

4. 服务层

算法服务

通用算法
基础算法
算法共享，协作开发

模型服务

Model directory
Model sharing
Model deployment
Model validation
Model Assessment

数据服务

统一数据访问
基础数据大联通
特征工程数据共享
数据处理工具链

总结

1. 通过在线服务提供机器学习一站式平台。
2. 通过容器实现多租户隔离，版本管理，和资源弹性扩展。
3. Notebook和DAG Editor相辅相成成为用户入口。
4. 强调共享协作。数据共享，模型共享，算法共享。
5. 以提高生产力和支撑前沿探索系统需求为最高目标。

Thank you!

唯品会
全球精选 正品特卖

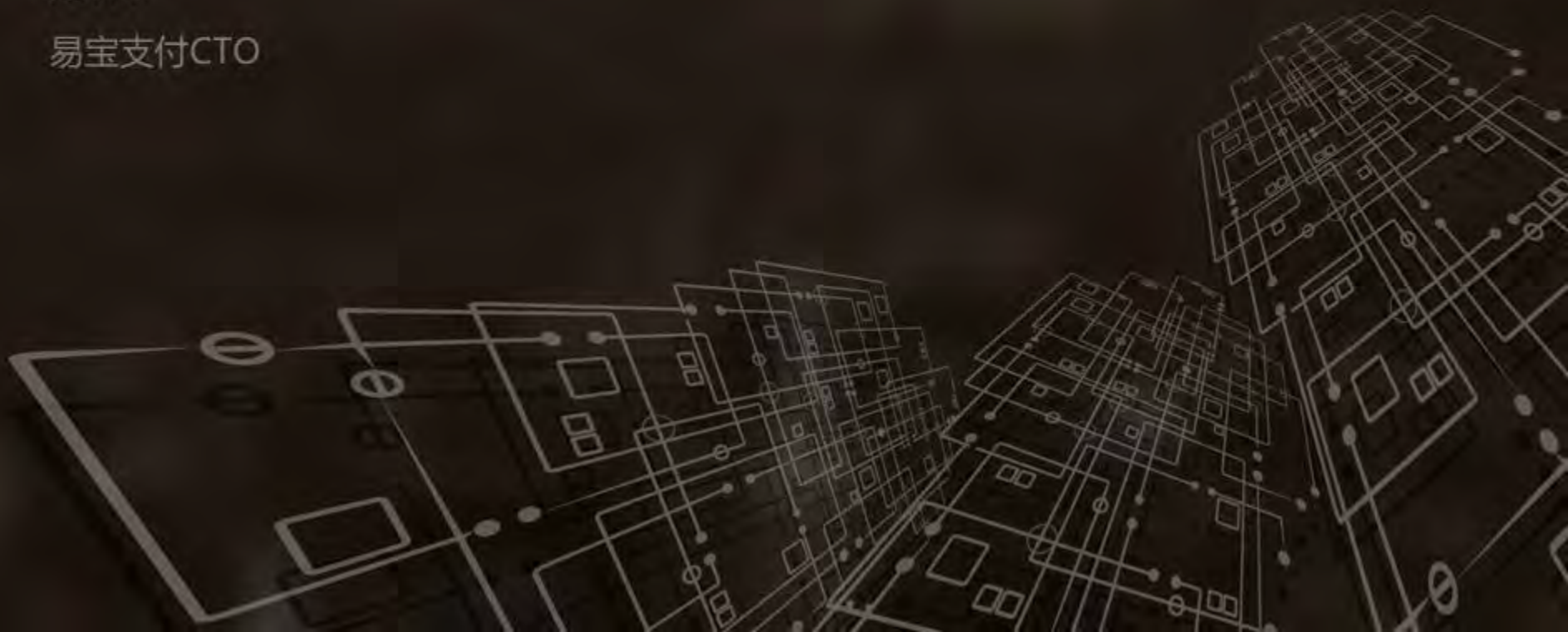


2017 Software Architecture Summit

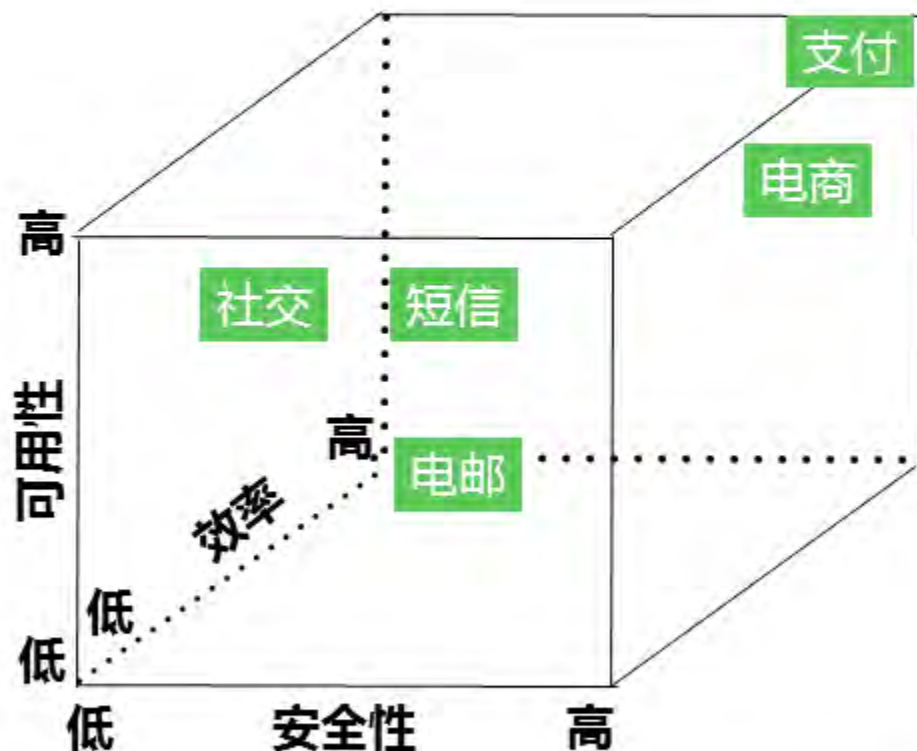
如何构建大型支付交易平台

陈斌

易宝支付CTO



