HUAWEI

扫码添加小助手，发送“Istio”加群

HUAWEI

CloudNativeLives

istio入门级实训

Istio 服务发现和Pilot的架构机制

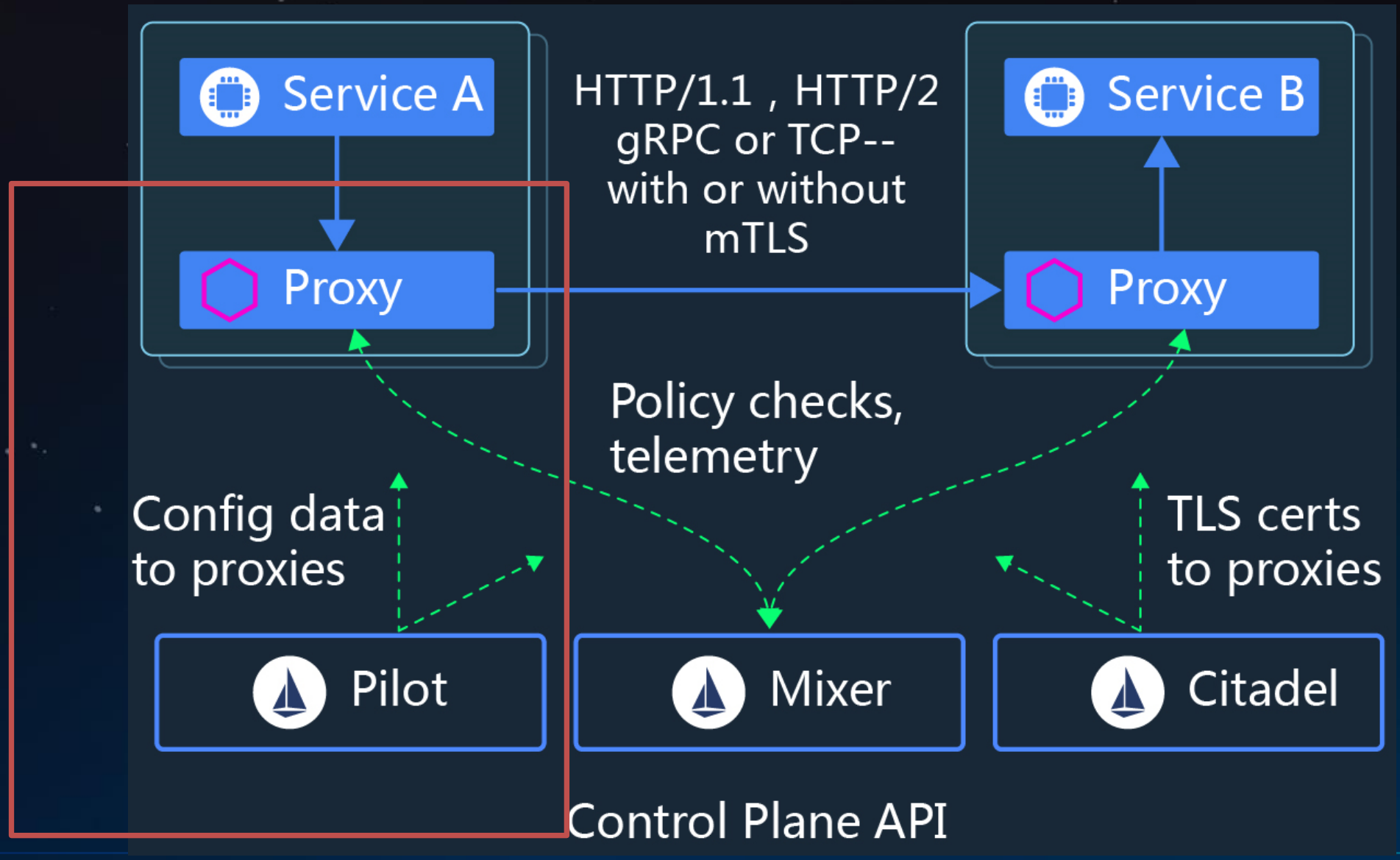
华为云容器团队核心架构师 & CNCF社区主要贡献者倾力打造

Agenda

HUAWEI

- Istio架构回顾&Pilot介绍
- Istio服务发现
- Istio服务配置
- Istio服务发现&规则管理与Kubernetes结合
- ShowCase

