



Open Ta//k

云原生应用架构最佳实践

HANGZHOU 2020

云原生网络代理 MOSN 的进化之路

王发康（毅松）

主办方：



自我介绍

王发康（毅松）

蚂蚁金服 可信原生技术部，技术专家

蚂蚁金服技术专家，专注于高性能网络服务器研发，MOSN、Tengine 开源项目核心成员，目前关注云原生 ServiceMesh、Nginx、Istio 等相关领域。

喜欢开源，乐于分享。

<https://github.com/wangfakang>

大纲

- MOSN 介绍
- 云原生演进
- 总结与展望
- Q&A

MOSN 介绍

- MOSN 诞生背景
- MOSN 发展历程
- MOSN 架构解析
- MOSN 落地情况

MOSN 介绍— 诞生背景

MOSN (Modular Open Smart Network) GoLang 语言开发的网络代理软件，作为云原生的网络数据平面，旨在为服务提供多协议，模块化，智能化，安全的代理能力。

业务痛点

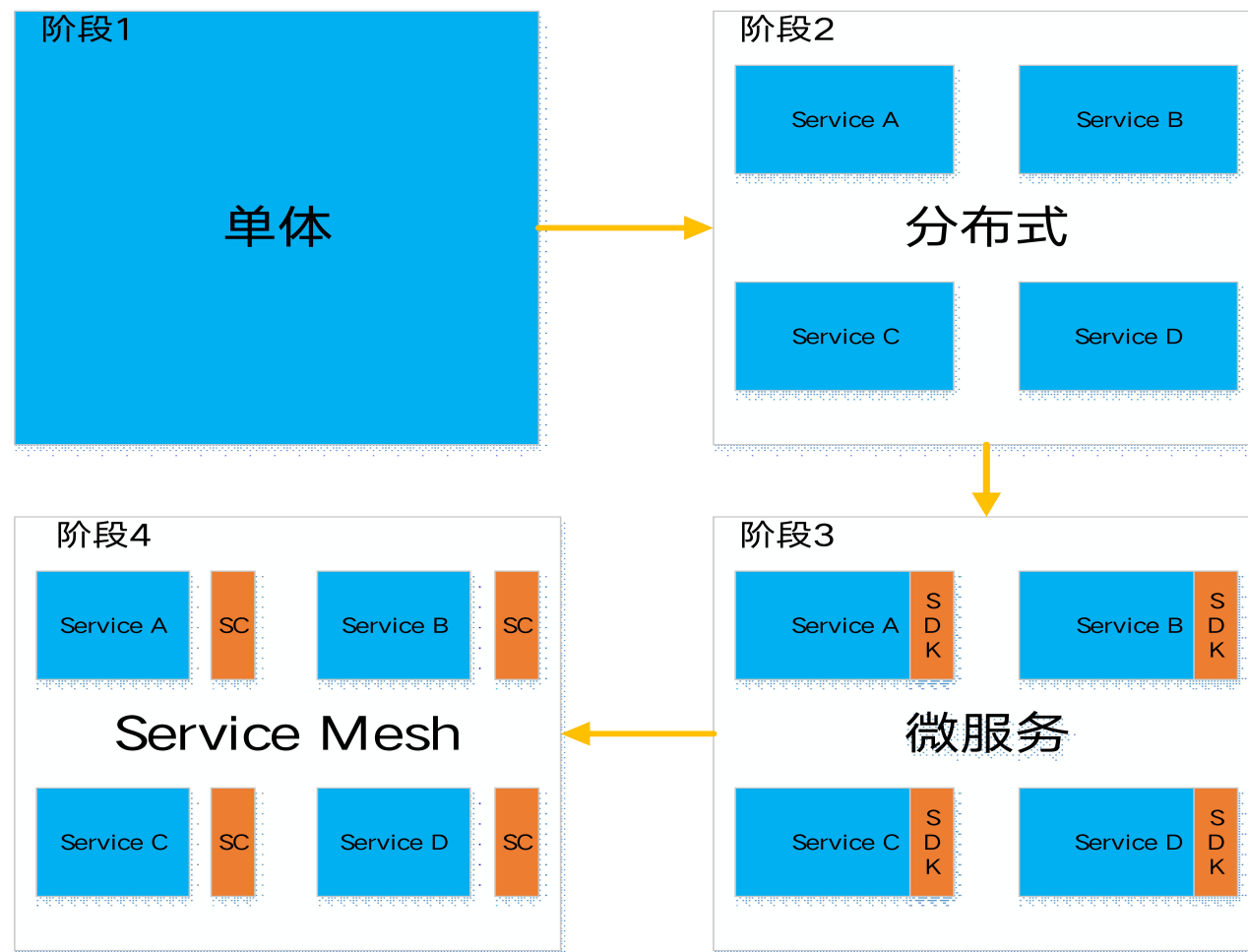
- 多语言，中间件组件开发适配成本高
- SDK 升级困难
- 服务治理能力弱
- 技术不通用，无法复用

业界方案

- Envoy (C++)
- Linkerd (活跃度较低)
- NginxMesh (活跃度低)

MOSN 问世

- GoLang 语言开发
- 兼容 xDS/UDPA

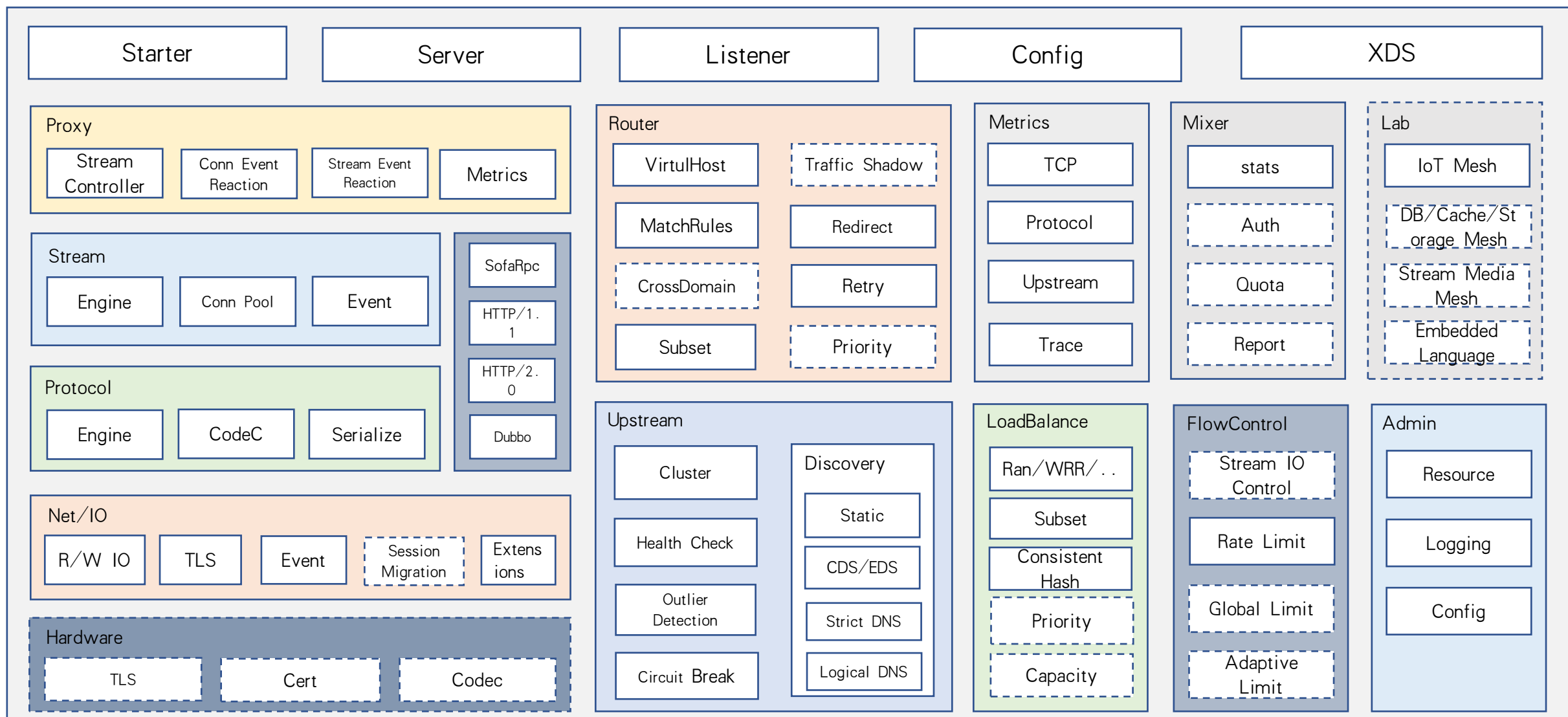


MOSN 介绍 — 发展历程

MOSN 从 Service Mesh 技术调研，到产品孵化，历经重重困难，最终通过双 11 规模化验证。借力开源、反哺开源，进行 Cloud Native 生态融合，在实践的道路上一步步的走向云原生。



MOSN 介绍 — 功能视图



MOSN 介绍 — 架构解析

MOSN 整体框架采用分而治之的架构思想，其中每一层通过工厂设计模式向外暴露其接口，方便用户灵活的注册自身的需求，采用协程池的方式使得用户以同步的编码风格就可以实现异步功能特性。

热生效

- 配置热生效 (xDS)
- 二进制平滑升级 (连接迁移)

多协议 (Xprotocol)

