

# 高可用架构之中间件技术

吴湘@美团点评-基础架构

# 大纲

---

- 互联网业务的技术挑战
- 解决之道-构建微服务高可用架构
- 如何实践-中间件基础设施的建设

# 互联网业务的技术挑战

---

稳定性

性能

扩展性

效率

成本

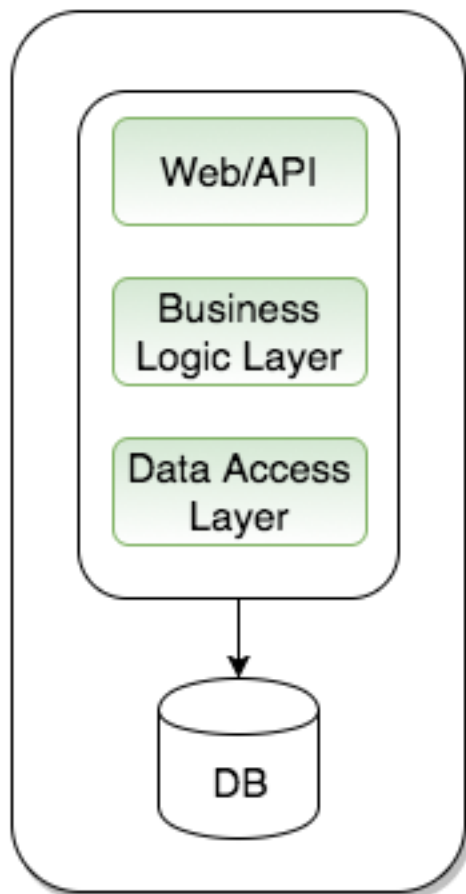
# 什么是微服务

---

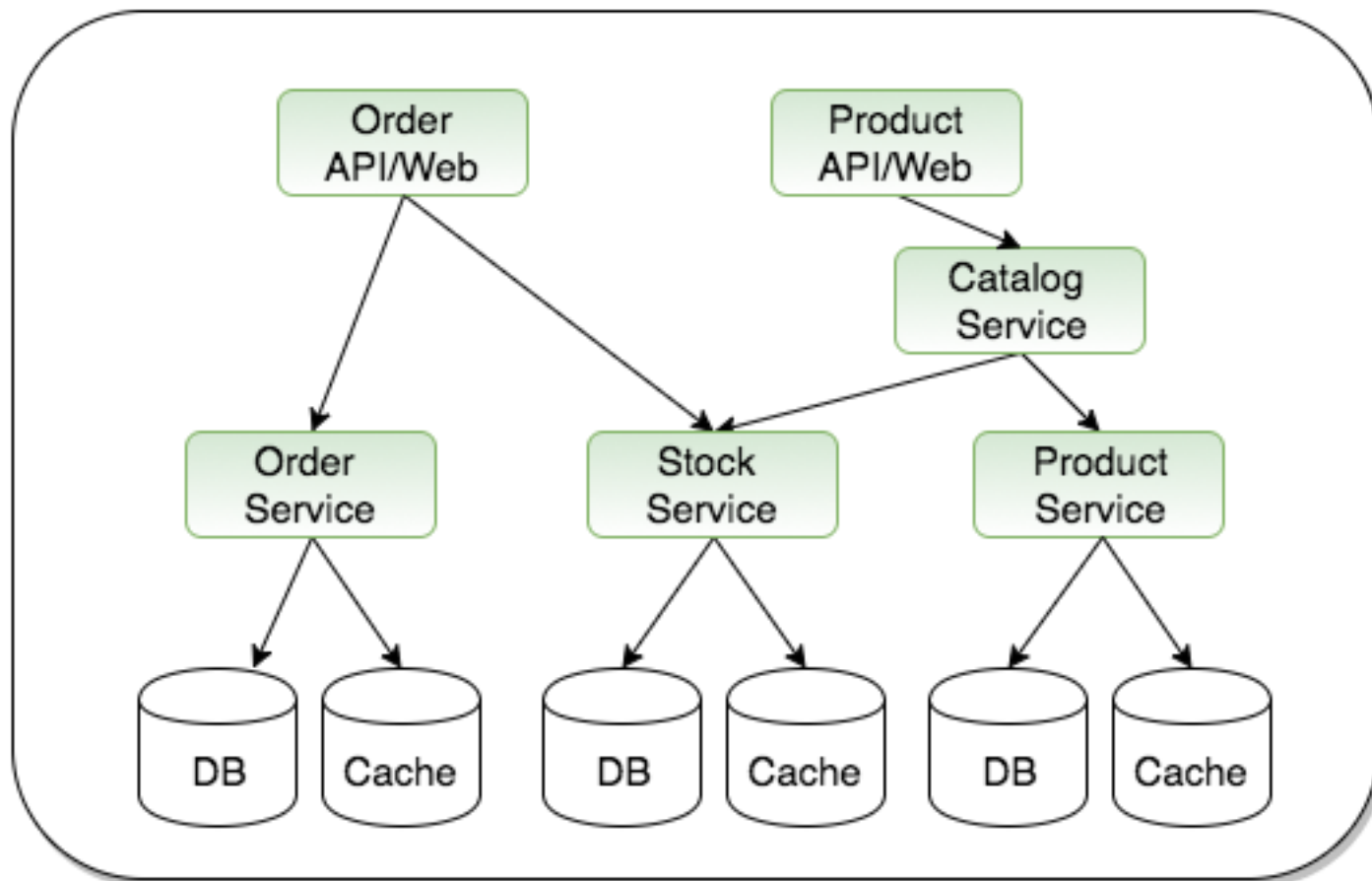
- 微服务是一种软件体系结构类型，复杂的应用程序由许多微小并且相互独立的服务组成：
  - 这些服务相互之间通过API接口通信;
  - 这些服务是微小的，高度松耦合，并且只关注在一个小的任务;
  - 服务是自治并且完整的。控制所有组件，包括UI、中间件、存取数据和事务。

