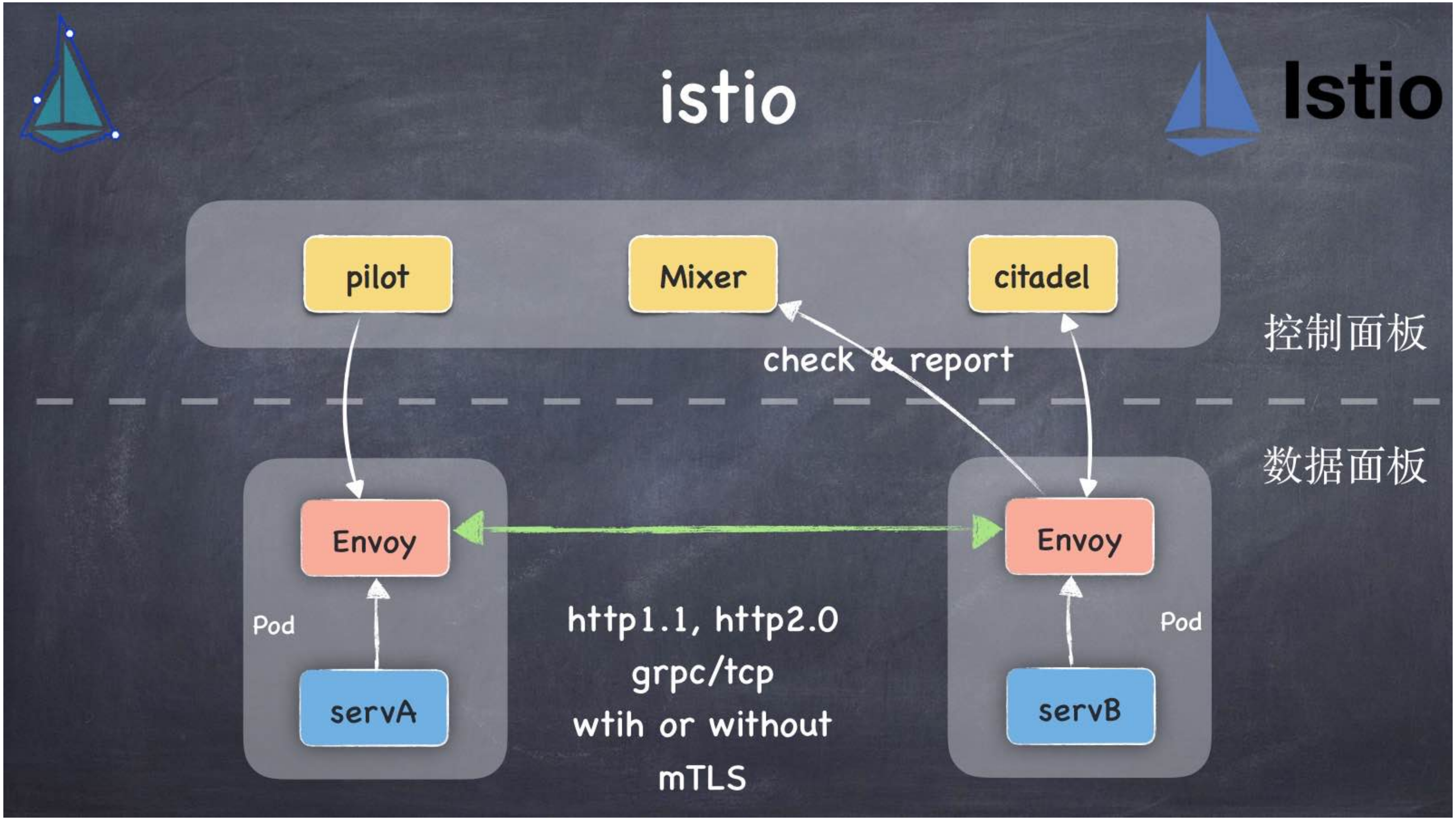
istio那些事儿

- xiaorui.cc

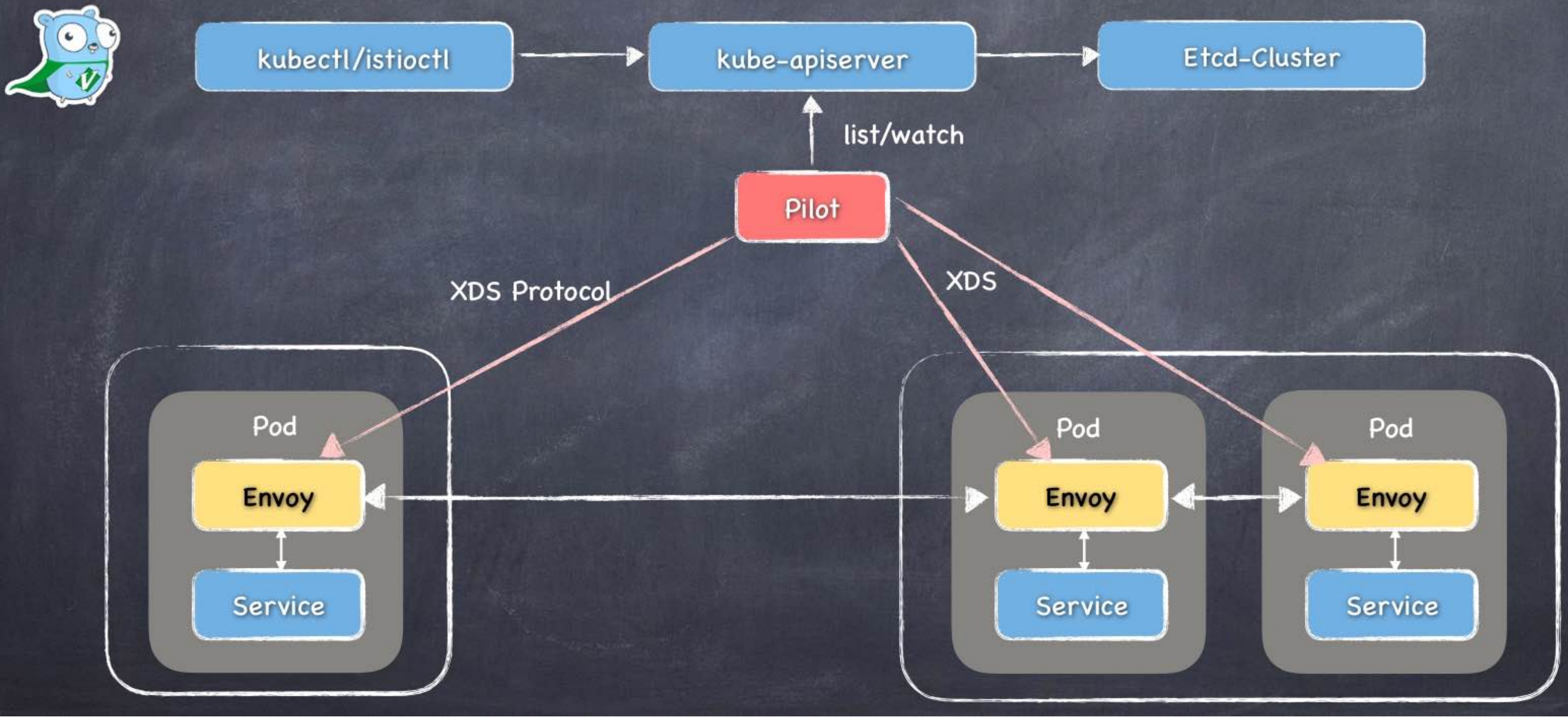
istio


Istio
pilot
Mixer
citadel
控制面板
check & report
数据面板
Envoy
Envoy
Pod
Pod
servA
servB
http1.1, http2.0
grpc/tcp
wtih or without
mTLS

