

移动端WEB自动化测试



自我介绍

- weibo [@达峰的夏天](#)
- twitter [@2009xdf](#)
- github [@xudafeng](#)
- site [xdf.me](#)
- company alipay.com

WEB UI 自动化实践之路





2015.07 - WEB 更加工程化、全栈化

- 2015.06 - ES6 正式获批
- 2014.10 - HTML5 制定完成
- 2013.05 - React 发布
- **2009.05 - Node.js 诞生**
- 2008.09 - Chrome 发布
- 2006.01 - jQuery 发布
- 2005.02 - Ajax 概念出现

WEB 与 Native

- Hybrid App 已数见不鲜
- 丰富的 device API
- WEB 与 Native 的融合在高速地自我突破
- 定制化的 webkit
- 既要迎合技术栈的更新频率，又要迎合市场

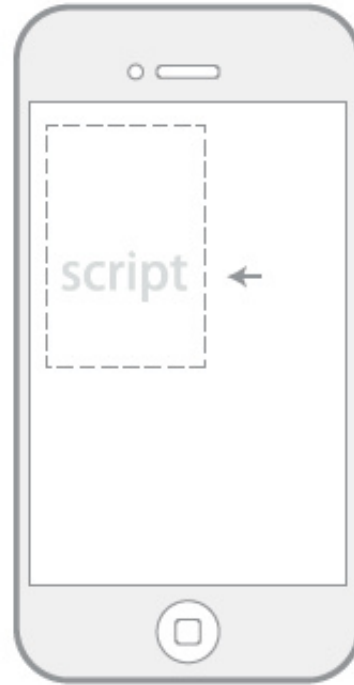
移动带来的挑战
体验要求更高
迭代频率更快
高品质保障更有力



start...

赶快“自动化”起来！

写个静态服务器就搞定。。。。



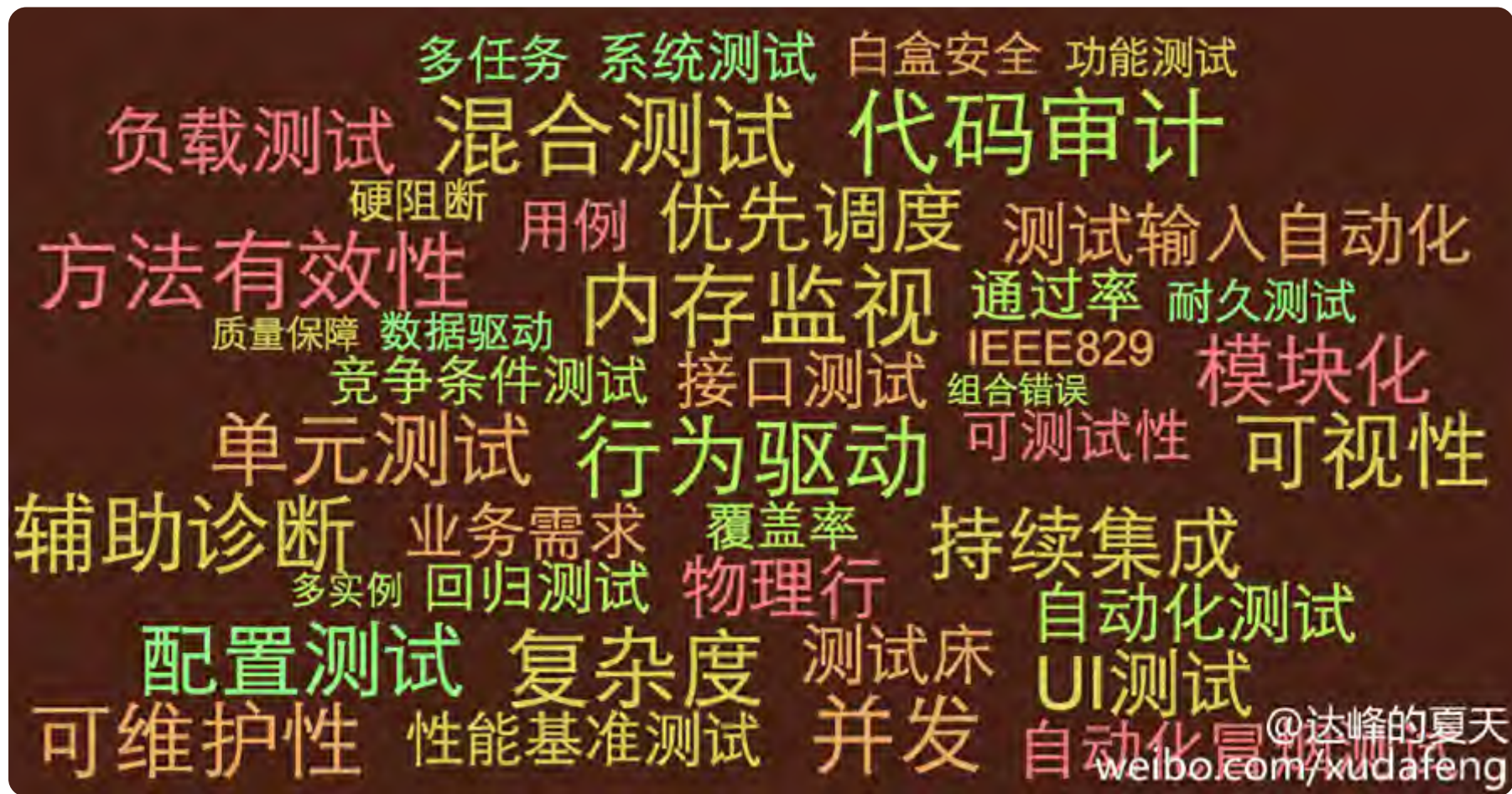
- HTTP 或网关层面实现
- 代理端注入测试套件 **testsuit**
- 代理端插入测试脚本

