

滴滴Passport设计之道

——一切为了高可用

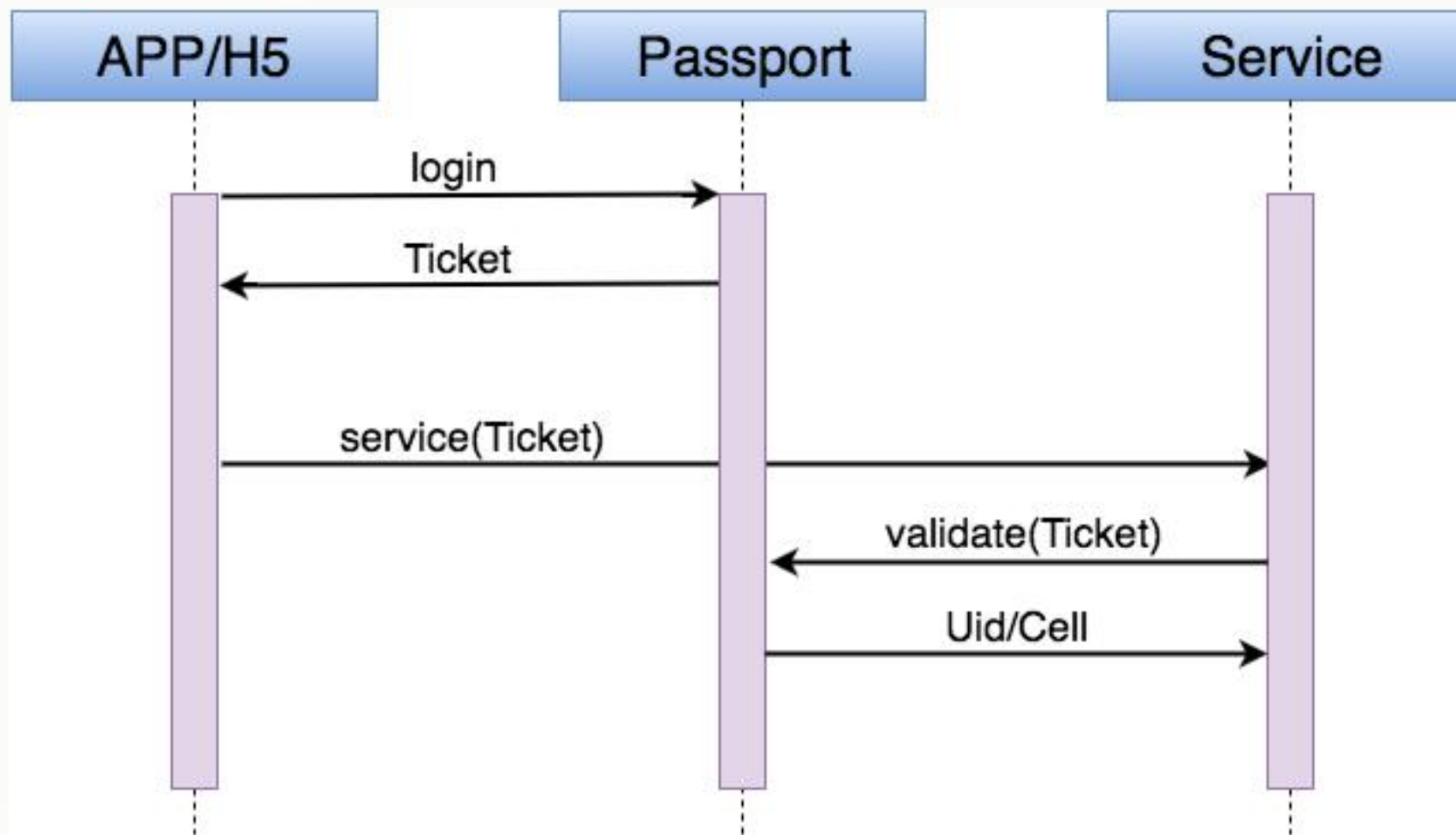
洪泽国 2016-06-07



Agenda

- 业务场景
- 一切为了高可用
- 效果
- 坑

业务场景



一切为了高可用

- 编程语言
- 最小闭环
- 柔性 & 降级
- 异地多活
- 访问控制
- 接口拆分

编程语言

PHP -> 全栈golang

最小闭环

- account: uid、cell、password|name、nick、id_card...
- 能登录 vs 不能登录
- 最小闭环: uid/cell/password
- 高可用变得简单

