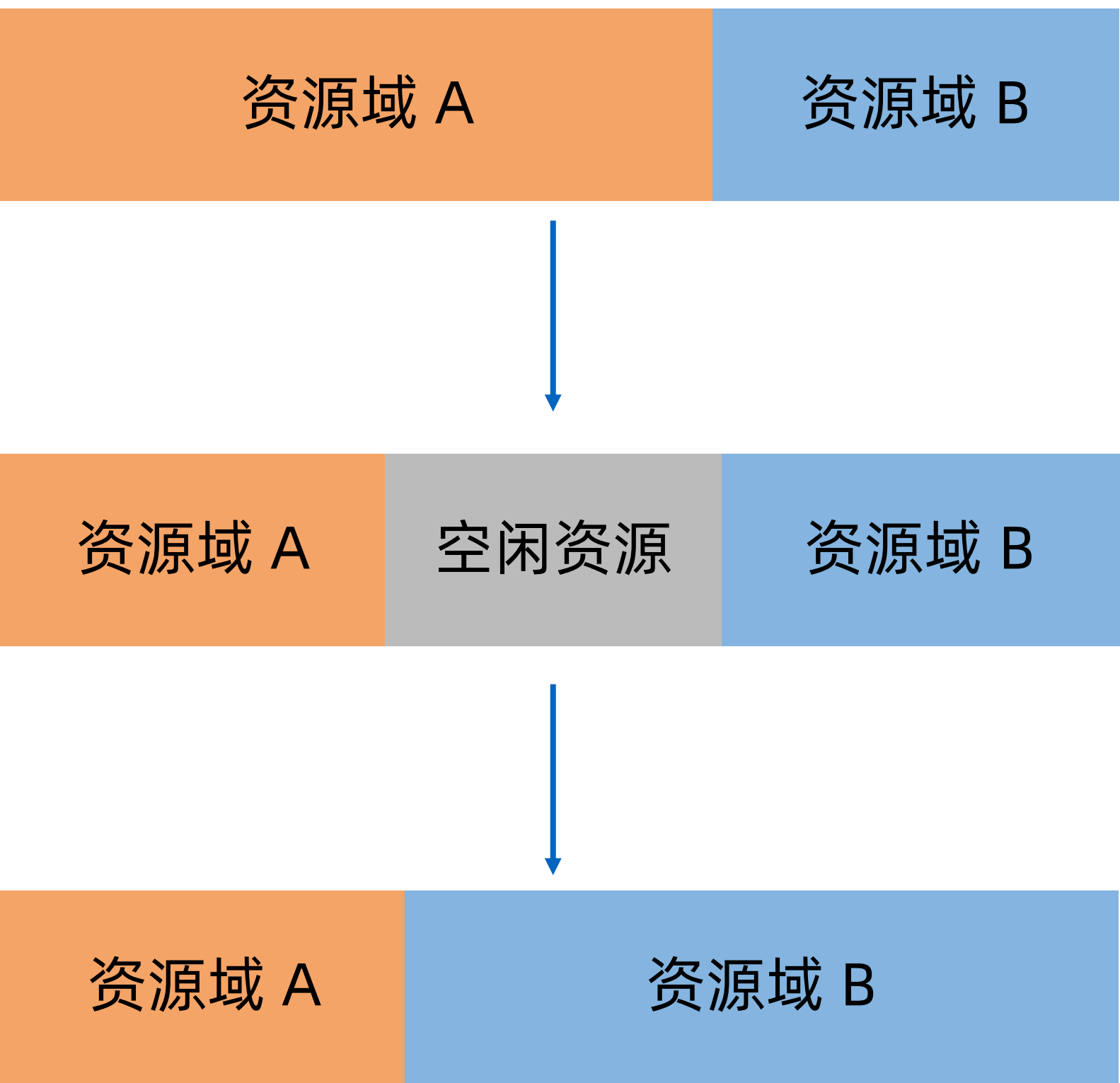
业务价值



分时调度



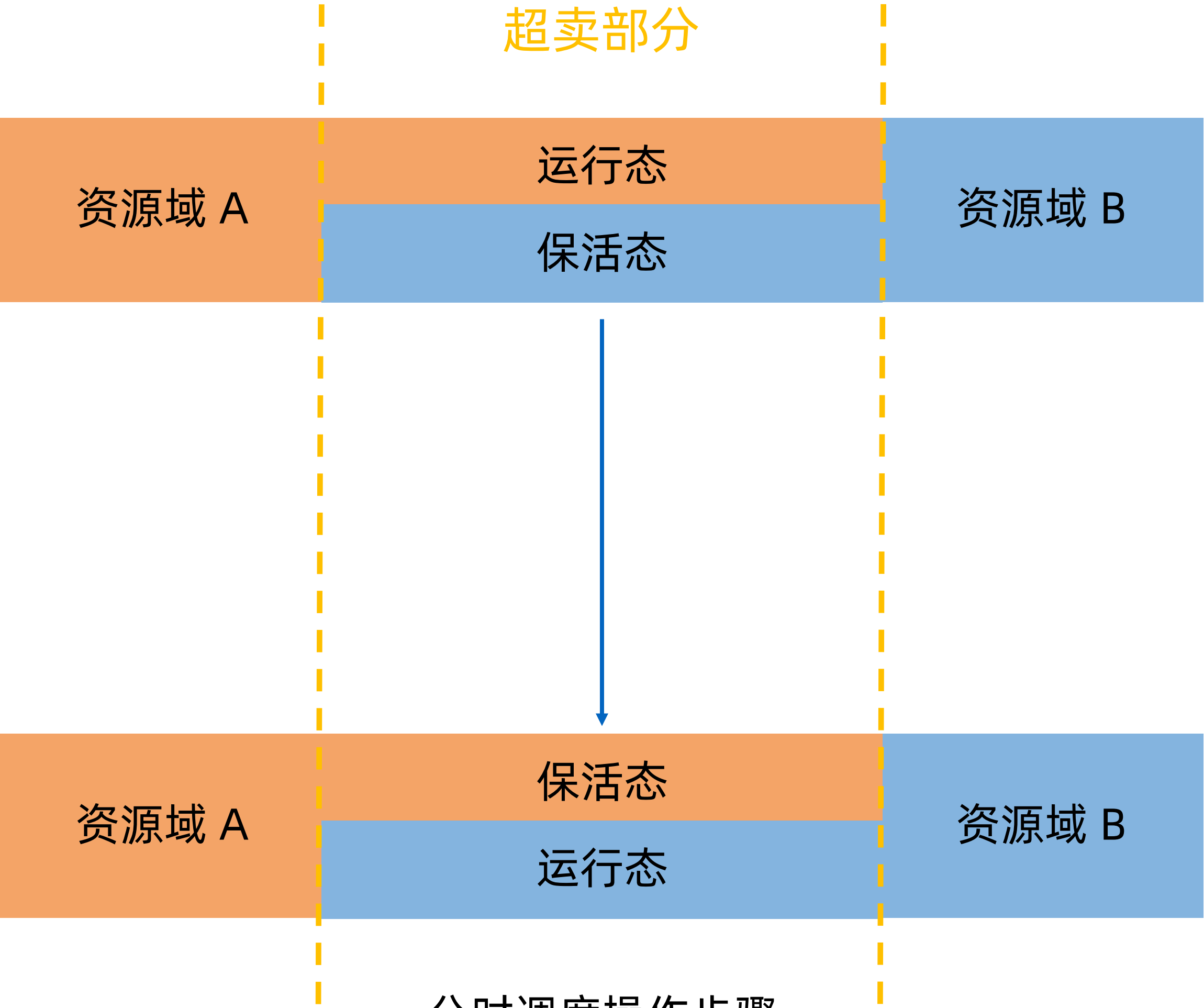
