

DevOps管理专家

基于DevOps的应用运维实践方案

彭鲤航，优维科技（深圳）有限公司



个人 简介

彭鲤航 联合创始人&COO

黎明网络 1998

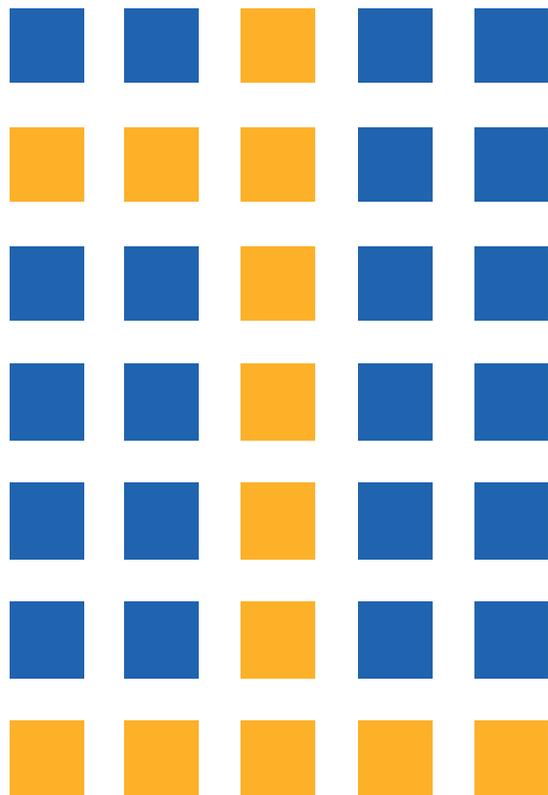
最内第一批的网络工程师及互联网运维

腾讯科技 2006

曾负责腾讯SNG、IEG等多个大型业务的运维及团队管理。

优维科技 2015

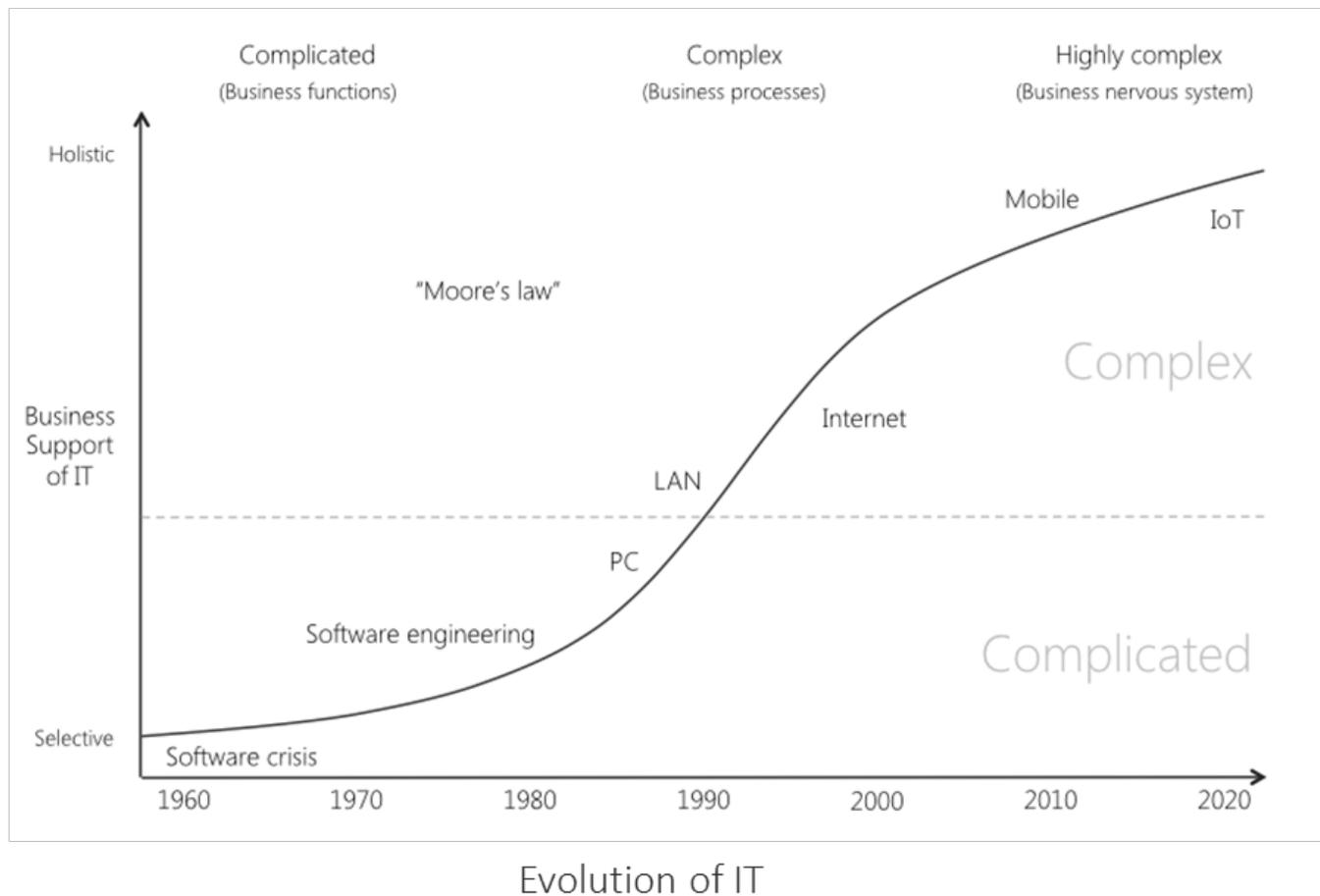
负责优维科技的项目交付团队管理



为什么需要DevOps ?



IT已经从企业内部的一种运营工具，逐步演进为一种竞争优势

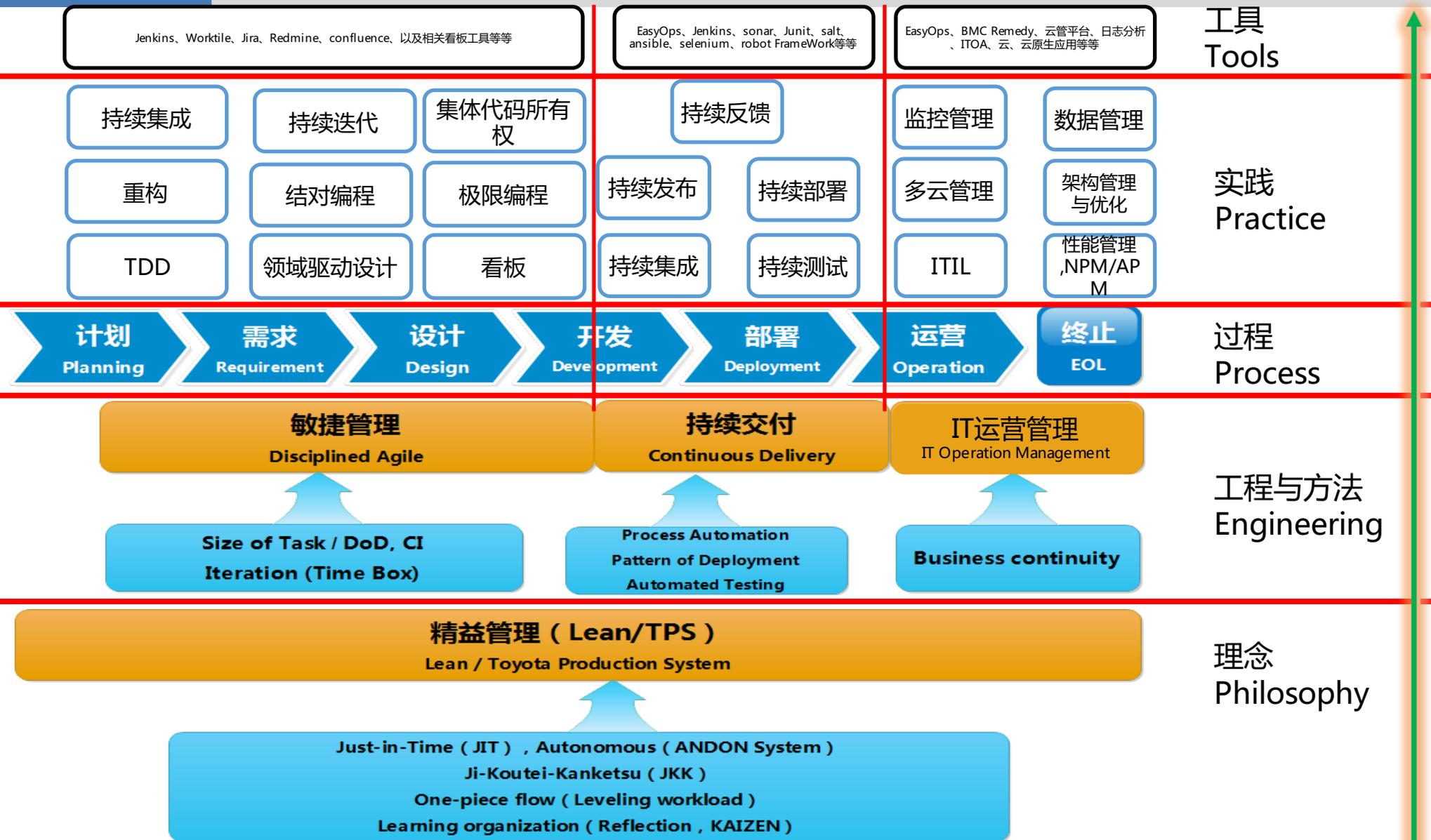


- 传统的软件工程方法
 - 系统的应用工程方法
 - 基于预测性
 - 重管控、结构化
 - 有时避免或拒绝合理变更

Volatility	Uncertainty	Complexity	Ambiguity
易变	不确定	复杂	模糊
- unexpected	- cause is unknown	- many interconnected components	- little or no information
- unstable	- outcome is unknown	- volume of information overwhelming	- unclear relationships
- of unknown duration	- impact is unknown		

- VUCA新常态下，IT的目标
 - 需要**最大化业务产出**
 - 强调适应性
 - 快速交付价值
 - 灵活响应变化

DevOps整体体系框架



- DevOps是一连串的工程实践的有机组合，其中包括敏捷管理、持续交付、IT服务管理等等。
- DevOps是关注整个业务/应用/服务生命周期的管理，把业务和IT的战略进行了对齐。
- DevOps以精益思想为基础，强调自动化、拉动式、“拒绝浪费、创造价值”等

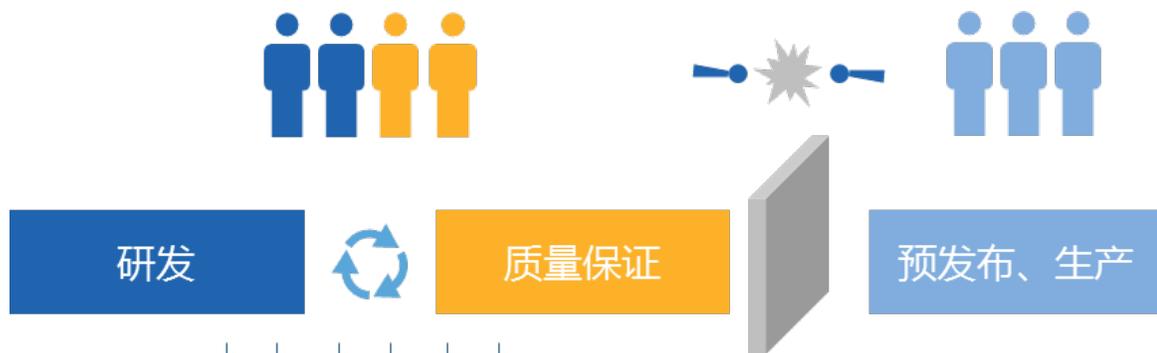
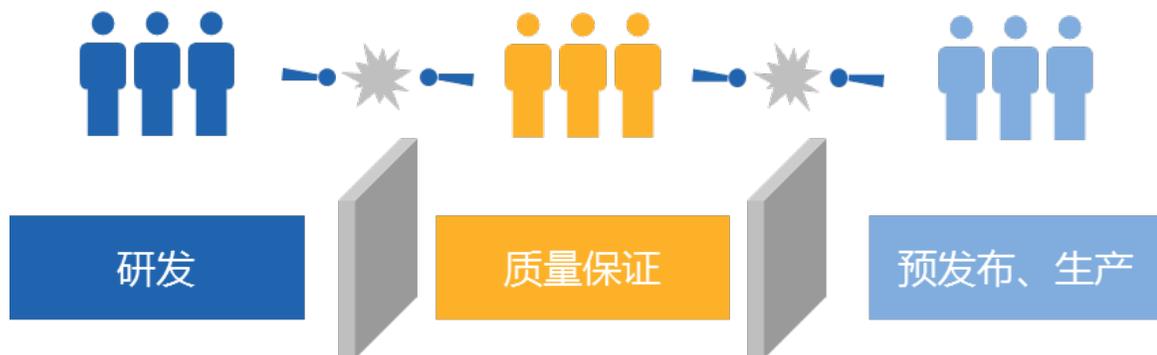


- 1、理念与价值先行
- 2、顶层设计与全局规划
- 3、Start Small，从小做起
- 4、构建IT元数据平台，驱动IT平台间整合
- 5、痛苦的事情优先解决
- 6、工具也是一种文化
- 7、组织二元性，加强落地力
- 8、价值拉动，而非事务驱动
- 9、平台+插件化=服务能力产品化，和组织一致
- 10、自动化别人，先自动化自己
- 11、持续交付是DevOps落地的最佳实践
- 12、IT运营管理驱动Ops能力建设
- 13、构建面向应用的最强管理驱动力
- 14、构建指标，驱动DevOps落地

