

当“配置”遇到“服务”

配置服务的前世今生

关于我

- 7年DevOps@Sina
- 曾负责系统平台研发与运维
- 同时支撑新浪与微博
- Huawei工程师
- 喜欢做一切有趣的事





前方高能
-WARNING-



南

北

干

货

湿乎乎的话题

- 让我们一起设计一个配置服务
 - 为什么要“配置”？
 - 为什么要“配置服务”
 - “配置服务”粗狂设计

配置



CORSAIR
DOMINATOR GT



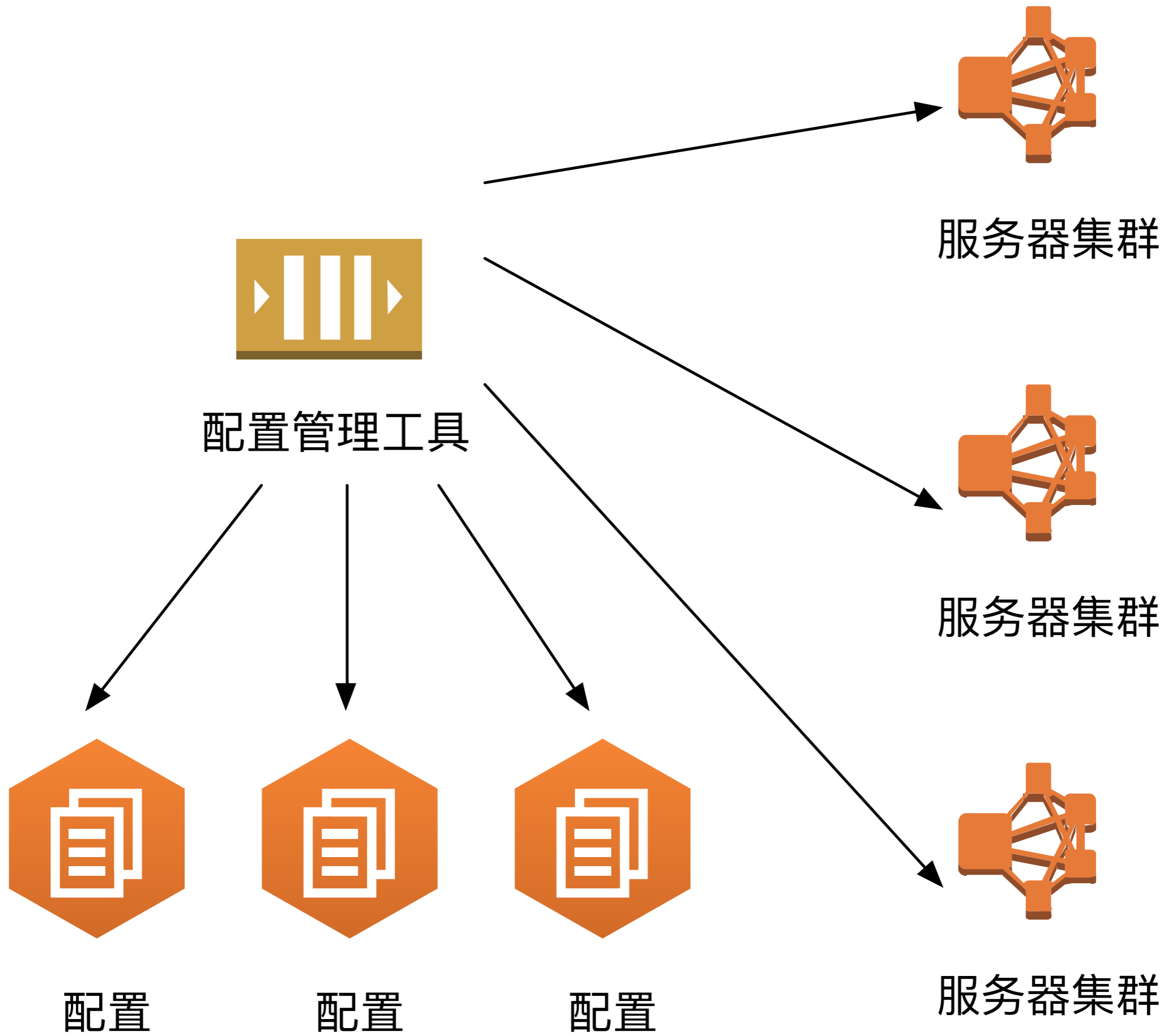
DICKINSON
GAMERSKY



配置文件？ 还有其他的吗？

环境变量

当服务器变多。。。。



配置管理出现了。。。。

云计算到底带来什么
不一样的东西？

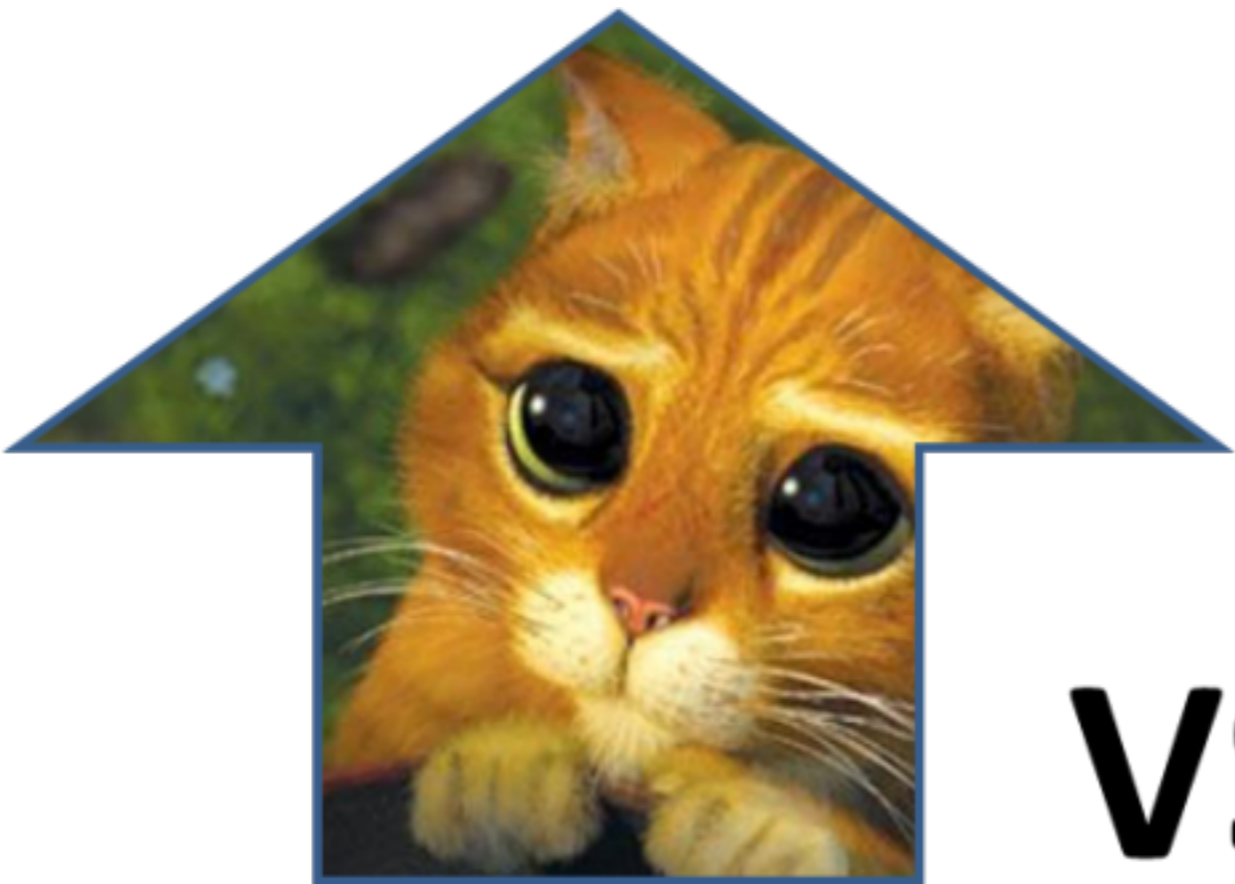
单应用大规模或多实例

