# eBay基于Istio的应用网关的生产化实践

**陈佑雄、eBay上海 、Network SIG/Service-Mesh**

# 议程

数据中心流量管理现状

基于Istio的应用网关实践

- Istio部署模式
- 应用高可用接入架构
- 流量管理模型

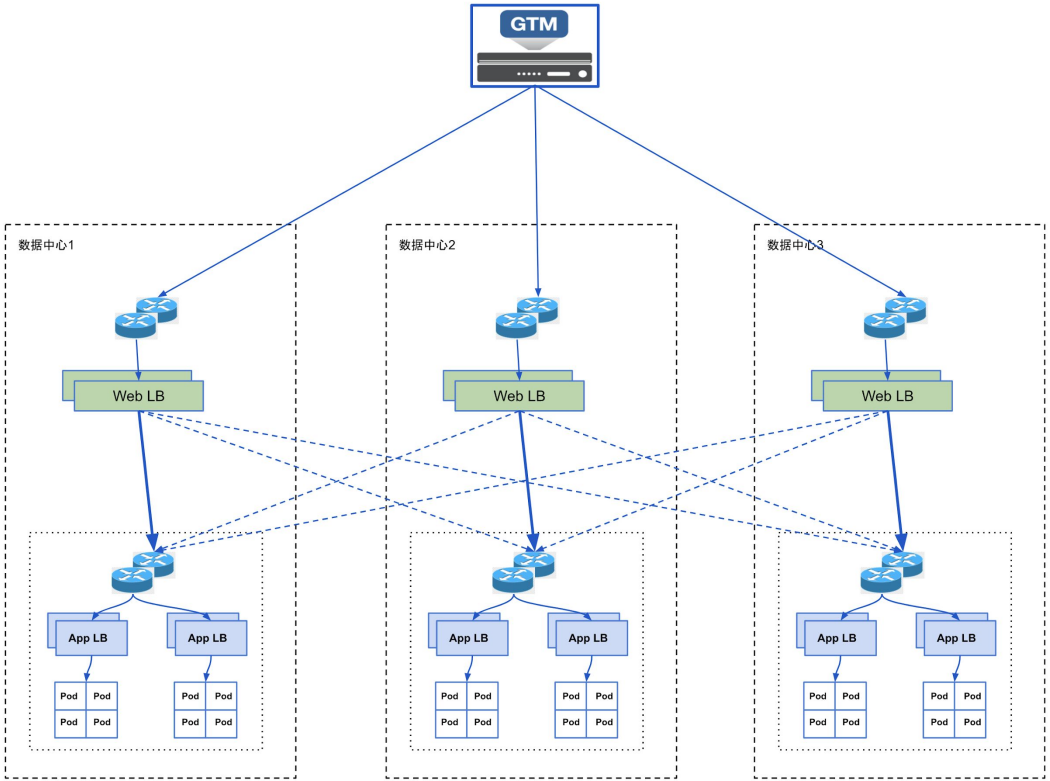
Istio应用中遇到的挑战

未来展望

# 数据中心流量管理现状

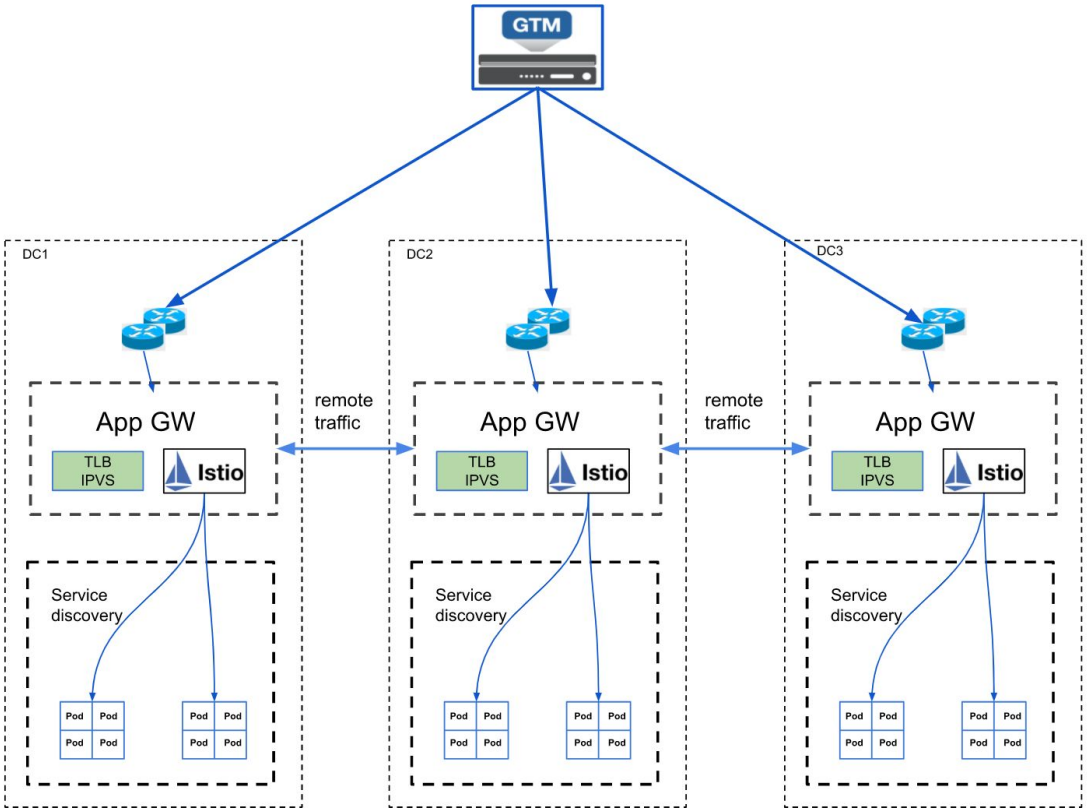
- 生产应用跨三个数据中心部
- 2000多对硬件负载均衡设备
- 数据中心内部应用多副本部署并配置应用层负载均衡
- 服务间调用以南北流量为主
- Web层LB流量管理
  - 99%请求转入本地数据中心
  - 1%请求跨数据中心
- 数据中心负载均衡特征：
  - 微服务架构，VIP数量众多
  - 同时具有公网和内网VIP
  - VIP上配置有少量L7规则