上期回顾：Istio架构



Pilot功能

HUAWEI

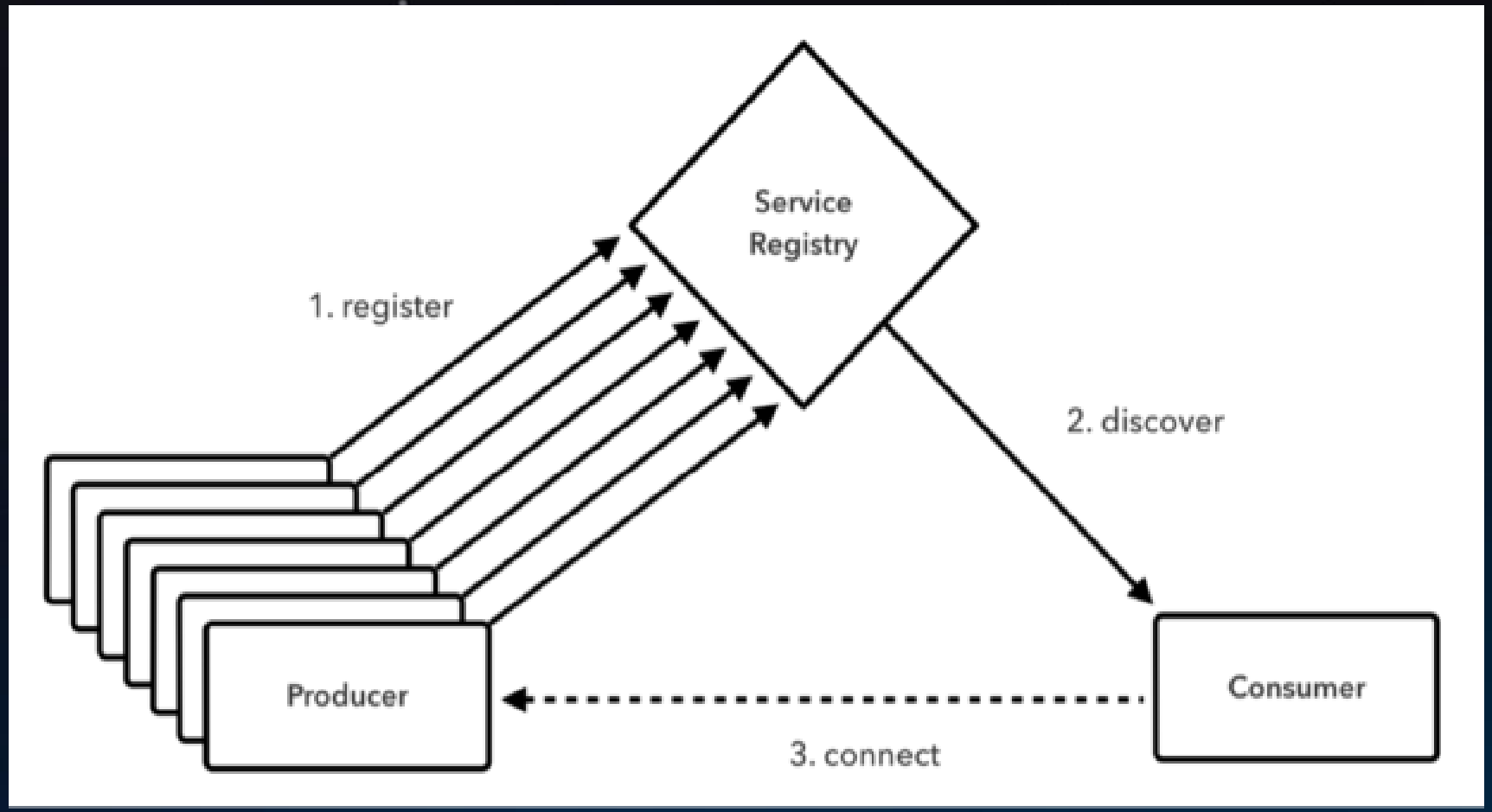
- 服务发现
- 服务配置

Agenda

HUAWEI

- Istio架构回顾&Pilot介绍
- Istio服务发现
- Istio服务配置管理
- Istio服务发现&规则管理与Kubernetes结合

服务发现基本原理

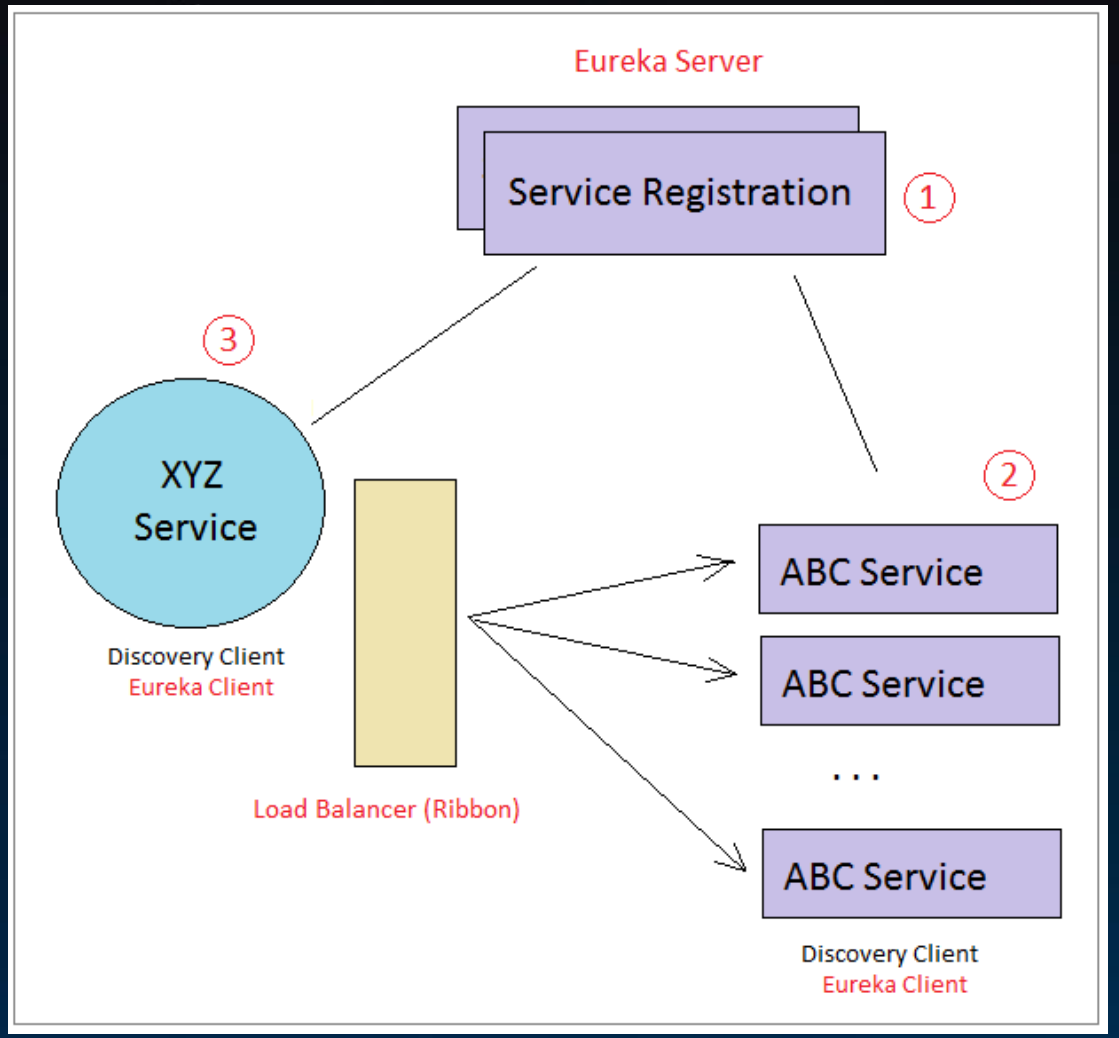


a.app 88.88.88.66
a.app 88.88.88.77
a.app 88.88.88.88

b.app 88.88.88.99
b.app 88.88.88.55

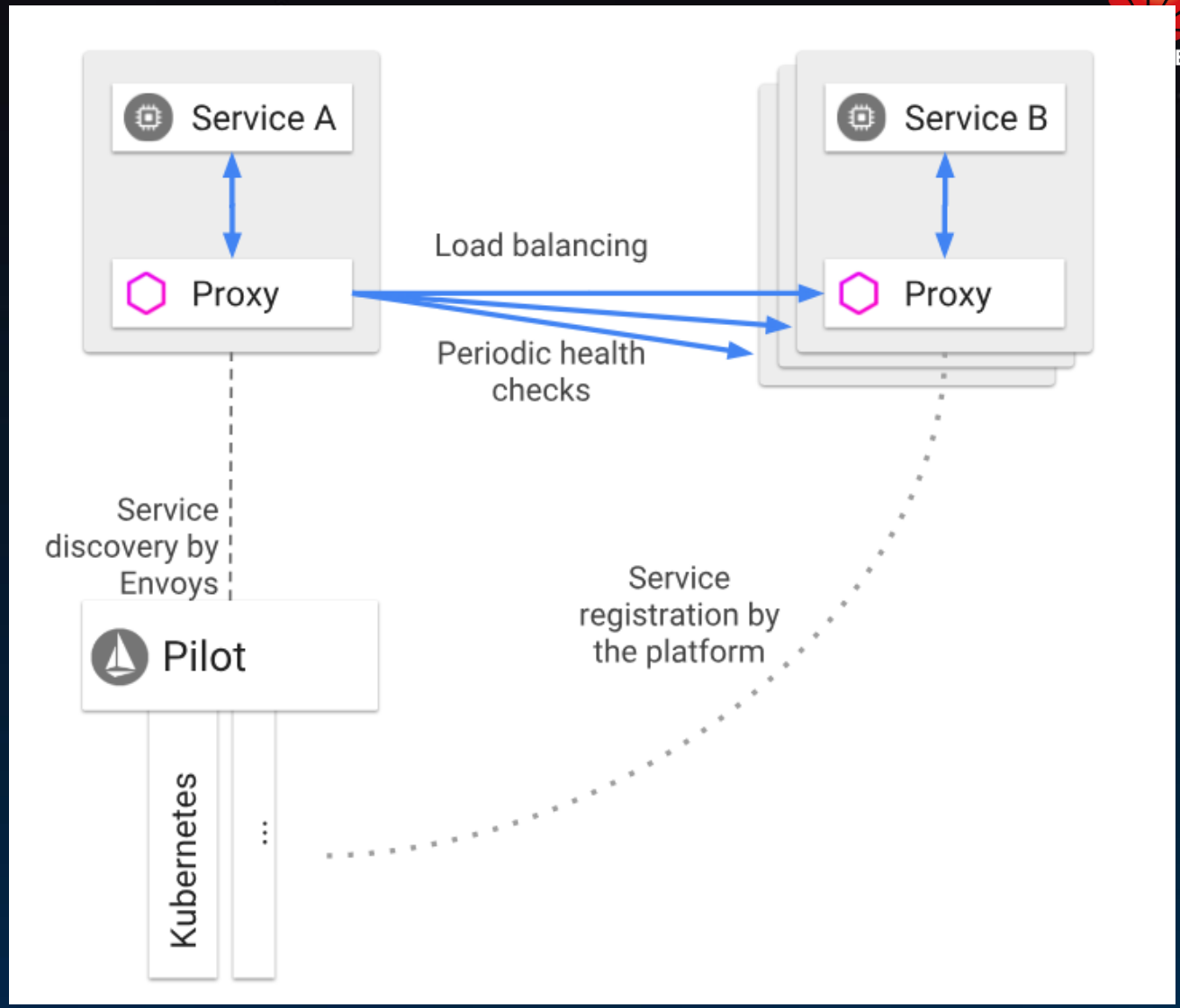
SpringCloud的服务(注册与)发现流程

- 服务注册表：如Springcloud中一般Eureka服务；
- 服务注册：服务配置文件中配置服务名和本实例地址，实例启动时自动注册到服务注册表；
- 服务发现：访问目标服务时连服务注册表，获取服务实例列表。根据LB根据策略选择一个服务实例，建立连接去访问。

