

微服务架构核心20讲

杨波 @2018

微服务架构核心20讲

杨波 / 拍拍贷基础框架研发总监

讲师介绍



杨波

拍拍贷基础框架研发总监

超过10年的互联网分布式系统架构和研发经验的老司机，曾先后就职于：

eBay中国研发中心(eBay CDC)，任资深研发工程师，参与eBay开放平台研发；

携程旅游网(Ctrip)，任框架研发总监，主导携程大规模SOA体系建设；

目前专注于微服务基础架构，DevOps实践，PaaS云平台建设和团队管理

微服务定义 (一)



马丁·福勒 (Martin Fowler)

- 一种架构风格，将单体应用划分成一组小的服务，服务之间相互协作，实现业务功能
- 每个服务运行在独立的进程中，服务间采用轻量级的通信机制协作（通常是HTTP/JSON)
- 每个服务围绕业务能力进行构建，并且能够通过自动化机制独立地部署
- 很少有集中式的服务管理，每个服务可以使用不同的语言开发，使用不同的存储技术
- 参考：<https://www.martinfowler.com/articles/microservices.html>

微服务定义 (二)



Adrian Cockcroft

- Loosely coupled service oriented architecture with bounded context
- 基于有界上下文的，松散耦合的面向服务的架构

微服务的利和弊

- 利

- 强模块化边界
- 可独立部署
- 技术多样性

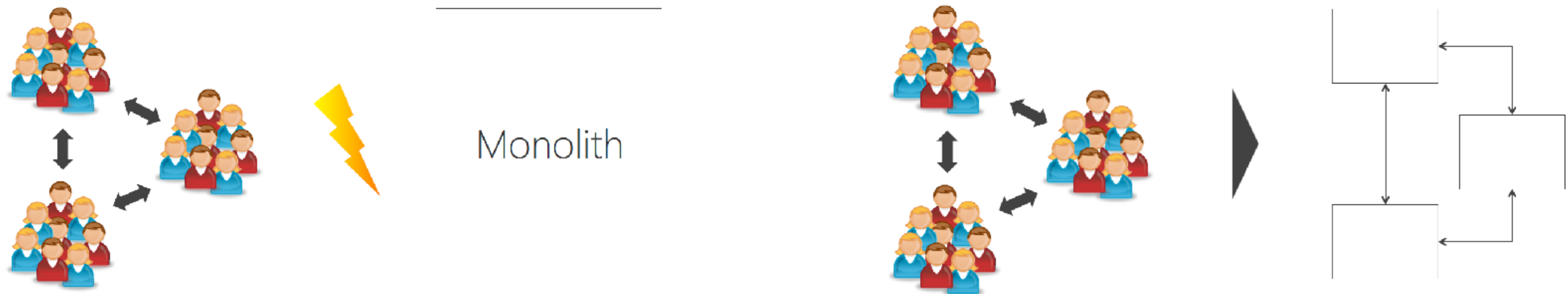
- 弊

- 分布式系统复杂性
- 最终一致性
- 运维复杂性
- 测试复杂性

如果你搞不定一个单块应用，别指望微服务能够拯救你！

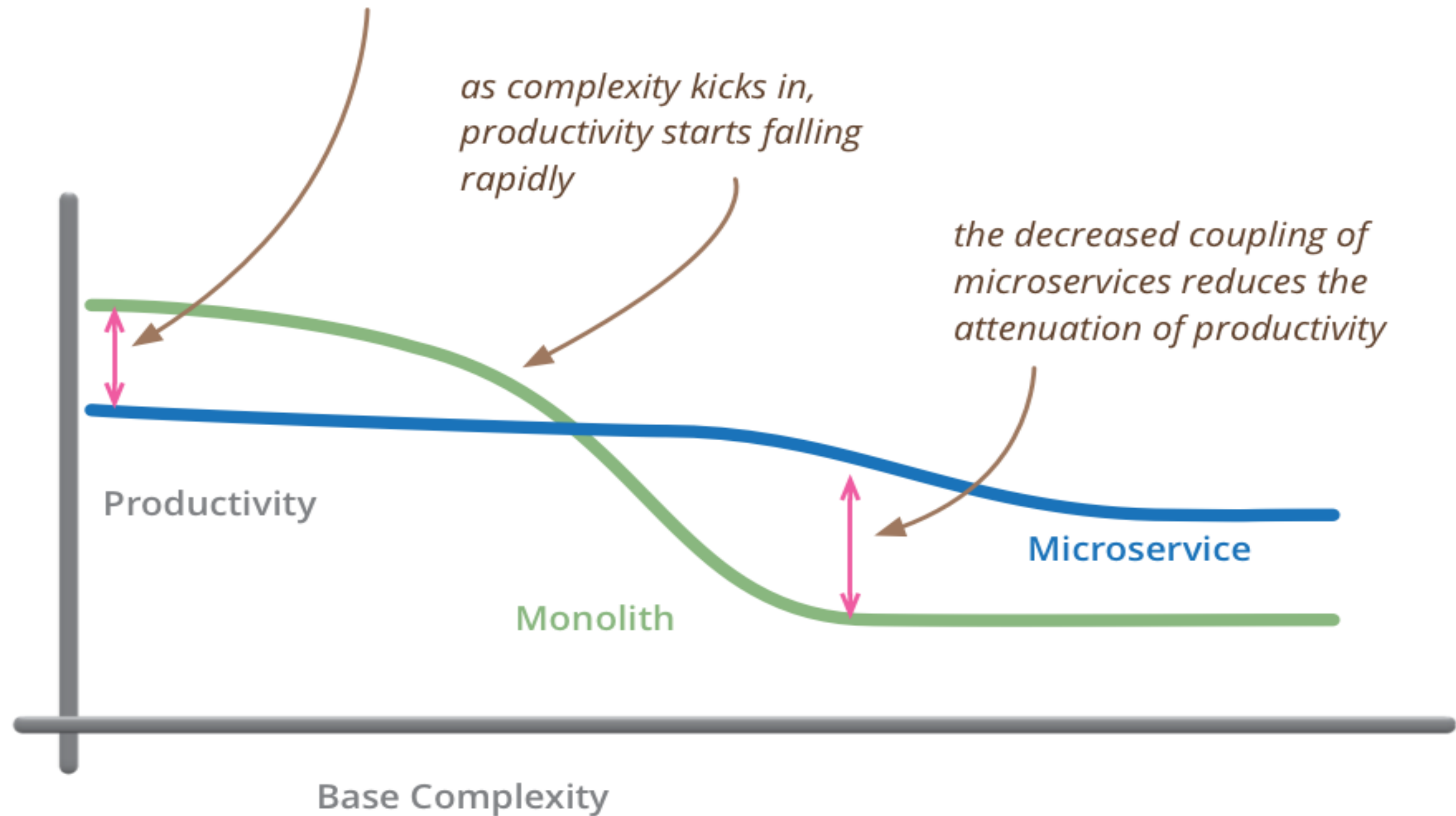
康威法则

- Organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations.
- 设计系统的组织，其产生的架构设计等价于组织间的沟通结构



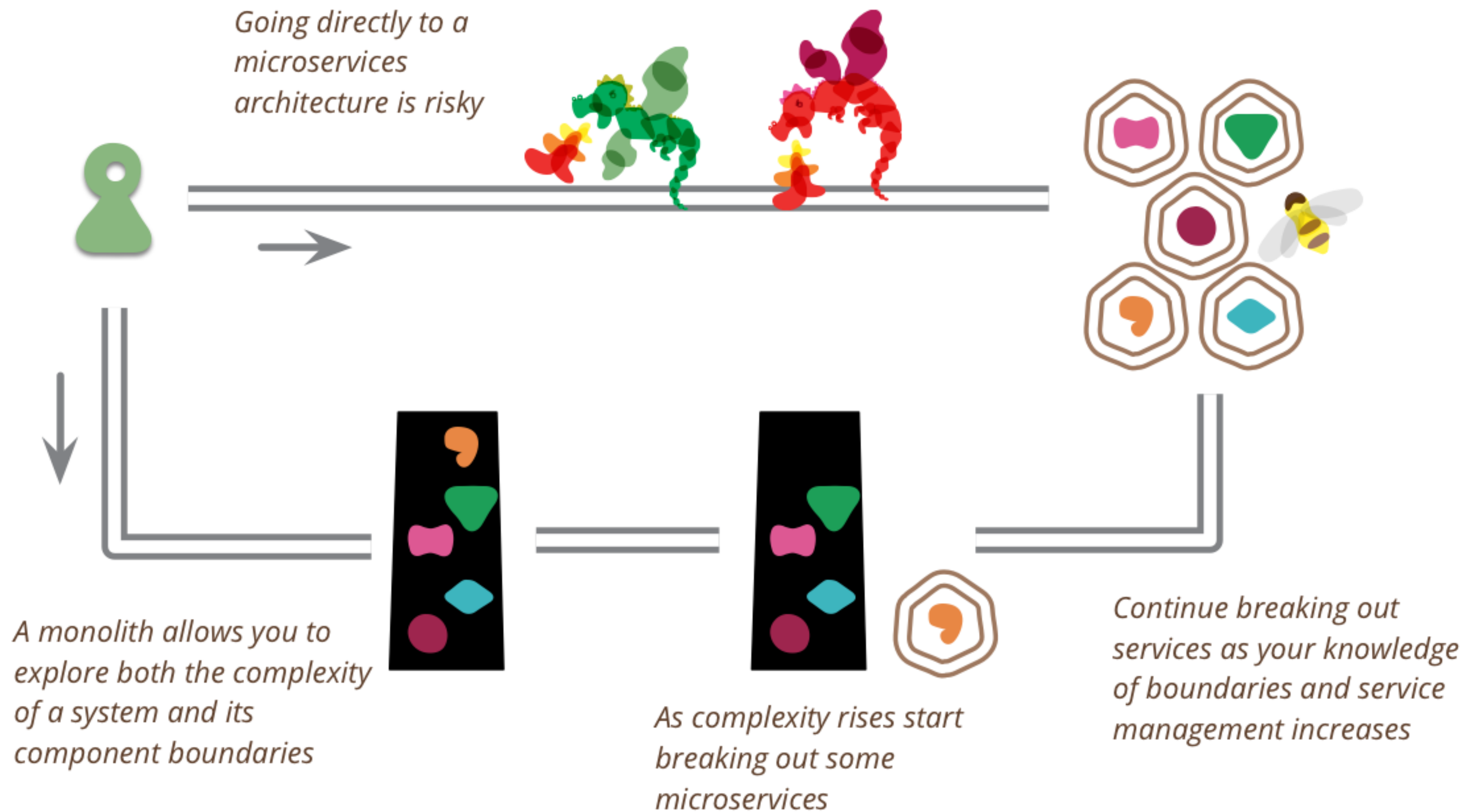
微服务的适用性

for less-complex systems, the extra baggage required to manage microservices reduces productivity



