诗和远方

蚂蚁金服 Service Mesh 深度实践

敖小剑 @ 蚂蚁金服

蚂蚁金服
ANT FINANCIAL

前言

- 2017年：“Servicemesh: 下一代微服务”，布道Servicemesh技术
- 2018年：“长路漫漫踏歌而行：蚂蚁金服Service Mesh实践探索”，介绍蚂蚁金服在ServiceMesh领域的探索性实践

今天，有幸第三次来到QCon，给大家带来的依然是蚂蚁金服在Servicemesh领域的实践分享。和去年不同的是，今年蚂蚁金服进入了Servicemesh落地的深水区，规模巨大，而且即将迎来双十一大促考验。

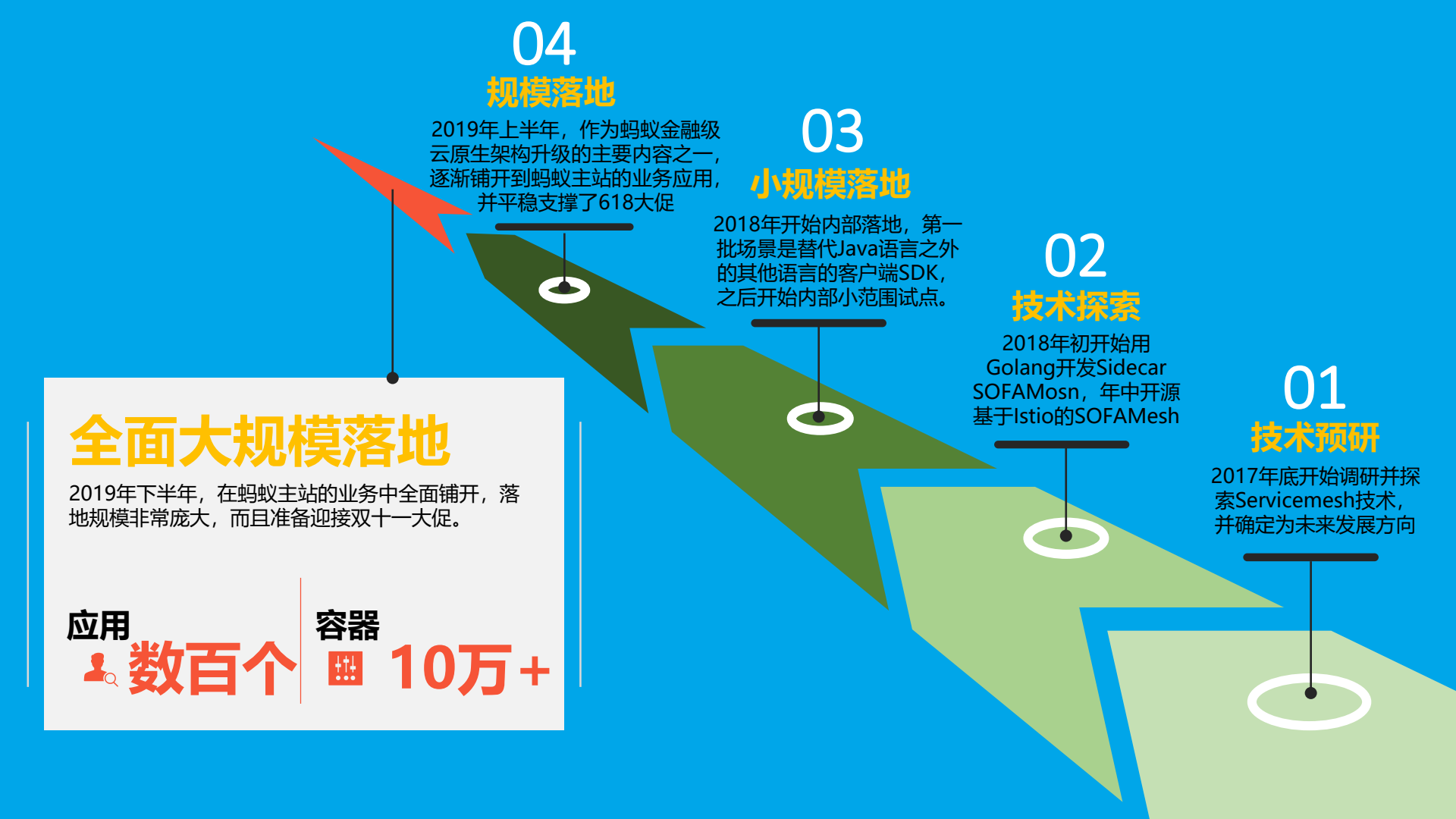
内容

1. 蚂蚁金服落地情况介绍（双十一）
2. 大规模落地的困难和挑战
3. 是否采用ServiceMesh的建议

蚂蚁金服 ANT FINANCIAL

1 蚂蚁金服落地情况介绍（双十一）

蚂蚁金服
ANT FINANCIAL


04
规模落地
2019年上半年，作为蚂蚁金融级云原生架构升级的主要内容之一，逐渐铺开到蚂蚁主站的业务应用，并平稳支撑了618大促
03
小规模落地
2018年开始内部落地，第一批场景是替代Java语言之外的其他语言的客户端SDK，之后开始内部小范围试点。
02
技术探索
2018年初开始用Golang开发Sidecar SOFAMosn，年中开源基于Istio的SOFAMesh
01
技术预研
2017年底开始调研并探索Servicemesh技术，并确定为未来发展方向
全面大规模落地
2019年下半年，在蚂蚁主站的业务中全面铺开，落地规模非常庞大，而且准备迎接双十一大促。
应用
数百个
容器
10万+

