Deep Dive into the Network Traffic Path of the Coexistence of Ambient and Sidecar

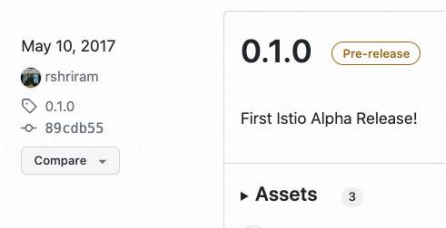

Huailong Zhang

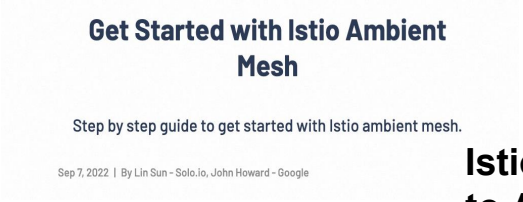

Yuxing Zeng

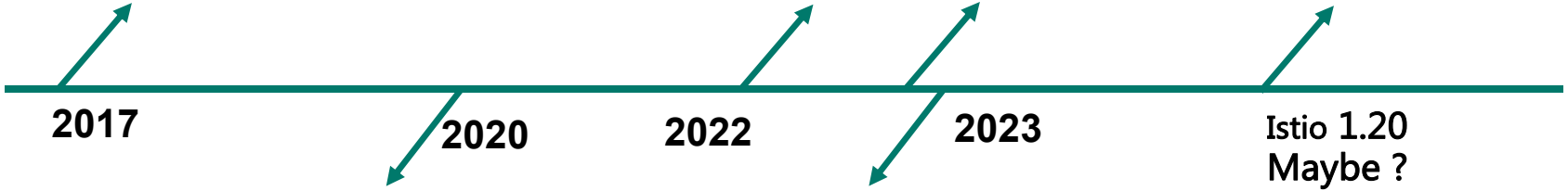
IstioCon

目录

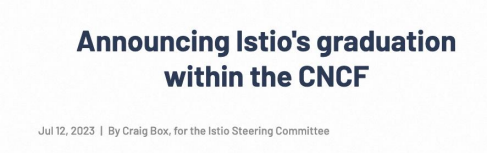
Istio Ambient 发展历程







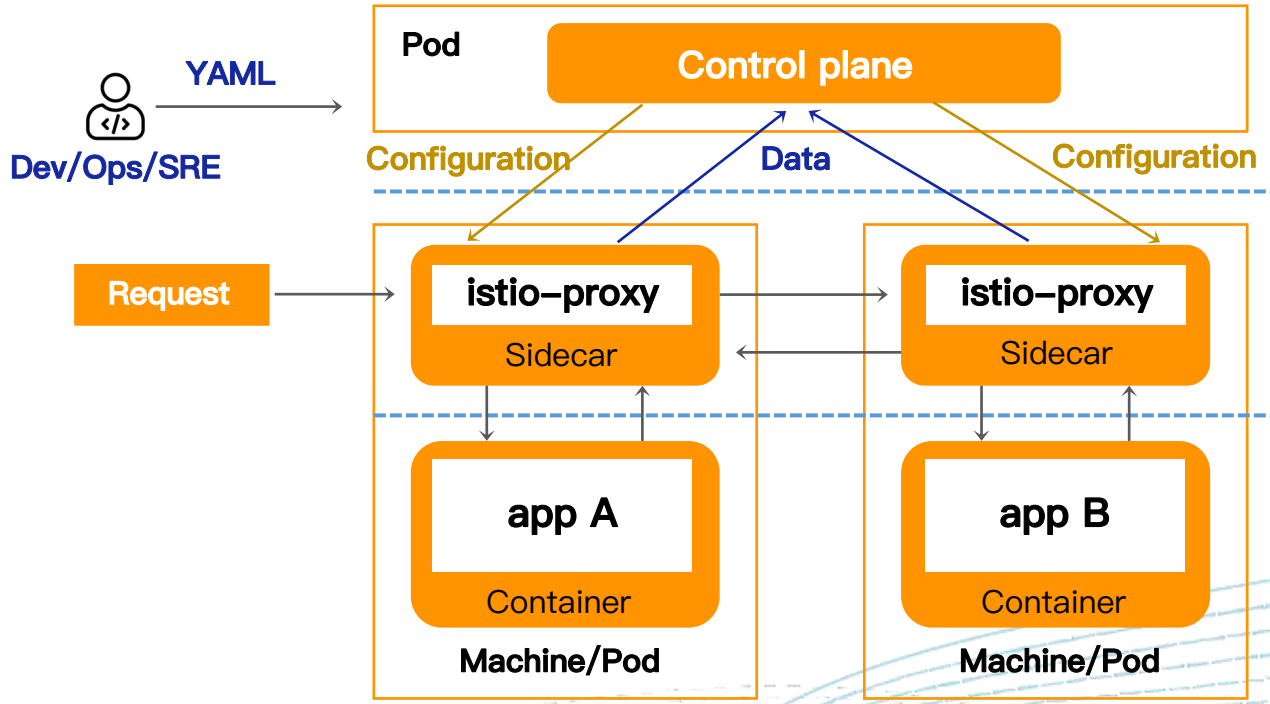
Istio 1.5,
Simplified
Istio (istiod)



Istio 1.18, Ambient to Alpha

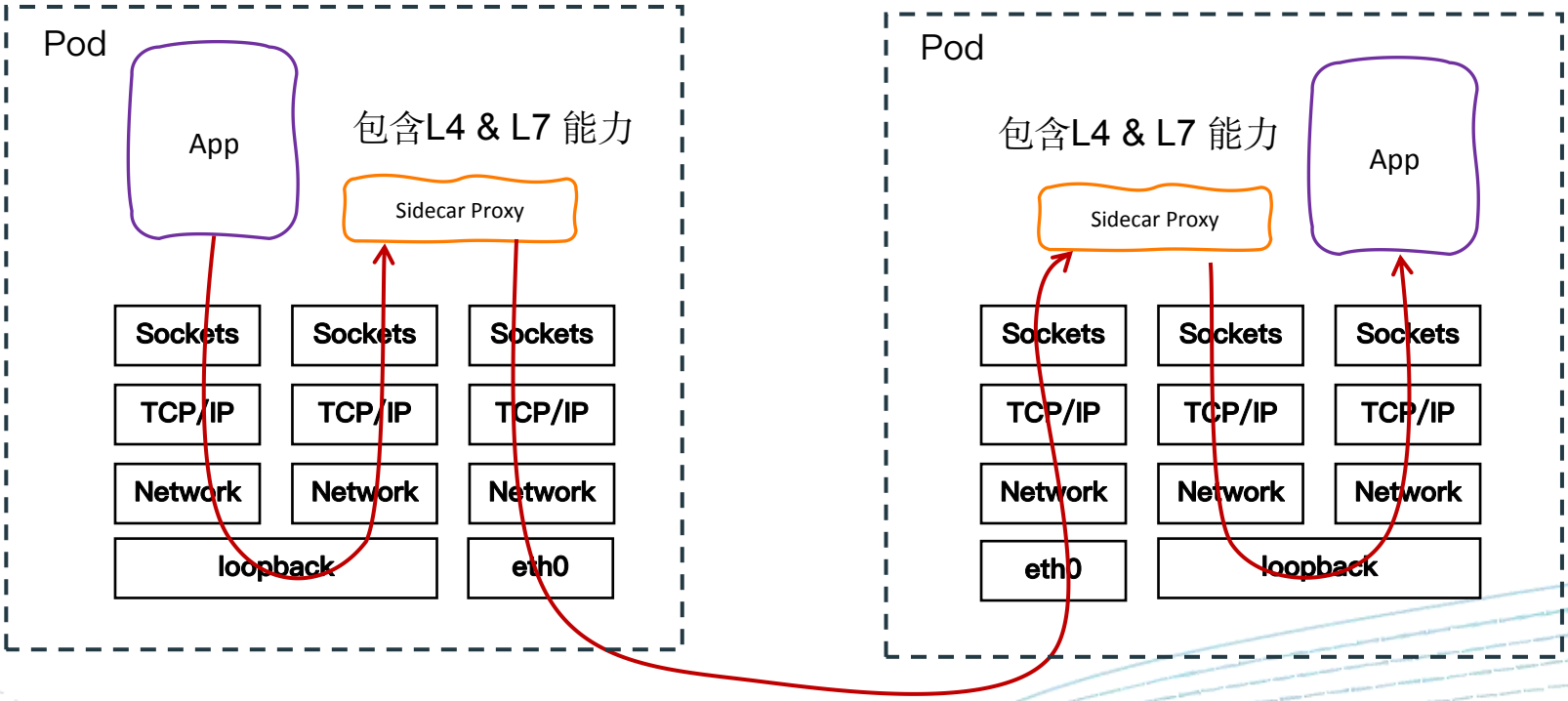
Ambient to Beta?

