玩转 Envoy
落地自研 ServiceMesh

殷湘
思源 高级架构专家

SPEAKER

INTRODUCE

殷湘　高级架构专家

- 2018.01 - present 思源
- 2017.04 - 2018.01 华为
- Apache ServiceComb committer

TABLE OF

CONTENTS 大纲

- 背景
- 拼：Envoy 工作原理
- 满：Why Envoy?
- 借：自研ServiceMesh

服务化带来的挑战

分 - 服务设计与拆分	**管** - 多服务运维管理	**控** - 不稳定的网络
• 服务耦合度 • 业务复杂度 • 业务架构与系统架构统一	• 部署升级 • 监控、调用链追踪 • 灰度发布 • 网关 • 日志 • 动态配置	• 服务发现 • 服务路由 • 超时延迟重试 • 数据一致性

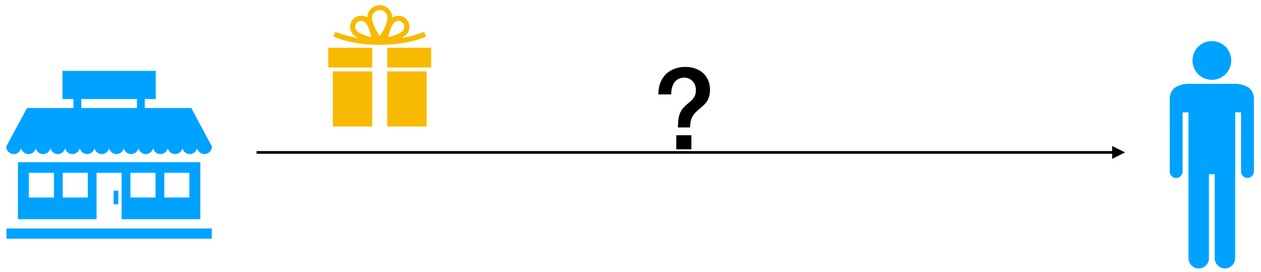
诸多问题与网络相关

分 - 服务设计与拆分	**管** - 多服务运维管理	**控** - 不稳定的网络
• 服务耦合度 • 业务复杂度 • 业务架构与系统架构统一	• 部署升级 • 监控、调用链追踪 • 灰度发布 • 网关 • 日志 • 动态配置	• 服务发现 • 服务路由 • 超时延迟重试 • 数据一致性

理想的解决方案

全	易	广
解决所有与网络相关的服务治理问题 • 监控 • 调用链追踪 • 灰度发布 • 网关 • 服务发现 • 服务路由 • 超时延迟重试	节省业务团队集成成本 • 少改 (最好不改) 现有代码 • 升级对业务影响小 (无影响) • 学习、集成门槛低 • 可拔插	支持多语言 • Java • C++ • C# • Go • Python • Nodejs