应用集群后端服务器整体宕机不会造成数据面影响



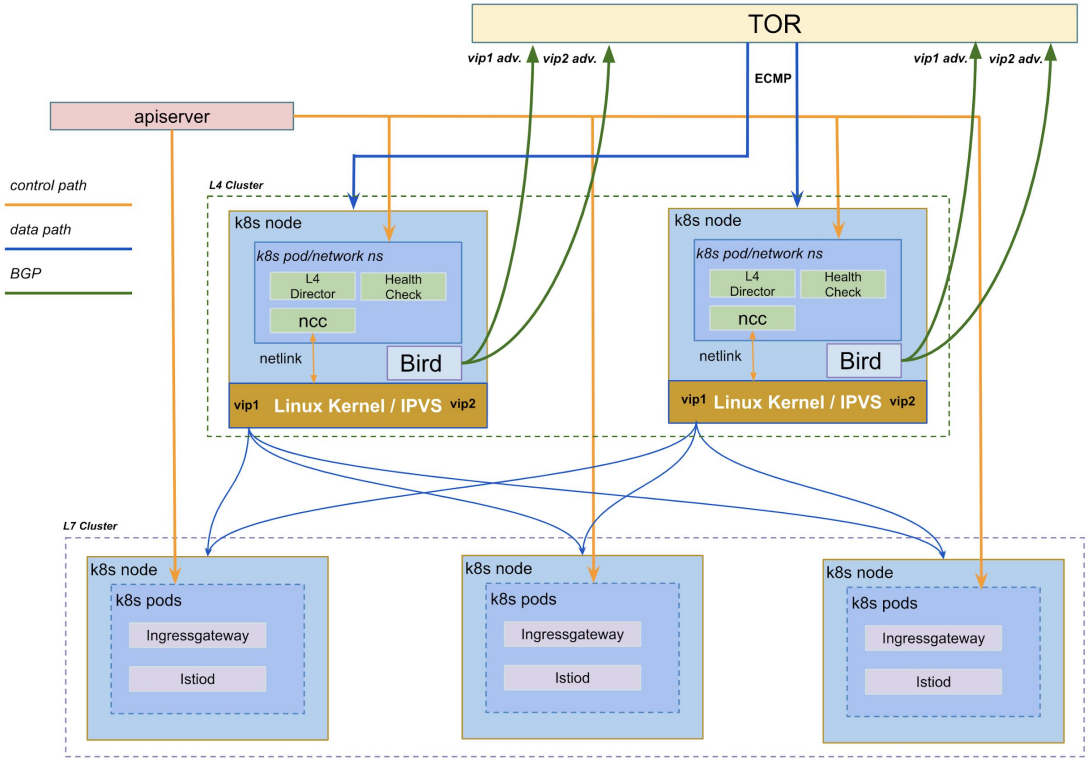
# 数据中心云原生网络目标

- 高安全
  - 支持全链路加密
  - 支持认证，授权
  - 抵御已知的网络安全威胁
- 高吞吐
  - 弹性扩容
  - 避免流量过载
- 高可用
  - 故障域为N+1
  - 跨数据中心故障转移
  - 精准流量控制



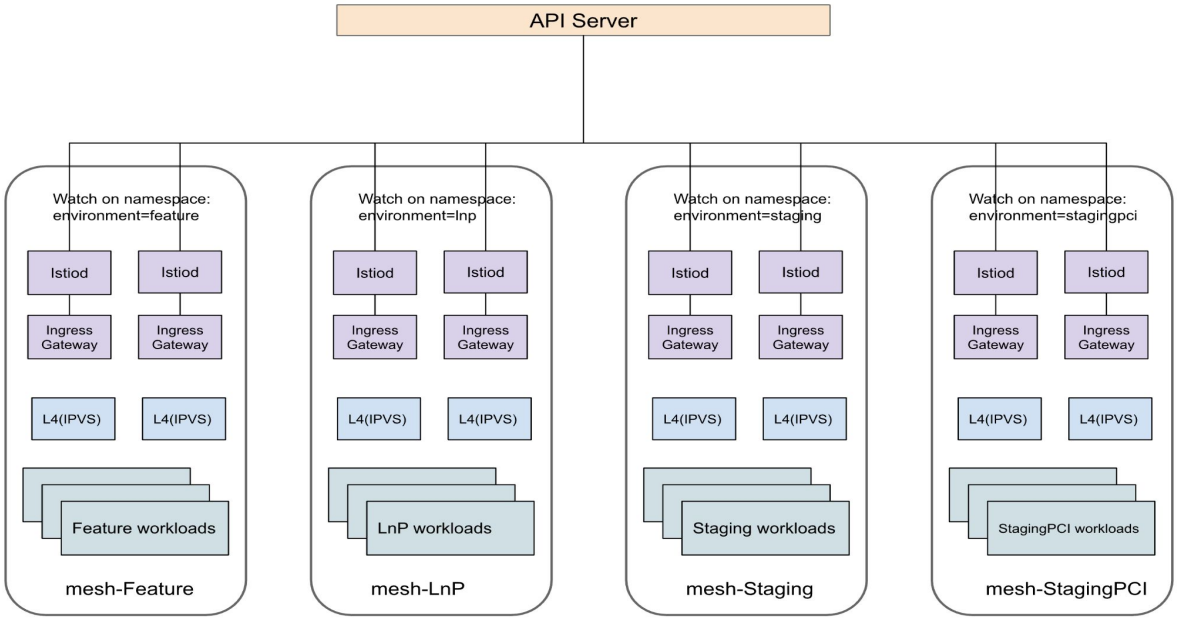
# 基于IPVS和Istio的网络云原生架构

- 基于IPVS的L4 Service控制器：
  - 四层网关调度，VIP地址分配
  - 不同应用配置独立网关VIP
  - 配置IPIP Tunnel模式的IPVS规则
  - 基于BGP VIP子网路由宣告
  - 配置IngressGateway Tunnel接口
  - 支持Direct Server Return (DSR)
- Istio作为应用网关控制器：
  - 管理应用L7规则
  - 自动化生成eBay证书
  - 管理和注入sidecar
  - 网格内部请求mTLS