- HTTP 系
- SOFARPC
- Dubbo

流量管理及路由

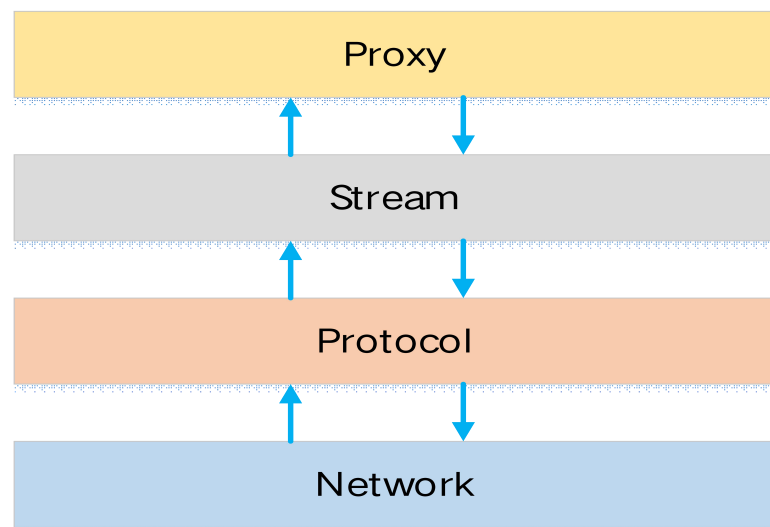
- headers/uri 分流
- 连接池、健康检查、熔断保护、故障注入
- TLS/国密

转发能力

- 四层及七层
- SWRR、Random、EDF、Maglev etc

可观察性

- 连接、请求、流量

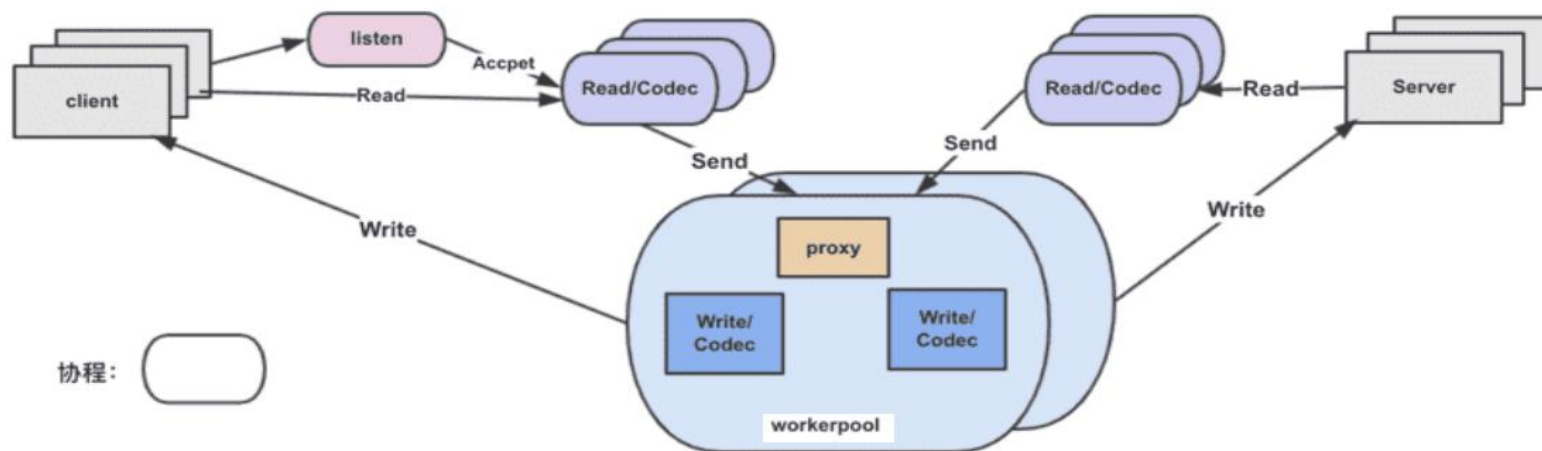


提供proxy框架（分阶段处理、router、loadbalancer、connection pool）

Stream_filter(health_check、faultinject)

多协议层（HTTP系、SOFARPC、Dubbo），提供自定义协议能力（Xprotocol）

Listener_filter（original_dst）、network_filter（tcp、faultinject）

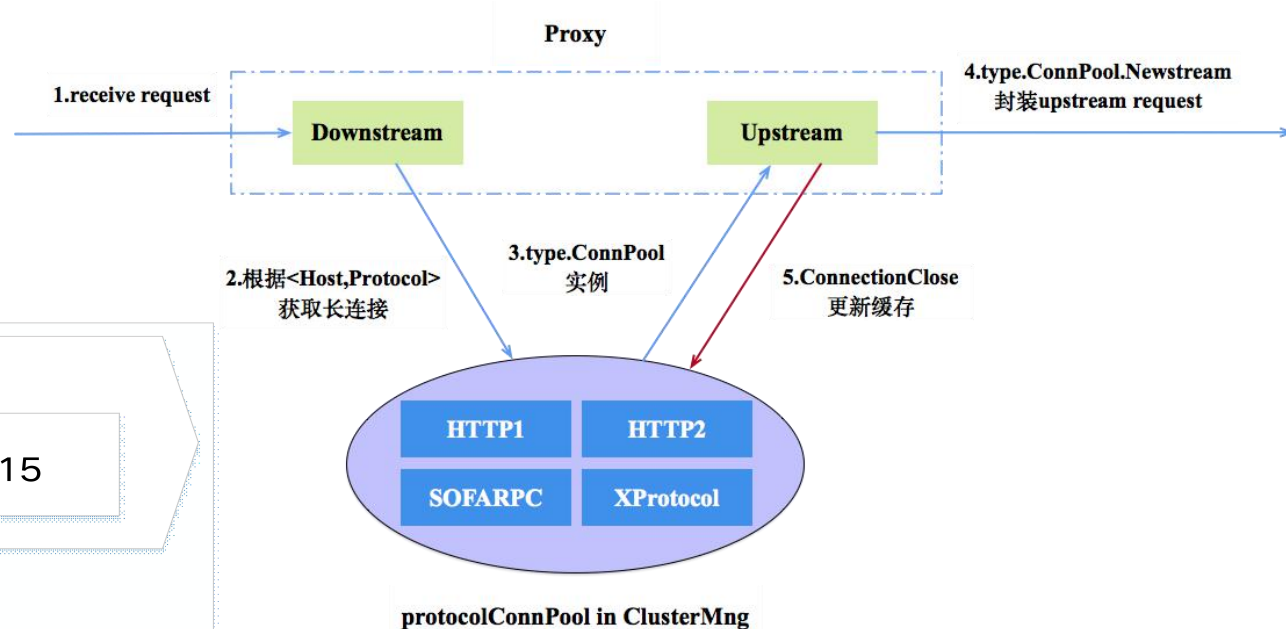
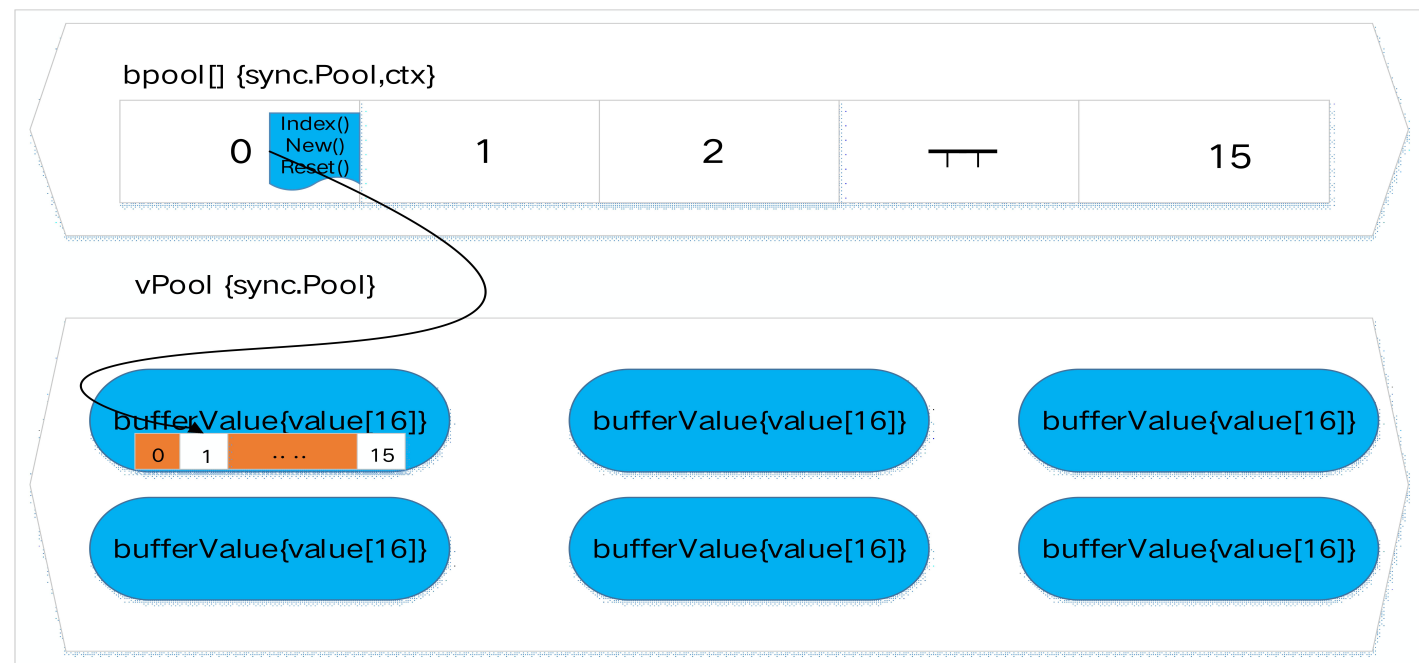


MOSN 介绍 — 内存&连接池

MOSN 为了降低 Runtime GC 带来的卡顿，自身做了内存池的封装，方便多种对象高效的复用内存。为了提升服务网格之间的建连能力，设计封装了多种协议的连接池，方便的实现连接复用及管理。