[anchor#1] movie

满足要求么？

- 数据环境不满足
- 权限环境不满足
- 环境因素导致的上下文限制
- 代码侵入 APP

自动化测试本质是软件开发



首要考虑

- 界面抽象粒度
- 用例的可维护性
- 标准的脚本语言

自动化的目的

- 耐久性测试
- 性能基准测试

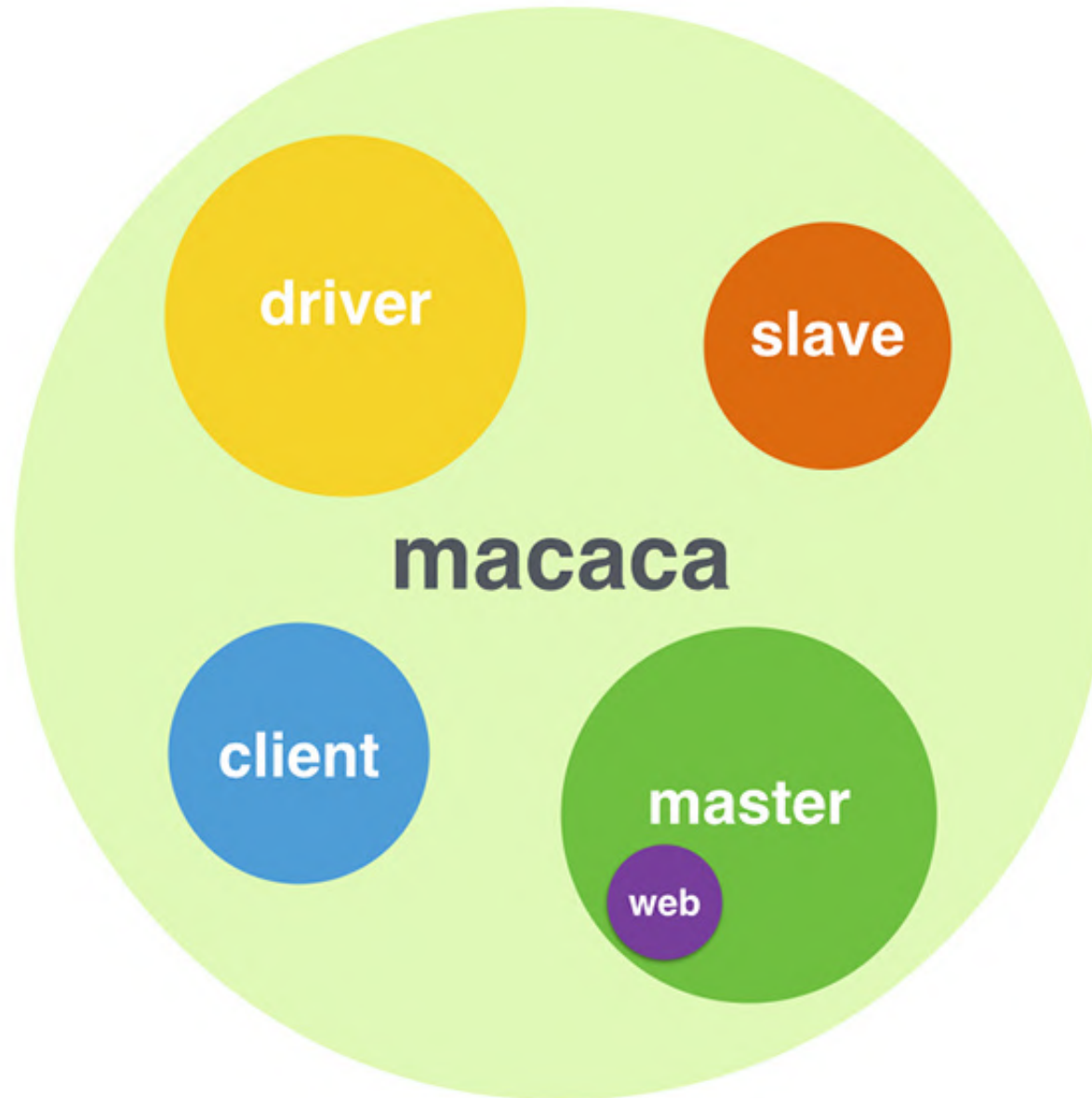
我们还需要解决什么问题？

版本差异

- 操作系统版本不统一
- 多个 SDK 版本共存