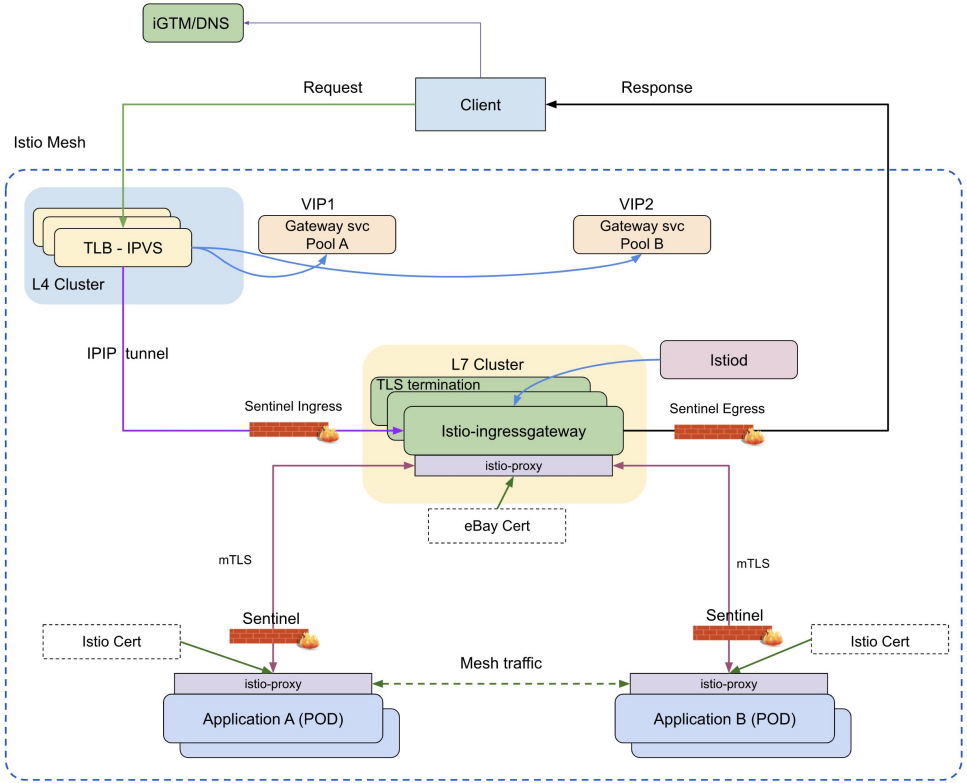


# Istio单集群多环境部署

- 非生产环境
  ：Feature/LnP/Staging/StagingPC
- 生产环境
  ：Pre-Prod/Prod/PCI(Payment Car Industry)
- 单个k8s集群部署多套
  Istiod，IngressGateway和L4集群
- Cheery-pick 社区Pilot scoped namespaces PR
- 不同环境控制面数据面隔离



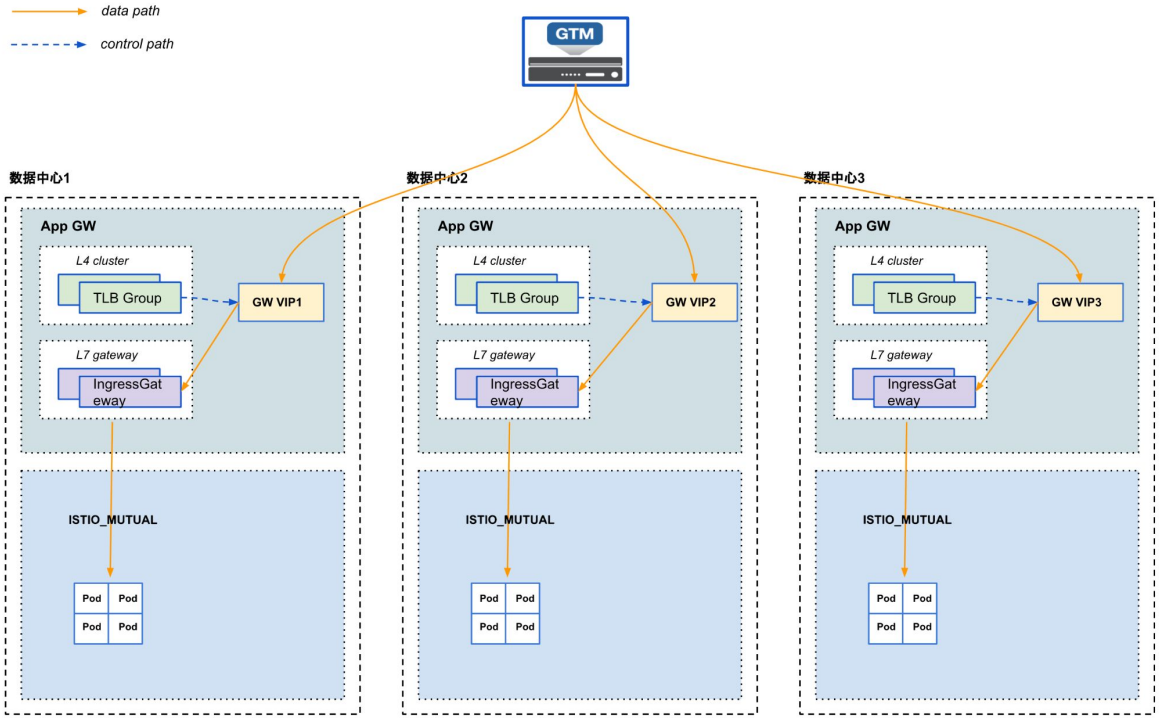
# 单网关全链路加密模式




- 应用场景
  - Feature/LnP/Staging Secure测试环境
  - 全链路加密(E2E TLS)
  - 可用性要求不高
- 同时支持API Gateway和Mesh
  - 应用配置独立Gateway VIP
  - External访问API Gateway为Simple TLS
  - Mesh内部访问为mTLS
  - 不同环境配置专有L4/L7集群
- 软件防火墙集成(Sentinel)
  - 默认阻止所有访问
  - 保护Ingress/Egress流量
  - 保护进入Pod的流量
- 模拟生产环境

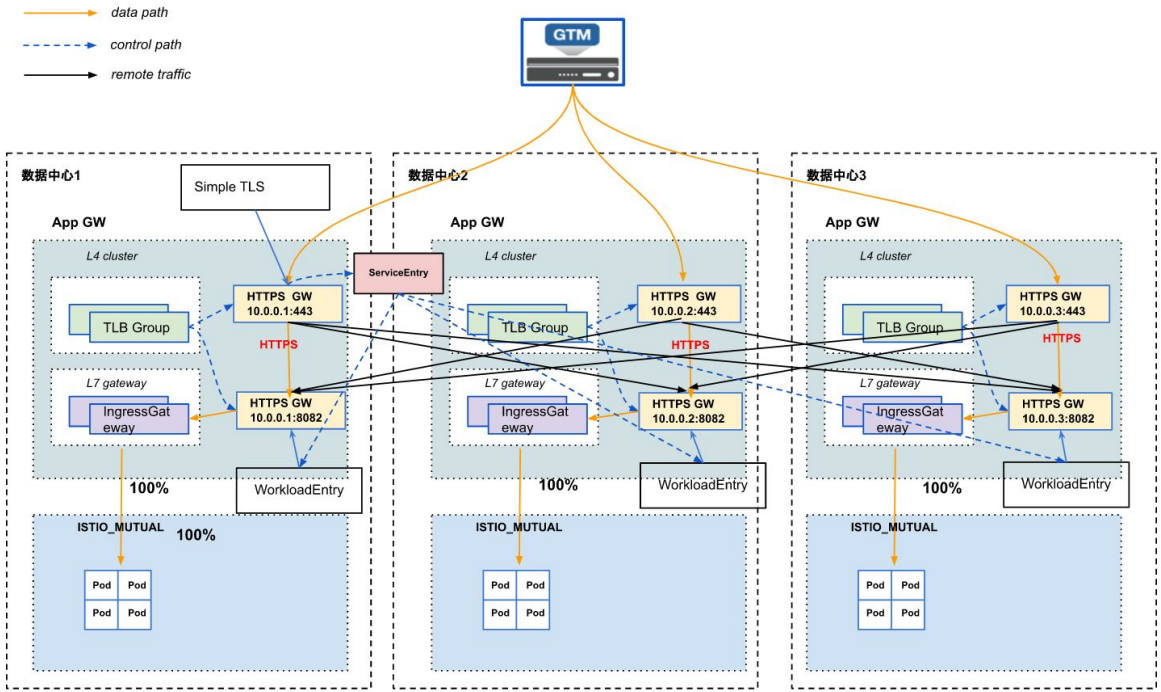
# 应用高可用接入方案-多集群1层

- 流量管理
  - GTM（智能DNS）配置多个VIP，负责健康检查
  - GTM（智能DNS）配置不同数据中心流量权重
  - L4 IPVS 为流量入口
  - 没有跨数据中心流量
  - 应用数据面为网关Envoy到Sidecar Envoy
- 故障容灾
  - GTM监控VIP状态，自动mark down故障VIP
  - Istio 网关宕机会影响客户端DNS缓存
  - 单集群应用后端Pod整体宕机会造成数据面影响



