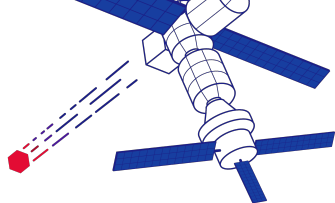


# TiDB在二维火的实践 及二维火海外机房构建

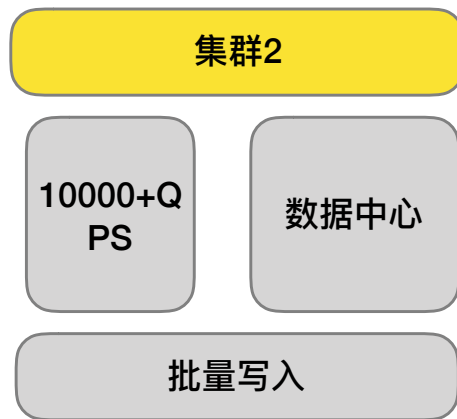
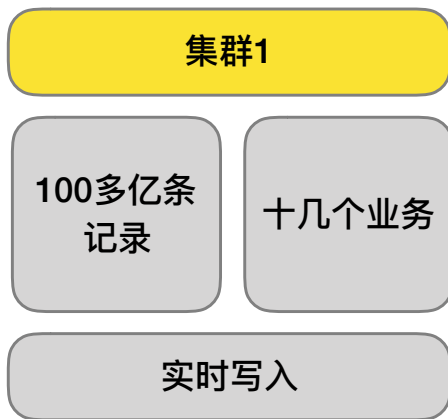
二维火-童云峰(火烧)

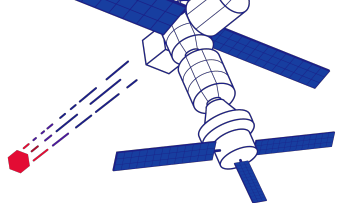




## TiDB 在二维火的使用状况

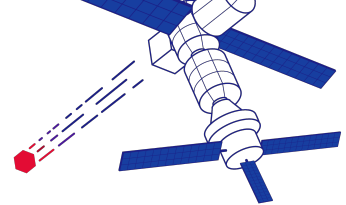
- ▶ 2017年9月至今
- ▶ Pre-GA => 2.1
- ▶ 涵盖OLTP&OLAP
- ▶ 两套集群



