

网易云Service Mesh的产品架构与实现

刘超

网易研究院云计算技术部首席架构师

关于我

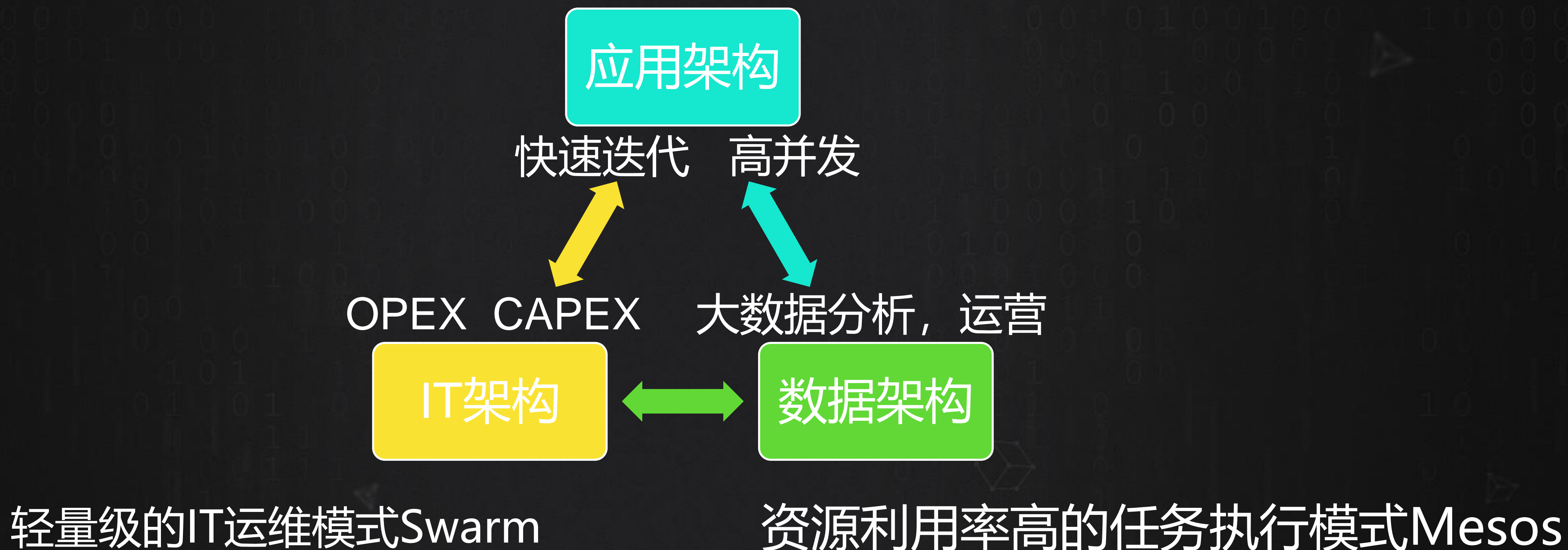
- **刘超**
- **网易云 解决方案总架构师**
- 10余年云计算领域研发及架构经验，先后在EMC，CCTV证券资讯频道，HP，华为，网易从事云计算和大数据架构工作
- 毕业于上海交通大学。
- 曾出版《Lucene应用开发揭秘》
- 多次作为邀请讲师参加Dockone容器技术大会，Segmentfault开发者大会，InfoQ全球架构师峰会（明星讲师），CSDN SDCC大会，51CTO WOTA大会等
- 知名技术博主，博客可搜索popsuper1982，多篇文章推荐至全球最大IT社区CSDN首页及《程序员》杂志
- 在工作中积累了大量运营商系统，互联网金融系统，电商系统等容器化和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

目录

- 01 微服务与Docker、Kubernetes
- 02 网易云微服务框架介绍
- 03 基于容器服务的微服务架构实践

一、微服务与Docker、Kubernetes

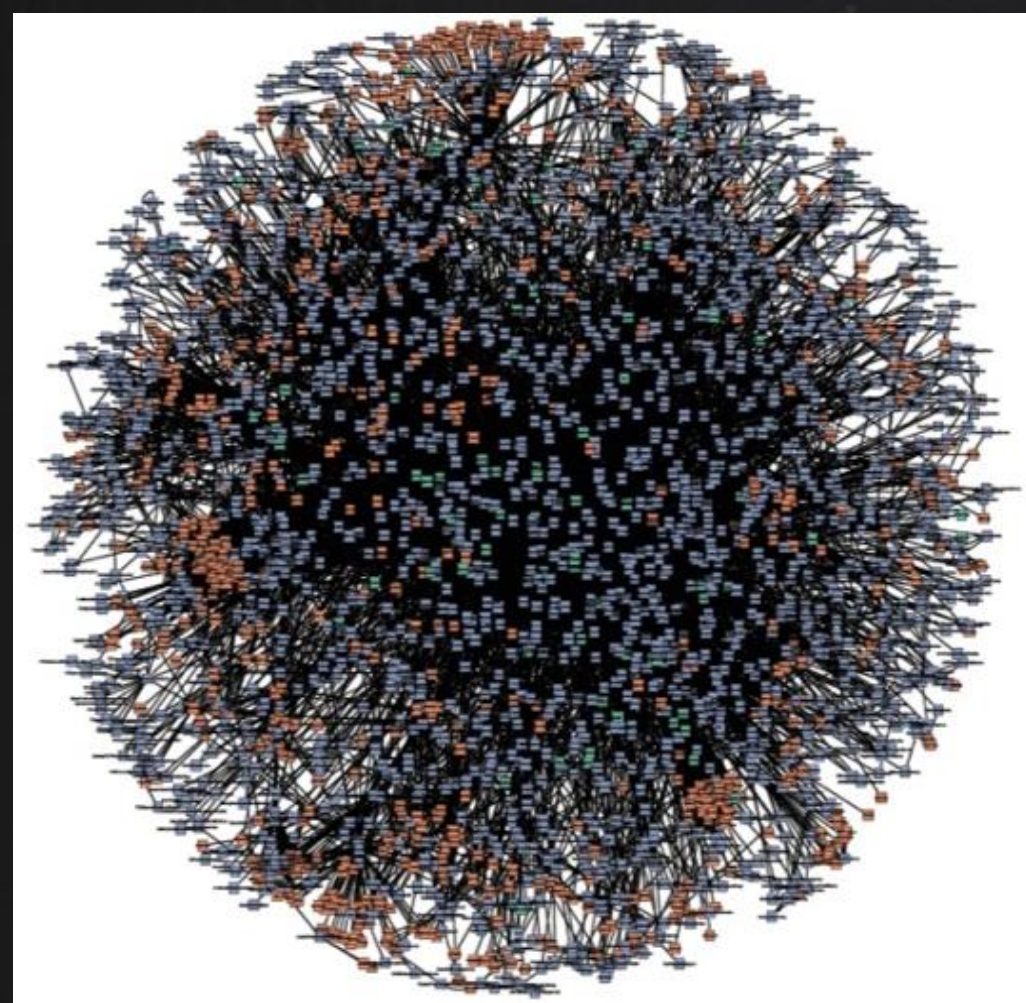
微服务的交付形式Kubernetes



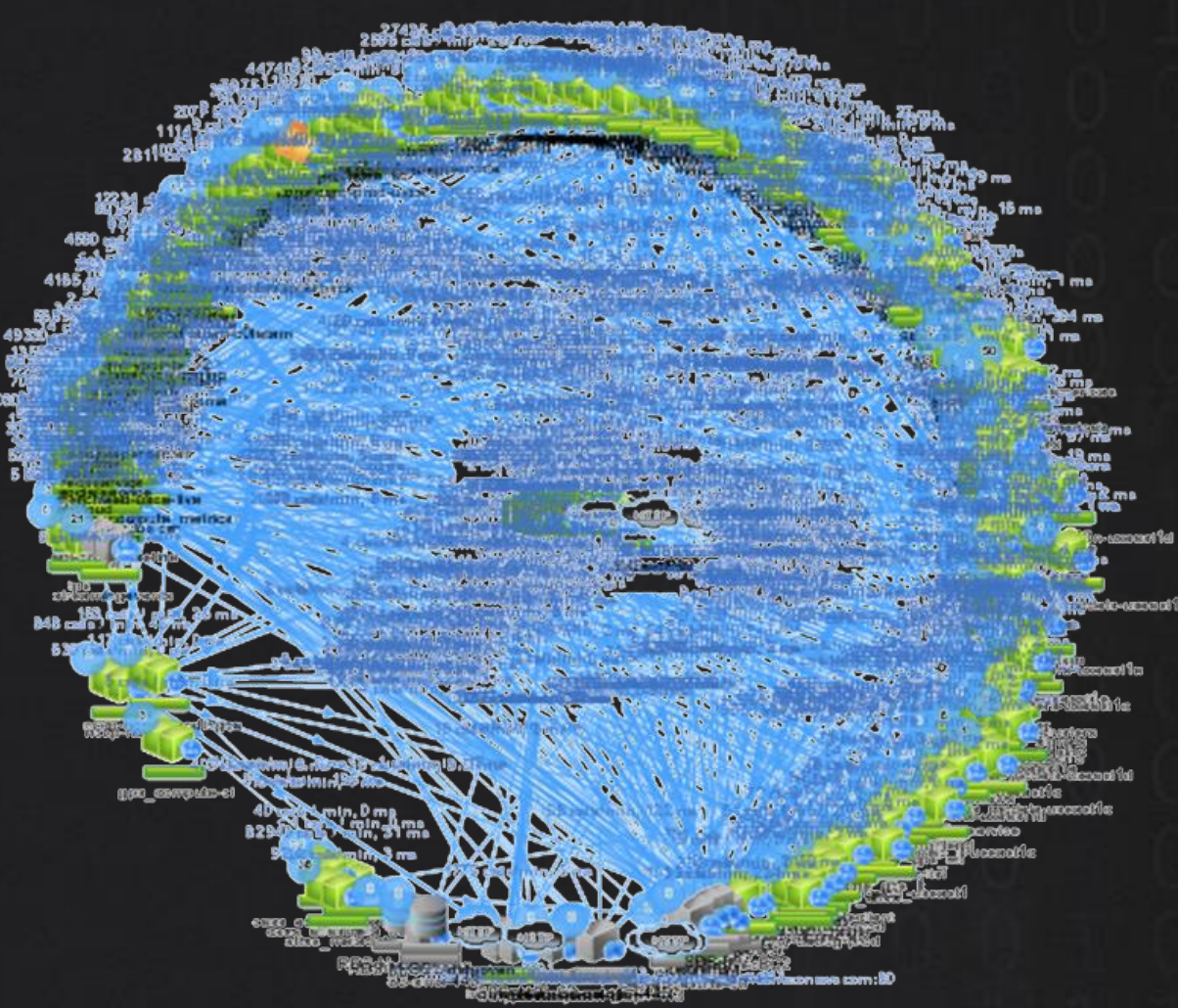
微服务形态



Hailo



Amazon



Netflix



Kaola

来源: <https://www.linkedin.com/pulse/astonishingly-underappreciated-azure-service-fabric-ben-spencer/>

