

Brought by **Geekbang**  **InfoQ**
极客邦科技



Connect Container Community

全球容器技术大会

剖析容器企业实践 关注容器生态圈开源项目



分布式平台微服务架构演化实践

郭峰

联合创始人@DaoCloud

README.md

应用交付的难题

微服务的辩证法

系统架构演化实践

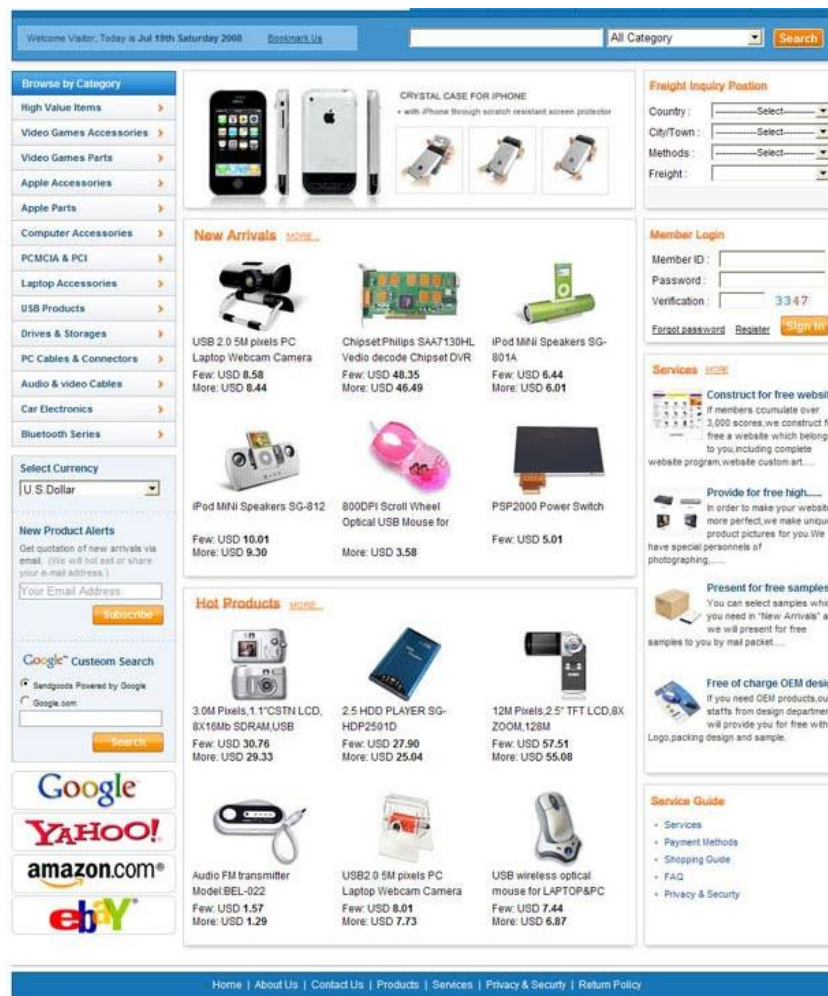
微服务的持续交付

总结

互联网速度

•迭代 1: 网站框架

•迭代 2: 用户管理





迭代.N

订单

用户

产品

支付

展示

库存

购物车

报表

推荐

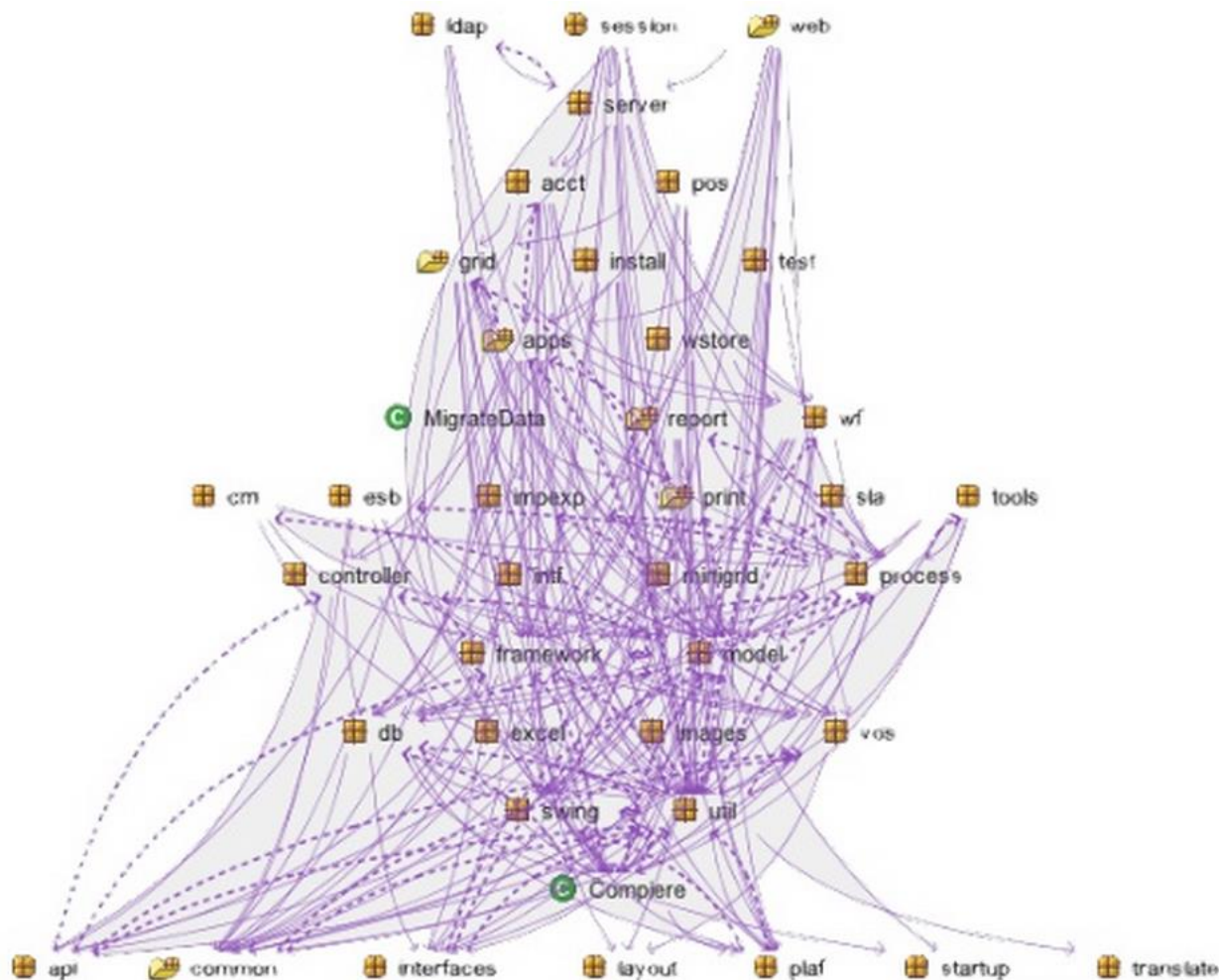
供应链

监控

评论

开发壁垒

```
▼ > daoci [daocloud-daoci feature/support_
  ► bin
  ► pkg
  ▼ > src
    ► bitbucket.org
    ► code.google.com
    ▼ > github.com
      ► andybons
      ► Bugagazavr
      ► BurntSushi
      ► codegangsta
      ► daaku
      ▼ > DaoCloud
        ► > DaoCI
        ► dgrijalva
        ► docker
        ► drone
        ► franela
        ► garyburd
        ► GeertJohan
        ► go-sql-driver
        ► gogits
        ► goji
        ► google
        ► gorilla
        ► lib
        ► mattn
        ► pelletier
        ► russross
        ► Sirupsen
        ► stretchr
        ► thoj
        ► zenazn
```



不可能的快速迭代

- 小功能点需要大更新