# 应用高可用接入方案-多集群2层

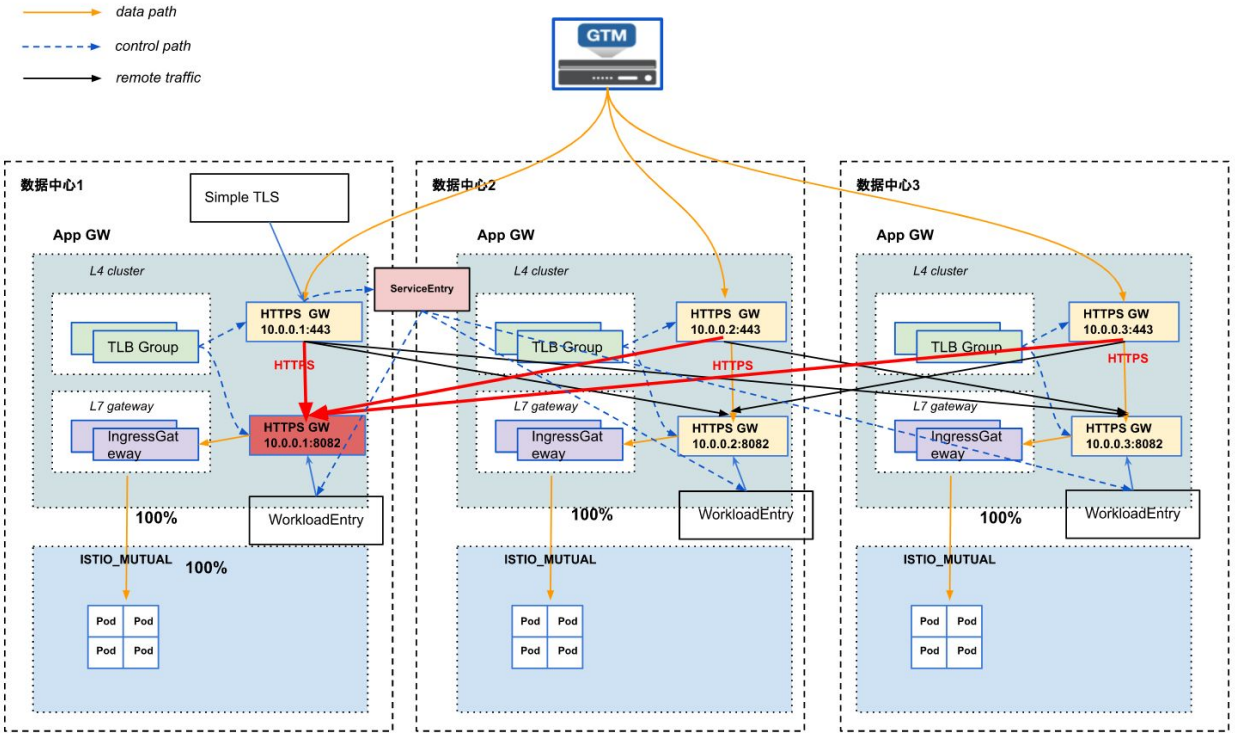
- 流量管理
  - 两层VIP，适配现有模型，精确控制流量
  - ServiceEntry选择WorkloadEntry
  - WorkloadEntry表示远端VIP并且控制weight
  - 本数据中心98%，远端各1%
  - 所有流量都经过Istio
- 故障容灾
  - 单集群应用后端Pod整体宕机不会造成数据面影响
  - Istio 网关宕机会受客户端DNS缓存影响



# 数据中心故障转移

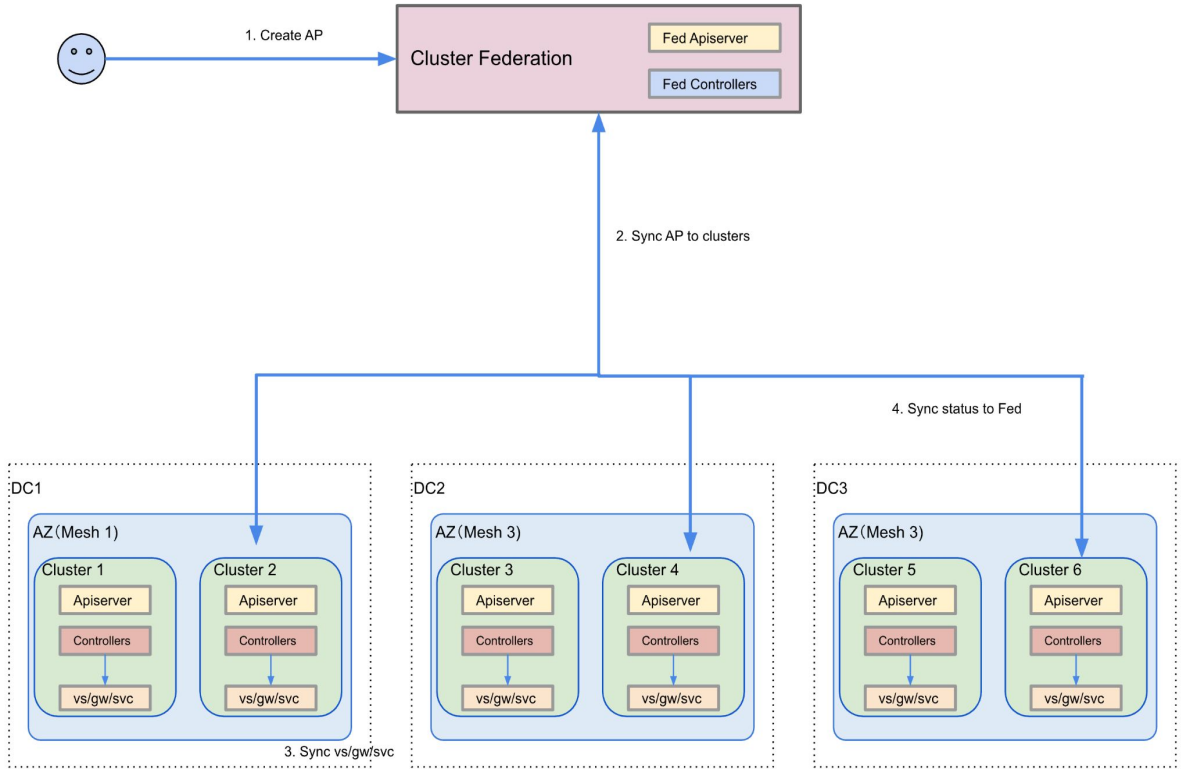
## 将远端WLE健康状态标志为False

```
apiVersion: networking.istio.io/v1beta1
kind: WorkloadEntry
metadata:
  annotations:
    proxy.istio.io/health-checks-enabled: "true"
  creationTimestamp: null
  name: https-foo-we-140
spec:
  address: foo-ns-foo-gateway.example.com
  labels:
    app: foo
  weight: 33
  Locality: region01/az01
status:
  conditions:
    - lastProbeTime: "2022-02-26T22:06:34Z"
      lastTransitionTime: "2022-02-26T22:06:34Z"
      message: 'Region Exit'
      reason: unhealthy
      status: "False"
      type: Healthy
```



