



SERVICE MESH
SUMMIT 2022
服务网格峰会

主办方



云原生社区
Cloud Native Community

如何基于Istio 实现Mesh自定义扩展功能

曾宇星 阿里云云原生架构师

自我介绍



- 阿里云技术专家、云原生架构师
- 长期从事服务端开发和架构工作，10 多年分布式领域后台开发经验，目前主要关注于云原生、高性能、高可用分布式架构。
- 有多年 Service Mesh 、Envoy 网关、Kubernetes 容器平台等云原生领域相关开发工作经验。目前在阿里云服务网格团队从事 Service Mesh 云产品研发和架构设计工作。

Agenda

- 为什么需要自定义扩展能力
- Istio 下几种自定义扩展方式详细介绍
- 开发/运维人员使用最佳实践
- 总结

为什么Mesh需要自定义扩展能力？

为什么Mesh需要自定义扩展能力？



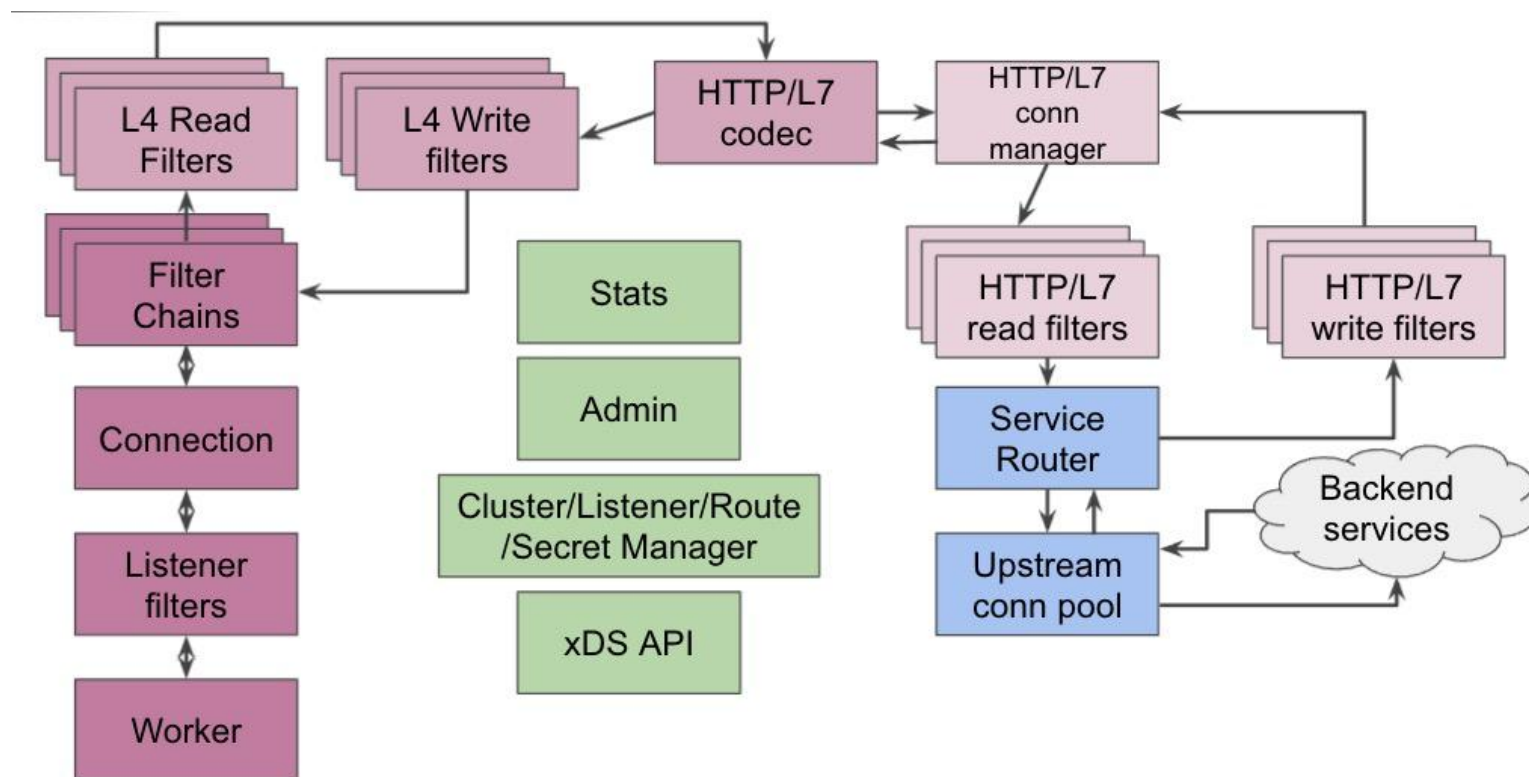
SERVICE MESH
SUMMIT 2022
服务网格峰会

- 业务应用的多样性和差异性，导致Mesh用户需求的多样性
- Istio 社区功能和用户实际需求有差异，需要补齐，特别是长尾非通用的小众需求

Envoy 扩展方式原理



SERVICE MESH
SUMMIT 2022
服务网格峰会



- Lua
- Wasm
- RPC 请求转发到外部进程

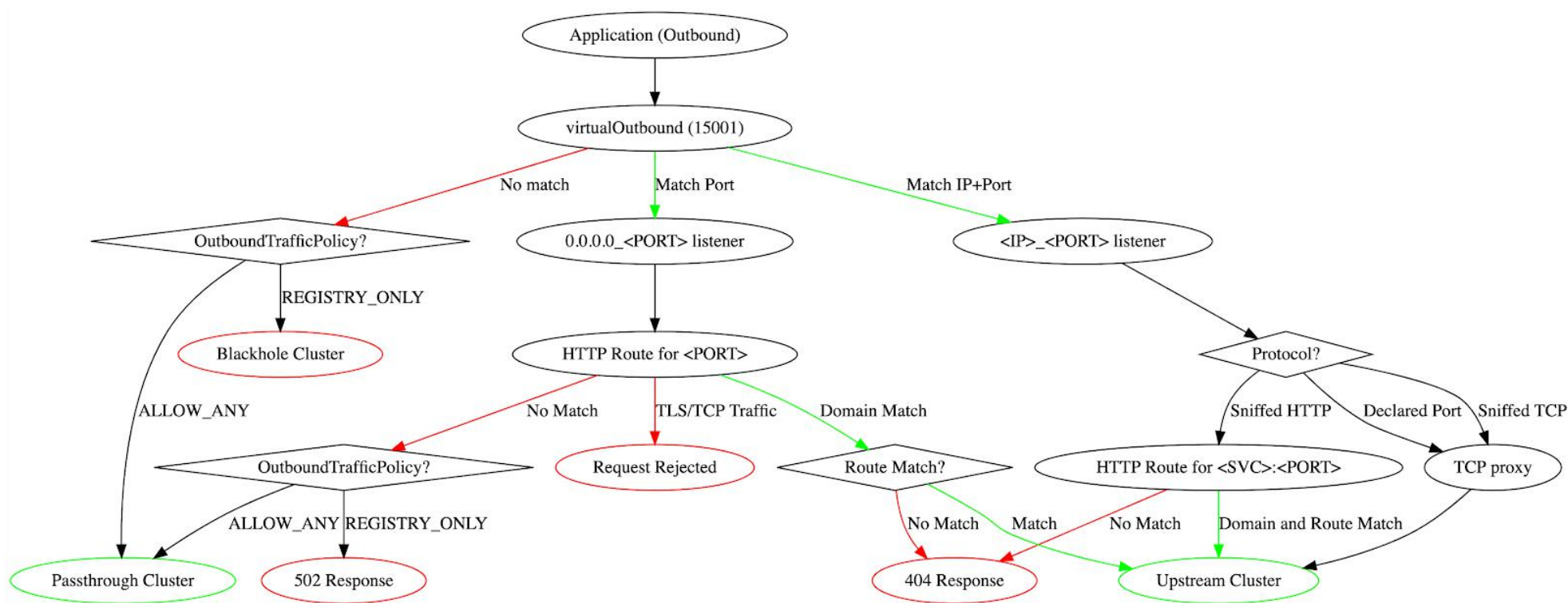


Istio 扩展方式

- EnvoyFilter API
- WasmPlugin API
- xDS 扩展（比如dubbo RDS)
- xDS 之ECDS
(type.googleapis.com/envoy.config.core.v3.TypedExtensionConfig)