- 完整QA周期下的“慢速迭代”
- 中断后台运行job
- 失败风险



超负荷IDE



The background of the slide is a photograph of a gymnasium. A basketball hoop is visible on the left side. On the wall, there is large, bold text that reads 'THE OFFICIAL GUIDE TO AN AWESOME LOCK-IN'. The text is white and black, with 'LOCK-IN' being the most prominent part.

THE OFFICIAL GUIDE TO AN AWESOME LOCK-IN

被某个技术栈长期绑定

微服务架构

*Loosely coupled service oriented architecture
with **bounded contexts***

1990s and earlier

Pre-SOA (monolithic)

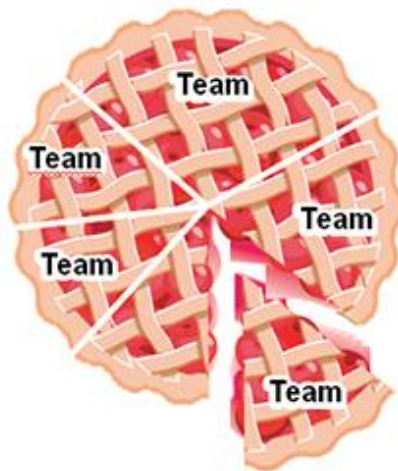
Tight coupling



2000s

Traditional SOA

Looser coupling



2010s

Microservices

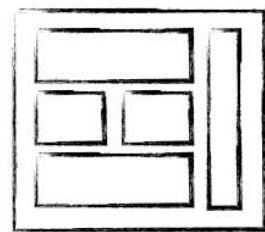
Decoupled



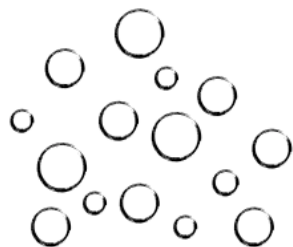
Ref: [Giant Swarm](#)

微服务架构优势

- 化繁为简
- 用最合适的技术实现
- 主人翁精神
- 升级 / 替换模块变得更简单
- 独立部署，减少沟通成本



MONOLITHIC/LAYERED

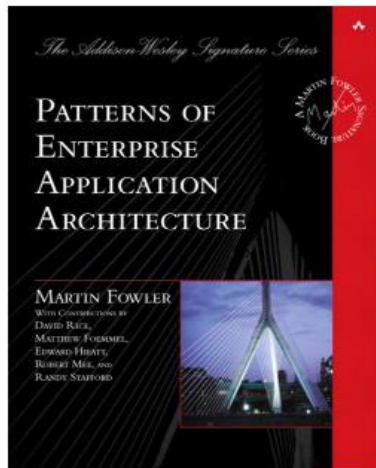


MICRO SERVICES

微服务的抉择

“don't even consider microservices unless you have a system that's too complex to manage as a monolith.”