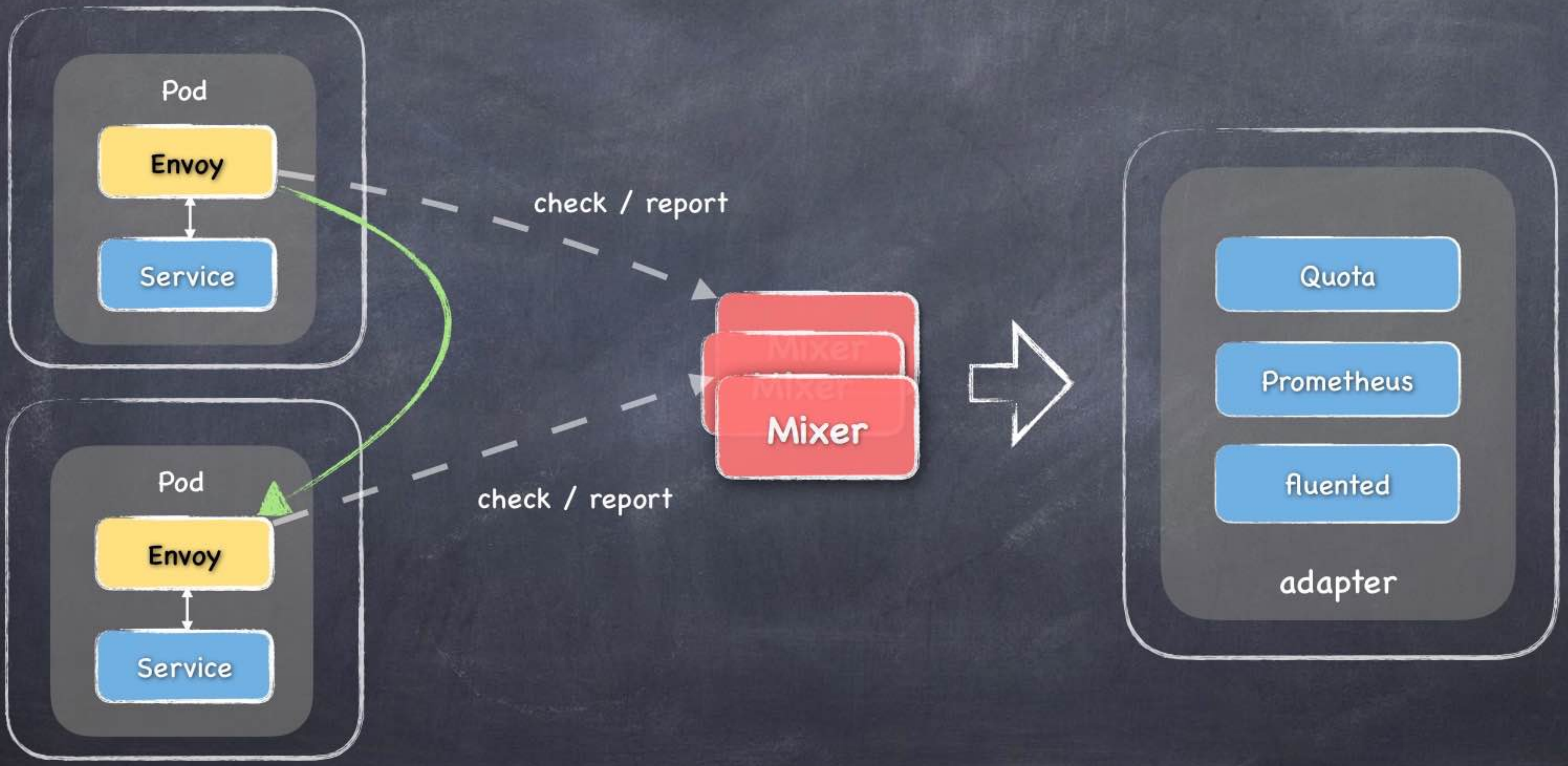
istio 组件

- Pilot-x 服务发现
- Mixer
  - istio-policy 检查权限，配额
  - istio-telemetry 收集调用metrics
- citadel 证书
- galley 校验正确性
- ingressgateway 网关
- jaeger
- prometheus
- zipkin
- fluented
- ...

istio pilot


kubectl/istioctl
kube-apiserver
Etcd-Cluster
list/watch
Pilot
XDS Protocol
XDS
Pod
Envoy
Service
Pod
Envoy
Service
Pod
Envoy
Service

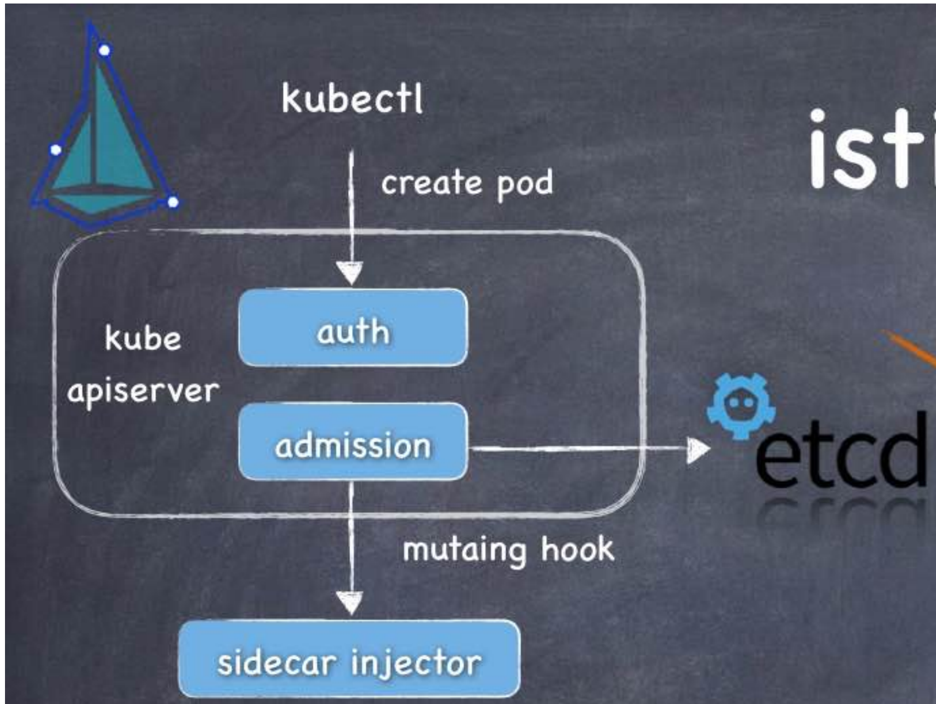
istio mixer adapter



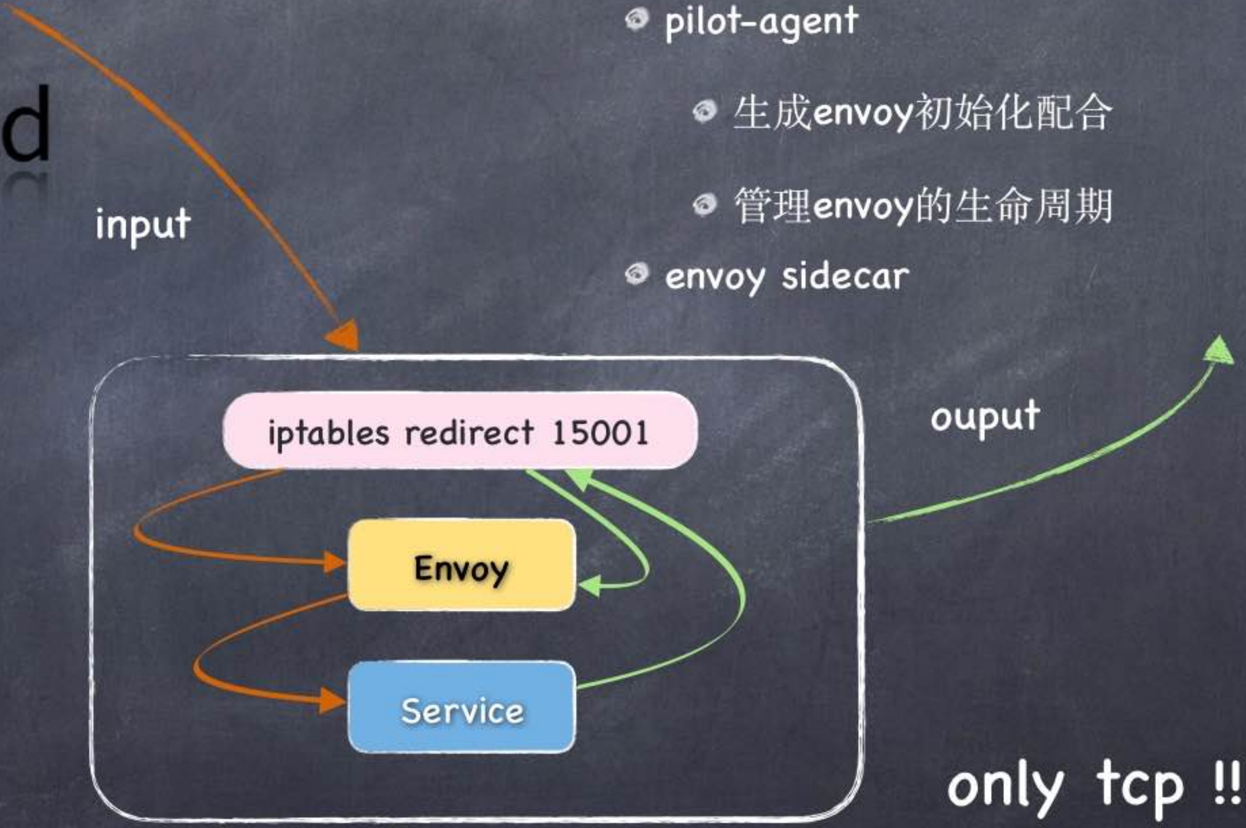
istio 流量治理流程

- 控制面板流程：
  - 管理员通过kubectl/istioctl或者API创建流量规则
  - Pilot从kubernets apiserver获取数据，并规则转换为Envoy xds
  - Pilot将xds推送给envoy
- 数据面板流程：
  - Envoy动态载入xds配置，并初始化新的资源监听
  - Envoy 拦截 pod上的本地容器的Inbound和Outbound流量
  - 在流量经过Envoy时执行对应的流量规则，执行流量治理

istio inject



- pilot-proxy
  - pilot-agent
    - 生成envoy初始化配合
    - 管理envoy的生命周期
  - envoy sidecar



- pilot-init
  - ./prepare_proxy.sh -p 15001 -u 1337
  - init exit !