- 软件包、WEB 容器多版本
- APP 自身的版本

环境和配置差异

- 多套部署环境



macaca-driver

- WebDriver Wire Protocol
- vs headless-webkit

Nodejs 成为主要的技术选型

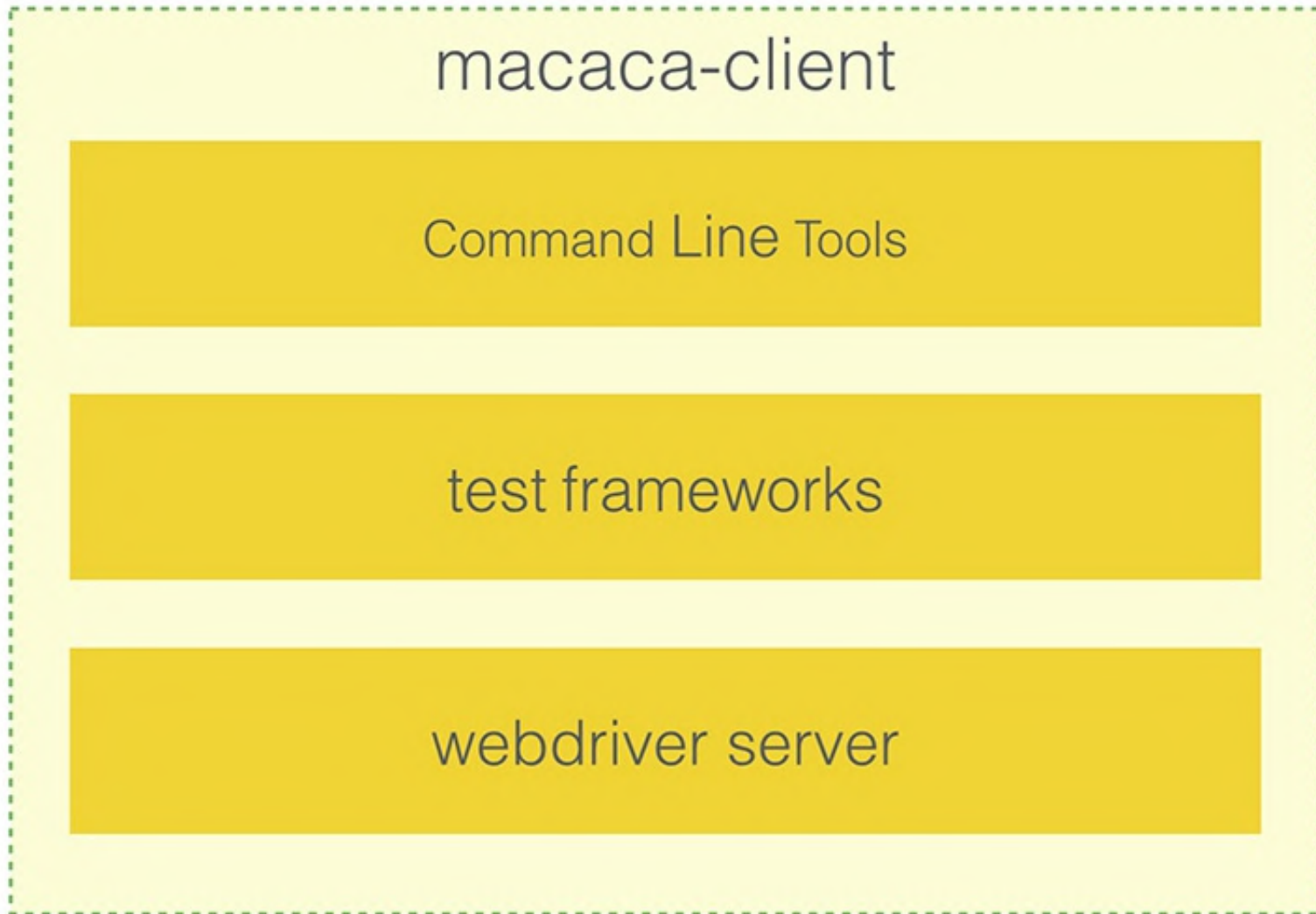
- 跨平台的 WEB 实时应用首选
- 事件驱动与非阻塞 I/O 模型使其轻量、高效
- 繁荣和极速膨胀的生态 NPM