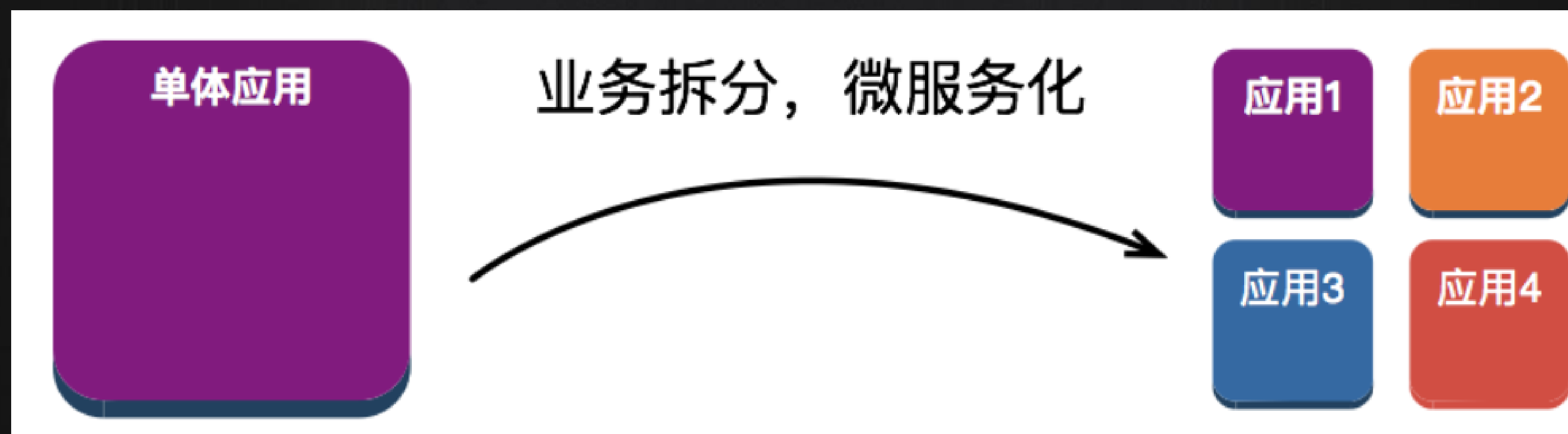
微服务拆分

开发独立: 代码耦合度比较高, 修改代码通常会对多个模块产生影响, 操控难度大, 风险高

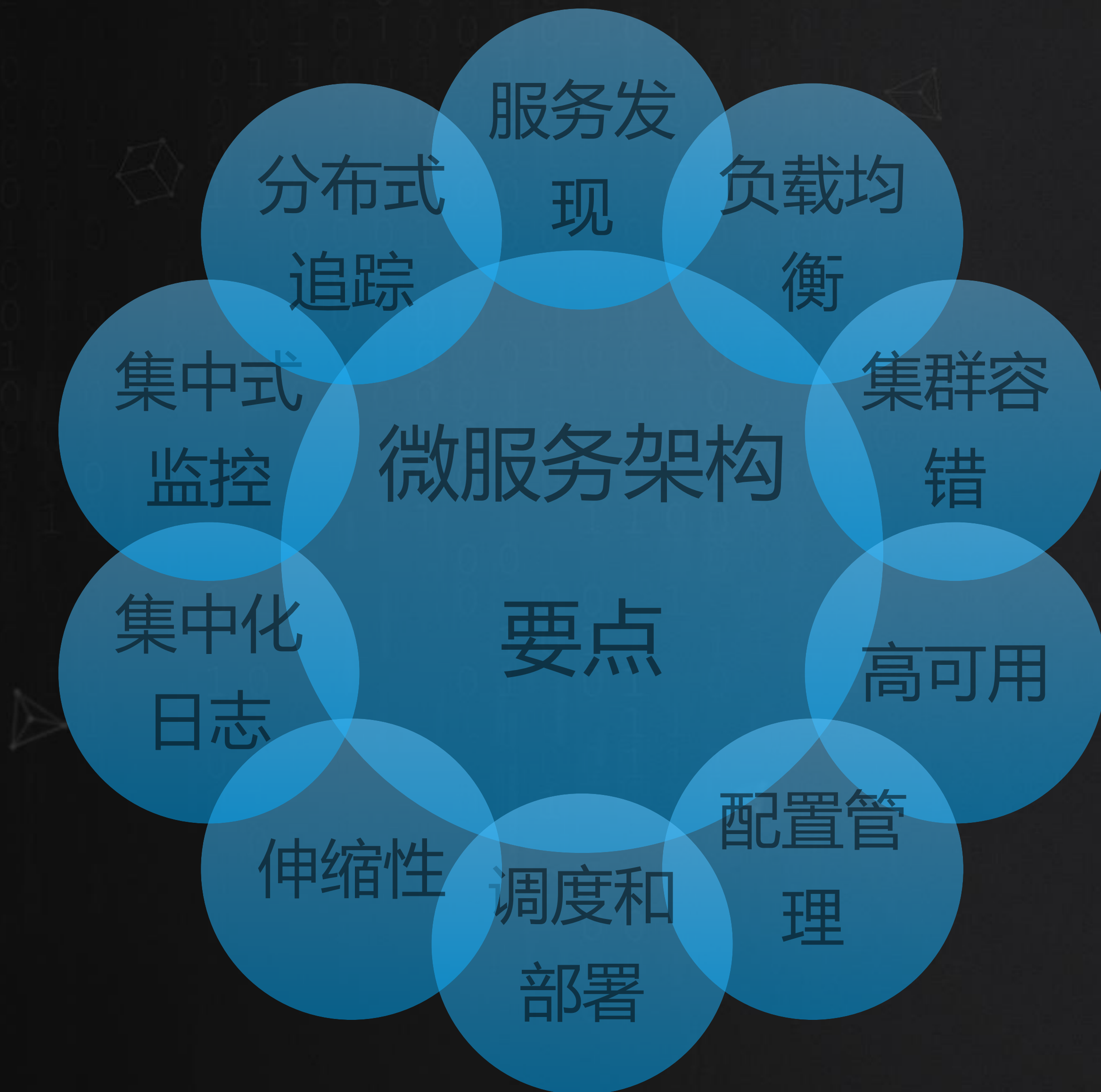
上线独立: 单次上线需求列表多, 上线时间长, 影响面大

简化扩容: 由于业务多, 每一次扩容需要增加的配置比较杂。一些不起眼的小业务虽然不是扩容的主要目的, 也需要慎重考虑

容灾降级: 核心业务与非核心业务耦合, 在关键时候互相影响



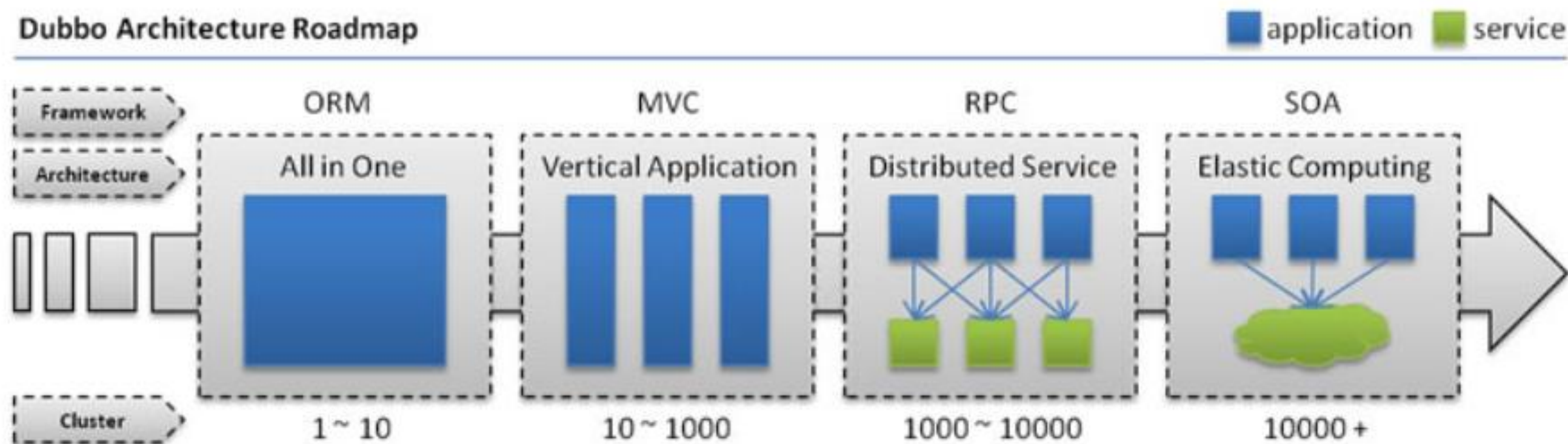
微服务架构要点



微服务框架

- Dubbo
- Spring Cloud
- Kubernetes
- Service Mesh
-

Dubbo产生背景



单一应用架构

垂直应用架构

分布式服务架构

弹性计算架构

当网站流量很小时，只需一个应用，将所有功能都部署在一起，以减少部署节点和成本。此时，用于简化增删改查工作量的 **数据访问框架 (ORM)** 是关键。

当访问量逐渐增大，单一应用增加机器带来的加速度越来越小，将应用拆分成互不相干的几个应用，以提升效率。此时，用于加速前端页面开发的 **Web框架 (MVC)** 是关键。

当垂直应用越来越多，应用之间交互不可避免，将核心业务抽取出来，作为独立的服务，逐渐形成稳定的服务中心。此时，用于提高业务复用及整合的 **分布式服务框架 (RPC)** 是关键。

当服务越来越多，容量的评估，小服务资源的浪费等问题逐渐显现，此时需增加一个调度中心基于访问压力实时管理集群容量。此时，用于提高机器利用率的 **资源调度和治理中心 (SOA)** 是关键。

