

分布式监控系统的容器化变迁与 CI/CD 实践



光载无限 海外研发中心

欧曜玮
Kordan Ou

www.wifire.net

Contents

01

分布式监控系统

02

微服务的
运维难题

03

持续集成
的实践

04

回顾与总结

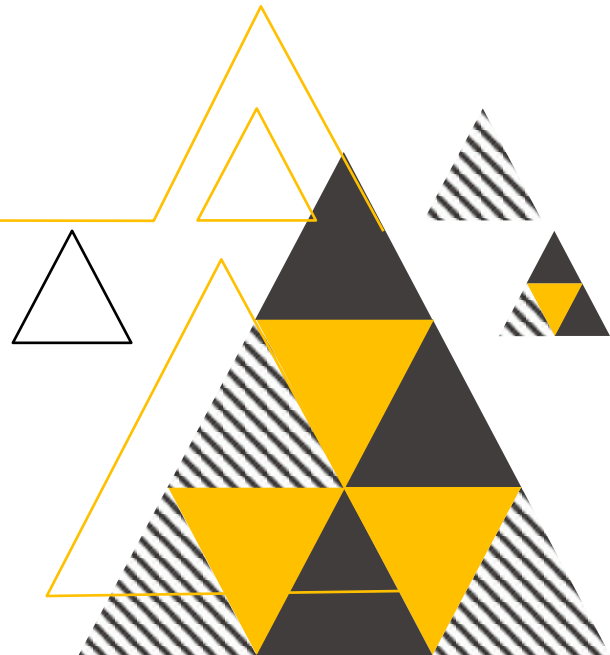


微服务架构的优缺点

优点	缺点
<ul style="list-style-type: none">➤ 易于横向扩展➤ 容错机制易实现➤ 技术选型灵活➤ 独立部署➤ 复杂度可控	<ul style="list-style-type: none">➤ DevOps 技术栈➤ 运维成本➤ 接口相依与版本控管➤ 分布式系统的复杂性

A decorative arrow pointing right, composed of a black triangle, a yellow triangle, and a grey triangle with diagonal stripes.

分布式监控系统





Open-Falcon 介绍 (1/2)



- Open-Falcon 是一款企业级、高可用、可扩展的开源监控解决方案
 - 让运维标准化的理念可以在不同公司之间达成一致，进一步推动和提升国内运维行业的自动化水平。
- 国内已有上百家公司在重度使用
 - 包括小米、滴滴、美团、金山云、快网（光载无限）、迅雷、360、平安云、七牛、赶集、又拍云、链家网、一点资讯、爱奇艺、汽车之家、微博等
- 选择了 Open-Falcon，不仅仅是选择了一个监控系统，而是选择了一个「运维的生态」



falcon





Open-Falcon 介绍 (2/2)

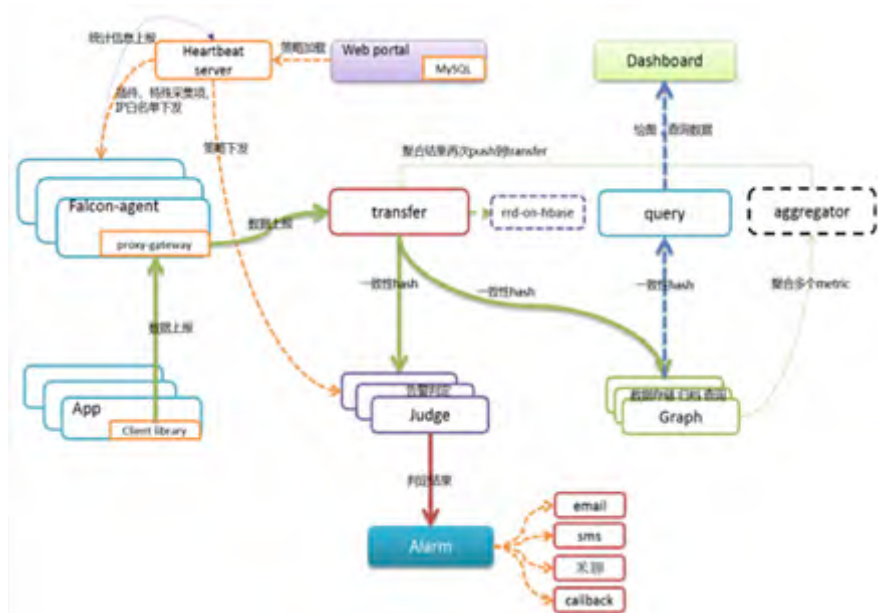




Open-Falcon 架构图



- 模块多，开发复杂度稍高
- 系统不易分发，部署难度高
- 安全性考量不到位
- 没有通盘考虑的 API 权限设计
- 每个 Graph 实例均是单点、扩容有损
- 上下游组件无动态管理机制
- 报警没有入库、现场没有保存



