

微服务介绍

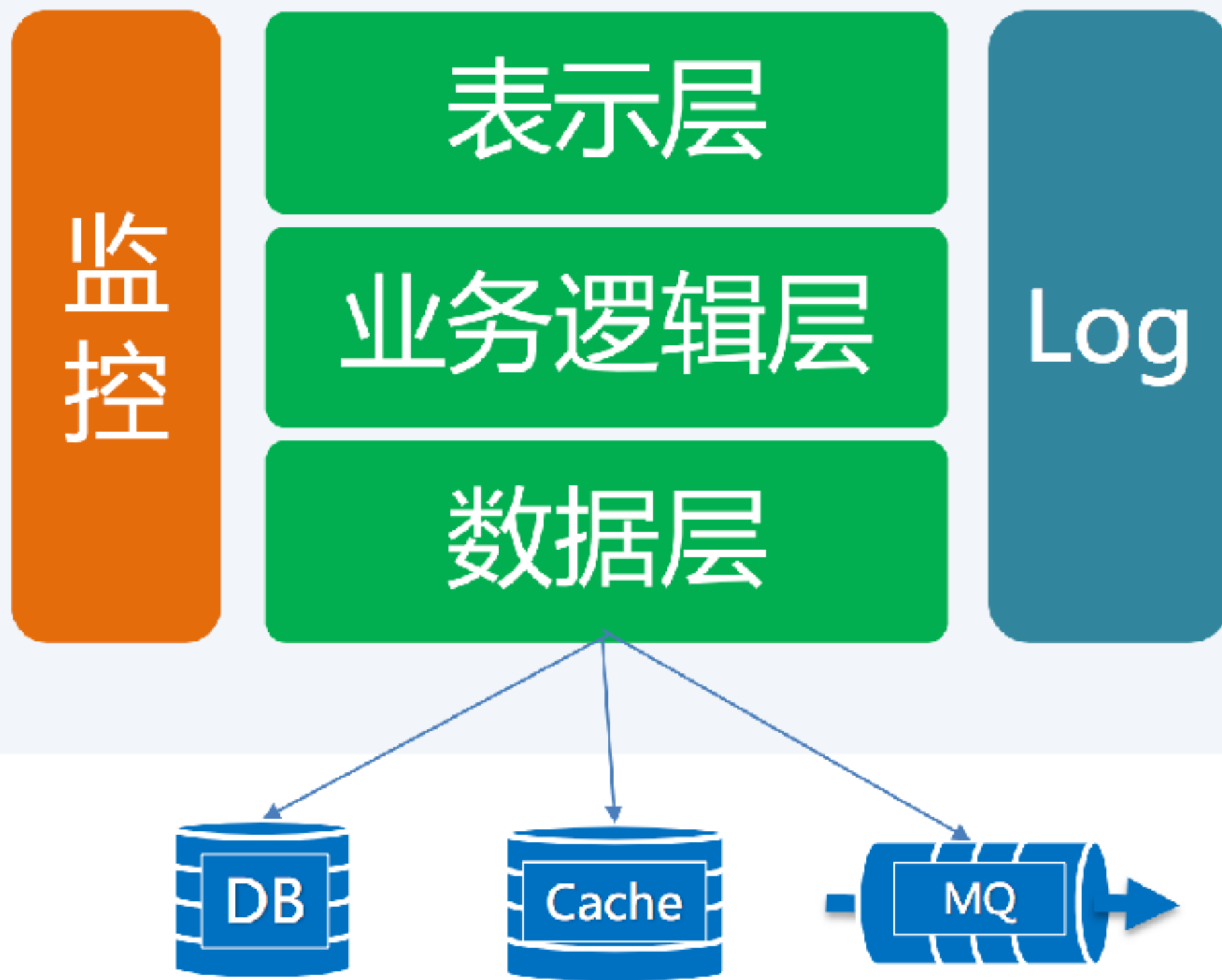
—荣夫弟

微服务演进

- 单体架构
- 服务化架构 (SOA)
- 多元化架构
- 微服务架构

单体架构

应用



单体架构

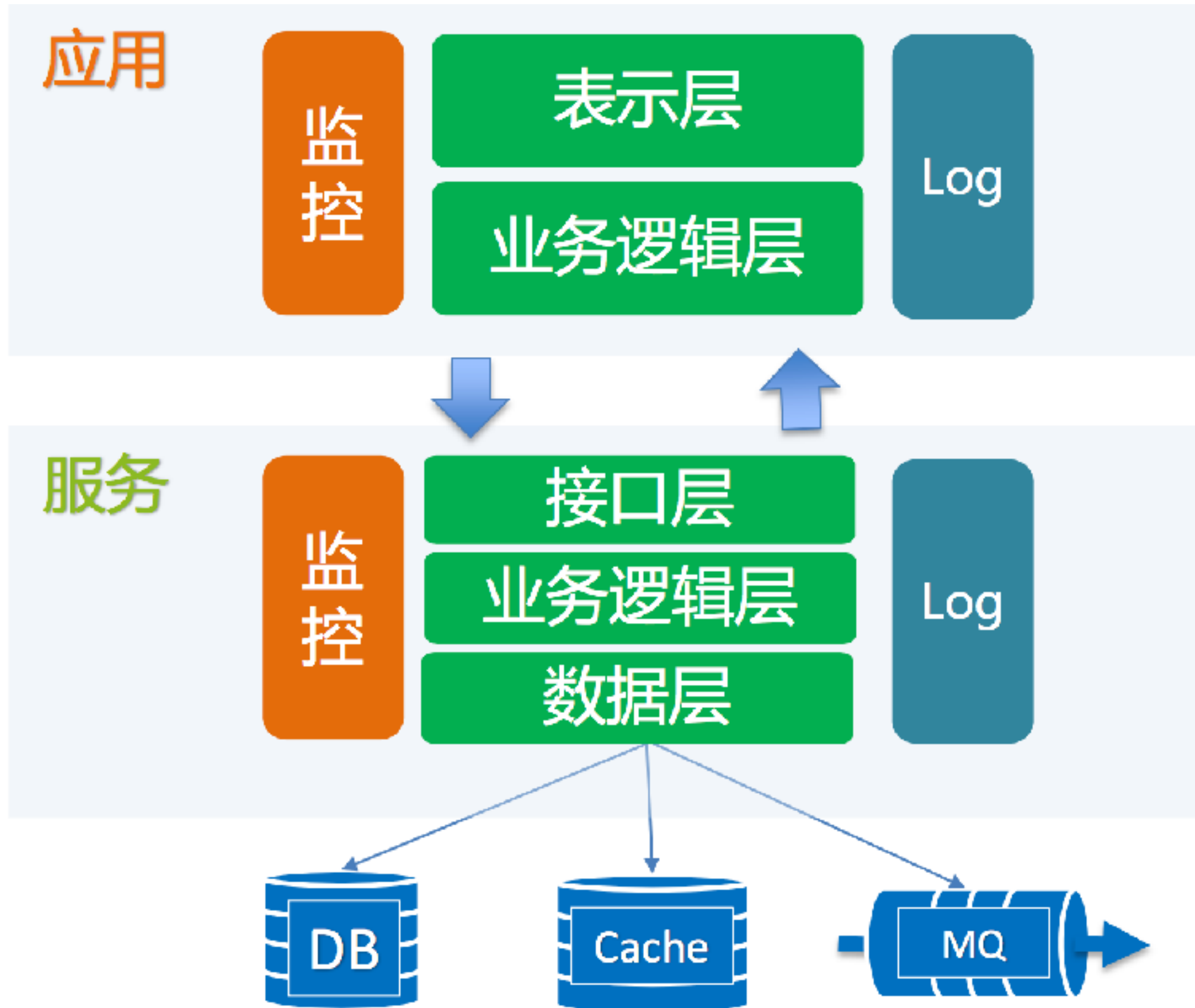
