



SOFAStack钉钉群

“金融级分布式架构”微信公众号





# Service Mesh 在蚂蚁落地后的探索与实践

李唯（良恩）

SOFARPC Committer



**01**

**能力建设**

**02**

**价值思考**

**03**

**研发效能**

# 1.能力建设

得益于 Service Mesh 将业务和基础设施解耦，在过去一年中我们的基础设施能力得到了飞快的发展。大量能力大规模落地蚂蚁。例如：链路加密，可信身份认证，服务鉴权，自适应限流，集群限流，精细化引流，服务自愈等。

# 01

## 能力建设-链路加密

我要发布，你又卡我干嘛？

有什么推荐方案不

我只想听要我改代码不。。  
项目急今天我要上线。



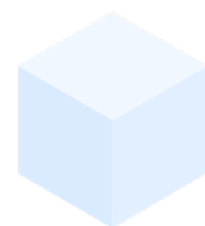
这个应用发起的请求里面有敏感信息，整改后放行



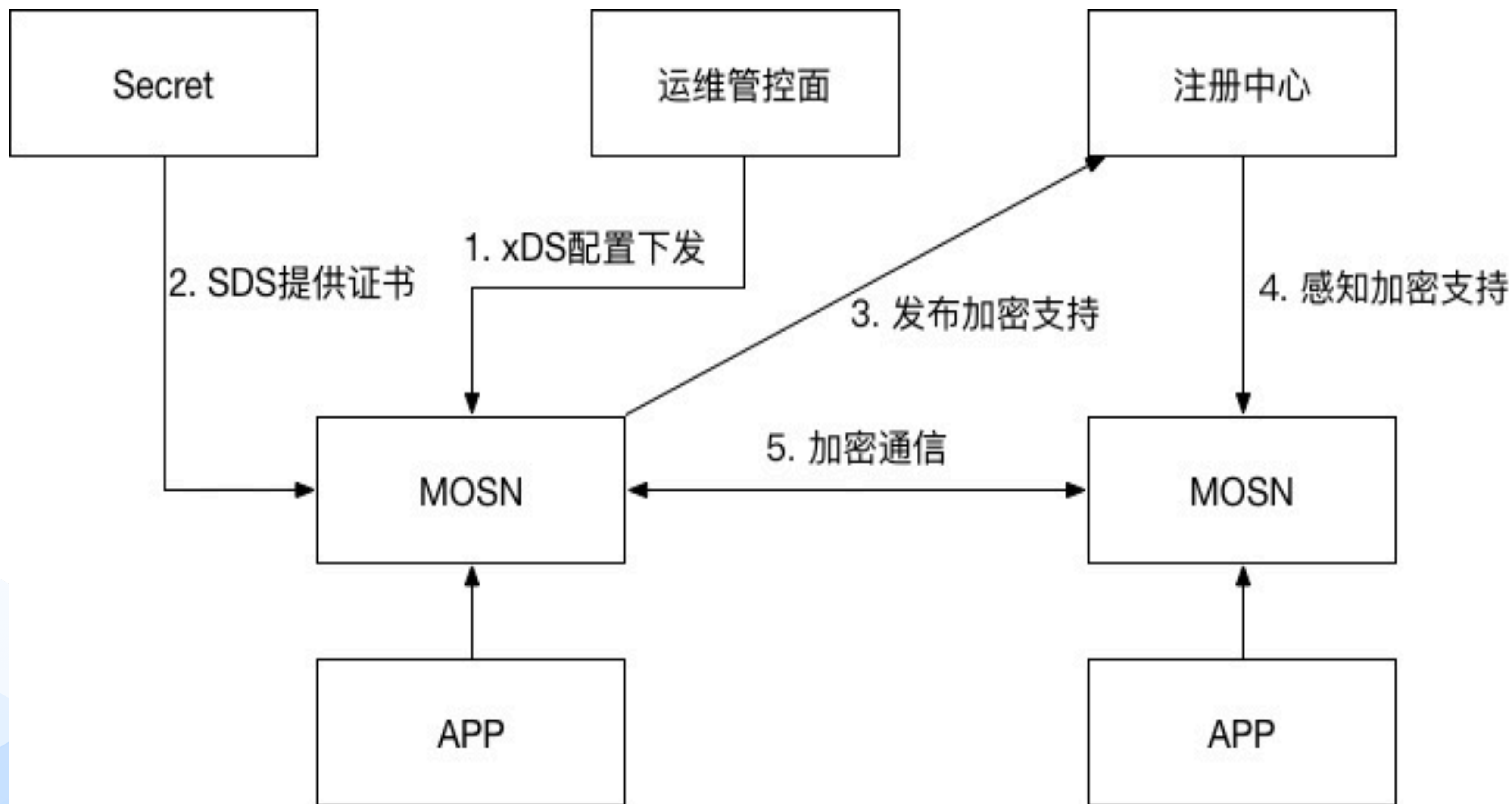
我们有加密SDK，我给你讲一下它的实现原理和强大功能吧，  
你想先听哪一个。

# 01

## 能力建设-链路加密



- 证书与配置下发到服务端
- 通过注册中心发布加密状态
- 客户端发起加密调用



# 01

## 能力建设-自适应限流

昨晚一个活动峰值我的机器  
又被打卦了？

限高了。。。迭代了几个月，  
早扛不住这个量了。

来不及了，我先限个低点的，  
申请加点机器吧。

不是配了QPS限流吗，还是我之前精心压测配的一  
完美兼顾稳定性和资源利用率。

重新压一个准确的限流值吧。

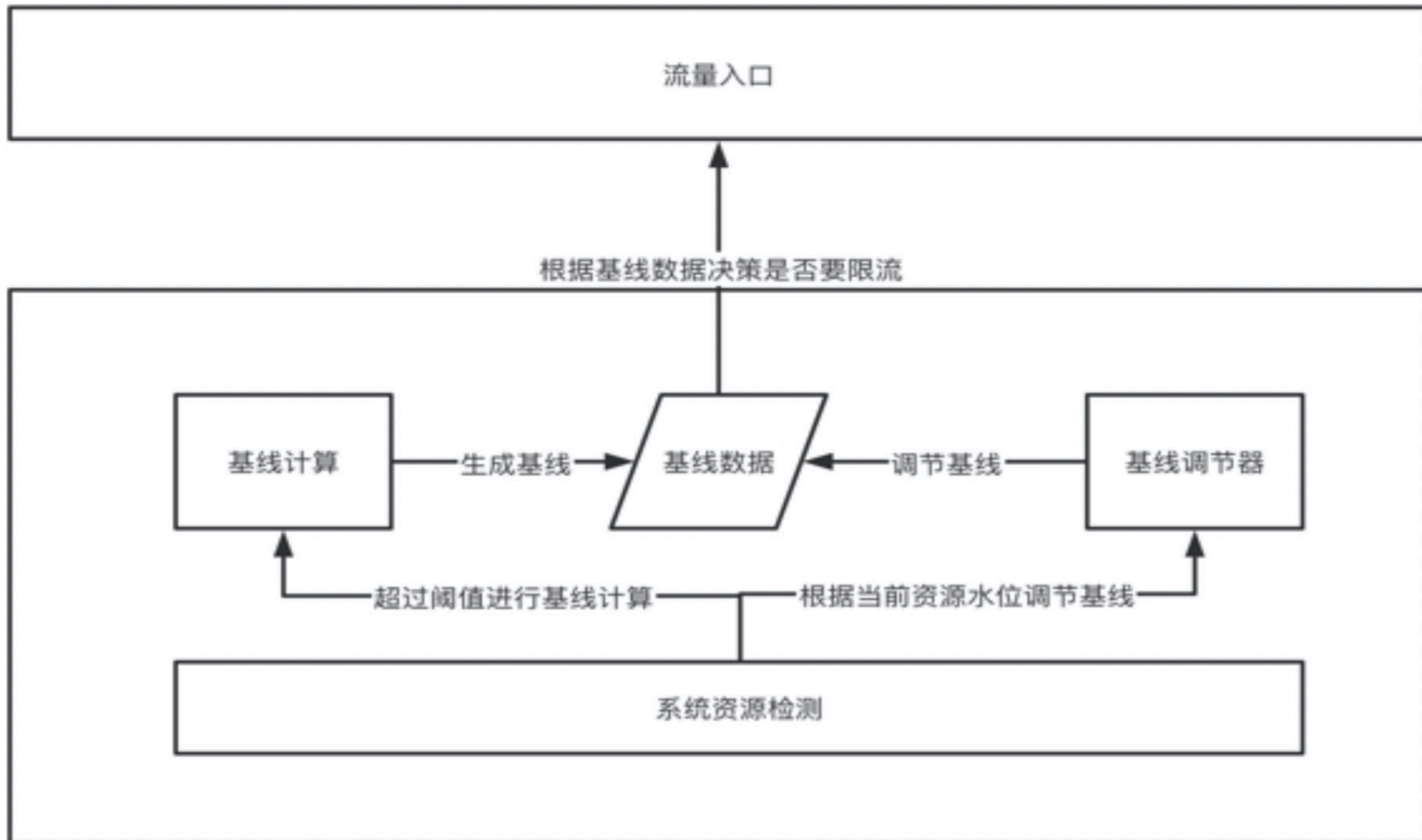
。。。



# 01

## 能力建设-自适应限流

- 系统资源检测：不断检测资源占用是否超过阈值
- 基线计算：找出资源消耗大，上涨明显的接口
- 基线调节器：资源水位 > 阈值，限流基线 ↓  
资源水位 < 阈值，限流基线 ↑
- 根据接口对应的基线决策是否限流