使用者端

- 本地命令行工具
- WEB 操作平台

macaca-client

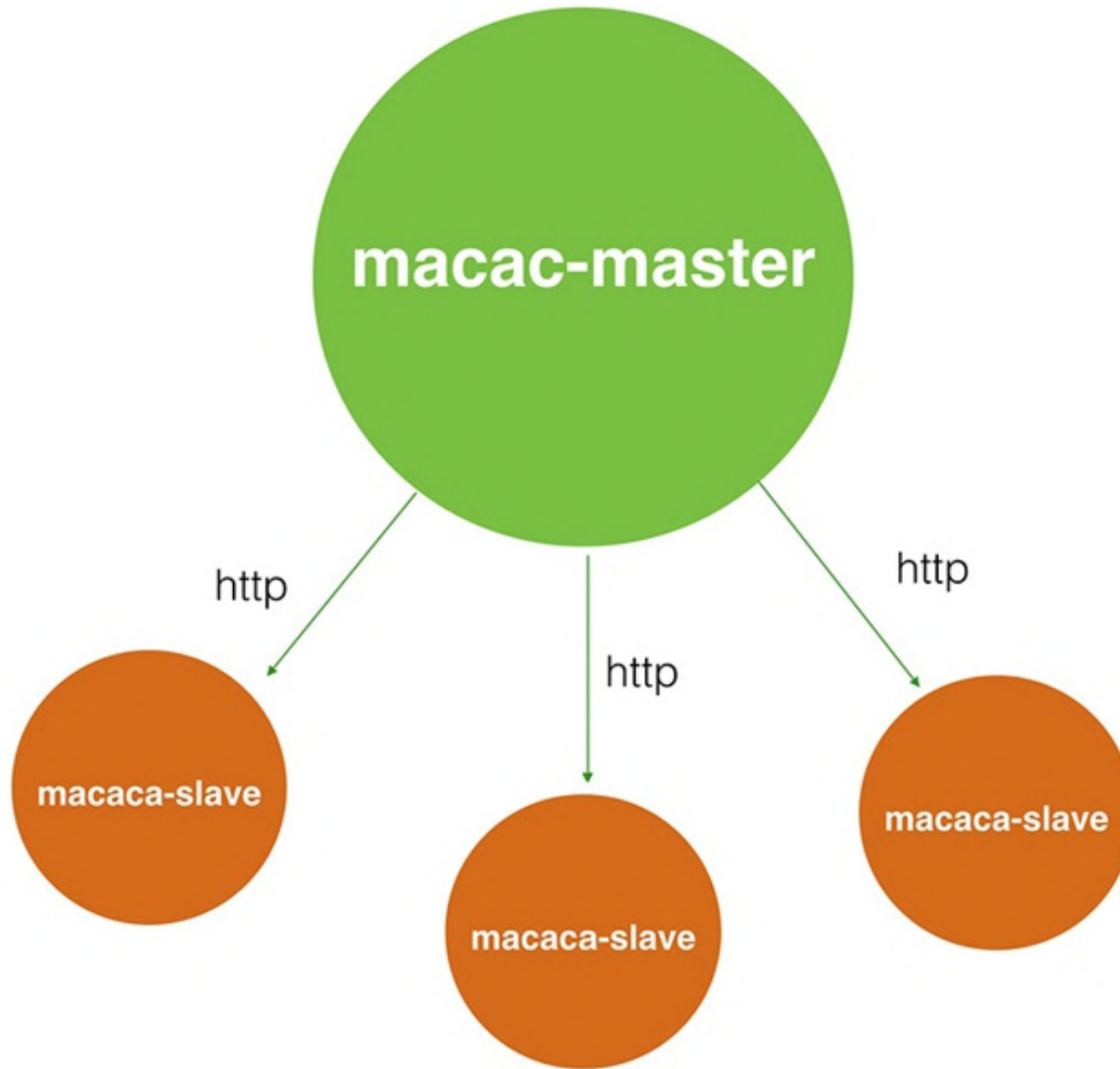
- 一体化的测试体验
