Istio Ambient Mesh + Kata 场景下的快速数据平面路径

Intel 云原生软件研发工程师 / 张怀龙

蚂蚁集团高级技术专家 / 李福攀

演讲议题

- 服务网格流量劫持的基本原理和演进现状
- Istio + Kata 流量劫持的实现方案及其存在的问题
- Istio Ambient Mesh + Kata 快速数据平面方案介绍

演讲议题

- **服务网格流量劫持的基本原理和演进现状**
- Istio + Kata 流量劫持的实现方案及其存在的问题
- Istio Ambient Mesh + Kata 快速数据平面方案介绍

服务网格流量劫持的基本实现

目前流量劫持的实现主要有基于 iptables/netfilter 和 eBPF 两种方案:

方案名称	实现分类	实现原理	代表项目	备注
iptables/netfilter	Sidecar Pre Pod	通过 Kubernetes 的 init-container 或者方案自身实现 CNI 组件去操作 CNI Network Plugin 完成 iptables rules 规则在 Pod 或者 Node 上的配置，最终实现服务网格中网络流量的劫持。	Istio and Linkerd	Istio Sidecar Istio Ambient Mesh
	Sidecar Per Node			
eBPF	Sidecar Pre Pod	通过方案自身实现 CNI 组件去加载流量劫持相关的 eBPF Program，并将其关联到 Linux 对应的 eBPF 挂载点，以此实现服务网格中网络流量的劫持和监控，以及网络通信的加速。	Cilium and Flomesh	
	Sidecar Per Node			
	Sidecarless			

Sidecar 模式下，mesh 和业务应用是相同pod中不同容器之间的流量交互。Ambient 模式下mesh和业务应用是相同节点中不同pod之间的流量交互

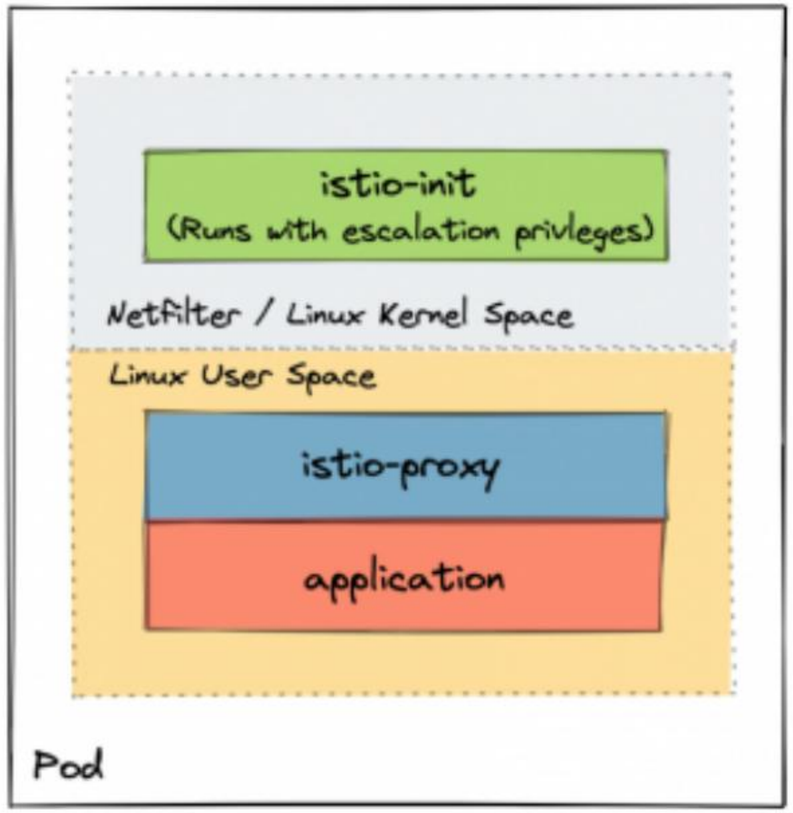
Istio 流量劫持之边车组件

❑ istio-init

- 作为 kubernetes 的 init container 为应用 Pod 的网络进行初始化为 Envoy 容器配置其应用的 iptables 规则
- 关键端口：15001 和 15006
- 新增规则链：ISTIO_INBOUND，ISTIO_IN_REDIRECT，ISTIO_OUTPUT 和 ISTIO_REDIRECT

