

搜索广告引擎测试持续集成 进阶之路

Qtest 杨会会

目录

CONTENTS

part1

背景介绍

part2

改进方案

part3

效果介绍

part4

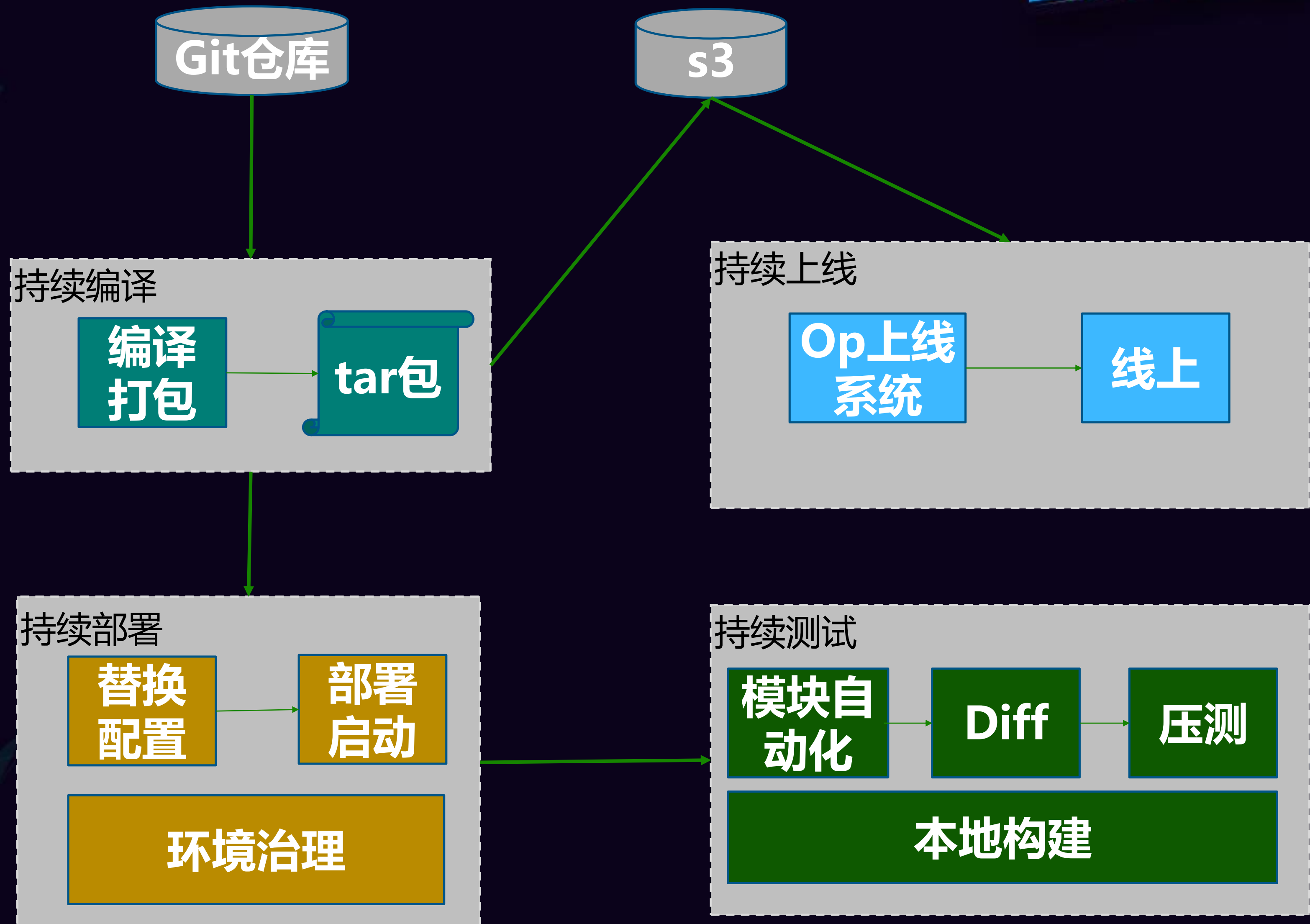
未来展望

Part 1：背景介绍

代码管理

- 1、纯手工
- 2、效率低
- 3、RD繁忙
- 4、QA作用不大

Part 2：改进方案



代码
管理

所有代码git管理

Git代码VS线上代码

编译

一键式编译

静态编译VS动态编译

打包

全量打包

全量打包VS增量打包

两套环境

多套环境

容器化

■优点:

- 环境测试维护
- 系统diff和压测

■缺点

- 系统级测试
- 并行项目排队
- 机器利用率低

■优点:

- 任务不排队
- 机器可以重复利用

■缺点

- 多人使用污染严重
- 利用率仍有提升空间
- 不稳定, 隐藏问题多

■优点:

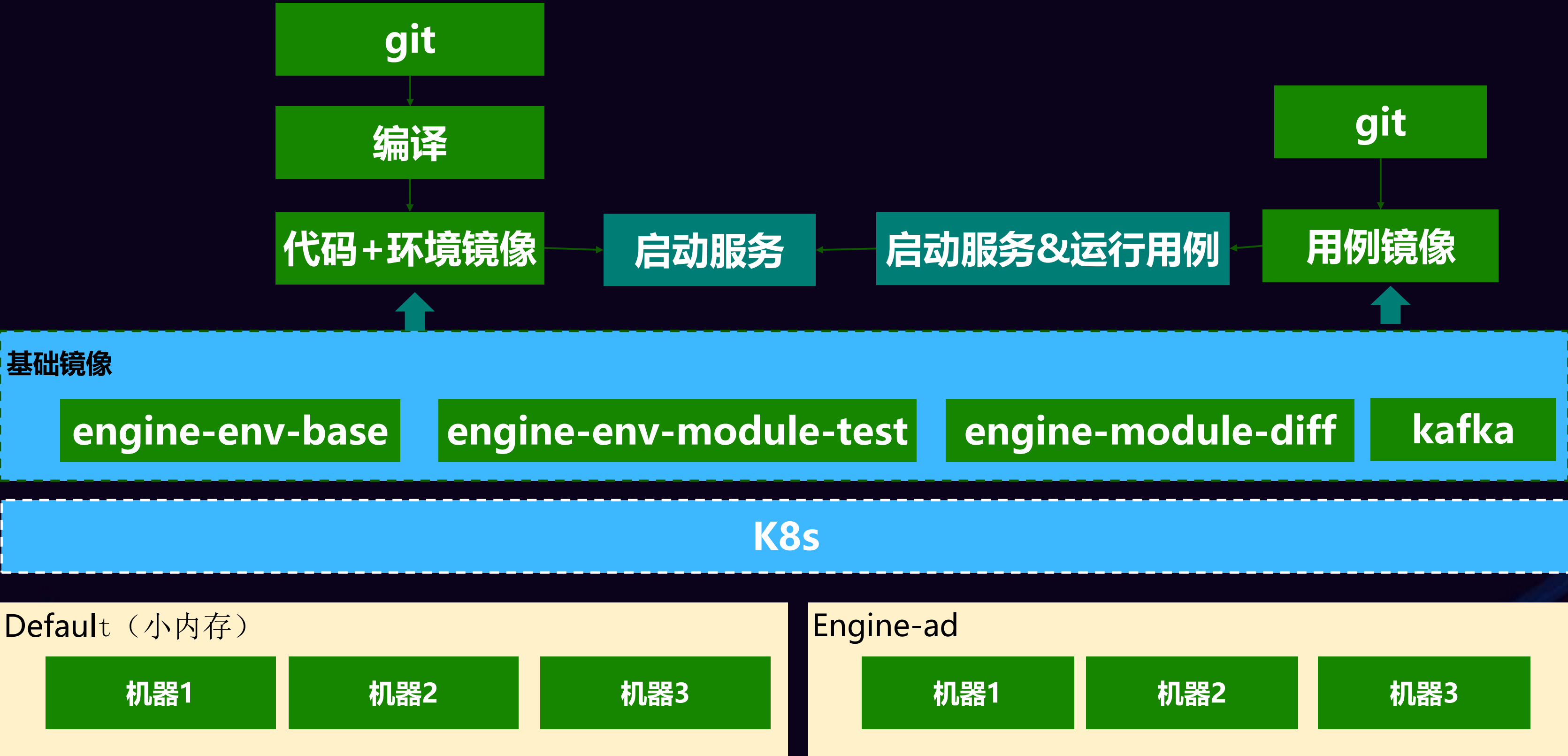
- 任意套环境
- 环境完全隔离
- 机器利用率提高
- 稳定

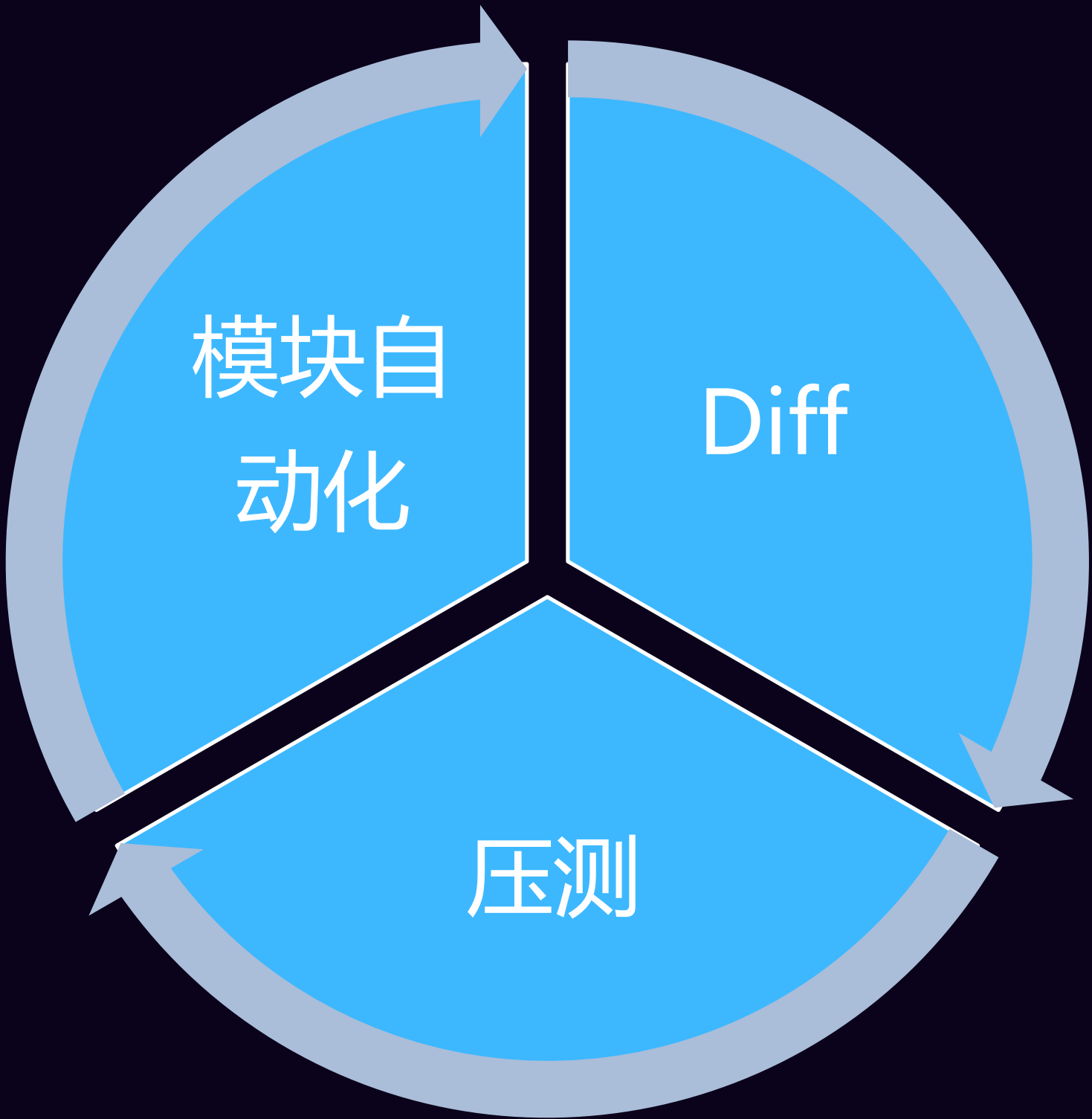
■缺点:

- 资源仍有浪费

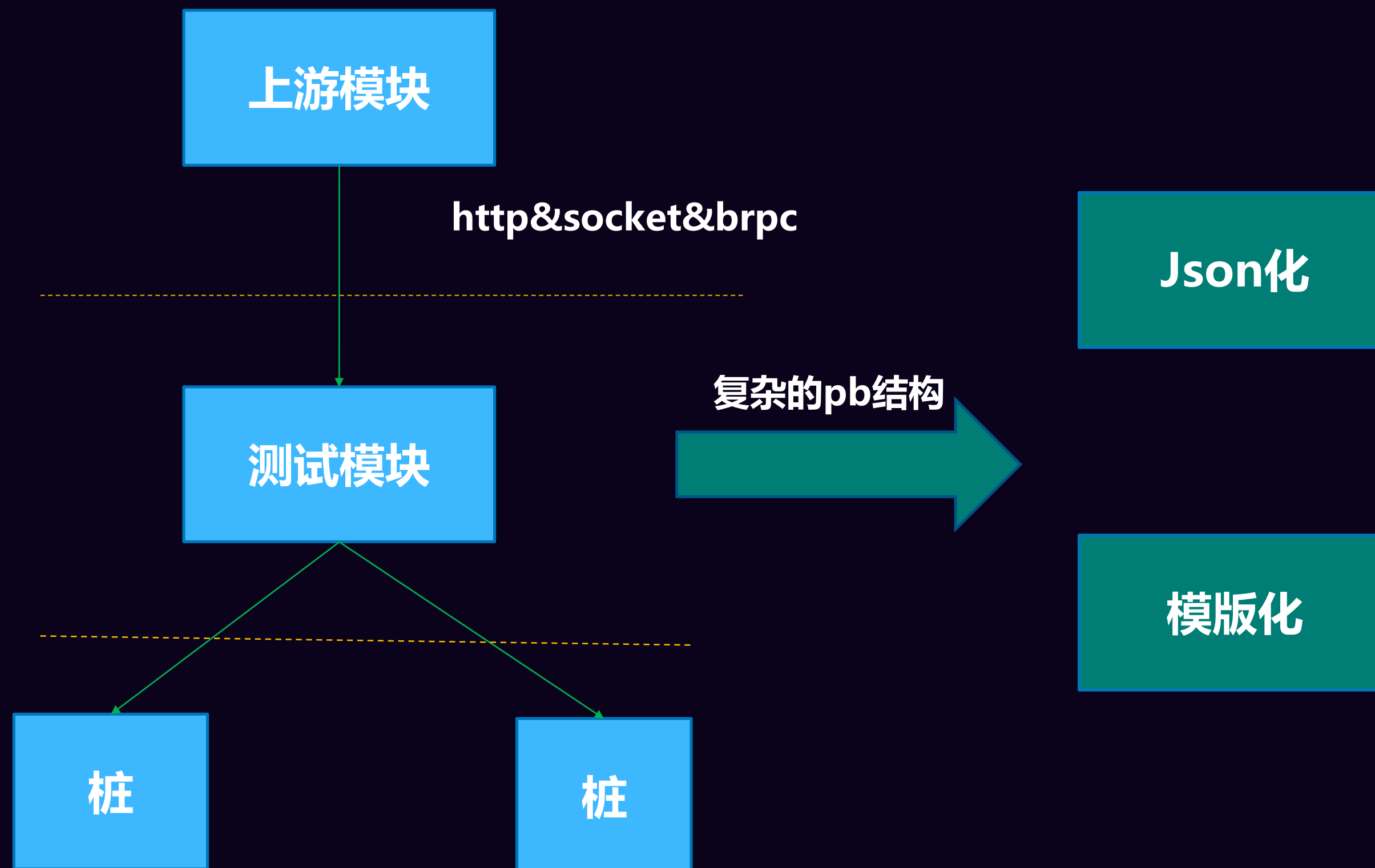
➤ 容器化

● Jenkins+镜像仓库

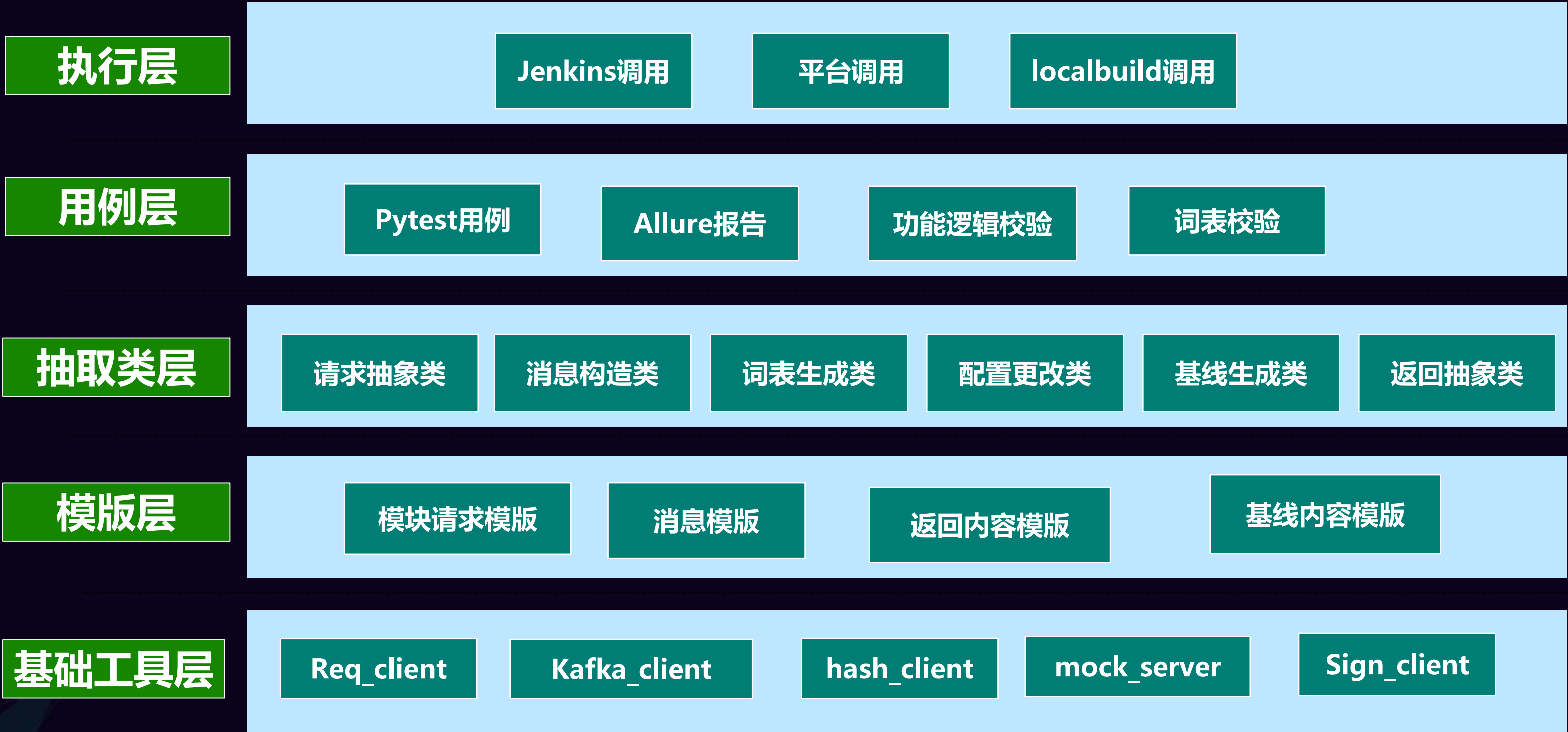




➤ 模块自动化



➤ 系统架构



➤ 模块自动化-平台化

➤ Case模版化