支付交易平台的特点



支付交易平台

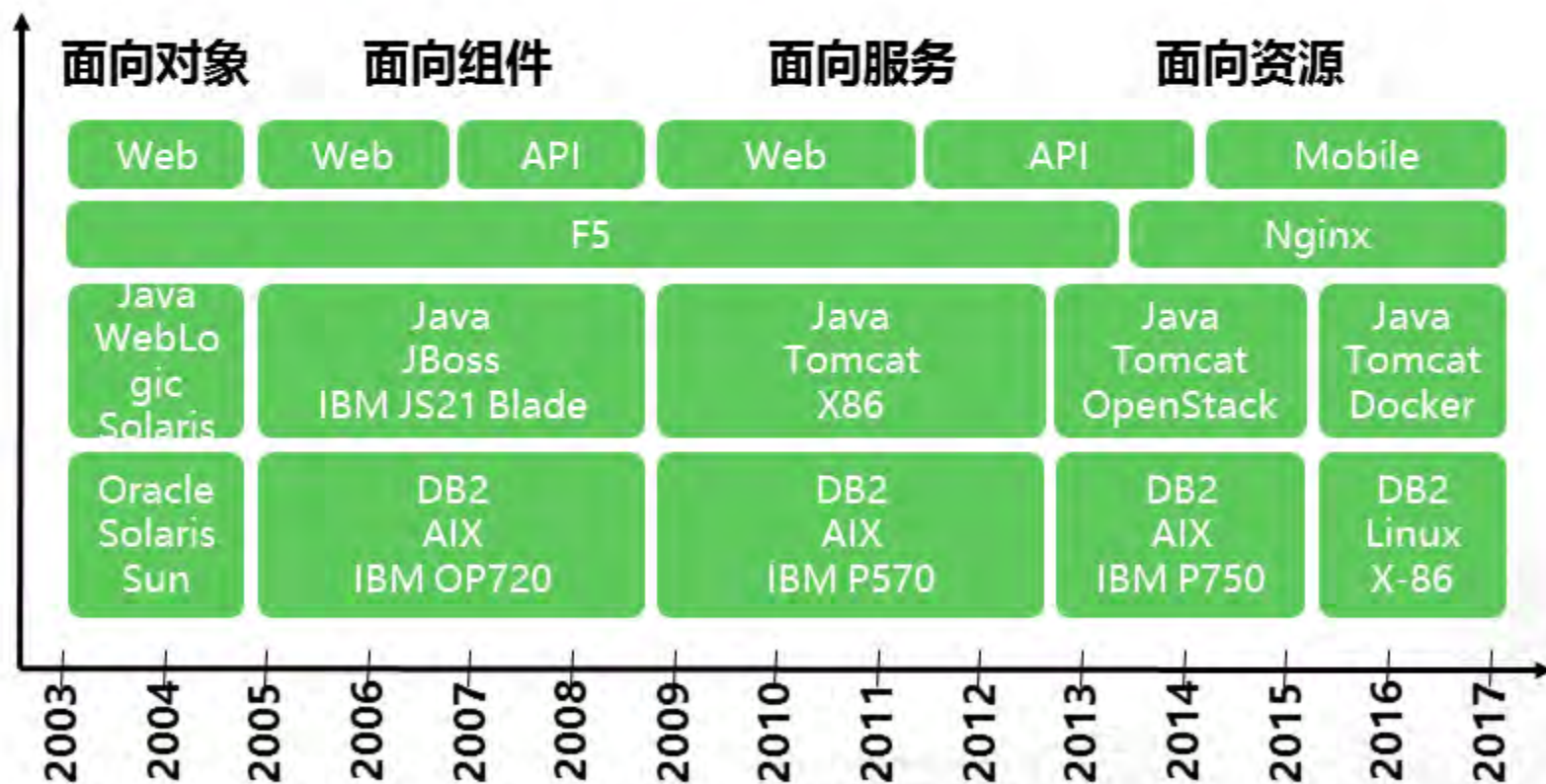
高效率

高安全性

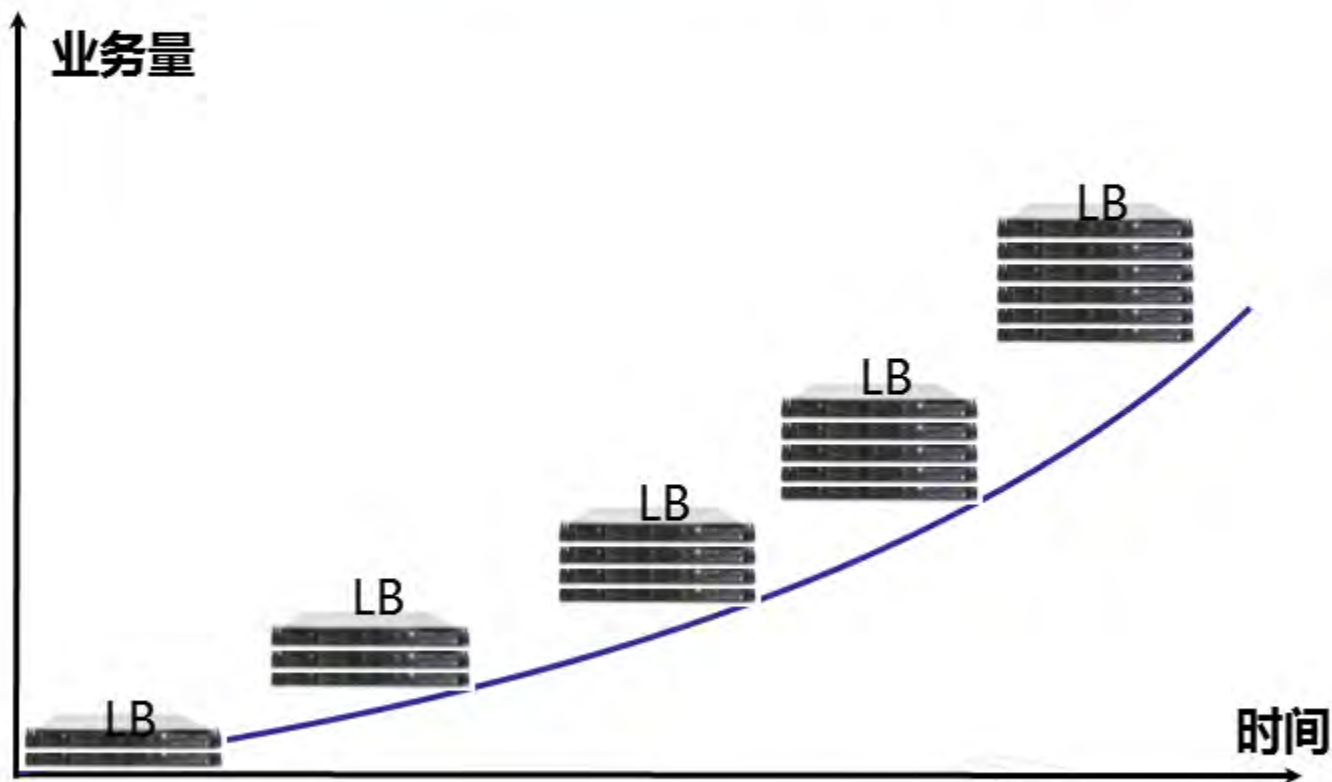
高可用性

- 高可用性：7 X 24 X 365随时可用
- 高安全性：需满足PCI-DSS的要求
- 高效率：每笔交易的成本要够低

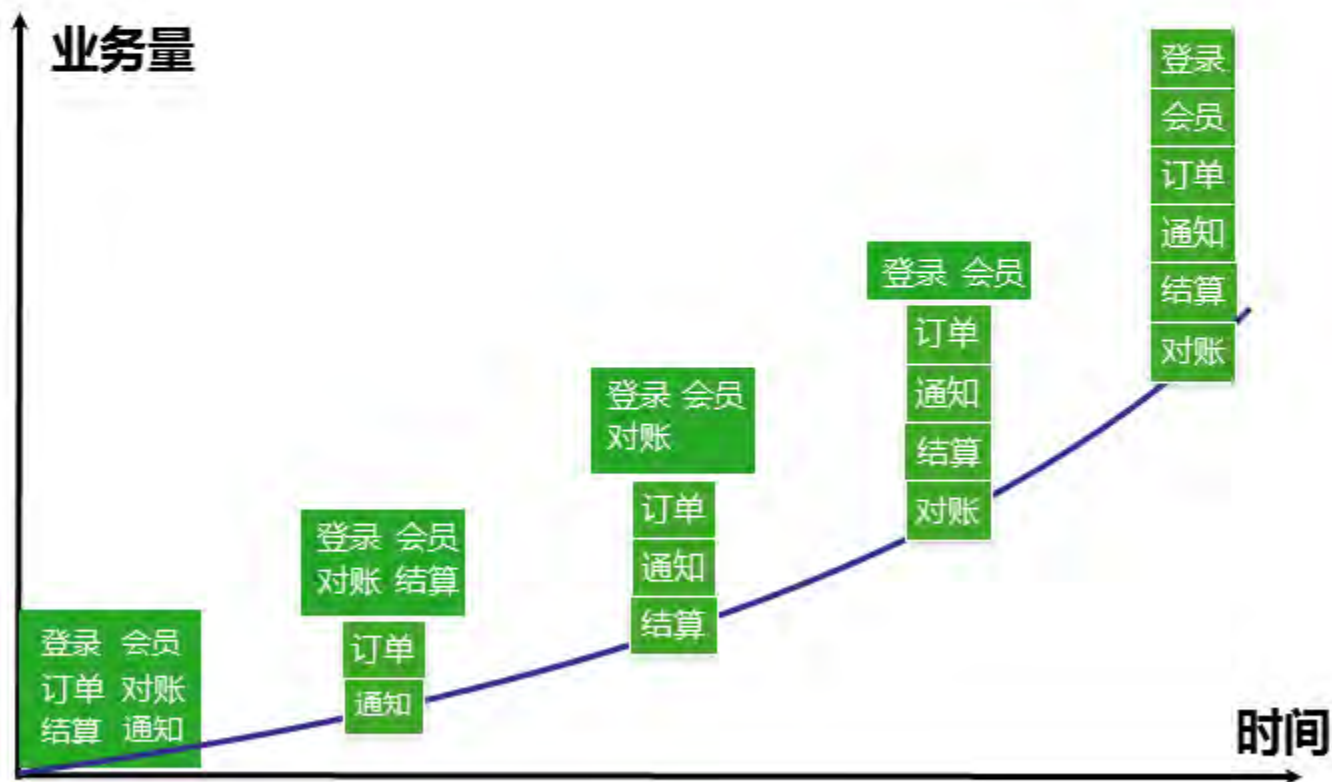
易宝支付交易平台的发展历程



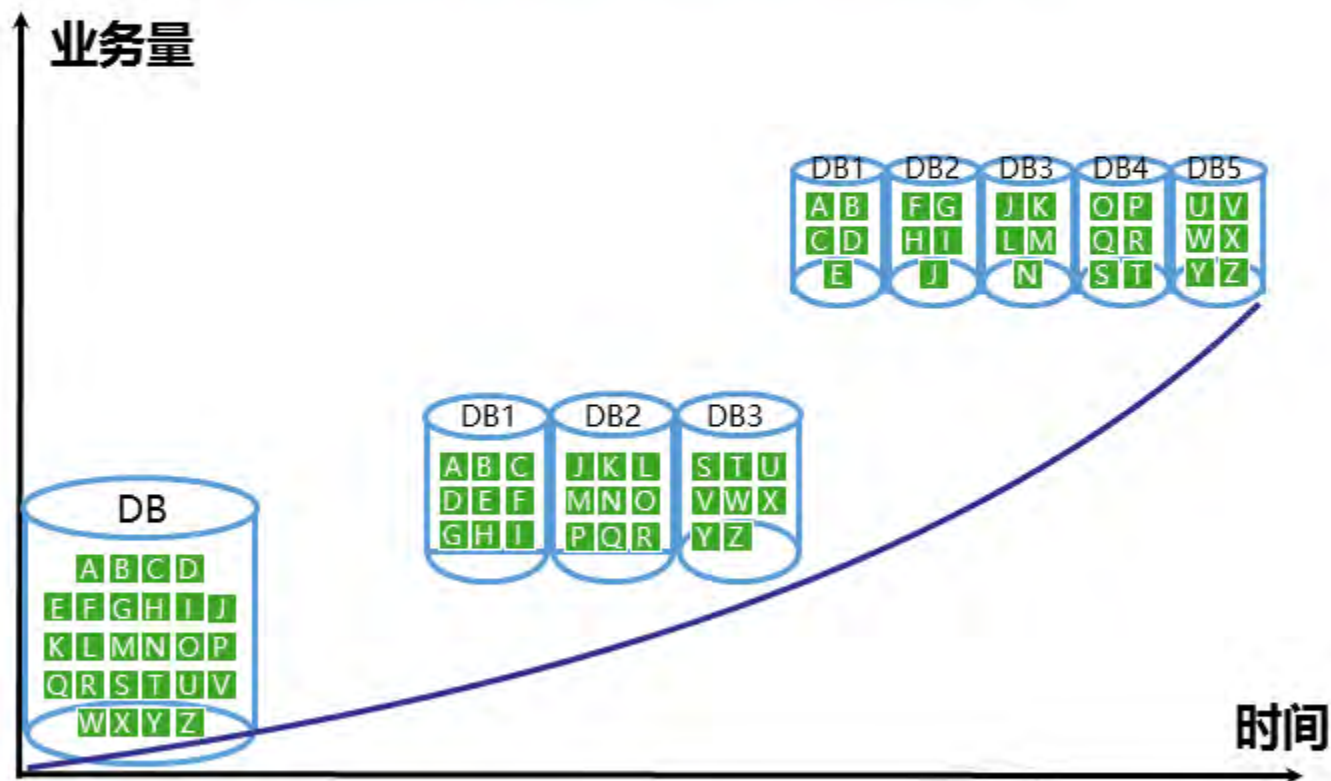
X轴扩展-水平扩展



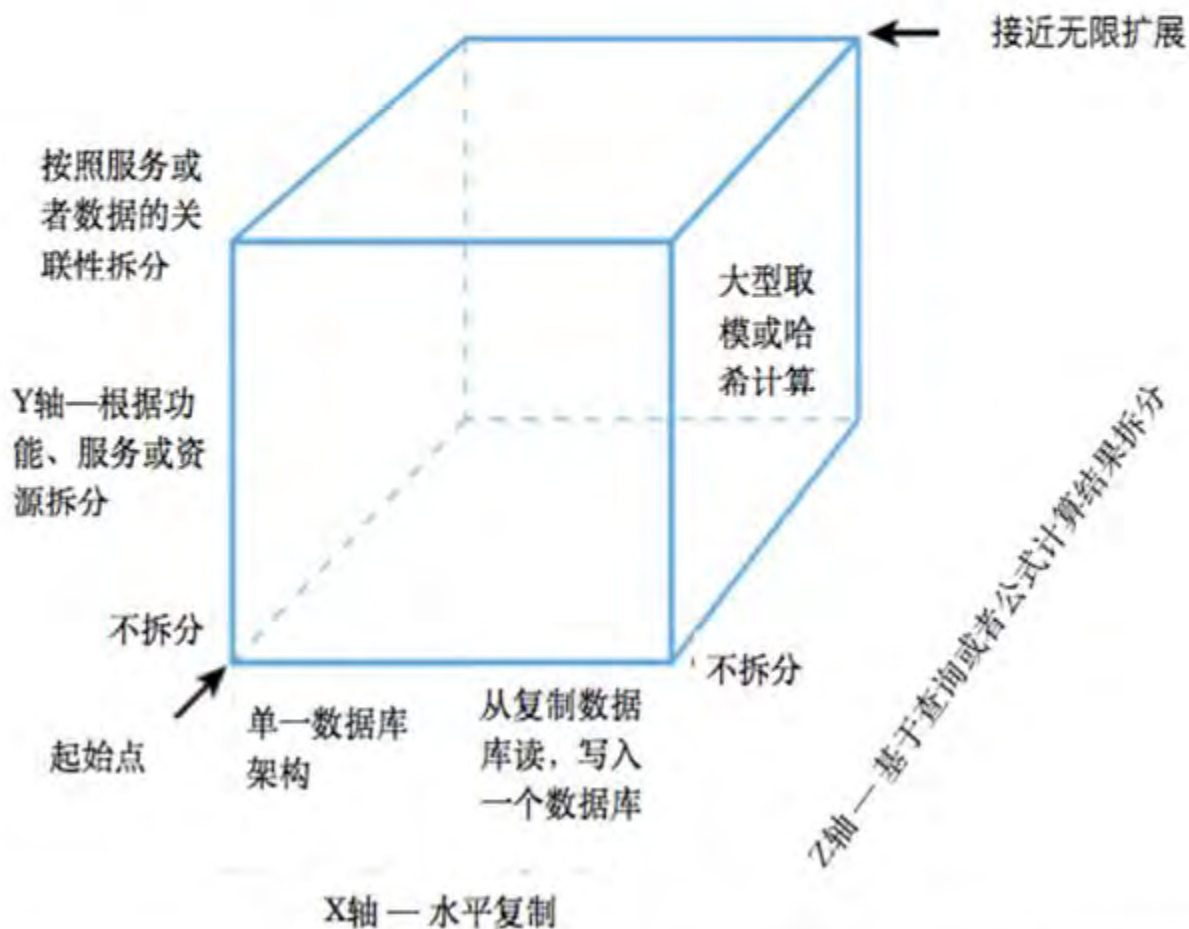
Y轴扩展-应用拆分



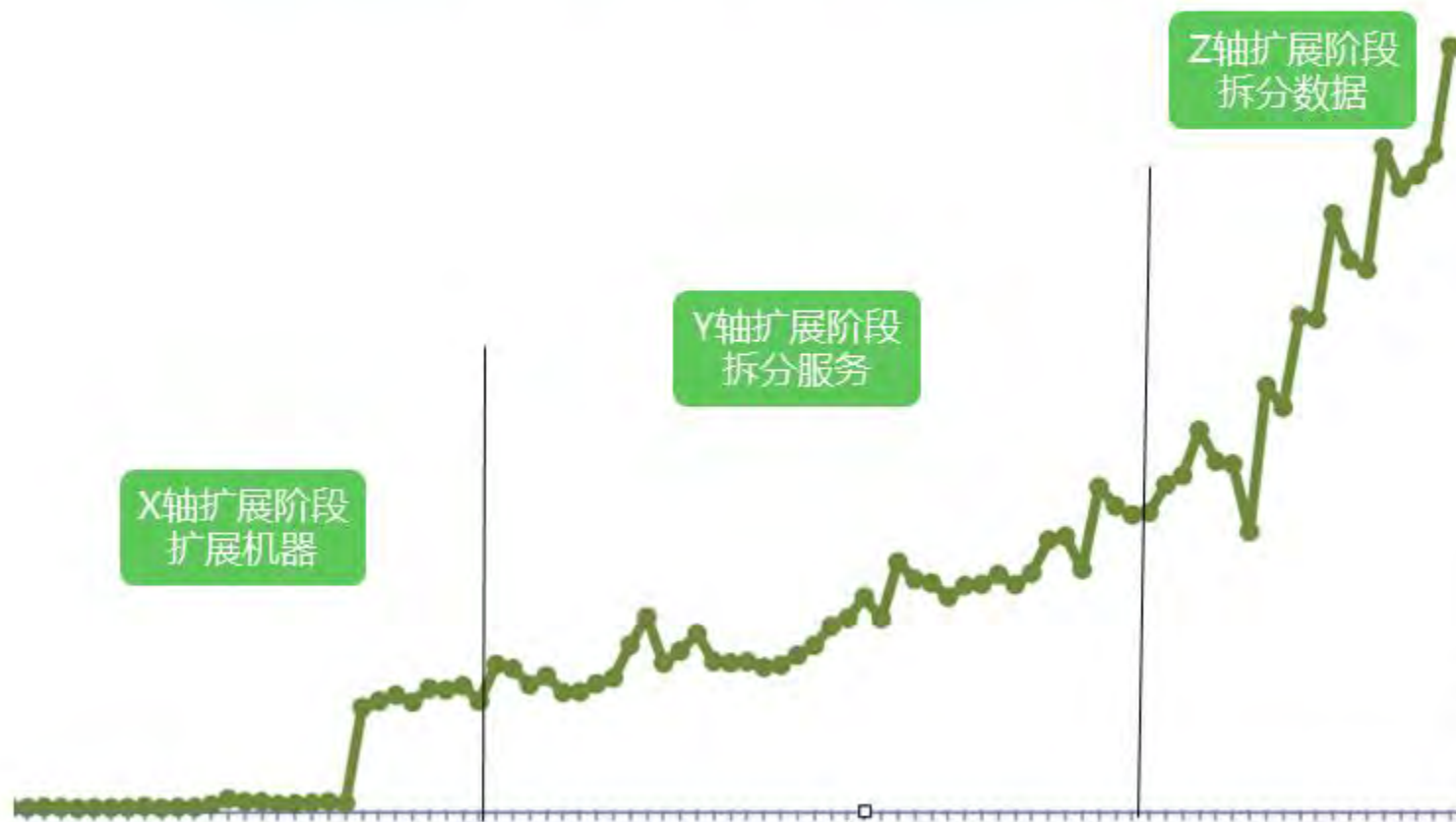
Z轴扩展-数据拆分



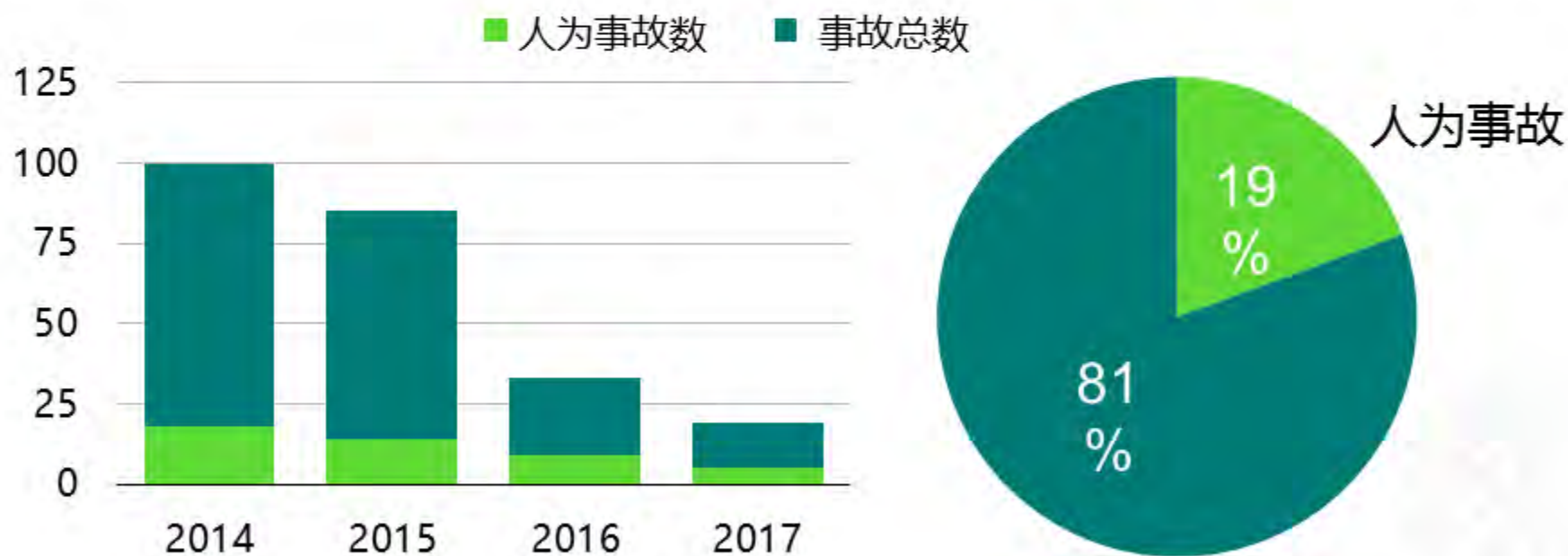
NKF的三轴扩展理论



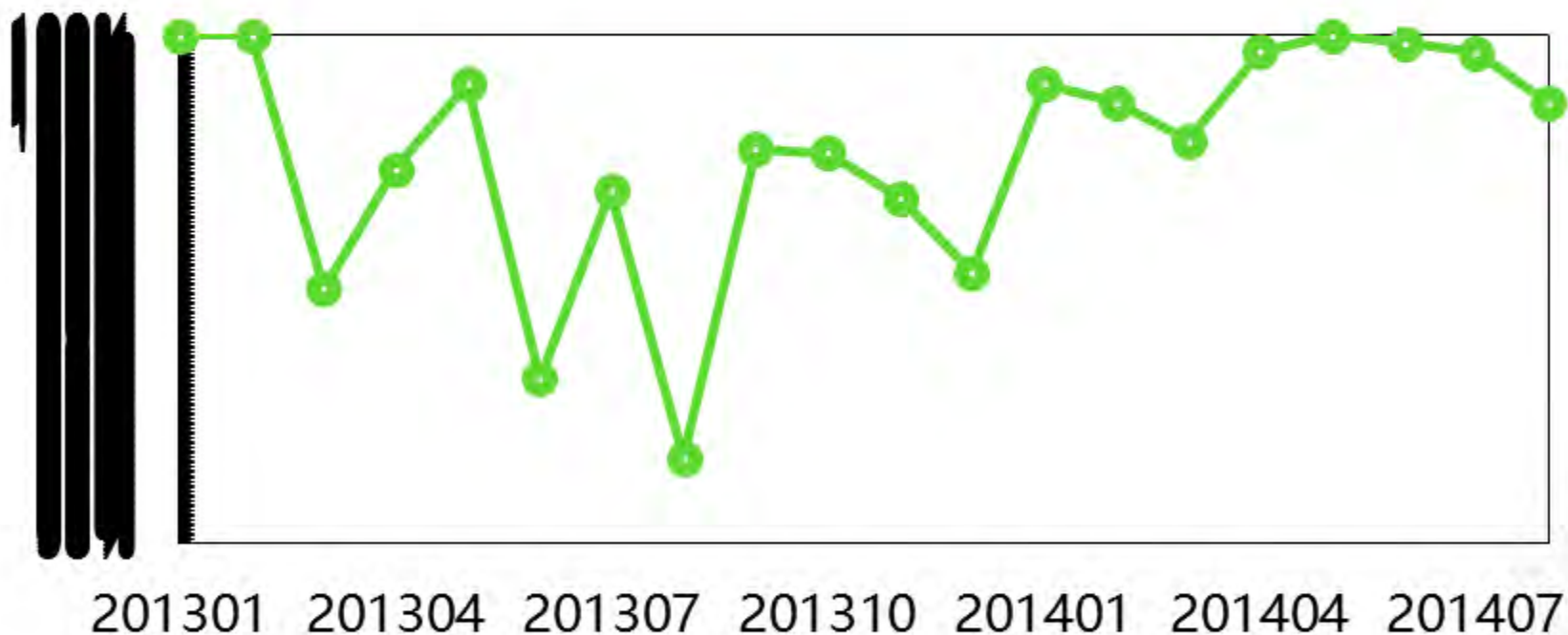
