云计算开源产业联盟

IT大咖说 知识分享平台

博云 BoCloud

Docker容器在传统行业的落地实践

传统行业容器化-为谁服务？

属性	传统模式	敏捷模式/互联网模式
价值	稳定、可靠、经济	业务价值、客户体验
核心目标	降低成本、成本可预测；基于规格文档开发；可靠性、安全性、控制风险	弹性、效率；管理不确定性；验证、学习、试点
需求管理	计划和可预知的功能；已知的性能和容量需求；	需求变更频繁，不明确；无法确定的性能和容量需求，需通过弹性来满足
变化频率	较少的、增量的改变	快速、经常的变化
技术来源	成熟的技术；成熟的供应商；长期合作	可能不成熟的技术；可能不成熟/小规模的供应商；可能短期合作
基础设施	IOE 传统小型机、Oracle/DB2数据库、传统商业存储	**X86化、LAMP** X86化、分布式存储、软件定义网络、开源架构
运维体系	**流程化、半自动化** 以流程为核心构建运维体系，以保障可用性为第一目标，逐步/稳健提升自动化水平	**标准化、全自动化、DevOps** 以标准化为基础，在遵从流程的基础上，实现全自动化，逐步推进DevOps等创新模式的落地

对容器化的期望-解决什么问题

营销化 | 移动化 | 服务化 | 合作开放 | 大数据化

需求的迫切性 | 访问的不确定性 | 系统的复杂性 | 海量用户

挑战

弹性不足：应用系统部署以虚拟机为单位构建，系统的扩容经历资源分配，软件安装，应用部署测试，切割入网等过程，在业务量突变的情况下无法进行快速扩展和缩容，无法应对突发性访问的挑战

