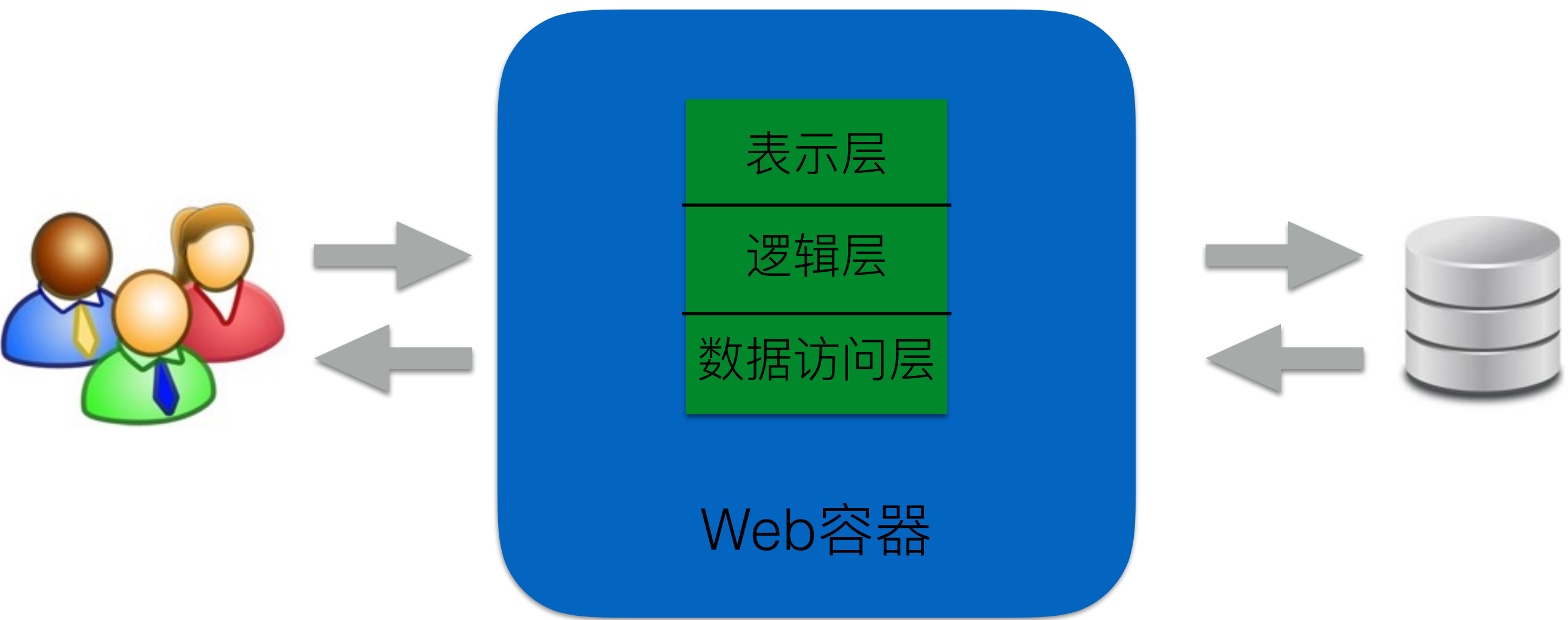


微服务架构

传统单块应用架构





用户

应用

数据库

单块应用架构的优势

- 易于开发
- 易于测试
- 易于部署
- 易于水平伸缩

最近十年互联网行业的
迅猛发展

单块应用架构面临的挑战

- 维护成本增加
- 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
- 构建全功能团队难

什么是微服务架构

“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，每个服务运行在其独立的进程中，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。”

——Martin Fowler

“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，每个服务运行在其独立的进程中，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。”