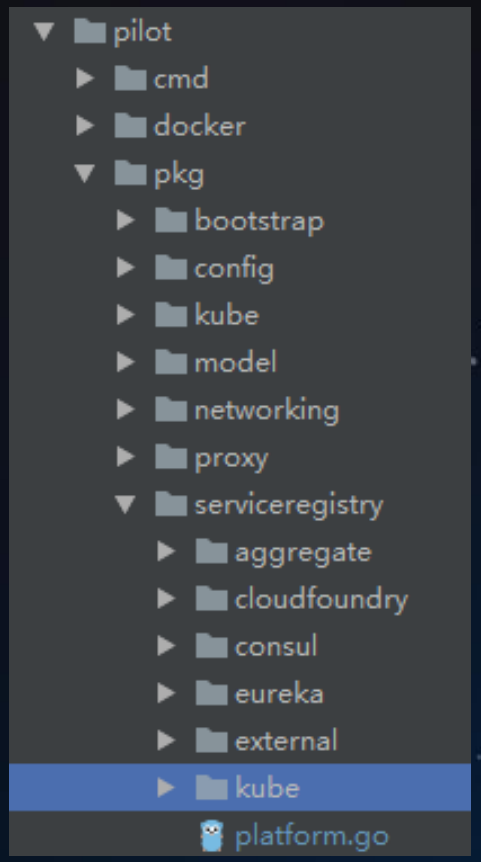


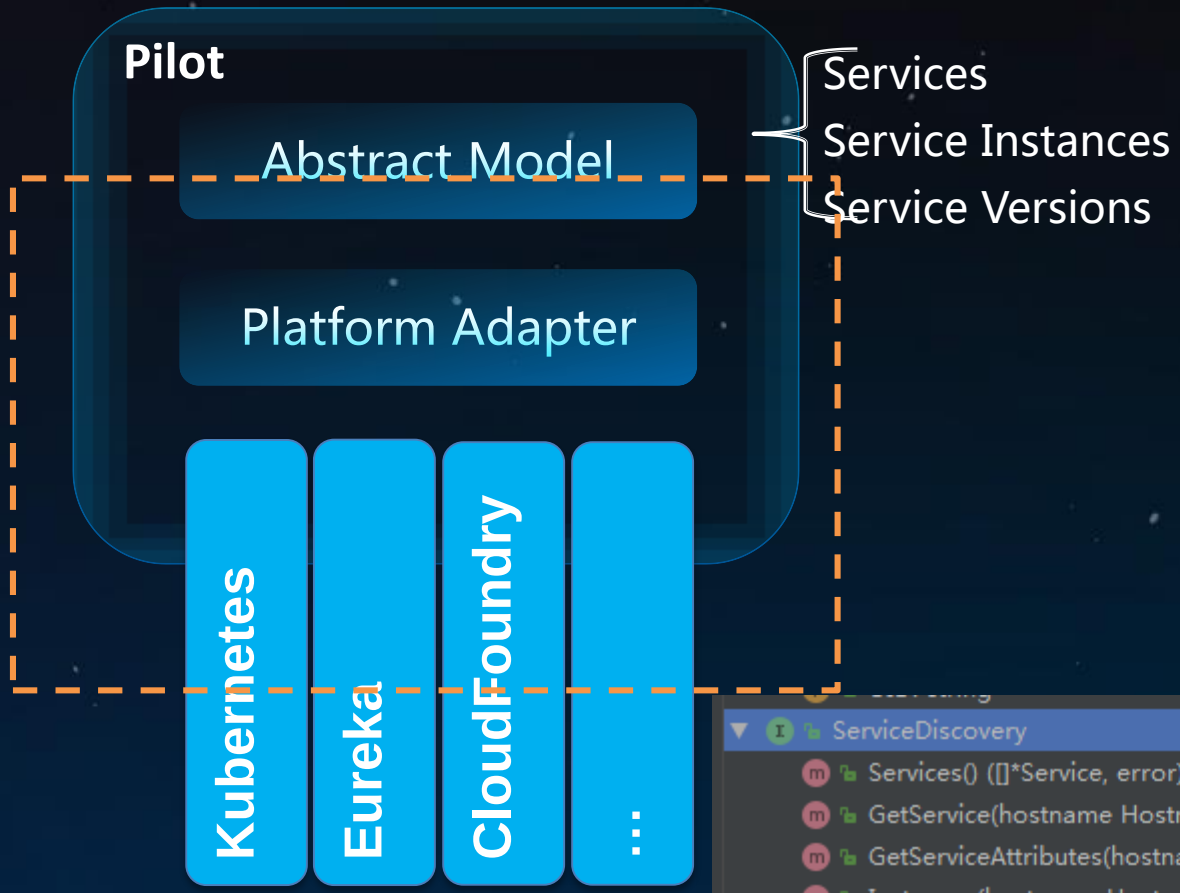
Istio服务发现流程

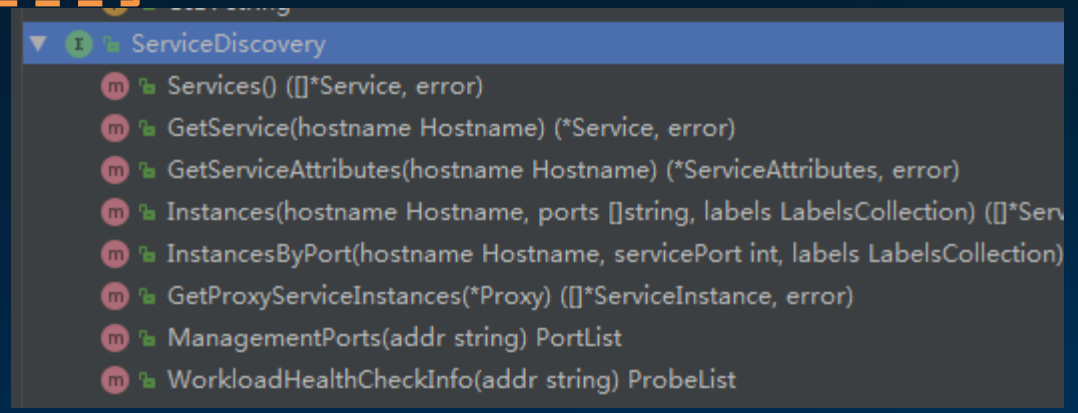
- 服务注册表：Pilot 从平台获取服务发现数据，并提供统一的服务发现接口。
- 服务注册：无
- 服务发现：Envoy 实现服务发现，动态更新负载均衡池。在服务请求时使用对应的负载均衡策略将请求路由到对应的后端。