Spring Cloud技术栈

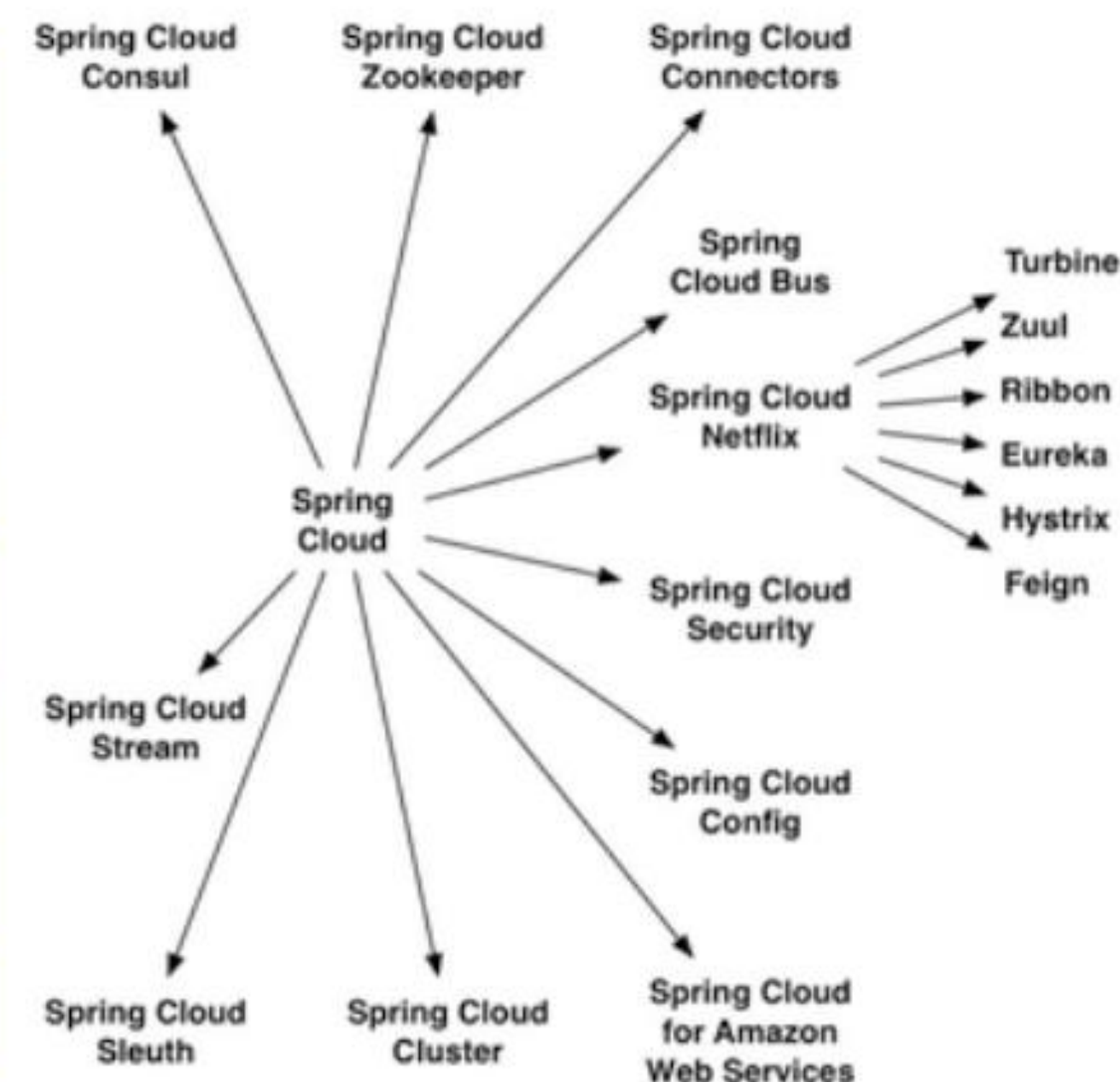
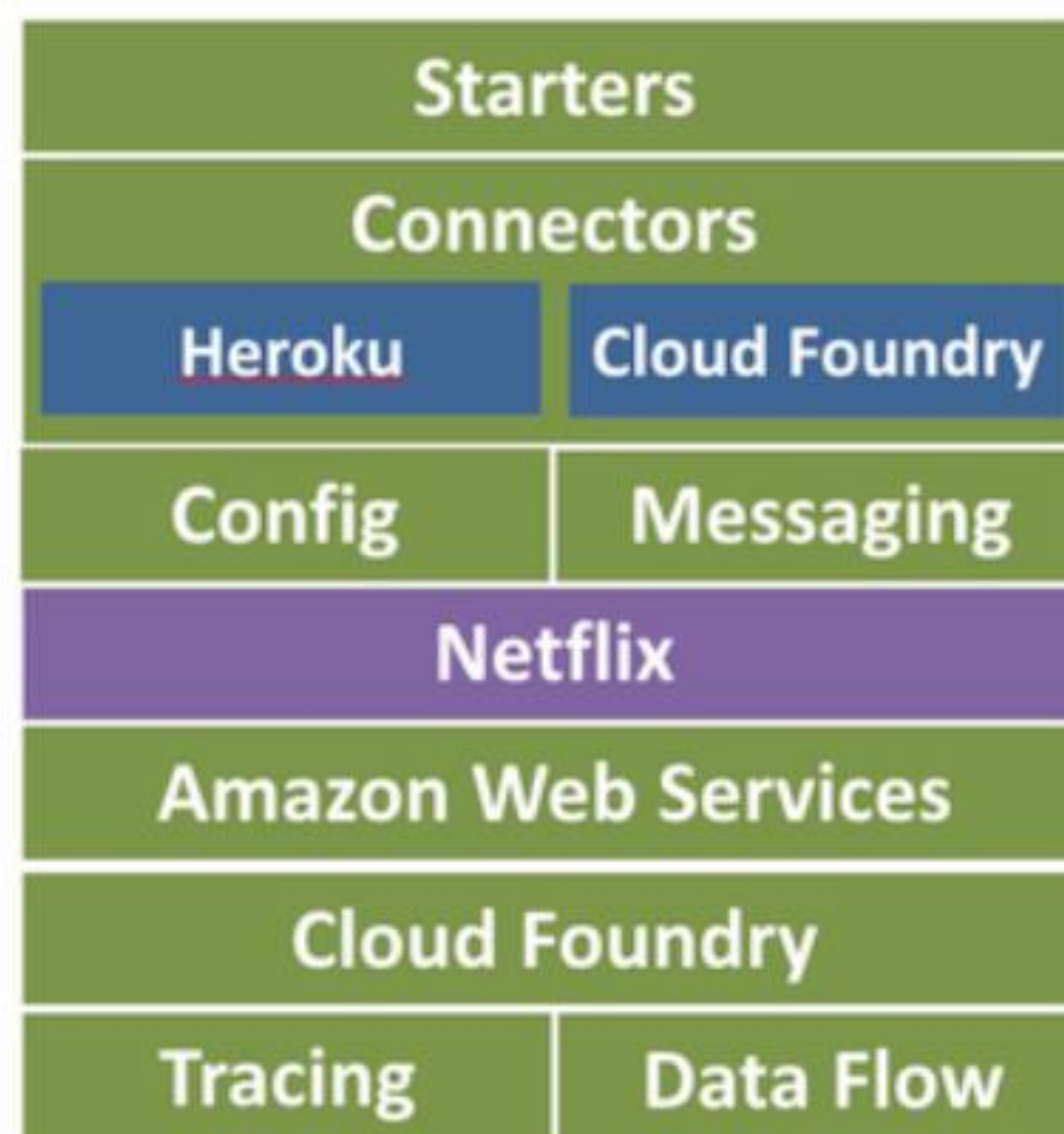
由封装成熟框架（主要）和部分分布基础服务设施两部分组成。

三方成熟框架：

- **Spring Cloud Netflix**：包括服务的发现和注册（**Eureka**），负载均衡（**Ribbon**）、断路器（**Hystrix**）、**REST**客户端（**Feign**）、请求路由（**Zuul**）等服务组件组成

微服务基础设施：

- **Spring Cloud Config**：配置中心
- **Spring Cloud Bus**:消息总线
- **Spring Cloud Stream**：消息驱动



SpringCloud 生态圈

Spring Cloud优缺点

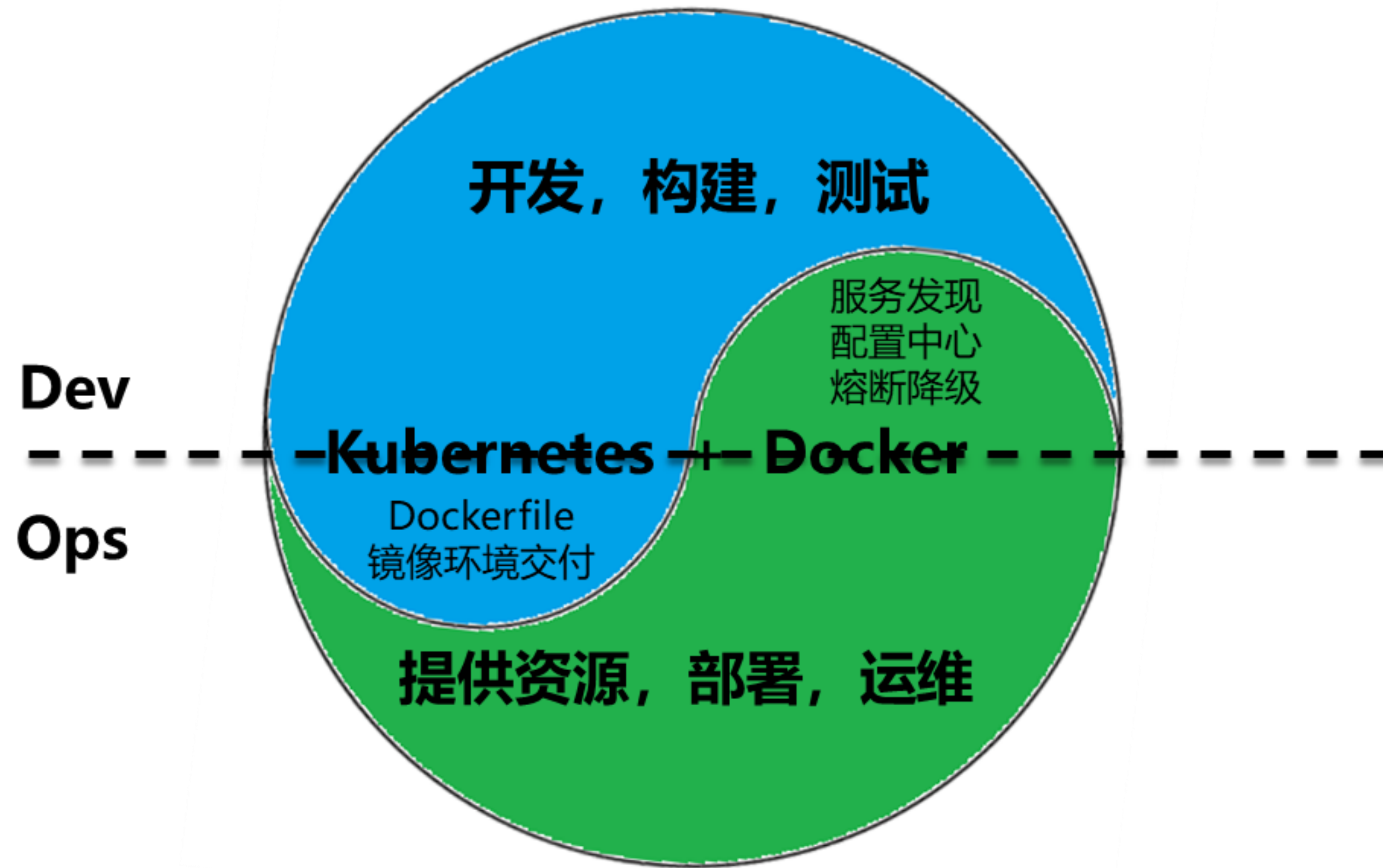
优点

1. 可定制性强，可以通过定制满足各种业务场景。
2. 基于**Spring Boot**的统一编程模型，有快速创建应用的能力。
3. 丰富的**SDK**和类库支持，覆盖大多数运行时需求。
4. 配置灵活，控制能力强。
5. 注解驱动，**Spring Cloud**应用兼容性好。

缺点

1. 只适用于**Java**技术栈，不支持跨语言的能力。
2. 使用门槛较高，开发人员需处理大量与业务无关的逻辑。
3. 覆盖面有限，缺少完整的微服务生命周期支持，比如自动化部署、调度、资源管理、进程隔离、故障恢复等。