- 可定制化和可扩展的 API
- 消灭配置



[anchor#2] movie

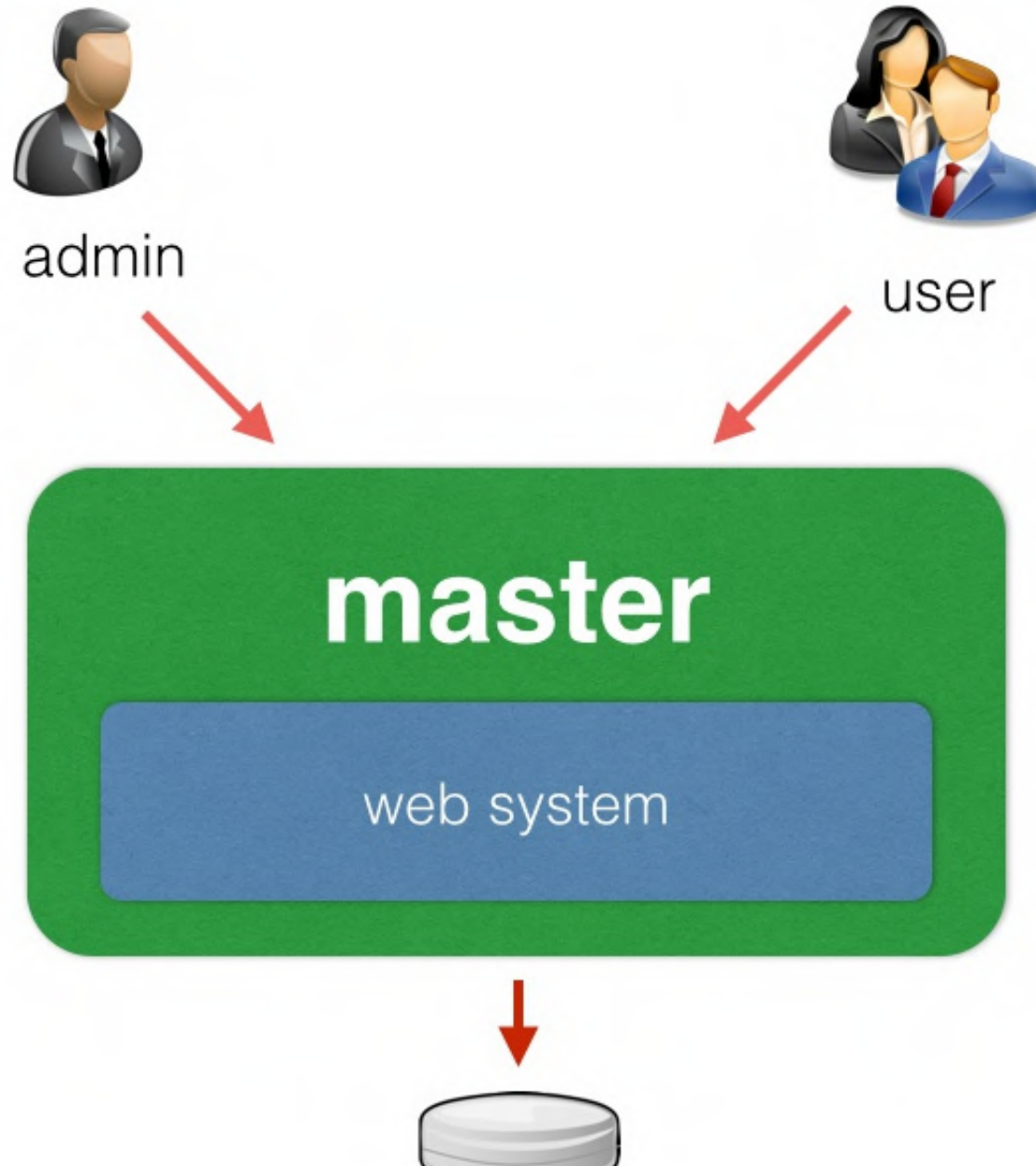
Then?