Pilot服务发现机制的Adapter机制

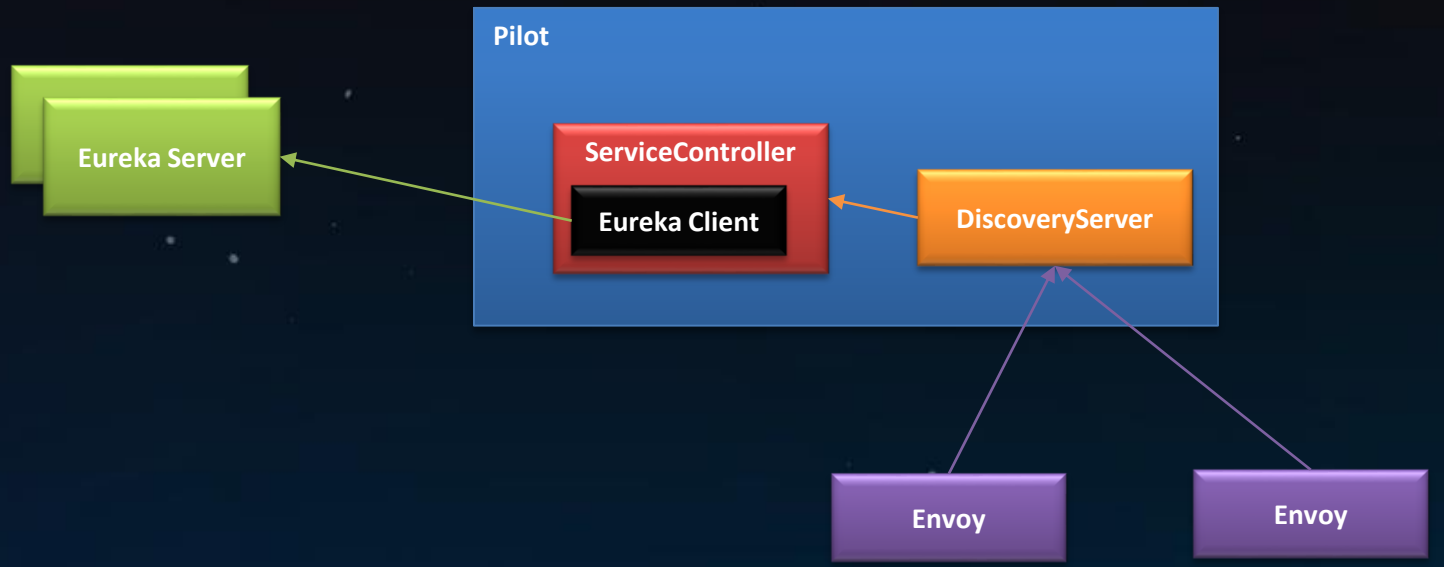






Istio服务发现实现：基于 Eureka

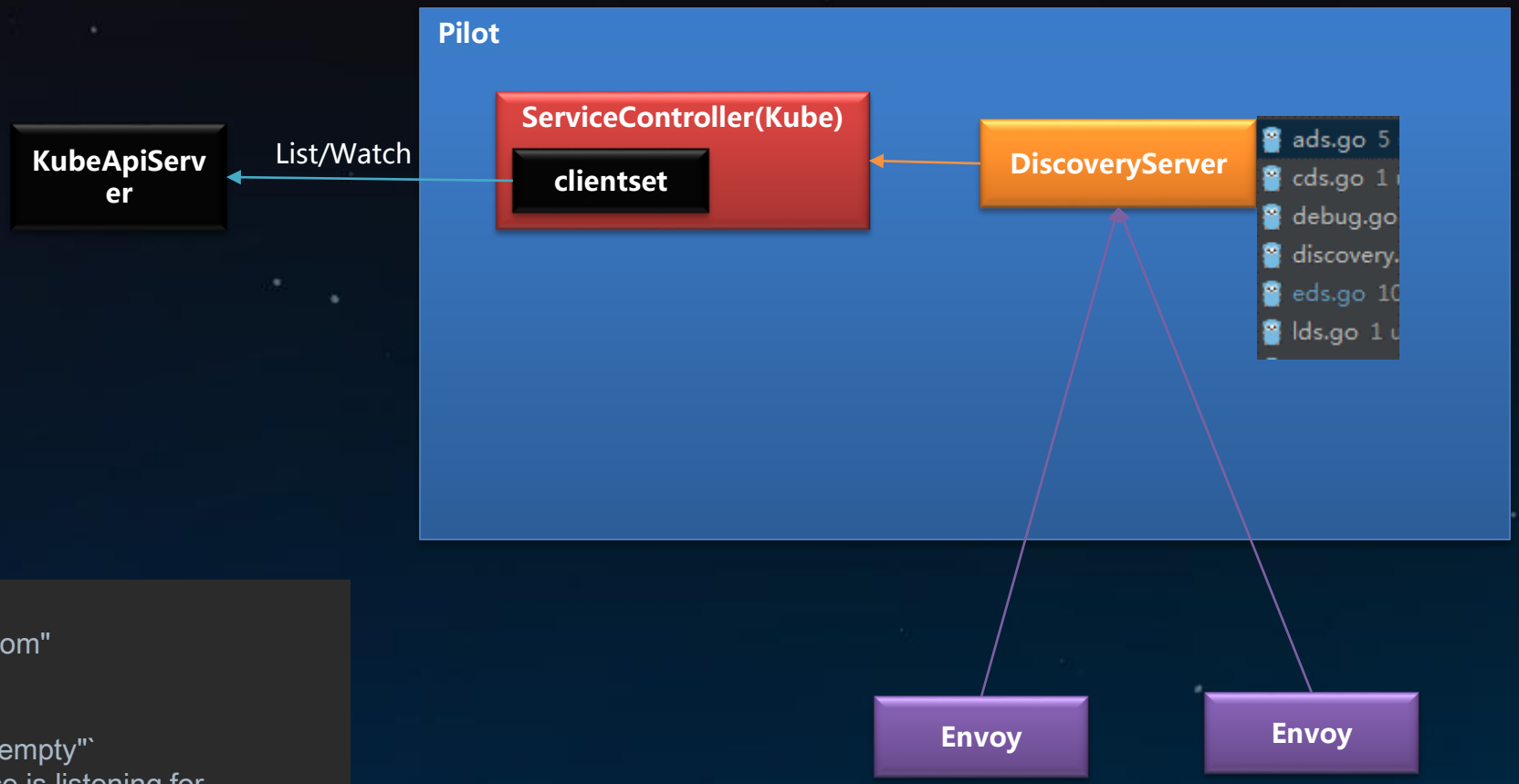
1. Pilot 实现若干服务发现的接口定义
2. Controller使用EurekaClient来获取服务列表，提供转换后的标准的服务发现接口和数据结构；
3. Discoveryserver基于Controller上维护的服务发现数据，发布成gRPC协议的服务供Envoy使用。
4. 当有服务访问时，Envoy在处理Outbound请求时，根据配置的LB策略，选择一个服务实例发起访问




```
public interface ServiceInstance {
   String getServiceId();
   String getHost();
   int getPort();
   boolean isSecure();
   URI getUri();
   Map<String, String> getMetadata();
}
```

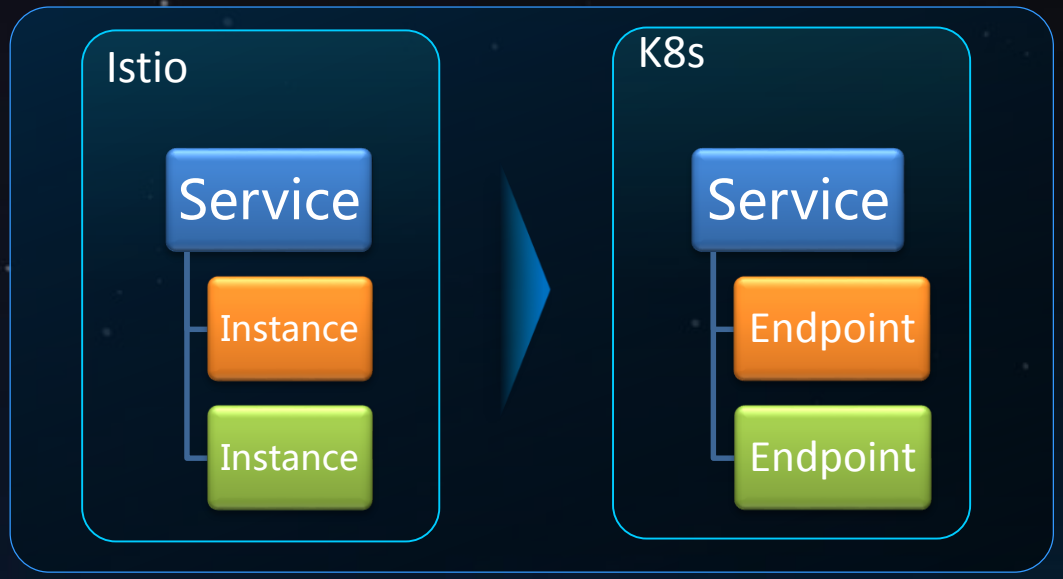
Istio 服务发现实现：基于Kubernetes

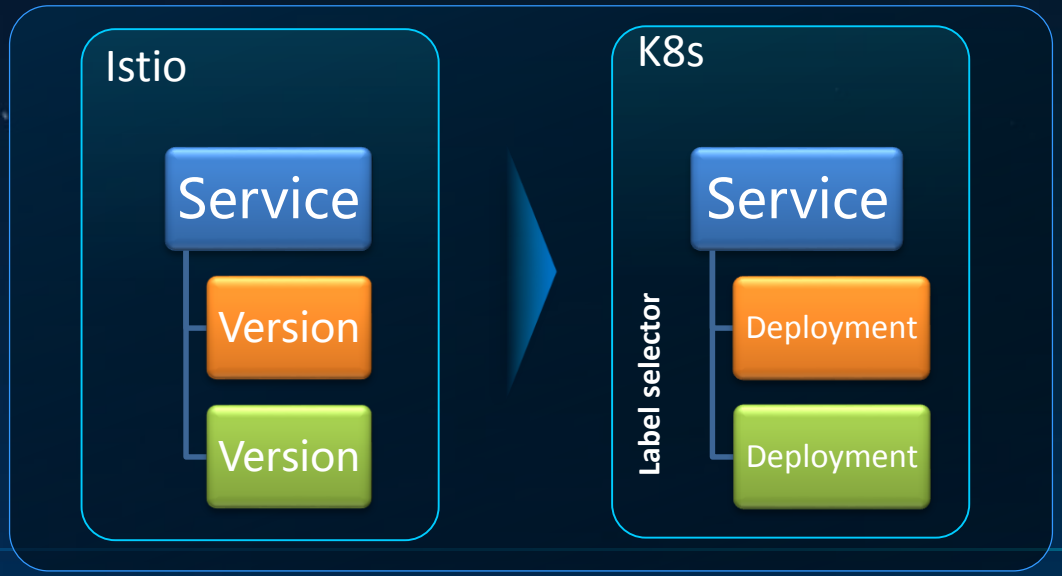
1. Pilot 实现若干服务发现的接口定义
2. Pilot 的Controller List/Watch KubeAPIserver上service、endpoint等资源对象并转换成标准格式。
3. Envoy从Pilot获取xDS，动态更新
4. 当有服务访问时，Envoy在处理Outbound请求时，根据配置的LB策略，选择一个服务实例发起访问。



