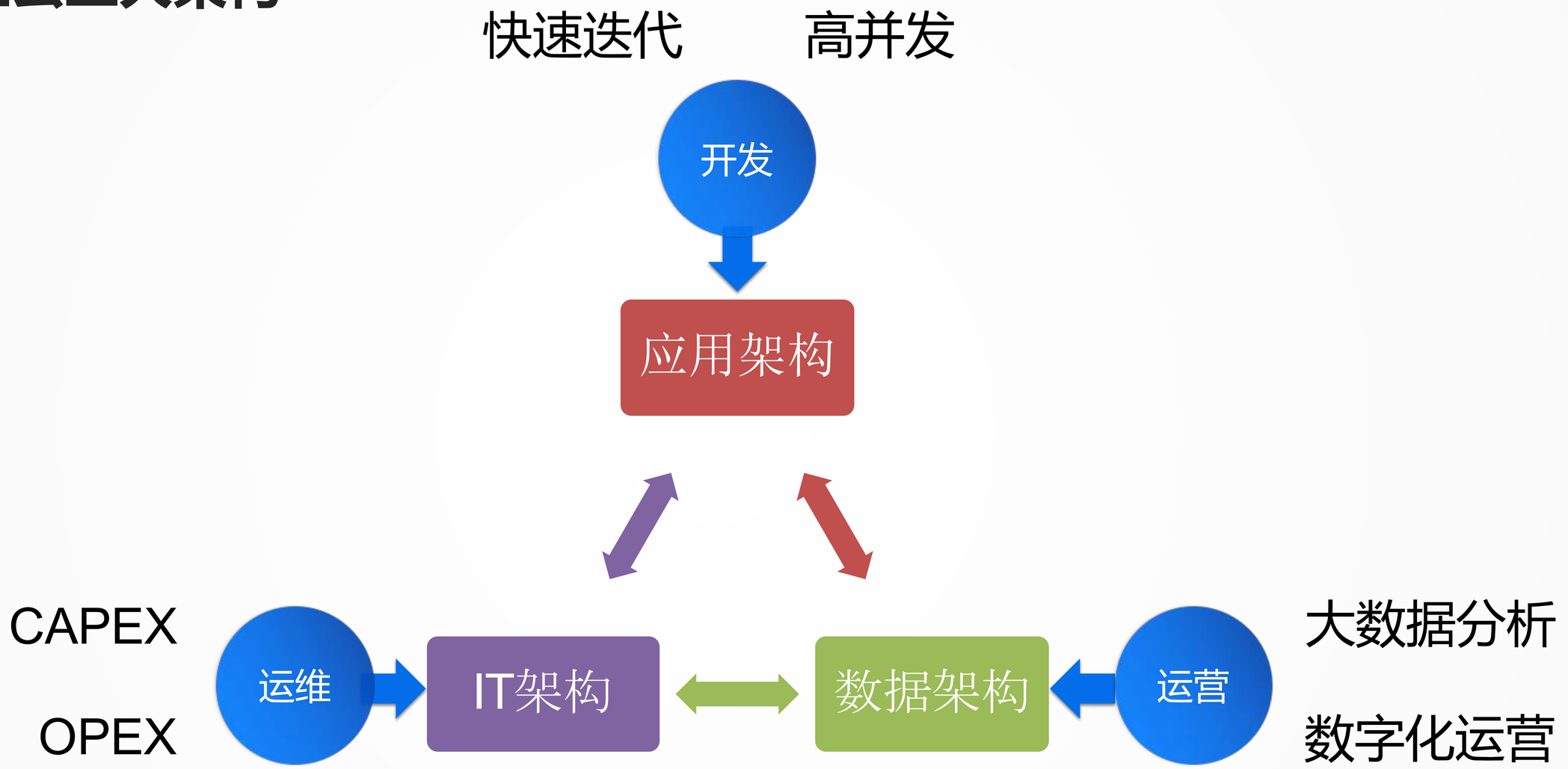
基于Service Mesh的海量容器管理平台实践

关于我

- **刘超**
- **网易云　解决方案总架构师**
- 10余年云计算领域研发及架构经验，先后在EMC，CCTV证券资讯频道，HP，华为，网易从事云计算和大数据架构工作
- 毕业于上海交通大学。
- 曾出版《Lucene应用开发揭秘》
- 多次作为邀请讲师参加Dockone容器技术大会，Segmentfault开发者大会，InfoQ全球架构师峰会（明星讲师），CSDN SDCC大会，51CTO WOTA大会等
- 知名技术博主，博客可搜索popsuper1982，多篇文章推荐至全球最大IT社区CSDN首页及《程序员》杂志
- 在工作中积累了大量运营商系统，互联网金融系统，电商系统等容器化和微服务化经验