超市的送货问题

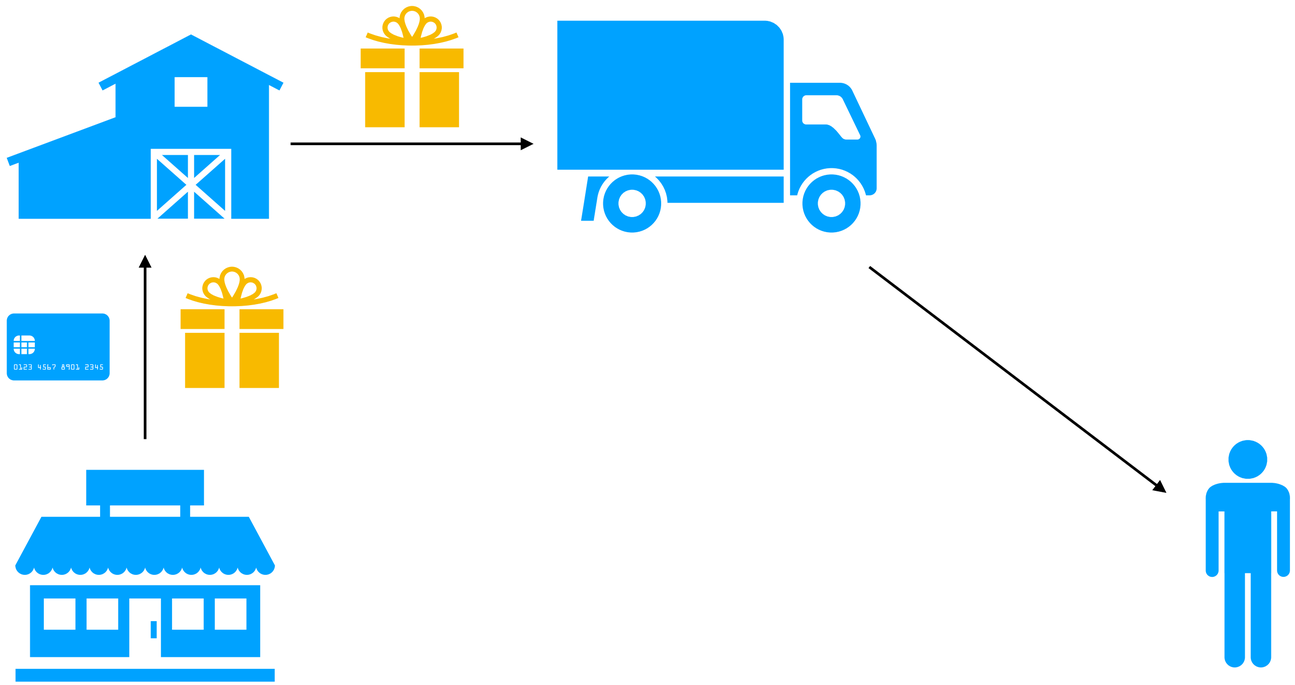


自建 vs 代理

自建

送货业务成为公司的**一部分**

代理



公司只**专注**超市业务

假如

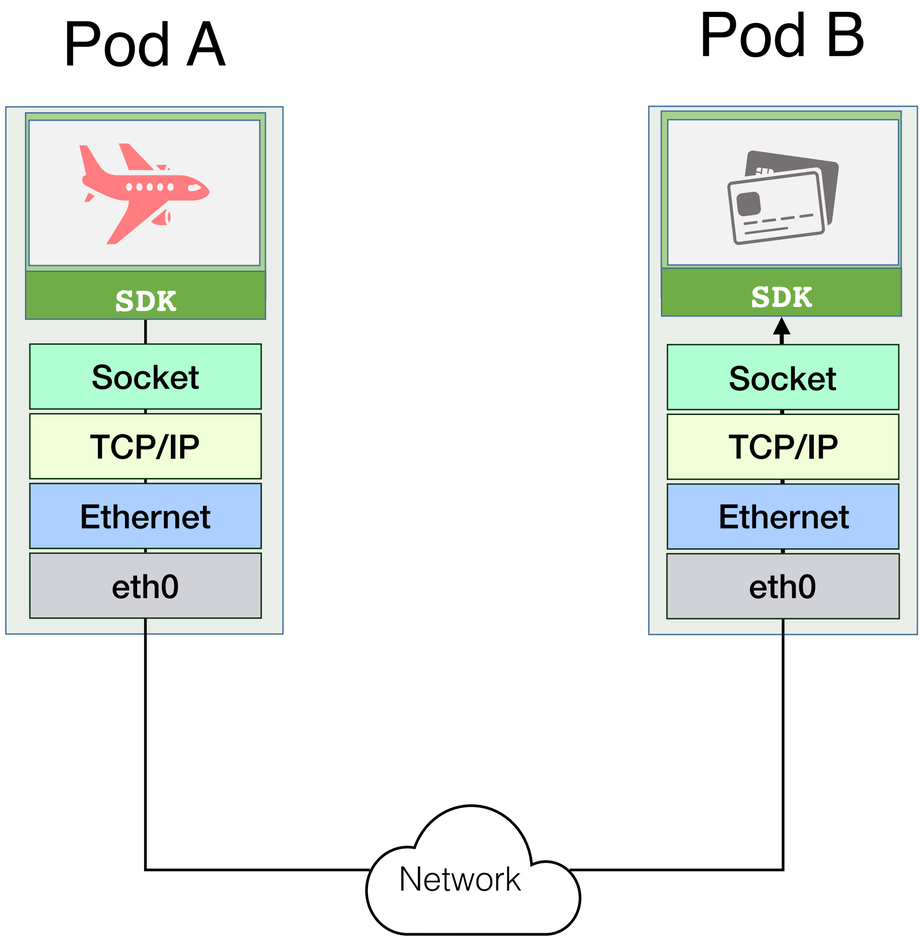
业务团队 = 超市运营

送货问题 = 服务治理

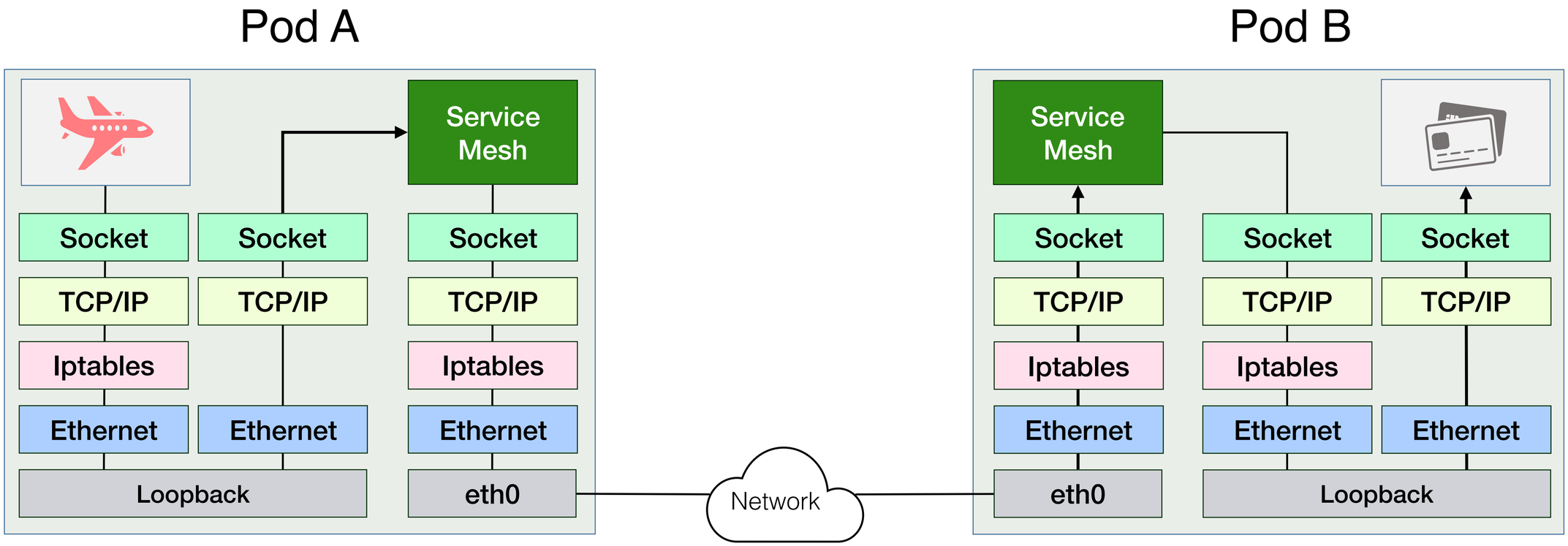
SDK模式：自建

- 以SDK的形式嵌入到服务进程中

SDK成为所有业务团队的问题



服务网格：代理 - 业务无感知



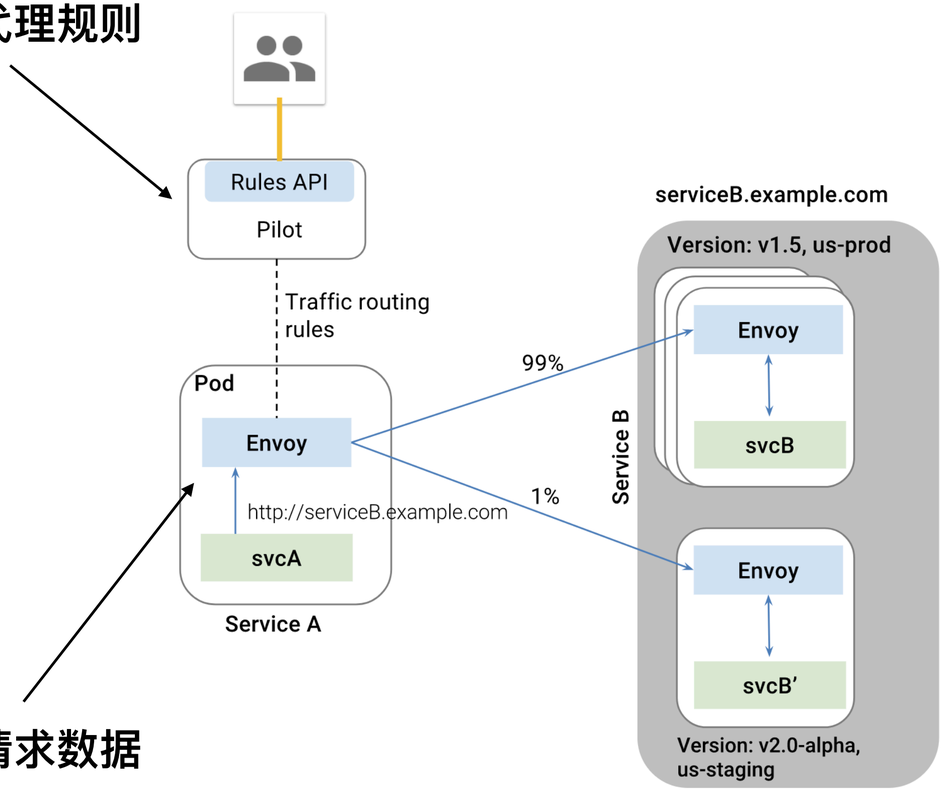
服务网格由专门技术团队负责，业务团队只专注业务

理想的解决方案

全	易	广
解决所有与网络相关的服务治理问题 • 监控 • 调用链追踪 • 灰度发布 • 网关 • 服务发现 • 服务路由 • 超时延迟重试	节省业务团队集成成本 • 少改 (最好不改) 现有代码 • 升级对业务影响小 (无影响) • 学习、集成门槛低 • 可拔插	支持多语言 • Java • C++ • C# • Go • Python • Nodejs