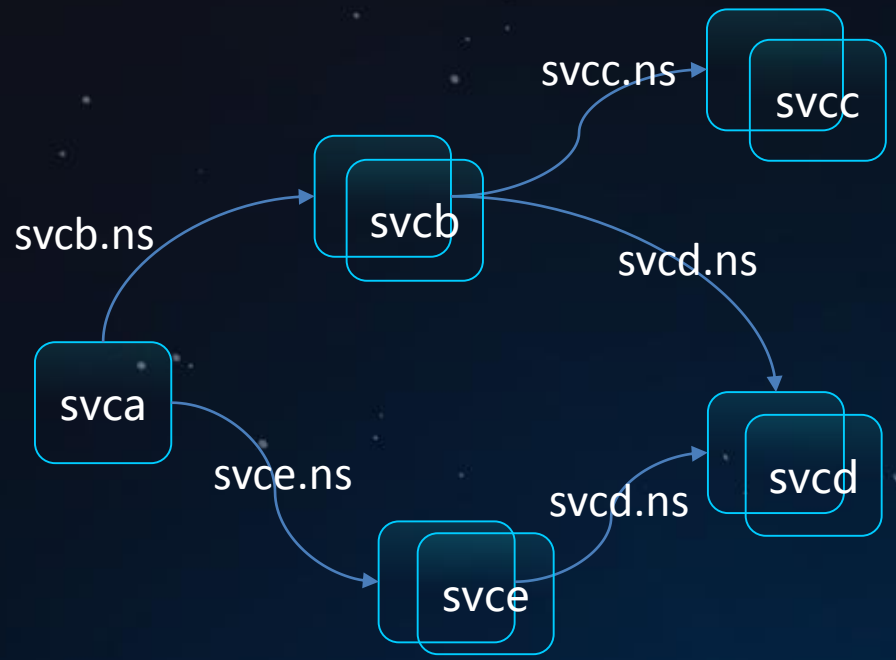

```
type Service struct {
   // Hostname of the service, e.g. "catalog.mystore.com"
   Hostname Hostname `json:"hostname"`
   Address string `json:"address,omitempty"`
   Addresses map[string]string `json:"addresses,omitempty"`
   // Ports is the set of network ports where the service is listening for
connections
   Ports PortList `json:"ports,omitempty"`
   ExternalName Hostname `json:"external"`
   ...
}
```