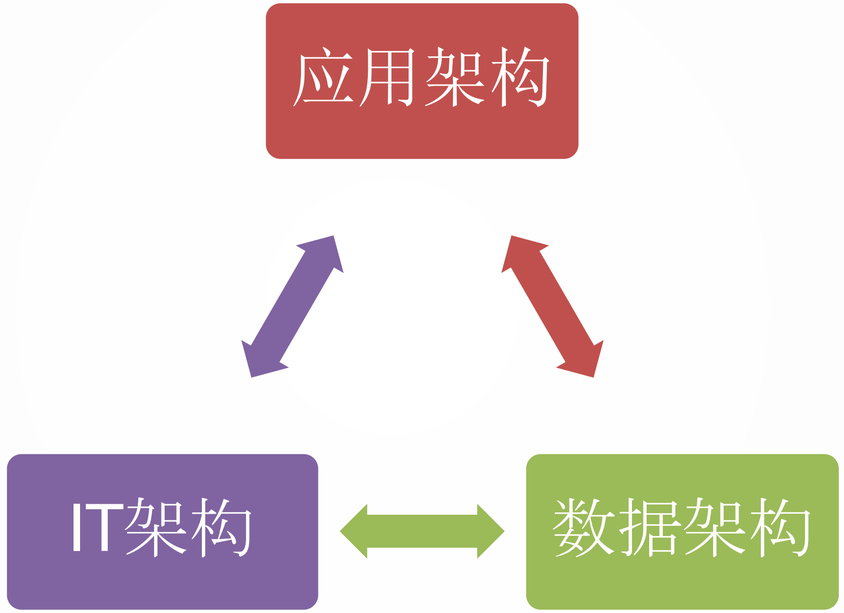
目录

企业上云三大架构



容器技术的三种视角

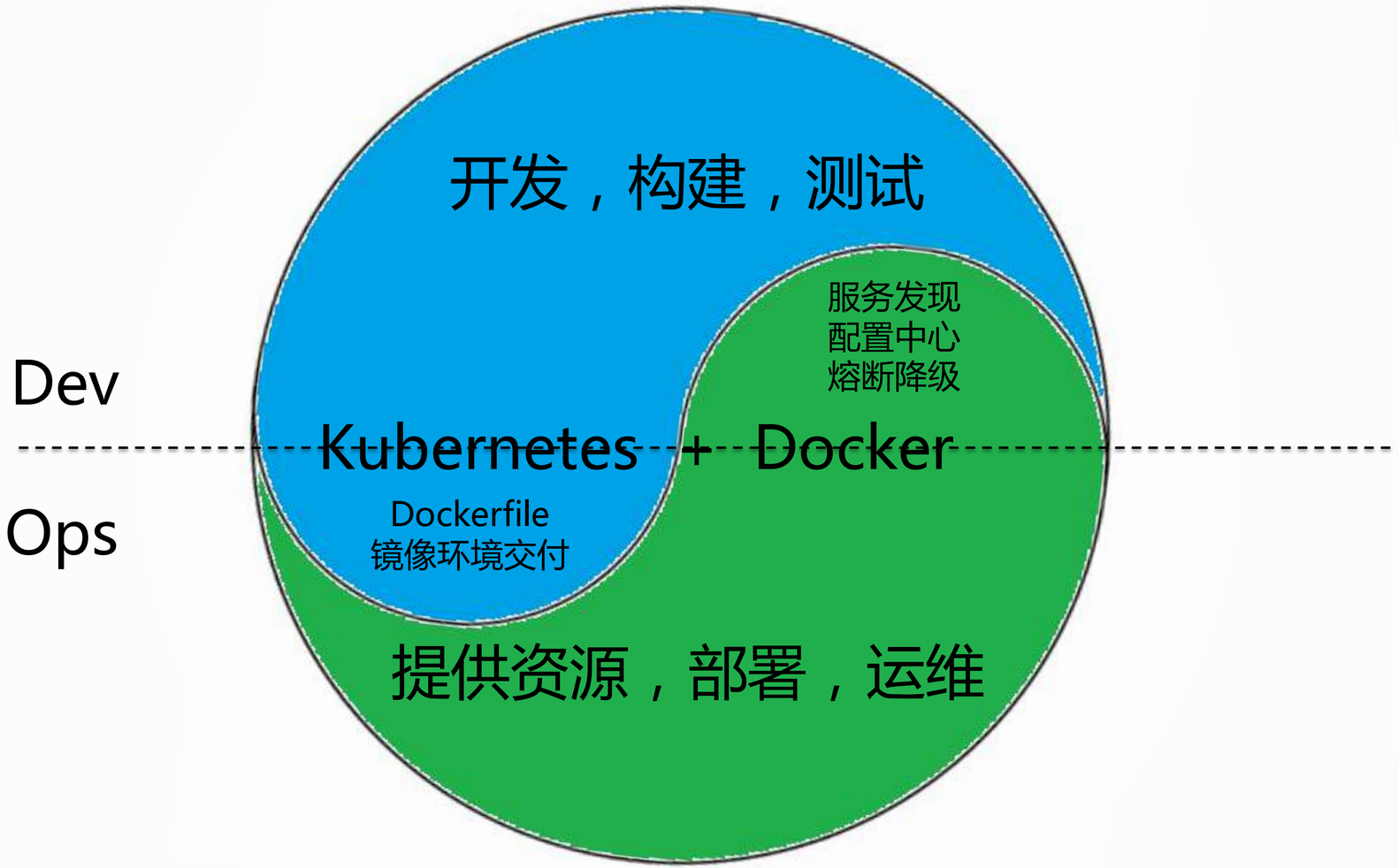
微服务的交付形式Kubernetes



轻量级的IT运维模式Swarm

资源利用率高的任务执行模式Mesos

Kubernetes + Docker 是 Dev 和 Ops 融合的一个桥梁。



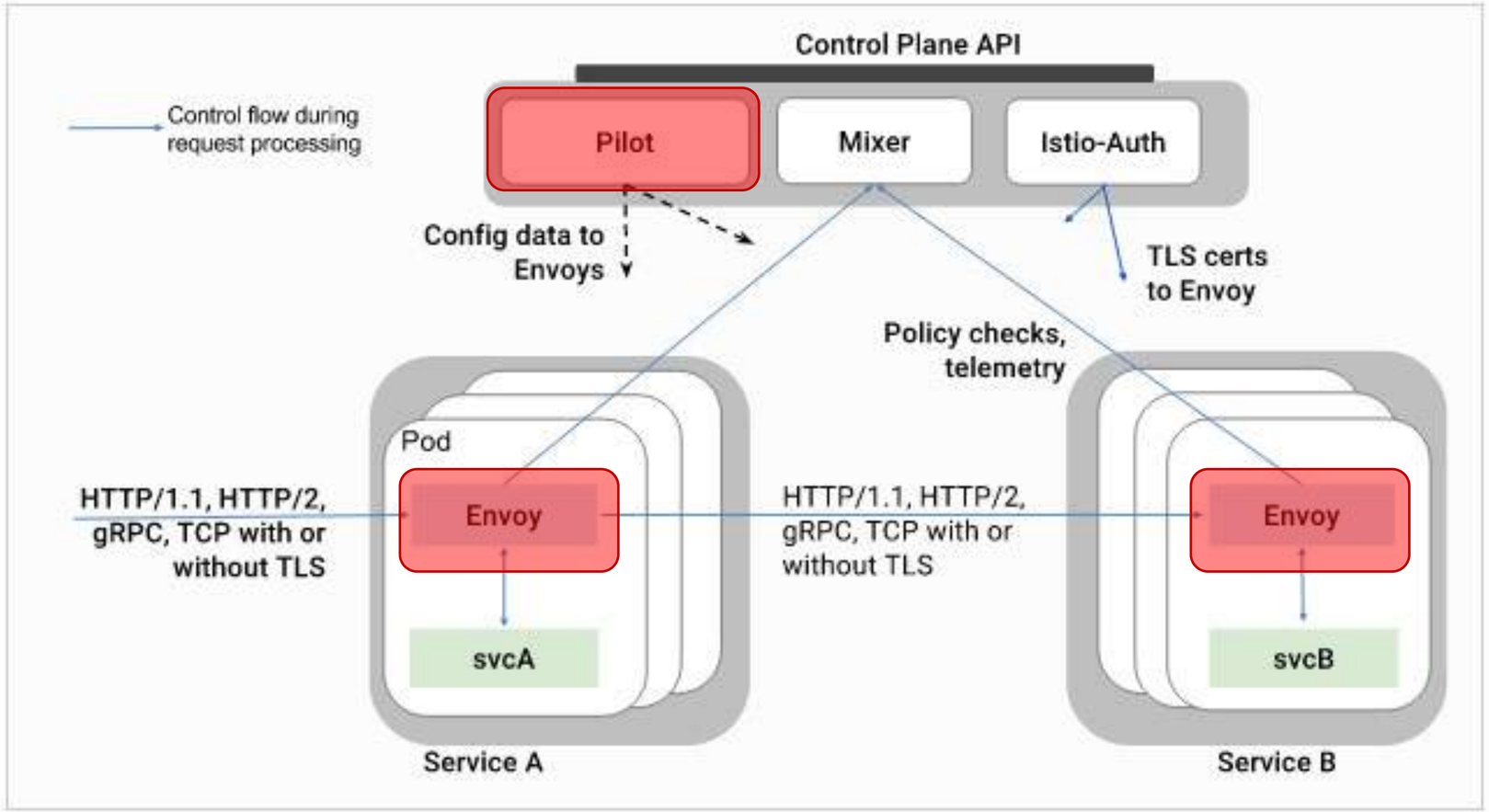
Kubernetes更加适合微服务和DevOps的设计

网易云

微服务设计	Kubernetes功能
设计要点一：API 网关	Ingress
设计要点二：无状态化，区分有状态的和无状态的应用。	无状态对应Deployment，有状态对应 StatefulSet
设计要点三：数据库的横向扩展。	headless service指向PaaS服务，或者StatefulSet部署
设计要点四：缓存	headless service指向PaaS服务，或者StatefulSet部署
设计要点五：服务拆分和服务发现	Service
设计要点六：服务编排与弹性伸缩	Deployment的Replicas
设计要点七：统一配置中心	ConfigMap
设计要点八：统一的日志中心	DaemonSet部署日志Agent
设计要点九：熔断，限流，降级	Service Mesh
设计要点十：全方位的监控	Cadvisor，DaemonSet部署监控Agent

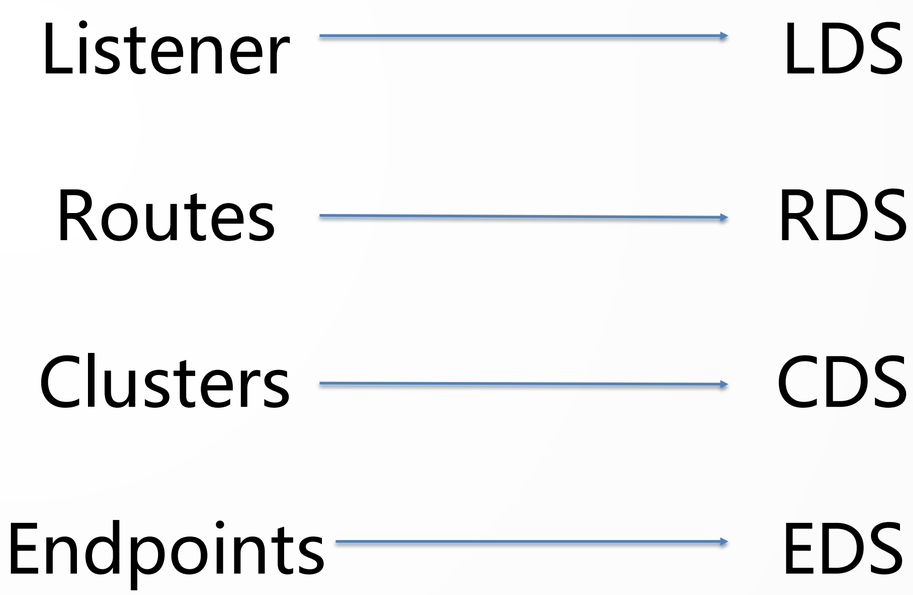
目录

Service Mesh的范例Istio



一切从envoy开始

- 轻量级的proxy
- 静态配置，热加载，热重启
- 动态配置，拉取模式



Envoy之静态配置

```
admin:
  access_log_path: /tmp/admin_access.log
  address:
    socket_address: { address: 127.0.0.1, port_value: 9901 }

static_resources:
  listeners:
  - name: listener_0
    address:
      socket_address: { address: 127.0.0.1, port_value: 10000 }
    filter_chains:
    - filters:
      - name: envoy.http_connection_manager
        config:
          stat_prefix: ingress_http
          codec_type: AUTO
          route_config:
            name: local_route
            virtual_hosts:
            - name: local_service
              domains: ["*"]
              routes:
              - match: { prefix: "/" }
                route: { cluster: some_service }
          http_filters:
          - name: envoy.router
  clusters:
  - name: some_service
    connect_timeout: 0.25s
    type: STATIC
    lb_policy: ROUND_ROBIN
    hosts: [{ socket_address: { address: 127.0.0.2, port_value: 1234 }}]
```

Listener

Routes

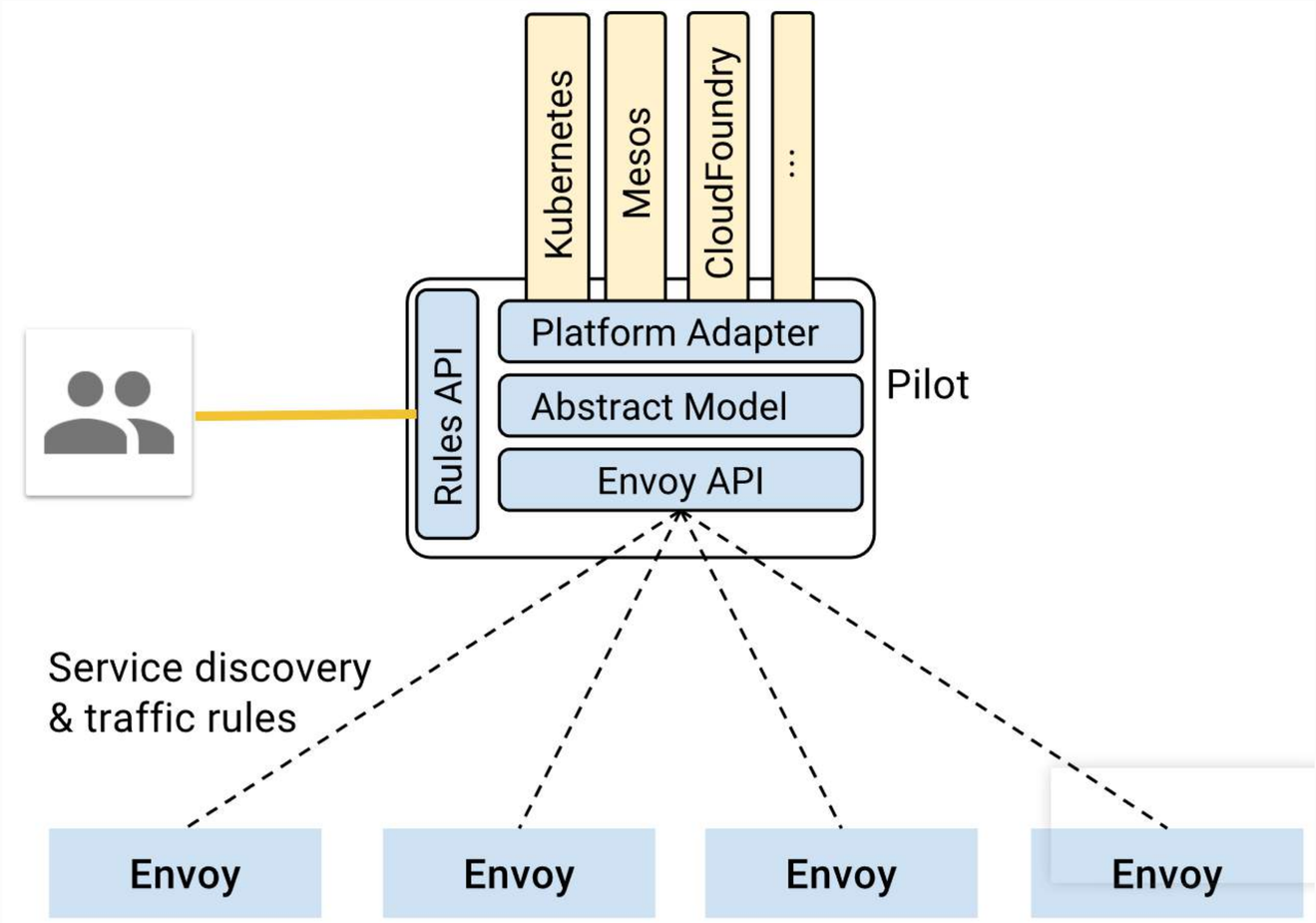
Clusters

Endpoints

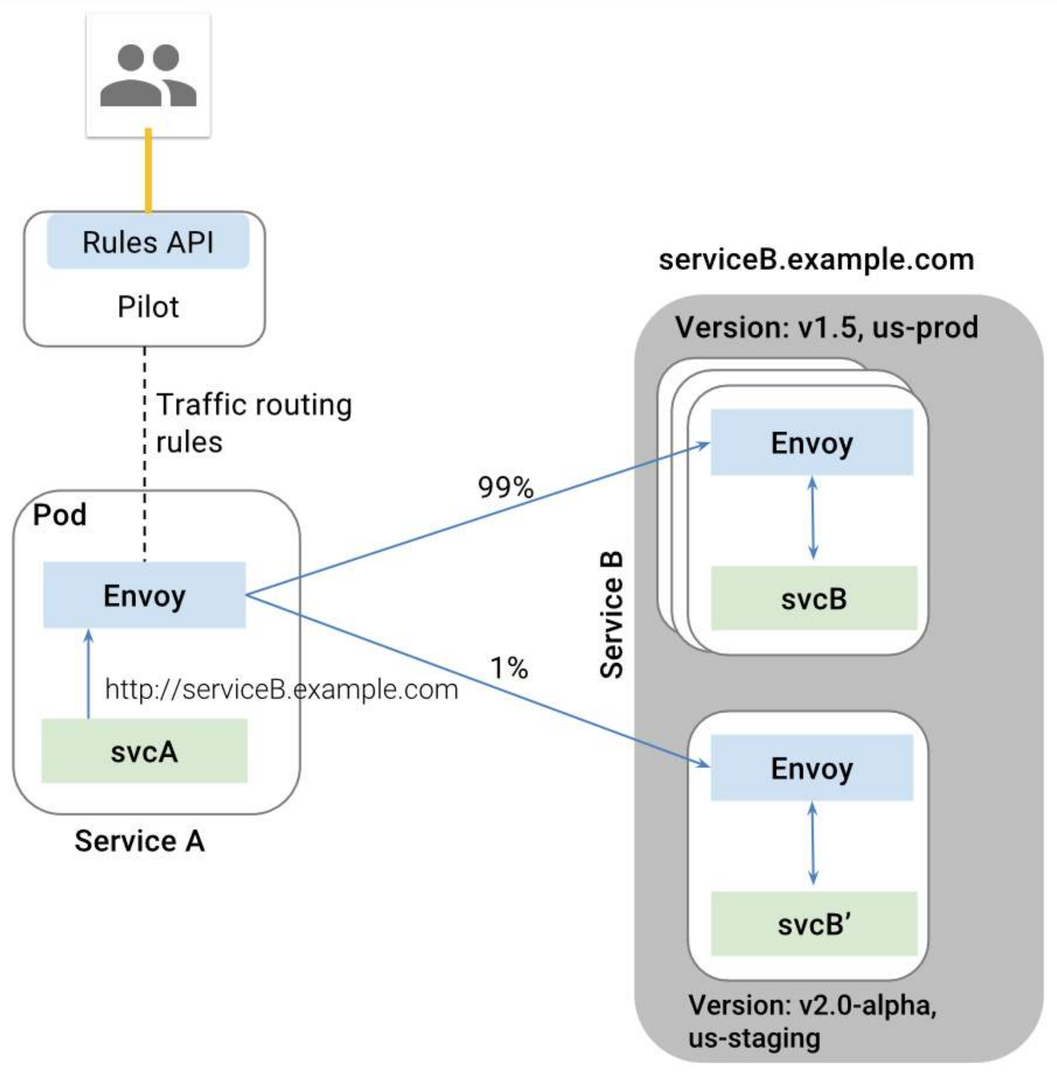
Envoy之动态配置

```
admin:
  access_log_path: /tmp/admin_access.log
  address:
    socket_address: { address: 127.0.0.1, port_value: 9901 }

dynamic_resources:
  lds_config:
    api_config_source:
      api_type: GRPC
      cluster_names: [xds_cluster]
  cds_config:
    api_config_source:
      api_type: GRPC
      cluster_names: [xds_cluster]

static_resources:
  clusters:
  - name: xds_cluster
    connect_timeout: 0.25s
    type: STATIC
    lb_policy: ROUND_ROBIN
    http2_protocol_options: {}
    hosts: [{ socket_address: { address: 127.0.0.3, port_value: 5678 }}]
```