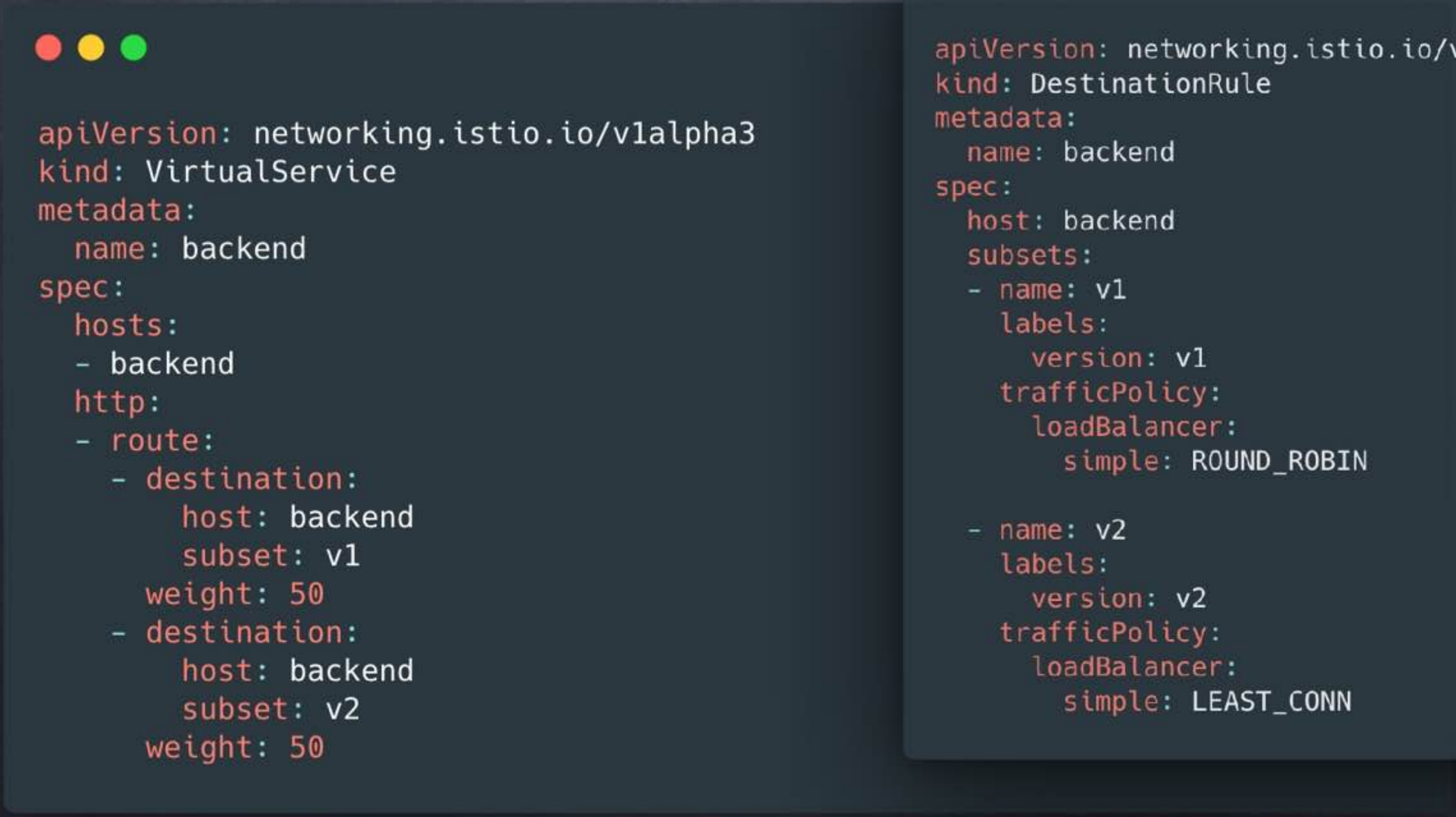
only tcp !!!

istio crd resource

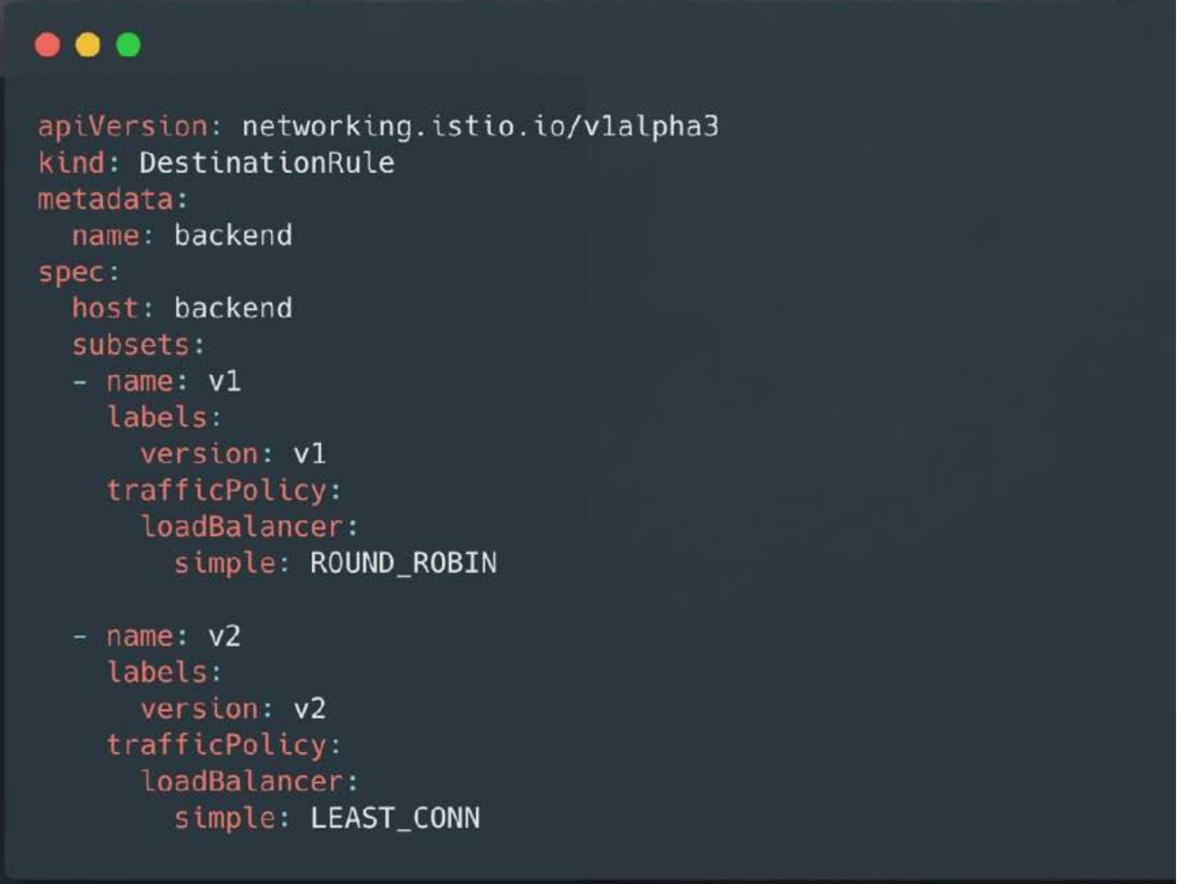
- VirtualService
  - 定义路由规则
  - 流量管理
- DestinationRule
  - 定义可路由的目的服务的子集
  - 目的服务的策略 (断路器/负载均衡/TLS ...)
- ServiceEntry
  - 定义网格之外的资源
- Gateway
  - 为网格配置网关
- Sidecar
  - 服务隔离
- EnvoyFilter
  - 集成Lua可自定义envoy的过滤链规则