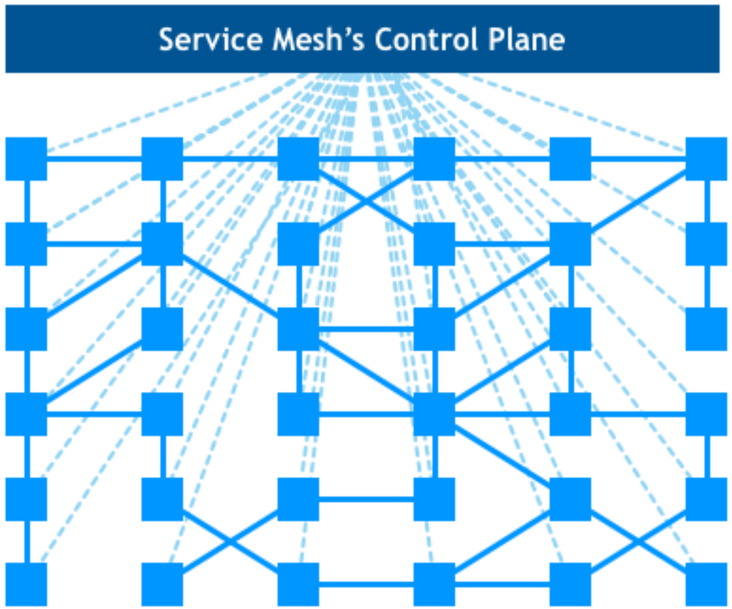
ServiceMesh = 控制面 + 数据面

管理代理规则



代理请求数据

集中管控整个系统网络

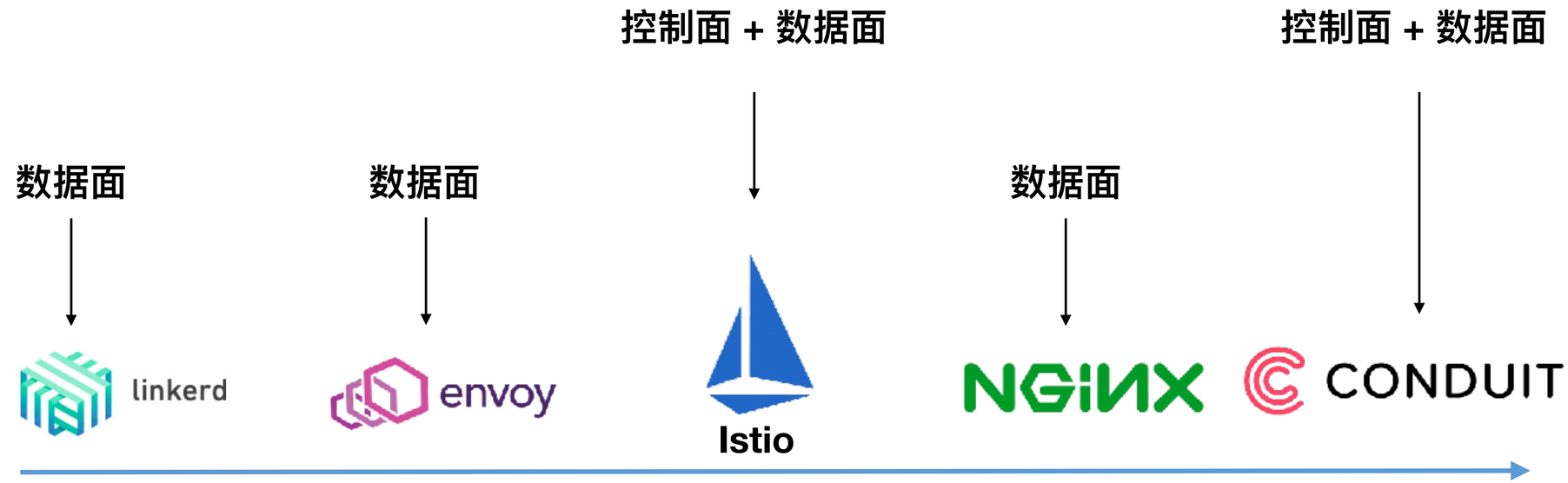


微观

宏观

http://philcalcado.com/2017/08/03/pattern_service_mesh.html

服务网格的演进



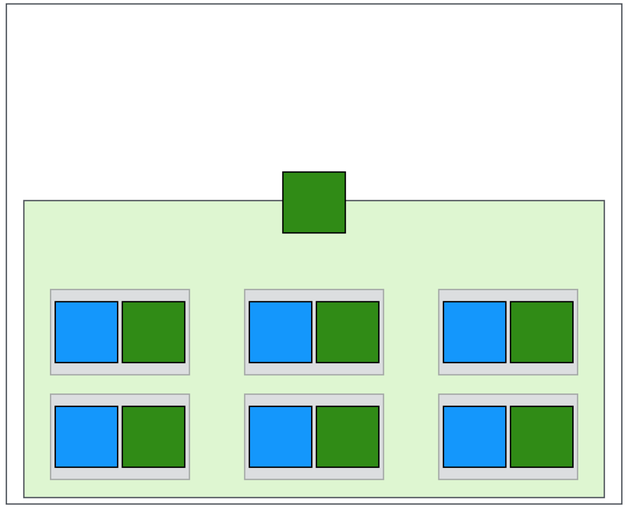
可商用的控制面姗姗来迟

TABLE OF

CONTENTS 大纲

- 背景
- 拼：Envoy工作原理
- Why Envoy
- 自研ServiceMesh

Envoy



服务网关/服务代理

7	application	HTTP
6	presentation	
5	session	
4	transport	TCP
3	network	IP
2	data-link	
1	physical	