- 易于开发
- 易于测试
- 易于部署
- 易于水平伸缩

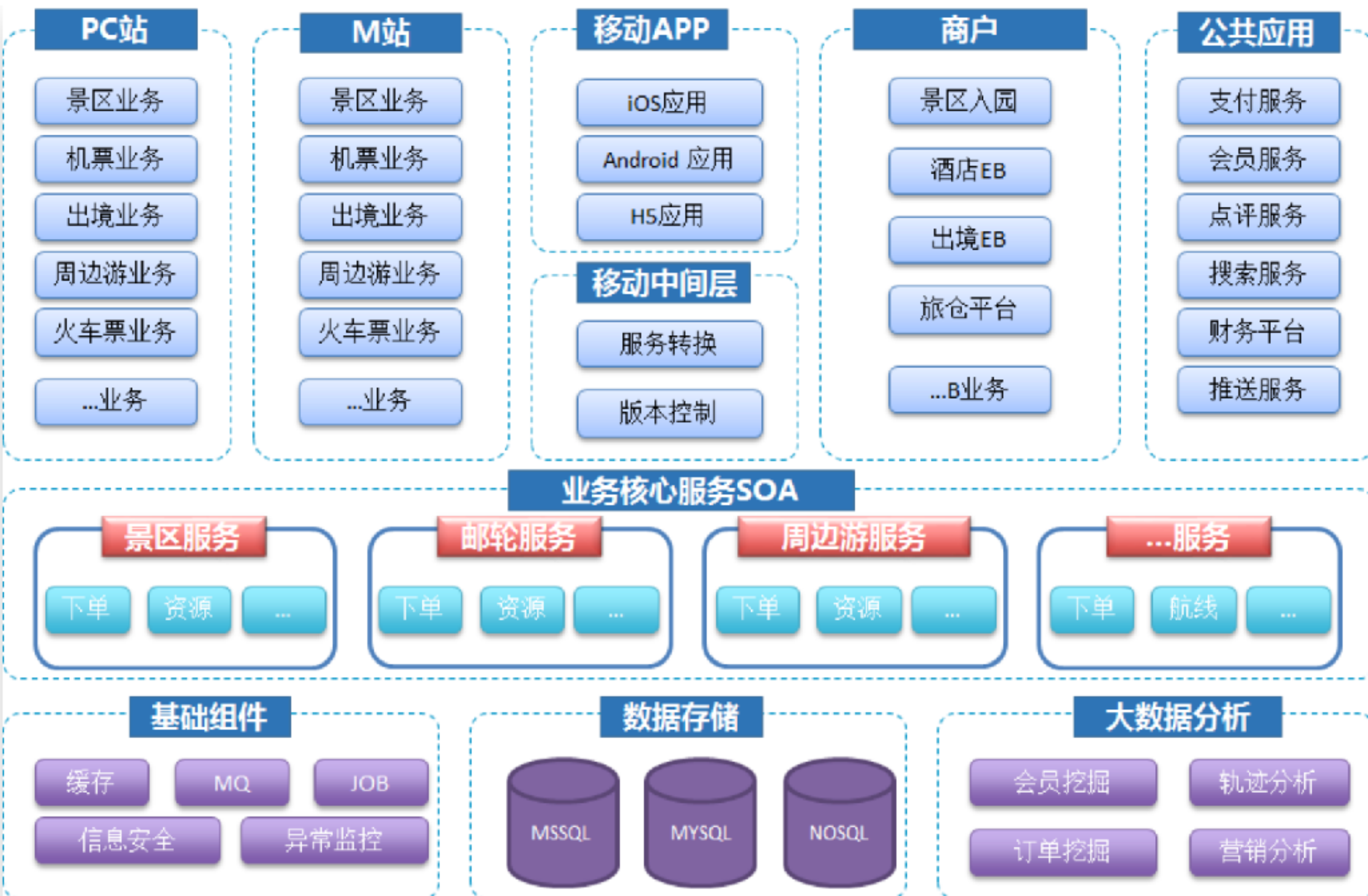
长大的烦恼

- 维护成本大
- 持续交付周期长
- 新人培养周期长
- 技术选型成本高
- 可扩展性差



SOA架构