Istio 数据面的演变过程 Sidecar -> Ambient



- 流量管理
- 安全
- 可观测性

Sidecar 代理下的网络数据包的传输过程



Istio Ambient- 一种新的数据平面模式

设计理念：将数据平面分层， 以此允许用户以更渐进增量的方式采用服务网格技术

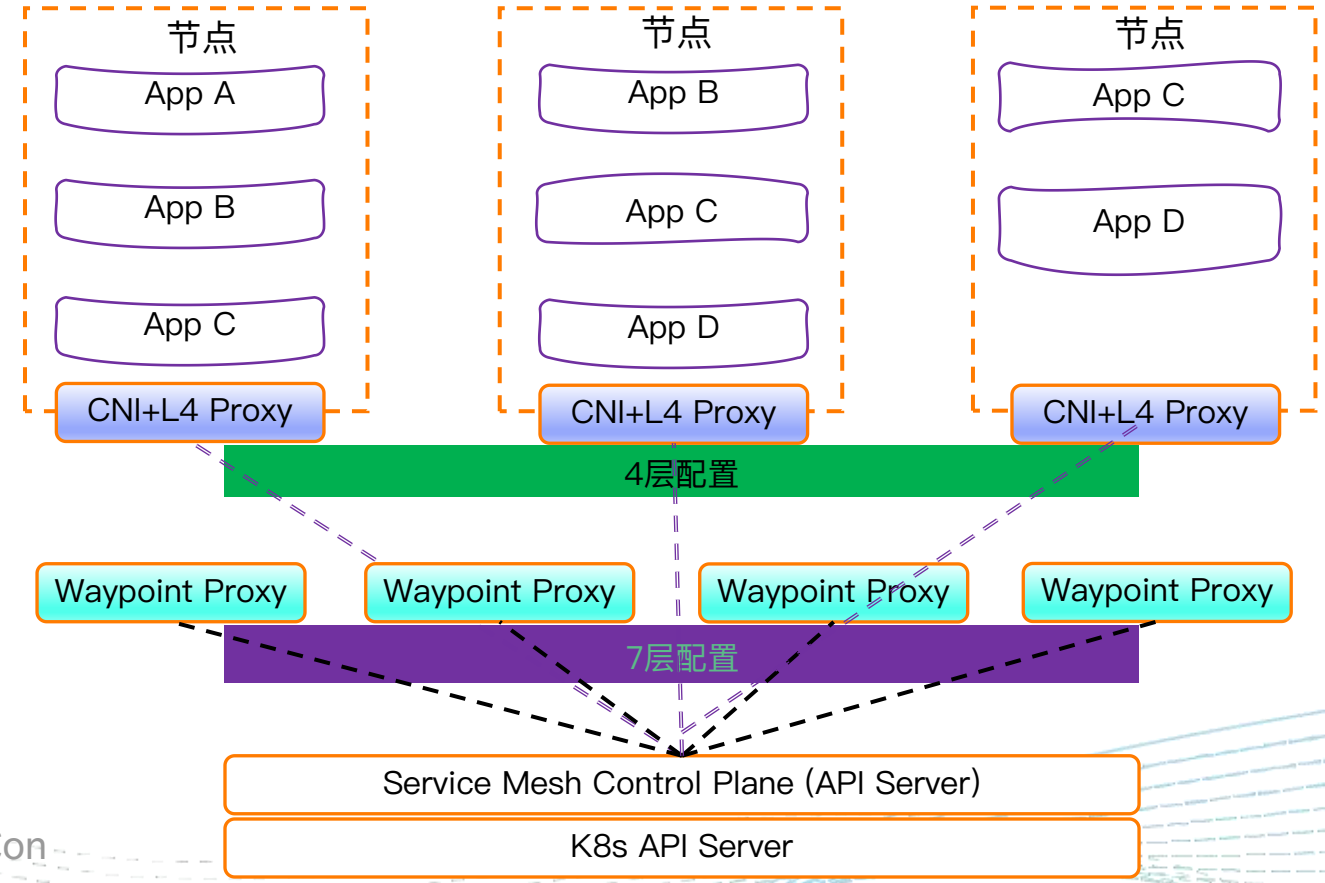
7层高级处理：
功能丰富

- 流量管理：HTTP路由、负载均衡、熔断、限流、故障容错、重试、超时等
- 安全：面向7层的精细化授权策略
- 可观测：HTTP监控指标、访问日志、链路追踪

4层基础处理：
低资源
高效率

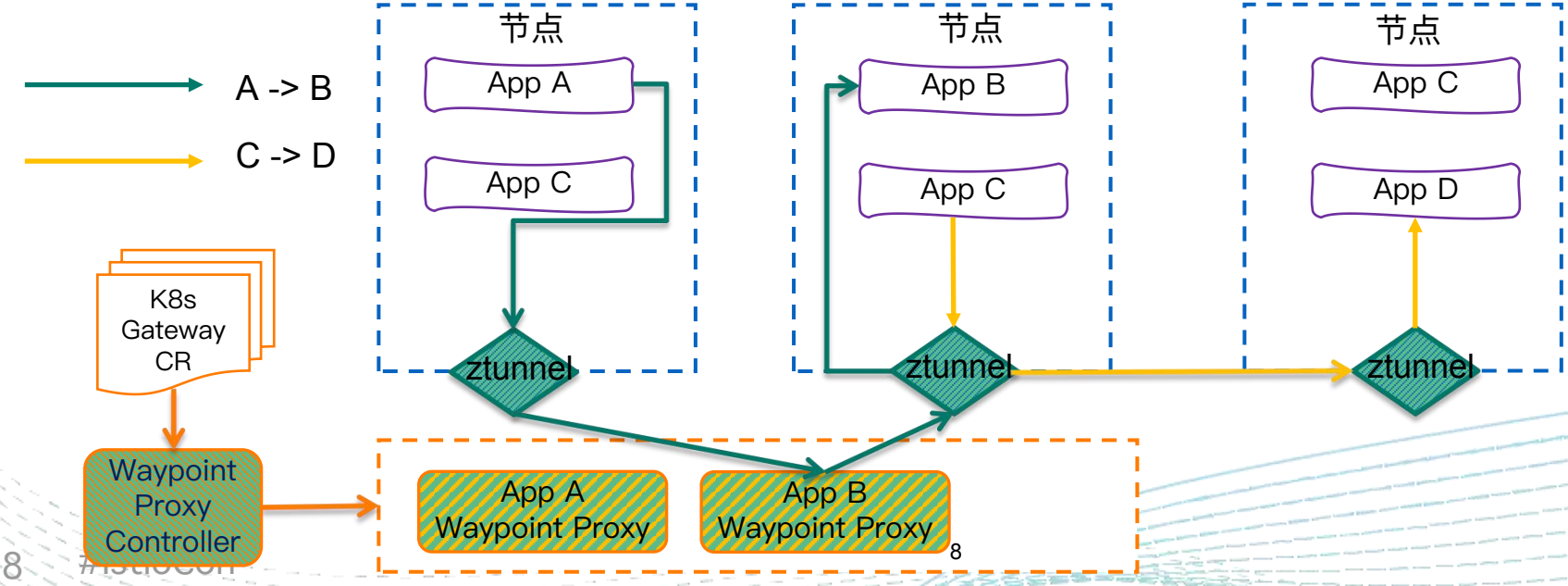
- 流量管理：TCP路由
- 安全：面向4层的简单授权策略、双向TLS
