Istio

通过sidecar proxy拦截所有微服务之间的通信。

istio简化服务级属性配置，如熔断器，超时，重试。让服务管理更轻松（广告语）。

追迹/监控/日志

虚拟服务+目的地规则

从简单的负载均衡到依据规则的路由。kubernetes是基于服务实例扩展进行负载均衡。可结合gateway控制ingress和egress流量。

service entry

服务入口

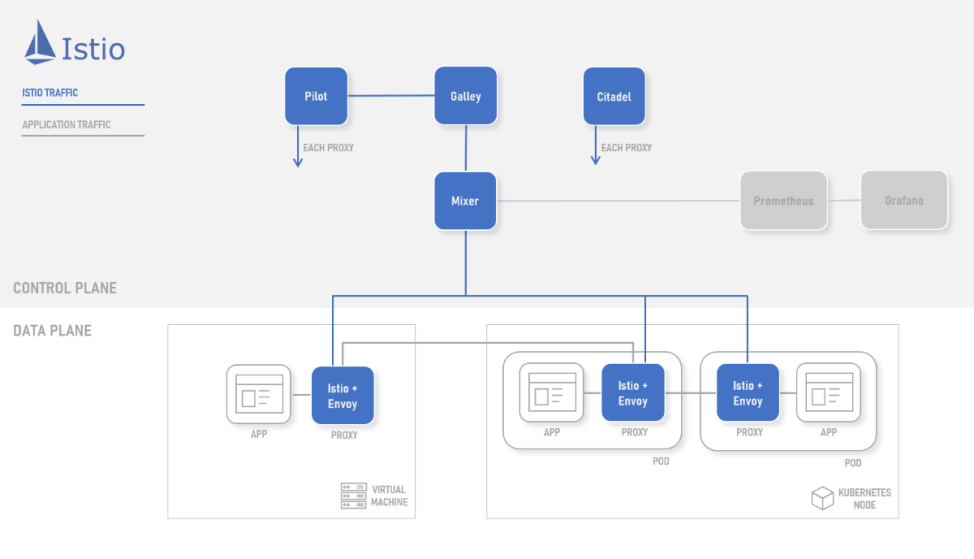
Service mesh重点在通信管理，包括信息加密/基于策略的路由/流量管理/本身行为可见性（日志/追迹/监控）

包括控制层和数据层

控制层负责管理规则策略和监控，数据层面实现功能（表现为sidecar形式的proxy）。

相比于k8s，service mesh提供更多/细的控制能力（服务间安全，跟踪，和细粒度负载均衡）。

* Consul - 服务管理框架，支持k8s
* Istio - 原生的k8s服务网格
* Linkerd - 和istio架构类似，强调简单性。



**Mixer** - Enforces access control and usage policies. Collects telemetry from the proxies that is pushed into [Prometheus](https://prometheus.io/). 执行访问控制和使用策略，收集代理信息。

**Pilot** - Provides service discovery and traffic management policy/configuration for the proxies. 服务发现和流量管理策略/配置

**Citadel** - Provides identity and security capabilities that allow for mTLS between services. 安全认证和加密

**Galley** - Abstracts and provides configuration to components. 配置

[1] <https://www.envoyproxy.io/>

[2] <https://platform9.com/blog/kubernetes-service-mesh-a-comparison-of-istio-linkerd-and-consul/>