分时调度



常规方案操作步骤

t1 时刻

t2 时刻



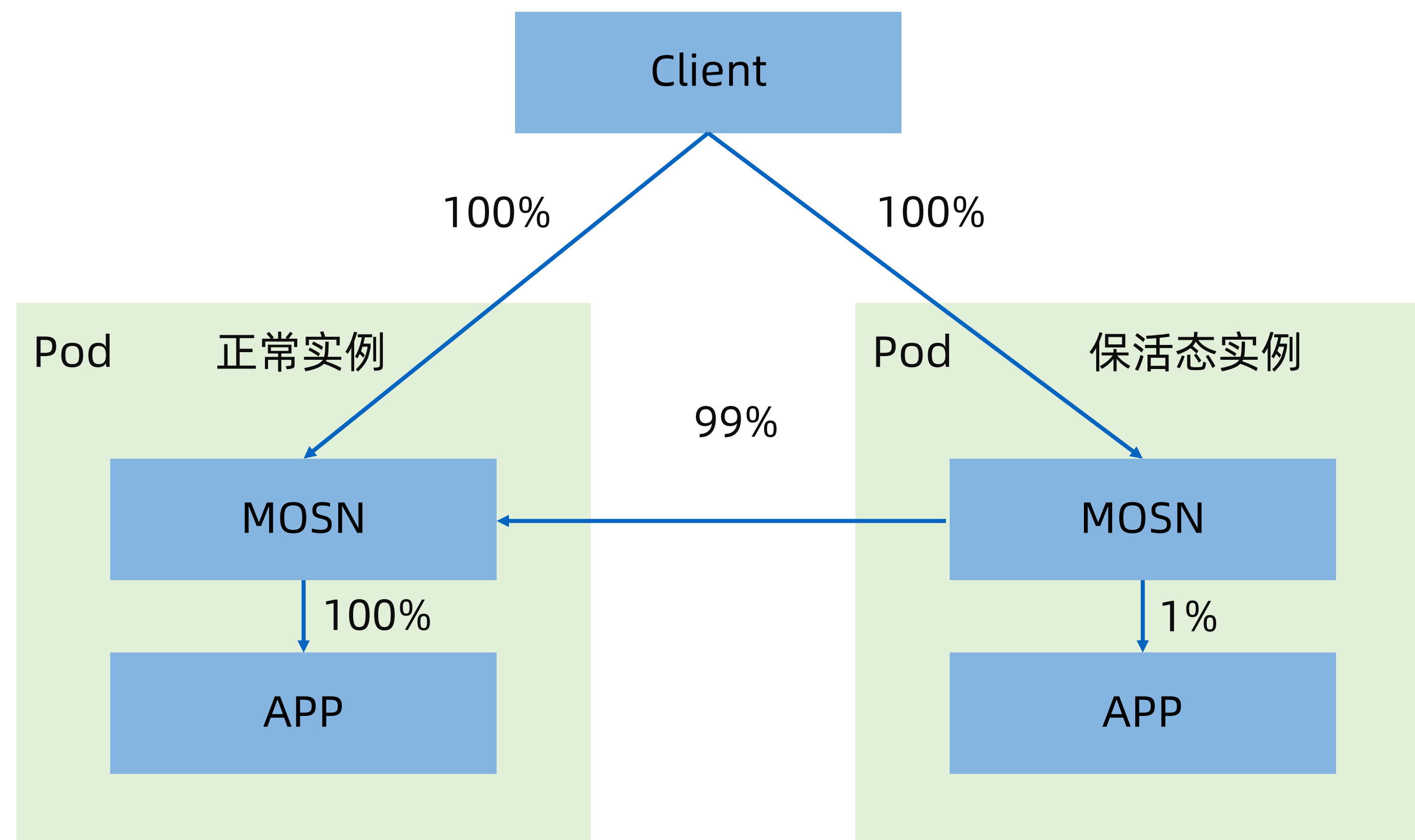
分时调度操作步骤