❑ istio-proxy

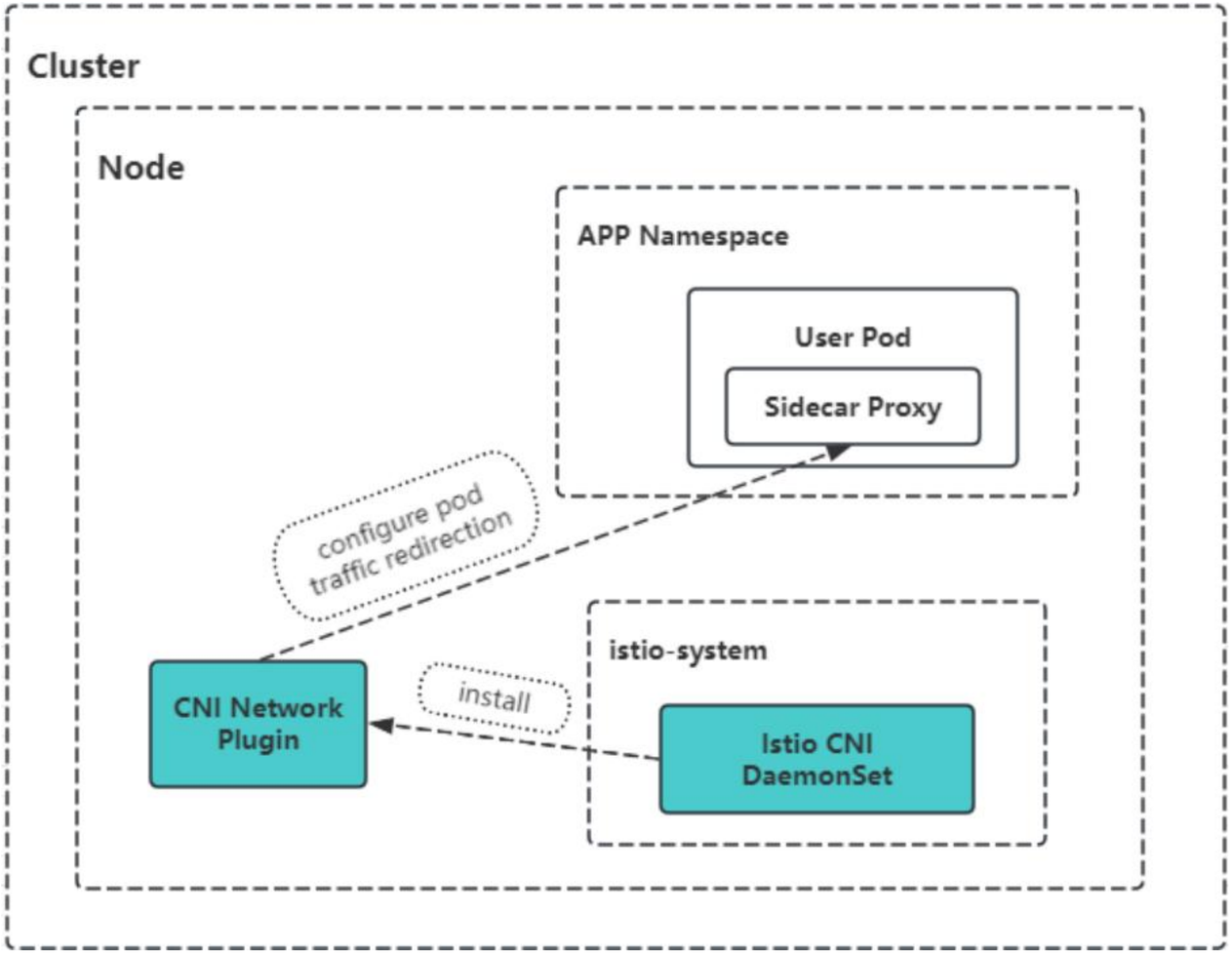
- 可以理解为对 Envoy proxy 的增强或者扩展，具体可以查看官方文档



图片来源

Istio 流量劫持之边车 Istio CNI

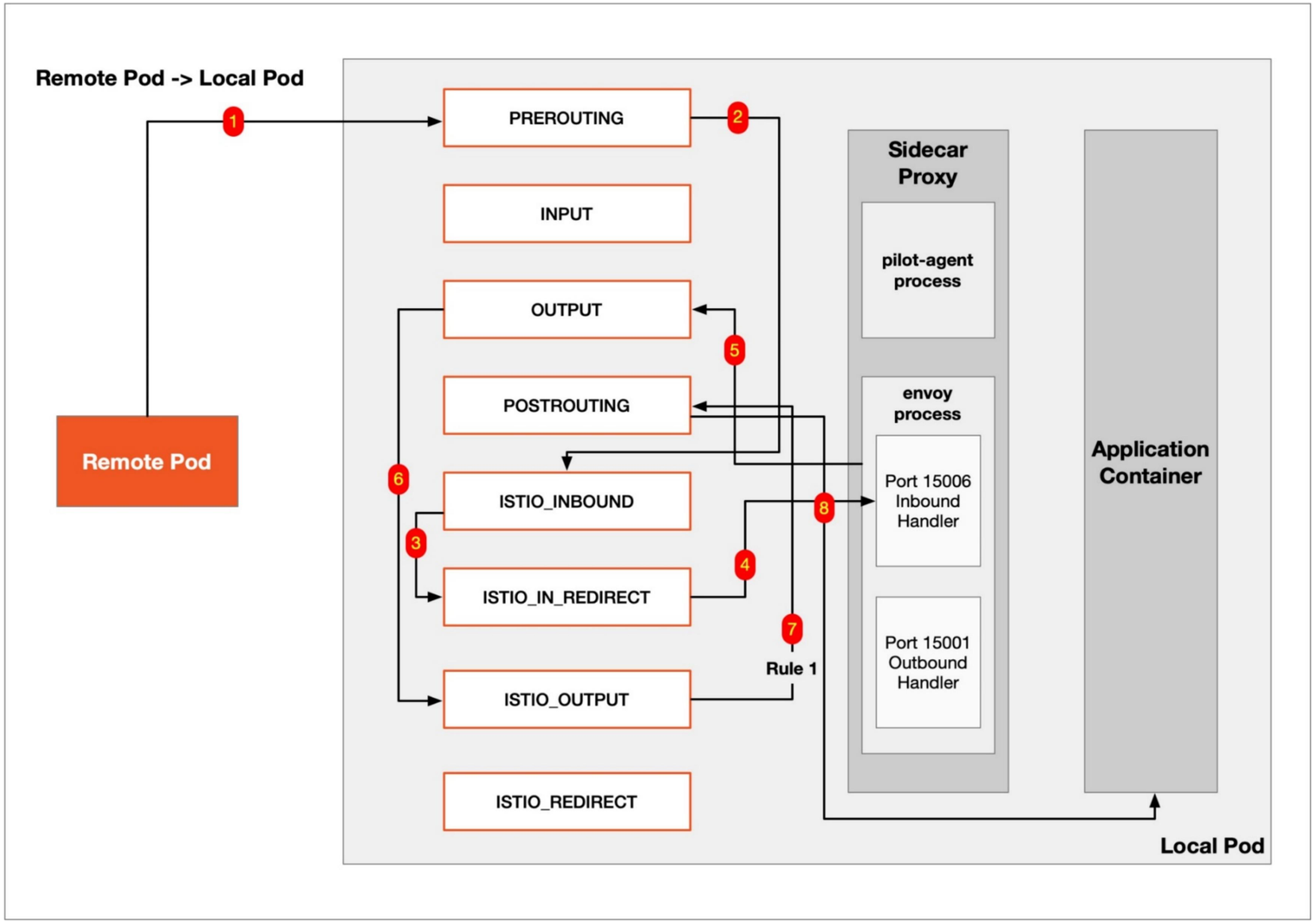
- **istio-validation**
  - istio-validation 与常规情况下的 istio-init 对等，负责设置为 Envoy 容器配置其应用的 iptables 规则
- **istio-cni**
  - 部署 istio-cni-node DaemonSet，负责安装CNI 网络插件完成 User Pod 边车代理的流量转发规则配置



Istio 边车流量劫持：从远端 Pod 到本地 Pod

对于从远端 Pod 到本地 Pod 数据包经过的链路如下：

- Remote Pod –> RREROUTING –> ISTIO_INBOUND –> ISTIO_IN_REDIRECT –> Envoy 15006 (Inbound) –> OUTPUT –> ISTIO_OUTPUT RULE 1 –> POSTROUTING –> Local Pod
- 备注： 数据包仅通过一次 Envoy Inbound 15006 端口