——Martin Fowler

多微才够微？

微服务划分原则

- 业务独立性
- 团队自主性

微服务架构的特点

单一职责

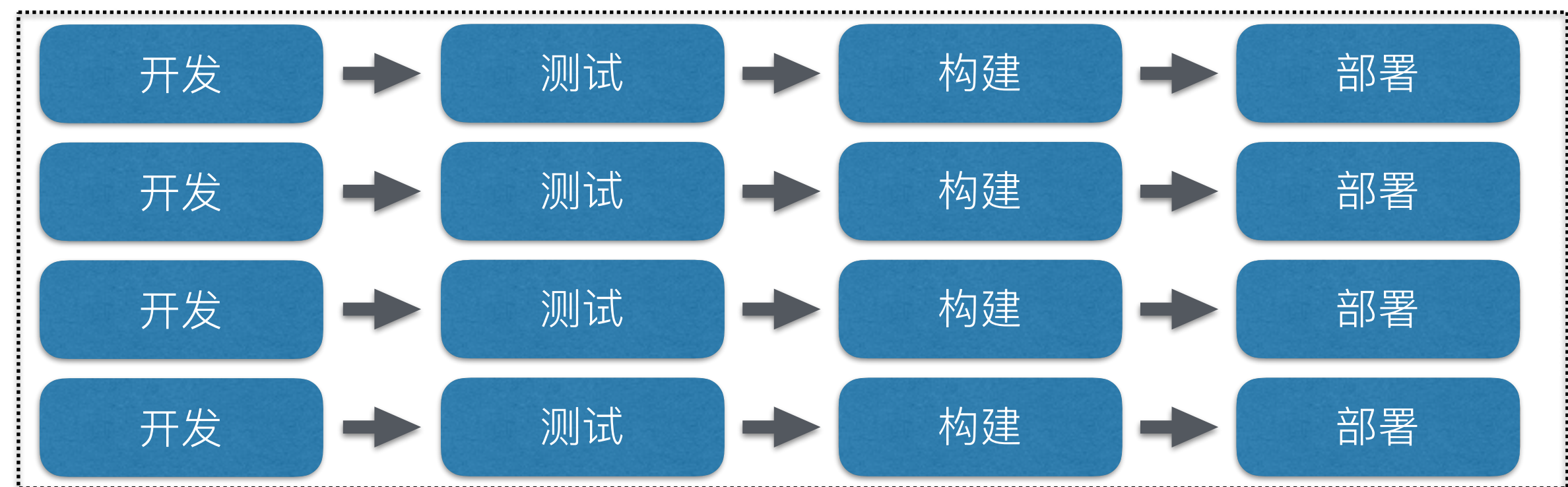
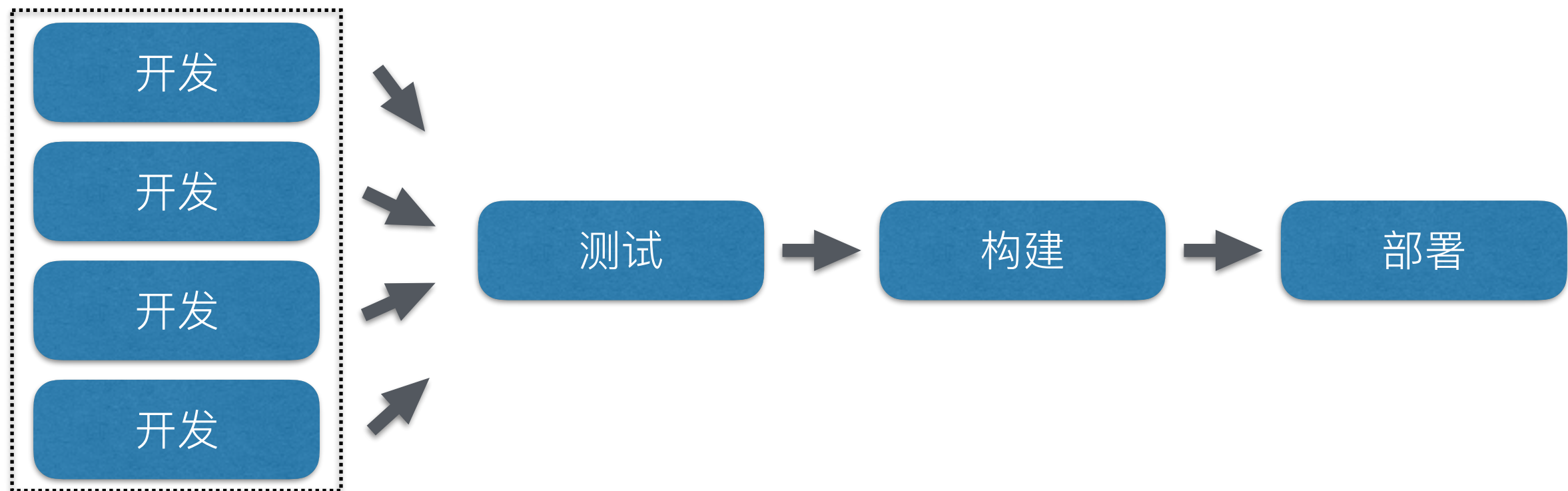
也就是说，微服务架构中的每个服务，都是具有业务逻辑单一的，符合高内聚、低耦合原则以及单一职责原则的单元，不同的服务通过“RPC”的方式灵活组合，从而构建出庞大的系统。