内存复用

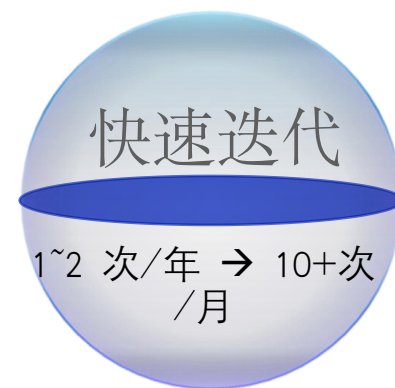
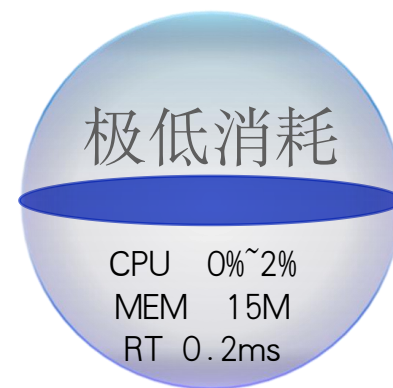
- ByteBufferPool
- IoBufferPool
- CtxBuffer



连接池

- 连接复用
- 连接维度的熔断保护
- 预建连

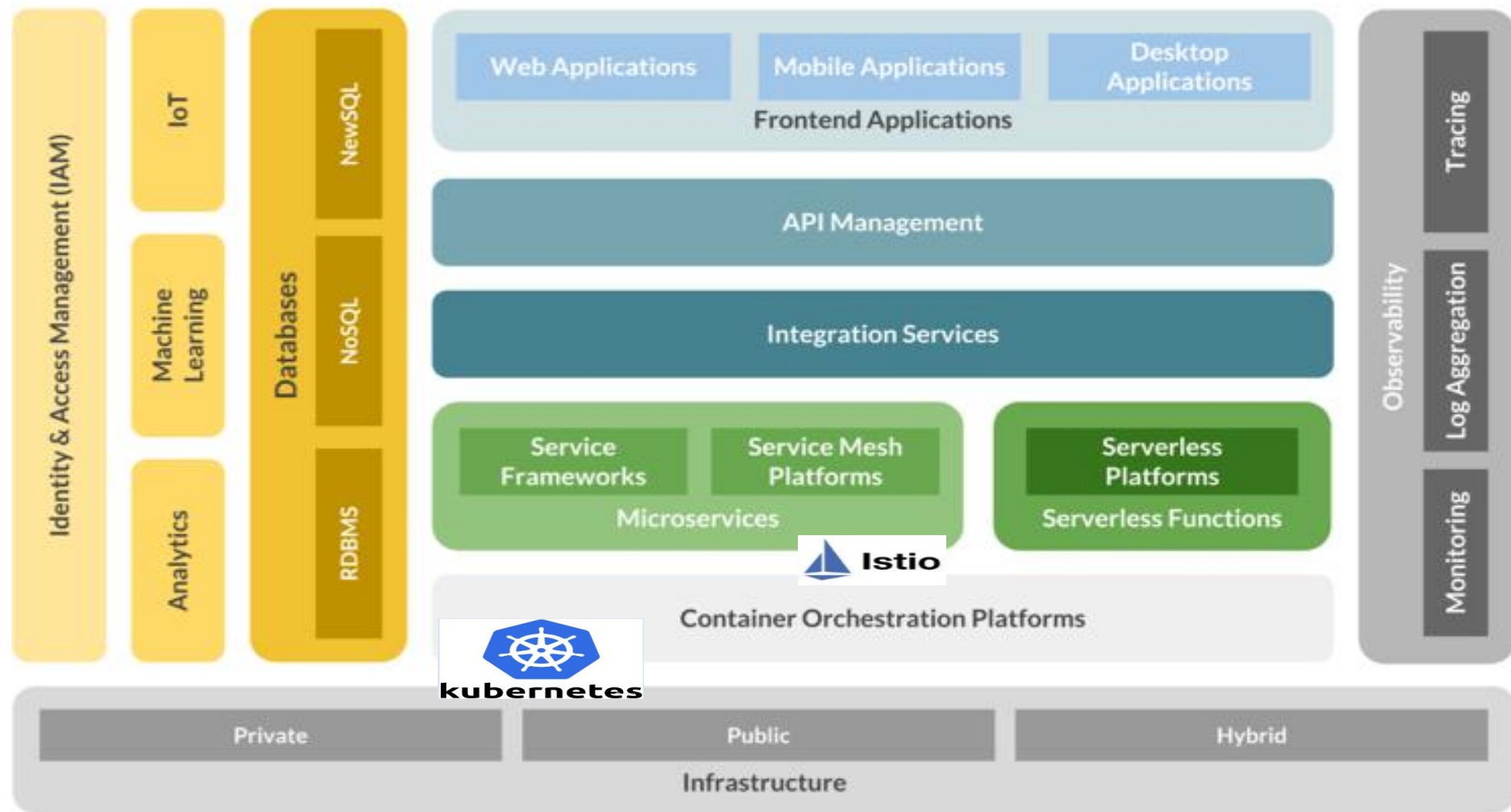
MOSN 介绍 — 落地情况



MOSN 云原生演进

- Could Native Arch
- Istio 简介
- MOSN with Istio 演进历程
- 开源生态建设
- 标准化演进

MOSN 云原生演进 — Could Native Arch



图片来源: <https://dzone.com/articles/implementing-cloud-native-enterprise-applications>

MOSN 云原生演进 — Istio 简介

为什么出现 Istio

- Docker（环境、弹性、微服务）
- K8s（容器编排、调度、管理）
- Service Mesh（微服务流量治理）

Istio 解决的问题

- 弥补K8s的服务治理短板
- 微服务网格管理

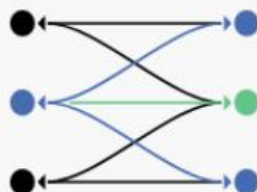
Istio 的功能

- 服务互连
- 流量安全
- 流量控制
- 可观测性



Istio

Connect, secure, control, and observe services.



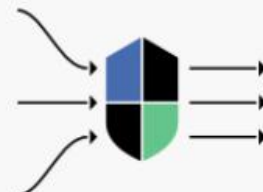
Connect

Intelligently control the flow of traffic and API calls between services, conduct a range of tests, and upgrade gradually with red/black deployments.



Secure

Automatically secure your services through managed authentication, authorization, and encryption of communication between services.



Control

Apply policies and ensure that they're enforced, and that resources are fairly distributed among consumers.

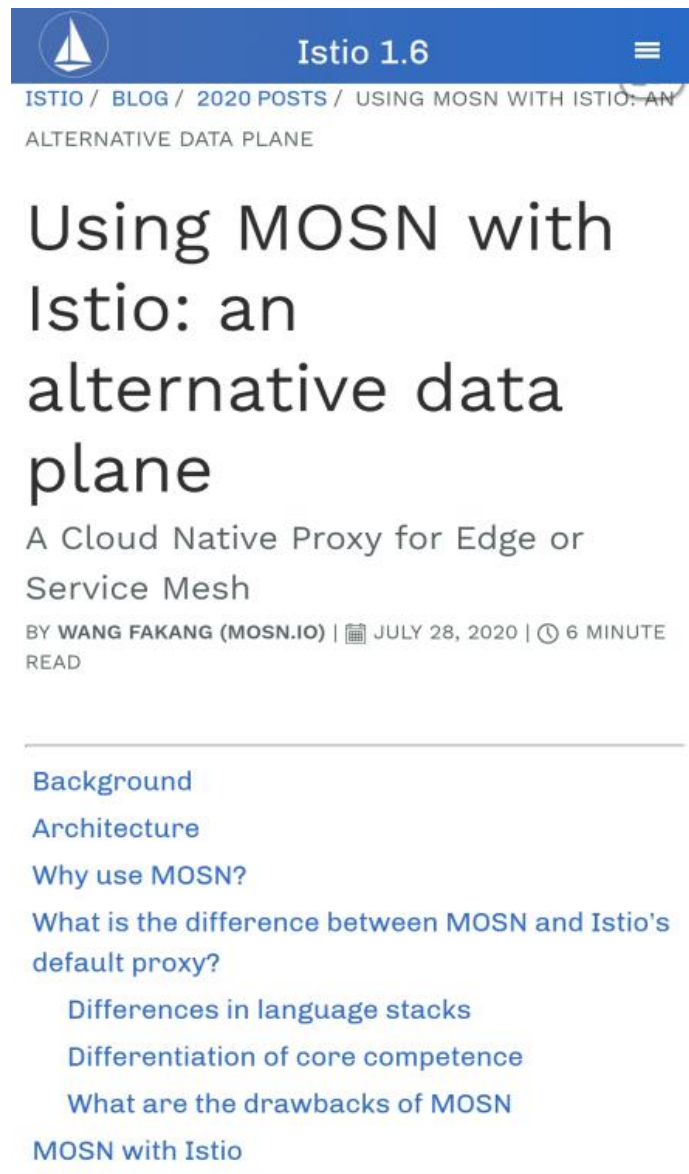


Observe

See what's happening with rich automatic tracing, monitoring, and logging of all your services.