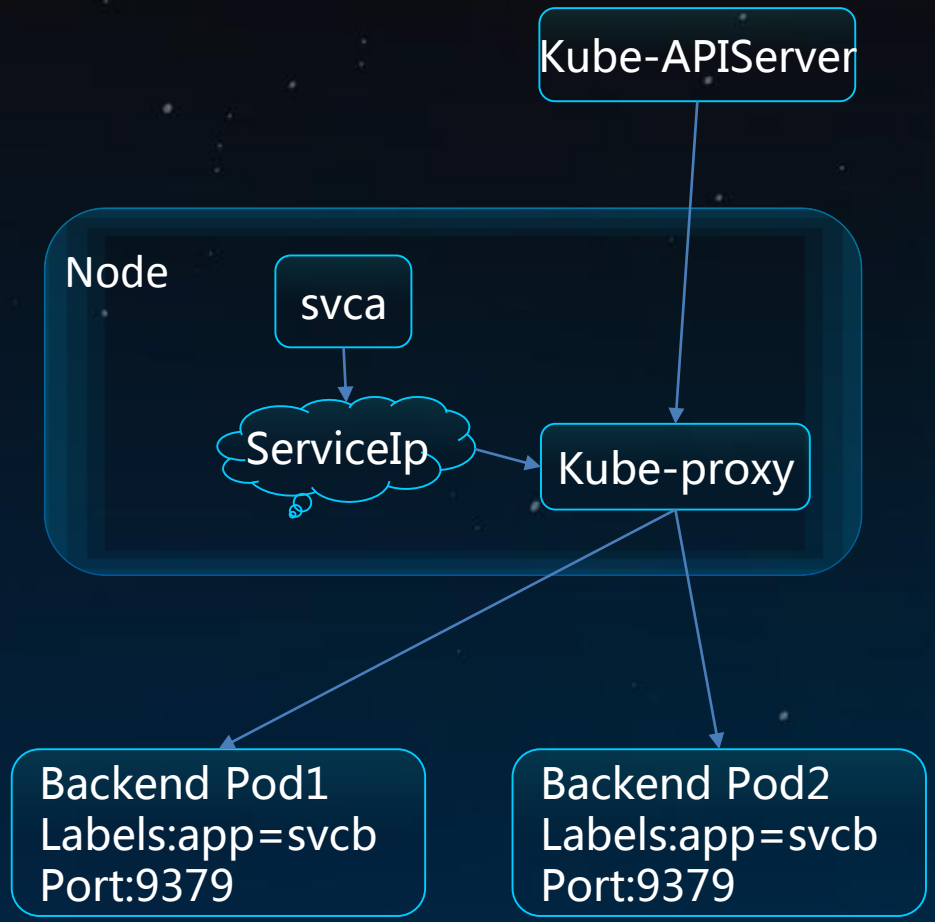
Kubernetes & Istio 服务模型





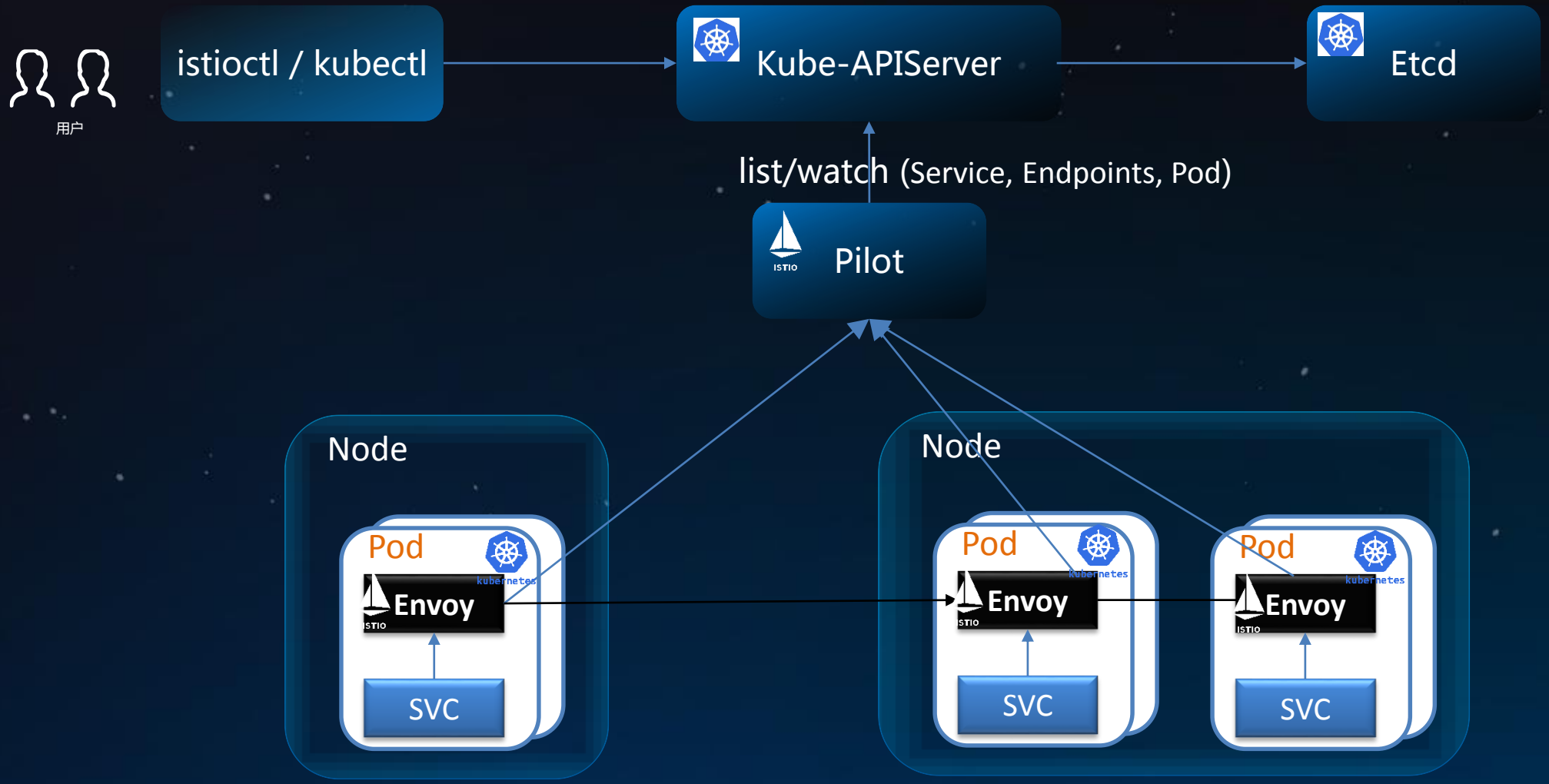
Kuberntes的服务发现





Istio Upon Kubernetes场景

Istio（upon Kubernetes）服务发现和配置



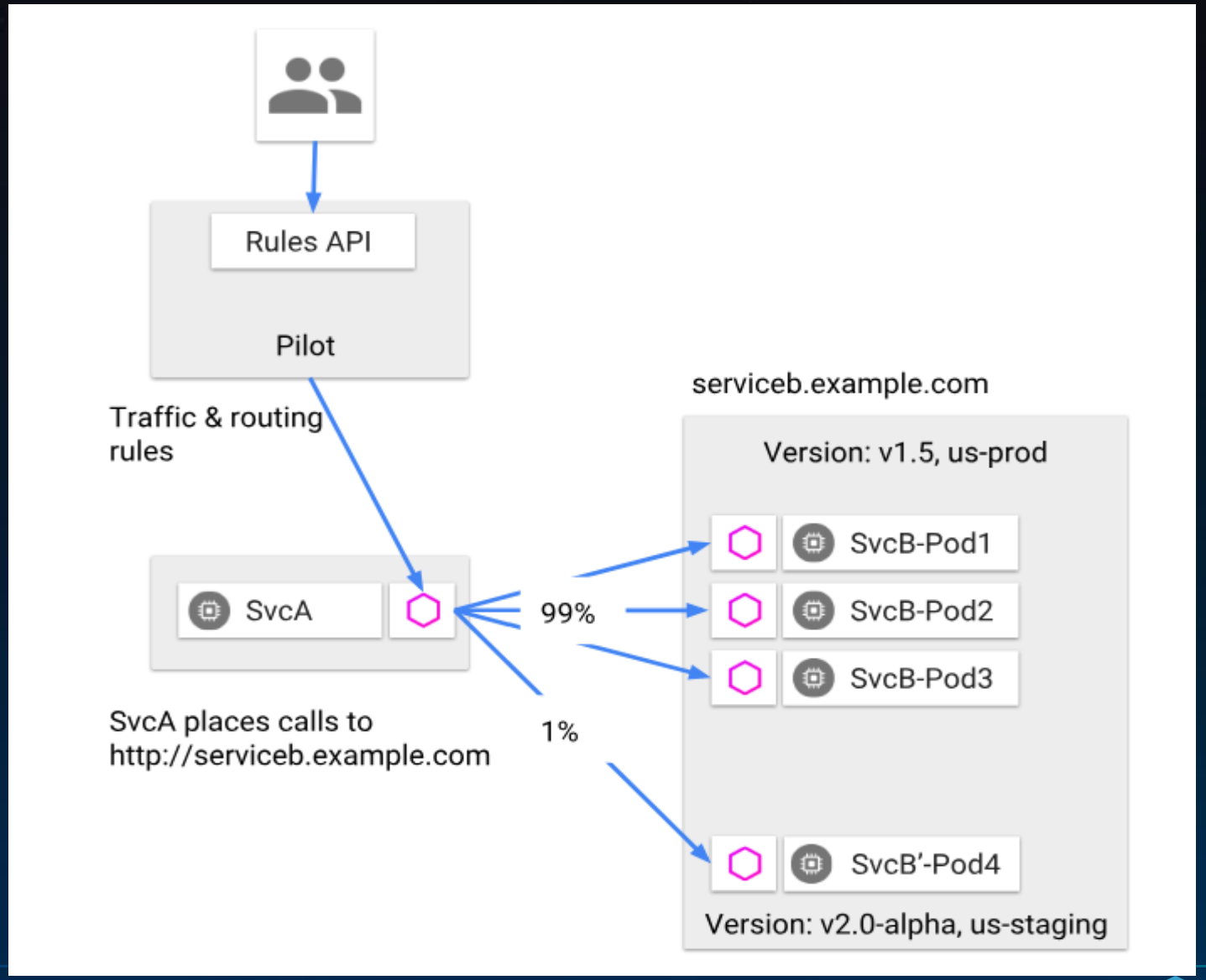
Agenda

HUAWEI

- Istio架构回顾&Pilot介绍
- Istio服务发现
- **Istio服务配置管理**
- Istio服务发现&规则管理与Kubernetes结合

Istio 服务访问规则维护和工作机制

1. 配置：管理员通过Pilot配置治疗规则
2. 下发：Envoy从Pilot获取治理规则
3. 执行：在流量访问的时候执行治理规则



Istio治理规则

HUAWEI

- VirtualService
- DestinationRule
- Gateway
- ServiceEntry
- …

https://istio.io/docs/reference/config/istio.networking.v1alpha3/

Istio流量规则： VirtualService

```
kind: VirtualService
metadata:
  name: reviews
spec:
  hosts:
  - reviews
  http:
  - match:
    - headers:
        Foo:
          exact: bar
    fault:
      delay:
        fixedDelay: 5s
      abort:
        percent: 10
        httpStatus: 400
    route:
    - destination:
        host: reviews
        subset: v2
  - route:
    - destination:
        host: reviews
        subset: v1
```

服务访问路由控制。满足特定条件的请求流到哪里，过程中治理。包括请求重写、重试、故障注入等。

https://preliminary.istio.io/docs/reference/config/istio.networking.v1alpha3/#VirtualService

协议支持：

http	HTTPRoute[]
tls	TLSRoute[]
tcp	TCPRoute[]

HTTP协议流量规则：

Field	Type
match	HTTPMatchRequest[]
route	HTTPRouteDestination[]
redirect	HTTPRedirect
rewrite	HTTPRewrite
timeout	google.protobuf.Duration
retries	HTTPRetry
fault	HTTPFaultInjection
mirror	Destination

Istio流量规则： DestinationRule

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: reviews
spec:
  host: reviews
  trafficPolicy:
    loadBalancer:
      simple: RANDOM
  subsets:
  - name: v1
    labels:
      version: v1
  - name: v2
    labels:
      version: v2
    trafficPolicy:
      loadBalancer:
        simple: ROUND_ROBIN
  - name: v3
    labels:
      version: v3
```

目标服务的策略，包括目标服务的负载均衡，连接池管理等。

https://preliminary.istio.io/docs/reference/config/istio.networking.v1alpha3/#DestinationRule

host	string
trafficPolicy	TrafficPolicy
subsets	Subset[]

loadBalancer	LoadBalancerSettings
connectionPool	ConnectionPoolSettings
outlierDetection	OutlierDetection
tls	TLSSettings
portLevelSettings	TrafficPolicy.PortTrafficPolicy[]

name	string
labels	map<string, string>
trafficPolicy	TrafficPolicy

Istio流量规则： ServiceEntry & Gateway

HUAWEI

ServiceEntry：

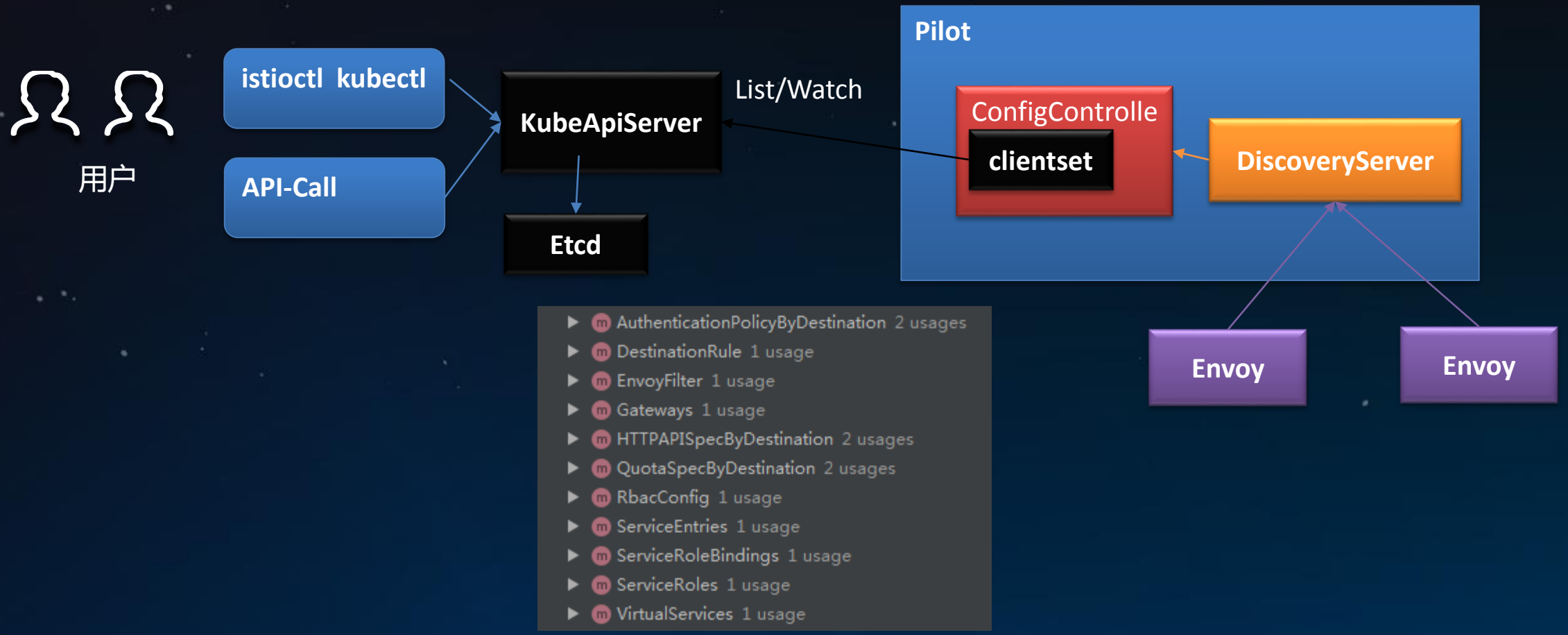
- 功能： Mesh外的服务加入到服务发现中，向Mesh里面的服务一样的被治理
- 机制： 将ServiceEntry描述的服务加入到服务发现中；对这些服务的outbound流量进行拦截，进而进行治理

Gateway：

- 功能：将mesh内的一个服务发布成可供外部访问。
- 机制：在入口处部署一个ingress的Envoy，在其上执行服务治理。

参见后面课程详解。

Istio配置规则维护和下发流程



治理规则定义 Istio VS Envoy

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: details-route
  namespace: bookinfo
spec:
  hosts:
  - details
  http:
  - match:
    - headers:
        User-Agent:
          regex: .*((OS [\d.]+)).*
        cookie:
          exact: aa=bb
    route:
    - destination:
        host: details
        subset: v2
  - route:
    - destination:
        host: details
        subset: v1
```