易宝支付平台扩展演进历史



版本多、环境杂，步骤多、指令烦



支付平台的可用性不断波动

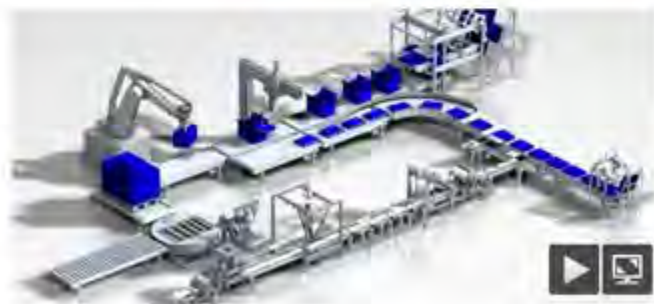


运维面对挑战

人常犯错误，而且往往多次犯同样的错误。人也容易对重复性或琐碎的工作失去兴趣。对企业而言，人员成本都是显著的，而且好的人才价值不菲。

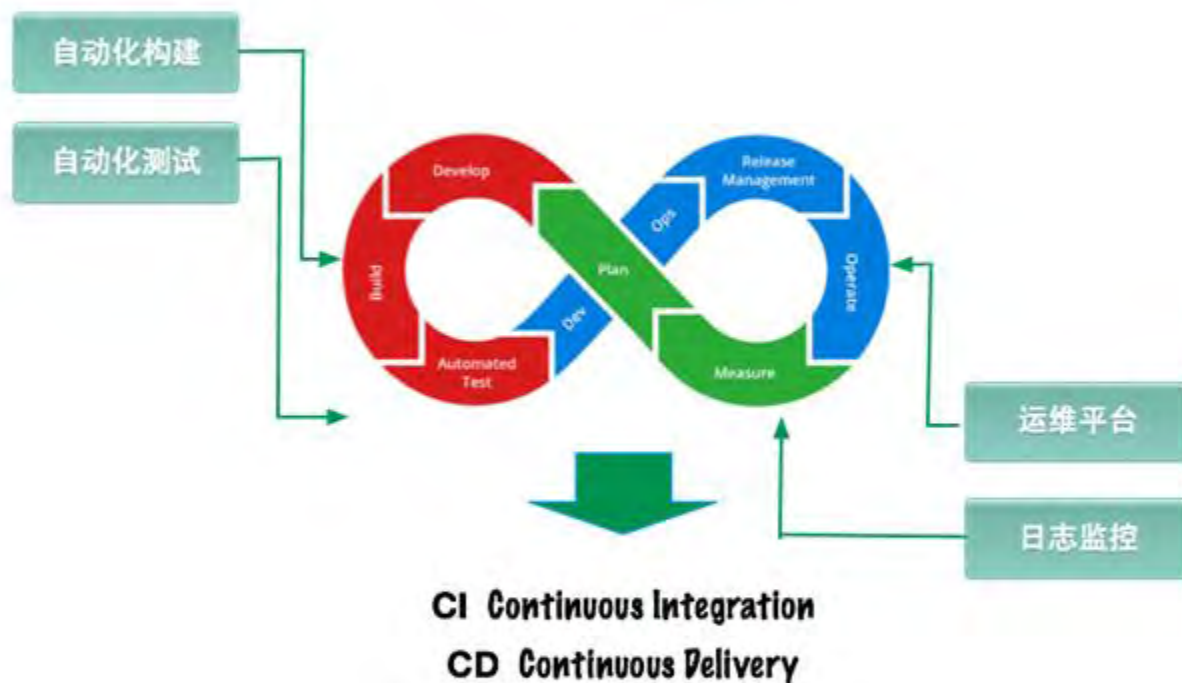


编码 → 测试 → 集成 → 部署 → 监控



自动化机器每次都会完成简单重复的任务，或取得相同的成功，或犯同样的错误。所有系统都该在设计阶段考虑自动化部署、构建、测试、监控。

DEVOPS



技术选型



功能比较简单
技术栈简单
非稳定版



偏重资源抽象
技术栈复杂
大规模场景



自主研发
投入产出比低
项目风险大
机会成本太高

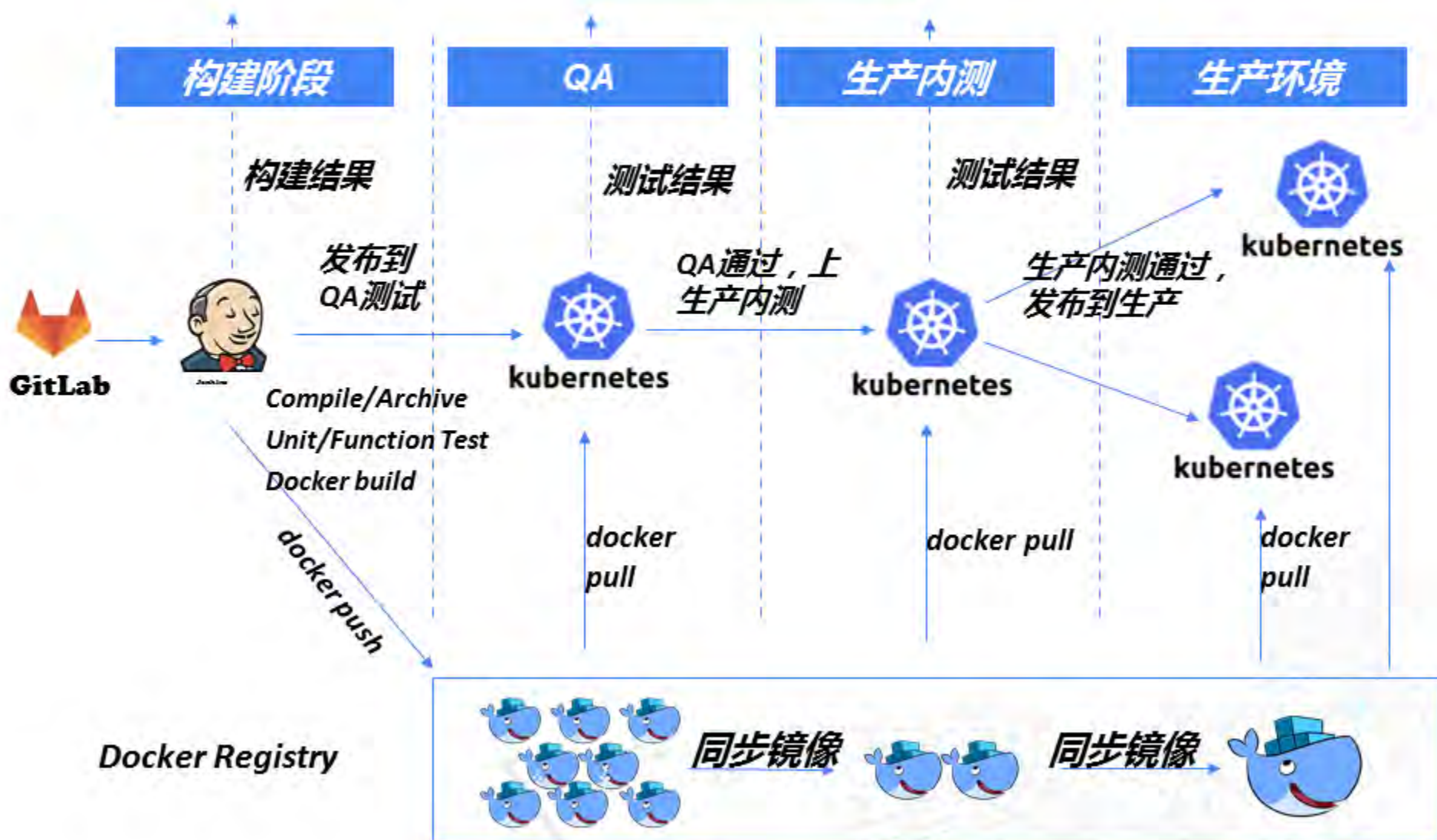


kubernetes

理念先进
以应用为中心
Web应用为主
功能完善

Pass!