什么是微服务

从业界的讨论来看，微服务本身并没有一个严格的定义，不过ThoughtWorks的首席科学家（Martin Fowler）的描述更加通俗易懂

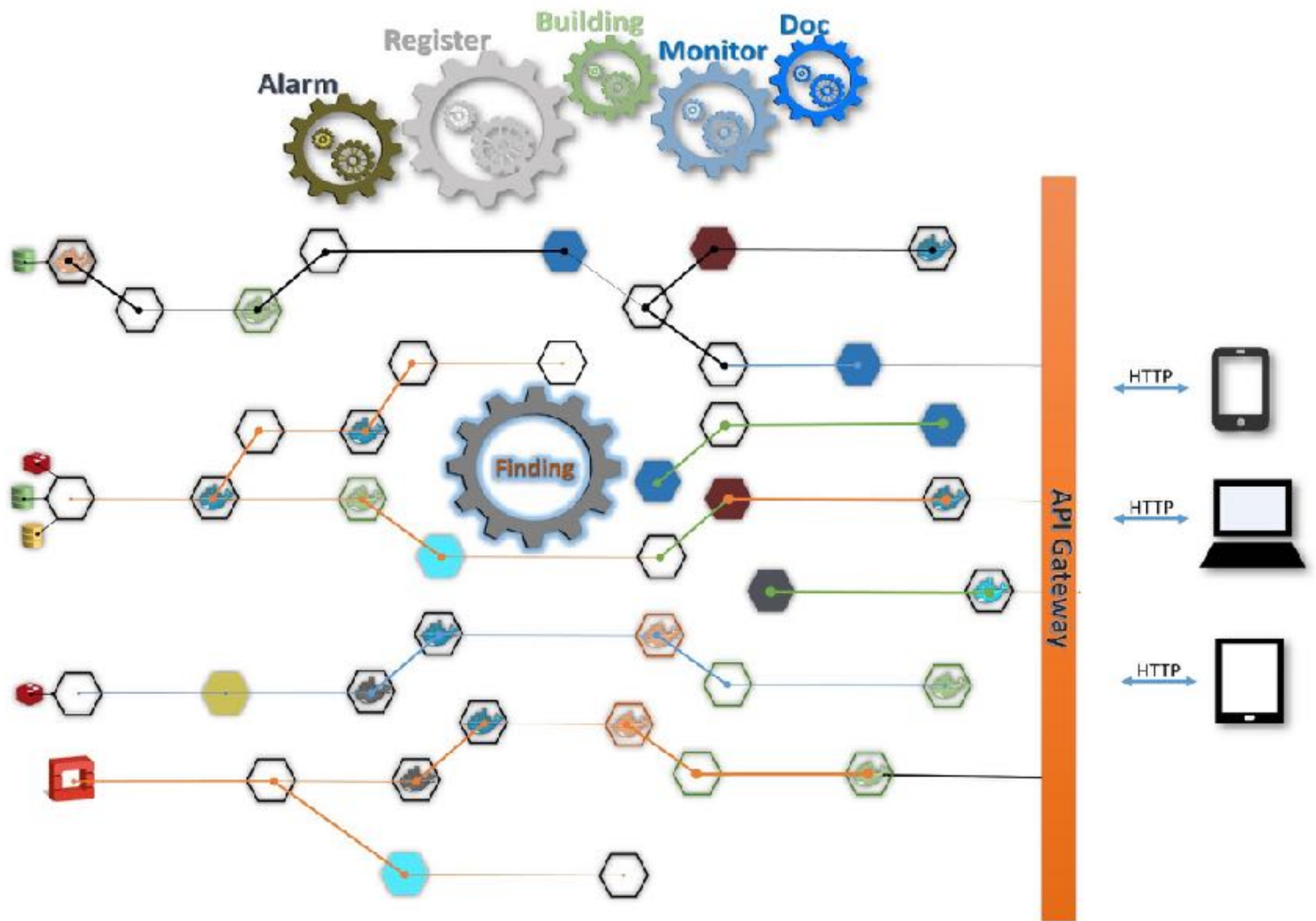
微服务架构是一种架构模式，它提倡将单一应用程序划分成一组小的服务，服务之间互相协调、互相配合，为用户提供最终价值。每个服务运行在其独立的进程中，服务与服务间采用轻量级的通信机制互相沟通（通常是基于HTTP的RESTful API）。每个服务都围绕着具体业务进行构建，并且能够被独立地部署到生产环境、类生产环境等。另外，应尽量避免统一的、集中式的服务管理机制，