发布周期长：大部分的应用系统有开发，测试，准发布和生成四个部署环境，各部署环境不一致，导致代码从开发到上线环节多，部署复杂，容易出错，无法满足业务快速上线的要求

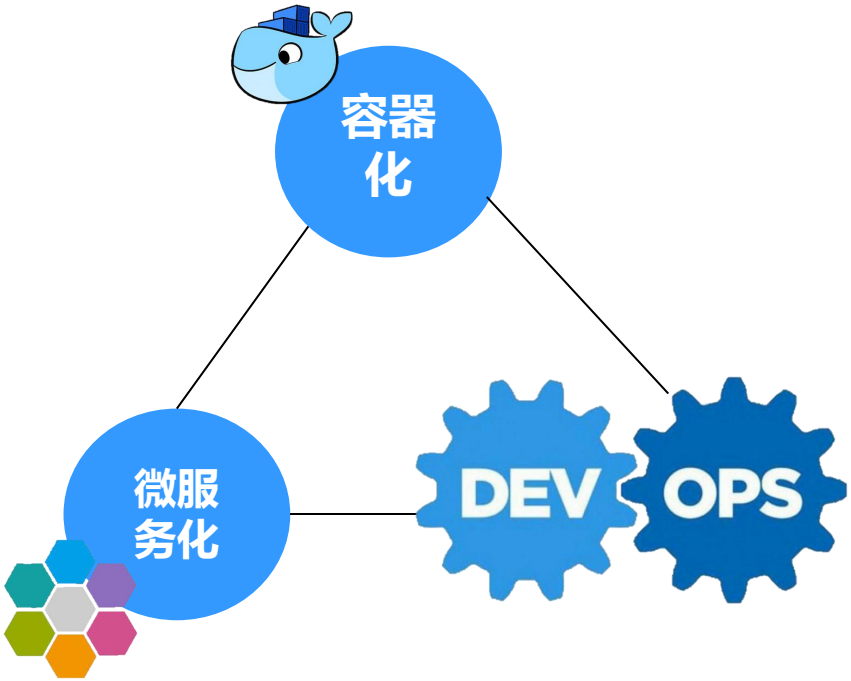
并发能力不足：业务架构无法满足千万级用户、互联网业务的高并发能力要求

问题

快速部署/迭代 | 弹性 | 易运维 | 高并发

仅仅落地容器平台是不够的

- 弹性不足
- 发布周期长
- 并发能力不足



架构改造，容器化和DevOps的基础

对数据架构的要求

- 分布式
- 配置与应用分离
- 无状态
- 可快速启动和停止

- 单服务粒度小
- 服务间松耦合，服务可独立部署

容器平台建设的关键问题

弹性伸缩问题

- 如何快速，有效的弹性伸缩

配置管理问题

- 理想情况
- 传统应用怎么办

网络问题

- 应用需求满足
- 性能

流程适配问题

- 应用镜像入库流程
- 镜像环境切换流程
- 服务目录申请审批流程
- 运维响应流程

平台对接问题

- 监控方案及对接
- 告警对接
- 日志对接
- 审计对接

可用性保障问题

- 应用可用性方案设计
- 数据可靠性设计
- 容灾设计

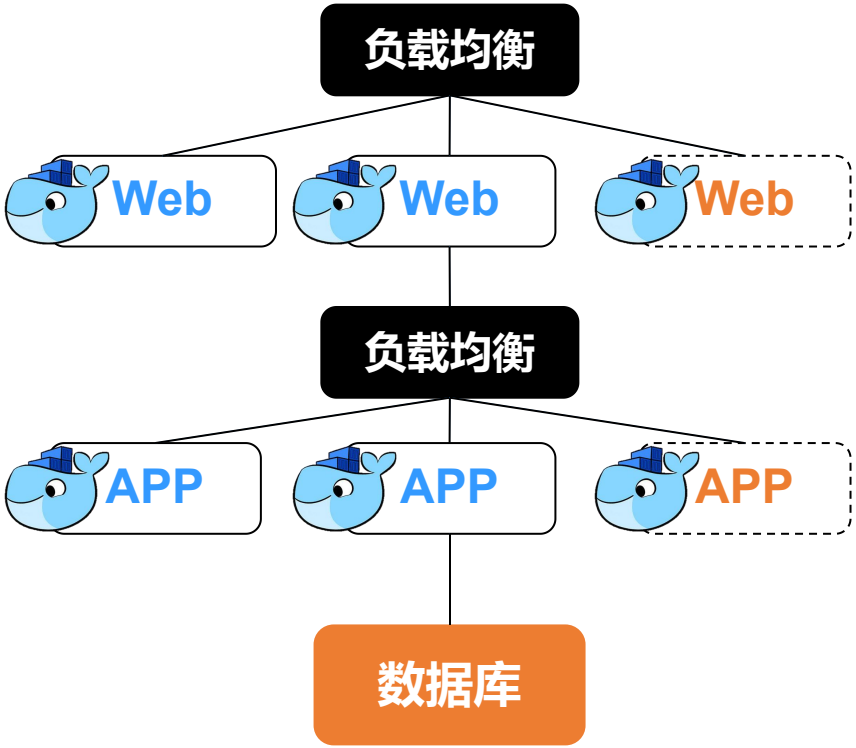
可运维性问题

- 容器故障如何诊断定位
- 容器应急处理容器监控与状态巡检

安全合规性问题

- 网络隔离合规性
- 数据保护合规性
- 访问控制

快速弹性伸缩，只容器化是不够的



- 依据什么指标做弹性伸缩？
- 应用弹了，负载均衡器怎么配置？
- 什么时候把用户调到新容器上？
- 如何缩容才能保证用户无感知？
- 应用扩容了，就一定能满足并发要求吗？
- 底层资源不足怎么办？