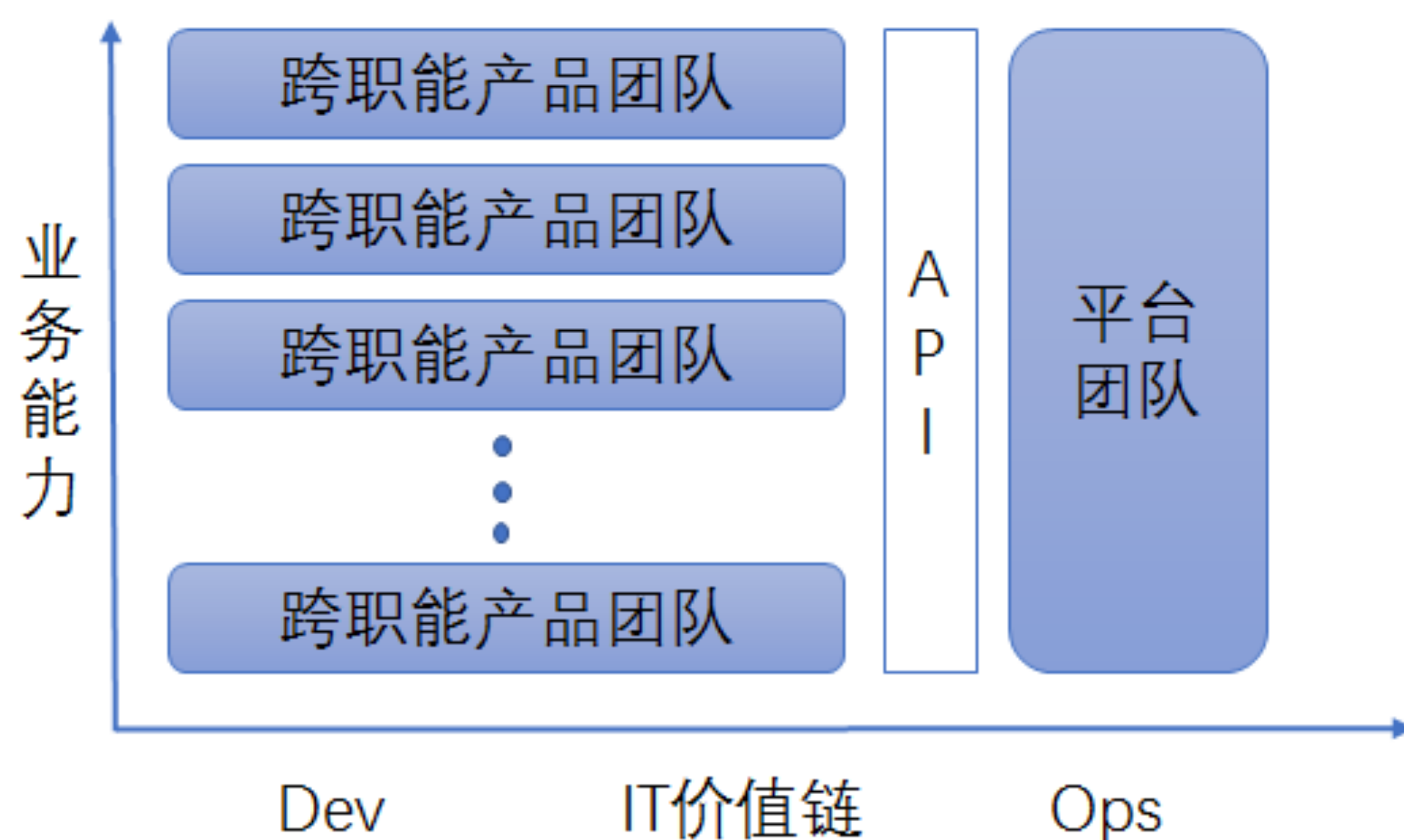
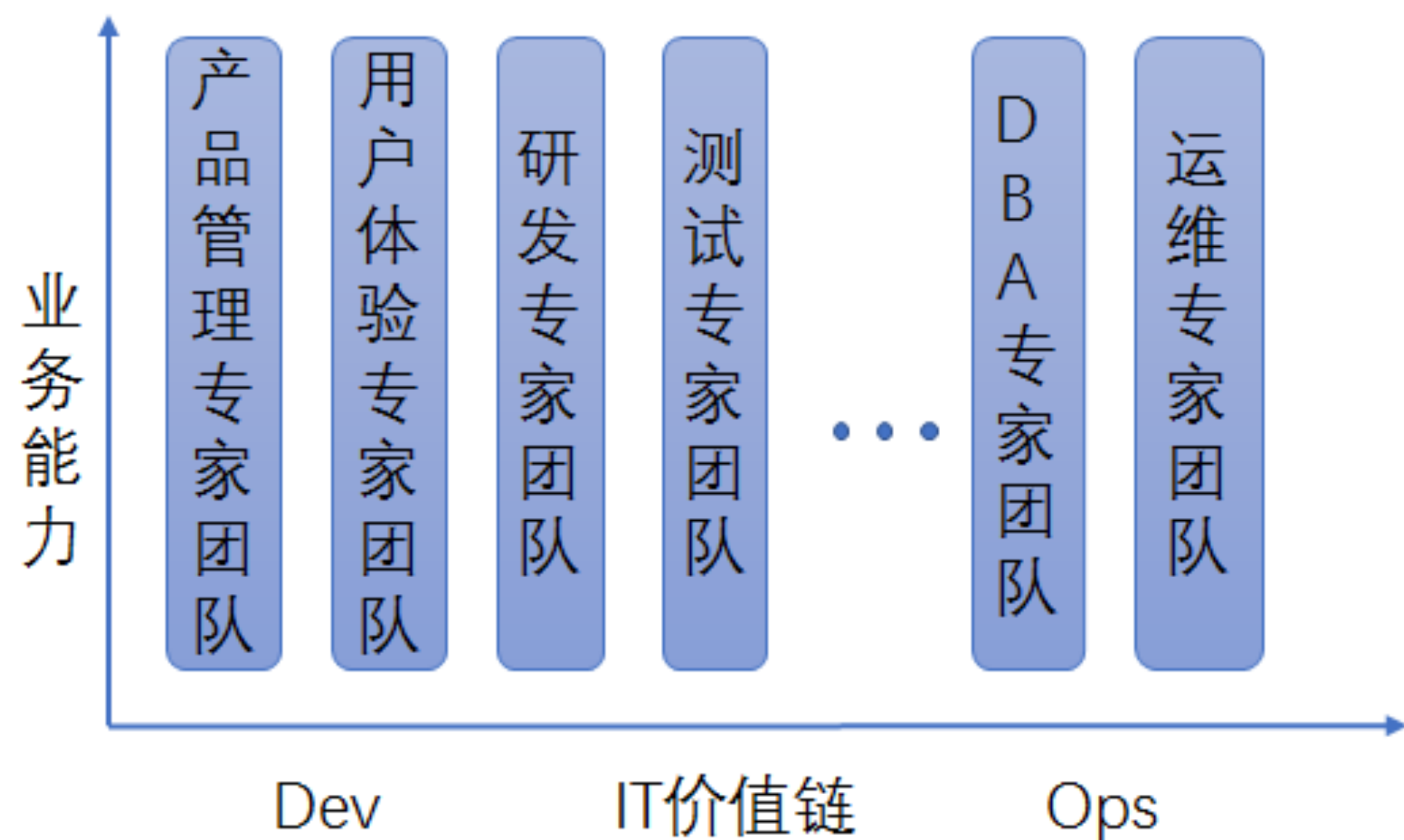
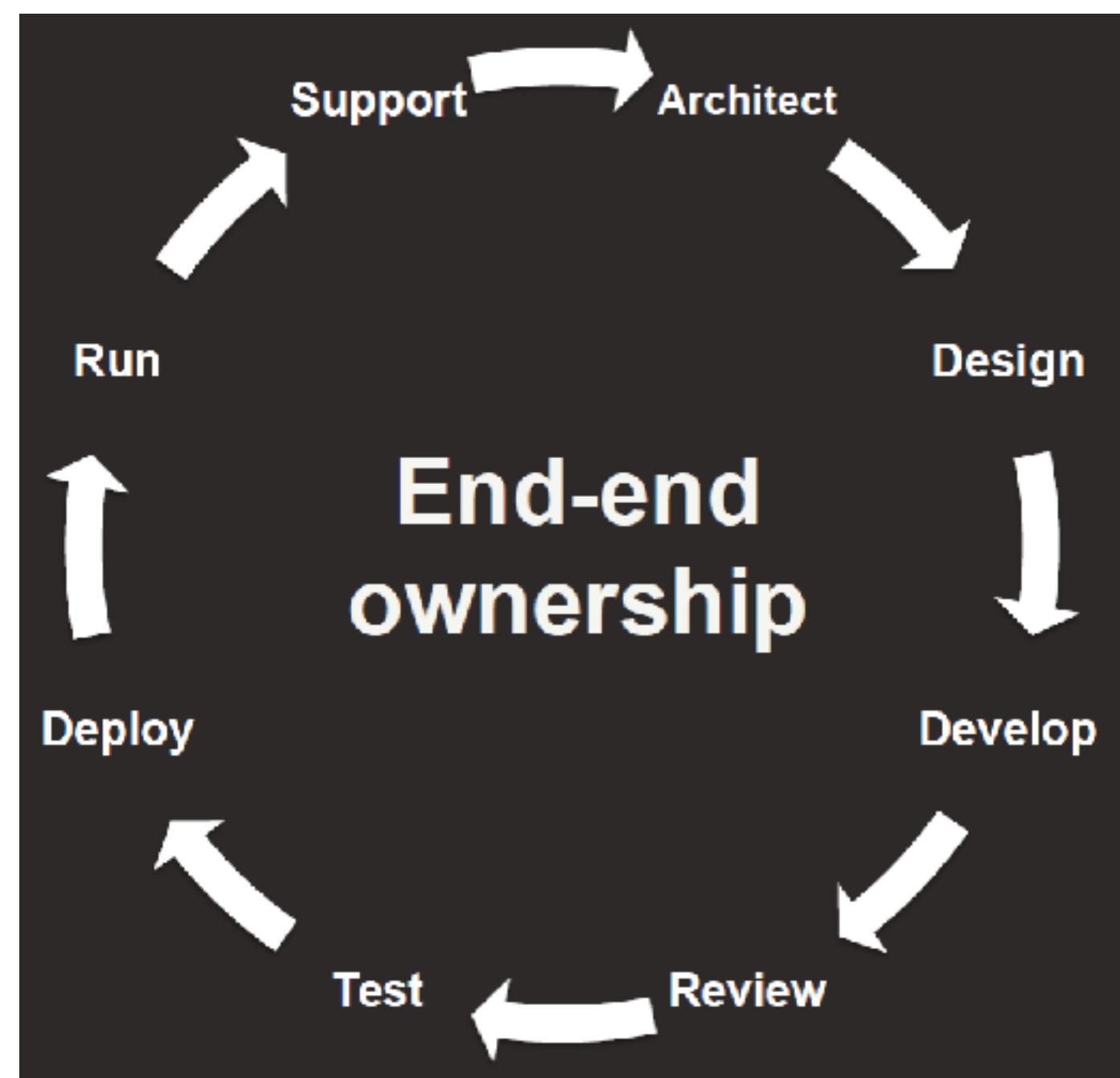
but remember the skill of the team will outweigh any monolith/microservice choice