- 选择模版，更改生成用例

➤ 选择环境运行

- 和环境对接，
- 可切换环境运行

➤ 方便的调试和运行

- 界面调试用例
- 方便报告查看

➤ Diff测试

Http请求diff

- 覆盖全部URL请求
- 各种类型词1.5万个
- 重复执行
- 可视化报告



socket请求diff

- 定期收集线上请求
- Pb返回对比
- 覆盖所有引擎模块
- 重复执行

➤ 压测

■ tcpcopy 在线导流和离线回放

离线导流

- 定期从线上导流生成pcap包

流程化运行

- Jenkins一键式部署环境+导流

在线导流

- 实时导流量到测试机
- 增大流量压测检查

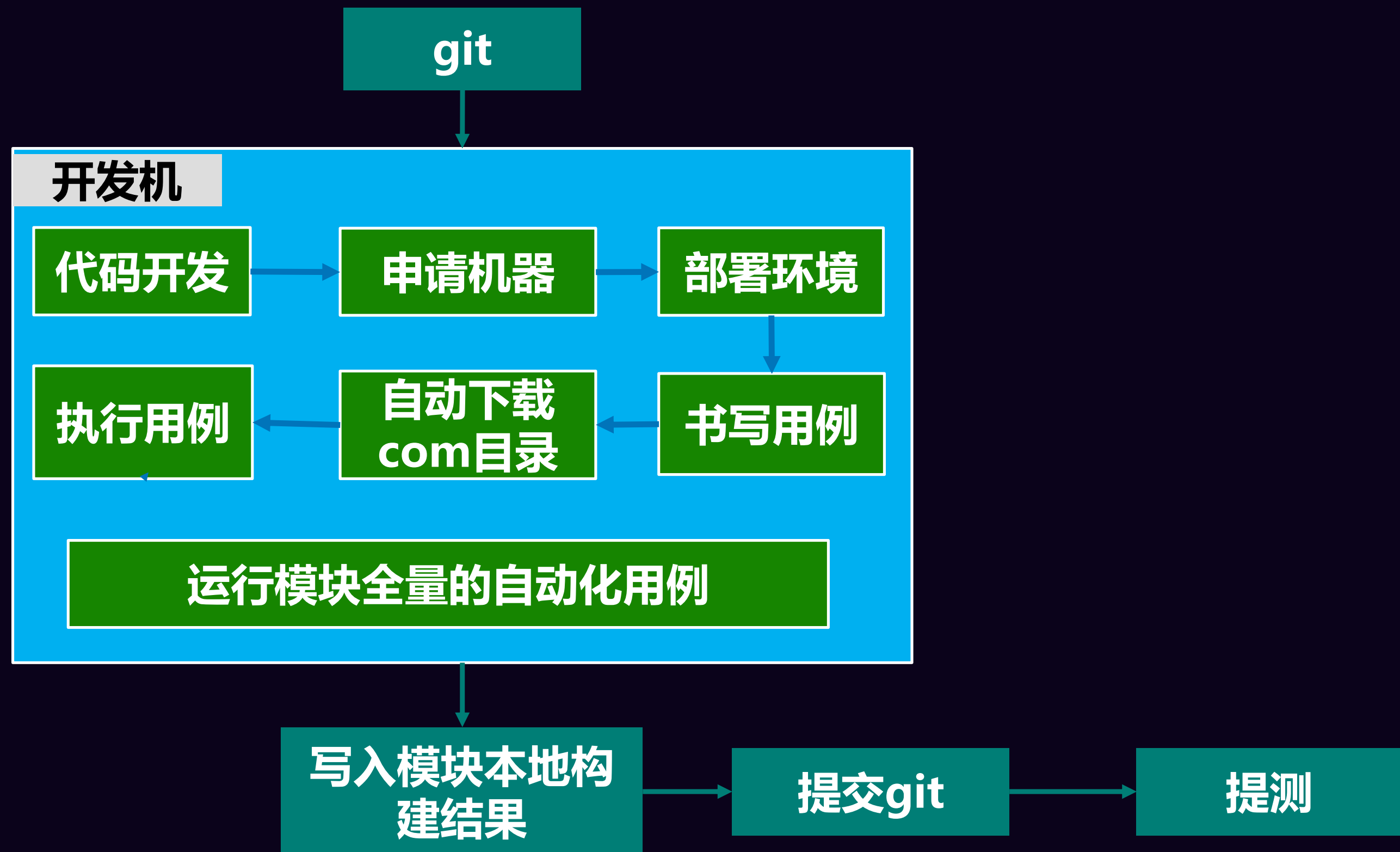
可视化报告

- 和opentsdb对接
- 实时获取压测的数据进行展示



➤ 本地构建

- 一键编译部署本地代码
- 一键运行自测用例



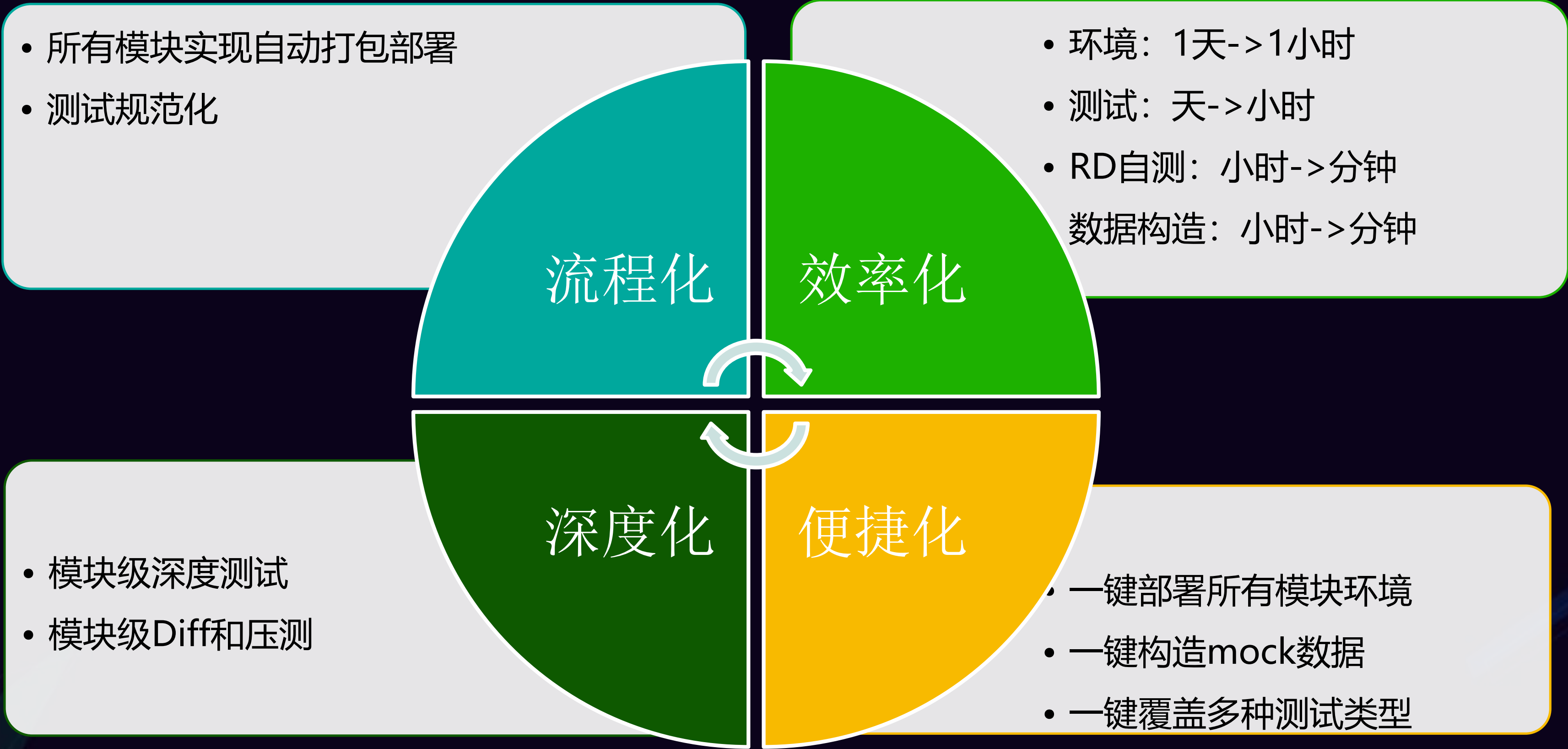
Part 3：效果介绍