- 可观测：TCP监控指标及日志

Ambient数据面: L4 与L7 代理解耦

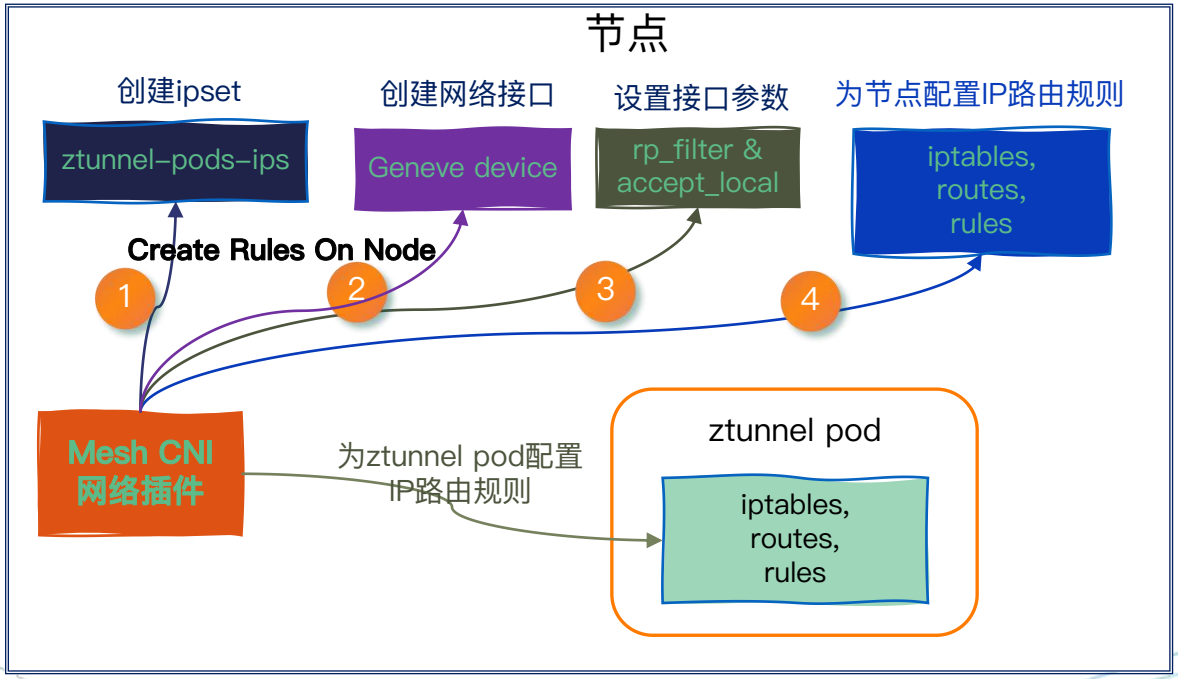


Istio 中业务应用程序和数据平面代理分离

Waypoint代理	• L7 组件完全独立于应用程序运行， 安全性更高；每个身份（Kubernetes 中的服务帐户）都有自己专用的 L7 代理，避免多租户 L7 代理模式引入的复杂度与不稳定性； • 通过K8s Gateway CRD定义触发启用；
ztunnel	将 L4处理下沉到 CNI级别，来自工作负载的流量被重定向到 ztunnel， 然后ztunnel识别工作负载并选择正确的证书来处理；
与Sidecar模式兼容	Sidecar模式仍然是网格的一等公民，可以与部署了 Sidecar 的工作负载进行本地通信；