易宝容器云



搭建基于容器云的基础设施



保持弹性



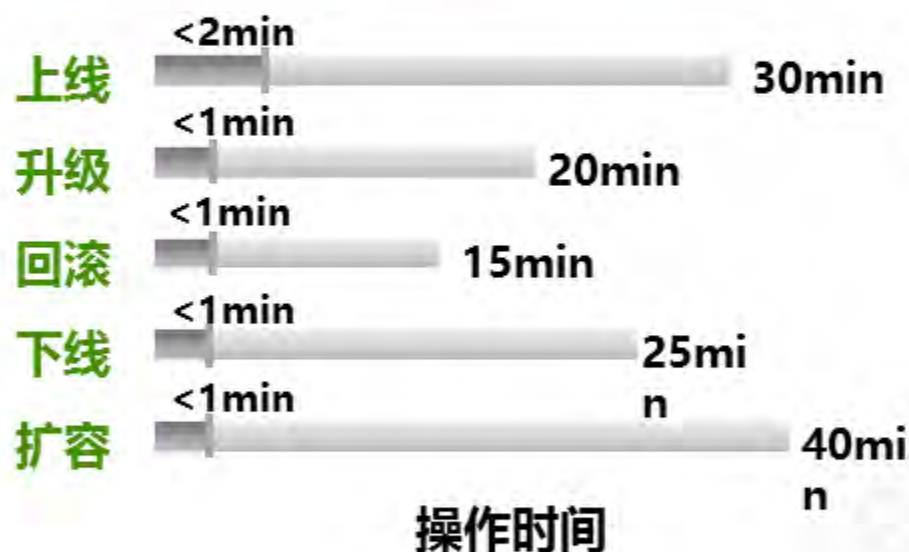
支付平台的运维效率得到了明显改善



5 : 1

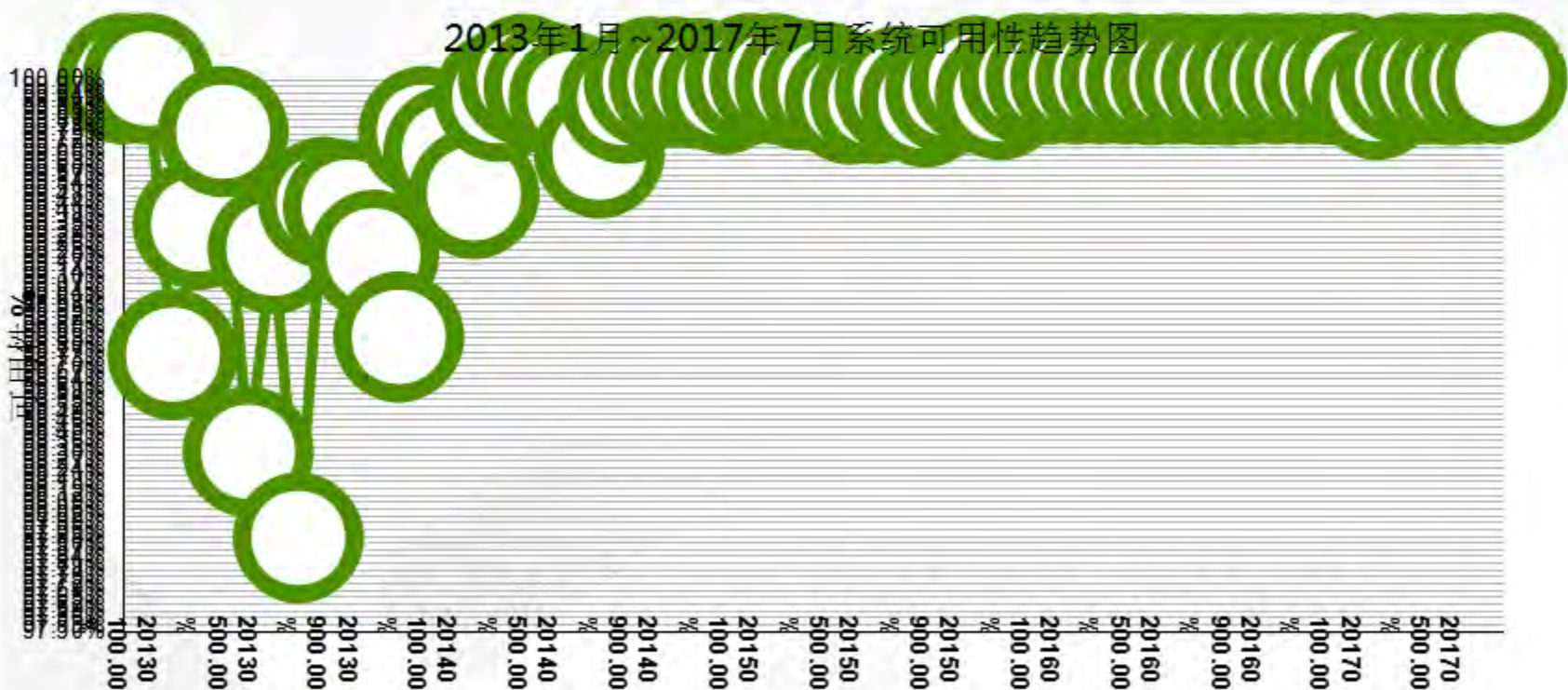


迁移前后

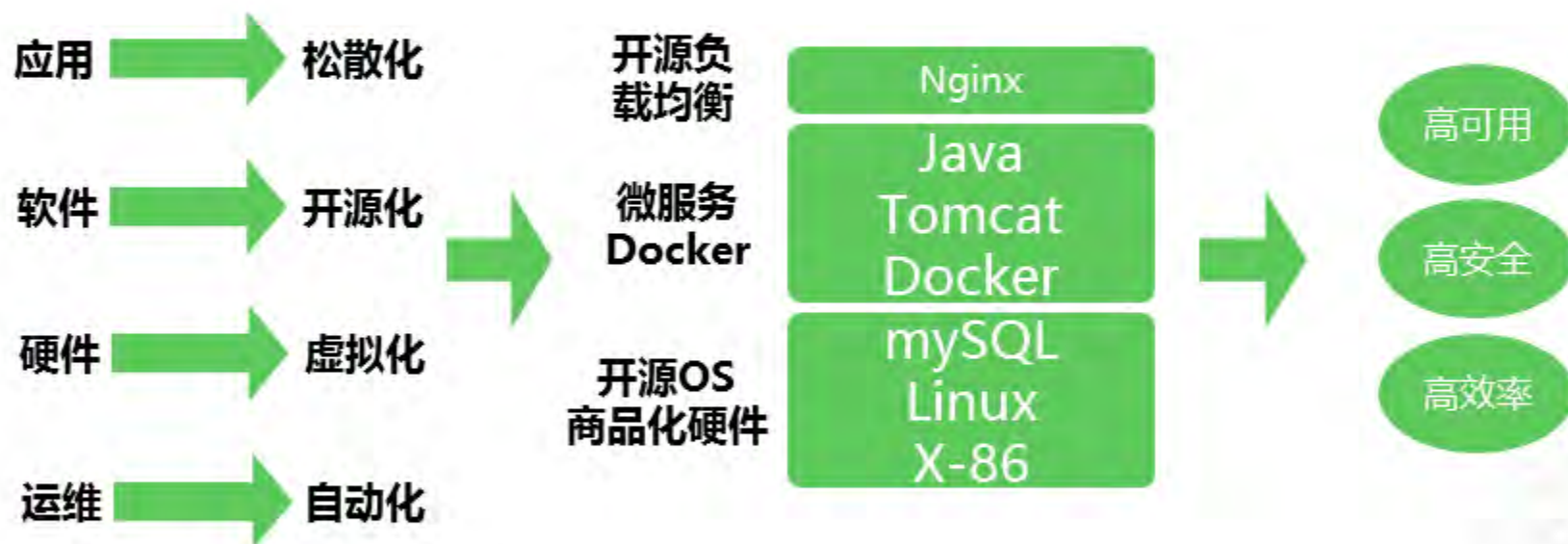


支付平台的可用性得到了大幅度的提高

2013年1月~2017年7月系统可用性趋势图



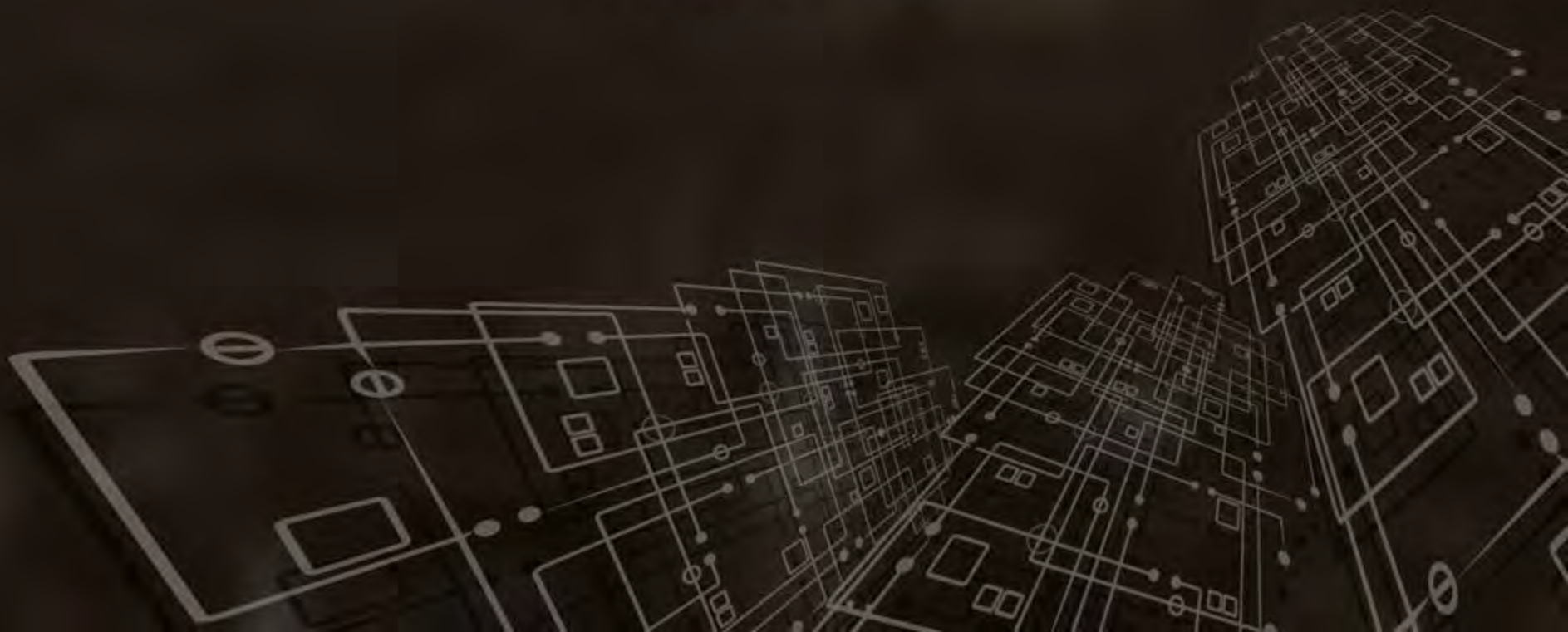
易宝支付平台的发展趋势





2017 Software Architecture Summit

ThankYou !



DevOps在链家上研的实践与探索

郭飞

2017年10月

 链家
连接每个家的故事

目录

- 为什么用DevOps ?
- DevOps的探索
- 构建要点

为什么用DevOps？

存在什么痛点？

- 新项目资源申请与发布过程繁杂
- 项目众多，各项目资源使用不均
- 资源分配、配置调整为人工操作，出错率高
- 项目动态伸缩困难
- 项目下线、资源不能及时回收

为什么用DevOps ?

什么是DevOps ?

- 一组工具、过程及系统的统称
- 是一种理念，提倡开发和运维之间的高度协作，打破部门墙
- 强调对应用全生命周期的管理
- 为什么是现在？
 1. 微服务与容器技术的应用
 2. 敏捷开发模式的普及
 3. 系统架构越来越复杂