注：老王有一个专题分享专门讨论了这14个实践原则。



瀑布流, Waterfall



敏捷迭代、Sprint、测试驱动研发

DevOps

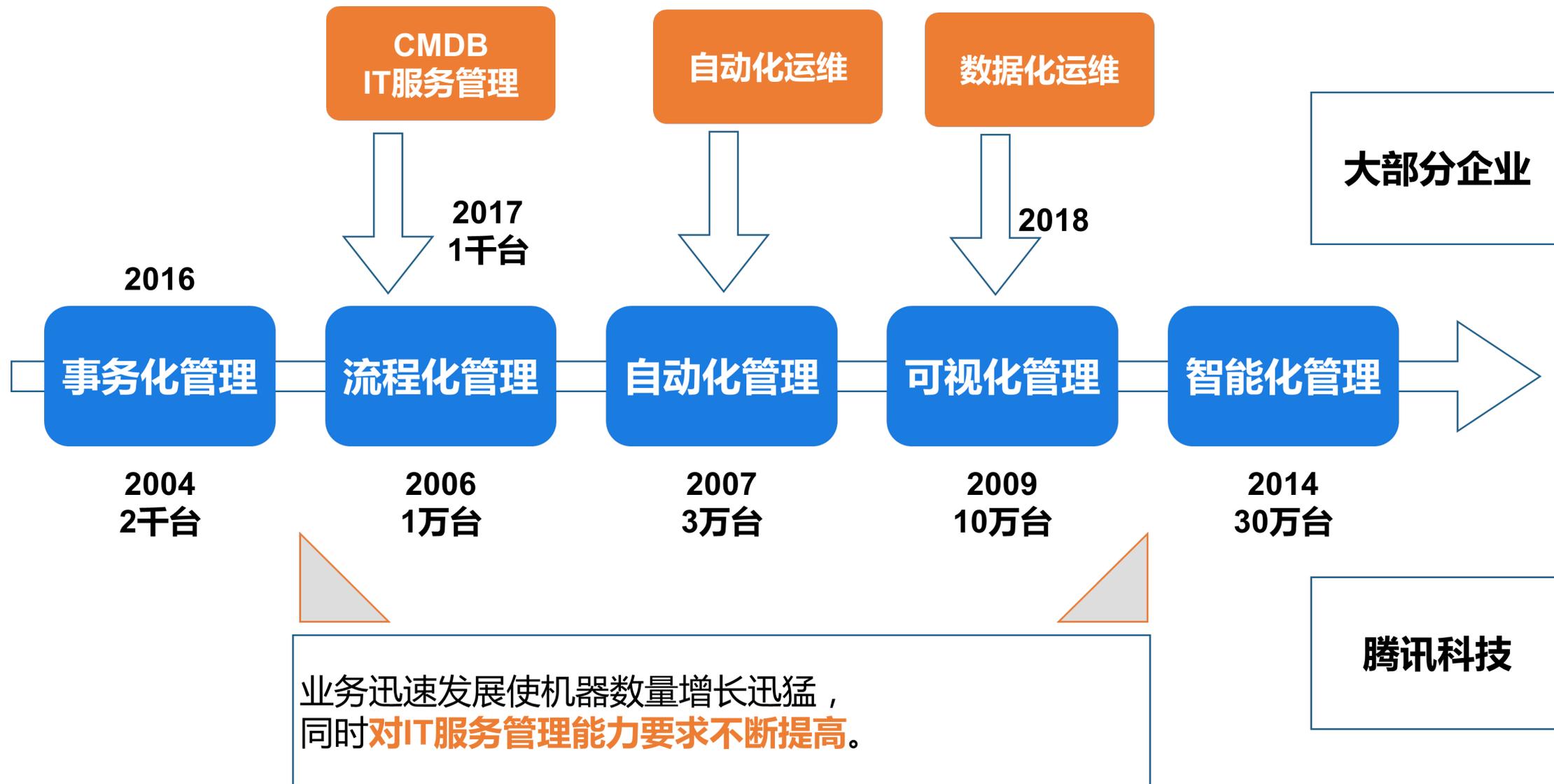


- DevOps让团队共享面向客户的价值、共享集成目标、共享质量责任

注：刘劲辉有一个专题讨论了基于DevOps的产品研发实践。

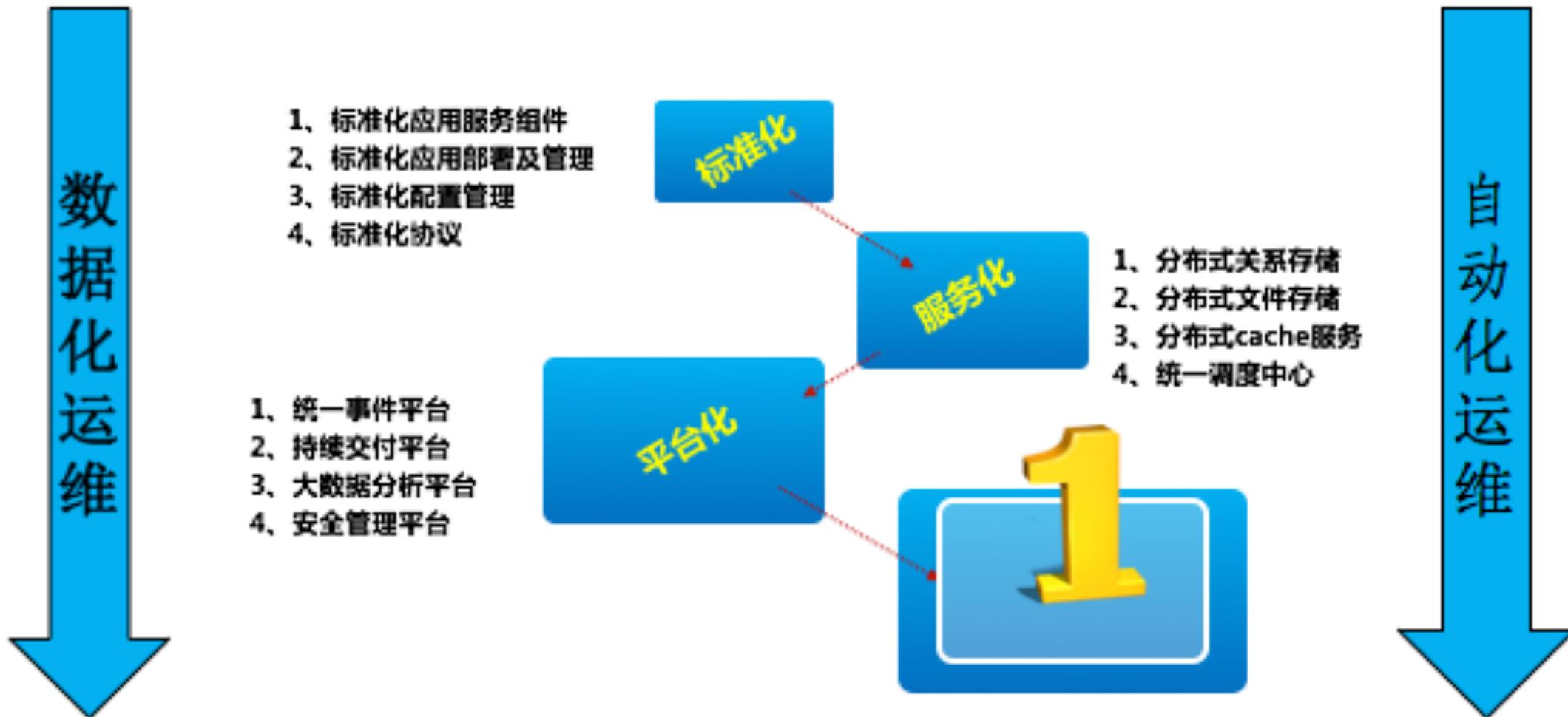


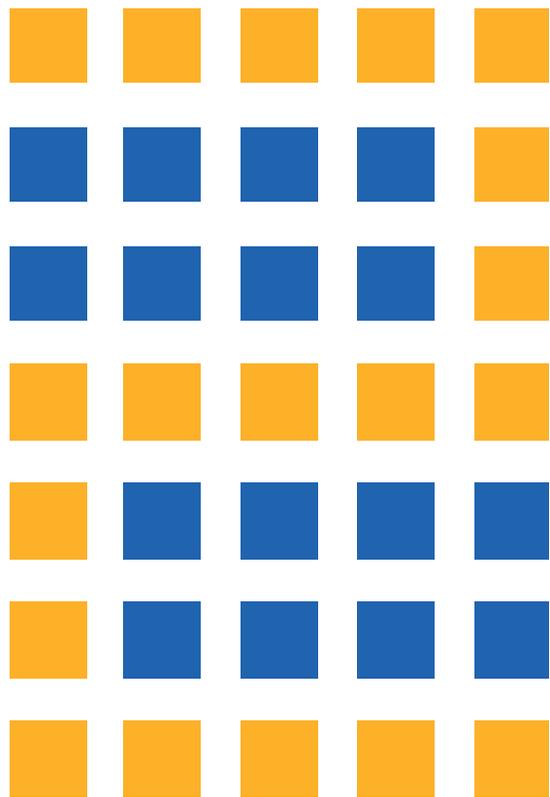
国内互联网运维的DevOps之路





运维能力建设的最佳实践

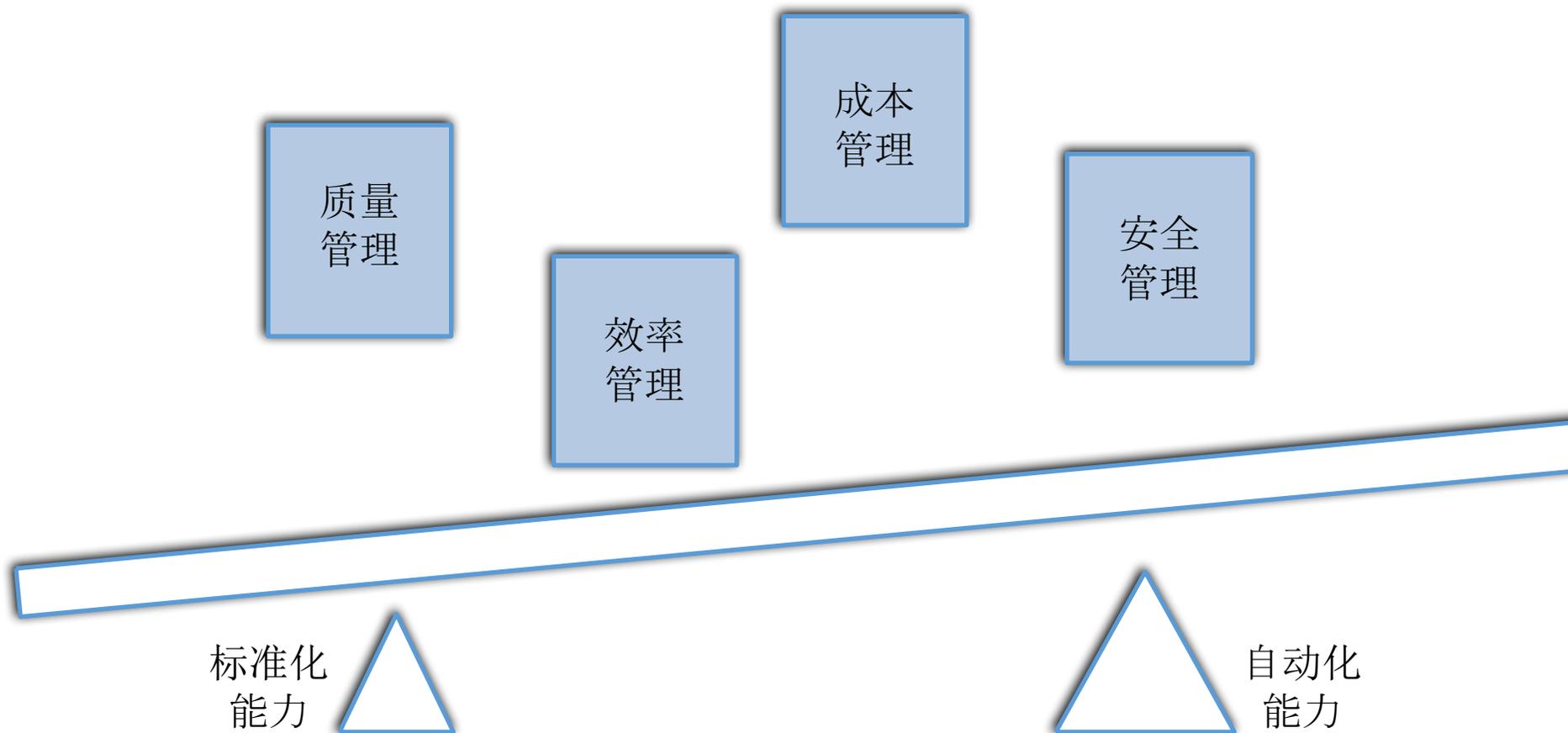




DevOps之标准化实现

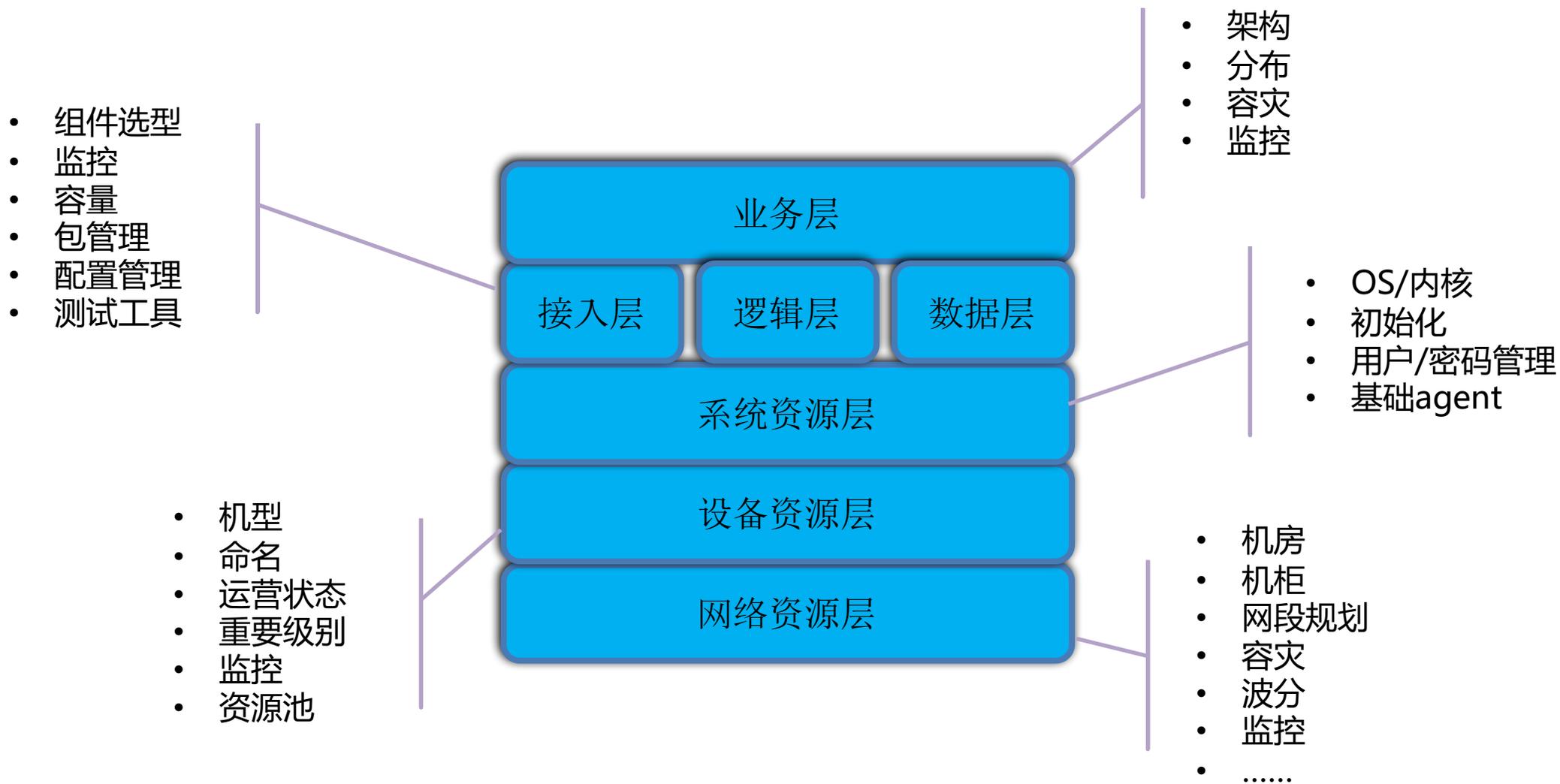


运维标准化的价值--IT运营管理的重要支点





DevOps之运维标准体系





标准化体系建设的方法论

标准化
不是
文档

标准化
不是
口号

标准化
不是
规章制度

V.S.

标准化
是
团队文化

标准化
是
生产过程

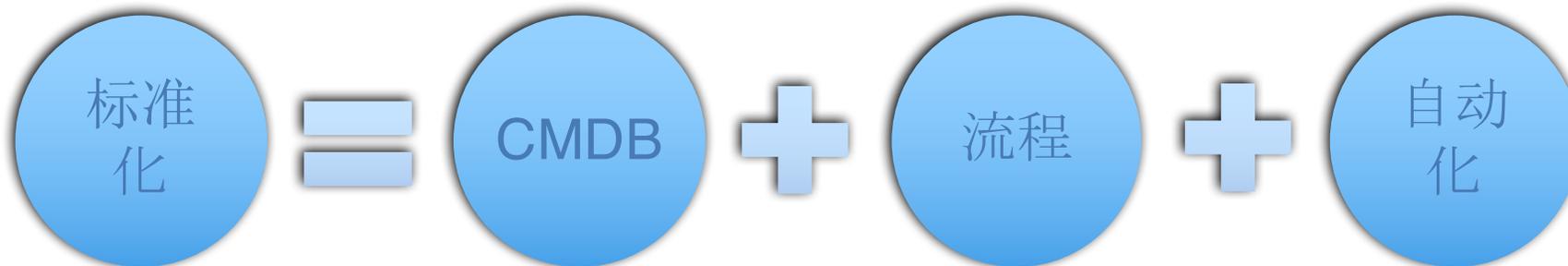
标准化
是
变更工具

标准化
是
CMDB



标准化体系建设的成功要素

让标准化建设融入生产过程，成为无法感知和不可或缺的关键环节。



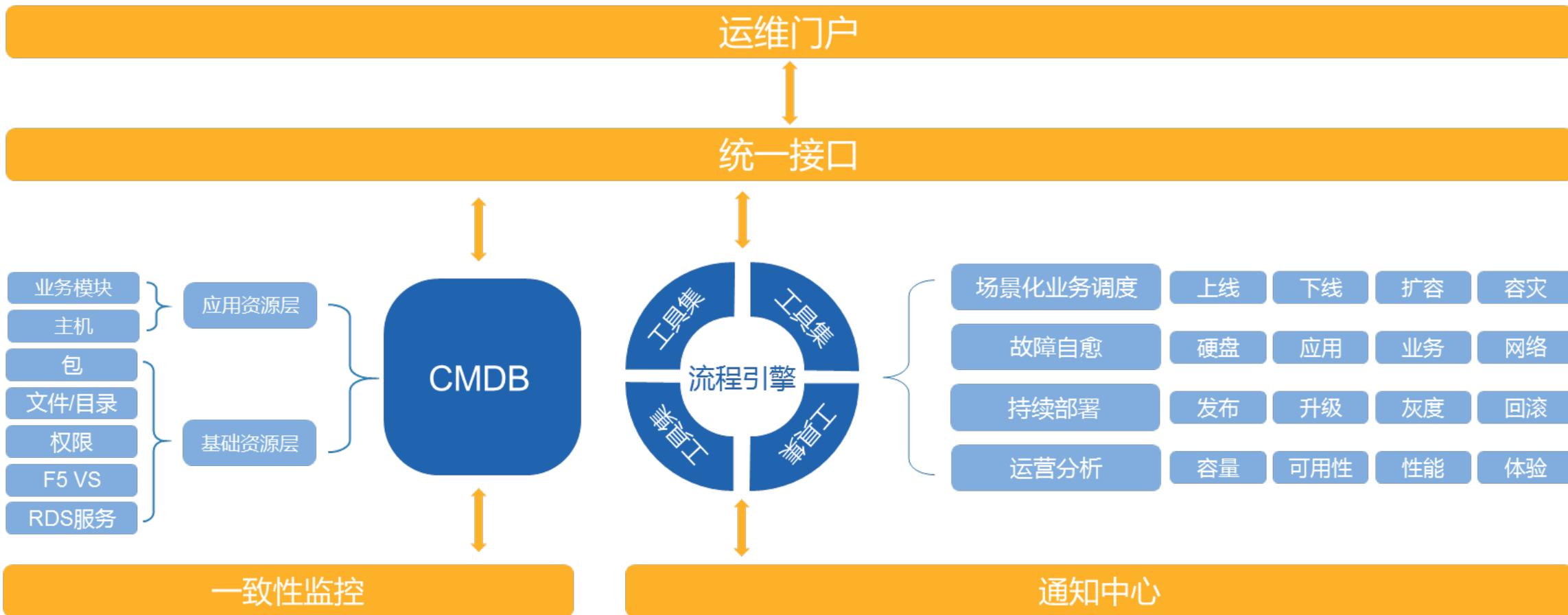
CMDB：承载和检验标准化的结果

流程：让标准化融入管理过程

自动化：让标准化融入生产过程



面向应用层的CMDB—90%的同类项目都失败了？



- 面向业务的配置管理，即业务信息管理平台，建立CMDB资源管理的核心驱动力
- 面向资源的资产管理，即CMDB

轻量级的New ITSM vs 传统的ITSM



传统的ITSM



- ❖ 强调过程标准规范
- ❖ 分阶段持续完善
- ❖ 重质量，轻效率
- ❖ 重建设，轻运营

New ITSM



- ❖ 重视运营大于建设
- ❖ 实现运营自动化，提高运行效率
- ❖ 加强持续改善，缩短周期

流程与实际变更动作的深度结合

从角色维度思考 效率与规范并存

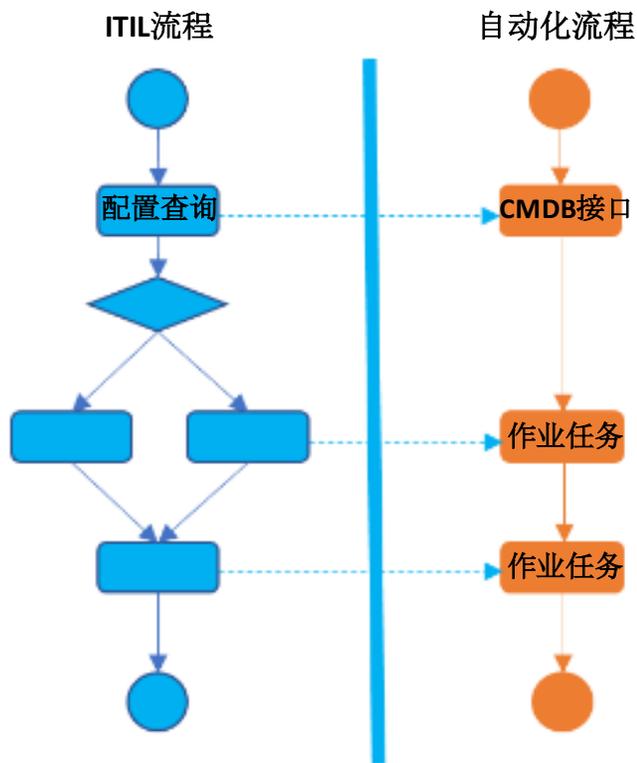
开放的RESTful风格的丰富接口

整合各大系统信息作统一呈现



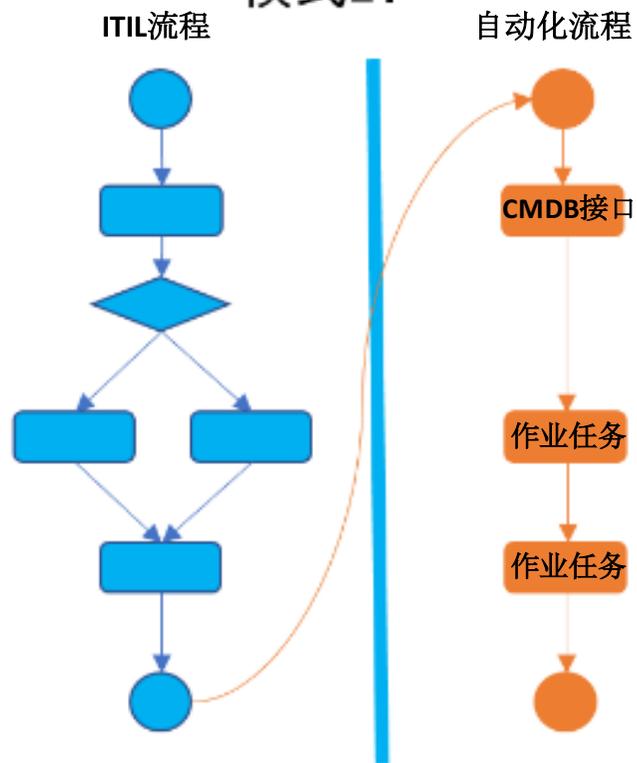
CMDB与ITSM流程之间的融合模式

模式1:



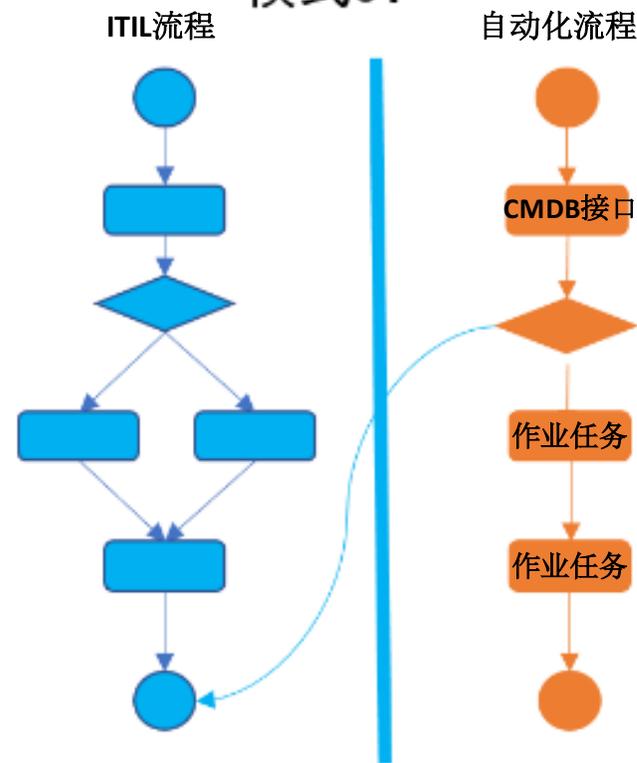
- 在线服务开通流程
- 资源管理流程

模式2:

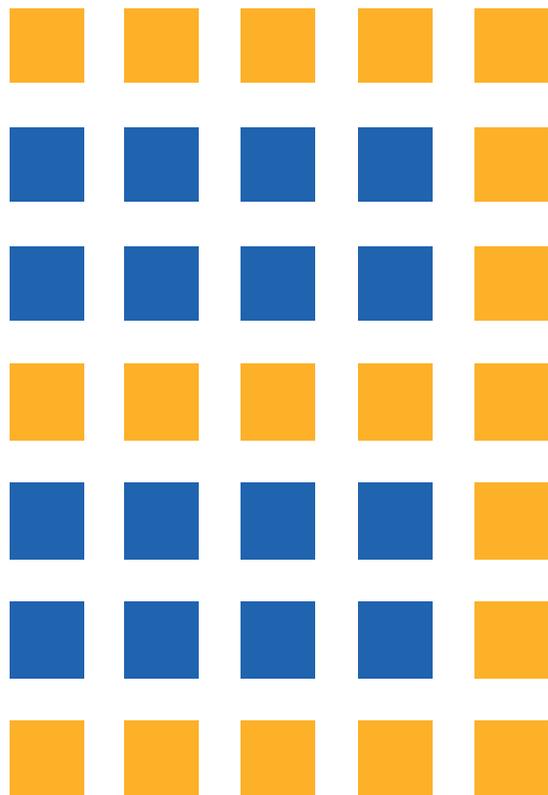


- 重大变更流程
- 高稳定性服务保障流程、变更流程

模式3:



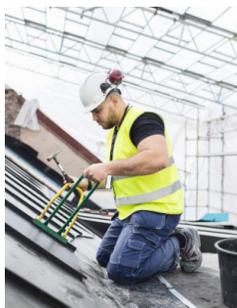
- 敏捷发布流程
- DevOps变更流程



DevOps之服务化实现

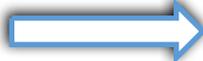


DevOps服务化的价值----实现能力升级和知识沉淀



手工服务阶段

能力升级



自动化服务阶段



沉淀



- 管理目的驱动
- 知识的收集和整理
- 转化过程中有失真
- 取决于使用的个人

- 业务场景驱动
- 知识的积累和沉淀
- 转化过程中无失真
- 摆脱对人的依赖

作业平台 + 新建工具 工具商店 查看历史任务

按名称搜索

经常使用的工具 AutoDiscover AWS ORACLE Poc-Tools Shadow VMware Tools weblogic
Windows常用命令 基础工具 常用命令 打包上传 推荐工具 故障排查 构建工具 系统管理 网络管理
部署 默认

AutoDiscover 编辑

 激活网段所有IP	 采集主机与交换机端口 连接关系	 采集主机存储HBAW WN和存储序列号	 采集存储设备信息	 采集本机网段活跃IP- MAC关系
 采集网络设备OS版本	 采集网络设备端口信息	 采集虚拟机与主机机关 系	 Jenkins构建	 Jenkins触发构建工具
 查询主机名	 查询安装应用	 查询本机服务	 查询机器信息	 查询磁盘空间



DevOps服务化的价值---一些重要的场景

01

发布
管理

02

变更
管理

03

配置
管理

04

故障
处理

05

影响
分析

06

数据库
维护

07

数据
备份

08

域名
维护

09

日志
查询



服务化体系建设的方法论---先上线再优化

快速尝试

- 没有完美的方案，只有可用和不可用的方案
- 先用起来再慢慢优化
- 让服务的最终用户来投票

实现闭环

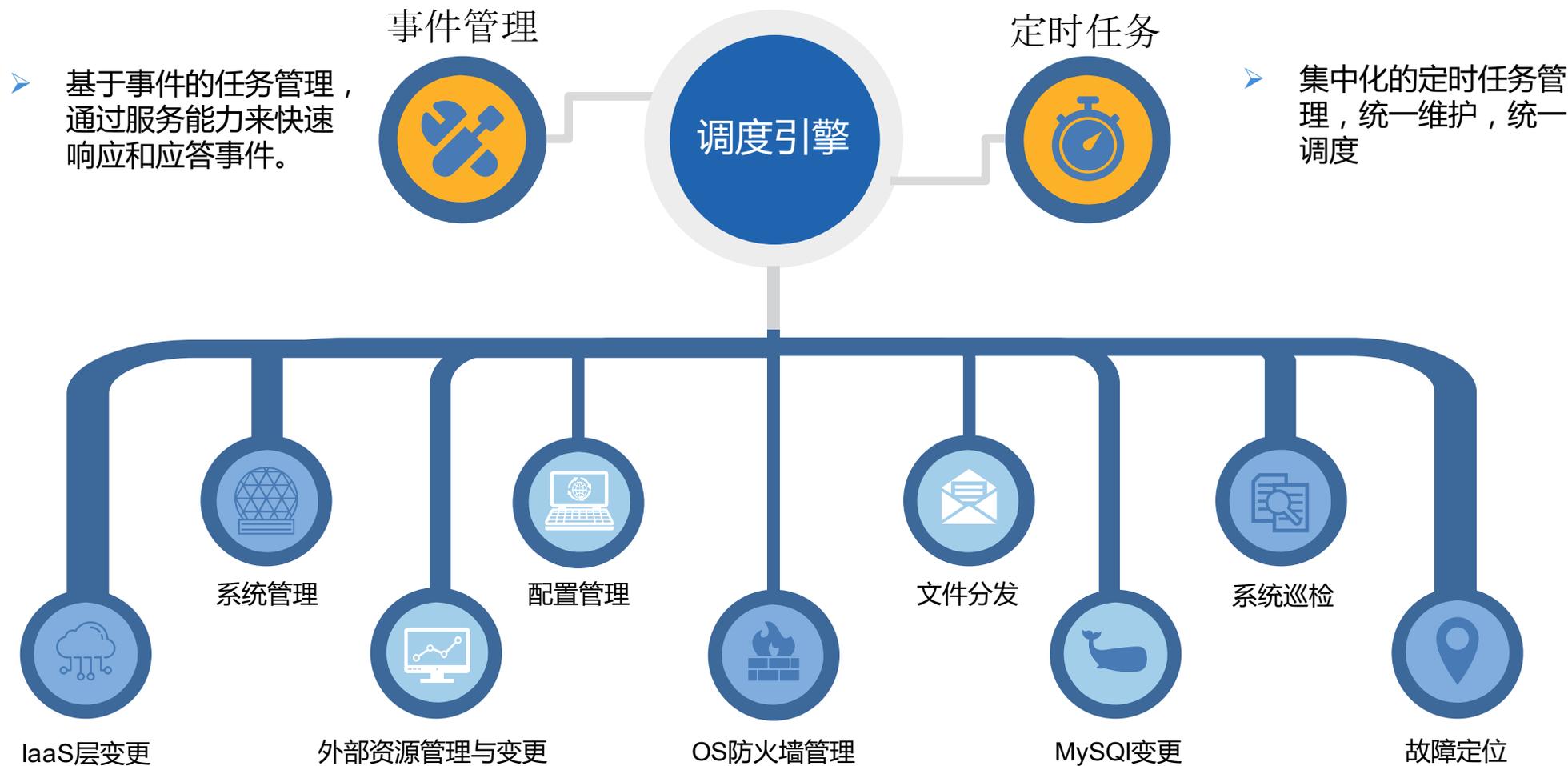
- 能够面向完整的应用场景
- 通过关联服务的协同，来实现能力闭环

从0到1 再从1到N

- 运维场景很复杂，需要分而治之
- 每个服务尽量简单和独立
- 一定要提供服务接口（输入和输出）

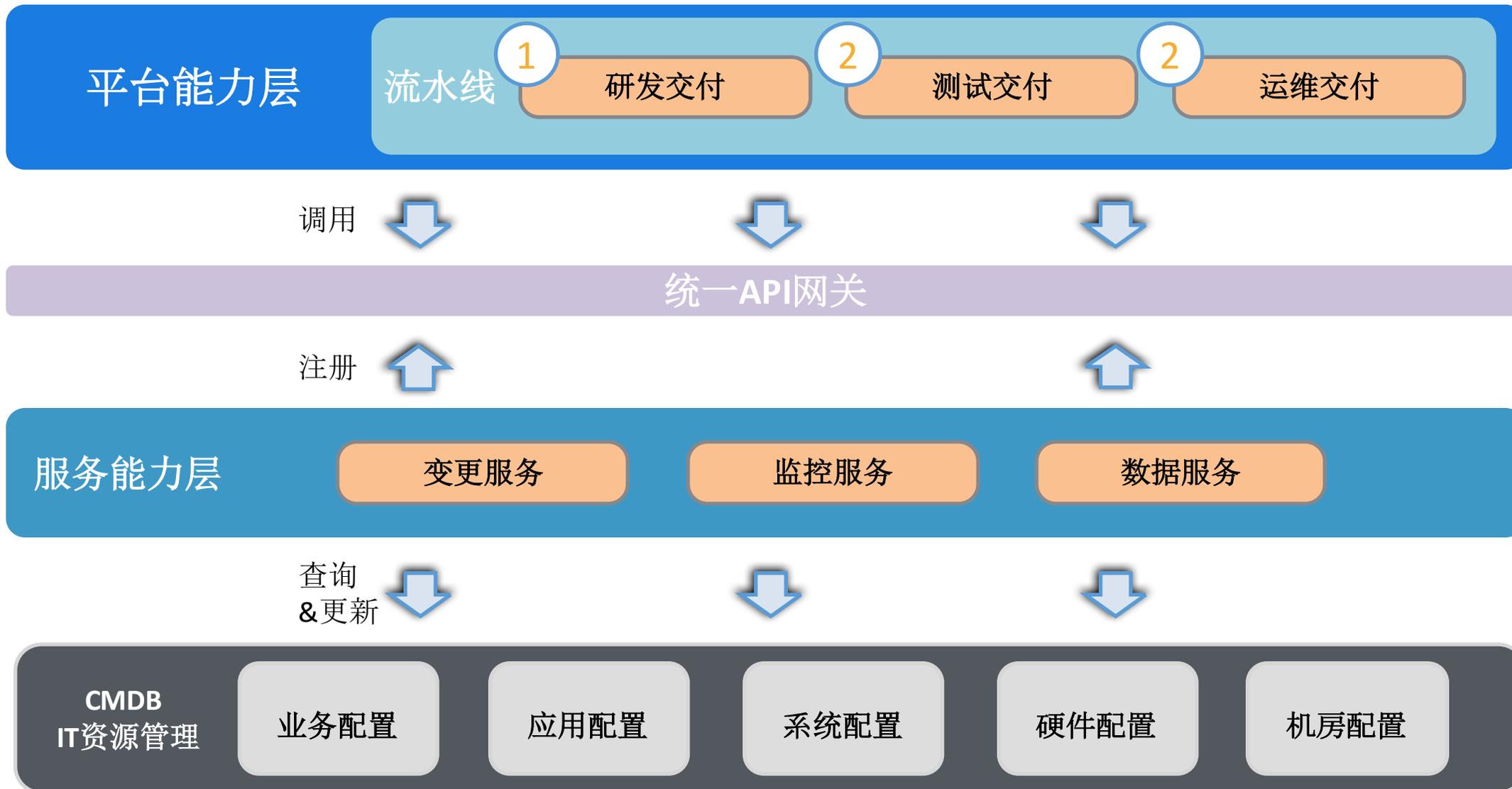


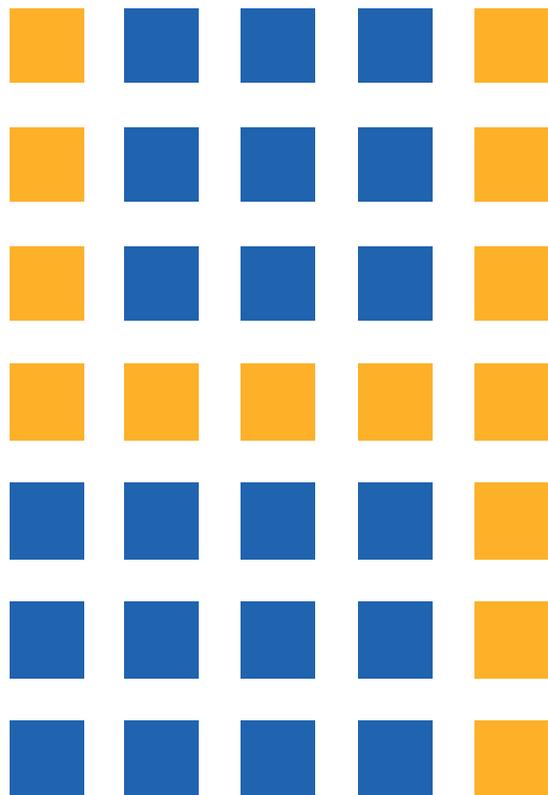
服务之间的协同和调度管理机制





服务能力的体系架构设计

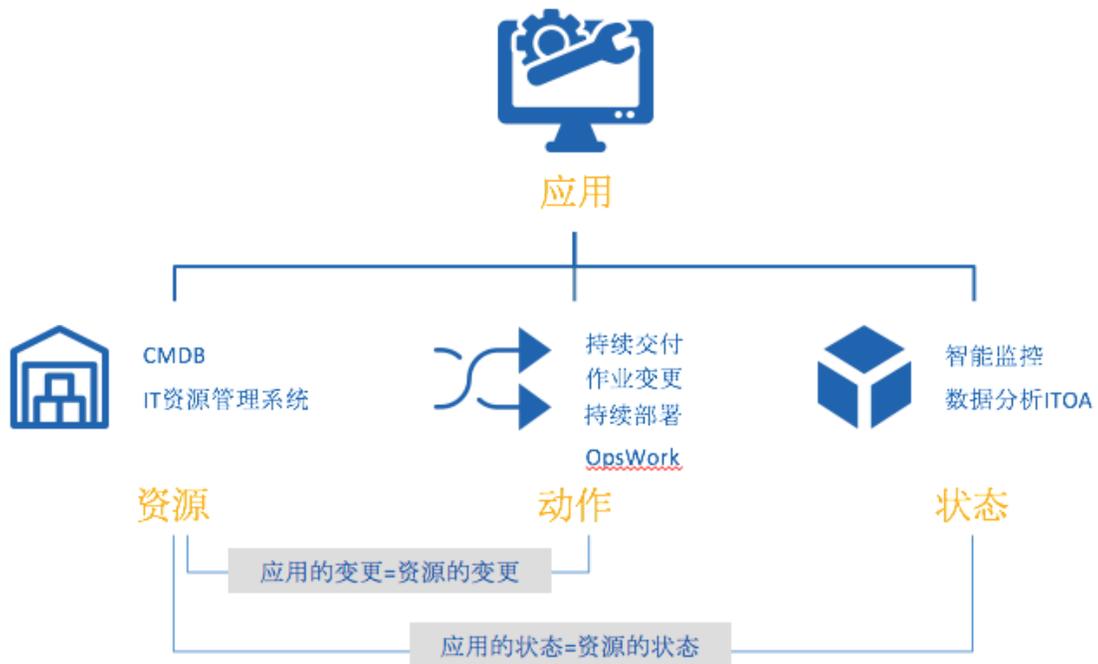




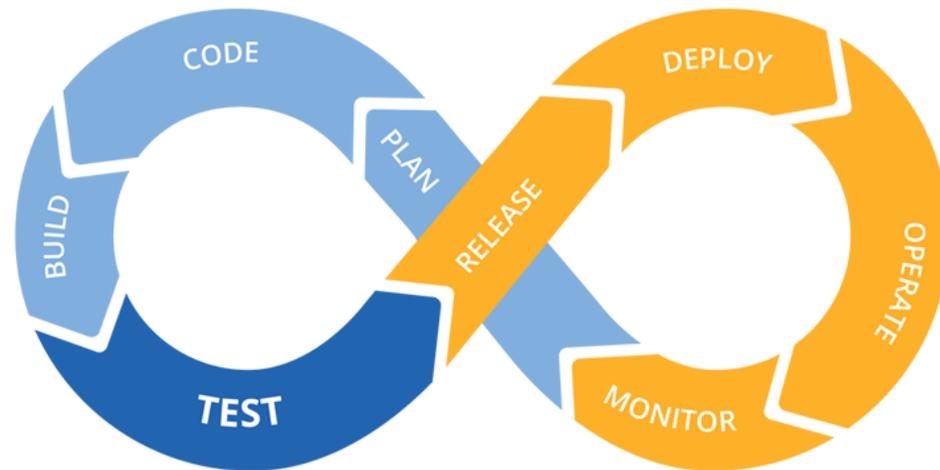
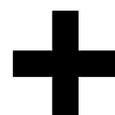
DevOps之平台化实现