一个实例

Istio

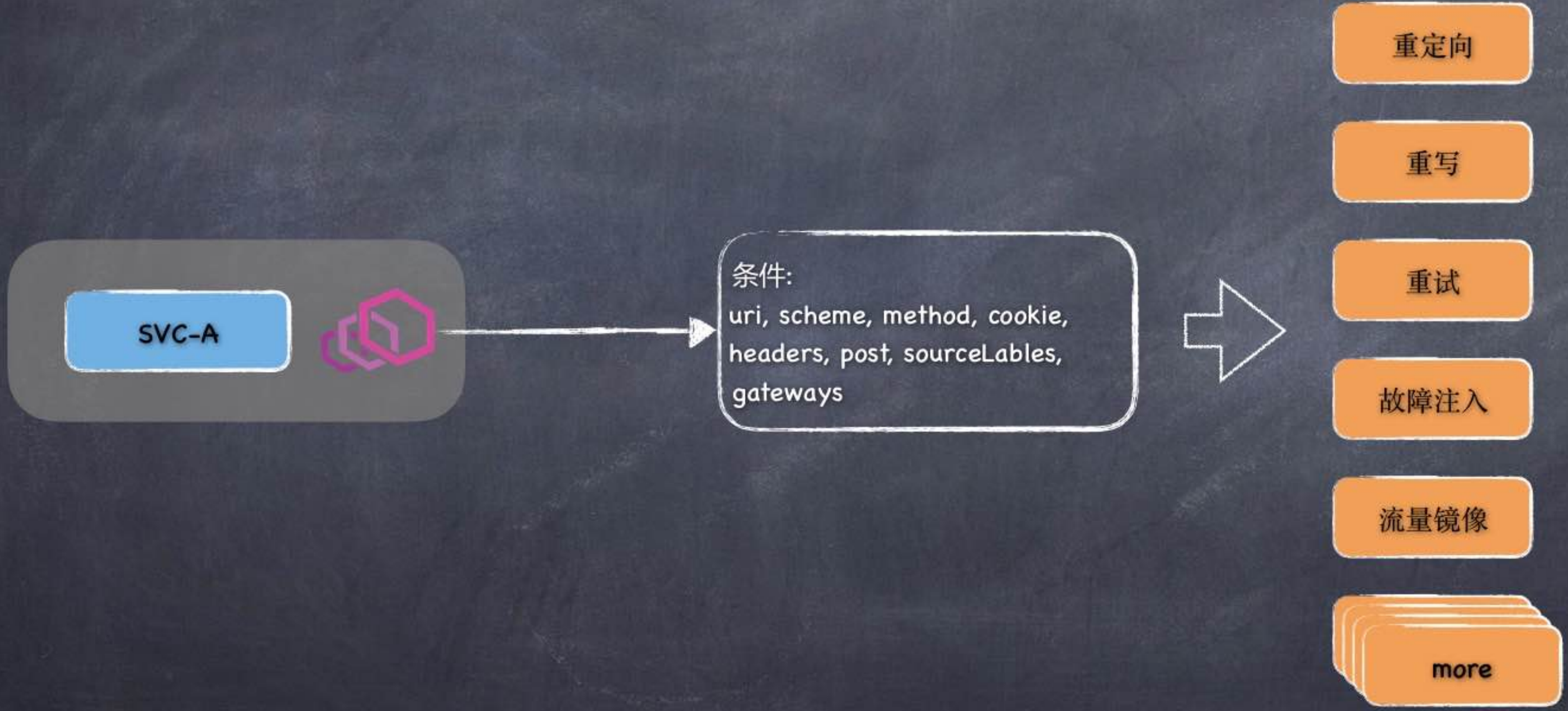


```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: backend
spec:
  hosts:
  - backend
  http:
  - route:
    - destination:
        host: backend
        subset: v1
      weight: 50
    - destination:
        host: backend
        subset: v2
      weight: 50
```

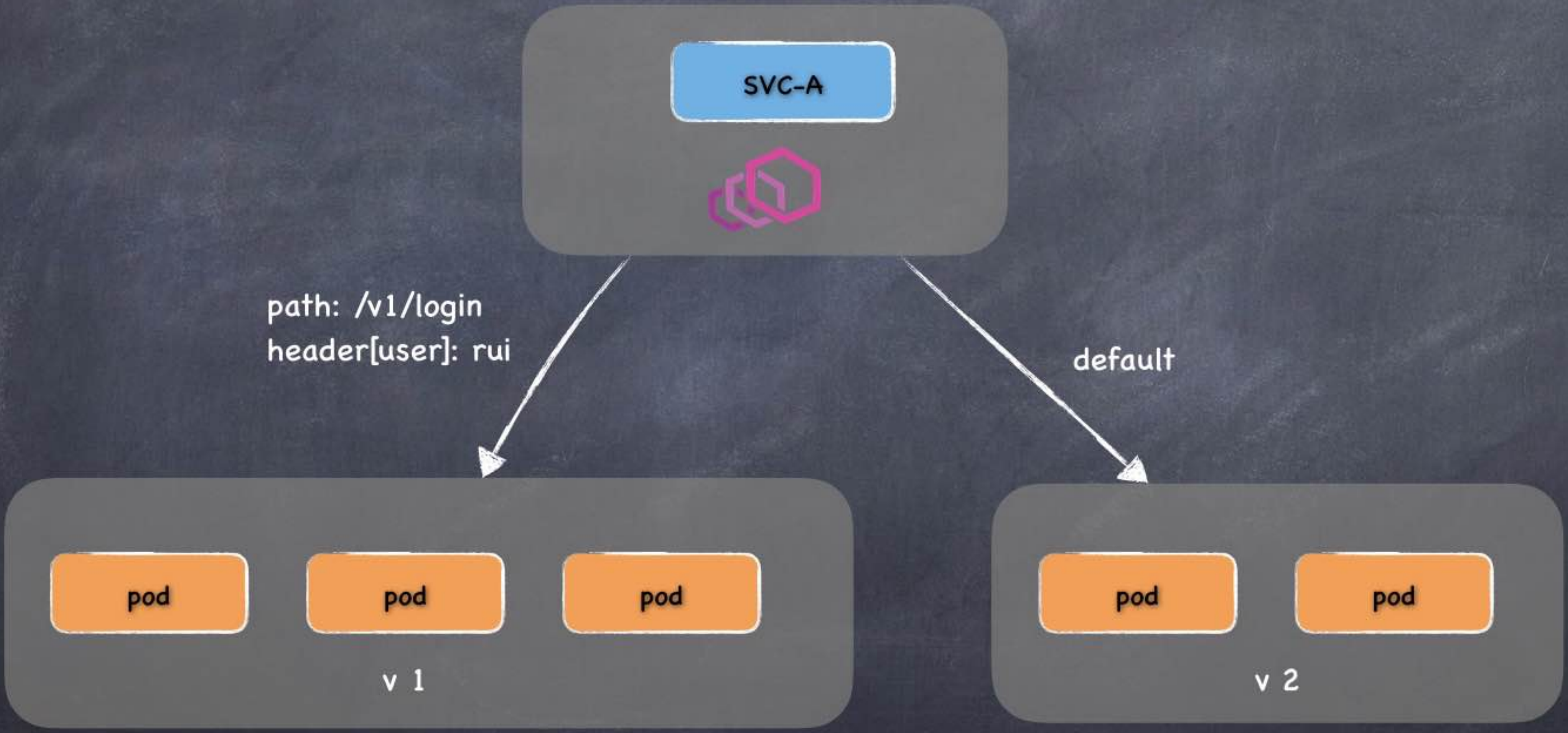


```yaml
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: backend
spec:
  host: backend
  subsets:
  - name: v1
    labels:
      version: v1
    trafficPolicy:
      loadBalancer:
        simple: ROUND_ROBIN

  - name: v2
    labels:
      version: v2
    trafficPolicy:
      loadBalancer:
        simple: LEAST_CONN
```

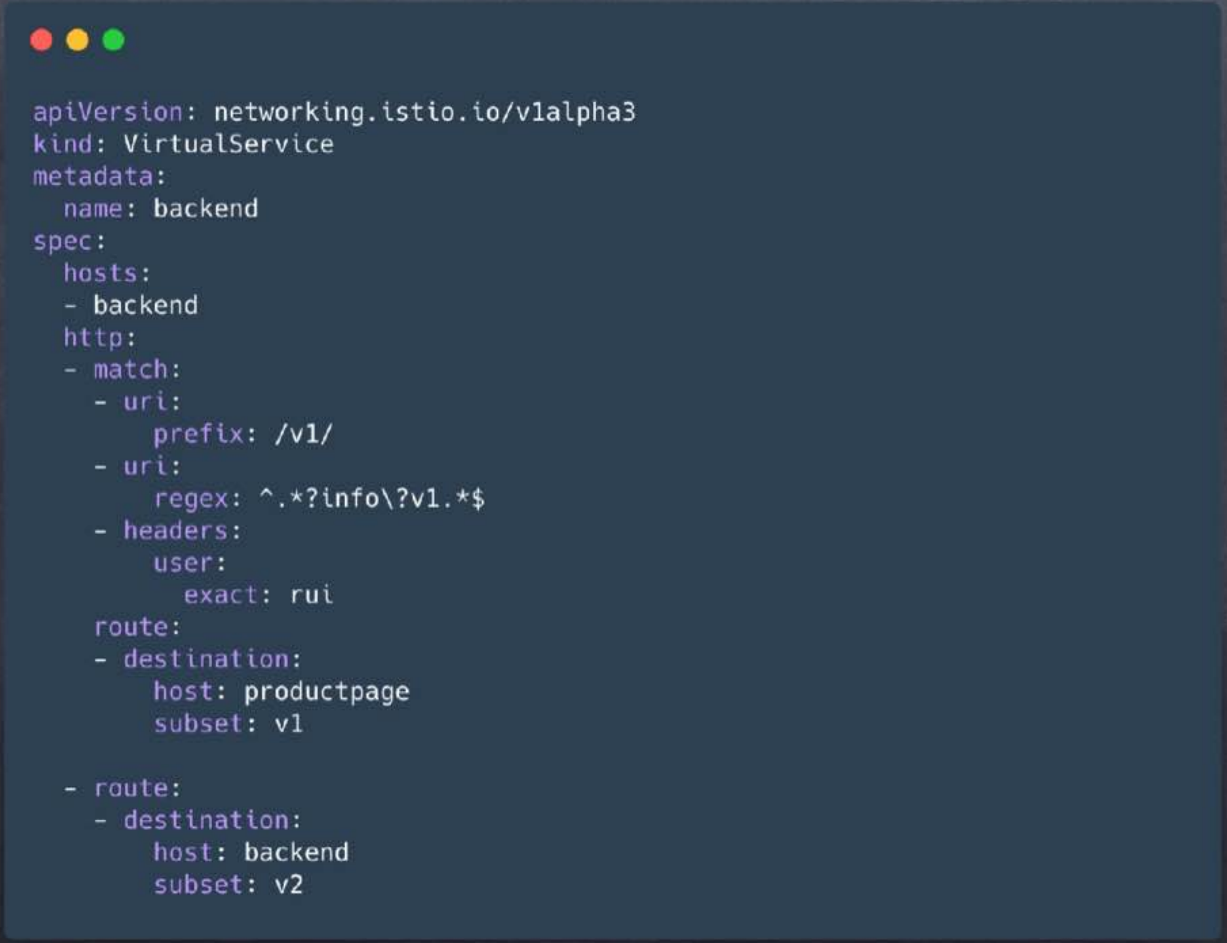
rules


SVC-A
条件:
uri, scheme, method, cookie,
headers, post, sourceLables,
gateways
重定向
重写
重试
故障注入
流量镜像
more