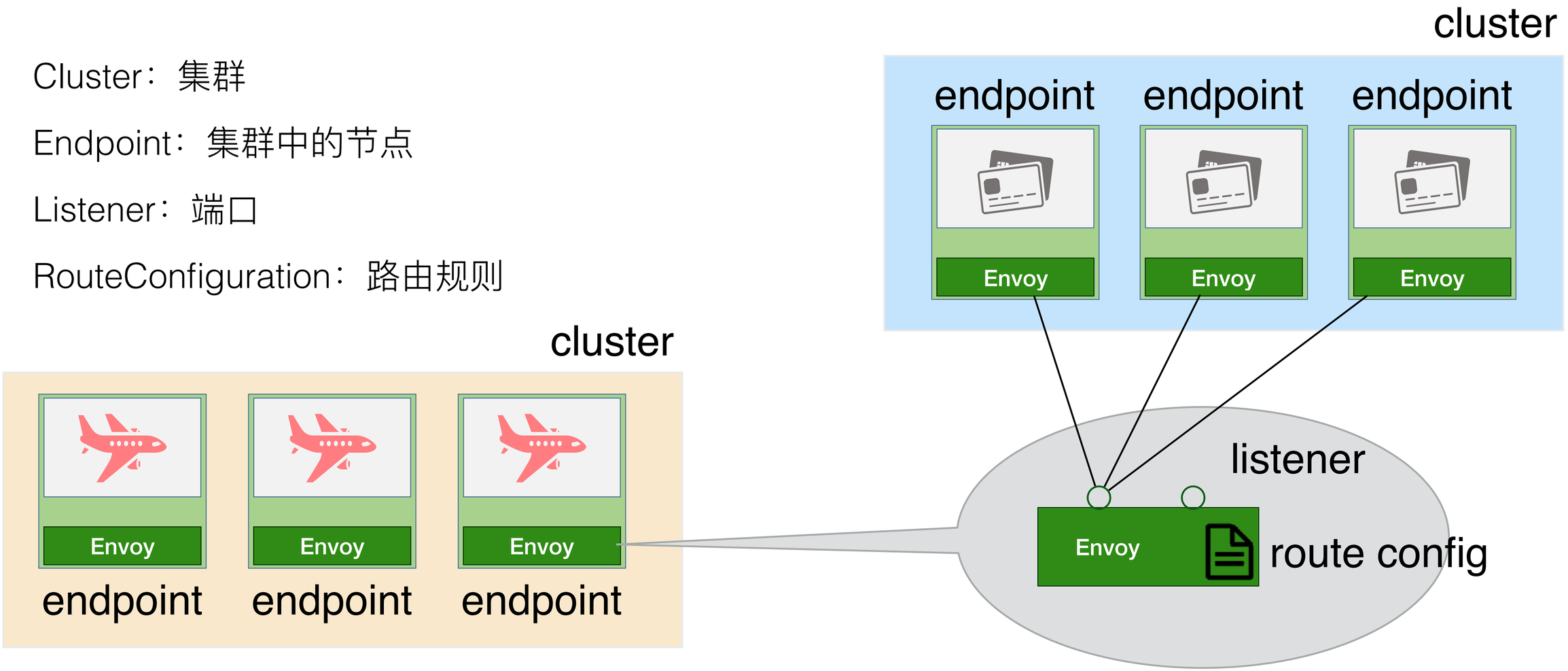
L3/4/7 proxy
支持SSL
HTTP2.0

服务发现	健康检查
负载均衡	网络监控
限流	调用链追踪
熔断	流量复制
重试	热重启
超时	动态路由

服务治理功能

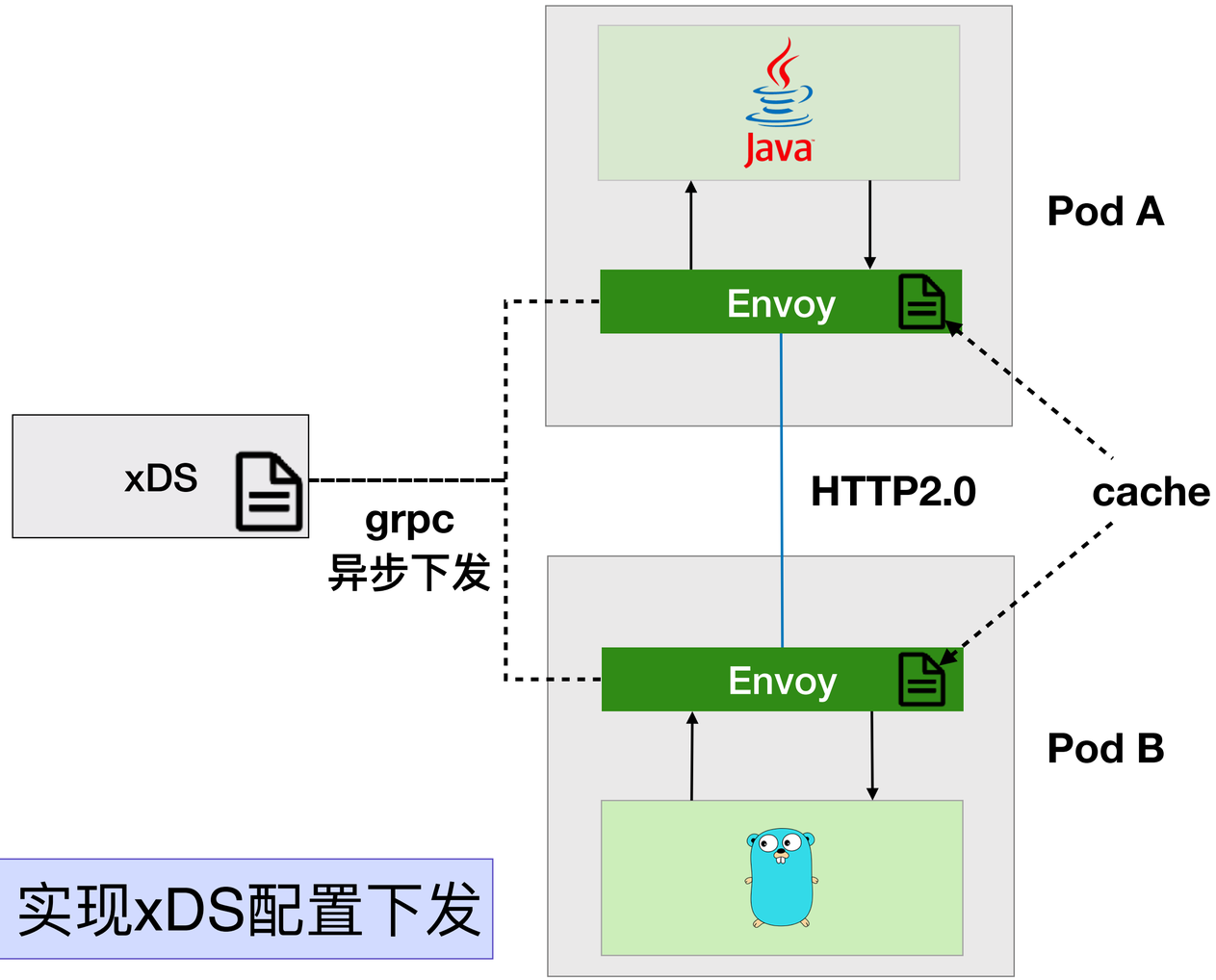
Envoy 核心概念

- Cluster：集群
- Endpoint：集群中的节点
- Listener：端口
- RouteConfiguration：路由规则



Envoy 动态配置

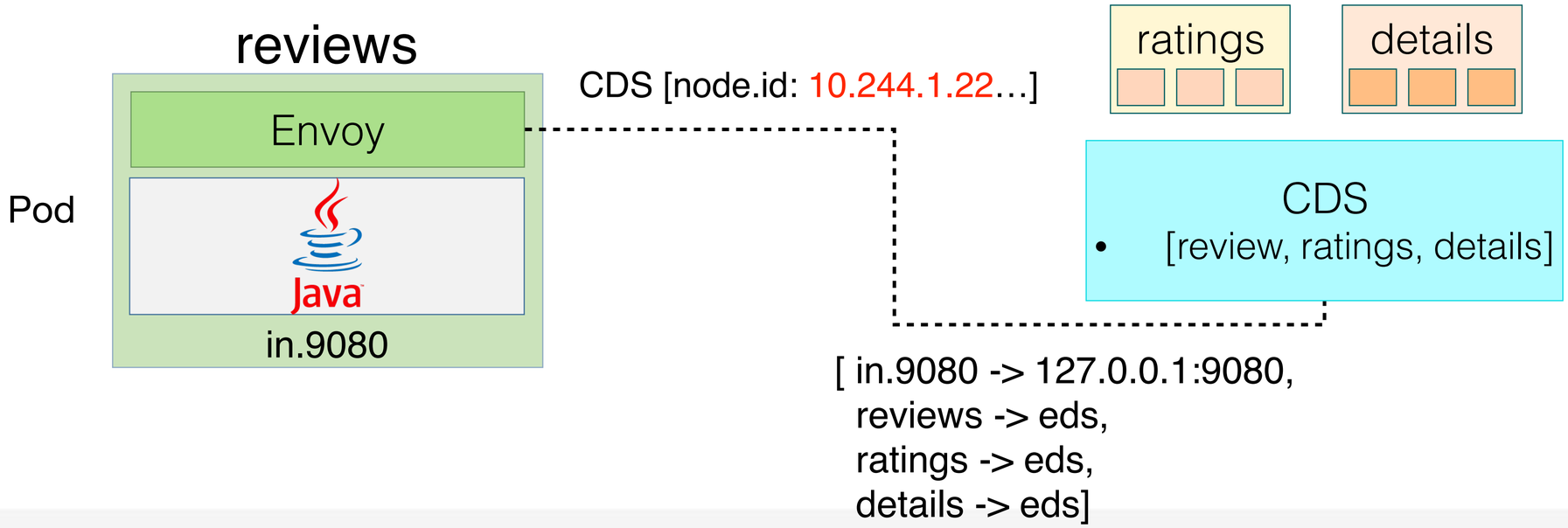
- xDS: x discovery service
  - CDS: Clusters DS
  - EDS: Endpoints DS
  - LDS: Listeners DS
  - RDS: RouteConfigurations DS
- ADS = LDS + LDS + CDS + EDS



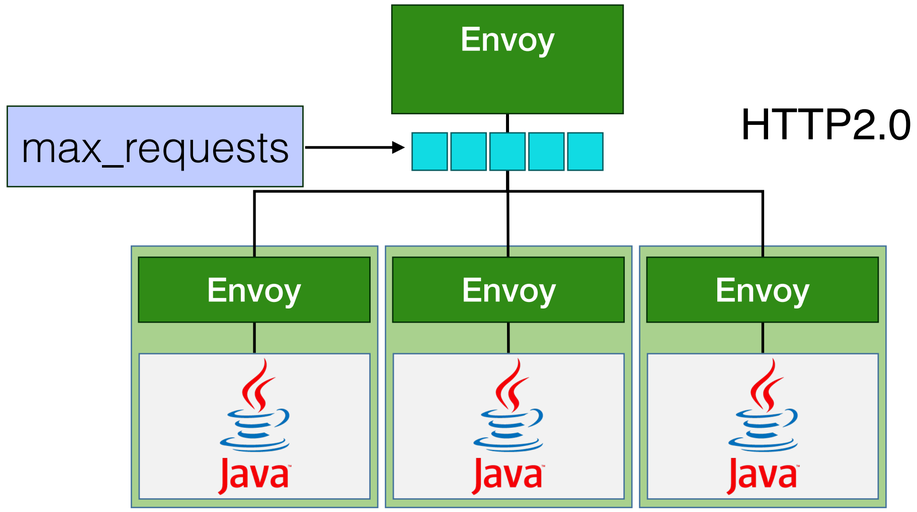
实现控制面 = 实现xDS配置下发

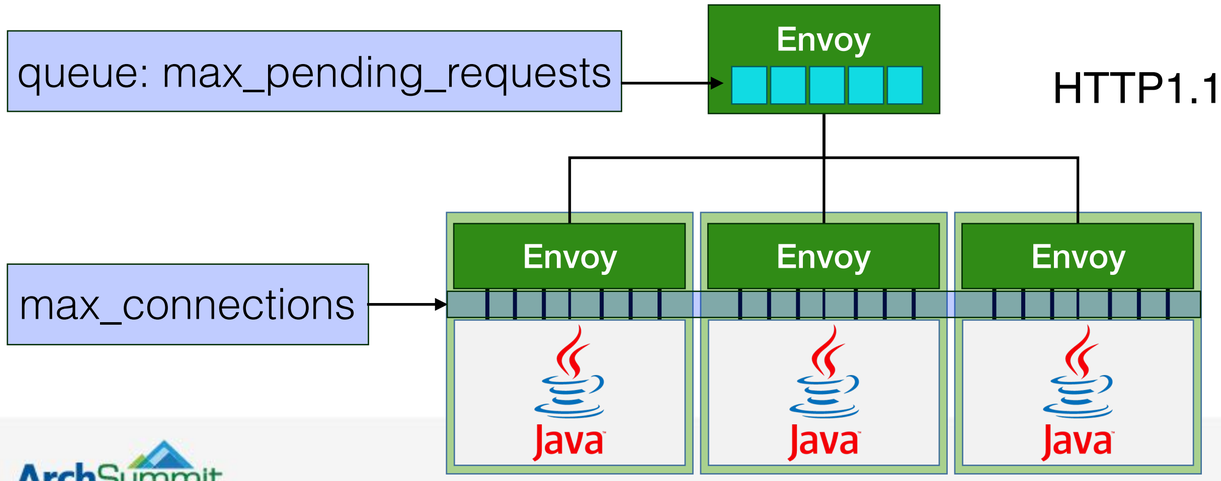
CDS：集群信息

- Local cluster
  - in.9080
- Remote clusters
  - reviews
  - ratings
  - details



CDS示例：熔断

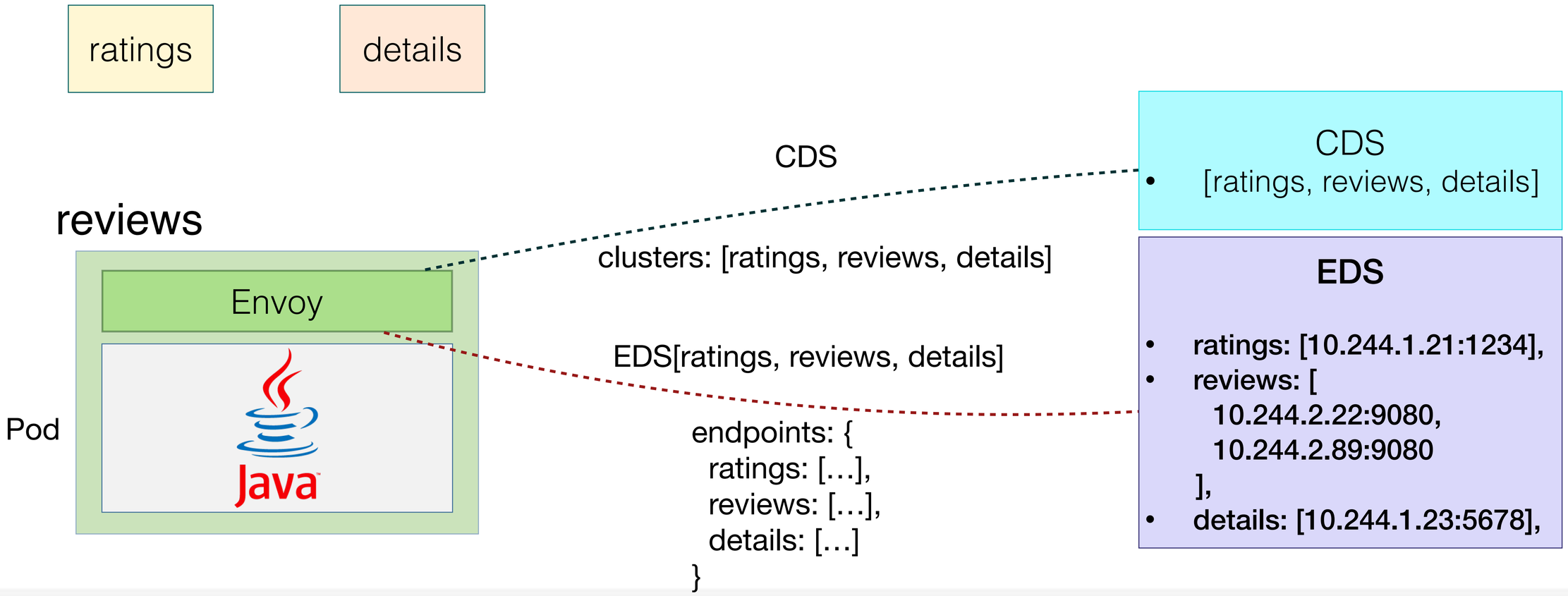




CDS

```
"name": “reviews",
"circuit_breakers": {
   "default": {
     "max_connections": 1024,
     "max_pending_requests": 1024,
     "max_requests": 1024,
     "max_retries": 3
   }
}
type: EDS
eds_cluster_config {
  eds_config {
    ads {
    }
  }
  service_name: “reviews"
}
```