<http://www.jianshu.com/p/623b96175915>

微服务面临的普遍问题



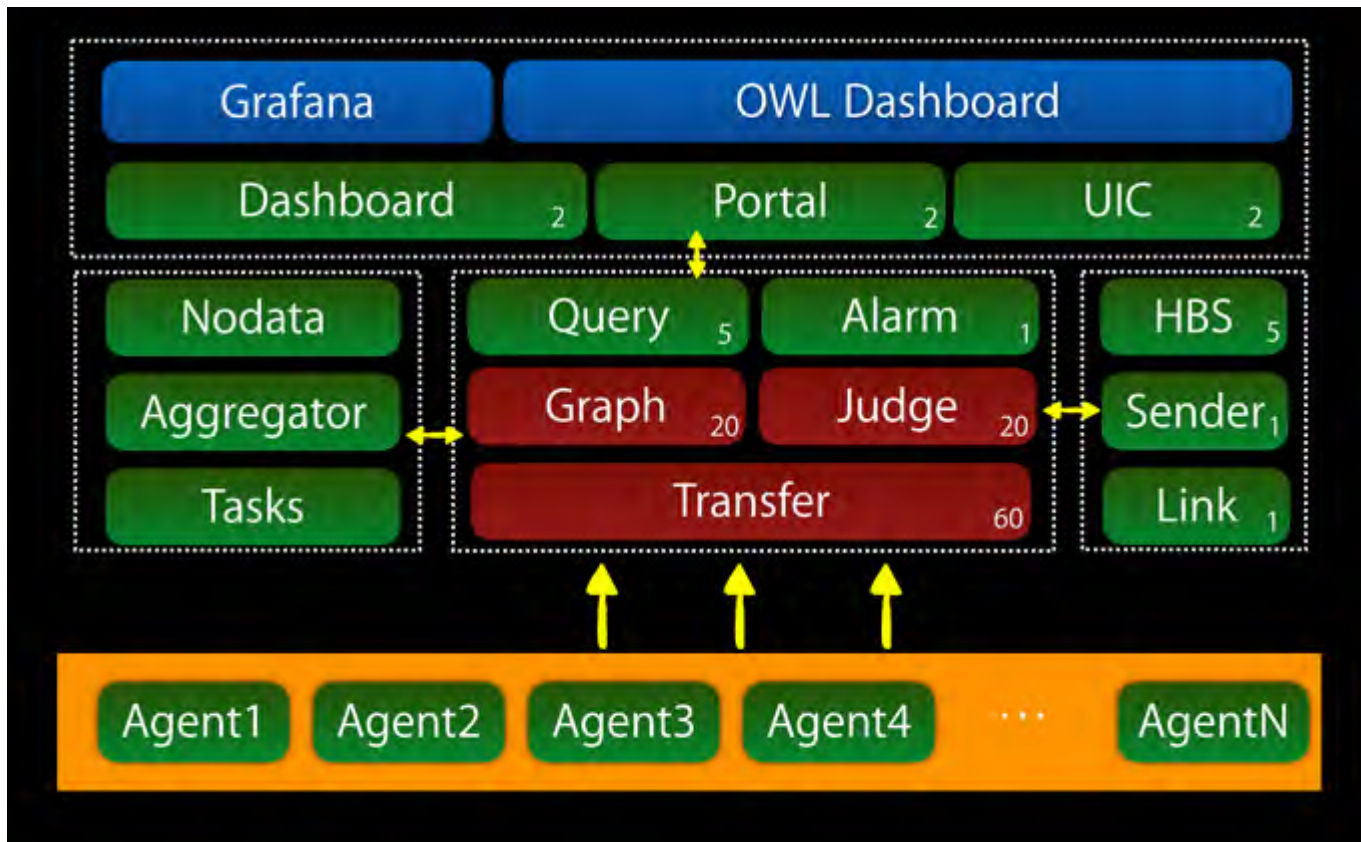
微服务

微服务 (Microservices) 是一种软件架构，它是以专注于单一责任与功能的小型功能区块 (Small Building Blocks) 为基础，利用模块化的方式组合出复杂的大型应用程式，各功能区块使用与语言的 API 相互通讯。