柔性&降级 – Ticket设计

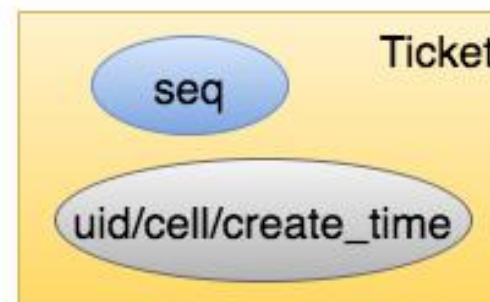
- Ticket: 可剔除



uid : Ticket

Ticket == Ticket

1. Ticket无语义;
2. 不好做降级。

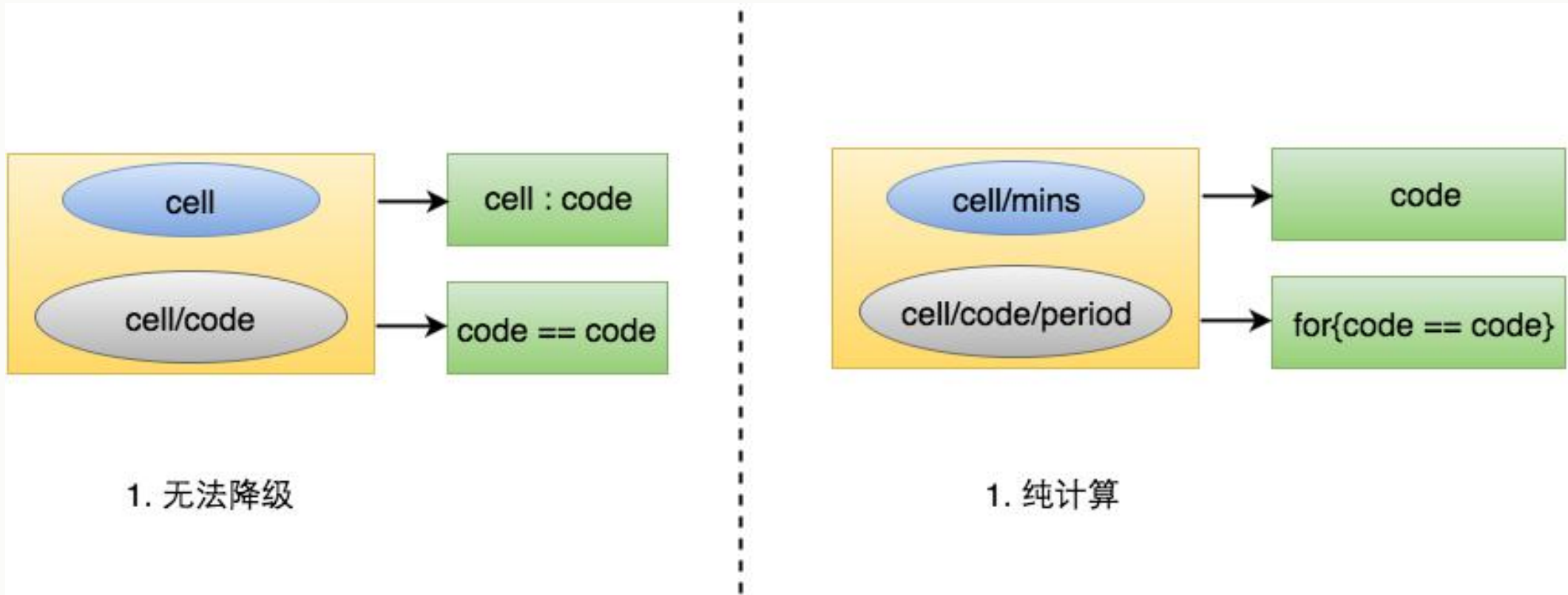


seq++

seq == seq

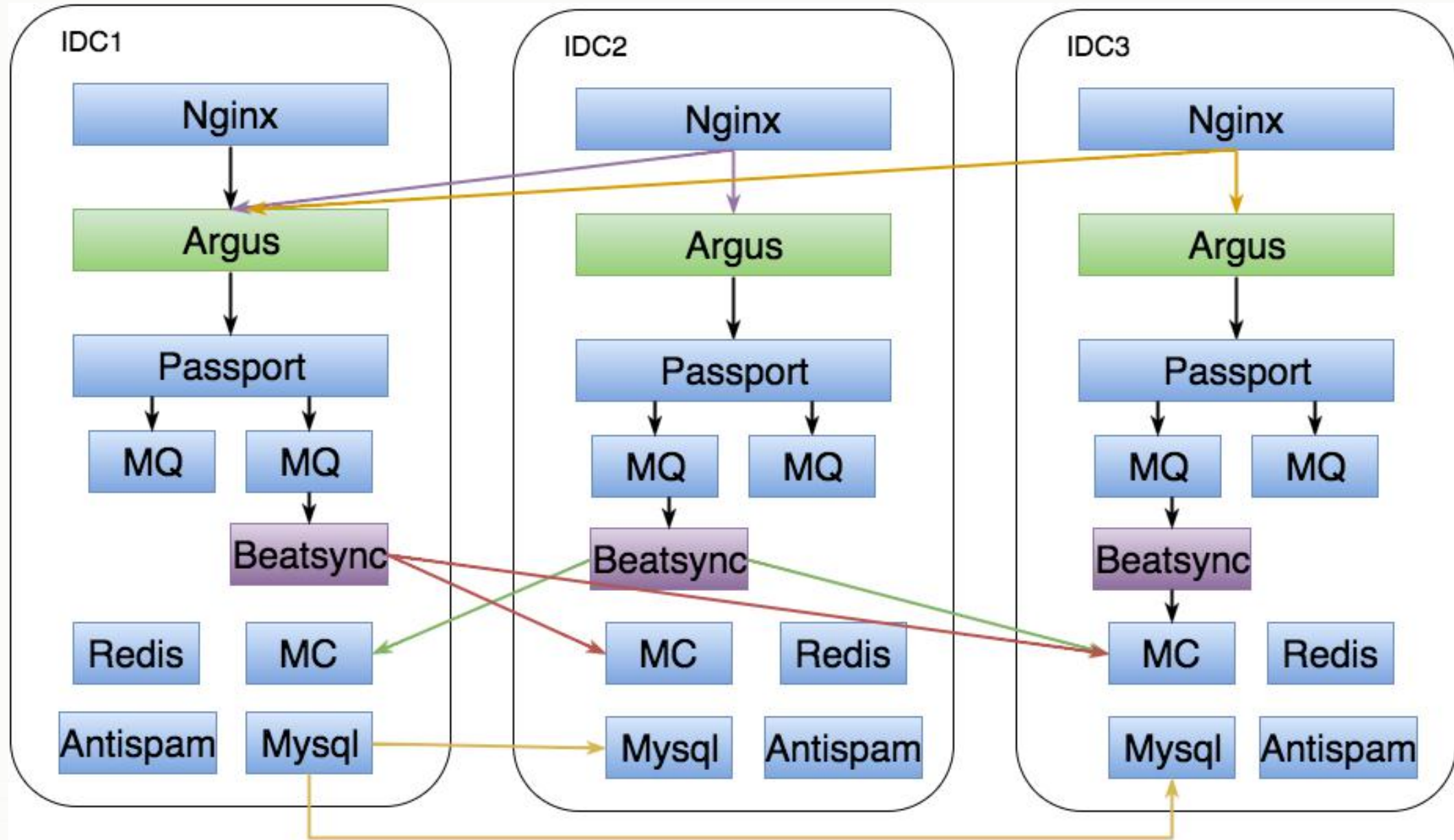
1. self-contained
2. no refresh-token
3. 柔性降级
4. 少存储