单块优先

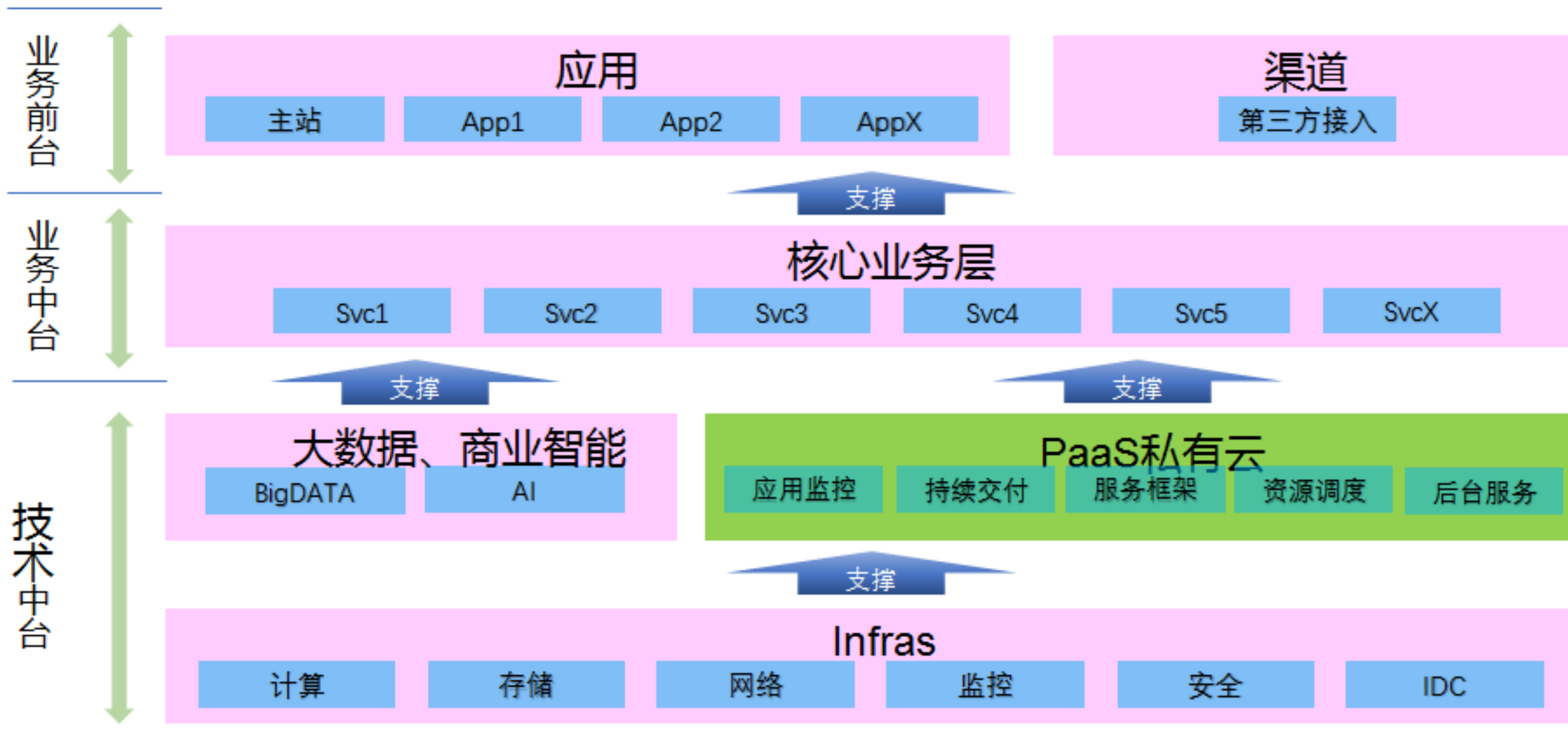


微服务组织架构

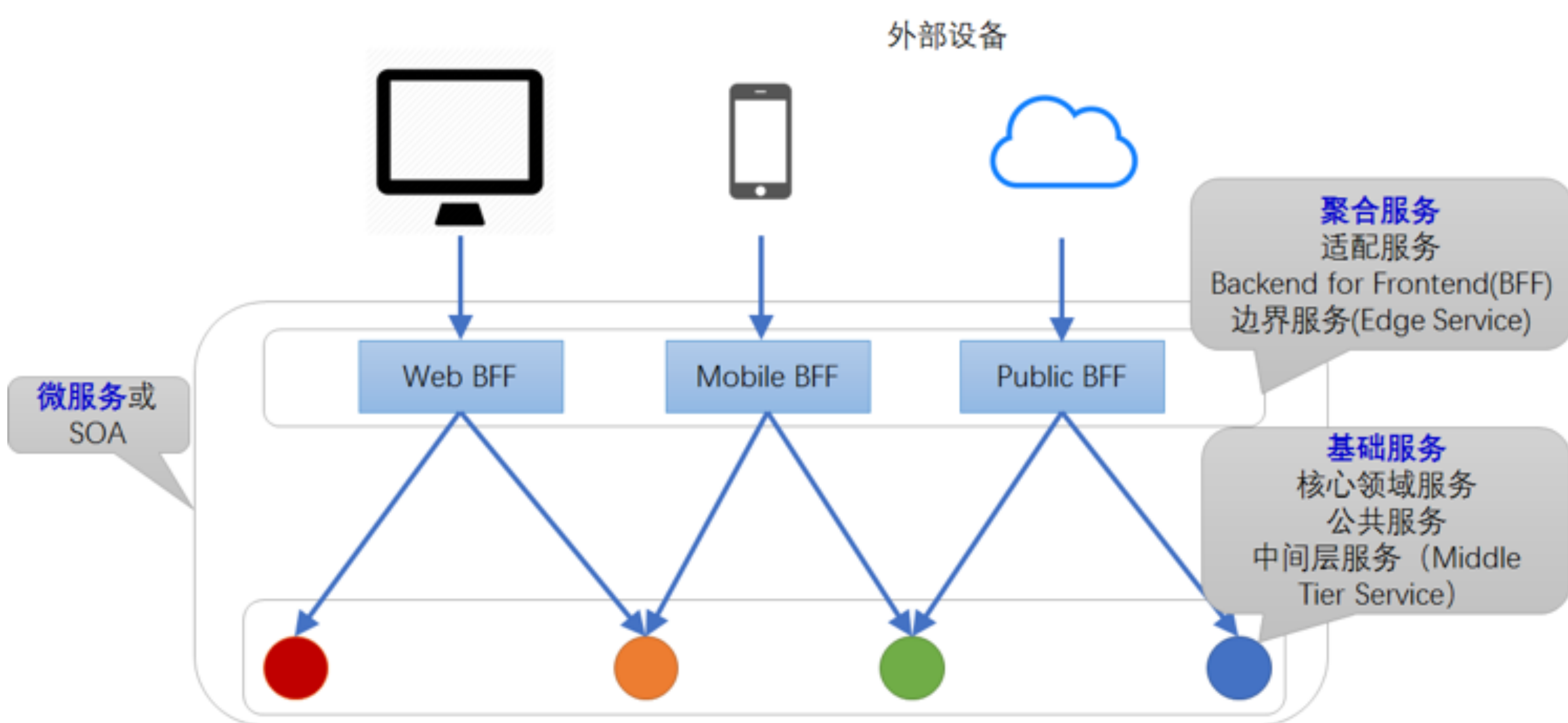
- 从传统职能型到跨职能产品型



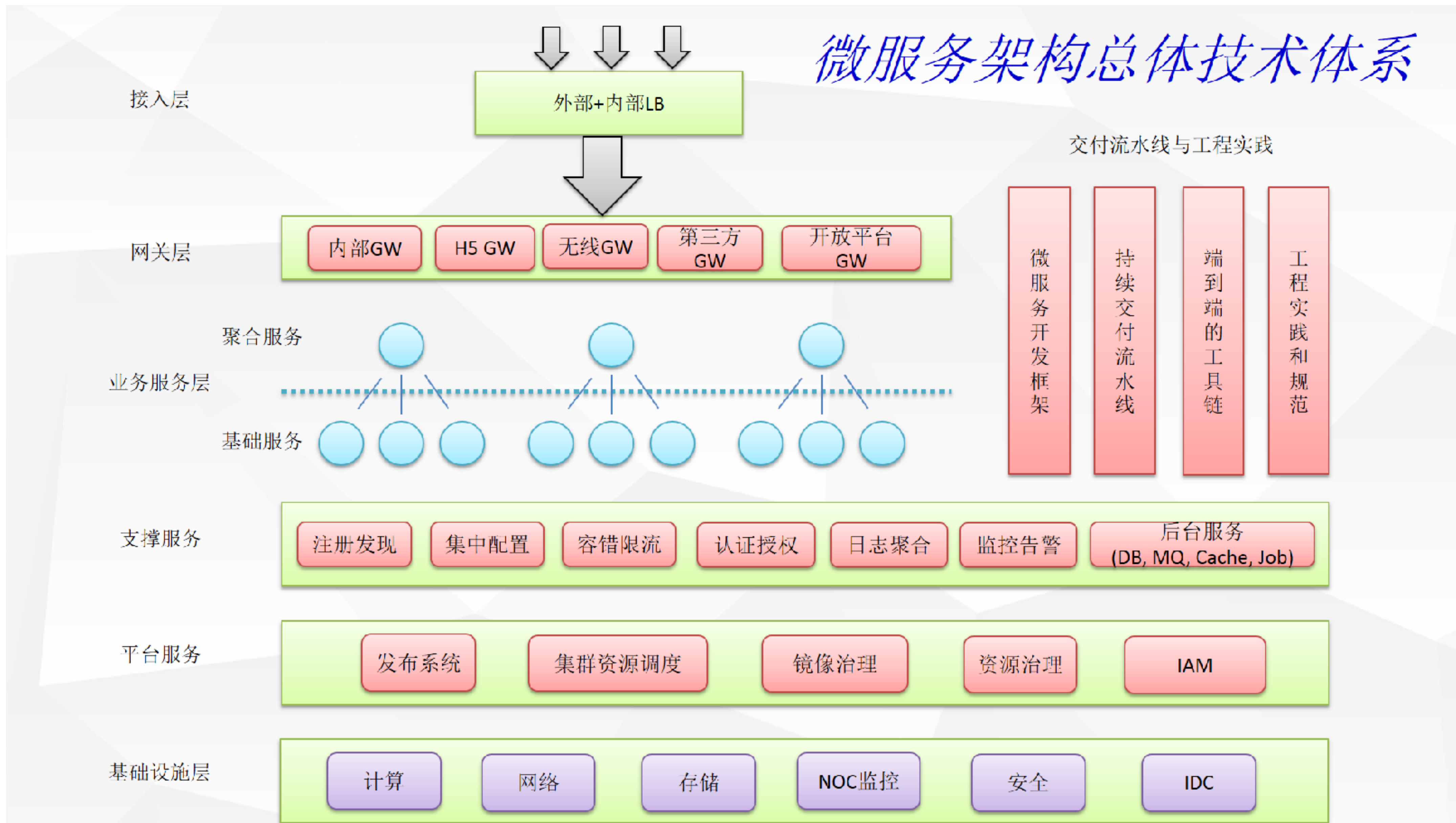
中台战略和微服务



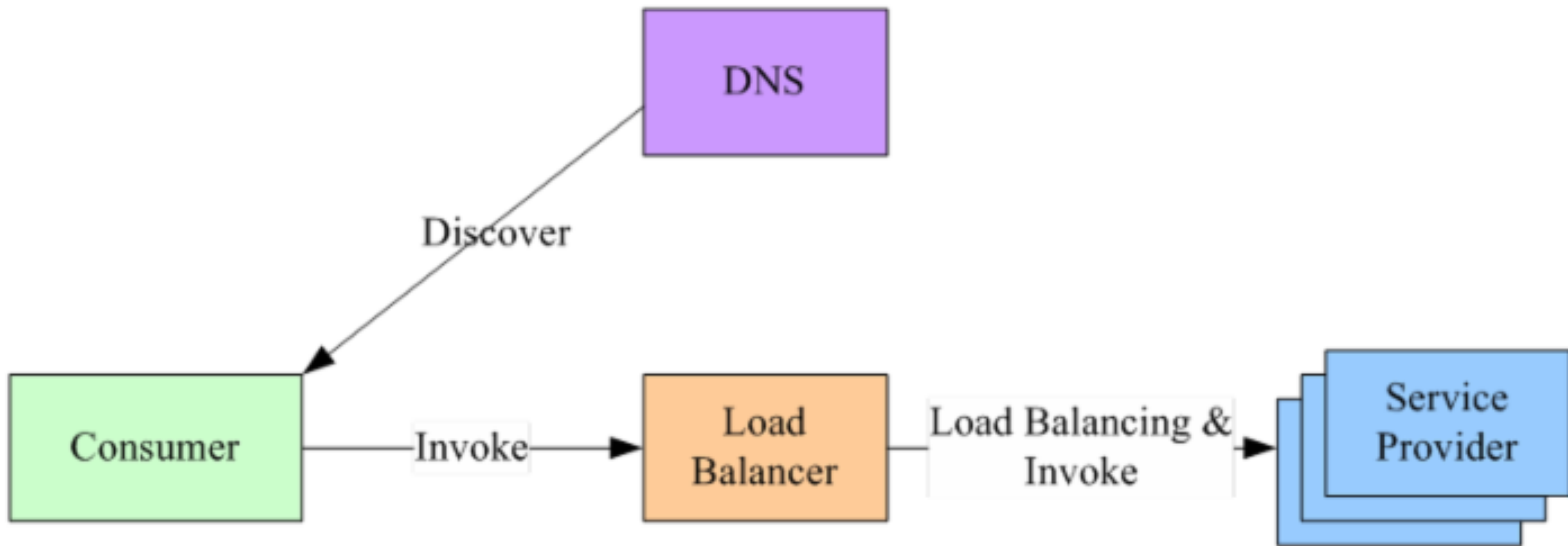