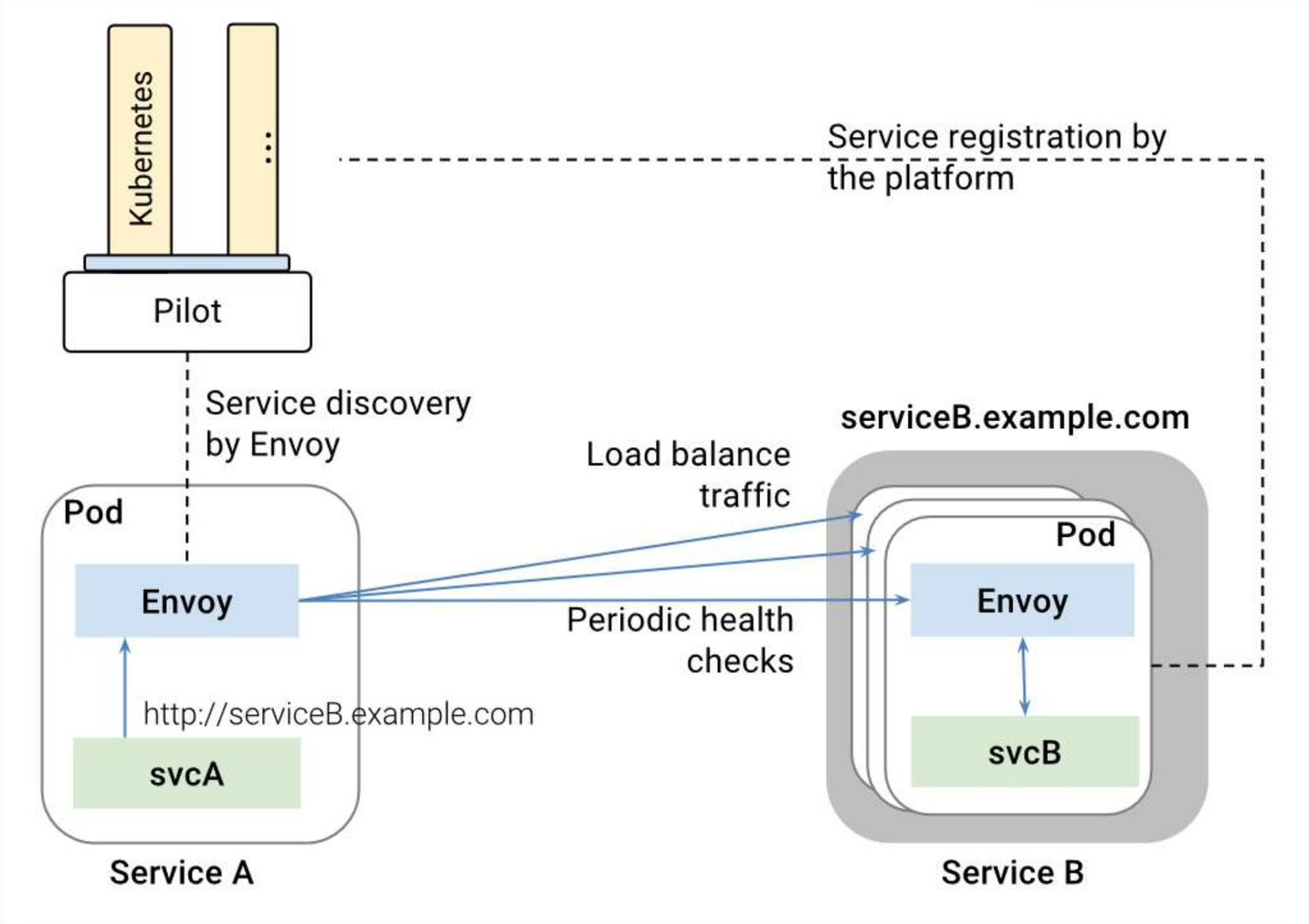
加入控制面Pilot


Kubernetes
Mesos
CloudFoundry
...
Rules API
Platform Adapter
Abstract Model
Envoy API
Pilot
Service discovery
& traffic rules
Envoy
Envoy
Envoy
Envoy

Pilot路由策略

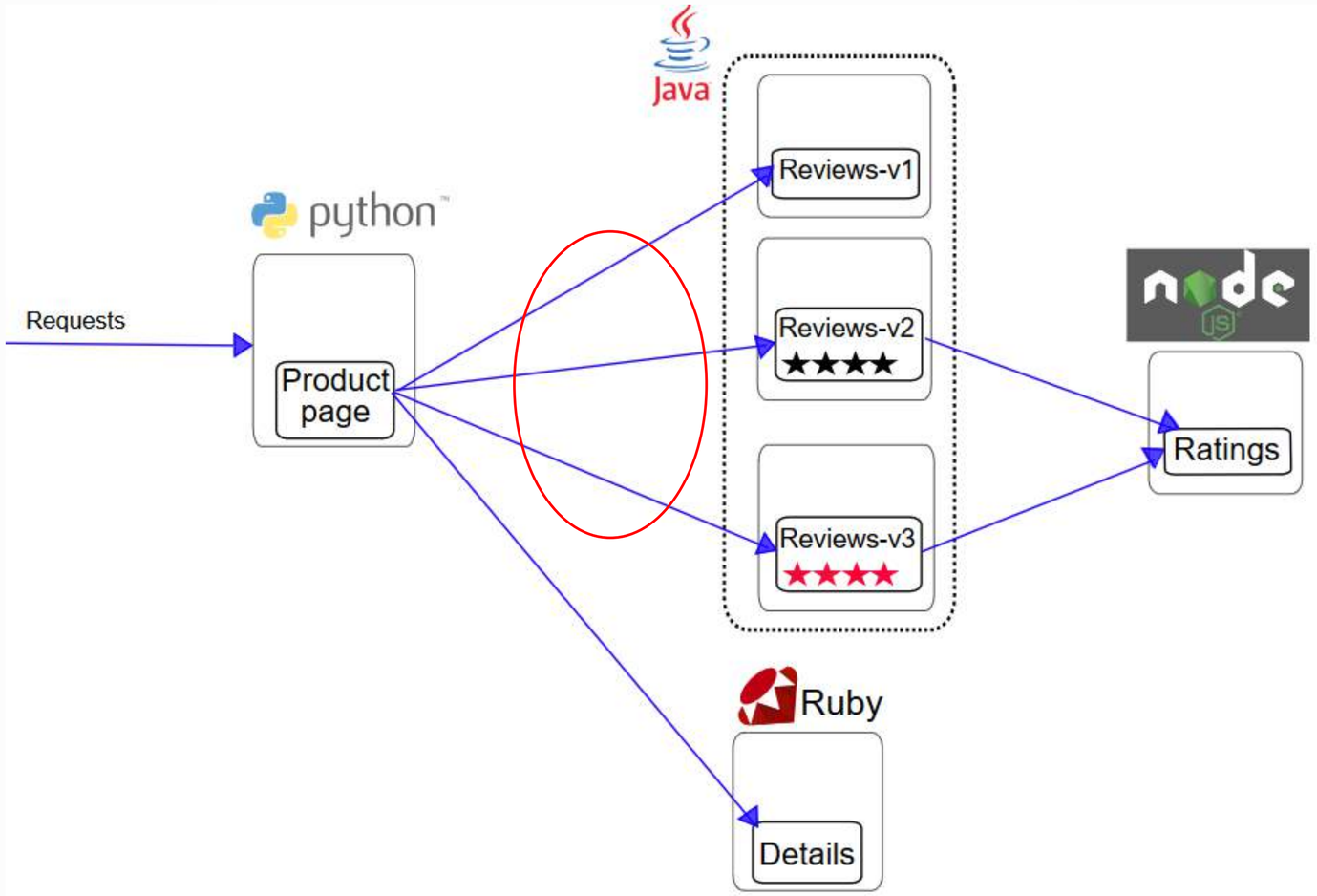

Rules API
Pilot
Traffic routing rules
Pod
Envoy
http://serviceB.example.com
svcA
Service A
99%
1%
serviceB.example.com
Version: v1.5, us-prod
Envoy
svcB
Service B
Envoy
svcB'
Version: v2.0-alpha, us-staging

Pilot的负载均衡


Kubernetes
...
Pilot
Service registration by the platform
Service discovery by Envoy
serviceB.example.com
Load balance traffic
Pod
Envoy
Periodic health checks
Pod
Envoy
http://serviceB.example.com
svcA
svcB
Service A
Service B

目录

这个简单例子背后发生的事情



Productpage是一个python程序

```
@app.route('/api/v1/products')
def productsRoute():
    return json.dumps(getProducts()), 200, {'Content-Type': 'application/json'}


@app.route('/api/v1/products/<product_id>')
def productRoute(product_id):
    headers = getForwardHeaders(request)
    status, details = getProductDetails(product_id, headers)
    return json.dumps(details), status, {'Content-Type': 'application/json'}


@app.route('/api/v1/products/<product_id>/reviews')
def reviewsRoute(product_id):
    headers = getForwardHeaders(request)
    status, reviews = getProductReviews(product_id, headers)
    return json.dumps(reviews), status, {'Content-Type': 'application/json'}


@app.route('/api/v1/products/<product_id>/ratings')
def ratingsRoute(product_id):
    headers = getForwardHeaders(request)
    status, ratings = getProductRatings(product_id, headers)
    return json.dumps(ratings), status, {'Content-Type': 'application/json'}
```

```
details = {
    "name" : "http://details:9080",
    "endpoint" : "details",
    "children" : []
}

ratings = {
    "name" : "http://ratings:9080",
    "endpoint" : "ratings",
    "children" : []
}

reviews = {
    "name" : "http://reviews:9080",
    "endpoint" : "reviews",
    "children" : [ratings]
}
```

使用Kubernetes编排productpage

```
apiVersion: v1
kind: Service
metadata:
  name: productpage
  labels:
    app: productpage
spec:
  ports:
  - port: 9080
    name: http
  selector:
    app: productpage
---
```

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: productpage-v1
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: productpage
        version: v1
    spec:
      containers:
      - name: productpage
        image: istio/examples-bookinfo-productpage-
v1:1.5.0
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 9080
---
```

使用Kubernetes编排reviews

```
apiVersion: v1
kind: Service
metadata:
  name: reviews
  labels:
    app: reviews
spec:
  ports:
  - port: 9080
    name: http
  selector:
    app: reviews
---
```

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: reviews-v1
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: reviews
        version: v1
    spec:
      containers:
      - name: reviews
        image: istio/examples-bookinfo-reviews-v1:1.5.0
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 9080
---
```