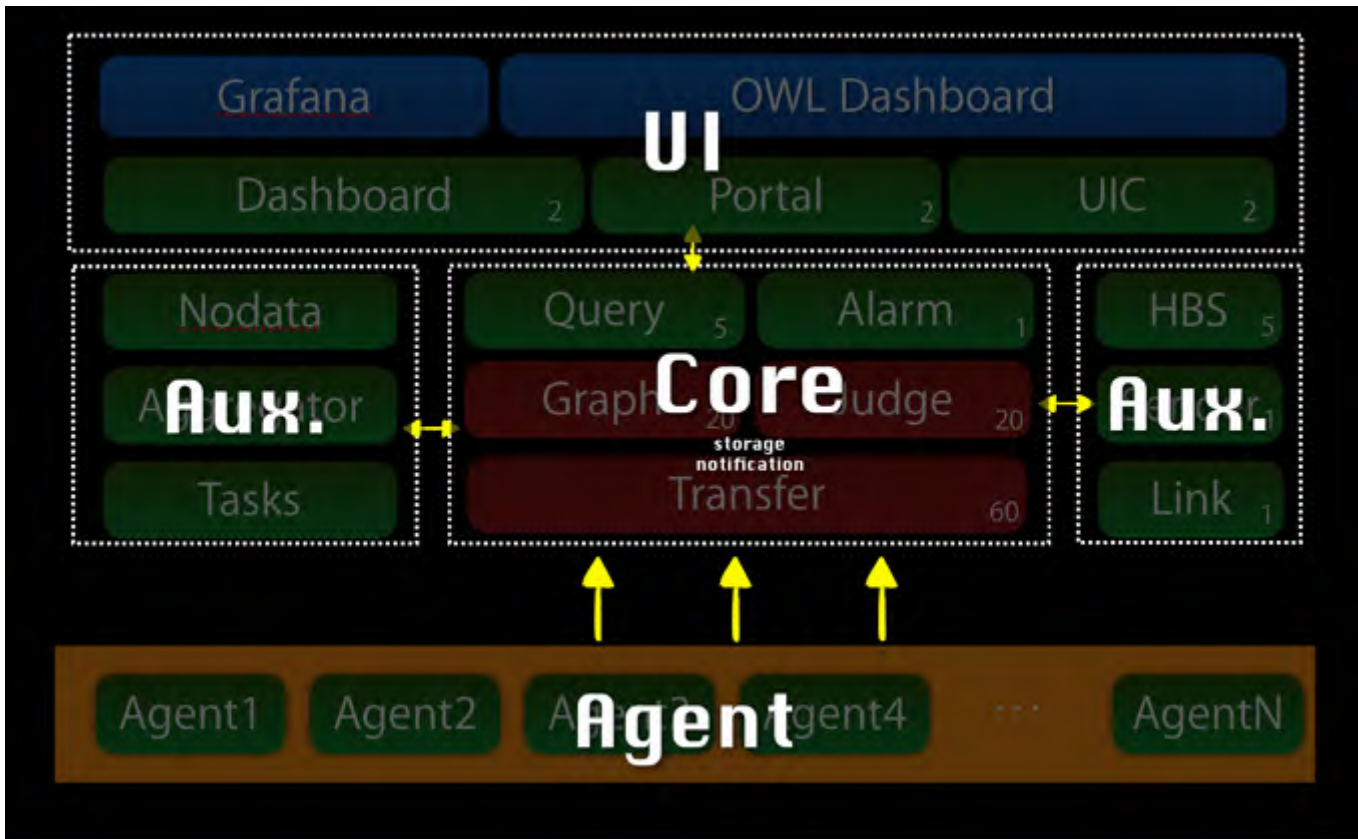


分层后的系统架构 (1/2)





分层后的系统架构 (2/2)



持续集成的实践：基于容器的微服务架构设计



Docker 的优势

- 可移植性封装依赖，标准统一的打包/部署/运行方案
- 简单、易用、活跃的社区
- Infrastructure as Code
- 容器之间互相隔离，混合部署解决依赖复杂度
- 相比传统虚拟化速度，有更好的性能，尤其是内存和 IO 的开销
- 轻量，启动快速，最小化停机时间
 - 系统秒级回滚
 - 热备份
 - 镜像分层技术
- 易于构建，良好的 RESTful API，也很适合自动化测试和持续集成