Kubernetes作为微服务平台



Kubernetes作为微服务框架

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

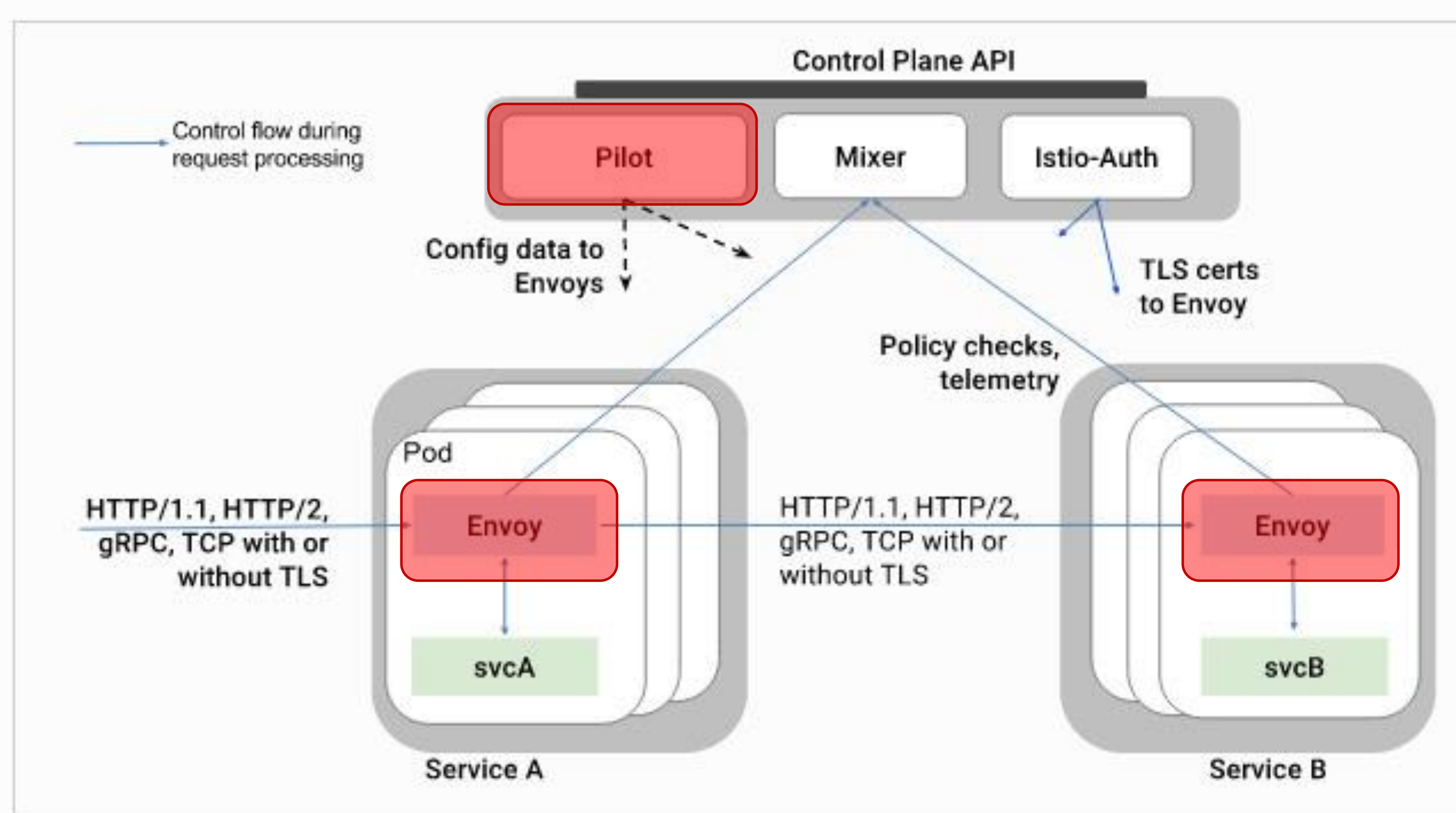
优点

1. 多语言支持。
2. 平台层解决微服务痛点问题，业务基本无侵入。
3. 完整地覆盖了微服务生命周期，从运行时到部署、调度、资源管理、隔离、故障恢复等。
4. **Google**主导，容器技术的事实标准。

缺点

1. 定义为通用平台，定制优化能力较弱。
2. 学习曲线陡，使用门槛较高，运维或者**DevOps**人员需维护一个稳定的**K8S**集群。
3. 年轻，迭代快。

Service Mesh作为微服务框架

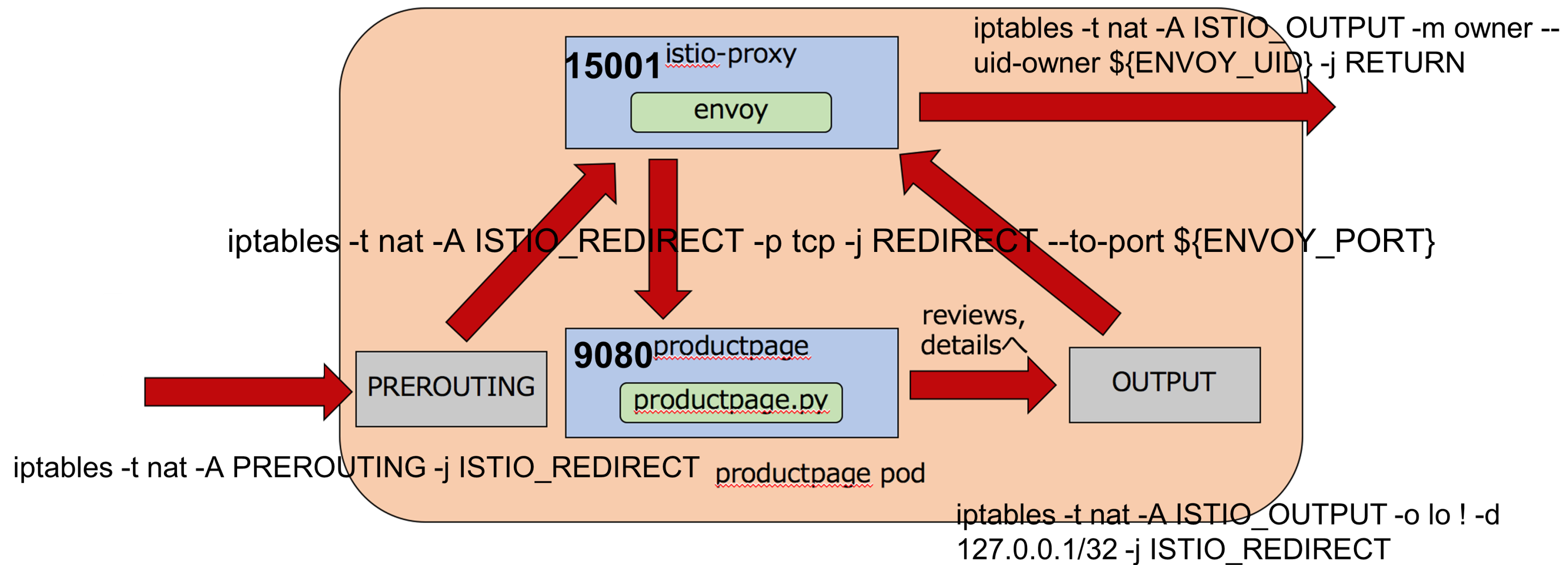
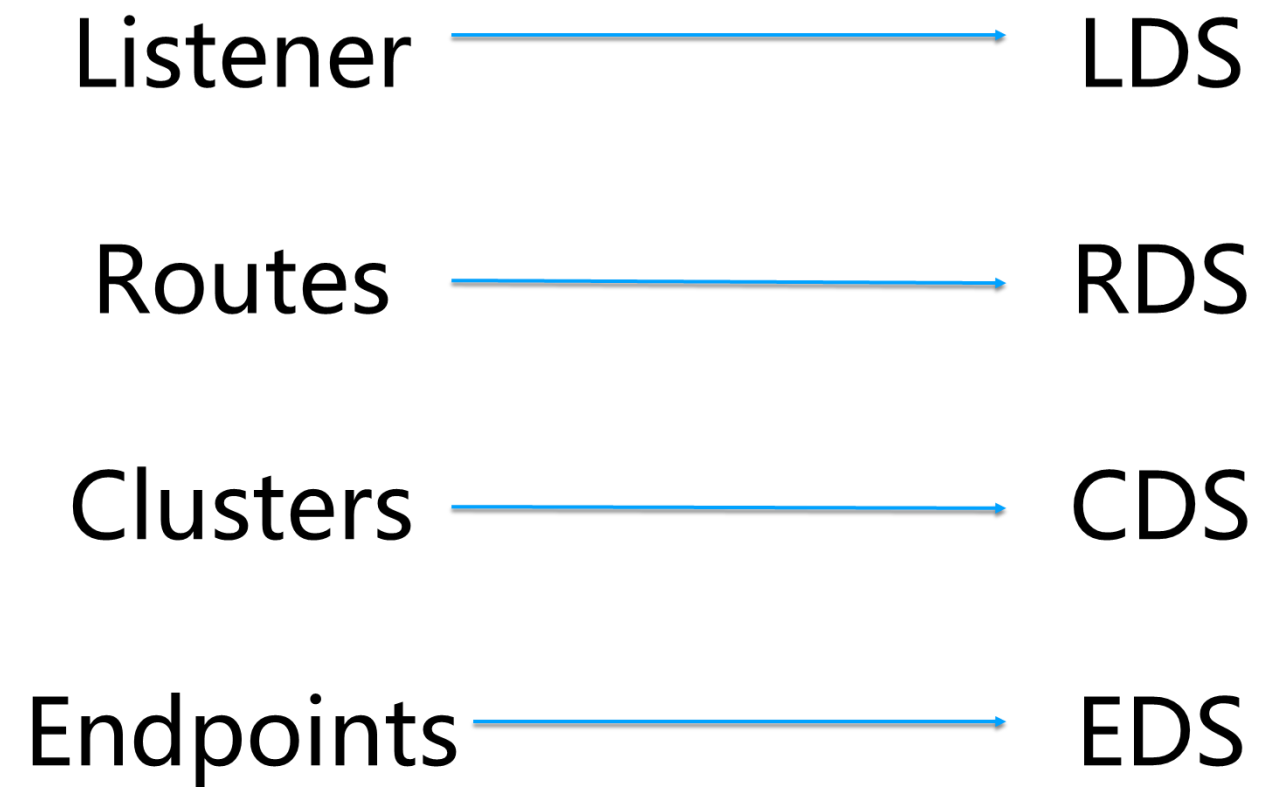