-- Martin Fowler



Kent Beck
@KentBeck

 Follow

any decent answer to an interesting question begins, "it depends..."

1:45 AM - 7 May 2015



462

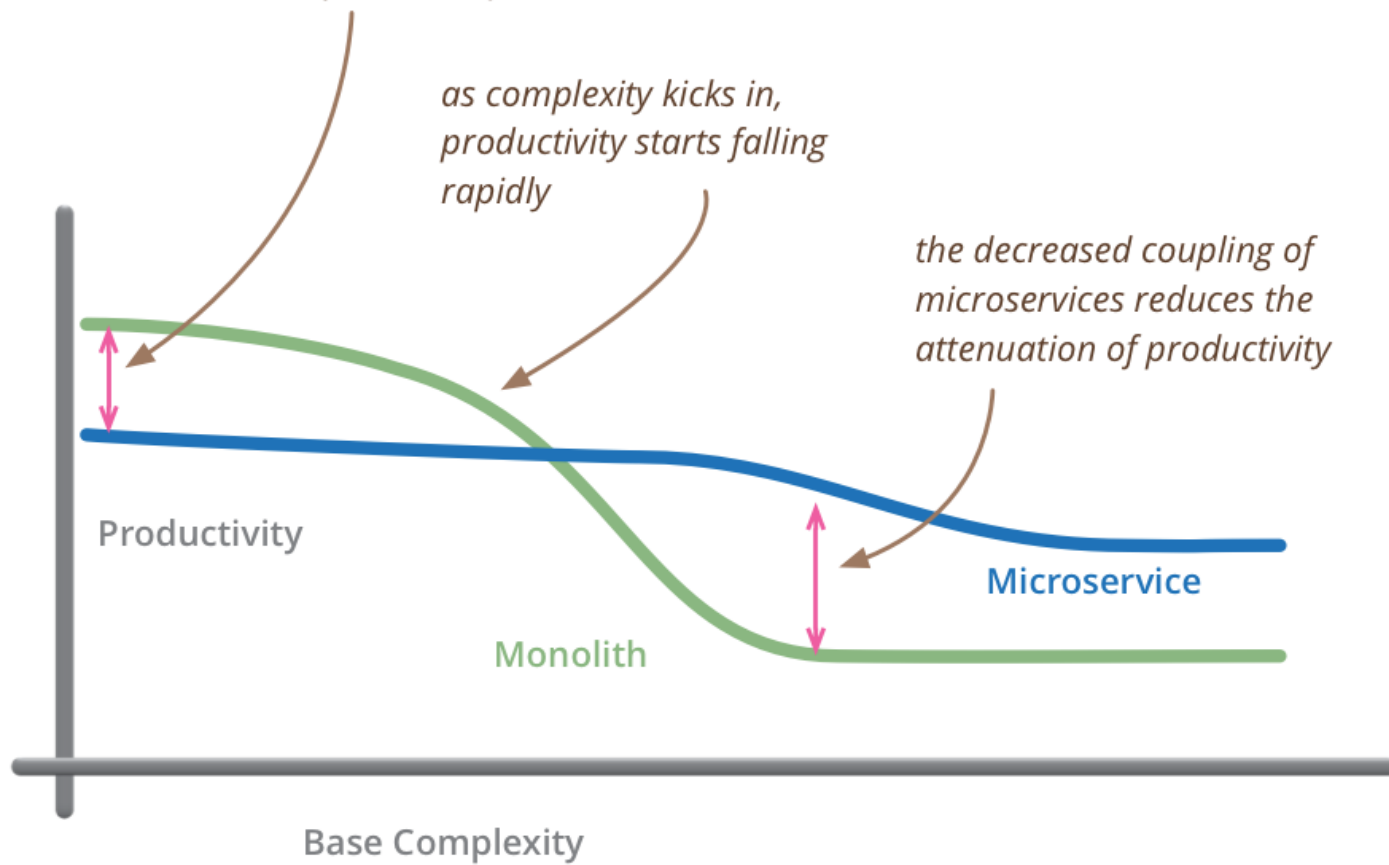


280

for less-complex systems, the extra baggage required to manage microservices reduces productivity