->

多应用小规模单实例

->

多应用大规模多实例



Scale Up

VS



Scale Out

更频繁的配置变化
更快速的配置生效

后端服务访问入口

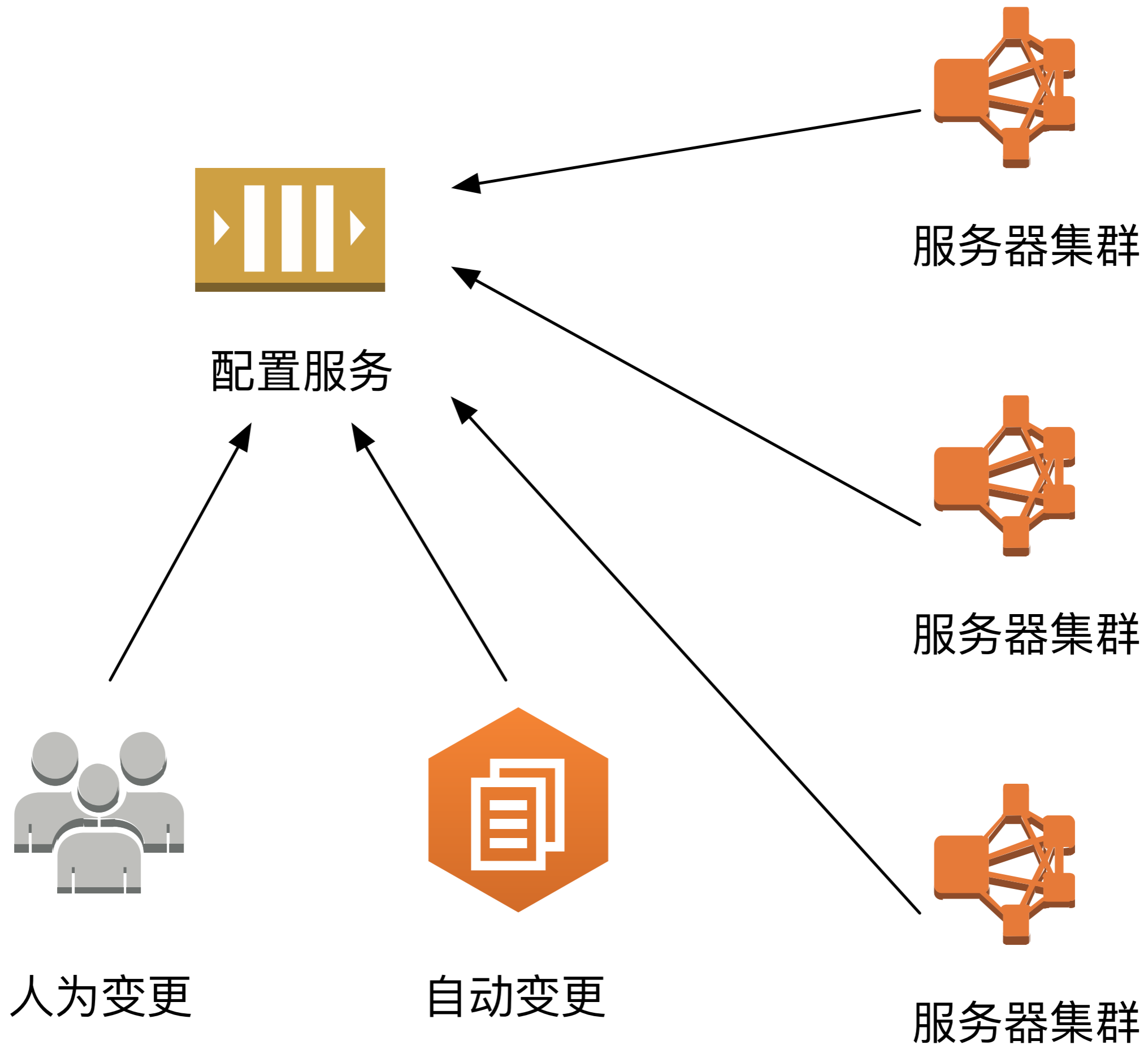
根据地理位置或角色产生的差异配置

降级开关

吞吐-延迟



嗯，于是配置服务出现了



弱耦合

Push

如何设计一个配置服务

- 架构设计原则