图片来源: <https://istio.io>

MOSN 云原生演进 — An alternative data plane

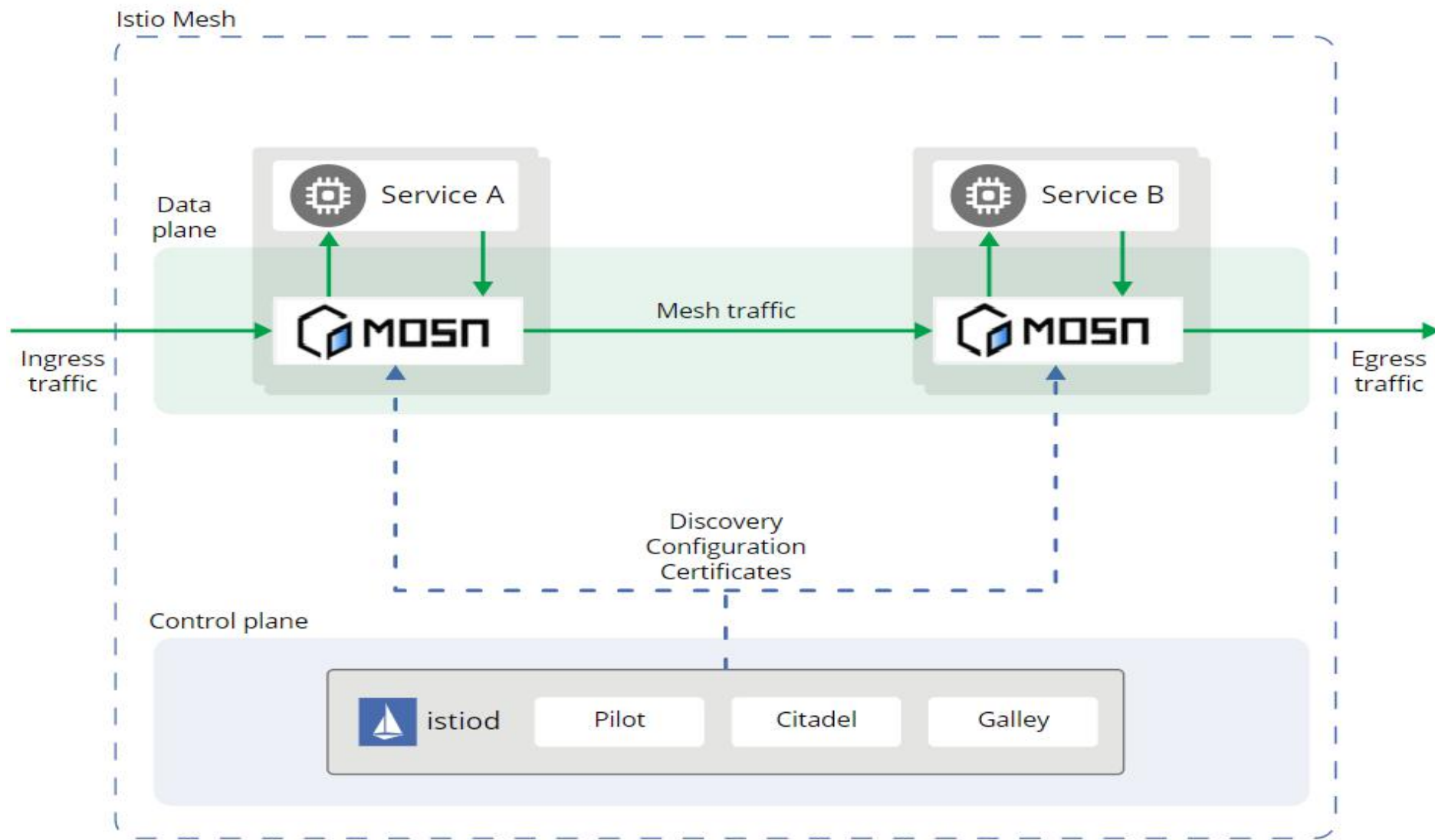


Istio 数据平面的另一个选择 —— MOSN

Istio 官方博客介绍

<https://istio.io/latest/blog/2020/mosn-proxy>

MOSN 云原生演进 — MOSN with Istio Arch



图片来源: <https://www.servicemeshes.com/istio-handbook/concepts/mosn.html>

云原生演进 — MOSN with Istio

云原生标准 Sidecar

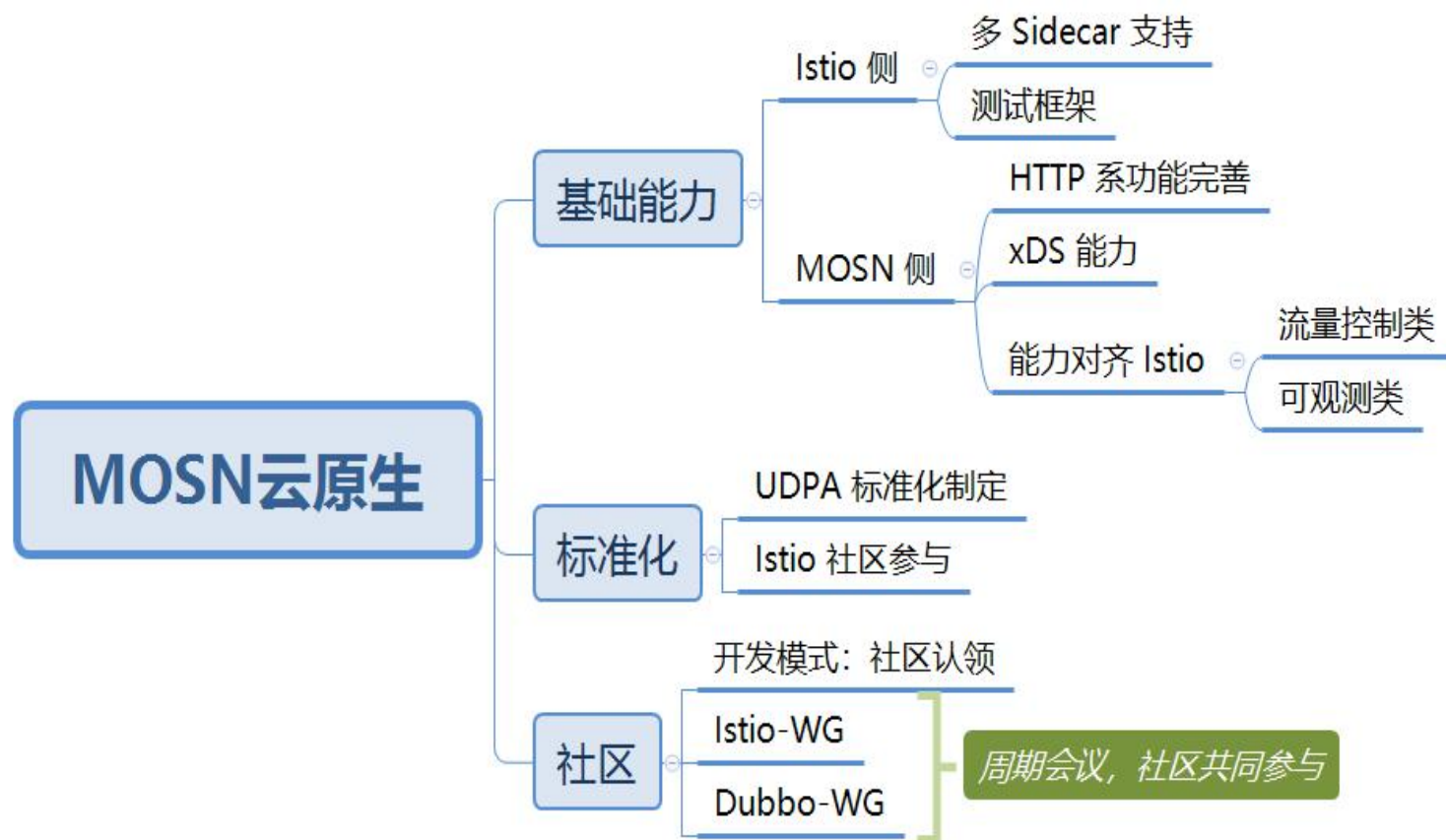
- Istio 侧多 Sidecar 支持
- MOSN 侧功能对齐 Istio

标准化

- 参与 UDPA 讨论及标准制定
- 参与 Istio 的开发

社区共建

- 建立 [Istio-WG](#)、[Dubbo-WG](#)
- 多开发者参与[共建](#)