对具体的一个服务而言，应根据业务上下文，选择合适的语言、工具对其进行构建。

——摘自 马丁·福勒先生的博客
(<http://martinfowler.com/articles/microservices.html>)



一个理想的微服世界

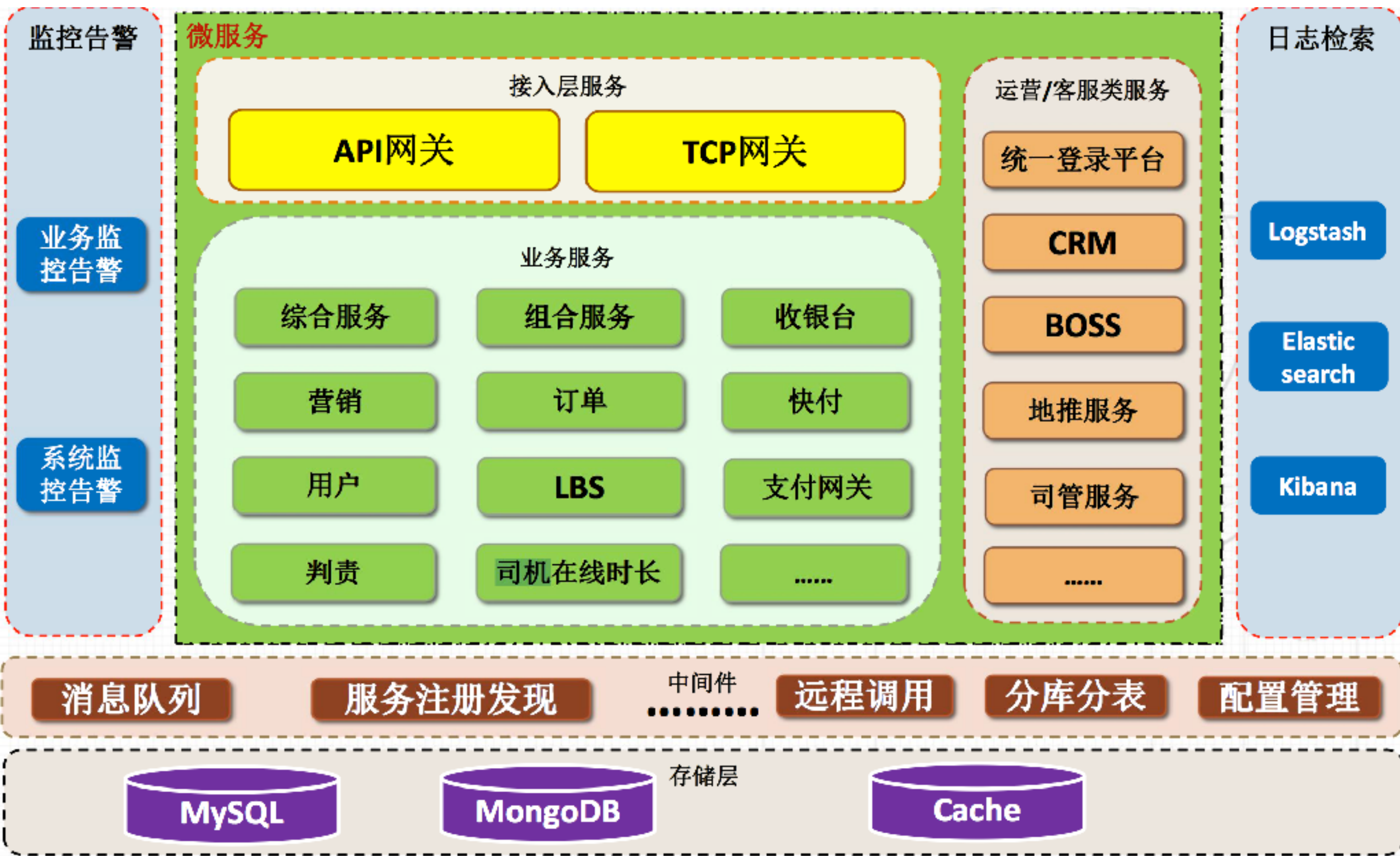