打造面向应用的全生命周期管理



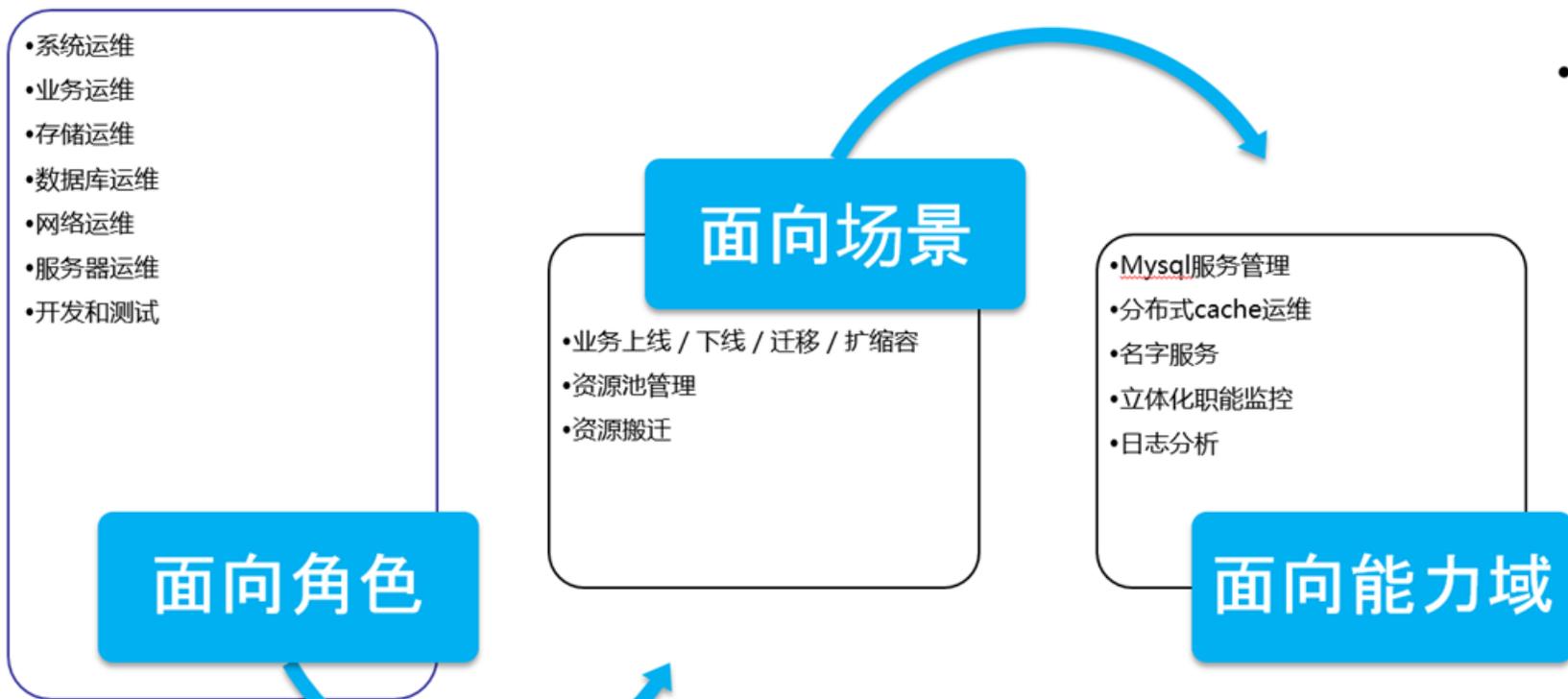
应用管理



生命周期管理



从运维管理向运营管理转换



- 全栈的运维平台，是要考虑各种角色 / 场景 / 能力下的交付需求，面向能力域的交付能力是一种专业化的服务能力交付。



- 价值流分析法是基于场景的分析，端到端的交付实现。



面向业务，提升IT的交付能力



角色：各类运维角色，网络、存储、服务器等等

场景：运维各类自动化、配置、变更等

能力：
面向资源层的自动化能力，包括IaaS和PaaS层自动化

产品：云管理平台CMP



角色：Dev、Test、Ops

场景：持续部署、配置管理、灰度发布、端到端交付

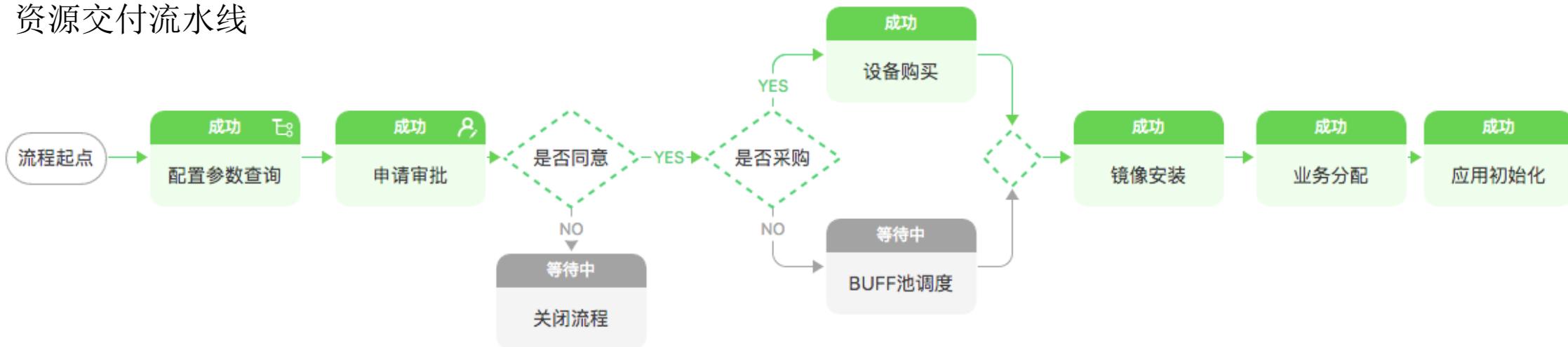
能力：面向应用的自动化能力，包括持续交付的自动化和运营维护过程自动化

产品：DevOps平台

构建资源的持续交付能力



资源交付流水线

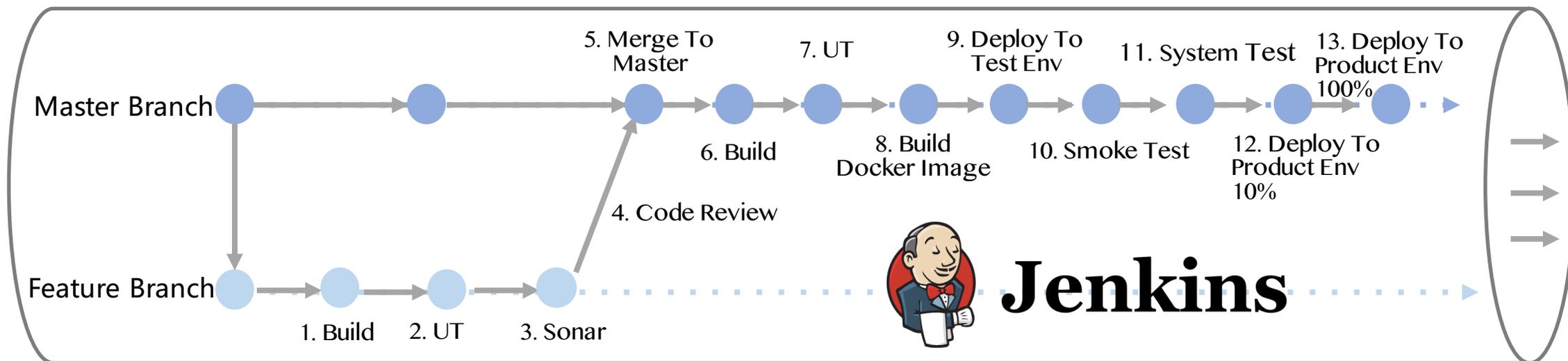


- 通过资源的快速生产、分配能力来加快设备生命周期的循环。
- 通过快速循环来淘汰过时和非标准的资源类型
- 资源快速交付意味着更低的成本





构建应用的持续交付能力



GitLab



Nexus



sonarqube



HARBOR™



Apache JMeter™



kubernetes

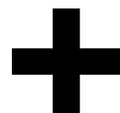
elastic



构建业务的持续反馈能力



端到端的数据采集能力



智能的数据分析处理闭环

IT运营管理平台的整体架构



运维统一门户

运维服务目录

研发者目录

任务中心

状态中心

调度编排系统

Web类业务

语音类业务

游戏类业务

互联网金融业务

金融类业务

运维服务器API (由各个服务平台提供)

ITIL服务管理 (基础)

HelpDesk

事件管理

问题管理

突发事件管理

变更管理

配置管理

IT服务管理 (高级)

容量管理

可用性管理

连续性管理

运维者服务平台

机房管理平台

交换机管理

LVS/F5管理

DNS管理平台

云管平台CMP

作业平台/配置管理

研发者服务平台

分布式NoSql存储

分布式Cache服务

统一RDS服务

统一文件、块、文件、
储存

名字服务

分布式消息队列服务

持续交付平台

持续构建

持续审查

持续测试

持续发布

持续反馈

统一监控平台

用户侧监控平台

服务级监控平台

应用级监控平台

系统及组件级监控平台

基础设施级监控平台

ITOA分析平台

业务质量监控品台

机房质量平台

用户体验/舆情分析

容量分析

日志分析平台

业务可用性分析

IT服务中心

系统管理员

网络管理员

数据库管理员DBA

驻场运维

存储运维

.....

流程管理

权限管理

CMDB统一接口层 (API)

配置管理CMDB

逻辑资源管理 (组织/人/代理商)

组织管理 服务商 代理商 服务商
人员管理 地址管理 角色管理 权限管理

应用资源管理

应用包 流程 包配置 工具
crontab 权限 进程段口 系统配置

基础资源管理

服务器 机房 网络设置 IP
机架 域名 储存 虚拟机

对象关系

对象关系

IaaS 基础设施平台

OpenStack

公有云

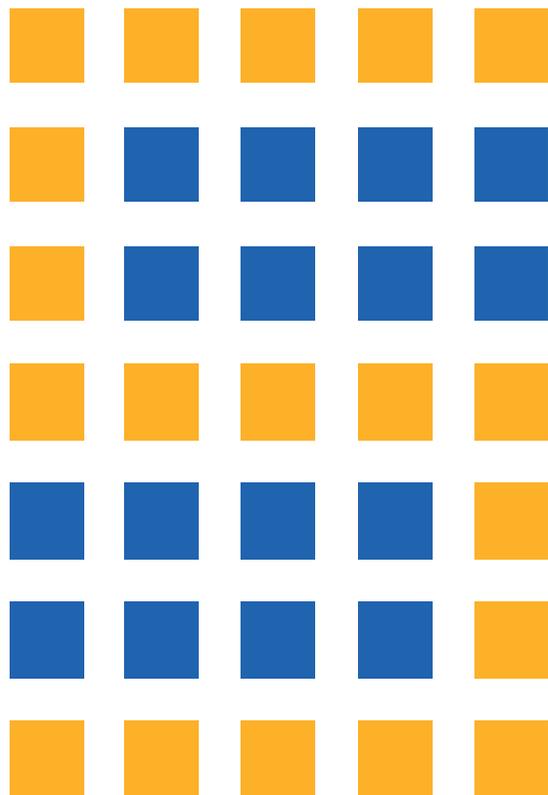
私有云

服务器

网络

储存

机房



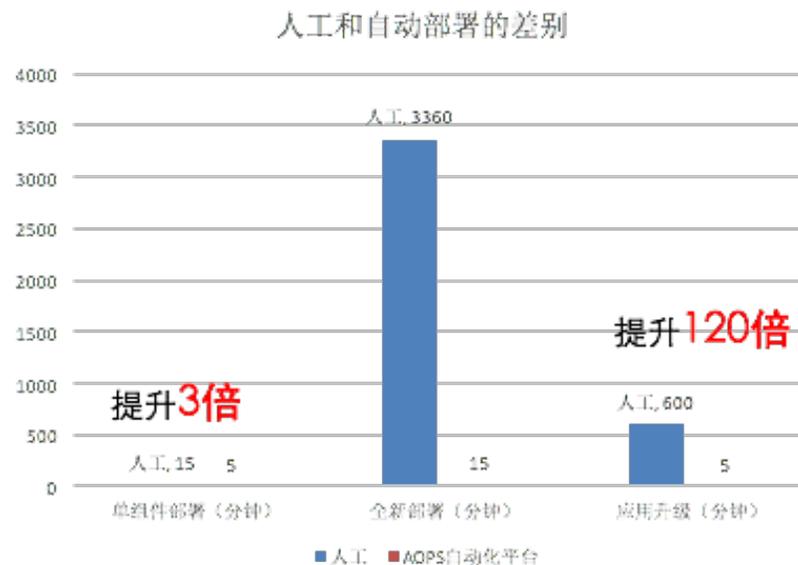
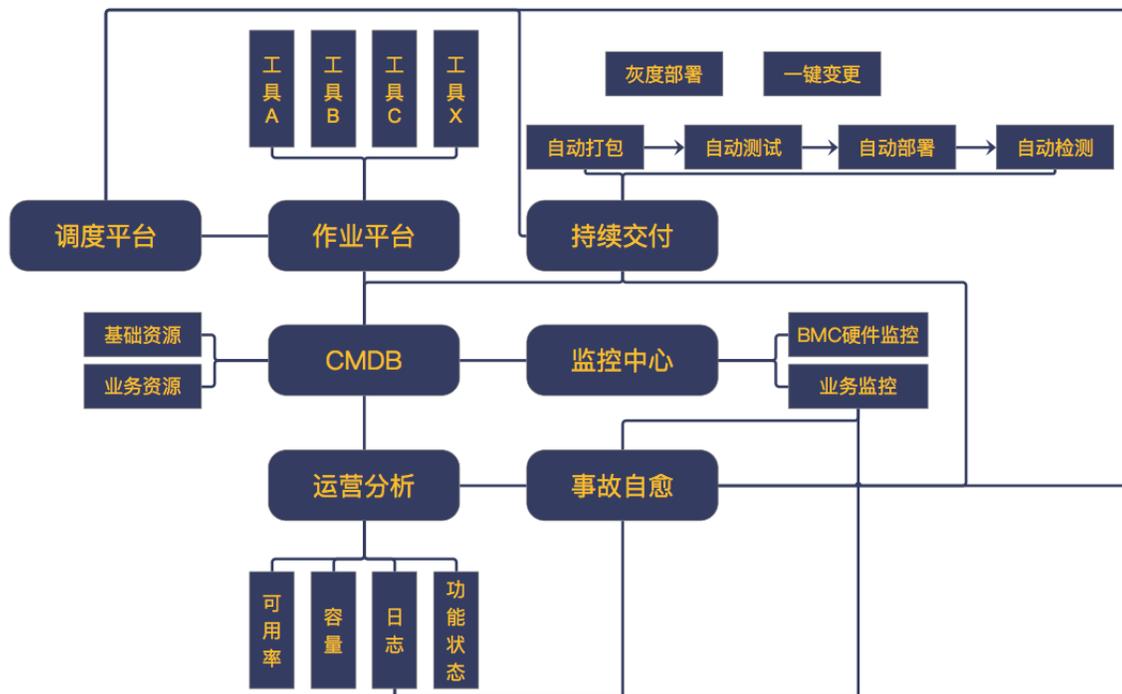
EASYOPS及企业实践



EASYOPS的主要客户



物流客户的EASYOPS建设方案



- ❖ 快递业务部署接入EasyOps后：
 - 一键自动并行部署ECS系统37个组件，批量上传只需2分钟以内，部署仅需 3 分钟；单个系统部署过程不超过15分钟。所有的核心系统都完成标准化，且测试和生产完成了接入AOPS！
- ❖ 接入前：
 - 需要手工准备环境，组件需要逐个部署，需要数小时
 - 组件上传需要3-5分钟
 - 每次环境准备与部署需要数小时，现有所有系统部署一次甚至需要一周



项目建设分为三个阶段



一个月：

- **CMDB**模型规划
- 配置信息导入
- 自动发现能力分析
- 应用程序解耦分析
- 程序包构建
- 运维场景分析
- 日志格式标准化
- 流程接口标准化
- 监控平台接口标准化
- 统一登陆协议确定

二个月：

- 应用程序改造
- 运维工具开发
- 运维流水线开发
- **CMDB**插件开发
- 监控及恢复插件开发
- 已有运维平台接口改造
- 已有监控平台接口改造
- 已有流程平台的改造

二个月：

- 平台正式上线
- 统一认证平台对接
- **BMC**监控平台对接
- **OA**流程平台对接
- **ELK**大数据平台对接
- 配置管理流程上线
- 资源交付流水线上线
- 应用交付流水线上线



项目实施时所面临的挑战和机会

挑战：

- 无标准化积累
- 系统有JENKINS+少量脚本工具+BMC监控
- 内网环境管控严格
- 日志无统一格式
- 人工CMDB
- 无IT流程支持
- 员工离职率高
- 发布需要停机
- 56个业务需要进行接入
- 1000台以上的机器要纳管

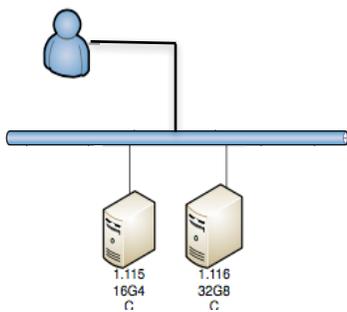
机会：

- 快递业务要上线
- 领导非常重视，高级架构师作为项目经理参与建设
- 项目完整：咨询服务+产品+培训+个性化开发
- 优维：4名高级工程师驻场
- 甲方：从各个业务维护团队中抽调人员成立项目组（脱产）