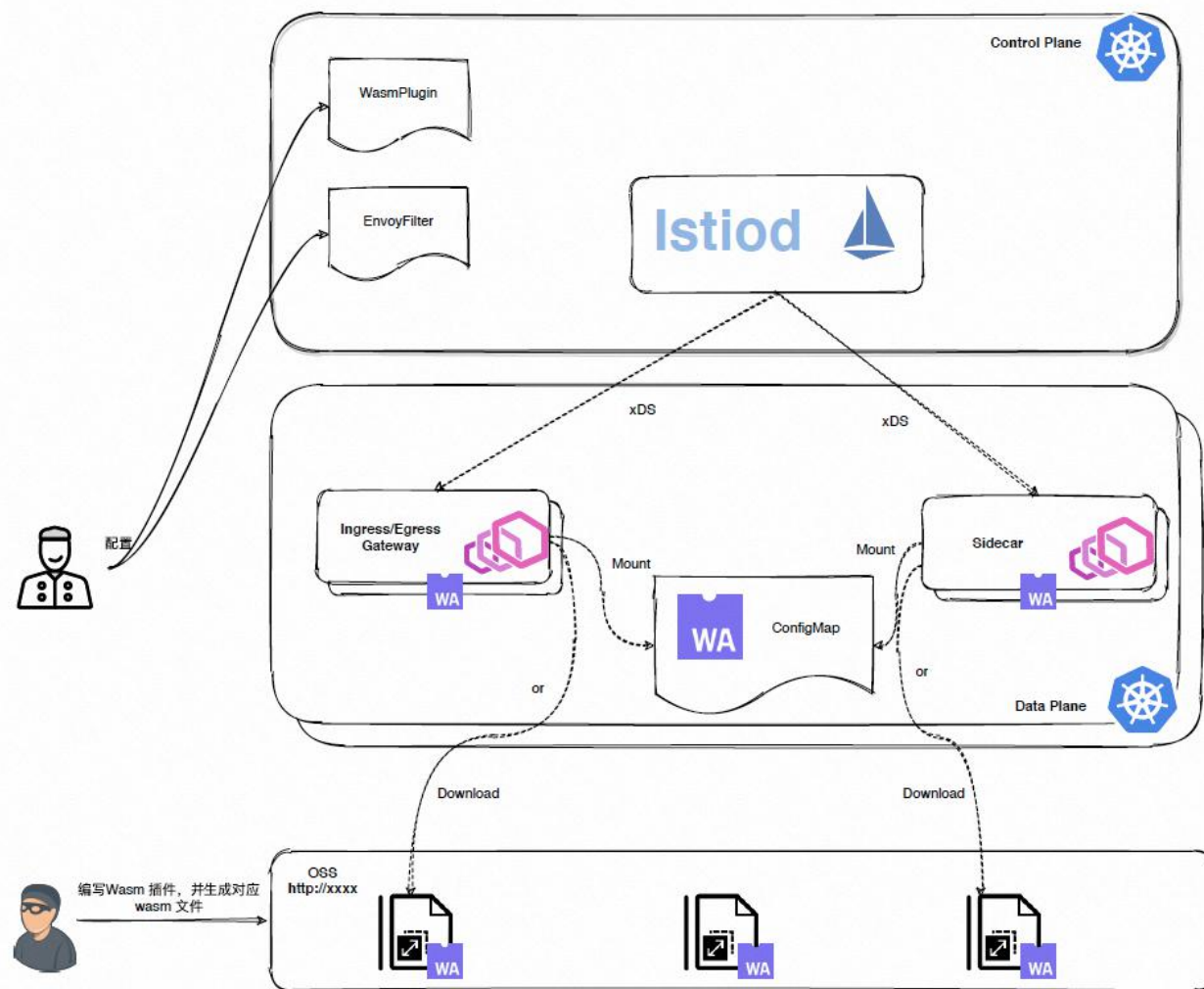
xDS data model



Istio 下自定义扩展方式详细介绍

自定义扩展方式之一：Wasm

- `kubectl create configmap new-filter \`
 `--from-file=new-filter.wasm=mycode.wasm`
- `kubectl create envoyfilter xxx` 或者
- `kubectl create wasmpugin xxx`

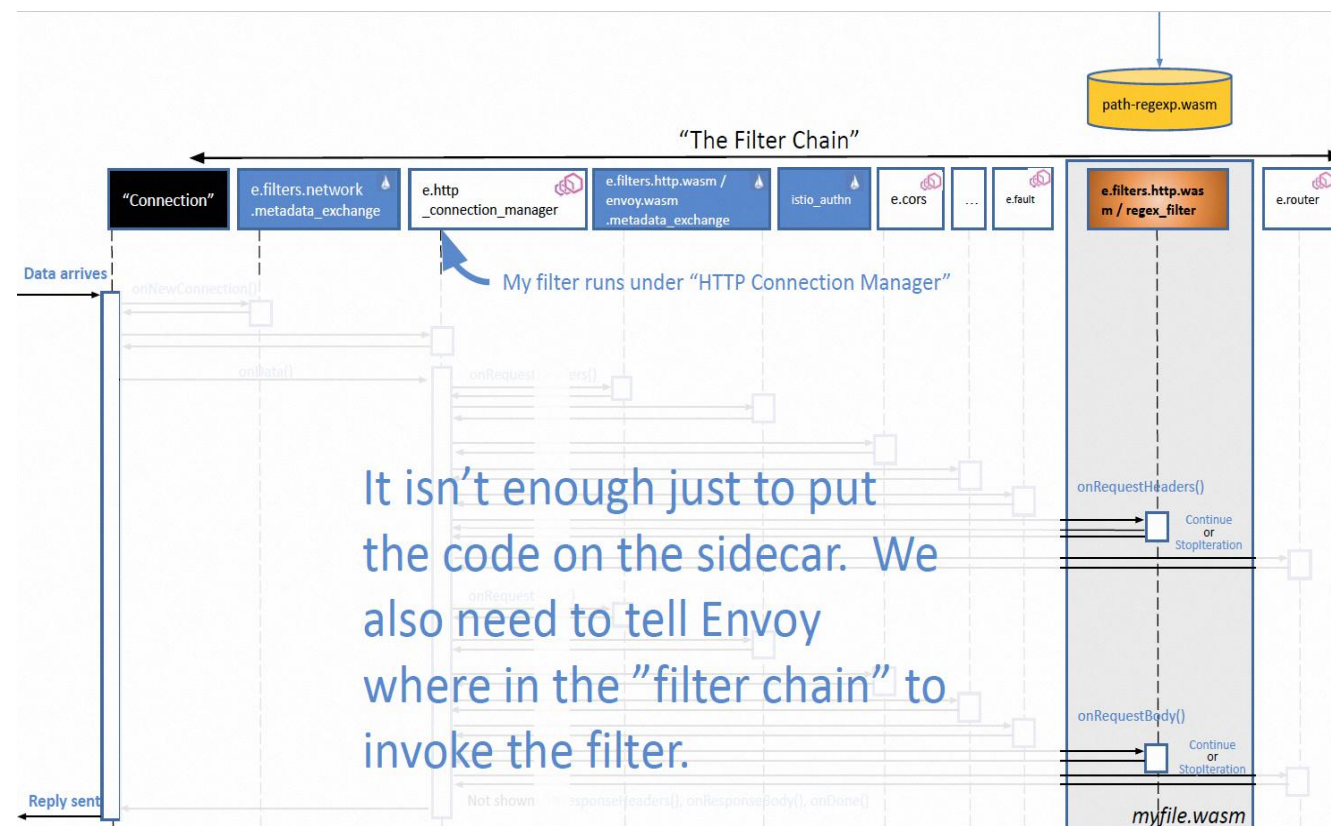


自定义扩展方式之一：数据面Wasm 支持



SERVICE MESH
SUMMIT 2022
服务网格峰会

- Wasm runtime
- 支持主流语言 C++、Go(TinyGo)、AssemblyScript、Rust
- 基于filter_chain 可配置生效位置



<https://istio.io/latest/docs/concepts/wasm/>

自定义扩展方式之一：控制面接口EnvoyFilter



SERVICE MESH
SUMMIT 2022
服务网格峰会

通用配置方式：

需要对数据面的配置格式有一定的了解

```
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
metadata:
  name: productpage-v1-examplefilter
  labels:
    asm-system: 'true'
    provider: asm
spec:
  configPatches:
    - applyTo: HTTP_FILTER
      match:
        context: SIDECAR_INBOUND
        proxy:
          proxyVersion: '^1\.*.*'
        listener:
          filterChain:
            filter:
              name: envoy.filters.network.http_connection_manager
              subFilter:
                name: envoy.filters.http.router
      patch:
        operation: INSERT_BEFORE
        value:
          typed_config:
            "@type": type.googleapis.com/envoy.extensions.filters.http.wasm.v3.Wasm
            config:
              name: example-filter
              rootId: my_root_id
              vmConfig:
                code:
                  local:
                    filename: /var/local/lib/wasm-filters/example-filter.wasm
                runtime: envoy.wasm.runtime.v8
                vmId: example-filter
                allow_precompiled: true
              name: envoy.filters.http.wasm
      workloadSelector:
        labels:
          app: productpage
          version: v1
```