服务分层概念



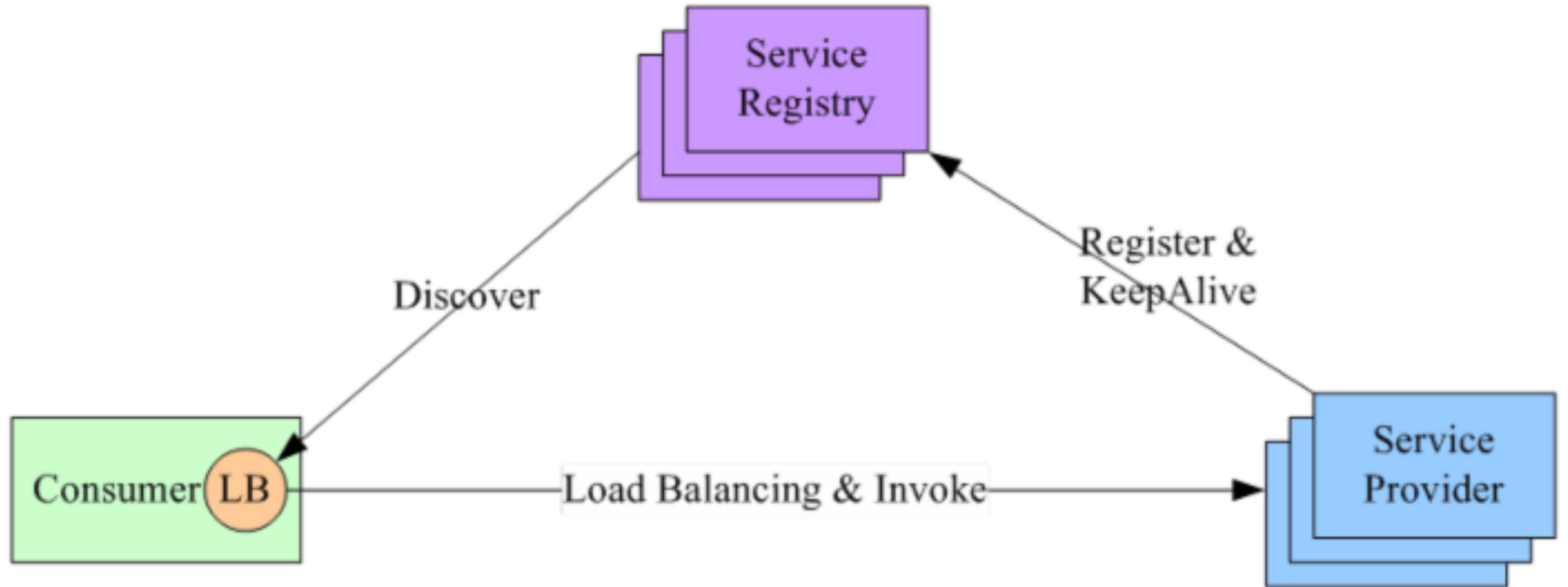
微服务总体架构体系图



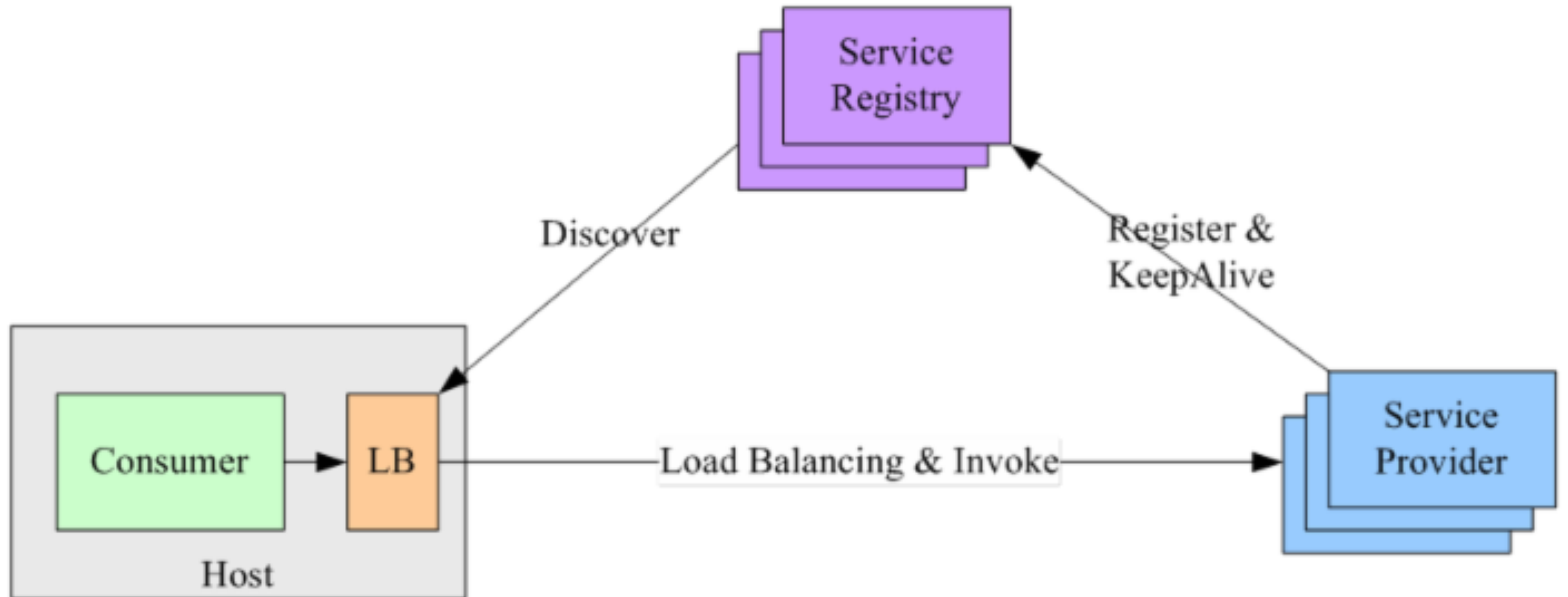
服务发现~独立LB



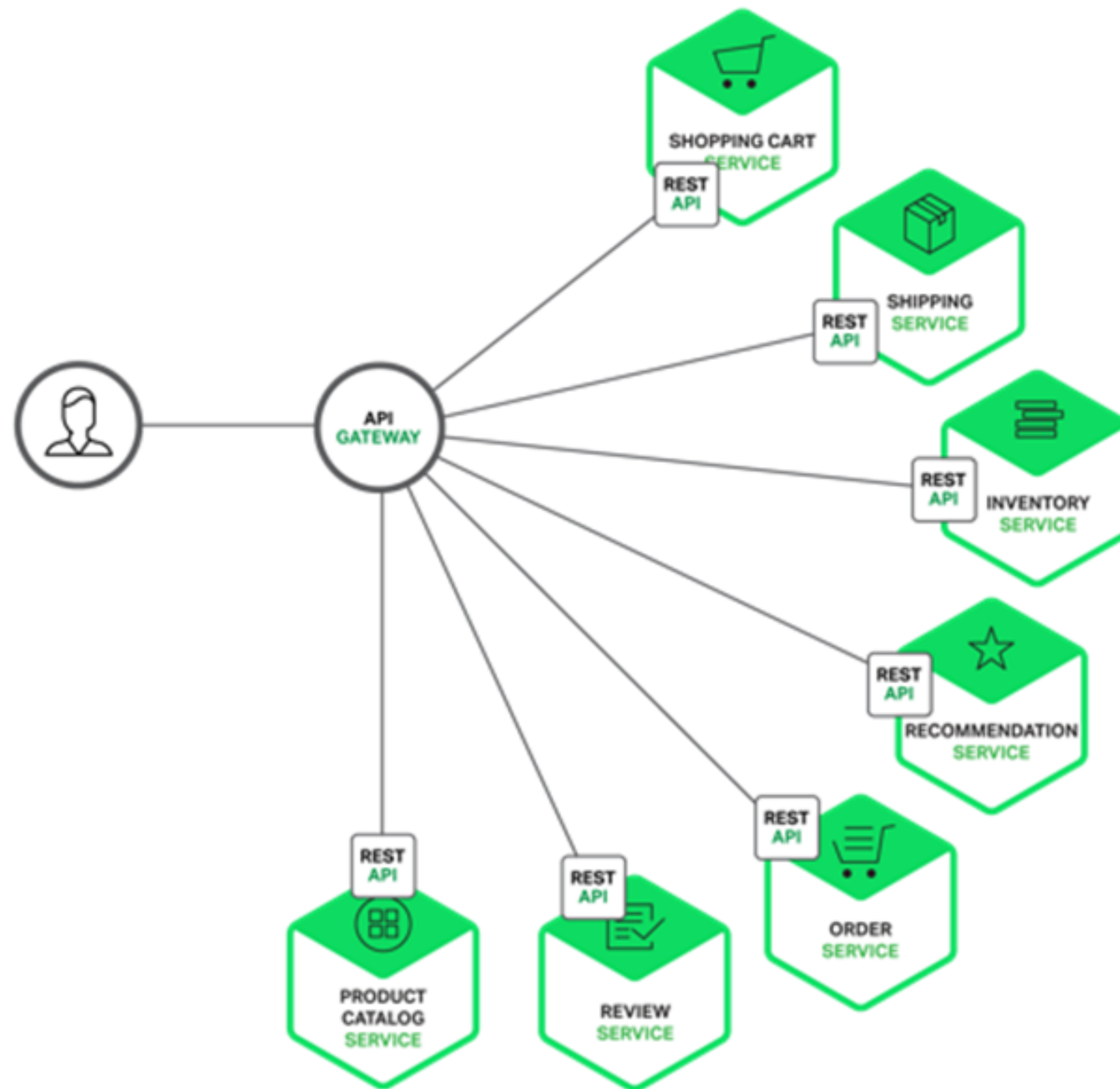
服务发现~进程内LB