# 统一流量模型 - AccessPoint

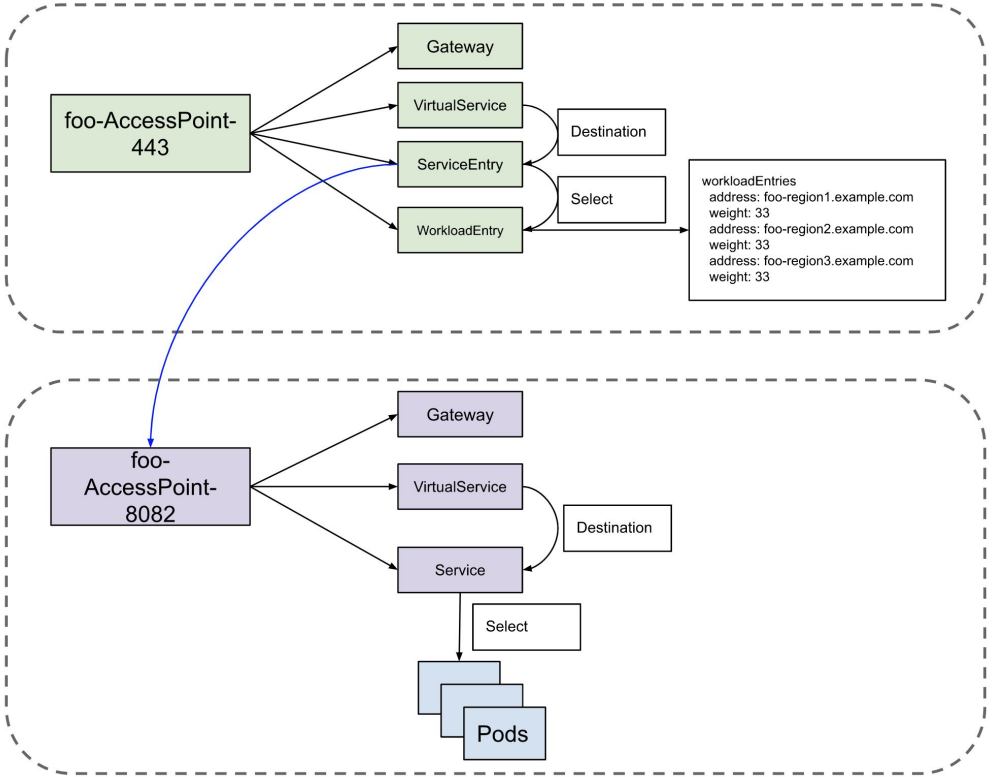
- Spec
  - Scope
    - AZ/ClusterID
  - TrafficTemplate
    - Istio models
    - kubernetes Services
  - Override
    - 可为不同目标集群修改模板属性值
    - 支持多种override方法
      - jsonPatch
      - mergePatch
  - Policy
    - RolloutPolicy
    - RuntimePolicy
- Status
  - Conditions
    - 四层网关状态
    - 证书状态
    - 网关服务IP和FQDN



# 流量管理模型--Service to Service

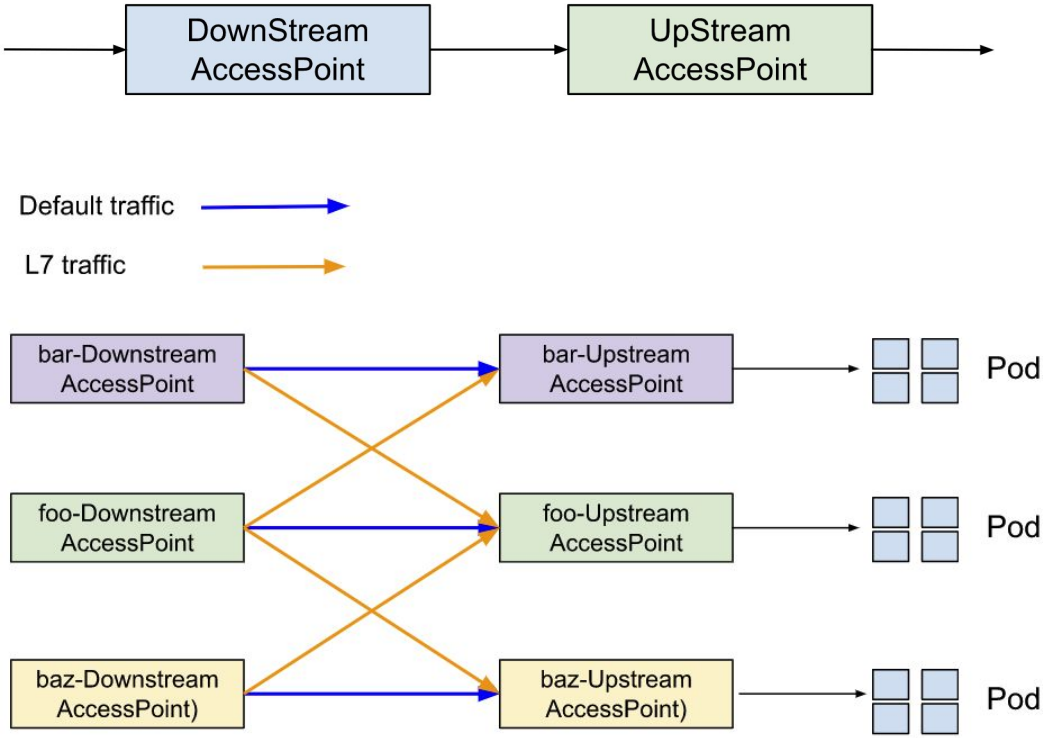
- eBay定义的AccessPoint作为流量抽象资源，管理Istio的CRD
- 使用社区CRD的语义，实现多层网络拓扑架构
- 利用WorkloadEntry表示Gateway VIP
- 利用ServiceEntry选择WorkloadEntry并且作为VirtualService的Destination

| Istio | Kubernetes |
| --- | --- |
| ServiceEntry | Service |
| WorkloadEntry | Pod |