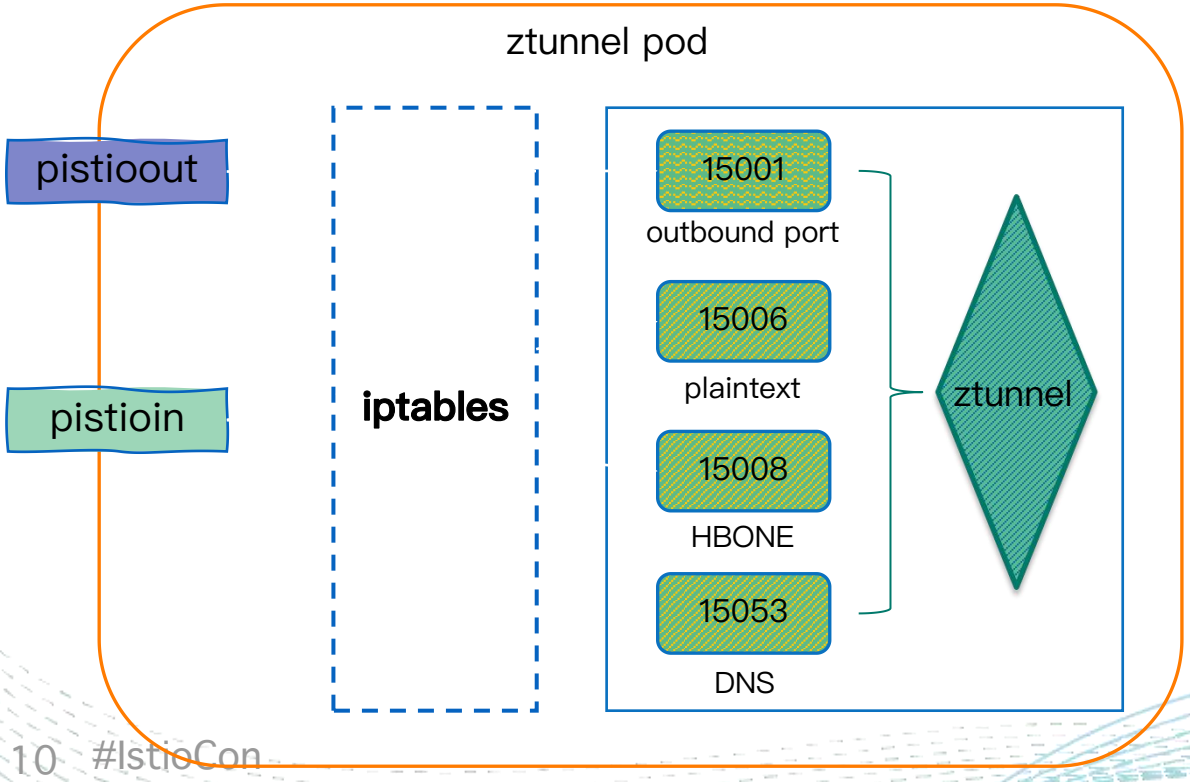


Ambient 的实现： 流量拦截之CNI 插件



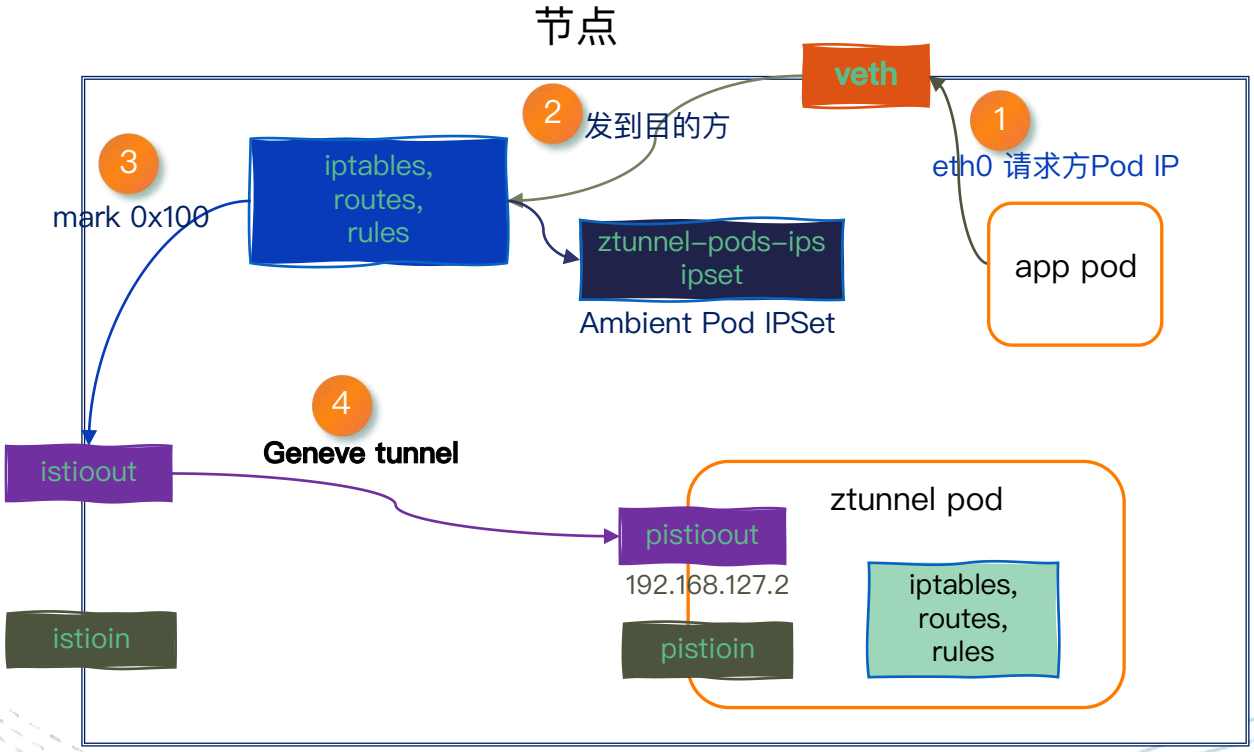
- 节点：在每个节点上配置网络命名空间，以将进出节点的流量透明地路由到节点，并相应地将其路由到ztunnel代理。
- ztunnel：在ztunnel pod的网络命名空间中配置路由，将进出流量路由到ztunnel代理上的特定端口。
- CNI插件在每个节点上初始化路由并设置iptables和ipset。
- 注意： CNI插件所做的配置不会直接影响任何工作负载pod。更改仅在节点网络命名空间和ztunnel pod的网络命名空间中进行，与流量重定向机制无关。

CNI 插件为ztunnel 配置iptables



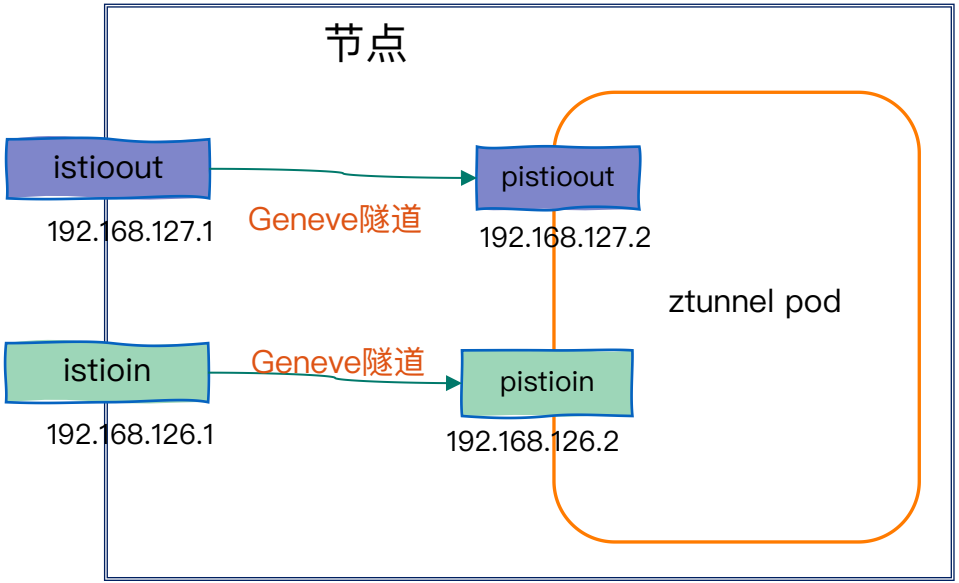
- 在ztunnel pod上，pistioin接口接收到的任何内容都会被转发到端口15008（HBONE）和15006（纯文本）。
- 同样，pistioout接口接收到的数据包最终会到达端口15001。
- ztunnel还捕获端口15053上的DNS请求，以提高网格的性能和可用性。