规则匹配

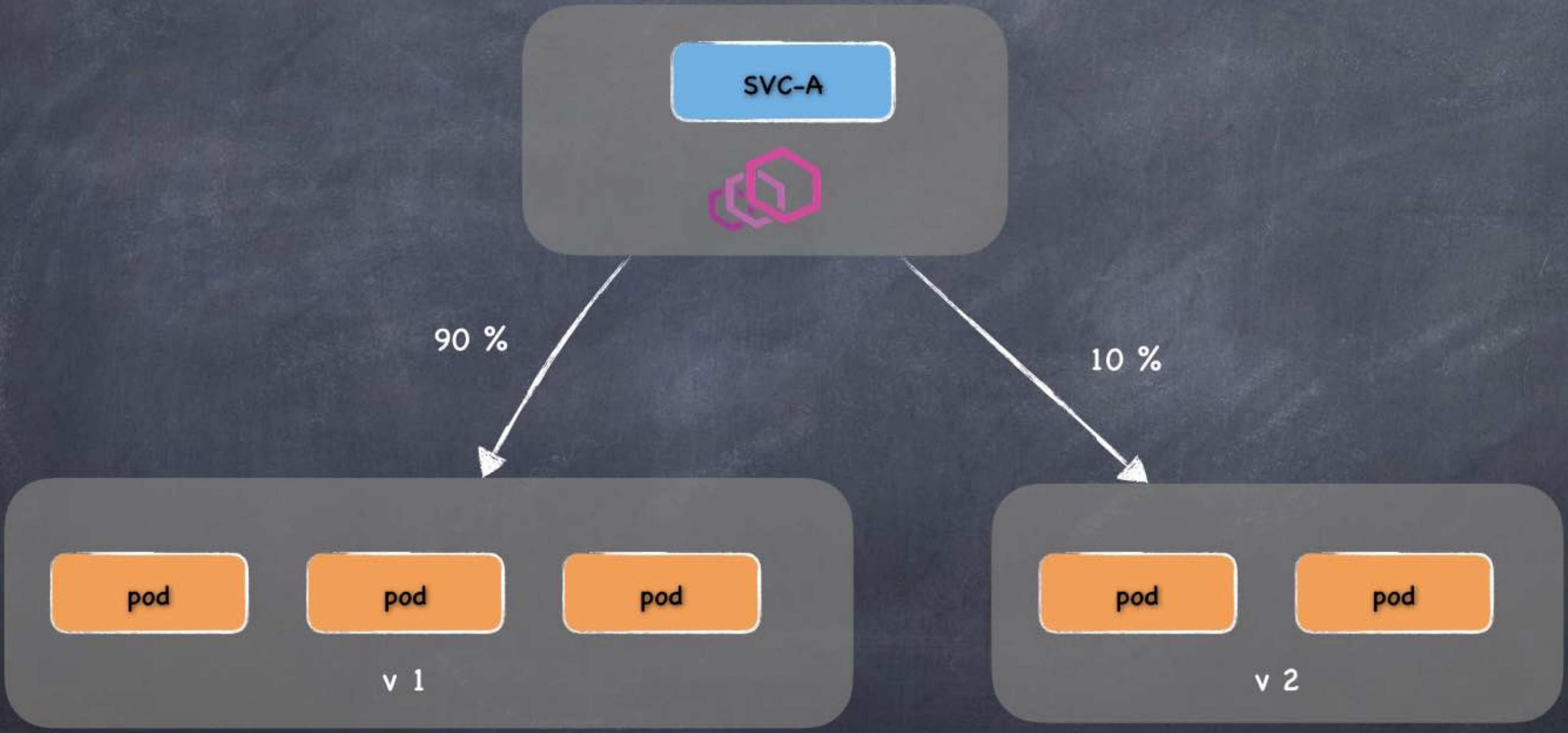

SVC-A
path: /v1/login
header[user]: rui
default
pod
pod
pod
v 1
pod
pod
v 2

规则匹配



```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: backend
spec:
  hosts:
  - backend
  http:
  - match:
    - uri:
        prefix: /v1/
    - uri:
        regex: ^.*?info\?v1.*$
    - headers:
        user:
          exact: rui
    route:
    - destination:
        host: productpage
        subset: v1

  - route:
    - destination:
        host: backend
        subset: v2
```