微服务容器化





Docker 目前的局限

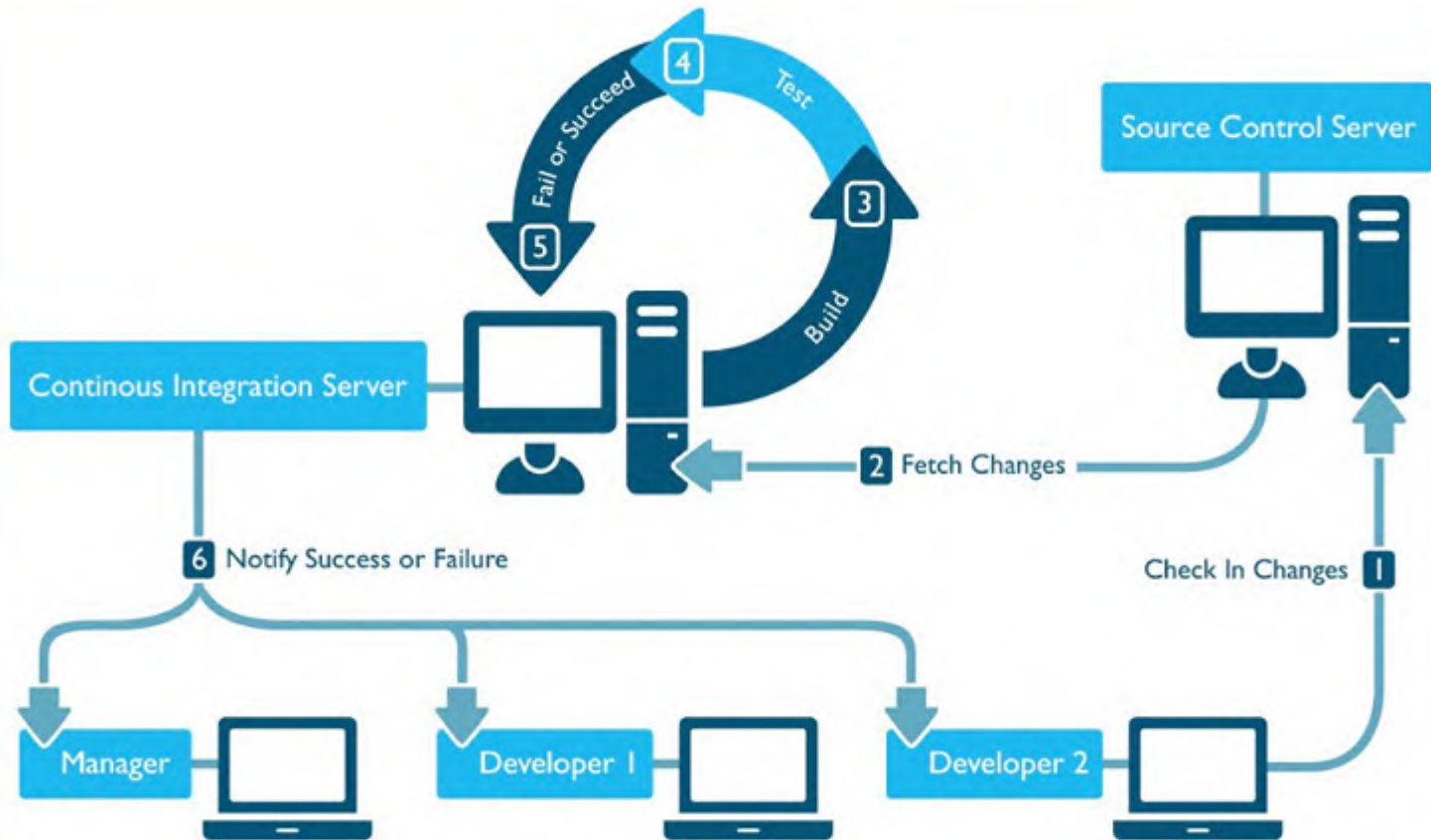


- 系统/内核层面的隔离性
- ~~缺乏成熟的集群管理~~
- 业务无感知的升级
 - Docker daemon live upgrade





持续集成 (1/2)





持续集成 (2/2)