Ambient 网格内部POD 的流量路径



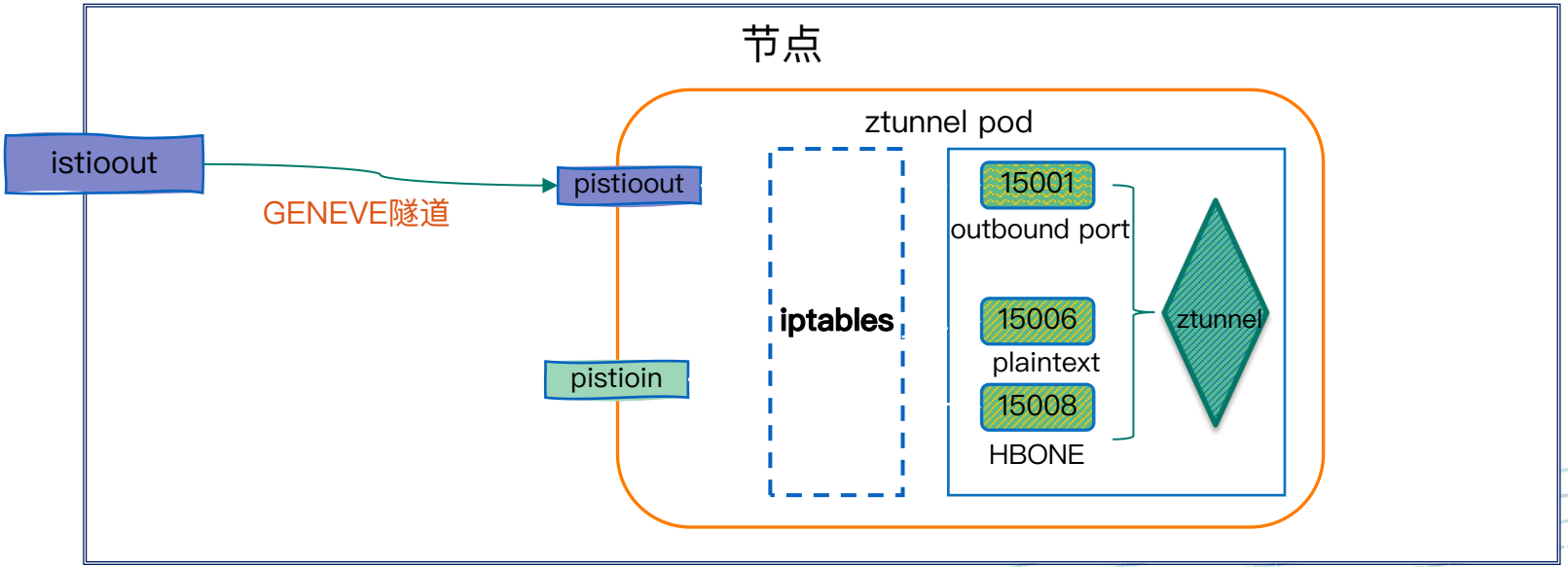
- 1. Ambient模式下的应用Pod会被CNI插件将其IP地址写入到ipset中，当发起请求时，流量数据包进入到该节点上对应的veth接口。
- 2.数据包来自Ambient Pod，会被iptables拦截。
- 3.使用0x100/0x100标记数据包，并进入到istioout网络接口。
- 4.通过istioout接口将流量透明劫持到pistioout网络接口，其中pistioout用于接收 Geneve 隧道中的发来的数据包 。

使用Geneve 隧道连接

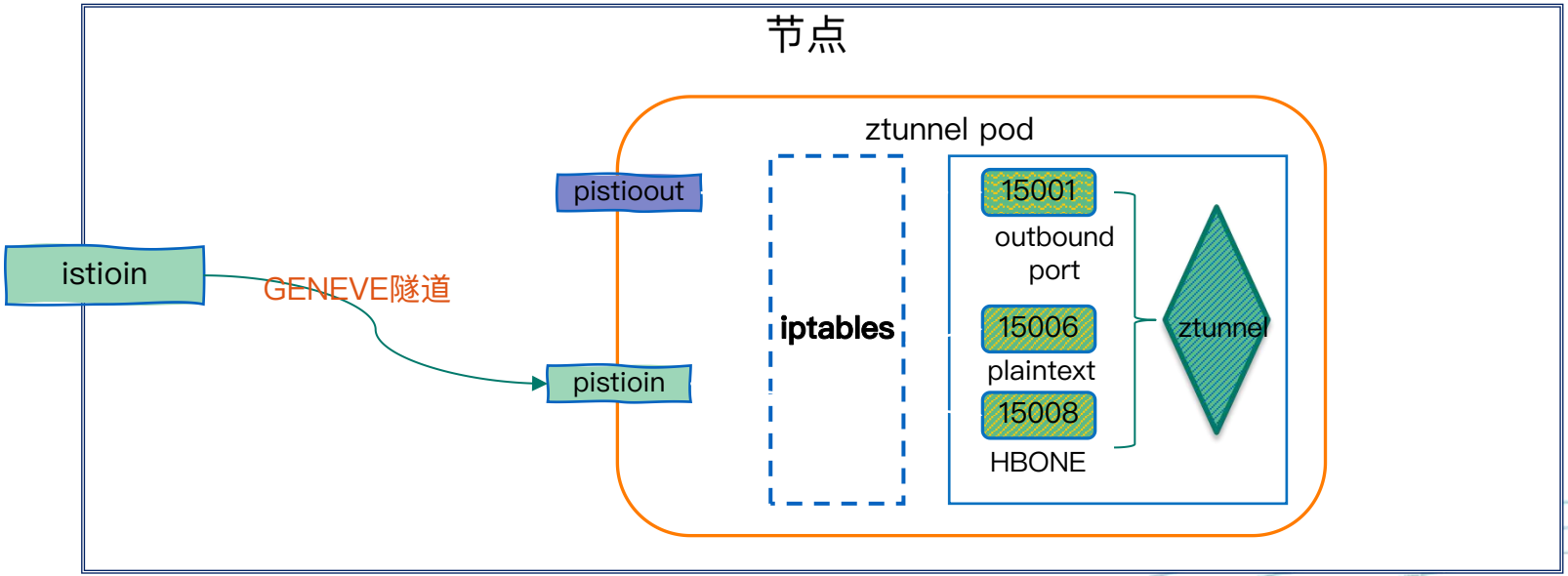


- CNI插件在每个节点上初始化路由并设置iptables和ipset规则。在每个节点上，设置了两个虚拟接口 – istioin和istioout，用于处理节点上的入站 (istioin) 和出站 (istioout) 流量。
- 这两个接口使用GENEVE（通用网络虚拟化封装）隧道连接到在同一节点上运行的ztunnel pod的接口上。
- 结合节点上的iptables规则和路由表，确保来自ambient pods的流量被拦截，并根据方向（入站或出站）分别发送到istioout或istioin。发送到这些接口的数据包最终会到达在同一节点上运行的ztunnel pod的pistioout或pistioin。