权重

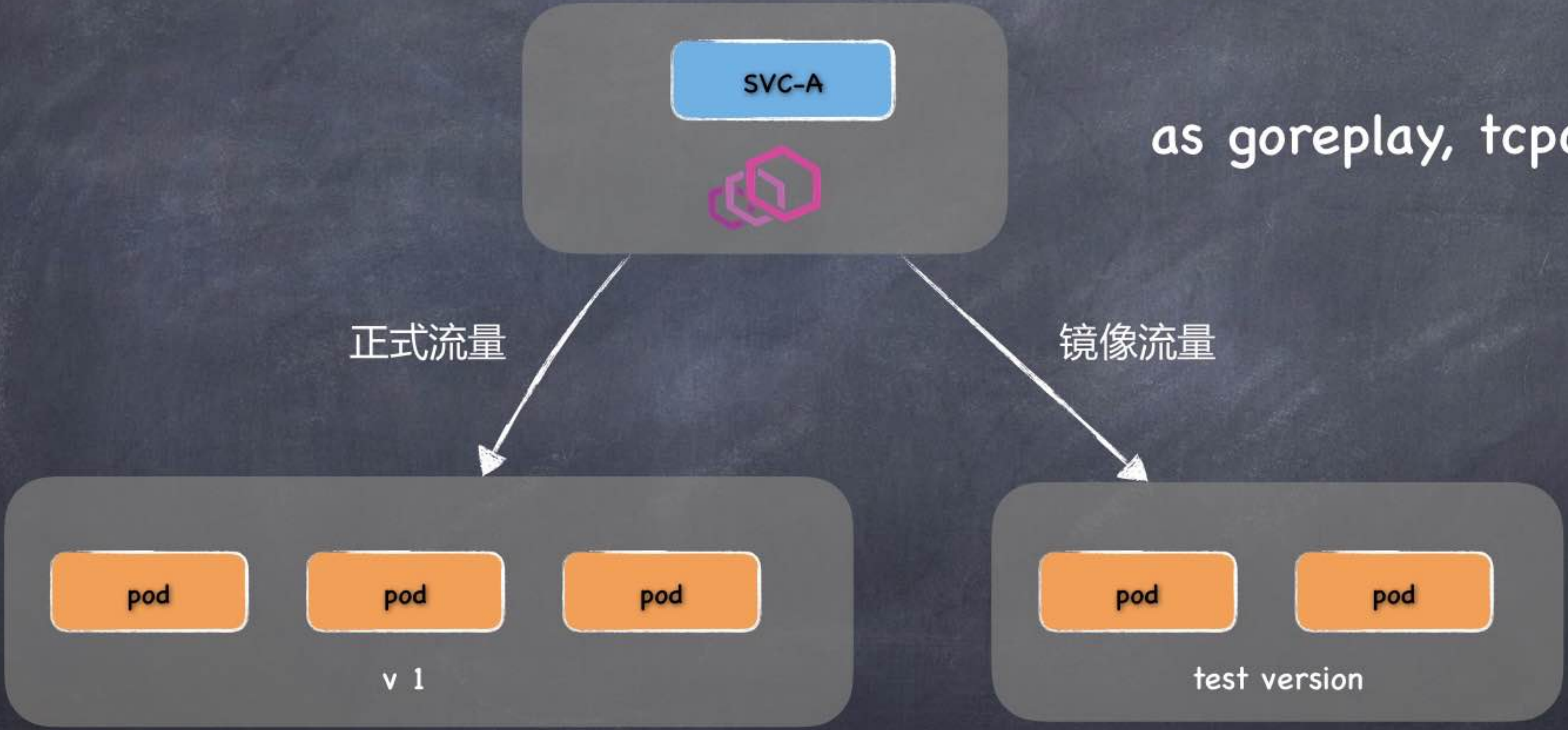


权重

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: backend
spec:
  hosts:
  - backend
  http:
  - route:
    - destination:
        host: backend
        subset: v1
      weight: 90
    - destination:
        host: backend
        subset: v2
      weight: 10
```

镜像

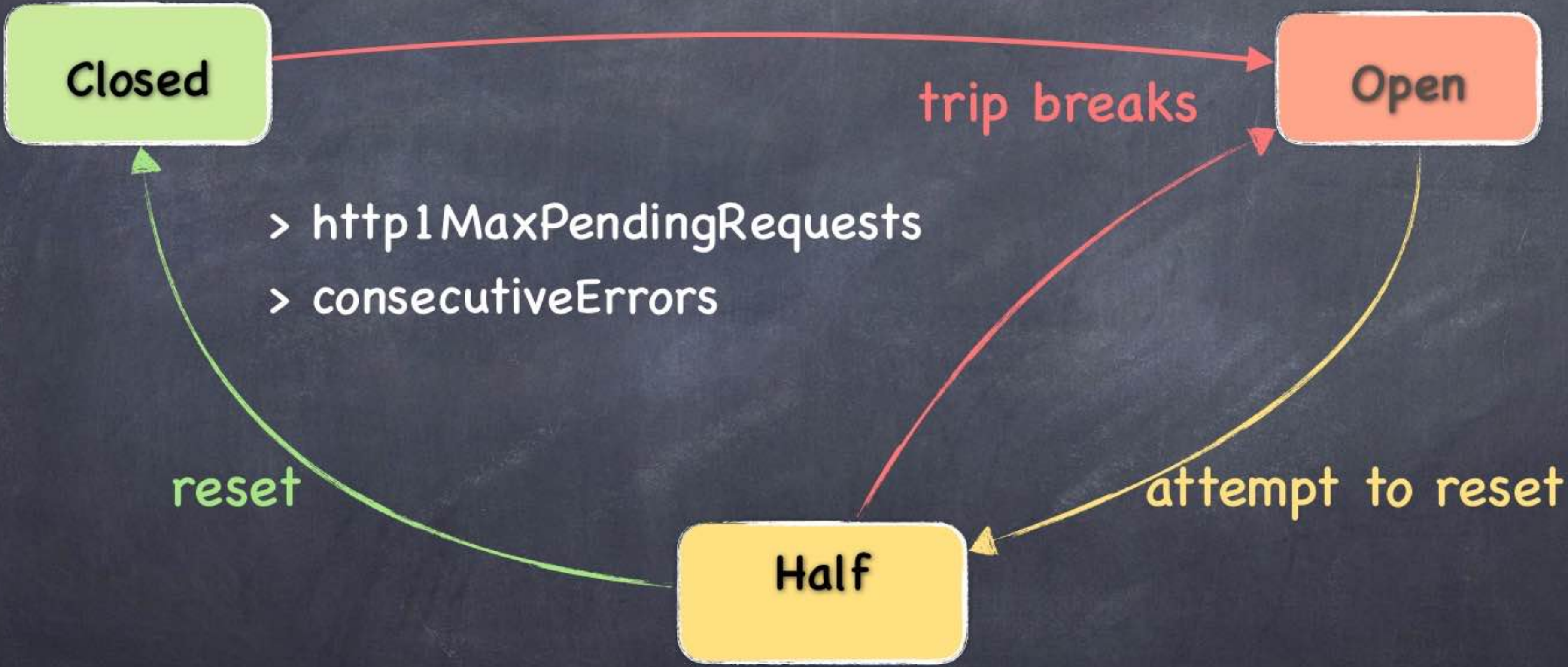
as goreplay, tcpcopy



镜像

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: backend
spec:
  hosts:
  - backend
  http:
  - route:
      - destination:
          host: backend
          subset: v1
        weight: 100
    mirror:
        host: backend
        subset: test
```

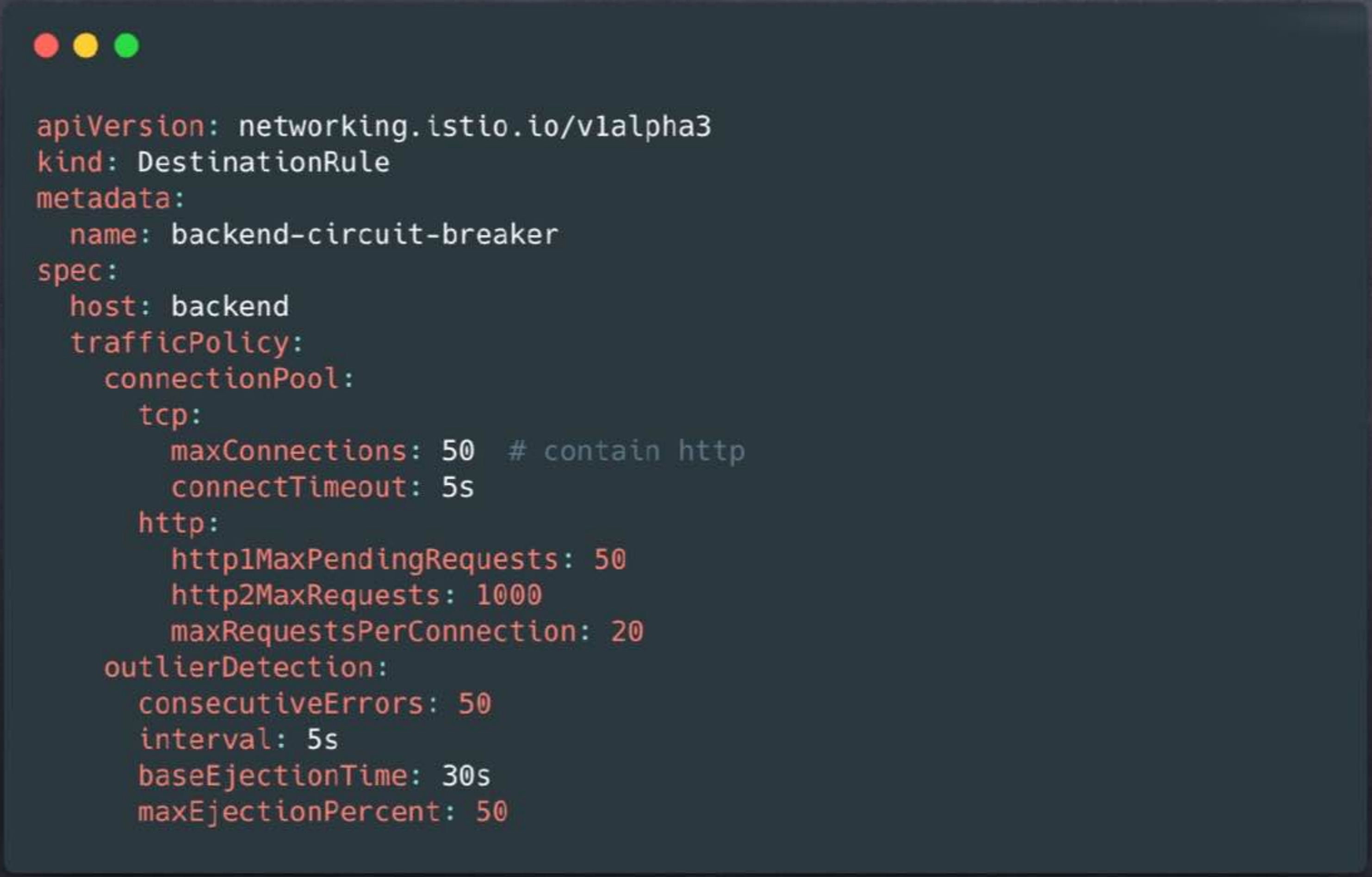
熔断



> http1MaxPendingRequests

> consecutiveErrors

熔断


```yaml
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: backend-circuit-breaker
spec:
  host: backend
  trafficPolicy:
    connectionPool:
      tcp:
        maxConnections: 50  # contain http
        connectTimeout: 5s
      http:
        http1MaxPendingRequests: 50
        http2MaxRequests: 1000
        maxRequestsPerConnection: 20
    outlierDetection:
      consecutiveErrors: 50
      interval: 5s
      baseEjectionTime: 30s
      maxEjectionPercent: 50
```

other

- ratelimit
- timeout
- session affinity
- ...

- retry
- rewrite
- redirect
- ...