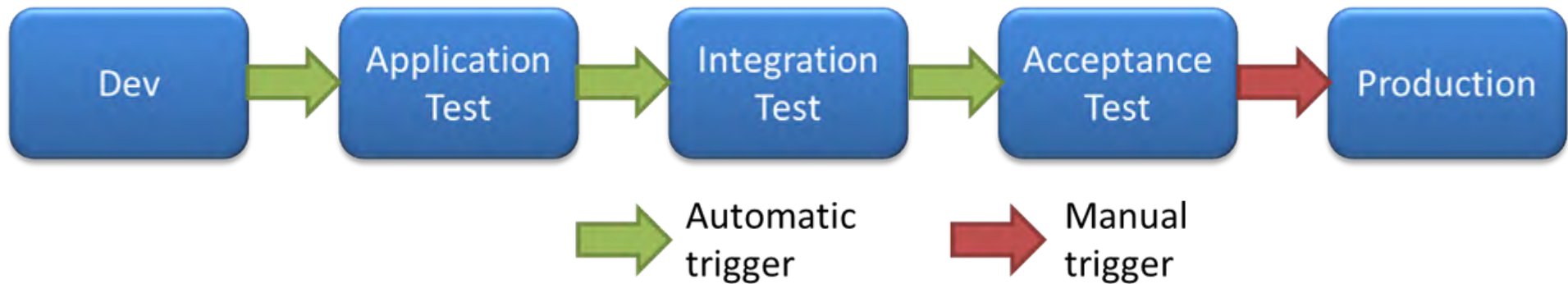




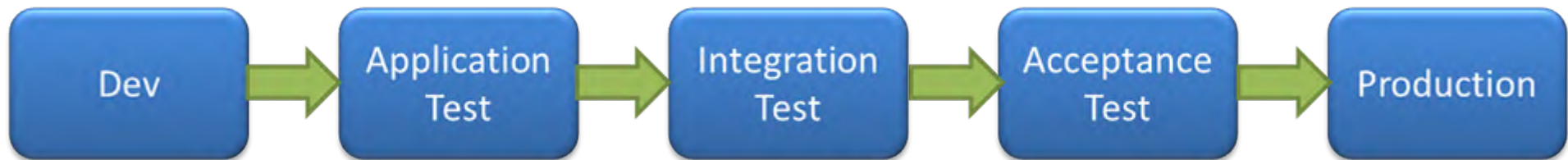
CI 与 CD 的不同



Continuous Delivery

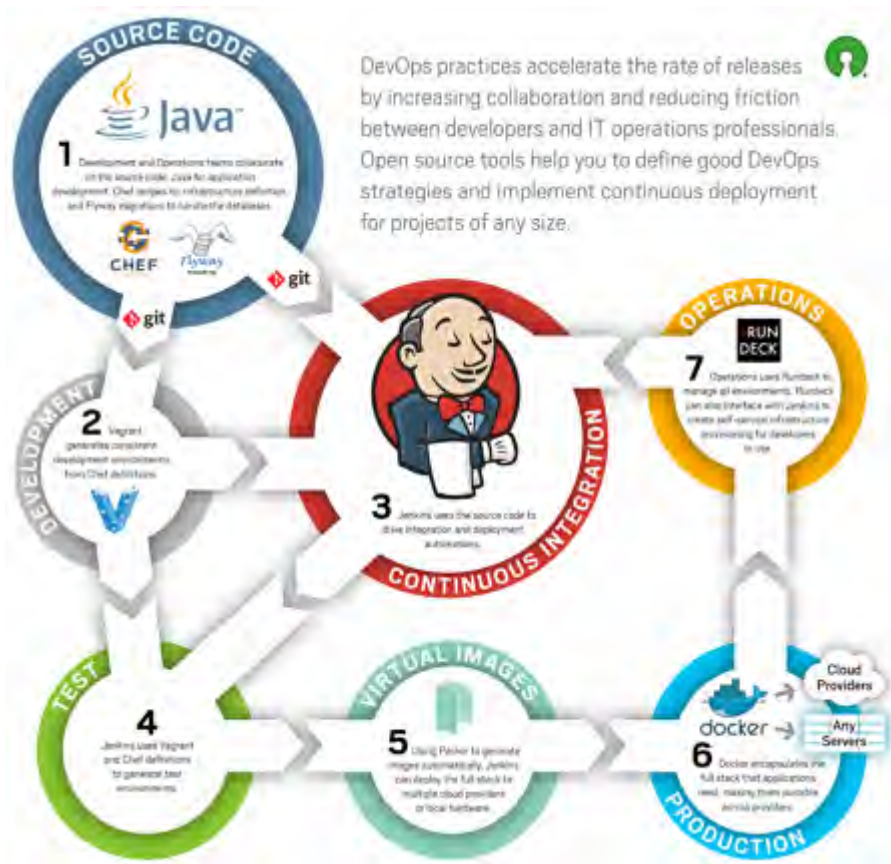


Continuous Deployment





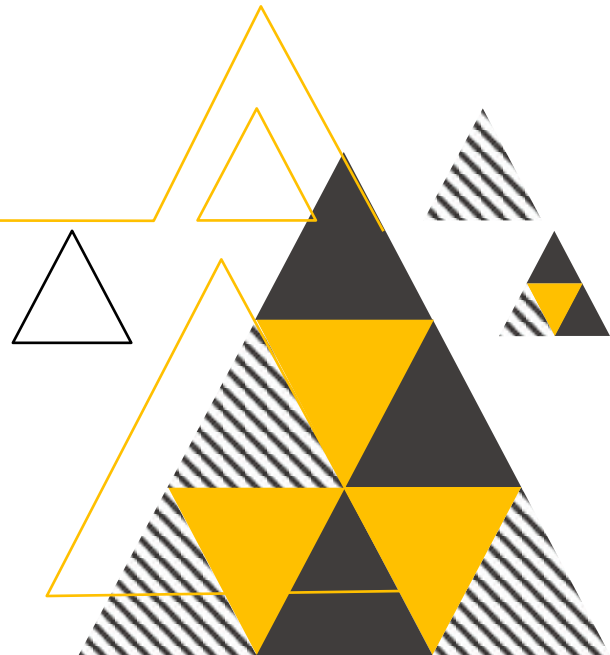
