任务也是微服务

漫谈微服务

AGENDA

- 什么是微服务
- ON DEMAND任务调度系统（JUICE）
- 架构拆解过程
- 测试
- 集成与部署
- 监控

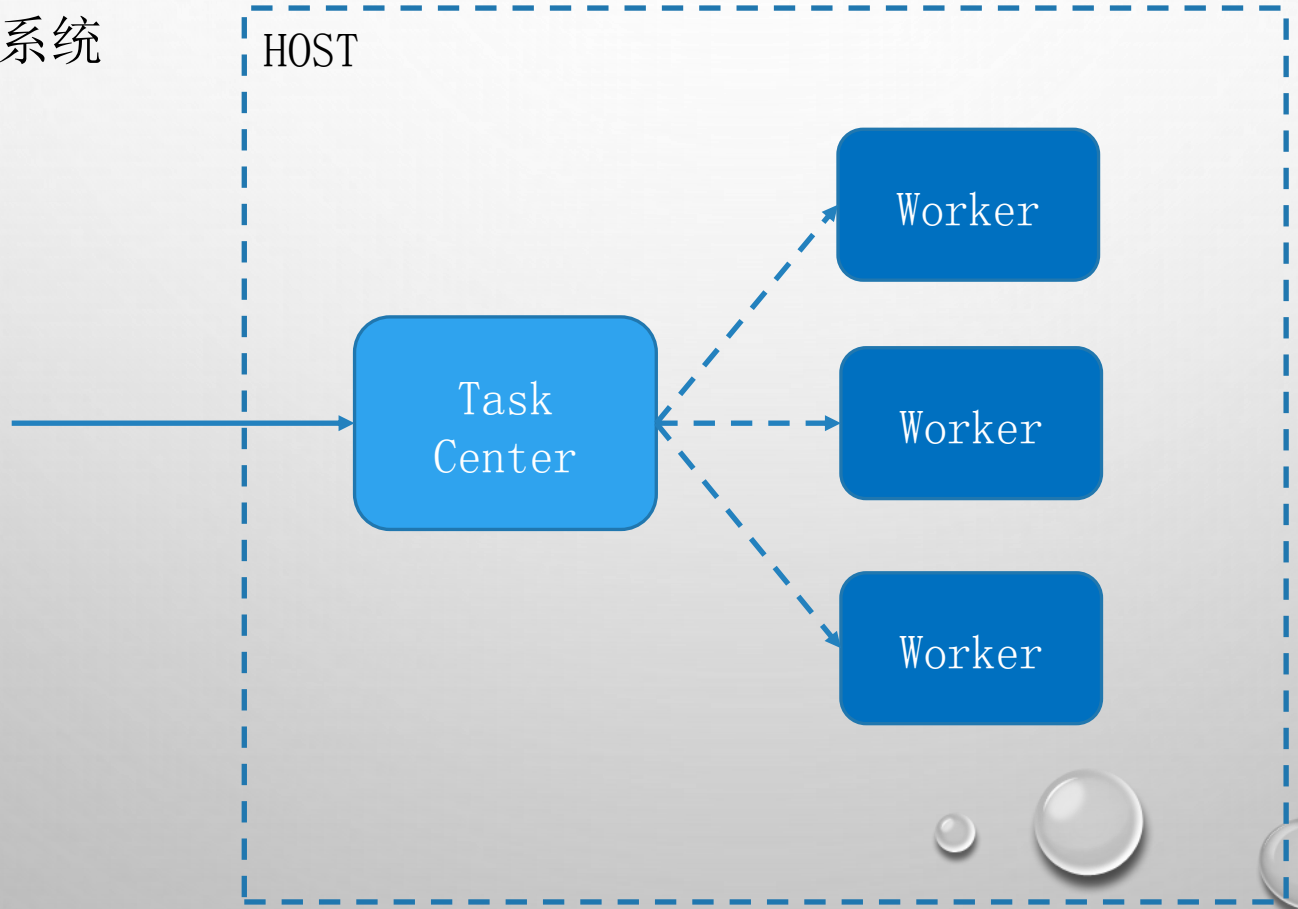
什么是微服务

- 一些协同工作，小而自治的服务
- SOA与微服务
  - 通信协议
  - 三方组件选择
  - 颗粒度
- 足够的小？
  - 服务越小，微服务架构的优点和缺点也就越明显
  - 能在两个星期重构