EDS：集群节点信息

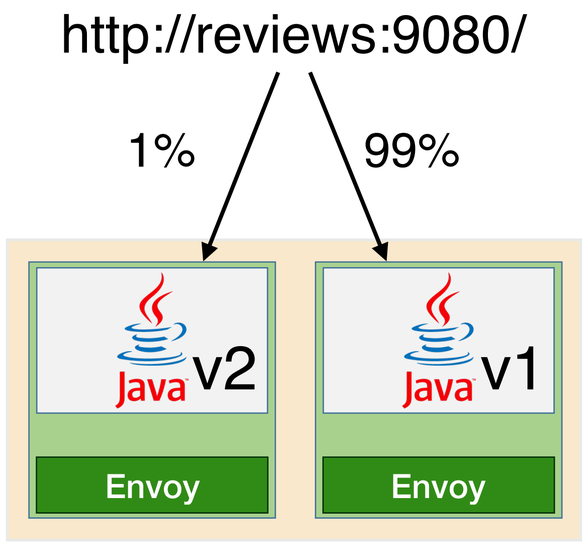

ratings
details
reviews
Envoy
Java
Pod
CDS
clusters: [ratings, reviews, details]
EDS[ratings, reviews, details]
endpoints: {
ratings: […],
reviews: […],
details: […]
}
CDS
• [ratings, reviews, details]
EDS
• ratings: [10.244.1.21:1234],
• reviews: [
10.244.2.22:9080,
10.244.2.89:9080
],
• details: [10.244.1.23:5678],

EDS示例：灰度发布

负载比例计算 (如不考虑AZ)

v1 lb% = weight_v1 / (weight_v1 + weight_v2)

v2 lb% = weight_v2 / (weight_v1 + weight_v2)

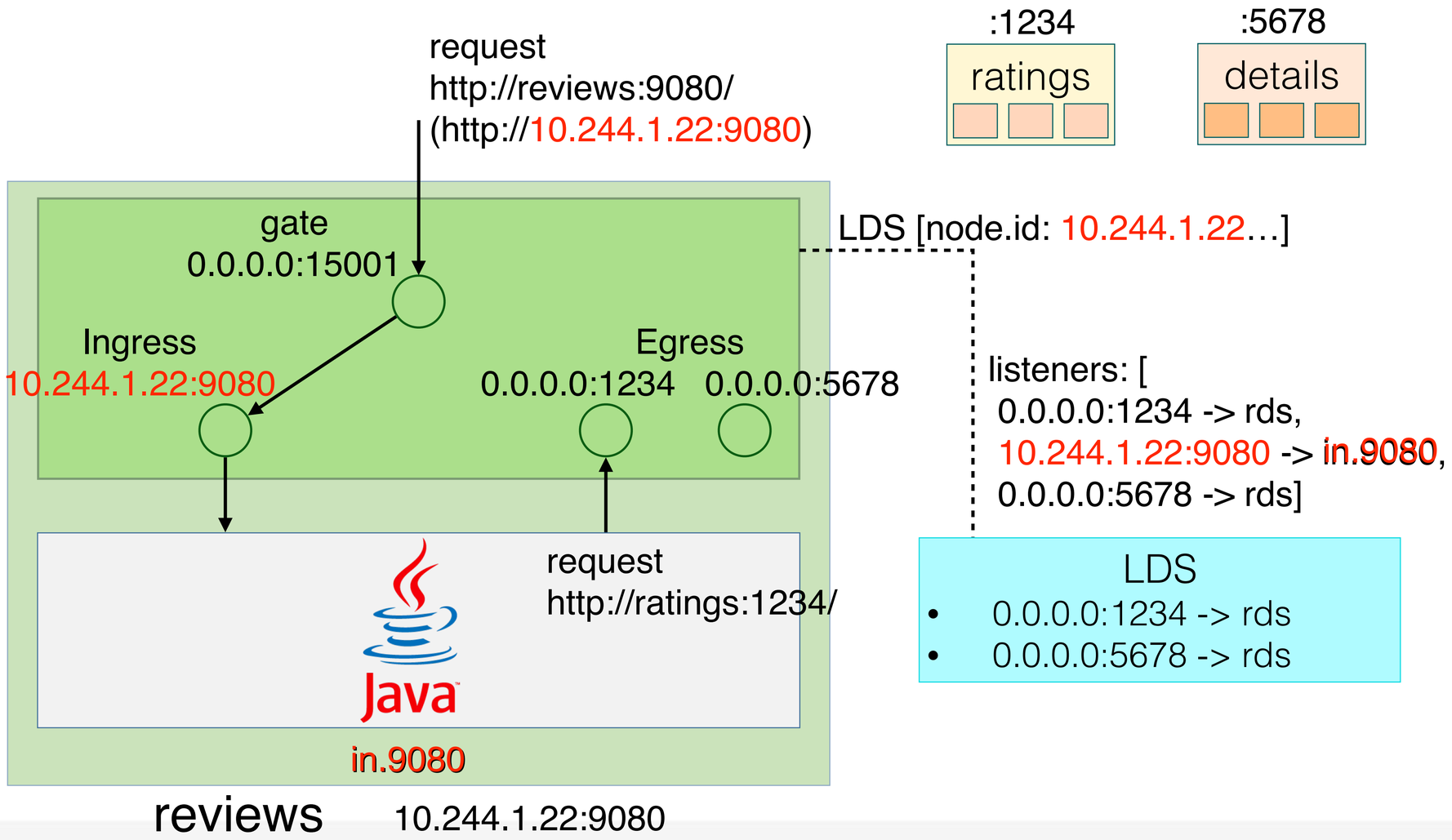


cluster: reviews

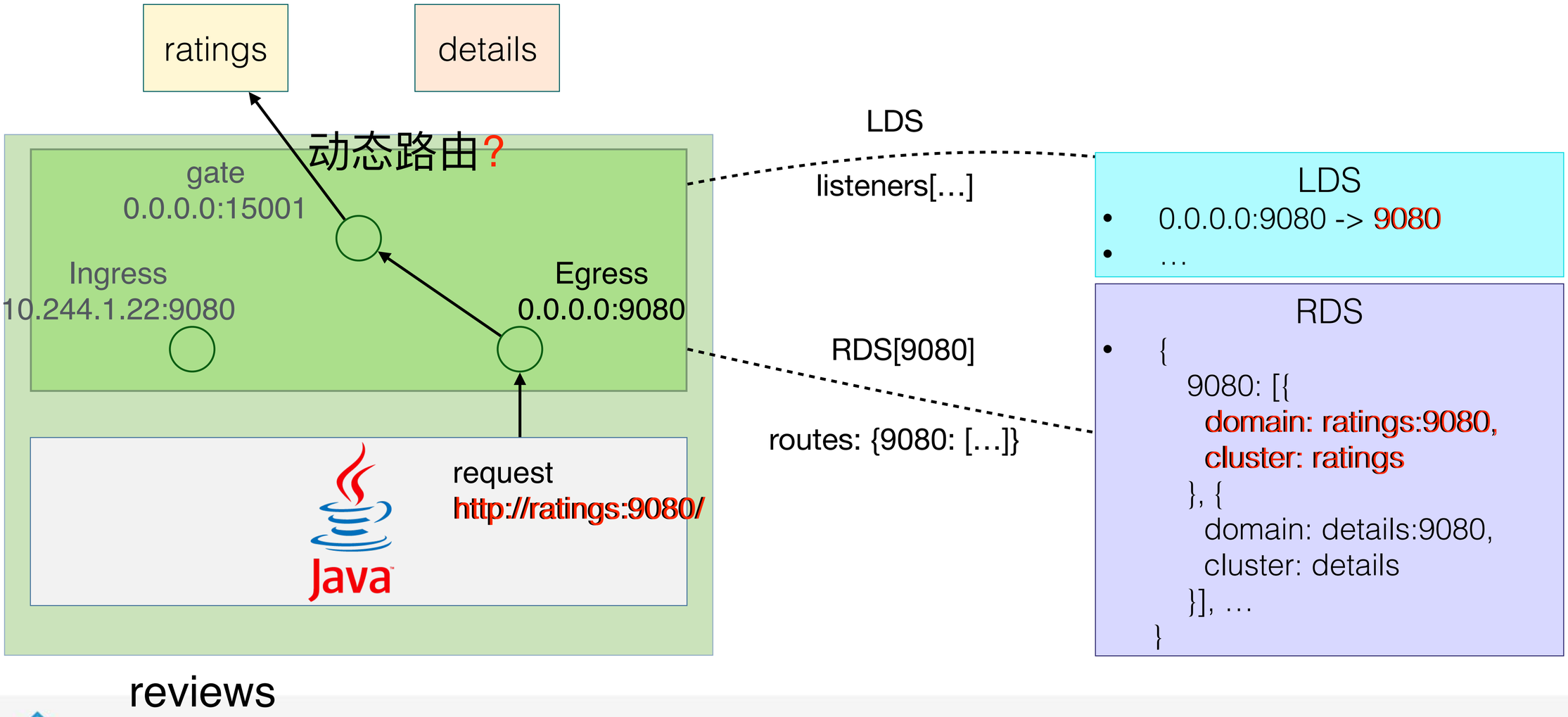
```
cluster_name: "reviews"
endpoints {
  lb_endpoints {
    endpoint {
      address {
        socket_address {
          address: "10.244.1.21"
          port_value: 9080
        }
      }
    }
    load_balancing_weight {
      value: 1
    }
  }
  lb_endpoints {
    endpoint {
      address {
        …
      }
    }
    load_balancing_weight {
      value: 99
    }
  }
```