以 Jenkins 驱动的 CI/CD



常见的误区：
持续集成就是
用上 Jenkins

A decorative arrow pointing right, composed of a black triangle, a yellow triangle, and a grey triangle with diagonal stripes.

回顾与总结





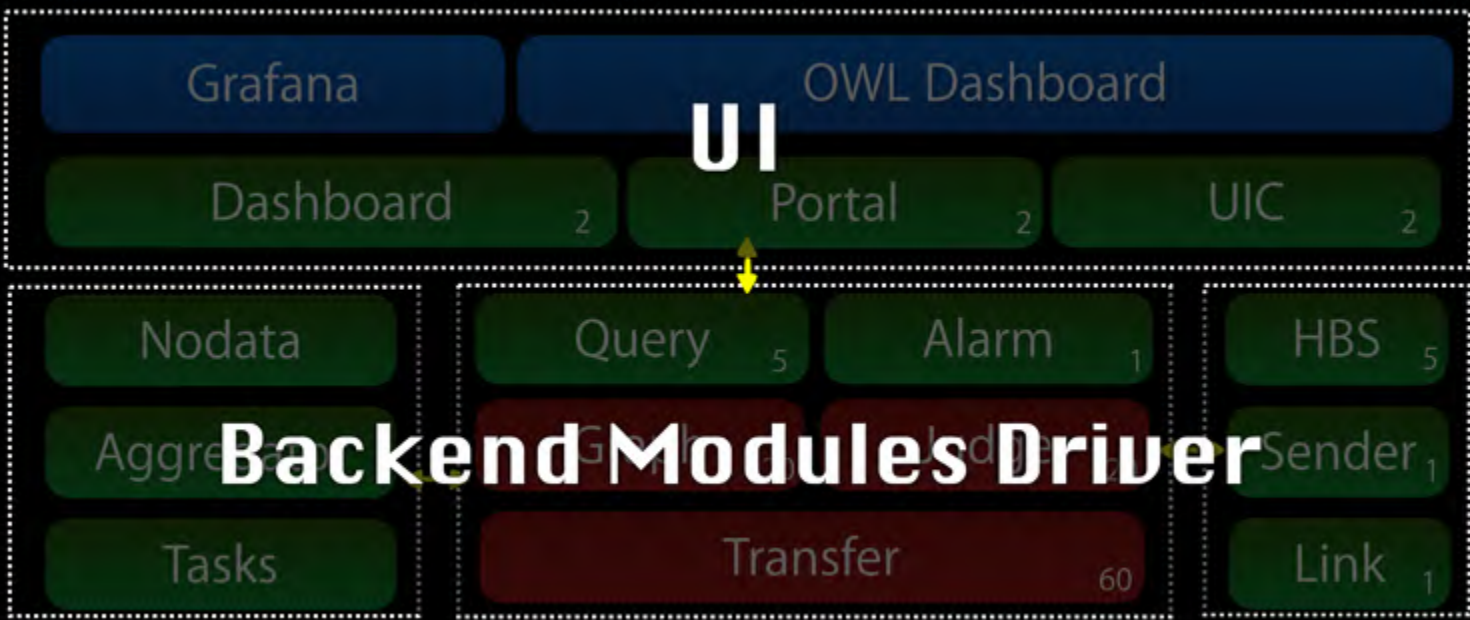
回顾与总结 (1/3)