分时调度

💡 Mosn 转发保活态应用流量，降低资源占用

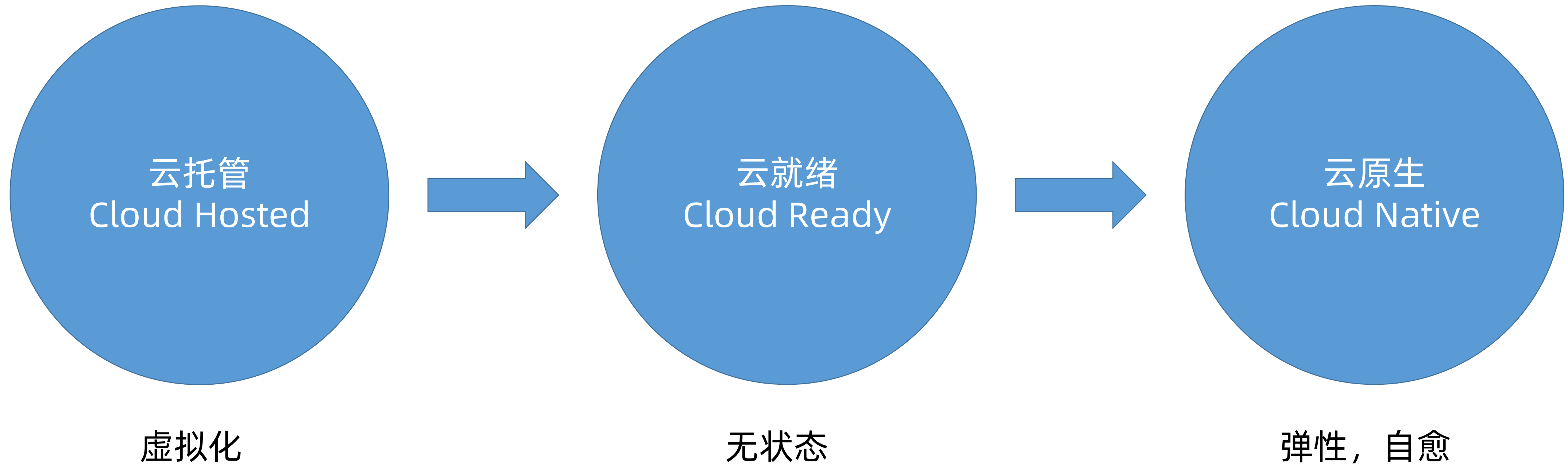
💓 保活意义

1. 维持应用状态：各种长连接、缓存