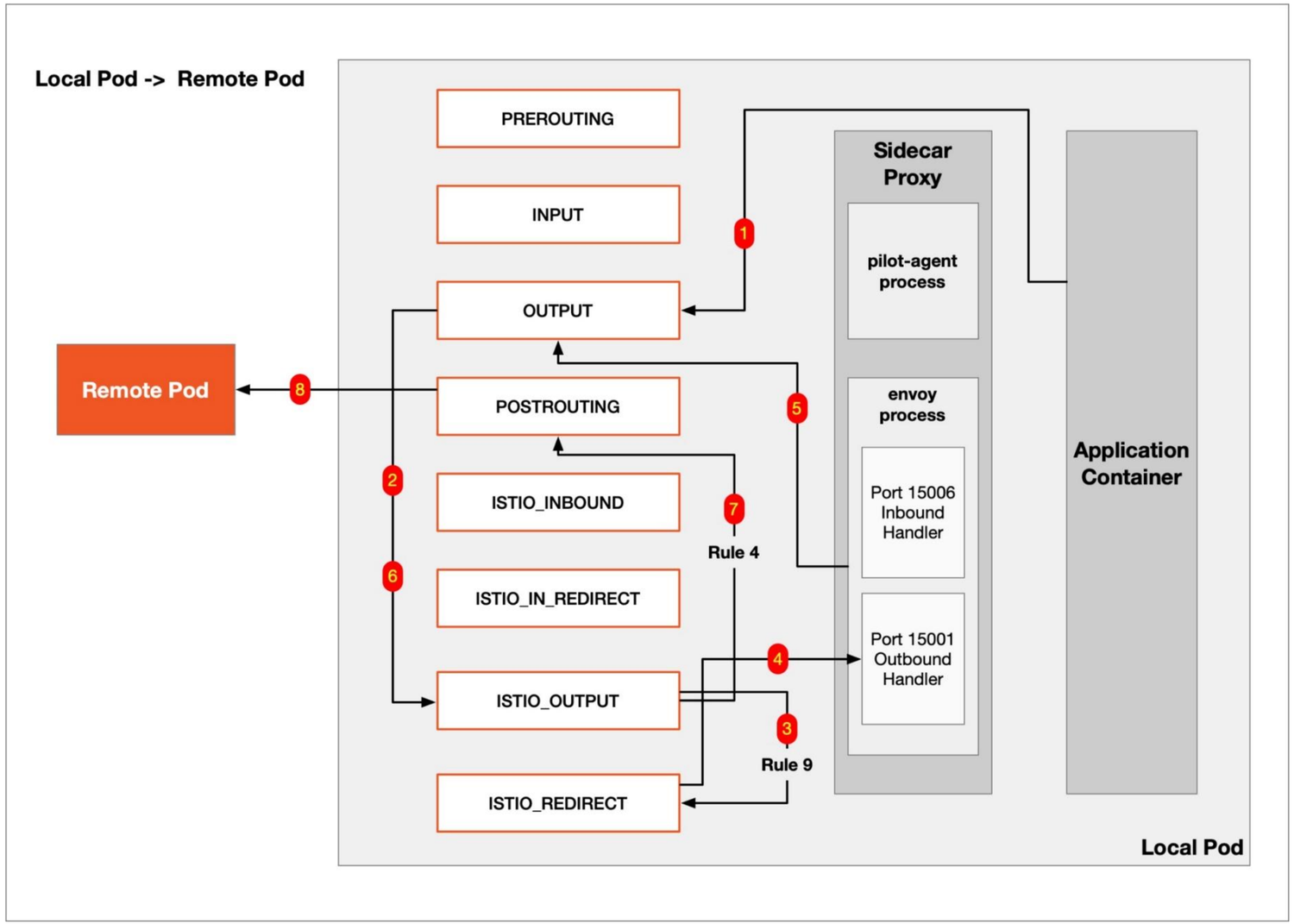


图片来源

Istio 边车流量劫持：从本地 Pod 到远端 Pod

对于从本地 Pod 到远端 Pod 数据包经过的链路如下：

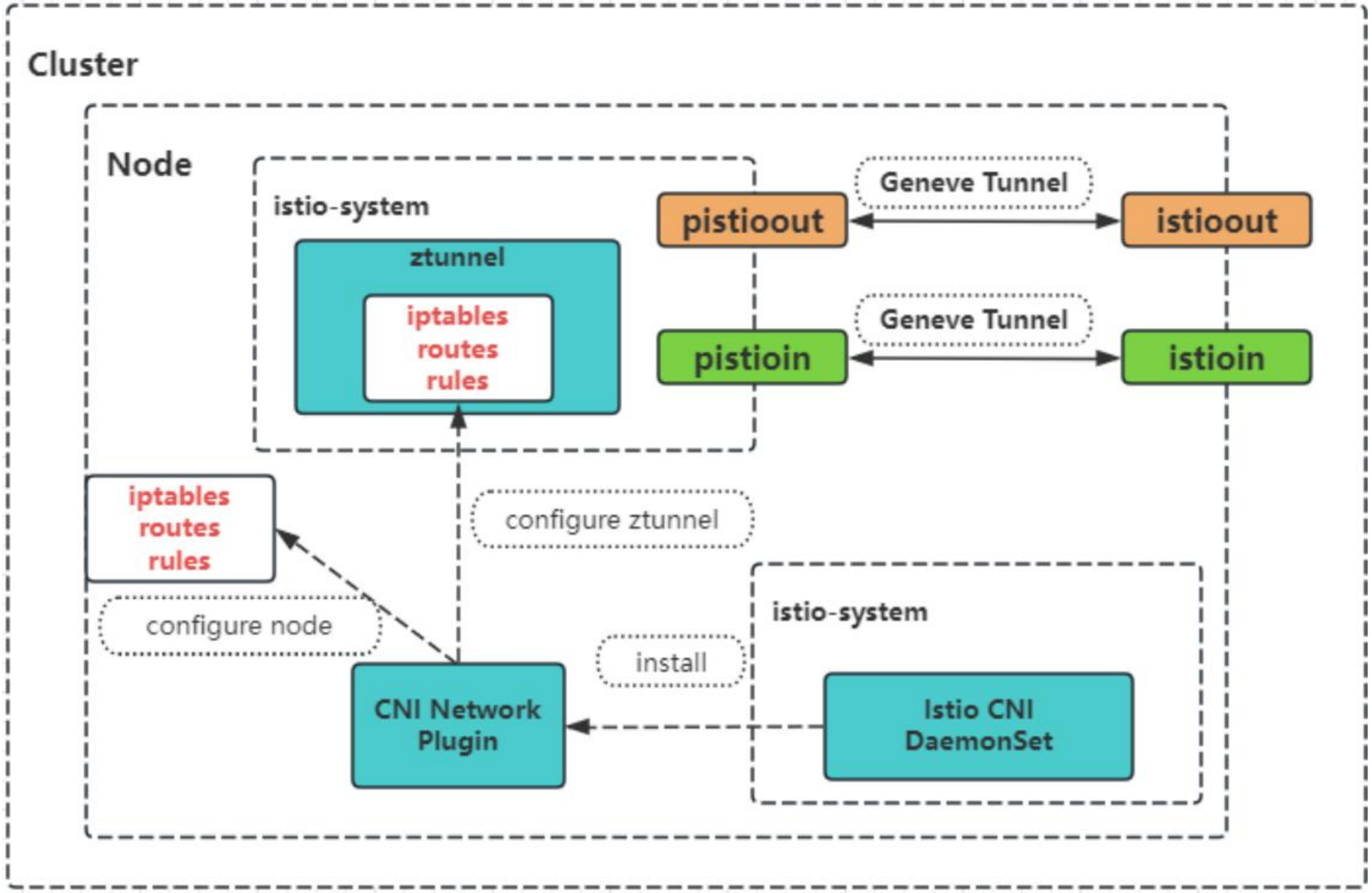
- Local Pod –> OUTPUT –> ISTIO_OUTPUT RULE 9 –> ISTIO_REDIRECT –> Envoy 15001 (Outbound) –> OUTPUT –> ISTIO_OUTPUT RULE 4 –> POSTROUTING –> Remote Pod
- 备注： 数据包仅通过一次 Envoy Outbound 15001 端口



图片来源

Istio 流量劫持之 Ambient Mesh 组件

- ztunnel
  - 是一个 Per-Node 的 DaemonSet，实现对本节点上所有用户 Pods 的流量劫持
- istio-cni
  - 部署 istio-cni-node DaemonSet，负责安装CNI 网络插件完成 node 和 ztunnel 的流量转发规则配置