Istio 规则

```
{
        "name": "details.bookinfo.svc.cluster.local:80",
        "domains": [
         "details.bookinfo.svc.cluster.local"
        ],
        "routes": [
         {
          "match": {
           "prefix": "/",
           "headers": [
            {
             "name": "User-Agent",
             "regex_match": ".*((OS [\\d.]+)).*"
            },
            {
             "name": "cookie",
             "exact_match": "aa=bb"
            }
           ]
          },
          "route": {"cluster": "outbound|80|v2|details.bookinfo.svc.cluster.local"...},
          "decorator": {"operation": "details.bookinfo.svc.cluster.local:80/*"...},
          "per_filter_config": {...}
         },
         {
          "match": {
           "prefix": "/"
          },
          "route": {
           "cluster": "outbound|80|v1|details.bookinfo.svc.cluster.local",
           "timeout": "0s",
           "max_grpc_timeout": "0s"
          },
          "decorator": {"operation": "details.bookinfo.svc.cluster.local:80/*"...},
          "per_filter_config": {...}
         }
        ]
       }
```

Envoy规则

Istio治理能力执行位置

治理能力	治理执行	
	服务发起方	服务提供方
路由管理	●	
断路器	●	
负载均衡	●	
调用链分析	●	●
服务认证	●	●
遥测数据	●	●
重试	●	
重写	●	
重定向	●	
鉴权		●
请求限流		●
…		