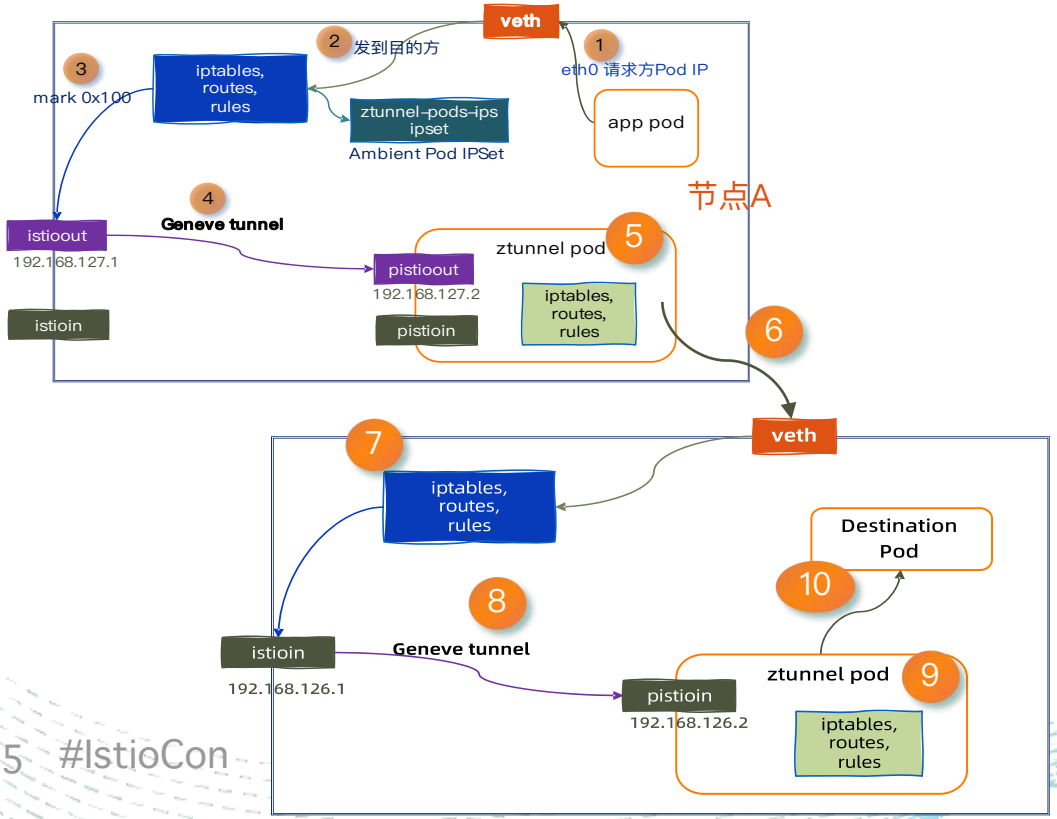
使用Geneve 隧道从节点istioout 到ztunnel 的pistioout



使用Geneve 隧道从节点istioin 到ztunnel的pistioin



Ambient L4 请求处理下的端到端流量路径

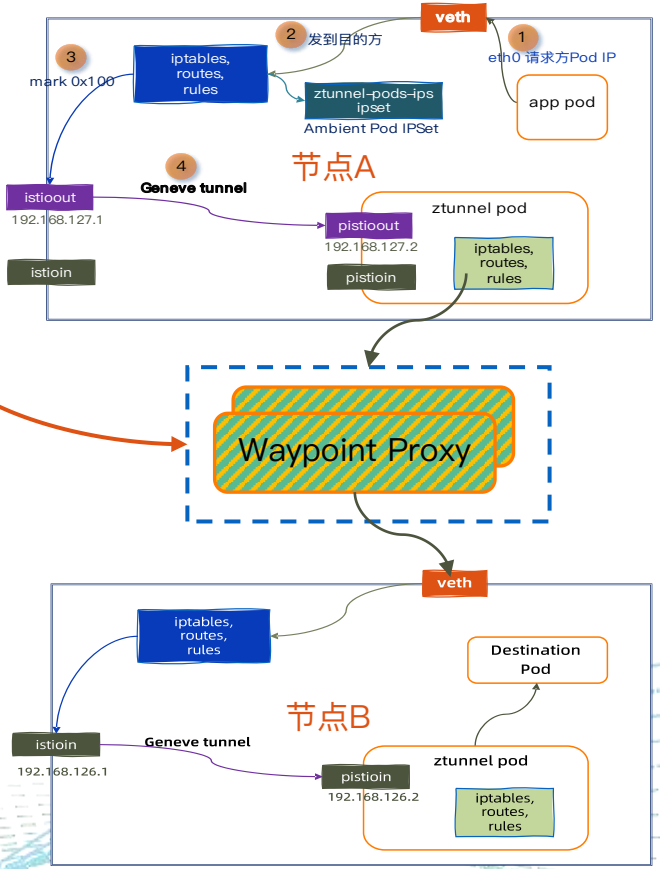


1. Ambient模式下的应用Pod会被CNI插件将其IP地址写入到ipset中，当发起请求时，流量数据包进入到该节点上对应的veth接口。
2. sleep app pod 的请求被节点上的规则和 iptables 配置捕获。
3. 因为该 pod 是环境网格的一部分，它的 IP 地址被添加到节点上的 IP 集合中，并且数据包被标记为 0x100。
4. 节点上的规则指定任何标记为 0x100 的数据包都要通过 istio 出口接口定向到目标 192.168.127.2。
5. ztunnel 代理上的规则透明地代理来自 pistioout 的数据包到 ztunnel 出站端口 15001。
6. ztunnel 处理数据包并将其发送到目标服务（httpbin）的 IP 地址。该地址在节点 B 上为 httpbin创建专用接口veth，请求在该接口上被捕获。
7. 入站流量的规则确保数据包被路由到 istioin 接口。
8. istioin 和 pistioin 之间的隧道使数据包落在 ztunnel pod 上。
9. iptables 配置捕获来自 pistioin 的数据包，并根据标记将它们定向到端口 15008。
10. ztunnel pod处理数据包并将其发送到目标 pod。

L4 到L7 的流量路径-ztunnel 转发到Waypoint

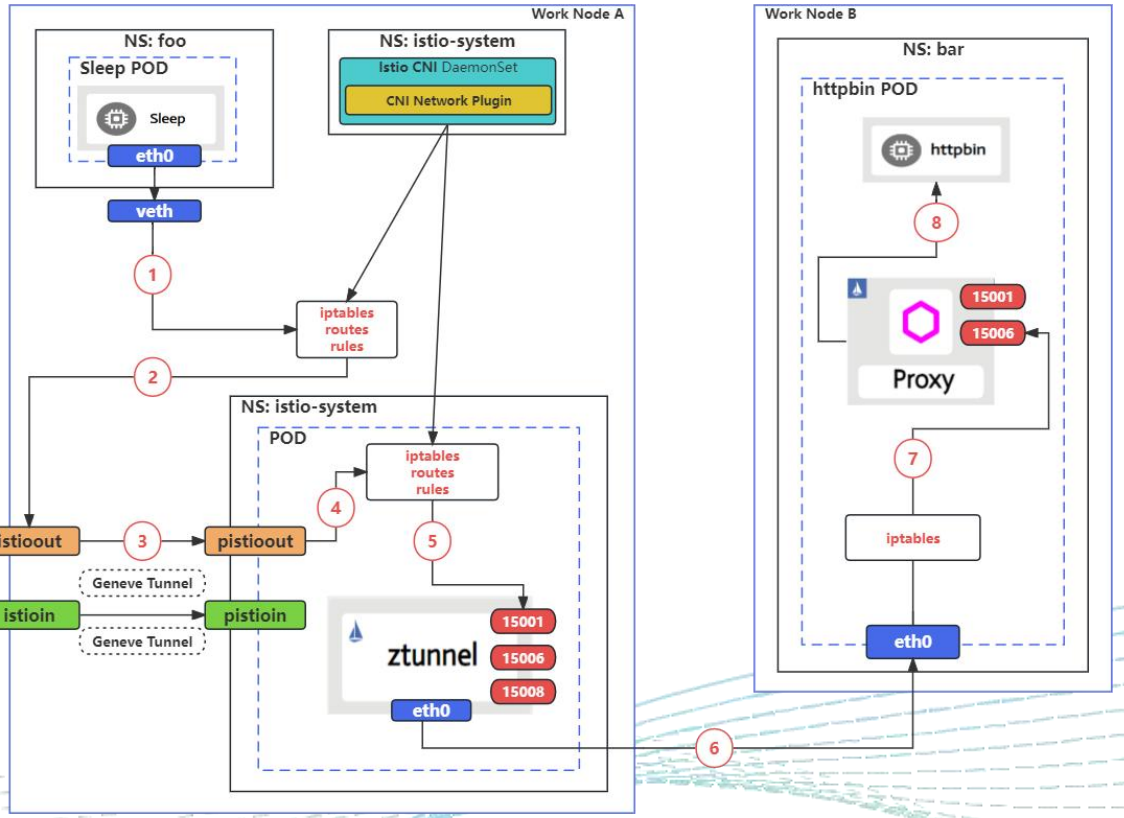
ztunnel中的配置


```
"10.0.0.211": {
    "workloadIp": "10.0.0.211",
    "waypointAddresses": [
        "10.0.0.221"
    ],
    "gatewayAddress": null,
    "protocol": "HBONE",
    "name": "productpage-v1-7b4dbf9c75-pktj7",
    "namespace": "default",
    "trustDomain": "cluster.local",
    "serviceAccount": "bookinfo-productpage",
    "workloadName": "productpage-v1",
    "workloadType": "deployment",
    "canonicalName": "productpage",
    "canonicalRevision": "v1",
    "node": "cn-beijing.10.0.0.210",
    "nativeHbone": false,
    "authorizationPolicies": [],
    "status": "Healthy",
    "clusterId": "Kubernetes"
},
```