MOSN 云原生演进 — 适配 Istio 列表

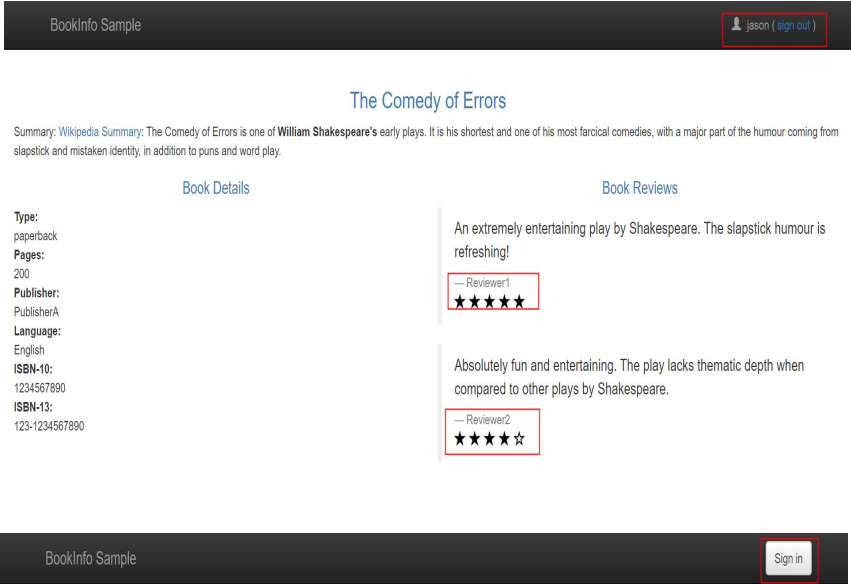
Type	Feature	Description	Application scenarios	Pb on the Istio	priority	issue	status
Load balance	original_dst	Forward the request based on the original destination	In the Istio transparent hijacking scenario, if no matching	https://github.com/envoyproxy/go-control-plane/issues/1161	p0	https://github.com/mosn/mosn/issues/982	Done
	WRR	weight round robin	Load balancing according to machine weight	https://github.com/envoyproxy/go-control-plane/issues/1161	3/30/2020	https://github.com/mosn/mosn/pull/1161	Done
	least req	Choose the machine with the least request	Load balancing according to the number of machine request	https://github.com/envoyproxy/go-control-plane/issues/1054	p2	https://github.com/mosn/mosn/issues/1054	Done
	Maglev	Consistent hash based on the request	Keep the session	https://github.com/envoyproxy/go-control-plane/issues/1118	p2	https://github.com/mosn/mosn/pull/1118	Done
Protocol	WebSocket	Implement WebSocket protocol	WebSocket proxy forwarding		p3		
	UDP/QUIC	support UDP/QUIC	proxy forwards the UDP/QUIC request		p3	https://github.com/mosn/mosn/issues/1062	WIP
xDS	xDS adapter	Adapt to Pilot's xDS API	Manage MOSN by Istio		p0	https://github.com/mosn/mosn/tree/feature-istio-adapter	Done
Service governance	Fault Injection	Injecting an delay or abort fault	Simulate the application of exception scenarios, can inject	https://github.com/envoyproxy/go-control-plane/issues/1063	p0	https://github.com/mosn/mosn/pull/1063	Done
	limit req	limit request	limiting protection	https://github.com/envoyproxy/go-control-plane/issues/1111	p1	https://github.com/mosn/mosn/pull/1111	Done
	Passive health	If the number of errors exceeds	Passive health check, disable abnormal service	https://github.com/envoyproxy/go-control-plane/issues/1055	p0	https://github.com/mosn/mosn/issues/1055	Done
	Circuit Breaking	Limits on the maximum number	Services to protect	https://github.com/envoyproxy/go-control-plane/issues/395	p1	https://github.com/mosn/mosn/issues/395	WIP
	External Author	External Authorization	Request authentication based on external authentication	https://github.com/envoyproxy/go-control-plane/issues/1056	p2	https://github.com/mosn/mosn/issues/1056	
	Traceing	Service observability	Service observability		p1	https://github.com/mosn/mosn/issues/1027	Done
	Traffic mirror	traffic mirror	traffic mirror	https://istio.io/latest/zh/docs/tasks/traffic-r	p1		
	Enhance rewrite	regex_rewrite host_rewrite	Rewrite the request's uri and host etc.	https://github.com/envoyproxy/go-control-plane/issues/1191	p1	https://github.com/mosn/mosn/issues/1191	WIP
Framework extension	listener filter	Optimize the listener filter frame	Support listener filter registration		p1	https://github.com/mosn/mosn/pull/1014	Done
	Strict DNS clus	Support STRICT_DNS type Clus	Service discovery by DNS	https://github.com/envoyproxy/go-control-plane/issues/1149	p1	https://github.com/mosn/mosn/pull/1149	Done
	Logical DNS cl	Dynamic resolve domain and us	Forward proxy by domain	https://github.com/envoyproxy/go-control-plane/issues/1005	p1	https://github.com/mosn/mosn/pull/1005	Done
Programmable capability	Lua	Support Lua	Support Lua	https://github.com/envoyproxy/go-control-plane/issues/1151	p2		TBD
	Wasm	Support Wasm	Support Wasm		p2		TBD
others	gzip filter	Support gzip response body	Gzip		p2	https://github.com/mosn/mosn/pull/1151	Done
	pilot-agent prot	Support serverinfo api	Pilot agent health check		p0	https://github.com/mosn/mosn/issues/1057	Done
	istio.stats filter	Metrics support custom Lables	Metrics support custom Lables		p1	https://github.com/mosn/mosn/pull/1209	WIP
	parameter exte	Parameter extension	Compatible with pilot starting mosn		p0	https://github.com/mosn/mosn/issues/1023	Done
Istio	image build	Make proxyv2 image building oper	Build proxyv2 image for multiple data planes		p1	https://github.com/istio/istio/issues/24457	WIP
	test framework	Modify istio test framework, make c	One test framework could testing multiple data planes		p1	https://github.com/istio/istio/issues/24483	WIP

Roadmap: <https://docs.google.com/spreadsheets/d/1fALompY9nKZNIImOuxQw23xtMD-5rCBrXWziJZkj76bo/edit?usp=sharing>

基于请求 Header 路由

Istio VirtualService

MOSN Router



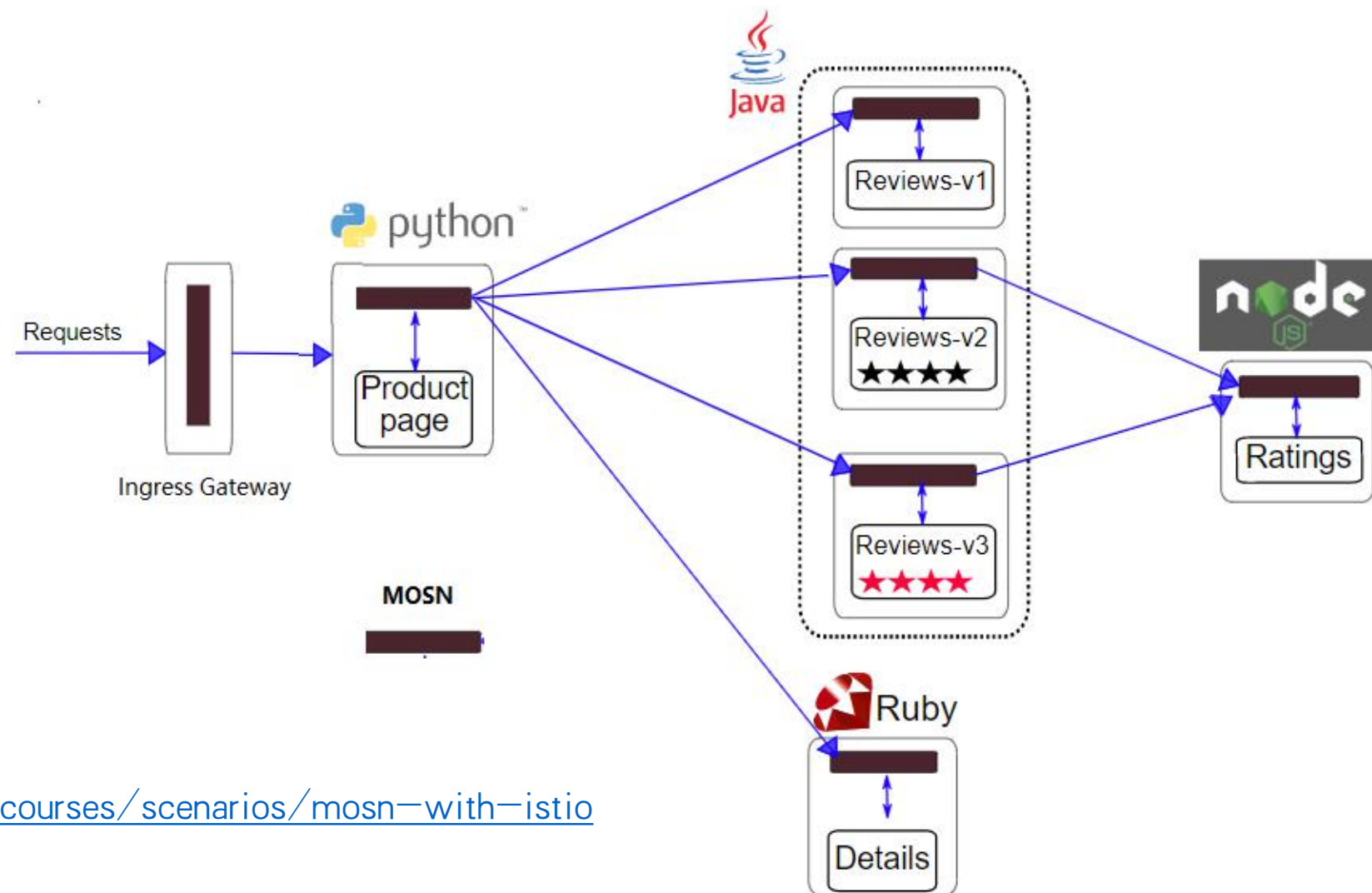
```
#kubectl get virtualservice reviews -oyaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: '{"apiVersion": "networking.istio.io/v1alpha3", "kind": "VirtualService", "spec": {"hosts": ["reviews"], "http": [{"match": [{"headers": {"host": "reviews", "subset": "v3"}}]}]}'}'
  creationTimestamp: "2020-03-10T08:34:57Z"
  generation: 6
  name: reviews
  namespace: default
  resourceVersion: "1053451"
  selfLink: /apis/networking.istio.io/v1alpha3/namespaces/default/virtualservices/reviews
  uid: 05e2e1ca-62aa-11ea-93b9-00163e09e68e
spec:
  hosts:
  - reviews
  http:
  - match:
    - headers:
      end-user:
        exact: jason
    route:
    - destination:
        host: reviews
        subset: v2
  - route:
    - destination:
        host: reviews
        subset: v3
```

```
{
  "name": "reviews.default.svc.cluster.local:80",
  "domains": [
    "reviews.default.svc.cluster.local",
    "reviews.default.svc.cluster.local:80"
  ],
  "routers": [
    {
      "match": {
        "prefix": "/",
        "headers": [
          {
            "name": "end-user",
            "value": "jason"
          }
        ]
      },
      "route": {
        "cluster_name": "outbound|80|v2|reviews",
        "timeout": "0s",
        "retry_policy": {
          "retry_on": true,
          "retry_timeout": "0s",
          "num_retries": 2
        }
      }
    },
    {
      "match": {
        "prefix": "/"
      },
      "route": {
        "cluster_name": "outbound|80|v3|reviews",
        "timeout": "0s",
        "retry_policy": {
          "retry_on": true,
          "retry_timeout": "0s",
          "num_retries": 2
        }
      }
    }
  ]
}
```