锚点

patch 操作

wasm 插件文件

自定义扩展方式之一：控制面接口WasmPlugin



SERVICE MESH
SUMMIT 2022
服务网格峰会

Istio
原生
API

```
apiVersion: extensions.istio.io/v1alpha1
kind: WasmPlugin
metadata:
  name: openid-connect
  namespace: istio-ingress
spec:
  selector:
    matchLabels:
      istio: ingressgateway
  url: oci://private-registry:5000/openid-connect/openid:latest
  imagePullPolicy: IfNotPresent
  imagePullSecret: private-registry-pull-secret
  phase: AUTHN
  pluginConfig:
    openid_server: authn
    openid_realm: ingress
  vmConfig:
    env:
      - name: POD_NAME
        valueFrom: HOST
      - name: TRUST_DOMAIN
        value: "cluster.local"
```

生效范围

wasm 插件对应镜像

镜像获取 secret

插件配置

```
apiVersion: extensions.istio.io/v1alpha1
kind: WasmPlugin
metadata:
  name: openid-connect
  namespace: istio-ingress
spec:
  selector:
    matchLabels:
      istio: ingressgateway
  url: https://private-bucket/filters/openid.wasm
  imagePullPolicy: Always
  phase: AUTHN
  pluginConfig:
    openid_server: authn
    openid_realm: ingress
  vmConfig:
    env:
      - name: POD_NAME
        valueFrom: HOST
      - name: TRUST_DOMAIN
        value: "cluster.local"
```

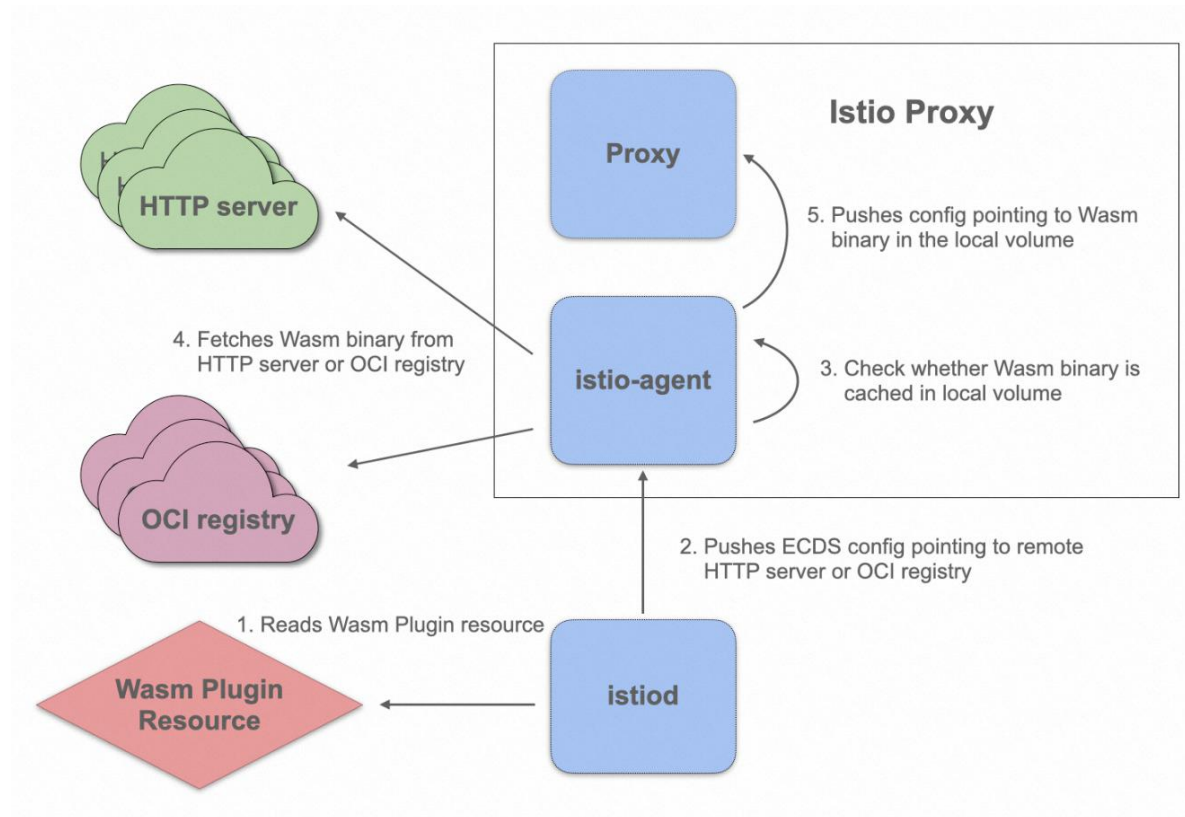
https://istio.io/latest/docs/reference/config/proxy_extensions/wasm-plugin/

自定义扩展方式之一：WasmPlugin 实现原理



SERVICE MESH
SUMMIT 2022
服务网格峰会

Istio
原生
API



https://istio.io/latest/docs/reference/config/proxy_extensions/wasm-plugin/

自定义扩展方式之二：Lua + EnvoyFilter



SERVICE MESH
SUMMIT 2022
服务网格峰会

SpringCloud 流量模型如何适配 istio ?

GET HTTP 1.1
Host: 10.0.0.222



GET HTTP 1.1
Host: provider.default.svc.cluster.local

```
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
metadata:
  labels:
    provider: "asm"
    asm-system: "true"
  name: nacos-subscribe-lua
  namespace: istio-system
spec:
  configPatches:
    # The first patch adds the lua filter to the listener/http connection manager.
    - applyTo: HTTP_FILTER
      match:
        proxy:
          proxyVersion: "^1.*"
          context: SIDECAR_OUTBOUND
          listener:
            portNumber: 8848
            filterChain:
              filter:
                name: "envoy.filters.network.http_connection_manager"
                subFilter:
                  name: "envoy.filters.http.router"
      patch:
        operation: INSERT_BEFORE
        value: # lua filter specification
        name: envoy.lua
        typed_config:
          "@type": "type.googleapis.com/envoy.extensions.filters.http.lua.v3.Lua"
          inlineCode: |
            -- copyright: ASM (Alibaba Cloud ServiceMesh)
            function envoy_on_request(request_handle)
              local request_headers = request_handle:headers()
              -- /nacos/v1/ns/instance/list?healthyOnly=false&namespaceId=public&clientIP=11.122.63.81&serviceName=DEFAULT_GROUP%40%40service-provider&udpPort=53174&encoding=UTF-8
              local path = request_headers:get(":path")
              if string.match(path, "/nacos/v1/ns/instance/list") then
                local servicename = string.gsub(path, ".*&serviceName=%40{[%w._\\-]+}&.*", "%1")
                request_handle:streamInfo():dynamicMetadata():set("context", "request.path", path)
                request_handle:streamInfo():dynamicMetadata():set("context", "request.servicename", servicename)
                request_handle:logInfo("subscribe for serviceName: " .. servicename)
              else
                request_handle:streamInfo():dynamicMetadata():set("context", "request.path", "")
              end
            end
            function envoy_on_response(response_handle)
              local request_path = response_handle:streamInfo():dynamicMetadata():get("context")["request.path"]
              if request_path == "" then
                return
              end
              local servicename = response_handle:streamInfo():dynamicMetadata():get("context")["request.servicename"]
              response_handle:logInfo("modified response ip to serviceName: " .. servicename)
              local bodyObject = response_handle:body(true)
              local body = bodyObject:getBytes(0, bodyObject:length())
              body = string.gsub(body, "%s+", "")
              body = string.gsub(body, "(ip:\"\\\")(%d+.%d+.%d+.%d+)\", \"%1\"..servicename)
              response_handle:body():setBytes(body)
            end
```

- 依赖Envoy Lua filter
- EnvoyFilter patch 配置
- Example: 使用Lua 脚本支持SpringCloud 上Mesh