- fault inject
  - delay
  - abort
- denier
  - attribute
  - iplist

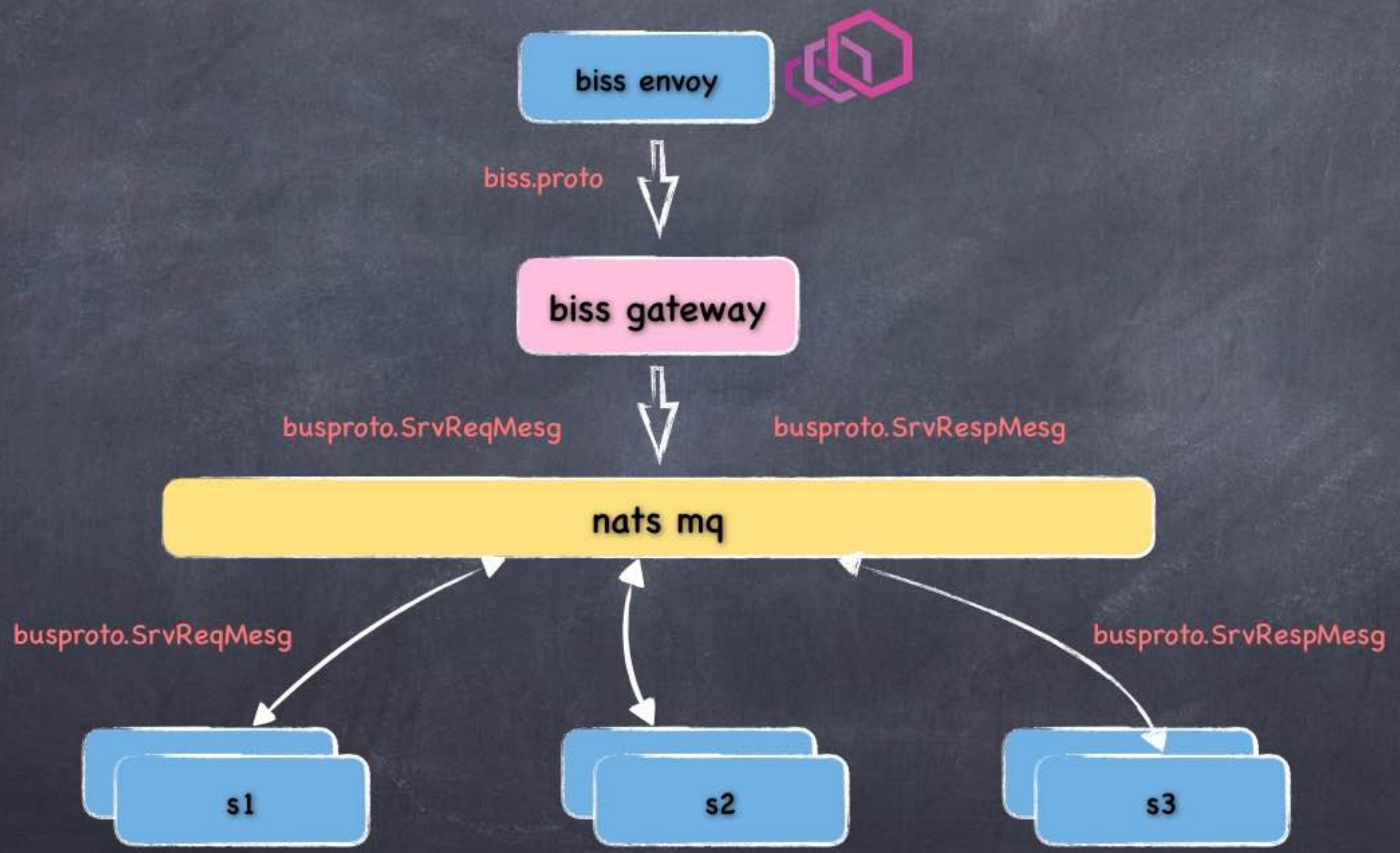
例子

https://github.com/rfyiamcool/istio-http-lb

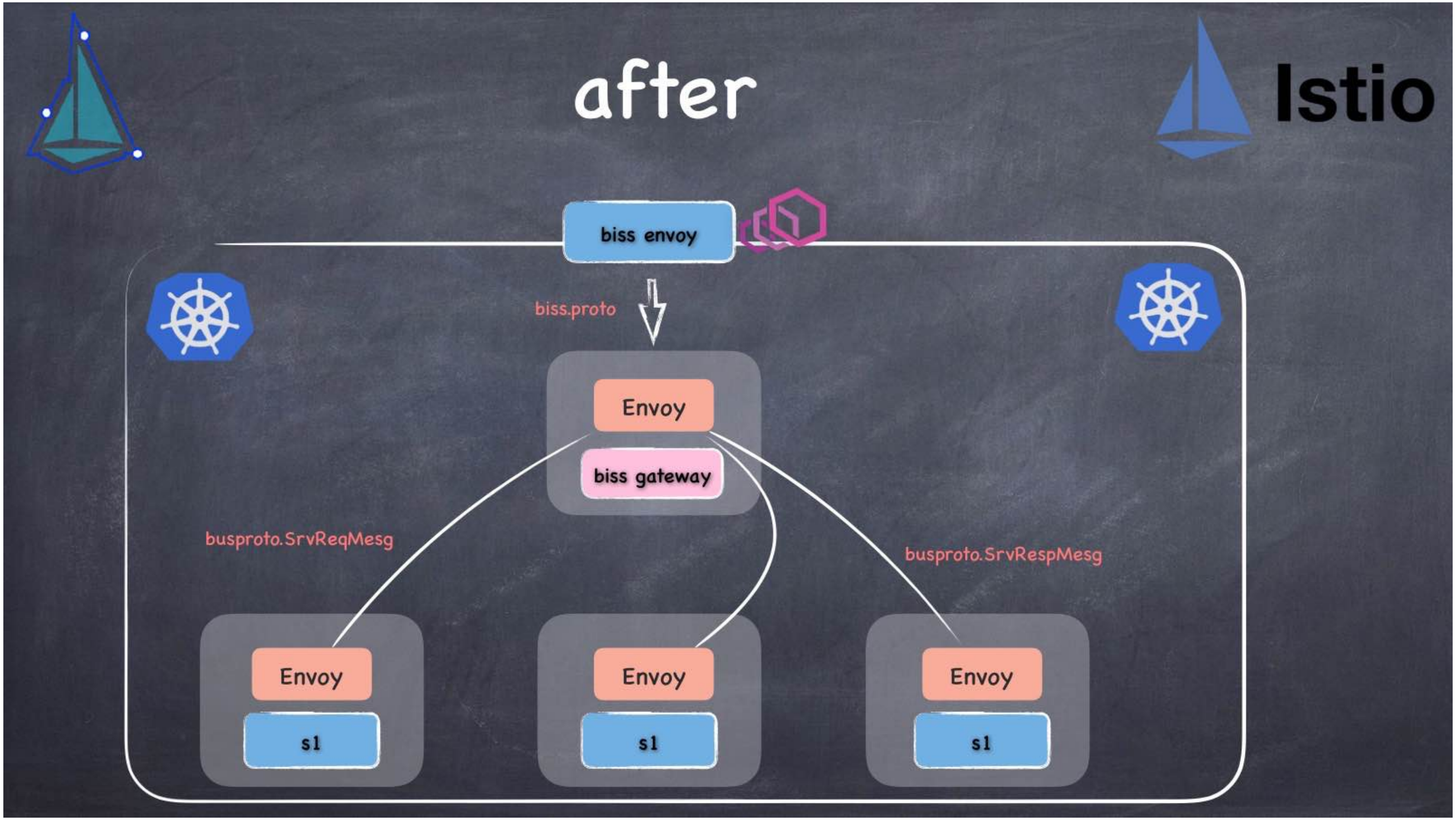
- k8s
- istio
- 涵盖大多数功能测试

before

Istio



after



Istio
biss envoy
biss.proto
Envoy
biss gateway
busproto.SrvReqMesg
busproto.SrvRespMesg
Envoy
s1
Envoy
s1
Envoy
s1

平滑迁移

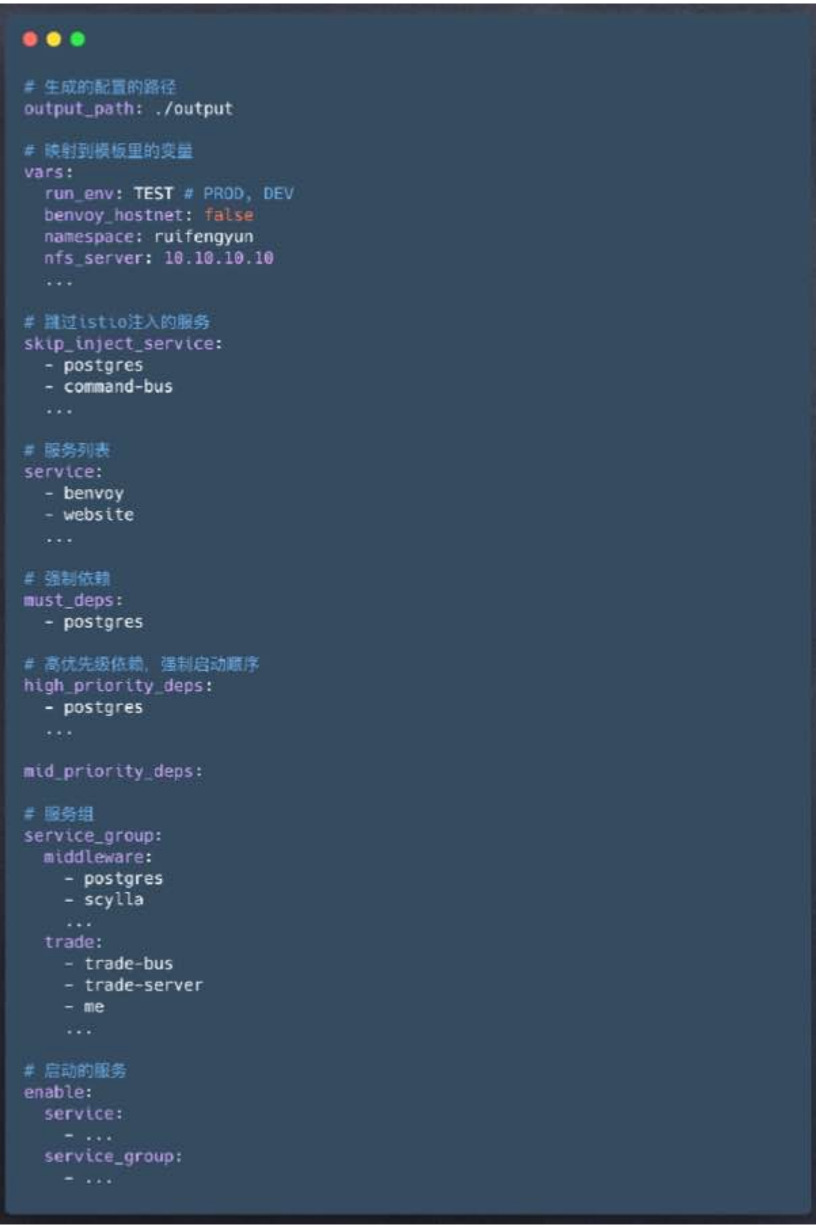
业务只需要更换srvFrame库包即可 !!!