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: reviews-v2
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: reviews
        version: v2
    spec:
      containers:
      - name: reviews
        image: istio/examples-bookinfo-reviews-v2:1.5.0
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 9080
---
```

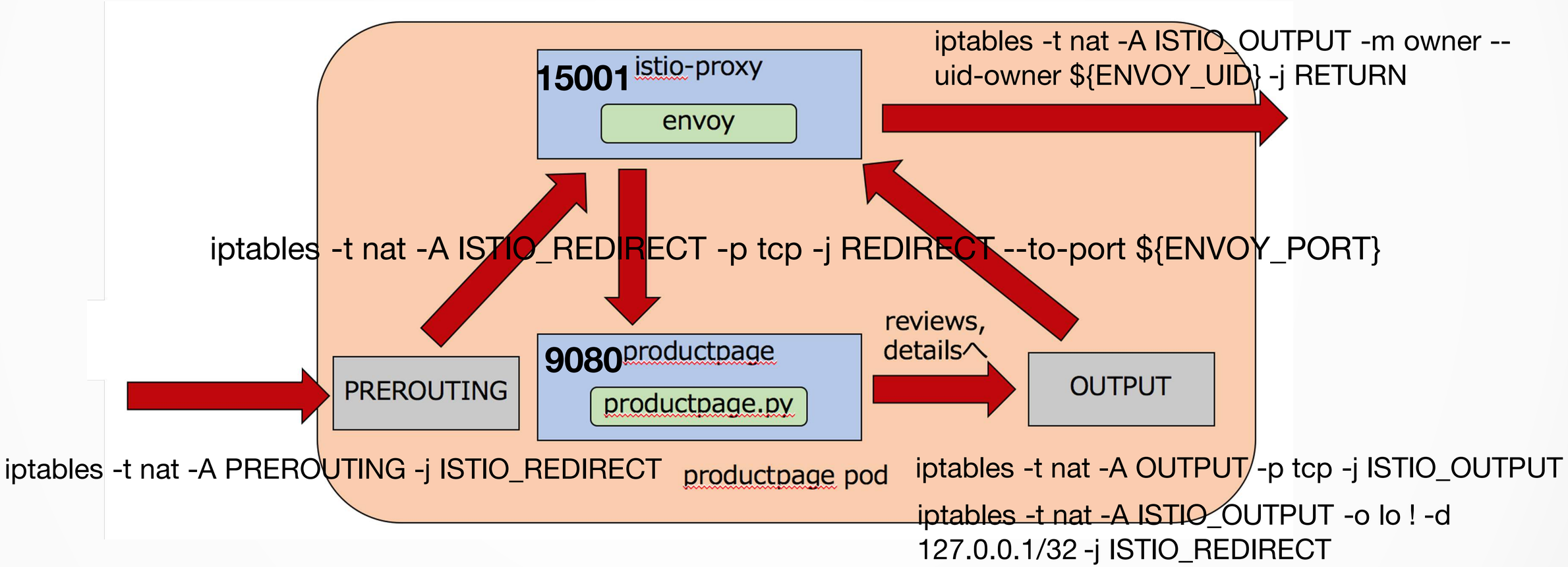
```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: reviews-v3
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: reviews
        version: v3
    spec:
      containers:
      - name: reviews
        image: istio/examples-bookinfo-reviews-v3:1.5.0
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 9080
---
```

嵌入proxy_init作为InitContainer

```
Init Containers:
  istio-init:
    Image:      docker.io/istio/proxy_init:0.7.1
    Port:       <none>
    Host Port:  <none>
    Args:
      -p
      15001
      -u
      1337
    Environment:  <none>
    Mounts:       <none>
  enable-core-dump:
    Image:      alpine
    Port:       <none>
    Host Port:  <none>
    Command:
      /bin/sh
    Args:
      -c
      sysctl -w kernel.core_pattern=/etc/istio/proxy/core.%e.%p.%t && ulimit -c unlimited
    Environment:  <none>
    Mounts:       <none>
```

proxy_init设置iptables规则

/usr/local/bin/prepare_proxy.sh -p PORT -u UID



嵌入proxy容器作为sidecar

```
istio-proxy:
    Image:      docker.io/istio/proxy_debug:0.7.1
    Port:       <none>
    Host Port:  <none>
    Args:
      proxy
      sidecar
      --configPath
      /etc/istio/proxy
      --binaryPath
      /usr/local/bin/envoy
      --serviceCluster
      productpage
      --drainDuration
      45s
      --parentShutdownDuration
      1m0s
      --discoveryAddress
      istio-pilot.istio-system:8080
      --discoveryRefreshDelay
      1s
```

SideCar内启动的进程

```
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
istio-p+     1  0.0  0.2  32656 19108 ?        Ssl  08:55   0:00 /usr/local/bin/pilot-agent proxy sidecar --configPath
/etc/istio/proxy --binaryPath /usr/local/bin/envoy --serviceCluster productpage --drainDuration 45s --
parentShutdownDuration 1m0s --discoveryAddress istio-pilot.istio-system:8080 --discoveryRefreshDelay 1s --
zipkinAddress zipkin.istio-system:9411 --connectTimeout 10s --statsdUdpAddress istio-mixer.istio-system:9125 --
proxyAdminPort 15000 --controlPlaneAuthPolicy NONE

istio-p+    11  3.9  0.5 127352 41068 ?        Sl   08:55  13:19 /usr/local/bin/envoy -c /etc/istio/proxy/envoy-rev0.json
--restart-epoch 0 --drain-time-s 45 --parent-shutdown-time-s 60 --service-cluster productpage --service-node
sidecar~192.168.1.114~productpage-v1-8666ffbd7c-mss5p.default~default.svc.cluster.local --max-obj-name-len
189 -l info --v2-config-only
```

Envoy进程的配置

```
"admin": {
  "access_log_path": "/dev/stdout",
  "address": {
   "socket_address": {
    "address": "127.0.0.1",
    "port_value": 15000
   }
  }
},
```

管理端口

```
"dynamic_resources": {
  "lds_config": {
   "api_config_source": {
    "api_type": "REST_LEGACY",
    "refresh_delay": {"seconds": 1,
"nanos": 0},
    "cluster_names": [
     "rds"
    ]
   }
  },
  "cds_config": {
   "api_config_source": {
    "api_type": "REST_LEGACY",
    "refresh_delay": {"seconds": 1,
"nanos": 0},
    "cluster_names": [
     "rds"
    ]
   }
  }
```

动态资源

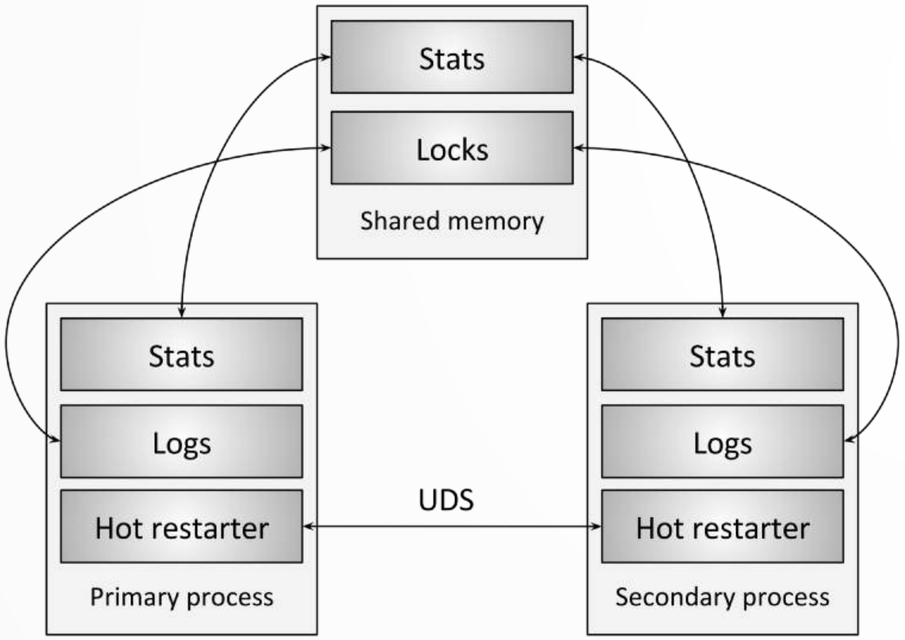
```
 "static_resources": {
   "clusters": [
    {
     "name": "rds",
     "type": "STRICT_DNS",
     "connect_timeout": {"seconds": 10,
"nanos": 0},
     "lb_policy": "ROUND_ROBIN",

    "hosts": [
      {
       "socket_address": {"address": "istio-
pilot.istio-system", "port_value": 8080}
      }
     ]

   },
```

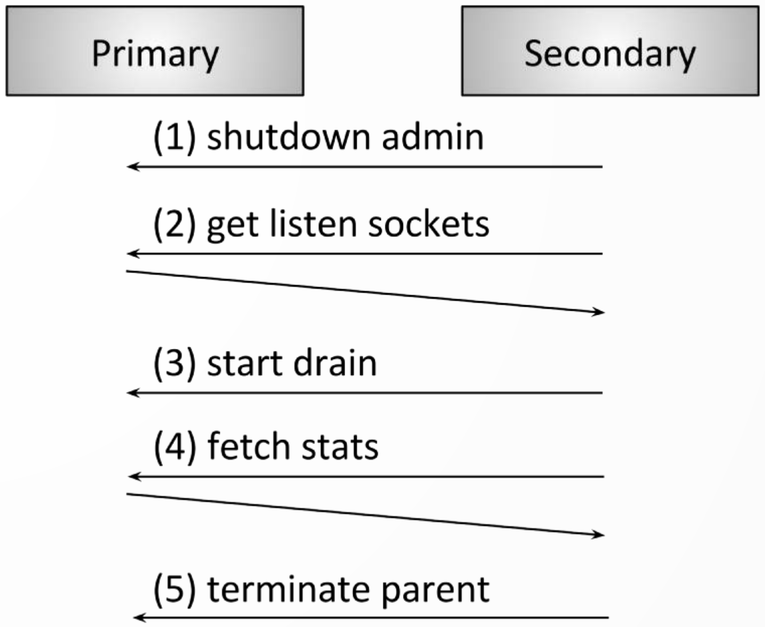
静态资源

Pilot Agent的作用

Wrapper of envoy

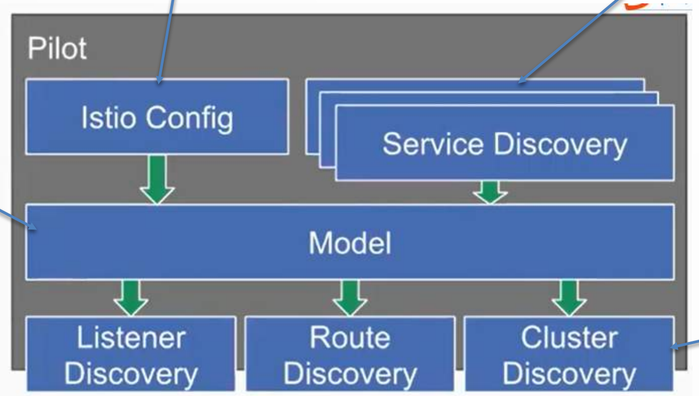
当envoy的静态配置改变的时候，对envoy进行热重启后生效(例如TLS)





Pilot的工作机制

```
apiVersion: config.istio.io/v1alpha2
kind: RouteRule
metadata:
  name: reviews-default
  namespace: default
spec:
  destination:
    name: reviews
  precedence: 1
  route:
  - labels:
      version: v1
    weight: 50
  - labels:
      version: v3
    weight: 50
```



在pilot上配置route

```
apiVersion: config.istio.io/v1alpha2
kind: RouteRule
metadata:
  name: reviews-default
spec:
  destination:
    name: reviews
  precedence: 1
  route:
  - labels:
      version: v1
```

查看envoy的管理端口

curl http://127.0.0.1:15000/clusters