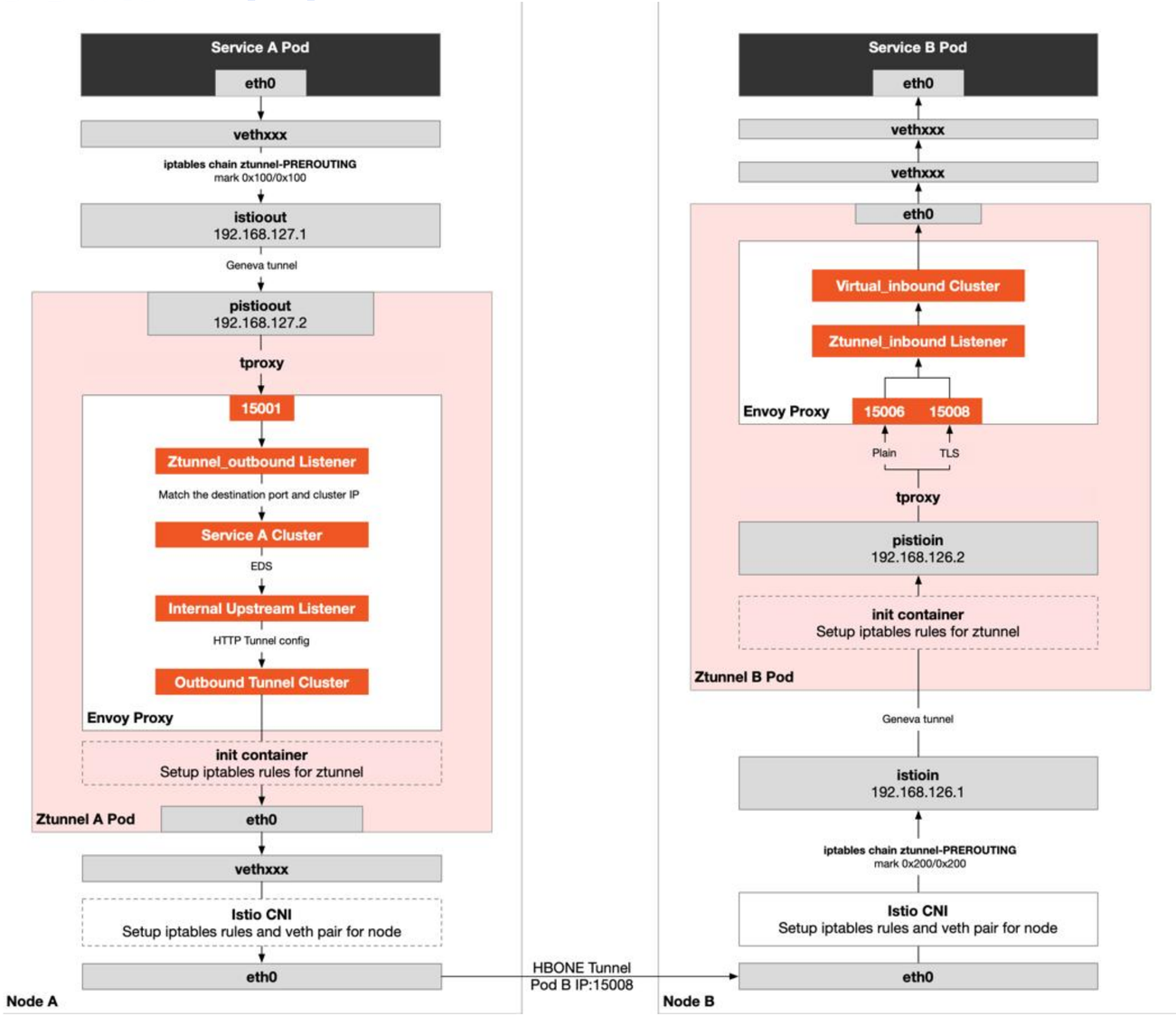
Istio Ambient Mesh 流量劫持流程图

对于从 SVC A Pod 到 SVC B Pod 数据包经过的链路如下：

Local Pod –> ztunnel–PREROUTING(使用 0x100/0x100 标记流量) –> **GENEVE**(istioout to pistioout) –> **TProxy** 15001 –> ztunnel–OUTPUT –> **HBONE** Tunnel –> ztunnel–PREROUTING(使用 0x200/0x200 标记流量) –> **GENEVE**(istioin to pistioin) –> **TProxy** (15006 for plain text, 15008 for HBONE) –> Remote Pod

Ambient Mesh 采用的主要技术如下：

1. 使用 GENEVE (Generic Network Virtualization Encapsulation) 实现节点与同节点的 ztunnel pod 之间的隧道链接
2. 使用 HBONE 建立隧道以在 Ztunnel 之间传递 TCP 流量
3. 使用 TPROXY 透明地拦截从主机 Pod 到 Ztunnel（Envoy Proxy）的流量



图片来源

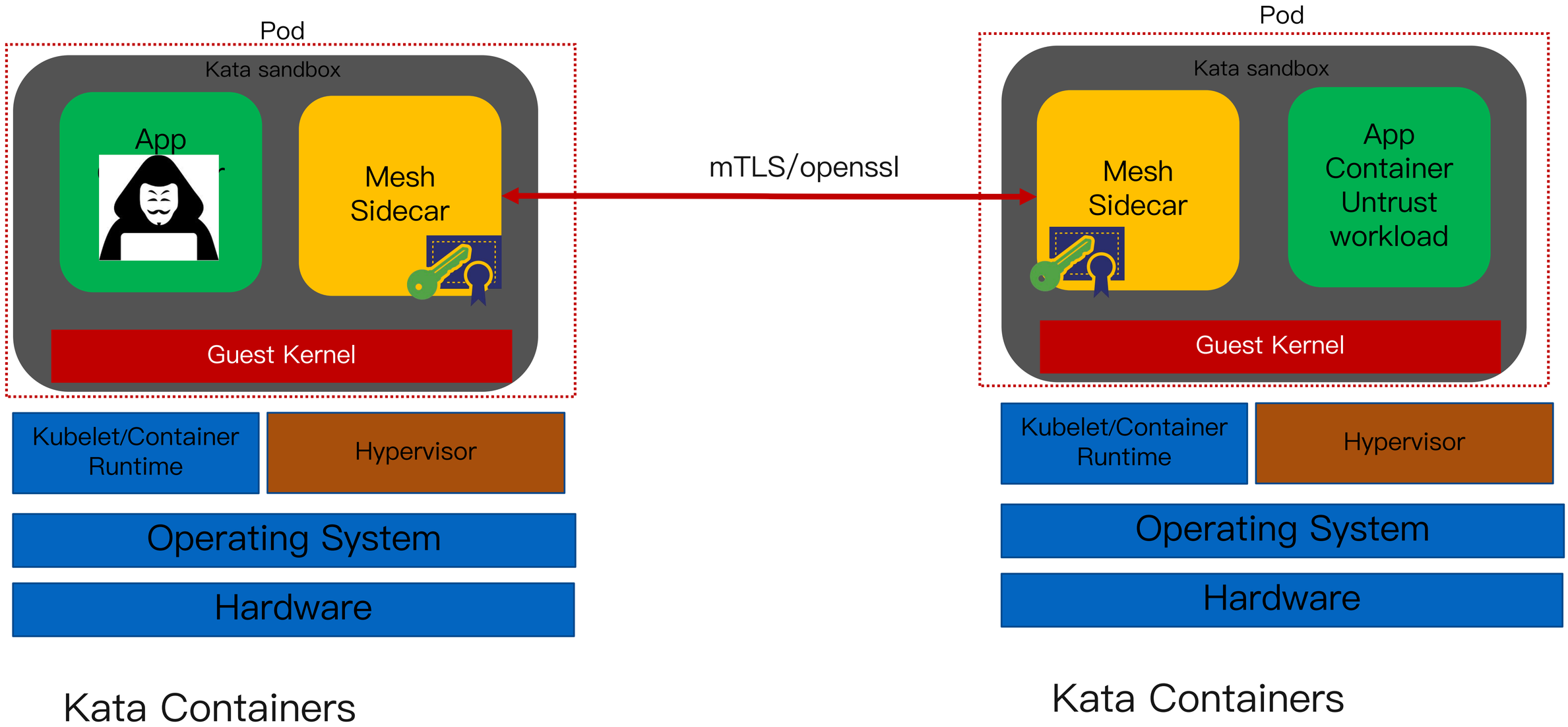
演讲议题

- 服务网格流量劫持的基本原理和演进现状
- Istio + Kata 流量劫持的实现方案及其存在的问题
- Istio Ambient Mesh + Kata 快速数据平面方案介绍