蚂蚁金服
ANT FINANCIAL

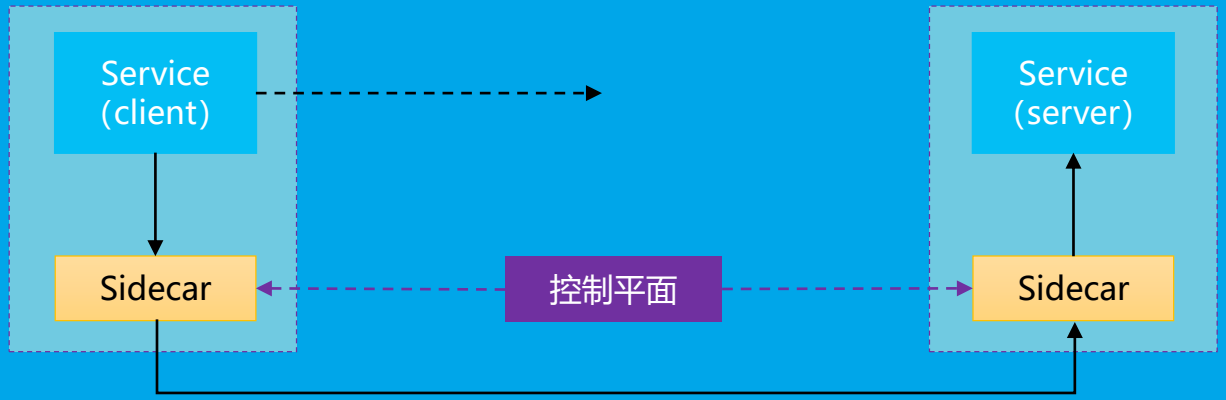
支付宝
ALIPAY

- 多语言支持
- *应用无感知的升级*
- 流量控制
- RPC协议支持
- 可观测性

主要落地场景

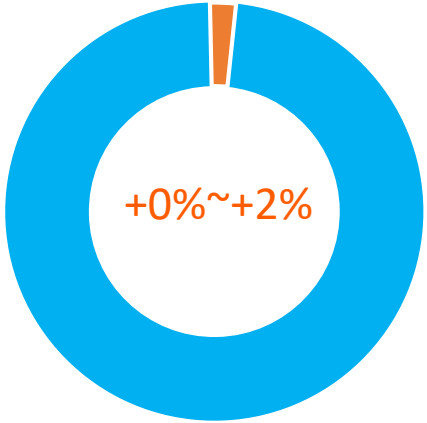
调查：大家最关心、最想看到的是什么？

性能



蚂蚁金服
ANT FINANCIAL

大促压测数据：对比带MOSN和不带MOSN



CPU

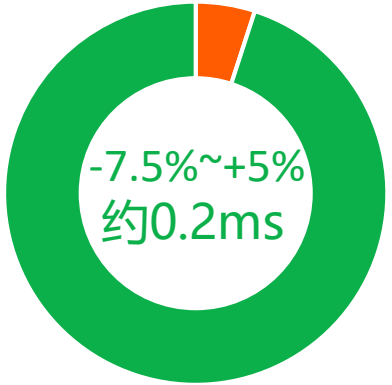
CPU在峰值情况下增加8%，均值约2%。

最新的一次压测，CPU已经基本持平。

内存

带 Mosn 的节点比不带 Mosn 的节点内存平均多15M

是不是感觉不科学?