# 微服务架构

单体应用

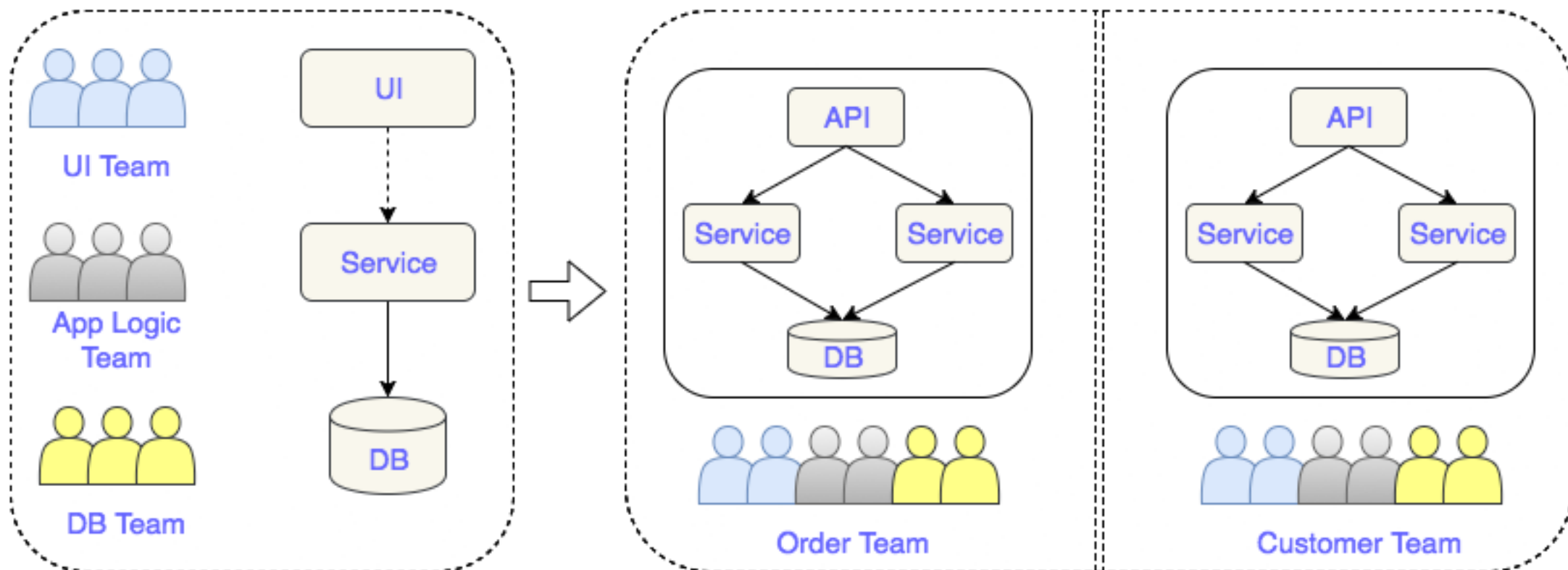


微服务架构

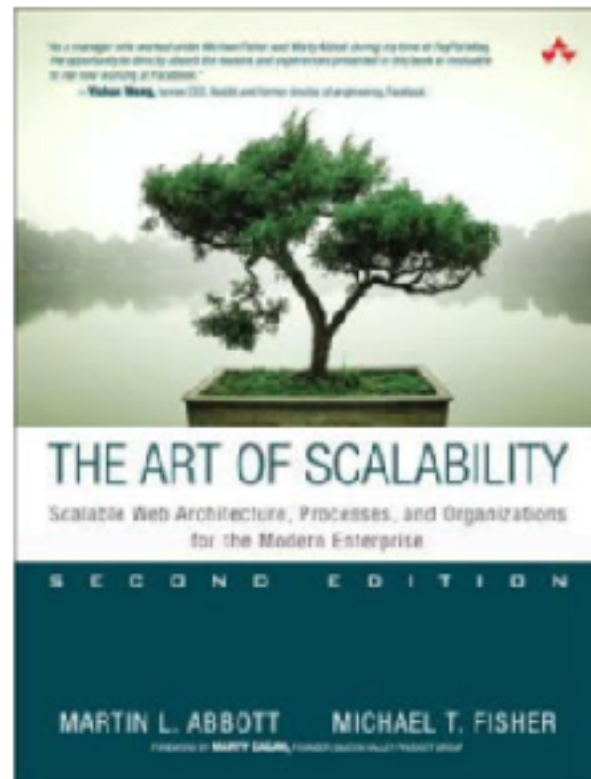
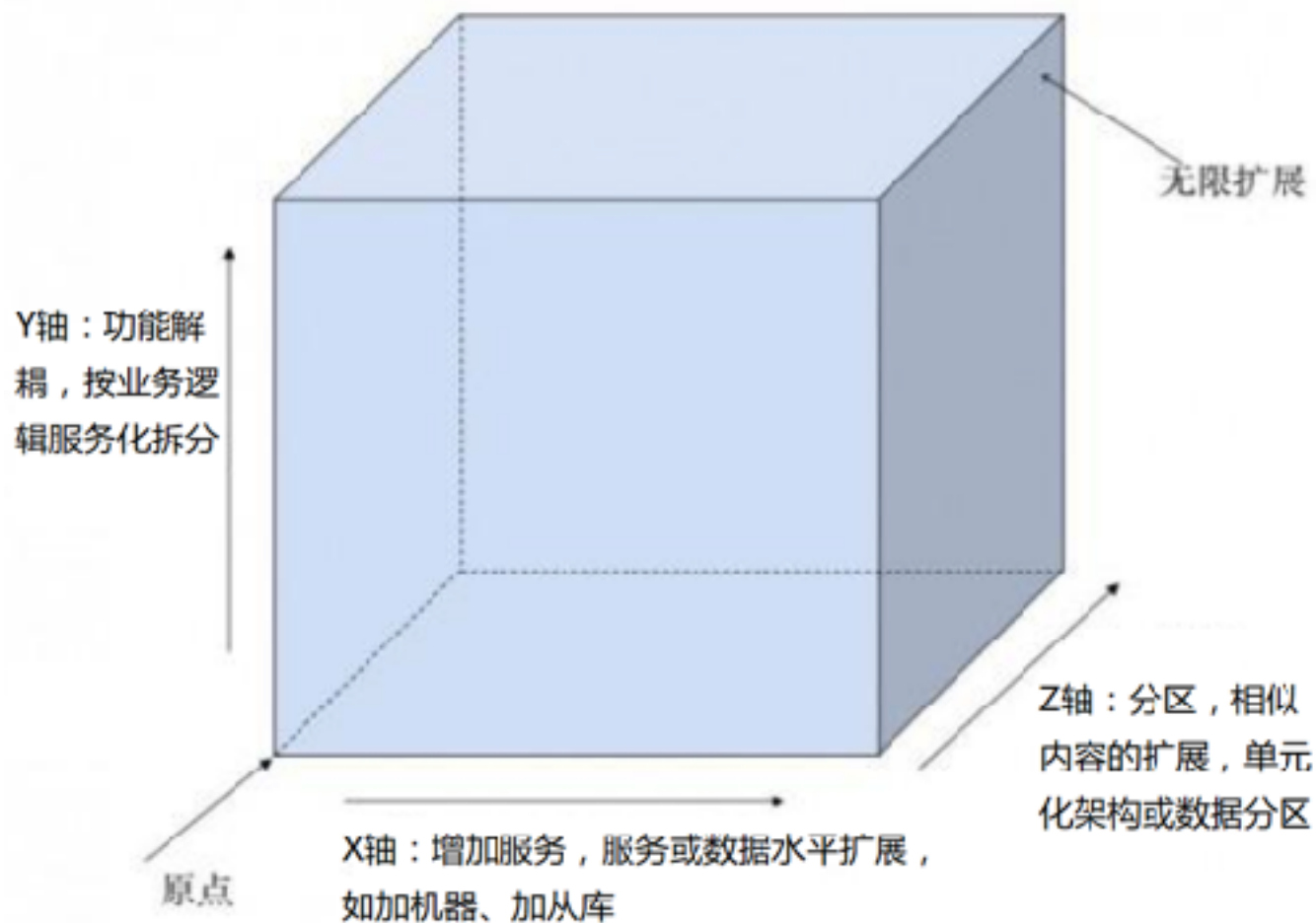