DevOps的探索

DevOps的探索

目标是什么？

- 能对资源的使用情况进行全局掌控，合理分配资源
- 对项目全生命周期进行线上化管理
- 尽可能减少人工操作，降低风险
- 减少跨部门的不必要沟通，提升持续交付效率
- 提升持续交付的健壮性与安全性

DevOps的探索

- 核心关键词

1

自动化

2

自助化

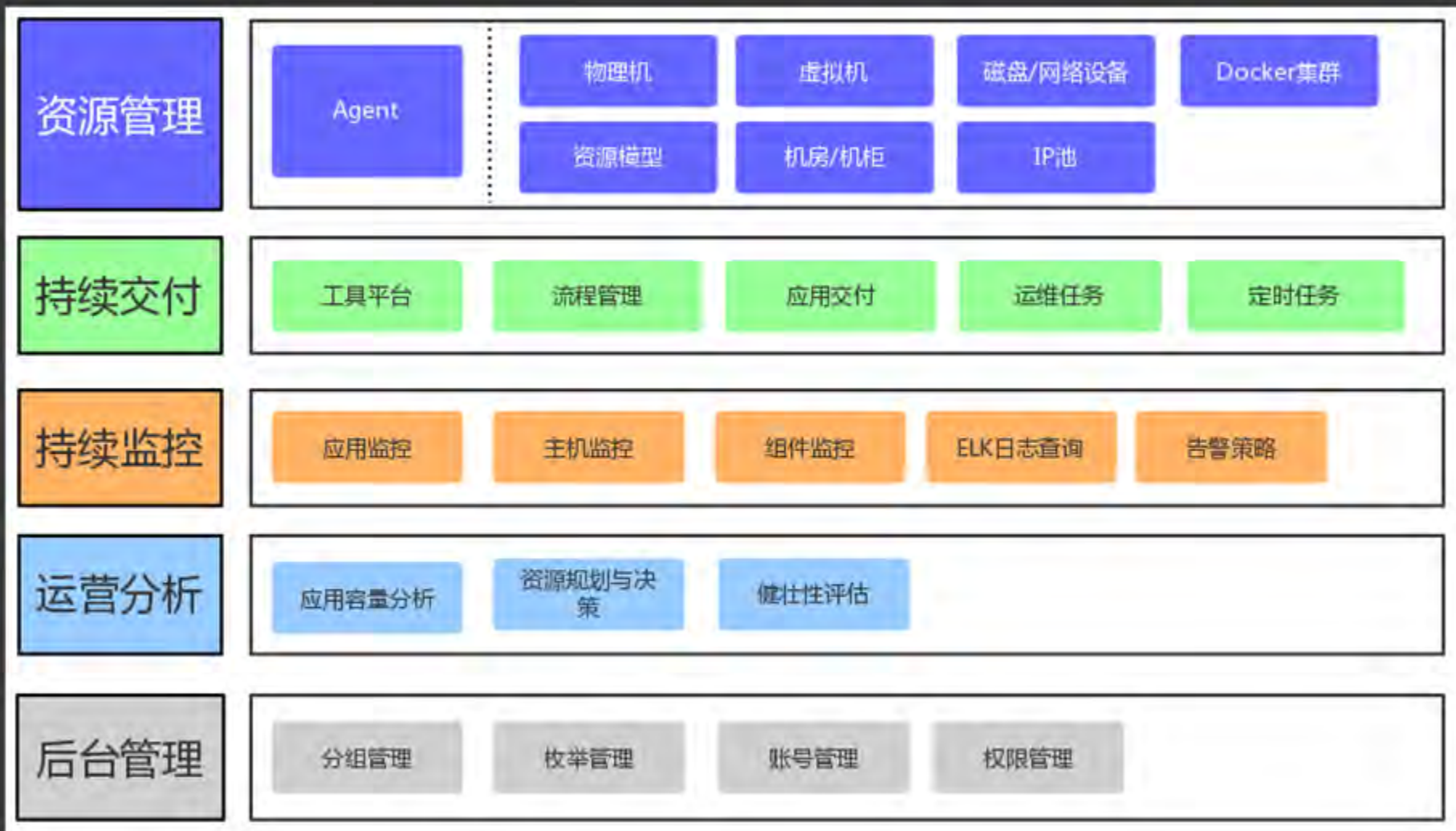
3

协作

4

智能

系统结构



一、资源管理

主要资源

物理机

K8S集群

IP池

虚拟机

其他资源

机房

机柜

网络设备

存储设备

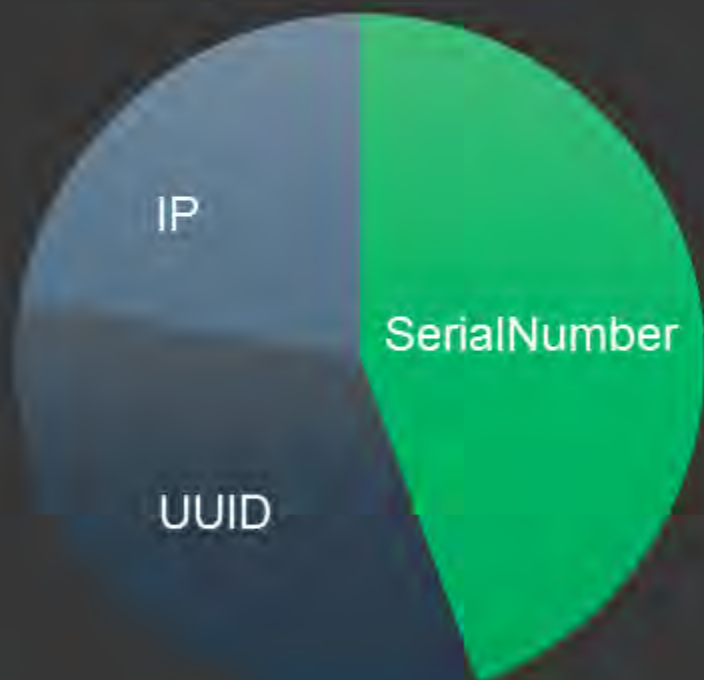
一、资源管理

核心是自动化 (Agent)

- 上下架自动发现
- 资源配置变更自动同步
- 状态变更自动处理
- 运行进程自动采集
- 使用容量动态更新

一、资源管理

资源唯一标识设计

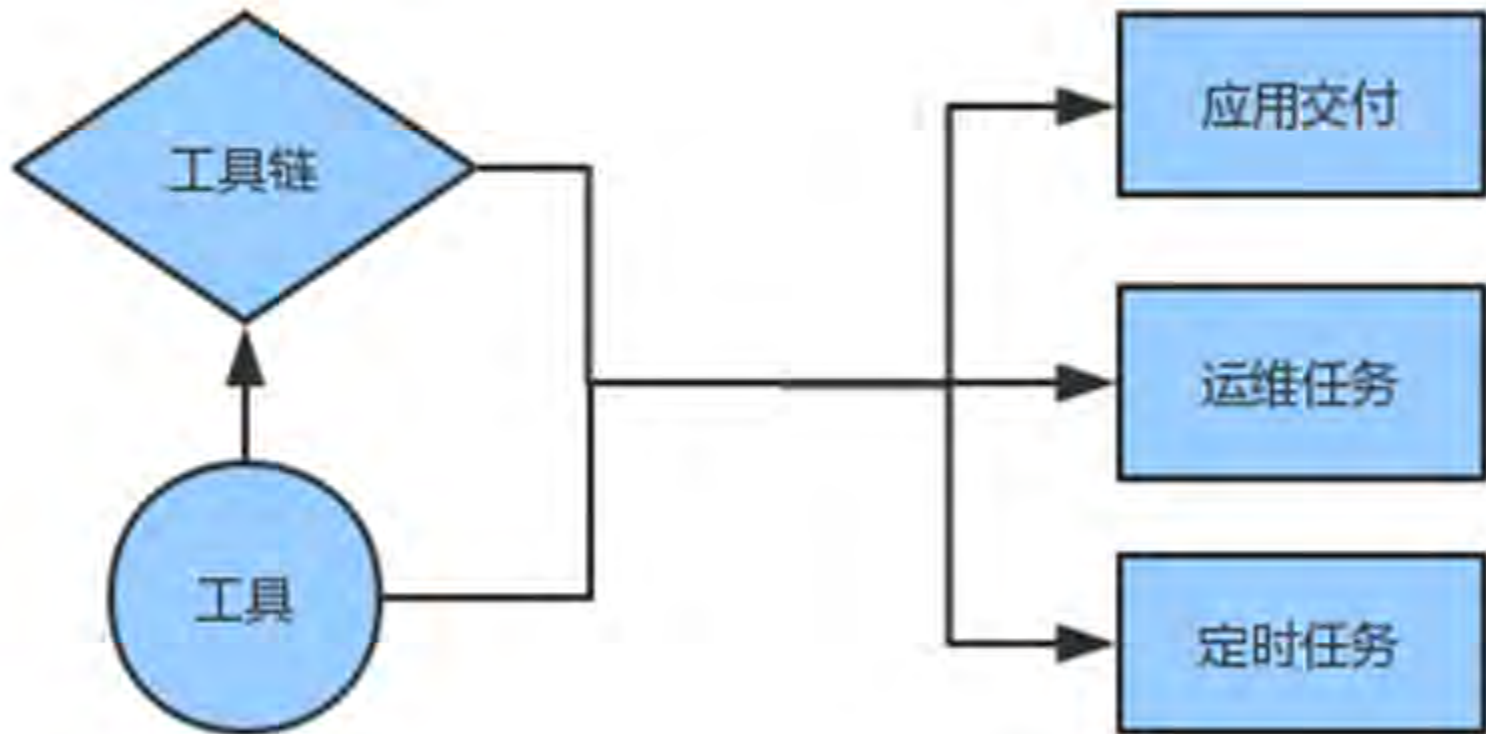


一、资源管理

资源调度决策

- K8S调用API
- Xen调用API
- 资源申请的自动决策
- IP申请的自动规划

二、持续交付



二、持续交付

工具管理平台（在线脚本管理）

- 丰富的脚本类型
 - Shell、Python、PowerShell、Java
- 标准输入、输出
- 开放的工具平台
 - 系统内置
 - 面向全体工程师群体
 - 也对接外部api