```
{
 "name": "reviews.default.svc.cluster.local|http",
 "domains": [
  "reviews:9080",
  "reviews",
  "reviews.default:9080",
  "reviews.default",
  "reviews.default.svc:9080",
  "reviews.default.svc",
  "reviews.default.svc.cluster:9080",
  "reviews.default.svc.cluster",
  "reviews.default.svc.cluster.local:9080",
  "reviews.default.svc.cluster.local",
  "10.104.14.93:9080",
  "10.104.14.93"
 ],
 "routes": [
  {
   "match": {
    "prefix": "/"
   },
   "route": {
    "cluster": "out.reviews.default.svc.cluster.local|http|version=v1",
    "timeout": "0s"
   },
   "decorator": {
    "operation": "reviews-default"
   }
  }
 ]
}
```


```
istio-proxy@productpage-v1-8666ffbd7c-mss5p:/$ curl http://127.0.0.1:15000/clusters | grep reviews
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
out.reviews.default.svc.cluster.local|http|version=v1::default_priority::max_connections::1024
100 52770    0 52770    0     0  16.1M      0 --:--:-- --:--:-- --:--:-- 25.1M
out.reviews.default.svc.cluster.local|http|version=v1::default_priority::max_pending_requests::1024
out.reviews.default.svc.cluster.local|http|version=v1::default_priority::max_requests::1024
out.reviews.default.svc.cluster.local|http|version=v1::default_priority::max_retries::3
out.reviews.default.svc.cluster.local|http|version=v1::high_priority::max_connections::1024
out.reviews.default.svc.cluster.local|http|version=v1::high_priority::max_pending_requests::1024
out.reviews.default.svc.cluster.local|http|version=v1::high_priority::max_requests::1024
out.reviews.default.svc.cluster.local|http|version=v1::high_priority::max_retries::3
out.reviews.default.svc.cluster.local|http|version=v1::added_via_api::true
out.reviews.default.svc.cluster.local|http|version=v1::192.168.2.23:9080::cx_active::0
out.reviews.default.svc.cluster.local|http|version=v1::192.168.2.23:9080::cx_connect_fail::0
out.reviews.default.svc.cluster.local|http|version=v1::192.168.2.23:9080::cx_total::0
out.reviews.default.svc.cluster.local|http|version=v1::192.168.2.23:9080::rq_active::0
out.reviews.default.svc.cluster.local|http|version=v1::192.168.2.23:9080::rq_error::0
out.reviews.default.svc.cluster.local|http|version=v1::192.168.2.23:9080::rq_success::0
out.reviews.default.svc.cluster.local|http|version=v1::192.168.2.23:9080::rq_timeout::0
out.reviews.default.svc.cluster.local|http|version=v1::192.168.2.23:9080::rq_total::0
out.reviews.default.svc.cluster.local|http|version=v1::192.168.2.23:9080::health_flags::healthy
out.reviews.default.svc.cluster.local|http|version=v1::192.168.2.23:9080::weight::1
out.reviews.default.svc.cluster.local|http|version=v1::192.168.2.23:9080::region::
out.reviews.default.svc.cluster.local|http|version=v1::192.168.2.23:9080::zone::
out.reviews.default.svc.cluster.local|http|version=v1::192.168.2.23:9080::sub_zone::
out.reviews.default.svc.cluster.local|http|version=v1::192.168.2.23:9080::canary::false
out.reviews.default.svc.cluster.local|http|version=v1::192.168.2.23:9080::success_rate::-1
```

修改pilot的route

```
root@ip-172-31-15-9:~# cat ./istio-0.7.1/samples/bookinfo/kube/route-rule-reviews-50-v3.yaml
apiVersion: config.istio.io/v1alpha2
kind: RouteRule
metadata:
  name: reviews-default
spec:
  destination:
    name: reviews
  precedence: 1
  route:
  - labels:
      version: v1
    weight: 50
  - labels:
      version: v3
    weight: 50
```

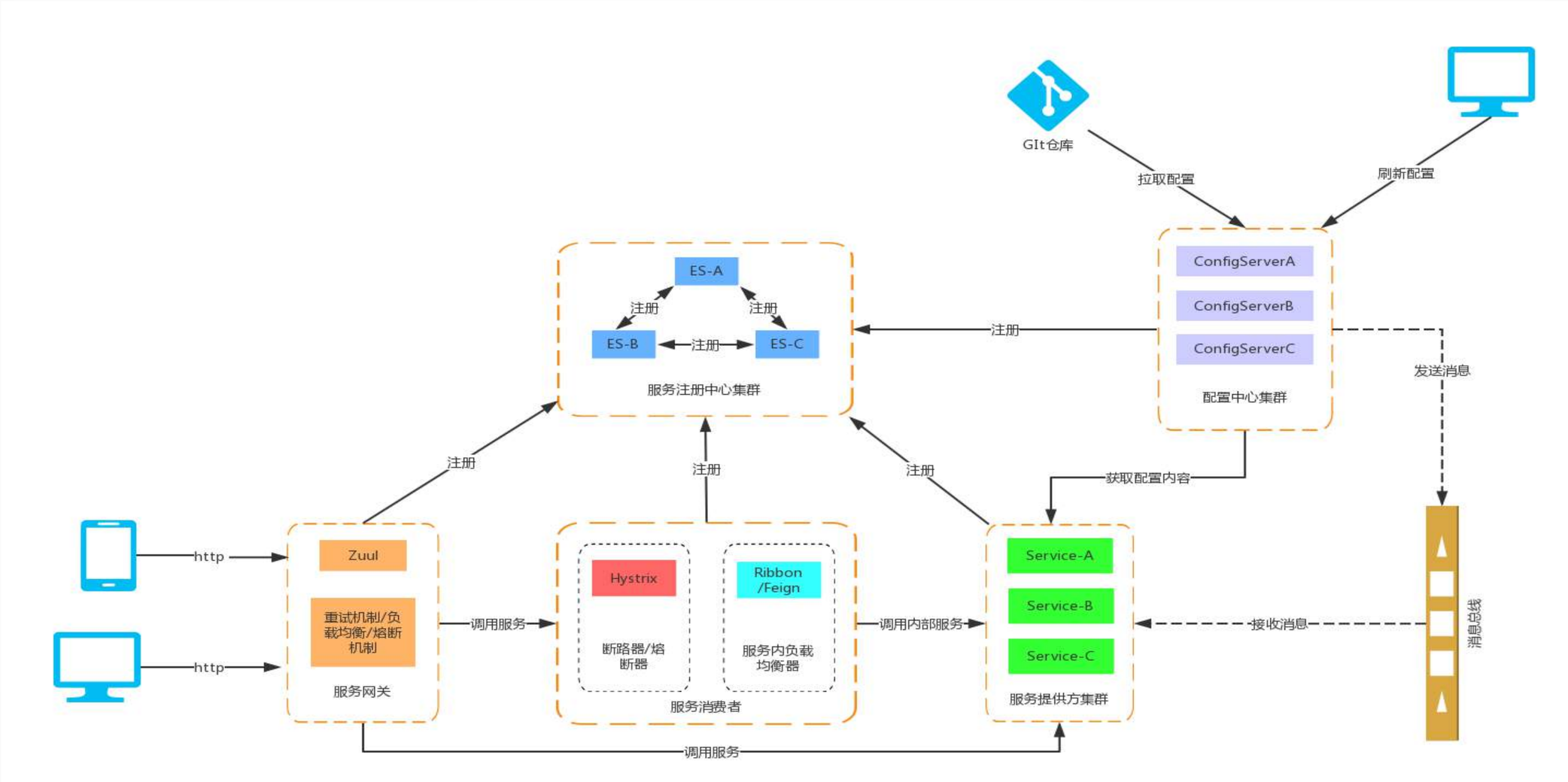
Envoy中的规则被修改


```
{
 "name": "reviews.default.svc.cluster.local|http",
 "domains": [
  "reviews:9080",
  "reviews",
  "reviews.default:9080",
  "reviews.default",
  "reviews.default.svc:9080",
  "reviews.default.svc",
  "reviews.default.svc.cluster:9080",
  "reviews.default.svc.cluster",
  "reviews.default.svc.cluster.local:9080",
  "reviews.default.svc.cluster.local",
  "10.104.14.93:9080",
  "10.104.14.93"
 ],
 "routes": [
  {
   "match": {
    "prefix": "/"
   },
   "route": {
    "weighted_clusters": {
     "clusters": [
      {
       "name": "out.reviews.default.svc.cluster.local|http|version=v1",
       "weight": 50
      },
      {
       "name": "out.reviews.default.svc.cluster.local|http|version=v3",
       "weight": 50
      }
     ]
    },
    "timeout": "0s"
   },
   "decorator": {
    "operation": "reviews-default"
```

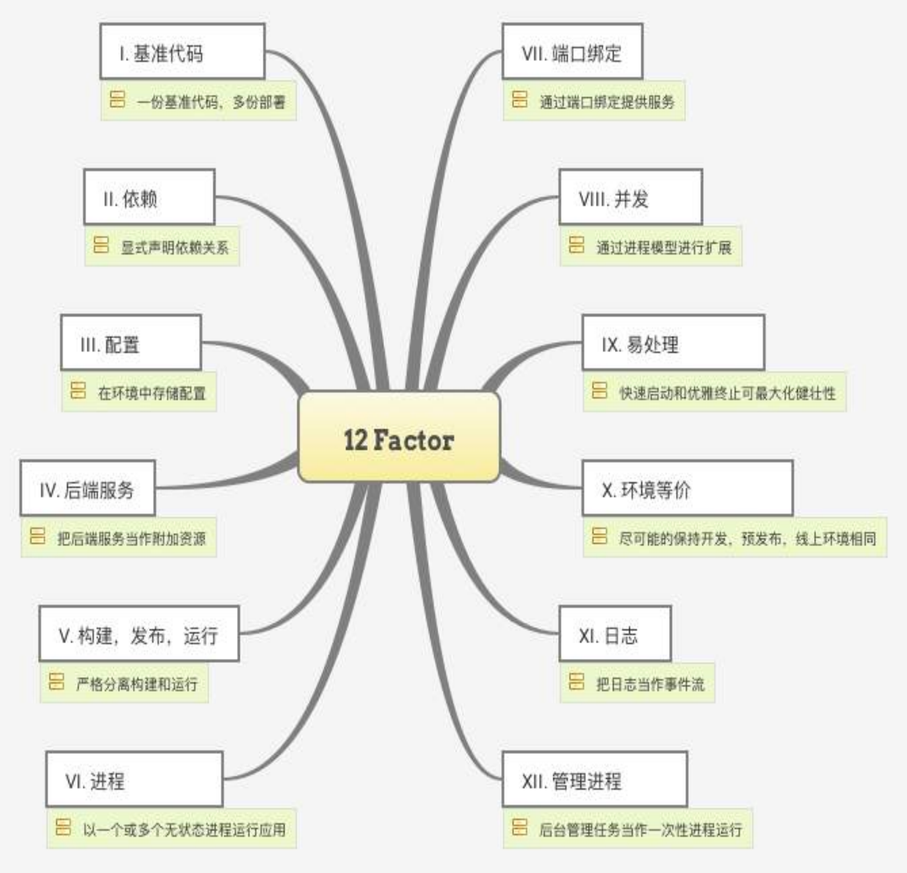