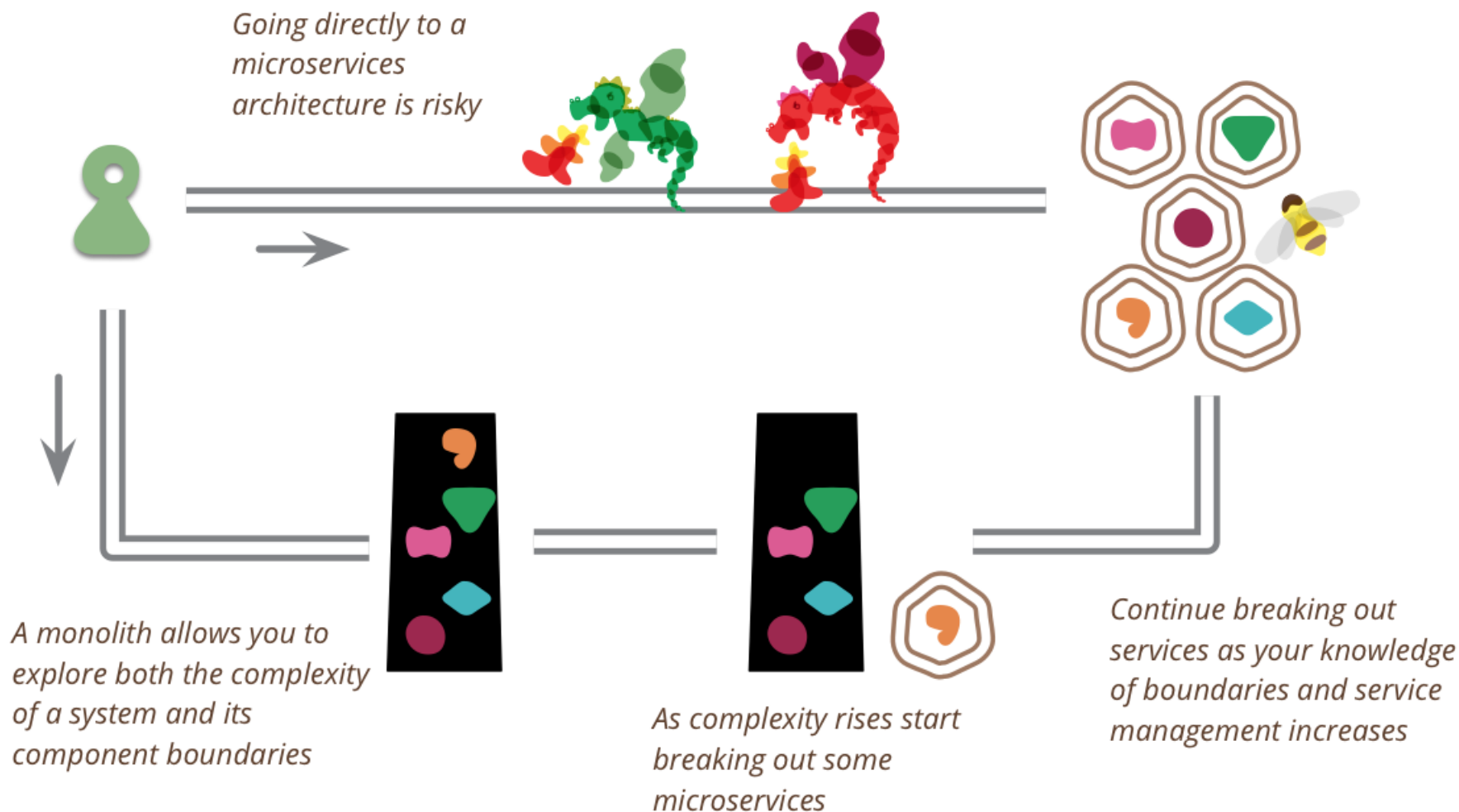
as complexity kicks in, productivity starts falling rapidly

the decreased coupling of microservices reduces the attenuation of productivity



Ref: [Microservice Premium](#)

需求驱动的微服务化



微服务拆分驱动力

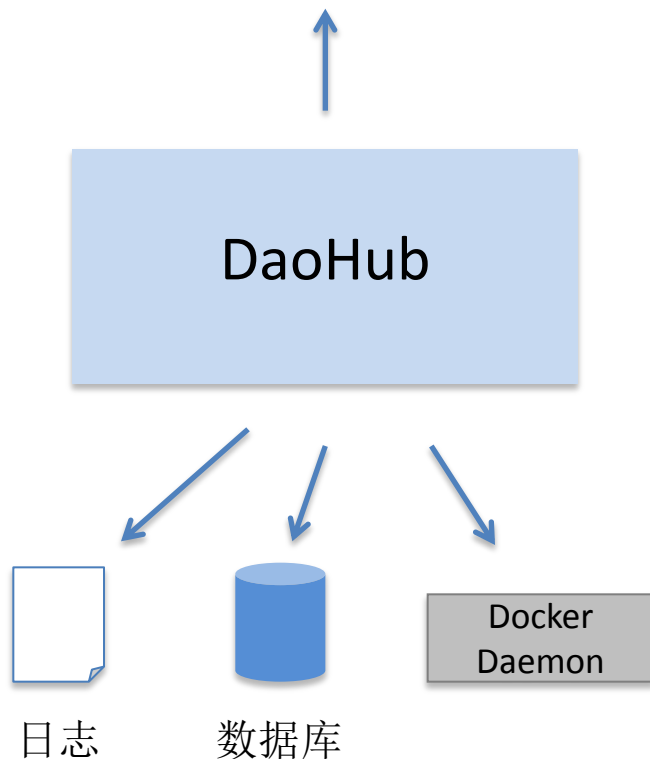
- 快速迭代
 - 解耦服务的独立更新
- 独立扩展
 - 三层架构
- 技术栈
 - 最合适的技术
 - 团队能力
- 组织结构
 - Conway法则
 - 团队和服务对应
- 特殊的考虑因素
 - 数据安全
 - 审计
 - 合规