Agenda

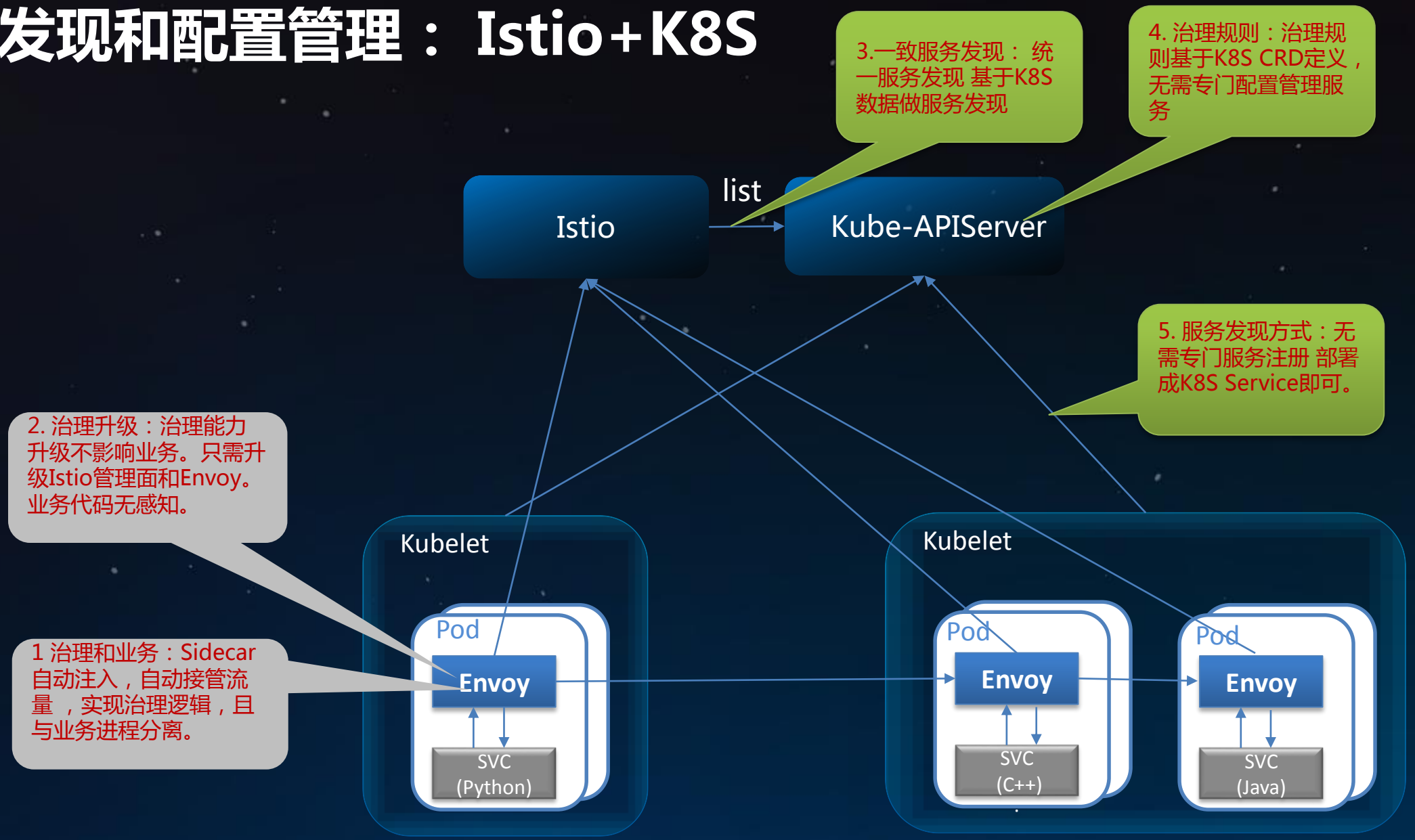
HUAWEI

- Istio架构回顾&Pilot介绍
- Istio服务发现
- Istio服务配置管理
- Istio服务发现&规则管理与Kubernetes结合

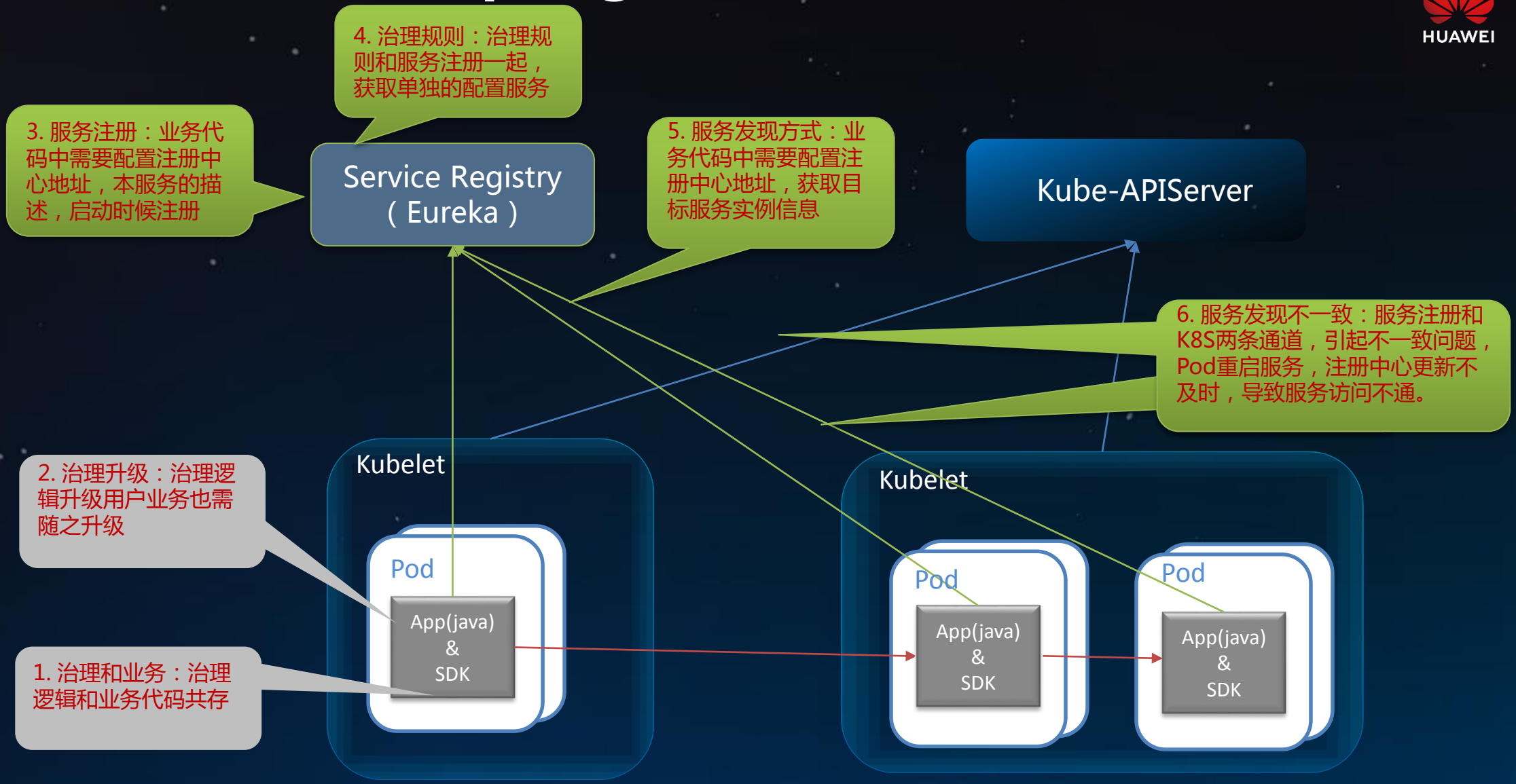
Kubernetes & Istio 结合

对于云原生应用，采用kubernetes构建微服务部署和集群管理能力，采用Istio构建服务治理能力，将逐渐成为应用微服务转型的标准配置。

服务发现和配置管理： Istio+K8S



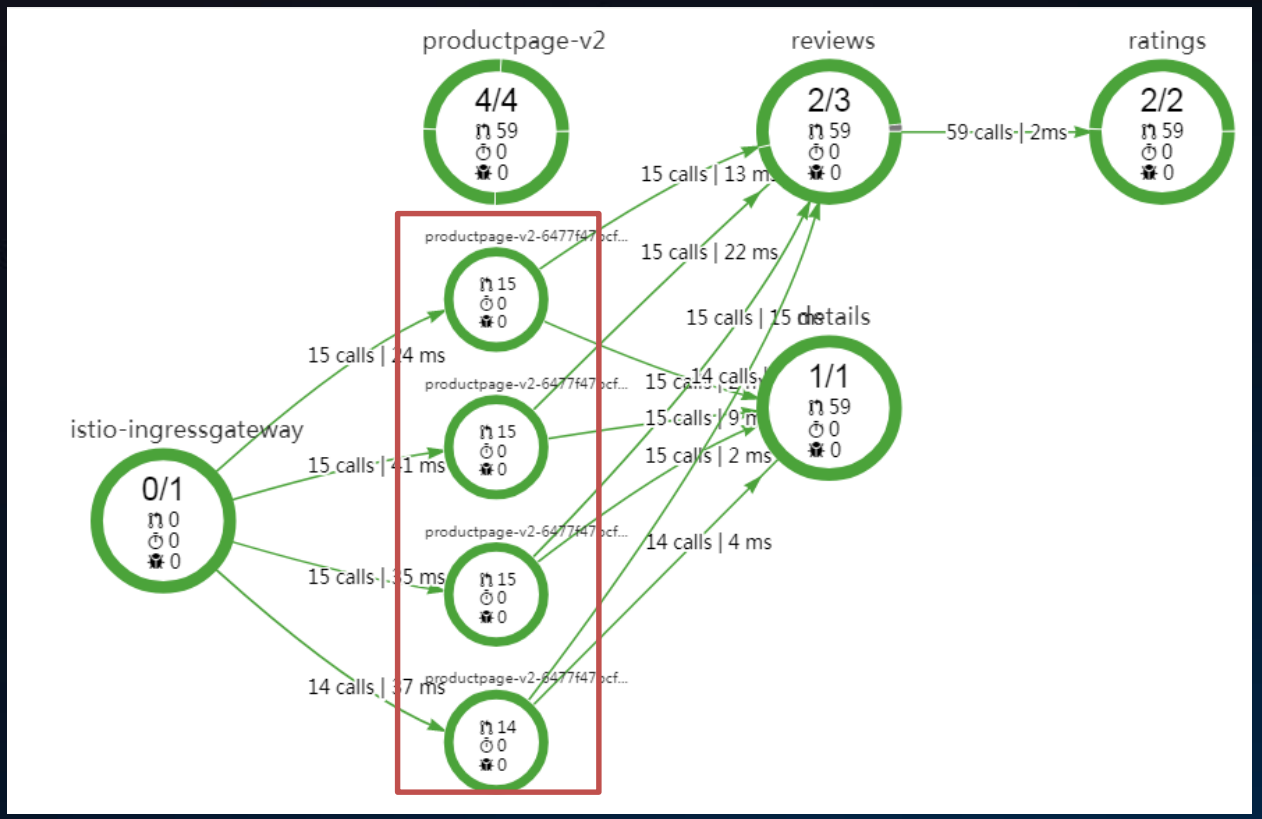
服务发现和配置管理： Spring Cloud+K8S

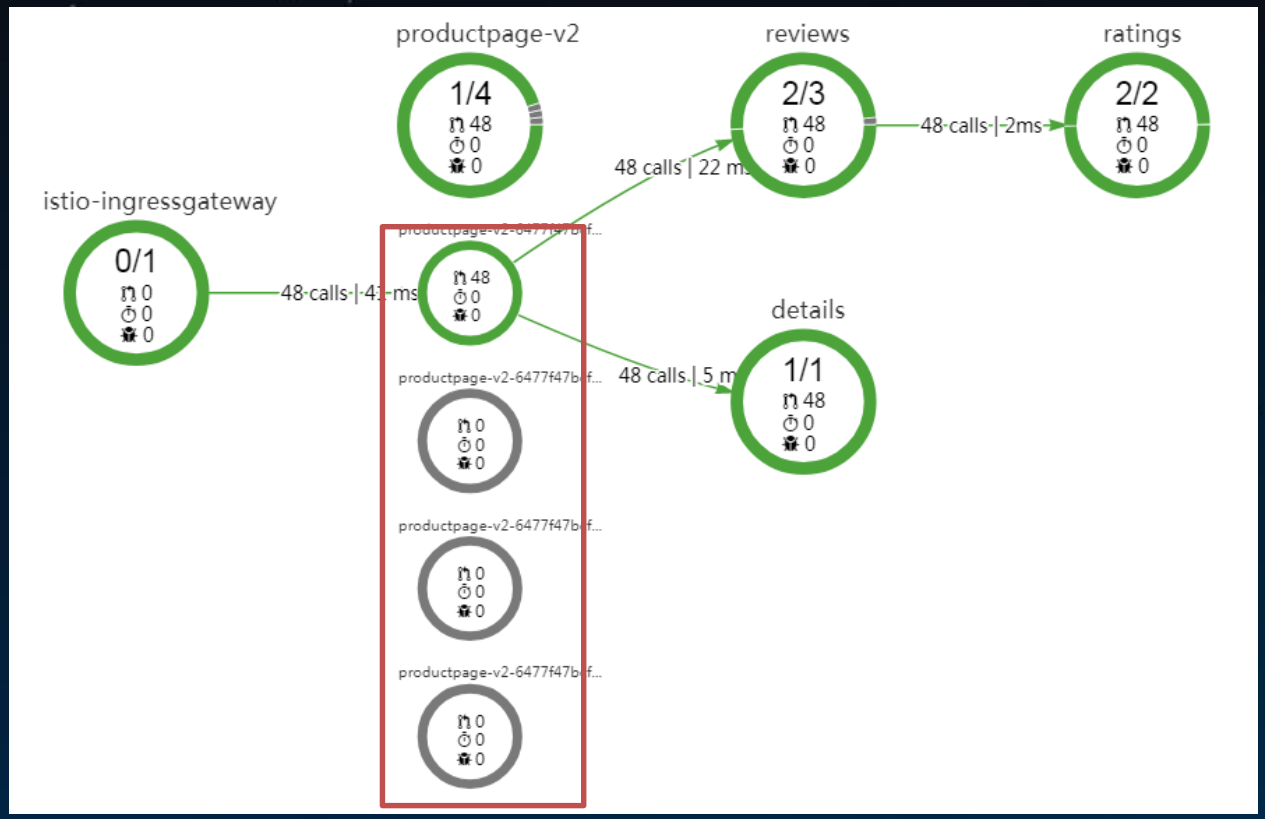


Show Case

https://console.huaweicloud.com/istio

回话保持





HUAWEI

HUAWEI

Thank You

直播 每周四 晚20:00

CloudNativeLives

Kubernetes Docker Istio

附： Istio治理能力执行位置

治理能力	治理执行	
	服务发起方	服务提供方
路由管理	●	
断路器	●	
负载均衡	●	
调用链分析	●	●
服务认证	●	●
遥测数据	●	●
重试	●	
重写	●	
重定向	●	
鉴权		●
请求限流		●
…		