自定义扩展方式之二：Lua + EnvoyFilter

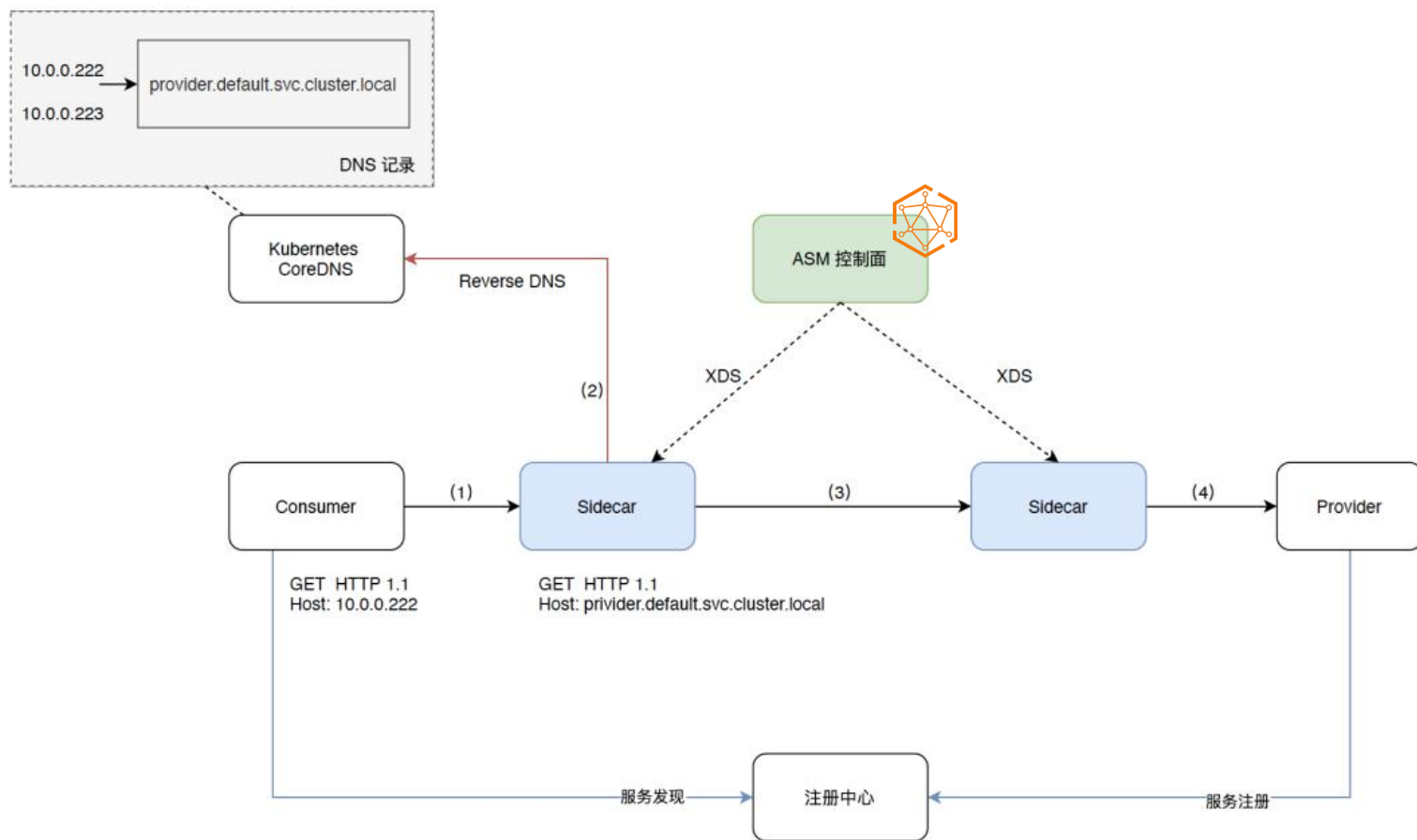


SERVICE MESH
SUMMIT 2022
服务网格峰会

```
inlineCode: |
-- copyright: ASM (Alibaba Cloud ServiceMesh)
function envoy_on_request(request_handle)
    local request_headers = request_handle:headers()
    -- /nacos/v1/ns/instance/list?healthyOnly=false&namespaceId=public&clientIP=11.122.63.81&serviceName=DEFAULT_GROUP%
    local path = request_headers:get(":path")
    if string.match(path, "^/nacos/v1/ns/instance/list") then
        local servicename = string.gsub(path, ".*&serviceName.*40([%w.\\_\\-]+)&.*", "%1")
        request_handle:streamInfo():dynamicMetadata():set("context", "request.path", path)
        request_handle:streamInfo():dynamicMetadata():set("context", "request.servicename", servicename)
        request_handle:logInfo("subscribe for serviceName: " .. servicename)
    else
        request_handle:streamInfo():dynamicMetadata():set("context", "request.path", "")
    end
end
function envoy_on_response(response_handle)
    local request_path = response_handle:streamInfo():dynamicMetadata():get("context")["request.path"]
    if request_path == "" then
        return
    end
    local servicename = response_handle:streamInfo():dynamicMetadata():get("context")["request.servicename"]
    response_handle:logInfo("modified response ip to serviceName:" .. servicename)
    local bodyObject = response_handle:body(true)
    local body= bodyObject:getBytes(0,bodyObject:length())
    body = string.gsub(body, "%s+", "")
    body = string.gsub(body, "(ip\\:\\\\)(%d+.%d+.%d+.%d+)", "%1"..servicename)
    response_handle:body():setBytes(body)
end
```

https://help.aliyun.com/document_detail/383257.html

自定义扩展方式之三样例：支持 SpringCloud



自定义扩展方式之三样例：支持 SpringCloud



SERVICE MESH
SUMMIT 2022
服务网格峰会

- EnvoyFilter patch 配置
- Example:
使用com.aliyun.reverse_dns filter 支持SpringCloud

```
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
metadata:
  labels:
    provider: "asm"
    asm-system: "true"
  name: any-spring-cloud-support
  namespace: istio-system
spec:
  configPatches:
  - applyTo: HTTP_FILTER
    match:
      proxy:
        proxyVersion: "^1.*)"
        context: SIDECAR_OUTBOUND
      listener:
        portNumber: 8070
        filterChain:
          filter:
            name: "envoy.filters.network.http_connection_manager"
            subFilter:
              name: "envoy.filters.http.router"
    patch:
      operation: INSERT_BEFORE
      value: # reverse_dns filter specification
      name: com.aliyun.reverse_dns
      typed_config:
        "@type": "type.googleapis.com/udpa.type.v1.TypedStruct"
        type_url: type.googleapis.com/envoy.config.filter.reverse_dns.v3alpha.CommonConfig
        value:
          pod_cidrs:
            - "10.0.128.0/18"
```