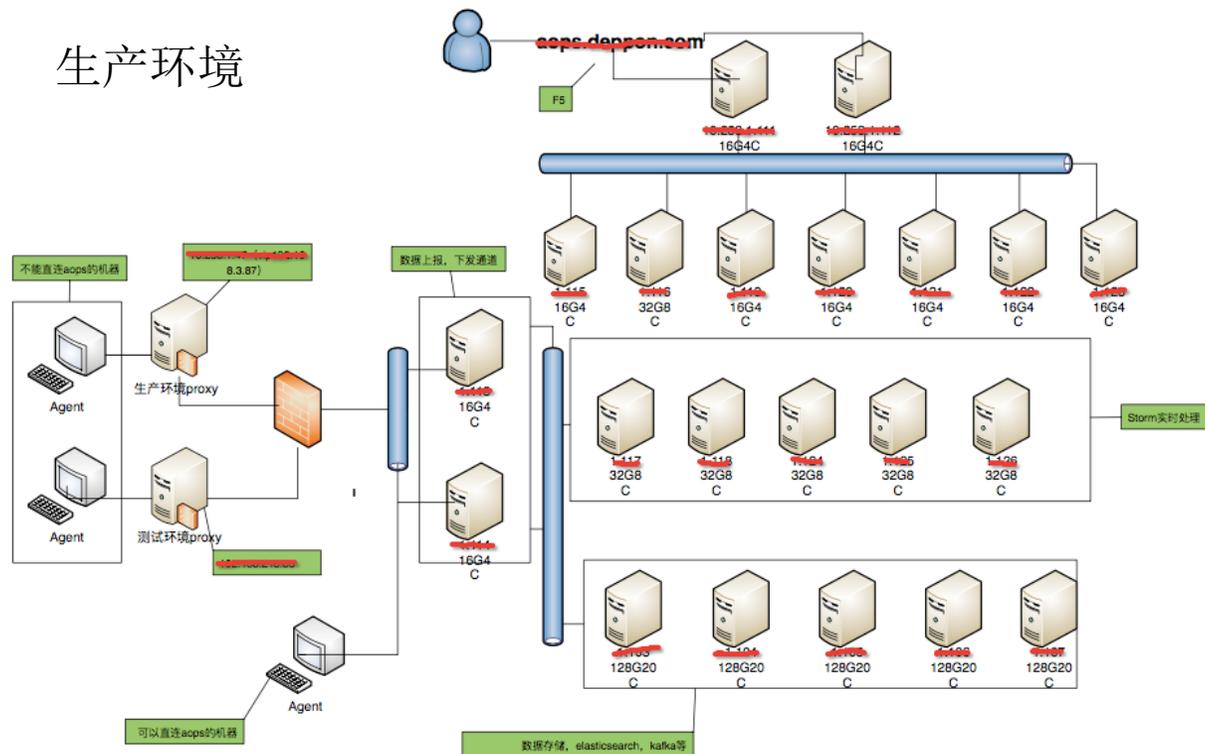
交付两个平台：测试平台+生产平台

测试环境



- 功能验证
- 稳定性验证
- 新版升级方案验证
- 程序包测试
- 业务接入前的测试

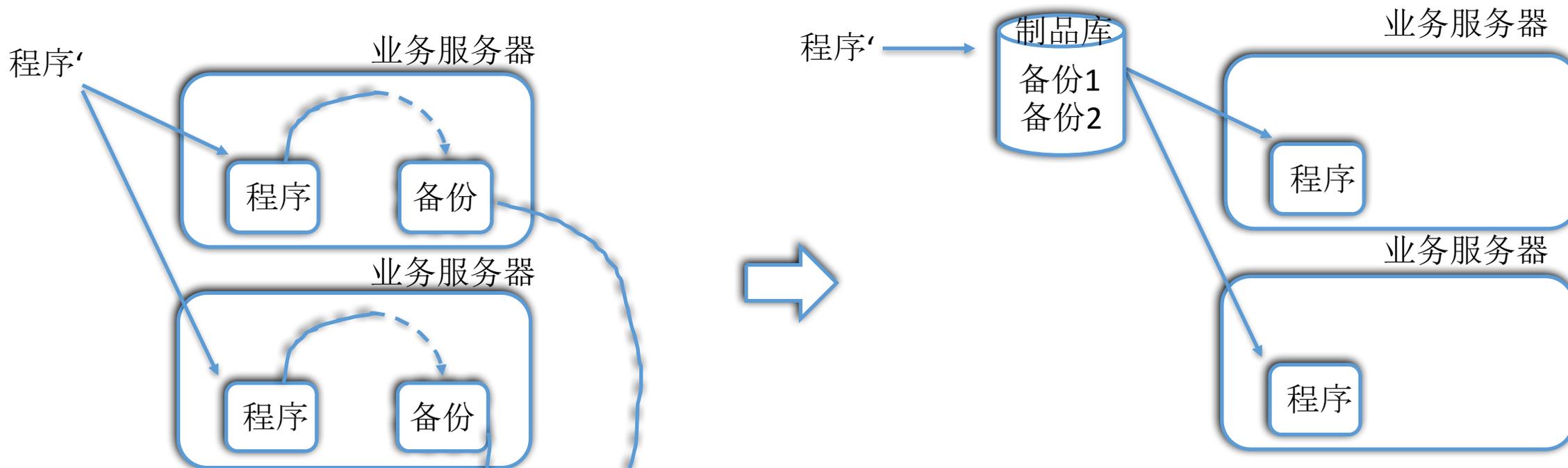
生产环境



- 资源管理 10%
- 应用管理 5%
- 应用交付 70%
- 应用监控和告警 15%



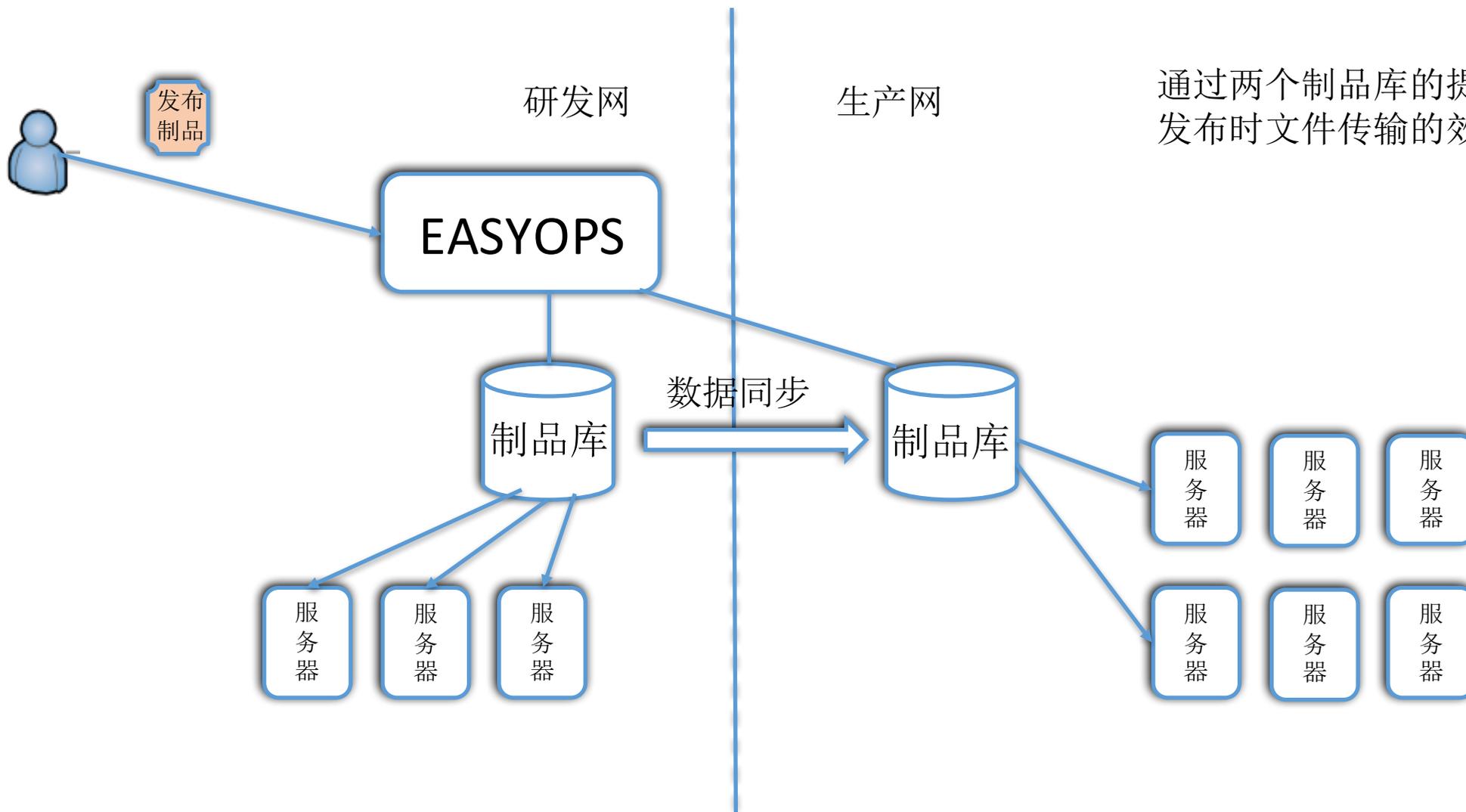
集中还是离散？



```
penglihangs-MacBook-Air:easyops penglihang$ ls -l
total 0
drwxr-xr-x  2 penglihang  staff   68  8 26 13:20 20150123
drwxr-xr-x  2 penglihang  staff   68  8 26 13:21 20150324
drwxr-xr-x  2 penglihang  staff   68  8 26 13:21 20161230
drwxr-xr-x  2 penglihang  staff   68  8 26 13:21 20170322
drwxr-xr-x  2 penglihang  staff   68  8 26 13:21 20170522
drwxr-xr-x  2 penglihang  staff   68  8 26 13:21 20170825
drwxr-xr-x  2 penglihang  staff   68  8 26 13:20 easyops
penglihangs-MacBook-Air:easyops penglihang$
```