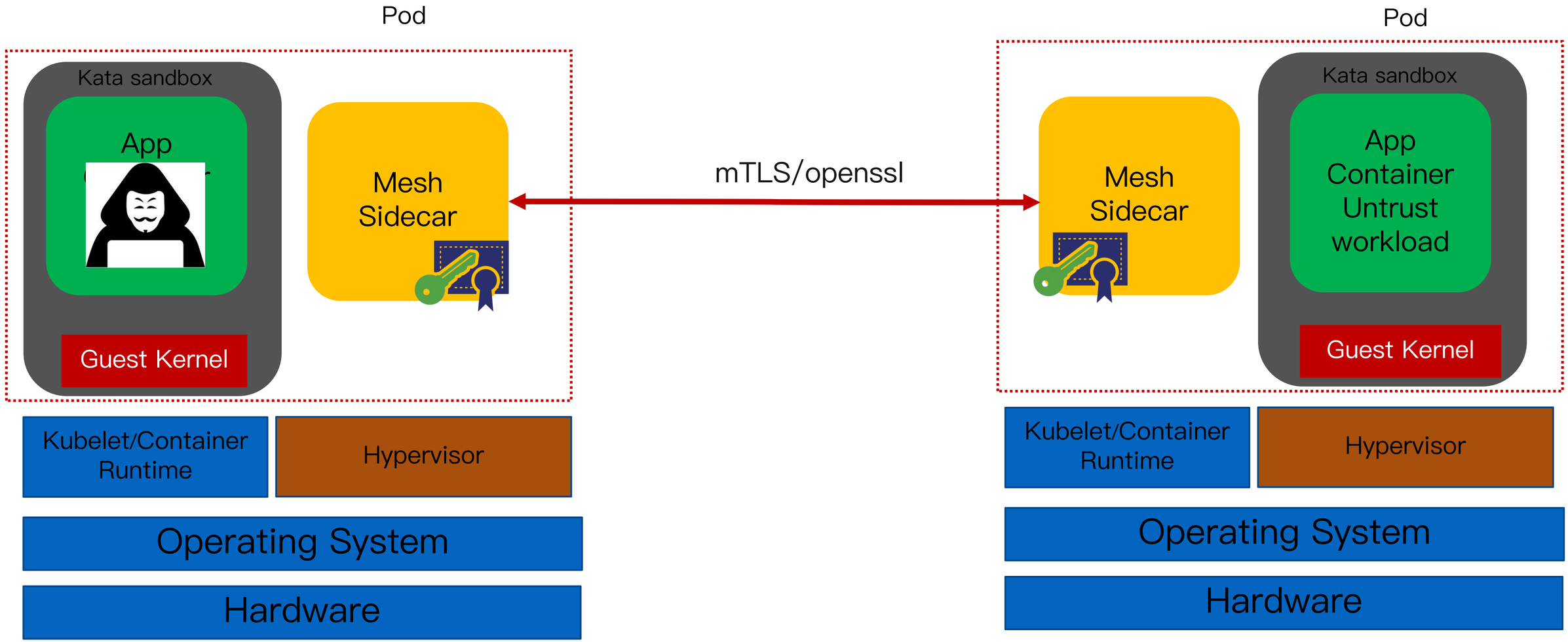
演讲议题

- 服务网格流量劫持的基本原理和演进现状
- **Istio + Kata 流量劫持的实现方案及其存在的问题**
- Istio Ambient Mesh + Kata 快速数据平面方案介绍