微服务实践

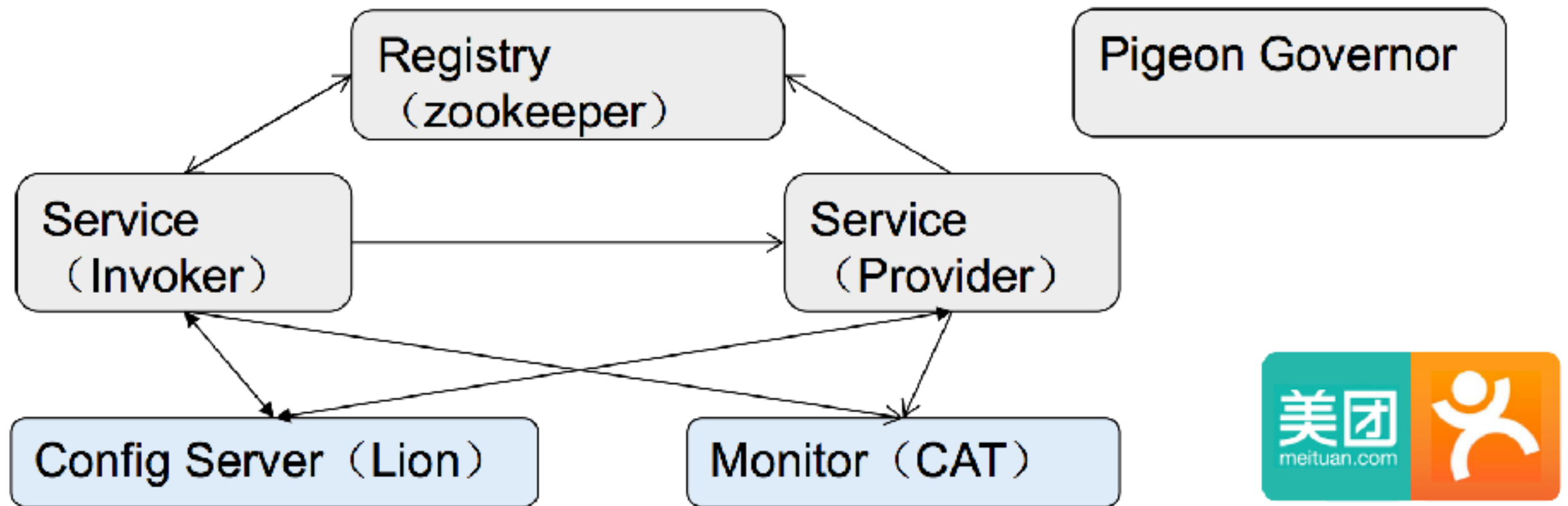
- 技术异构性
- 弹性
- 扩展性
- 简化部署
- 可组合型

为什么要微服务？

滴滴微服务



美团大众点评微服务



Pigeon提供jar包接入，线上运行在tomcat里