DaoCloud 分布式构建系统 DaoHub

- Docker 应用项目的镜像构建和持续集成
 - 全容器化解决方案
 - 缓存服务
 - 镜像同步
- 分布于全球 4 个数据中心
- 由 6 个微服务组成

微服务演化之路

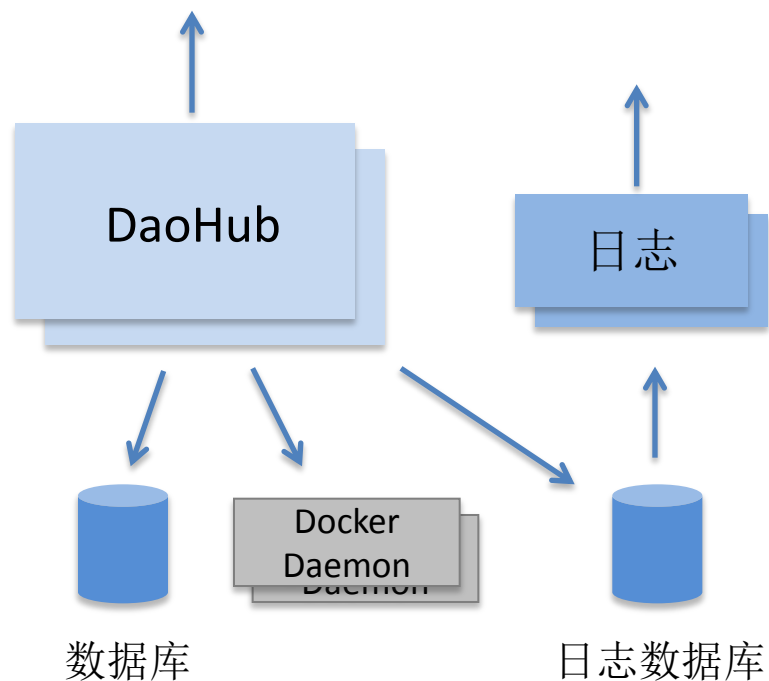


- 单机部署模式
- 镜像构建 / 持续集成
- 调用本地 Docker Daemon
- 日志存本地文件
- REST / Golang



- 有限的服务能力
- 单点故障
- 日志访问压力大

独立的横向扩展

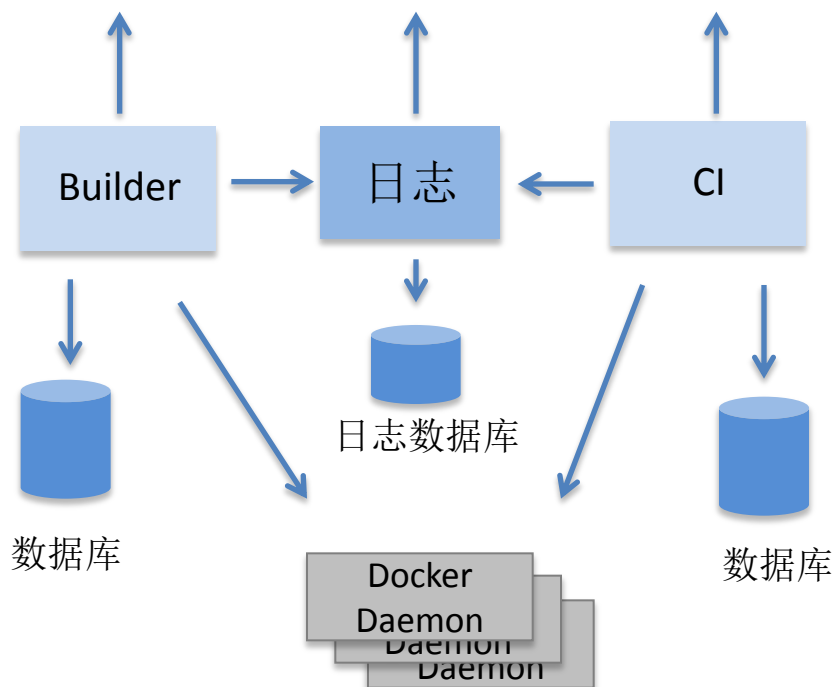


- 支持独立横向扩展