```
istio-proxy@productpage-v1-8666ffbd7c-mss5p:/$ curl http://127.0.0.1:15000/clusters | grep reviews
0out.reviews.default.svc.cluster.local|http|version=v3::default_priority::max_connections::1024
out.reviews.default.svc.cluster.local|http|version=v3::default_priority::max_pending_requests::1024
out.reviews.default.svc.cluster.local|http|version=v3::default_priority::max_requests::1024
out.reviews.default.svc.cluster.local|http|version=v3::default_priority::max_retries::3
out.reviews.default.svc.cluster.local|http|version=v3::high_priority::max_connections::1024
out.reviews.default.svc.cluster.local|http|version=v3::high_priority::max_pending_requests::1024
out.reviews.default.svc.cluster.local|http|version=v3::high_priority::max_requests::1024
out.reviews.default.svc.cluster.local|http|version=v3::high_priority::max_retries::3
out.reviews.default.svc.cluster.local|http|version=v3::added_via_api::true
out.reviews.default.svc.cluster.local|http|version=v3::192.168.2.24:9080::cx_active::0
out.reviews.default.svc.cluster.local|http|version=v3::192.168.2.24:9080::cx_connect_fail::0
out.reviews.default.svc.cluster.local|http|version=v3::192.168.2.24:9080::cx_total::0
out.reviews.default.svc.cluster.local|http|version=v3::192.168.2.24:9080::rq_active::0
out.reviews.default.svc.cluster.local|http|version=v3::192.168.2.24:9080::rq_error::0
out.reviews.default.svc.cluster.local|http|version=v3::192.168.2.24:9080::rq_success::0
out.reviews.default.svc.cluster.local|http|version=v3::192.168.2.24:9080::rq_timeout::0
out.reviews.default.svc.cluster.local|http|version=v3::192.168.2.24:9080::rq_total::0
out.reviews.default.svc.cluster.local|http|version=v3::192.168.2.24:9080::health_flags::healthy