2. 无需预热即可快速恢复



三、展望未来

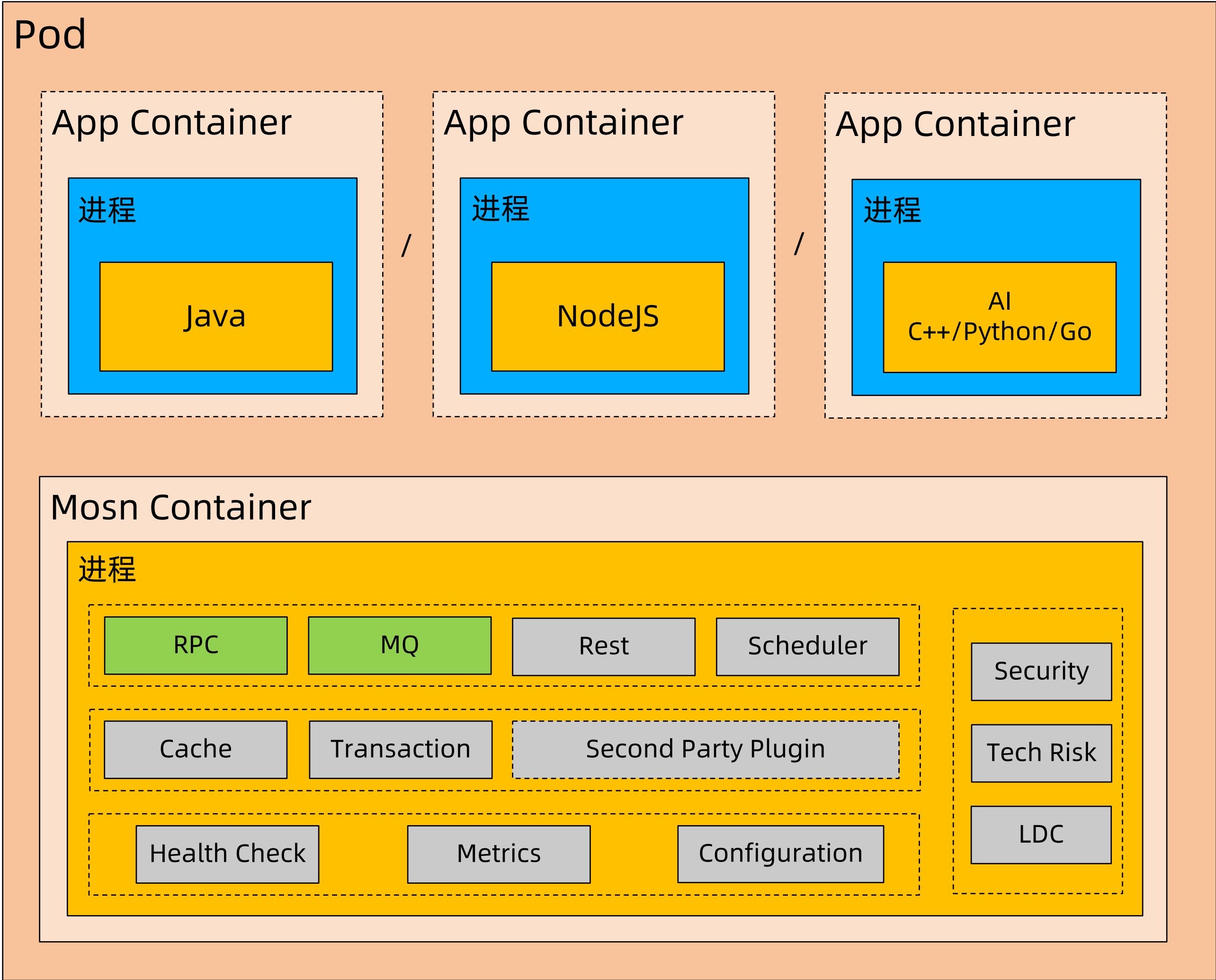
What's Next?