扩展的 native Filter : com.aliyun.reverse_dns

https://help.aliyun.com/document_detail/383257.html

EnvoyFilter 的不足



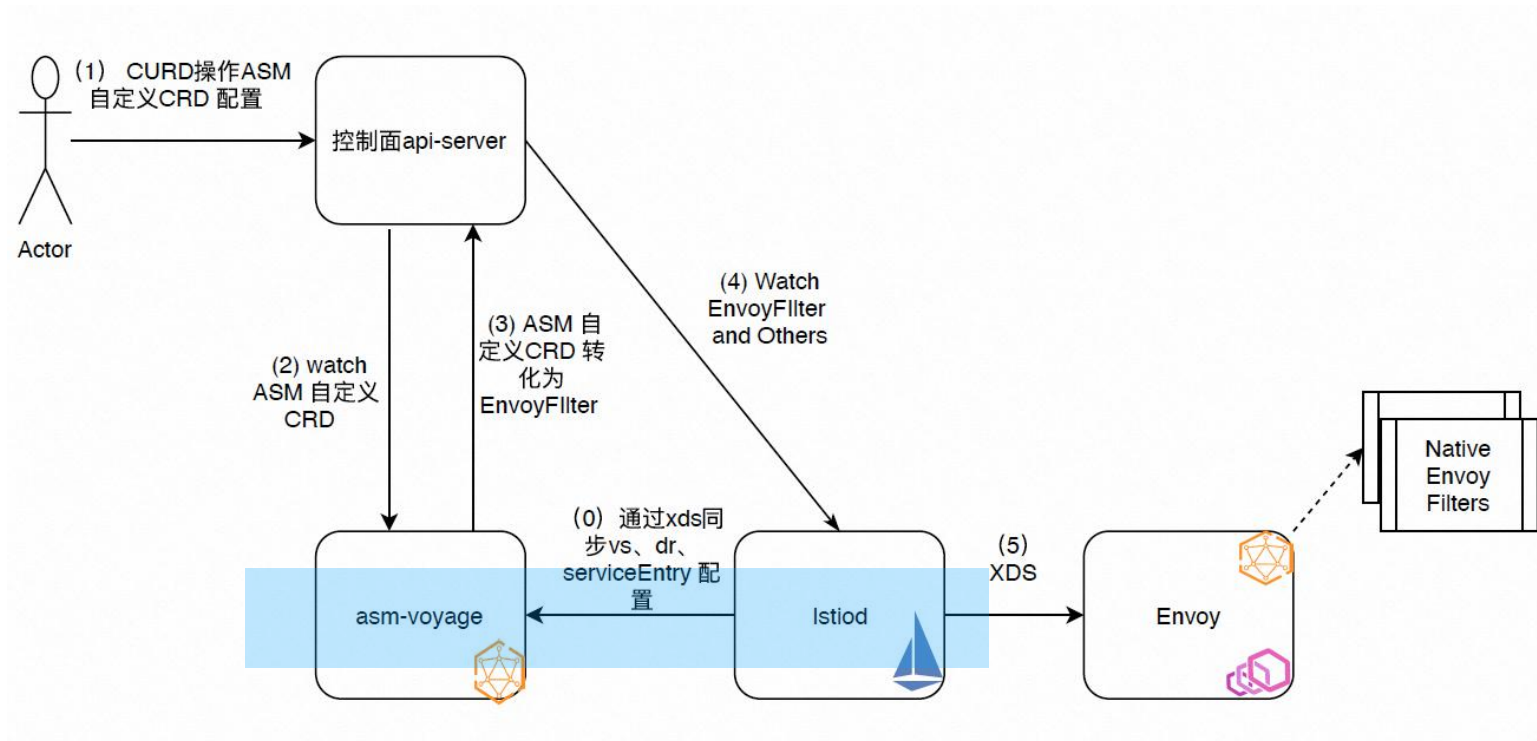
SERVICE MESH
SUMMIT 2022
服务网格峰会

- ✓ EnvoyFilter 功能很强大，但也非常具有破坏性，需要使用者对数据面Envoy 对应配置文件有一定的了解
- ✓ EnvoyFilter 的patch 顺序很脆弱，和创建时间相关
- ✓ Istiod 只能对EnvoyFilter 进行非常有限的验证

自定义扩展方式之三：Envoy Native Filter + Voyage



SERVICE MESH
SUMMIT 2022
服务网格峰会



asm-voyage 是阿里云ASM 的一个控制面扩展组件

自定义扩展方式之三：控制面CRD 抽象



SERVICE MESH
SUMMIT 2022
服务网格峰会

- EnvoyFilter patch 配置语义复杂，难度大
- 通过自定义CRD简化配置，方便用户配置，降低使用门槛

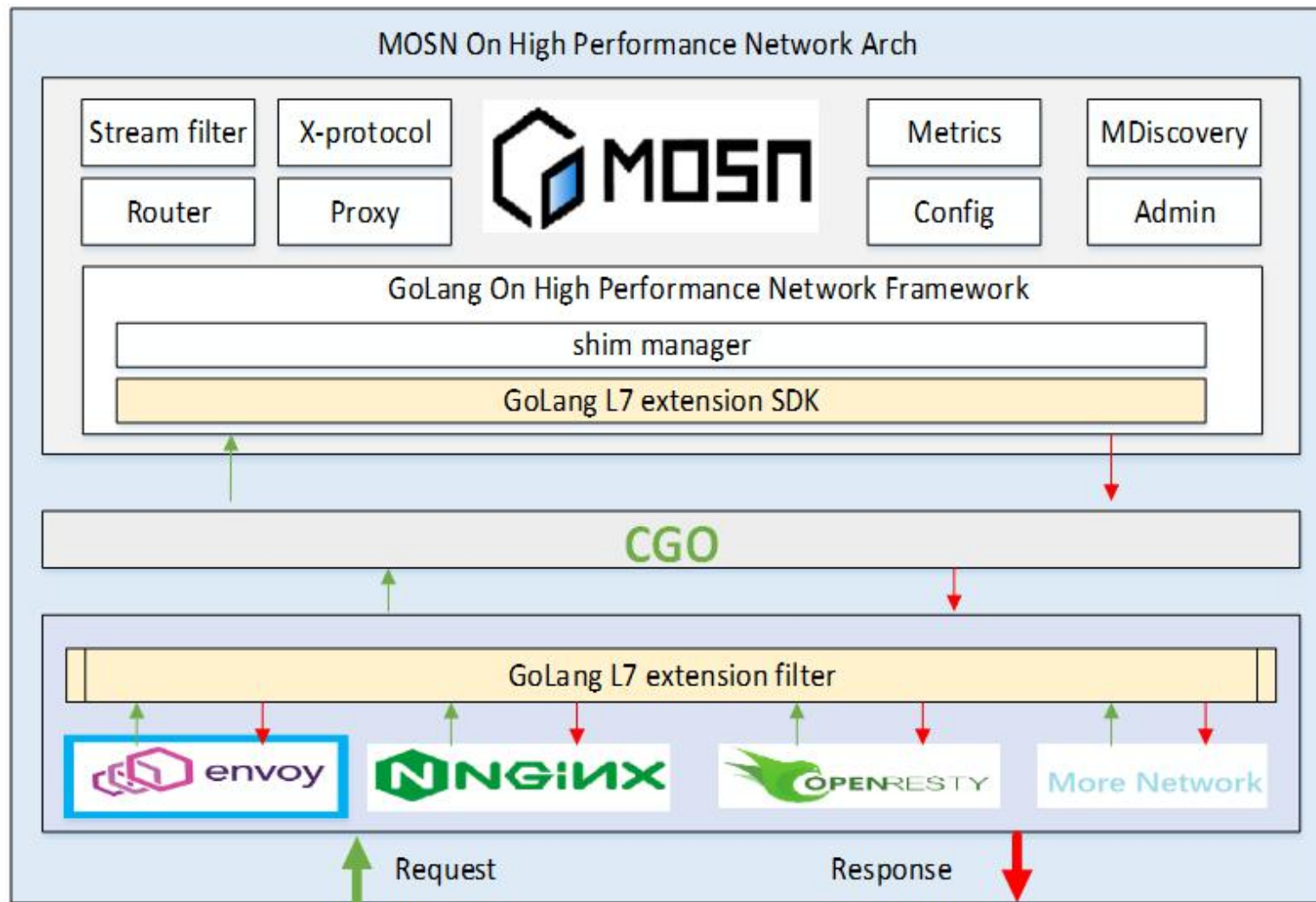
```
apiVersion: istio.alibabacloud.com/v1beta1
kind: ASMLocalRateLimiter
metadata:
  name: for-api-test
  namespace: default
spec:
  workloadSelector:
    labels:
      app: istio-ingressgateway
  isGateway: true
  configs:
    - match:
        vhost:
          name: "www.example1.com"  ## 如果gateway中配置了多个host域名，填最后一个即可。
          port: 80
          route:
            name_match: "test1"  ##VirtualService路由配置中对应route的name,若VirtualService路由配置下没有对应name的路由，则不生效。
        limit:
          fill_interval:
            seconds: 1
          quota: 10
    - match:
        vhost:
          name: "www.example2.com"
          port: 80
          route:
            name_match: "test1"
        limit:
          fill_interval:
            seconds: 1
          quota: 100
```

https://help.aliyun.com/document_detail/407660.html
<https://istio.io/latest/docs/tasks/policy-enforcement/rate-limit/>