Juice

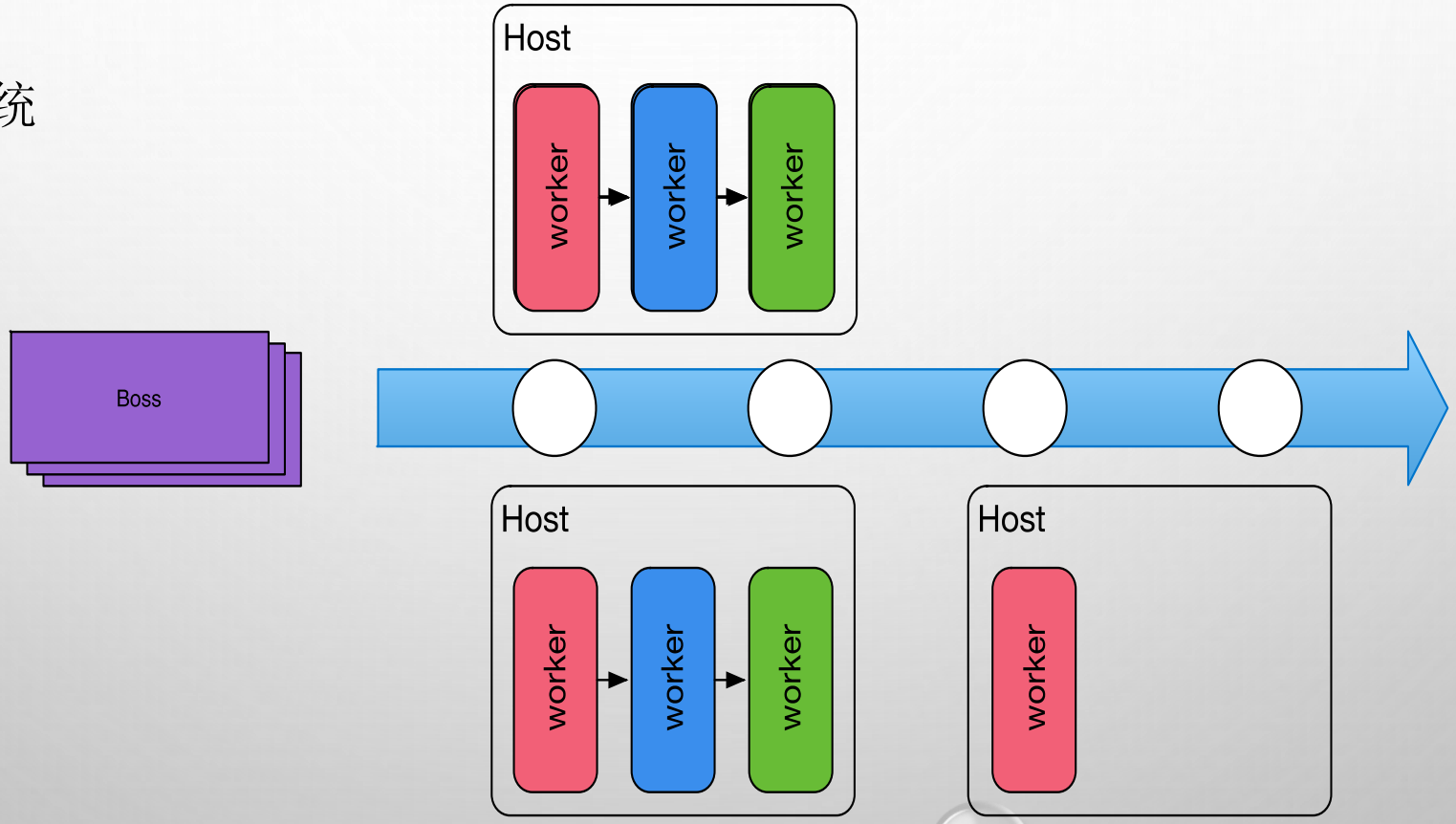
架构拆解过程

- 第一代调度系统



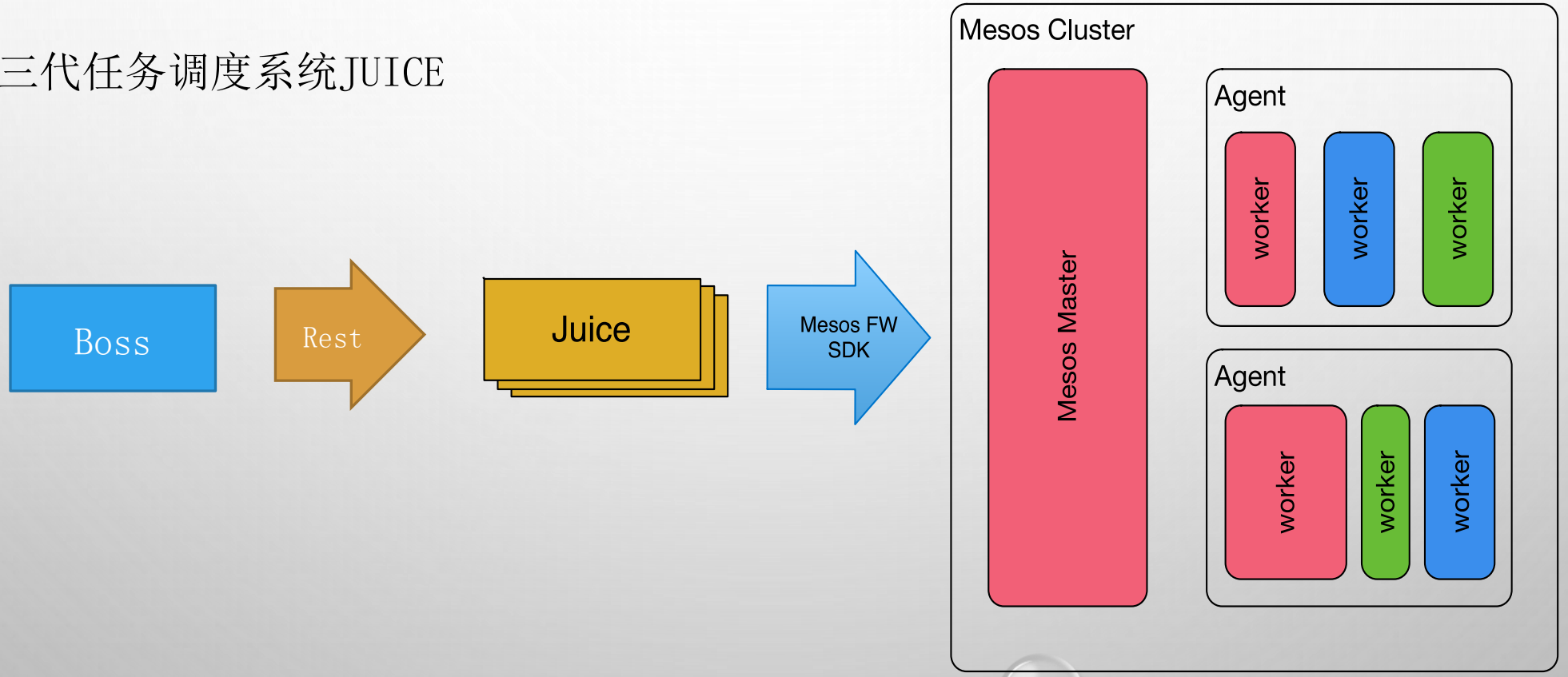
架构拆解过程

- 第二代调度系统