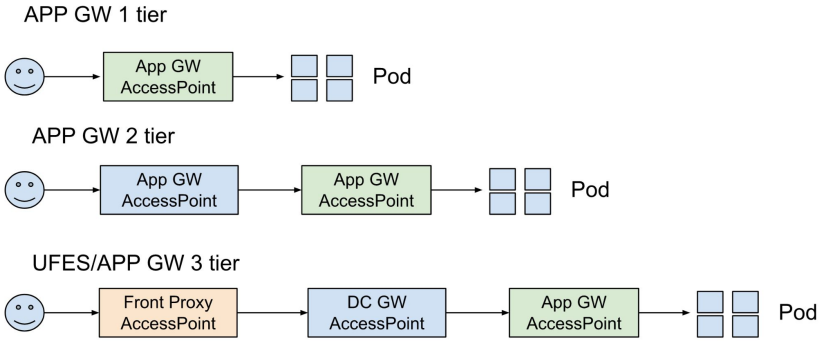
# 流量管理模型--上下游AccessPoint

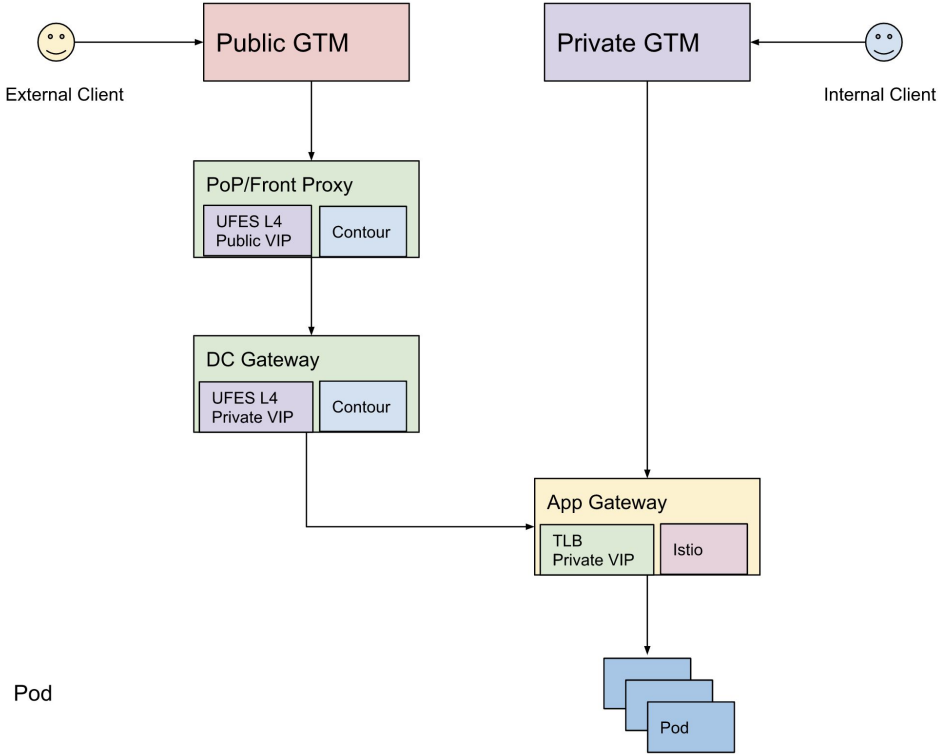
- AccessPoint为一组接入点VIP，作为流量入口
- Downstream AccessPoint将流量接入Upstream AccessPoint，最后再接入Pod
- Downstream AccessPoint通过服务发现Upstream AccessPoint中的VIP并且创建WorkloadEntry
- 为每个Upstream AccessPoint创建一个ServiceEntry并且选择WorkloadEntry，作为DownStream AccessPoint的目的地
- 统一管理应用默认流量以及L7转发流量
- 通过级联的方式实现多层架构



# 统一模型管理公网/内网流量

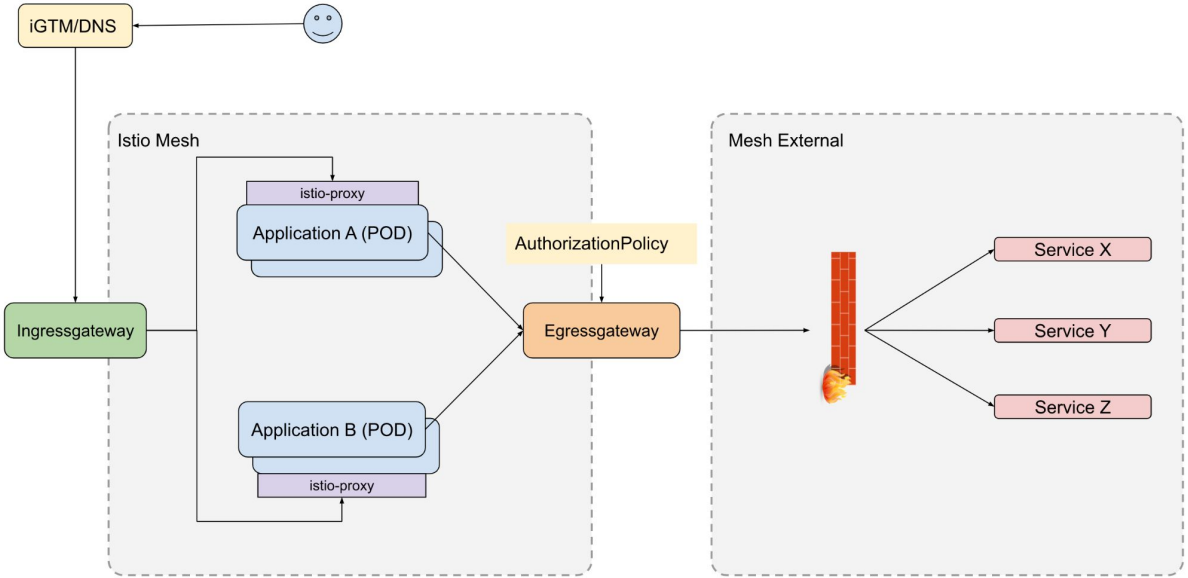
- 公网流量为3层，内网流量为1层
- 统一外网，内网，以及L7流量管理模型
- AccessPoint模型是流量管理的基础构建
- 将多层拓扑拆解成上下游AccessPoint，构建高可用的架构
- 支持全链路加密



# Egress Gateway处理硬件防火墙

- 传统的硬件防火墙基于L4（IP+Port)
- 在云原生的场景下，Pod IP会频繁发生变化，进而会导致硬件防火墙出现性能问题。
- Egress Gateway主要控制网格内部出站流量：
  - 出站流量的防火墙规则转移到Egress Gateway与硬件防火墙
  - 通过AuthorizationPolicy控制网格内部应用到Egress Gatweay的访问权限
  - Egress流量全链路加密