自定义扩展方式之四：MOE



SERVICE MESH
SUMMIT 2022
服务网格峰会

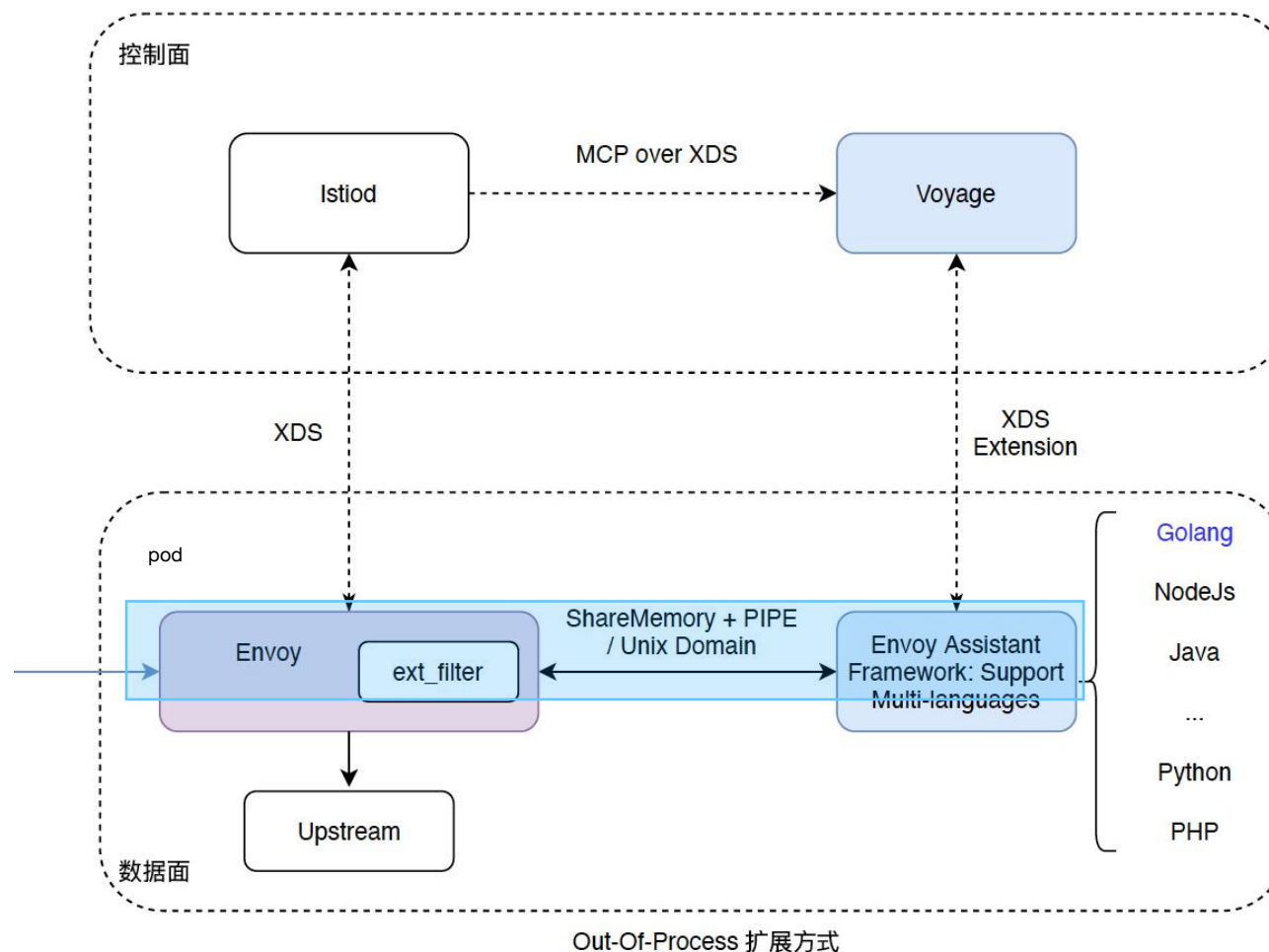


自定义扩展方式之五：Out Of Process 方式



SERVICE MESH
SUMMIT 2022
服务网格峰会

非侵入式扩展Istio
和 Envoy

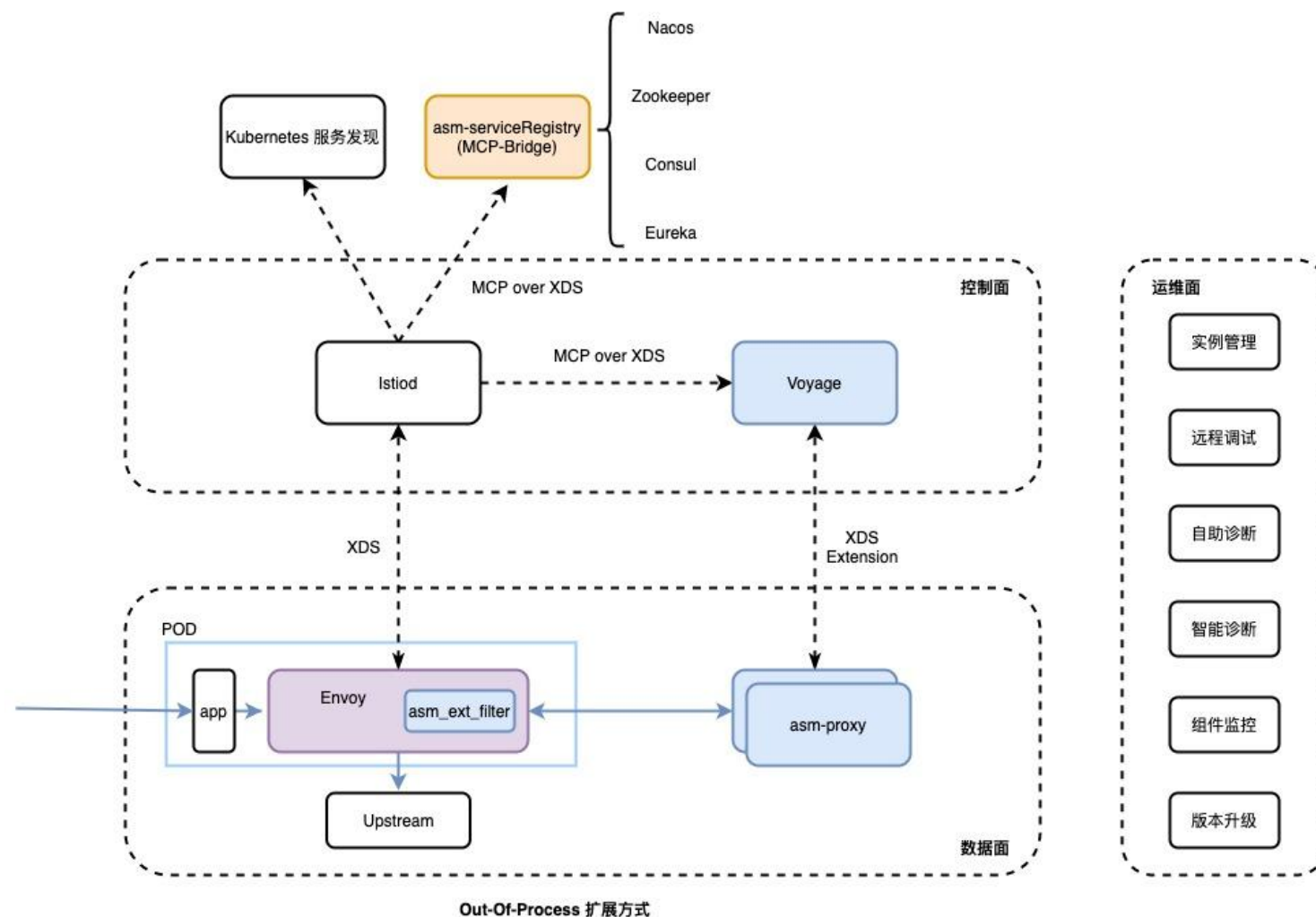


自定义扩展方式之五：Out Of Process 方式



SERVICE MESH
SUMMIT 2022
服务网格峰会

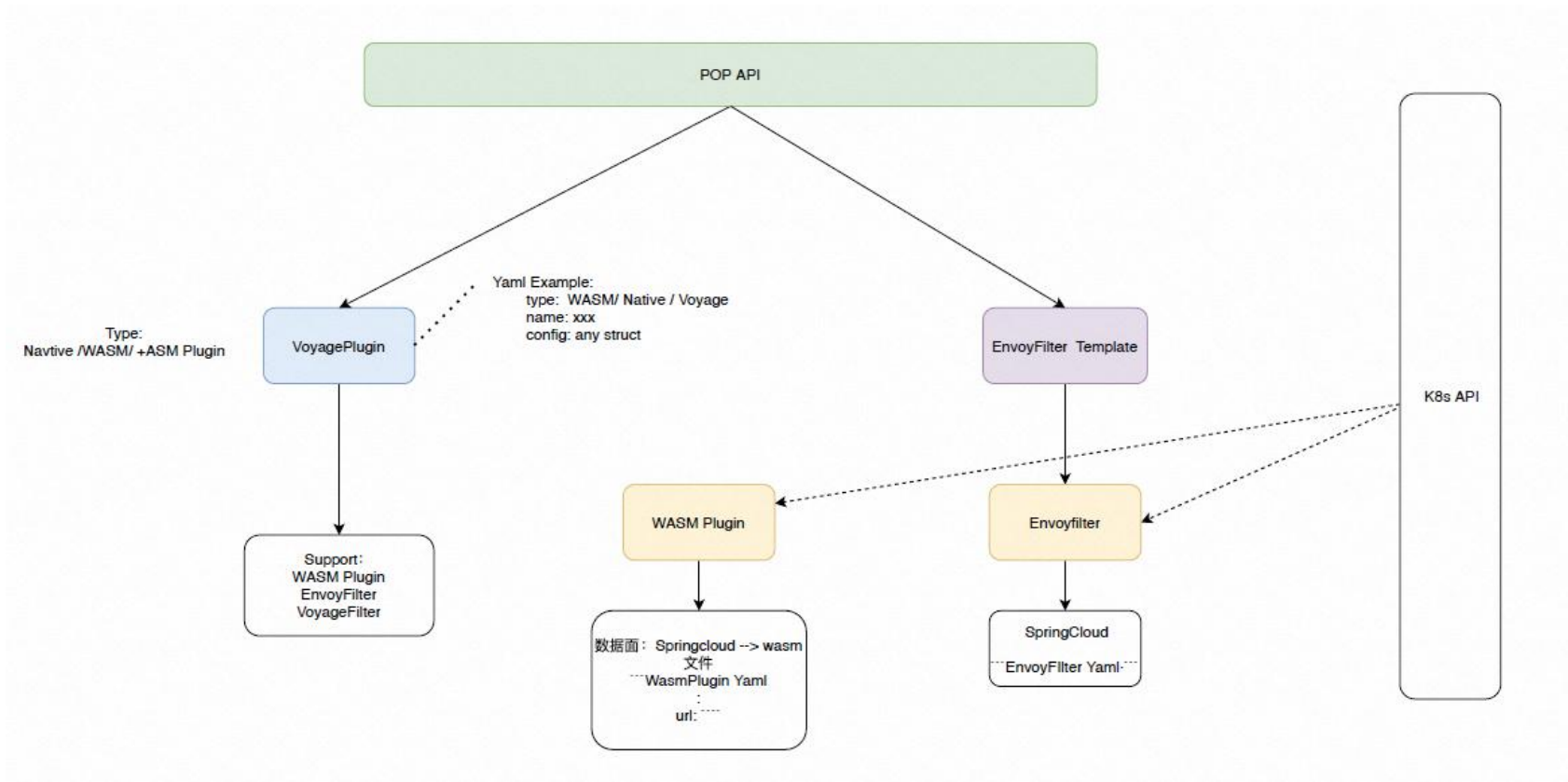
扩展的asm-proxy
可以有多种部署方
式



自定义扩展产品化: 以插件形式进行产品化封装



SERVICE MESH
SUMMIT 2022
服务网格峰会



EnvoyFilter 原生API 使用，针对插件执行顺序、插件配置等具有一定的复杂度，不太适合直接面向普通用户，生产环境使用需要有封装过的产品能力

开发/运维人员使用最佳实践

适合自己的才是最好的



SERVICE MESH
SUMMIT 2022
服务网格峰会

扩展方式	Native C++	Lua	WASM	Out Of Process方式 (Envoy Assistant Framework)
性能	优	较优	一般	较优
开发效率	需要有一定的c++ 技术栈储备，是否有c++ 开发人员是关键	优，上手很快	较优（支持多语言）	框架有学习成本，较优，支持多语言
成熟度	优	优	一般	仅阿里云内部，还未开放
支持热加载	否，需重新构建镜像	是	是	否，需重新构建镜像（集中式部署方式除外）
社区	活跃（需具备c++ 知识储备）	一般	活跃	仅阿里云内部，还未开放

- 对性能要求不苛刻的场景小功能可以使用Lua、Wasm 插件方式
- 对性能有一定要求，且逻辑较复杂的场景建议使用C++ Native Filter 或者采用Out Of Process方式