# 组织结构的变化

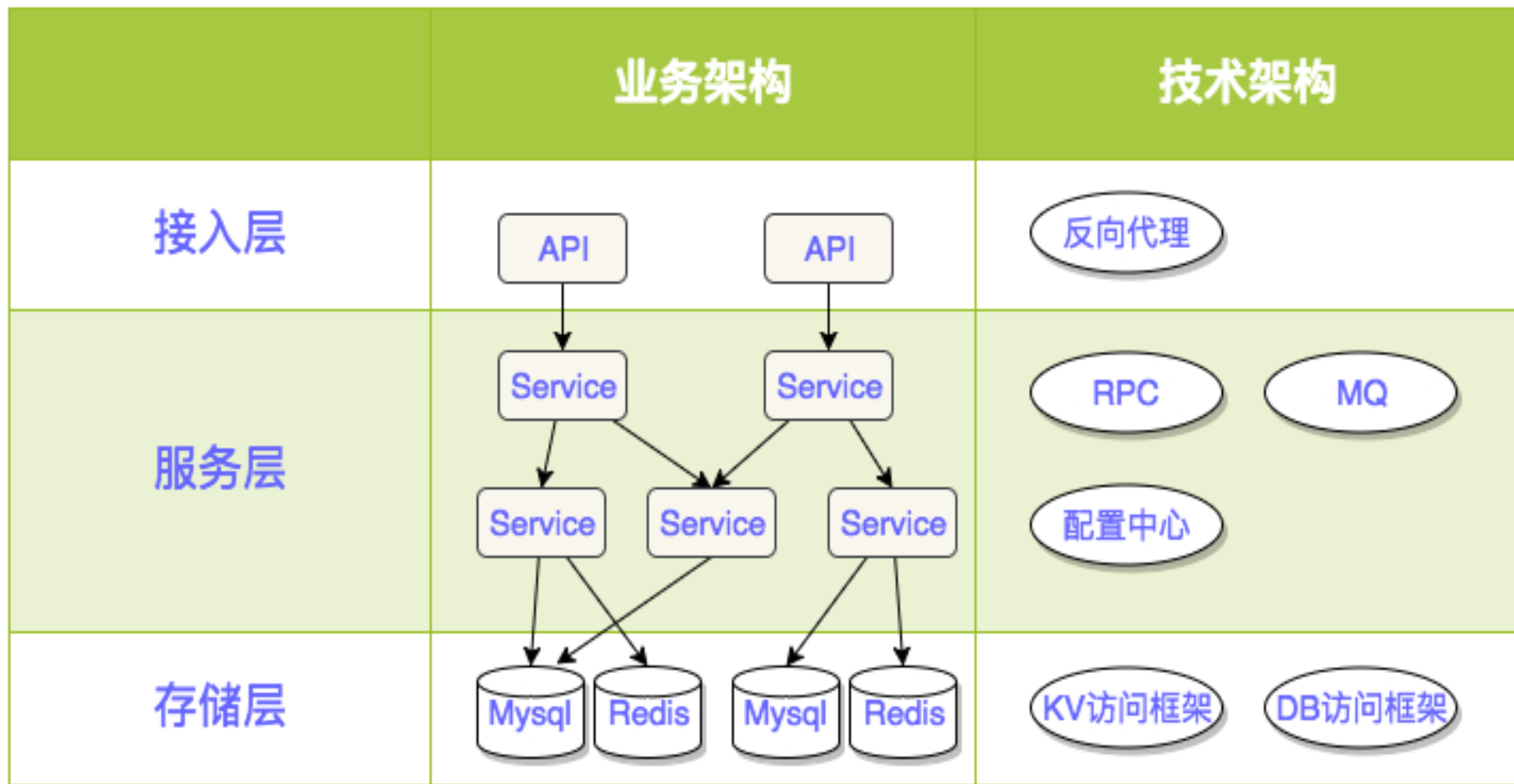
- 按技术能力组织->按业务职责组织，微服务的一个组织的职能包括产品、开发、测试、部署、支持等



# 扩展立方



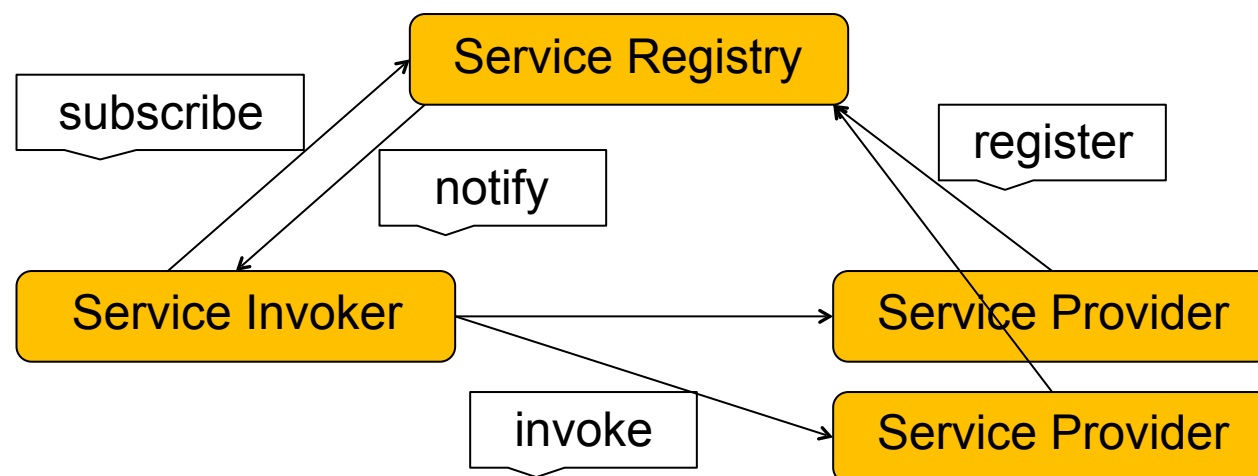
# 微服务如何实践





# RPC

- 服务数量多，调用关系变得复杂，需要有统一的服务管理机制来支撑X轴扩展
- RPC框架让调用方只关注接口，面向接口编程，开发效率高
- 推荐采用具备服务注册、服务发现机制的RPC框架，相比常规通过LB节点作负载均衡的方式可用性、性能更好