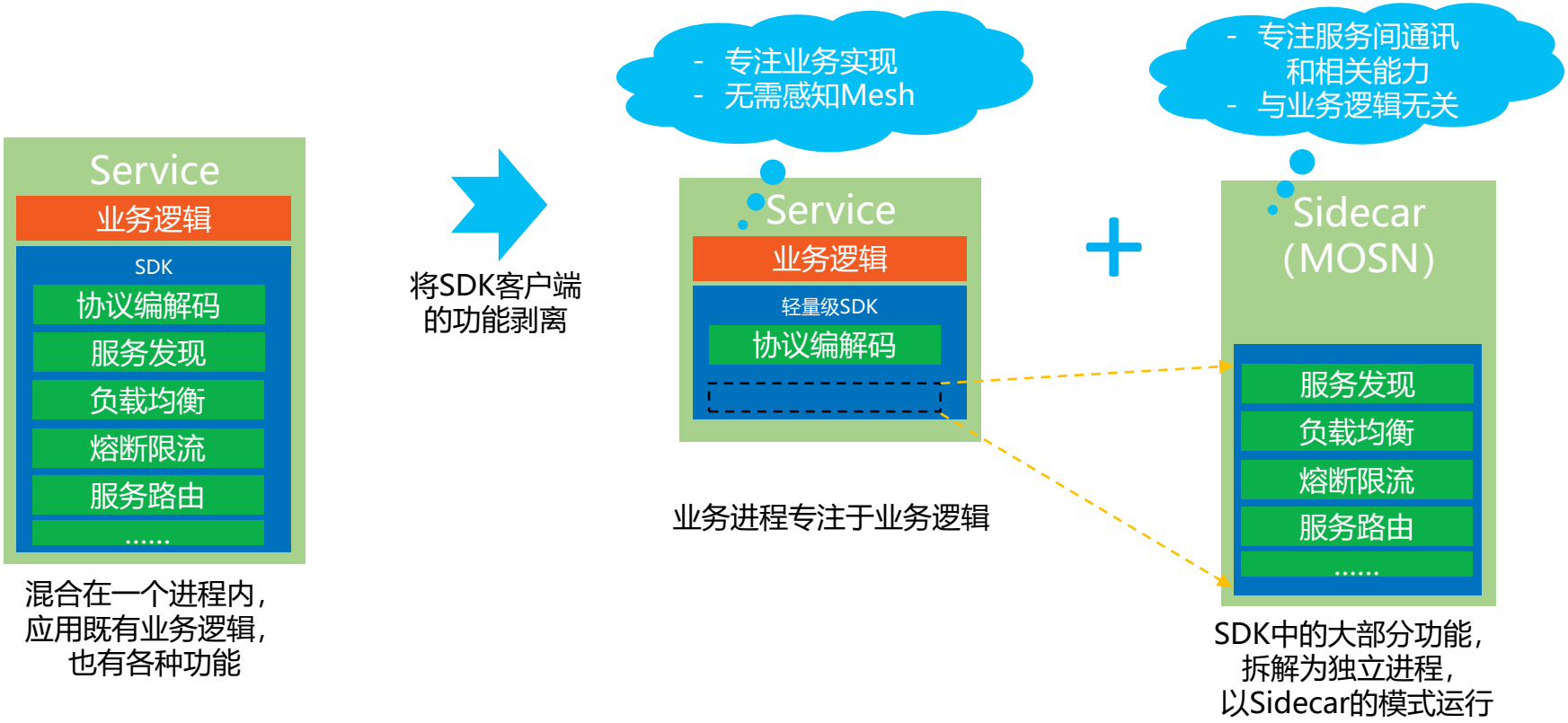
延迟

延迟增加约0.2ms。

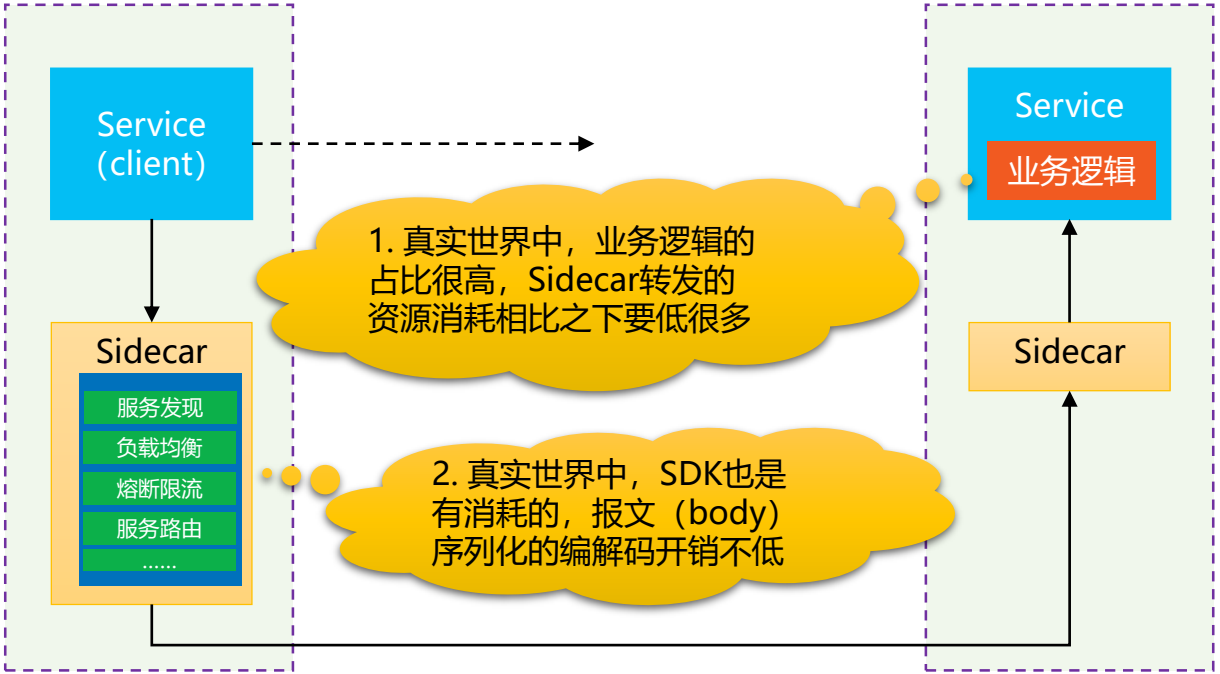
部分场景带MOSN比不带MOSN RT增加约5%。

部分场景带MOSN比不带MOSN RT反而降低7.5%

回顾ServiceMesh的基本思路：关注点分离 + 独立维护



真实世界中的应用和ServiceMesh



推导结果：3倍

直连： 业务逻辑=0，SDK=0，总开销=远程调用*1+0+0=1

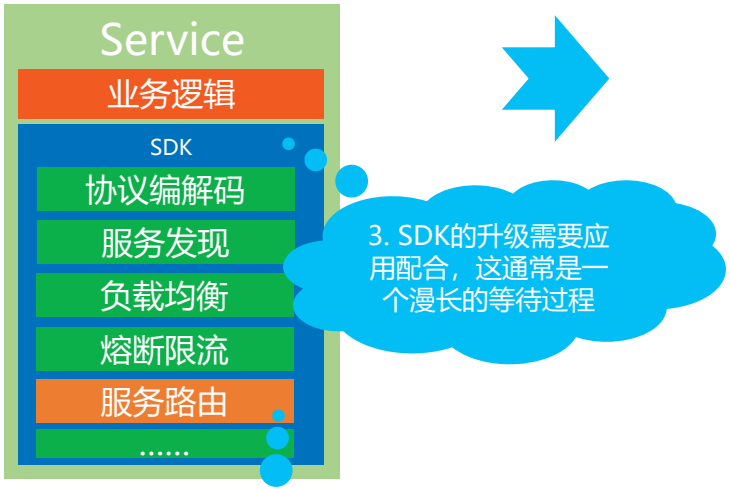
Mesh：业务逻辑=0，SDK=0，总开销=远程调用*3+0+0=3

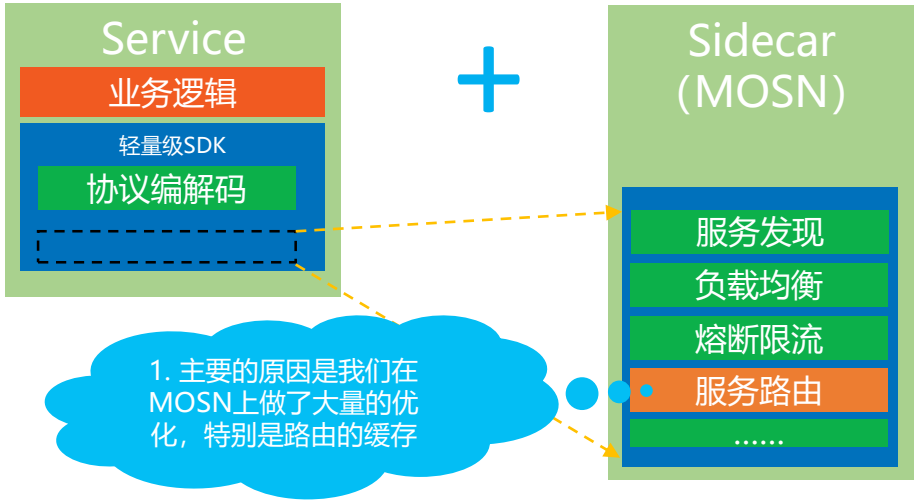
推导结果：+2%

直连： 业务逻辑=100，SDK=10，总开销=远程调用*1+100+10=111

Mesh：业务逻辑=100，SDK=10，总开销=远程调用*3+100+10=113

意外惊喜：持续不断的优化+无感知升级=快速获得收益





推导结果：-7%

直连： 业务逻辑=100，SDK=10，总开销=远程调用*1+100+10=111

Mesh：业务逻辑=100，SDK=1 ，总开销=远程调用*3+100+ 1=104

SDK下沉到基础设施后具备独立升级能力带来的红利

2 大规模落地的困难和挑战

蚂蚁金服
ANT FINANCIAL

困难和挑战：当Servicemesh遇到蚂蚁金服的规模

当规模达到一定程度时，很多原本很小的问题都会急剧放大

CPU优化

- Golang writev 优化：多个包拼装一次写降低 syscall 调用，我们在 go 1.9 的时候发现 writev 有内存泄露的bug，内部使用的目前是 patch 了 writev bugfix 的 go 1.12。writev bugfix 已经集成在 go 1.13 中。