# 01

## 能力建设-引流压测

能不能平时空了就做下压测  
压测，把数据记录下来。

容量评估也有实时依据。

。 。 。

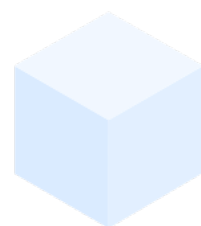
有道理，日常迭代就解决掉性能瓶颈，  
不然临近大促搞不定又加机器

关键是我平时能空吗？

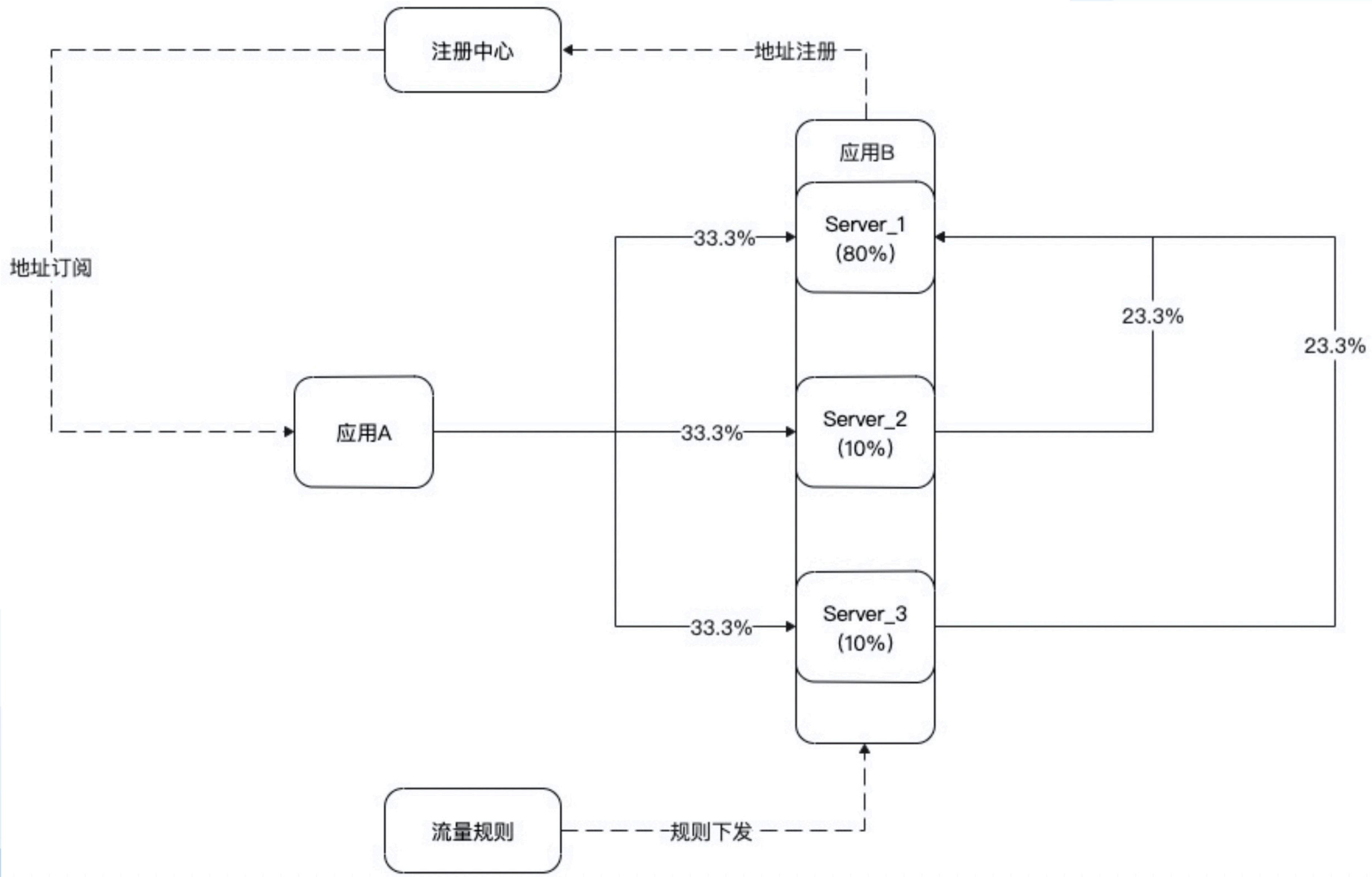


# 01

## 能力建设-引流压测



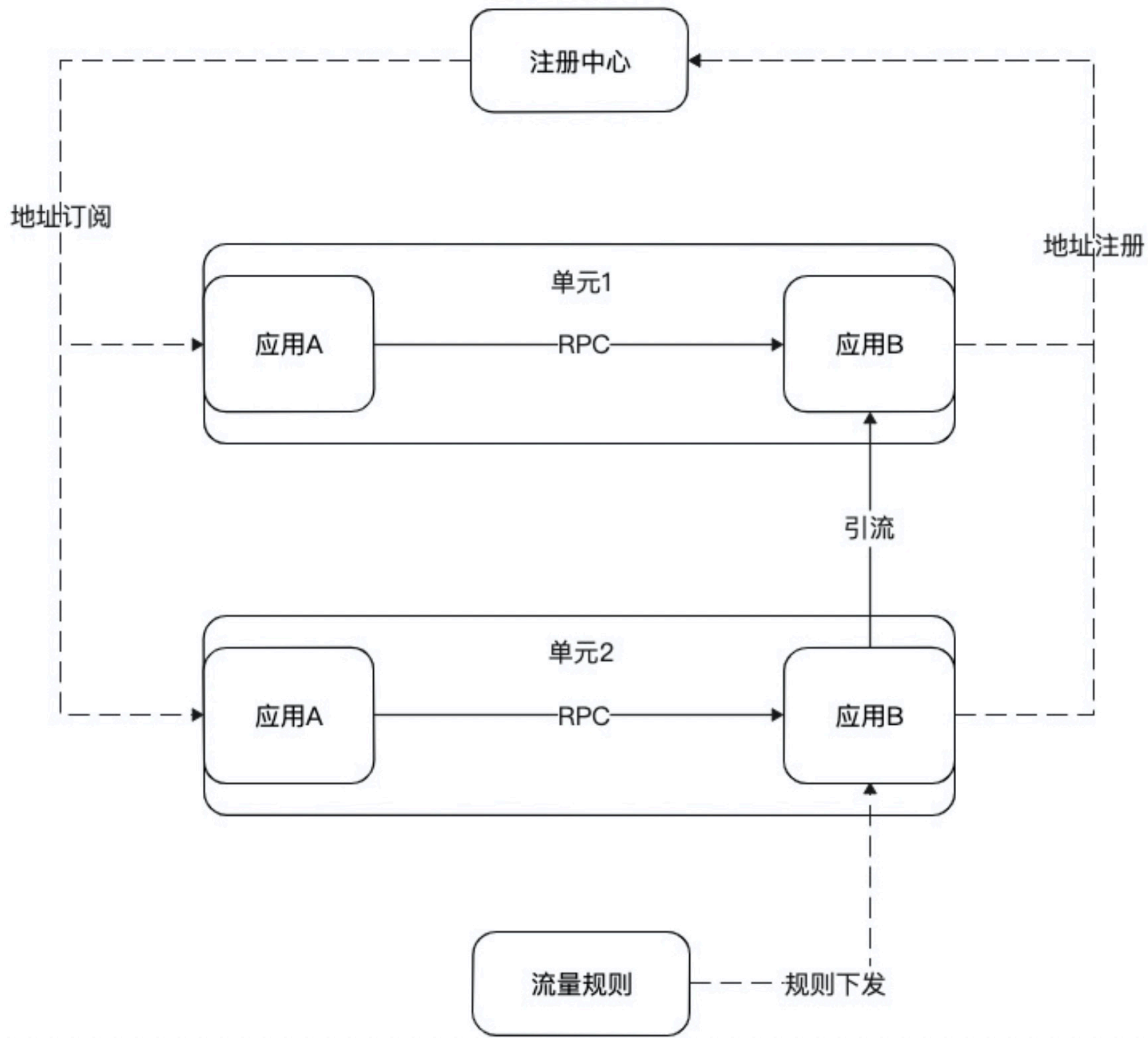
- 同应用 pod 维度引流。
- 常态化绘制性能基线。



# 01

## 能力建设-单应用引流

- 基于单元化架构，同应用不同单元间切流。
- 灰度发布，容灾，机房新建。

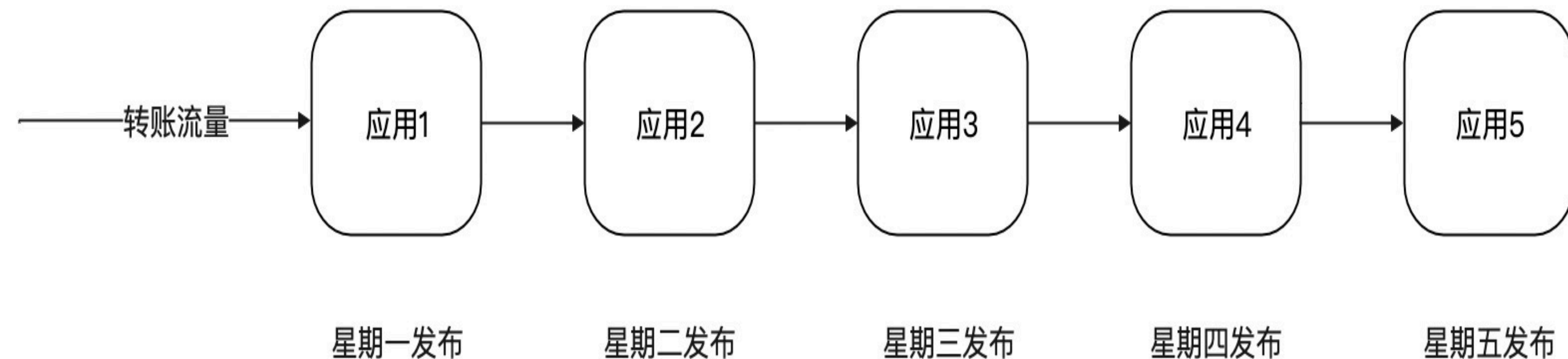


# 01

## 能力建设-业务链路隔离

- 一次转账经过5个系统处理。

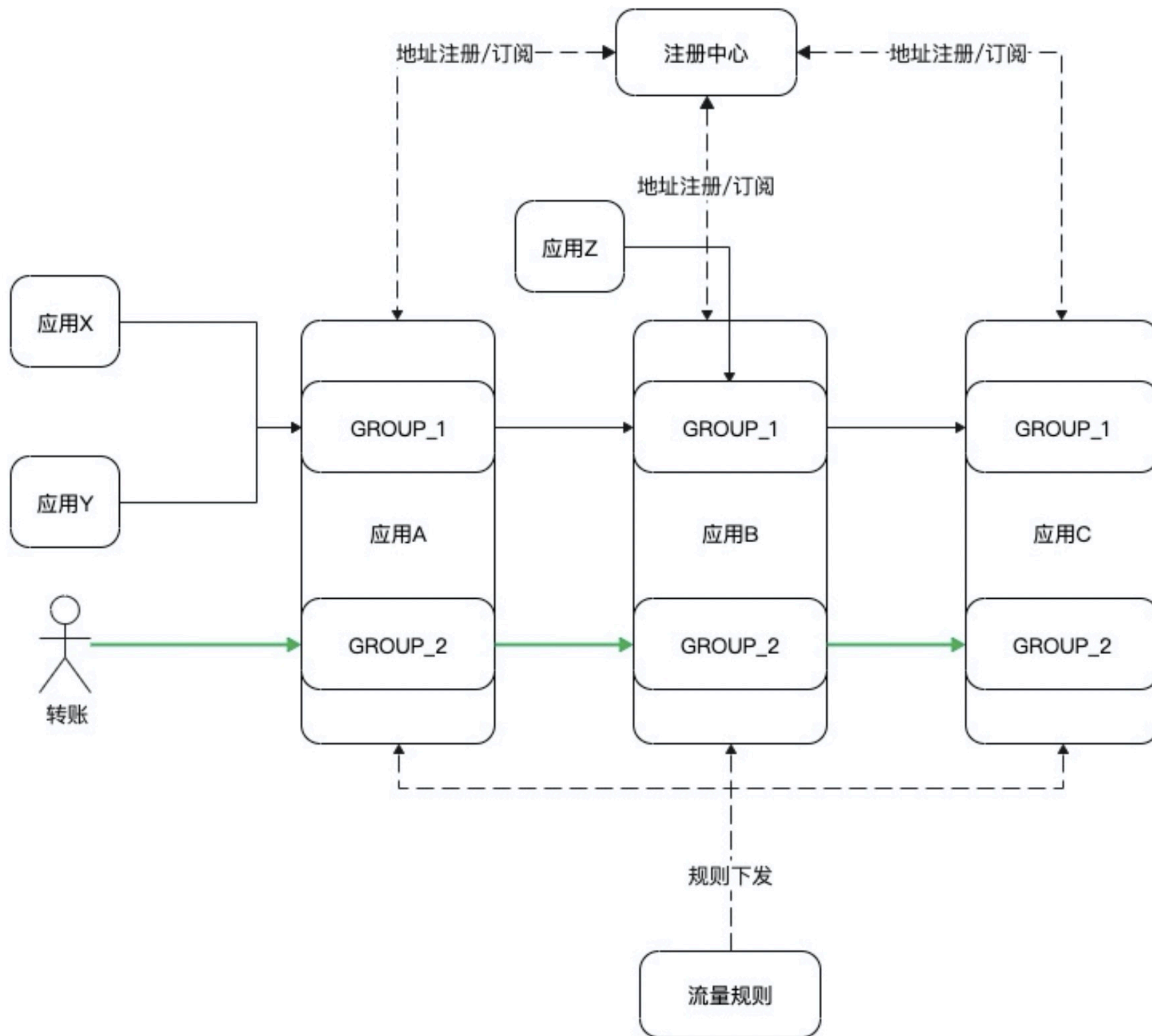
- 这条链路上每个工作日都有应用在发布，你慌不慌？



# 01

## 能力建设-业务链路隔离

- 业务对流量进行打标
- 客户端根据（标的+规则）将流量引流到不同分组
- 隔离，高保，容灾



## 2. 价值思考

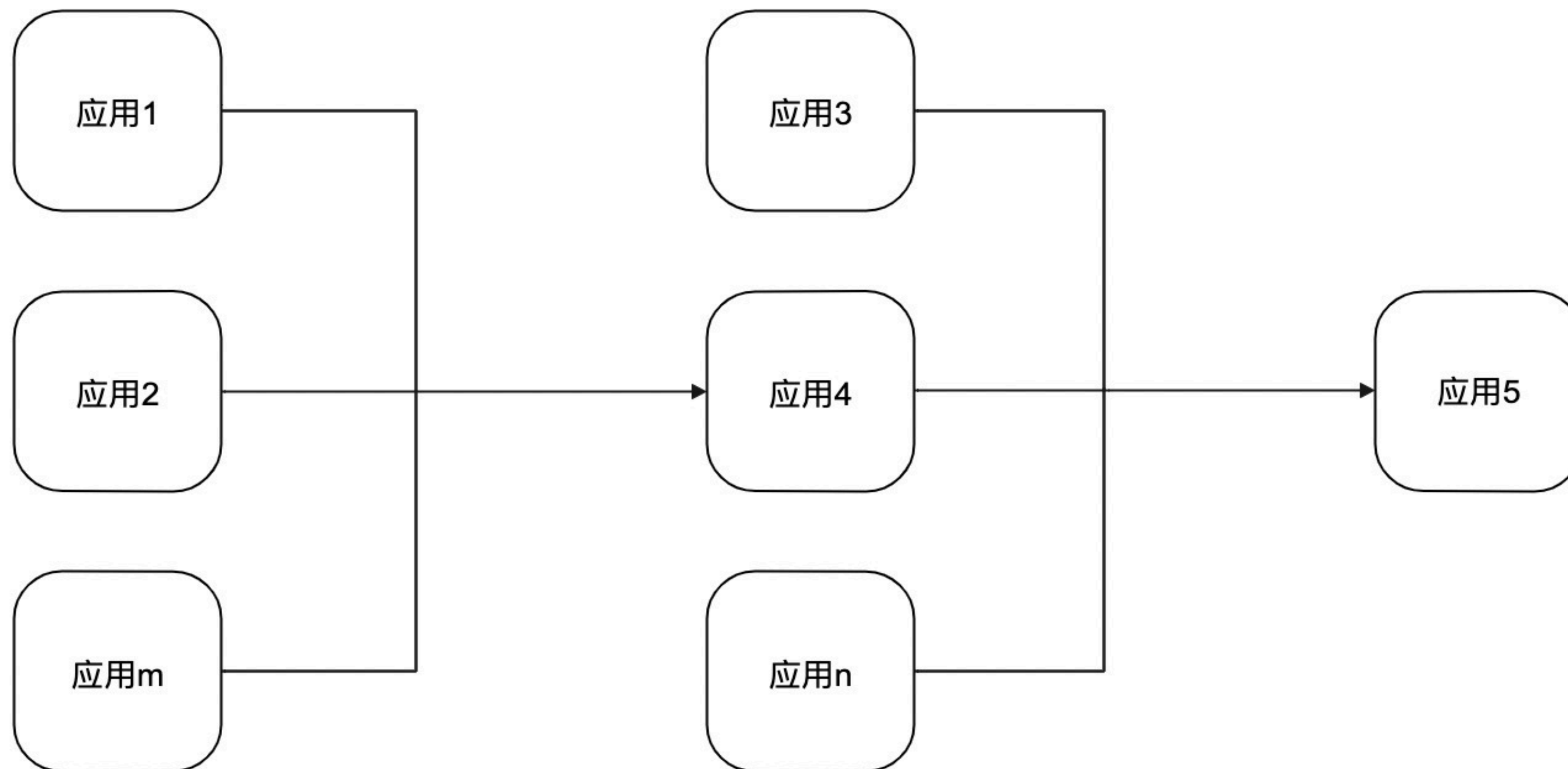
MOSN 上的能力我们其实在 SDK 上都能实现

所以 Service Mesh 的价值是什么？

## 02

### 价值思考

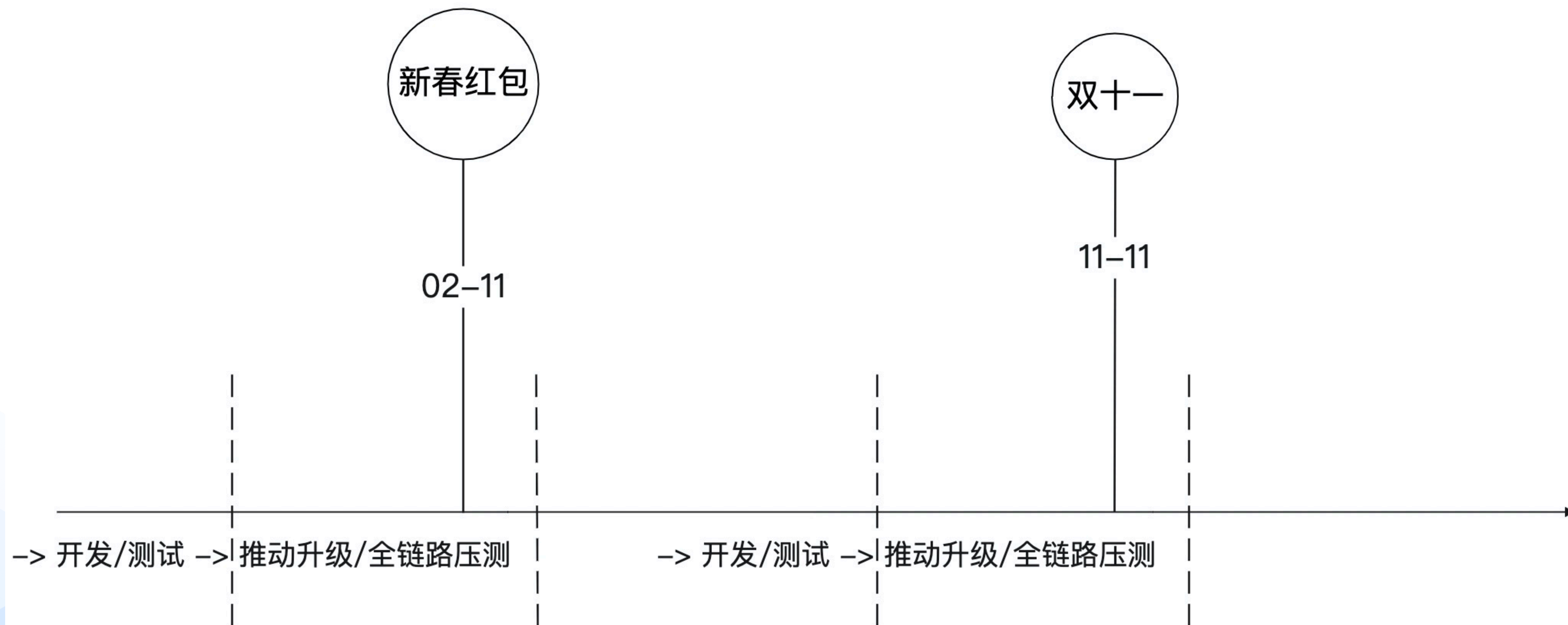
- 业务链路隔离，需要整个链路及周边应用升级



## 02

### 价值思考

- 如自适应限流需要多次迭代进行验证优化。





## 02 价值思考

- 打扰业务
- 推进困难
- 迭代周期长
- 版本碎片化严重



- 在不打扰业务的情况下将基础设施快速稳定地升级到全集群

### 3. 研发效能

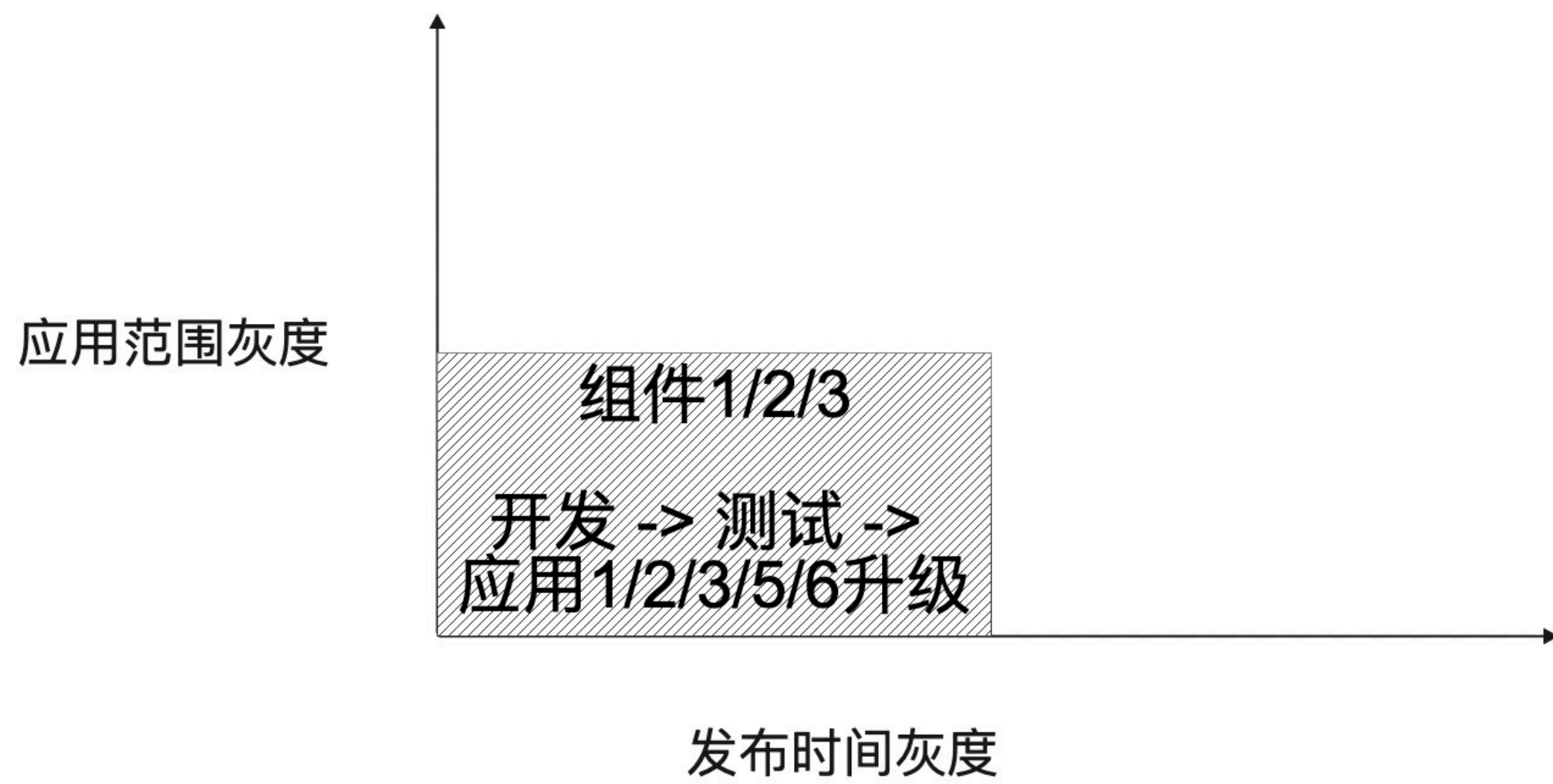
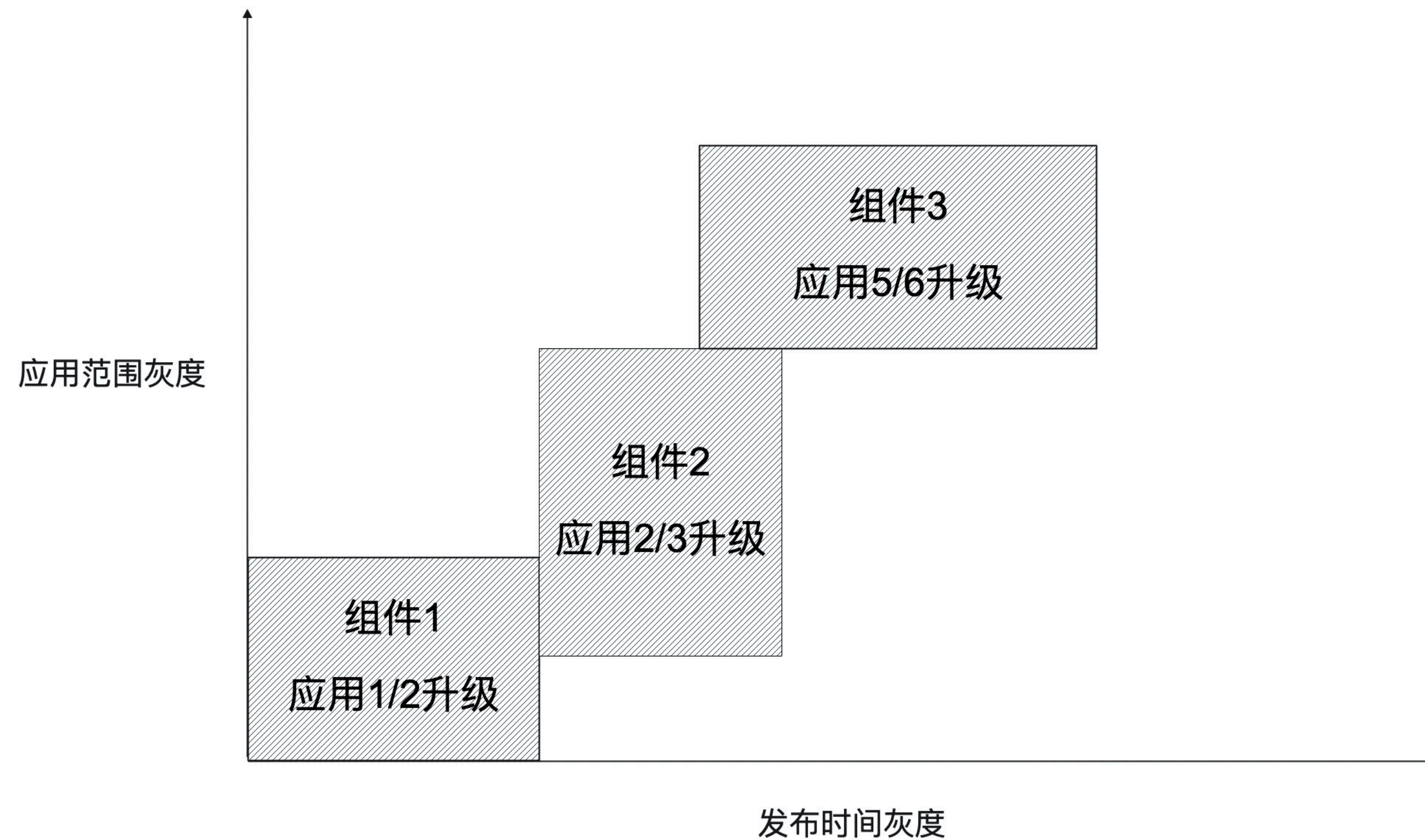
当 Service Mesh 的优势逐渐显现，越来越多的能力希望下沉到 MOSN 中。

而我们逐渐发现 MOSN 在实际迭代过程中遇到的一些困难，其中比较突出的两个问题：

# 03

## 研发效能

- 变更量大
- 集中式发布

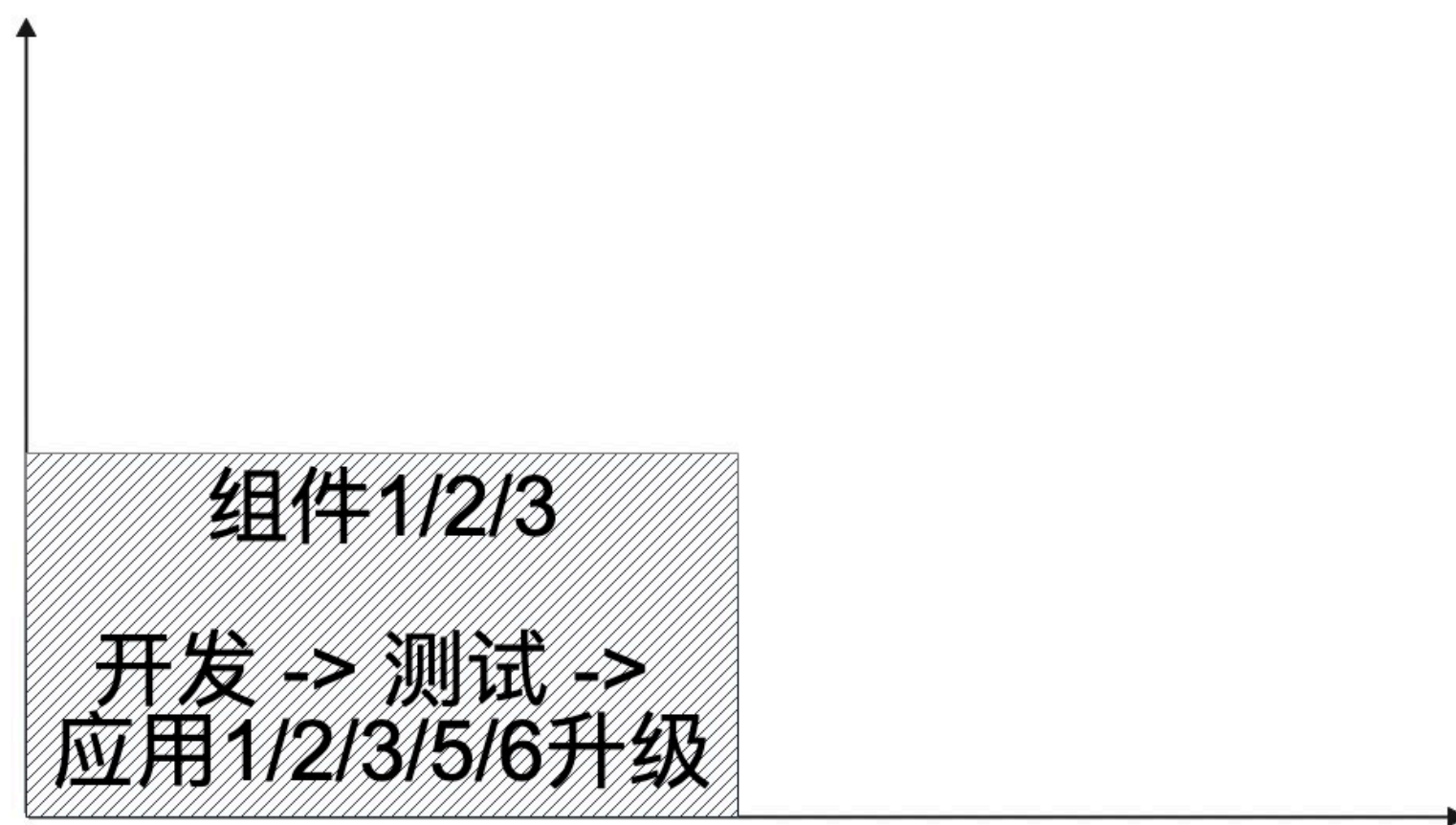


# 03

## 研发效能

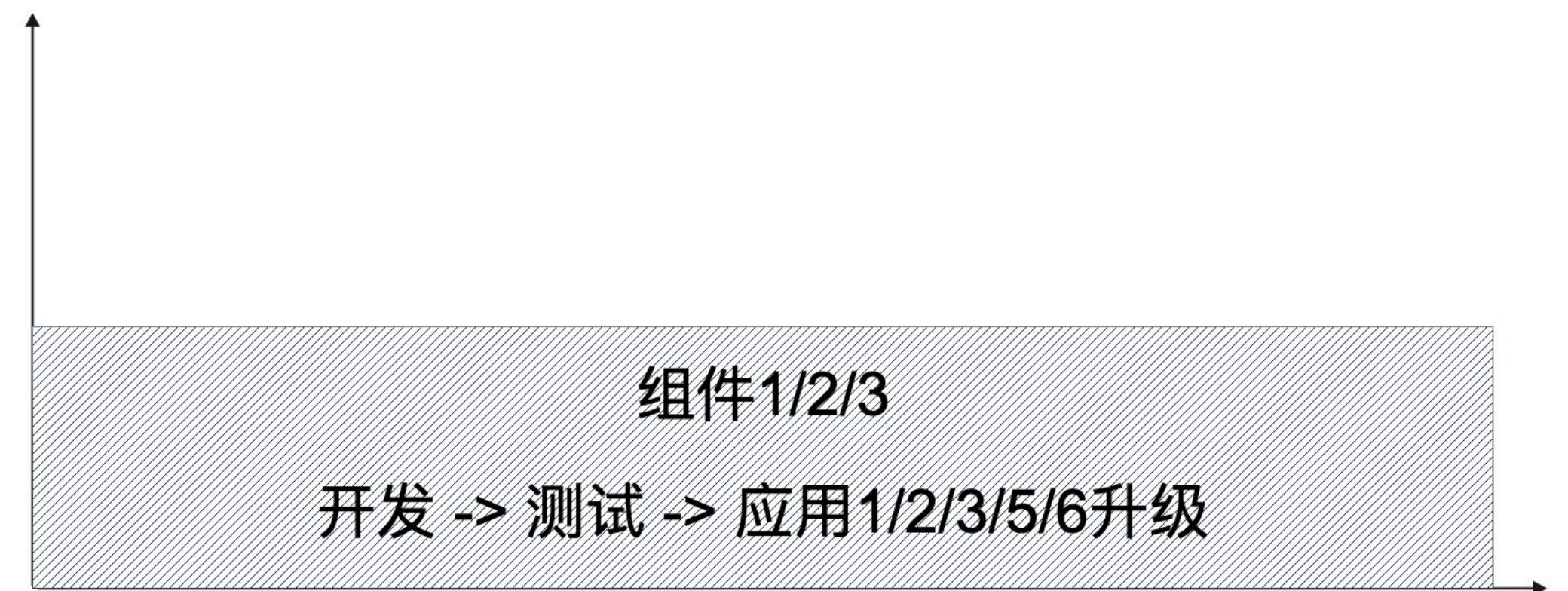
- 灰度时间拉长

应用范围灰度



发布时间灰度

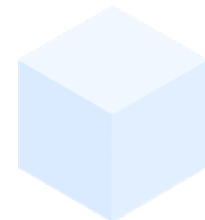
应用范围灰度



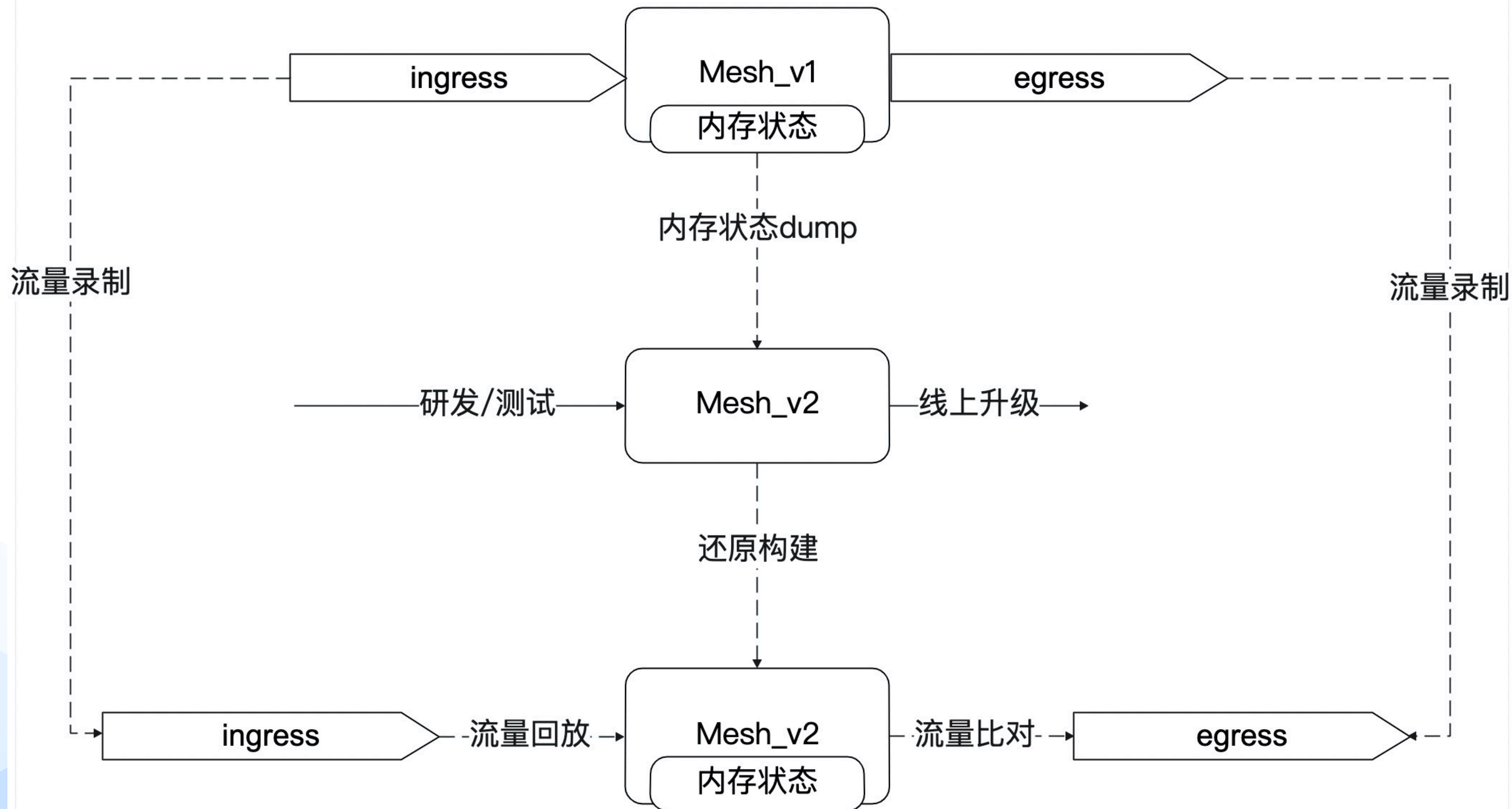
发布时间灰度

# 03

## 研发效能-质量保障



- 流量录制
- 状态 dump 和重建
- 流量回放



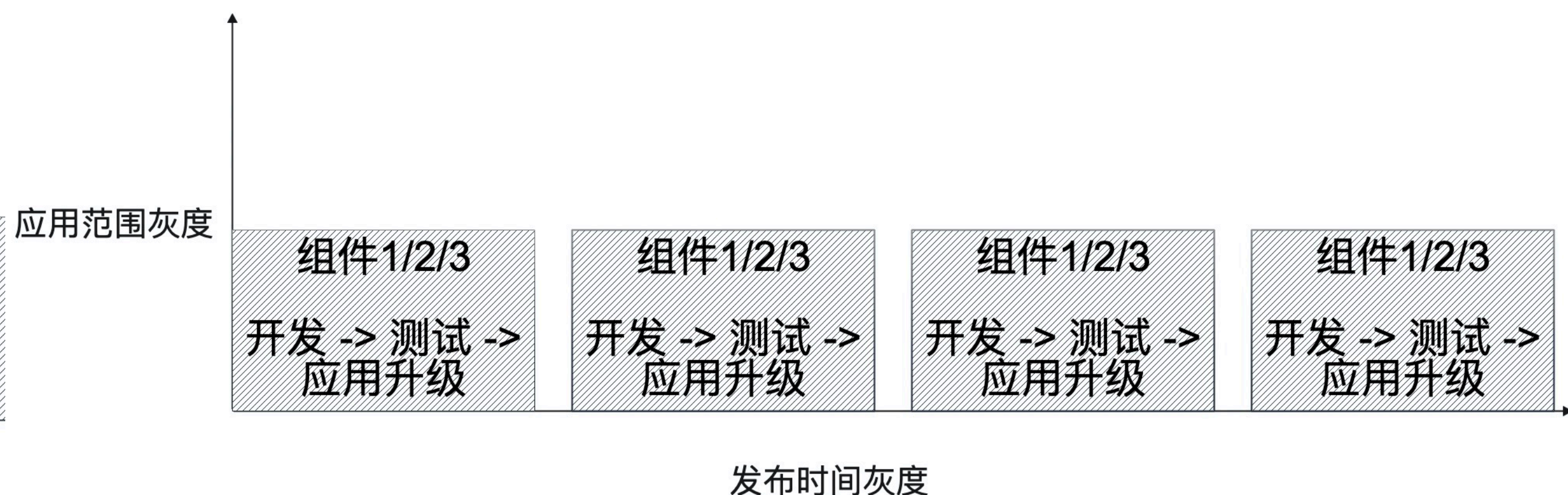
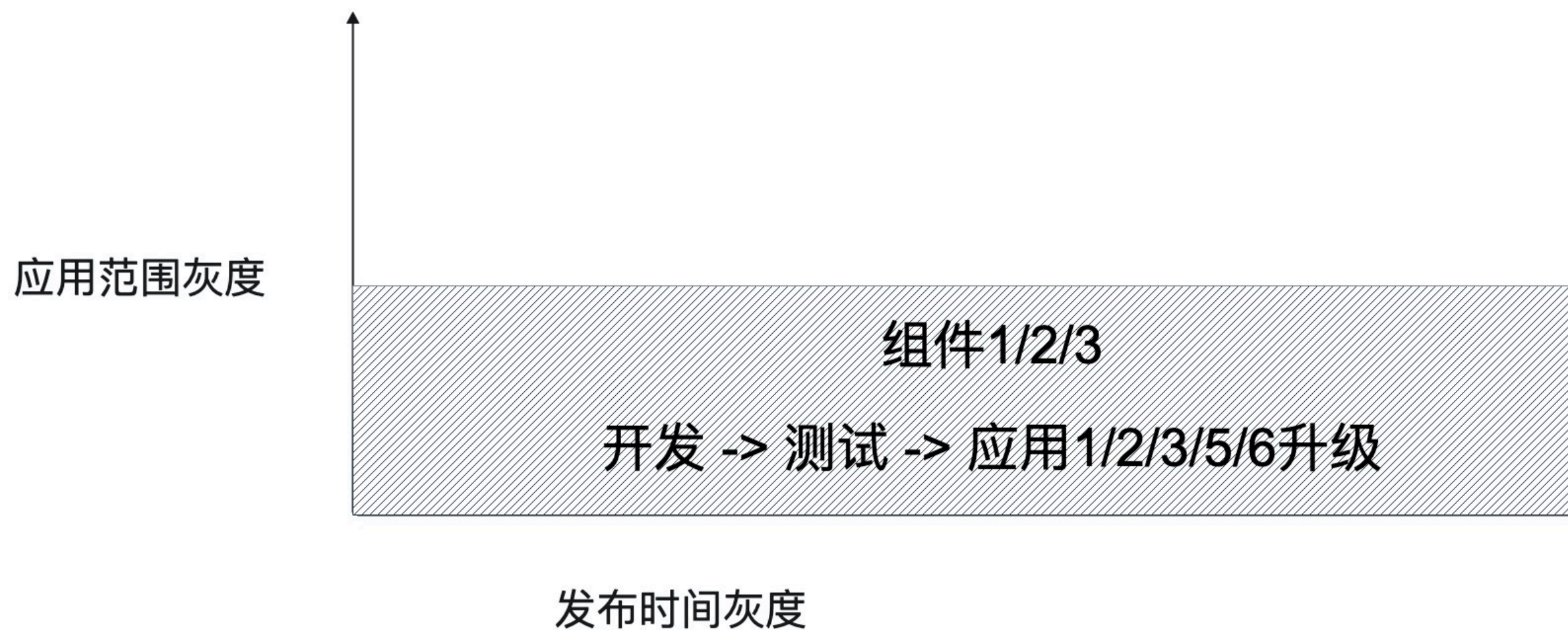
# 03

## 研发效能-技术风险

• 高频率的发布升级方式

挑战

• 升级的效率，侵入性，稳定性

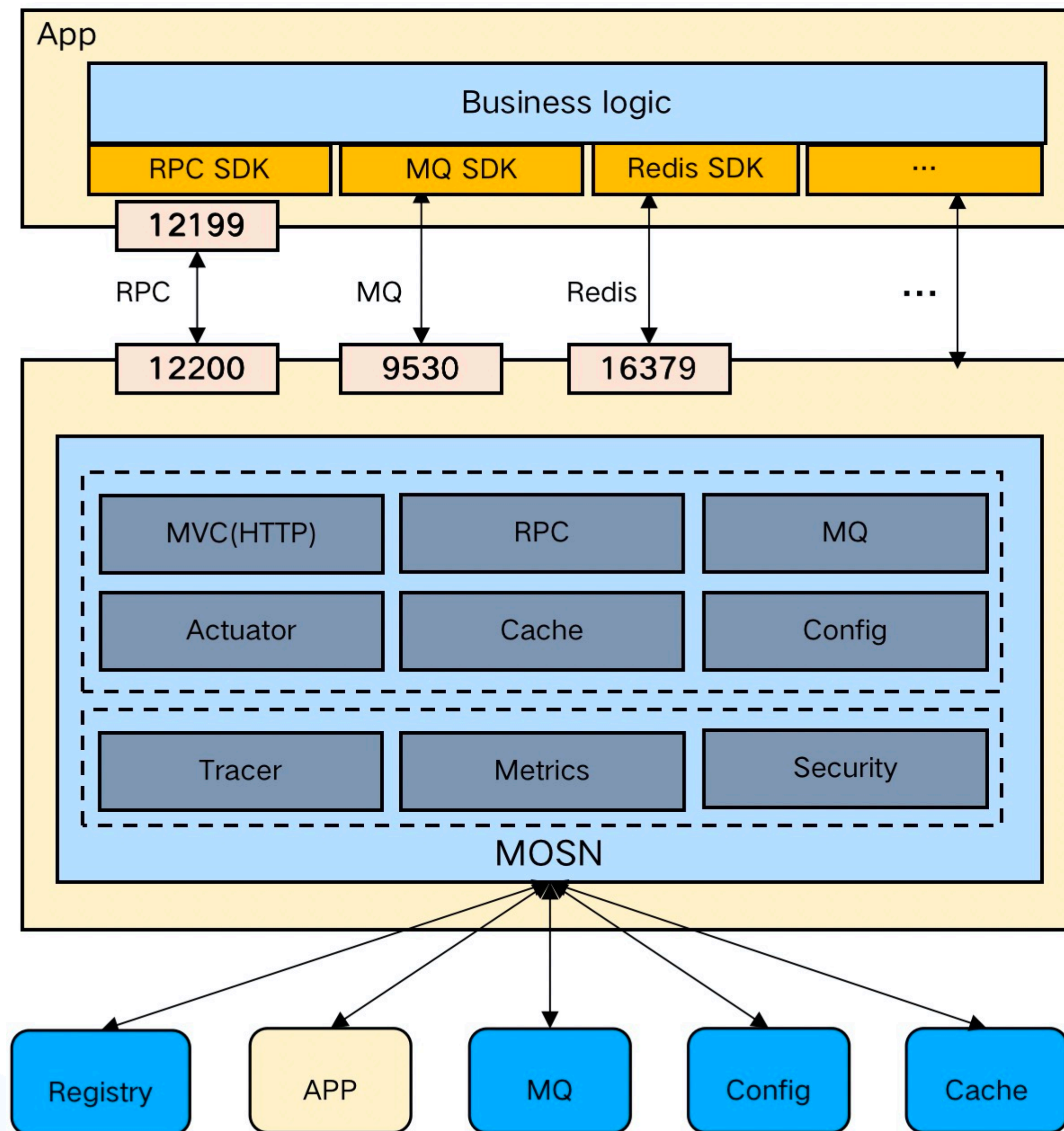


## 4.未来方向和思考

# 03

## 未来方向和思考

- 现状

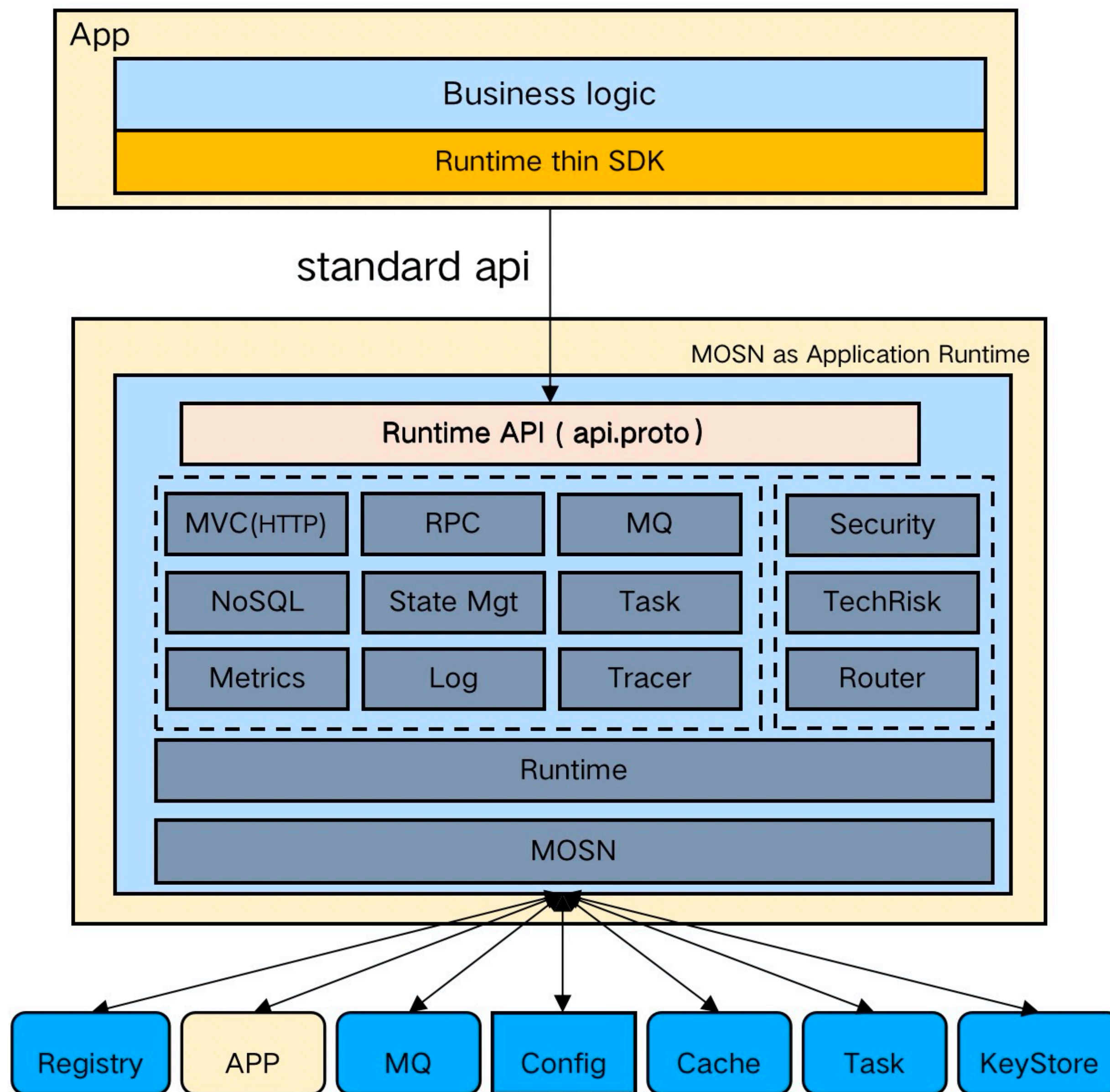




# 03

## 未来方向和思考

- 1.能力下沉，流量接管。
- 2.统一的交互标准



# 03

## 未来方向和思考

- 3.结合业务挖掘有价值的场景，通过 Service Mesh 的优势为业务带来帮助。
- 4.继续投入开源。社区输入 + 蚂蚁内部大规模场景实践。



钉钉扫码，入群交流

THANKS. 