快速弹性伸缩，只容器化是不够的

- 与应用监控平台对接，基于应用监控指标做弹性伸缩
- 基于CPU、内存的弹性伸缩
- 手动弹性伸缩

F5对接和配置自动化

七层调度策略配置

确保只调度到可提供服务的容器上

快速弹性伸缩，只容器化是不够的

- Session机制
- 闲时缩容
- 用户在线状态检测

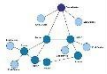

Vcenter对接

Openstack对接


公有云对接

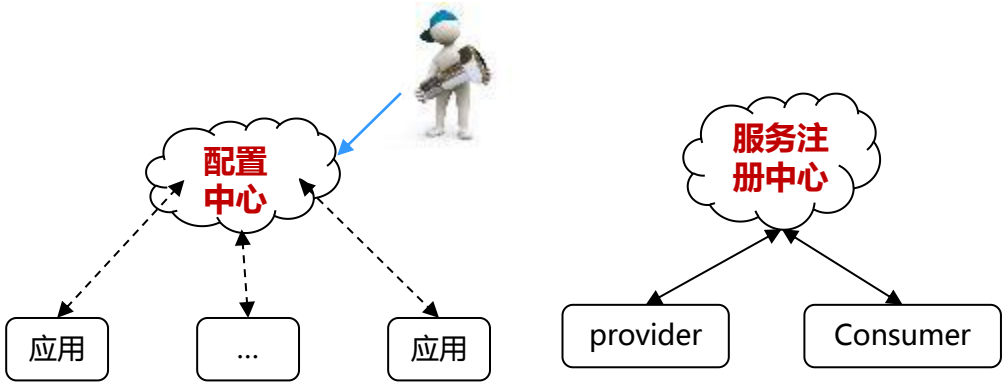