Service Mesh生态

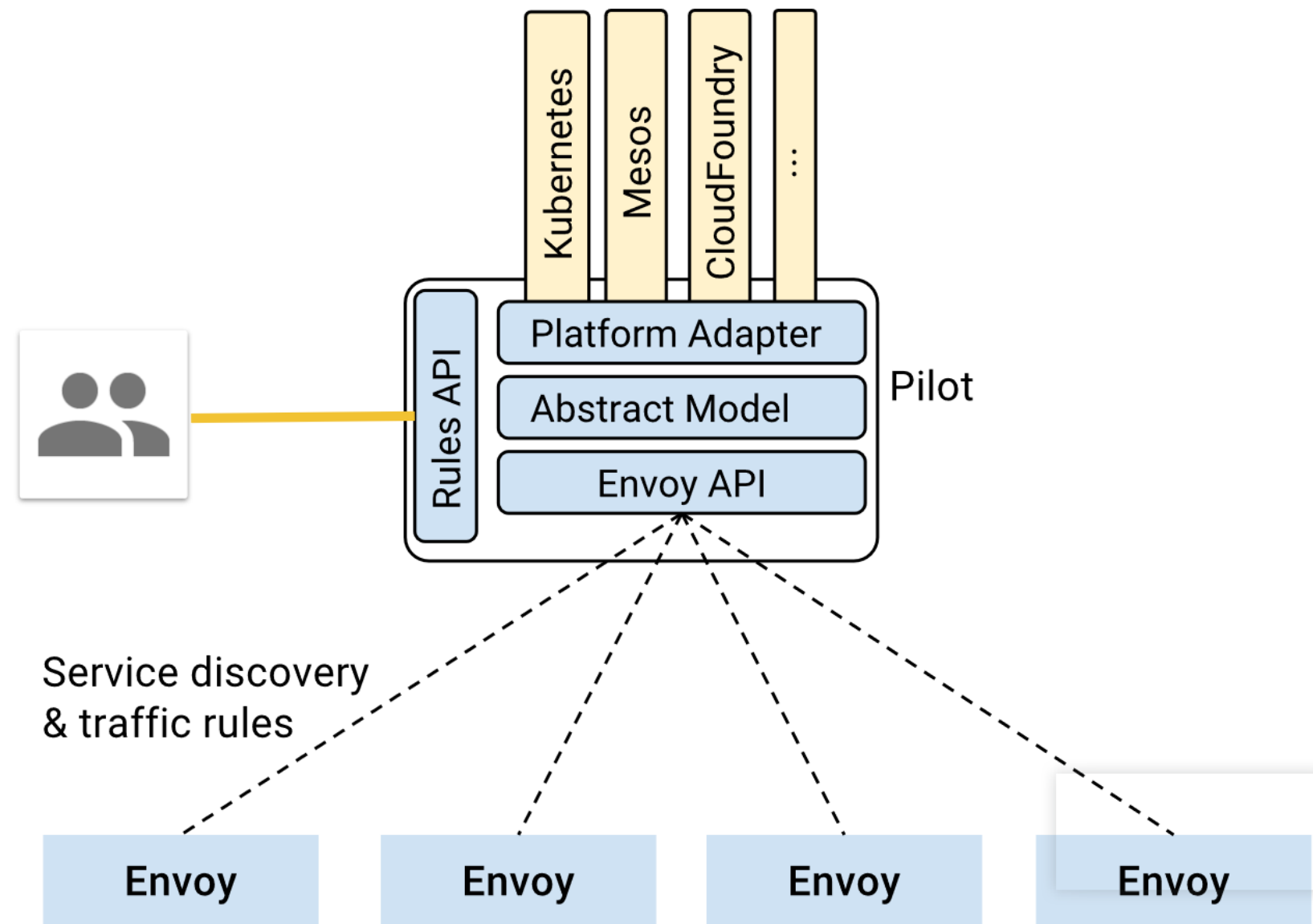
- 数据面：
 - Envoy(star 5000+, Latest commit 4 hours ago)
 - Linkerd(star 4000+, Latest commit 19 hours ago)
 - nginmesh(star 300, Latest commit 29 days ago)
- 控制面：
 - Istio(star 8500+, Latest commit 8 hours ago)

数据面Envoy

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

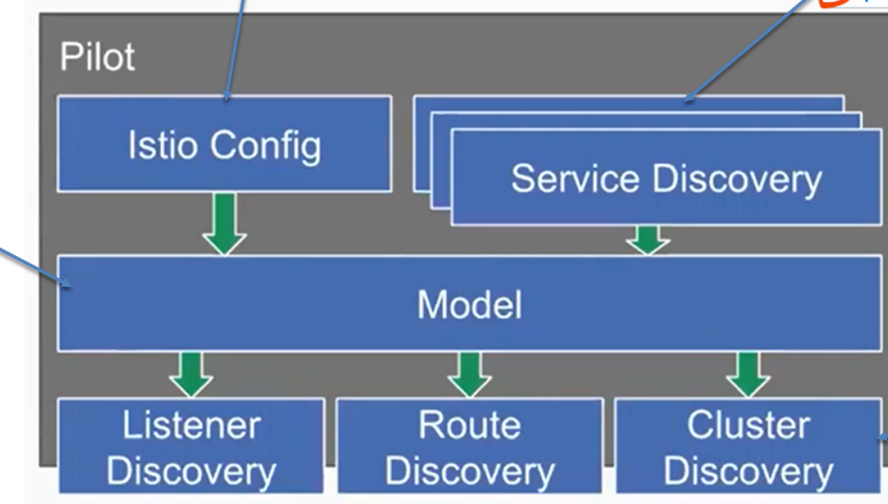


控制面Pilot



```
model  
├── test  
│   ├── authentication.go  
│   ├── authentication_test.go  
│   ├── config.go  
│   ├── config_test.go  
│   ├── context.go  
│   ├── context_test.go  
│   ├── controller.go  
│   ├── conversion.go  
│   ├── conversion_test.go  
│   ├── egress_rules.go  
│   ├── egress_rules_test.go  
│   ├── gateway.go  
│   ├── gateway_test.go  
│   ├── service.go  
│   ├── service_test.go  
│   ├── validation.go  
│   └── validation_test.go
```

```
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
```



```
proxy  
├── envoy  
│   └── v1  
│       ├── cds.go  
│       ├── discovery.go  
│       ├── eds.go  
│       ├── eds_test.go  
│       ├── lds.go  
│       ├── xds.go  
│       ├── agent.go  
│       ├── agent_test.go  
│       ├── net.go  
│       └── resolve.go
```

```
serviceregistry  
├── aggregate  
├── cloudfoundry  
├── consul  
├── eureka  
├── external  
└── kube  
    ├── testdata  
    │   ├── cache.go  
    │   ├── cache_test.go  
    │   ├── client.go  
    │   ├── controller.go  
    │   ├── controller_test.go  
    │   ├── conversion.go  
    │   ├── conversion_test.go  
    │   ├── deregister.go  
    │   ├── deregister_test.go  
    │   ├── queue.go  
    │   ├── queue_test.go  
    │   ├── register.go  
    │   ├── register_test.go  
    └── platform.go
```

Service Mesh优缺点

优点

1. 对业务代码0侵入，开发者无需关心分布式架构带来的复杂性以及引入的技术问题。
2. 对于不适合改造的老旧单体应用，提供了一种接入分布式环境的方式。
3. 理念很新，并且由大企业主导(Google、IBM等)，社区活跃，是下一代微服务架构。

缺点

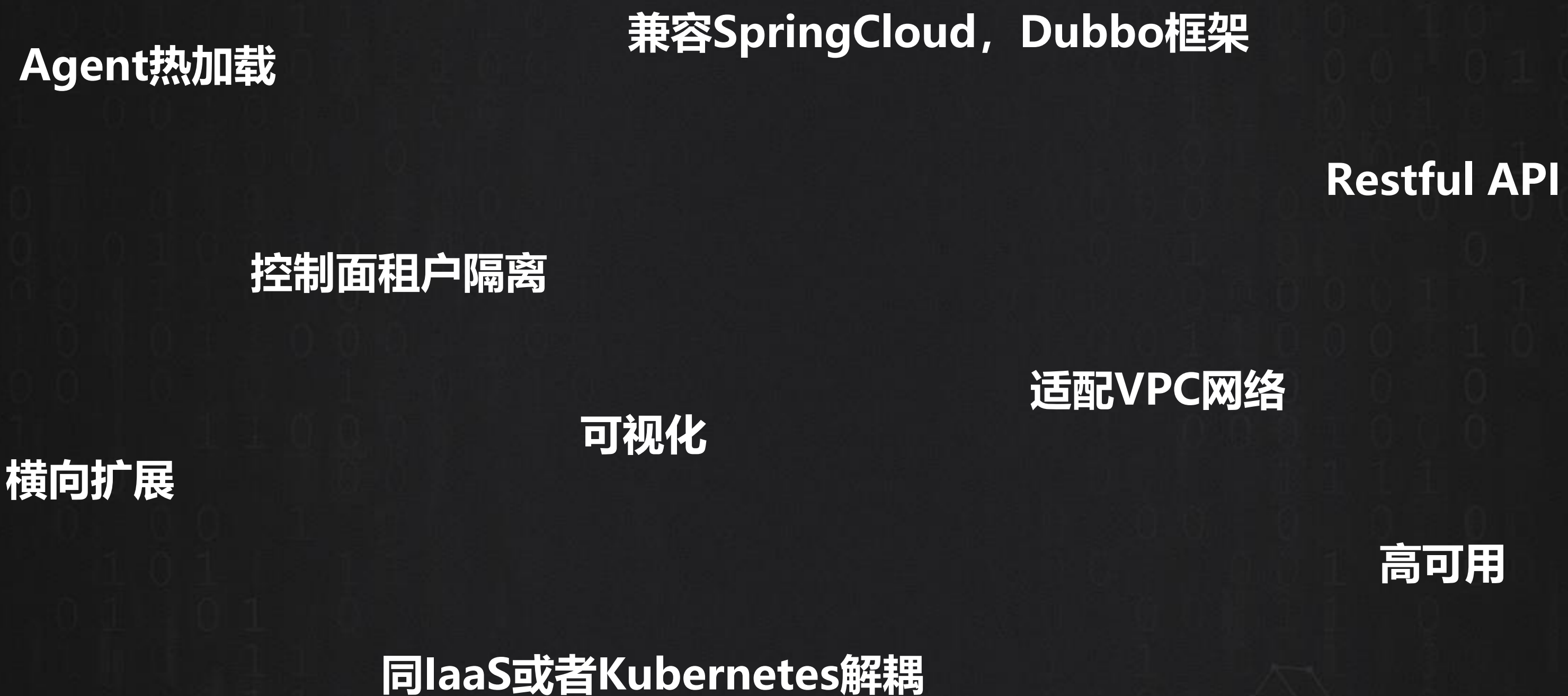
1. **Service Mesh**网格代理请求转发，会在一定程度上降低系统通信性能。
2. 依赖单独的**Service Mesh**代理，多了一个故障点。整个系统的运行和运维也强依赖于**Service Mesh**组件的能力。
3. 成熟度较差，开源方案存在性能问题，没有生产环境的最佳实践案例。