# MOSN 的无人值守变更实践

黄家琦（嘉祁）

SOFABolt-Python 组件维护者。



**01**

**规模化运维挑战**

**02**

**无人值守变更实践**

**03**

**技术风险的思考方向**



# 1.MOSN 规模化后的运维挑战

# 01

## 传统变更方式

- 变更流程：pod 粒度的变更工具支持，人工控制变更范围与进度
- 问题发现：上层业务的错误发现，配合 MOSN 的错误码
- 双轨版本：灰度小范围快速迭代功能，稳定版本大范围推进



# 01

## 传统方式带来的问题

- 覆盖对接的业务越来越多
- 技术栈多而复杂
- 新特性与 bugfix 堆积

风险高

- 变更影响评估面广
- 变更依赖链路长
- 需要控制潜在问题的影响范围

MOSN  
变更

迭代慢

周期长



01

## MOSN 规模化的运维挑战

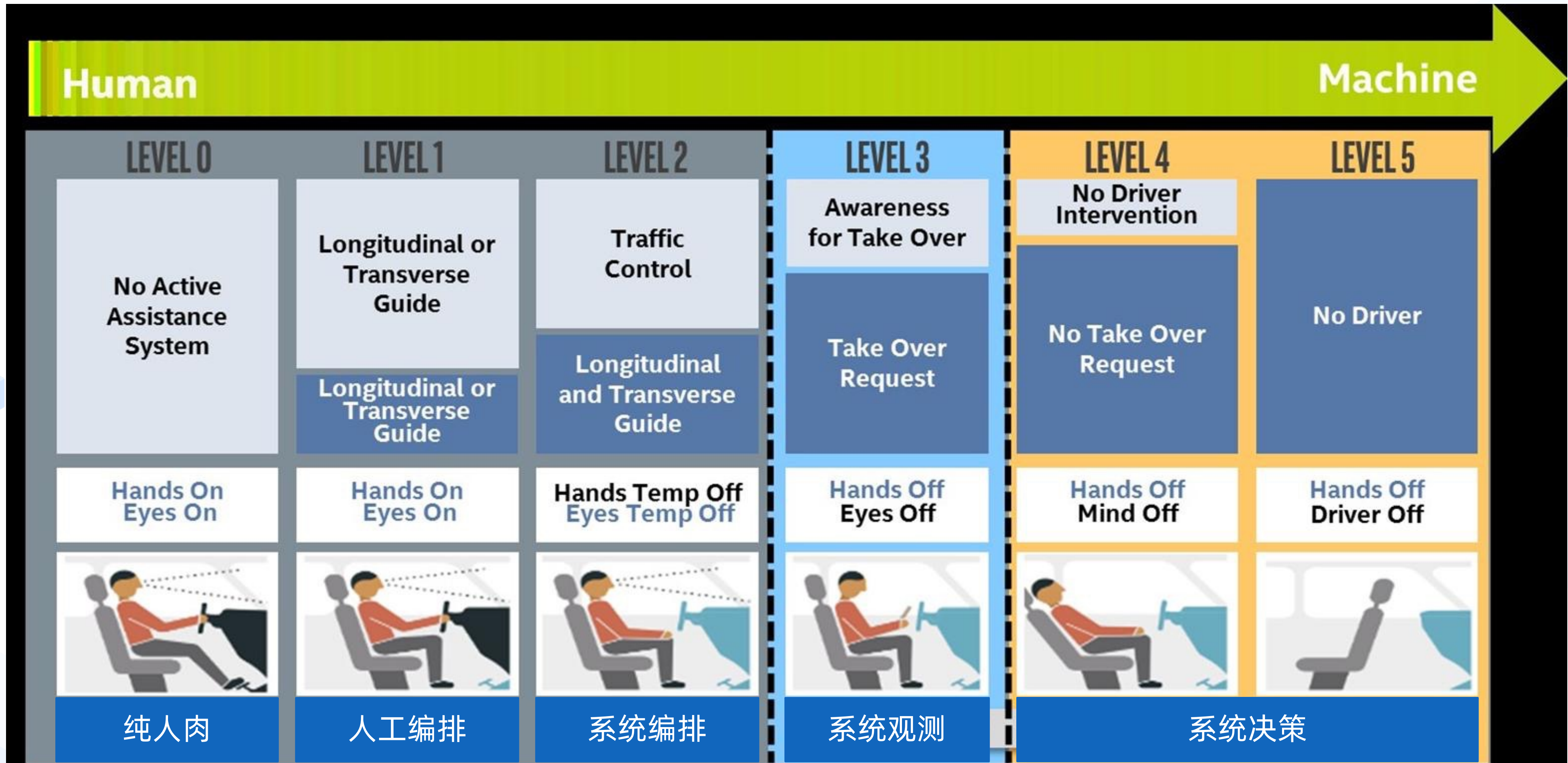


效率

稳定

## 2.MOSN 的无人值守变更

# MOSN 的无人值守变更——无人值守的分级



命令行

人工灰度

Hands on  
Eyes on  
Phone on

强制灰度

Hands off  
Eyes on  
Phone on

变更防御

Hands off  
Eyes off  
Phone on

变更自愈

Hands off  
Mind off  
Phone on

Hands off  
Mind off  
Phone off

# 02

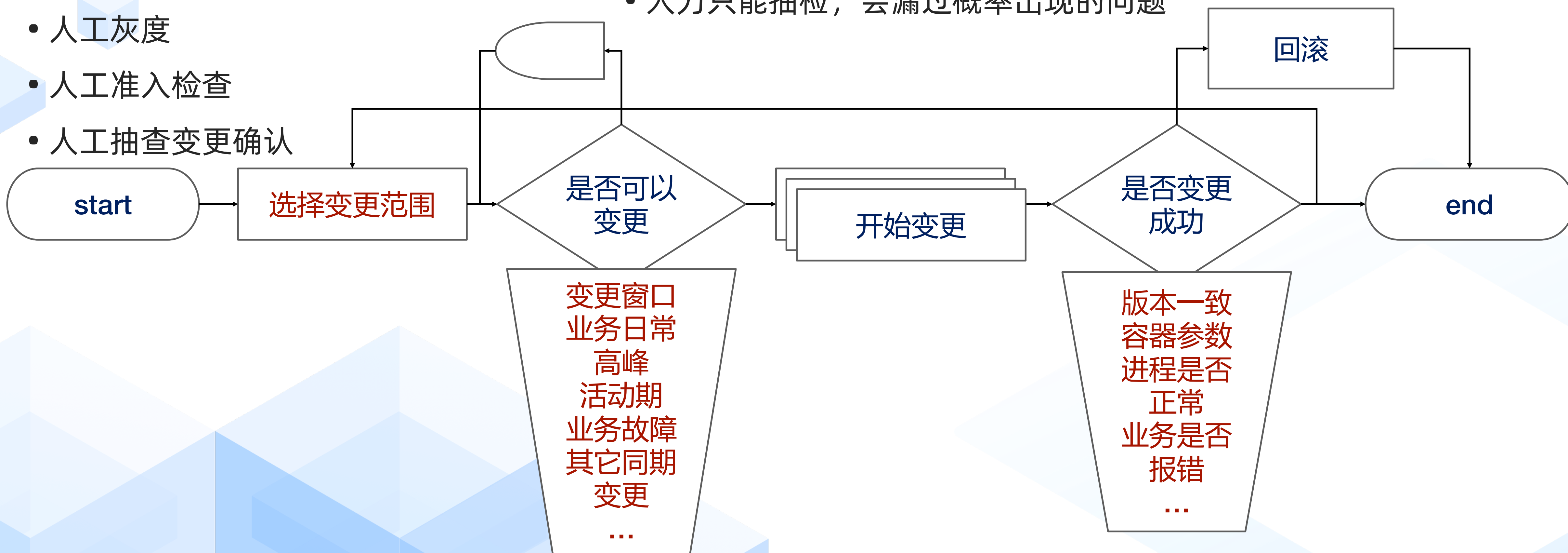
## 变更的风险与防控

### 人工保证带来的风险

- 灰度过程不完整或被跳过
- 变更窗口信息不同步、准入规则被遗漏
- 人力只能抽检，会漏过概率出现的问题

### 传统的变更流程

- 人工灰度
- 人工准入检查
- 人工抽查变更确认

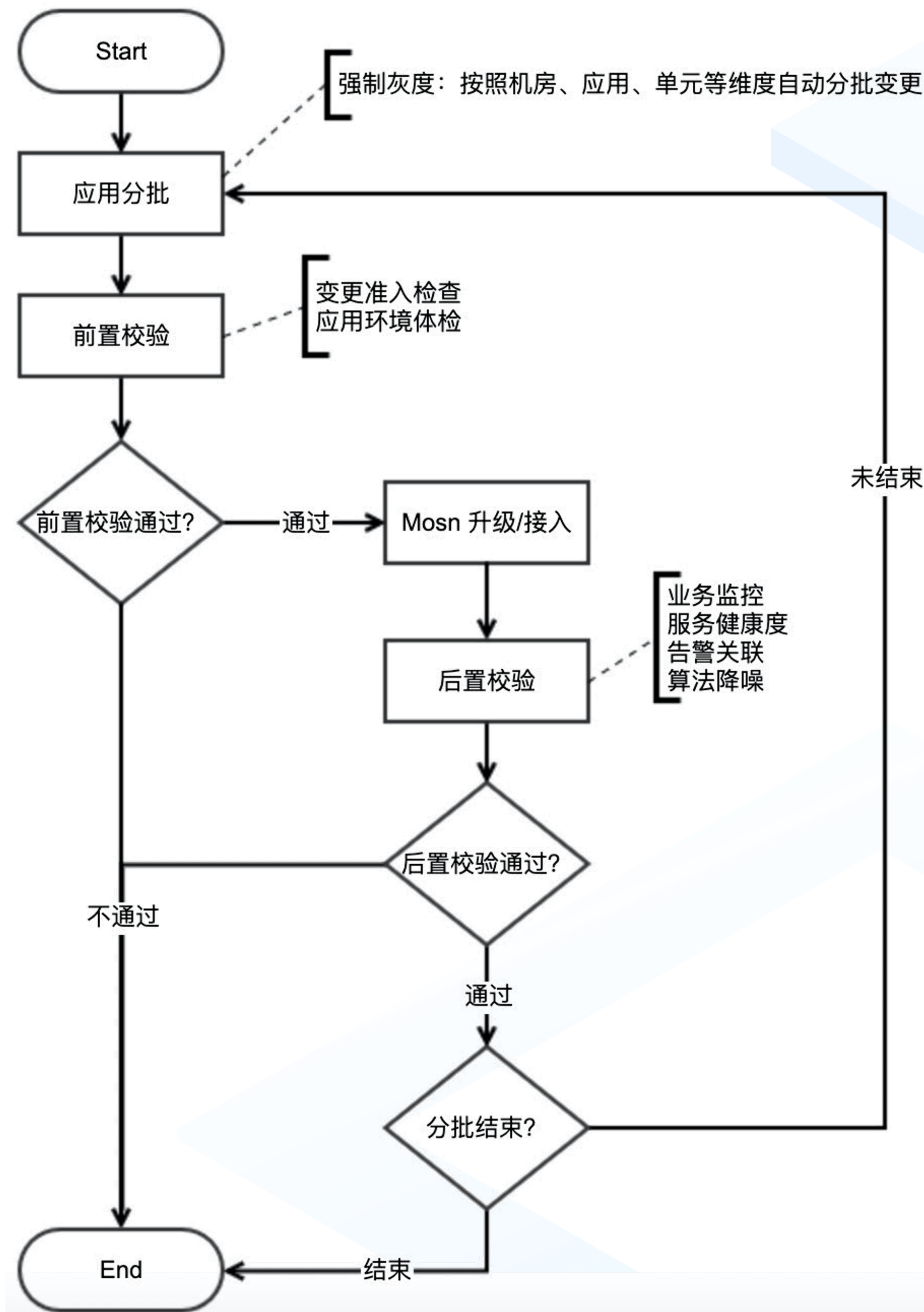


# 02

## 变更的风险与防控

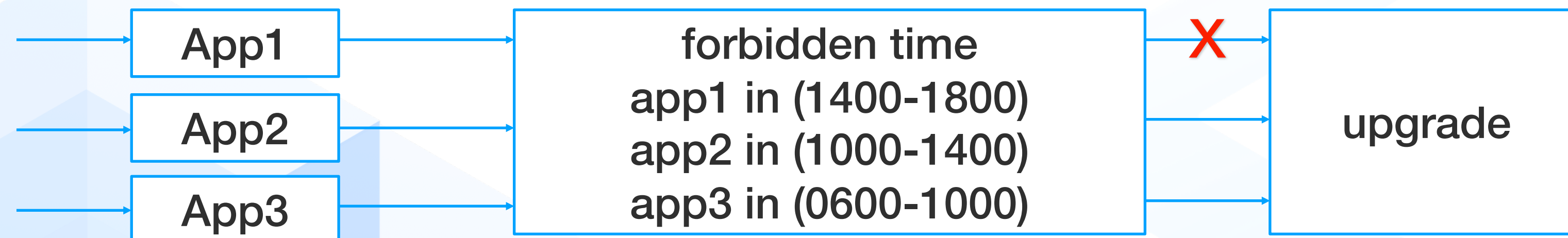
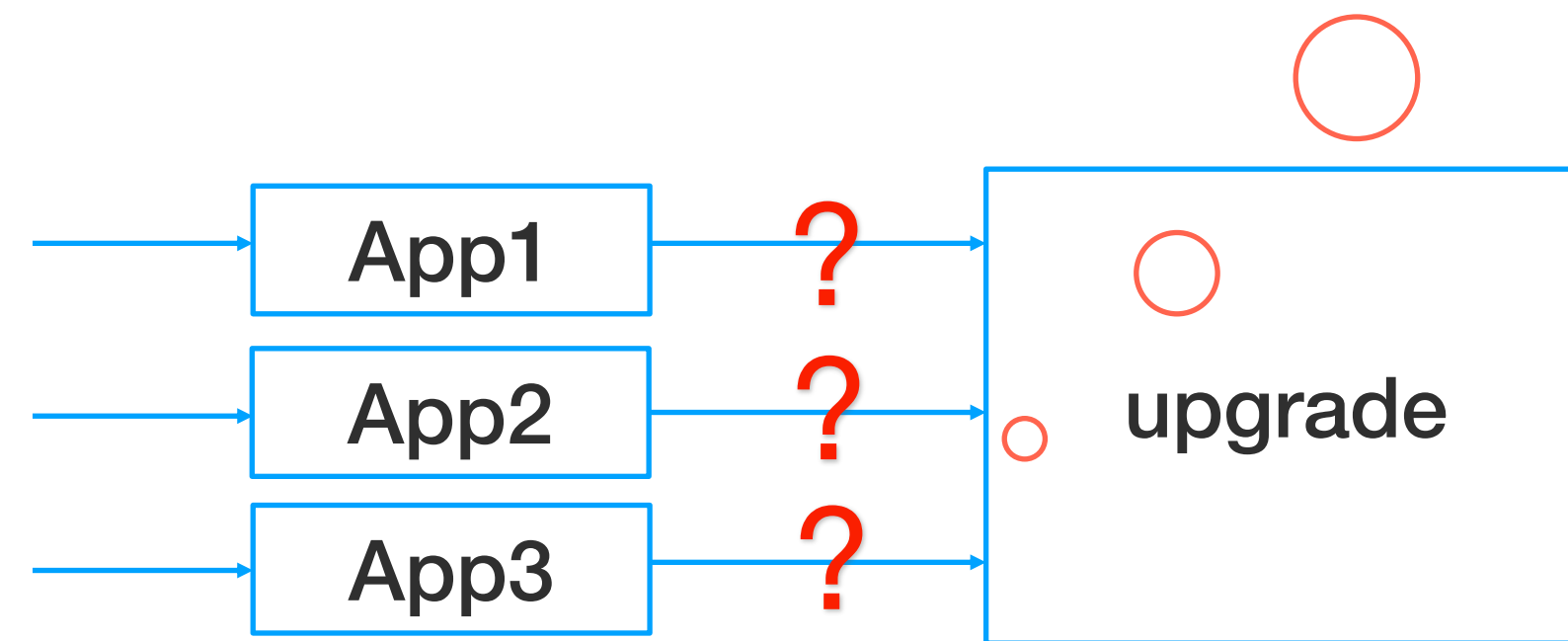
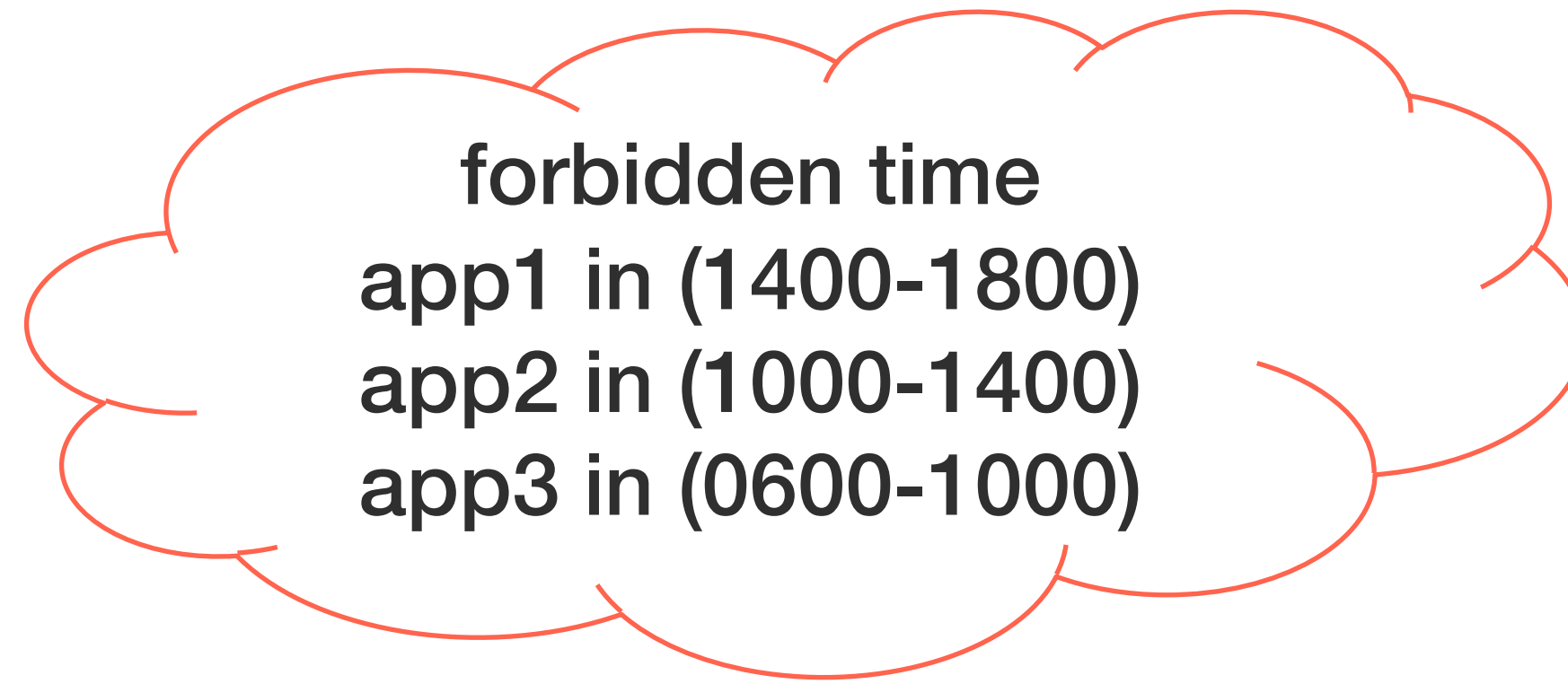
- Hands off
  - 全流程策略编排
  - 强制灰度
- Eyes off
  - 变更准入检测
  - 变更完整性检测
  - 完整的健康度检测、业务影响阻断
  - 完整的变更一致性检查、修复

### 变更防御流程



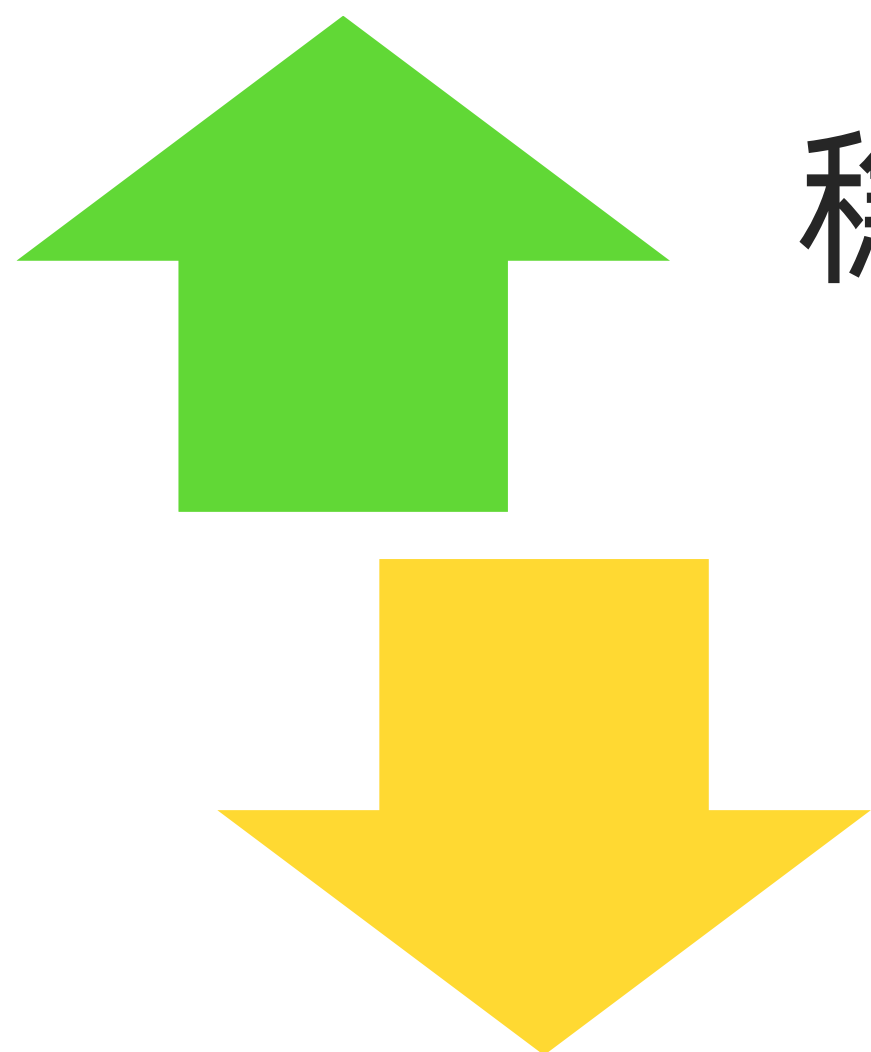
# 02

## 变更的风险与防控



# 02

## 变更的风险与防控



稳定性提升

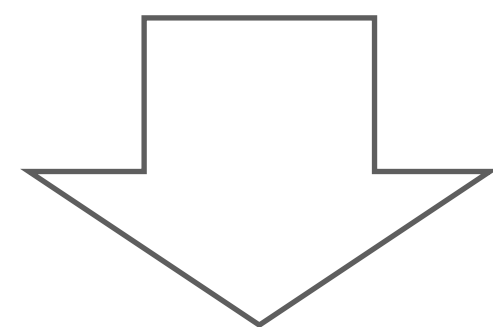
效率下降 **WHY?**

- 依赖业务错误检查，阈值未必合理
- 全面检查，进一步放大不合理阈值导致的中断

# 02

## MOSN 健康度与 SLO

变更健康检查 == 业务错误检查



变更健康检查 == MOSN 健康检查 => MOSN 健康度

- MOSN 健康检查导致
- 业务自身错误导致
- 其它中间件问题
- 其它基础设施问题



# 02

## MOSN 健康度与 SLO

环境代码问题  
无流量 启动自检

业务兼容性问题  
运行期 Error metrics



MOSN SLO

# 02

## 变更方式的演进

- 传统原地冷升级：upgrade sidecar 同时重启业务容器
  - 业务需要冷启动，通常较 MOSN 慢，较 MOSN 启动失败率高。
- 云原生滚动升级模式：create new pod, remove old pod
  - 新建 pod，避免旧配置影响
  - 可能受限于资源碎片等资源问题
  - 业务需要冷启动