MOSN 云原生演进 — 服务治理实践

Istio 的 BookInfo 事例

- 按 version 路由能力
- 按照权重路由能力
- 按照特定 header 路由能力
- 故障注入能力
- 服务熔断自护能力
- 透明劫持能力
- 超时重试机制
- etc



[演示教程《MOSN with Istio》](#)

<https://www.katacoda.com/mosn/courses/scenarios/mosn-with-istio>

云原生演进 — MOSN with Dubbo

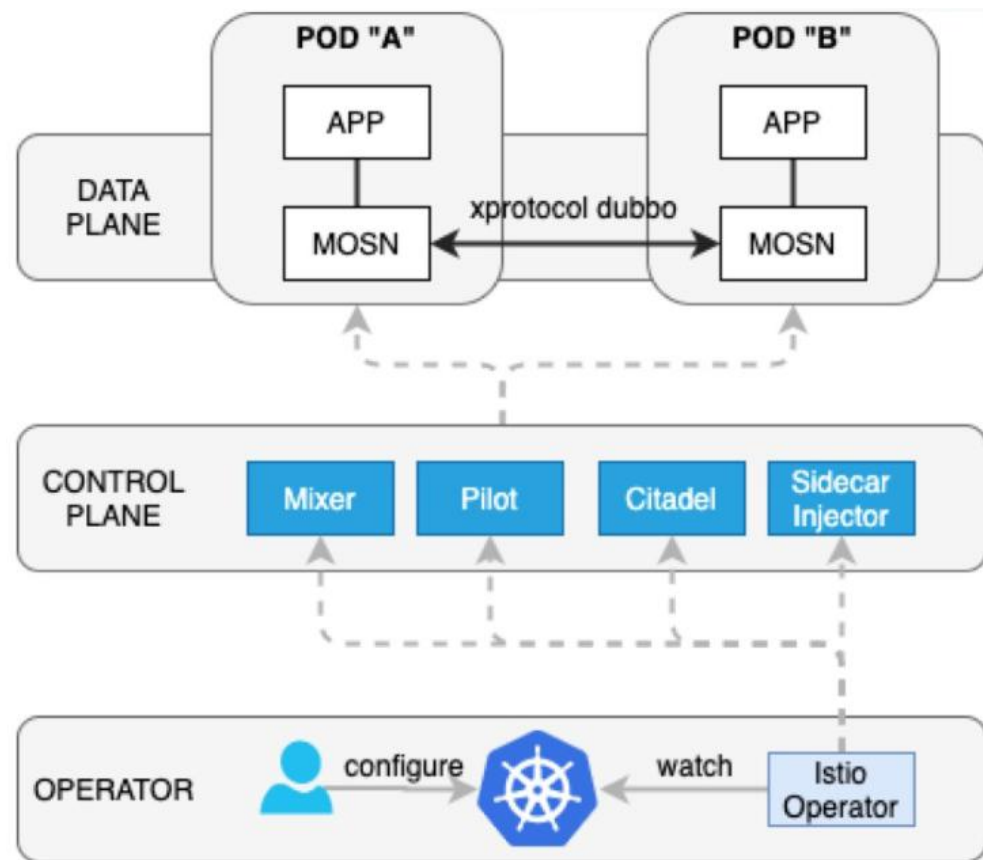
业务诉求（微服务治理）

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

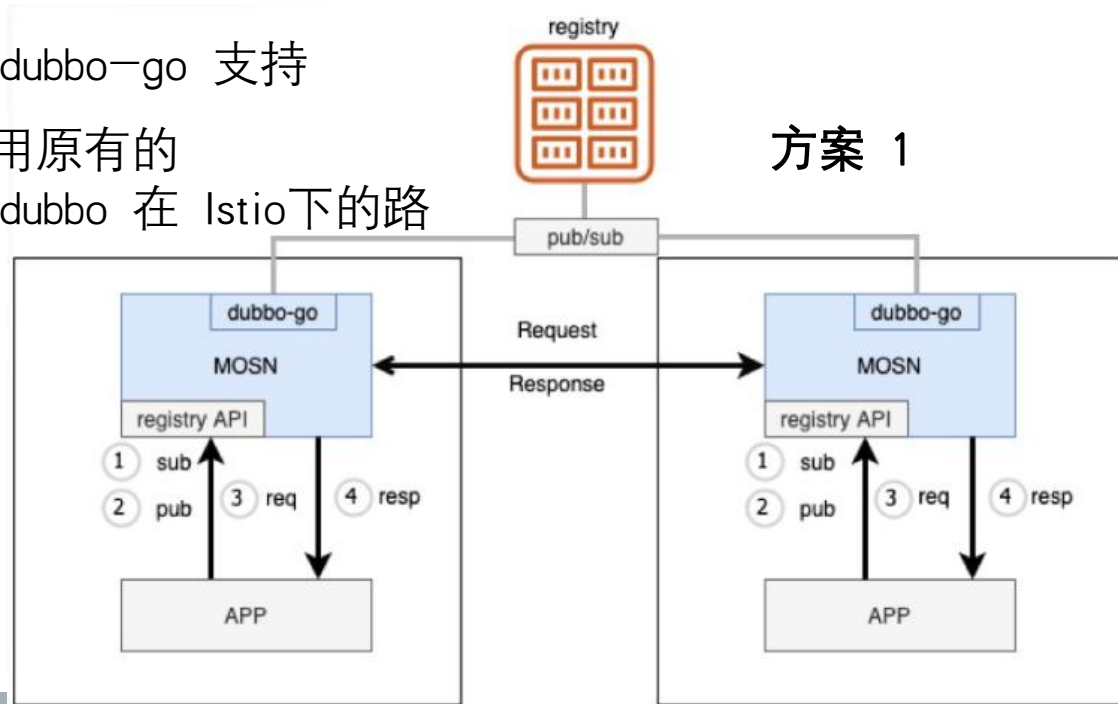
解决方案

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

方案 2



方案 1



云原生演进 — MOSN with Sentinel

背景描述

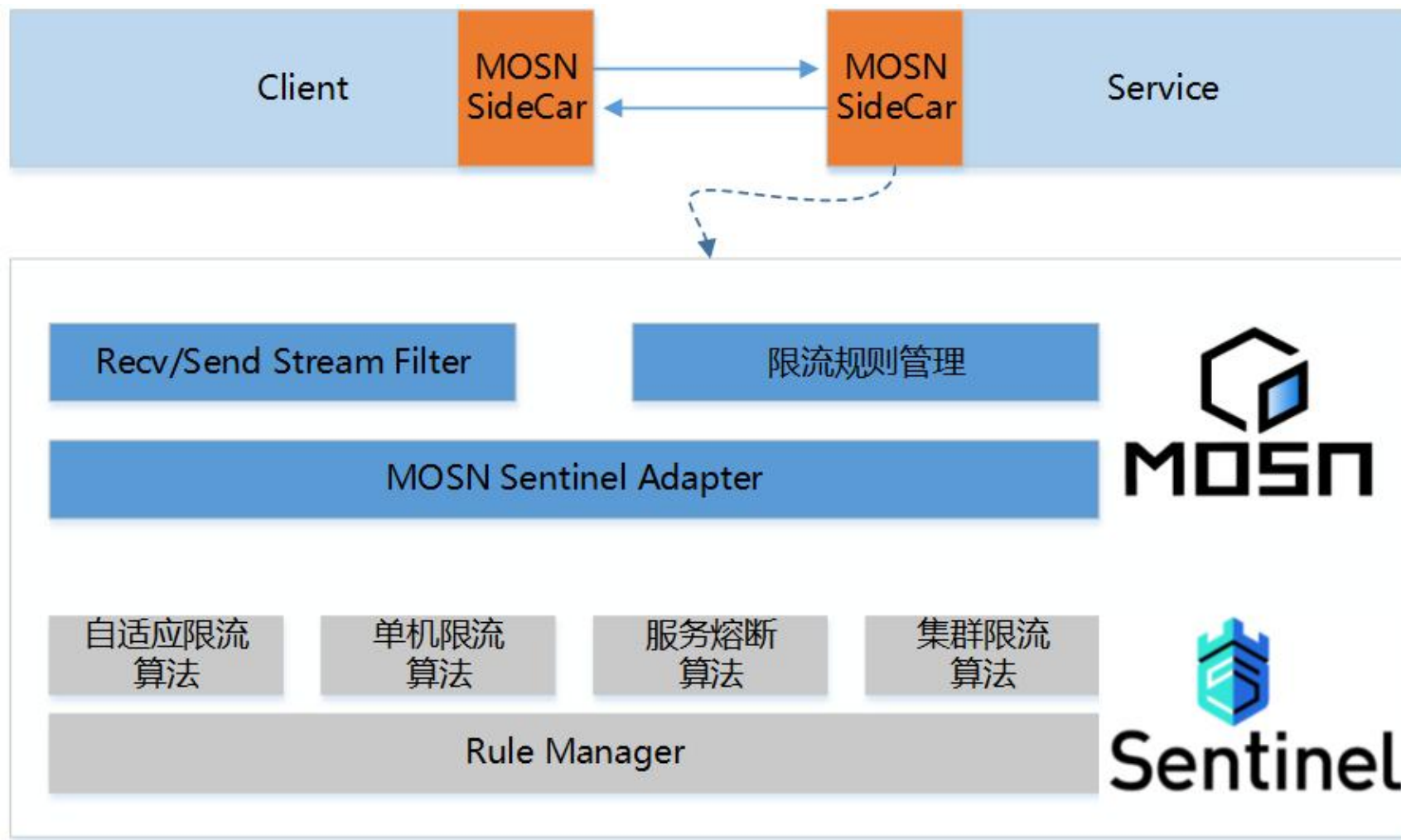
- 限流是微服务治理之一
- MOSN 集成 Sentinel, 复用底层的限流能力