- 边界明确的日志模块
- 服务基于 DB 通讯



- Builder/CI 必须一起更新
- 模型的耦合

解耦的服务更新

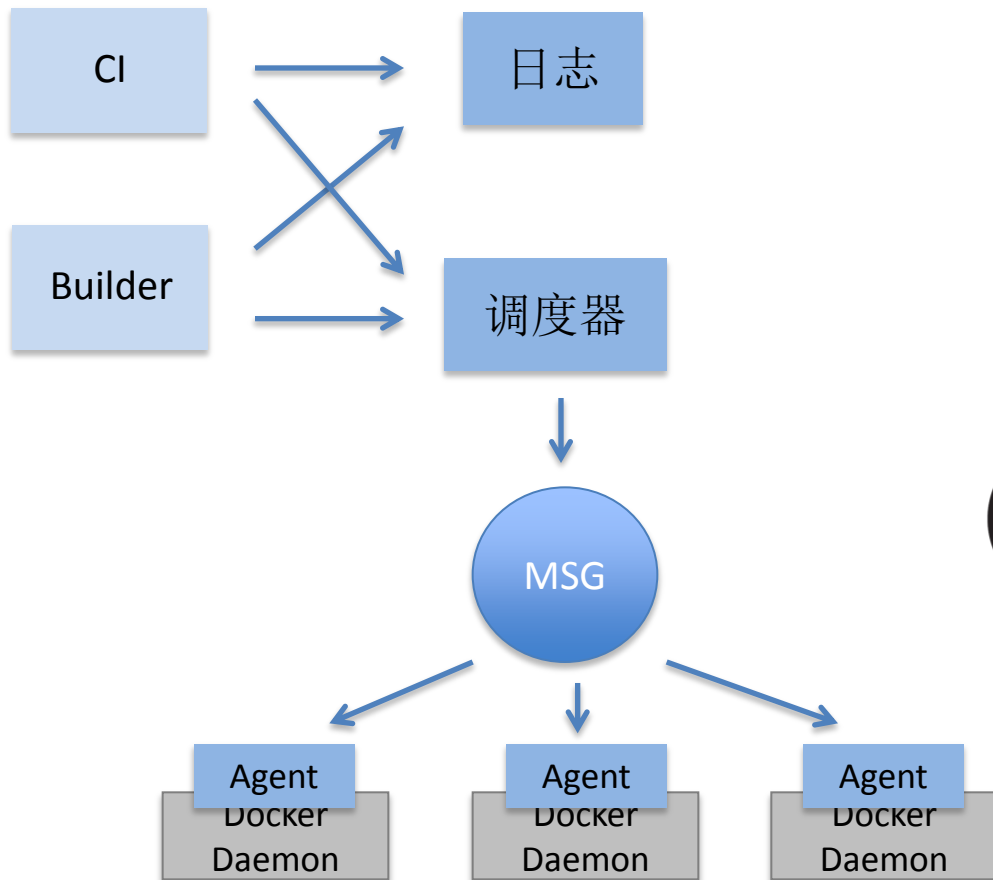


- Builder / CI 独立更新
- 拆分 team 维护 CI
- 隔离的数据模型
- 共享 Docker 集群
- 基于 Golang Channel 管理任务



- 健壮性
- 重复的逻辑
- 盲目调度

边界清晰的服务



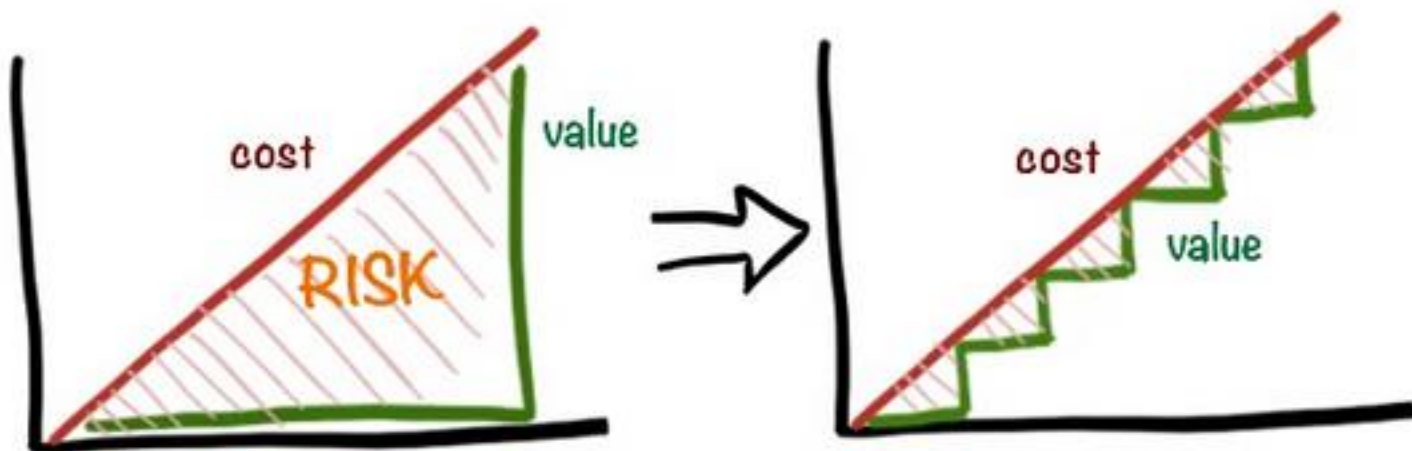
- 统一调度器
- 最合适的技术实现
- REST / MSG
- 高度动态的 Docker 集群