  详情见我们当时给go提交的PR：https://github.com/golang/go/pull/32138

内存优化

- 内存复用：报文直接解析可能会产生大量临时对象，mosn 通过直接复用报文字节的方式，将必要的信息直接通过 unsafe.Pointer 指向报文的指定位置来避免临时对象的产生。

延迟优化

- 协议升级，快速读取header：TR 协议请求头和 Body 均为 hessian 序列化，性能损耗较大。而 Bolt 协议中 Header 是一个扁平化map，解析性能损耗小。升级应用走 Bolt 协议来提升性能

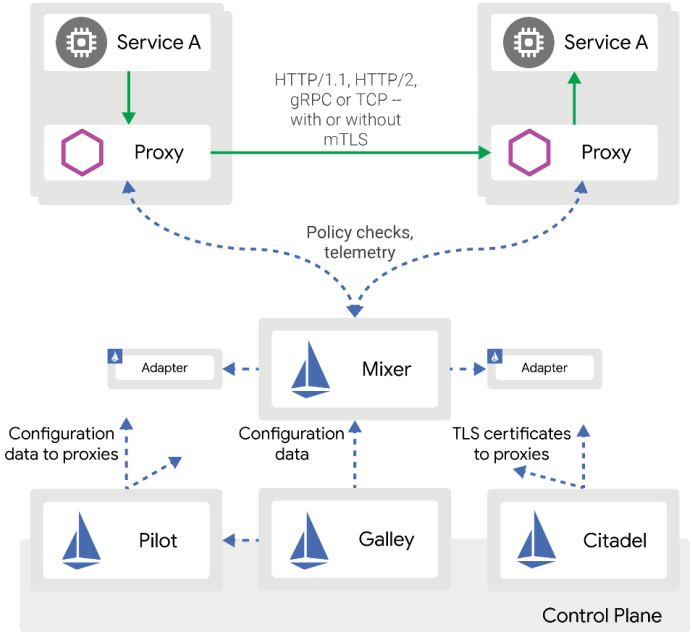
- 路由缓存：内部路由的复杂性（一笔请求经常需要走多种路由策略最终确定路由结果目标），通过对相同条件的路由结果做秒级缓存，我们成功将某核心链路的全链路 RT 降低 7%

性能优化@数据平面：MOSN的各种性能优化

性能优化@Mixer：老生常谈，避无可避

蚂蚁金服 ANT FINANCIAL

Mixer的性能问题，一直都是Istio中最被人诟病的地方。在Istio 1.1/1.2版本之后，引入 Out-Of-Process Adapter之后，更是雪上加霜



Mixer V1
（生命无法承受之重）

对于生产级落地而言，性能难于接受，更不要提大规模落地……

折衷落地方案
（眼前的苟且）

弃用Mixer v1，改为在MOSN中直接实现功能，并只提供最基本的策略检查功能如限流，鉴权等

Mixer V2
（诗和远方）

Mixer合并进Sidecar，引入web assembly进行Adapter扩展，这是Mixer的正确姿势。然而社区望穿秋水，但迟迟不能启动，远水解不了近渴

序列化优化

- xds序列化升级：全面使用types.Any类型，弃用types.Struct类型，序列化性能提升70倍，整体性能提升4倍
- CR序列化缓存，将序列化时机从Get/List操作提前至事件触发时，并缓存结果。大幅降低序列化频率，压测场景下整体性能提升3倍，GC频率大幅下降

预计算优化

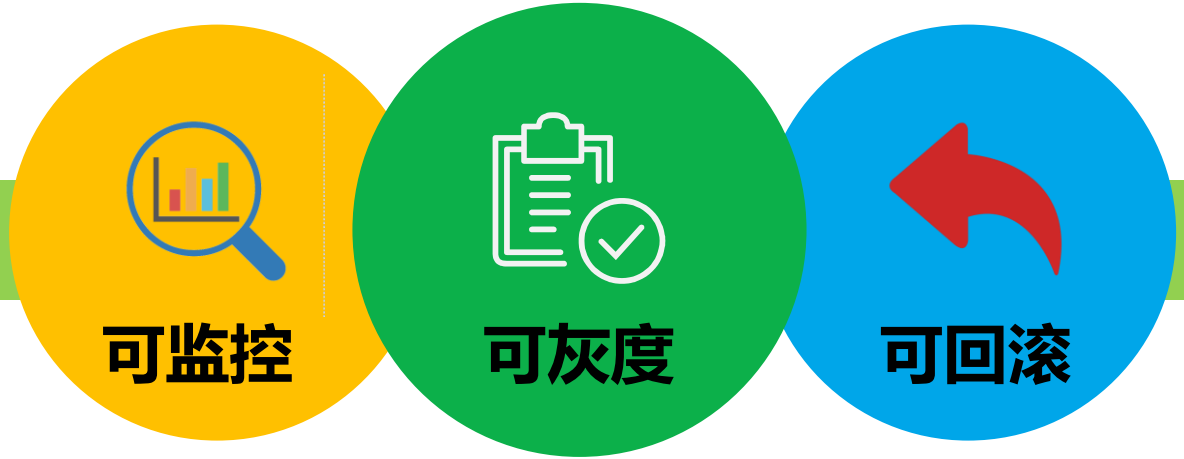
- 支持Sidecar CRD维度的CDS /LDS/RDS 预计算，大幅降低重复计算，压测场景下整体性能提升6倍
- 支持Gateway维度的CDS / LDS / RDS 预计算
- 计算变更事件的影响范围，支持局部推送，减少多余的计算和对无关sidecar的打扰

推送优化

- 支持运行时动态降级，支持熔断阈值调整，限流阈值调整，静默周期调整，日志级别调整
- 实现增量ADS接口，在配置相关处理上，sidecar cpu减少90%，pilot cpu减少42%