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



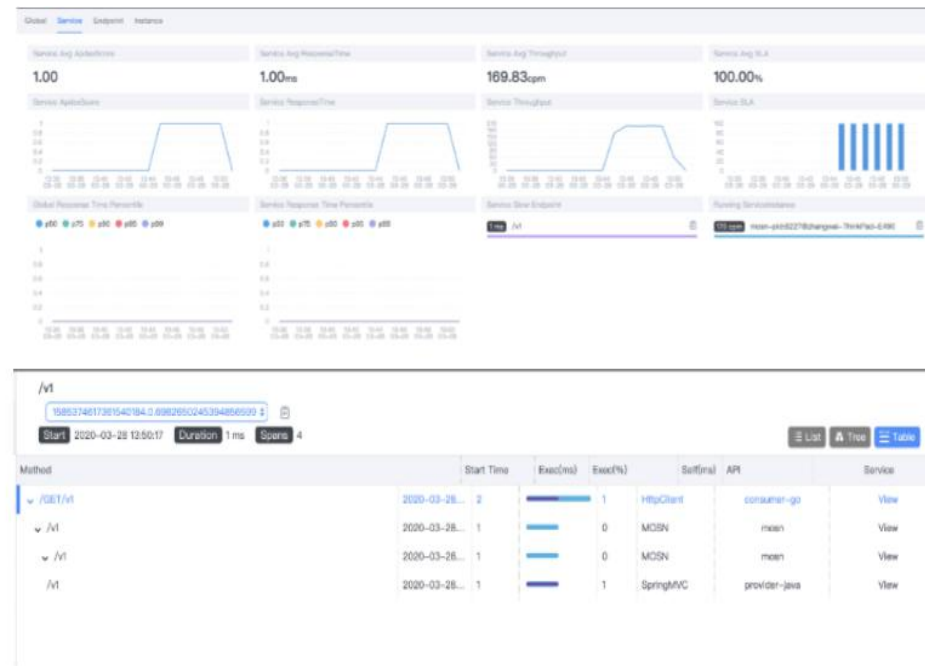
云原生演进 — MOSN with Skywalking

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

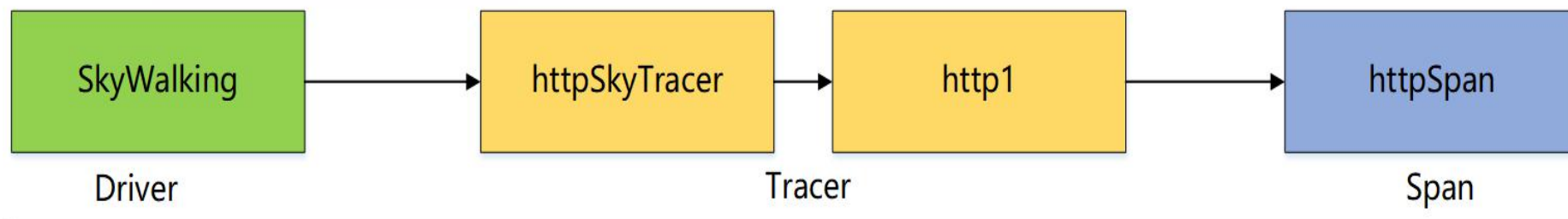
功能介绍

- 调用链路拓扑展示
- QPS 监控
- 细粒度 RT 展示



持续演进

- Dubbo Tracing 支持



云原生演进 — 标准化演进

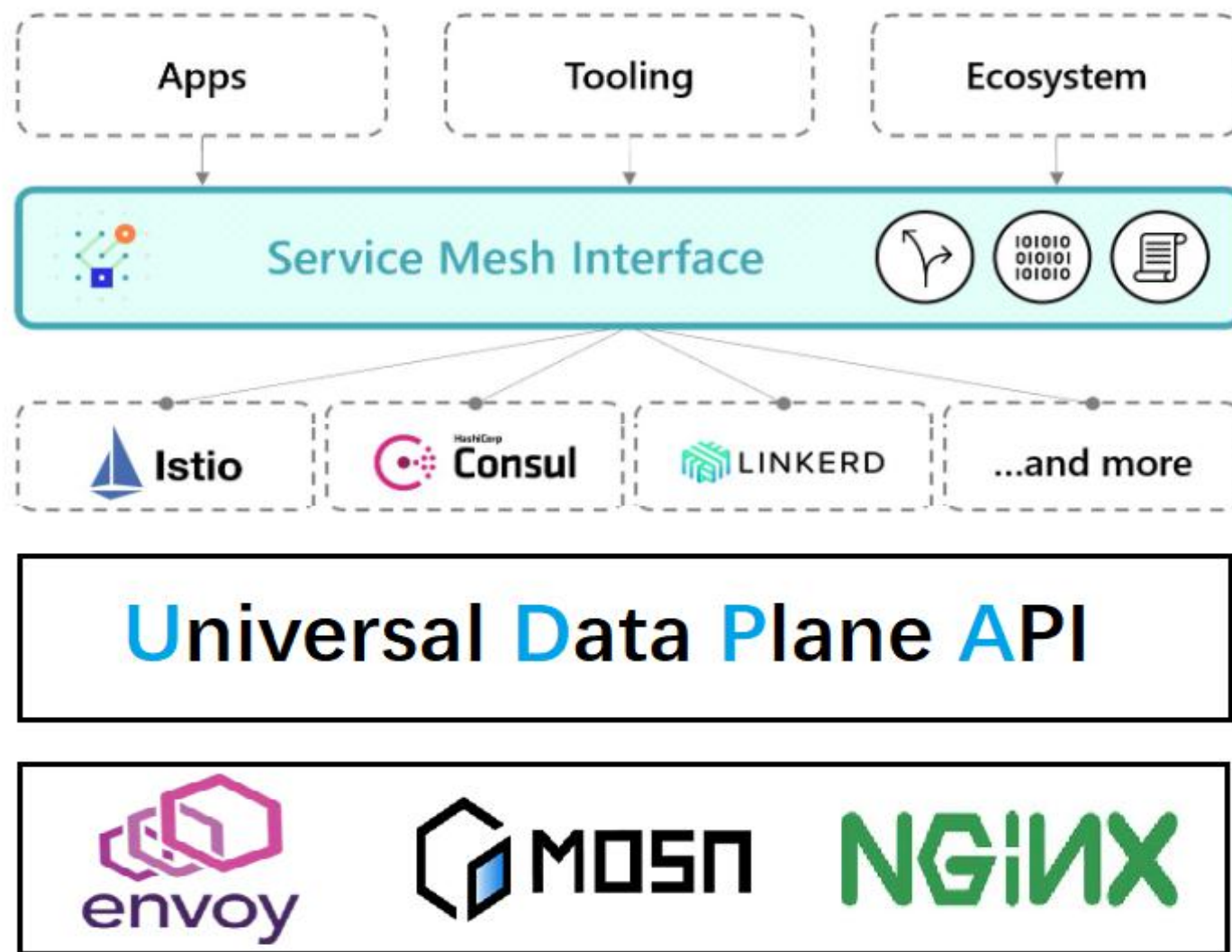
SMI

- 控制面之上的一层标准 API
- 解耦控制面和上层应用/工具等

UDPA

- 数据面之上的一层标准 API
- 解耦控制面和数据面通信

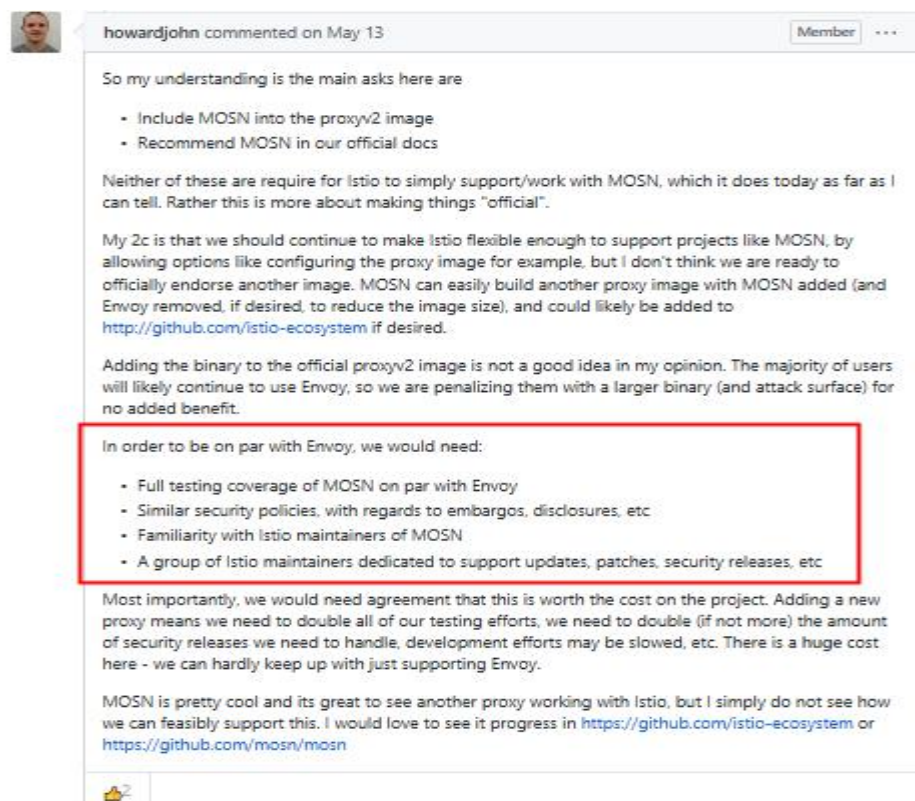
防止锁定，可方便用户
灵活切换



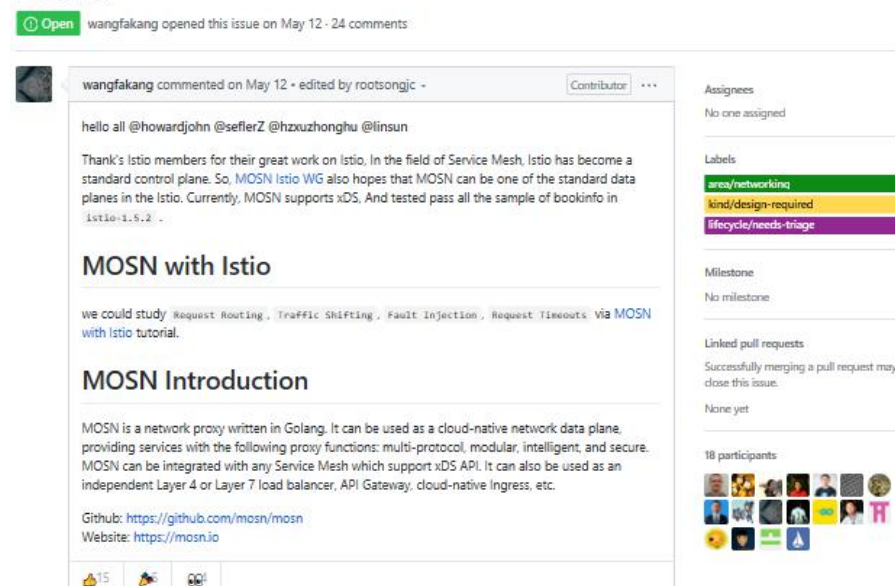
云原生演进 — 标准化演进

MOSN 兼容 Istio

- MOSN 支持 Istio 的功能
- MOSN 自己维护 proxyv2 镜像，官方可以将其纳入到 istio-ecosystem



Istio could supports multiple data planes? such as MOSN



MOSN 成为 Istio 官方推荐 Sidecar

- MOSN 支持 Istio 的功能
- 进入 Istio 官方推荐的 Sidecar

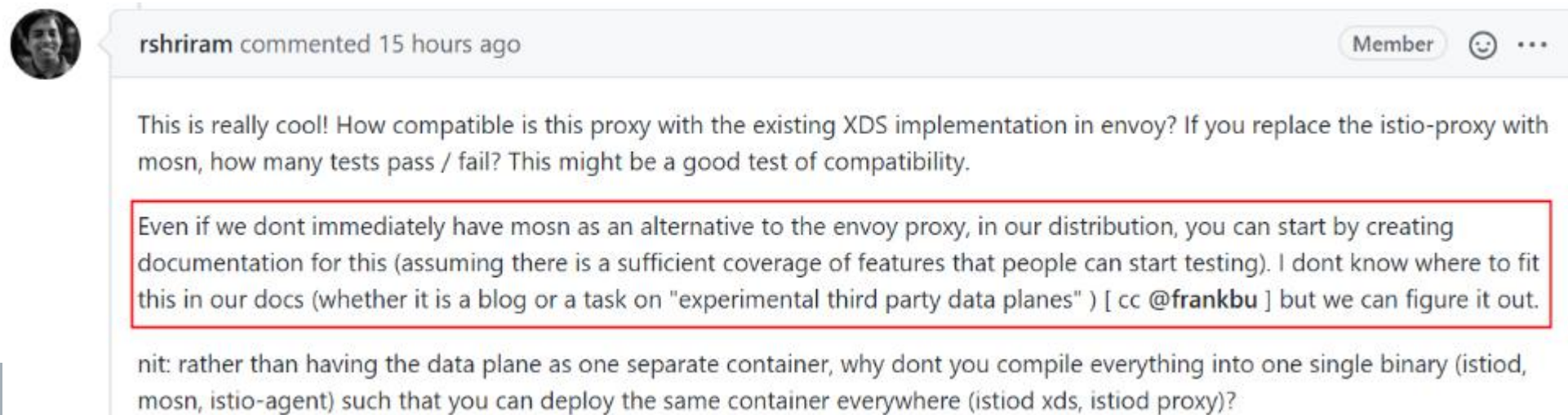
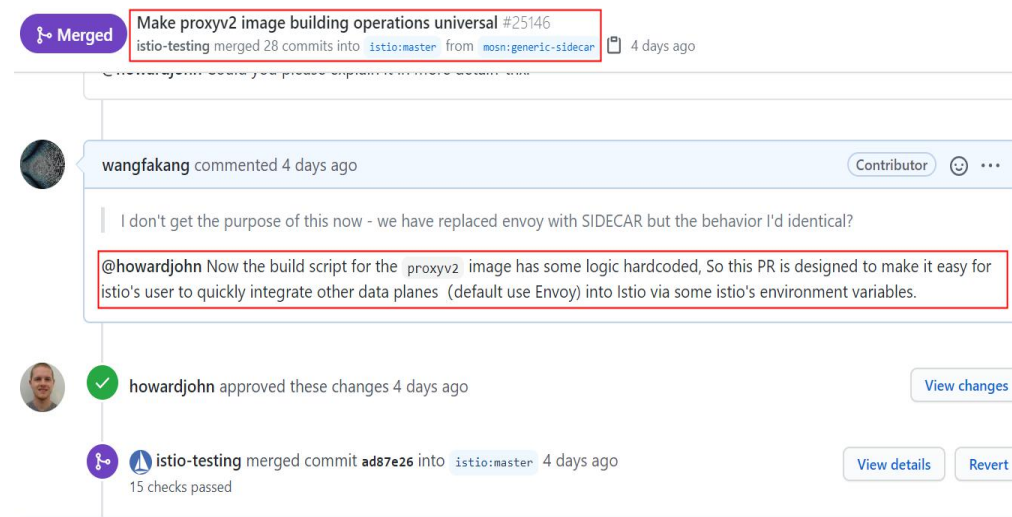
云原生演进 — 标准化演进

综合 MOSN 和 Istio 官方的讨论后，MOSN 社区主导并会参与 Istio 中数据面解耦的事情（比如测试集、镜像构建等），这样让 Istio 先变得更容易集成第三方的数据面，即 MOSN 社区的用户更方便的集成 Istio 使用。