- 自定义grpc编码
- grpc.UnknownServiceHandler
- grpc invoke raw
- grpc reflect & protobuf reflect

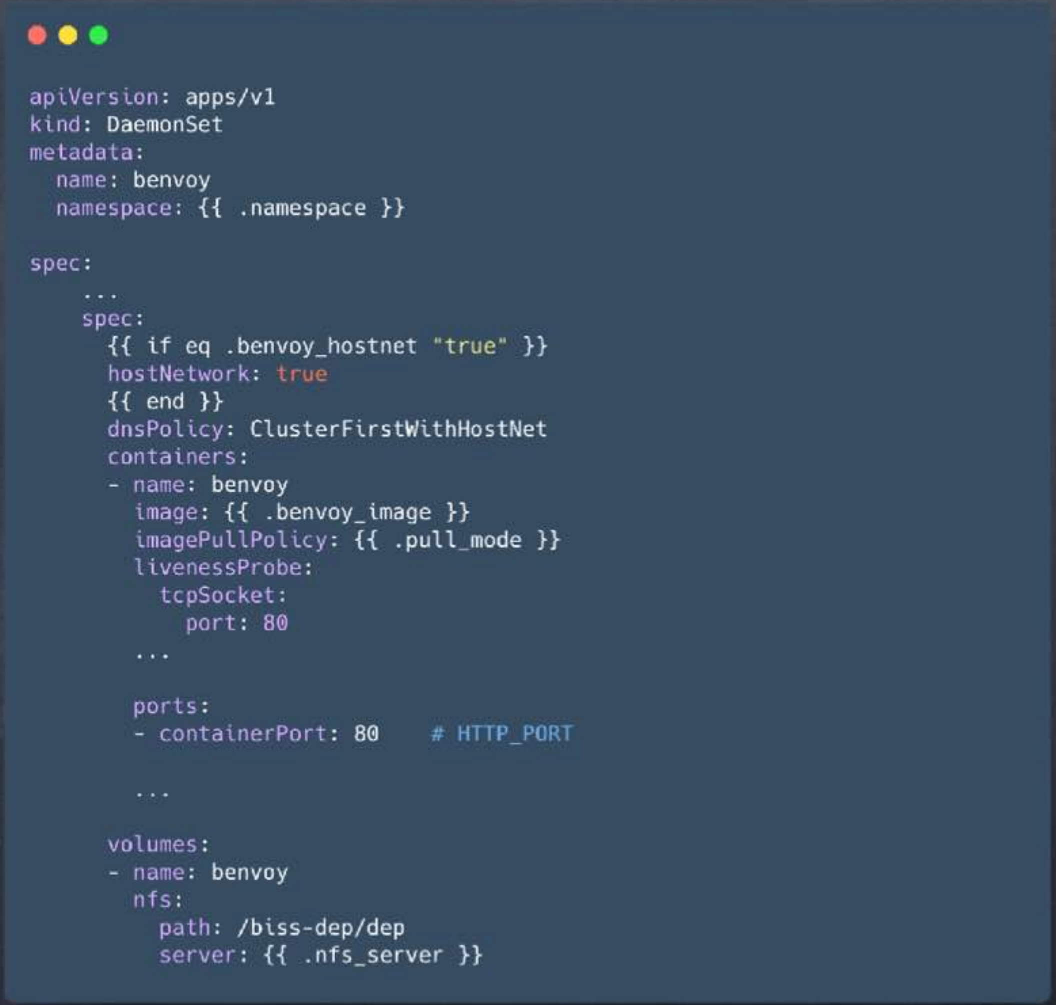
control script

- 通过env和template来生成k8s/istio配置
- 通过namespace来隔离每个developer的环境
- 可配置服务组启动，跳过注入及启动顺序等
- 通过control来管理mesh各资源的生命周期
  - gen; start; stop; restart; logs; pods; …

https://github.com/rfyiamcool/k8s-istio-control



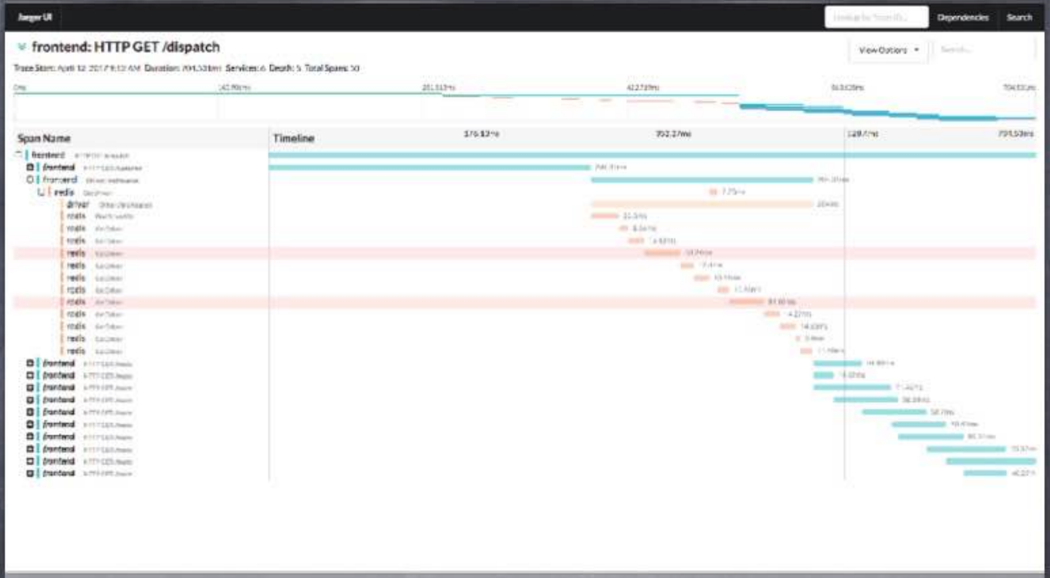
```yaml
# 生成的配置的路径
output_path: ./output

# 映射到模板里的变量
vars:
  run_env: TEST # PROD, DEV
  benvoy_hostnet: false
  namespace: ruifengyun
  nfs_server: 10.10.10.10
  ...

# 跳过istio注入的服务
skip_inject_service:
  - postgres
  - command-bus
  ...

# 服务列表
service:
  - benvoy
  - website
  ...

# 强制依赖
must_deps:
  - postgres

# 高优先级依赖，强制启动顺序
high_priority_deps:
  - postgres
  ...

mid_priority_deps:

# 服务组
service_group:
  middleware:
    - postgres
    - scylla
    ...
  trade:
    - trade-bus
    - trade-server
    - me
    ...

# 启动的服务
enable:
  service:
    - ...
  service_group:
    - ...
```



```yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: benvoy
  namespace: {{ .namespace }}

spec:
    ...
    spec:
      {{ if eq .benvoy_hostnet "true" }}
      hostNetwork: true
      {{ end }}
      dnsPolicy: ClusterFirstWithHostNet
      containers:
      - name: benvoy
        image: {{ .benvoy_image }}
        imagePullPolicy: {{ .pull_mode }}
        livenessProbe:
          tcpSocket:
            port: 80
        ...

        ports:
        - containerPort: 80    # HTTP_PORT

        ...

      volumes:
      - name: benvoy
        nfs:
          path: /biss-dep/dep
          server: {{ .nfs_server }}
```

other

- 分布式链路追踪
  - opentracing
    - jaeger
    - zipkin
  - design
    - trace ID
    - span ID



- 分布式日志追踪
  - 使用trace id追溯上下文

“ Q&A ”

- xiaorui.cc