使用阿里云服务网格构筑企业级能力



SERVICE MESH
SUMMIT 2022
服务网格峰会

用户界面/被集成能力: Web控制台v2.0/Open API/SDK

声明式云原生API, 兼容社区Istio, 支持数据面KubeAPI 操作控制面资源



托管ASM控制面核心组件-标准/pro版架构统一 柔性架构、多版本支持、定制能力增强

托管核心组件ASM
Infra

流量管理&
协议增强

零信任安全

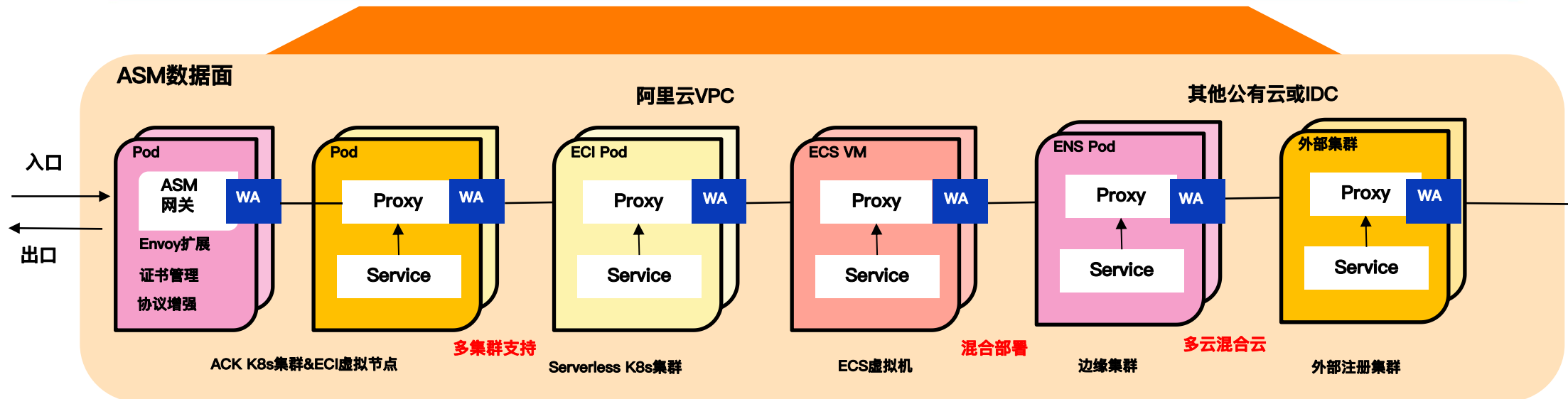
xDS推送优化

多注册中心

可观测性/
弹性伸缩

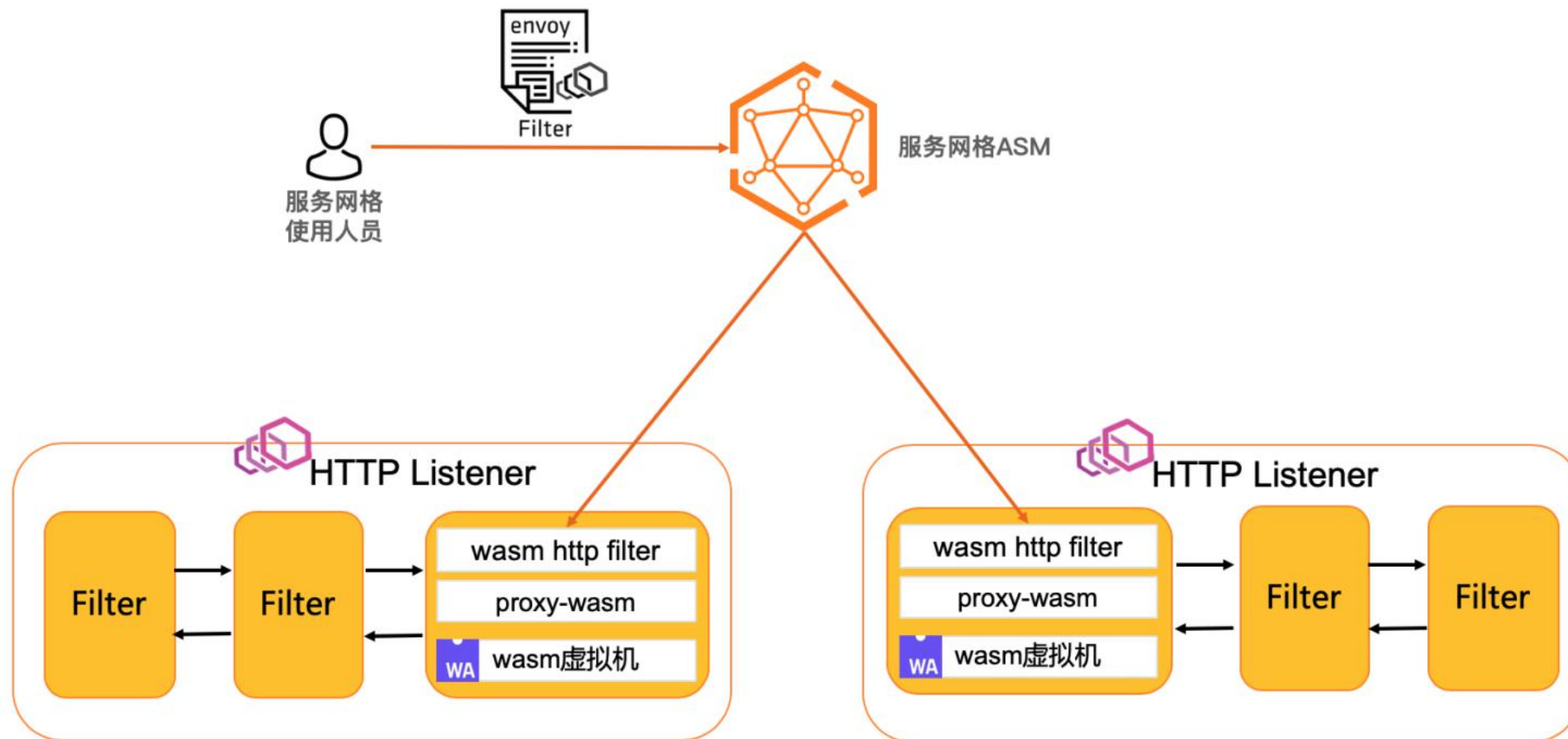
网络诊断
智能分析

插件扩展中心



<https://www.aliyun.com/product/servicemesh>

使用WASM 进行扩展ASM 能力



ASM 插件市场



SERVICE MESH
SUMMIT 2022
服务网格峰会

←

网络实例 ^

基本信息

全局命名空间

升级管理

网络状态

网络诊断

授权信息

集群与工作负载管理 v

Sidecar管理(数据面) v

服务管理

ASM网关

流量管理中心 v

可观测管理中心 v

网络安全中心 v

插件扩展中心 ^

插件市场

Envoy过滤器模板

Envoy过滤器

网络优化中心 v

服务网格 ASM / demo / 插件市场

插件市场

名称 请输入

支持Spring Cloud服务 未启用

v1

该Envoy过滤器模板通过令Sidecar拦截Spring Cloud服务注册流程来帮助服务网格支持Spring Cloud服务。
[详情查看: 管理Spring Cloud服务](#)

注意: 因为需要拦截注册流程, 您需要在Deployment创建之前创建此Envoy过滤器模板并绑定至istio-system命名空间。若某些业务Deployment先于EnvoyFilter创建, 您需要滚动更新该业务Deployment。

设置http2的初始流量窗口大小 已启用

v1

通过设置连接级别或者stream级别的滑动窗口大小, 可以在某些http2/gprc场景优化envoy的内存占用

在访问日志中打印HTTP body 未启用

v1

该Envoy过滤器模板可以通过让Sidecar拦截并解析请求的HTTP body, 进而在访问日志中输出服务所收到请求的HTTP body。
此功能需要与[自定义访问日志格式](#)搭配使用, 您需要在访问日志格式中增加变量值为DYNAMIC_METADATA(envoy.lua)%的访问日志变量(变量名称可随意设定), 才能在访问日志中查看请求的HTTP body。

保留请求与响应头部大小写 未启用

v1

在处理HTTP/1.1请求时, Envoy默认会将请求与响应的头部的Key都转化为小写, 这对于一些依赖头部大小写的的应用产生了很大影响。
该Envoy过滤器能够让网关与Sidecar保持HTTP请求/响应头部的大小写, 解决这个问题。模板创建后, 建议您将模板绑定至istio-system命名空间, 以保证请求链路上的每个Sidecar都不修改头部大小写。

开启AHAS流控 未启用

v1

开通AHAS限流功能

添加HTTP响应头 未启用

v1

该Envoy过滤器模板通过在应用程序中添加HTTP响应头来提高Web应用程序的安全性。默认提供了HTTP响应头的基准配置, 您也可以在填充模板参数时添加自定义响应头, 或覆盖默认提供的响应头。注意: 该Envoy过滤器模板需要被绑定至istio-system命名空间下的ASM网关服务才可生效。
[详情查看: 在ASM中通过EnvoyFilter添加HTTP响应头](#)

在响应头中增加请求头信息 未启用

v1

该Envoy过滤器模板可以在HTTP请求响应返回指定的若干状态码时, 在响应头中增加请求头的信息。

直接响应 未启用

v1

对于发往指定服务的指定路径的http请求, 不再向服务转发请求, 而是立即返回固定的响应内容。
此Envoy过滤器模板可以绑定至ASM网关或注入Sidecar的工作负载。当绑定至网关时, 将对请求的域名、端口、路径进行匹配; 当绑定至注入Sidecar的工作负载时, 将对请求发往的端口和路径进行匹配。

设置allow_connect为true允许升级的协议连接 未启用

v1

默认情况下, HTTP/2对WebSocket的支持是关闭的, 该Envoy过滤器模板设置allow_connect为true允许升级的协议连接, 以此支持WebSocket在HTTP/2流上进行隧道传输。
[详情查看: 在ASM中使用WebSocket协议访问服务](#)

<https://servicemesh.console.aliyun.com/>