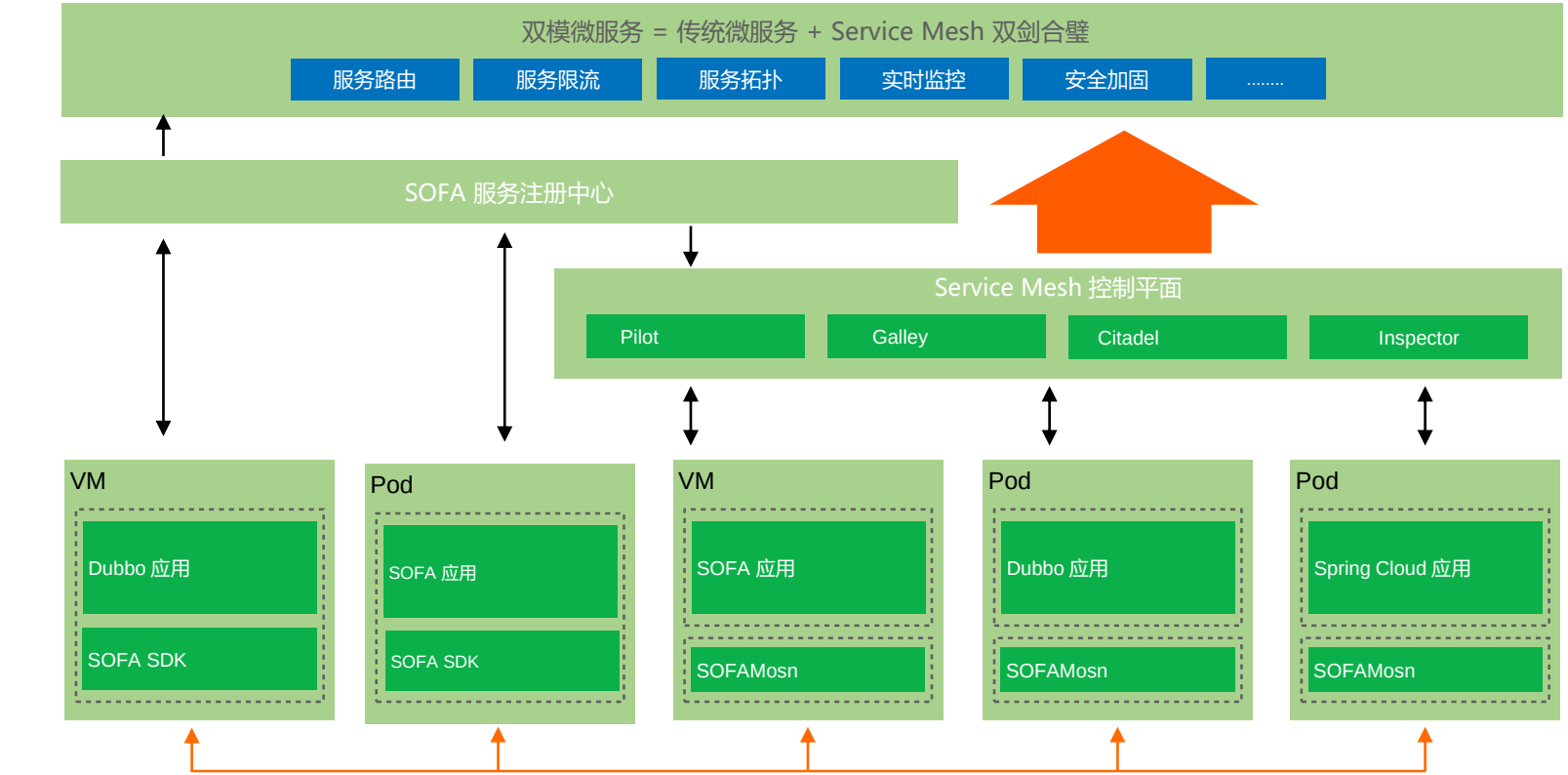
性能优化@Pilot：被Mixer掩盖的重灾区

运维：线上变更三板斧



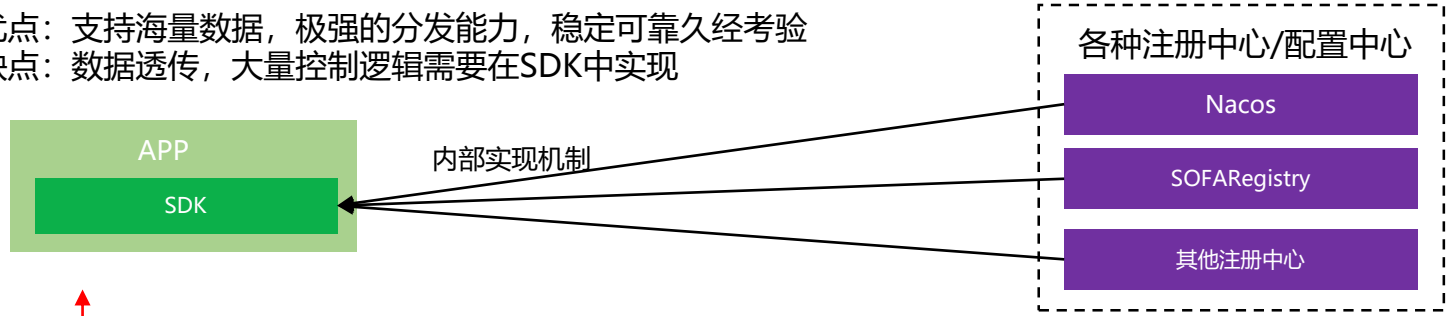
生产无小事，变更需谨慎

应用迁移：SOFA双模微服务平台

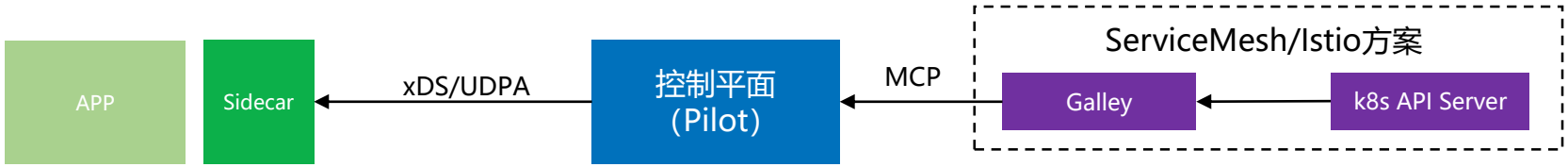


控制平面和传统注册中心/配置中心：各有千秋，如何结合？

优点：支持海量数据，极强的分发能力，稳定可靠久经考验
缺点：数据透传，大量控制逻辑需要在SDK中实现

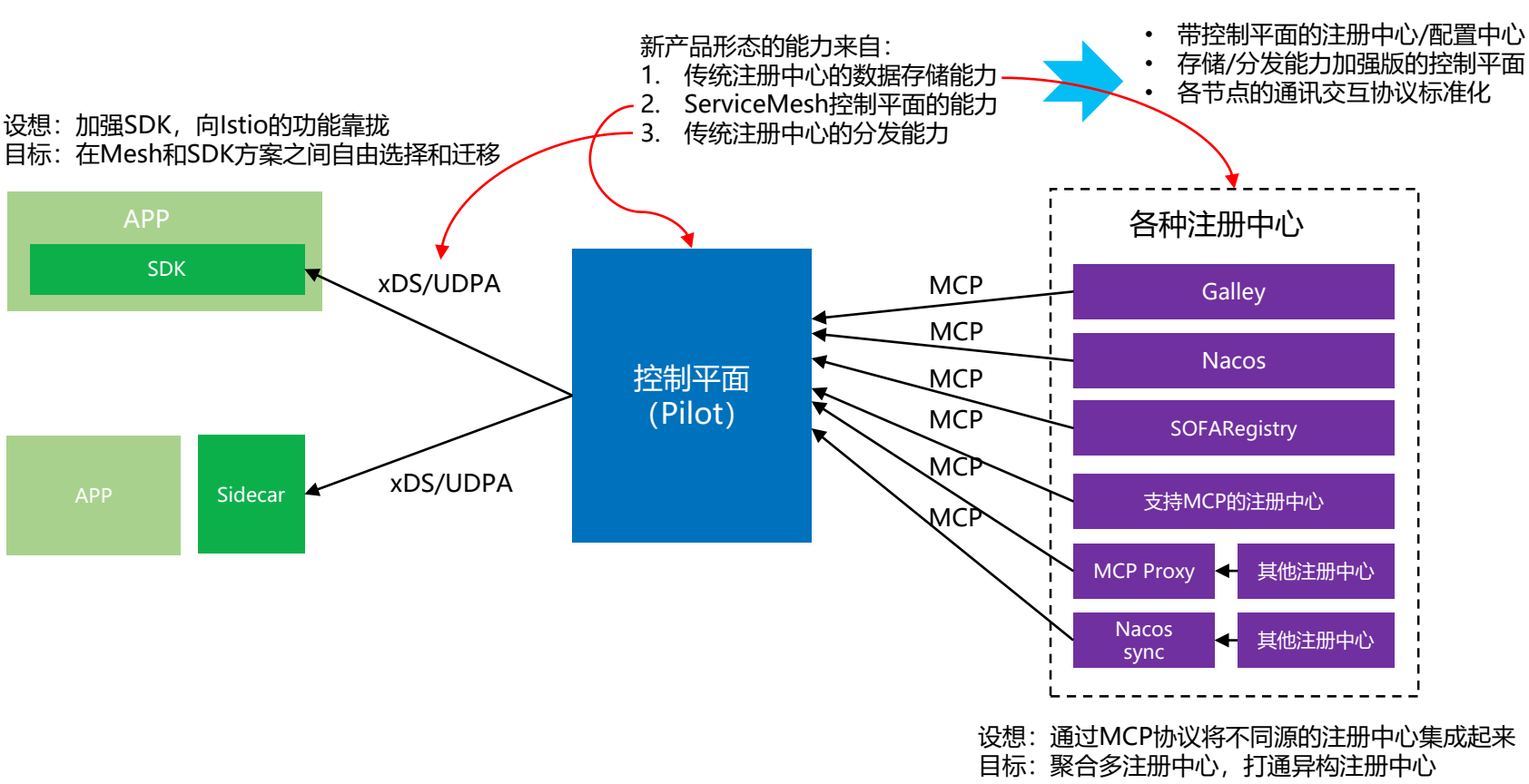


如何打通两个体系是ServiceMesh社区的老大难问题

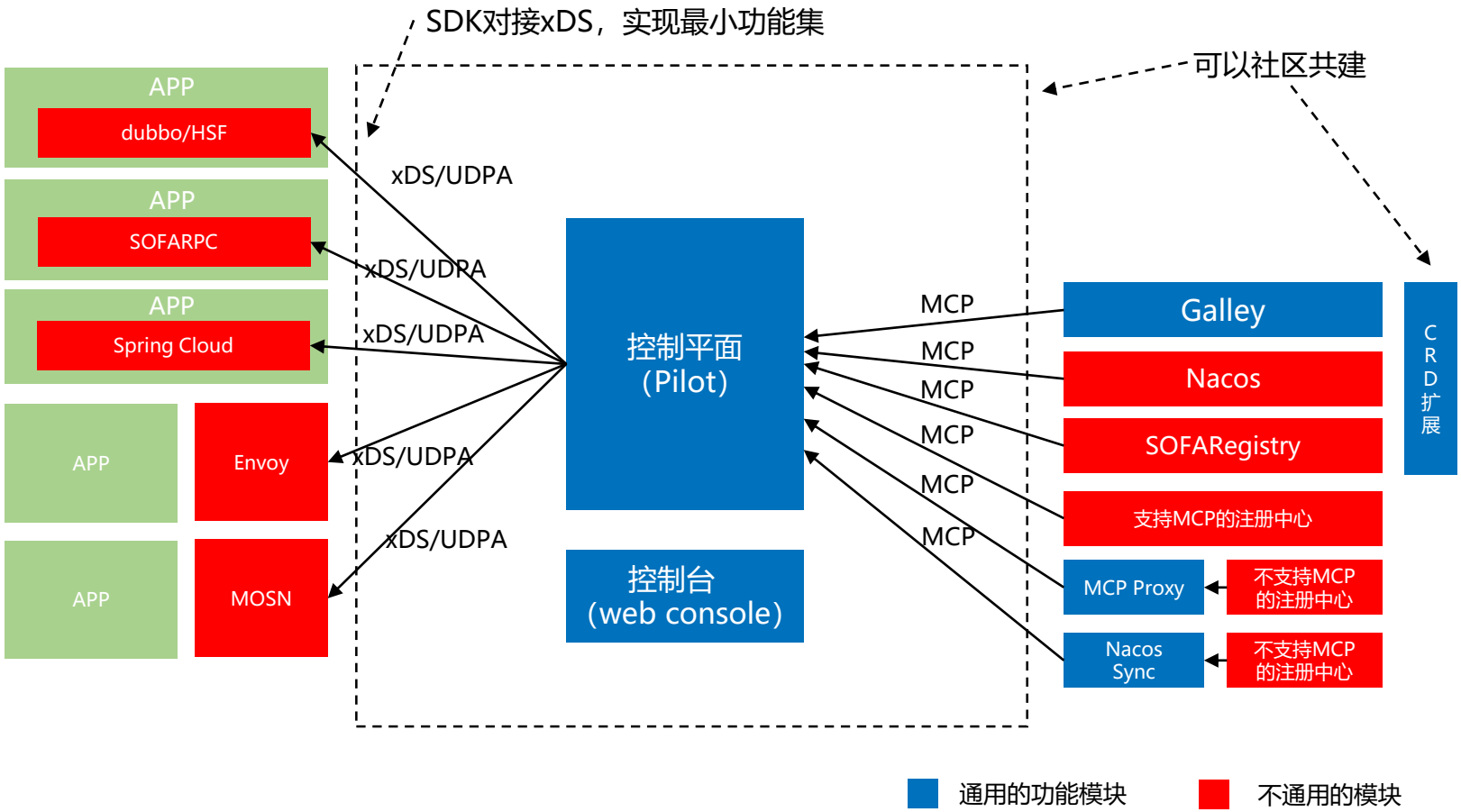