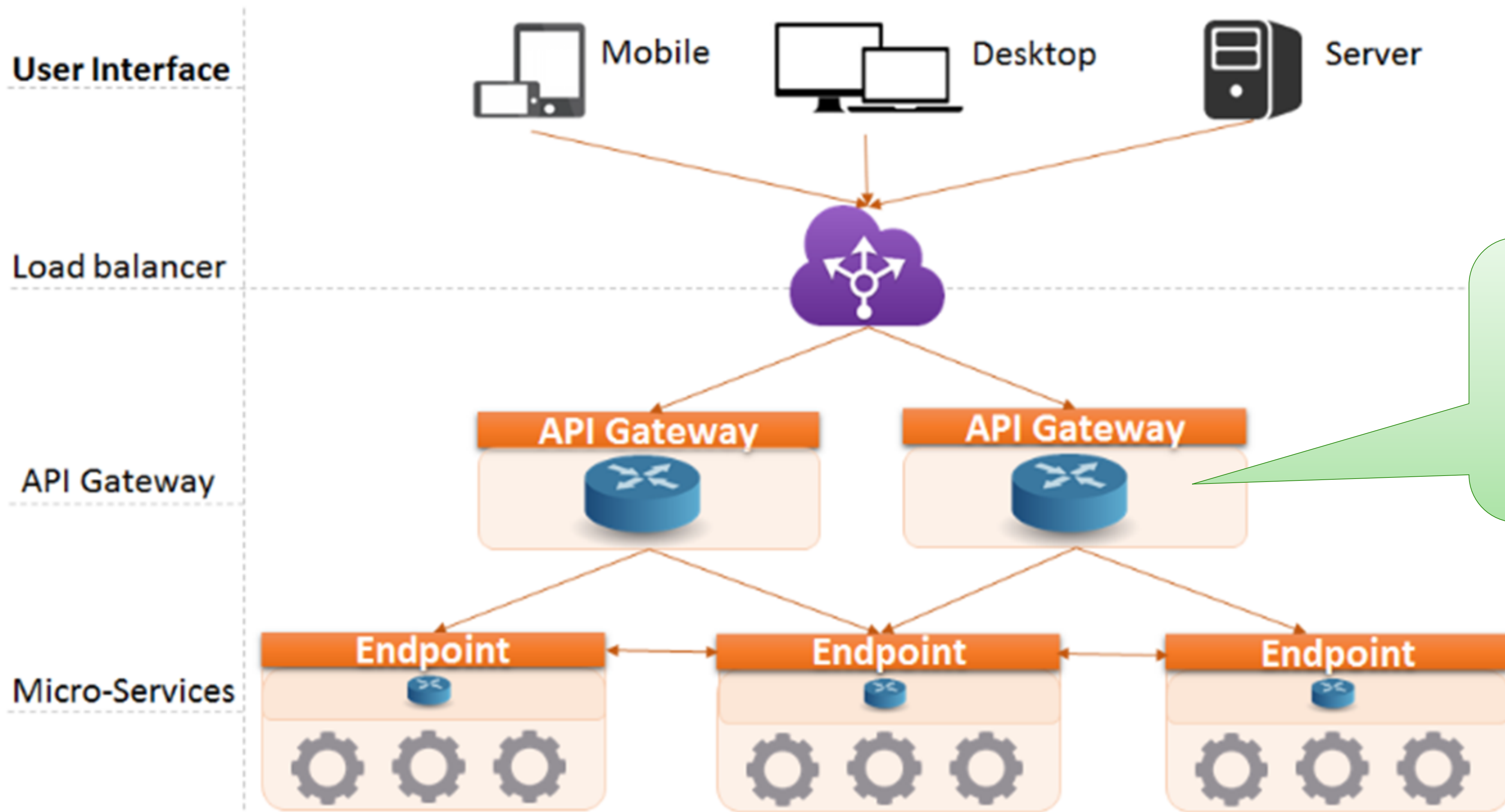
服务发现~主机独立进程LB



微服务网关

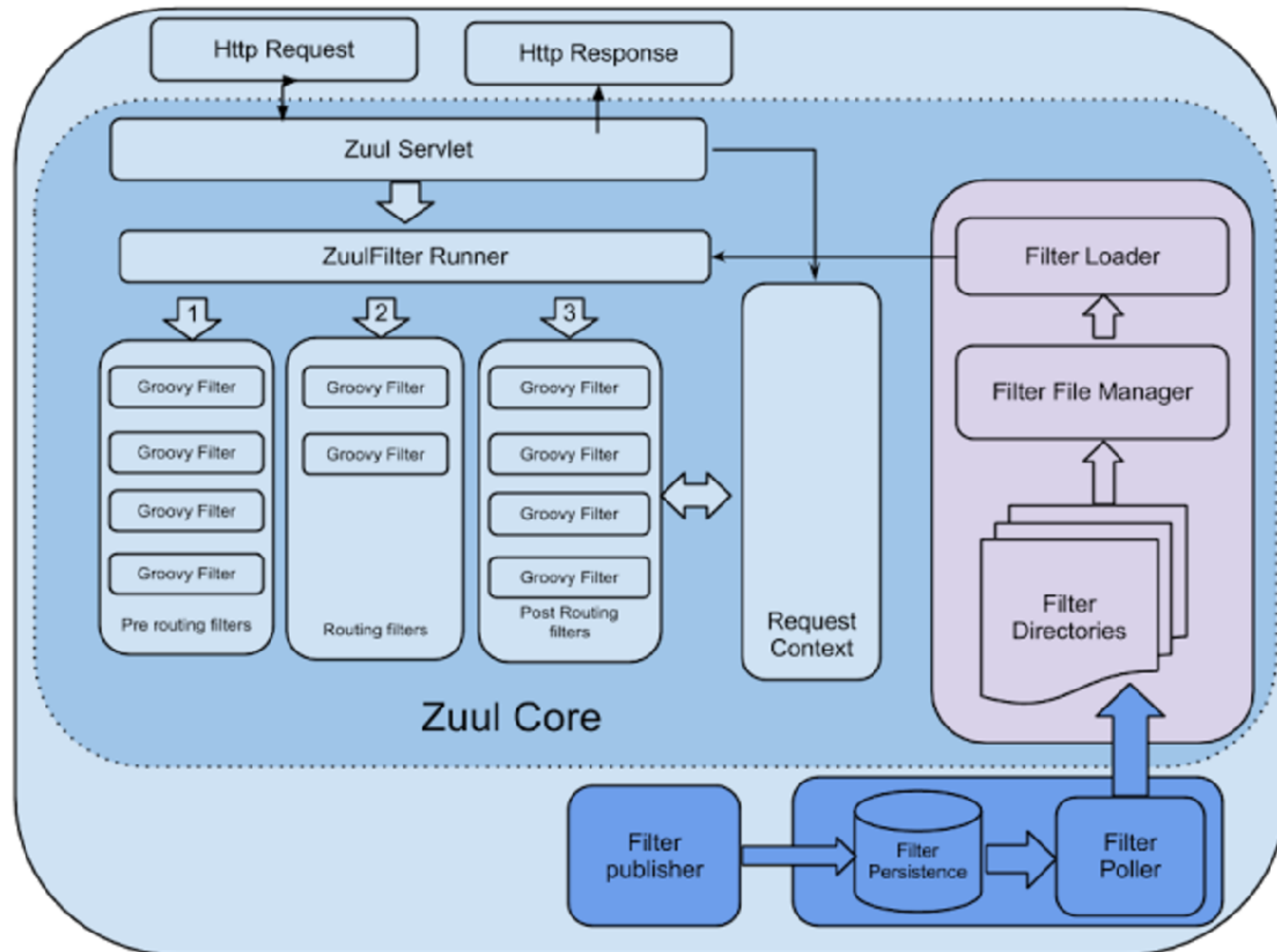


微服务网关



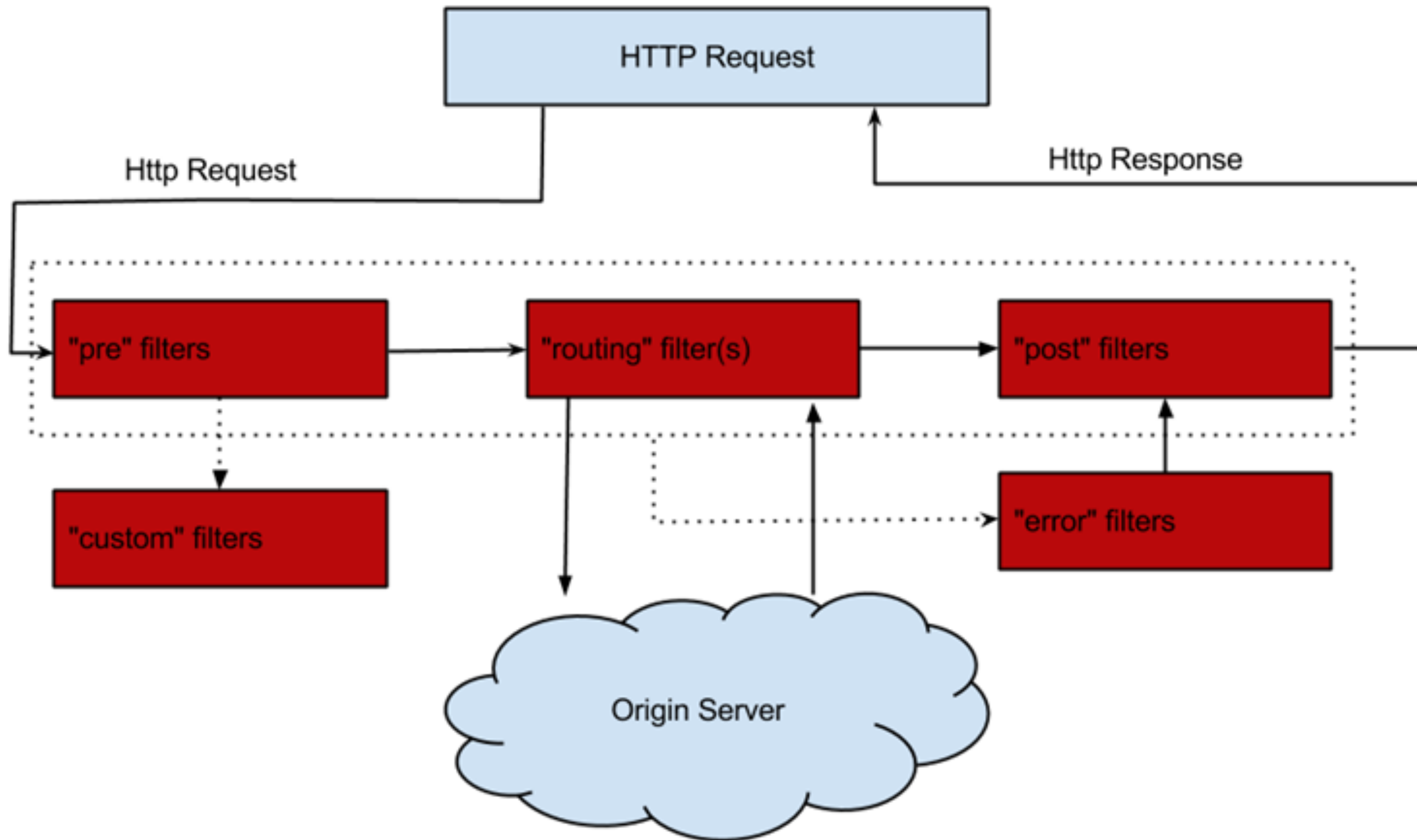
反向路由
认证安全
限流熔断
日志监控

Netflix Zuul 网关架构 (一)