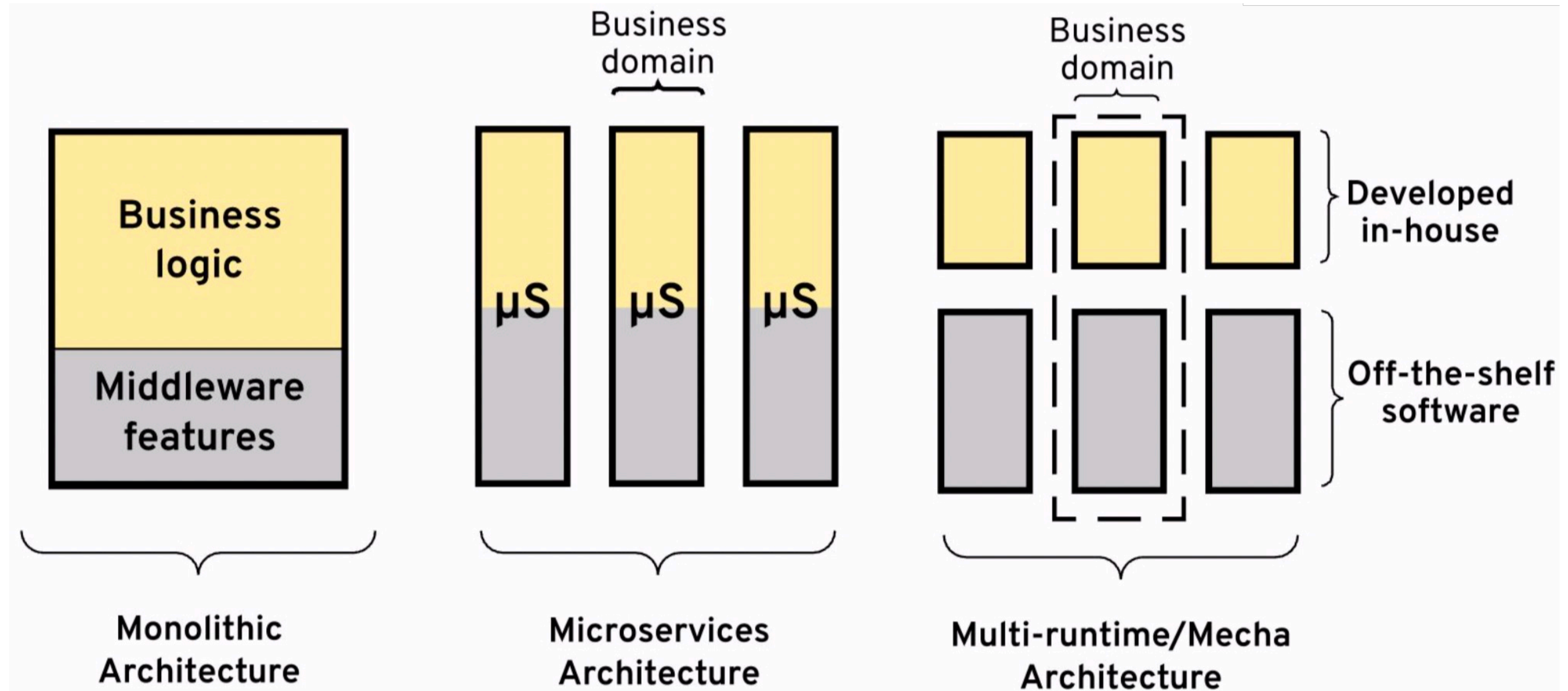
- 提升开发/运维**效率**, 降低资源**成本**
- 云原生不是目的, 而是手段