轻量级通信

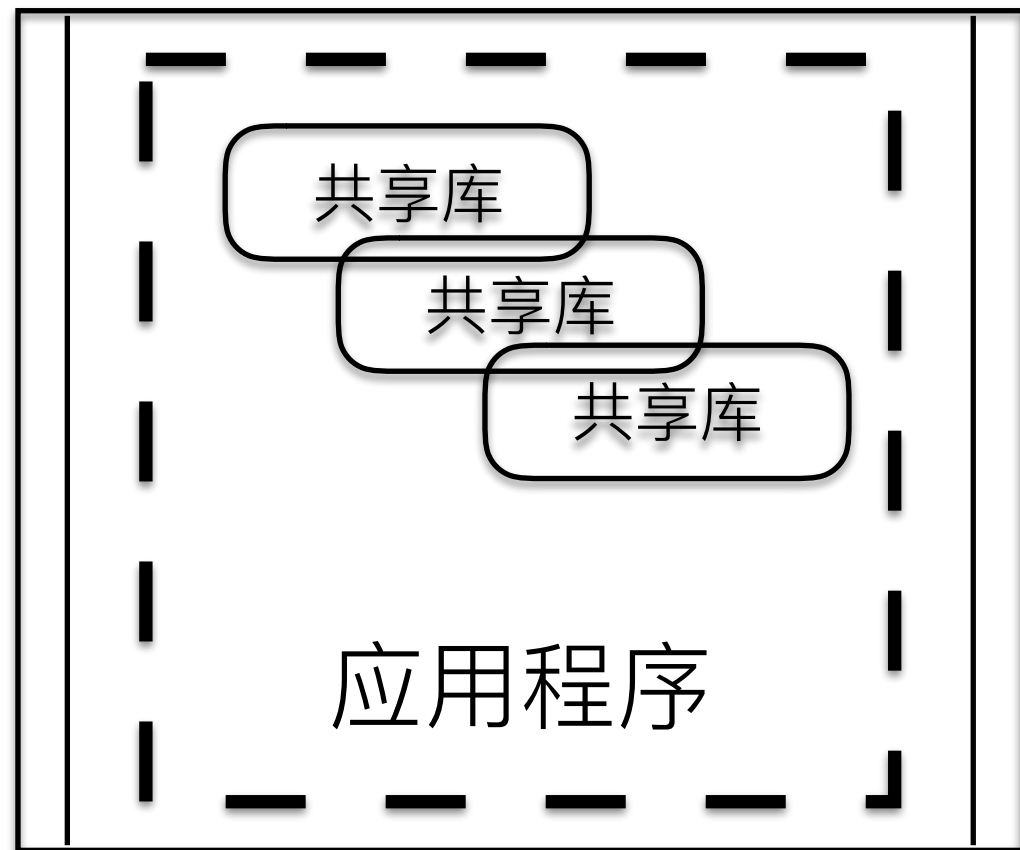
所谓轻量级通信，指语言无关、平台无关的交付方式。比如基于HTTP的Rest API