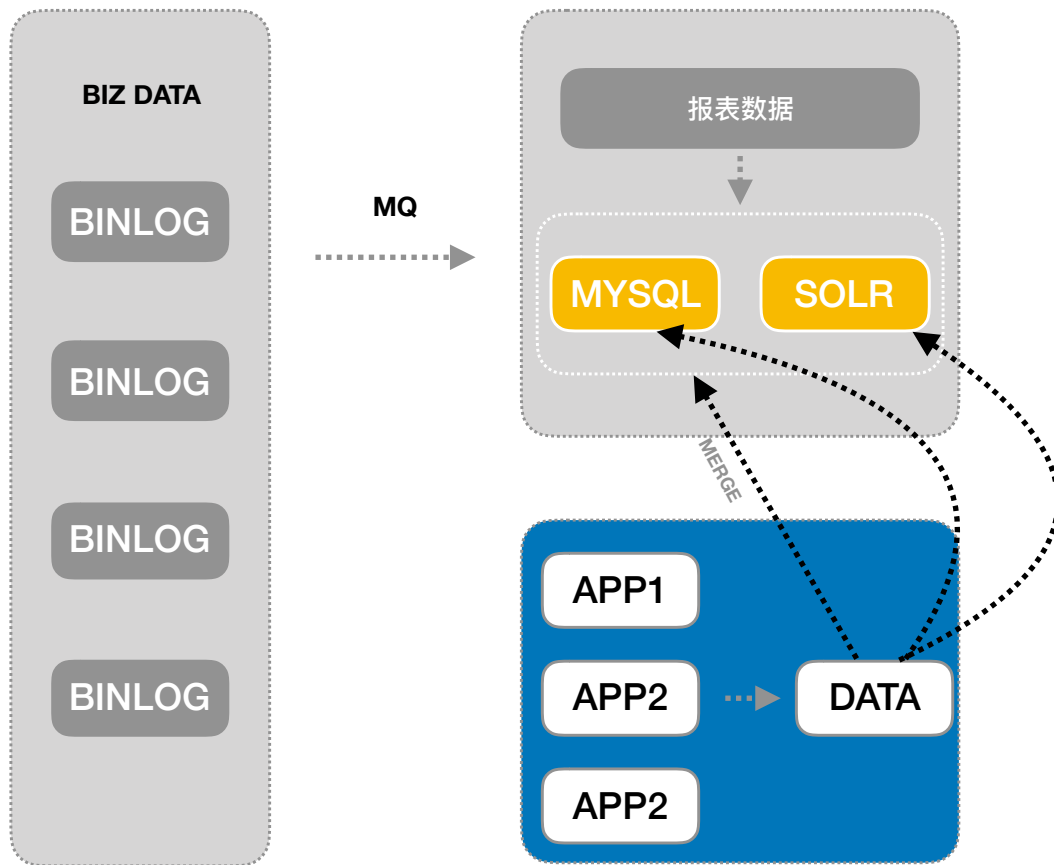


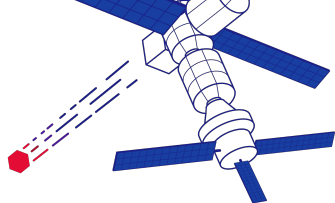
## 场景-数据大屏及报表原架构痛点

- ▶ 报表的几个场景需要上百个分库,扩容、合并麻烦
- ▶ 店铺关系调整,报表需要T+1计算
- ▶ 多数据源(mysql,solr)合并计算,代码冗长层层嵌套容易出错
- ▶ 连锁维度下多维度数据联合查询效率低下