柔性&降级 - 短信验证码

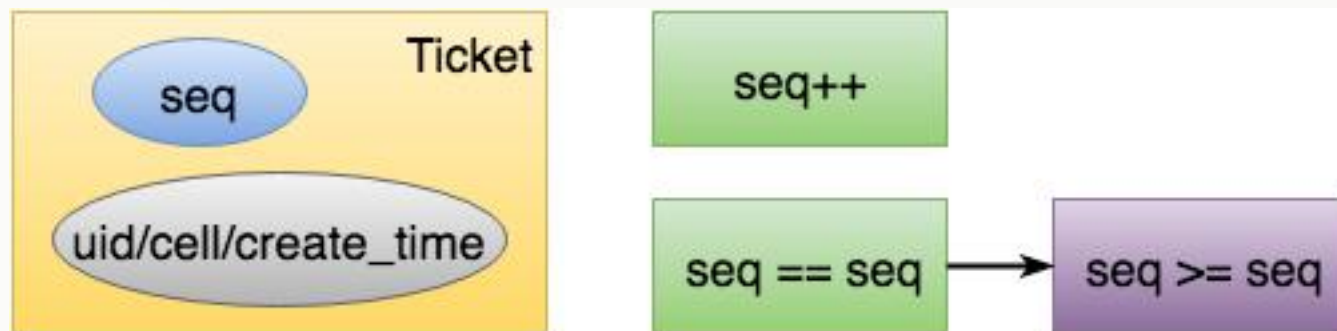


异地多活(1)

- 3 IDCs
- 调用方按IDC独立部署
- 场景分析：读、写（登录、注册）



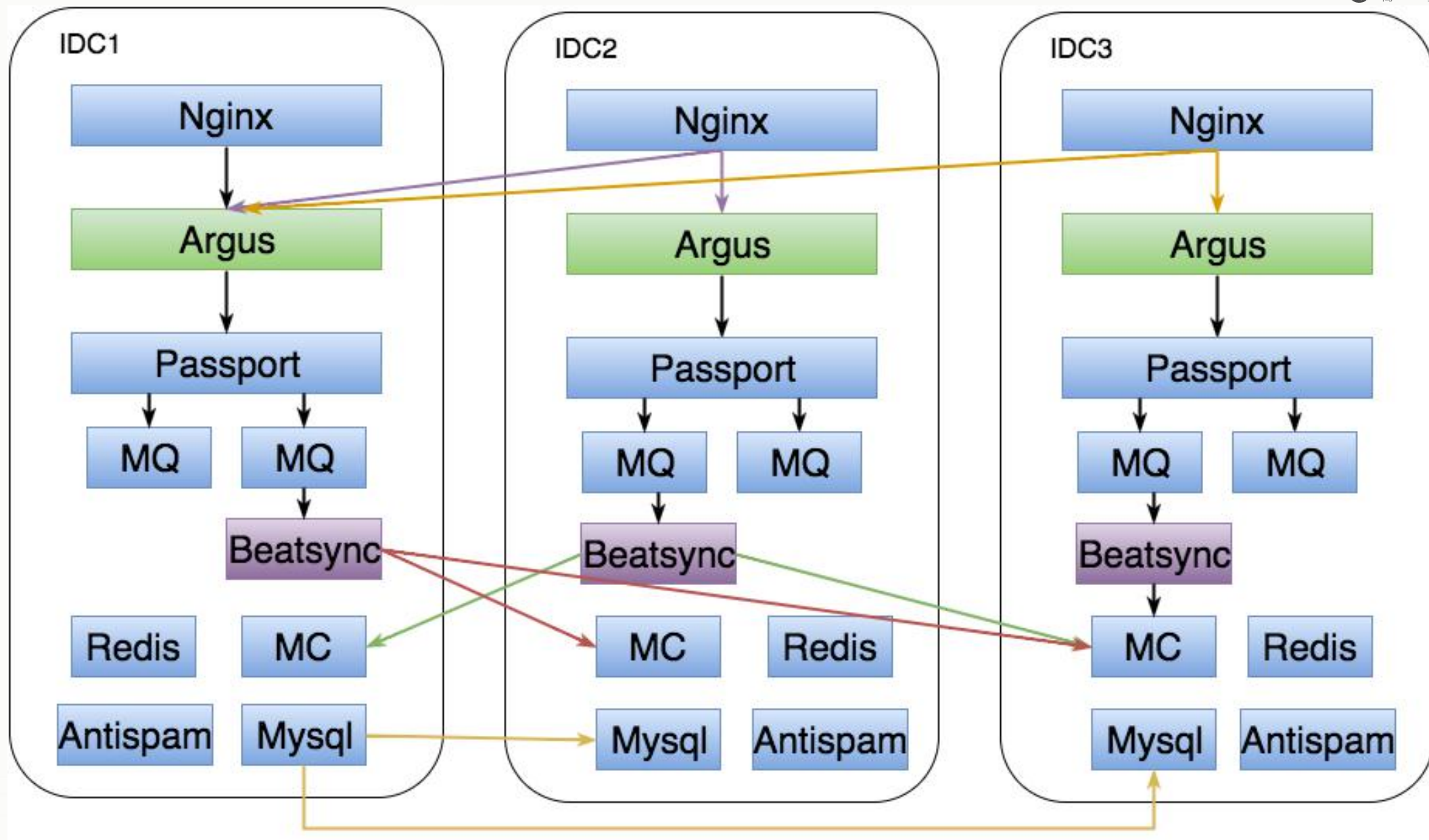
异地多活(2)