双制品仓库同步解决网路隔离问题

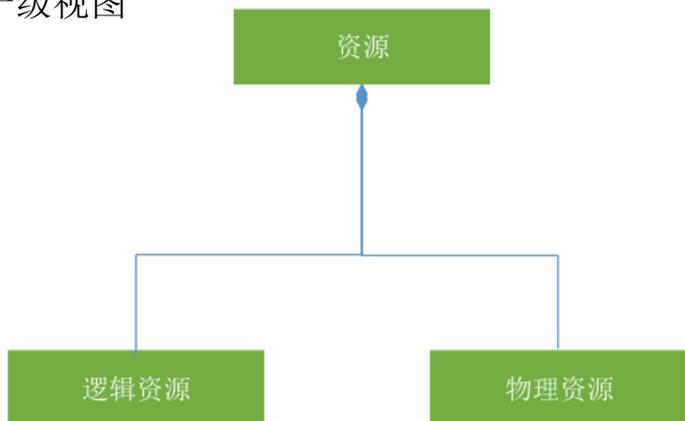


通过两个制品库的提前同步来提高发布时文件传输的效率问题。

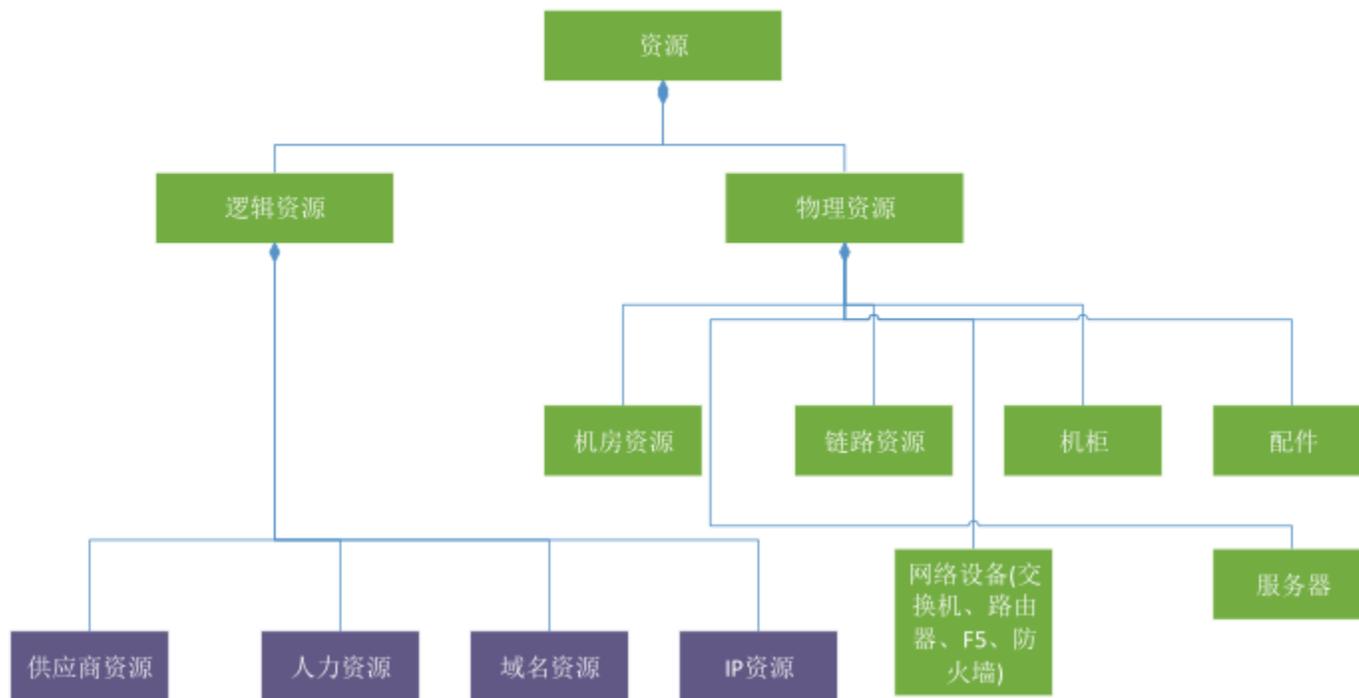
根据业务需求设计合理的资源模型



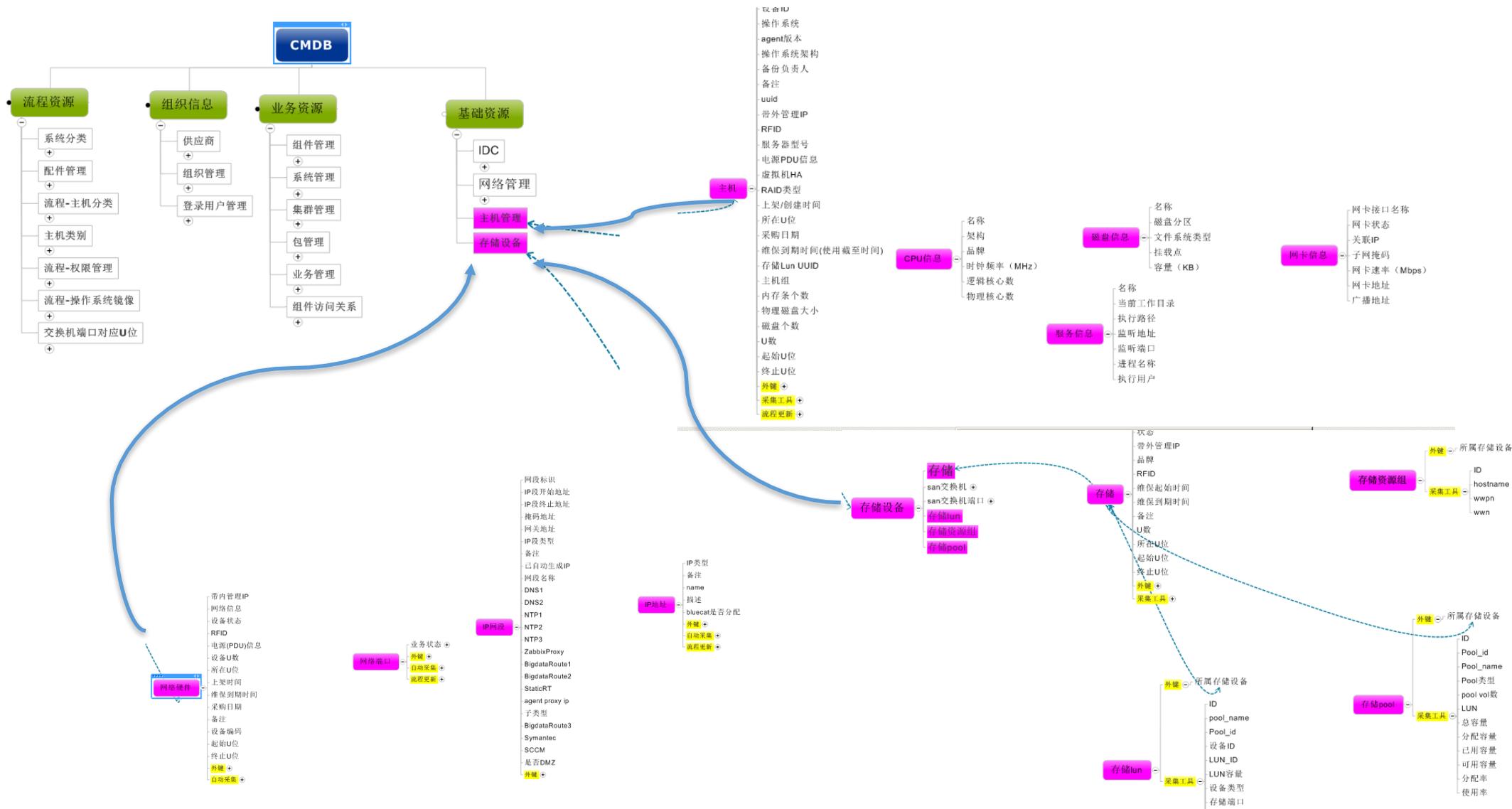
一级视图



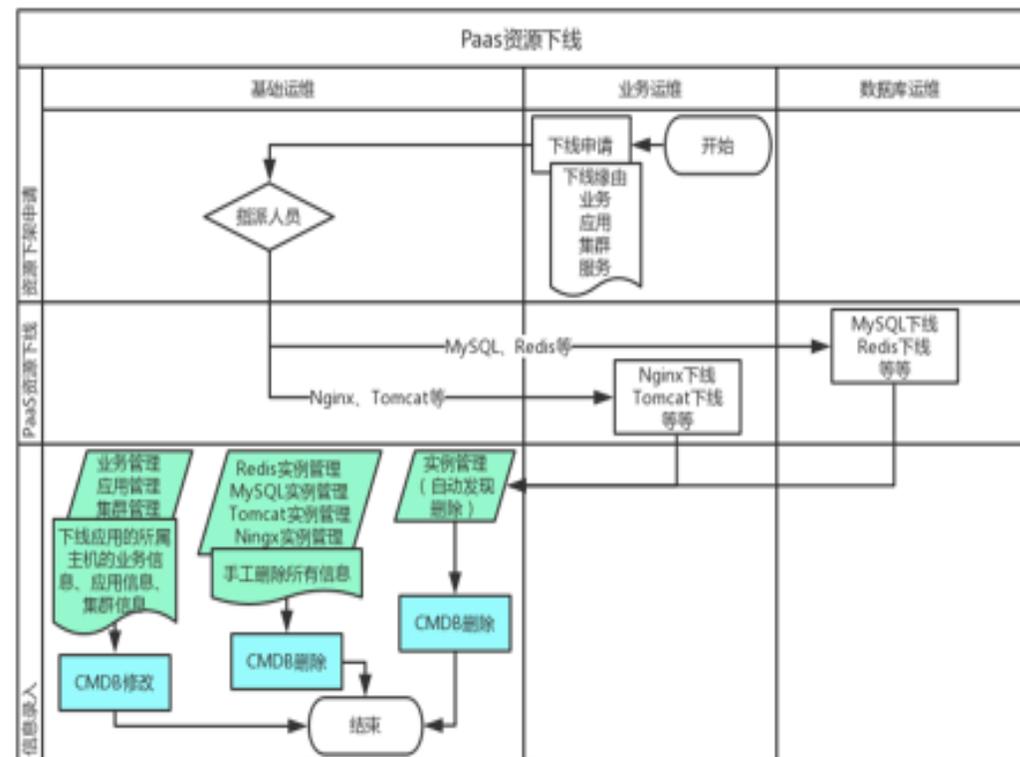
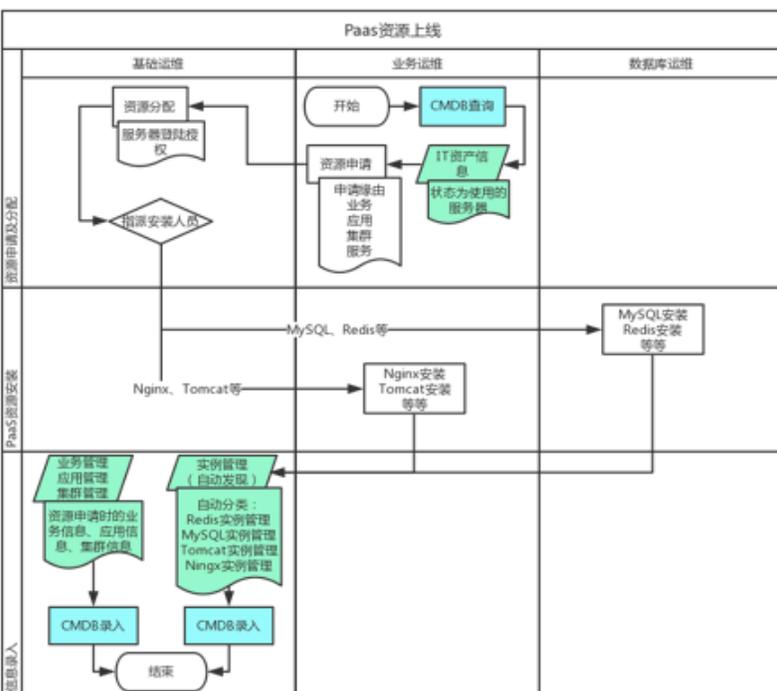
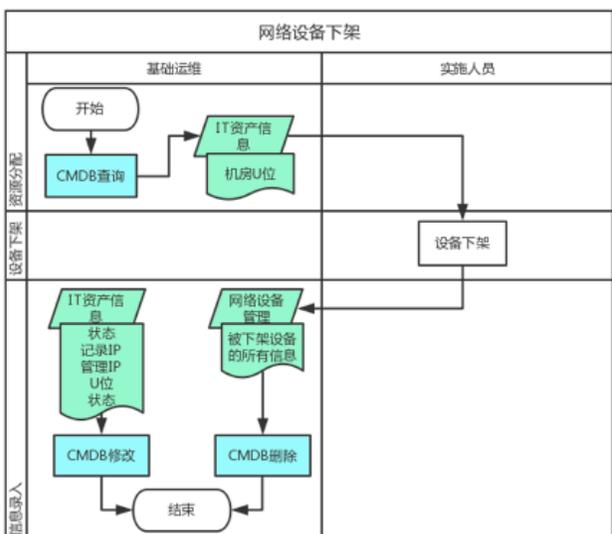
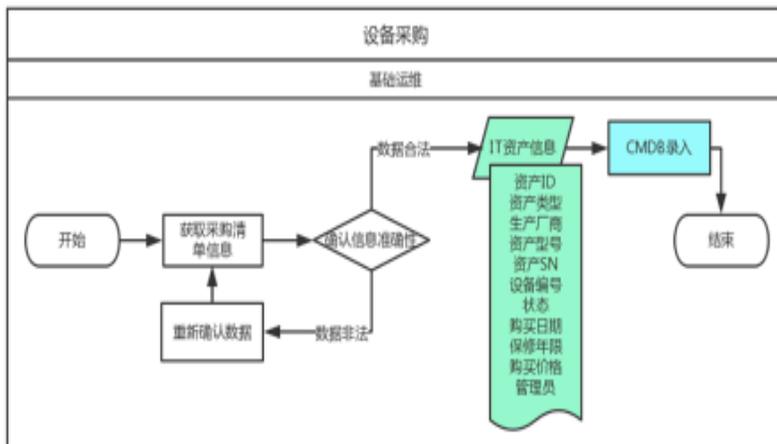
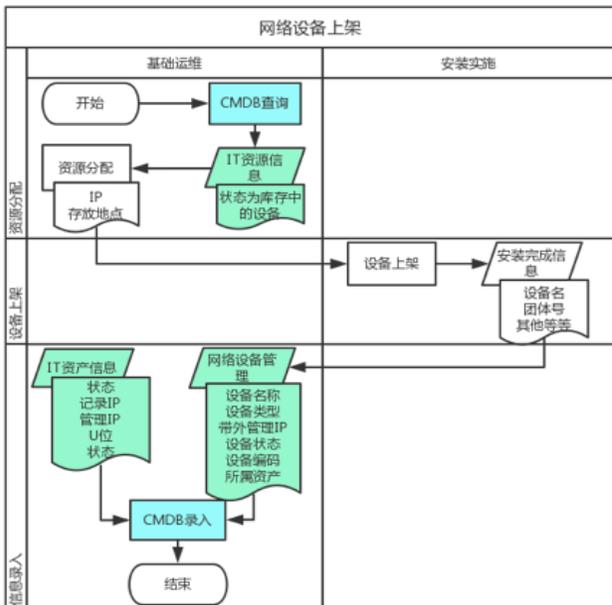
二级视图进一步对配置进行细化



资源模型分解

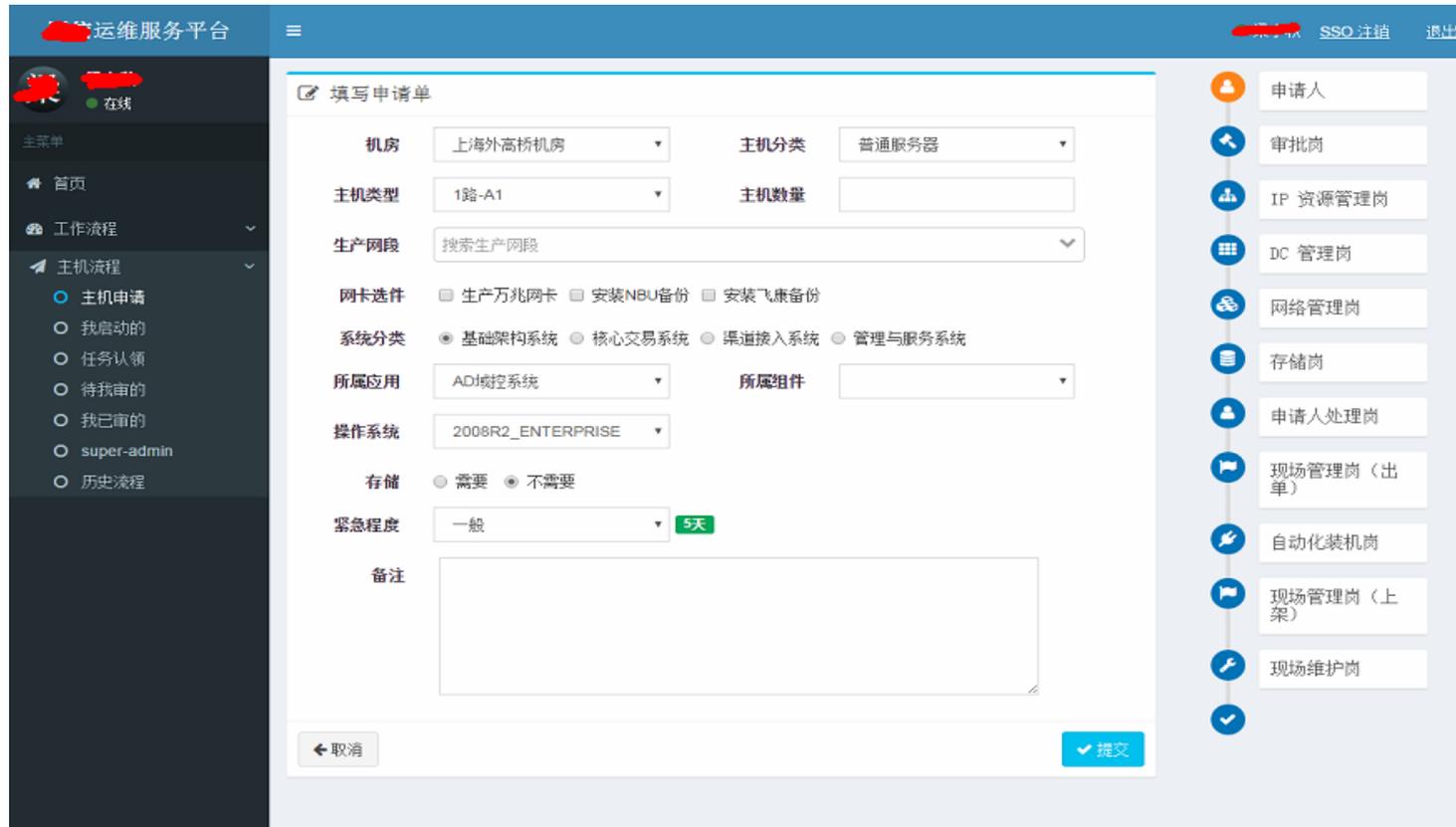
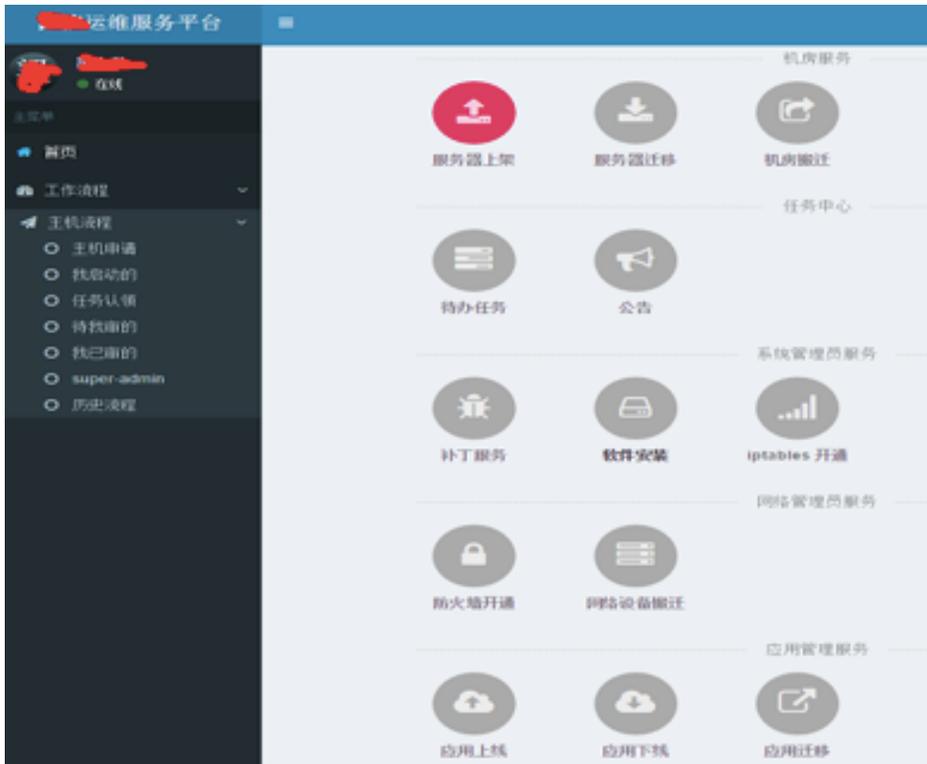


设计配置项的管理和维护过程





流程系统对接--让已经填过的信息自动录入





配置发现工具的开发--把配置维护的话让工具去干

CMDB自动发现



IP-MAC信息采集



IP网段管理与IP地址关联



VMWAREESXI宿主机信息采集



VMWARE虚拟机信息与宿主机关系采集



主机存储序列号W MN采集



主机管理与IT资产信息关联



域名信息与域名详情关联



存储设备采集



实例与业务管理关联



实例异常告警



实例归分类



实例自动发现



根据IP网段生成IP地址



物理机与交换机的关联



网络设备探测发现



网络设备端口管理



网络设备管理与IT资产信息关联



采集存储交换机硬件信息



采集网络设备硬件信息

采集插件	用途、场景	原理	依赖哪些第三方工具	关联哪些cmdb对象、属性
IP-MAC的发现 IP对应mac地址采集 (nmap)	针对交换机上没有mac表, 把本网段所有活跃IP的MAC, 将此关系录入到nmap	借助nmap采集ip和mac的关系; 使用nmap扫描本机网段 (-sP -n) 获得网段内所有活跃IP的MAC, 将此关系录入到nmap		对象: IP地址管理 属性:
IP-MAC的发现 IP对应mac地址采集 (snmp)	针对交换机上采集网络设备硬件信息	根据设备管理	依赖设备厂商自带的工具, 比如EMC: 存储序列号;	对象: 网络设备管理
物理机与交换机的关联	使用MAC地址和ARP表, 使主机建立关联	针对自定义3机相关信息比如主机挂存储, 以及主机的信息, 比如IBX-Fww	vmware esxi宿主机信息采集	对象: 网络设备管理 属性: 带外管理IP
网络设备端口管理	需要采集网络设备名称, 关联网络设备	vmware esxi宿主机信息采集	网络设备探测发现(初始化)	对象: SAN交换机 SAN交换机端口 属性: IP、序列号、WWN、微码版本
存储设备采集	通过工具采集在cmdb管理层的存储信息	vmware esxi宿主机信息采集	采集存储交换机硬件信息	属性: ID、所属交换机、端口索引、SlotName、PortName、device_port_wwn、端口类型、端口状态、端口速率、最大速率、端口名称、别名
		根据ip网段生成ip地址	根据san交换机的name, 采集san交换机硬件信息、以及	无



承载多少信息不重要，重要的是信息要流动起来

资源模型

三层简化结构:



资源项 **43**

属性 **750**

流程制度

5 维护动作

录入、修改、删除、人工变更、自动发现

2 关联流程

事件、主机上架流程

系统建设

即搜即得

100% 自动采集 (东莞)

标准化实时接口

与流程对应的维护权限控制

业务系统覆盖

业务系统 **215个**

组件 **632个**

组件关系 **2009条**

实例信息

机房 **8个**

机柜 **209个**

网络设备 **424个**

存储 **16个**

主机 **4426个**

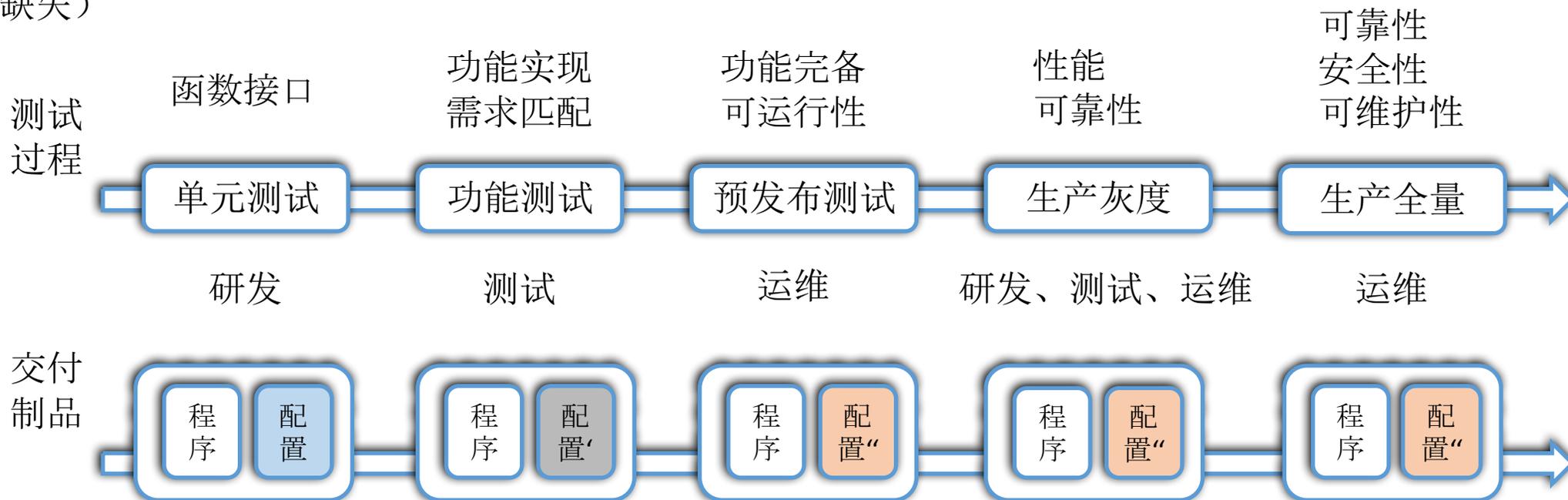
在用IP **3046个**

测试完了，但真得做够了吗？



测试目标：

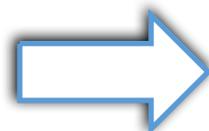
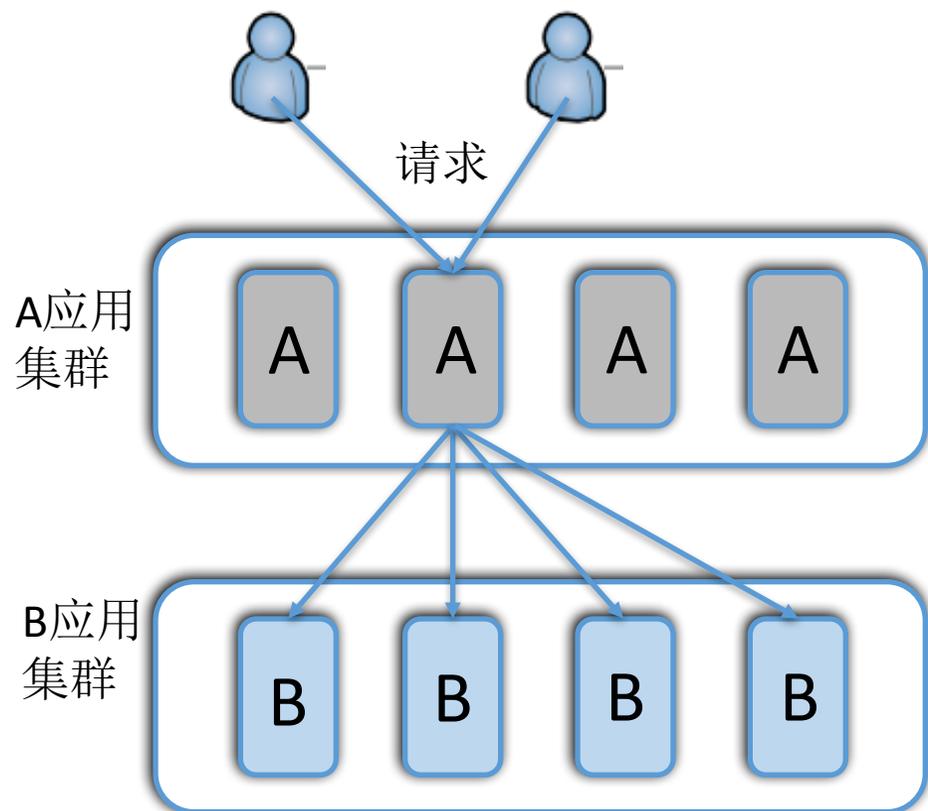
- 功能
- 性能（部分缺失）
- 可靠性（部分缺失）
- 安全性（缺失）
- 可运行性（缺失）
- 可维护性（缺失）