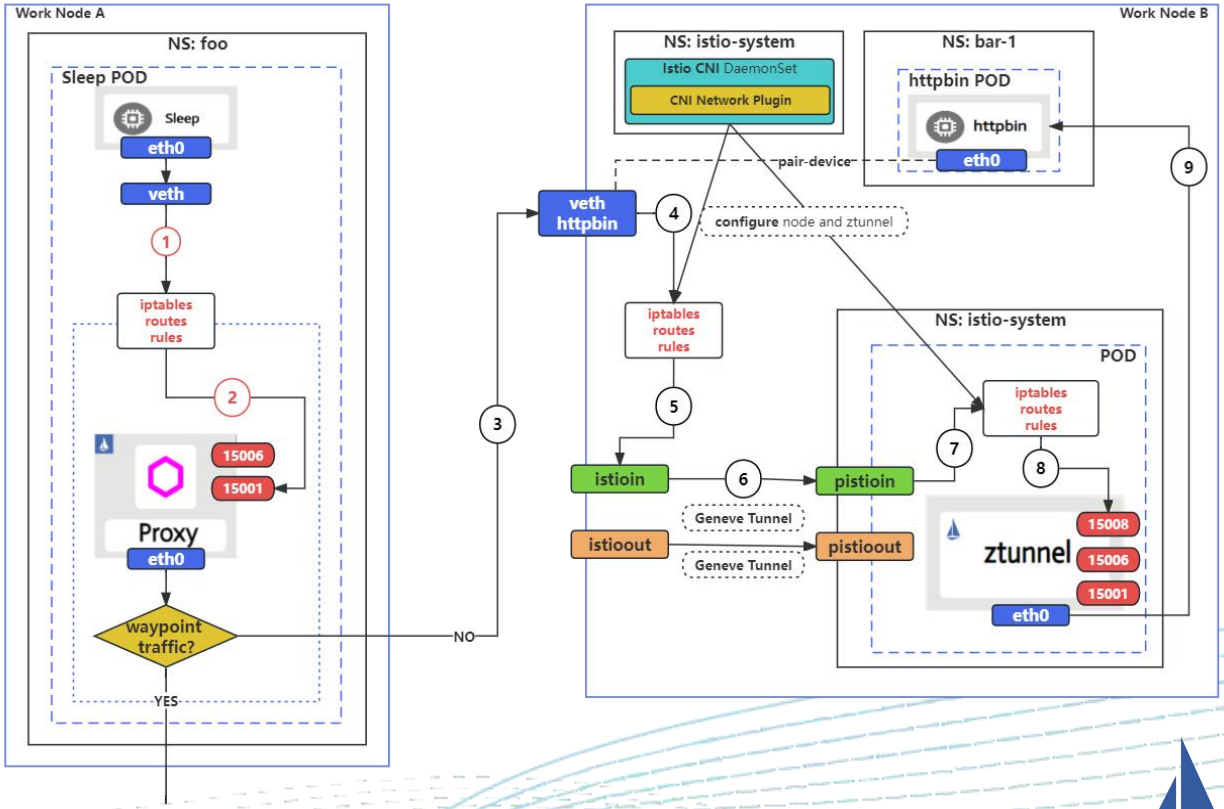
Sidecar 和 Ambient 共存下的流量路径分析

Ambient -> Sidecar



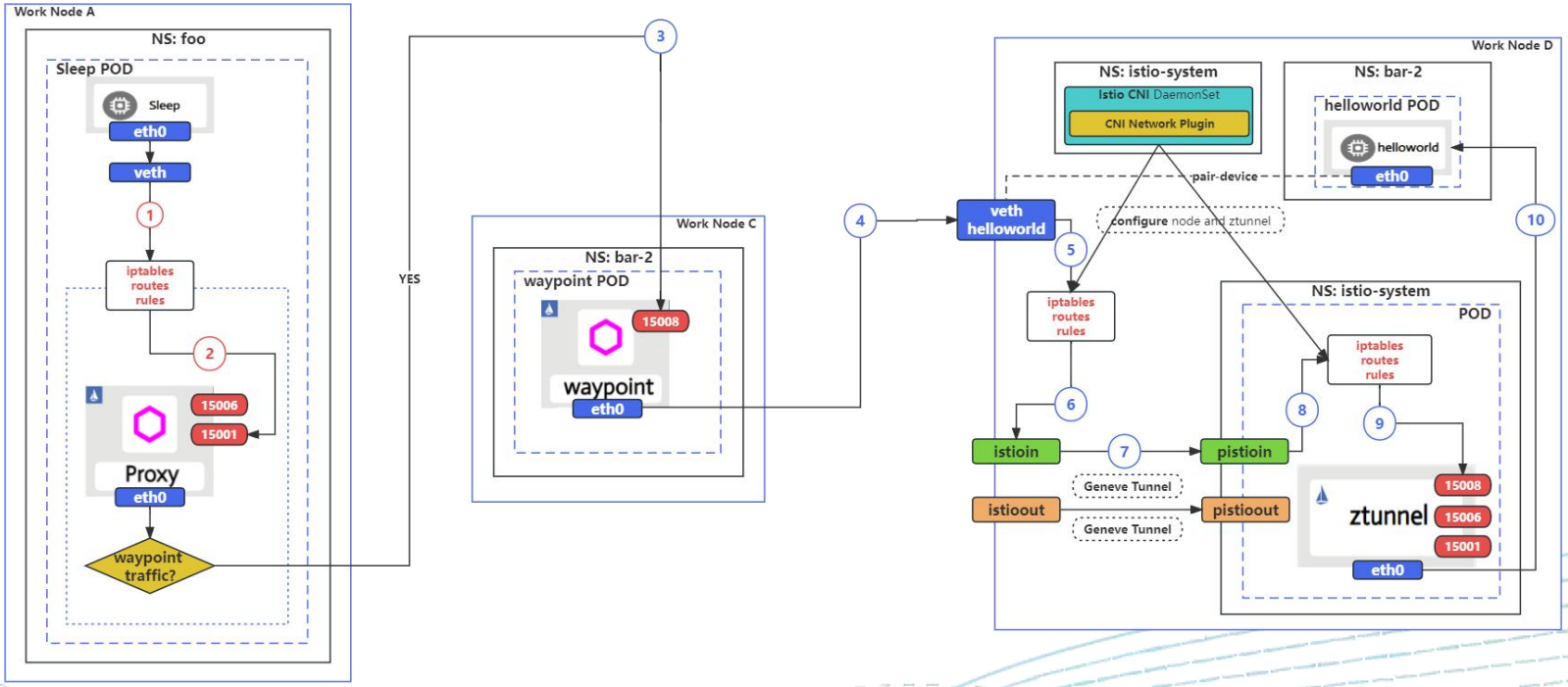
Sidecar 和 Ambient 共存下的流量路径分析

Sidecar -> Ambient (no waypoint)