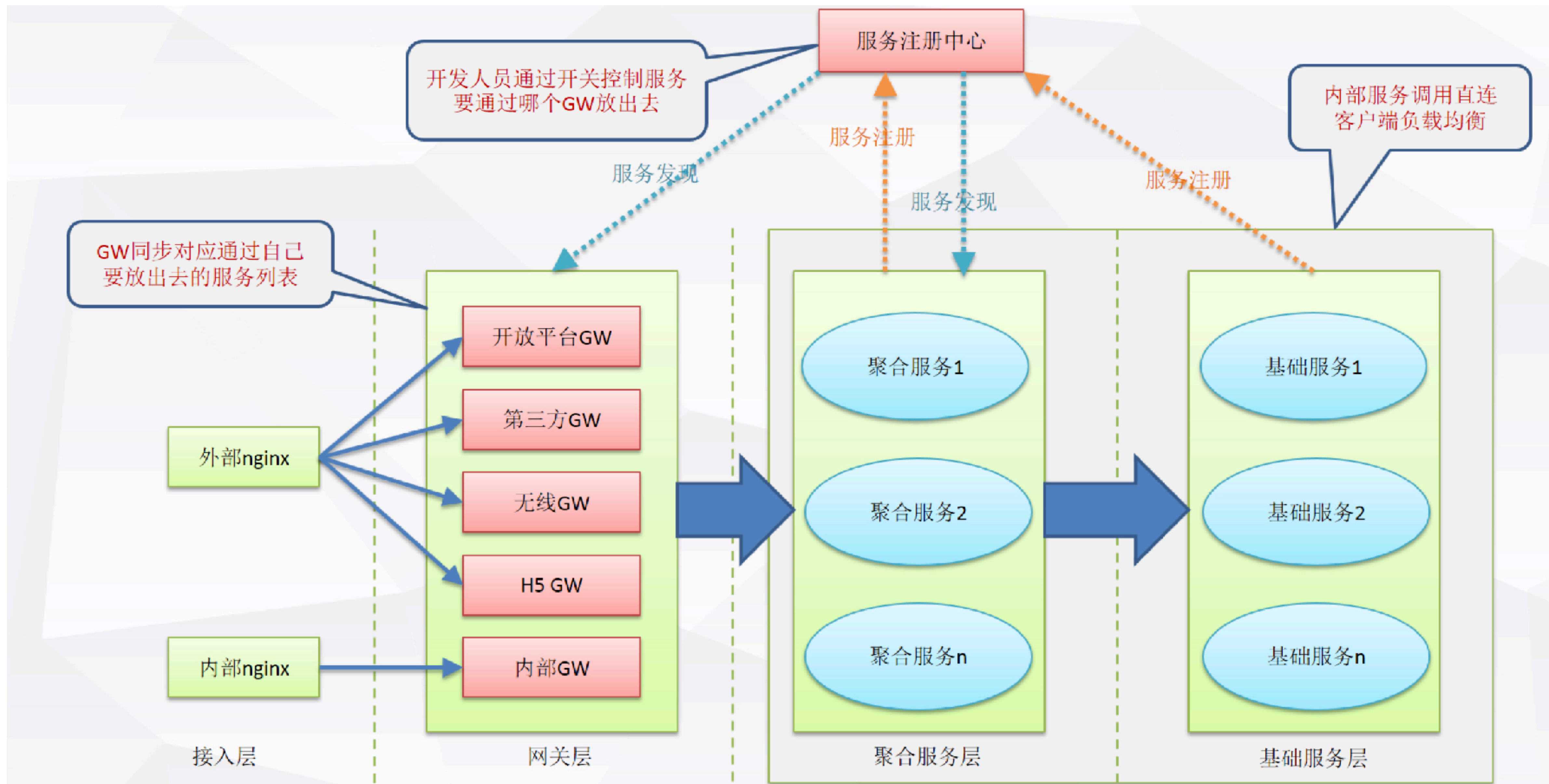


Zuul Core Architecture

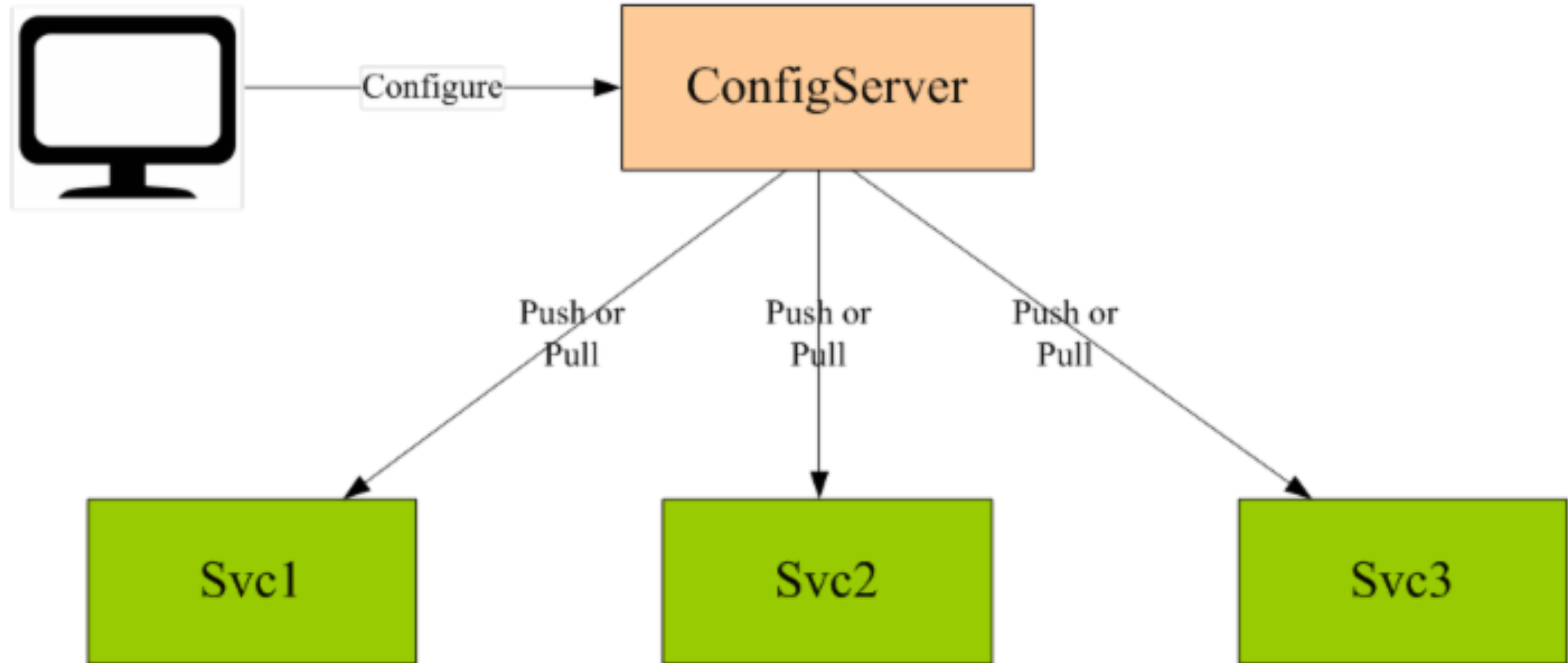
Netflix Zuul 网关架构 (二)



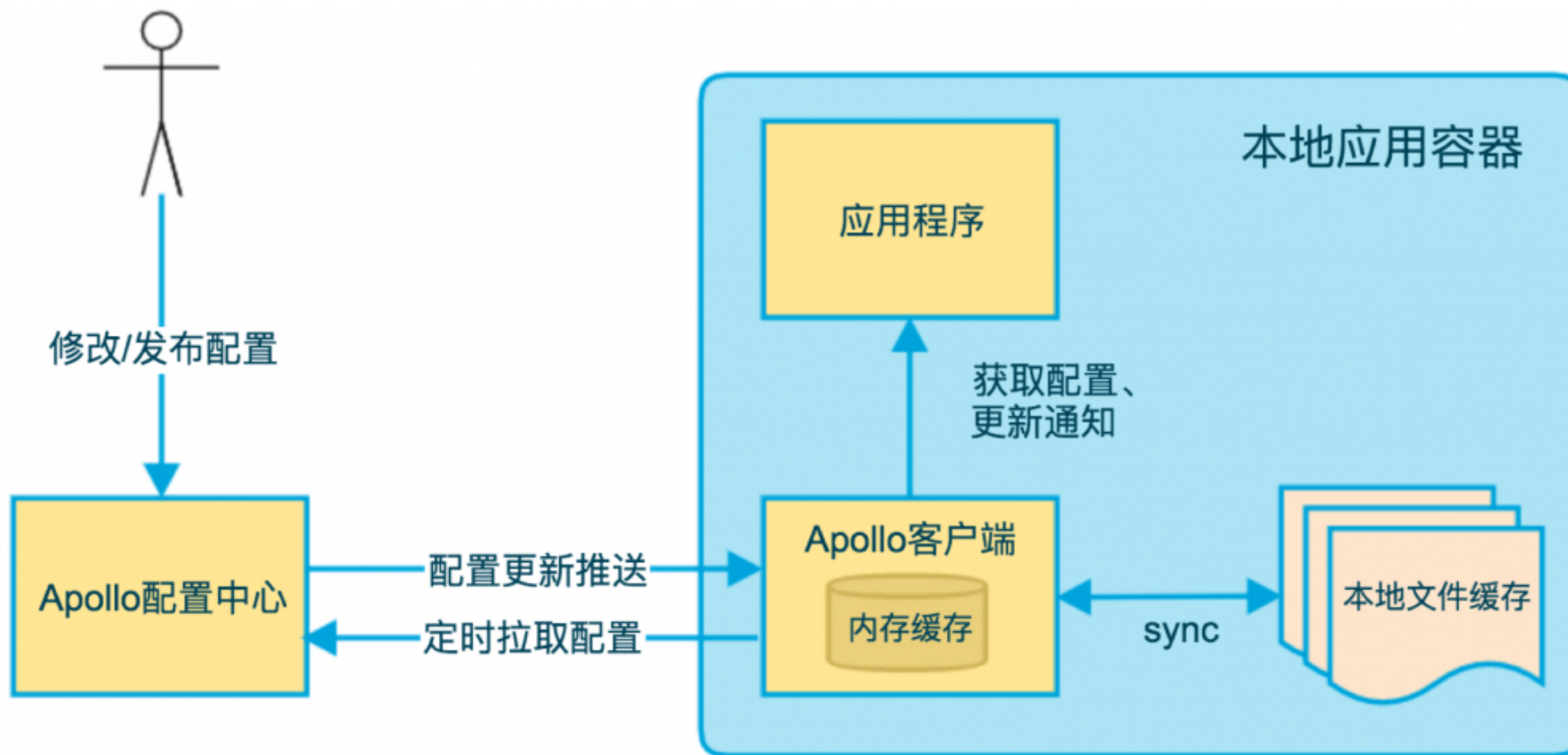
微服务服务发现路由体系



微服务配置中心原理（一）



微服务配置中心原理 (二)



RPC vs REST

	RPC	REST
耦合性	强耦合	松散耦合
消息协议	二进制thrift, protobuf, avro	文本XML, JSON
通讯协议	TCP	HTTP/HTTP2
性能	高	一般低于RPC
接口契约IDL	Thrift, protobuf idl	Swagger
客户端	强类型客户端, 一般自动生成, 可支持多语言客户端	一般HTTP client可访问, 也可以自动生成强类型客户端, 可支持多语言客户端
案例	Dubbo, motan, Tars, grpc, thrift	Spring MVC/Boot, Jax-rs, dropwizard
开发者友好	客户端比较方便, 但是二进制消息不可读	文本消息开发者可读, 浏览器就可以访问
对外开放	对外一般需要转换成REST/文本协议	直接可以对外开放

服务框架和治理



四层监控体系和监控分类

端用户体验监控
(性能, 返回码、城市、地区、运营商、版本、系统等)

业务监控
(核心指标监控, 登录注册, 下单, 支付等)

应用层监控
(url, service, sql, cache可用率, 响应时间, qps等)

系统层监控 (物理机, 虚拟机, OS)
(cpu, memory, network, disk等)

基础设施监控 (网络, 交换机)
(网络流量, 丢包, 错包, 连接数等)