Kata 容器安全威胁模型



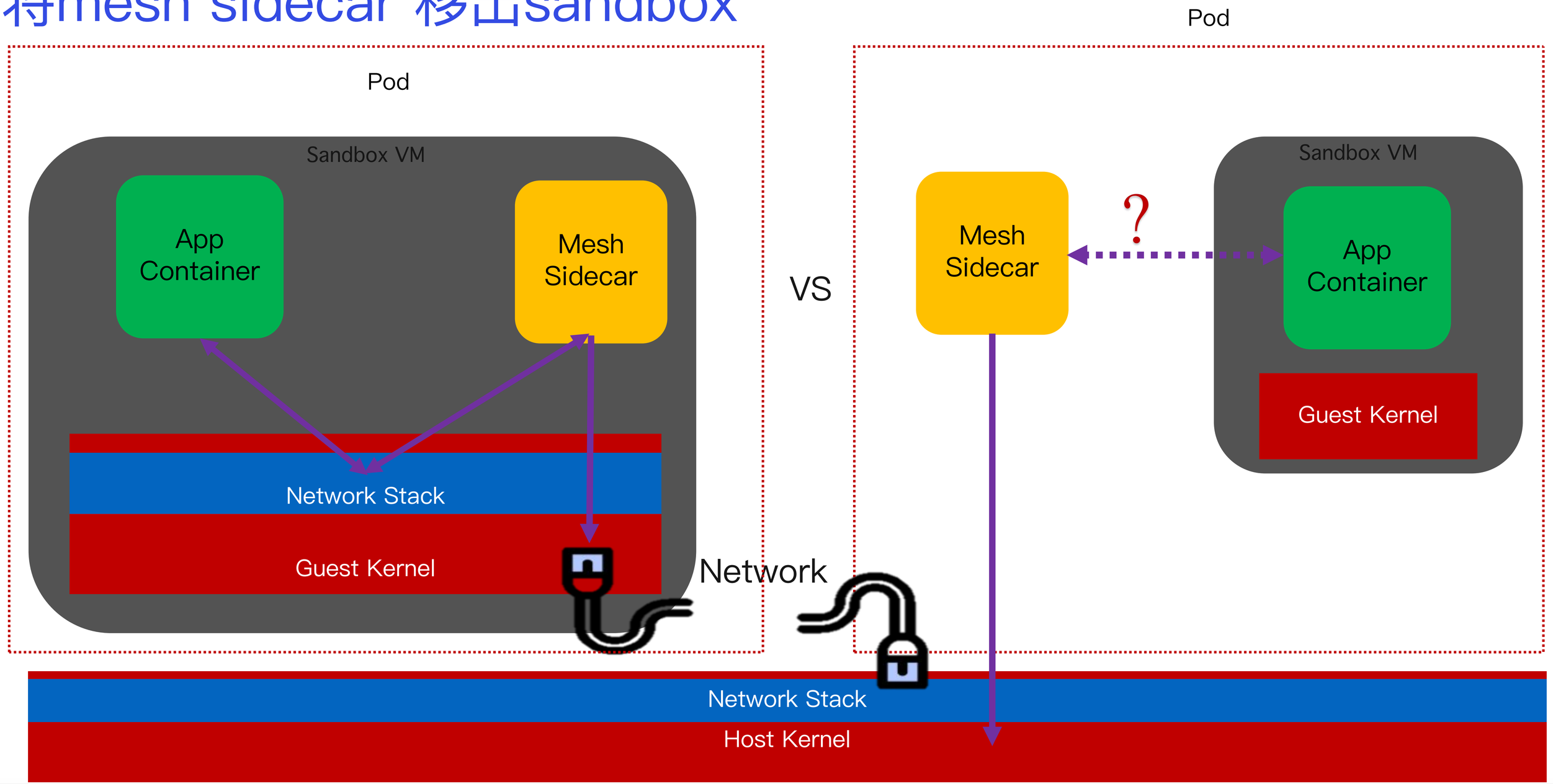
Kata 容器安全威胁模型



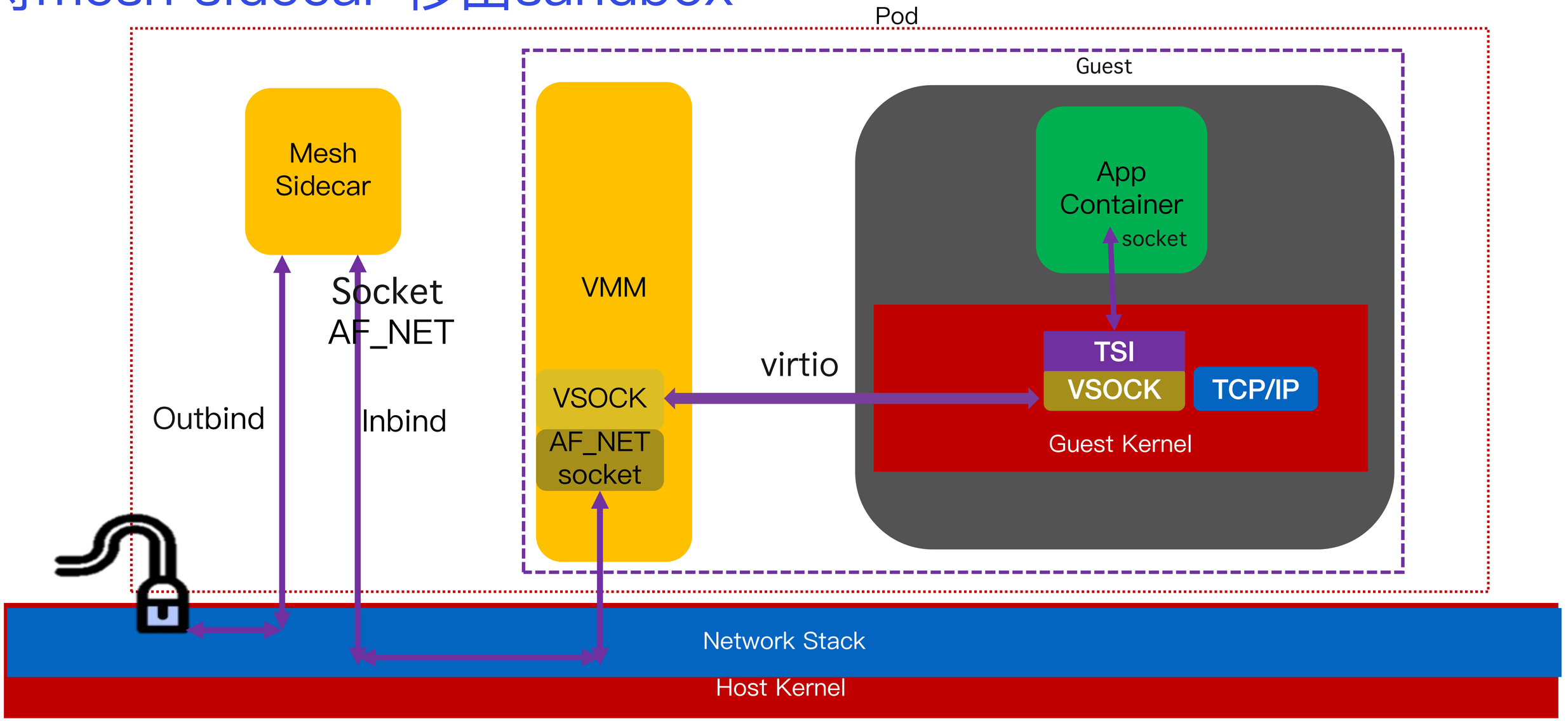
Kata Containers

Kata Containers

将mesh sidecar 移出sandbox



将mesh sidecar 移出sandbox

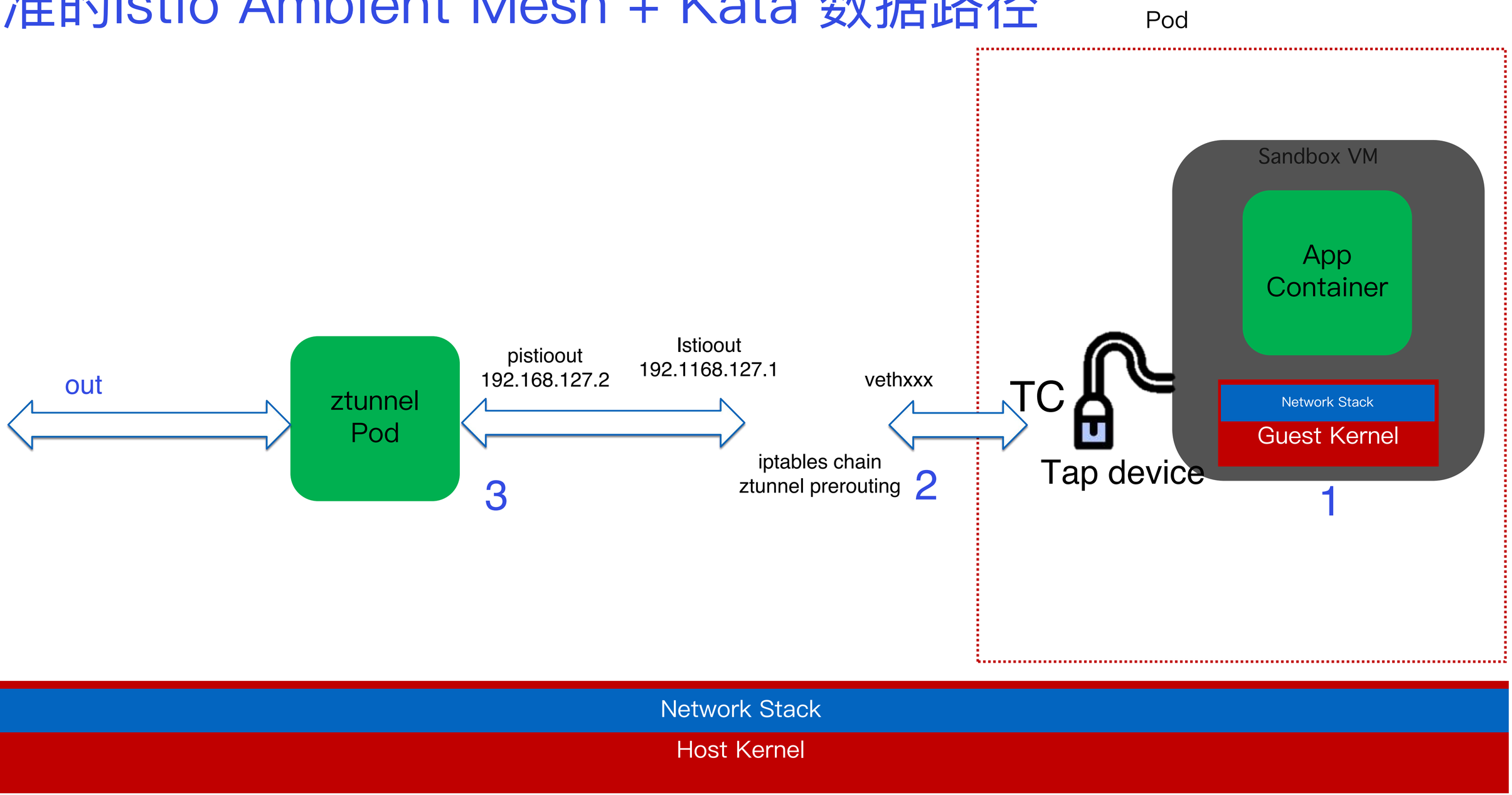


TSI(transparent socket Impersonation)： Developed by https://github.com/containers/libkrun