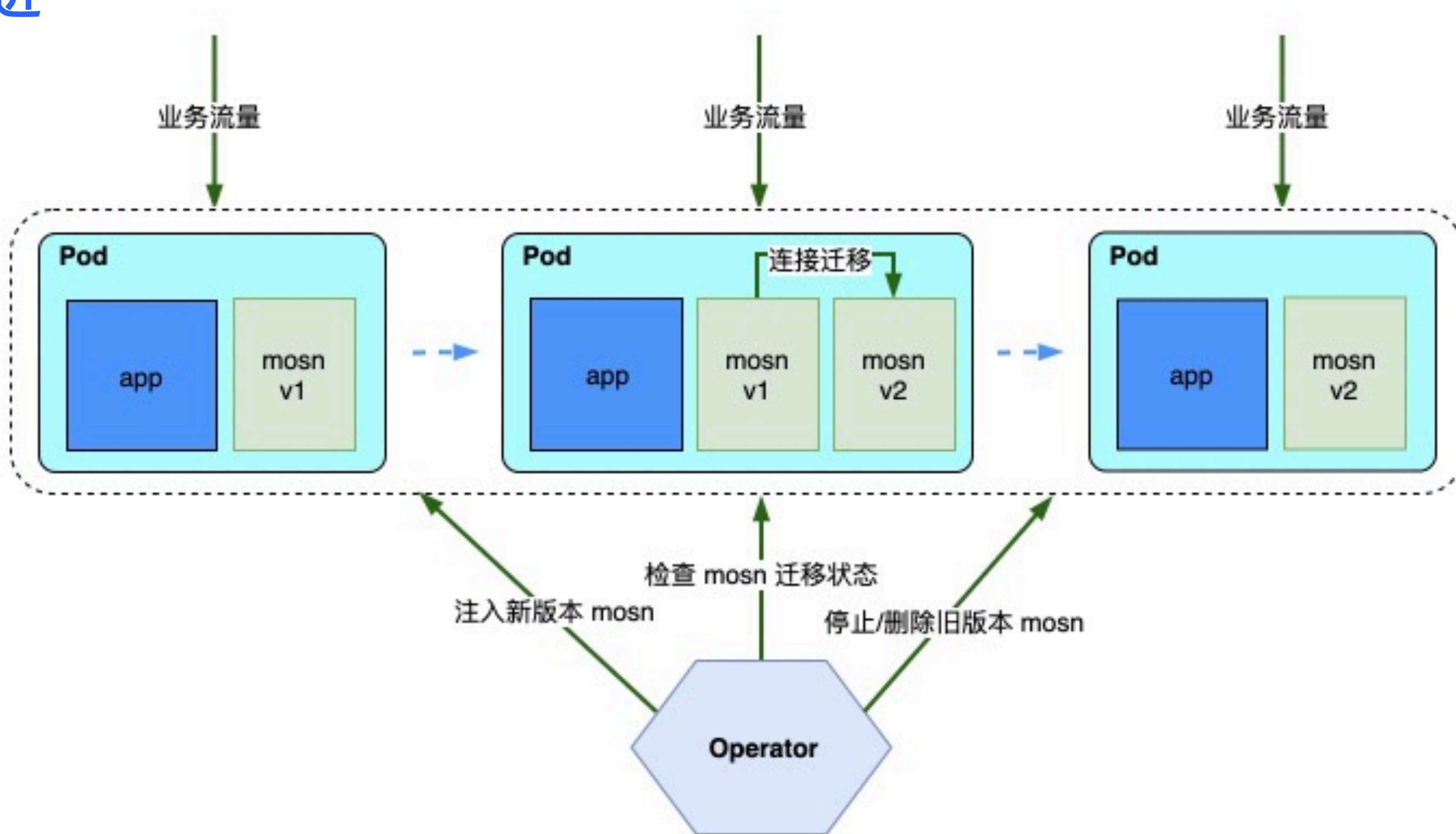
## 02

### 变更方式的演进

- MOSN 的热升级

- 有状态连接迁移复杂度高，新增协议开发难度高（如 TLS 密钥）

- 动态容器注入时资源分配逻辑复杂，回滚逻辑复杂

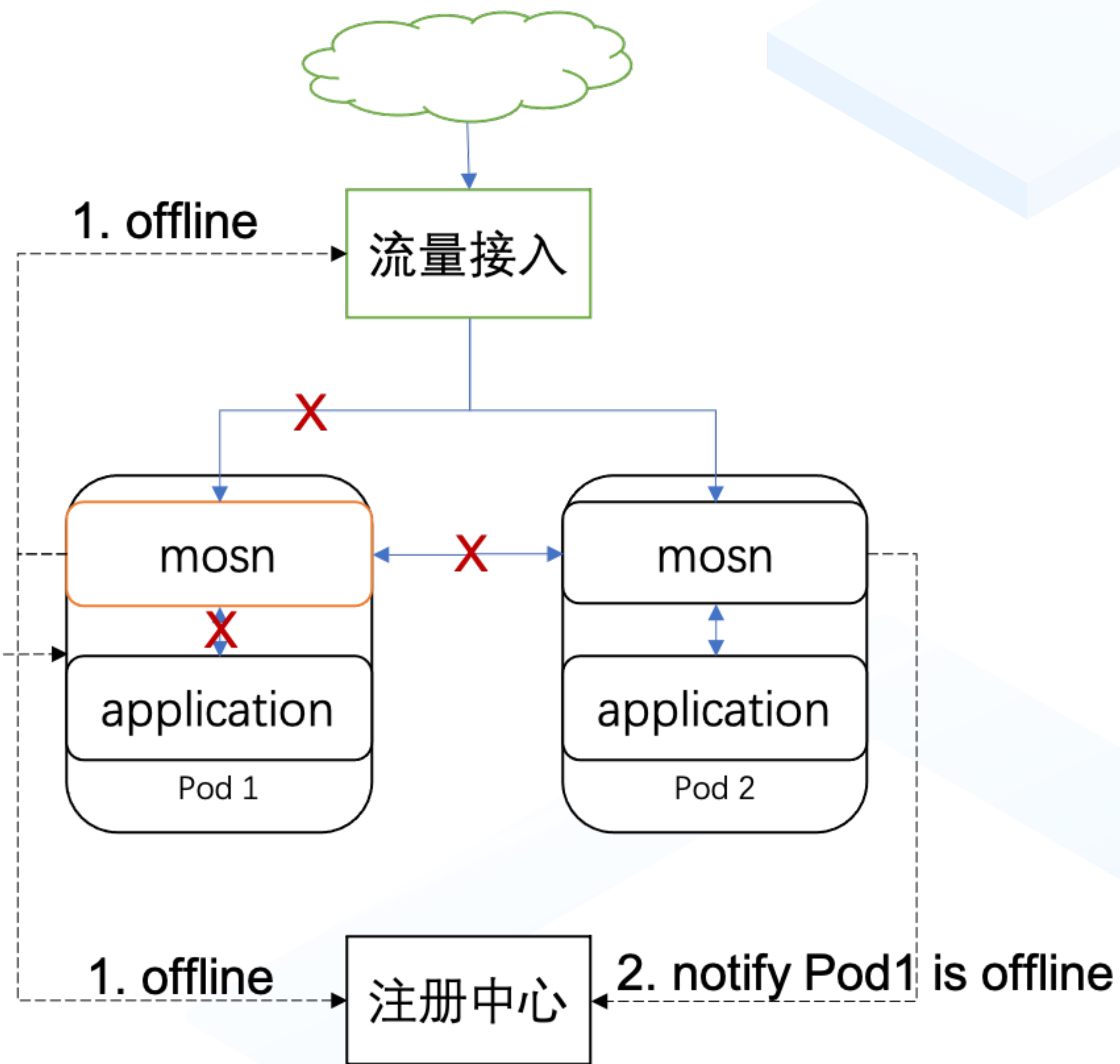


# 02

## 变更方式的演进

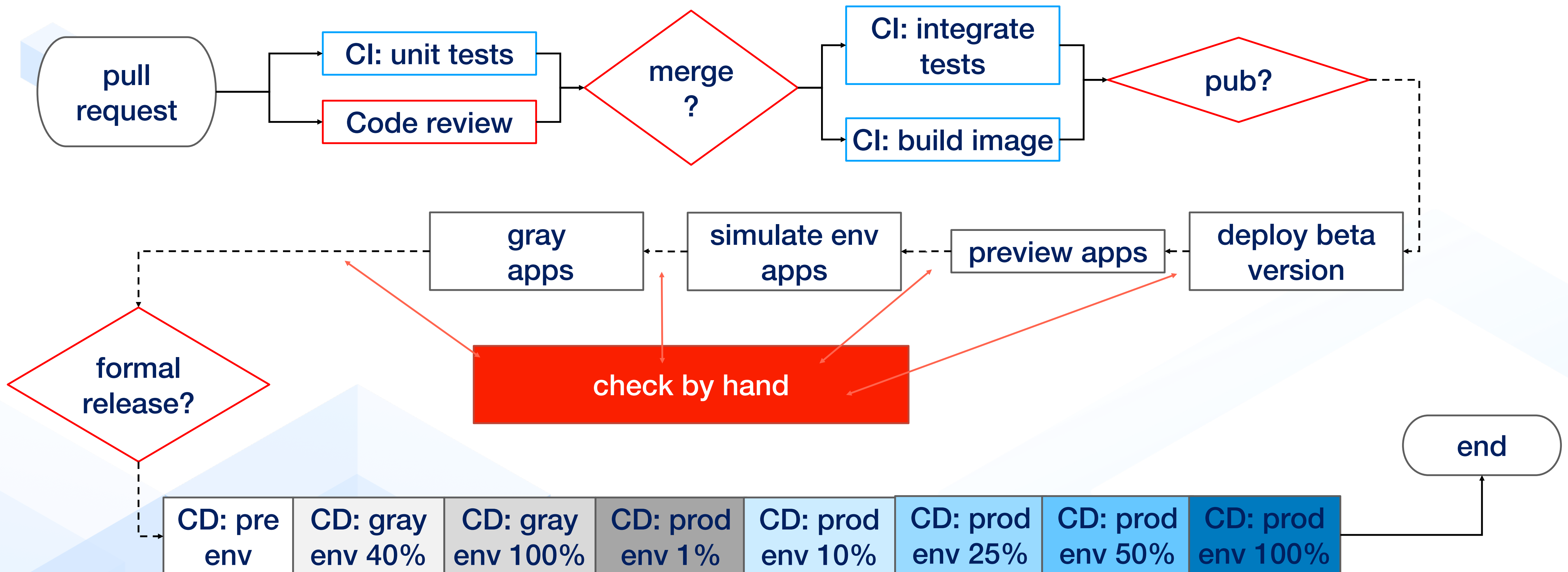
- 温升级模式
  - 去掉业务冷启动时间
  - 避免业务冷启动失败阻塞变更
  - 只要求配置向下兼容，简化开发

0. mosn upgrade



# 02

## 持续集成与持续发布



# 02

## 持续集成与持续发布

### CI

- 自动化CI流程
- 完善集成测试

### CD-nightly

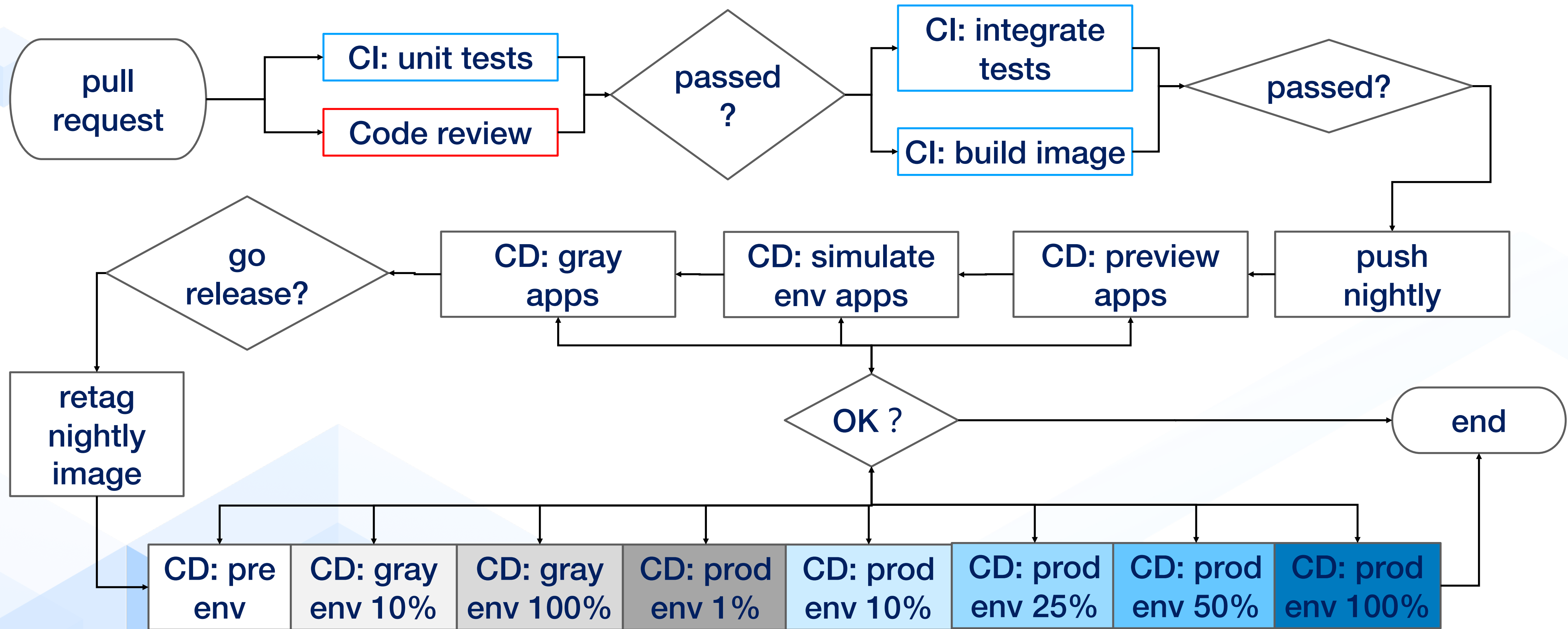
- 对接 CI 结果
- 常态 nightly 机制
- Nightly 反馈报告

### CD-正式

- Nightly 评估与准入
- 优化发布流

# 02

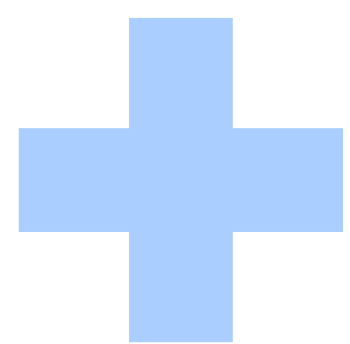
## 持续集成与持续发布



# 02

## 持续集成与持续发布

变更自  
动推进



自主  
决策



L5 无  
人值守



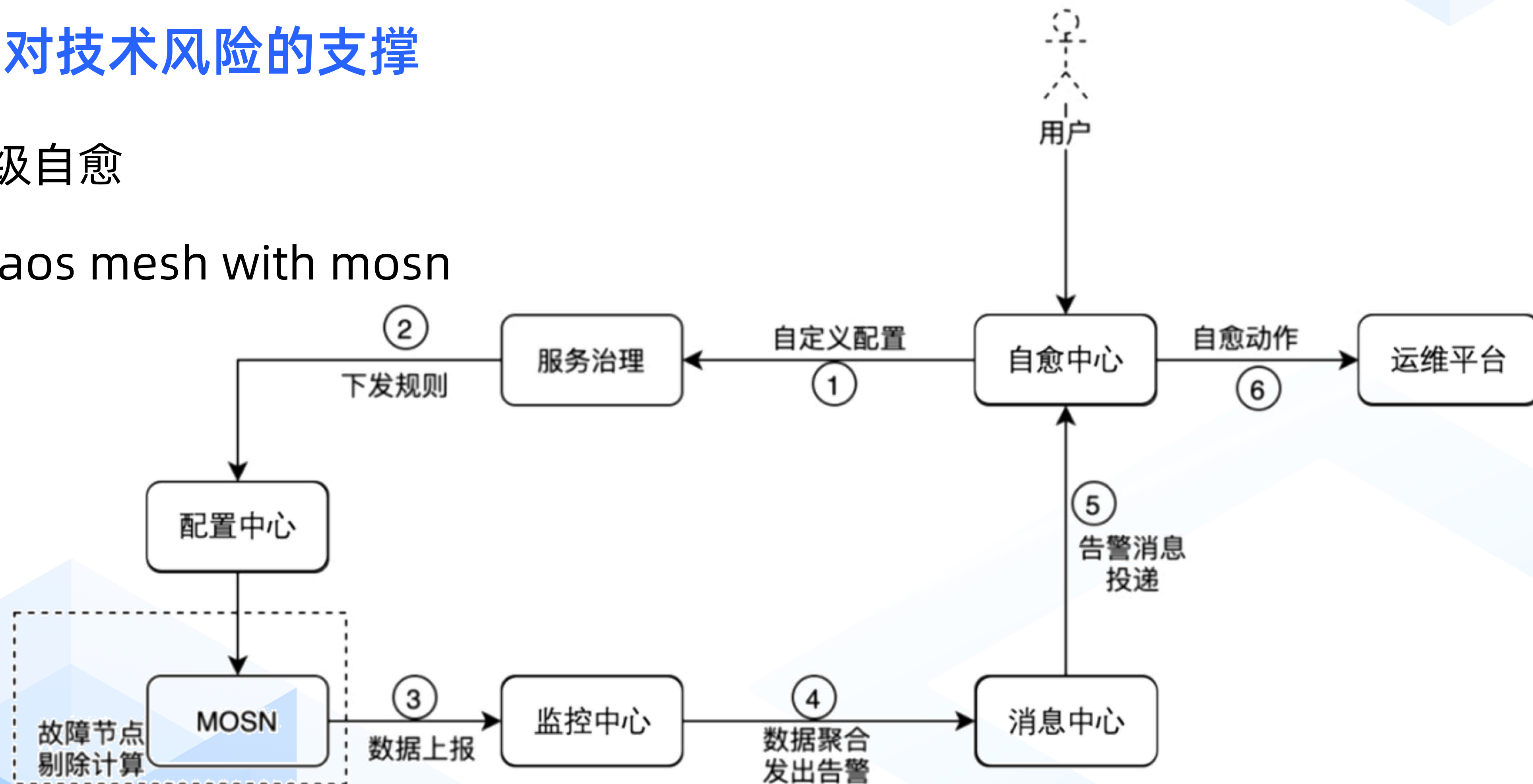


### 3.MOSN 技术风险的思考与方向

# 03

## MOSN 对技术风险的支撑

- 秒级自愈
- Chaos mesh with mosn



# 03

## MOSN 技术风险的演进

- 更稳
  - SLO
  - 自主决策
- 更快
  - 月度发布
  - L5 无人值守



钉钉扫码，入群交流

THANKS. 





SOFAStack钉钉群

“金融级分布式架构”微信公众号





# Service Mesh 在美团的落地挑战与实践

薛晨  
美团技术专家





01

美团服务治理  
演进与最新进展

02

Service Mesh  
落地挑战与破局

03

总结与展望

# 01

## 美团服务治理演进与最新进展

- 美团服务治理演进
- 遇到问题
- Mesh化架构



# 美团服务治理演进

阶段一  
标准化

服务框架统一  
命名服务统一  
监控预警  
数据分析

阶段二  
易用性/高性能

服务框架轻量化  
服务框架高性能  
链路追踪精细化  
测试工具可视化

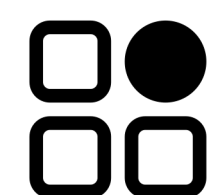
阶段三  
全方位服务治理

稳定性保障平台  
全链路压测平台  
链路级流量治理  
服务鉴权加密

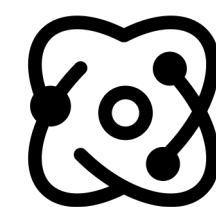
阶段四  
跨地域容灾扩展

Set 化架构支撑  
易用性平台建设

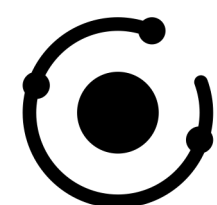
# 美团服务治理演进



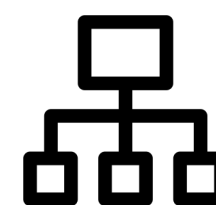
接入**数万**应用系统



日均**万亿**级业务调用



**数十万**级服务节点



覆盖**全量**多元业务线

Why Service Mesh?



## 效率低

迭代效率：业务代码与基础组件强耦合，彼此制约

融合效率：融合公司打通慢

## 赋能慢

小语种：无法享受完善的服务治理红利

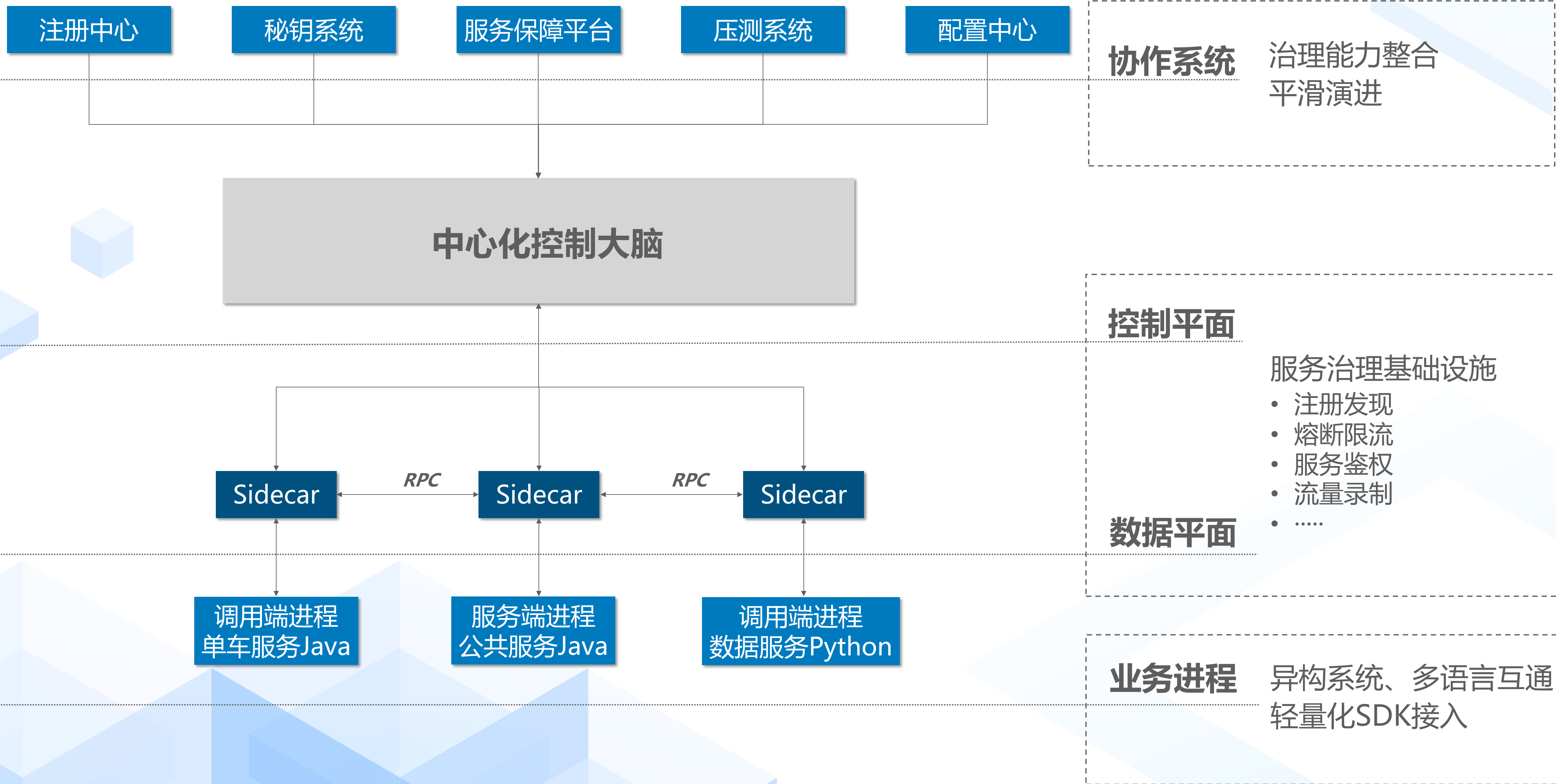
新业务：基础组件分散化使用

## 运维难

组件收敛：版本分散，难以收敛

问题排查：复杂问题业务进程排查，费时费力

# Mesh 化架构



协作系统

治理能力整合  
平滑演进

控制平面

数据平面

业务进程

服务治理基础设施

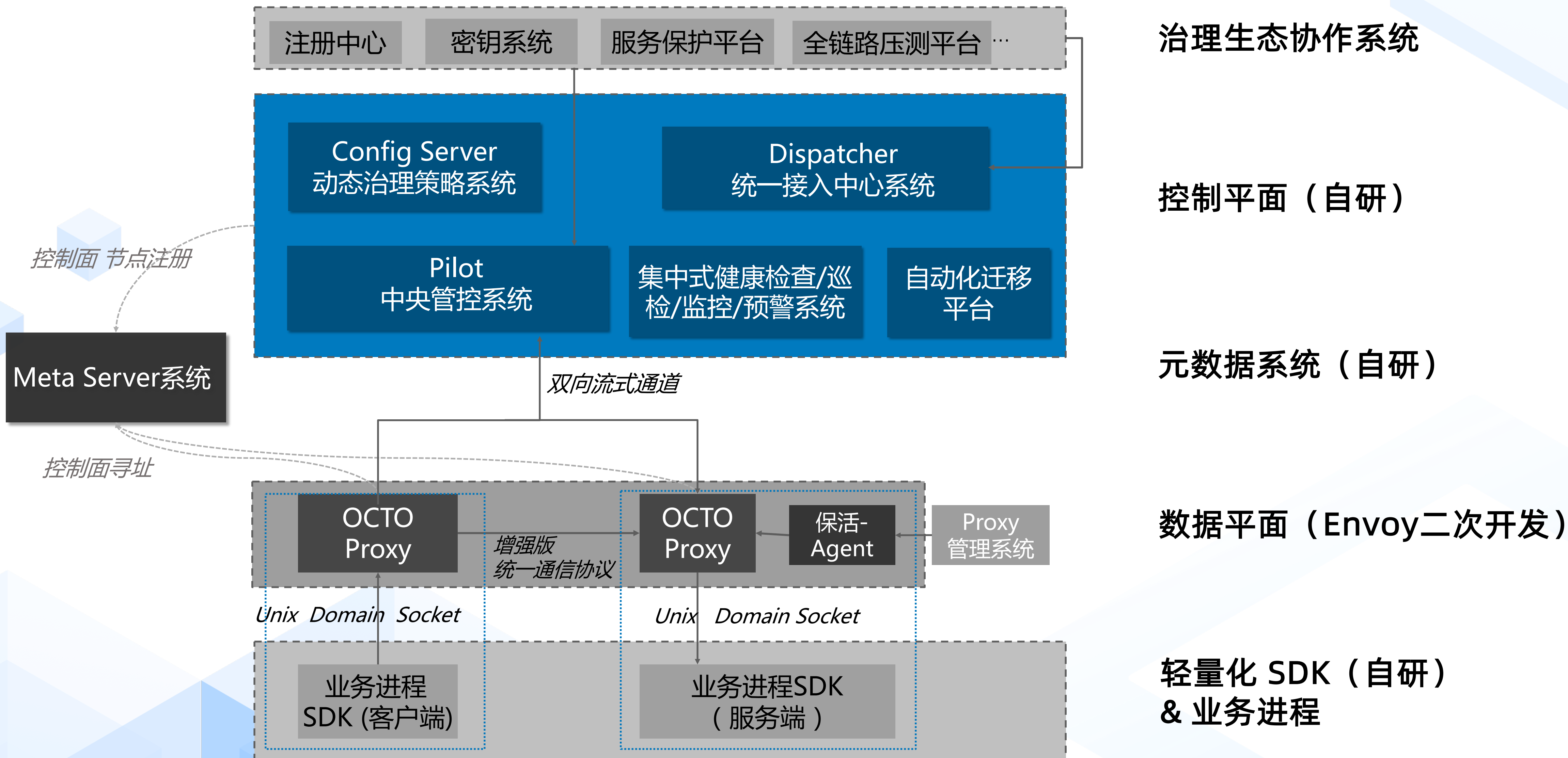
- 注册发现
- 熔断限流
- 服务鉴权
- 流量录制
- .....

异构系统、多语言互通  
轻量化SDK接入

# Mesh 化架构

- 管控数十万数据平面节点?  
中央管控系统 Pilot
- 打通异构服务治理协作系统?  
统一接入中心 Dispatcher
- 支撑用户的平滑接入?  
动态治理策略系统 ConfigServer、自动化迁移平台 Migrator
- 保障系统的稳定性?  
中心化健康检查/监控/巡检/预警系统

# Mesh 化架构



## 02

# Service Mesh 落地的挑战与破局

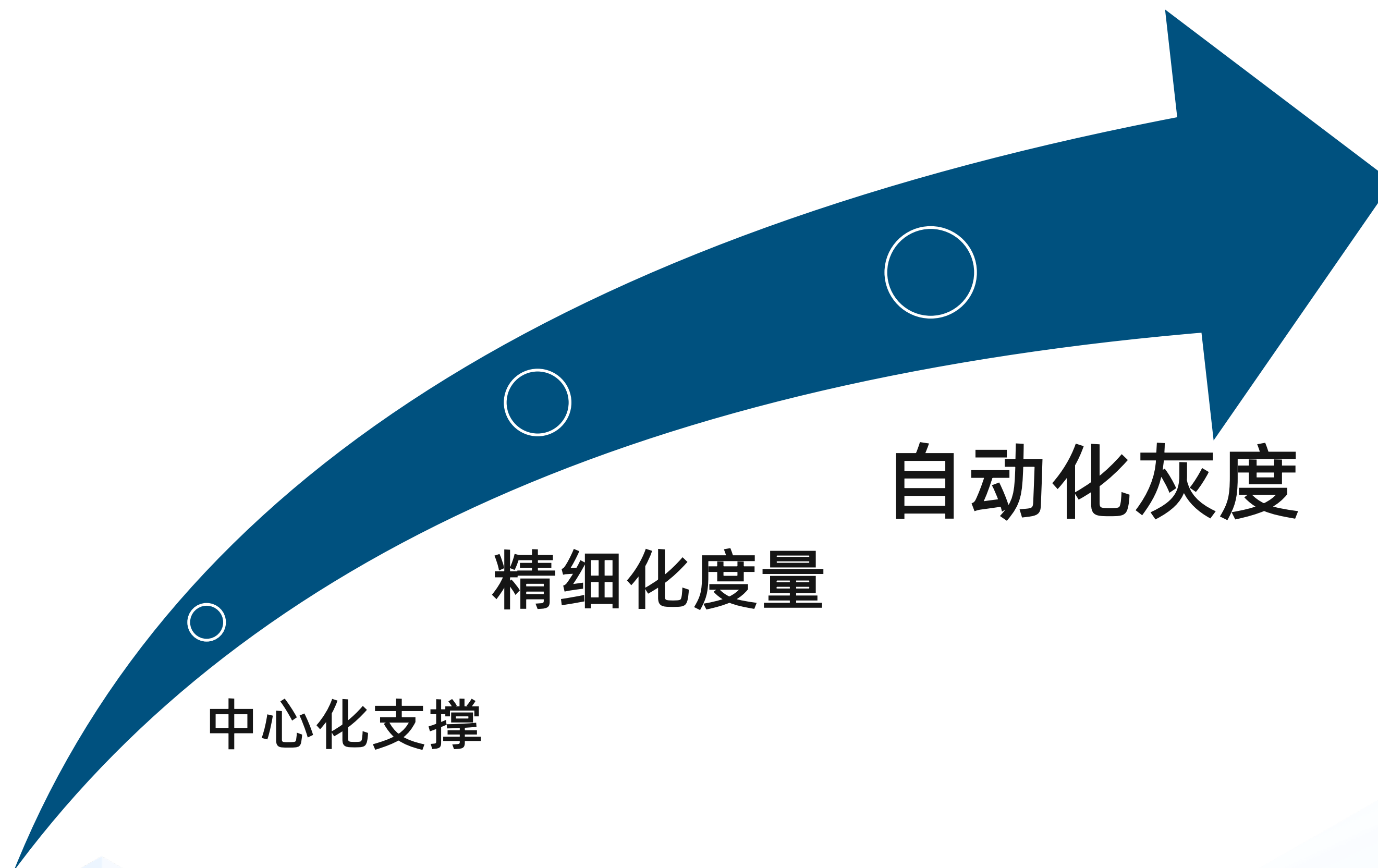
- 中心化支撑
- 精细化度量
- 自动化灰度

# 面临挑战

- 中心化管控治理  
扩展性、性能、统一化
- 业务价值导向，最小化接入成本  
产品价值 = (新体验 - 旧体验) - 替换成本
- 平滑完成基础设施升级  
兼顾 稳定、体验、效率