ASM 插件市场



SERVICE MESH
SUMMIT 2022
服务网格峰会

服务网格 ASM / demo / 插件市场

← 插件详情



插件名称 设置http2的初始流量窗口大小
插件类型 EnvoyFilterTemplate
插件优先级 0(修改优先级)

插件版本 v1
插件状态 已启用

使用指引

插件配置

插件生效范围

☐ 全局生效

插件能力将在所有的工作负载中生效

☒ 工作负载生效

可以灵活选择让插件生效于指定标签选择的工作负载，或整个命名空间中的工作负载

☐ 网关生效

可以选择让插件生效于ASM网关

添加工作负载到生效范围

添加命名空间到生效范围

已生效的Sidecar

名称	命名空间	类型	匹配标签	操作
kube-state-metrics	arms-prom	Service	k8s-app:kube-state-metrics	解绑

插件配置

YAML

[更新](#)

```
1 patch_context: ANY
2 port_number: '80'
3 initial_connection_window_size: '65536'
4 initial_stream_window_size: '65536'
5
```

生效开关



ASM 网格诊断



**SERVICE MESH
SUMMIT 2022**
服务网格峰会

网格诊断

命名空间		诊断时间: 2022年9月20日 09:59:00 运行诊断成功 运行	
序号	检查项	结果	详情
6	网关选择器检查 🔗	🔴 找不到与网关选择器匹配的工作负载	在命名空间default中找不到网关simple-springboot-gateway 选择器匹配的工作负载。请首先检查是否以"istio: 网关名称"或"app: istio-网关名称"对网关工作负载进行匹配, 再确认指定的网关工作负载是否存在。
28	默认路由检查 🔗	🔵 建议为每个虚拟服务的路由规则提供默认路由	虚拟服务的http路由配置建议: 命名空间default中的bookinfo最好提供不包含任何match字段的默认路由, 以保证所有流量都能够找到转发目标
28	默认路由检查 🔗	🔵 建议为每个虚拟服务的路由规则提供默认路由	虚拟服务的http路由配置建议: 命名空间default中的springboot-istio-client-vs最好提供不包含任何match字段的默认路由, 以保证所有流量都能够找到转发目标
25	网关规则引用检查 🔗	🔴 虚拟服务引用的网关规则不存在	命名空间default中虚拟服务bookinfo指定的网关规则bookinfo-gateway不存在。 请检查您的网关名称或创建相应的网关
32	虚拟服务路由多端口检查 🔗	🟡 虚拟服务的路由目标需要指定端口	虚拟服务springboot-istio-server-vs路由到公开多个端口18080,18888的服务spring-boot-istio-server。 需要在目标中指定端口以消除歧义
31	虚拟服务路由目标检查 🔗	🟡 没有找到虚拟服务的路由目标	未能找到在虚拟服务springboot-istio-server-vs的路由目标中引用的服务spring-boot-istio-server; subset: \$userdefine1。 建议您为服务所在命名空间开启Sidecar自动注入, 再检查该服务是否存在。
1	数据面组件版本检查 🔗	🟢 通过	
2	服务端口检查 🔗	🟢 通过	
3	app及version标签检查 🔗	🟢 通过	
4	网关规则端口冲突检查 🔗	🟢 通过	
5	网关规则TLS证书冲突检查 🔗	🟢 通过	
7	网关引用的证书对应的Secret格式是否正确 🔗	🟢 通过	
8	网关引用的证书对应的Secret是否存在 🔗	🟢 通过	
9	目标规则的命名空间检查 🔗	🟢 通过	
10	网关Pod unprivileged模式端口检查 🔗	🟢 通过	
11	数据面的命名空间注入标签值与控制面是否同步 🔗	🟢 通过	
12	控制面是否存在非ASM提供的EnvoyFilters 🔗	🟢 通过	
13	授权策略中的命名空间引用检查 🔗	🟢 通过	
14	工作负载引用检查 🔗	🟢 通过	
15	字段弃用检查 🔗	🟢 通过	
16	忽略字段检查 🔗	🟢 通过	

总结



- 基于Istio&Envoy 提供的可扩展能力，可以满足用户不同业务场景需求。
- 开发/运维人员可以根据实际情况选择适合自身团队的扩展方式。
- 使用和运维Istio 具有一定门槛和复杂度，需要进行产品能力封装，推荐使用阿里云产品ASM



SERVICE MESH
SUMMIT 2022
服务网格峰会

感谢观看



云原生社区
Cloud Native Community

活动由云原生社区主办