二、持续交付

工具的安全控制

- 安全级别定义
- 加入审批机制
- 脚本安全检测

二、持续交付

流程管理

- 多个工具的组合
- 核心是流程执行器的设计

二、持续交付

应用交付

- 场景定义
 - 初始化、部署、回滚、重启、下线、配置变更、节点伸缩。。。
 - 通过场景实现对应用全生命周期的管理
- 多环境下的不同流程
 - 非正式环境下自助化操作
 - 正式环境下的审批流设计
- 资源使用自动规划（如新系统资源分配、添加节点）
- 权限的开放与公司管理制度相匹配

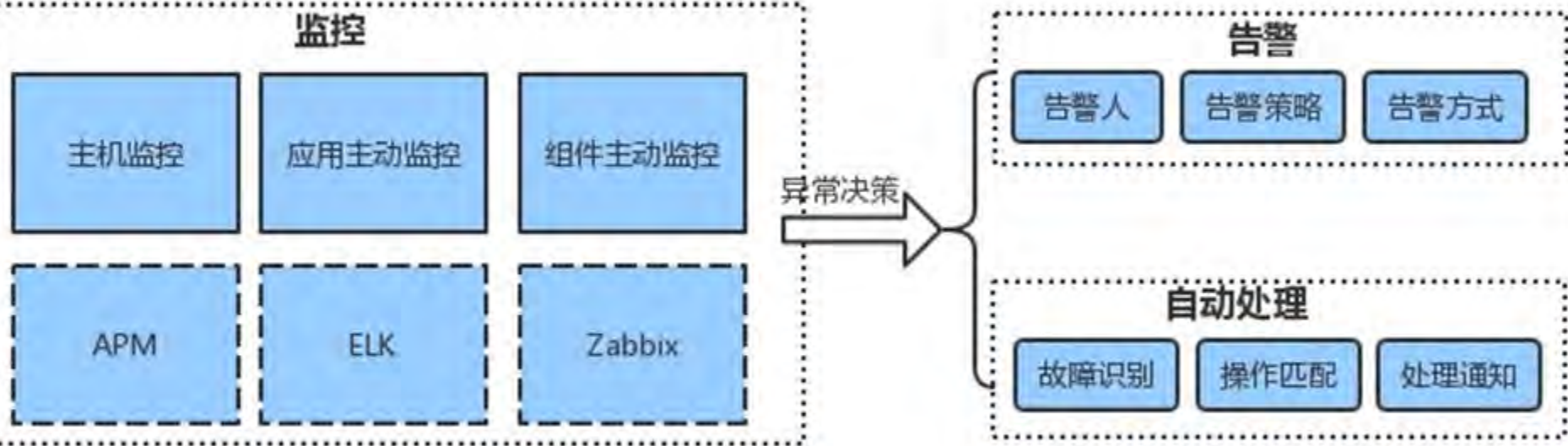
二、持续交付

- 运维任务

- 运维的常用操作线上化

- 定时任务

三、持续监控



三、持续监控

与原有监控系统是什么关系？

- 对原有监控报警的整合与补充
 - 对接zabbix、APM，让其纳入应用生命周期管理范畴
 - 提供更好、更准确的告警服务（告警人）
 - 整合各种组件监控与告警
- 侧重点不同
 - 关注点不同：更关注系统可用性，淡化故障排查过程
 - 结合APM系统达到自动故障定位、自动处理的目的

三、持续监控

应用监控

- 规范化应用可用性检测接口
- 主动心跳检测

主机监控

- 通过Agent完成对主机的运行状态监测

组件监控

- Redis、nginx、mysql、mongoDB、ES的心跳检查

告警策略

- 多种告警方式（结合工具的使用）
- 告警频率与阈值
- 告警人与分组告警

三、持续监控

自动处理

- 异常类型定义
 - 故障：失去响应
 - 负载过高：CPU、内存、线程
- 异常处理流程匹配

四、运营分析

解决什么问题？

- 帮助解决资源的合理使用问题
- 帮助进行资源规划
- 帮助对应用进行健壮性评估

四、运营分析

1.应用容量

- 帮助发现应用容量的瓶颈风险
- 帮助发现应用资源使用上的浪费
- 自动对应用资源进行伸缩
 - 如何做到？

四、运营分析

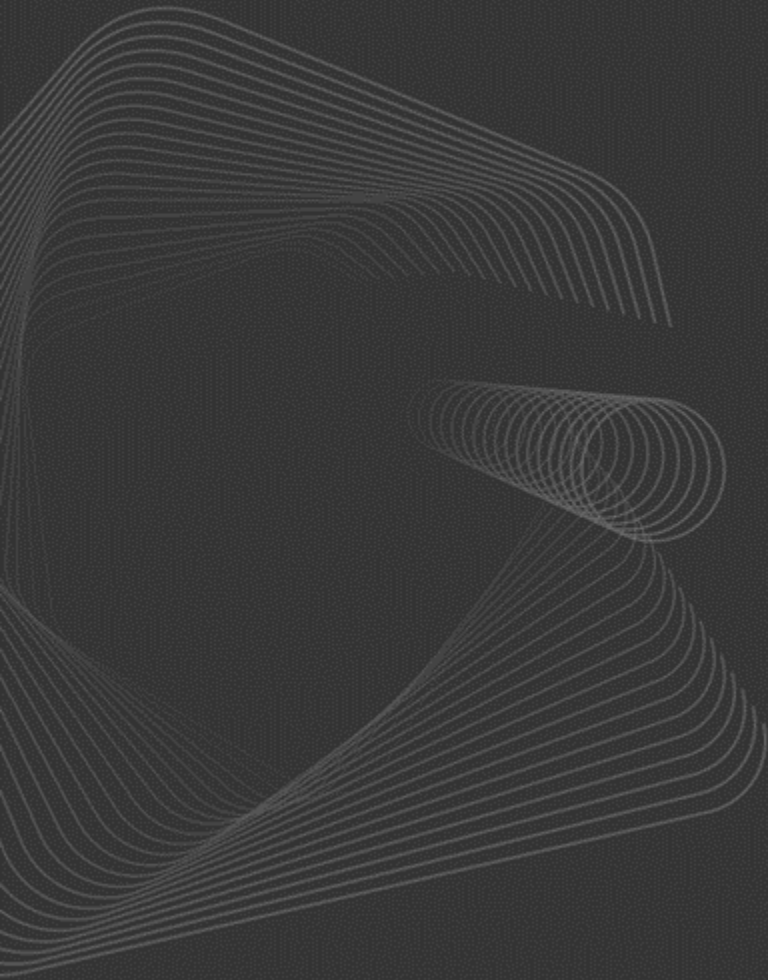
2. 资源规划与采购决策

- 资源剩余量预警
- 可用时长评估
- 业务组资源占用统计与费用测算
 - 资源使用是否合理？
 - 与业务价值是否符合？
- ...

四、运营分析

3.应用健壮性评估

- 部署成功率
- 故障率
- 故障时长
- 非正常发布频率



构建要点

构建要点

- 尽可能范围内自动化、自助化
- 零停机部署（工具与流程设计）
- 工具共享
- 从实践中学习
 - 忌大而全，与公司实际情况相结合
 - DevOps的完善也是持续迭代的过程
- 微服务与传统架构均可使用