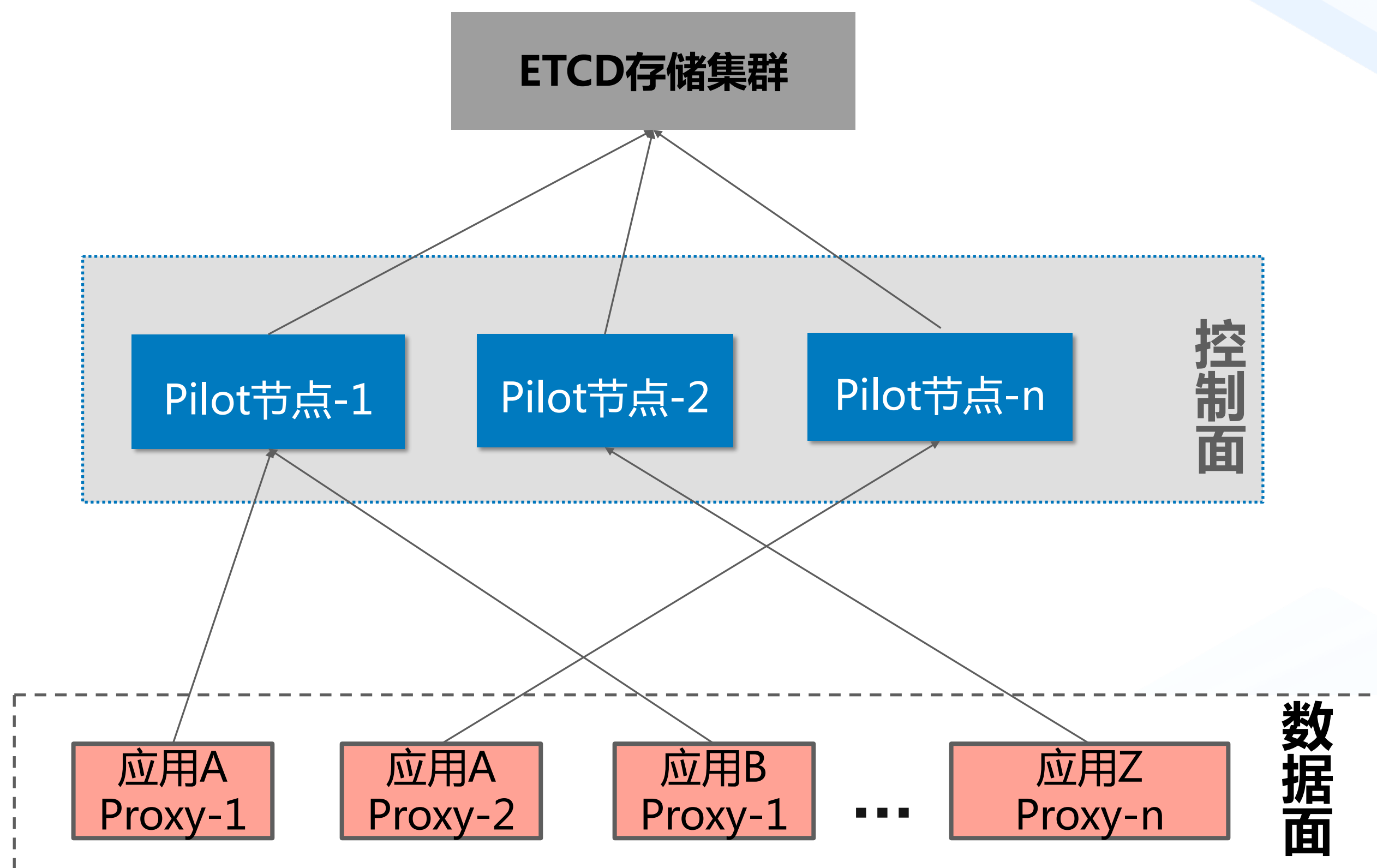


# Mesh 落地三步走



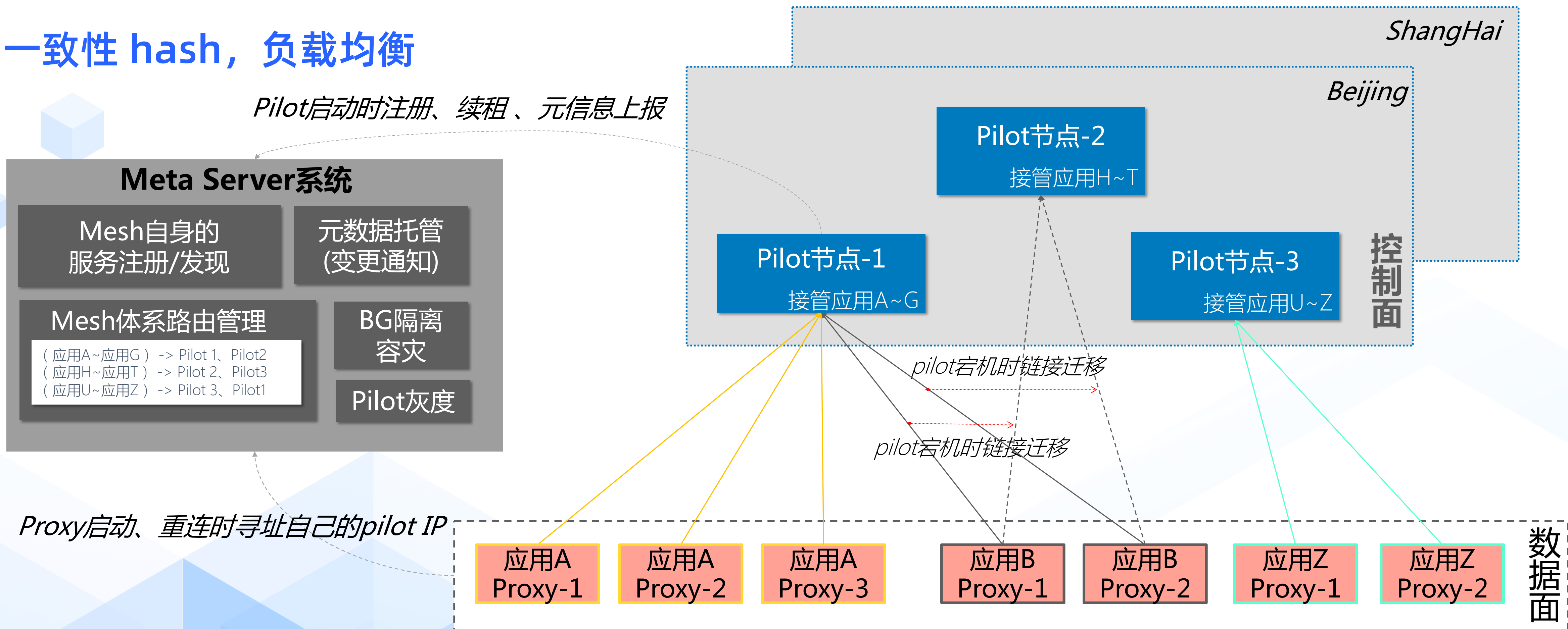
# 中心化支撑

- 全量数据，水平扩展
- IO 冗余，批量风暴



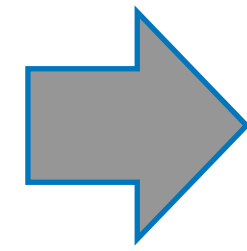
# 中心化支撑

- 横向数据分片，会话粘滞
- 服务分组，同地域优先
- 一致性 hash，负载均衡



# 中心化支撑

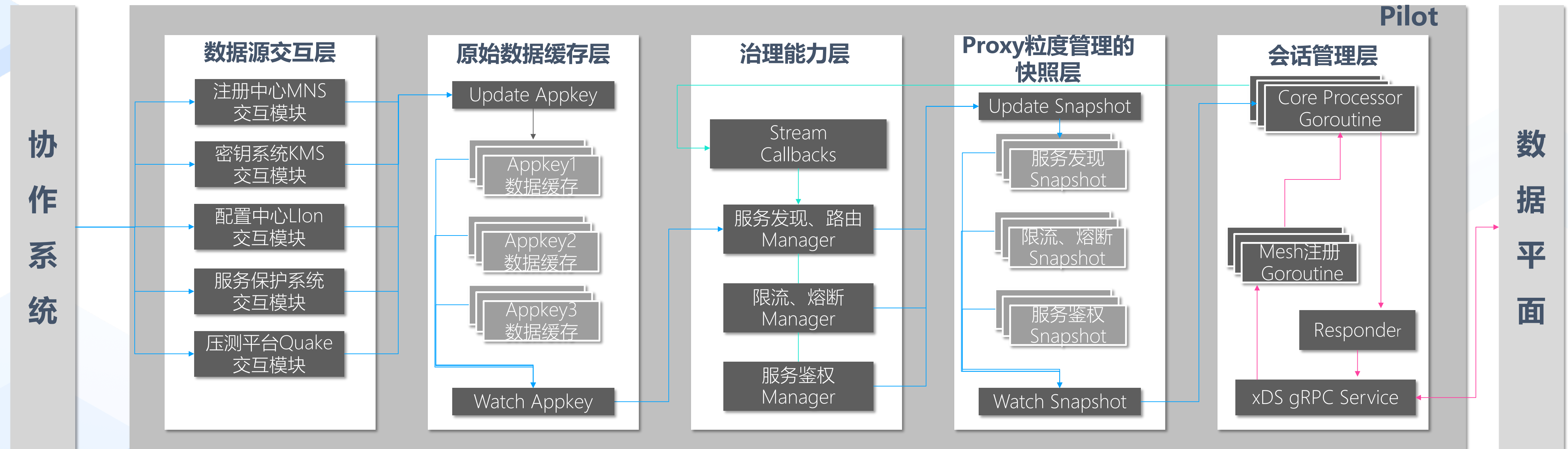
- 纵向分层订阅
- 减少 IO 冗余降低负载
- 类多路复用提高吞吐



- 有效应对批量风暴
- Pilot 单实例压测：6w+ 连接、tp99 < 2.3ms

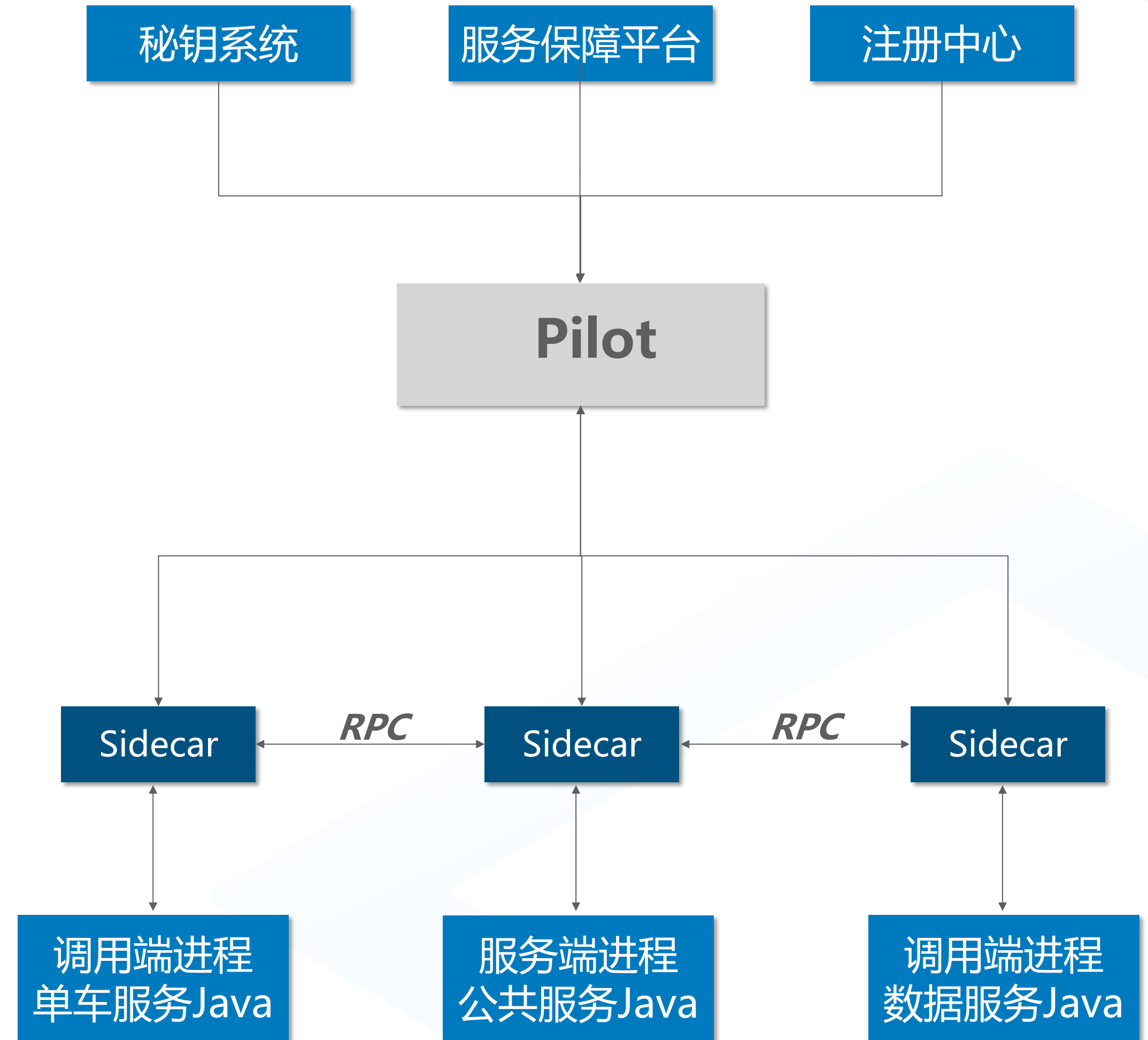
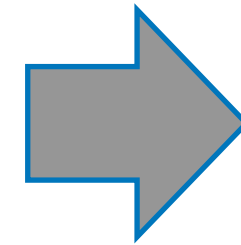
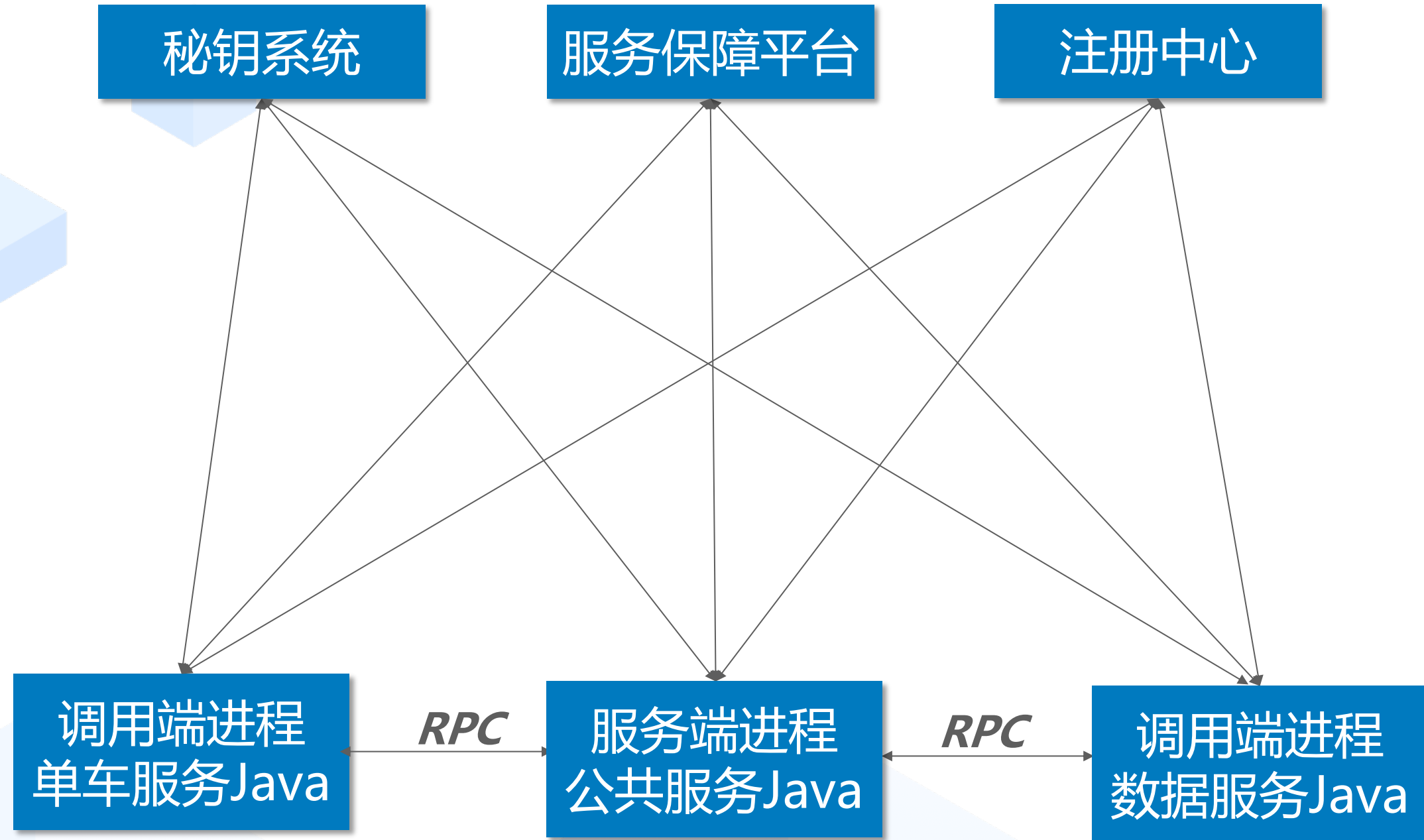
监听规则变更链路