二、网易云微服务框架介绍

设计思路

- 发现容易搞定，治理难搞定
- 发现不易替换，治理易替换
- 解耦容器平台与微服务
- 解耦服务发现与服务治理

设计思路



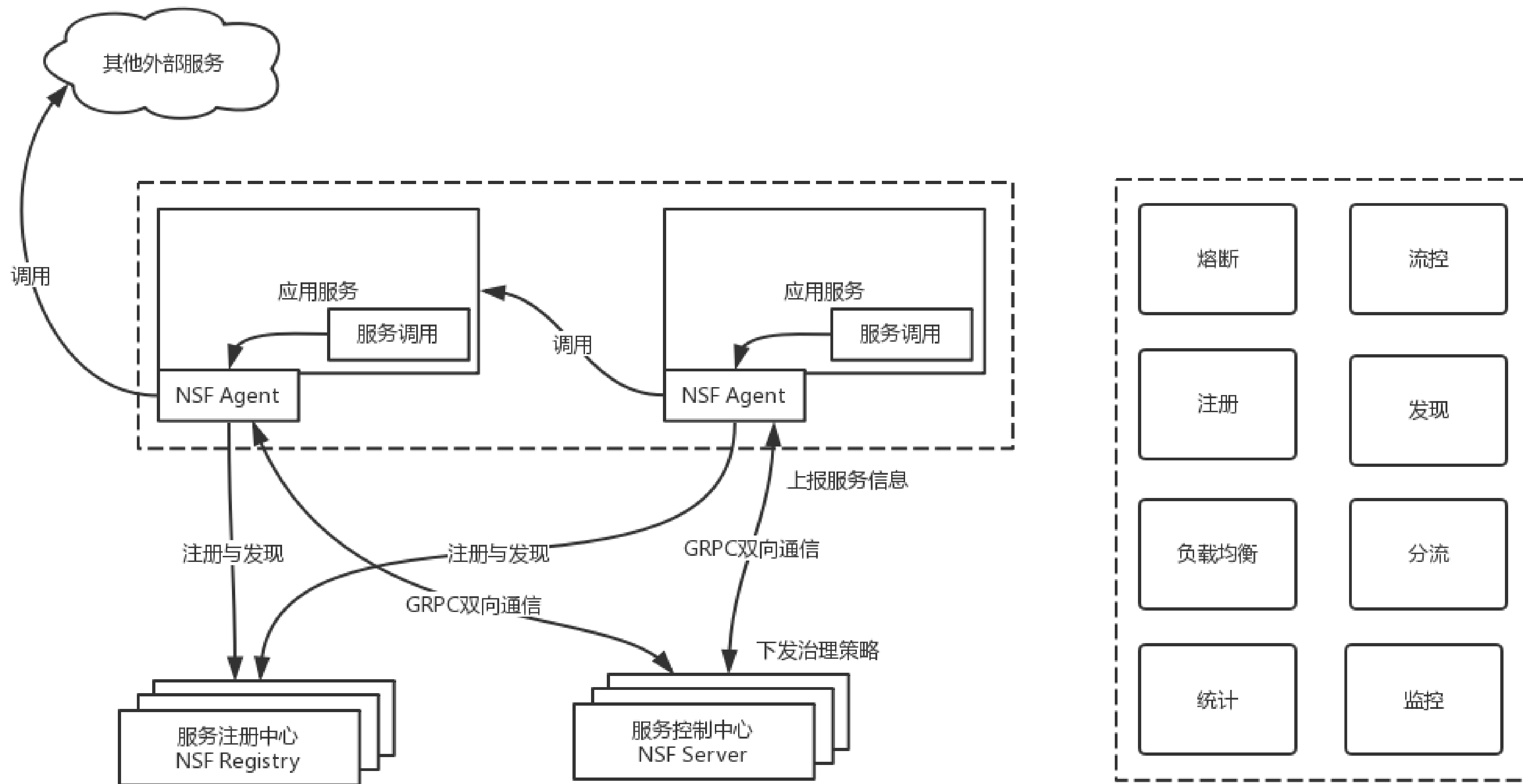
网易微服务框架

- 网易微服务框架NSF (NetEase Service Framework)是企业级微服务管理、治理、控制平台，提供包括服务注册、发现、熔断、降级、容错、限流、负载均衡、监控、统计等功能。

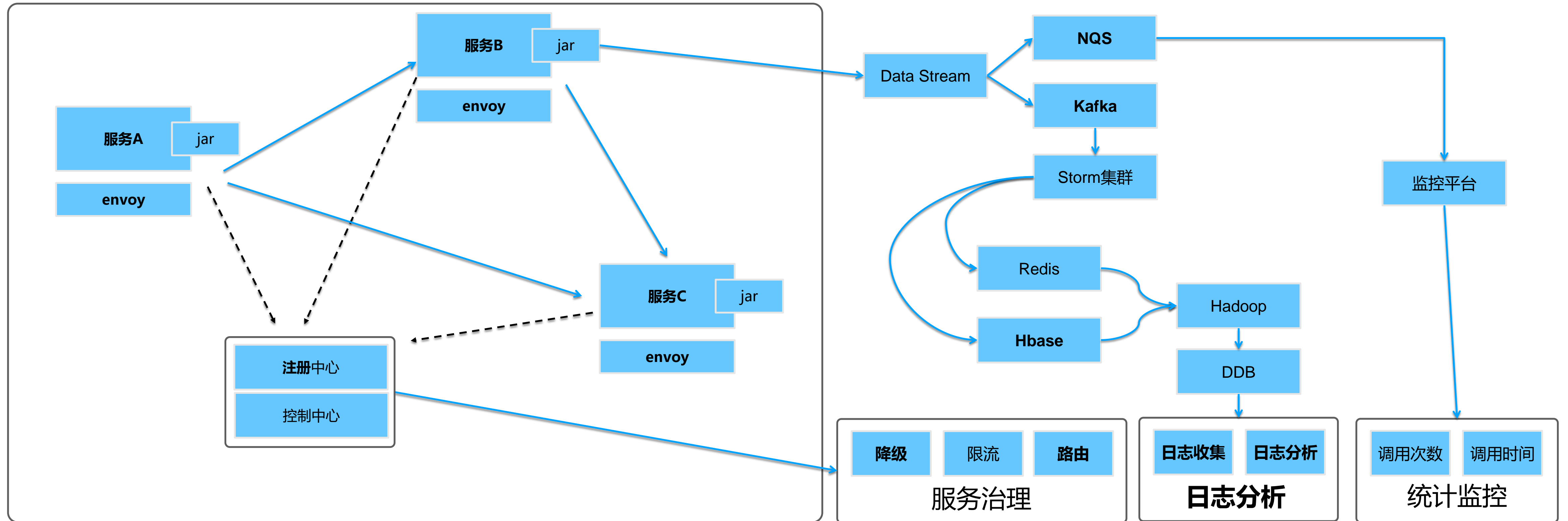
几点优势：

- 开源技术栈：基于Spring Cloud实现服务治理，自由可控
- 无侵入：通过字节码增强技术实现Agent逻辑，应用无感知
- 兼容性：兼容Spring Cloud和Dubbo应用
- 高可用：分布式系统架构，99.99%以上的可用性
- 高性能：单节点支持10000+服务实例运行，可水平扩展