out.reviews.default.svc.cluster.local|http|version=v3::192.168.2.24:9080::weight::1
out.reviews.default.svc.cluster.local|http|version=v3::192.168.2.24:9080::region::
out.reviews.default.svc.cluster.local|http|version=v3::192.168.2.24:9080::zone::
out.reviews.default.svc.cluster.local|http|version=v3::192.168.2.24:9080::sub_zone::
out.reviews.default.svc.cluster.local|http|version=v3::192.168.2.24:9080::canary::false
out.reviews.default.svc.cluster.local|http|version=v3::192.168.2.24:9080::success_rate::-1
out.reviews.default.svc.cluster.local|http|version=v1::default_priority::max_connections::1024
out.reviews.default.svc.cluster.local|http|version=v1::default_priority::max_pending_requests::1024
out.reviews.default.svc.cluster.local|http|version=v1::default_priority::max_requests::1024
out.reviews.default.svc.cluster.local|http|version=v1::default_priority::max_retries::3
out.reviews.default.svc.cluster.local|http|version=v1::high_priority::max_connections::1024
out.reviews.default.svc.cluster.local|http|version=v1::high_priority::max_pending_requests::1024
out.reviews.default.svc.cluster.local|http|version=v1::high_priority::max_requests::1024
out.reviews.default.svc.cluster.local|http|version=v1::high_priority::max_retries::3
out.reviews.default.svc.cluster.local|http|version=v1::added_via_api::true
out.reviews.default.svc.cluster.local|http|version=v1::192.168.2.23:9080::cx_active::0
out.reviews.default.svc.cluster.local|http|version=v1::192.168.2.23:9080::cx_connect_fail::0
out.reviews.default.svc.cluster.local|http|version=v1::192.168.2.23:9080::cx_total::0
out.reviews.default.svc.cluster.local|http|version=v1::192.168.2.23:9080::rq_active::0
```

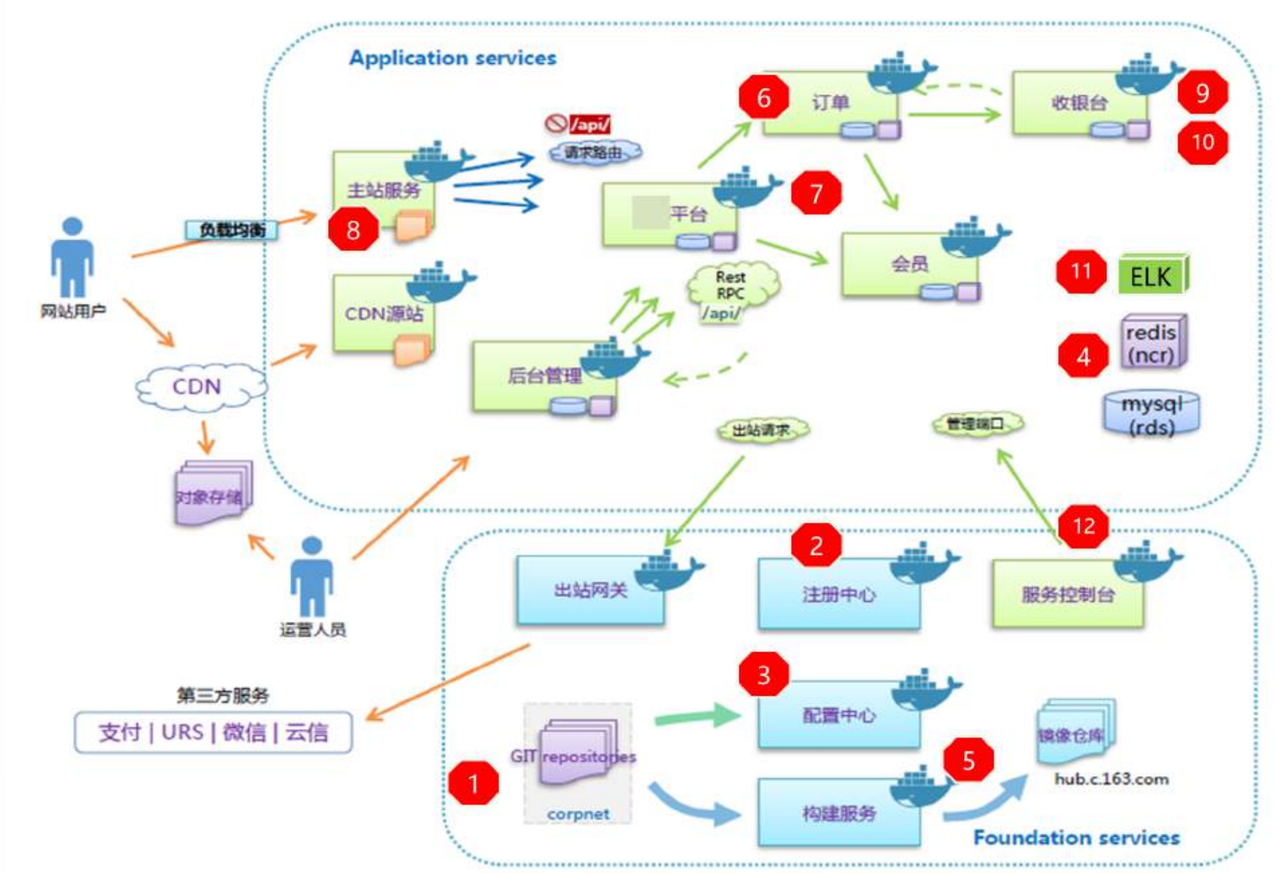