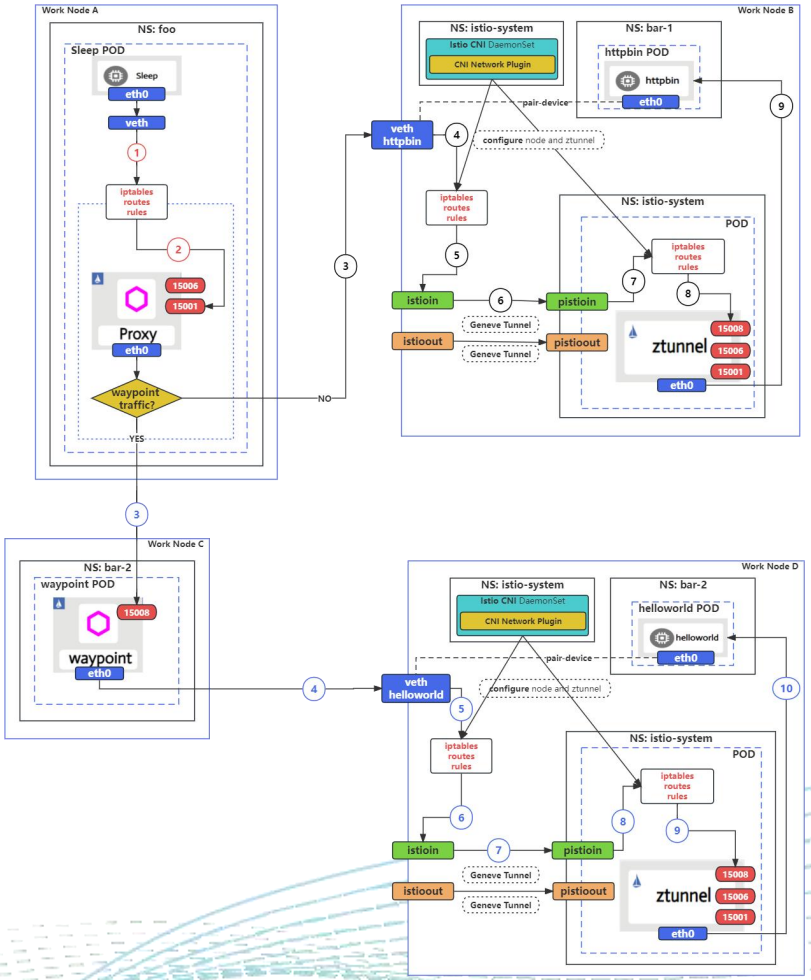
Sidecar 和 Ambient 共存下的流量路径分析

Sidecar -> Ambient （Waypoint)



Sidecar 和 Ambient 共存下的流量路径分析

Sidecar -> Ambient



总结

Ambient Mesh 采用的主要技术如下：

1. pod 流量基于iptables 和 策略路由被路由到istioin /istioout 设备
2. 使用 GENEVE (Generic Network Virtualization Encapsulation) 实现 istioin/istioout 与同节点的 ztunnel pod 之间的隧道链接
3. 使用 HBONE 建立隧道以在 Ztunnel 之间传递 TCP 流量
4. 使用 TPROXY 透明地拦截从主机 Pod 到 Ztunnel（Envoy Proxy）的流量

总结

Ambient Mesh 目前仍存在的一些问题，社区在积极推进优化：

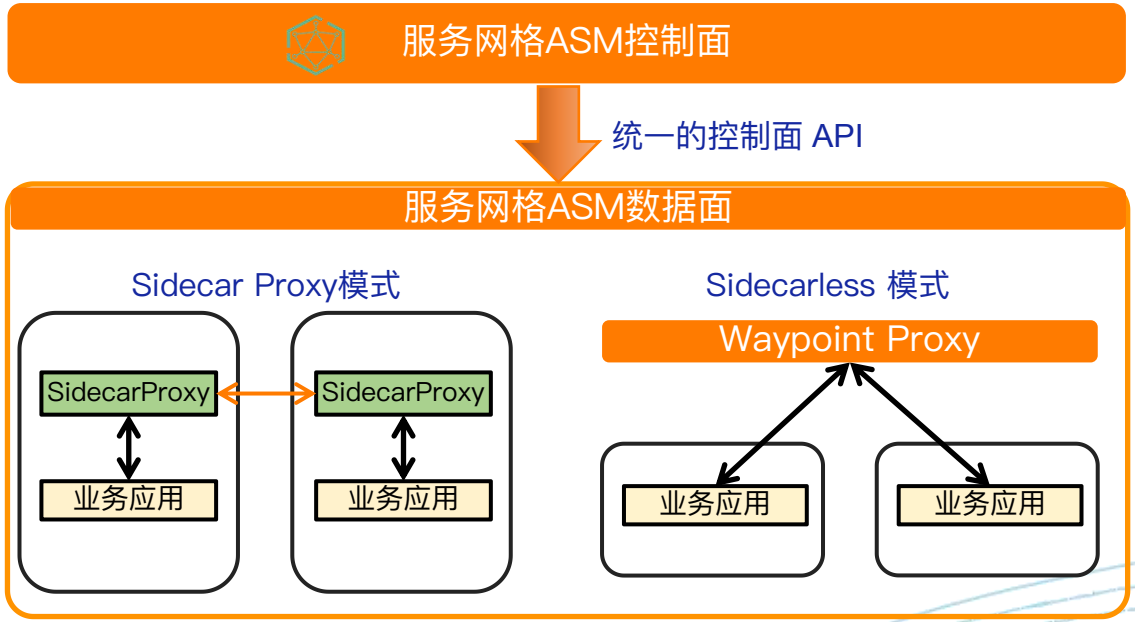
1. 对已有API 的覆盖度的问题
   - 对sidecar 模式下L7 traffic management API覆盖度的问题
     - sourceLabels (virtualservice)
   - Authentication 和 Authorization 策略相关的API覆盖度的问题
2. ztunnel 存在的问题
   - ztunnel 的L4 负载均衡目前实现是简单随机选择Endpoint
   - ztunnel 的单点故障和优雅升级问题，对mesh整体稳定性有较大影响
3. 性能和资源
   - 资源节省了多少？
   - 性能对比sidecar 有多少提升 ？
     - 整体性能提升
     - HBONE 协议对比mTLS 的差异
4. sidecar 如何平滑演进升级到Ambient mesh

Thank you!

zhlsunshine878@gmail.com
newday.jesse@gmail.com

Ambient 与sidecar 模式融合的技术架构

业界首个Sidecarless与Sidecar模式融合的服务网格平台，欢迎试用与交流！



多样的数据面形态，支持广泛的K8s环境（K8s版本 * K8s形态 * 节点规格 * OS版本）

钉钉群

扫一扫群二维码，立刻加入该群。