- 多组件的部署
- 快速测试发布
- 不同技术栈的运维

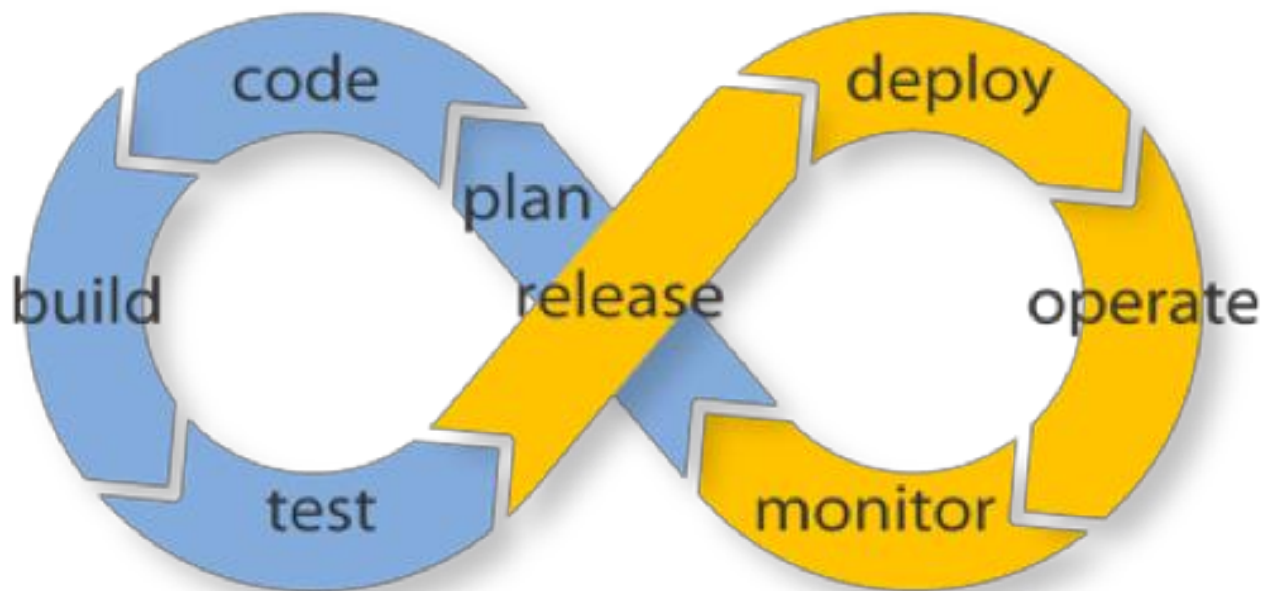
持续交付是...