优点：控制平面提供强大的控制逻辑，解藕数据源和下发数据，扩展性好
缺点：支持的容量有限，下发的性能和稳定性相比之下有差距

以MCP和xDS/UDPA为基础，融合控制平面和传统注册中心/配置中心



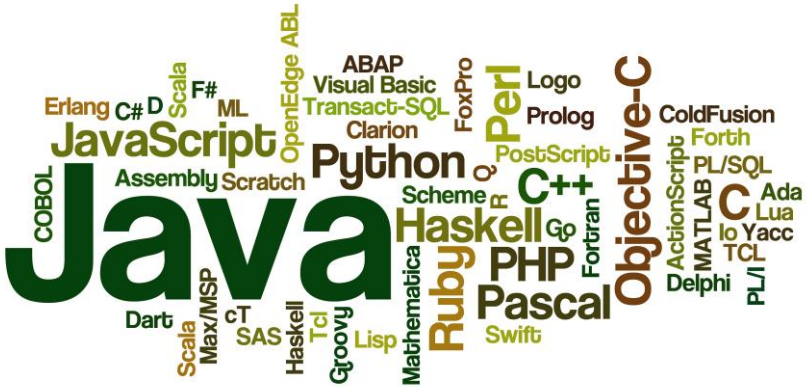
最大限度的整合传统微服务框架和ServiceMesh：求同存异，保持兼容



是否采用ServiceMesh的建议

蚂蚁金服
ANT FINANCIAL

如有下述痛点，可以考虑Servicemesh



多语言支持

类库升级困难

无侵入性：老应用无改动升级改造，堪称神器

蚂蚁金服 ANT FINANCIAL

掌控系统流量

- 精准的流量控制
- 完善的可观测性
- 无侵入

确保系统安全

- 零信任网络
- 身份/加密/访问控制
- 无侵入

观察系统行为

- 监控
- Metrics