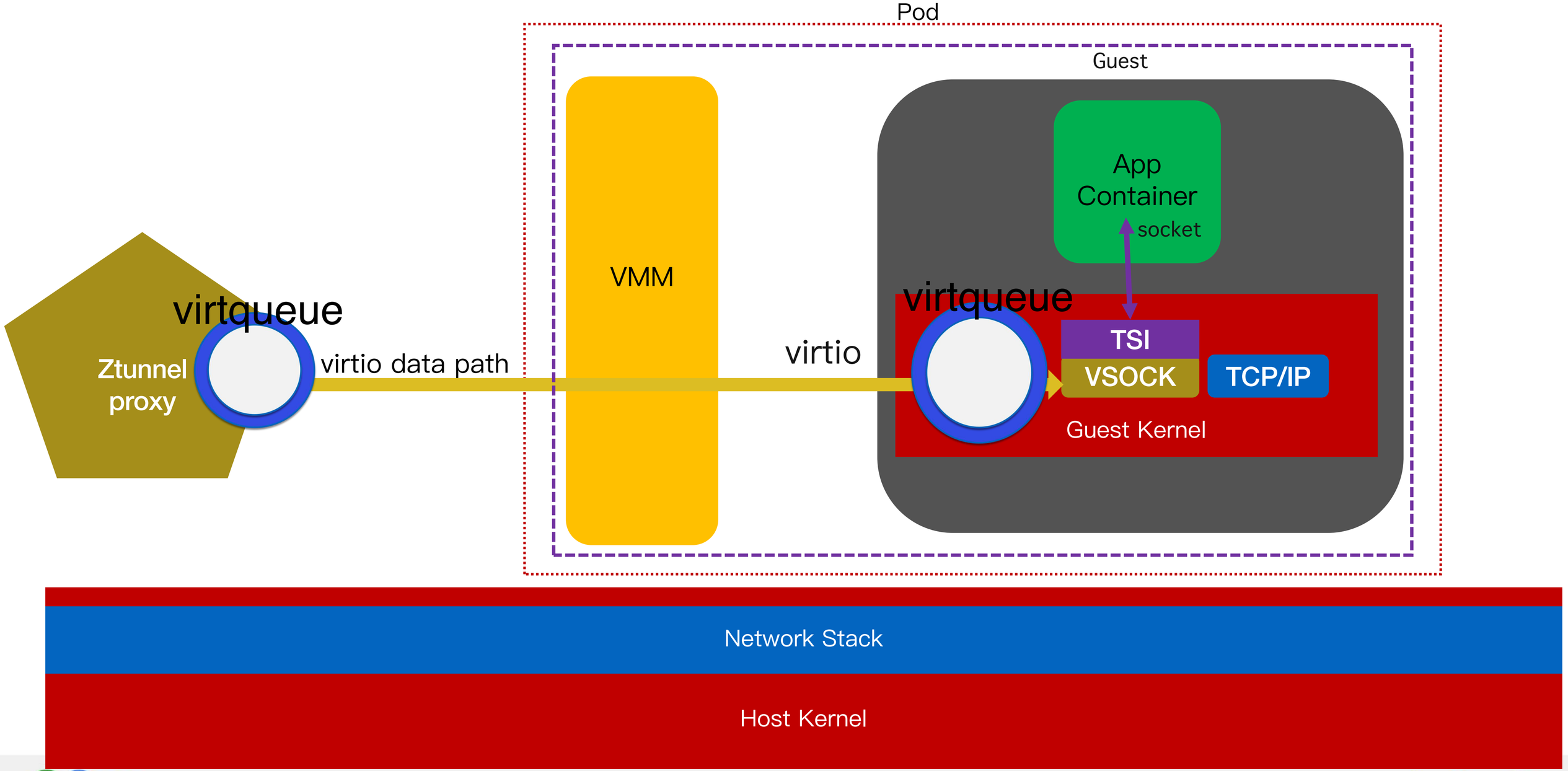
演讲议题

- 服务网格流量劫持的基本原理和演进现状
- Istio + Kata 流量劫持的实现方案及其存在的问题
- **Istio Ambient Mesh + Kata 快速数据平面方案介绍**

标准的Istio Ambient Mesh + Kata 数据路径

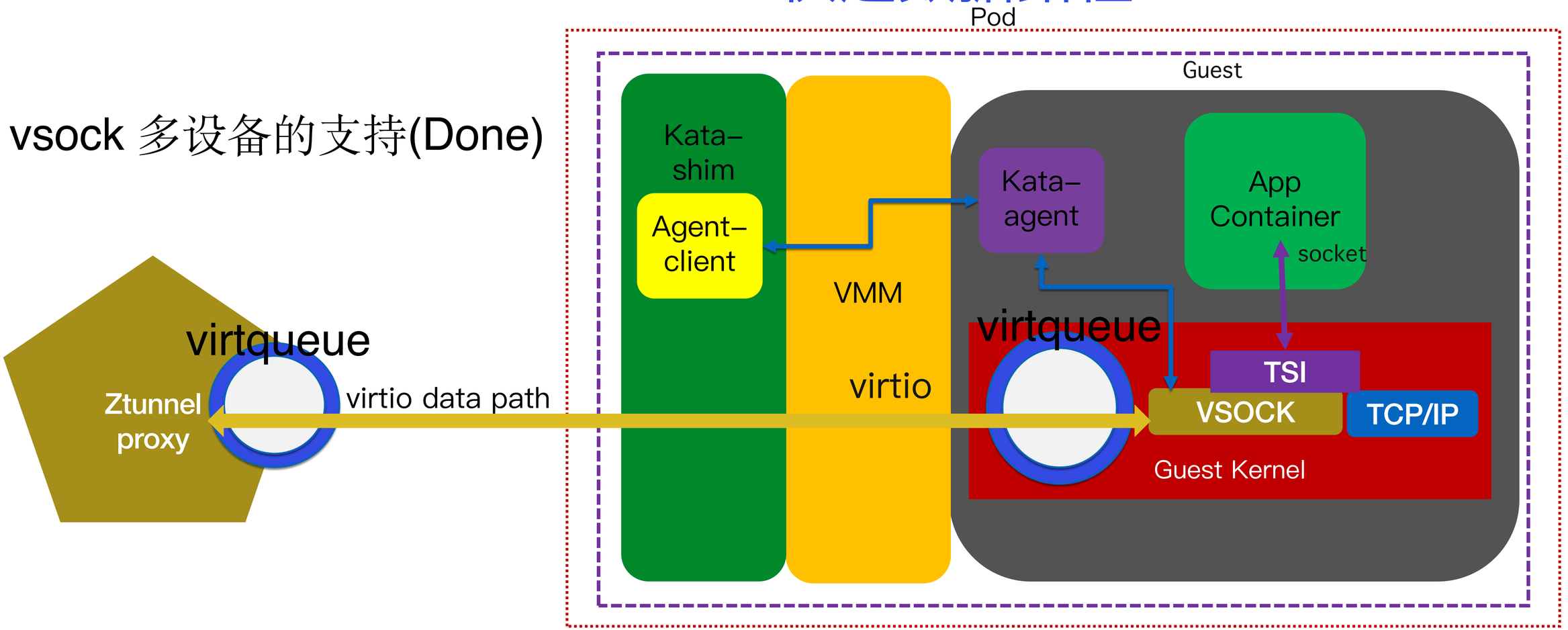


Istio Ambient Mesh + Kata 快速数据路径



Istio Ambient Mesh + Kata 快速数据路径

vsock 多设备的支持(Done)