LDS

- gate listener
  - 0.0.0.0:15001
- ingress listeners
  - pod_ip:endpoint_port
- egress listeners
  - 0.0.0.0:endpoint_port



RDS：路由规则


ratings
details
动态路由?
gate
0.0.0.0:15001
Ingress
10.244.1.22:9080
Egress
0.0.0.0:9080
request
http://ratings:9080/
Java
reviews
LDS
listeners[…]
LDS
• 0.0.0.0:9080 -> 9080
• …
RDS[9080]
routes: {9080: […]}
RDS
• {
9080: [{
domain: ratings:9080,
cluster: ratings
}, {
domain: details:9080,
cluster: details
}], …
}

RDS示例：根据用户名路由

- 用户名 = Jason -> reviews v2

cluster: reviewslv1

cluster: reviewslv2


```
name: 9080
virtual_hosts {
  name: "reviews"
  domains: "reviews:9080"
  routes {
    match {
      prefix: "/"
      headers {
        name: "cookie"
        value: "^(.*?;)?(user=jason)(;.*)?$"
        regex {
          value: true
        }
      }
    }
    route {
      cluster: "reviews|v2"
    }
…
```

拼：请求路由流程

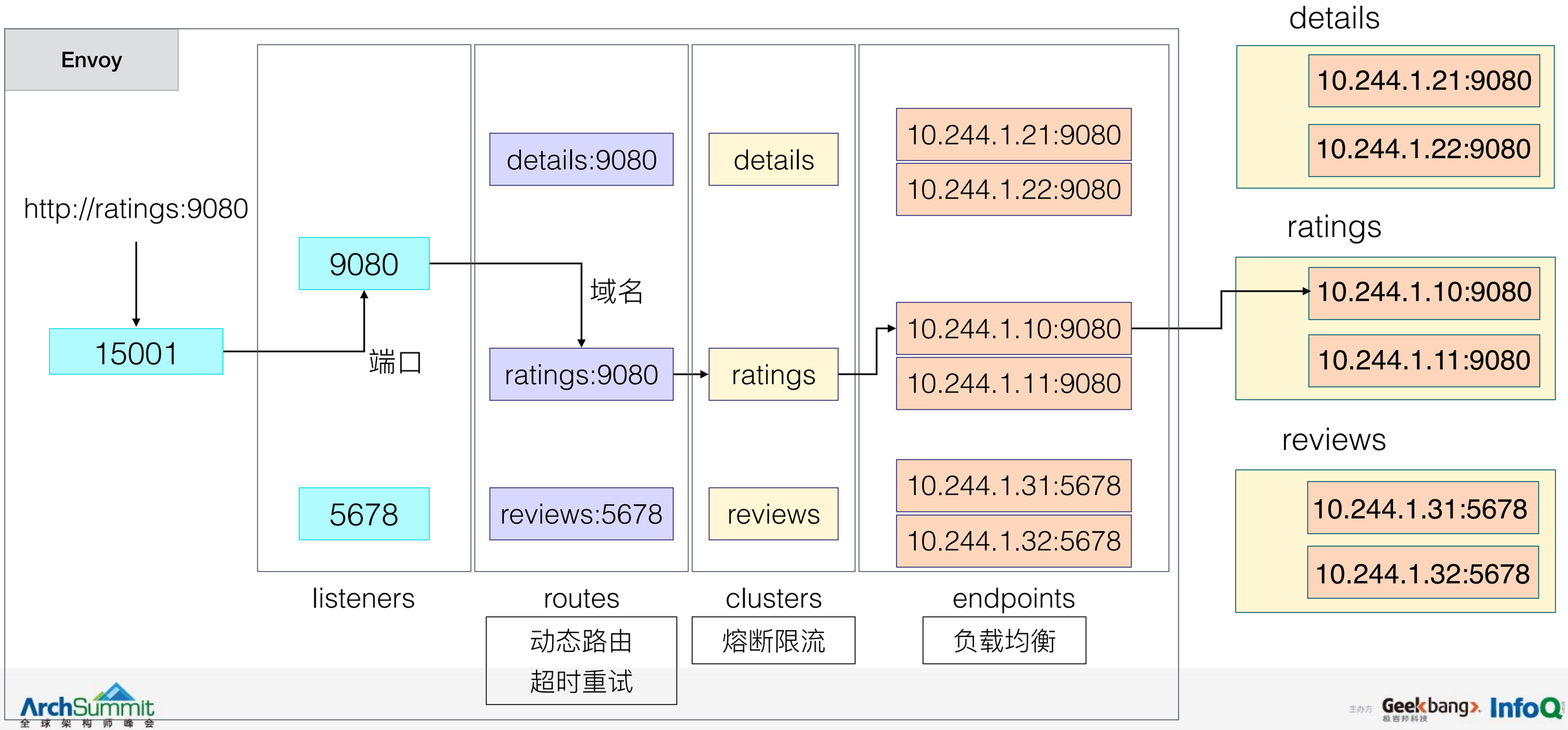

Envoy
http://ratings:9080
15001
9080
端口
域名
5678
details:9080
ratings:9080
reviews:5678
details
ratings
reviews
10.244.1.21:9080
10.244.1.22:9080
10.244.1.10:9080
10.244.1.11:9080
10.244.1.31:5678
10.244.1.32:5678
listeners
routes
clusters
endpoints
动态路由
超时重试
熔断限流
负载均衡
details
10.244.1.21:9080
10.244.1.22:9080
ratings
10.244.1.10:9080
10.244.1.11:9080
reviews
10.244.1.31:5678
10.244.1.32:5678

TABLE OF

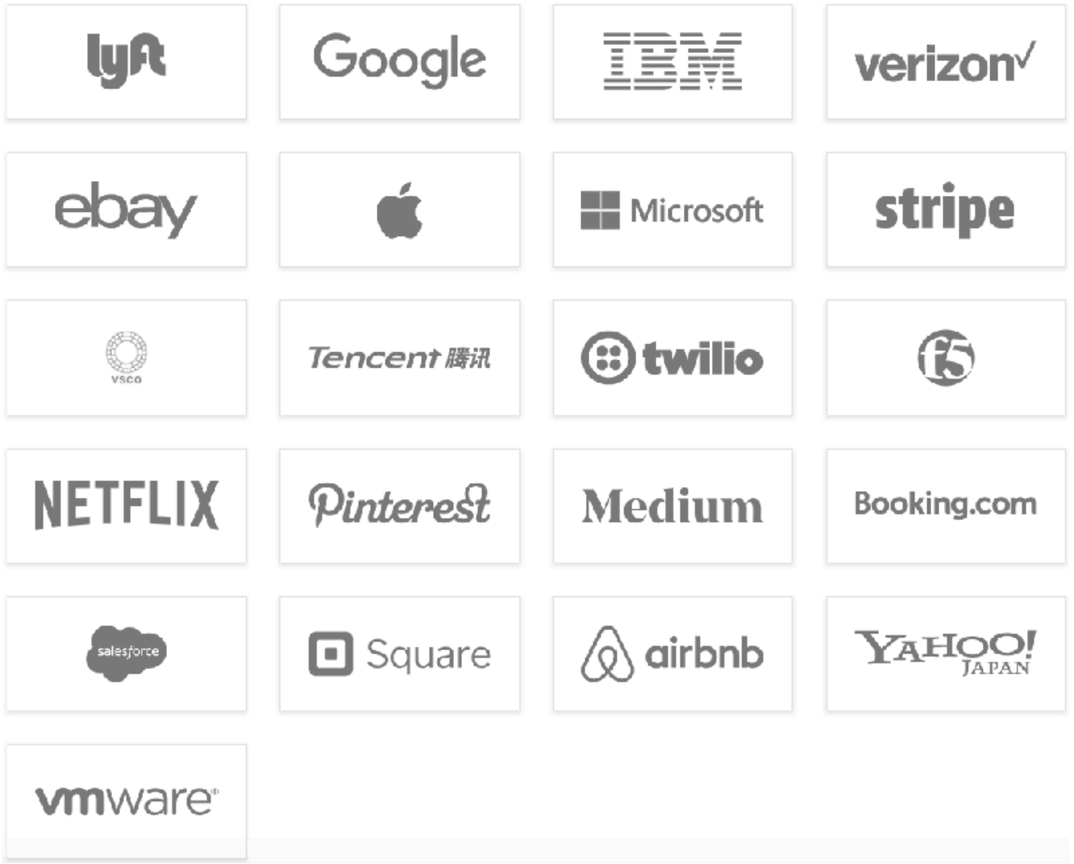
CONTENTS 大纲

- 背景
- Envoy工作原理
- 满： Why Envoy?
- 自研ServiceMesh

Envoy

实力 (@Lyft)	背书	维护者
• 管理 >100个服务 • 跨越10,000个虚拟机 • 每秒处理2百万请求 • 已加入CNCF	Istio 的默认数据面	Lyft Google Apple TurbineLabs