独立性

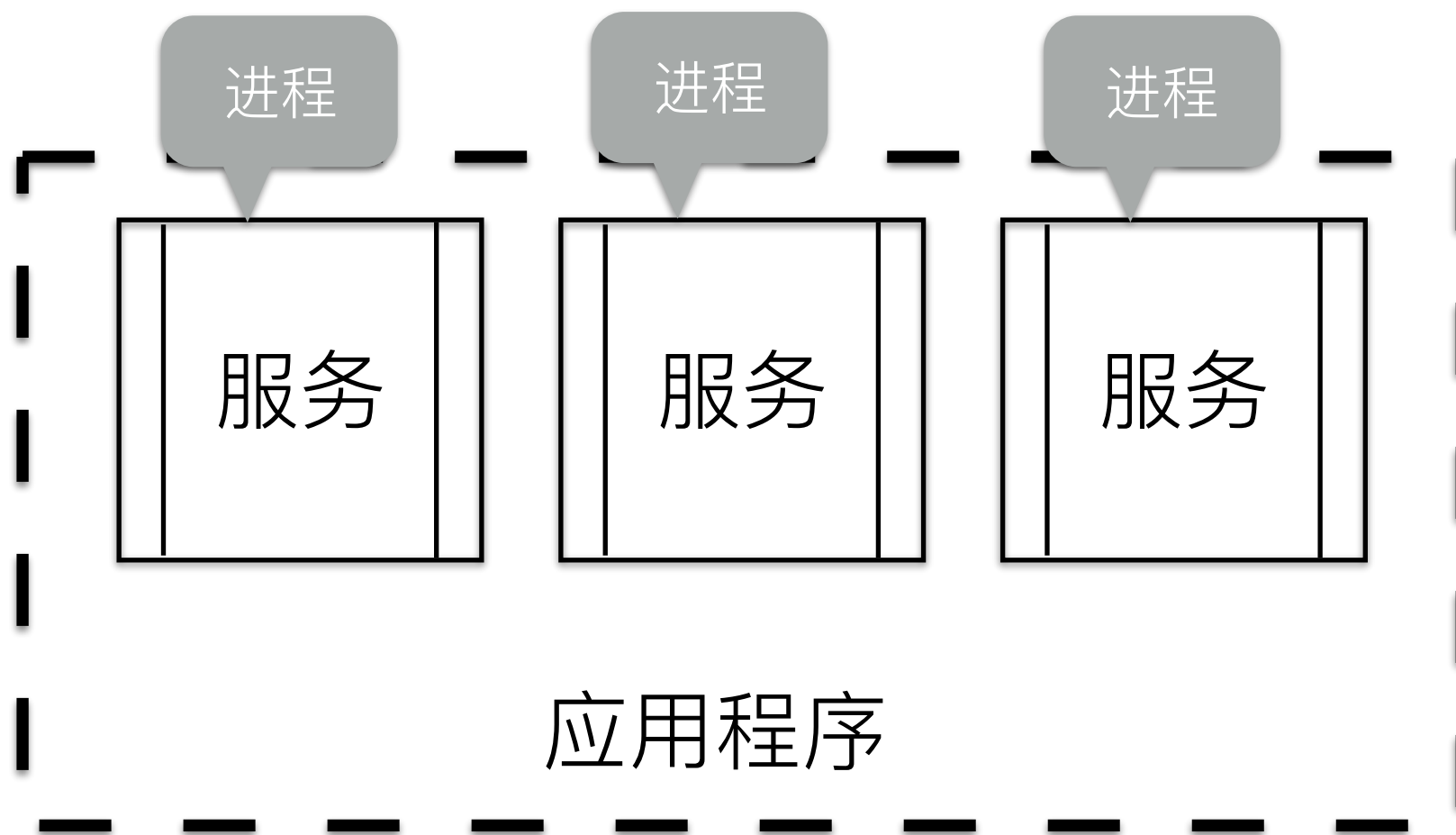
在应用的交付过程中，开发、测试以及部署的独立。



进程隔离



进程



微服务架构其实是将单一的应用程序划分成一组小的服务，每个服务都是具有业务属性的独立单元，同时能够被独立开发、独立运行、独立测试以及独立部署。

微服务架构与SOA

SOA实现	微服务架构实现
企业级，自顶向下开展实施	团队级，自底向上开展实施
服务由多个子系统组成，粒度大	一个系统被拆分为多个服务，粒度细
企业服务总线，集中式的服务架构	无集中式总线，松散的服务架构
集成方式复杂（ESB/WS/SOAP）	集成方式简单(HTTP/REST/JSON)
单块架构系统，相互依赖，部署复杂	服务能独立部署