[MOSN with Istio 适配的 Roadmap](#) 中新增如下事项：

- 推动 Istio 的镜像构建和数据面解耦，相关 [Issue](#)
- 推动 Istio 的测试框架和数据面解耦，相关 [Issue](#)

2020 年7月14号 Istio TOC（Istio 技术委员会）成员 [@Shriram Rajagopalan](#) 最新[回复](#)“也是支持 Istio 中支持多数数据面的方案，而且也建议先把 MOSN 做为实验性第三方数据平面纳入到 Istio 的官方博客中，方便用户来试用”

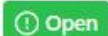


云原生演进 — 标准化演进

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



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
...
```


总结及展望

- 社区合作伙伴
- MOSN 云原生演进总结

合作伙伴及用户

Star 数: 2581
Contributor: 66
Committer : 7
意向合作企业: 14
意向合作社区: 5
企业用户: 20
4283 Commits, 20 Release

ISTIO / BLOG / 2020 POSTS

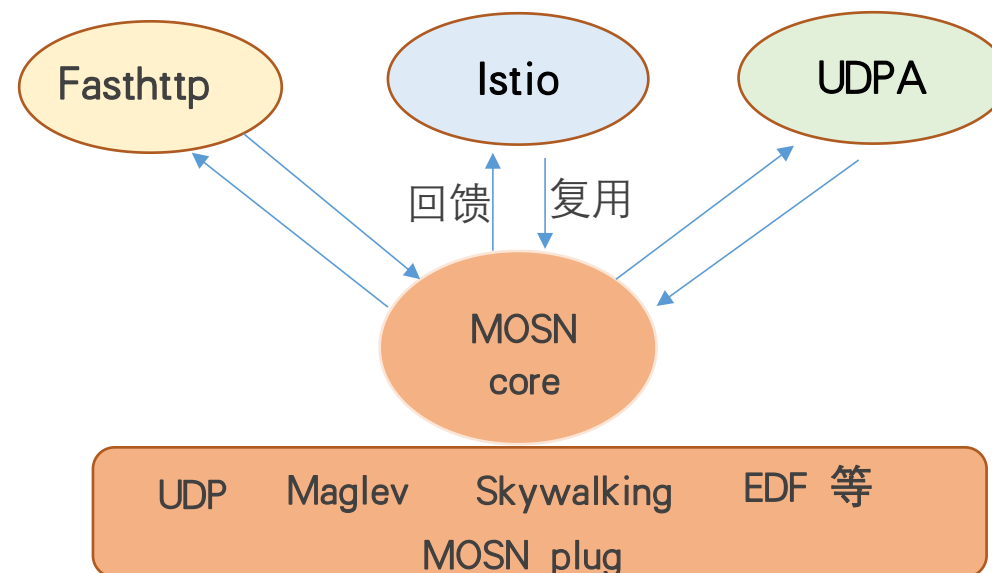
2020 Posts

Blog posts for 2020.

Using MOSN with Istio: an alternative data plane

July 28, 2020

An alternative sidecar proxy for Istio.



借力开源，反哺开源

在实践的道路上，一步步的标准化演进



总结&未来展望

云原生 = 标准 + 实践



Q&A

MOSN 官网 <http://mosn.io>
MOSN Github <http://github.com/mosn/mosn>
Service Mesh <https://www.servicemeshes.com>

MOSN 开源交流群



欢迎技术交流