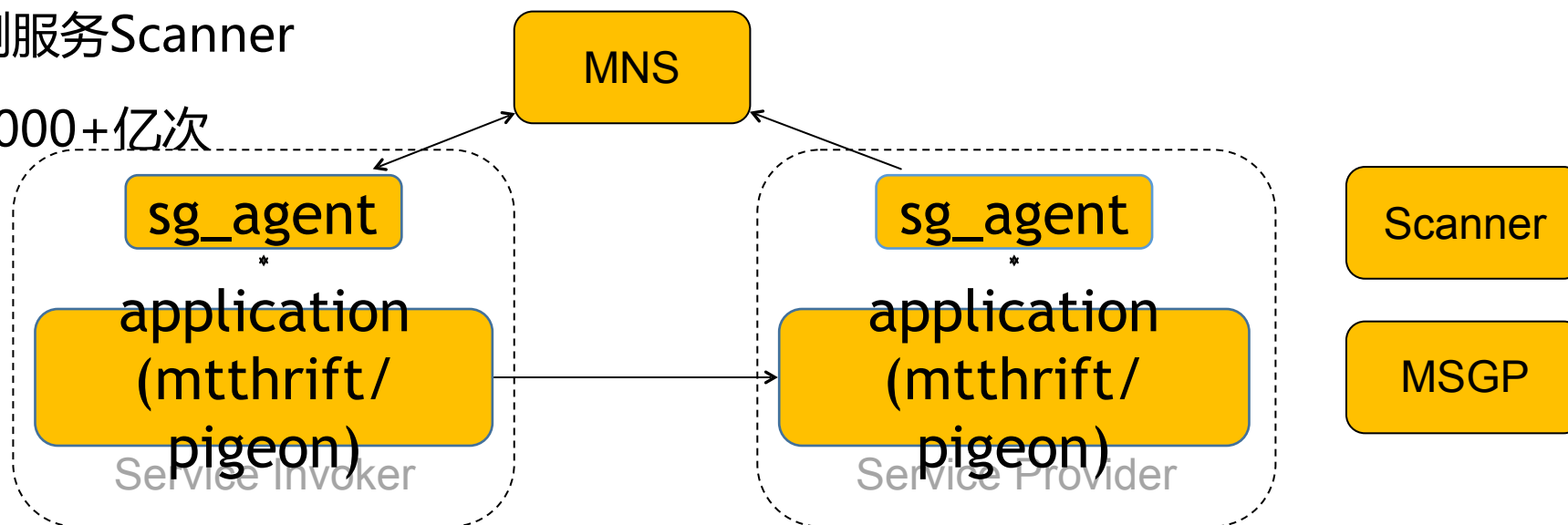
# RPC

---

- 开源的RPC框架Dubbo，腾讯Tars
- Spring Cloud体系（Eureka+Zuul+Ribbon+Feign+Hystrix）
- 云端PaaS服务，RPC相关组件如阿里云EDAS，腾讯云TSF，亚马逊AWS
- 美团点评OCTO、Pigeon

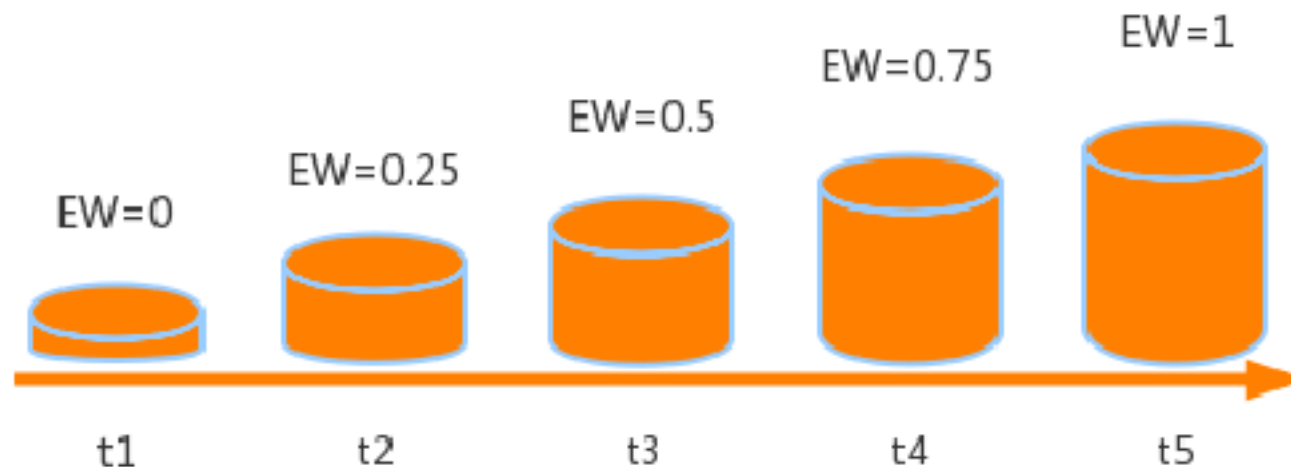
# 美团点评服务治理-整体架构

- 注册中心=MNS服务+sg\_agent进程
- RPC框架mtthrift/pigeon，基于TCP的私有协议，序列化支持thrift/hessian
- 服务治理门户MSGP
- 节点存活探测服务Scanner
- 日均调用量4000+亿次