数据库


内核/中间件


部署策略

容器化应用的配置管理

理想/未来情况



- 配置发布统一化
- 配置参数动态获取
- 配置更新无需重启容器
- 配置管理可视化

- 服务的动态发现和提供
- 服务的动态调度

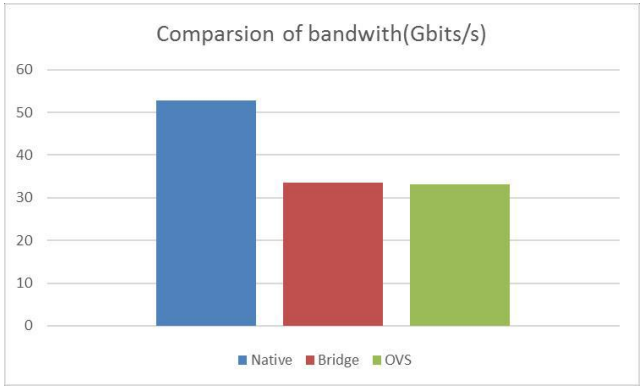
现实妥协

- 环境变量方式
- Services Name方式
- 配置文件方式

容器网络

需求和问题

- 容器间IP可达
- 网络性能问题
- 容器跑在虚拟网络上的问题：
  Overlay上的Overlay



解决方案

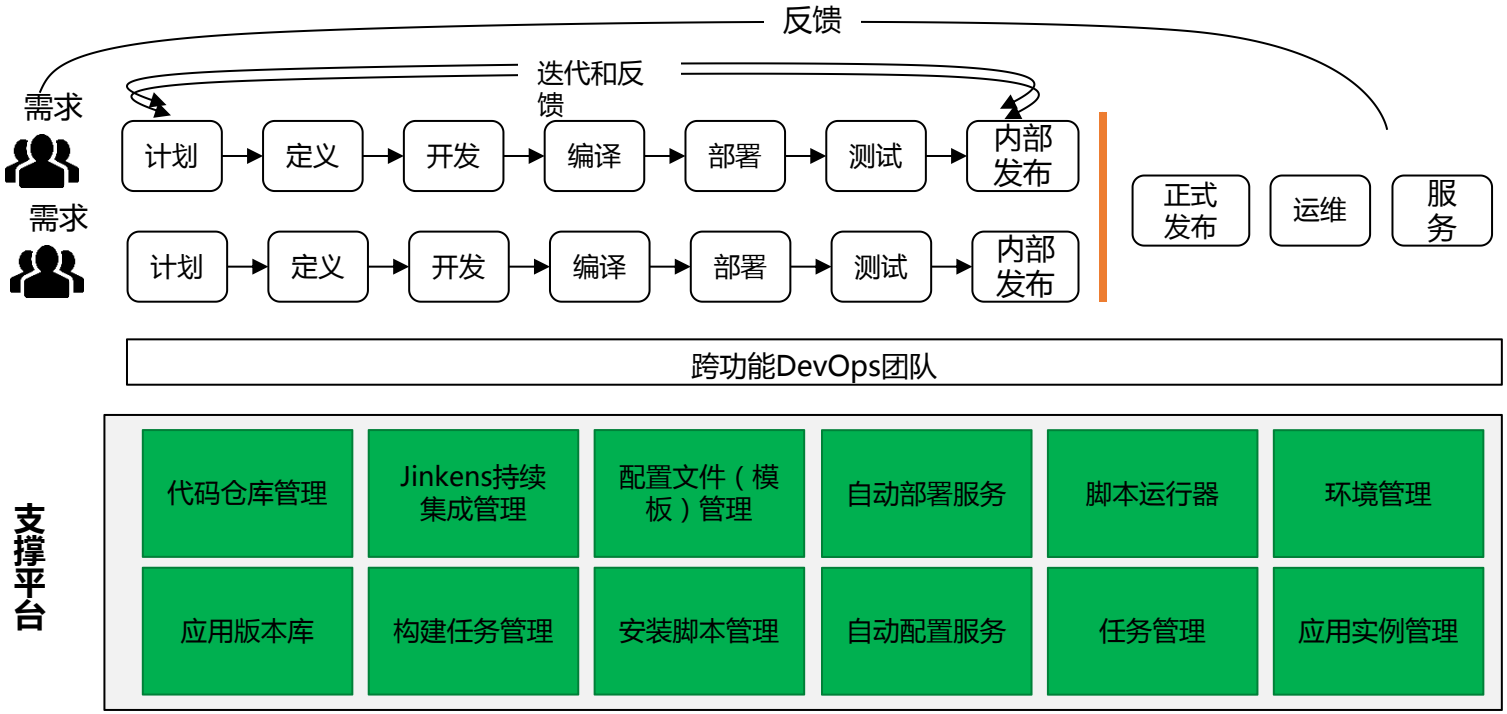
OPEN VSWITCH

+

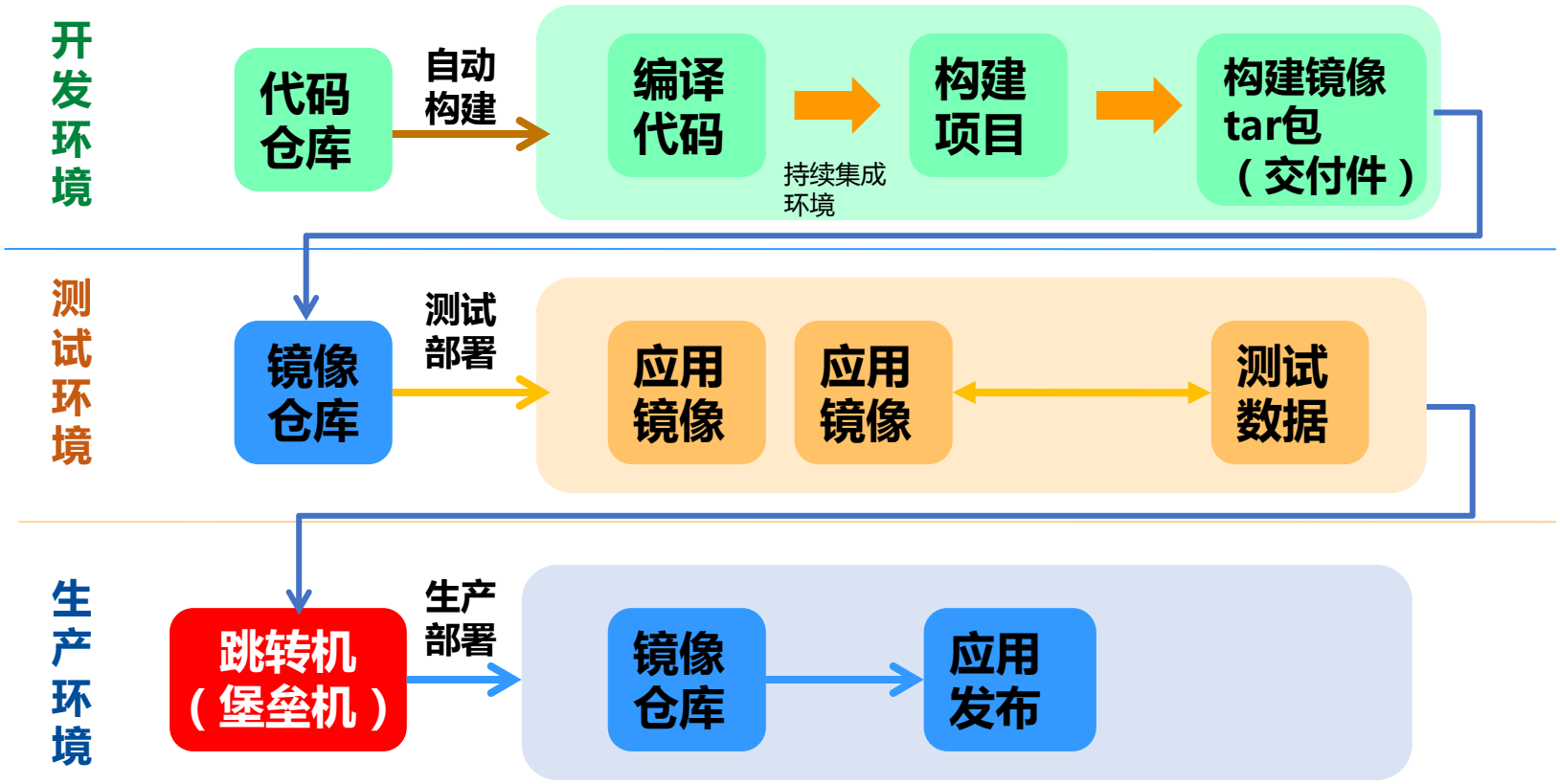
Intel DPDK

流程适配：DevOps支持

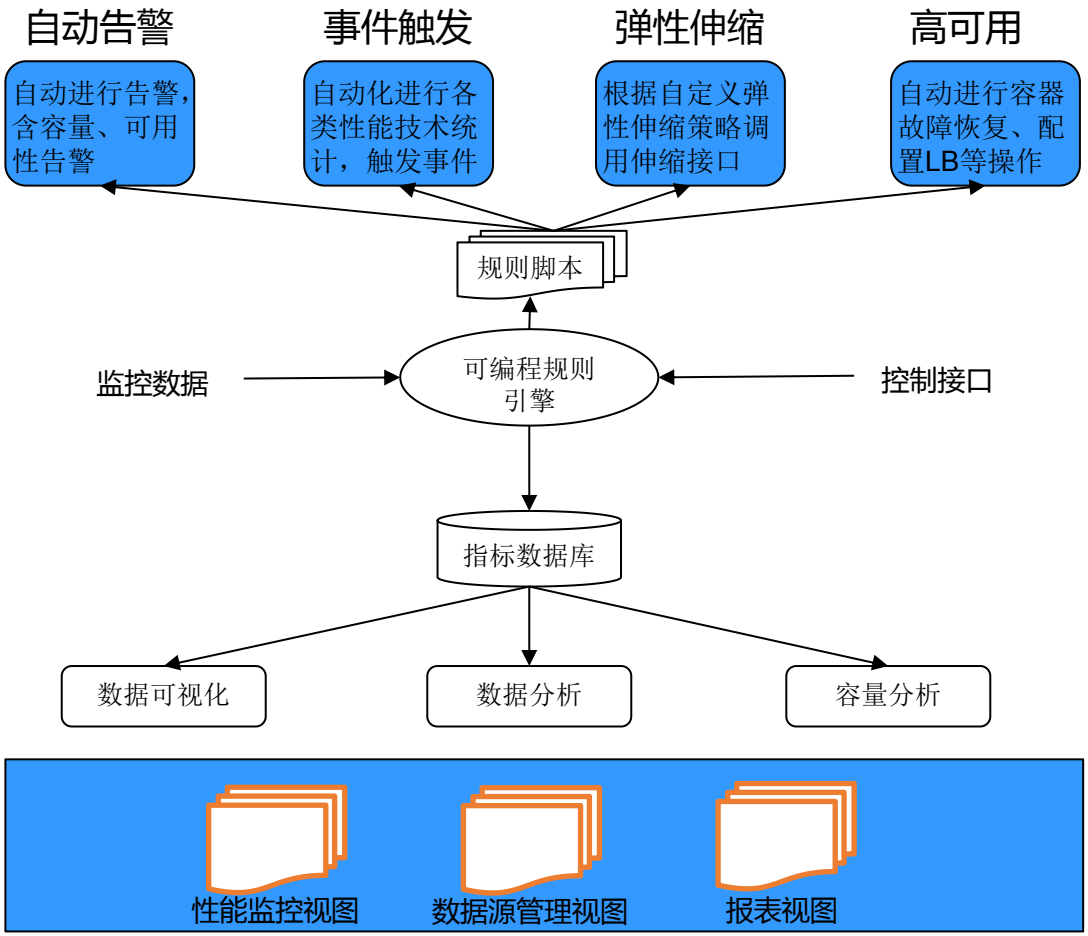
围绕开发-测试-部署-上线全生命周期流程构建自动化集成、部署、配置等能力，支撑DevOps落地