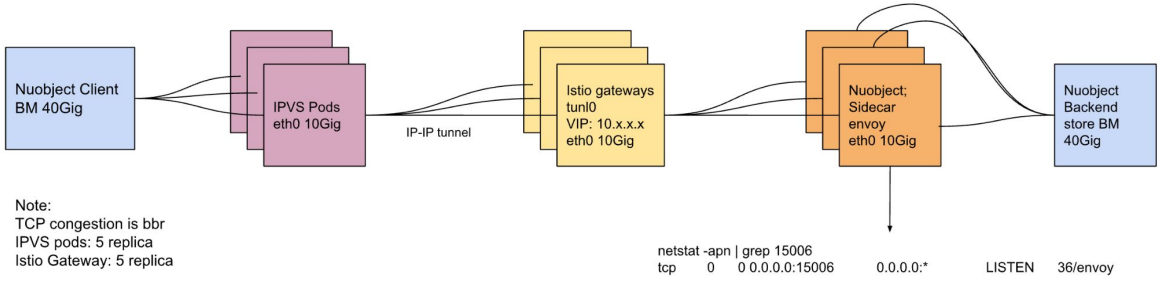
# 全链路加密存储服务-NuObject

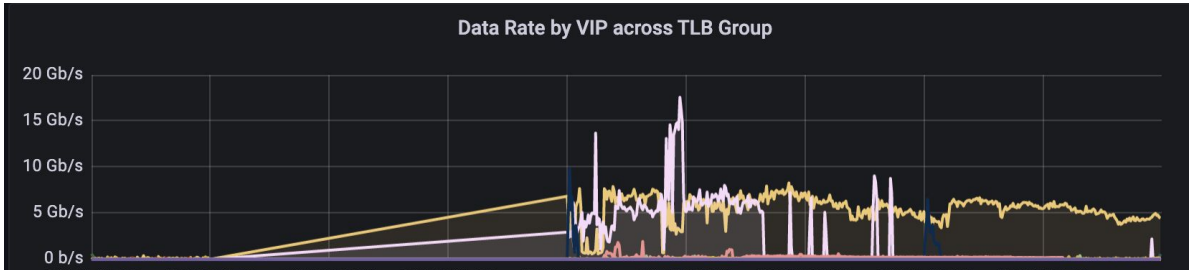
生产环境高吞吐

- 单数据中心单VIP达到18Gb/s
- 跨数据中心单VIP最高达40Gb/s
- 单个IPVS Pod达10Gb/s
- 生产环境IPVS/Gateway Pod各5个
- Ingress Gateway Envoy worker增加到40

高安全性存储服务

- 注入Sidecar到后端服务
- 网关配置Simple TLS
- 网格内部mTLS
- 集成软件防火墙

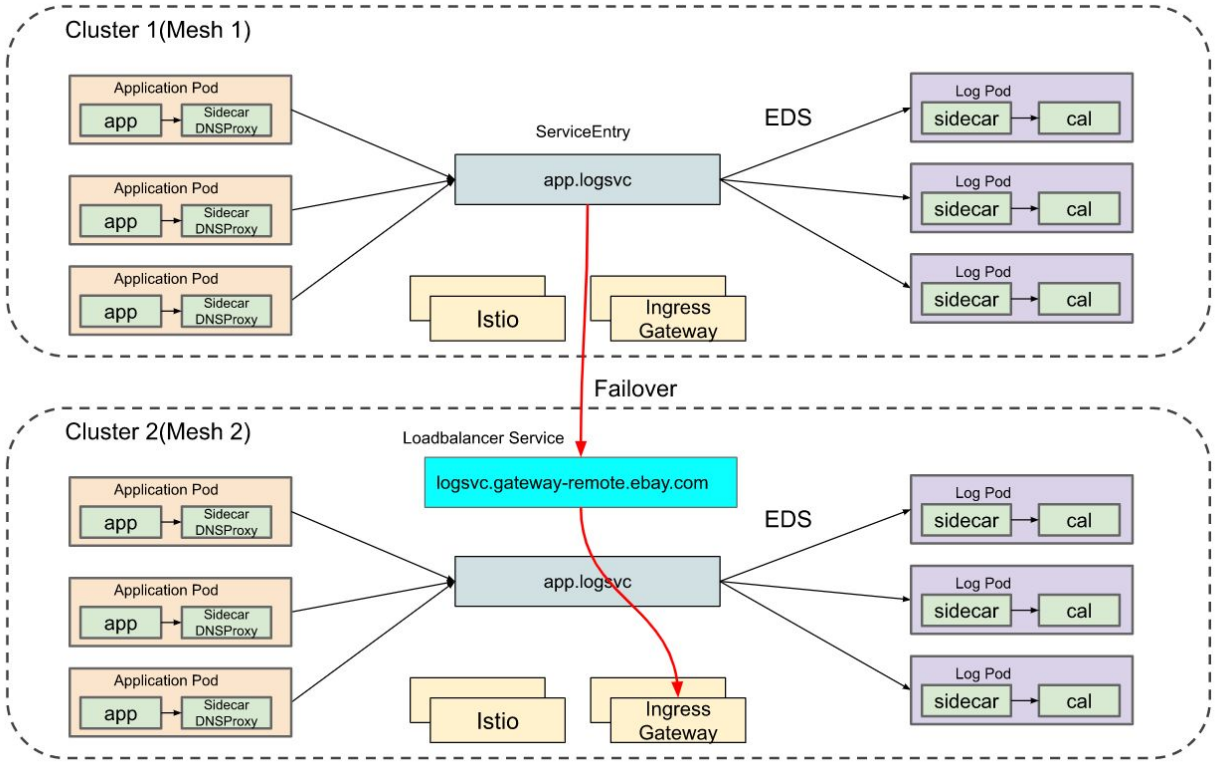


# CAL日志系统集成Mesh

- 不同网格同时部署应用以及日志服务器
- 同时注入sidecar到应用端以及日志系统服务端
- 网格内部mTLS东西流量
- 网格内部故障通过ServiceEntry选择WorkloadEntry切换成南北流量

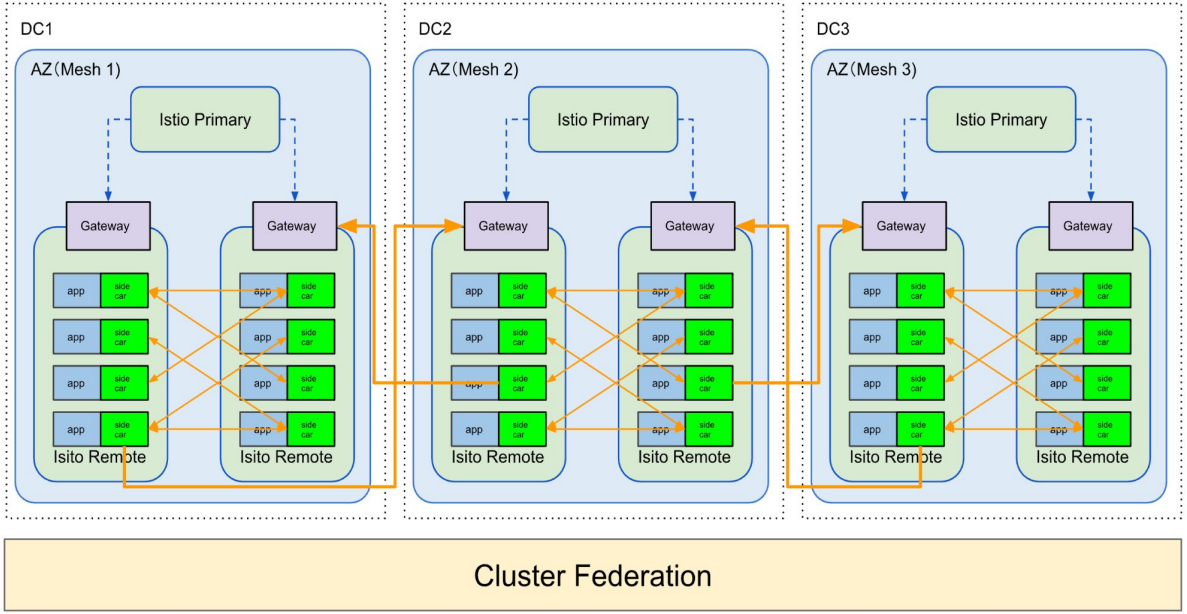
```
apiVersion: networking.istio.io/v1beta1
kind: ServiceEntry
metadata:
  name: log
spec:
  hosts:
  - log.example.com
  location: MESH_INTERNAL
  ports:
  - name: tls-1234
    number: 1234
    protocol: TLS
  resolution: DNS
  workloadSelector:
    labels:
      app: log-pods //Remote VIP WLE for failover
```