## 场景-数据大屏及报表原架构模式

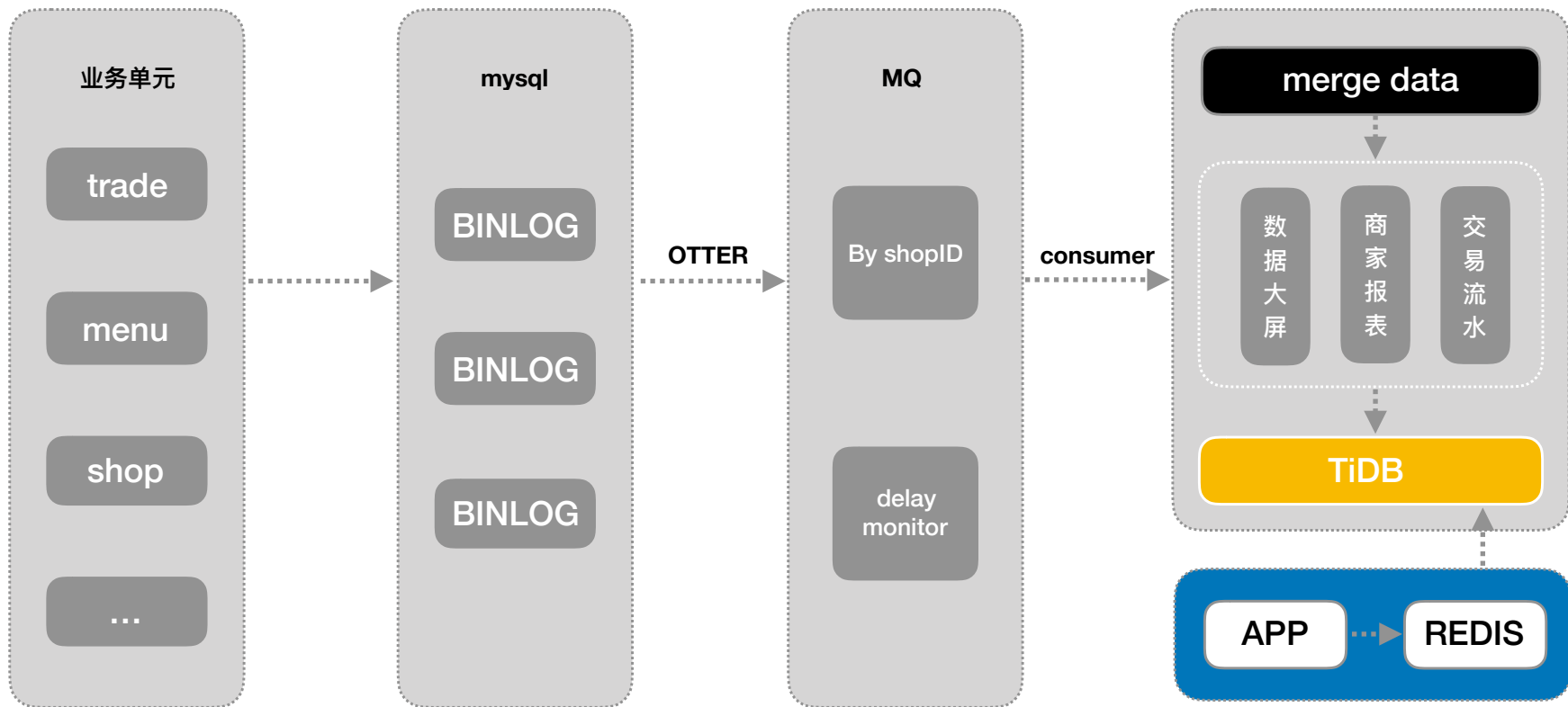


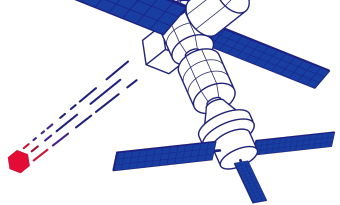


## 场景-数据大屏及报表之TiDB模式

- ▶ 不用分库,随时扩容
- ▶ 解决了分库情况下的连锁店报表问题
- ▶ 从T+1到准实时
- ▶ 解决了SOLR不断变更字段的FULL BUILD以及跨宽表难查询的问题
- ▶ 解决了多业务形态多维度数据的准实时查询

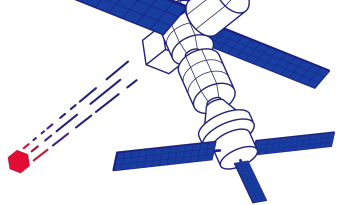
## 场景-数据大屏及报表之TiDB模式





## 场景-常规业务

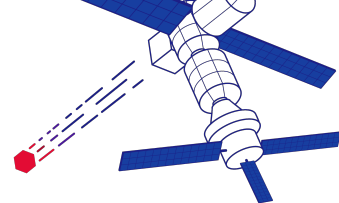
- ▶ 数据量较大
- ▶ 无法分库场景
- ▶ 减少依赖,DDL不用再和数据库中间件耦合
- ▶ 业务场景:审计日志、营销、设备关系等
- ▶ 系统稳定运行500天以上,节点增加无缝
- ▶ 大部分是OLTP类场景