一系列理念和最佳实践，使得软件构建，测试，发布更快，更频繁。



持续交付原则

- 发生在每次改变时
- 自动化
- 快速反馈



多样性的自动化难题

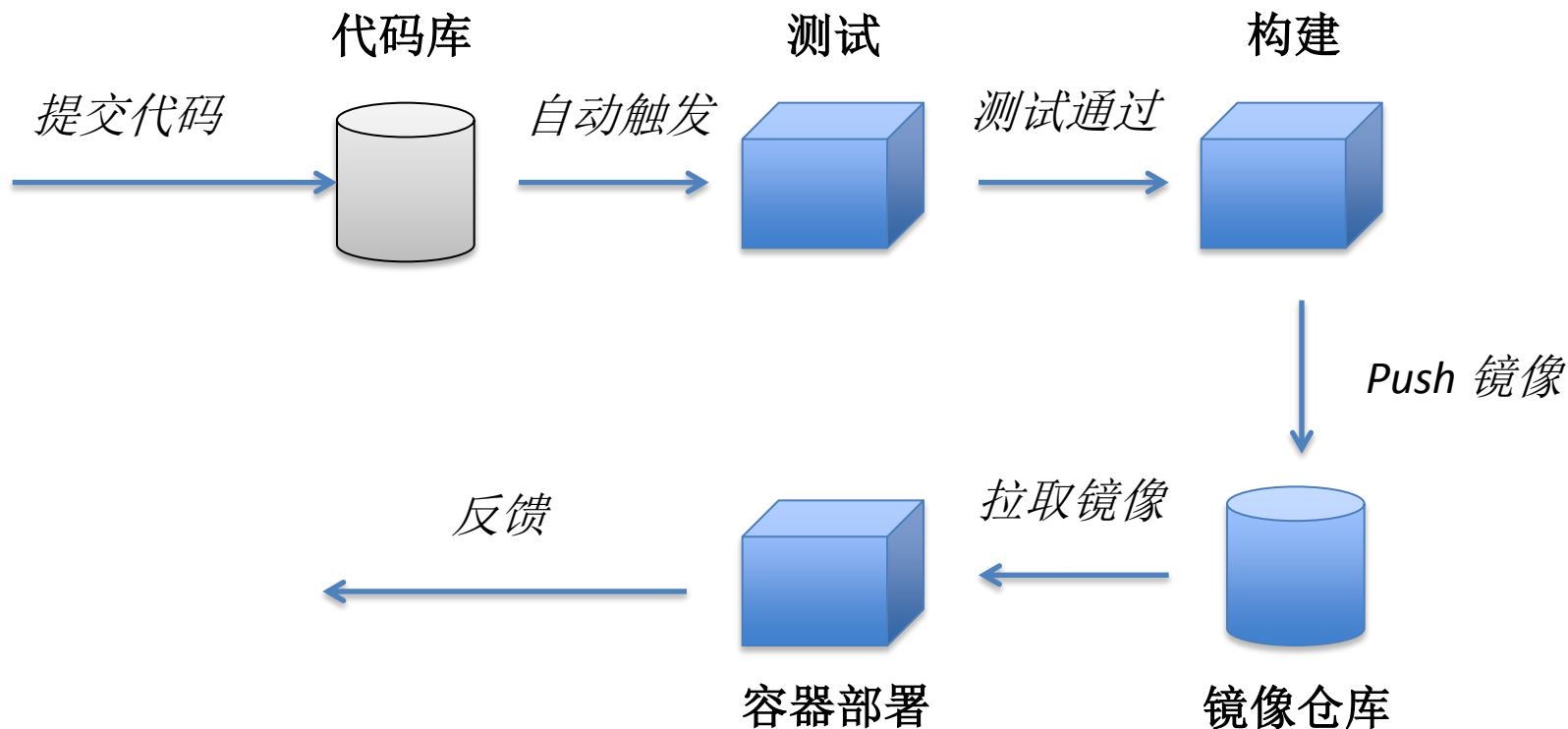
没错，想到的和没想到的，你都要面对



"DEFRAGMENT YOUR HARD DRIVE, REINSTALL YOUR OPERATING SYSTEM, UPDATE YOUR DRIVERS, AND BUY MORE MEMORY. THAT WILL KEEP YOU BUSY WHILE I FIGURE OUT WHAT'S WRONG WITH YOUR COMPUTER."

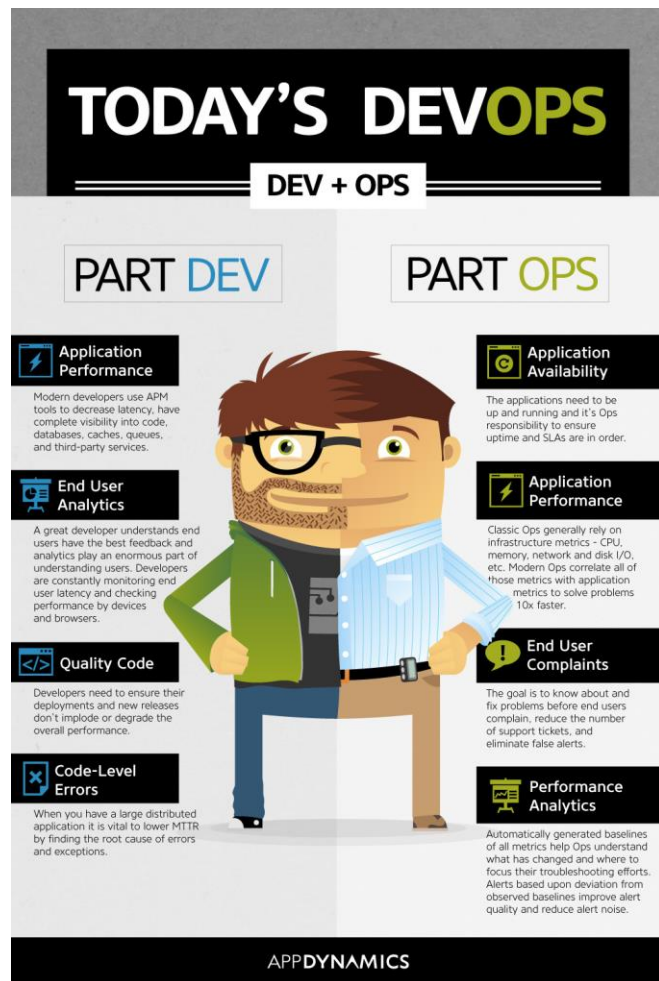
- Node.js
- Python
- Ruby
- Golang
- Java
- PHP
- ...

容器化微服务的持续交付



打造微服务精益研发流程

- 服务 Docker 化
- 从 Dev 到 DevOps
- 代码驱动的自动化
- Eat Your Own Dogfood



微服务持续交付 Demo

总结

- 微服务是个好东西
- 设计之初可以立足于应用而非服务拆分
- 微服务并非越“微”越好
- 微服务对持续交付提出新的需求
- 容器化和微服务珠联璧合
- 立足自身，实践获真知

We Are Hiring

发送简历至 jobs@daocloud.io



领先的专注于容器技术的云计算公司，提供容器化应用开发、测试、发布自动化流程，核心研发团队来自 EMC, Vmware, Oracle, Microsoft, 阿里, 盛大等一线高科技公司。

成立于2014年末，获得光速安振风险投资，总部位于上海，在北京及旧金山设有分支机构。

THANKS

全球容器技术大会

剖析容器企业实践 关注容器生态圈开源项目