网易微服务框架

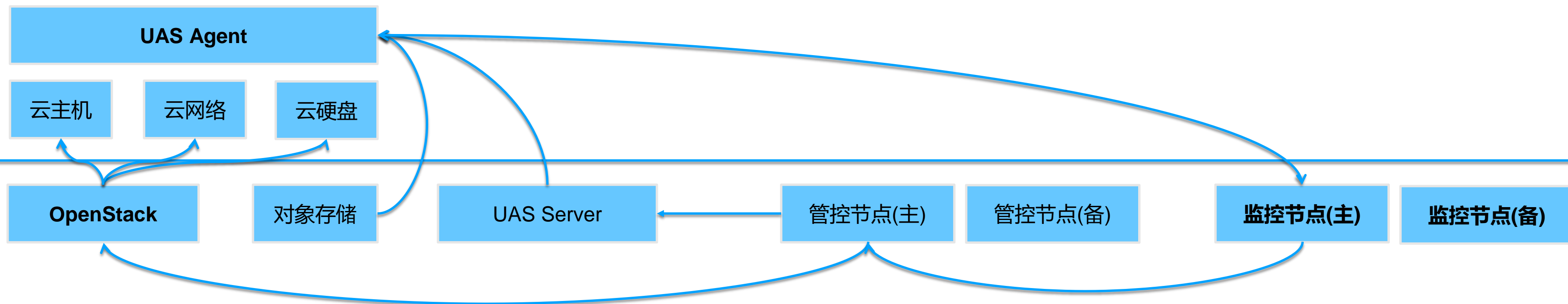


新一代微服务治理平台



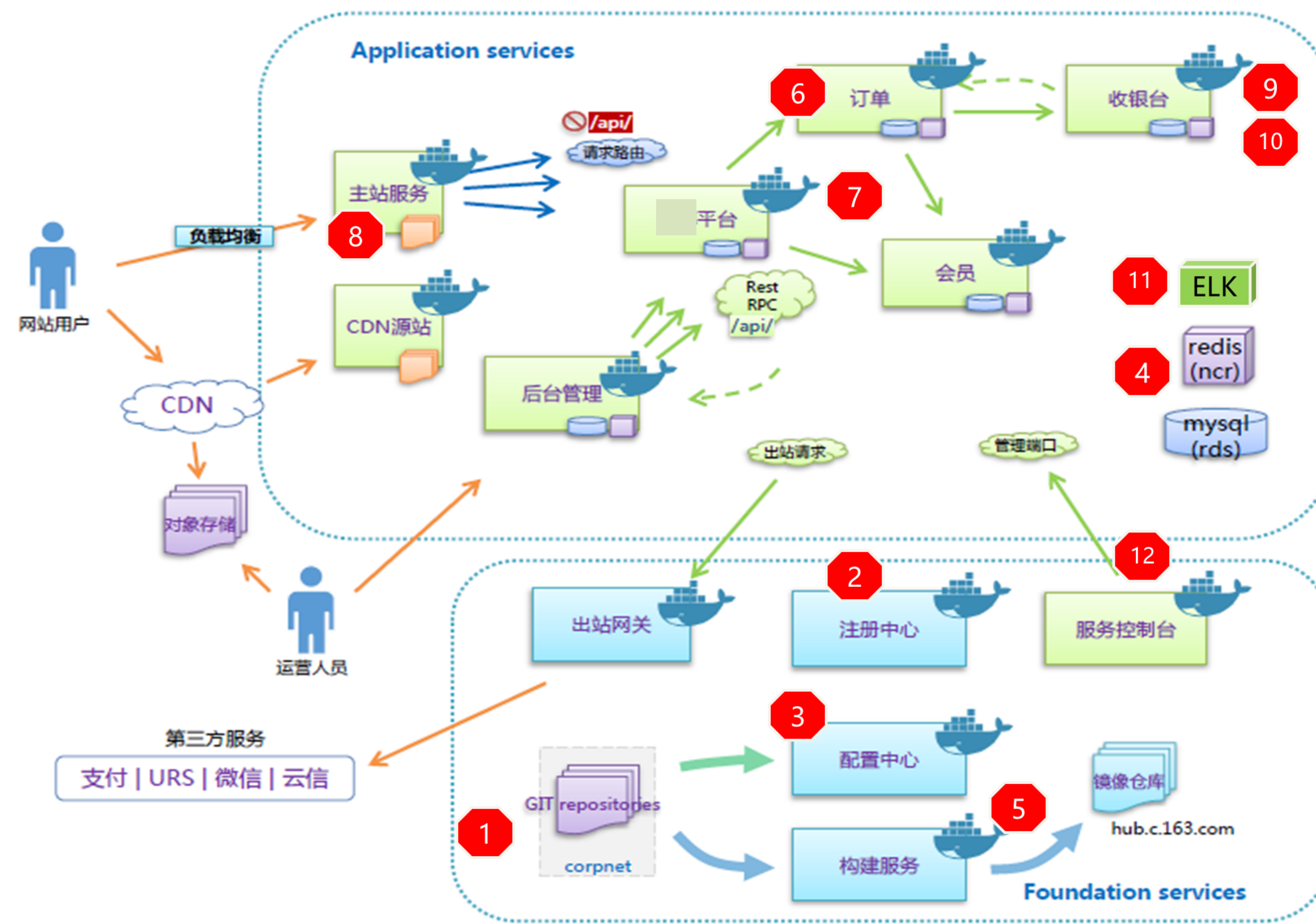
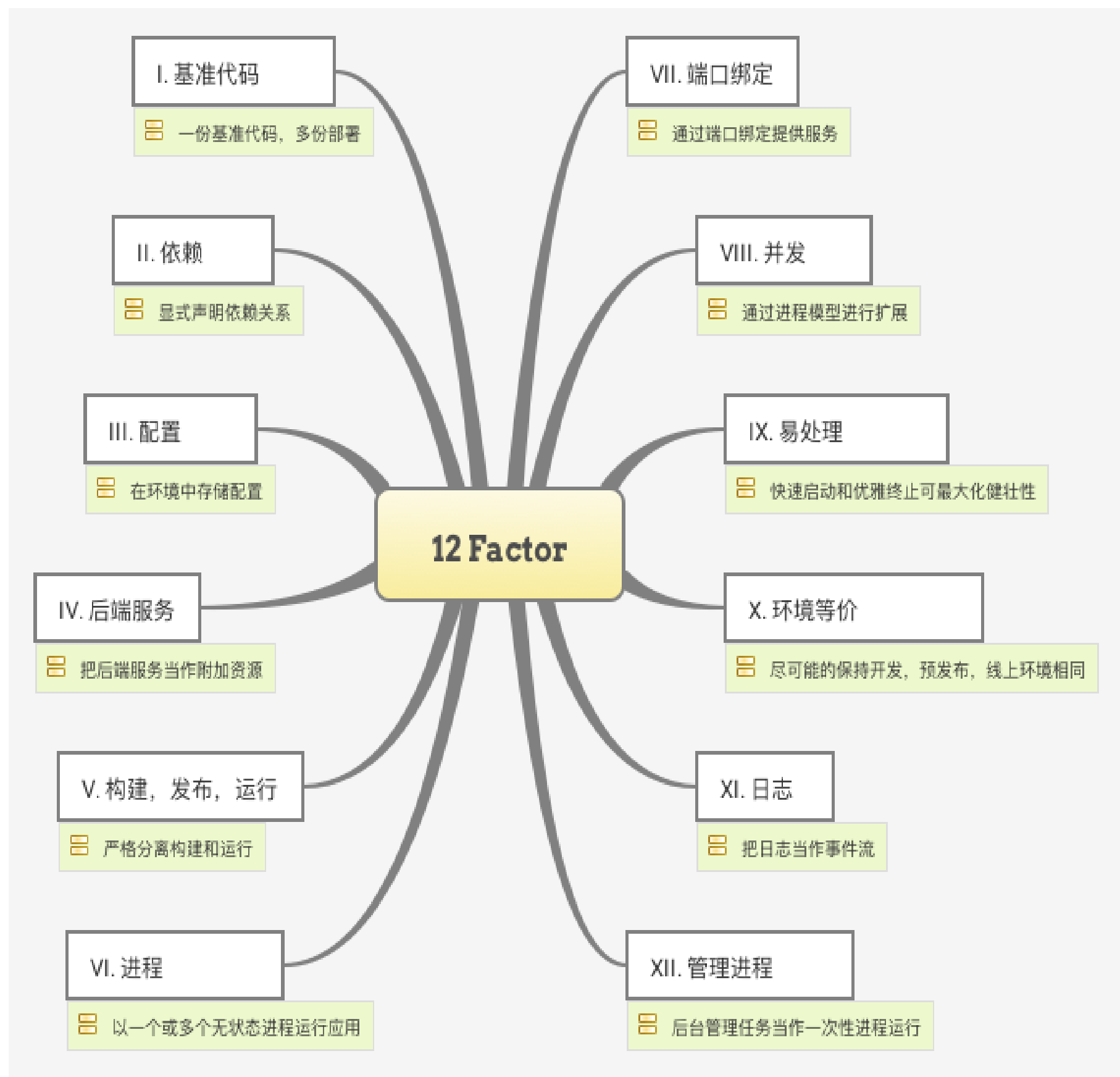
用户空间VPC