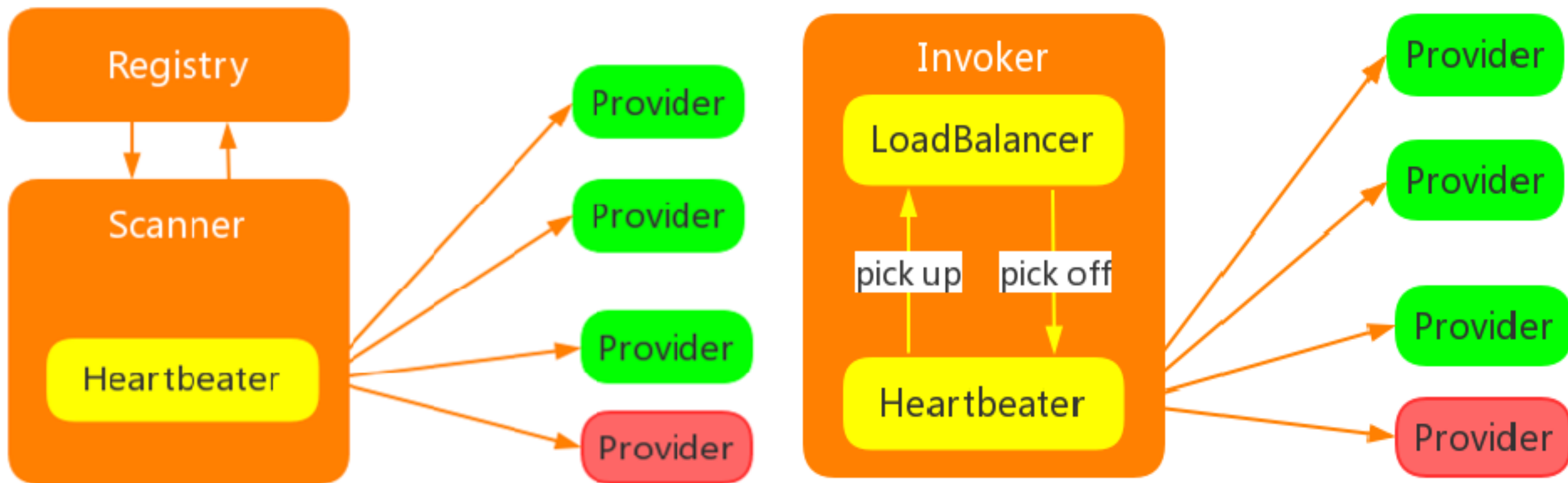


# 美团点评RPC-负载均衡

- 支持sync/future/callback/oneway等多种同步异步调用方式
- 多种负载均衡策略，默认是自适应策略，客户端会计算发往每个服务端节点的在途请求数，新的请求会优先选择在途请求数最小的节点发送
- 支持慢启动，针对服务端某个刚启动的节点，客户端按从慢到快的频率，将请求逐步发往这个节点，防止服务端刚启动的节点大量请求进来导致大量超时



# 美团点评RPC-服务节点存活检测

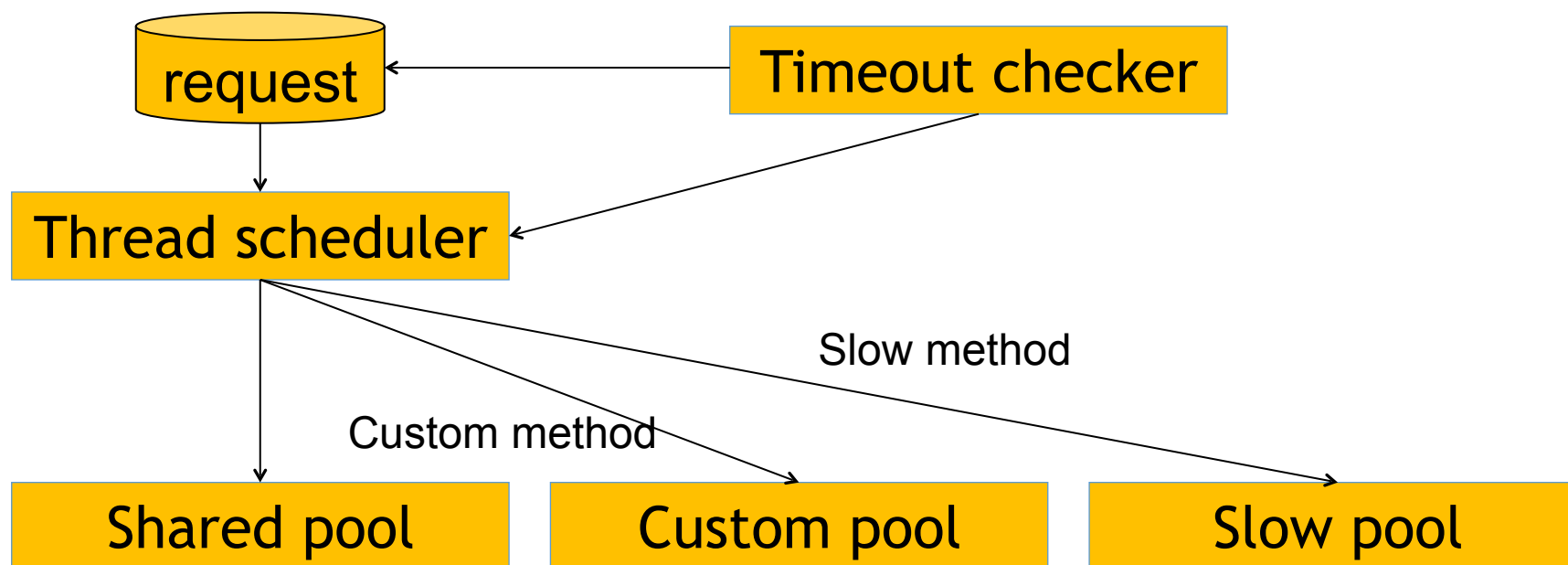


注册中心摘除

客户端心跳检测

# 美团点评RPC-服务限流、隔离

- 支持服务限流，针对接口、方法、调用者分别限流QPS
- 监控每个接口的超时情况，超时多的接口请求会自动路由到独立的慢线程池处理，如果该接口恢复正常，则会回到正常共享线程池，也可以为某些接口方法配置独立的线程池



# 美团点评RPC-服务降级

- 若依赖的服务是弱依赖，可以降级处理，Pigeon提供多种服务自动降级策略
- 降级的结果可以是自动返回默认值（支持json和groovy配置）、或抛出降级异常、或返回mock对象



# 美团点评RPC-服务测试

- Pigeon框架内置http服务，提供ip:4080/services服务测试页面，或通过管理端统一入口进行测试

服务名

方法名

参数类型:com.dianping.joy.event.api.request.GetSkusByShopReq

Response ^

```
▼ {
  respCode: 0,
  respMsg: "成功",
  skuList: [
    1,
    1
  ]
}
```