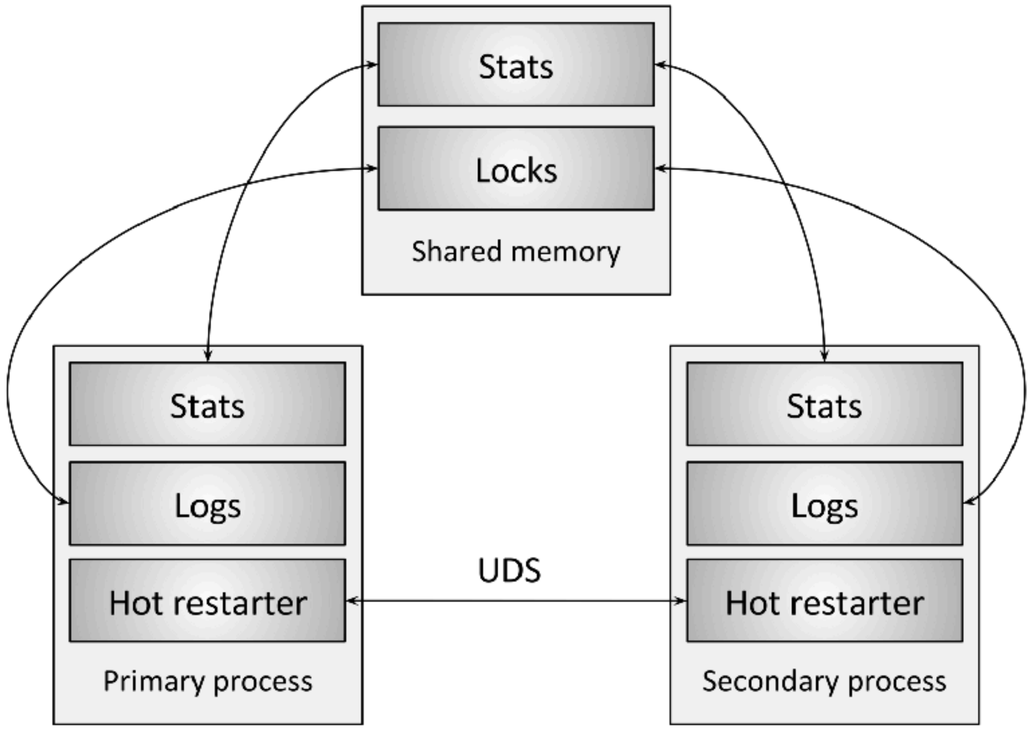
Envoy @ Global



信息来自 *https://www.envoyproxy.io/*

升级 Envoy 无需业务下线

- Envoy 的热重启保证升级时网络连接不中断，业务无感知



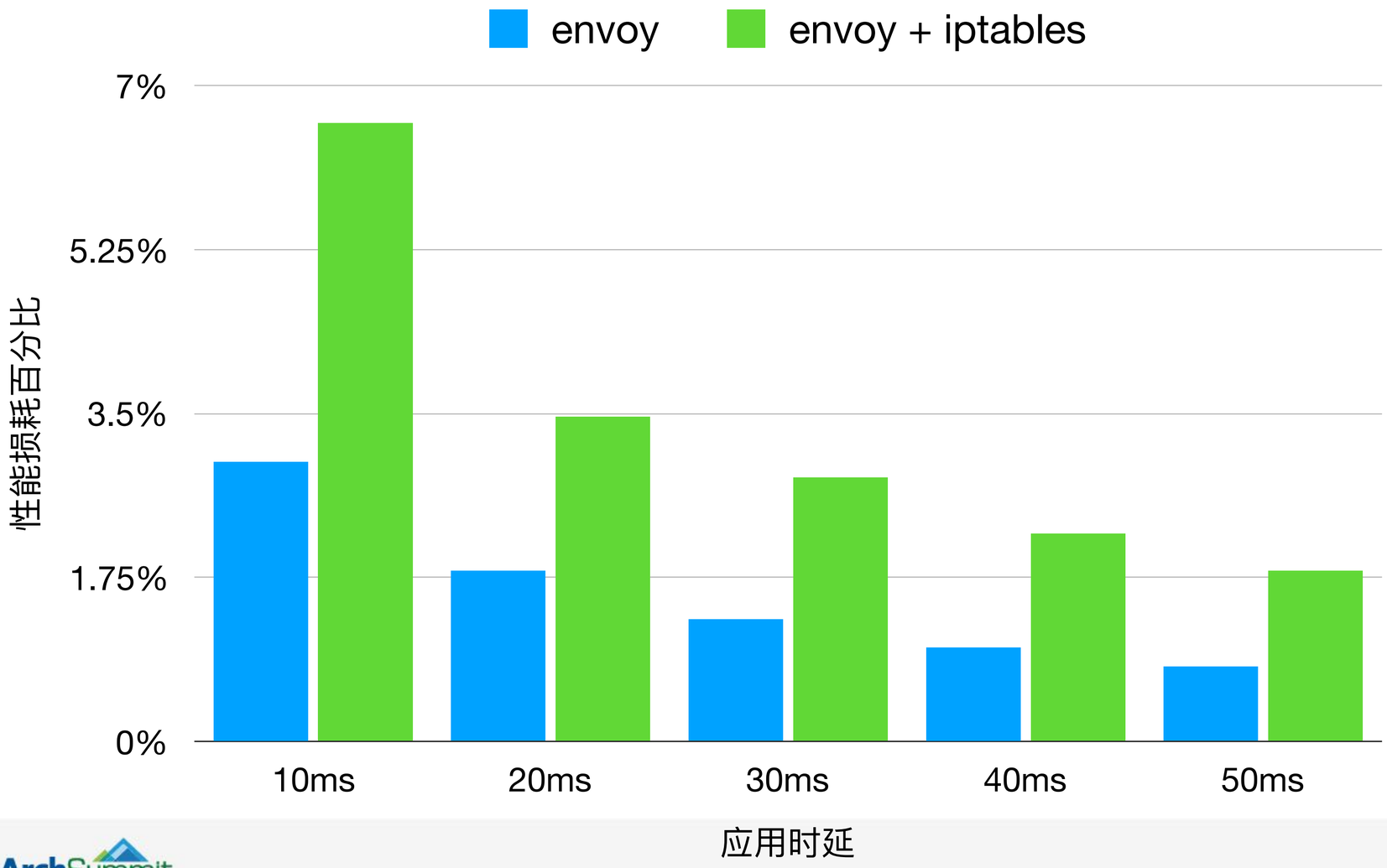
可见

- Tracing
- Metrics
- Logs

Jaeger

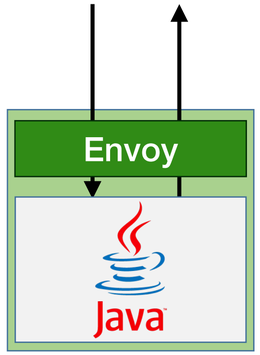
Prometheus

性能损耗



测试环境

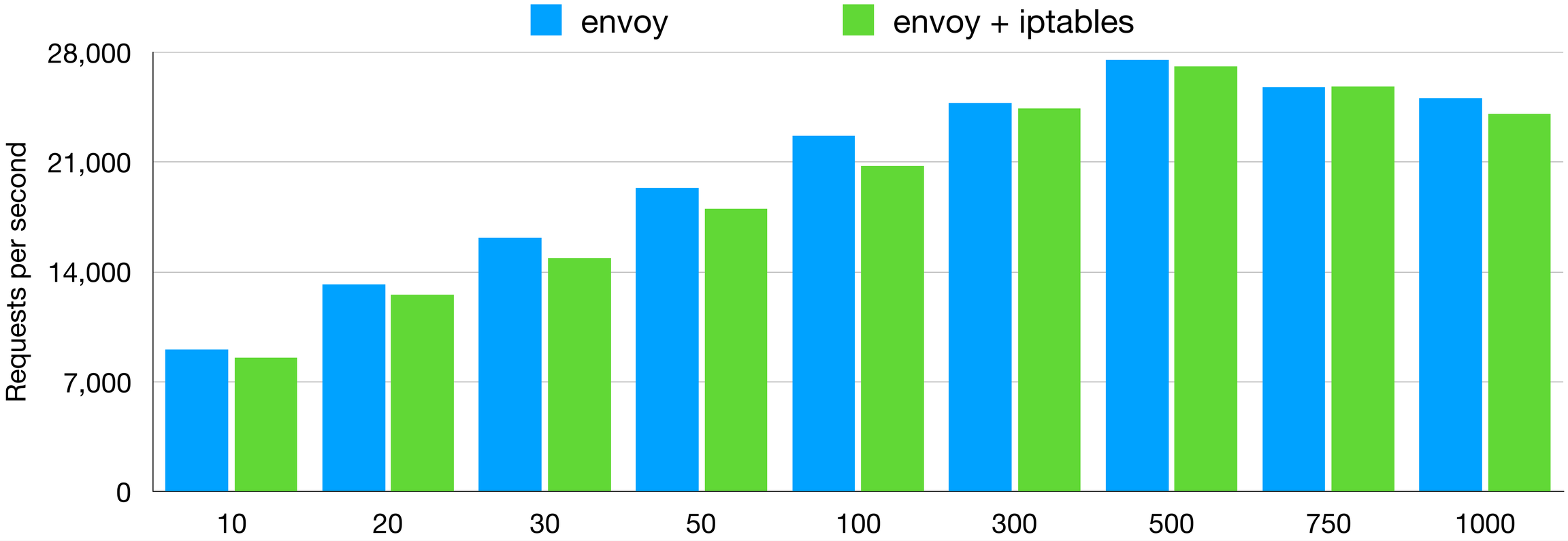
- 4C4G 虚拟机
- K8S 1.9.6
- Envoy 1.6.0



~0.4ms per envoy
~0.9ms envoy + iptables

连接数 vs RPS

- 500 连接 (~27K rps) < 15M
- 1万连接 (~18K rps) < 200M



ServiceMesh: 理想的解决方案

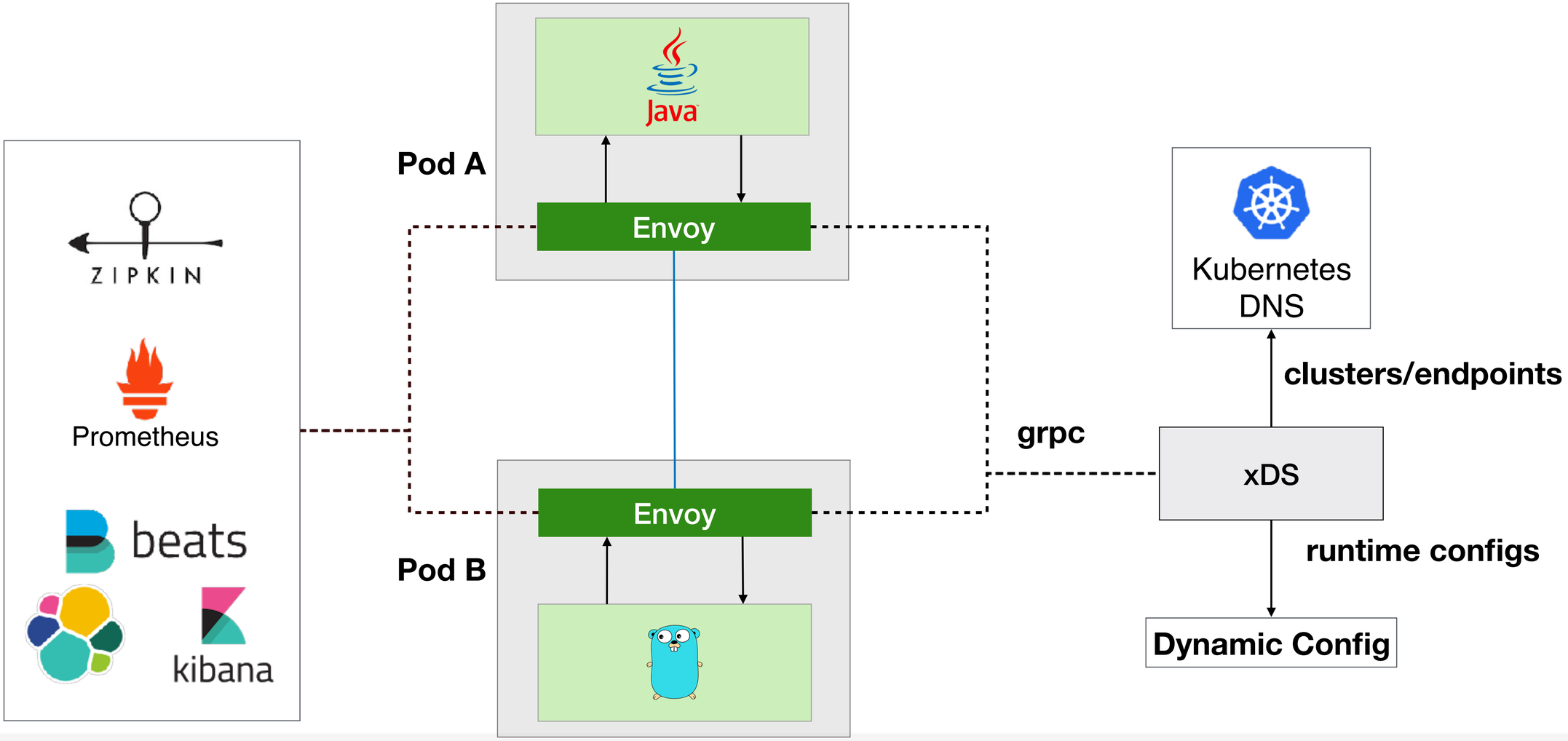
全	易	广