日志监控

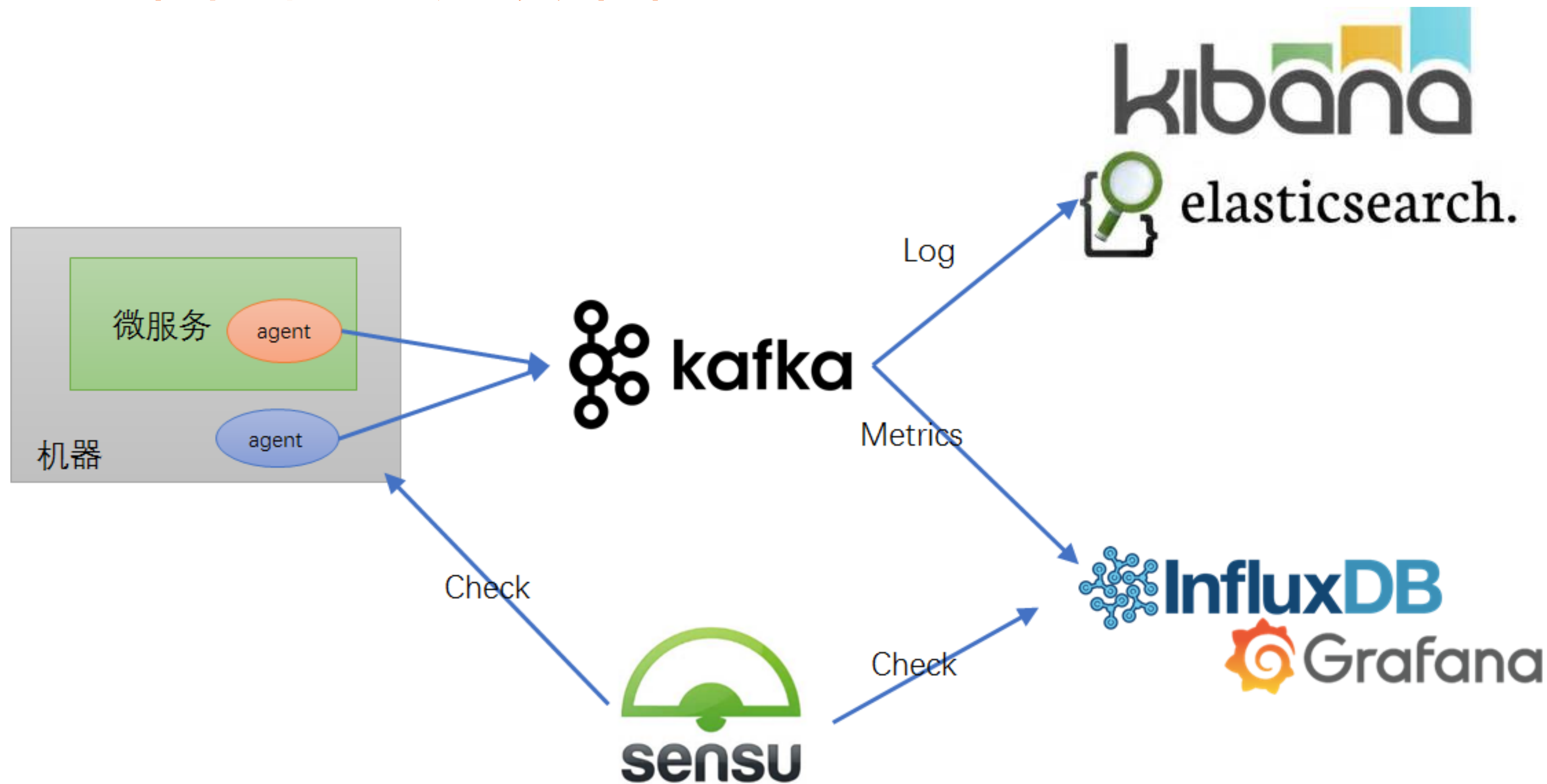
Metrics监控

调用链监控

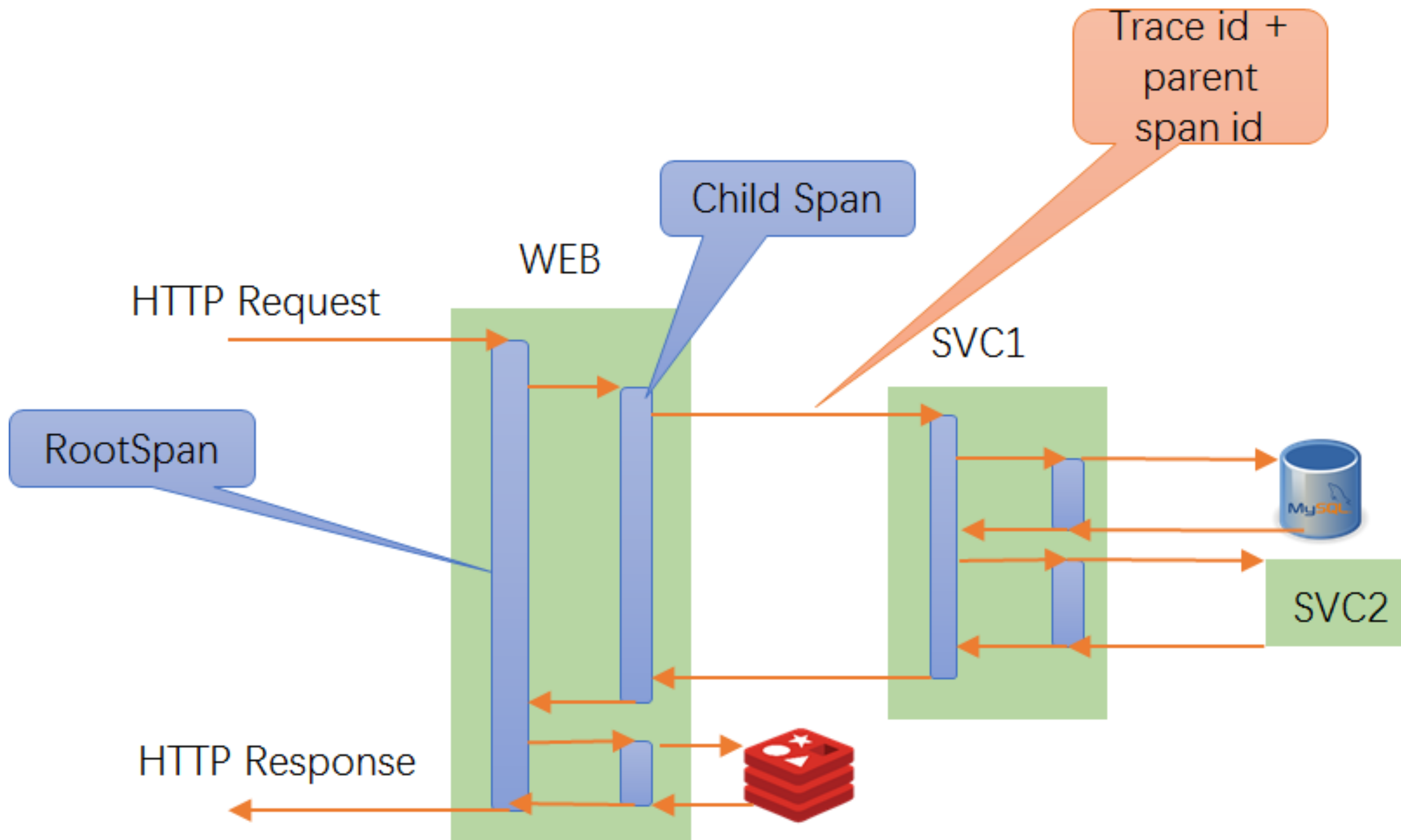
告警系统

健康检查

监控架构和主流技术栈



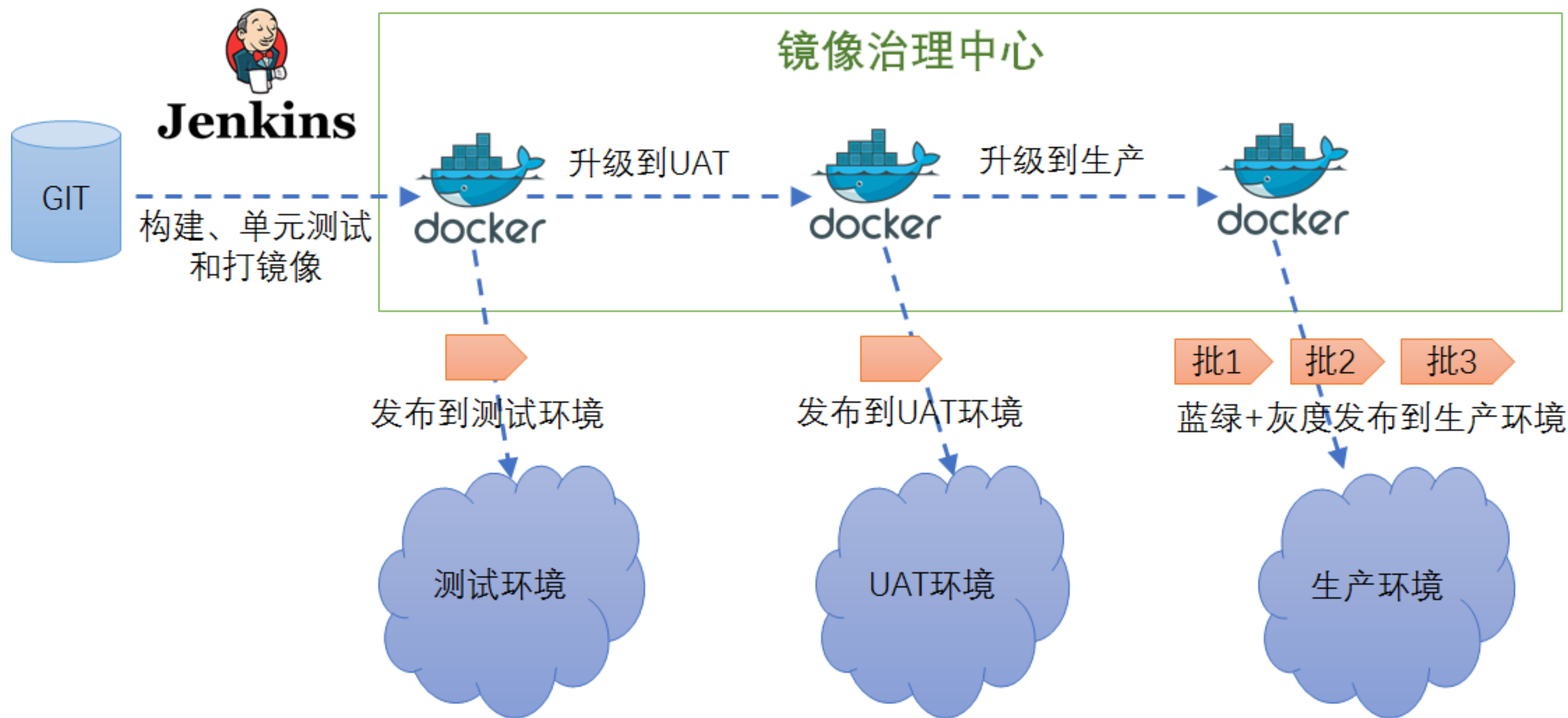
Trace调用链监控原理



调用链监控选型

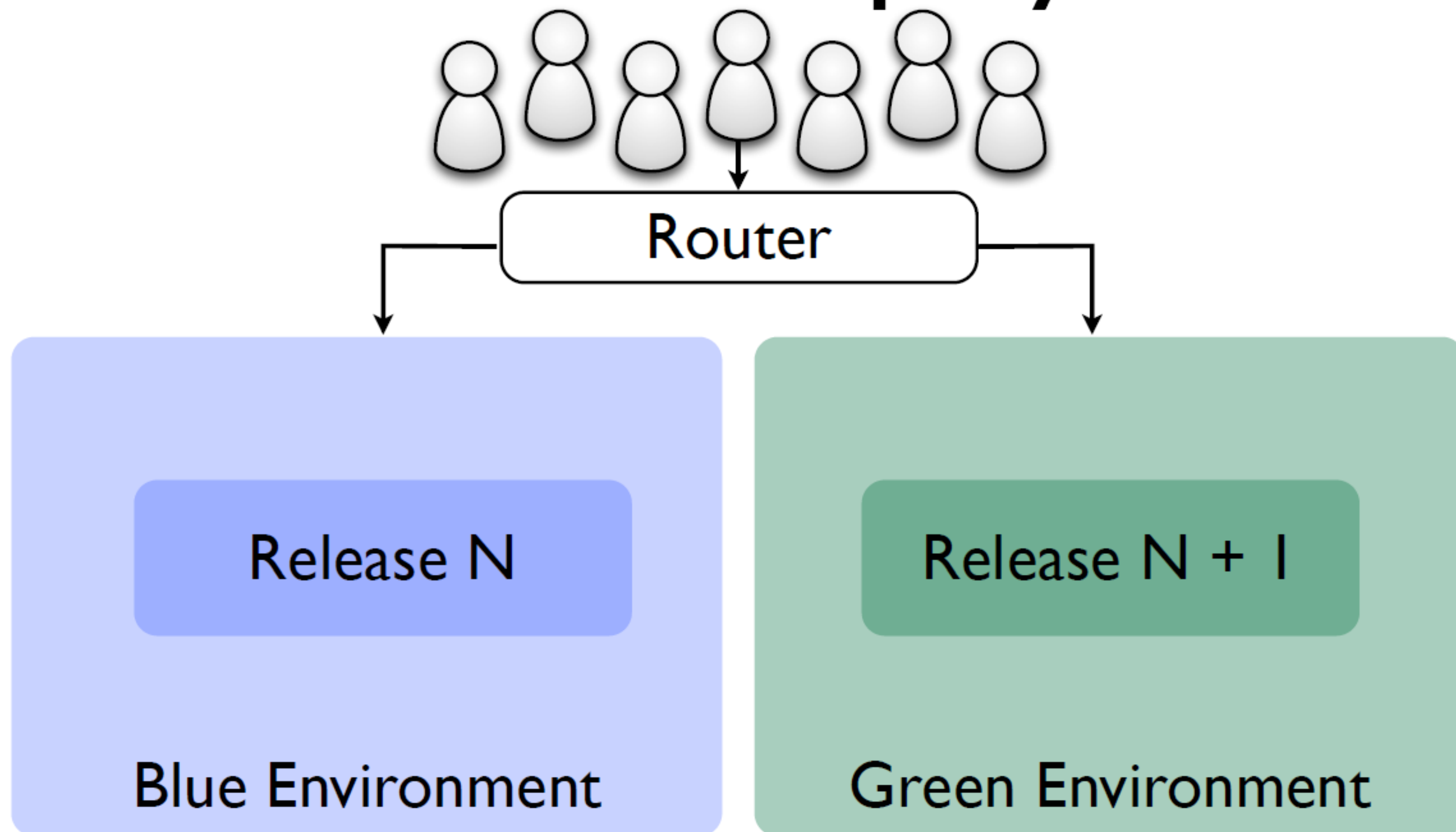
	CAT	Zipkin	Pinpoint
调用链可视化	有	有	有
报表	非常丰富	少	中
ServerMap	简单依赖图	简单	好
埋点方式	侵入	侵入	不侵入字节码增强
Heartbeat支持	有	无	有
Metric支持	有	无	无
Java/.Net客户端支持	有	有	只有Java