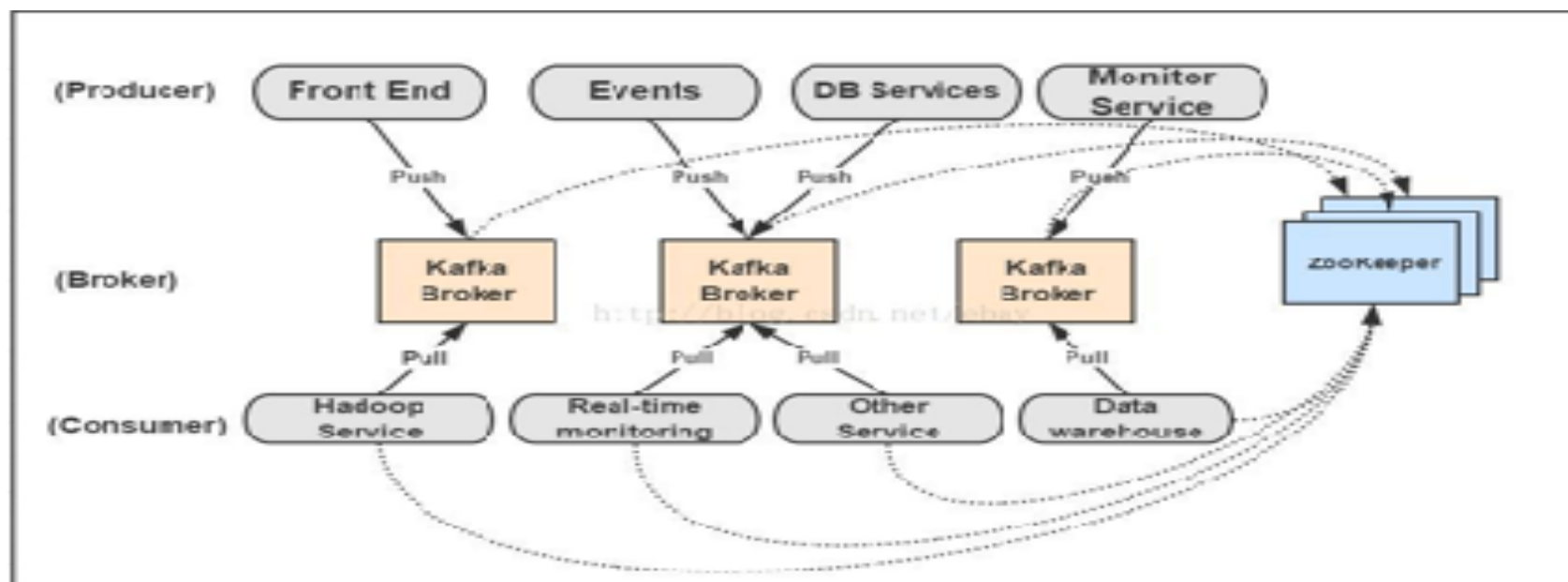


# 美团点评服务治理

- 底层存储如mysql、KV等尽量保证只有一个或少数几个业务功能模块相关的服务访问，每个应用只能访问自己的存储
- 面向业务领域定义服务接口，每个服务高内聚
- 服务可能适时的根据业务发展情况进行拆分或重构
- 服务层级调用，上层可以调用下层，同层可以互相调用，但下层不能调用上层，调用链路深度不超过3级，不循环，尽量异步化、并行化调用
- 区分基础服务和业务服务，区分MQ场景，离线数据抽取不走service
- 配套做好服务流控、降级、监报告警、服务链路分析、日志查询等
- 服务的治理，关注出度入度，可用性、性能基准，平均延时、top99延时

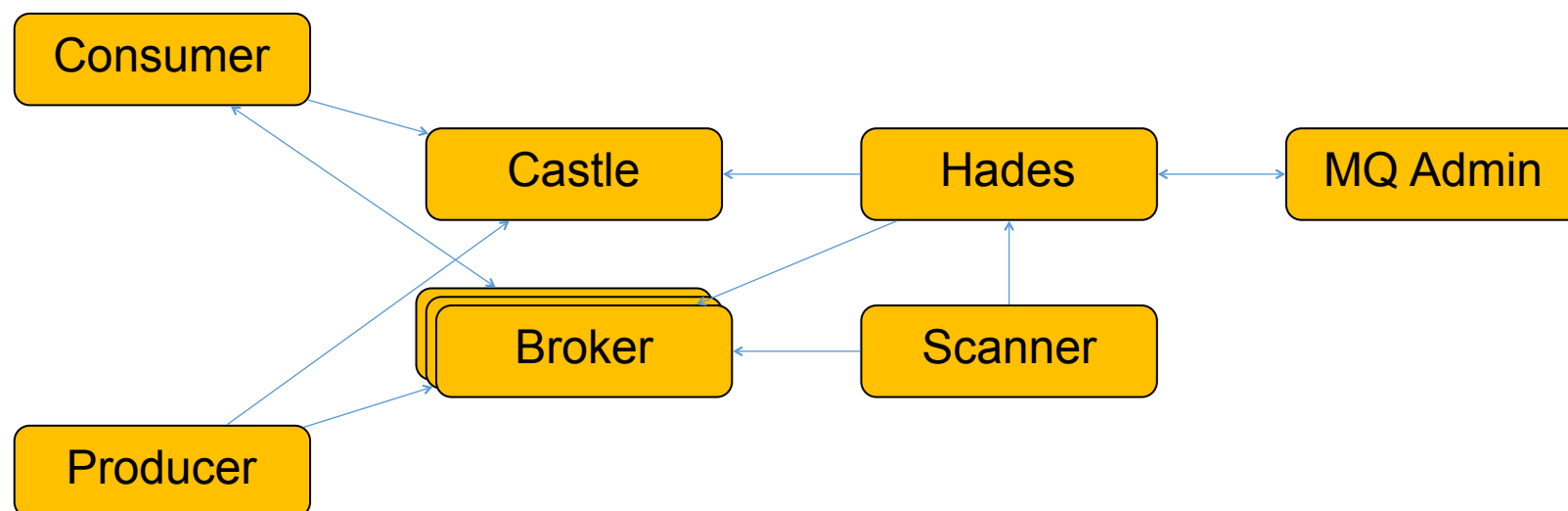
# MQ

- 开源的Kafka、RocketMQ、RabbitMQ、阿里云RocketMQ等
- 屏蔽异构平台的细节：发送方、接收方系统之间不需要了解对方，只需认识消息
- 异步：消息堆积能力，发送方接收方不需同时在线，削峰
- 解耦：避免过多RPC给系统的稳定性带来风险，可实现不同角色的系统解耦
- 复用：一次发送多次消费