- 无侵入

拿什么拯救你：技术栈不统一受制于人的企业

蚂蚁金服 ANT FINANCIAL

传统烟囱架构

- 重复建设，重复造轮子
- 不同时期，不同厂商，用不同的轮子
- 难以维护和演进，后续成本高昂
- 掌控力不足，容易受制于人

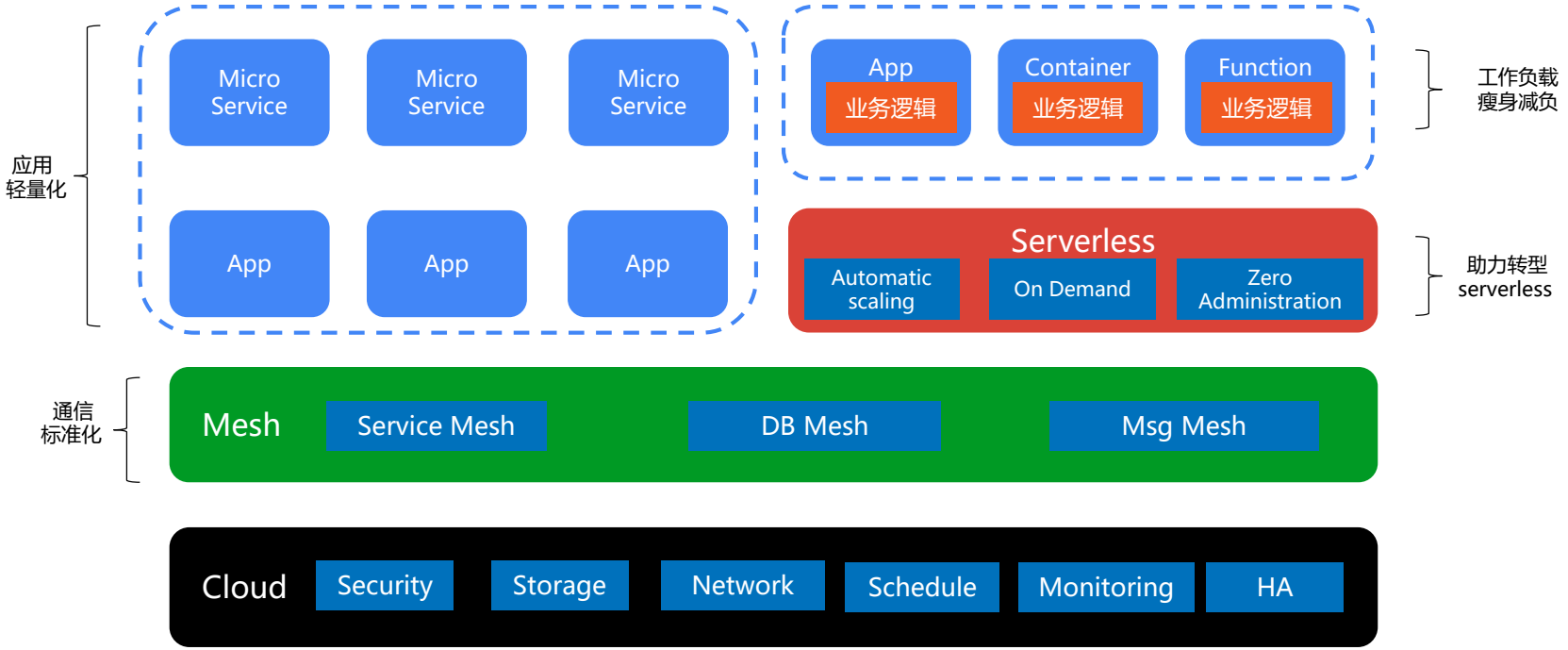
对于技术力量不足、严重依赖外包和采购的企业，尤其是银行/保险/证券类金融企业

将乙方限制在业务逻辑的实现上

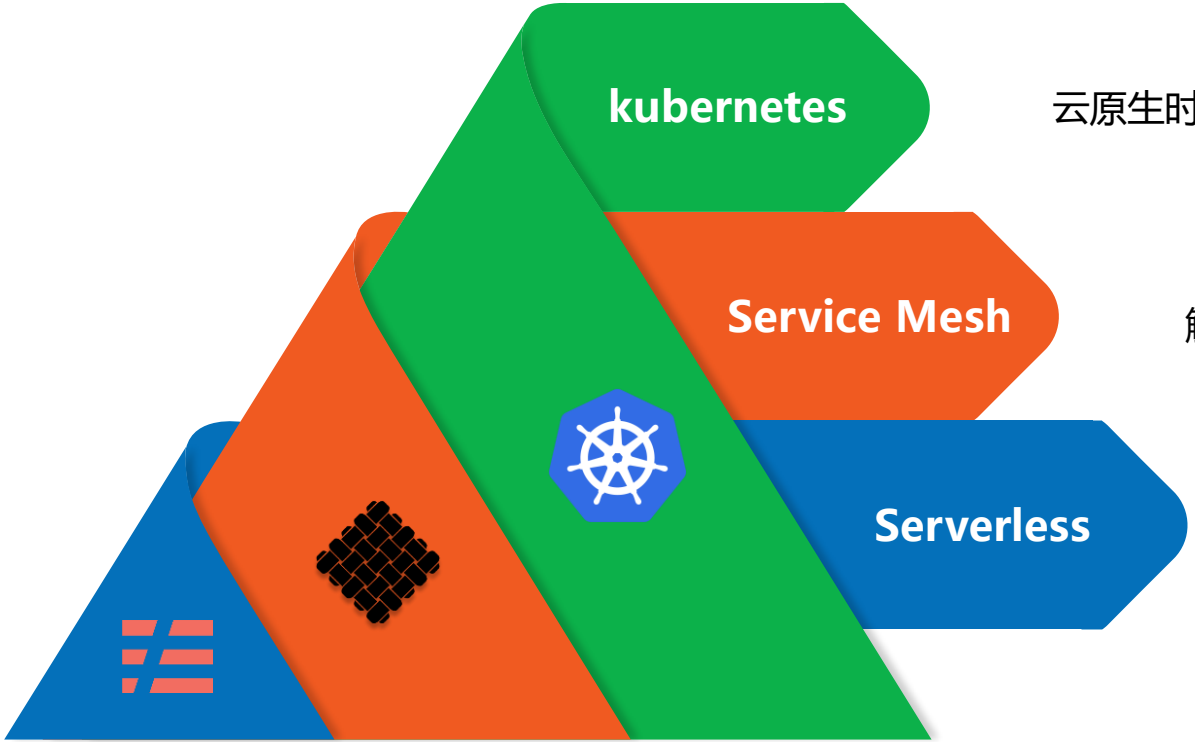
避免乙方公司借项目机会引入：

- 各种技术栈：造成技术栈混乱，后期维护成本超高
- 私有技术栈：造成对甲方事实上的技术绑定（甚至技术绑架）

如果云原生是你的诗和远方



Mesh化是云原生落地的关键步骤



云原生时代的操作系统

解耦应用与服务间通讯