解决所有与网络相关的服务治理问题 • 监控、调用链追踪 • 灰度发布 • 网关 • 服务发现 • 服务路由 • 超时延迟重试	节省业务团队集成成本 • 少改 (最好不改) 现有代码 • 升级对业务影响小 (无影响) • 学习、集成门槛低 • 可拔插	支持多语言 • Java • C++ • C# • Go • Python • Nodejs

TABLE OF

CONTENTS 大纲

- 背景
- Envoy工作原理
- Why Envoy
- 借： 自研ServiceMesh

xDS: Akka 异步无锁



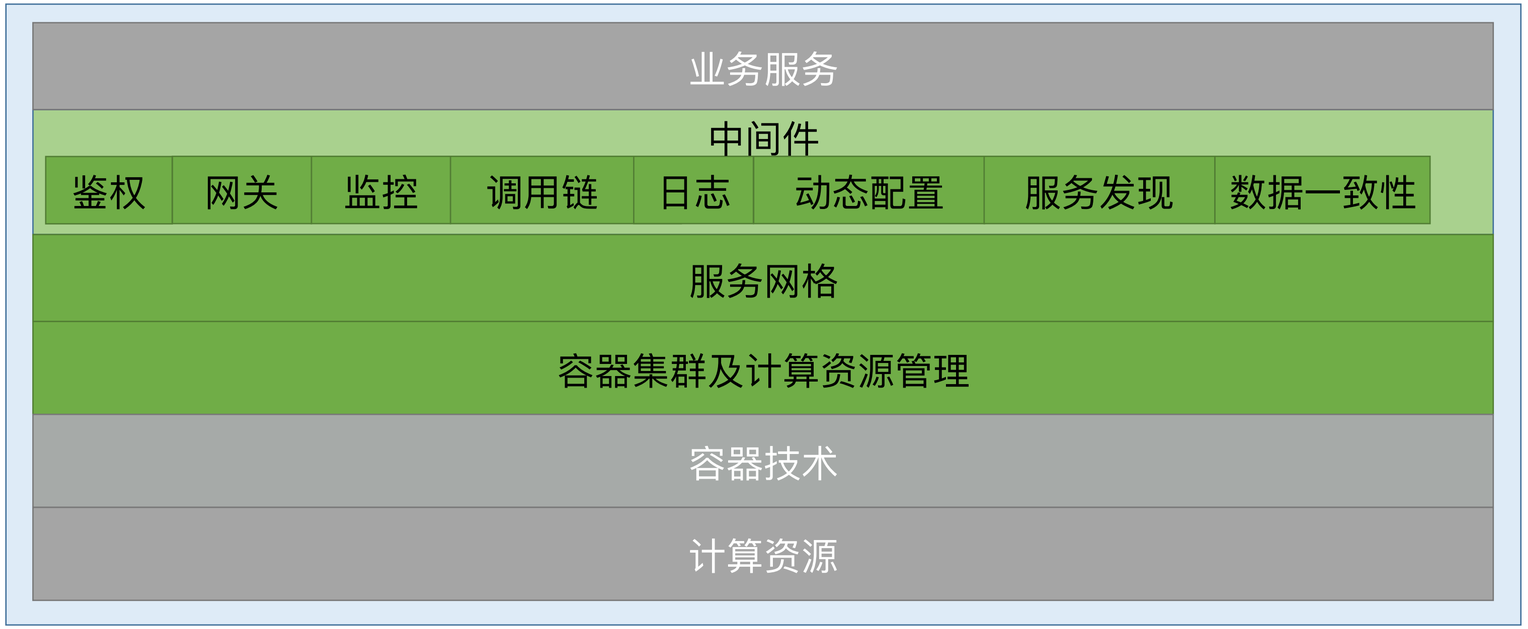
遇到的挑战

- Kubernetes自动化集成测试困难，手动测试花费大量时间
  - 写：自动化脚本
- 不同服务如果使用Tcp/http协议，并设置同一端口，Envoy路由时会出现协议错误
  - 绕：tcp/http服务使用不同端口
- 对Envoy工作原理了解不足 (local cluster / ingress listener)
  - 挖：istio / envoy java-control-plane 源码
- Envoy相关资料和示例不足
  - 问：envoy google groups / slack / github

Summary

- 拼：xDS拼装 - CDS/EDS/LDS/RDS
- 满：全易广
- 借：资料不足，借力社区

We are hiring



THANKS