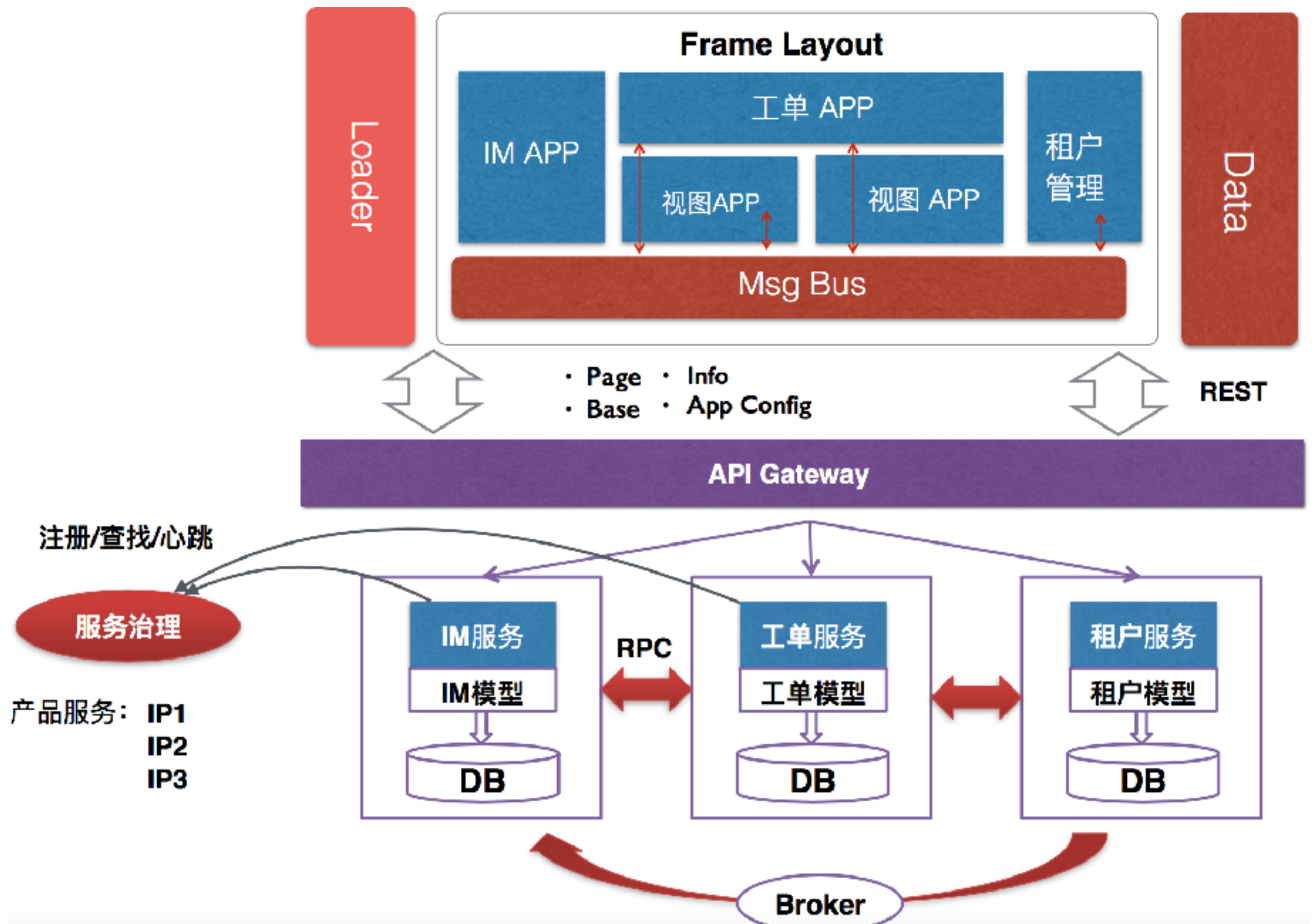
Monitor-CAT，负责调用链路分析、异常监控告警等

配置中心-Lion，负责一些开关配置读取

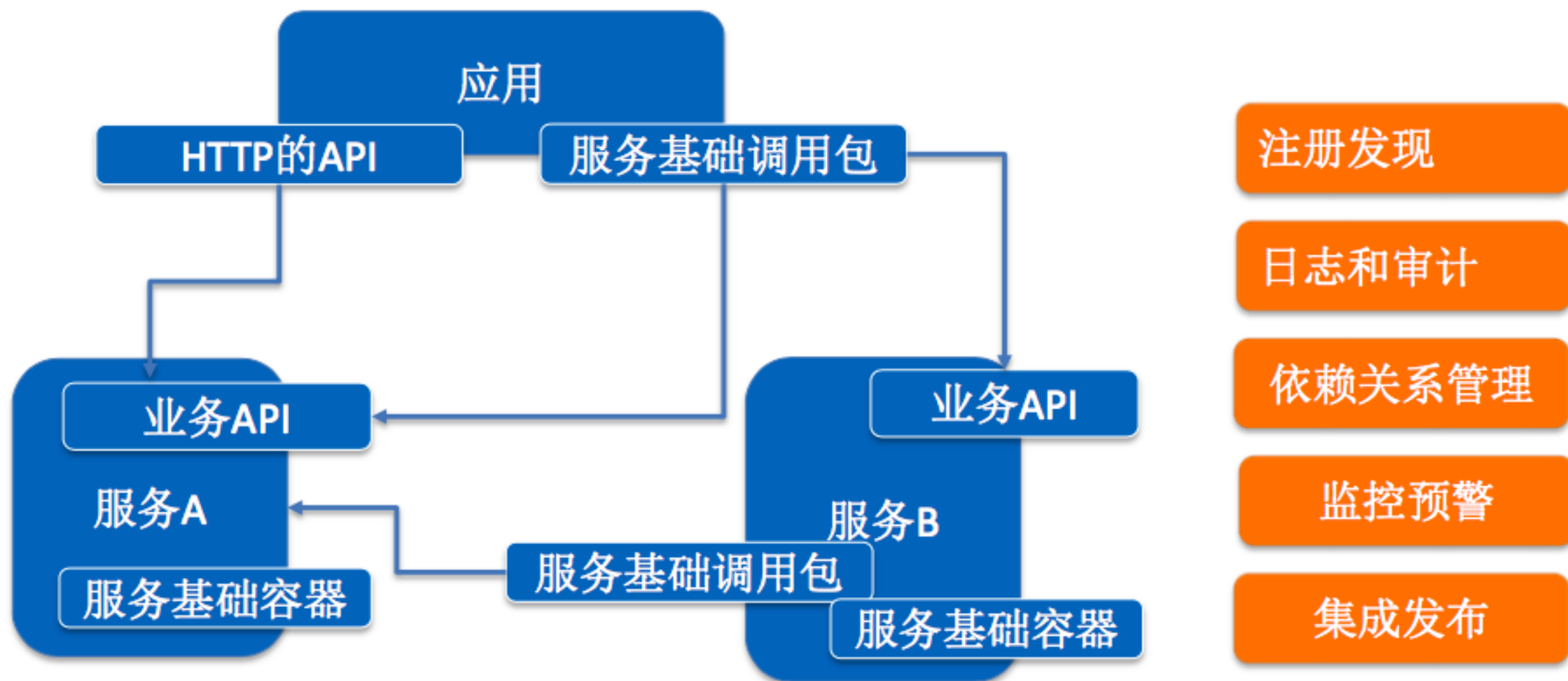
Governor-服务治理门户

一个interface定义为一个服务，每个服务有一个唯一标识

阿里CRM微服务



同程旅游微服务



服务注册发现，控制服务的API数量

统一代码框架，支持多种编程语言

服务依赖关系管理(服务的分级)

服务的CI/CD

没有银弹

- 多服务运维
- 服务间通信成本
- 数据一致性
- 系统集成测试
- 性能监控
- 管理复杂