解耦应用与服务器

云原生落地的三驾马车：相辅相成，相得益彰

Service Mesh的核心价值

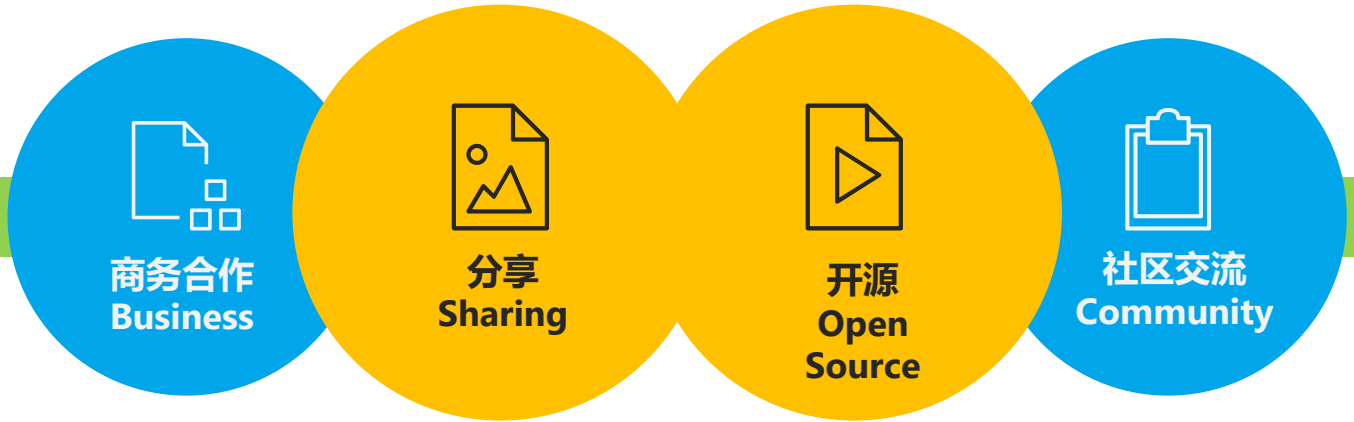
实现业务逻辑和非业务逻辑的分离

- 为下沉到基础设施提供可能
- 帮助应用轻量化，专注业务
- 进而实现应用云原生化

4 总结

静待双十一大考：ServiceMesh的历史时刻，敬请期待

蚂蚁金服 ANT FINANCIAL



蚂蚁金服即将推出ServiceMesh产品，提供商业技术支持

更多的技术分享：落地场景，经验教训，实施方案，架构设计...

将落地实践中的技术实现和方案以不同的方式回馈社区，推动Servicemesh落地实践

ServiceMesher技术社区继续承担国内Servicemesh布道和交流的重任；


我们的ServiceMesh，我们的诗和远方……