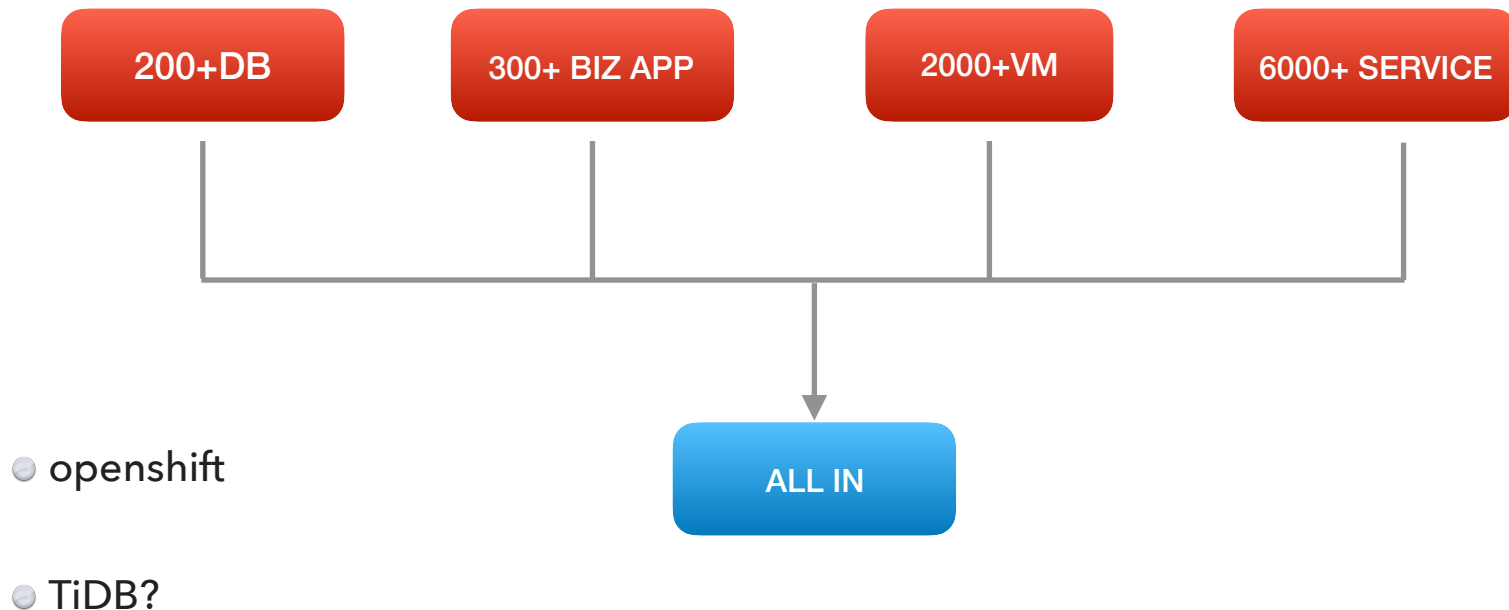
## 海外机房构建-背景

- ▶ GDPR
- ▶ 响应延迟
- ▶ 独立机房
- ▶ 全量构建
- ▶ 独立业务
- ▶ 一个半运维
- ▶ 交付时间一个月之内
- ▶ 网络慢,很慢

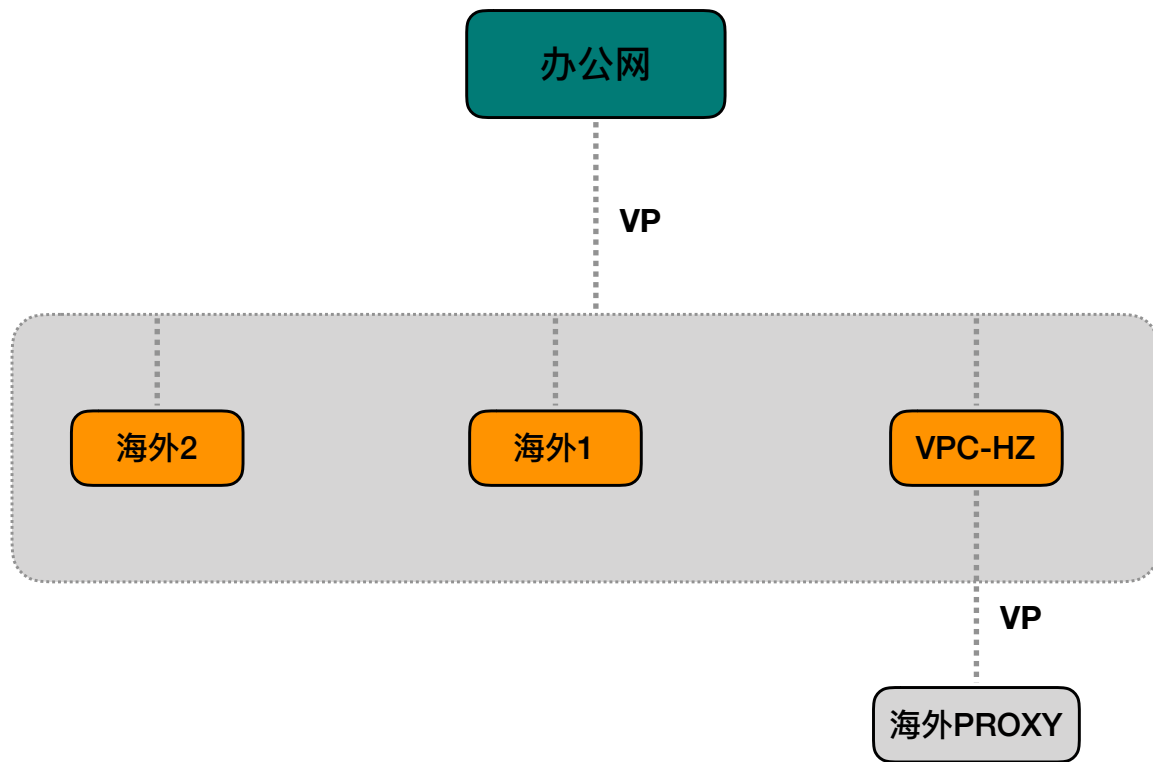
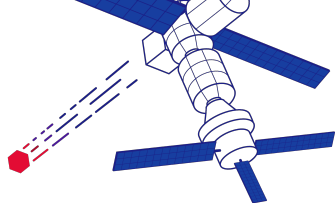