Continuous integration



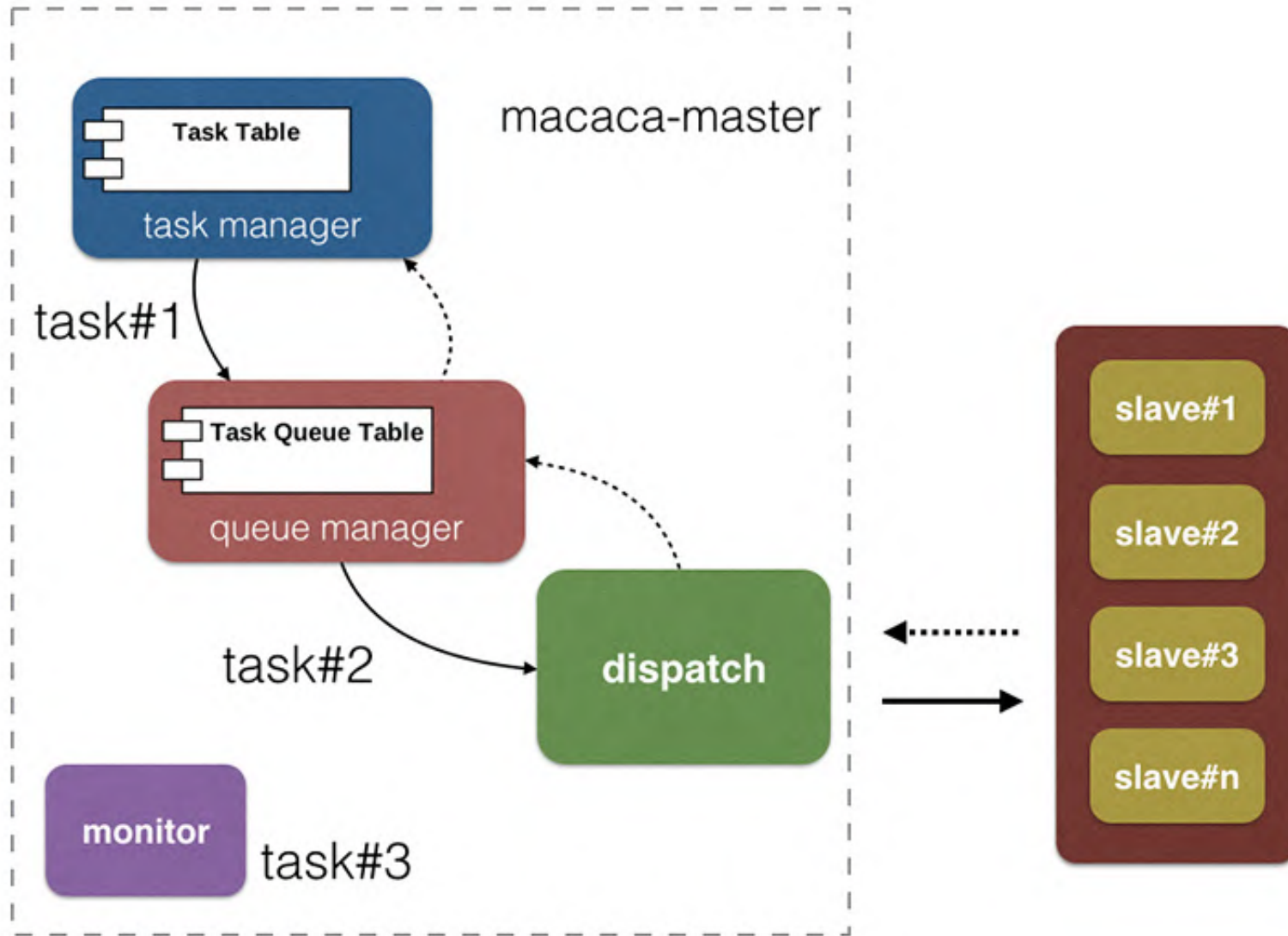
macaca-master

- 并发任务
- slave管理管理
- WEB平台集成
- 沉淀数据、报表





- 实现进程控制
- 处理数据存档
- 实现任务调度
- 调度策略



macaca-slave

- 接收资源申请
- 配置管理
- 测试框架集成
- 自身socket通信

- slave状态monitor
- slave条件负载

功能丰富的ci系统

- GitLab-ci
- Jenkins
- Gerrit
- strider
- travis-ci

Unattended Execution

- 如何监控





至少看上去“全自动”了

一点经验

- 实践认识论 \geq 方法论
- 可测试性高于任何框架、工具

Open Source?

- Testerhome.com

all passing (20sec)

2015.07.11