目录

基于SpringCloud的微服务框架

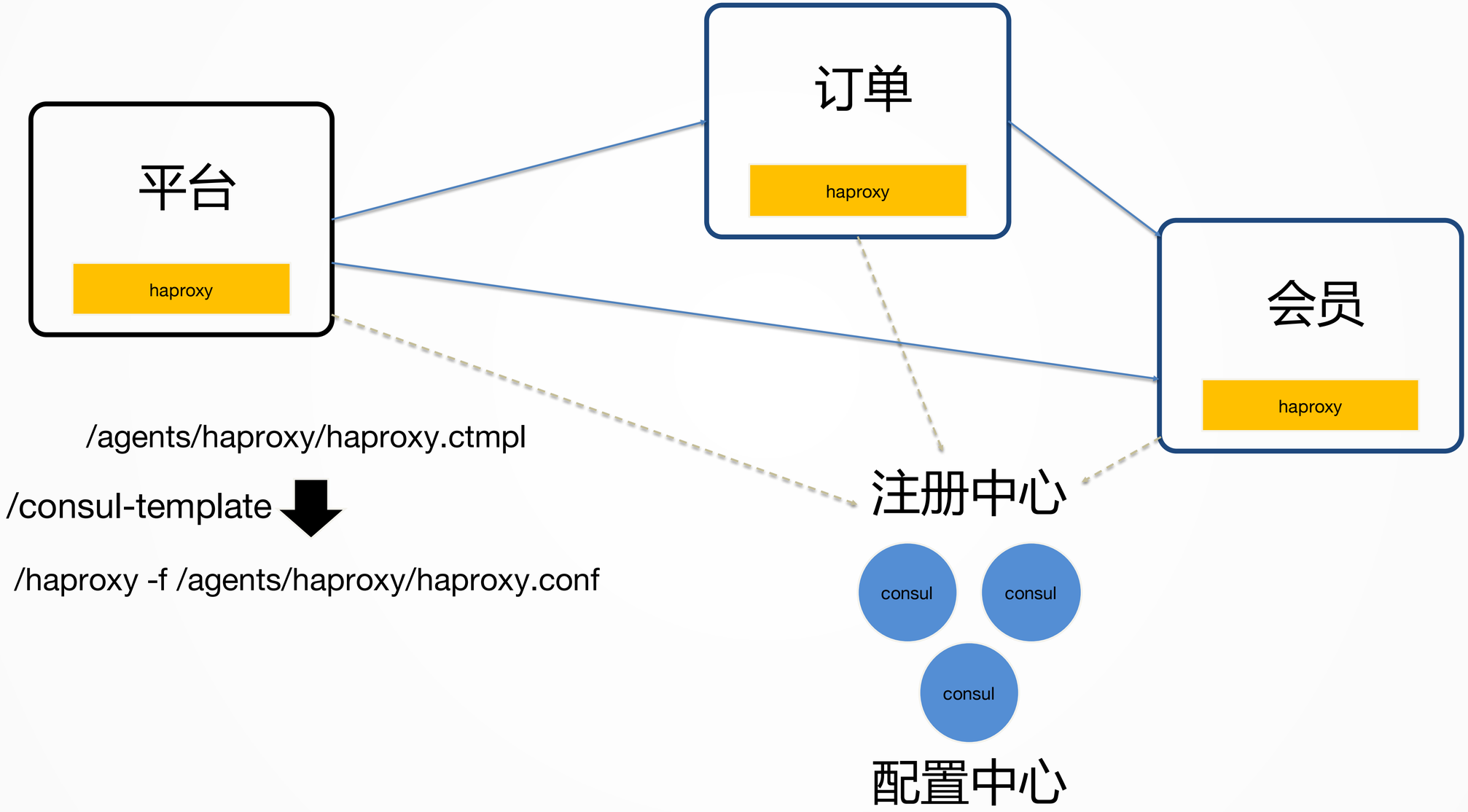


微服务十二原则

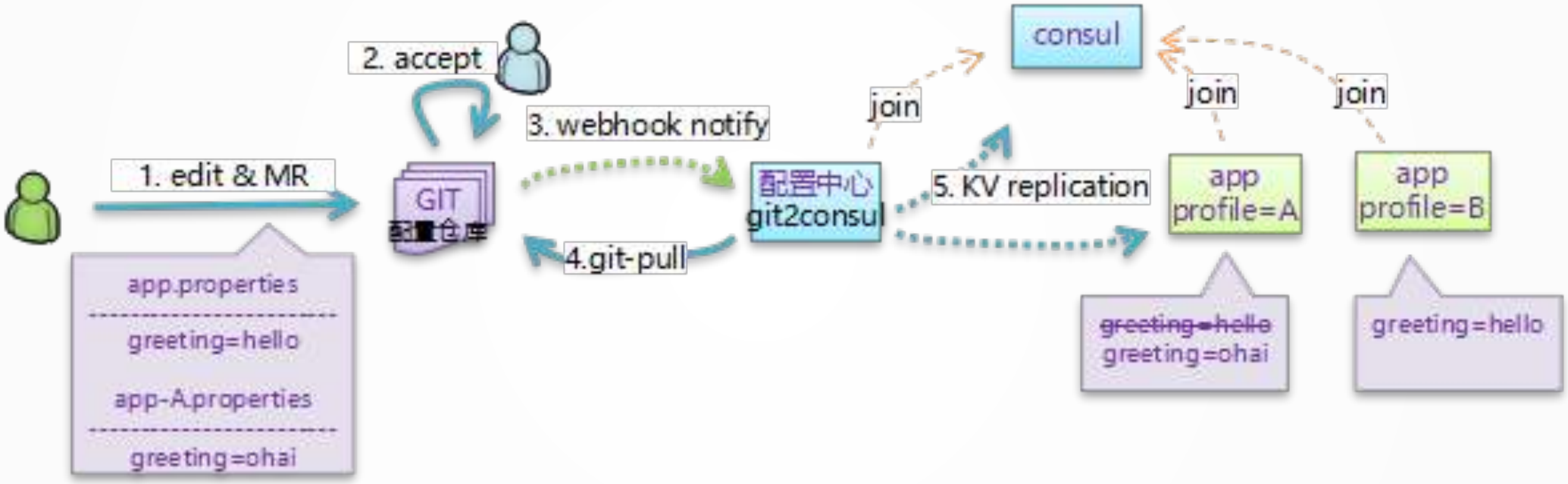

12 Factor
I. 基准代码
一份基准代码，多份部署
II. 依赖
显式声明依赖关系
III. 配置
在环境中存储配置
IV. 后端服务
把后端服务当作附加资源
V. 构建，发布，运行
严格分离构建和运行
VI. 进程
以一个或多个无状态进程运行应用
VII. 端口绑定
通过端口绑定提供服务
VIII. 并发
通过进程模型进行扩展
IX. 易处理
快速启动和优雅终止可最大化健壮性
X. 环境等价
尽可能的保持开发，预发布，线上环境相同
XI. 日志
把日志当作事件流
XII. 管理进程
后台管理任务当作一次性进程运行


Application services
网站用户
负载均衡
CDN
对象存储
CDN源站
主站服务
/api/
请求路由
订单
收银台
平台
会员
后台管理
Rest
RPC
/api/
ELK
redis
(ncr)
mysql
(rds)
出站请求
管理端口
运营人员
出站网关
注册中心
服务控制台
第三方服务
支付 | URS | 微信 | 云信
GIT repositories
corpnet
配置中心
构建服务
镜像仓库
hub.c.163.com
Foundation services
1
2
3
4
5
6
7
8
9
10
11
12

Haproxy作为service mesh的Agent



Haproxy作为service mesh的Agent



新一代微服务框架设计要点

Agent热加载

兼容SpringCloud框架

Restful API

控制面租户隔离

适配VPC网络

可视化

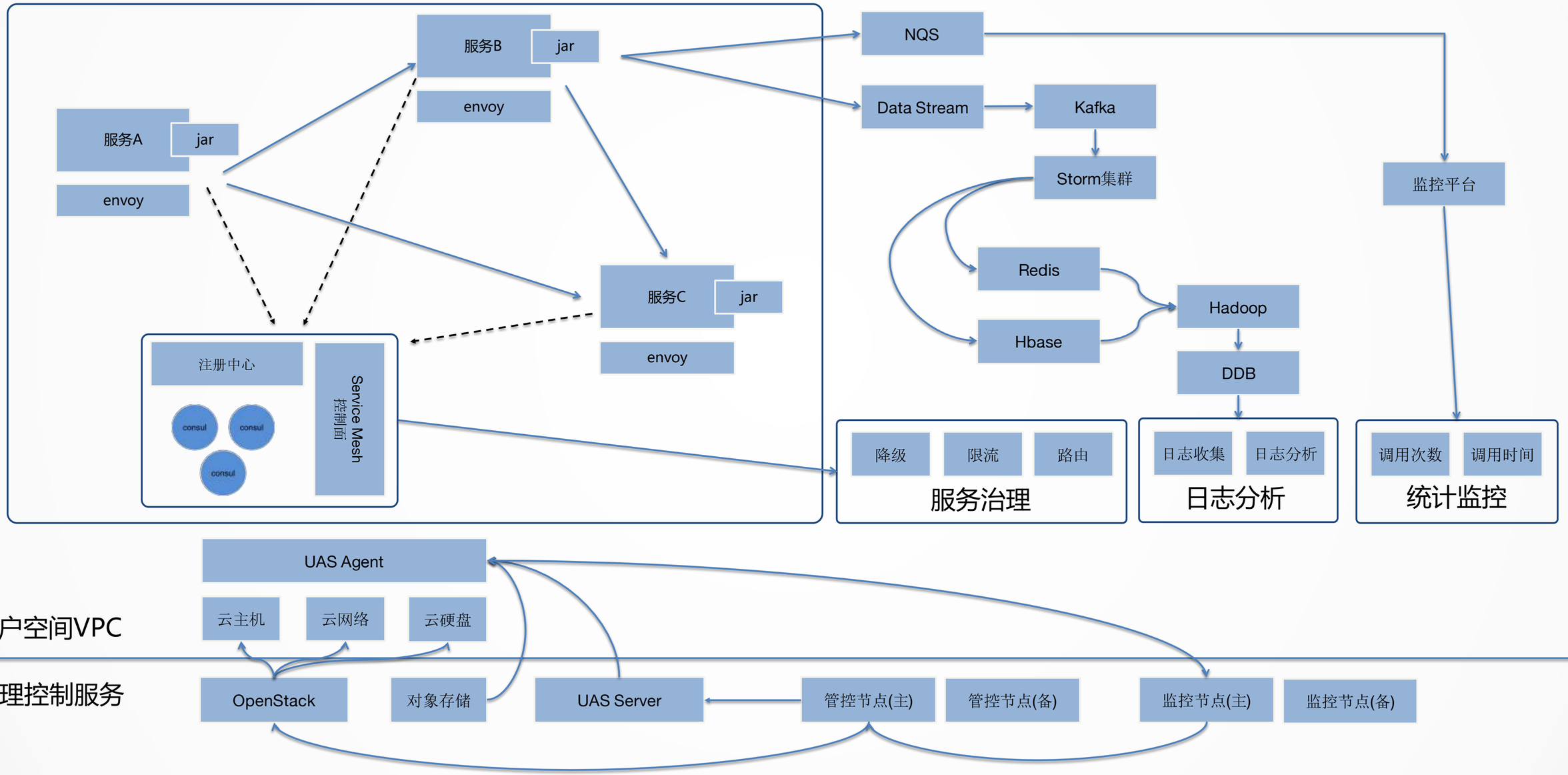
横向扩展

高可用

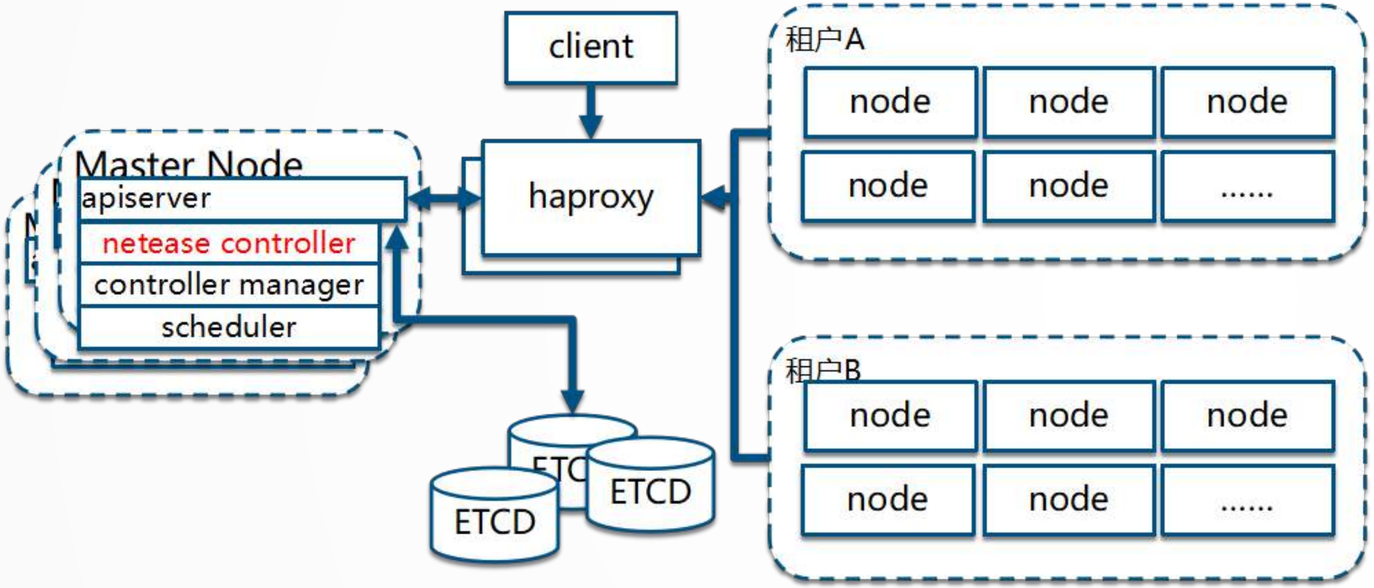
同IaaS或者Kubernetes解耦

熔断降级

新一代微服务治理平台


服务B
jar
envoy
服务A
jar
envoy
服务C
jar
envoy
注册中心
consul
consul
consul
Service Mesh
控制面
NQS
Data Stream
Kafka
Storm集群
Redis
Hbase
Hadoop
DDB
监控平台
降级
限流
路由
服务治理
日志收集
日志分析
日志分析
调用次数
调用时间
统计监控
UAS Agent
云主机
云网络
云硬盘
用户空间VPC
管理控制服务
OpenStack
对象存储
UAS Server
管控节点(主)
管控节点(备)
监控节点(主)
监控节点(备)

与VPC集成实现租户隔离



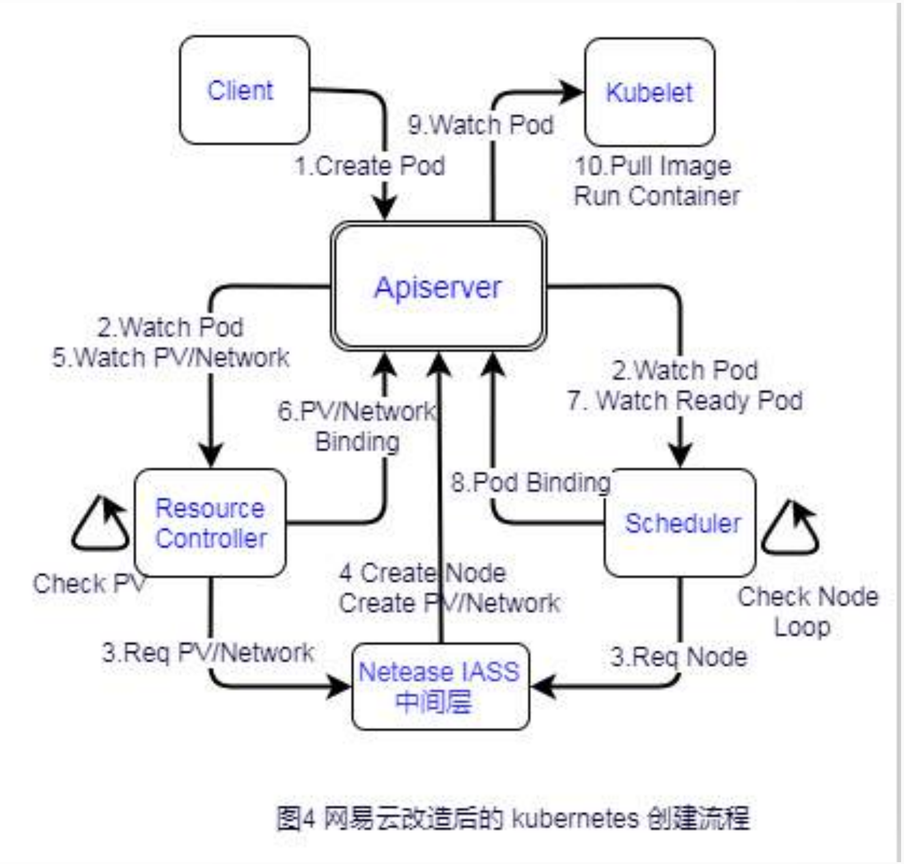
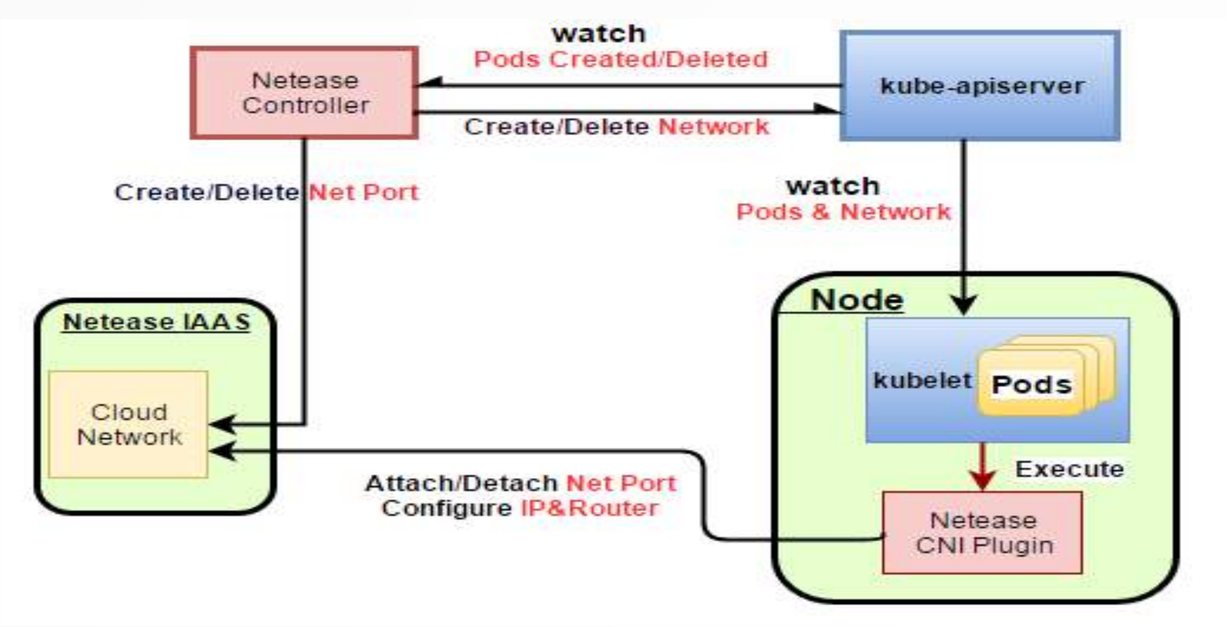
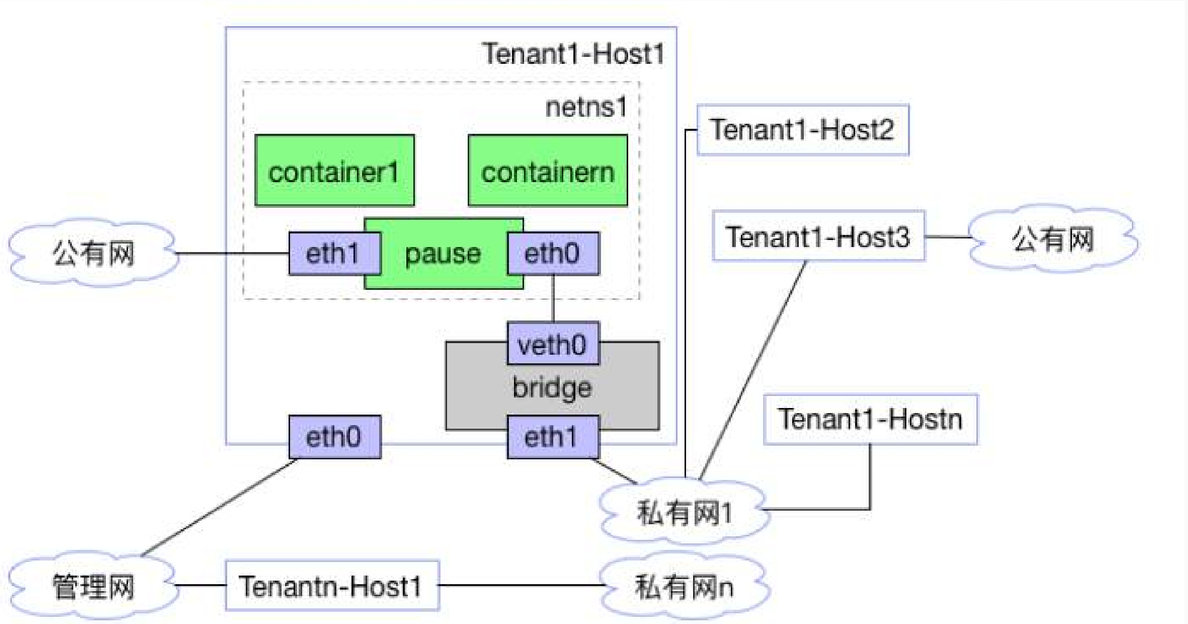


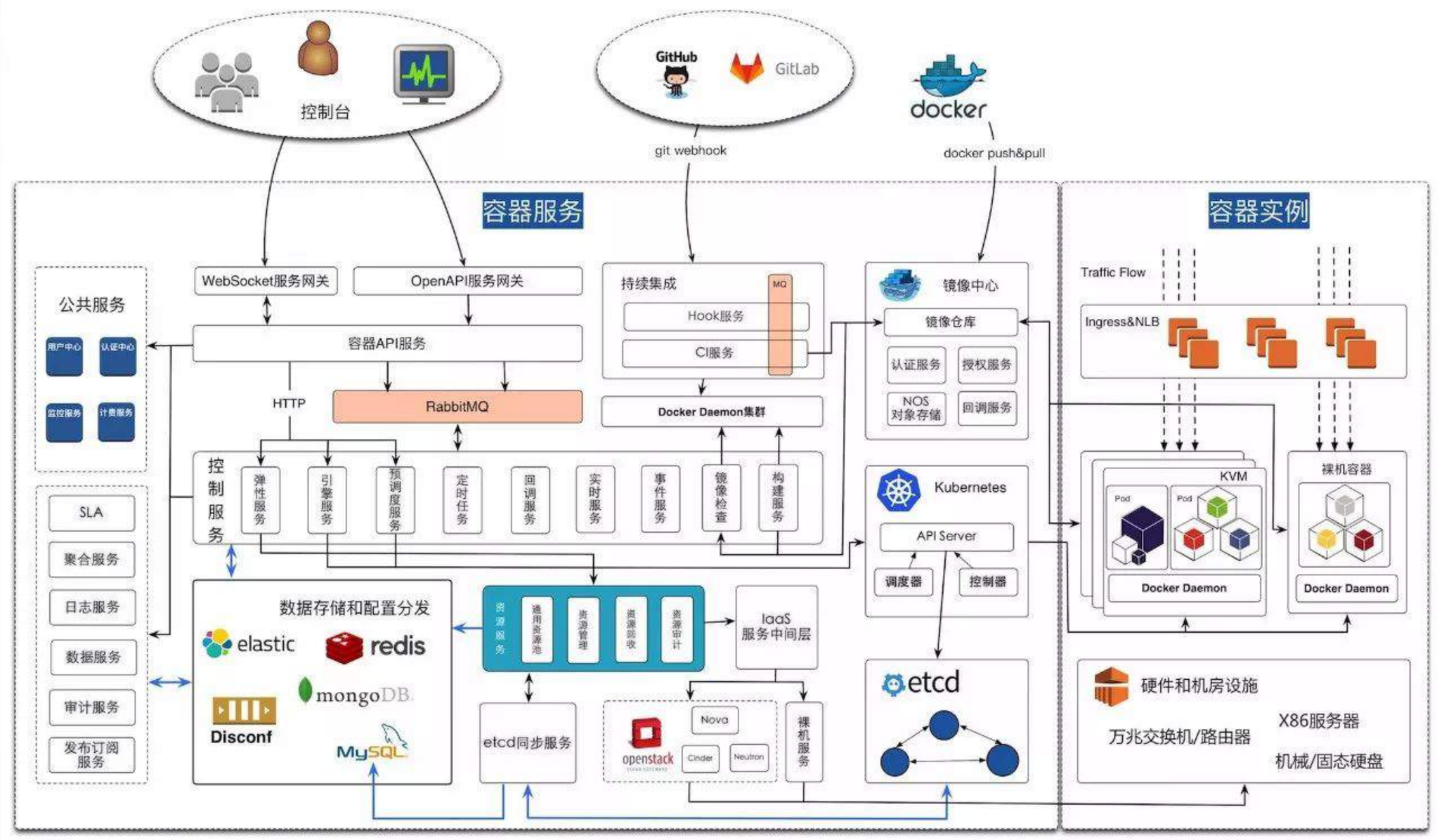
图4 网易云改造后的 kubernetes 创建流程

与VPC集成实现租户隔离





基于Service Mesh的容器管理平台



容器管理平台本身也是微服务

所有的多租户容器请求入口流量

高可用，横向扩展

对接多个业务：OpenStack，Kubernetes，所有PaaS，持续集成，镜像仓库，计费，用户，认证，……

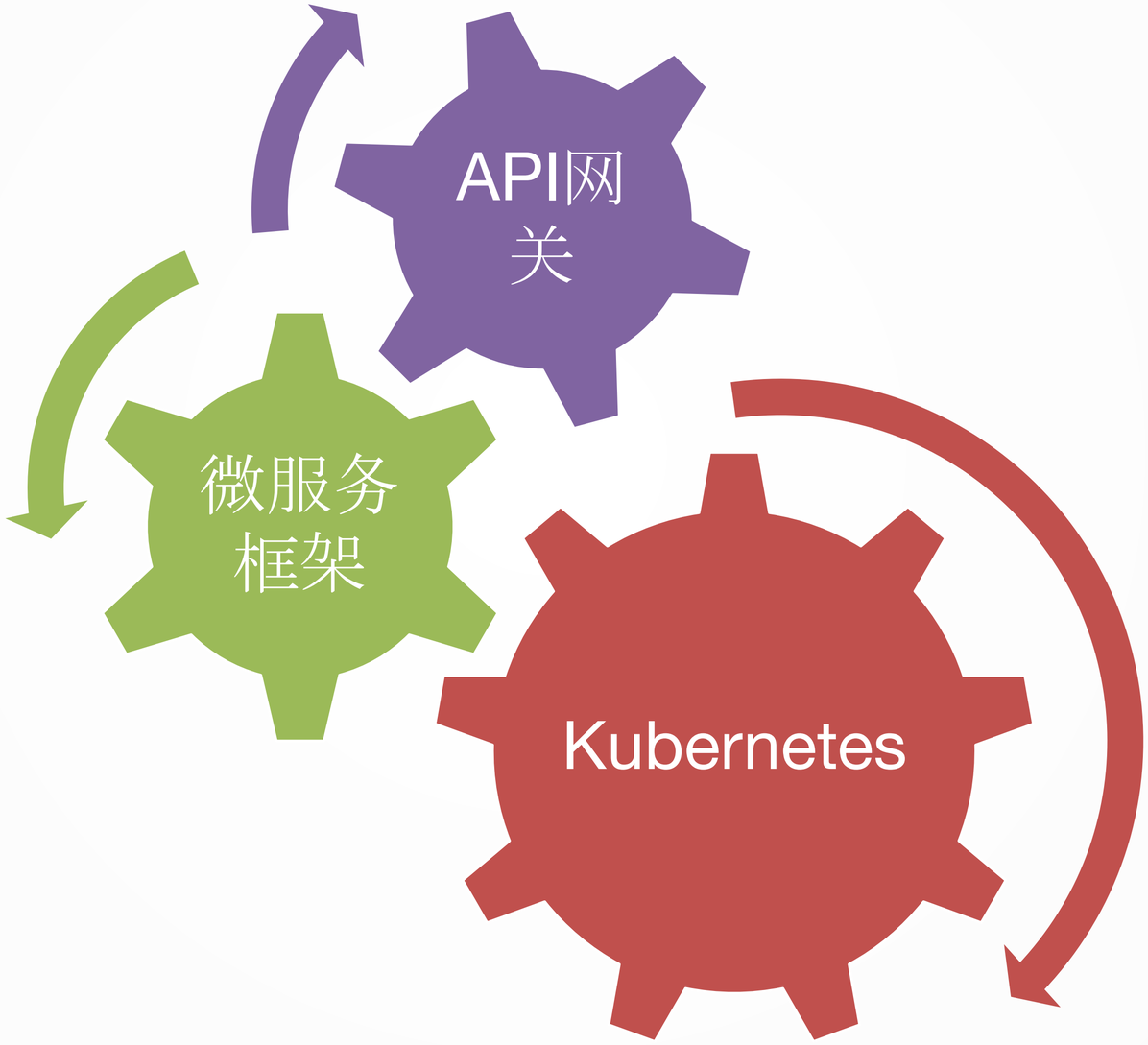
负载均衡，路由

熔断，限流，降级

可靠消息

监控，统计

容器管理平台也是用Kubernetes部署



网易云

共创云上精彩世界

www.163yun.com