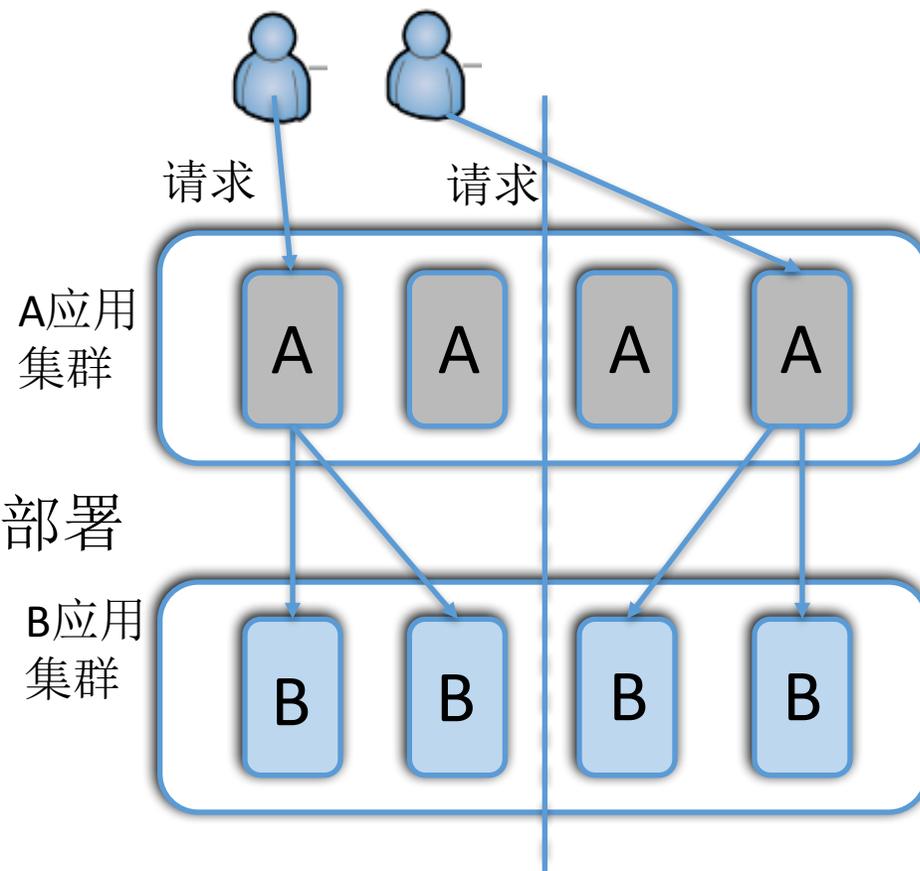
真灰度还是假灰度--蓝绿部署



A功能依赖B功力,
A和B要一起发布



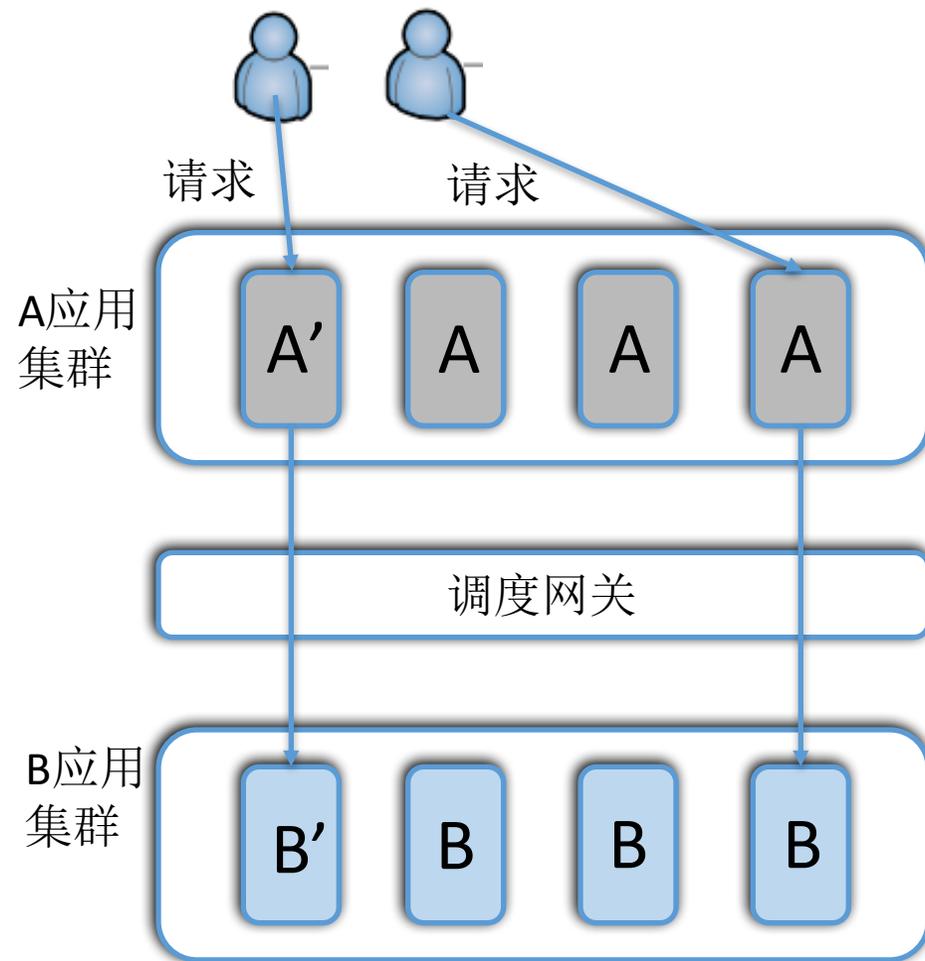
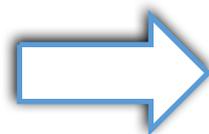
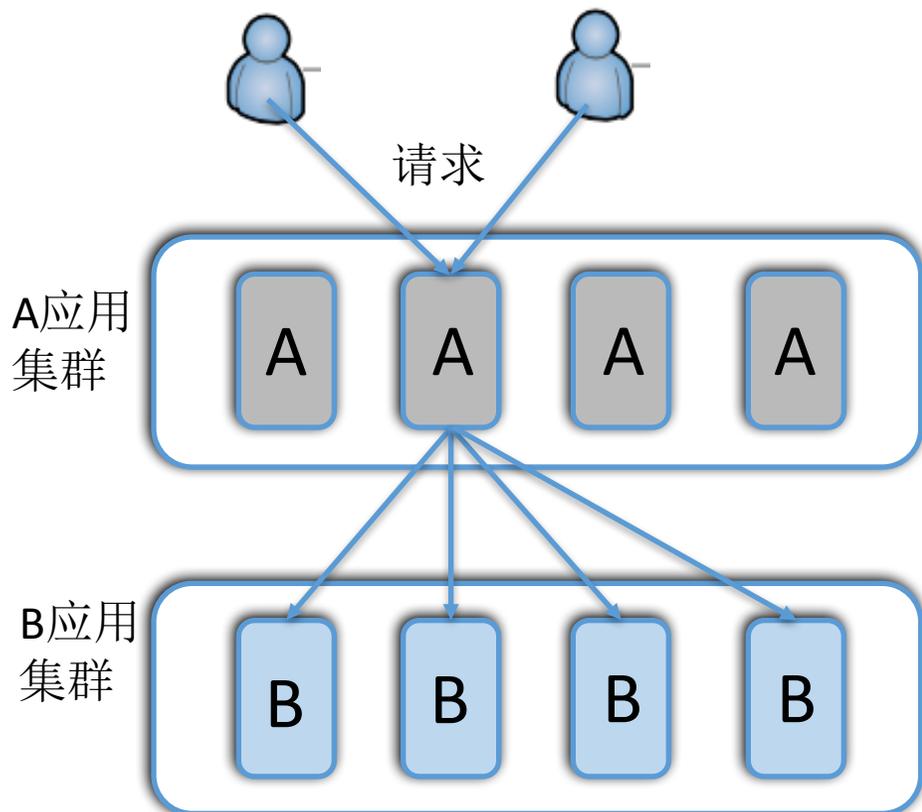
蓝绿部署



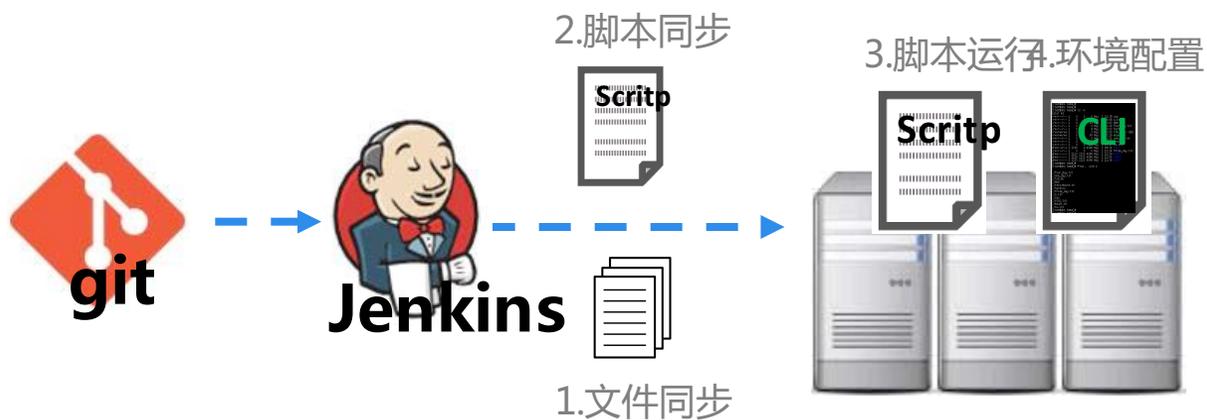


真灰度还是假灰度--灰度能力

A功能依赖B功力,
A和需要一起发布



从文件交付到程序包交付



优点：

- 实现简单
- 调整灵活

缺点：

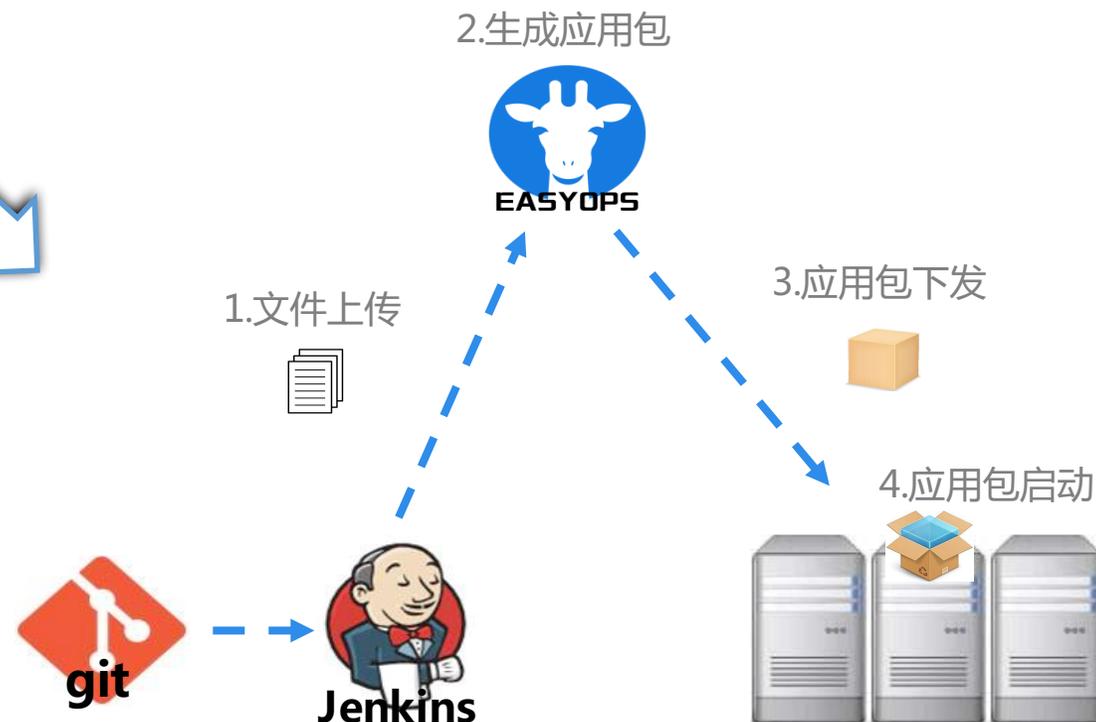
- 质量不可控
- 效率较低
- 环境变化时的管理成本高
- 设备迁移时成本很高

优点：

- 一键发布
- 快速部署及快速迁移
- 应用环境实现标准化
- 简化应用的维护

缺点：

- 应用需要接入容量管理平台





统一目录结构

统一的应用目录结构设计，主程序、配置、依赖库、文档、脚本等内容需要有明确统一的子目录

含程序及配置

保存全量的二进制主程序和对应的配置文件。

程序和配置分版本管理。

启动停止脚本

标准化的启动和停止脚本。通过传入参数来支持不同的服务环境。

记录环境变量

环境变量管理，记录应用所依赖的各种环境配置。

应用属性描述

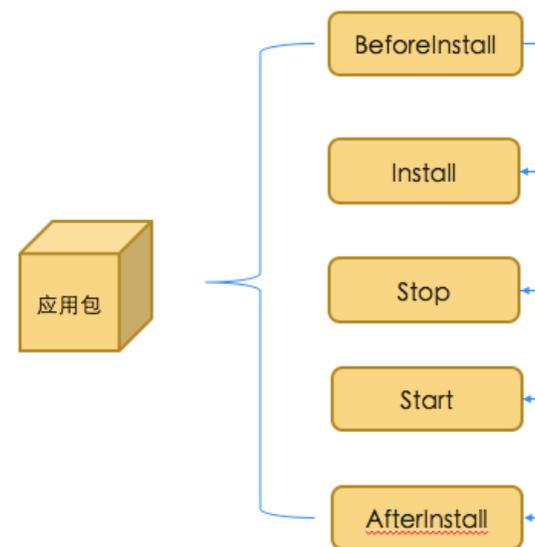
记录包启动后的应用状态描述信息。信息用于支持后续自动化处理。

包括：进程、端口、服务、接口等

定义维护方法

支持定义包的变更前后置任务来支持差异化的变更过程。

支持定义包的各种管理手段来支持后续维护的自动化。如：进程监控，日志清理，状态分析等

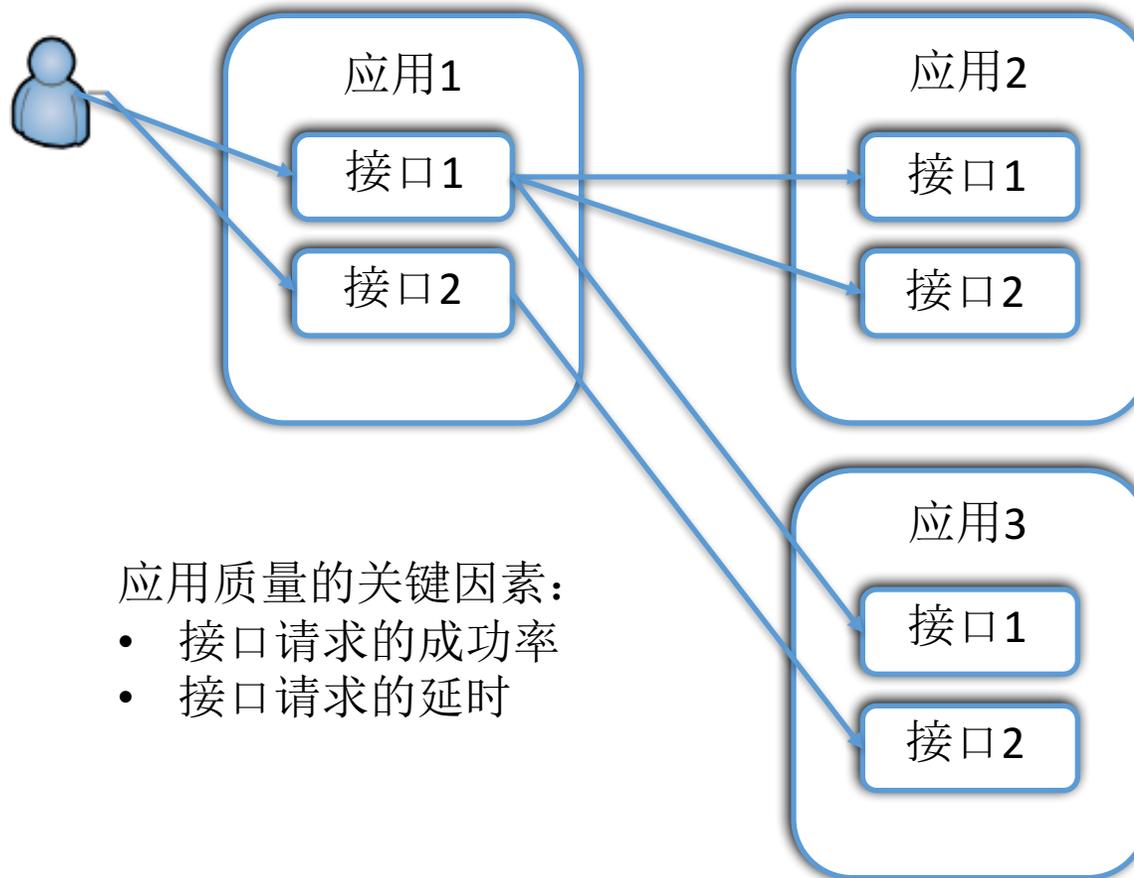


- 完整的应用包事件流机制

日志标准化及应用的质量监控



字段名	字段中文含义	是否可以为空	备注
Timestamp	日志产生的时间戳	否	
请求 ID	一次请求产生的 ID	可以为空	建议这个 requestID 能够一直警告
ParentId			
请求步骤 ID		可以为空	每次请求经历过的服务节点, 自动加一
主调 IP	发起一次 RPC 调用的主动 IP	否	如 1.1.1.1
主调服务名	系统中唯一的服务标识	否	Portal_web
被调服务名	系统中被调用的服务标识	否	Portal_server
被调接口名	系统中被调用的接口标识	否	如 login, 一个字符串, 研发定义
被调服务 IP	被调服务监听的 IP	否	如 2.2.2.2
被调服务 PORT	被调服务监听的端口	否	如 9080
调用状态	成功还是失败。以程序具体的返回码为准。0 为成功, 其他的都为失败	否	
调用延时	一次调用产生的延时	否	
代码位置	产生该调用的代码位置, 方便故障定位	是	
预留字段		是	