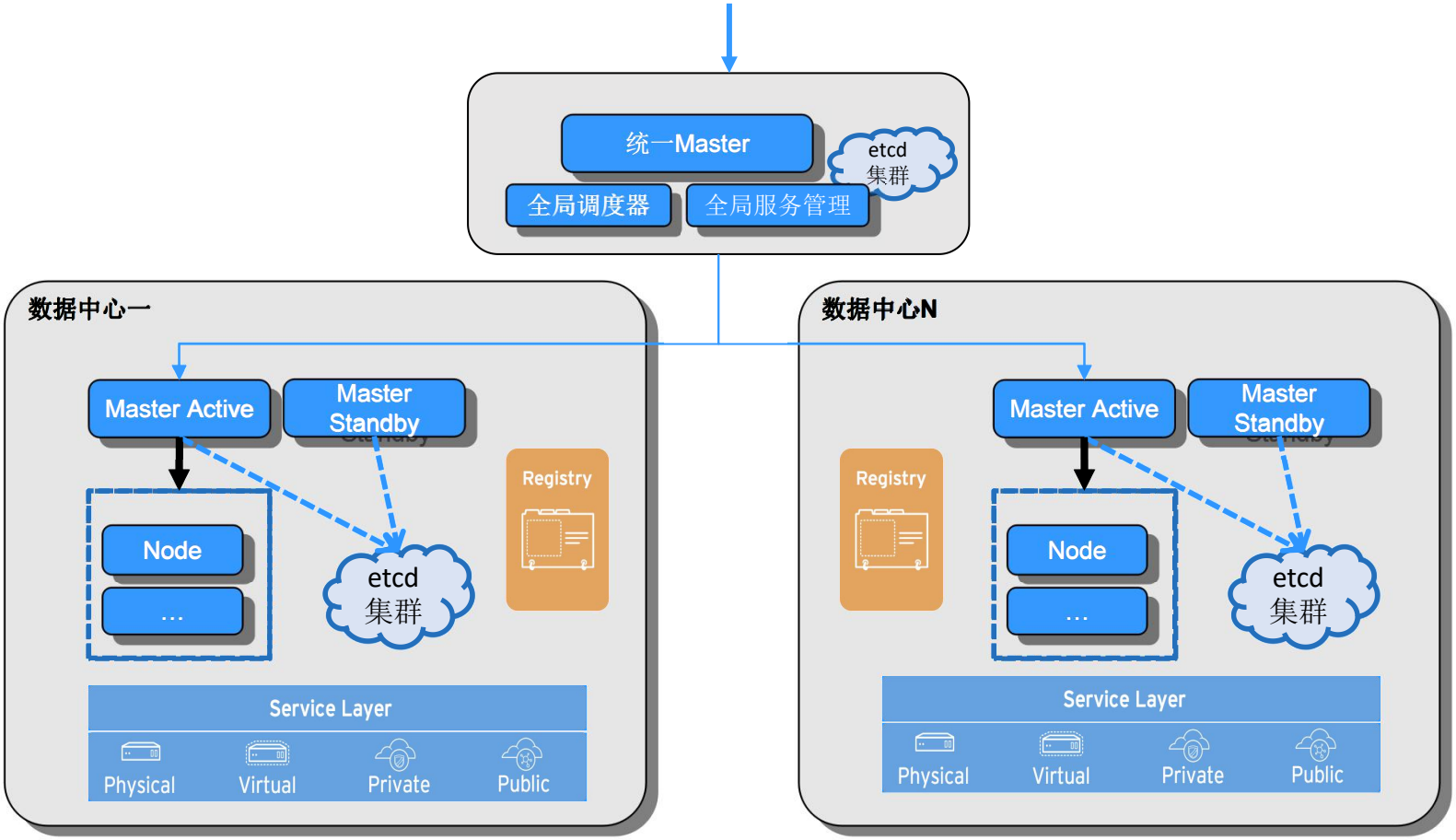
流程适配：多环境交互


开发环境
代码仓库
自动构建
编译代码
构建项目
构建镜像tar包（交付件）
持续集成环境
测试环境
镜像仓库
测试部署
应用镜像
应用镜像
测试数据
生产环境
跳转机（堡垒机）
生产部署
镜像仓库
应用发布

平台对接：容器监控与自动响应



容灾支持


统一Master
etcd
集群
全局调度器
全局服务管理
数据中心一
Master Active
Master
Standby
Node
...
etcd
集群
Registry
Service Layer
Physical
Virtual
Private
Public
数据中心N
Master Active
Master
Standby
Registry
Node
...
etcd
集群
Service Layer
Physical
Virtual
Private
Public