连接每个家的故事

2017 Software Architecture Summit

一站式软件交付：世界五百强企业中的DevOps转型之道

马致杰 *George Maloney*

JFrog 中国创始人 / CEO

前Marvell ARM服务器产品负责人

前Vungle亚洲总监



ORACLE®

CISCO

ING



1 个人介绍



马致杰 *George Maloney*
JFrog 中国创始人 / CEO

前Marvell ARM服务器
产品负责人

前Vungle亚洲总监

软件开发趋势

PROJECT EXECUTION METHODOLOGIES – THE CHANGE

WATERFALL 1985年



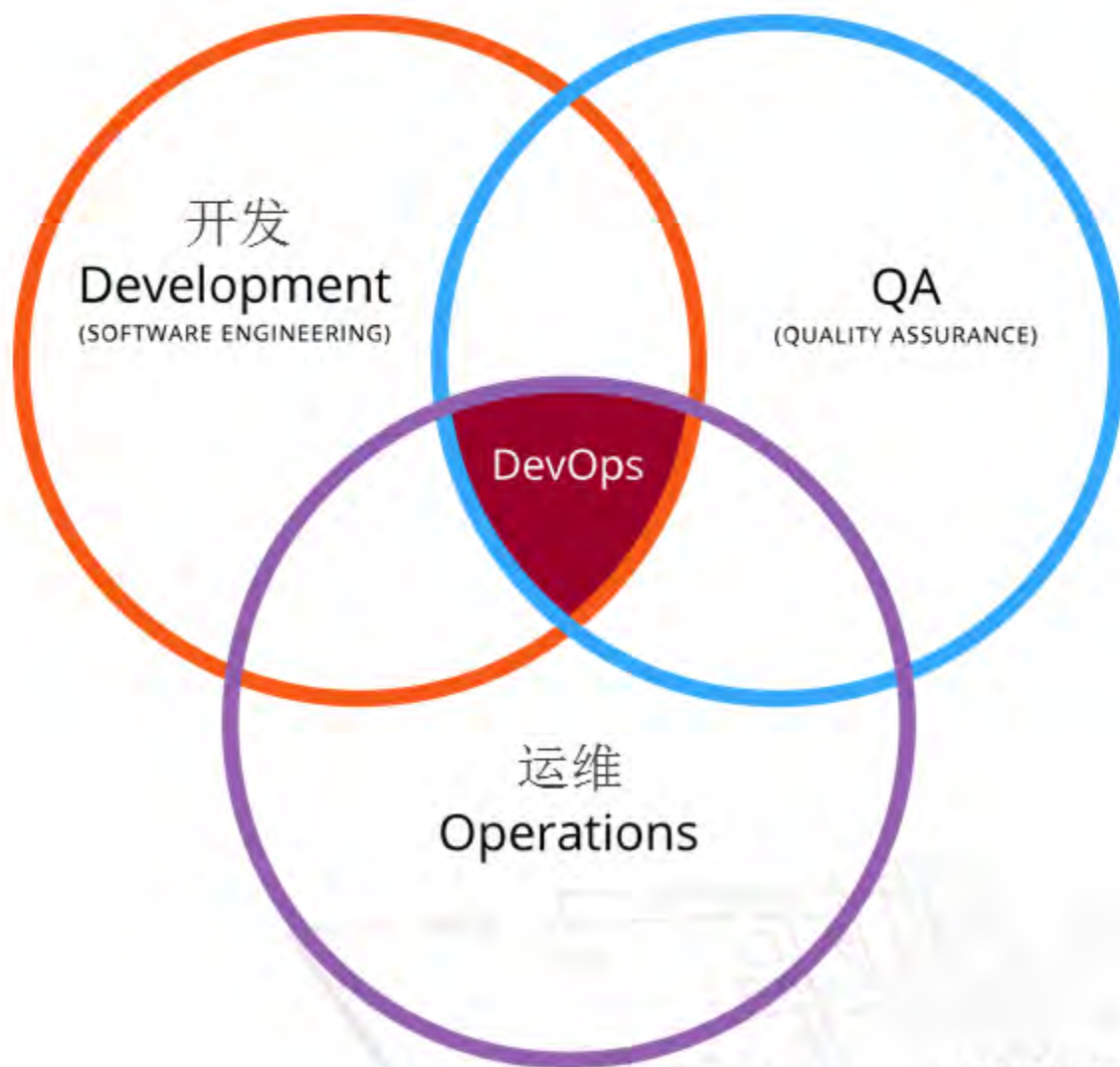
AGILE 2005年



DEVOPS 2009年



DevOps是什么？



DevOps 工具链

协作

构建

测试

部署

运行



https://www.pivotaltracker.com

DevOps的发展现状

2017 DevOps 现状调查报告

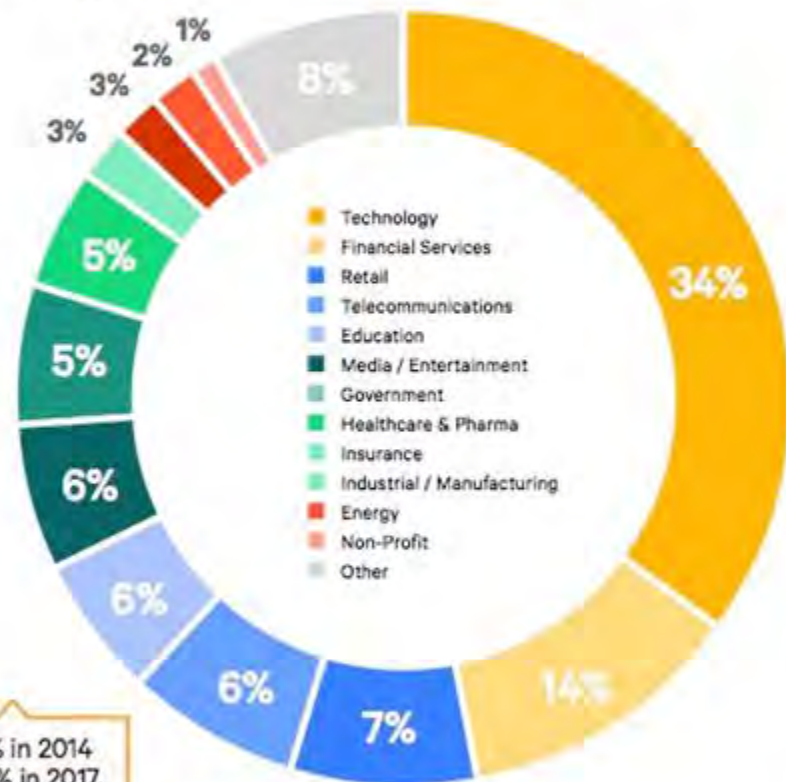
Presented by:



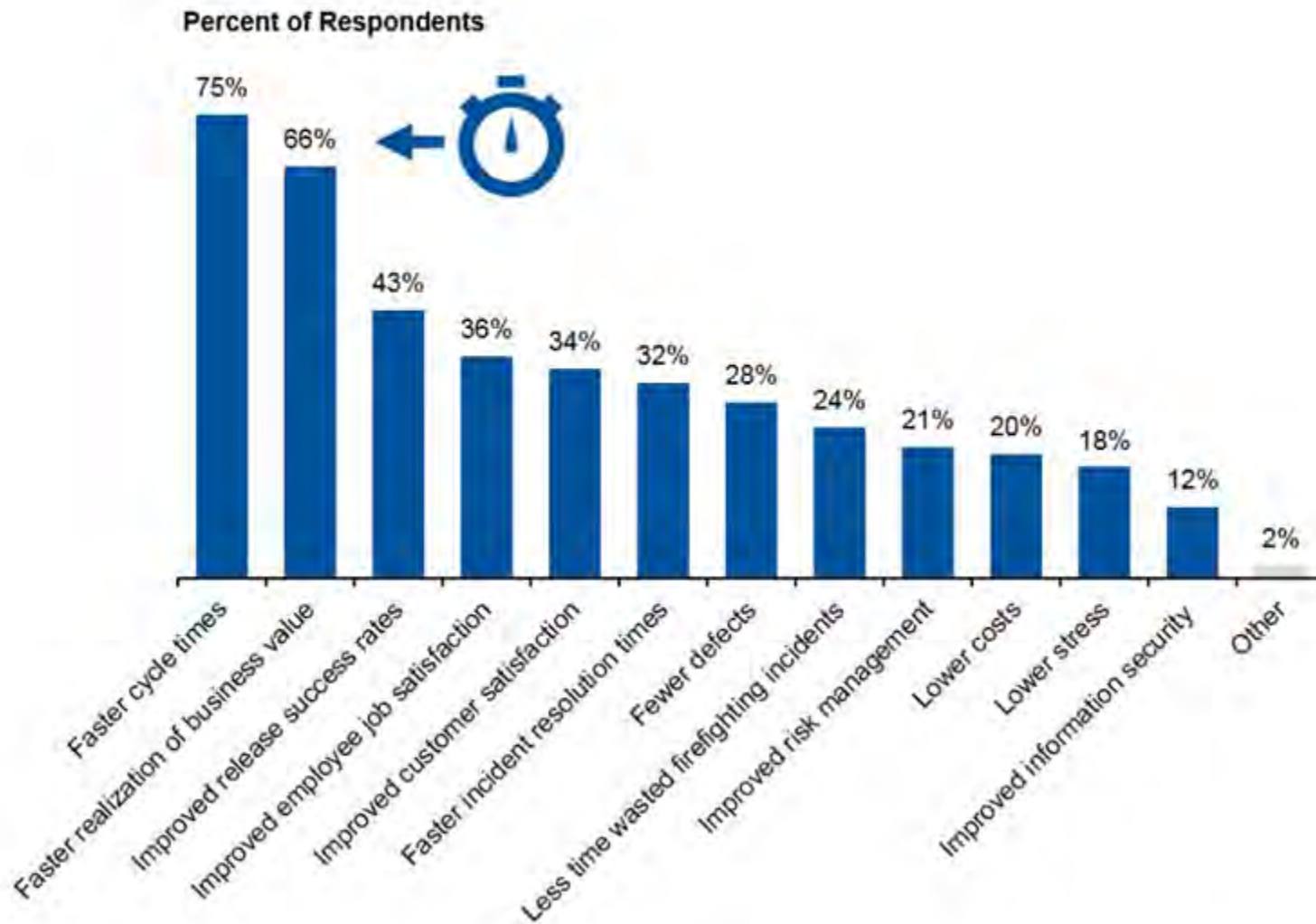
世界500强的DevOps员工
比前三年增长了两倍

DevOps teams increased from 16% in 2014 to 19% in 2015 to 22% in 2016 to 27% in 2017.