# 美团点评消息中间件-Mafka

- Castle负责生产者消费者心跳状态
- Hades负责Broker相关的元信息维护
- Scanner维护Broker存活状态
- 日均消息量150亿左右



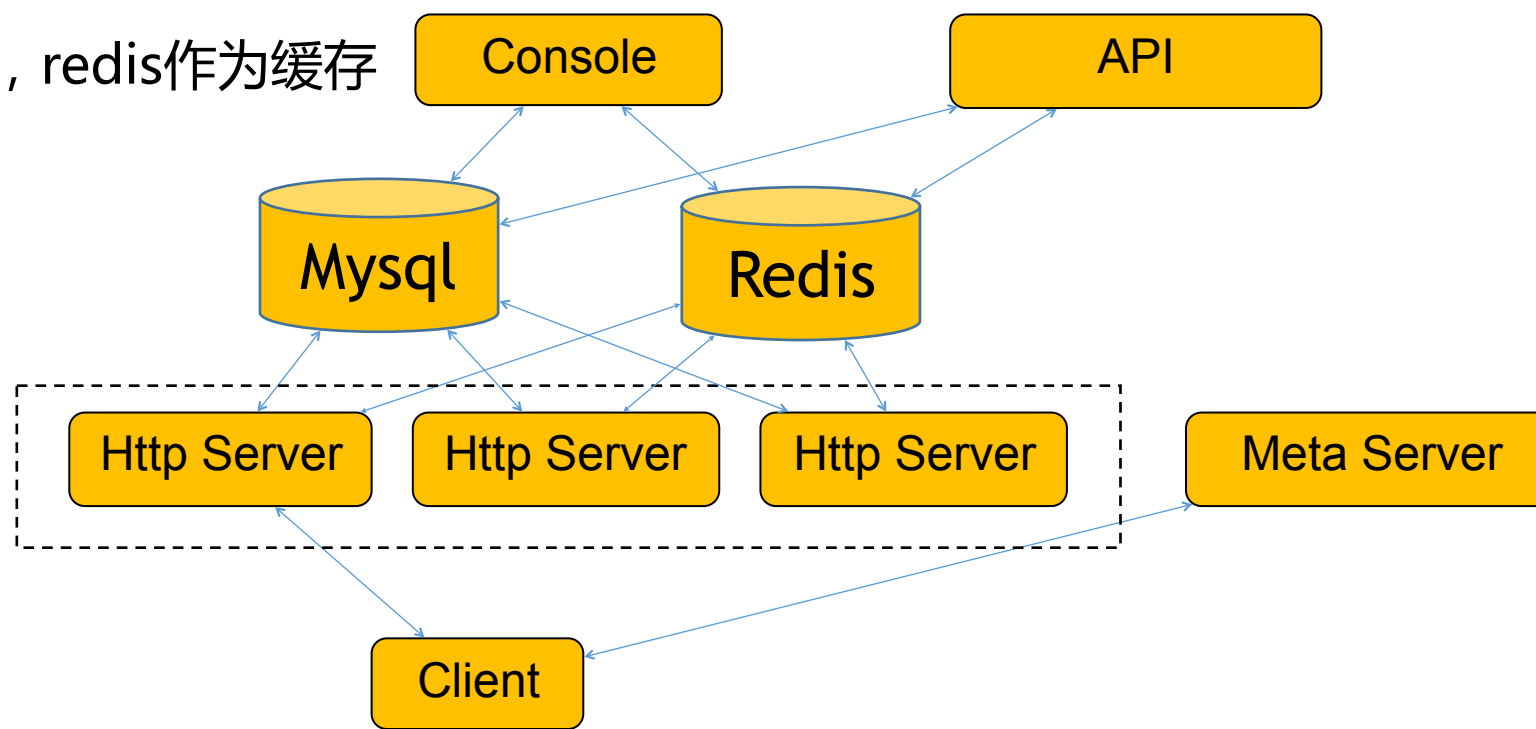
# 配置中心

---

- 解决痛点
  - 统一的管理平台修改配置，避免机器多需要人工每台机器修改配置，提高效率，避免改错、漏改
  - 配置改了可实时通知到应用，无需应用重启
- 业界产品
  - 阿里Diamond、百度Disconf
  - Spring cloud config
  - 基于zookeeper自研
  - 云端PaaS服务：阿里云ACM
  - 美团点评MCC/Lion

# 美团点评配置中心-Lion

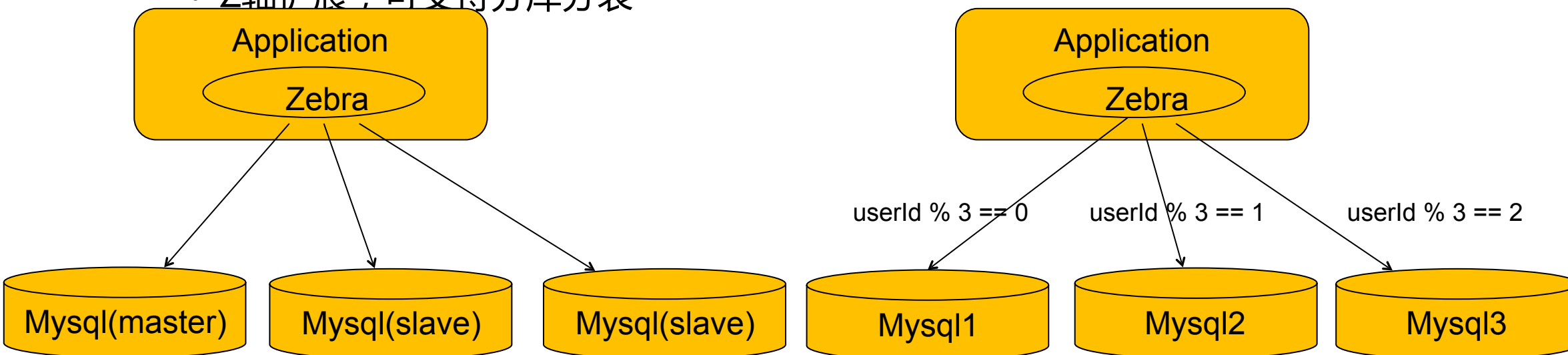
- 客户端通过Meta服务获取Http服务地址列表
- 客户端挑一台Http服务通过http长轮询获取配置并监听配置变化
- 管理端Console将配置写入Mysql和Redis
- 配置存储在mysql，redis作为缓存
- API提供写的入口