管理控制服务

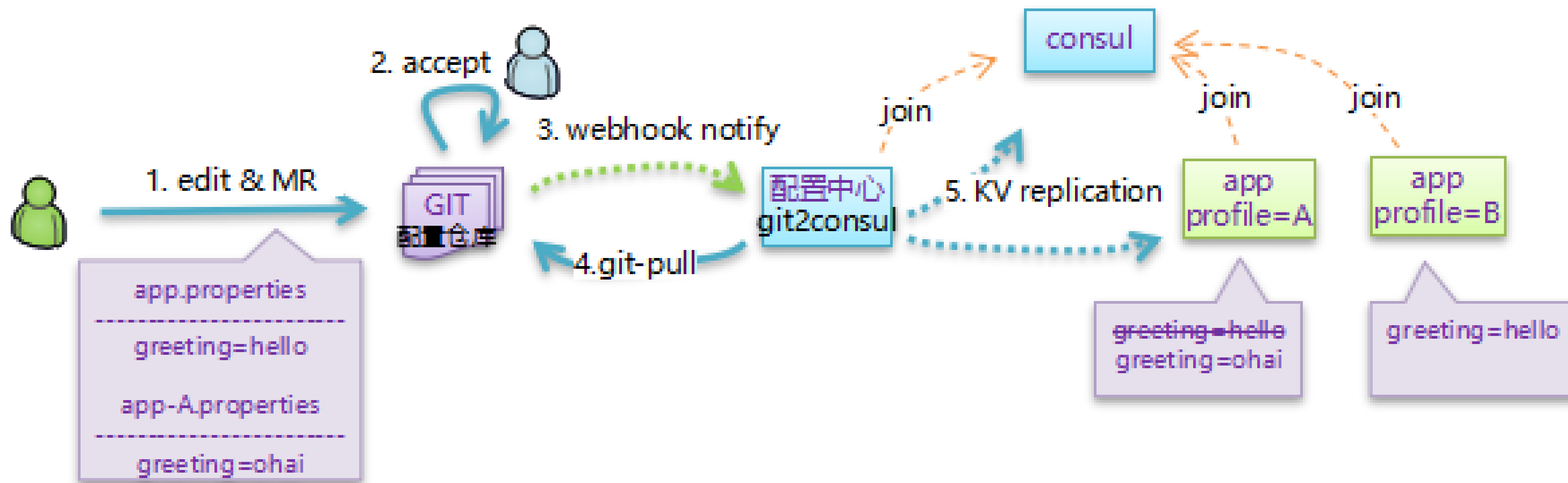


三、基于容器服务的微服务架构实践

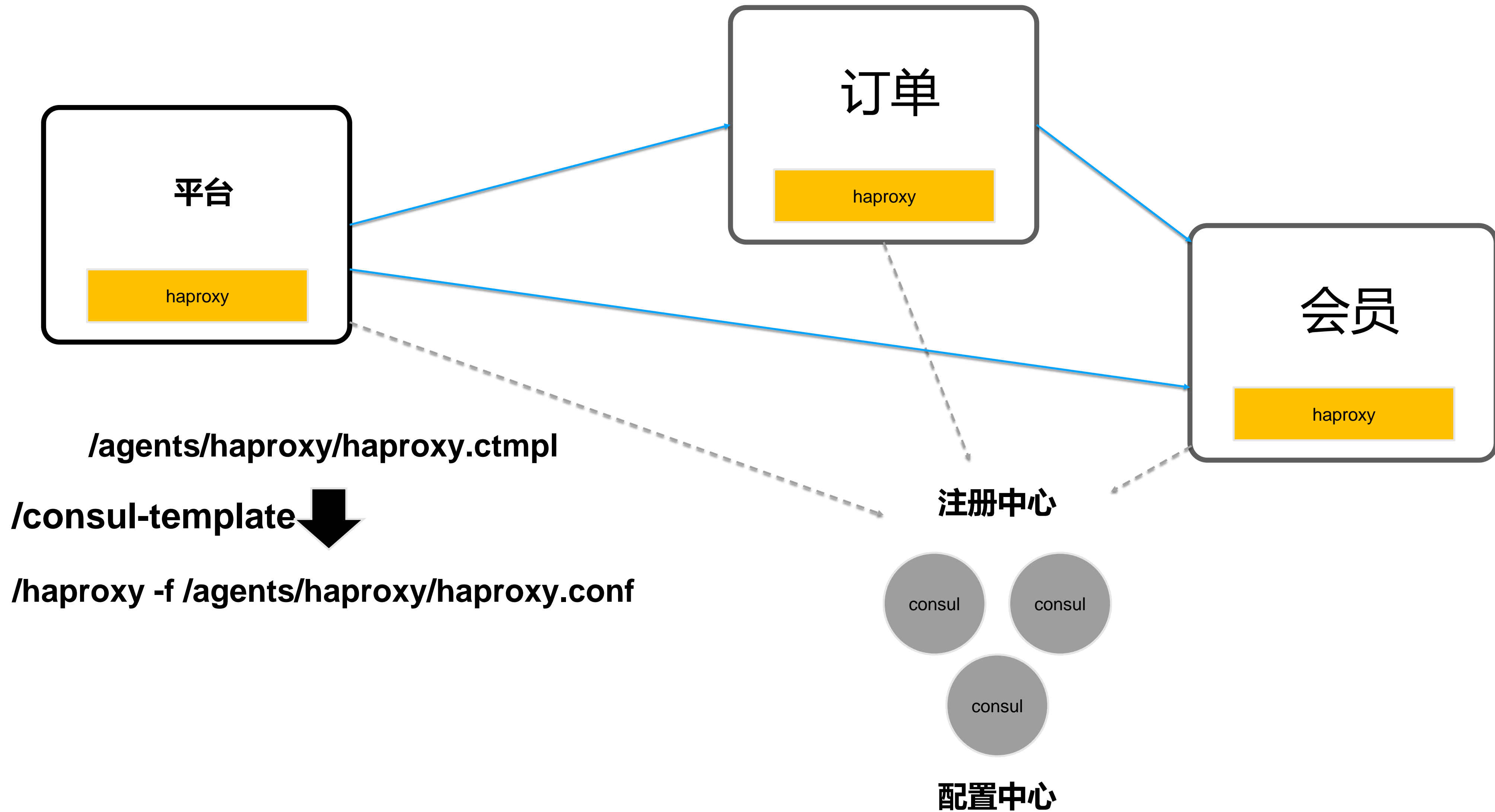
内部金融产品架构



Haproxy作为service mesh的Agent



从haproxy到envoy



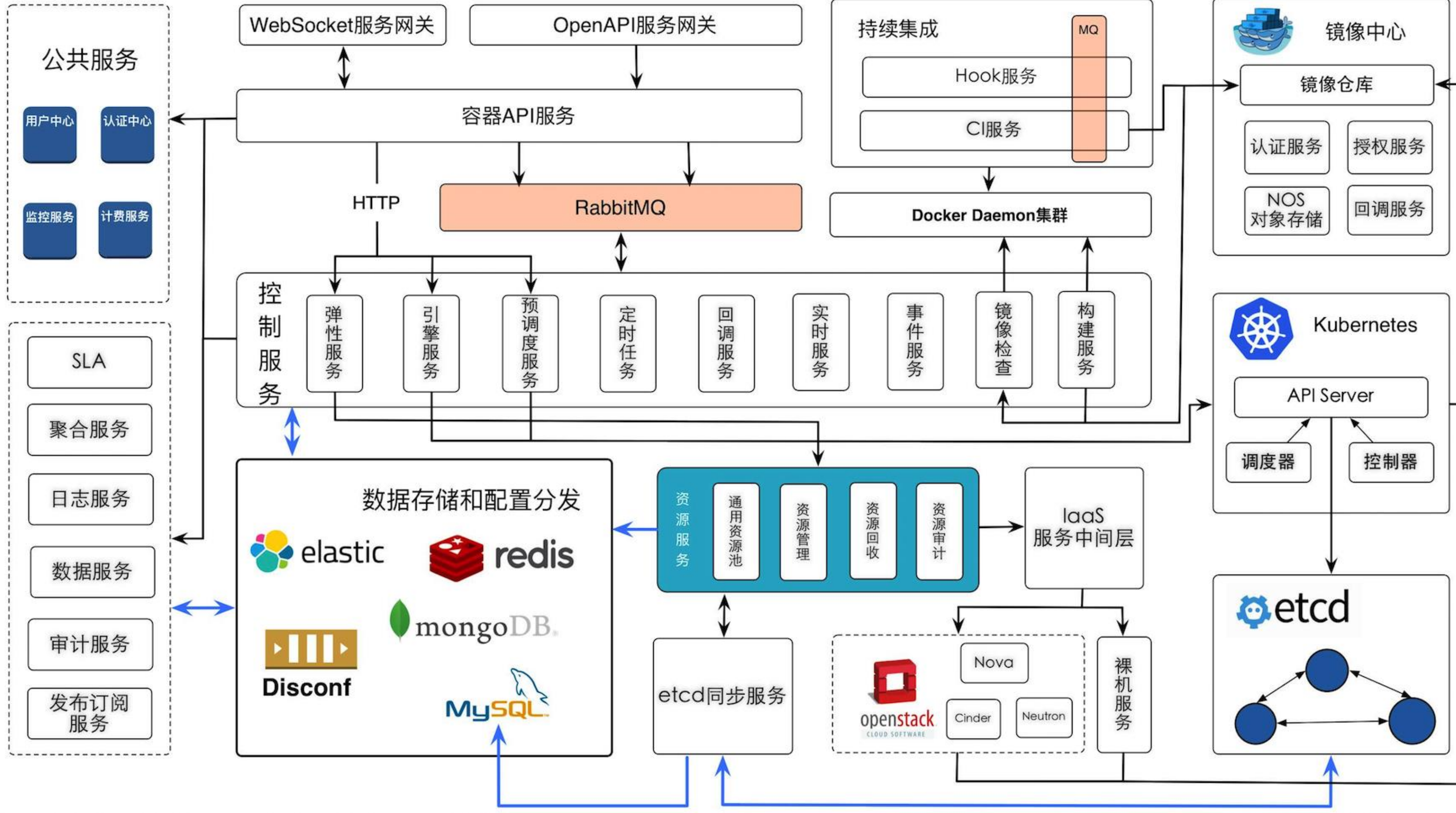
网易云容器服务逻辑架构



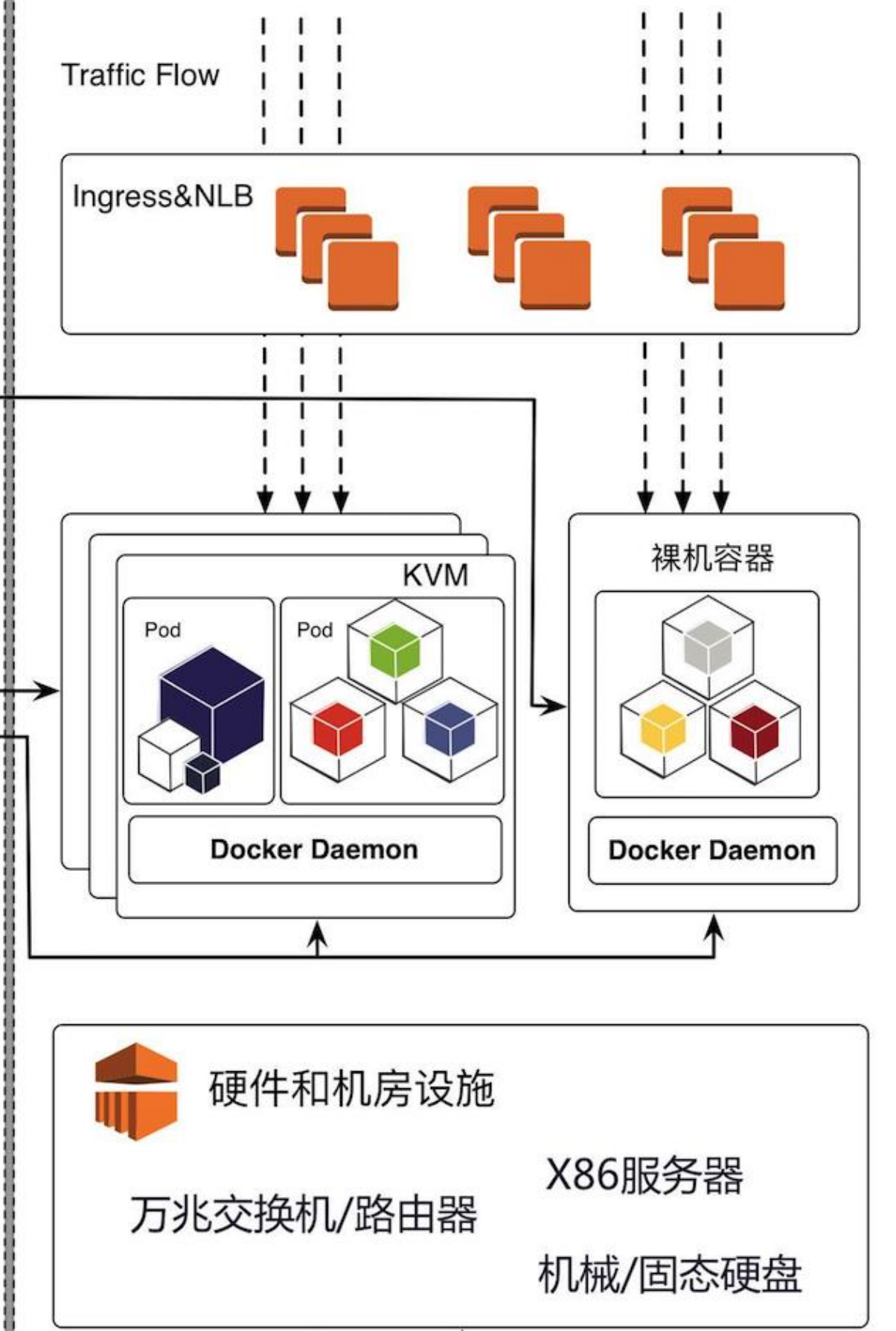
= Docker + Kubernetes + OpenStack + DevOps工具链

网易云容器服务架构

容器服务



容器实例



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

网关自助平台

快捷操作

您好, 裴斐 (服务管理员)

今日日期: 3月22日

概况总览

网关接入服务一览

GO网关API调用总次数统计

G1网关API调用总次数统计

云监控NCM 访问控制服务SAM OPS用户服务

- OPS用户服务 服务被创建, 负责人为郭晋, 20分钟前
- 云监控NCM 服务被创建, 负责人为杨劲柯, 2天前
- 访问控制服务SAM 服务被创建, 负责人为蒋文康, 2天前
- 云支付 服务被创建, 负责人为李超, 6天前
- g0test2 服务被创建, 负责人为张雨, 13天前
- 镜像仓库服务 服务被创建, 负责人为姜政考, 14天前
- 管理平台迁移接口 服务被创建, 负责人为裴斐, 1月前
- WebSocket服务 服务被创建, 负责人为朱凌晨, 1月前
- keystone 服务被创建, 负责人为管强, 1月前
- 镜像服务 服务被创建, 负责人为管强, 1月前

网关自助平台

快捷操作

网关概况总览

平台Dashboard

网关审计管理

服务审计管理

调用记录查询

调用统计查询

服务依赖关系

服务与API管理

网关环境管理

网关服务管理

网关API管理

已发布信息管理

平台管理

系统用户管理

服务账号管理

调用记录查询

网关审计管理 - 服务审计管理 - 调用记录查询

查询起始时间: 2018-03-21 00:00

查询结束时间: 2018-03-22 00:00

服务发起方标识: NCR

服务被调用方标识: *

查询服务调用记录

#	调用时间	调用方	被调用方	租户ID	请求方法	请求URL	返回码	响应时间
1	2018/03/21 23:57:05	NCR	adminagency	1abd760639944abfabe88dc546e44ac6	PUT	/api/v1/adminagency/rest/extend/v1/roles/718a2bfd01054a308b715dfefb471c0a	200	90ms
<p>用户信息: User ID=d09cafd59a464774b9cbe703065a2d90; User Name=ncr_yiqiao@163.com</p> <p>租户信息: TenantId=1abd760639944abfabe88dc546e44ac6; TenantName=Project_ncr_yiqiao@163.com</p> <p>请求信息: IP=10.176.91.105; UniqueId=a7b1fd4a-4adc-4b85-b7a2-7d626d4031c0; Token=3218a4d66cf6437a9425b91cb65f98aa</p> <p>请求数据: Request Bod=null</p> <p>响应数据: Response Body={"role":{"name":"service"}}</p>								
2	2018/03/21 23:25:45	NCR	adminagency	1abd760639944abfabe88dc546e44ac6	PUT	/api/v1/adminagency/rest/extend/v1/roles/95e9a13a8e3d4810a309a59a0aefdf53	200	87ms
3	2018/03/21 23:23:09	NCR	vpc	fd23b42940914231a3dde91bccb6d93c	DELETE	/api/v1/vpc/v2.0/ephemeralips/c95c66e6-5e00-4887-9c47-f64360525ce8	200	43ms
4	2018/03/21 23:23:09	NCR	vpc	fd23b42940914231a3dde91bccb6d93c	DELETE	/api/v1/vpc/v2.0/ephemeralips/b8d9e136-1d79-48e3-be5b-dba4c8316247	404	16ms
5	2018/03/21 23:23:09	NCR	vpc	fd23b42940914231a3dde91bccb6d93c	DELETE	/api/v1/vpc/v2.0/fixed-ips/2cf352fd-2c5b-4adc-b12b-9aedce081dbf	404	15ms
6	2018/03/21 23:22:24	NCR	vpc	fd23b42940914231a3dde91bccb6d93c	PUT	/api/v1/vpc/v2.0/ports/a4a5e053-c2c1-4f0e-9f66-f41e682a4760	200	31ms

 **网易云**
共创云上精彩世界