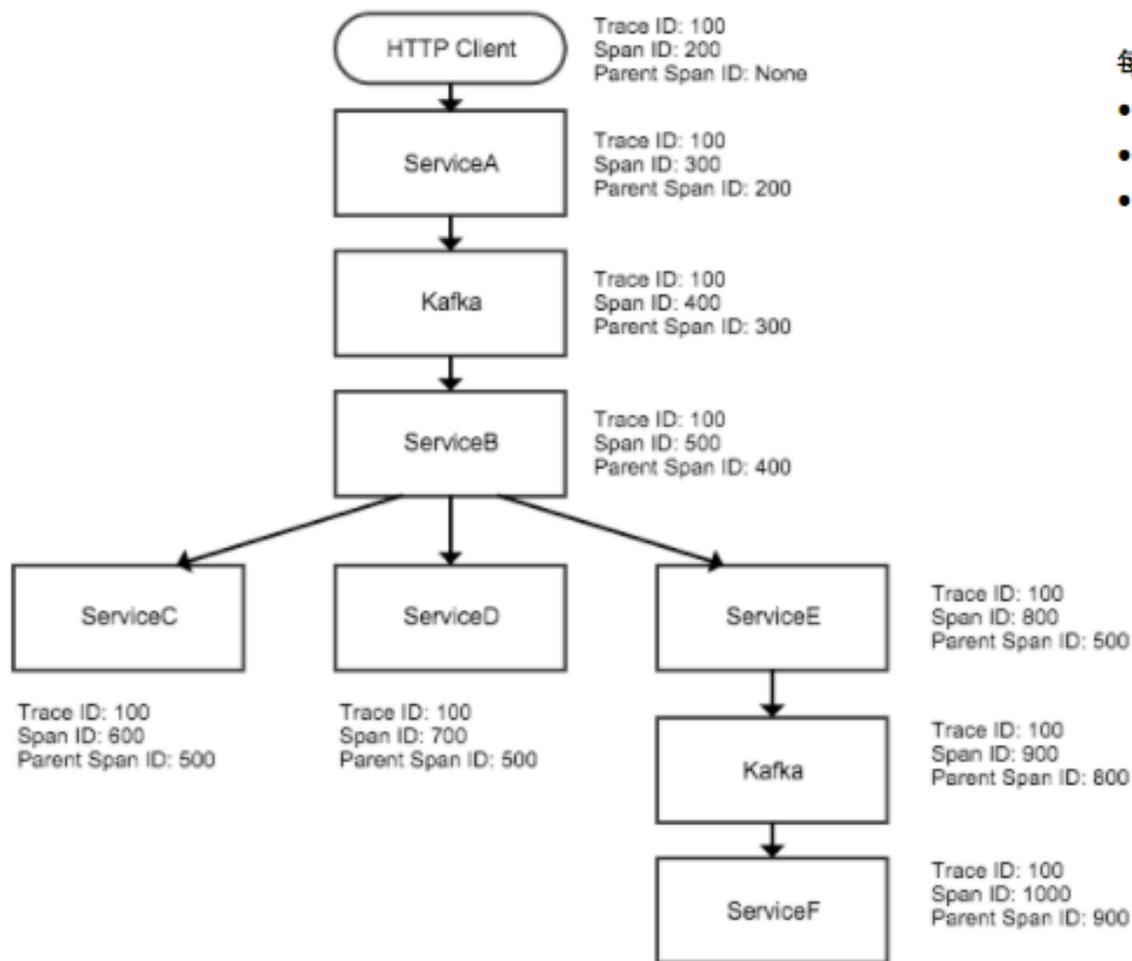
- 应用质量的关键因素:
- 接口请求的成功率
 - 接口请求的延时

应用的进程在运行, 但工作的怎么样?



分布式链路追踪--跟踪请求过程，直接发现问题点

Trace Tree with IDs

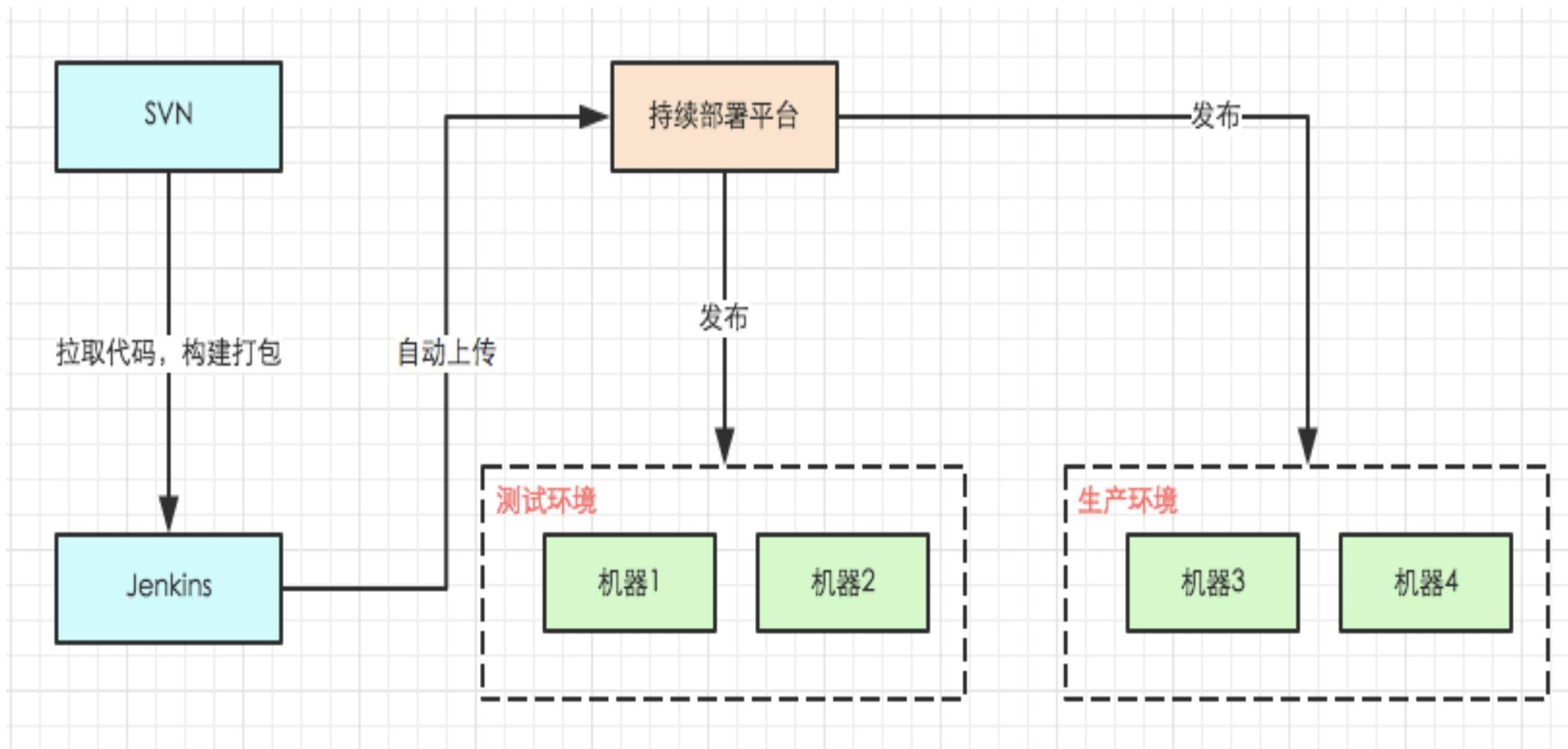


每一个跨度(span)有三个ID:

- Trace ID: 一个轨迹中所有的跨度(span)共享同一个Trace ID。
- Span ID: 用以标示不同的跨度(span)。Span ID与Trace ID不一定相同。
- Parent Span ID: 只有子跨度持有这个ID，根跨度没有Parent Span ID。

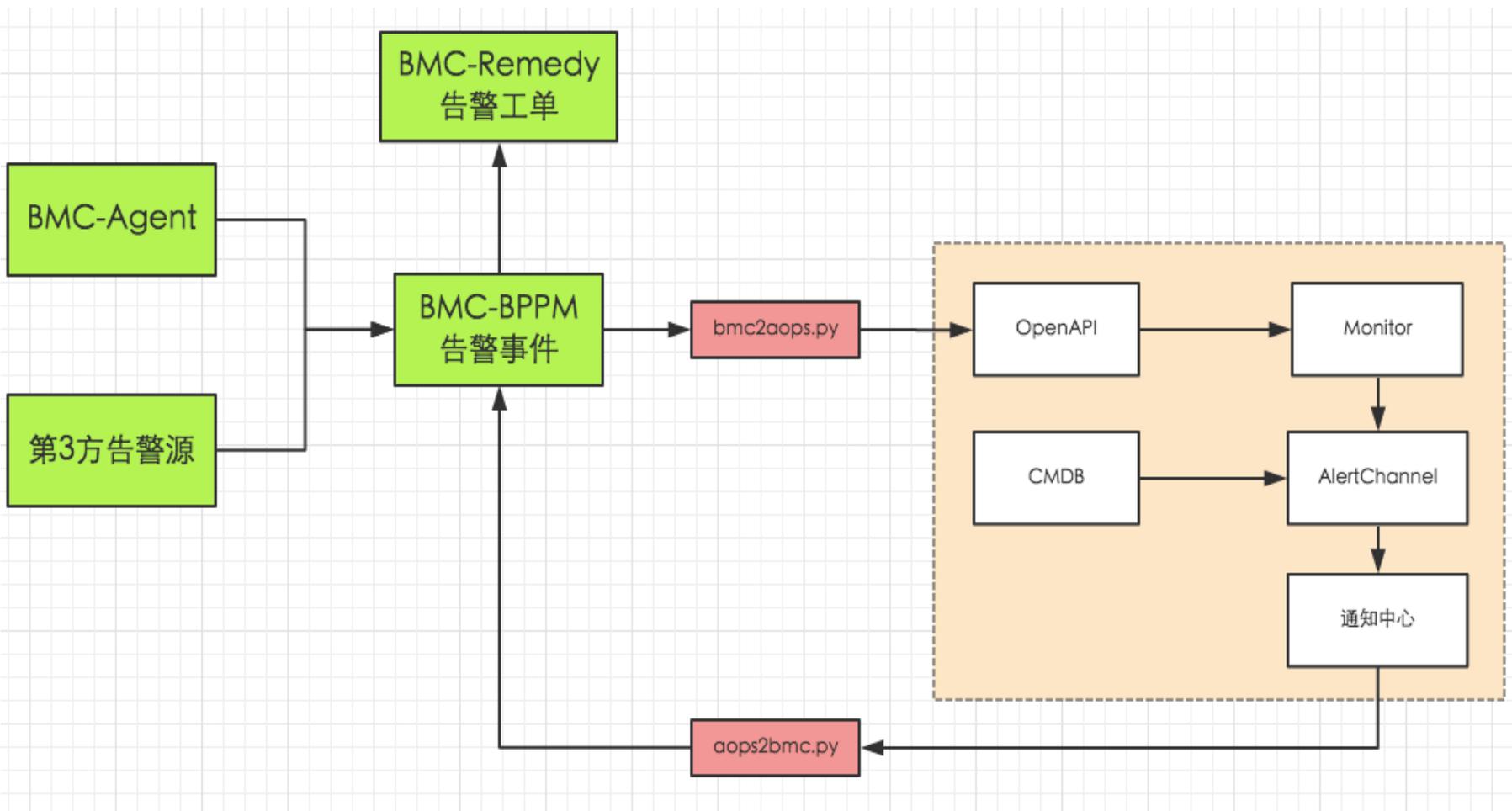


JENKINS对接实现从源码库到生产的全过程交付





告警能力对接--不要告警，要不告警！

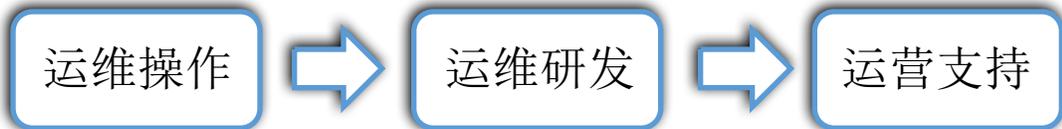


- BMC平台触发告警事件
- EASYOPS收到事件
- 查询CMDB进行影响分析和收敛
- 驱动恢复机制执行
- 结果反馈给BMC的消息通道
- 用户从BMC收到恢复通知



工具和平台都是为人服务的

优秀的人总是把各种不可能变为可能！



类目	培训大纲		
	课程	培训时间 (天)	讲师
DEVOPS理论知识	DEVOPS基础理论	0.25	
	DEVOPS实践	0.25	
研发管理及相关背景知识	研发管理基础理论	0.25	
	GIT及SVN基础实践	1	
	JENKINS基础实践	1	
测试管理及相关背景知识	测试管理基础理论	0.25	
	单元测试实践	0.5	
	自动化测试实践	0.5	
	功能测试实践	0.5	
运维管理及相关背景知识	运维管理基础理论	0.25	
	LINUX基础实践	0.5	
	SHELL开发实践	2	
	PYTHON开发实践	2	
EASYOPS平台理念及相关功能	EASYOPS设计理念	0.25	
	EASYOPS功能介绍	0.25	
	EASYOPS应用场景和企业案例介绍	0.25	
EASYOPS高级实践	平台安装 (单位、集群)	1	
	平台配置及维护	0.5	
	平台工具及流水线开发	2	
	平台插件开发 (采集+个性化界面)	3	



国际象棋



围棋



VS

AlphaGo

数据比算法更重要！



IT运维统一服务门户（建设中...）

EASYOPS
cmdb_test

IT 服务中心

运维系统地图

运维流程服务目录

API 服务网关

信息中心

任务中心

权限自定义

- 用户管理
- 角色管理
- 流程特权管理
- 流程环节管理

流程设计

- 自定义流程
- 自定义表单

流程平台

function

function

function

function

function

function

运维自动化平台

function

function

function

组件交付平台

function

function

我的信息中心

▼ 概览

主机状态		应用状态		组件状态	
43	2	18	33	35	1
正常	异常	正常	异常	正常	异常

▼ 告警

告警趋势

最新告警

- 17:23, 【告警】应用 dc_console...
- 17:23, 【告警】主机 192.168.100...
- 17:20, 【恢复】应用 paul 网络层应...
- 17:18, 【告警】程序包 192.168.10...
- 17:16, 【恢复】应用 paul 网络层应...

我的任务中心

平台名称	待处理	进行中	已完成
流程平台	11	3	123
运维自动化平台	15	11	311
组件交付平台	9	6	123

待处理

➤ 运维的统一服务入口

我的流程

我的工具

我的信息

我的任务

全面的应用资源管理模型



EASYOPS IT 资源管理 持续交付 智能监控 IT 运营分析

应用信息 应用交付 编辑 变更历史 更多操作

cmdb 应用信息

设备信息		人员信息			资源信息				
集群	主机	运维负责人	开发负责人	测试负责人	程序包	配置包	中间件	镜像	文件源
5个	4台	easyops			1个	2个	1个	1个	0个

➤ 资源对象自定义

对象自定义管理

对象属性自定义管理

➤ 资源对象关系自定义

对象关系管理

应用的运维和交付管理能力

The screenshot displays the EASYOPS interface for managing application deployment pipelines. The top navigation bar includes 'IT 资源管理', '持续交付', '智能监控', and 'IT 运营分析'. The left sidebar shows a navigation menu with categories like '基础信息', '流水线', and '应用维护'. The main content area is titled '发布流水线' (Release Pipeline) and shows a flowchart of the deployment process. Below the flowchart is a table of deployment history.

发布流水线 (Release Pipeline) Flowchart:

```

    graph LR
      Start([流程起点]) --> Step1[成功 获取版本信息]
      Step1 --> Step2[成功 获准部署?]
      Step2 --> Step3[成功 获取应用和集群信息]
      Step3 --> Step4[成功 屏蔽应用告警]
      Step4 --> Step5[成功 部署-生产环境]
      Step5 --> Step6[成功 开启应用告警]
      Step6 --> Decision{部署成功?}
      Decision -- YES --> End([成功 结束])
      Decision -- NO --> Wait[等待中 启动回滚流水线]
  
```

Deployment History Table:

时间	耗时	版本名称	执行者	结果
2017-08-21 09:06	1 分钟	2.0.4	easyops	成功
2017-08-21 08:23	几秒	2.0.2	easyops	成功
2017-08-21 08:03	1 分钟	2.0.1	easyops	成功
2017-08-17 15:52	几秒	1.10.14	easyops	暂停
2017-08-15 18:12	1 分钟	2.0.0	easyops	成功
2017-08-15 11:20	2 分钟	1.22.20	easyops	成功
2017-08-15 10:06	几秒	1.1.30	easyops	成功
2017-08-15 09:58	几秒	1.1.30	easyops	成功

- **可定制的流水线**
多级的交付流水线
根据需求自行定义
流程自动衔接
- **分场景的运维平台**
分场景组织维护能力
权限严格把控

应用的端到端监控分析能力



- 自定义数据看板
- 分模块数据采集
- 数据集中处理和汇聚
- 按场景组织视图报表

WHO WE ARE

优维科技专注于为企业转型升级实现“互联网+”能力提供IT动能，我们强调在安全、稳定的前提下，实现企业的IT精益管理能力及持续价值，以助力企业的持续商业创新。优维人将带着“DevOps管理专家”的使命，提供全栈的互联网化运维能力！



EASYOPS 优维

关于团队

优维的核心团队是由腾讯、阿里、网易的运维专家组成，具有10年的海量运维和服务运维经验，丰富的运维开发经验。

- 优维科技CEO，中国开放运维联盟发起人，“精益运维”理论提出者，中国第一批DevOps Master授权讲师，持续交付专家
- 优维科技CTO黎明，中国最大的运维管理平台织云设计者
- 公司85%是研发，坚持技术与产品驱动，最佳实践落地

我们坚持IT管理过程ERP产品化，提升所有企业的IT效能，驱动业务创新；坚持DevOps理念，带着“重新定义运维”的使命，做中国最好的IT运维服务厂商。



OUR TEAM



THANKS

释放DevOps的无限价值