安全性

增强容器稳定性，
明确环境边界

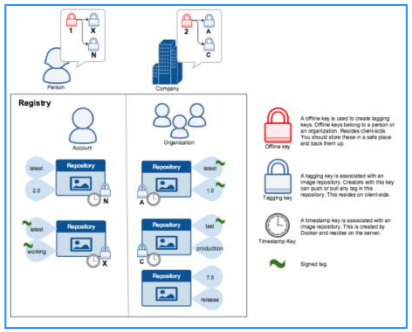
镜像安全扫描
镜像签名、认证镜像

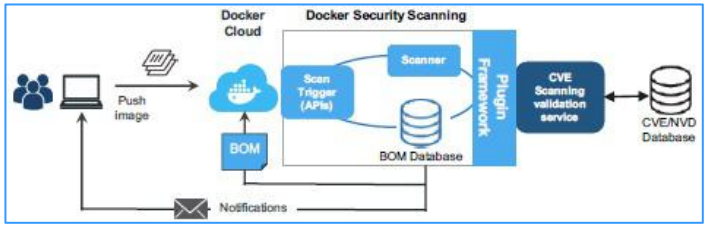
标准化容器/镜像

- 基础操作系统类容器
- 标准应用组件类容器
- 应用类容器
- 开发/测试仓库
- 线上运维仓库
- 标准镜像参数可自定义

自定义规格

- 调度参数定义
- 内存参数定义
- 磁盘容量定义
- 网络地址定义
- 环境变量定义
- 配置文件定义





可运维性：服务治理

调用链分析

- 跟踪服务调用链条，梳理服务依赖
- 分析服务的扇入/扇出信息，评估服务影响因子
- 服务故障根因定位与上下文分析
- 服务预警与风险评估

负载均衡/熔断保护

- 支持服务的自动化负载均衡
- 对故障服务进行熔断隔离保护
- 自动服务弹性，提升服务容量
- 服务版本管理，实现灰度发布

性能可视化

- 性能参数可视化
- 性能统计可视化
- 可用性信息可视化
- 性能对比
- 性能报表输出

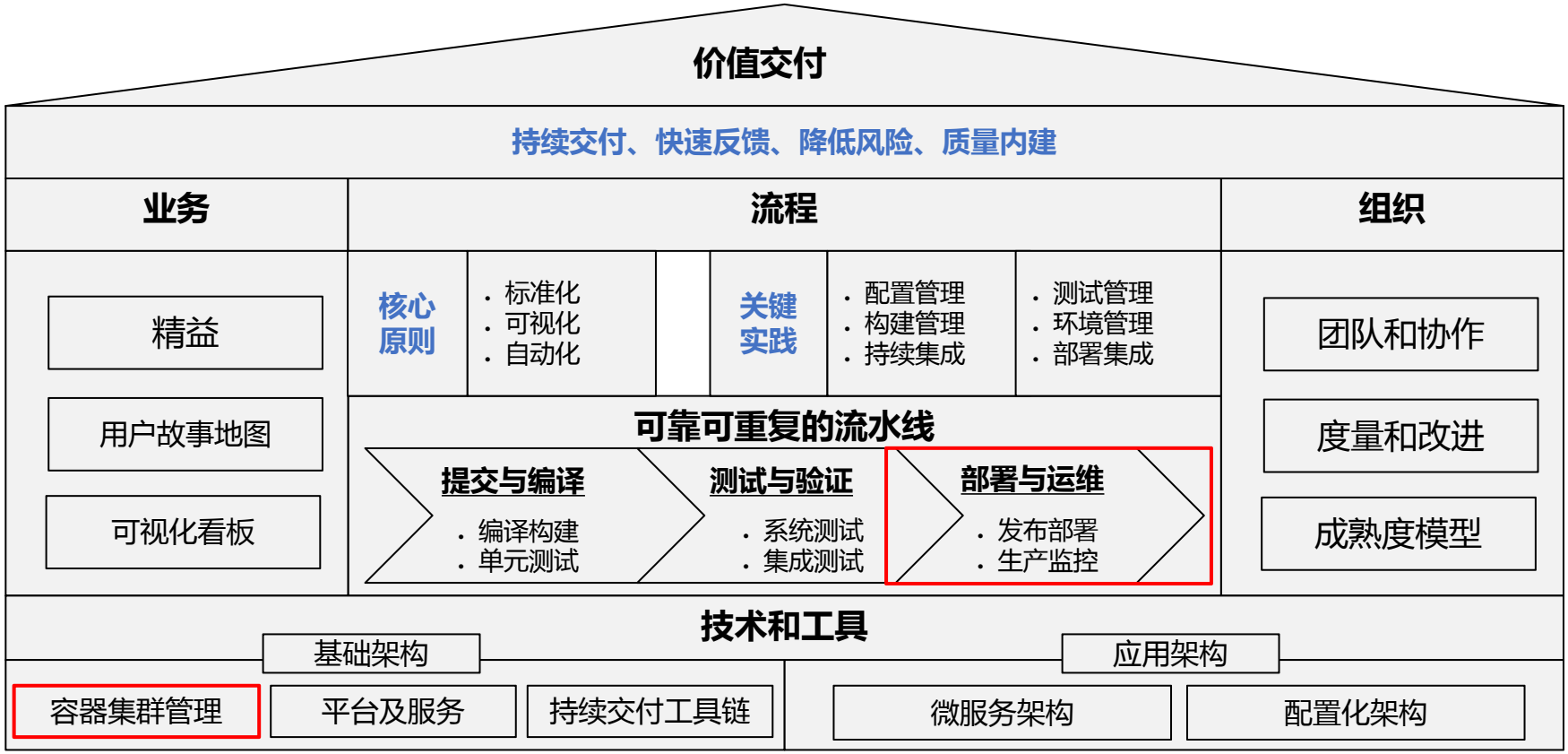
服务编排

- 支持服务网关功能
- 提供服务编排DSL语言，支持自定义服务
- 提供服务编排的生命周期管理等功能

DevOps落地：在传统行业的难

架构耦合	质量控制	交付和发布	局部优化
• 互影响、阻塞 • 交付无法并行	• 缺乏单元测试和集成测试 • 缺乏测试自动化	• 应用需求满足 • 性能	• 敏捷转型 • 建容器平台

DevOps全局化落地思路



From：《百度张乐-分享》