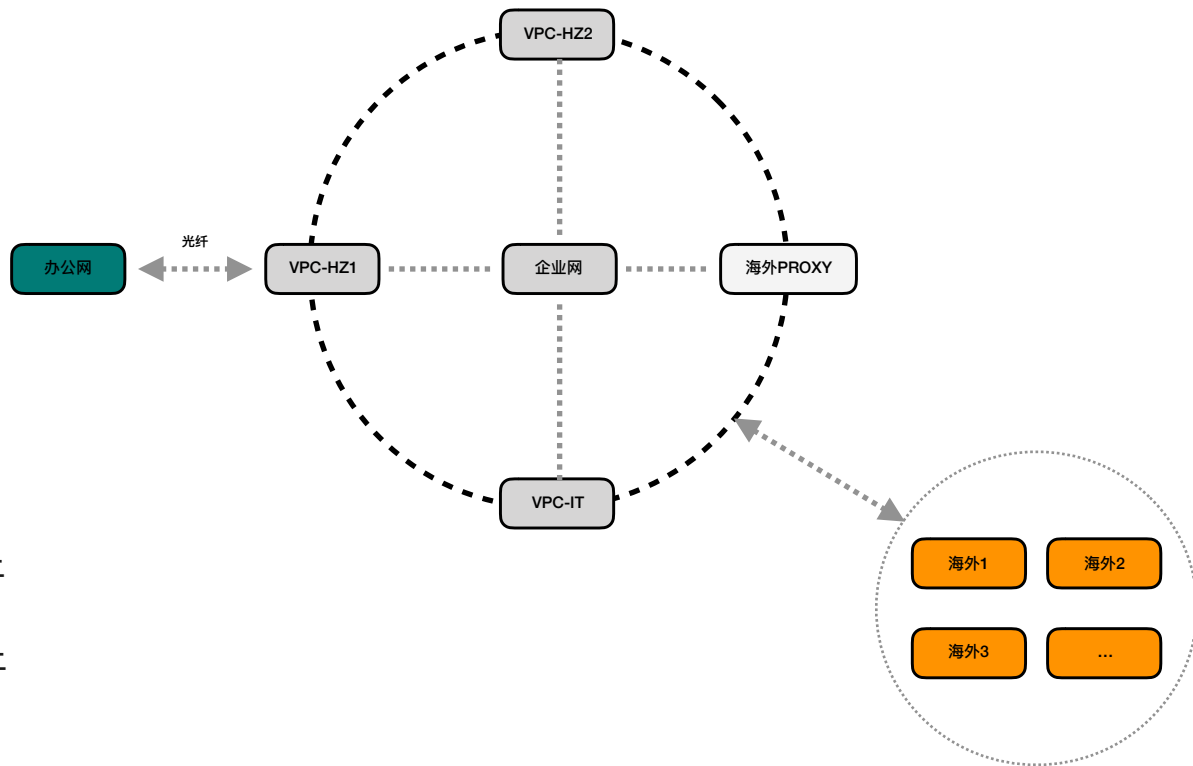
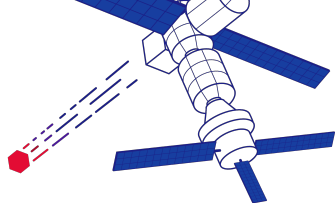
## 海外机房构建-服务状况