# Managed Stack全面集成Mesh

Managed Stack是使用eBay框架的应用，包括Java, Java Web, Batch以及NodeJS应用

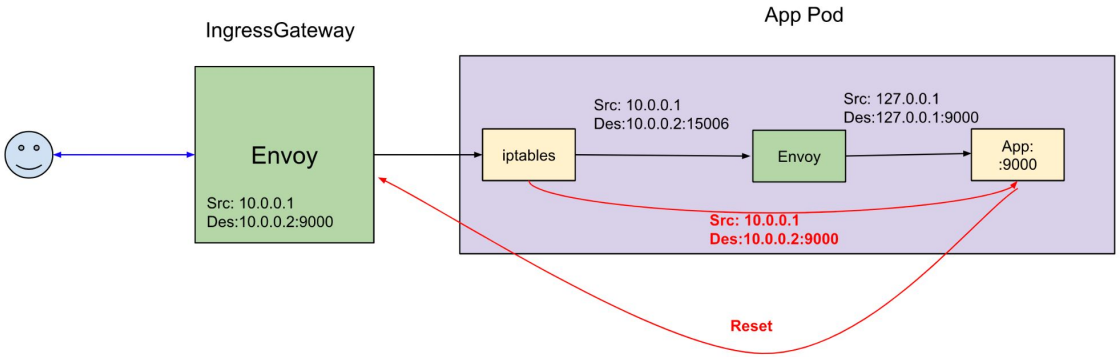
- 生产环境应用数量超过4000
- 个别应用部署后端服 务器超过3000
- 生产环境Pod总数超180000
- TLS/限速/授权与框架解耦
- Fat container全面转向Native以及Mesh



# Sidecar Envoy间歇性连接重置

原因:Sidecar的iptables没有给out of window的packet做DNAT

1. 让 conntrack 对数据包校验更加宽松，不要将out of window的数据包标记为INVALID。对linux来说只需要修改内核参数 echo 1 > /proc/sys/net/ipv4/netfilter/ip_conntrack_tcp_be_liberal
2. 专门添加一条 iptables 规则来丢弃标记为 INVALID 的数据包，这样它就不会到达客户端 Pod。




iptables -t mangle -A PREROUTING -m conntrack --ctstate INVALID -j DROP

Istio 1.13 Release

- **Fixed** an issue that sidecar iptables will cause intermittent connection reset due to the out of window packet. Introduced a flag `meshConfig.defaultConfig.proxyMetadata.INVALID_DROP` to control this setting. (Issue #36566)

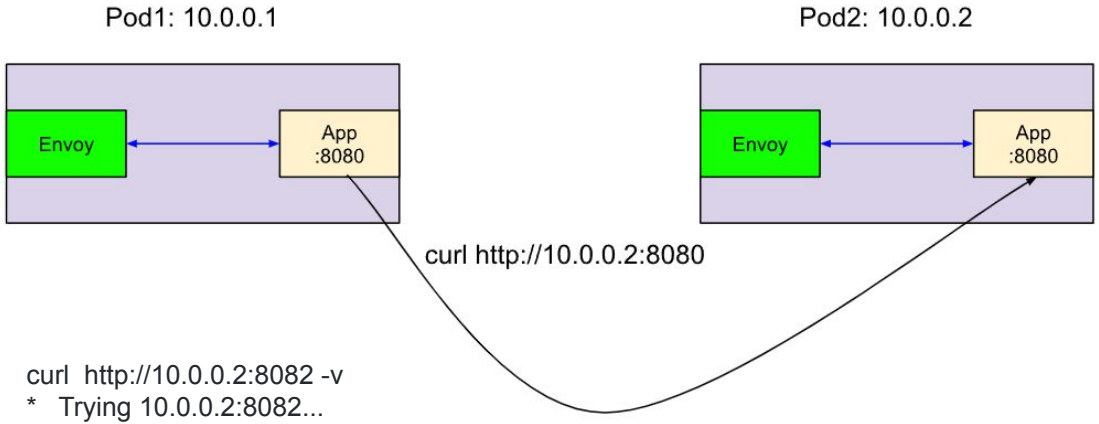
# STRICT mTLS Pod访问Pod IP

原因：直接访问Pod IP会被Envoy PassThrough cluster处理，目前还不支持mTLS.

短期解决方案：

1. 给每个需要被单独访问的Pod创建一个service
2. 通过访问service名字实现mTLS

社区正在解决这个问题：
https://github.com/istio/istio/issues/37431




```
curl  http://10.0.0.2:8082 -v
*   Trying 10.0.0.2:8082...
* TCP_NODELAY set
* Connected to 10.0.0.2 (10.0.0.2) port 8082 (#0)
> GET / HTTP/1.1
> Host: 10.0.0.2:8082
> User-Agent: curl/7.68.0
> Accept: */*
>
* Empty reply from server
* Connection #0 to host 10.0.0.2 left intact
    curl: (52) Empty reply from server
```

# Istio应用中遇到的挑战

1. 共享Wildcard listener带来的问题
   - 无法区分downstream envoy_listener_ssl_connection_error
   - 同一网关端口不能同时支持TCP/HTTPS
   - 解决方案：设置Gateway Bind属性，拆分listener
2. 单点从外部访问mTLS Mesh机器
   - 解决方案：利用Subset Load Balancing，EnvoyFilter从URL解析Pod IP并转发
3. AuthorizationPolicy性能问题
   - AuthorizationPolicy会被全量推送到listener rbac fitler
   - 解决方案：通过EnvoyFilter去除掉和Gateway无关的config
4. Egress Gateway E2E TLS
   - 应用outbound请求为HTTPs，经过Egress Gateway会导致double TLS.
   - 解决方案：Sidecar Egress Listener支持TLS termination
5. 控制面性能问题
   - 解决方案：sharding，将mesh切片，限制单个mesh Gateway/Service/Pod数量

# 未来展望

- 全面替换硬件负载均衡设备
- 南北流量接入软件应用网关
- 构建基于Mesh流量管理并实现端到端加密
- 全站应用全面转向Cloud Native/Service Mesh
- 数据平面加速Cilium

# Thanks