容器对DevOps的价值

交付件标准化！

交付动作标准化！

又轻又快！

简单

快速

可靠

可重复

环境无关

DevOps实施落地步骤建议

需求驱动

试点先行

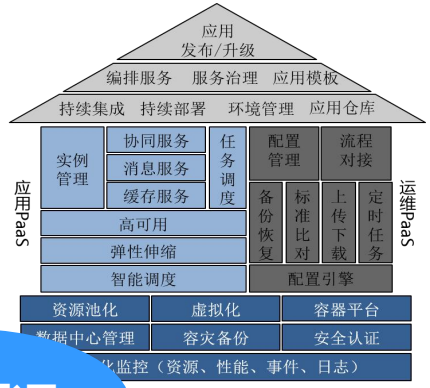
独立团队

先架构
再平台
后流程

博云概况与产品技术定位

客户群体：覆盖主要金融机构

产品体系：



云应用平台：

1. 应用编排
2. 服务治理；
3. 灰度发布；
4. 持续部署；

云运维平台：

1. 监控&告警
2. 配置/性能数据库；
3. 配置自动化；
4. 日志/故障数据库；

博云概况

- 公司拥有博士3人、硕士15人，核心成员多来自于微软、IBM、华为、阿里巴巴、百度等一流企业
- 2015年公司和中国科学院软件研究所签署战略合作协议
- 中科院软件所魏俊研究员、中科院计算所包云岗研究员为公司资深技术顾问
- 拥有完善的研发、方案、实施、运维人才团队，提供覆盖金融云全链条业务的服务能力
- 具备私有云建设、PaaS平台构建、服务治理平台管理、云运维的全栈技术与解决方案能力

- 2016年度优秀云管理平台提供商-商业资讯机构
- 2016年中国IT用户满意度调查金融及政府行业国产PaaS用户推荐品牌-中国质量协会用户委员会
- 2015-2016年度全球创新产品奖-全球云计算大会
- 数据中心联盟高级会员
- 云计算开源产业联盟企业会员单位
- 红帽redhat合作伙伴
- 京东云渠道合作伙伴

市场地位：业界领先的PaaS产品/服务提供商

云计算开源产业联盟

IT大咖说 知识分享平台

博云 BoCloud