数据下发链路



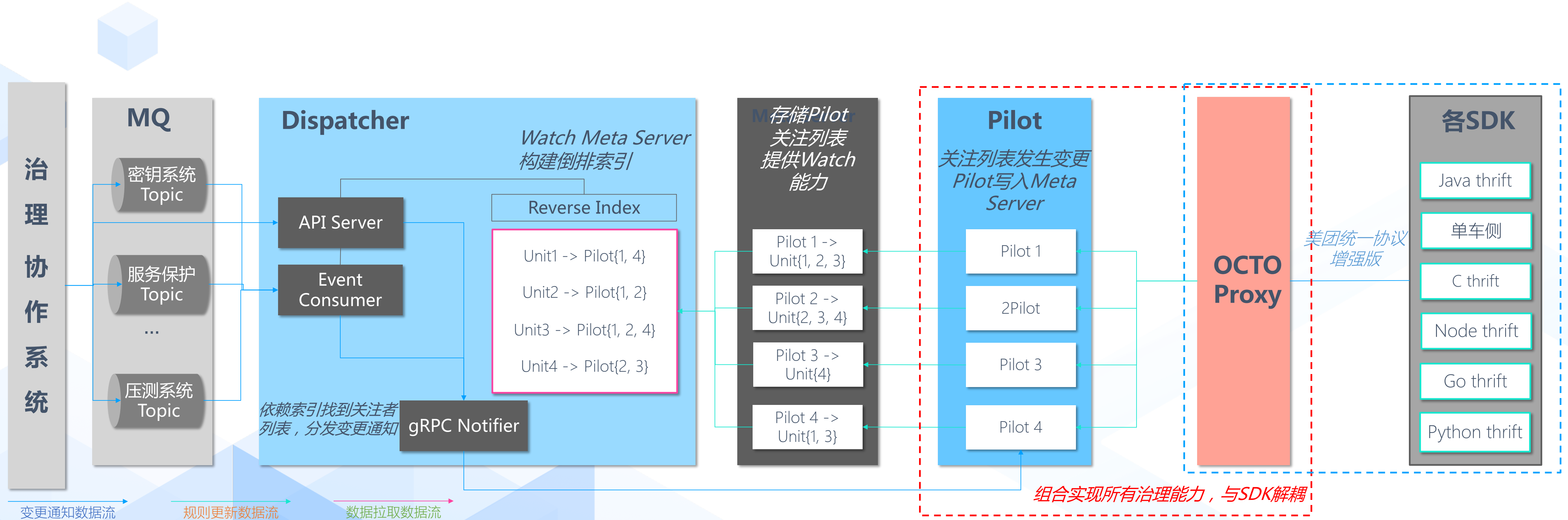
# 中心化支撑

分散化治理 -> 统一化治理



# 中心化支撑

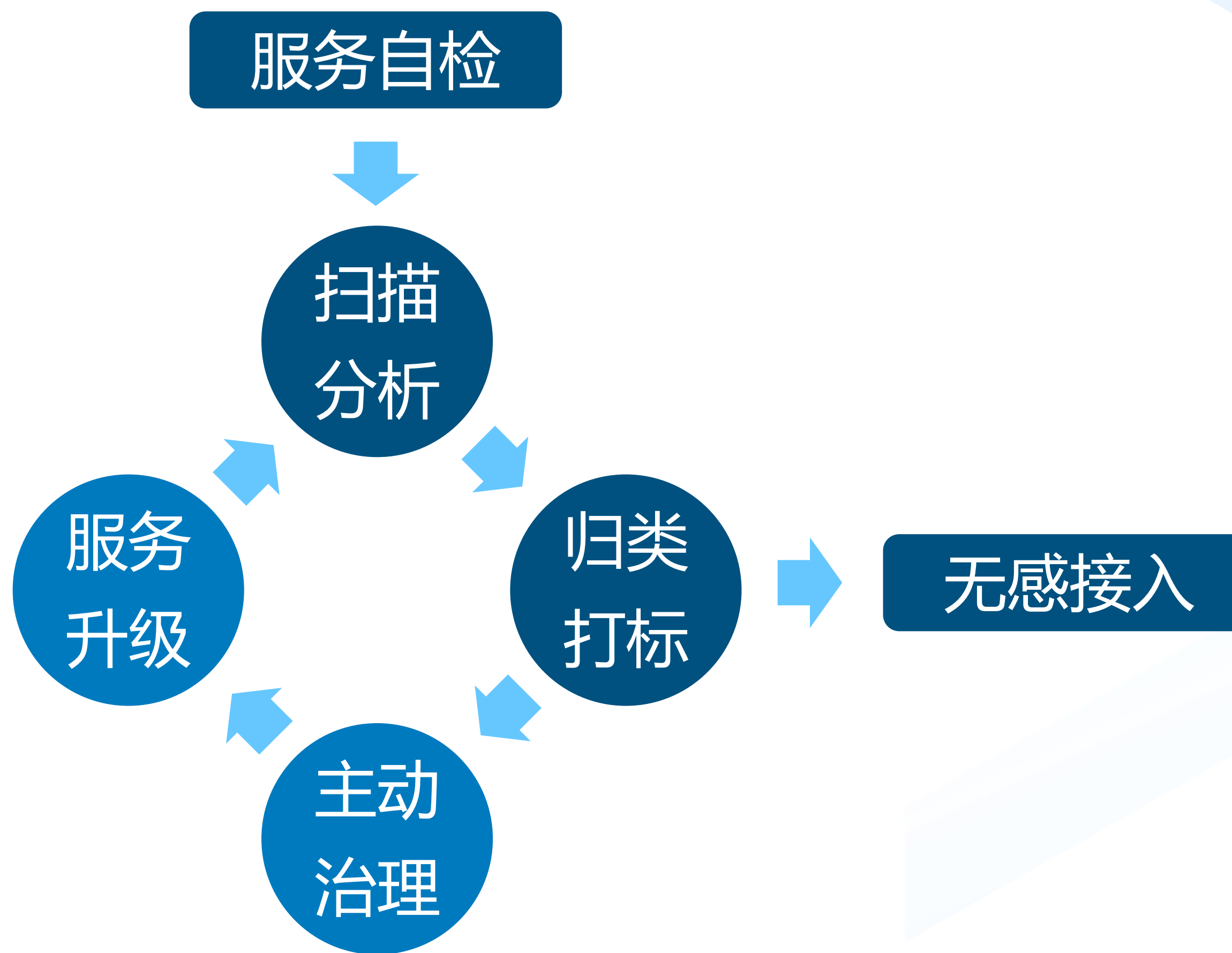
- 语种治理统一化
- 组件治理统一化
- 业务赋能统一化



# 精细化度量

## 1. 服务检测

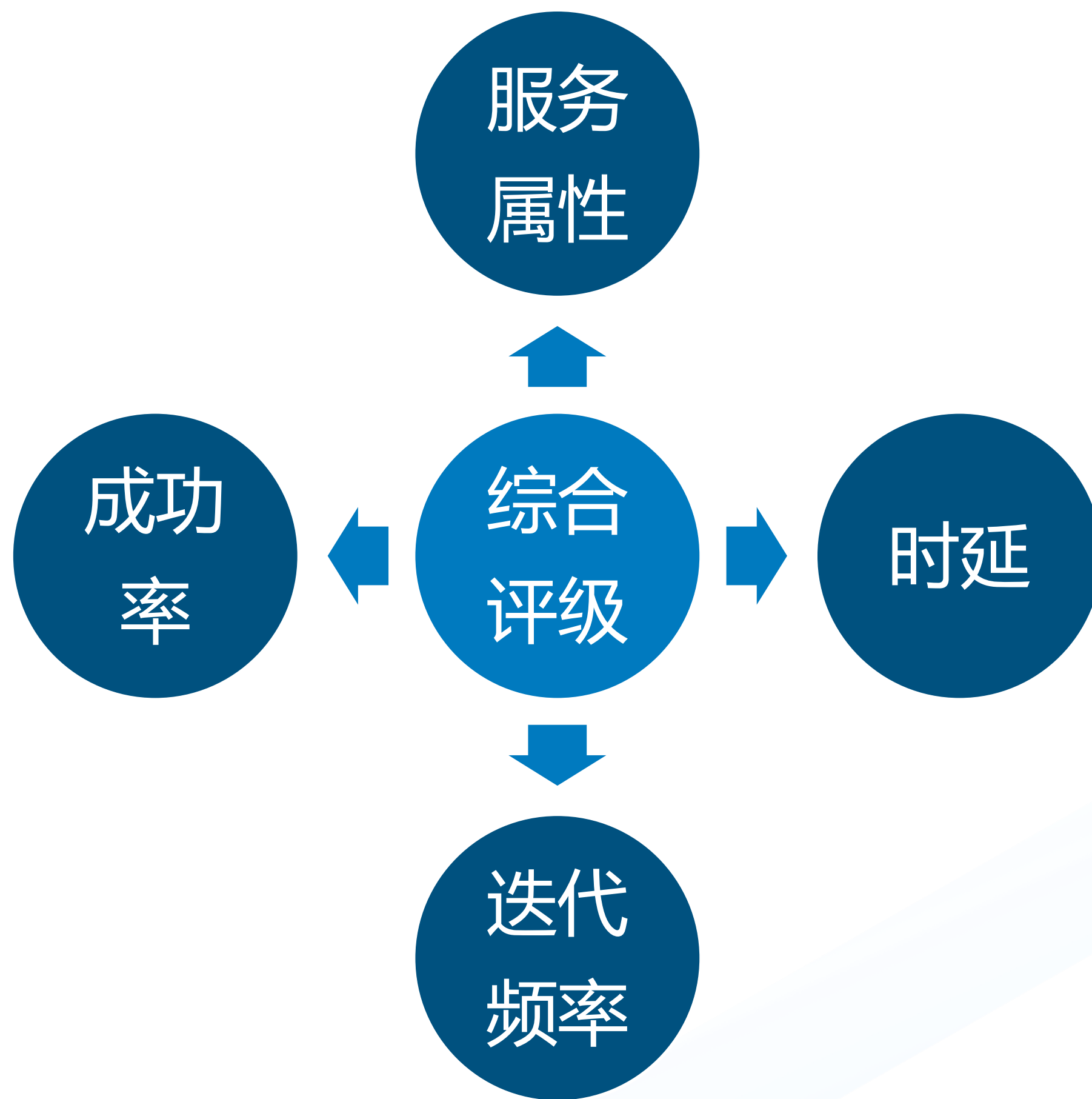
- 无感接入打标
- 有感接入治理



# 精细化度量

## 2. 综合评级

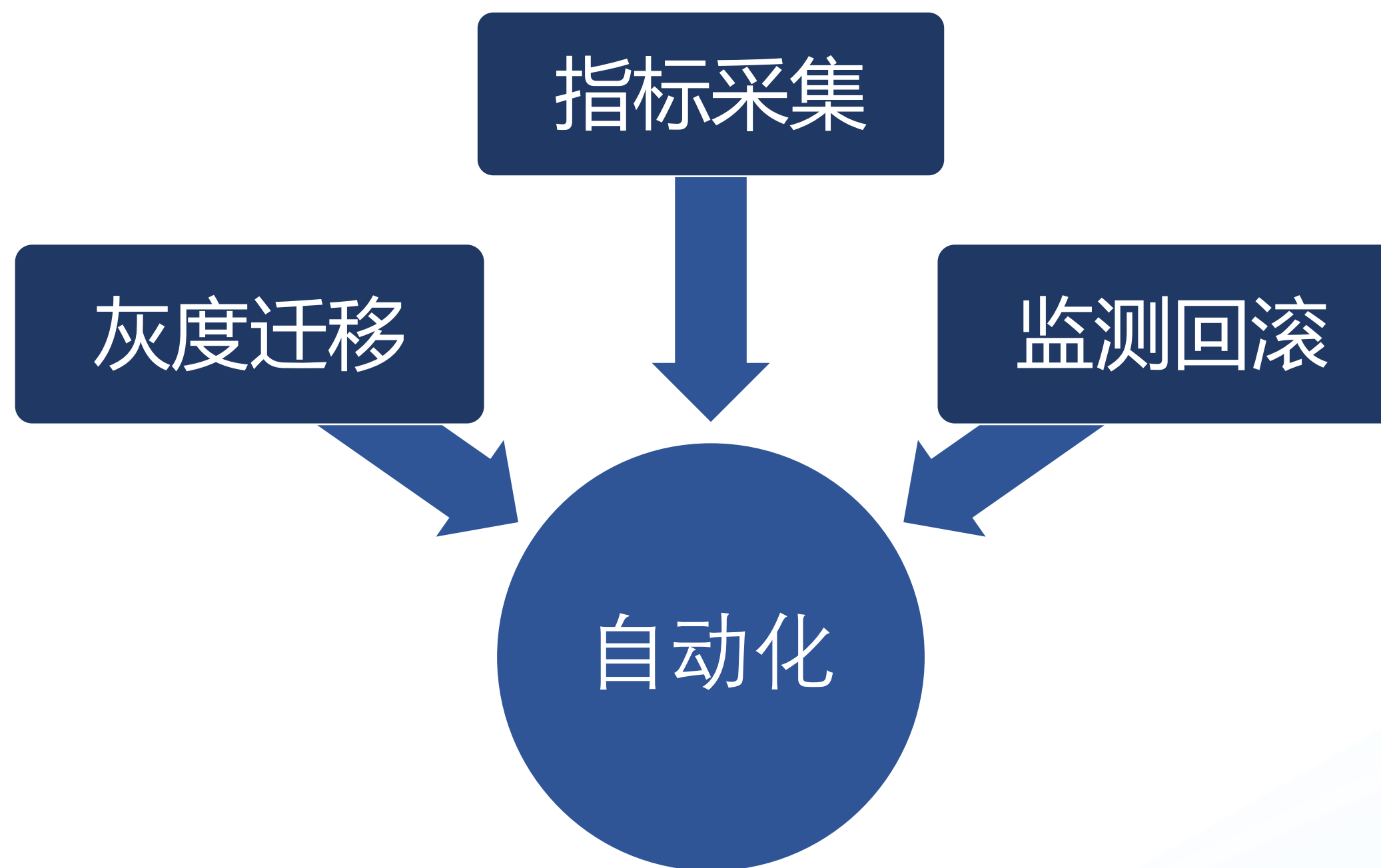
- 服务多维分级：  
服务属性、运行数据
- 自动化意愿收集





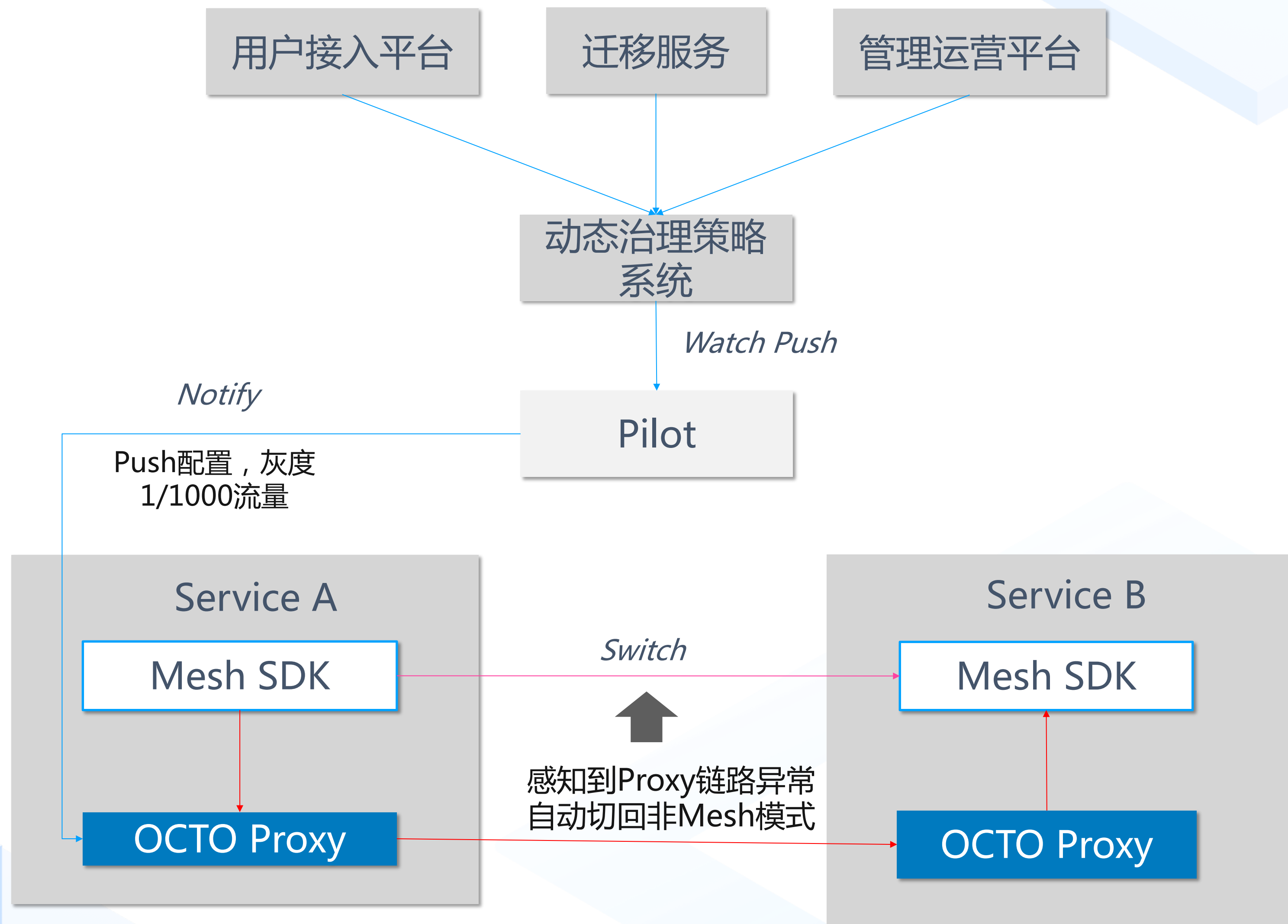
# 自动化灰度

- 可靠性：  
多重保护、数据驱动、多维评估
- 易用性：  
无感接入、快速接入
- 轻人力：  
自动化、平台化

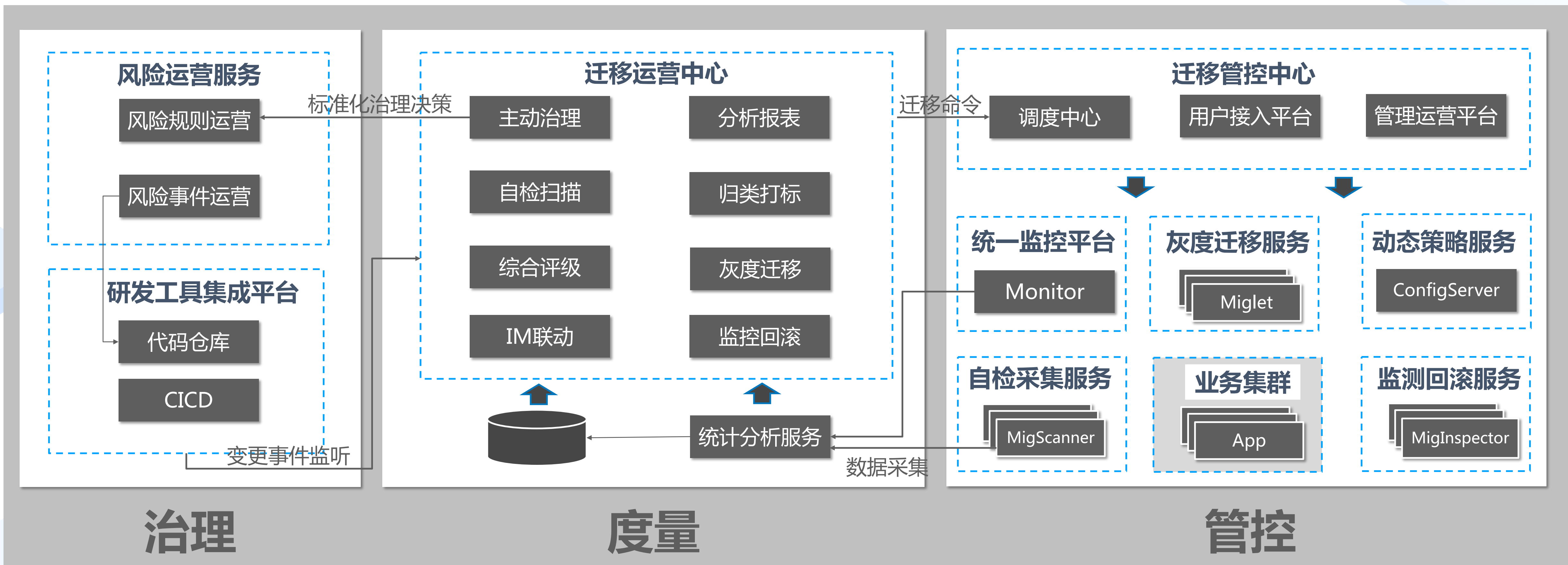


# 自动化灰度

- 灰度基础：  
动态治理策略系统
- 灰度升级：  
流量比例  
机器/组织维度  
调用角色
- 灰度保障：  
流量多级保护  
全链路巡检预警  
服务自保护



# 自动化迁移运营体系



## 03

### 总结与展望

- Service Mesh 落地原则、方法、实践
- 展望未来

# 总结与展望

- 美团服务治理现状

演进： 标准化 -> 易用性/高性能 -> 全方位 -> 跨地域容灾扩展

最新进展： Service Mesh（自研 + 开源二次开发）

- Service Mesh落地实践

原则： 平滑、简单

方法： 中心化支撑、精细化度量、自动化灰度

实践： 自动化迁移运营体系

# 总结与展望

- 未来已至，只是还未平均分布  
基础设施的未来 Cloud Native
- Service Mesh 不是目标，提效大型复杂应用的交付管控才是重点  
业务价值 matters
- 统一管控各类型流量是方向  
是否能全覆盖有待验证
- 中心化管控下更丰富的服务治理能力探索  
全局流量调度、服务安全等



THANKS.





# MOSN 社区周年回顾与展望

田阳 (烈元)  
MOSN Maintainer





**01**

**MOSN 历程**

**02**

**周年回顾**

**03**

**展望未来**

# 1.MOSN 历程

# MOSN 历程



**- 天猫双11 -**

## 支付宝新技术最佳演练场

**2019**

Service Mesh 搭建的全球最大金融级云原生集群首次登上大促舞台，保障支付宝和淘宝双端联动。支付宝技术出海赋能，6个基于支付宝技术的海外钱包首次服务全球电商。

通过融合计算引擎，协同流、图、并行计算、机器学习等不同计算模式，在支付过程中提供秒级智能决策能力。

随着图计算在花呗，蚂蚁森林等场景中大规模上线，图数据库Gea-base突破万亿边，在线图分析百亿关系特征计算秒级响应。

**2018**

区块链技术第一次全面参战，百余个国家和地区、1.5亿跨境商品的原产地溯源不再是难题。

生物支付首次成为双11主流支付方式，每10笔支付就有6笔，标志着密码正退出历史舞台，中国的生物支付时代来临。

支持双11的核心技术100%对外开放，帮助金融机构进行数字化转型，更加专注自身业务创新，创造属于自己的“双11”。

基于新计算框架的融合引擎首次上线花呗场景，计算模式正式开始发生改变。

2018年3月

MOSN 在蚂蚁诞生  
支持Service Mesh



2019年双十一

核心支付链路  
全覆盖

2019年12月28日

MOSN 宣布独立运营  
CNCF landscape



2020年7月  
Istio官方推荐数据平面

2020年6月  
V0.13.0 发布，进行云原生组件生态融合

2020年10月  
开源holmes问题诊断平台

2020年12月  
支持商业化落地  
江西农信Mesh  
阿里云CDN



## 2.周年回顾

- ✓ 社区建设
- ✓ 开源生态
- ✓ 核心能力

# 社区建设 - 开启开源社区

官网: <https://mosn.io>

github: <https://github.com/mosn>



# 社区建设 - 社区活动

## 技术分享

[基于 MOSN 和 Istio Service Mesh 的服务治理实践](#)

[Istio + MOSN 在 Dubbo 场景下的探索之路](#)

[云原生网络代理 MOSN 的扩展机制解析](#)

[云原生网络代理 MOSN 的多协议机制解析](#)

[Service Mesh 在 CDN 边缘场景的落地实践](#)

[阿里云 CDN 节点微服务架构演进之路](#)

[Go 开源说第五期：MOSN Go语言网络代理软件](#)

[云原生网络代理 MOSN 的进化之路](#)

[Dubbo 基于 MOSN 在 Service Mesh 场景下的落地实践](#)

[再启程，Service Mesh 前路虽长，尤可期许](#)

## MOSN 教学视频

<p>MOSN with Istio</p> <p>Run the Bookinfo sample using MOSN as data plane with Istio.</p> <p>Start Scenario</p>	<p>Istio with MOSN adapt dubbo</p> <p>Dubbo demo run with MOSN in Istio env.</p> <p>Start Scenario</p>	<p>MOSN with SkyWalking</p> <p>Use SkyWalking as a trace implementation.</p> <p>Start Scenario</p>	<p>MOSN with Istio Circuit Breaker</p> <p>Running Circuit Breaker using MOSN with Istio.</p> <p>Start Scenario</p>
--	--	--	--

## MOSN源码分析活动

MOSN 源码解析系列 [event/source-code-analyze](#) [kind/event](#)

#32 by rootsongjc was closed on 8 May 2020

MOSN 源码解析：HTTP系能力 [event/source-code-analyze](#)

#31 by rootsongjc was closed on 27 Mar 2020

MOSN 源码解析：连接池 [event/source-code-analyze](#)

#30 by rootsongjc was closed on 7 Apr 2020

MOSN 源码解析：XDS [event/source-code-analyze](#)

#29 by rootsongjc was closed on 21 Feb 2020

MOSN 源码解析：log系统 [event/source-code-analyze](#)

#28 by rootsongjc was closed on 10 Mar 2020

MOSN 源码解析：内存复用机制 [event/source-code-analyze](#)

#27 by rootsongjc was closed on 16 Mar 2020

MOSN 源码解析：协程模型 [event/source-code-analyze](#)

#26 by rootsongjc was closed on 16 Mar 2020

MOSN 源码解析：Plugin 机制 [event/source-code-analyze](#)

#25 by rootsongjc was closed on 14 Feb 2020

MOSN 源码解析：共享内存模型 [event/source-code-analyze](#)

#24 by rootsongjc was closed on 5 Mar 2020

MOSN 源码解析：变量机制 [event/source-code-analyze](#)

#23 by rootsongjc was closed on 21 Feb 2020

## 云原生标准 Sidecar

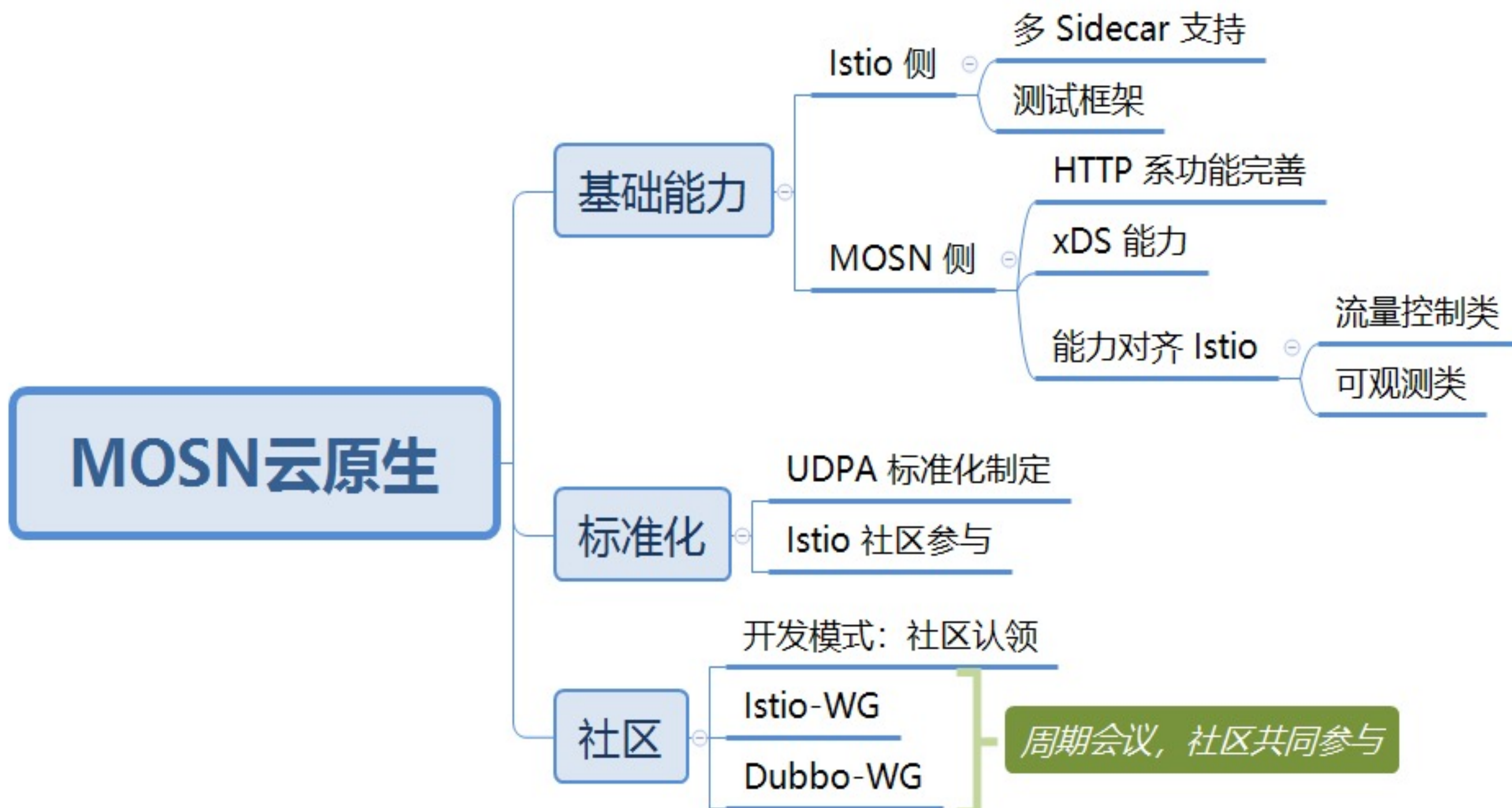
- Istio 侧多 Sidecar 支持
- MOSN 侧功能对齐 Istio

## 标准化

- 参与 UDPA 讨论及标准制定

## 社区推广

- 建立 [Istio-WG](#)
- 多开发者参与 [共建](#)



Type	Feature	Description	Application scenarios	Pb on the Istio	priority	issue	status
Load balance	original_dst	Forward the request based on t	In the Istio transparent hijacking scenario, if no matching	<a href="https://github.com/envoyproxy/go-control">https://github.com/envoyproxy/go-control</a>	p0	<a href="https://github.com/mosn/mosn/issues/982">https://github.com/mosn/mosn/issues/982</a>	Done
	WRR	weight round robin	Load balancing according to machine weight	<a href="https://github.com/envoyproxy/go-control">https://github.com/envoyproxy/go-control</a>	3/30/2020	<a href="https://github.com/mosn/mosn/pull/1161">https://github.com/mosn/mosn/pull/1161</a>	Done
	least req	Choose the machine with the lo	Load balancing according to the number of machine requ	<a href="https://github.com/envoyproxy/go-control">https://github.com/envoyproxy/go-control</a>	p2	<a href="https://github.com/mosn/mosn/issues/1054">https://github.com/mosn/mosn/issues/1054</a>	Done
	Maglev	Consistent hash based on the r	Keep the session	<a href="https://github.com/envoyproxy/go-control">https://github.com/envoyproxy/go-control</a>	p2	<a href="https://github.com/mosn/mosn/pull/1118">https://github.com/mosn/mosn/pull/1118</a>	Done
Protocol	WebSocket	Implement WebSocket protocol	WebSocket proxy forwarding		p3		
	UDP/QUIC	support UDP/QUIC	proxy forwards the UDP/QUIC request		p3	<a href="https://github.com/mosn/mosn/issues/1062">https://github.com/mosn/mosn/issues/1062</a>	WIP
xDS	xDS adapter	Adapt to Pilot's xDS API	Manage MOSN by Istio		p0	<a href="https://github.com/mosn/mosn/tree/feature-istio_ad">https://github.com/mosn/mosn/tree/feature-istio_ad</a>	Done
Service governance	Fault Injection	Injecting an delay or abort fault	Simulate the application of exception scenarios, can inject	<a href="https://github.com/envoyproxy/go-control">https://github.com/envoyproxy/go-control</a>	p0	<a href="https://github.com/mosn/mosn/pull/1063">https://github.com/mosn/mosn/pull/1063</a>	Done
	limit req	limit request	limiting protection	<a href="https://github.com/envoyproxy/go-control">https://github.com/envoyproxy/go-control</a>	p1	<a href="https://github.com/mosn/mosn/pull/1111">https://github.com/mosn/mosn/pull/1111</a>	Done
	Passive health	If the number of errors exceeds	Passive health check, disable abnormal service	<a href="https://github.com/envoyproxy/go-control">https://github.com/envoyproxy/go-control</a>	p0	<a href="https://github.com/mosn/mosn/issues/1055">https://github.com/mosn/mosn/issues/1055</a>	Done
	Circuit Breaking	Limits on the maximum number	Services to protect	<a href="https://github.com/envoyproxy/go-control">https://github.com/envoyproxy/go-control</a>	p1	<a href="https://github.com/mosn/mosn/issues/395">https://github.com/mosn/mosn/issues/395</a>	WIP
	External Author	External Authorization	Request authentication based on external authentication	<a href="https://github.com/envoyproxy/go-control">https://github.com/envoyproxy/go-control</a>	p2	<a href="https://github.com/mosn/mosn/issues/1056">https://github.com/mosn/mosn/issues/1056</a>	
	Traceing	Service observability	Service observability		p1	<a href="https://github.com/mosn/mosn/issues/1027">https://github.com/mosn/mosn/issues/1027</a>	Done
	Traffic mirror	traffic mirror	traffic mirror	<a href="https://istio.io/latest/zh/docs/tasks/traffic-r">https://istio.io/latest/zh/docs/tasks/traffic-r</a>	p1		
	Enhance rewrite	regex_rewrite host_rewrite	Rewrite the request's uri and host etc.	<a href="https://github.com/envoyproxy/go-control">https://github.com/envoyproxy/go-control</a>	p1	<a href="https://github.com/mosn/mosn/issues/1191">https://github.com/mosn/mosn/issues/1191</a>	WIP
Framework extension	listener filter	Optimize the listener filter frame	Support listener filter registration		p1	<a href="https://github.com/mosn/mosn/pull/1014">https://github.com/mosn/mosn/pull/1014</a>	Done
	Strict DNS clus	Support STRICT_DNS type Clu	Service discovery by DNS	<a href="https://github.com/envoyproxy/go-control">https://github.com/envoyproxy/go-control</a>	p1	<a href="https://github.com/mosn/mosn/pull/1149">https://github.com/mosn/mosn/pull/1149</a>	Done
	Logical DNS cl	Dynamic resolve domain and us	Forward proxy by domain	<a href="https://github.com/envoyproxy/go-control">https://github.com/envoyproxy/go-control</a>	p1	<a href="https://github.com/mosn/mosn/pull/1005">https://github.com/mosn/mosn/pull/1005</a>	Done
Programmable capability	Lua	Support Lua	Support Lua	<a href="https://github.com/envoyproxy/go-control">https://github.com/envoyproxy/go-control</a>	p2		TBD
	Wasm	Support Wasm	Support Wasm		p2		TBD
others	gzip filter	Support gzip response body	Gzip		p2	<a href="https://github.com/mosn/mosn/pull/1151">https://github.com/mosn/mosn/pull/1151</a>	Done
	pilot-agent prot	Support serverinfo api	Pilot agent health check		p0	<a href="https://github.com/mosn/mosn/issues/1057">https://github.com/mosn/mosn/issues/1057</a>	Done
	istio.stats filter	Metrics support custom Lables	Metrics support custom Lables		p1	<a href="https://github.com/mosn/mosn/pull/1209">https://github.com/mosn/mosn/pull/1209</a>	WIP
	parameter exte	Parameter extension	Compatible with pilot starting mosn		p0	<a href="https://github.com/mosn/mosn/issues/1023">https://github.com/mosn/mosn/issues/1023</a>	Done
Istio	image build	Make proxyv2 image building oper	Build proxyv2 image for multiple data planes		p1	<a href="https://github.com/istio/istio/issues/24457">https://github.com/istio/istio/issues/24457</a>	WIP
	test framework	Modify istio test framework, make c	One test framework could testing multiple data planes		p1	<a href="https://github.com/istio/istio/issues/24463">https://github.com/istio/istio/issues/24463</a>	WIP



## UDPA 规范制定

- 限流 Proto
  - 限流 key 的定义
  - 观察者模式
  - 限流后的 Action 抽象
- 通用的 Router Proto
  - 不局限于 HTTP 系
  - 层级路由支持
  - 路由条件的可扩展性增强

## About the definition of general rate-limit rule API. #27

Open wangfakang opened this issue on Apr 22 · 2 comments



wangfakang commented on Apr 22 · edited · ...

I am investigating UDPA of the `rate-limit` related rules recently, and found a related discussion [here](#), I hope it could be discussed in UDPA WG. @lizan @htuch @mattklein123  
thx



htuch commented on May 12

Collaborator ...

@wangfakang I think this is a good topic to cover, I'll work on scheduling a UDPA-WG meeting in the near future and we can include this.

## Design Ideas

Extend the VirtualService API to provide meta routing configuration support:

```
...
router: # (Optional): list of attributes based routes for a protocol
  protocol: # (Required): string (or enum) of supported protocols
  # list of attribute based matcher(s) and destinations
  # all match conditions within a block have AND semantics
  # multiple match blocks have OR semantics
  - match: # (Required): attribute match conditions
    name: # (Optional): name of the match block
    attributes:
      - key:
        value: # One of the following
          StringMatch: # exact, prefix, regex (Existing Spec for StringMatch)
          set: # Any of the values in the set are considered a match
            - StringMatch
          map:
            - <string, StringMatch>
      route:
        - RouteDestination: # (Required) Existing RouteDestination spec
...

```

Istio 数据平面的另一个选择 —— MOSN

[ISTIO](#) / [BLOG](#) / [2020 POSTS](#) / [USING MOSN WITH ISTIO: AN ALTERNATIVE DATA PLANE](#)

# Using MOSN with Istio: an alternative data plane

A Cloud Native Proxy for Edge or Service Mesh

BY WANG FAKANG (MOSN.IO) |  JULY 28, 2020 |  6 MINUTE READ

**MOSN** (Modular Open Smart Network) is a network proxy server written in GoLang. It was built at [Ant Group](#) as a sidecar/API Gateway/cloud-native Ingress/Layer 4 or Layer 7 load balancer etc. Over time, we've added extra features, like a multi-protocol framework, multi-process plug-in mechanism, a DSL, and support for the [xDS APIs](#). Supporting xDS means we are now able to use MOSN as the network proxy for Istio. This configuration is not supported by the Istio project; for help, please see [Learn More](#) below.

# 开源生态 - proxy-wasm

## 背景

proxy-wasm 是一个通用标准规范，满足该规范的 WASM 模块，可以在不同 proxy 代理服务器中使用，实现跨语言的 proxy 模块共享。

- MOSN 支持 proxy-wasm 标准
- 贡献 proxy-wasm 的 Golang 实现，支持其他 golang proxy server 复用

<https://github.com/mosn/proxy-wasm-go-host>

mosn / proxy-wasm-go-host

<> Code

! Issues

🔗 Pull requests



github.com/proxy-wasm/spec

## WebAssembly for Proxies (ABI specification)

This repository contains specification of the low-level Application Binary Interface (ABI) and conventions to use between L4/L7 proxies (and/or other host environments) and their extensions delivered as WebAssembly modules.

The event-driven streaming APIs and convenient utility functions were originally developed for the [WebAssembly in Envoy](#) project, but they are proxy-agnostic, and consumers can use the same Proxy-Wasm extensions across different proxies.