Before	Now
<ul style="list-style-type: none">• 模块多，开发复杂度稍高• 系统不易分发，部署难度高• 安全性考量不到位• 没有通盘考虑的 API 权限设计• 每个 Graph 实例均是单点、扩容有损• 上下游组件无动态管理机制• 报警没有入库、现场没有保存	<ul style="list-style-type: none">• 后端代码仓库统一• Docker-Compose, Docker-Swarm• 网络边界隔离• 加入没有通盘考虑的 API 权限设计• Graph 双写• Volume 容器统一配置管理• 报警事件入库



回顾与总结 (2/3)

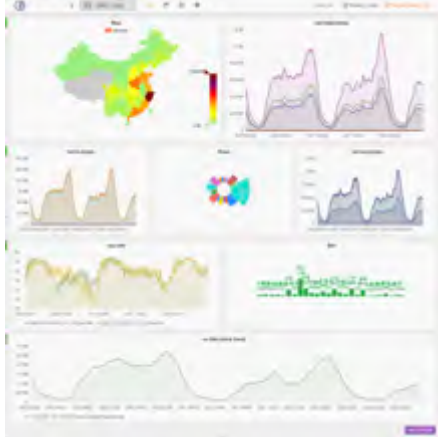
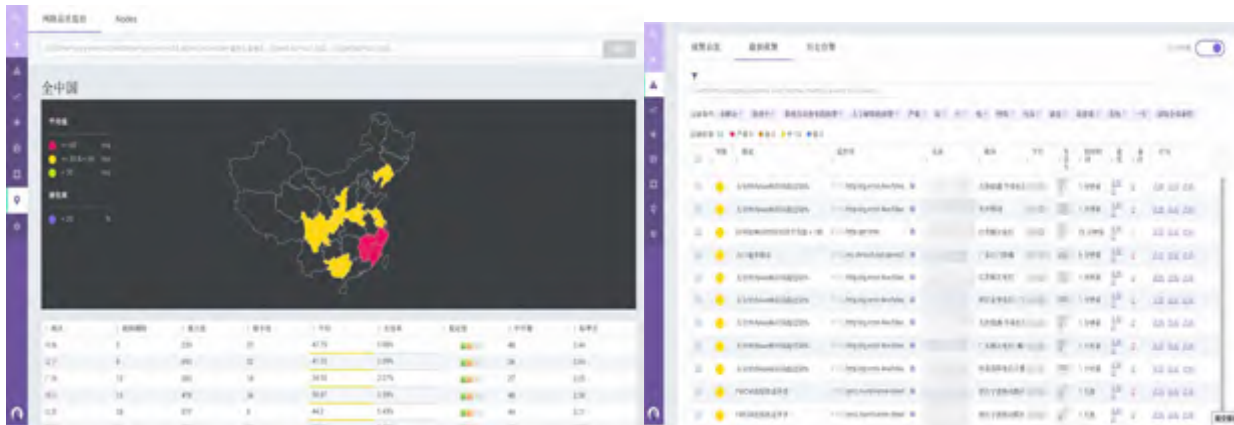


Docker





回顾与总结 (3/3)





Thanks

光载无限（北京）科技有限公司

DevOpsDays 即将首次登陆中国



DevOps 之父 Patrick Debois 与您相约
DevOpsDays 北京站 2017年3月18日



门票早鸟价仅限前100名，请从速哟

<http://2017-beijing.devopsdayschina.org/>





想第一时间看到
高效运维社区公众号
的好文章吗？

请打开高效运维社区公众号，点击右上角小人，如右侧所示设置就好





Thanks

高效运维社区
开放运维联盟

荣誉出品