SOA

云计算

敏捷

微服务

DevOps

持续
交付

微服务的本质

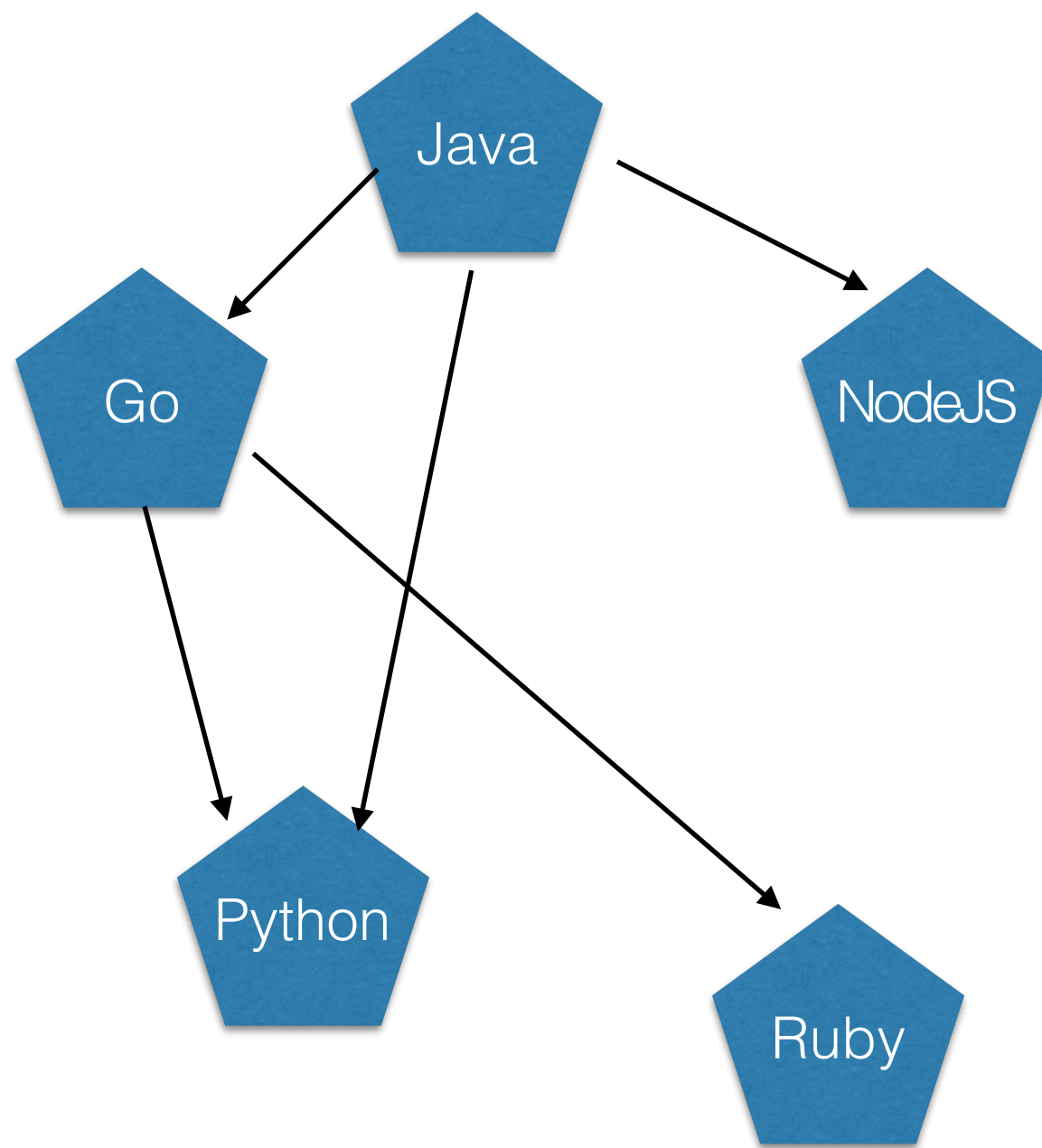
服务作为组件

围绕业务组织团队

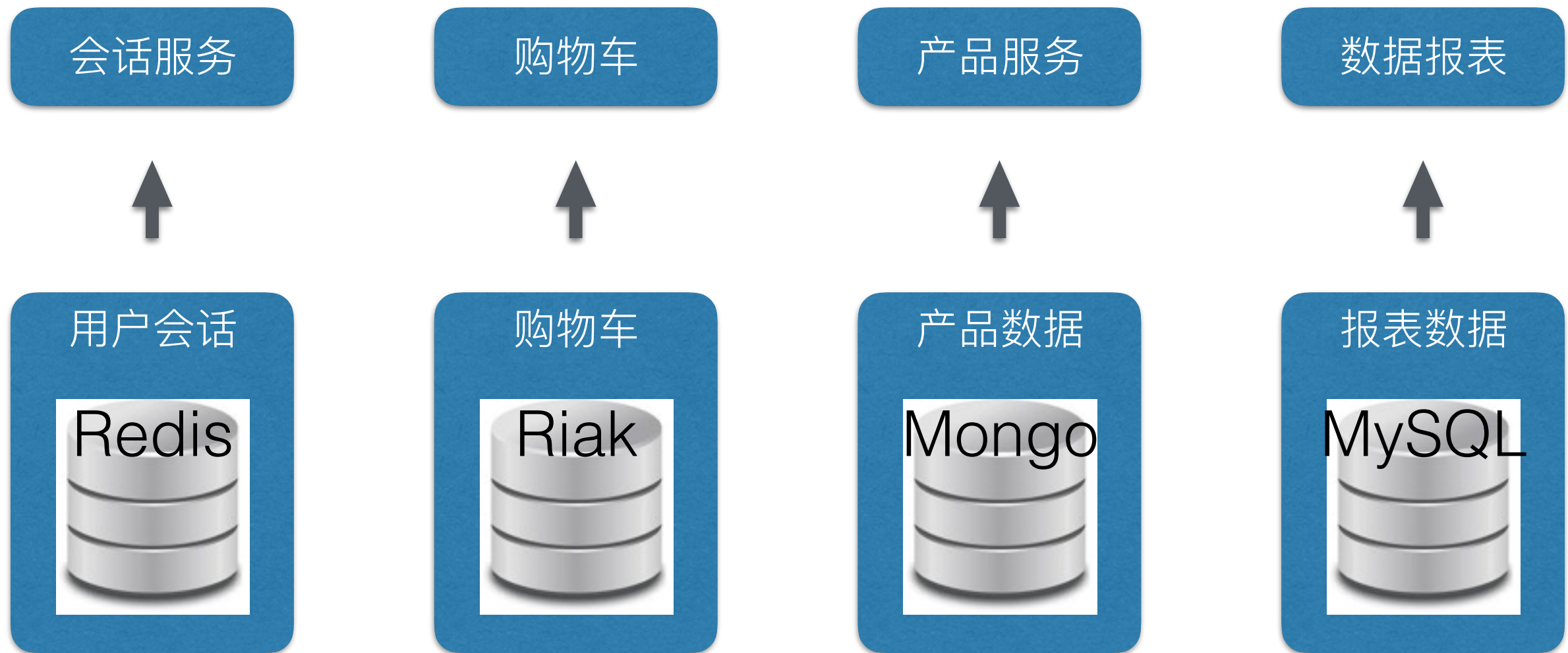
全功能团队

关注产品而非项目

技术多样性



业务数据独立



基础设施自动化

演进式架构

- 服务作为组件
- 围绕业务组织团队
- 关注产品而非项目
- 技术多样性
- 业务数据独立
- 基础设施自动化
- 演进式架构

微服务不是银弹

分布式系统的复杂度

- 性能
- 可靠性
- 异步
- 数据一致性

运维成本

- 配置
- 部署
- 监控与告警
- 日志收集

部署自动化

服务间的依赖测试

服务间的依赖管理

- 分布式系统的复杂度
- 运维成本
- 部署自动化
- 服务间的依赖测试
- 服务间的依赖管理

如何解决微服务架构
带来的挑战？