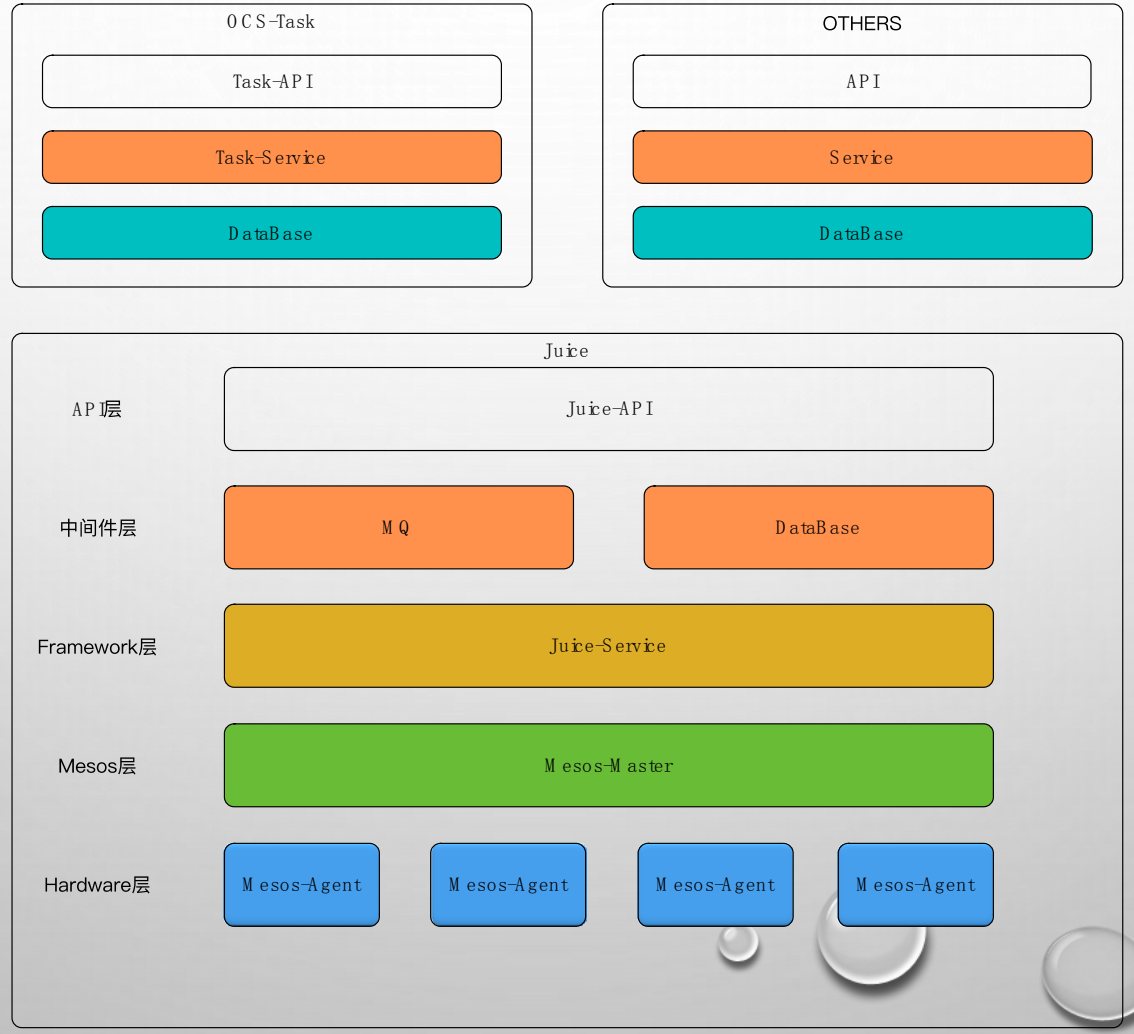
架构拆解过程

- 第三代任务调度系统JUICE



架构拆解过程

- 任务系统的结构



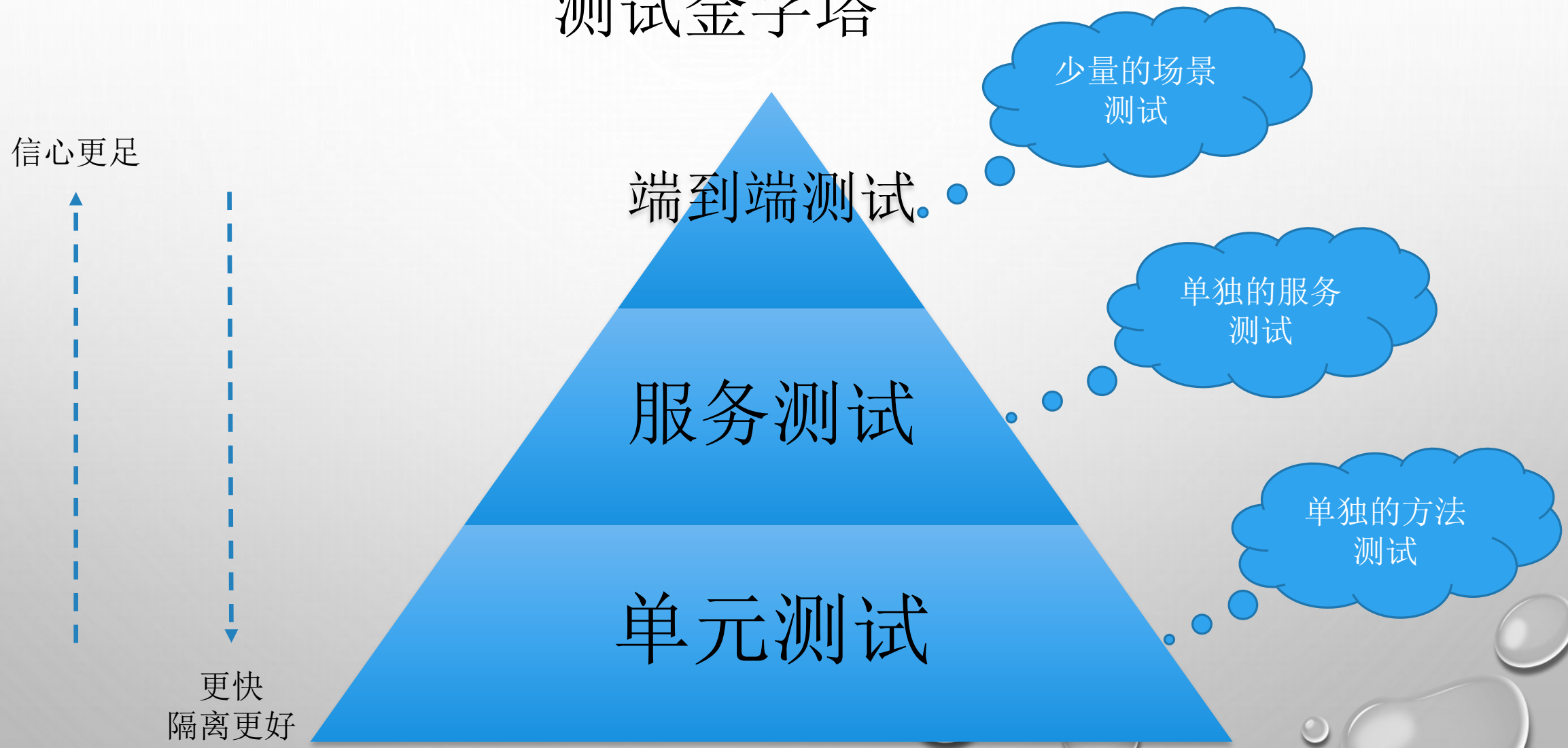
架构拆解过程

- 结构分离
  - 任务与调度分开
  - 任务生产者和消费者分开
  - 任务记录与业务逻辑分开
- 数据库分离
  - 打破外键关系
  - 分离共享数据
- 事务分离
  - 分布式事务
  - 最终一致性

测试

- TEST CASE增多
- 单个流程走不通
- 性能测试变得重要

测试金字塔



MOCK与打桩

- MOCK数据
  - 外部依赖的MOCK数据
  - 内部之间的MOCK数据
- 打桩服务
  - 做一个永远返回成功的CALLBACK服务
  - 做一个通过参数调节的智能打桩服务