Industry



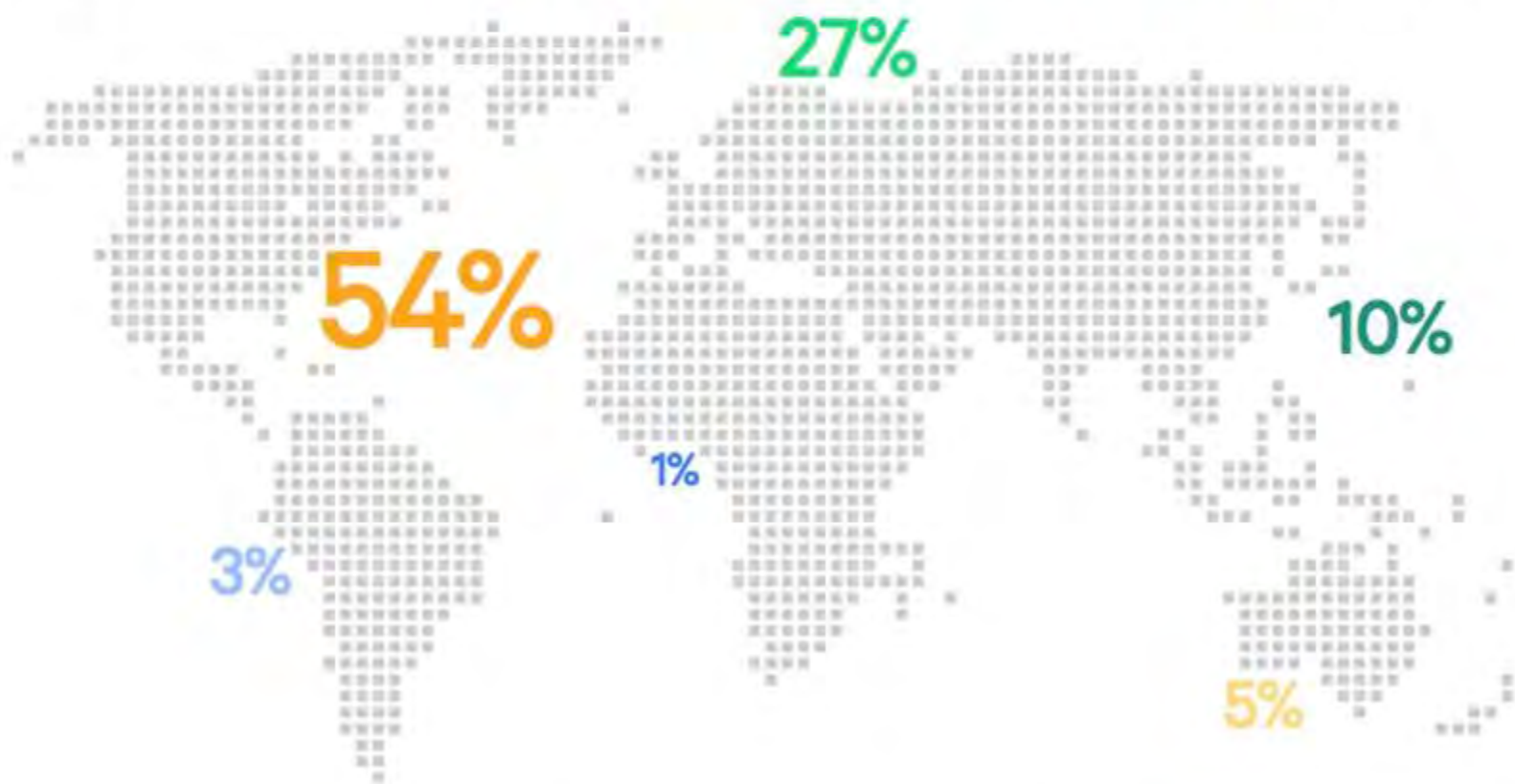
DevOps 的收益



DevOps全球开发者分布

Region

North America South America Europe + Russia Asia Africa Australia + New Zealand



全球最超前做DevOps的公司

Google

NETFLIX

ORACLE®

CISCO

Tencent 腾讯

HUAWEI

Alibaba Group
阿里巴巴集团

mobike
摩拜单车

自助式DevOps

自定义开源持续交付流水线

模版化交付任务
统一接入
跨团队数据分析

可视化报表
可扩展容器化管理





全球领先DevOps实践者

500种微服务

每天发布4000次



bakery



ORACLE®

ORACLE

40k
developers

59
countries

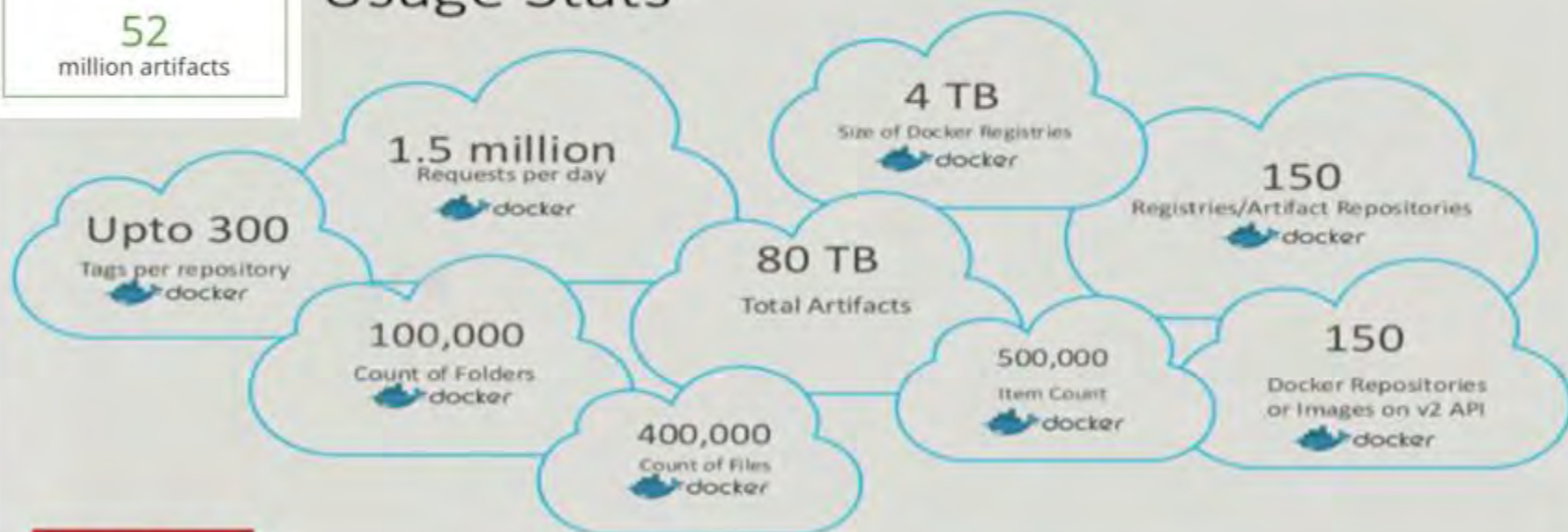
420k
customers

460
products

52
million artifacts

甲骨文的规模

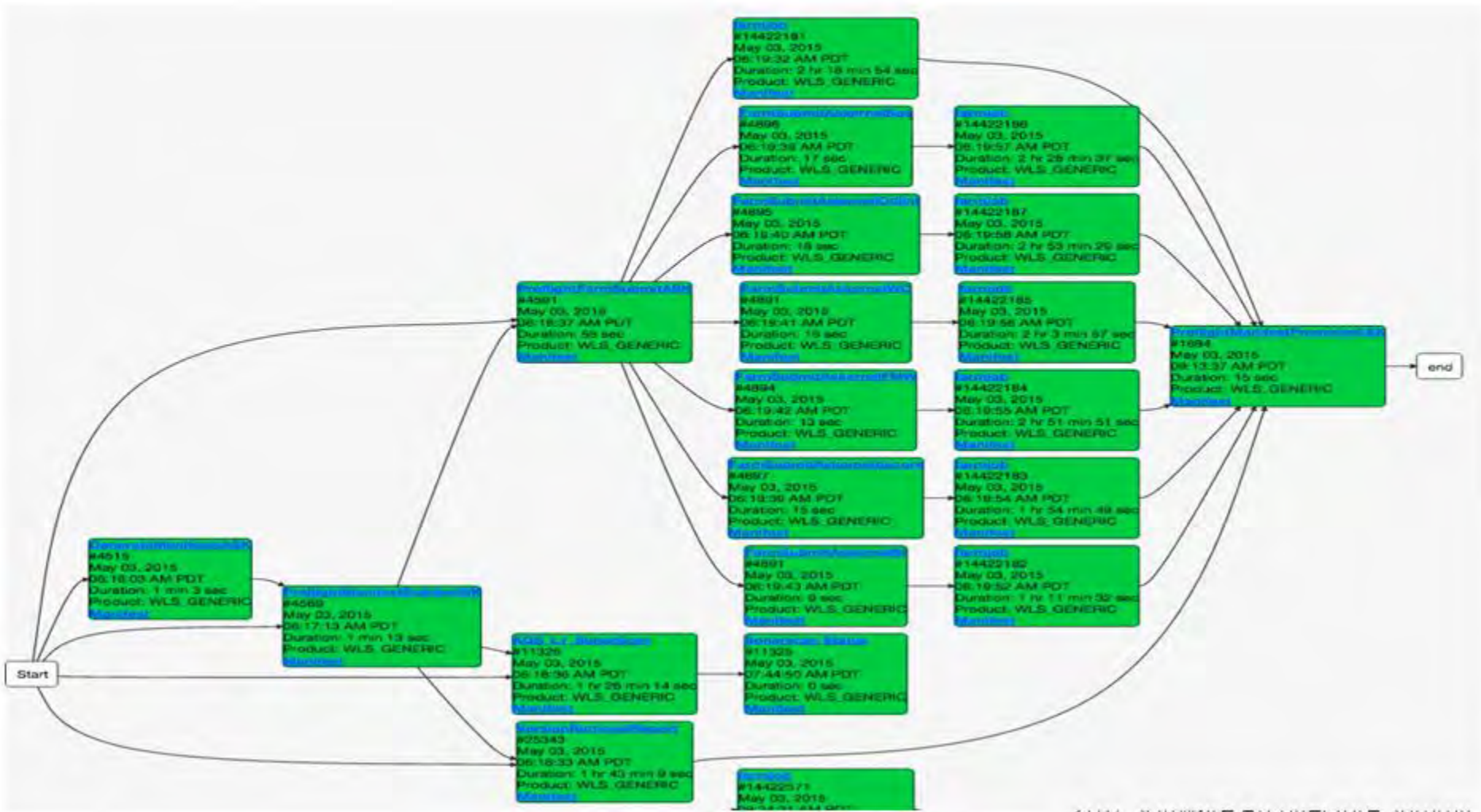
Usage Stats



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

甲骨文复杂流水线



集中工件管理

Artifact Diversity

2015

2016

Repository
Types

Maven



Databases

ORACLE DATABASE 12c

ORACLE DATABASE 12c

Management



CHEF

Operating
system

ORACLE
LINUX

ORACLE
LINUX

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

2017 Software Architecture Summit

可视化流水线

OrchViewer

OrchID: 883 - InstanceID: 1820965

[View Old Visualization](#)

Instance Information

Duration:
n/a

State:
ERROR

Start Event:
{
 CIM_TYPE: "SBE",
 RELEASE_BRANCH:
 "release-16-4-5",
 EM_ORACLE_HOME:
 "/u01/ops/em/core/12.1.0.1",
 EMAGENT_IMAGE_FILE:
 http://bigfiles.us.oracle.com/emagent/1210101/emagent1210101.jar,
 POOL: "pool12",
 MGMT_VM_NAME:
 "bdpmgmtpl1",
 SPARK_ASSEMBLY: "spark-assembly-1.6.1",
}

[Retrigger From](#) [Retrigger](#)

[Abort](#)

1820965

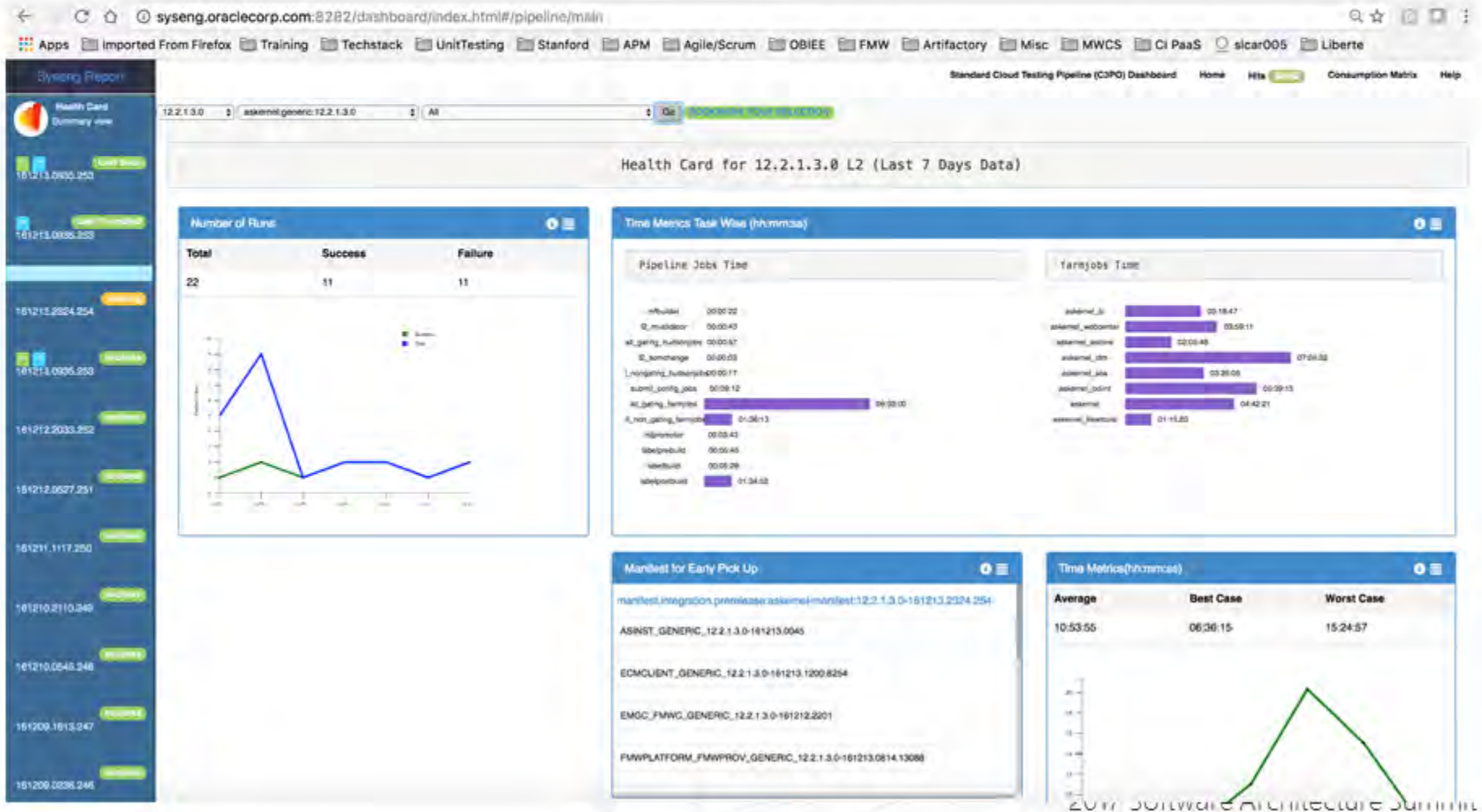
Graph View [JSON View](#)

■ Success ■ Failure ■ Running ■ Waiting ■ Skipped

```
graph LR; Start[Start] --> T1[EMAgentDownload]; T1 --> T2[EMAgentInstall]; T2 --> T3[EMAgentStart]; T3 --> T4[EMAgentCheck];
```

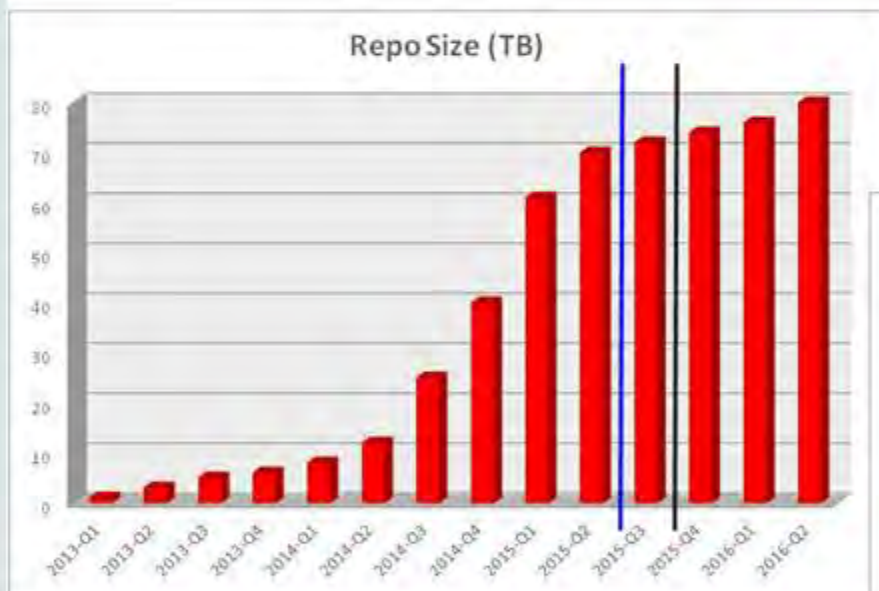
The visualization shows a linear sequence of five nodes connected by arrows. The first node is a white box labeled 'Start'. The second, third, and fourth nodes are green boxes with white text, indicating success. The fifth node is a red box with white text, indicating failure. The nodes contain details such as task names, start times, and durations.

统计与报表



甲骨文平台扩容

Ramp Up



- Continuous purging/clean-up began
- Quota Management put in place via Carson