Dashboard中文支持	好	无	无
社区支持	好，文档较丰富，作者在携程点评	好，文档一般，暂无中文社区	一般，文档缺，无中文社区
国内案例	携程、点评、陆金所	京东、阿里不开源	暂无
源头祖先	eBay CAL ~ Centralized Application Logging	Google Dapper	Google Dapper

基于镜像治理和多环境的持续交付流水线

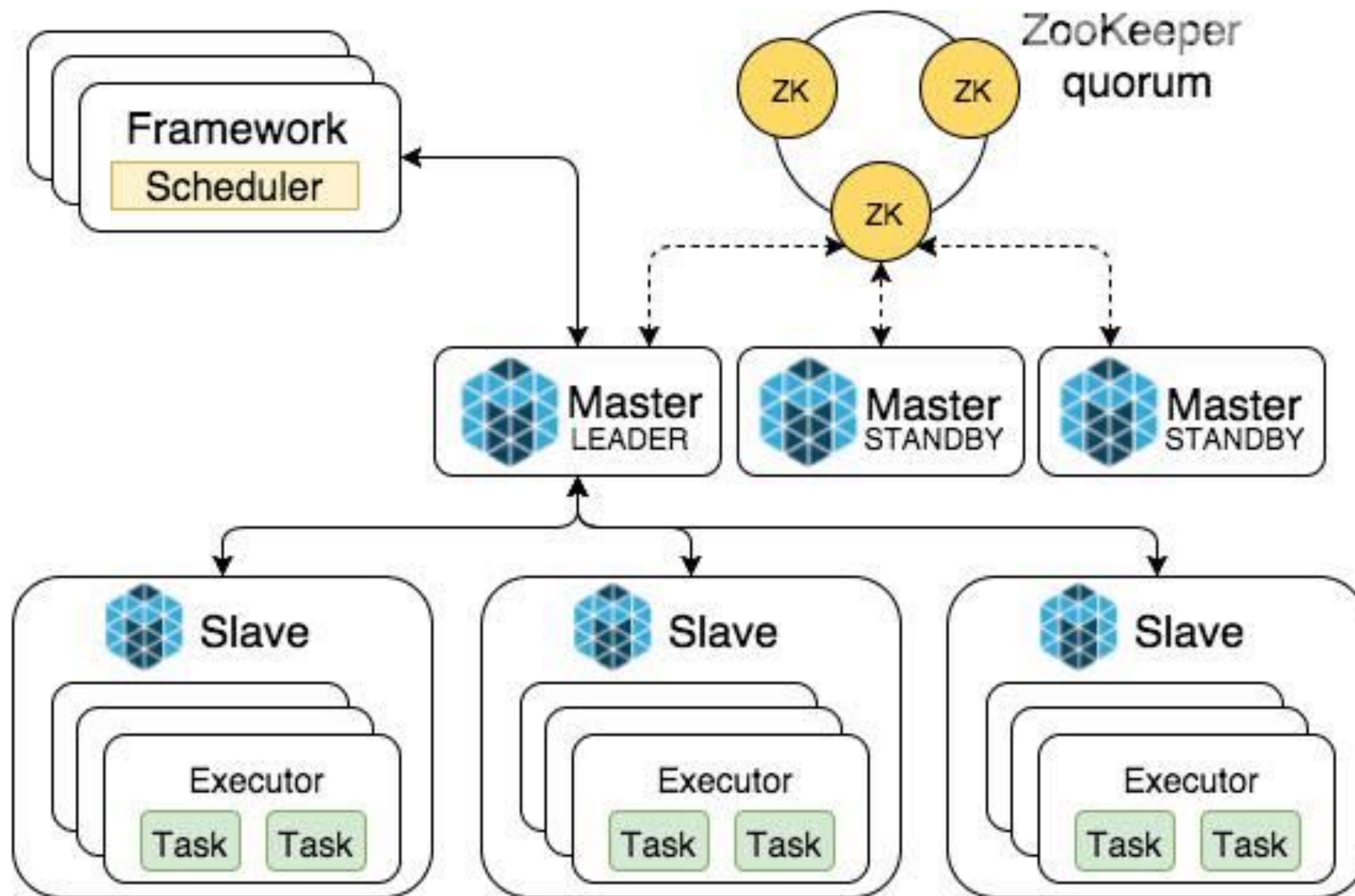


蓝绿、金丝雀和灰度发布

Blue Green Deployments

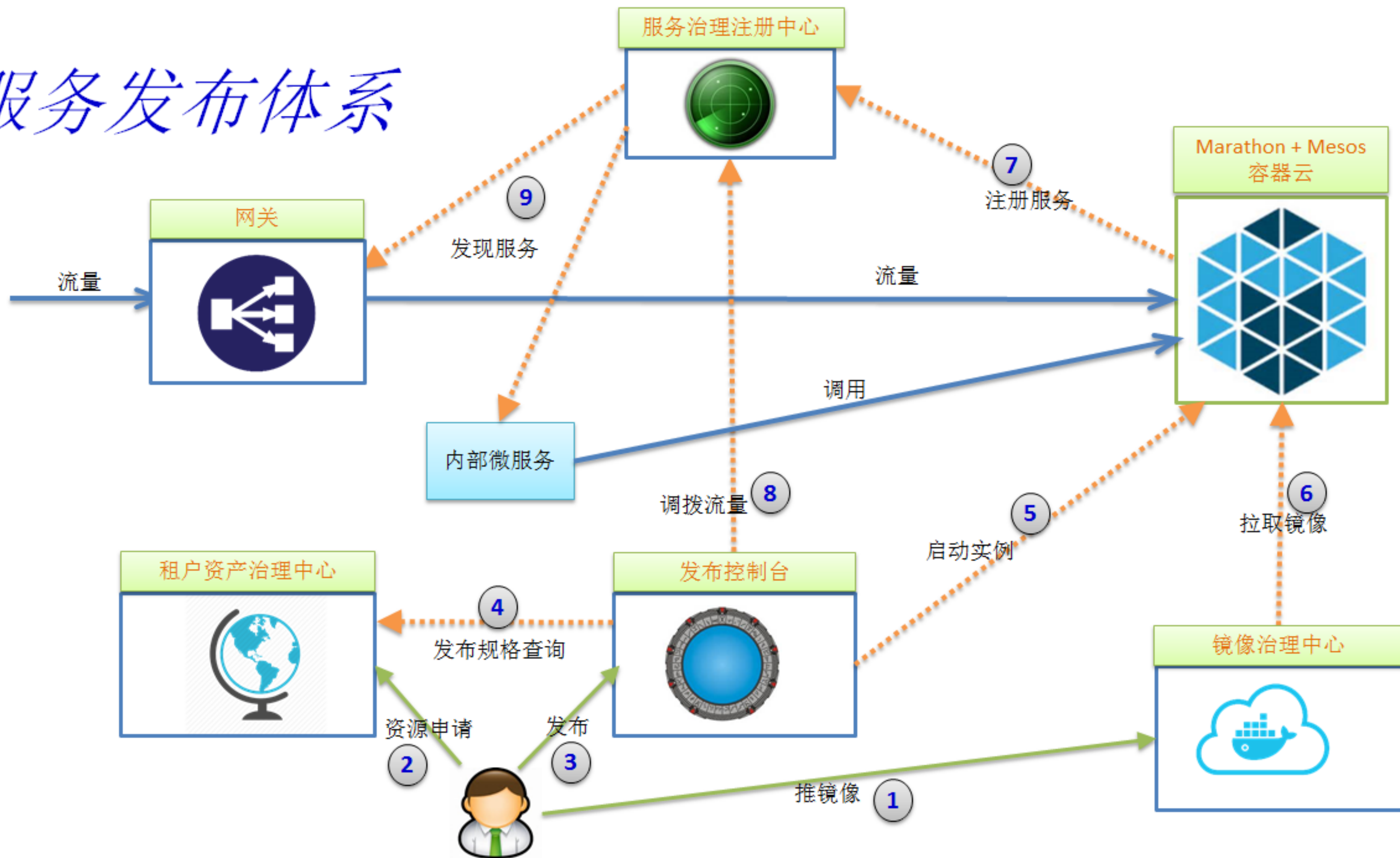


资源调度框架 Mesos 架构



基于容器云发布体系

服务发布体系



THANKS

重拾极客精神 · 提升技术认知

time.geekbang.org