### Implementations

#### SDKs

- [AssemblyScript SDK](#)
- [C++ SDK](#)
- [Go \(TinyGo\) SDK](#)
- [Rust SDK](#)

#### Host environments

##### Servers

- [Envoy](#)
- [Istio Proxy](#) (Envoy-based)
- [MOSN](#)
- [ATS](#) (proof-of-concept)

##### Libraries

- [C++ Host](#)
- [Go Host](#)

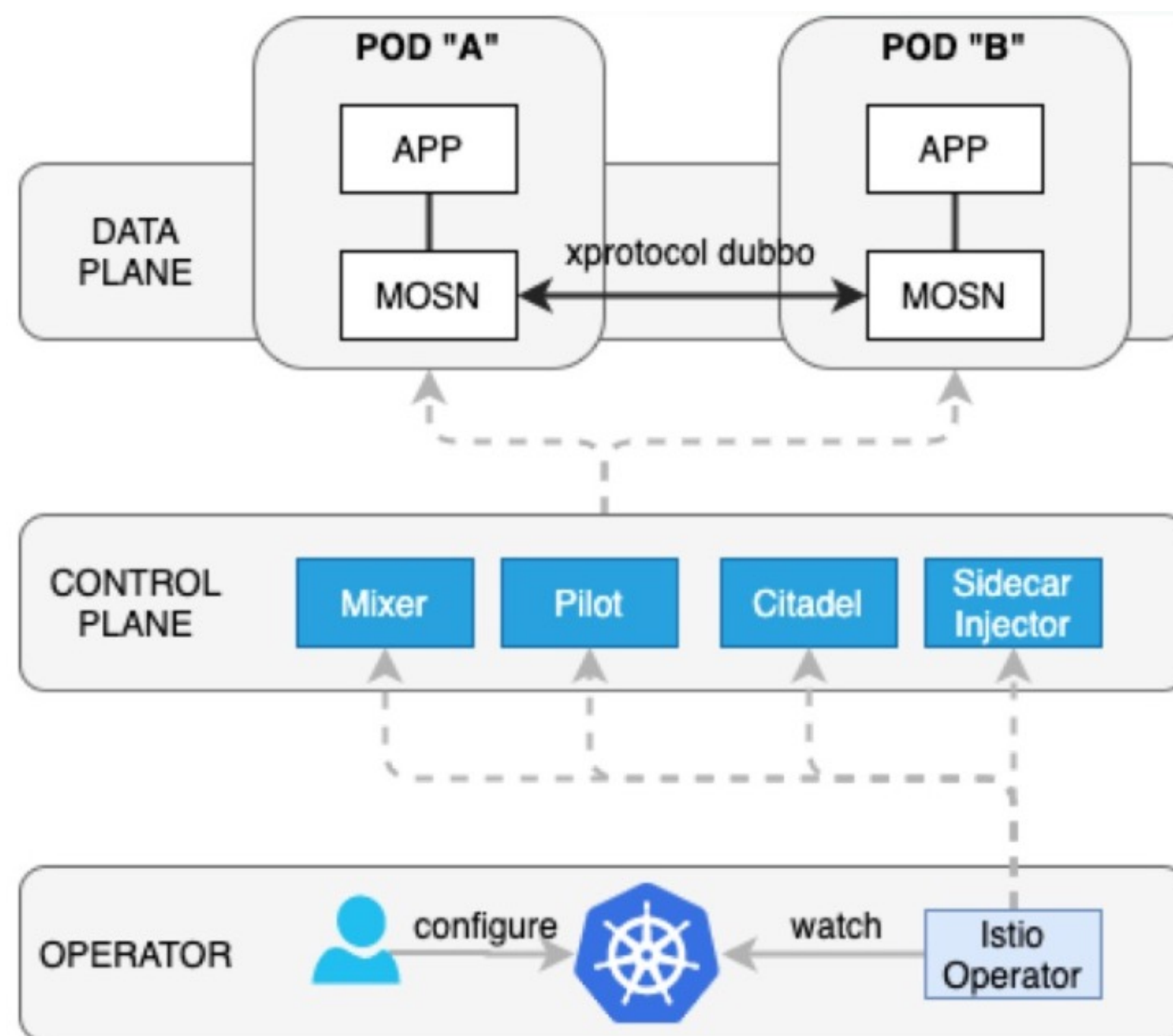
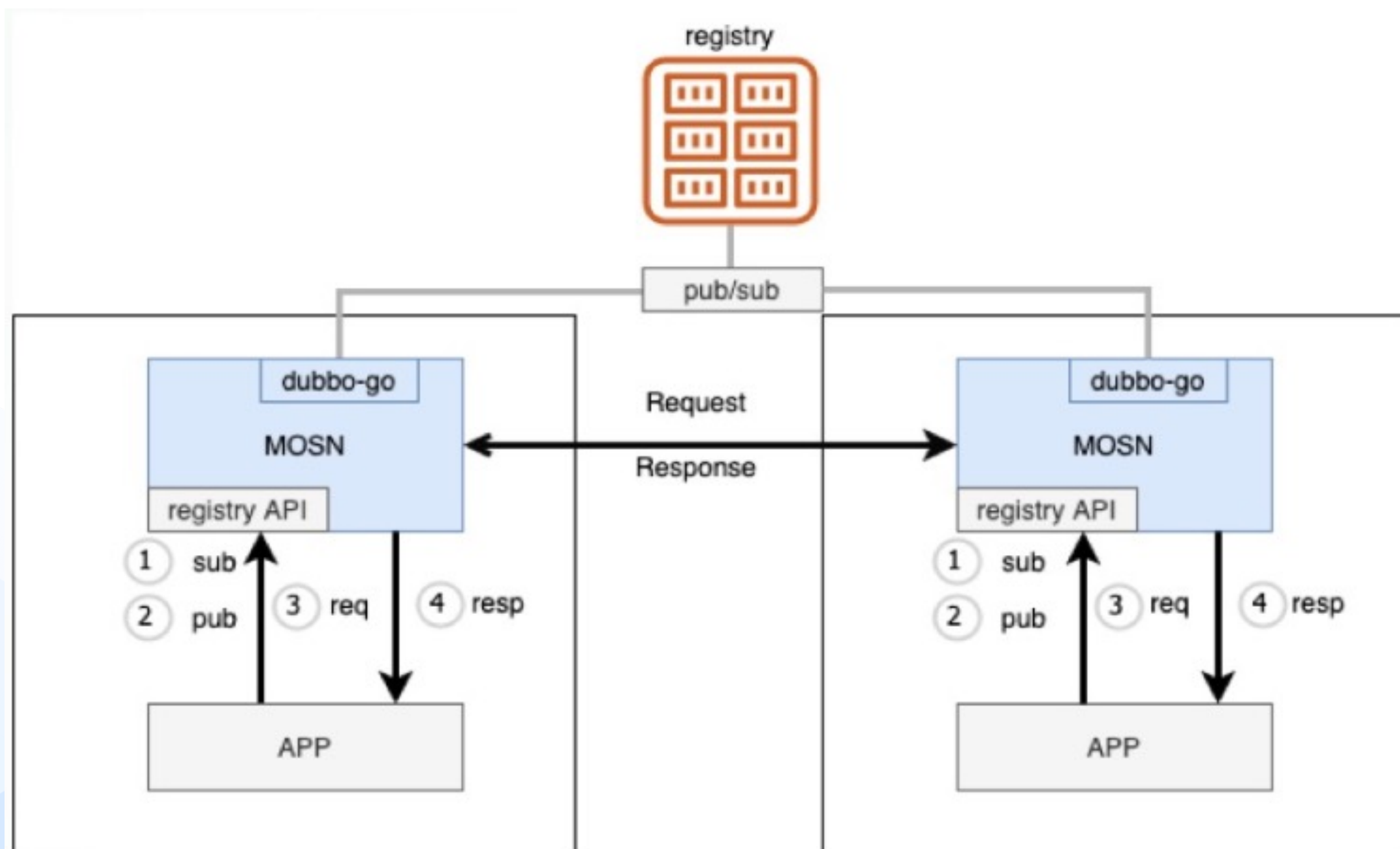
# 开源生态 -

## 业务诉求（微服务治理）

- 非 K8s 体系，历史包袱，双模架构
- 服务体系已 K8s 化，一步到位 Istio

## 解决方案

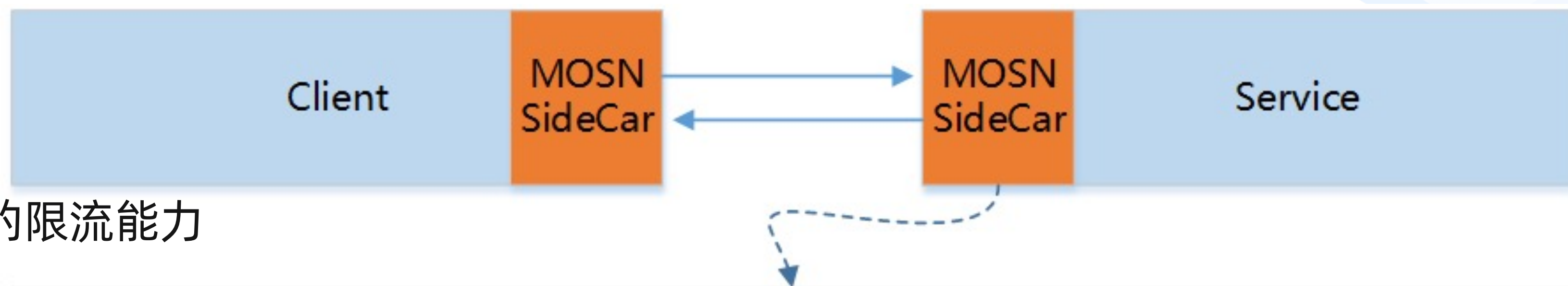
- MOSN 集成 dubbo-go 支持 pub/sub，复用原有的
- MOSN 支持 dubbo 在 Istio 下的路由



# 开源生态 - Sentinel

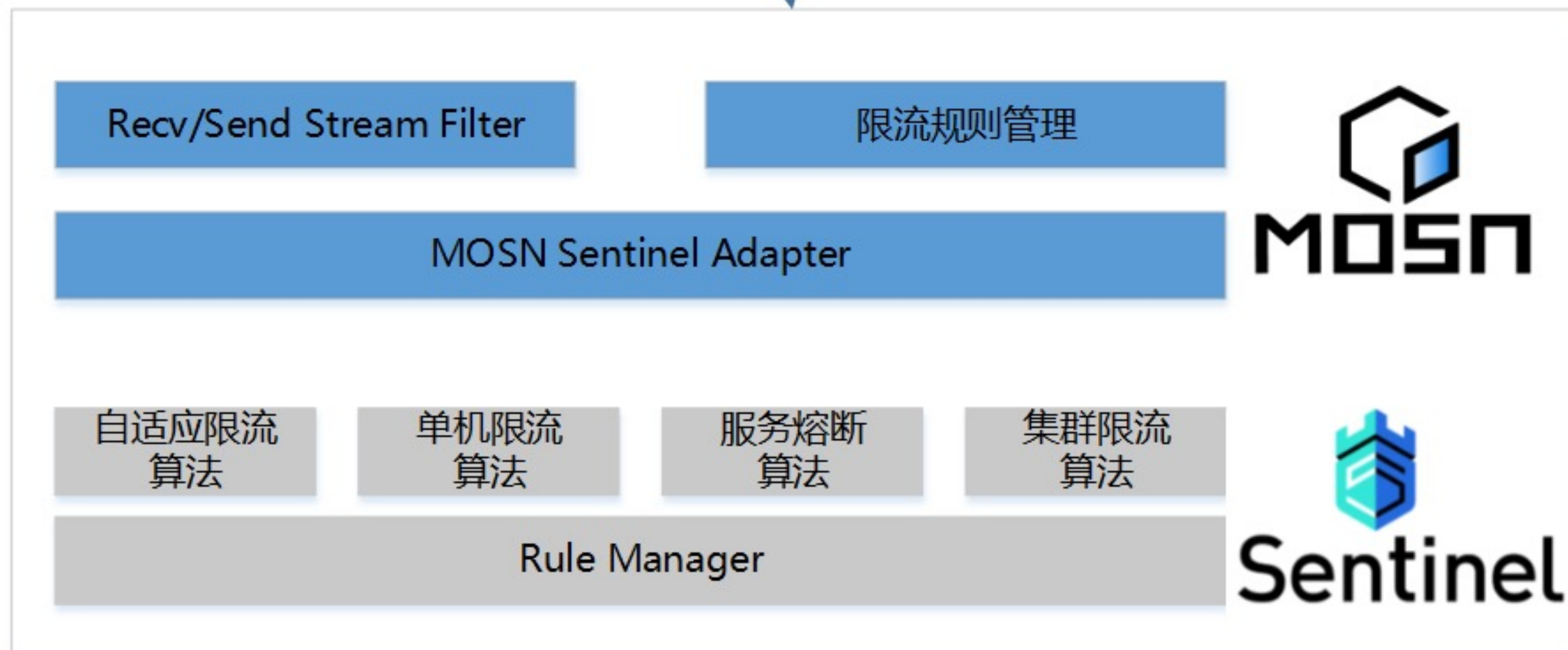
## 背景描述

- 限流是微服务治理之一
- MOSN 集成 Sentinel, 复用底层的限流能力



## 功能介绍

- 单机限流（令牌桶/漏桶结合）
- 服务熔断保护（服务成功率）
- 自适应限流（依据机器负载）



## 持续演进

- 限流算法丰富
- Istio 结合/UDPA 制定

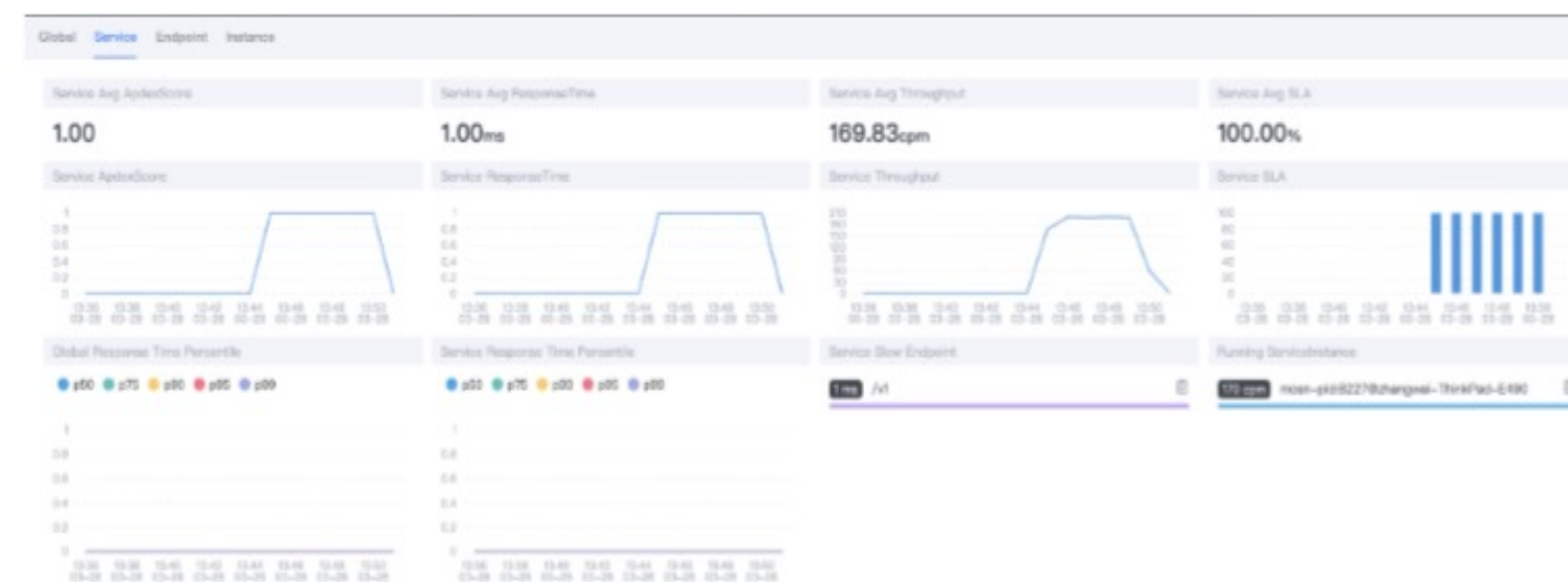
# 开源生态 - Skywalking

## 背景描述

- 调用依赖以及服务与服务之间的调用状态是微服务管理中一个重指标

## 功能介绍

- 调用链路拓扑展示
- QPS 监控
- 细粒度 RT 展示

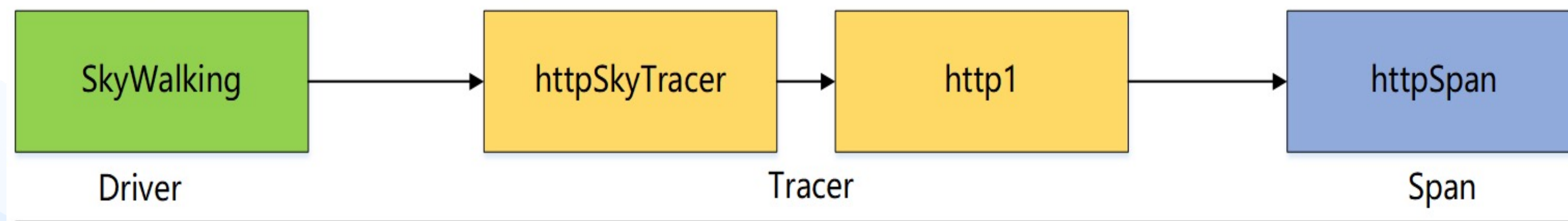


A screenshot of the SkyWalking trace view for a specific request. The URL is '/v1'. The start time is 2020-03-28 13:50:17, duration is 1ms, and spans are 4. The table below shows the execution flow:

Method	Start Time	Exec(ms)	Exec(%)	SelfTime	API	Service
/GET/v1	2020-03-28...	2	100		HttpClient	consumer-go
/v1	2020-03-28...	1	0		MOSN	main
/v1	2020-03-28...	1	0		MOSN	main
/v1	2020-03-28...	1	100		SpringMVC	provider-java

## 持续演进

- Dubbo Tracing 支持



# 开源生态 - holmes 在线诊断工具

## 背景

蚂蚁大规模落地遇到的实际问题

- 短时间内发生：抖了几秒，或者几分钟
- 无法拿到现场：发生之后程序已经崩溃/系统恢复

every 5 seconds



CPU util	Mem util	Goroutine
----------	----------	-----------

## 解决方案

1. 周期计算 CPU/RSS/Goroutine 平均值
2. 达到阈值，自动 dump 现场画像

CPU util ring	21	18	12	80	25	20	19	23	22	20
---------------	----	----	----	----	----	----	----	----	----	----

Mem util ring	21	18	12	25	25	20	19	23	80	20
---------------	----	----	----	----	----	----	----	----	----	----

goroutine ring	1400	1293	1388	1325	1322	8000	1363	1344	1400	1399
----------------	------	------	------	------	------	------	------	------	------	------

holmes

<https://github.com/mosn/holmes>

# 核心能力 - 扩展机制

## Filter 扩展机制

Network Filter

Proxy

TCPProxy

UDPProxy

Stream Filter

限流

鉴权

镜像

Listener Filter

透明劫持

## 多协议扩展

SOFA

Dubbo

Dubbo  
Thrift

Tars

msg

private

## 插件机制

多进程 Plugin 机制

WASM 扩展

So 动态库加载



# 核心能力 - 多协议扩展介绍

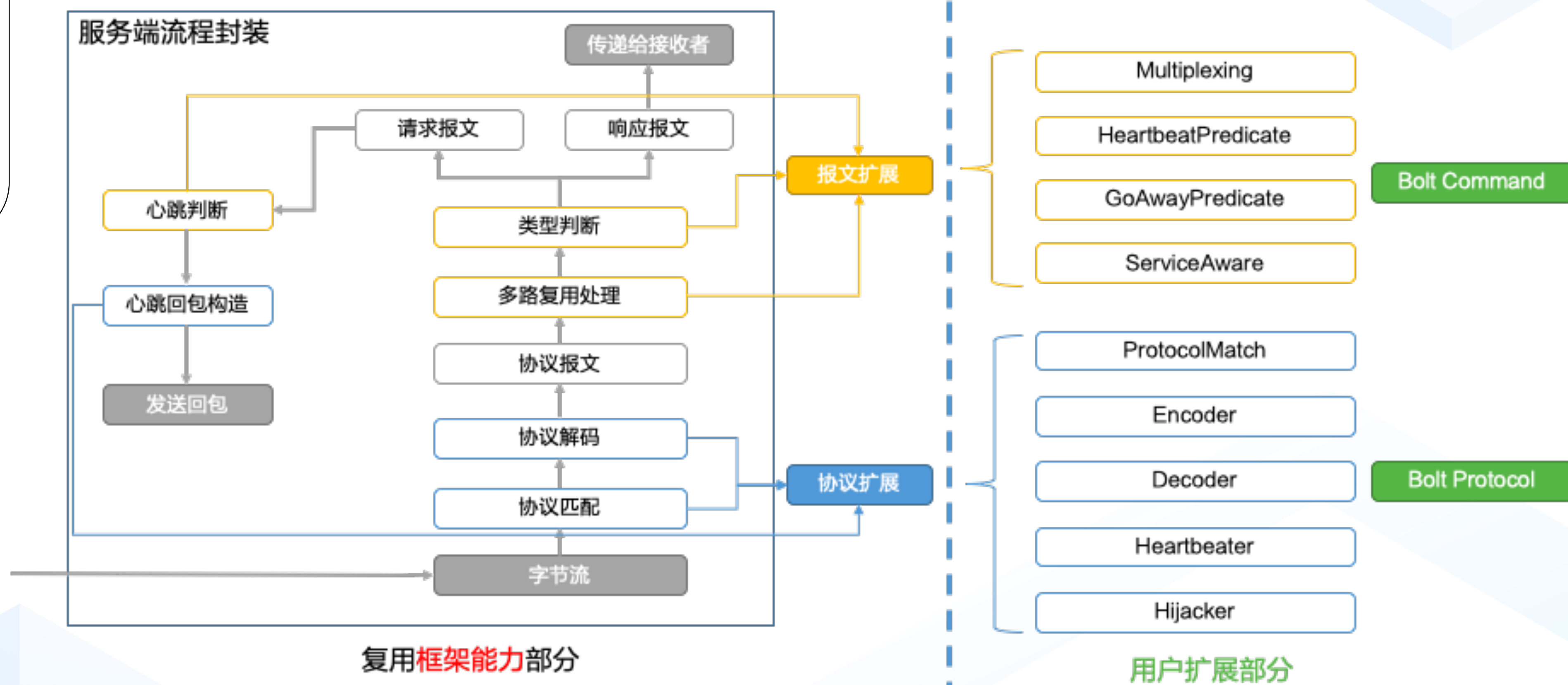
## 常规多协议框架

1. 框架生命周期管理
2. 连接类型 (如多路复用)
3. 连接池处理
4. 协议编码解析
5. 协议处理流程
6. 支持路由能力



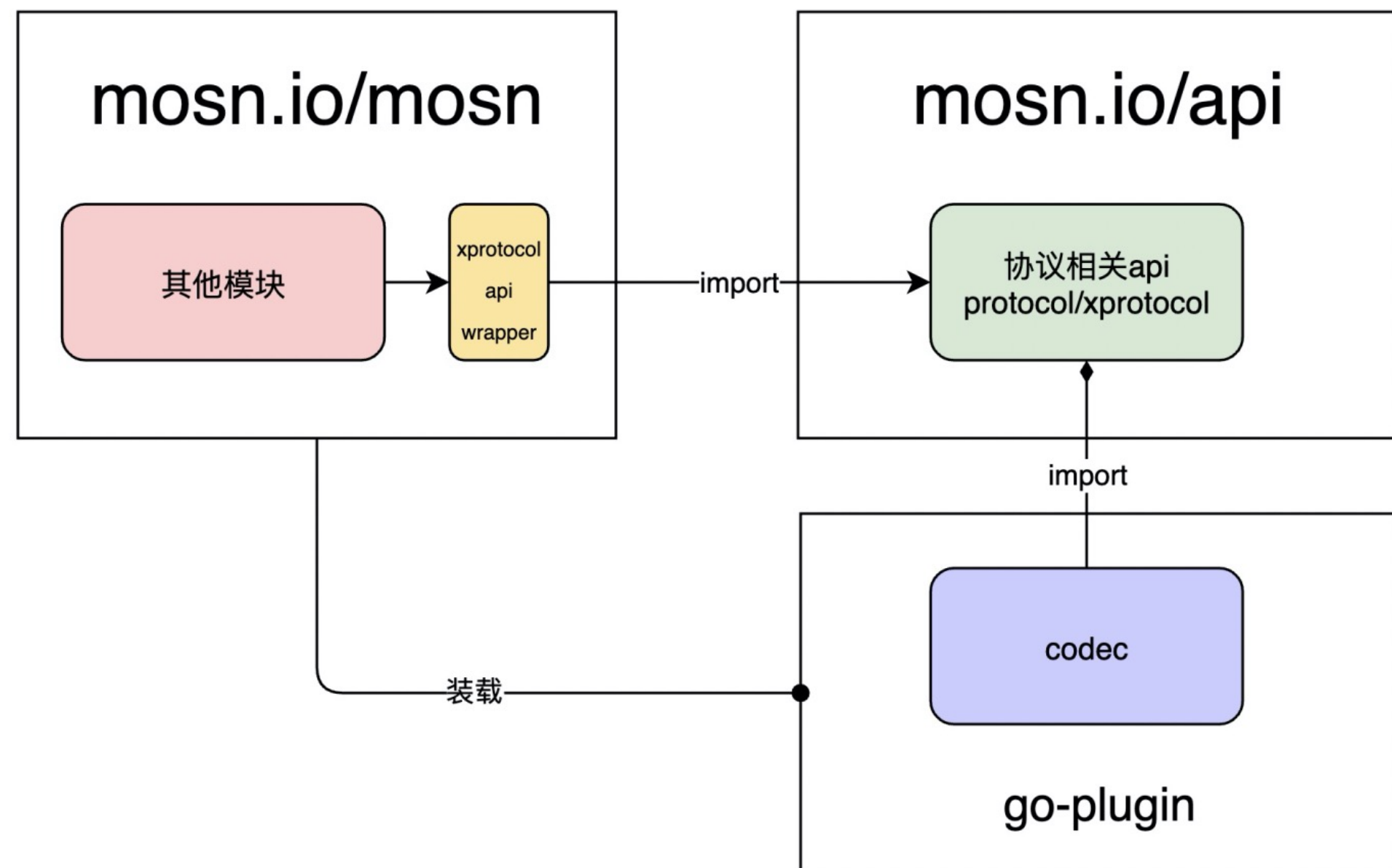
## MOSN多协议扩展框架

1. 协议编码解析

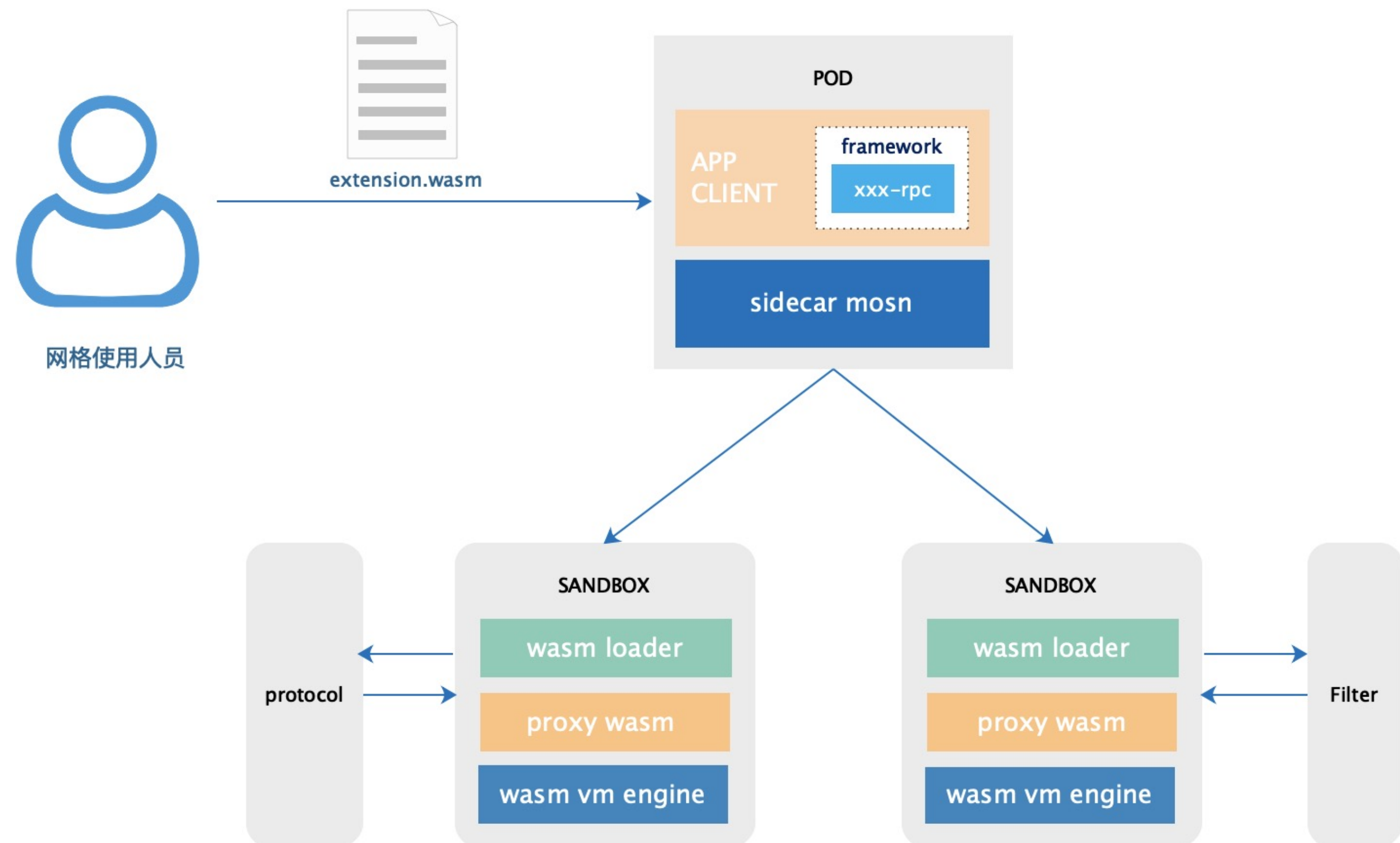


# 核心能力 - 扩展加载方式

## Go Plugin



## WASM



# 核心能力 - TLS优化

## 背景

蚂蚁全网加密项目的落地过程中，我们发现在大规模场景下，

MOSN 在 TLS 上占用的内存超出预期，频繁导致 OOM。

## 解决方案

通过具体分析，GO 的 TLS 实现会给每个连接预留一个 Buffer，

且不会缩容。

我们通过 sync.Pool 的方式进行优化，并反馈给了 Go 官方已接纳。

该优化在我们线上的场景会减少 50% 的内存占用。

golang / go

<> Code

Issues 5k+

Pull requests 289

Actions

Projects

### crypto/tls: pool Conn's outBuf to reduce memory cost of idle connections

Derived from CL 263277, which includes benchmarks.