- 鉴权、登录、其它读，注册暂时不支持

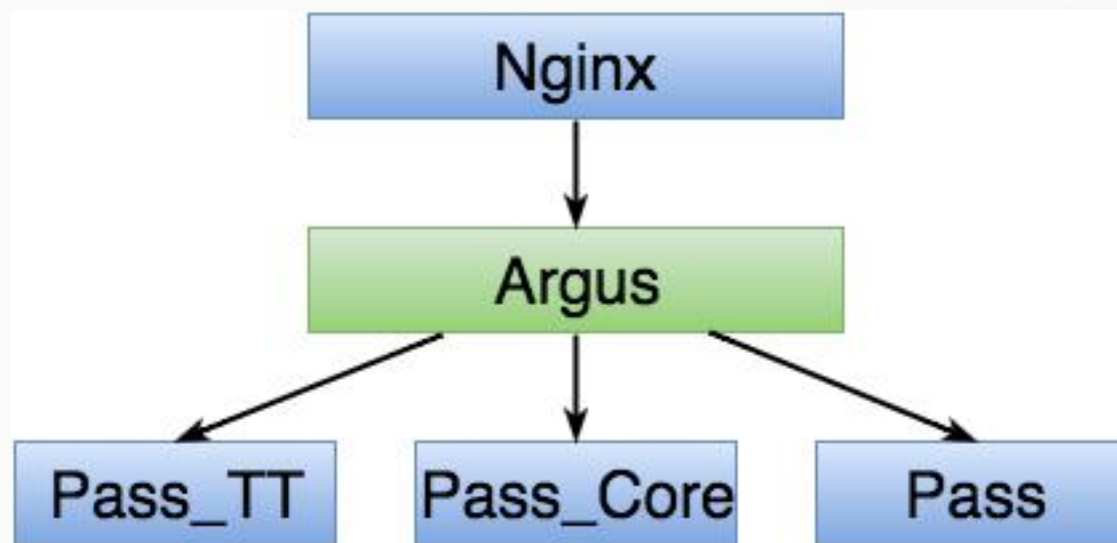
访问控制

- 过载保护、权限控制、白名单等
- 调用方多活，如何做过载保护？？？



接口拆分

- 资源占用：非对称加密接口
- 分级：鉴权等核心接口



效果

- SLA指标:
 - 单机QPS: >1w, 可平行扩展
 - RT: 高并发接口<5ms, 密码相关接口200ms
 - 可用性: 99.99? 零故障
- 口碑

一句话总结

什么是“一切为了高可用”？

柔性降级 + 冗余

坑

- 被攻击：发短信、登录、鉴权
 - 蜜罐、人机识别
- 全Golang，性能不错，连接池有些坑
 - 1.4不能设超时，链接打满处理策略
- 故障总是抛到Passport
 - 不是好差事！！！！

Q & A



北京滴滴无限科技发展有限公司

北京市海淀区东北旺路8号院尚东·数字山谷B1号楼