## 海外机房构建-原网络架构

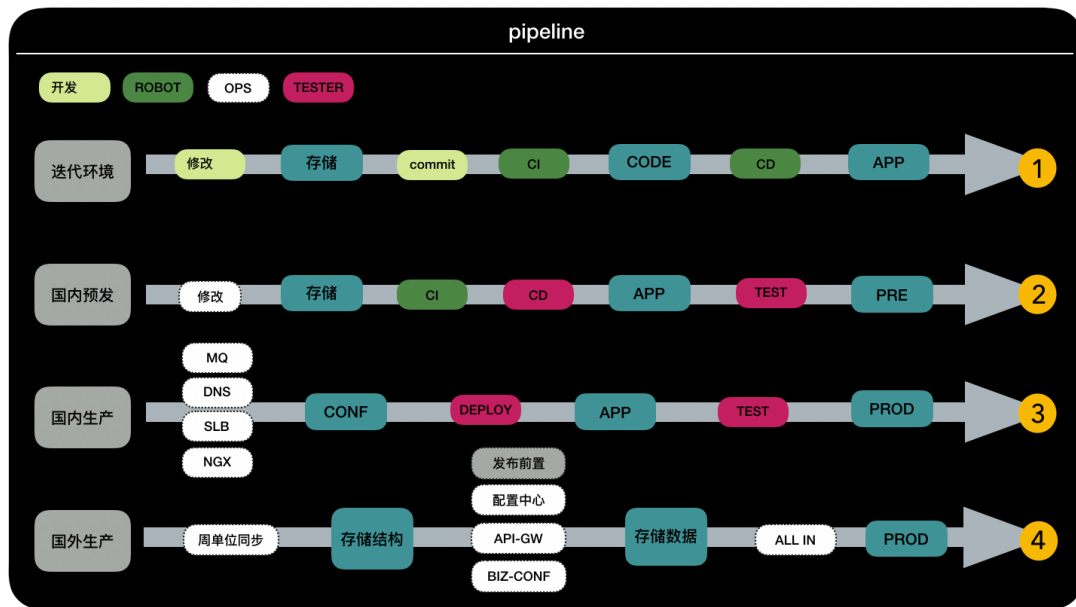


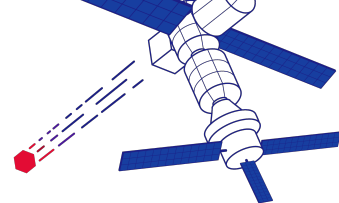
# 海外机房构建-网络



- 费用降低
- 稳定性提升3倍以上
- 延迟降低100%以上

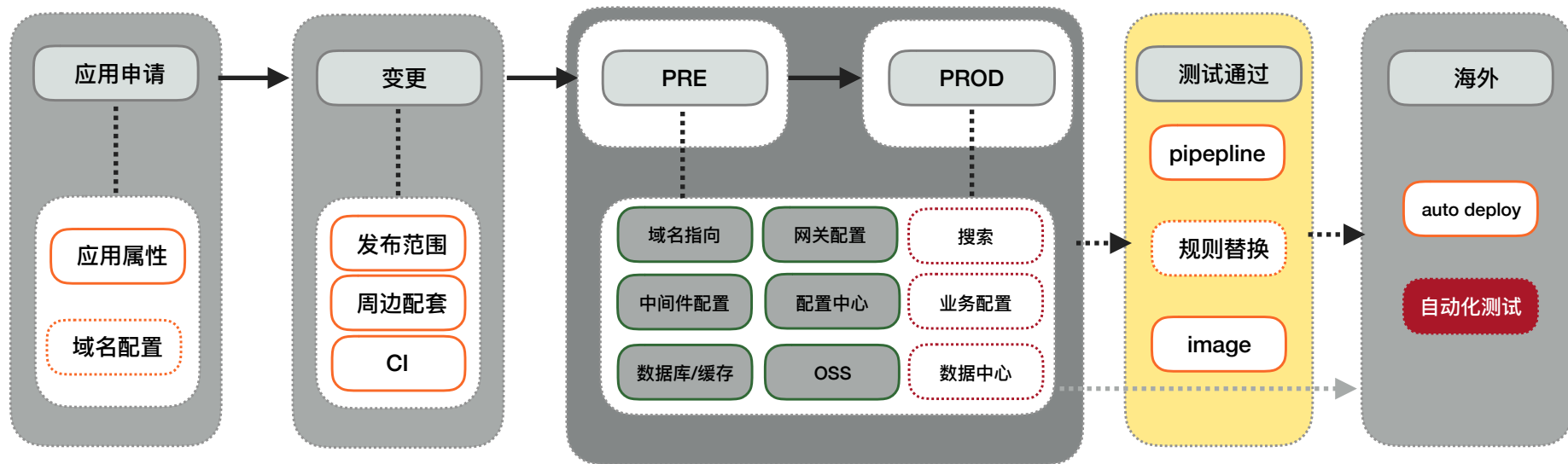
# 海外机房构建-ALL IN发布





# 海外发布-从ALL IN到增量

## 增量发布流程





# Thank You !