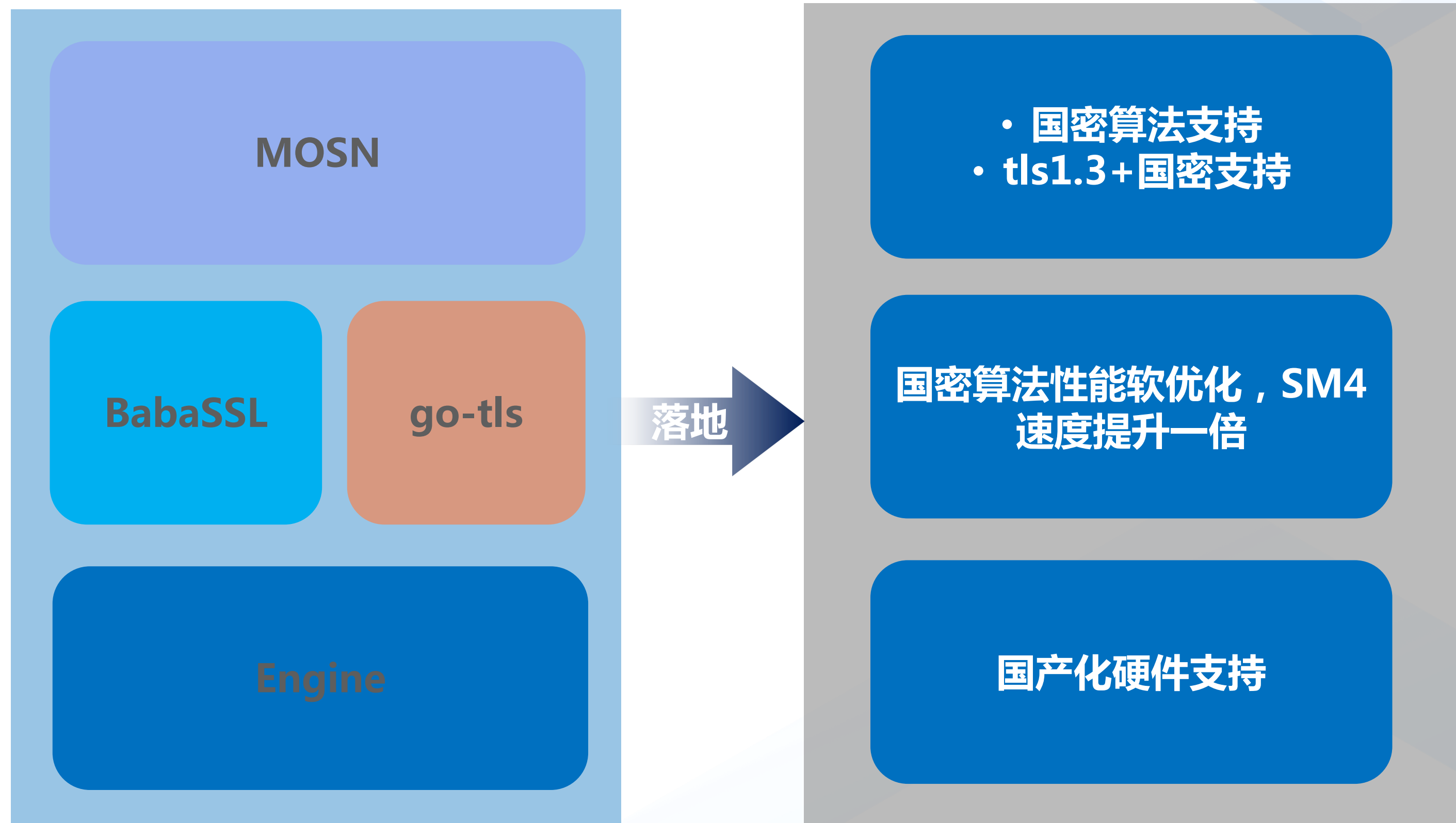
```
func (c *Conn) writeRecordLocked(typ recordType, data []byte) (int, error) {  
+   outBufPtr := outBufPool.Get().(*[]byte)  
+   outBuf := *outBufPtr  
+   defer func() {  
+       // You might be tempted to simplify this by just passing &outBuf to  
Put,  
+       // but that would make the local copy of the outBuf slice header  
escape  
+       // to the heap, causing an allocation. Instead, we keep around the  
+       // pointer to the slice header returned by Get, which is already on  
the  
+       // heap, and overwrite and return that.  
+       *outBufPtr = outBuf  
+       outBufPool.Put(outBufPtr)  
+   }()  
+}
```

# 核心能力 - 支持国密 BabaSSL

## BabaSSL

- 基于 OpenSSL
- 支持 SM 国密算法
- 支持 QUIC 接口
- MPK (运行时内存保护)
- Keyless
- 硬件加速卡

<https://github.com/mosn/BabaSSL>

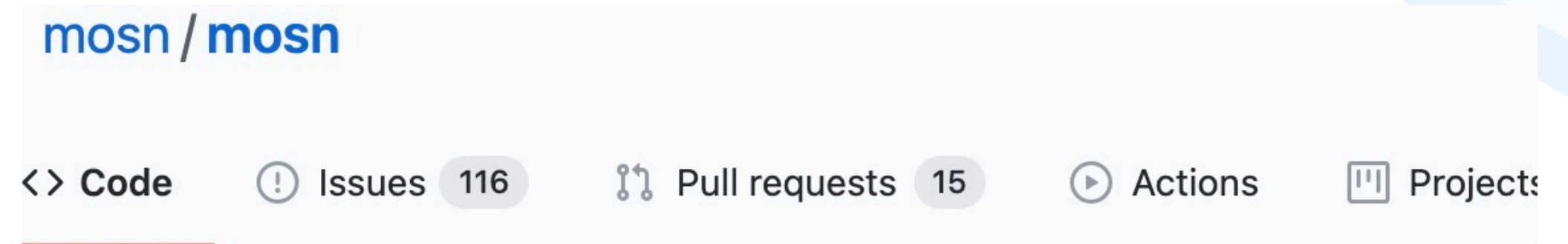


# 核心能力 - 更多能力

- 多进程 Plugin 扩展
- 支持 Listen Filter
- 支持 gzip
- 支持 strict dns cluster
- 支持 DSL
- 支持流量镜像
- 支持 UDS
- 配置可视化
- 协议自动识别
- 支持 netpoll 模式
- 支持广播能力
- 支持 metric 懒加载
- 支持 auto proc
- 支持多种负载均衡算法

详见: Changelog

<https://github.com/mosn/mosn/releases>



Releases Tags

Tags

**v0.21.0** ...

🕒 on 2 Feb 🔗 e9535c1 📦 zip 📦 tar.gz 📄 Notes ⬇ Downloads

**v0.20.0** ...

🕒 on 5 Jan 🔗 6e1b1f3 📦 zip 📦 tar.gz 📄 Notes ⬇ Downloads

**v0.19.0** ...

🕒 on 1 Dec 2020 🔗 2adbcbc 📦 zip 📦 tar.gz 📄 Notes ⬇ Downloads

**v0.18.0** ...

🕒 on 3 Nov 2020 🔗 43d6ac2 📦 zip 📦 tar.gz 📄 Notes ⬇ Downloads

**v0.17.0** ...

🕒 on 30 Sep 2020 🔗 91338c0 📦 zip 📦 tar.gz 📄 Notes ⬇ Downloads

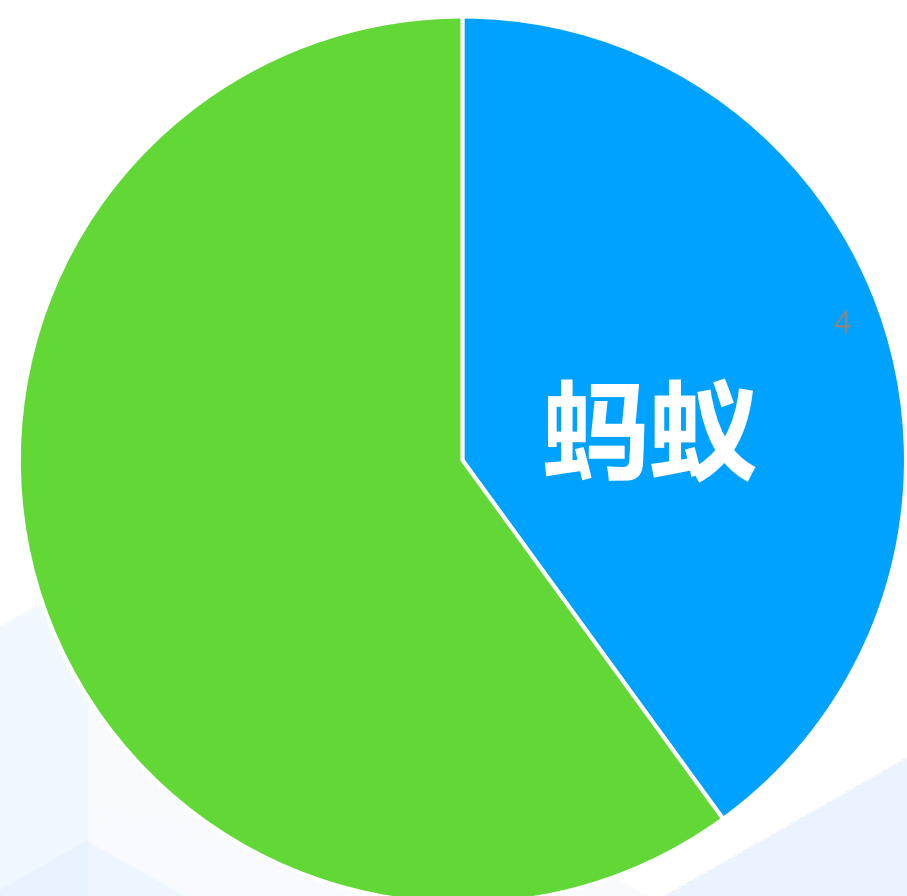
# 周年回顾

- 社区是开源软件发展的动力
- 借力开源，反哺开源
- 持续向云原生演进

Star 数: 2918

Committer: 10

Contributor: 73



## Service Proxy



### 3. 展望未来

# 2021 Roadmap

## 核心能力

- 云原生网络平台建设
- 升级Xprotocol框架
- 支持WASM
- 区块链网络框架
- 代码热更新

## 性能优化

- 支持网络层可扩展
- fastGRPC
- 协程收敛epoll模型
- cgo性能优化

## 微服务

- 支持zipkin, Jaeger等观测产品
- 支持zk, nacos等注册中心
- 支持Dubbo 3.0新特性
- 支持thrift, kafka等协议

## 云原生

- 支持Istio 1.8
- 支持Ingress和Gateway
- 推动UDPA多协议



# 高性能网络扩展层 - 总体架构

## 背景

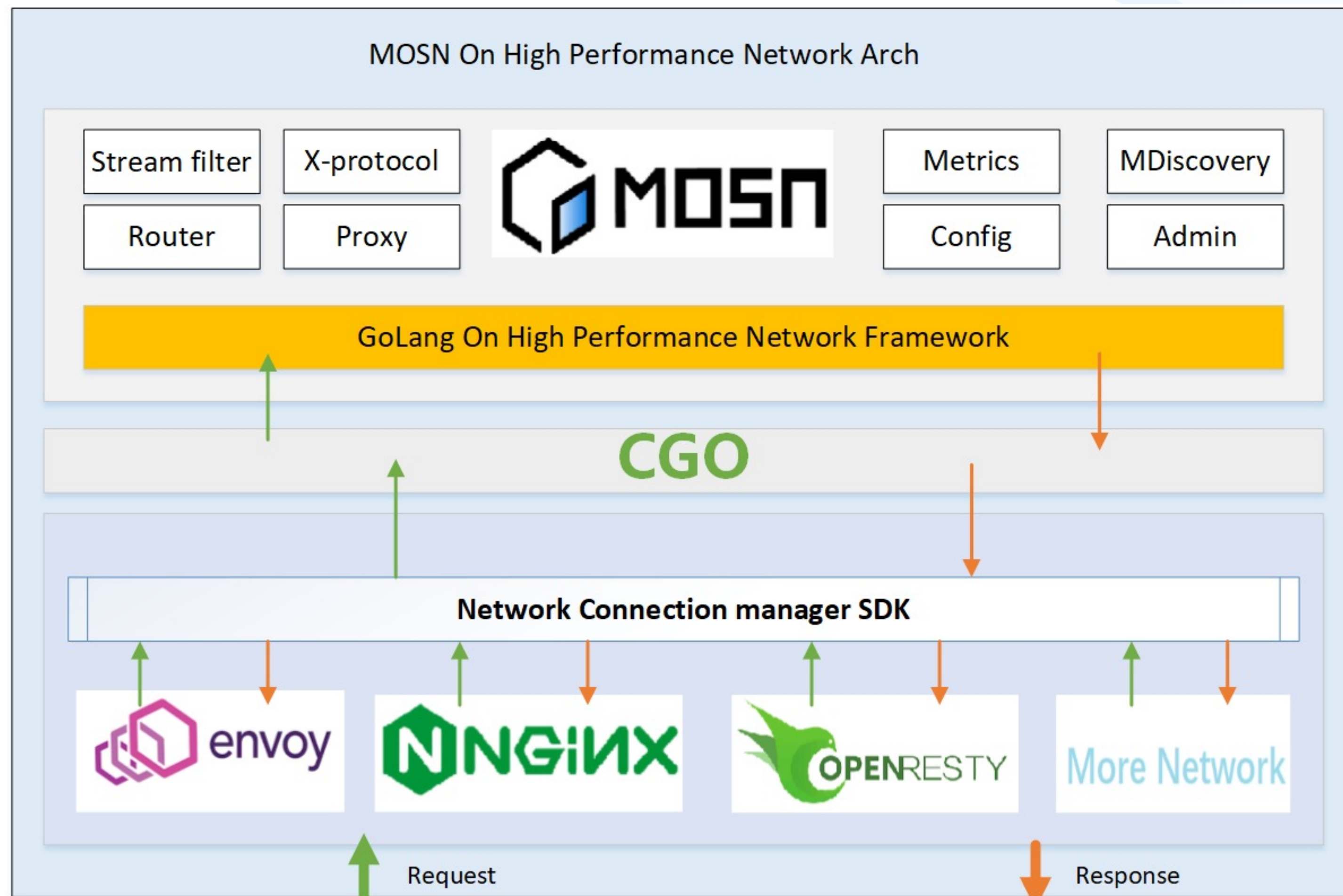
统一数据面（东西 Mesh，南北网关）

## 提议

高性能网络扩展层，和高性能网络库  
结合使用（如 Nginx、Envoy 等）

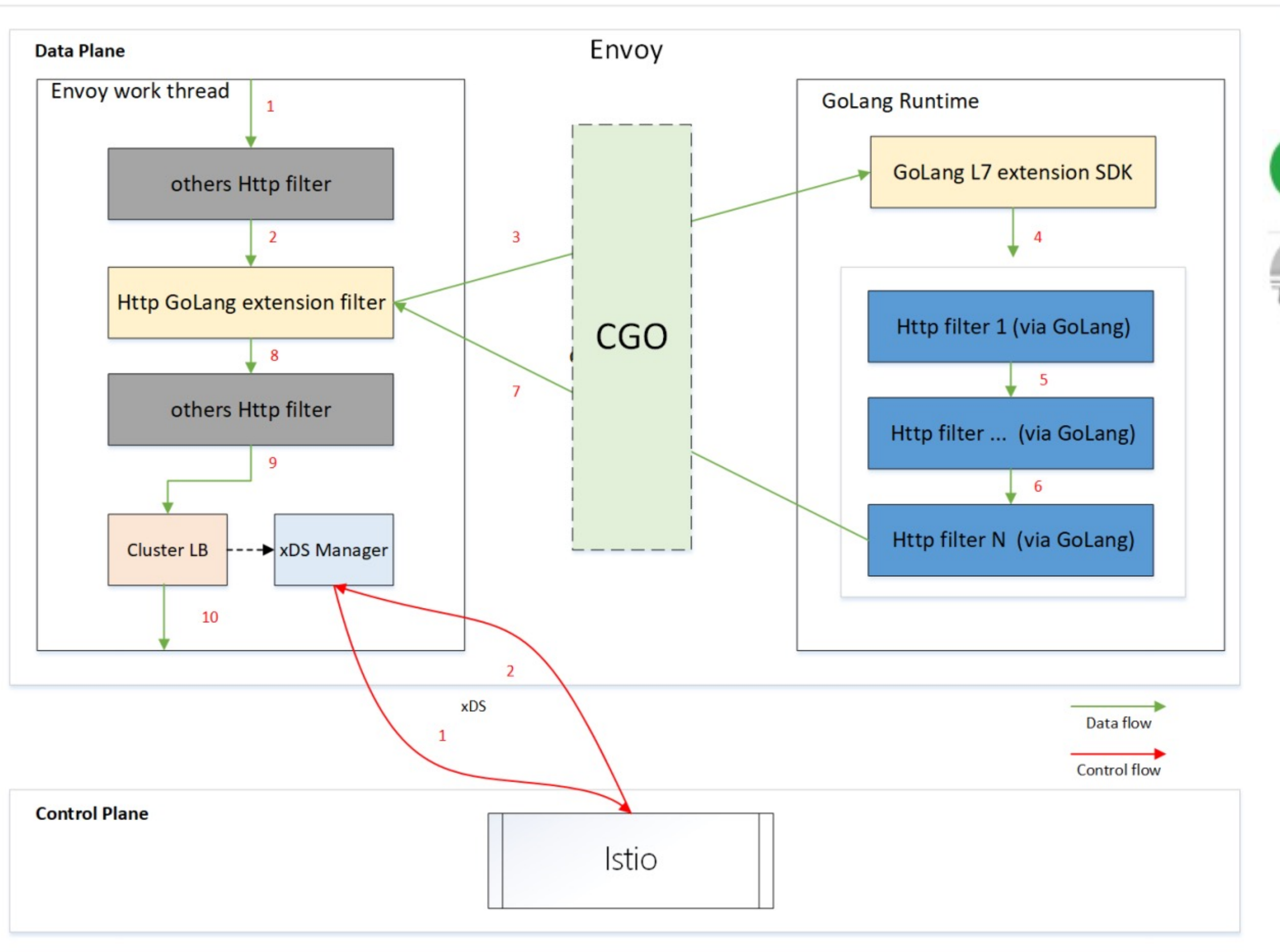
## 特点

- 高性能
- 高研发效能
- 生态打通



# 高性能网络扩展层 - Envoy 进展

## Envoy 方案



## Envoy 社区讨论

Open

A proposal of high-performance L7 network GoLang extension for Envoy. #15152

wangfakang opened this issue 4 days ago · 19 comments



mattklein123 commented 4 days ago

Member

Envoy maintainer 也赞同该方案

In general I'm in favor of doing this as I think it would unlock a huge amount of extensibility within the existing cloud native ecosystem. A few high level comments:

- Cross referencing #15113 which talks about using Envoy as an application platform (cc @nobodyiam). There is a lot of related concepts here, but in particular I really like the idea of allowing the embedding of Dapr style programs within Envoy that all exist out of the primary tree. L7 GoLang extension 方便引入 Dapr 能力
- I really want to figure out how to intersect this work with the work that the Cilium folks have done (cc @jrajahalme @tgraf). IMO we should upstream the Cilium L4 extension also and figure out how to make things consistent between L4 and L7 as much as possible. 同时也期望和 Cilium 交流共建 L4 和 L7 GoLang extension 方案

I think the next steps here are to put together a more long form gdoc/design doc on this topic, and hopefully connect with the Cilium folks to come up with a shared plan. Feel free to email me if folks need help connecting. Thank you!

已支持 Demo, 欢迎试用

<https://github.com/mosn/mosn/blob/master/pkg/networkextention/README-cn.md>

# Application Runtime

## 背景

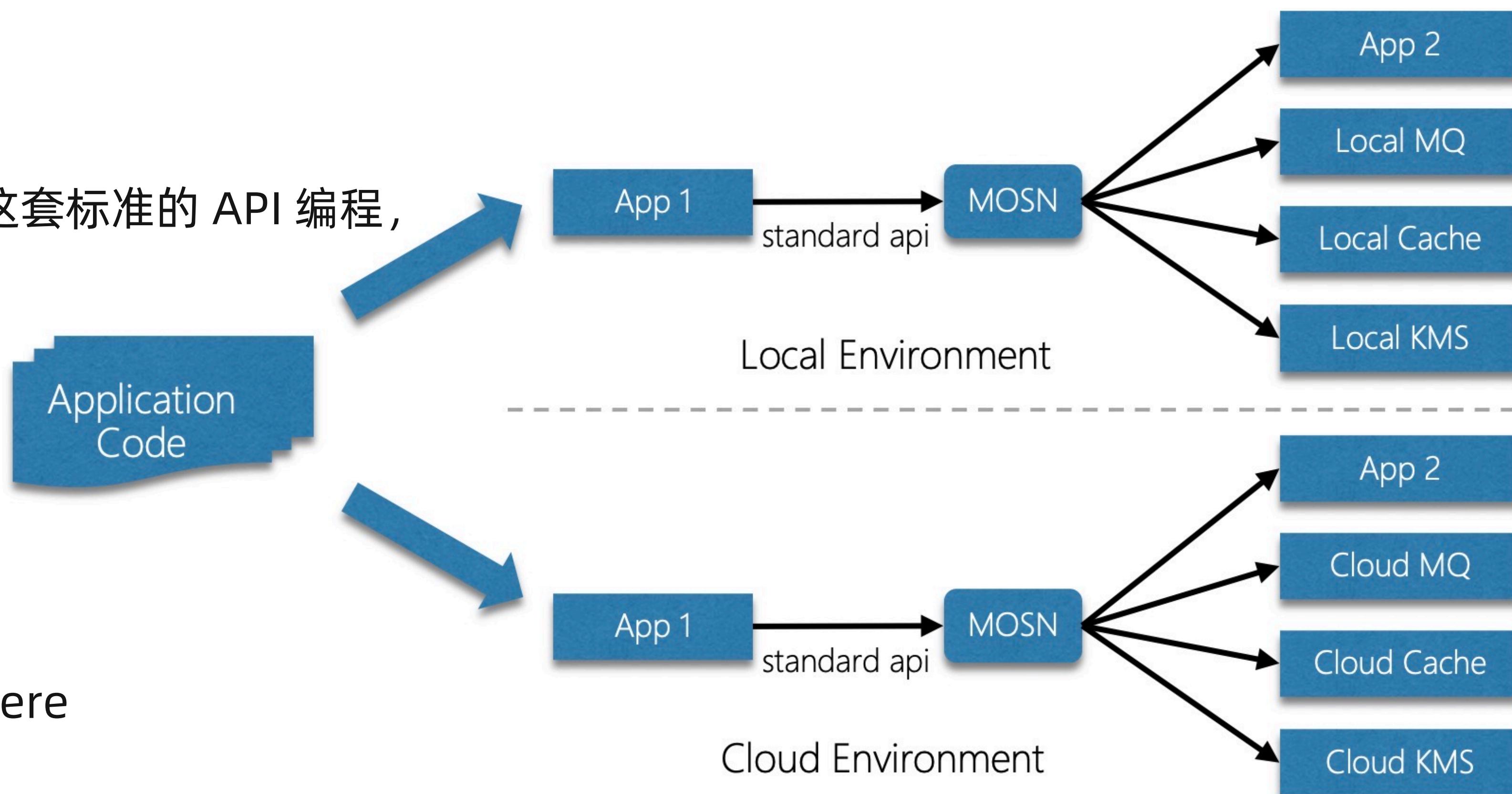
Mesh 解决了微服务方面的挑战，但在实际业务开发中，在缓存、数据库、消息队列、配置管理等领域，我们仍然需要维护一套重量级的 SDK 并且嵌在应用代码中。

## 提议

提供 API 抽象层，应用程序中只针对这套标准的 API 编程，无需考虑实际运行时的后端服务形态。

## 优点

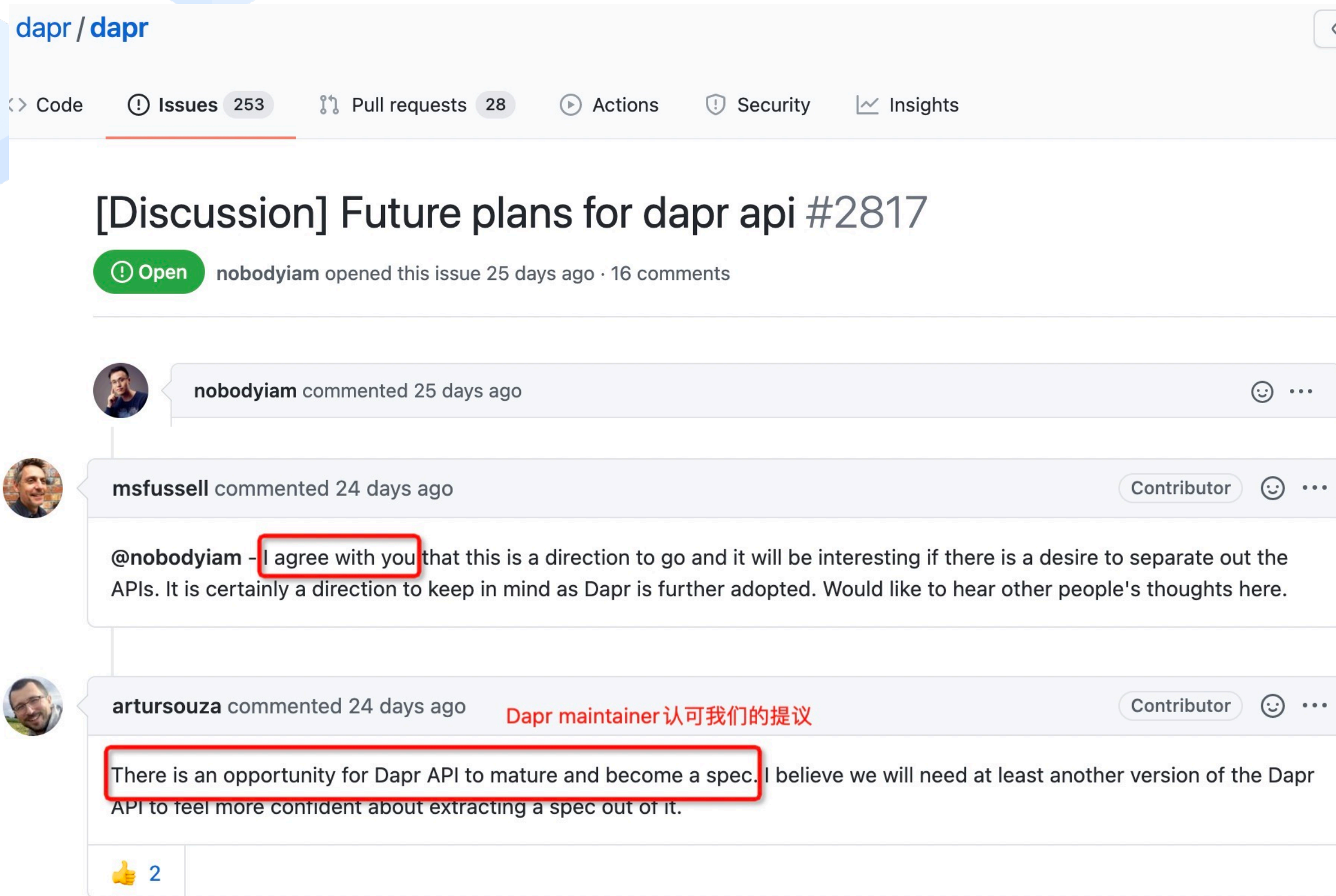
- 多语言友好
- 没有厂商绑定
- 真正实现 Write once, Run anywhere



# Application Runtime - API 标准

- 联合社区共同制定 API 标准
- 目前在 Dapr 和 Envoy 社区都有阶段性进展，已进入 CNCF 为目标

## Dapr 社区



dapr / dapr

<> Code Issues 253 Pull requests 28 Actions Security Insights

### [Discussion] Future plans for dapr api #2817

Open nobodyiam opened this issue 25 days ago · 16 comments

nobodyiam commented 25 days ago

msfussell commented 24 days ago Contributor

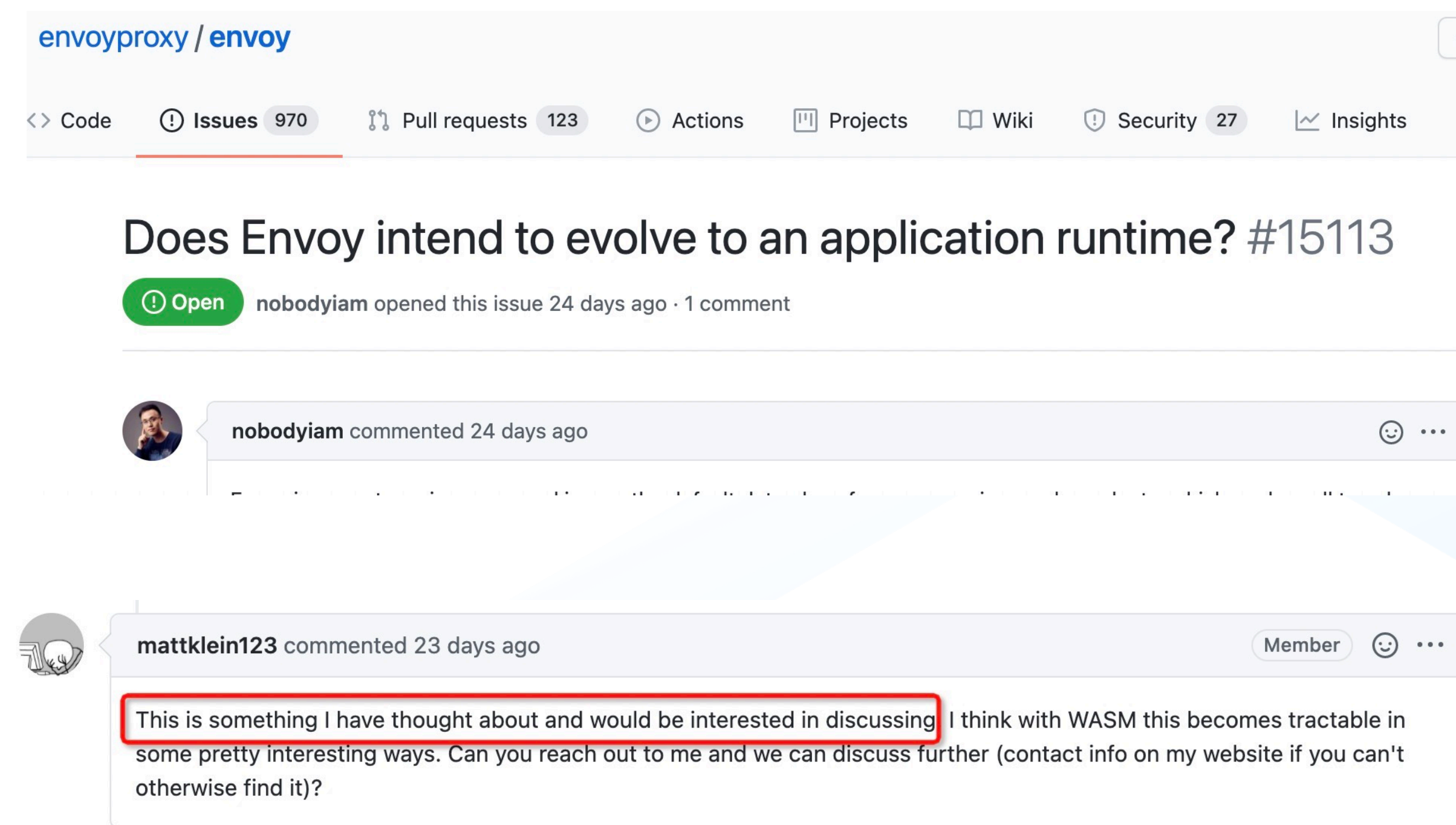
@nobodyiam - I agree with you that this is a direction to go and it will be interesting if there is a desire to separate out the APIs. It is certainly a direction to keep in mind as Dapr is further adopted. Would like to hear other people's thoughts here.

artursouza commented 24 days ago Dapr maintainer 认可我们的提议 Contributor

There is an opportunity for Dapr API to mature and become a spec. I believe we will need at least another version of the Dapr API to feel more confident about extracting a spec out of it.

2

## Envoy 社区



envoyproxy / envoy

<> Code Issues 970 Pull requests 123 Actions Projects Wiki Security 27 Insights

### Does Envoy intend to evolve to an application runtime? #15113

Open nobodyiam opened this issue 24 days ago · 1 comment

nobodyiam commented 24 days ago

mattklein123 commented 23 days ago Member

This is something I have thought about and would be interested in discussing. I think with WASM this becomes tractable in some pretty interesting ways. Can you reach out to me and we can discuss further (contact info on my website if you can't otherwise find it)?

# MOSN - 云原生网络代理平台

## 高性能

- 可扩展网络层 (envoy, nginx)
- 框架深度优化
- 系统级优化

## 易扩展

- Filter 扩展机制
- Xprotocol 多协议框架
- Golang 生态

## 云原生 网络代理平台

## 云原生生态

- 支持 Istio 生态
- 支持大量云原生组件

## 多场景

- Mesh
- Ingress
- Apigateway
- Application Runtime

# 欢迎加入



欢迎大家一起加入 MOSN 社区：

- MOSN 用户群/开发者群
- MOSN committers
- MOSN SIG
- MOSN 开发者会议

MOSN 官网 <http://mosn.io>

MOSN Github <http://github.com/mosn>



 扫一扫群二维码，立刻加入该群。



SOFAStack钉钉群

“金融级分布式架构”微信公众号



调研问卷，支付宝扫码，感谢您的聆听