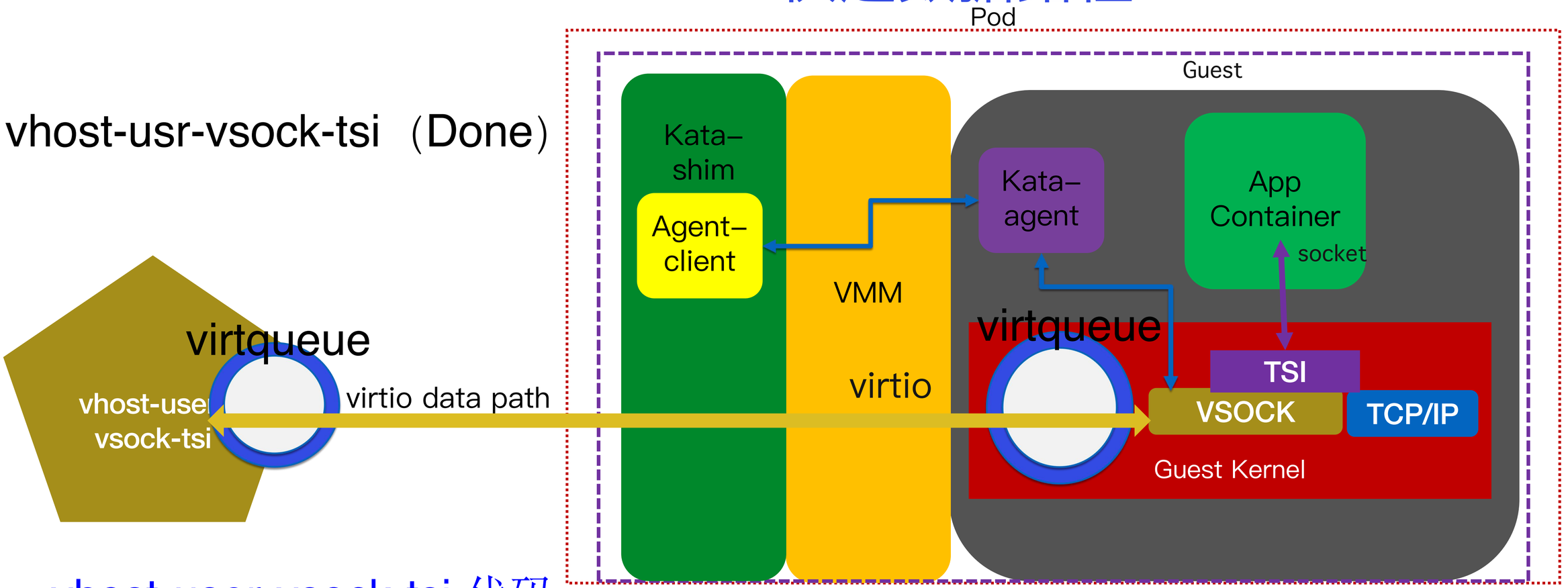


Network Stack

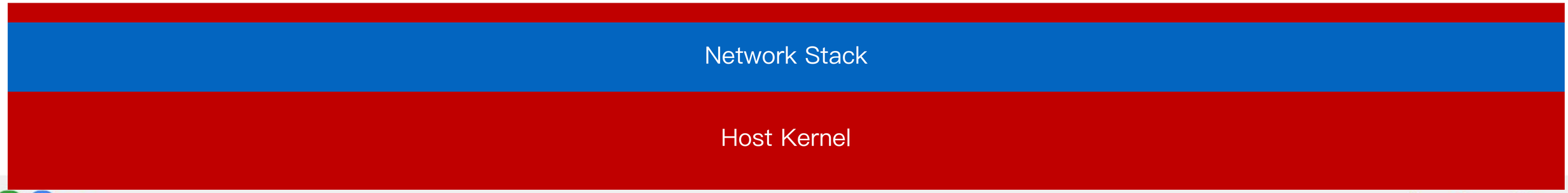
Host Kernel

Istio Ambient Mesh + Kata 快速数据路径

vhost-usr-vsock-tsi （Done）

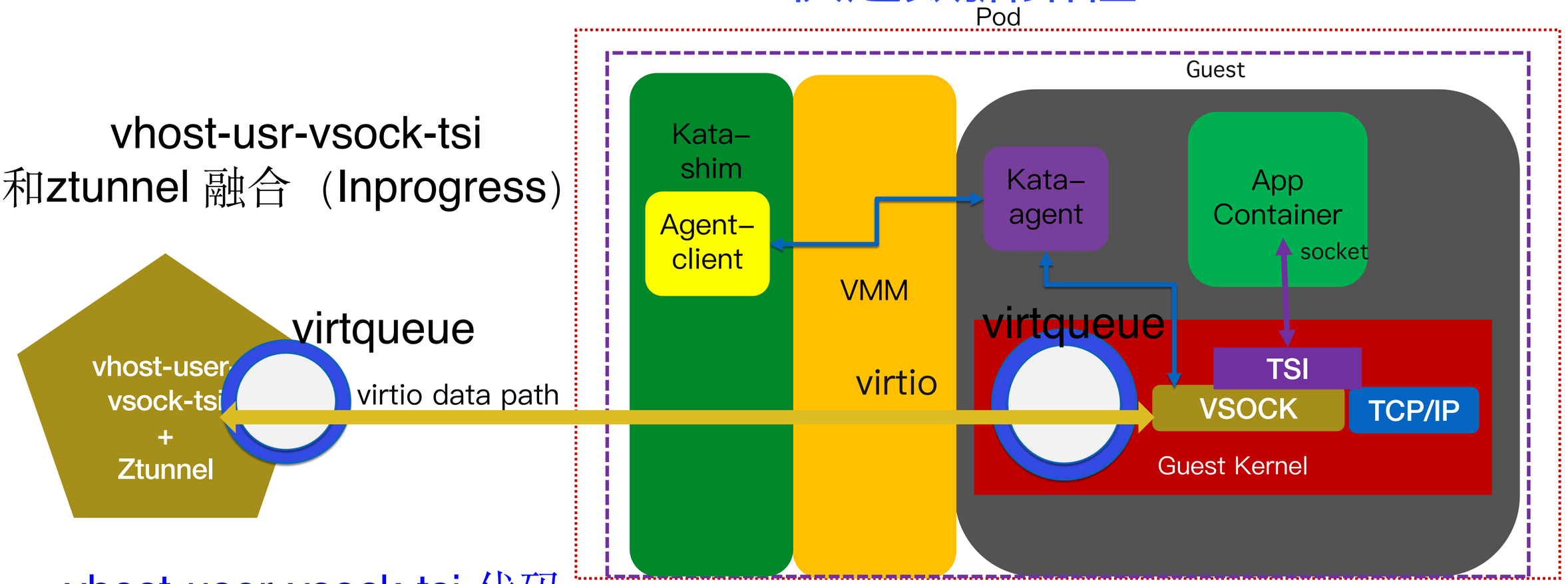


vhost-user-vsock-tsi 代码

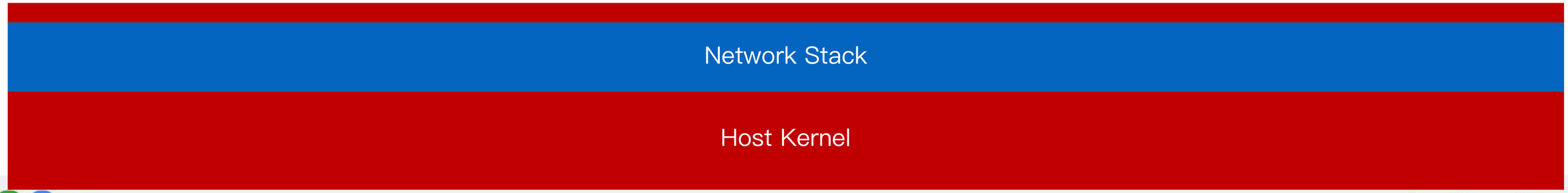


Istio Ambient Mesh + Kata 快速数据路径

vhost-usr-vsock-tsi
和ztunnel 融合（Inprogress）



vhost-user-vsock-tsi 代码



Ambient Mesh下的快速数据平面

1. 数据流量不在经过任何tcp/ip协议栈
2. 业务App 和ztunnel之间通过内存共享来交换数据

THANKS

软件正在重新定义世界

Software Is Redefining The World