# 美团点评数据库访问-Zebra

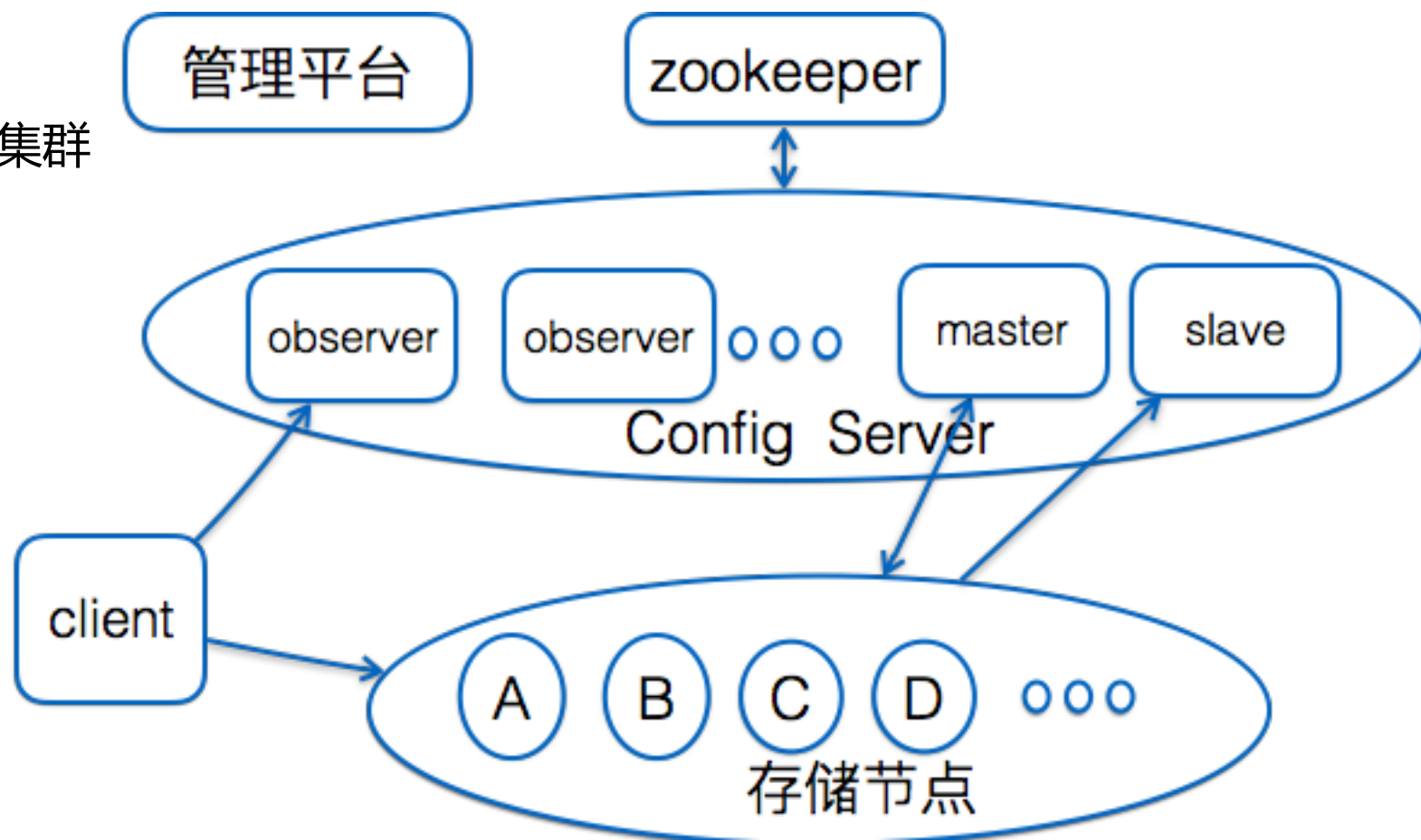
## 解决痛点

- 统一访问数据库的配置，提供丰富的数据库访问监控如SQL耗时、异常等
- 解决数据库访问的性能问题，可实现读写分离
- 可用性故障，多个读库之间自动failover
- Z轴扩展，可支持分库分表



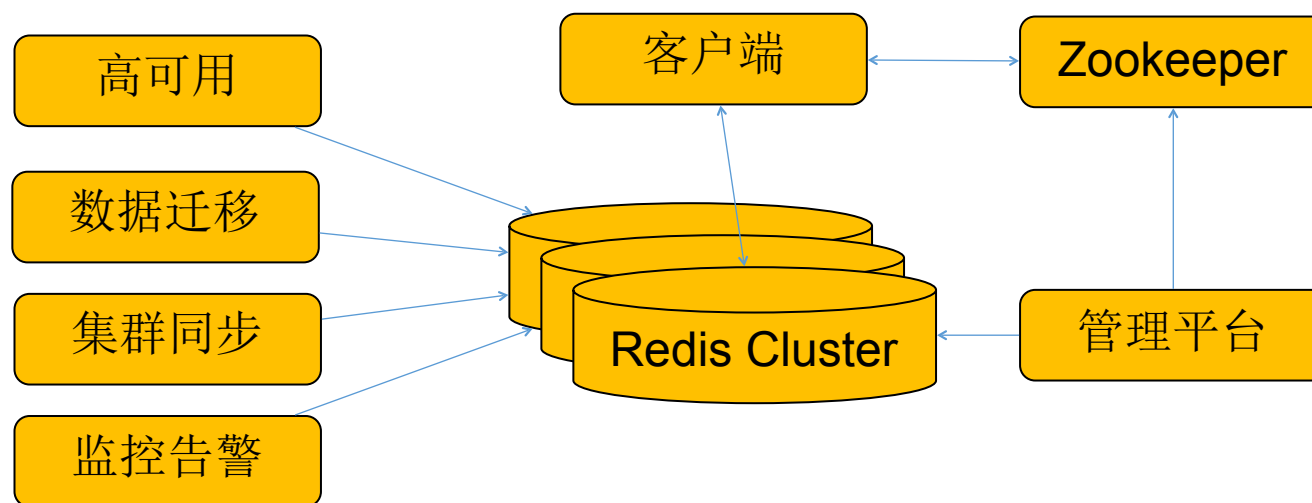
# 美团点评KV-Cellar

- Cellar适合持久化、数据量大、冷热数据明细的业务场景
- 支持leveldb、rocksdb引擎
- 日均调用量近3万亿次，200+集群



# 美团点评KV-Squirrel

- Squirrel基于redis cluster，适合缓存或数据量可控的持久化场景，数据结构丰富
- 日均请求量1万亿次，700+集群





# 选择合适的存储

- 根据业务选择合适的存储，尽量使用扩展性更好的Nosql产品
- 在线业务建议尽量使用KV系统，可用于分布式缓存，加速查询能力，也可直接作为持久化存储，取代关系型数据库
- 常见的关系型数据库访问层：阿里TDDL、美团点评Zebra、阿里云DRDS等
- 常见的KV系统如Redis、Tair、云端Redis等

# Q & A