Service Mesh -> Mesh

 流量代理 -> 中间件运行时



Microservice -> Micrologic + Sidecar



图片来源: <https://www.infoq.com/articles/multi-runtime-microservice-architecture/>

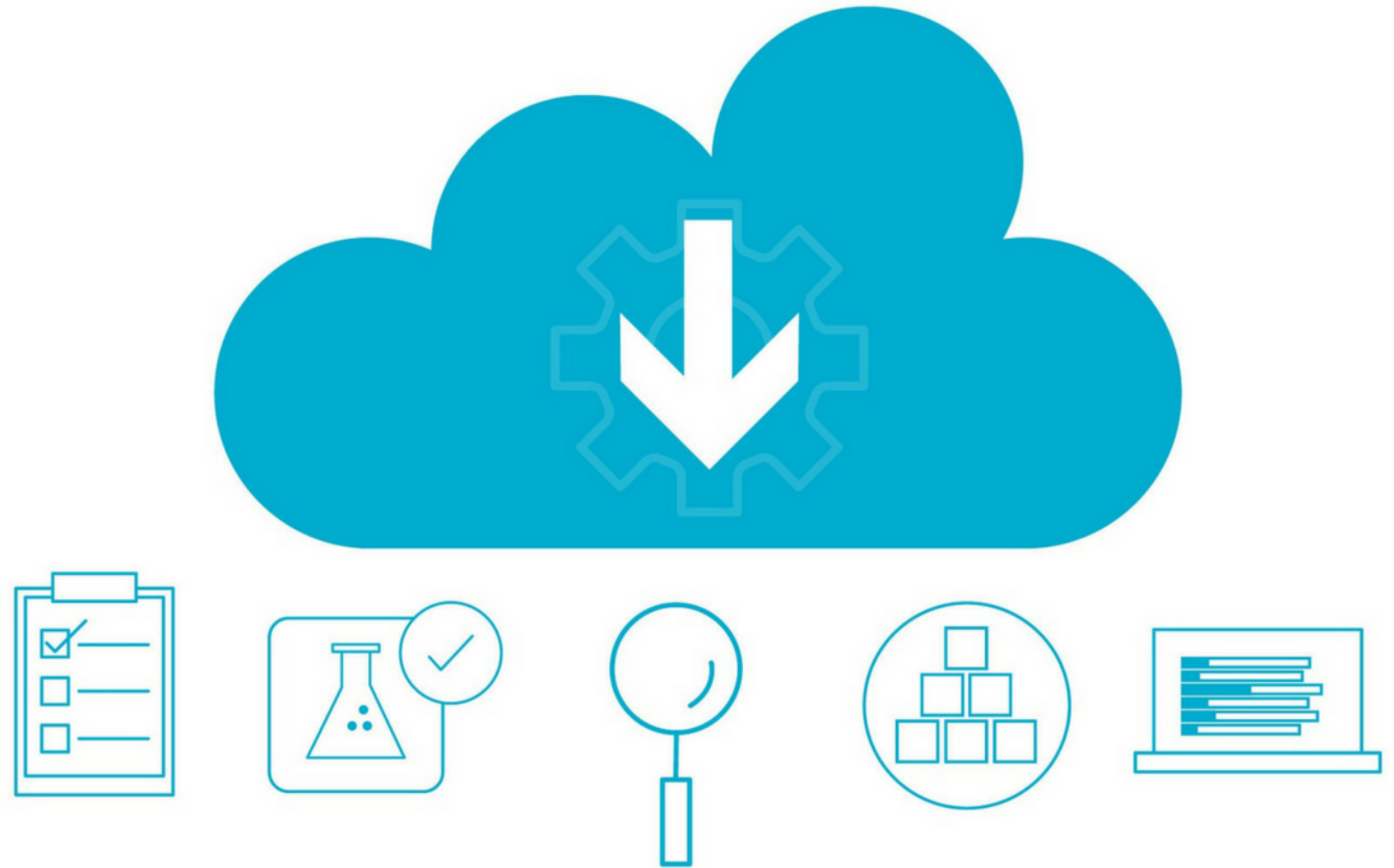
Serverless is more than FaaS

☑ Serverless

- 弹性、免运维、按需使用
- 对普通业务应用价值更大

🔗 Microservice -> Micrologic + Sidecar

- 应用体积更小
- 启动速度更快
- 基础设施可以做更多优化



图片来源: <https://vedarth-sharma.medium.com/going-serverless-bf86e380e51f>

总 结

Summary

- 一、为什么需要 Service Mesh?
- 二、蚂蚁 SOFAShark Mesh 的大规模落地实践
- 三、展望未来

THANKS