 - 简单
 - 扩展
 - 性能
 - 可靠
 - 通用
 - 功能

配置服务需求

- 数据模型
 - 数据结构
 - 数据容量
 - 数据变化频率
 - 数据一致性需求

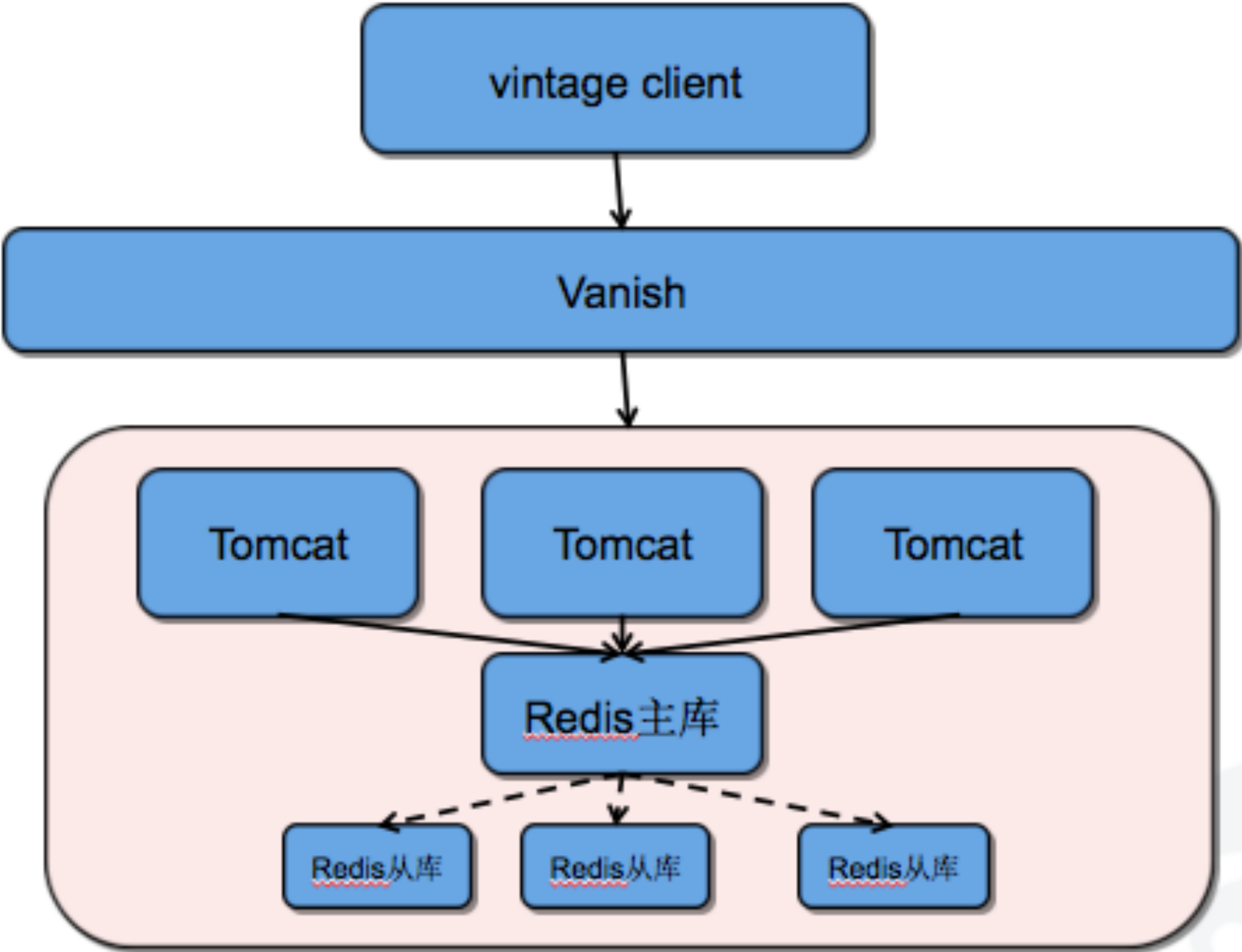
配置服务需求

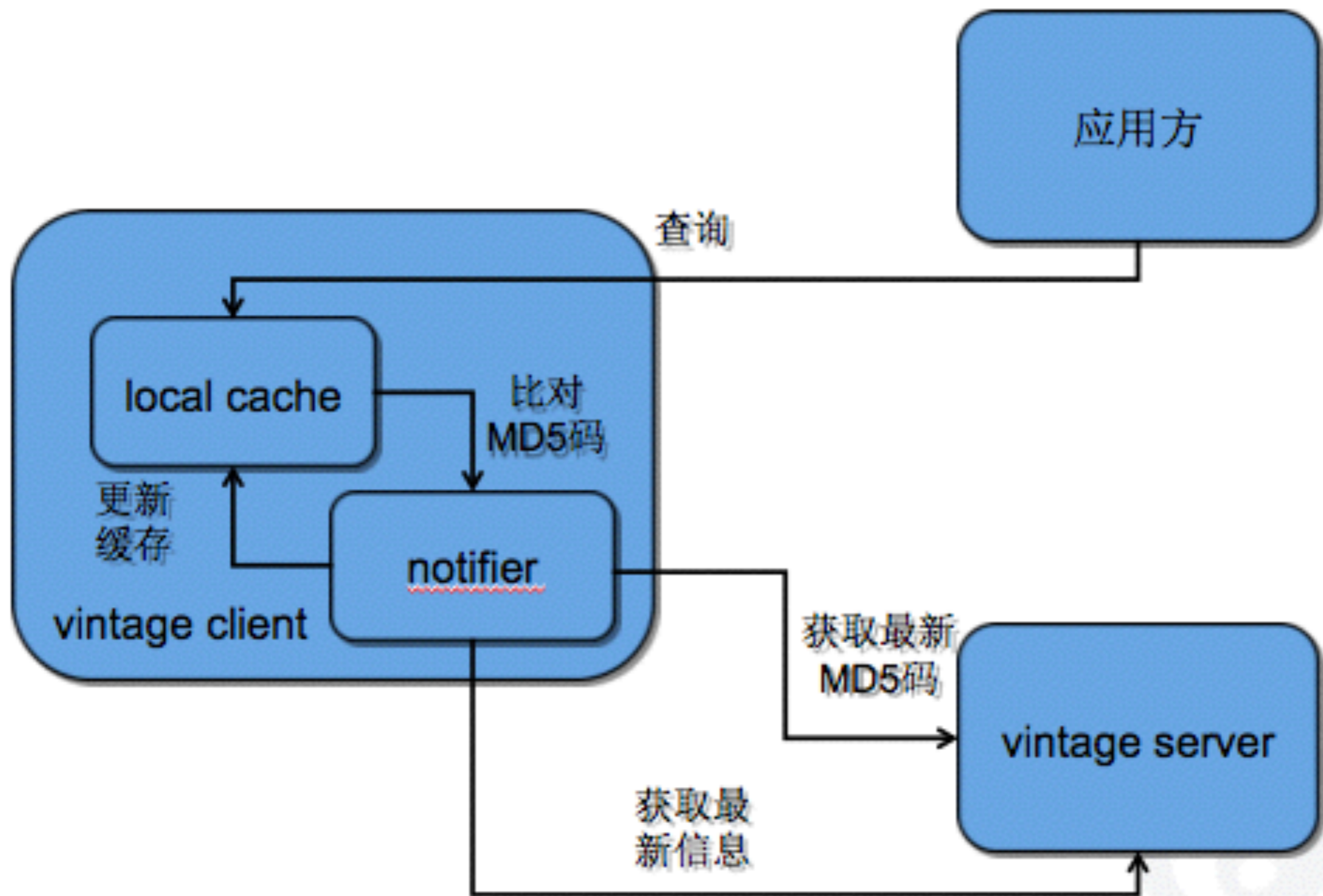
- 数据变化频率较低-可使用缓存
- 一致性要求-可容忍最终一致性
- 数据更新时-希望缓存主动过期或更新
- 可与监控程序联动

配置服务实现

- 取决于你应用与团队情况对于简单、功能、可用性与数据一致性的要求
- 存储：Redis? MySQL? ZooKeeper? etcd?
- 算法：Master-Slave? Raft? Paxos?
- 其他：HTTP API? RPC? Cache?

Weibo-Vintage





谢谢

Q&A