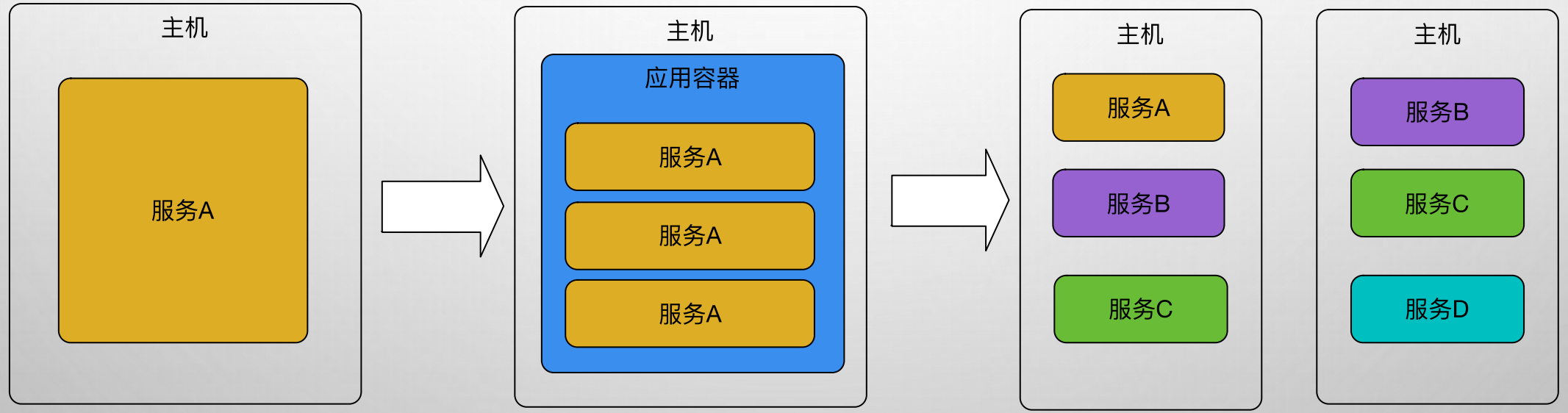
端到端测试

- 端到端测试的弊端
  - 服务越多，端到端就越不容易实现。
  - 谁来写这些测试用例？
  - 测试时间多长？
- 端到端测试的重要性
  - 保证业务的完整性
  - 保证性能

集成与部署

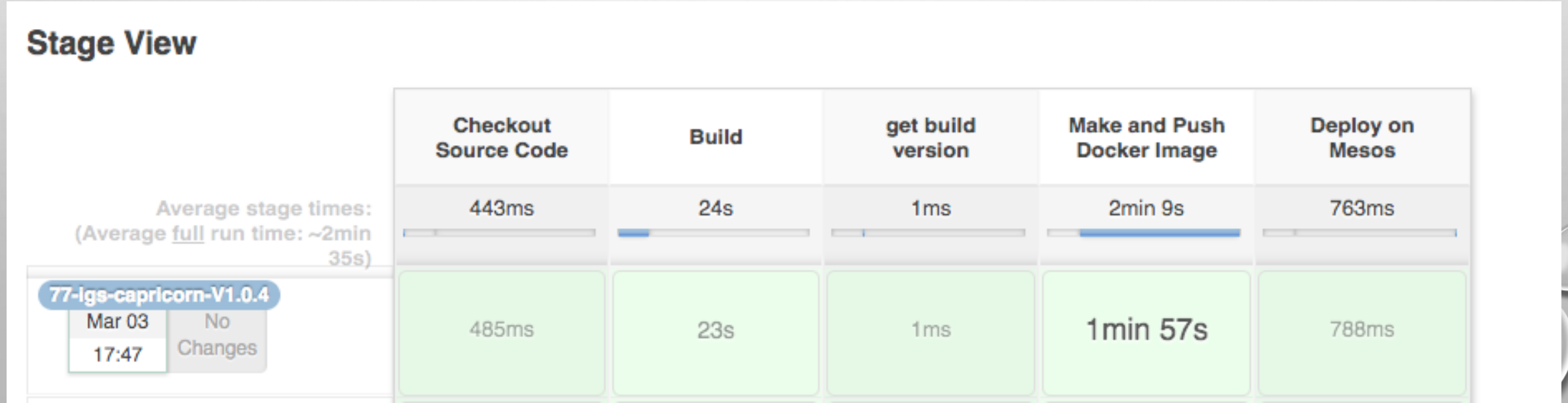
- 一台一服务到多台多服务的演变
- 持续集成
- 区分部署与上线

一台一服务到多台多服务的演变



持续集成

- 镜像作为构建物
  - 服务包含在镜像中，部署更快
  - 环境包含在镜像中，避免配置漂移
- 构建PIPELINE



区分部署和上线

- 蓝绿部署

监控

- 多服务多服务器的挑战
- 日志跟踪与指标跟踪
- 监控还能干什么？

多服务多服务器的挑战

- 海量的日志

```
<layout class="ch.qos.logback.classic.PatternLayout">
    <Pattern>[%date{ISO8601}][${APP_NAME}][${INSTANCE}][${TASK_ID}][%5level][%logger:%L] %msg%n</Pattern>
</layout>
```

- 程序的调用链找不到
  - 引入服务治理是个好方法
- 如何评估某一类服务的服务器占用？
  - 以服务来聚合监控

日志跟踪与指标跟踪

- 工具
  - ELK
  - MESOS METRICS
- 指标
  - 日志
  - CPU+MEMORY+DISK
  - 服务健康监测

监控还能干什么？

- BVT监控引入
- 服务的自动伸缩
- 业务评价

THANKS & QA