全新的研发测试流程

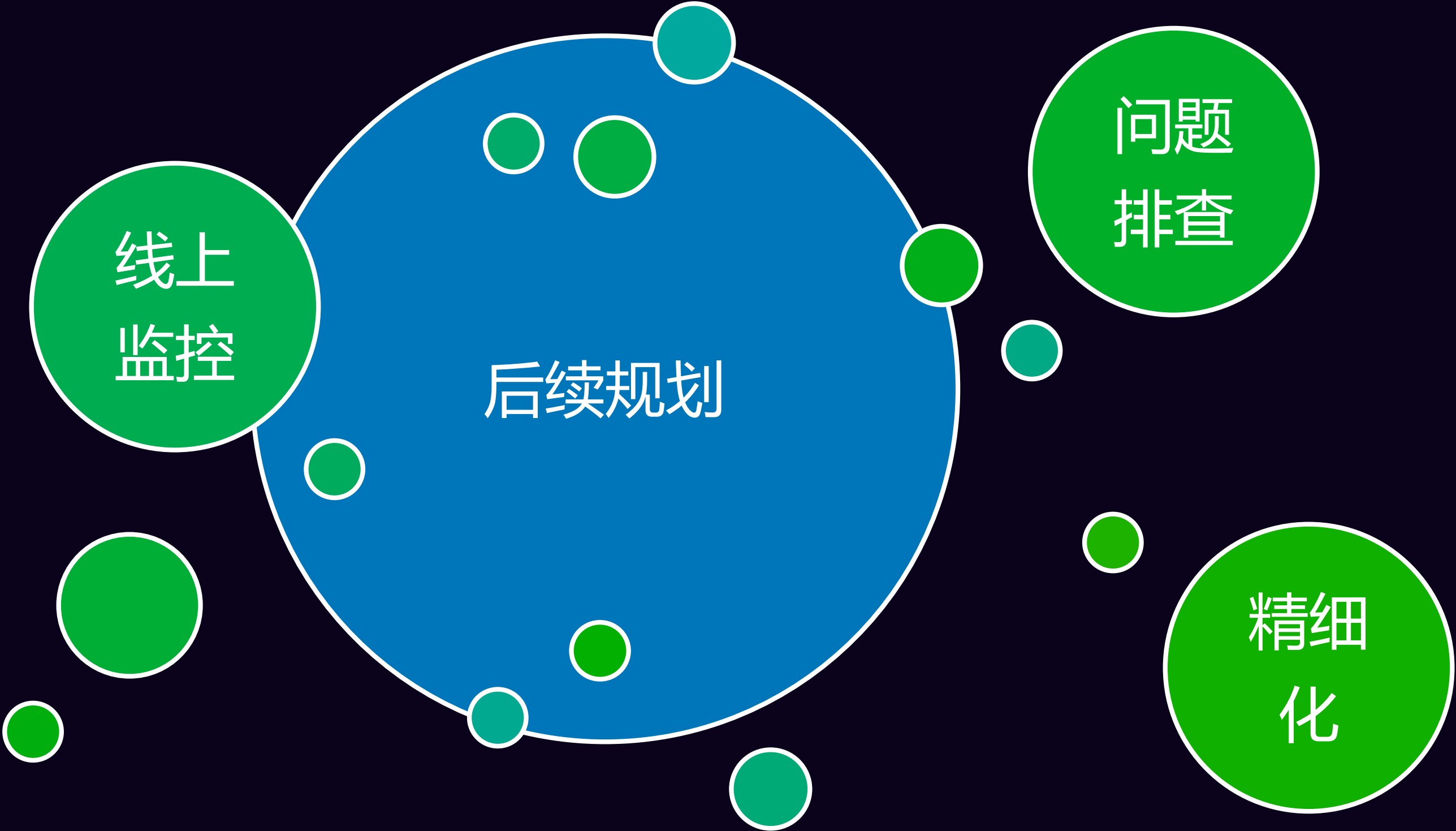
将机器申请，本地构建，自动化用例自测，自动化回归，diff，压测集合到一起，形成一套全新的研发测试流程。



➤ 100%提测上线都要经过CICD， 30%+项目开发自行完成



Part 4：未来展望



Thanks