



Ctrip
Continuous Delivery

携程持续交付之路

吴毅挺

Agenda

- 持续交付简介
- 携程持续交付现状&技术债
- 携程持续交付理念&产品
- 大迁徙
- 下一代持续交付平台

Code

Configuration

App

Data

Runtime

OS

Compute

Storage

Network

如何让变更越频繁的变得越容易越快?
稳定性 vs 灵活性
Ops vs Dev

变更
越来越频繁

何为持续交付

- 缩短产品从IDEA到交付的时间
- 开发、测试、发布高效迭代，增量发布
- 涵盖
 - scm,编译、打包、发布、持续集成、自动化测试、测试环境管理、生产灰度发布、回滚、扩容、缩容、配置服务等

携程持续交付现状

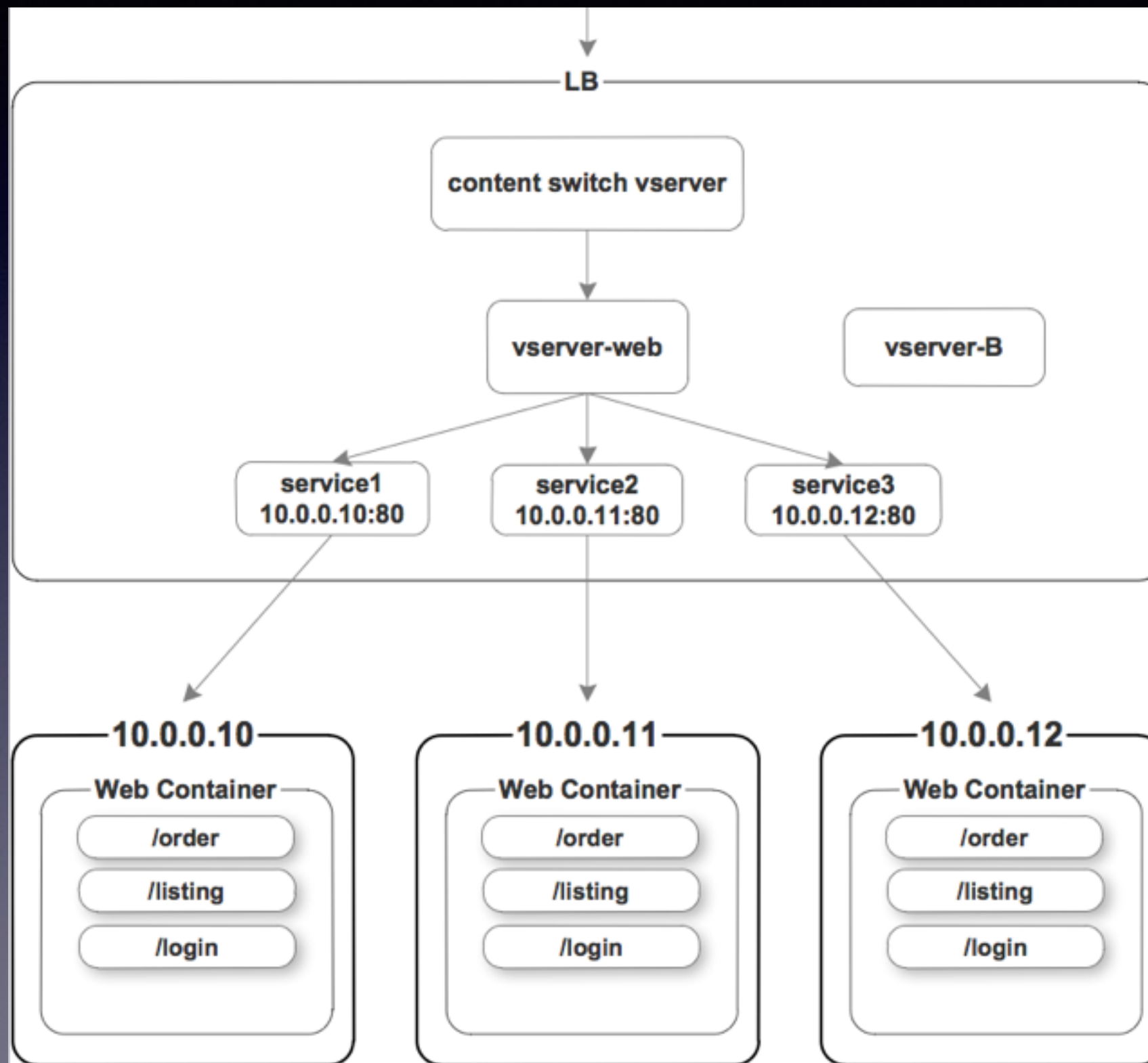
20+ BU 3000+ R&D

每周发布次数: FAT ~10K, UAT ~5k, Prod ~2K

技术栈: Java/Php/Python/.NET/Go

OS: Windows/Linux

部署模式



App细粒度
以VM/BM为单位

- 拉入拉出
- 扩容
- 混合部署

持续交付技术债

- 以VM/BM为单位拉入拉出 -> 发布排队/业务交付慢
- 混合部署 -> 缺乏有效资源隔离
- 新应用上线 -> 依赖人工分配
- 元数据不统一 -> 漏发
- FAT/UAT/Prod环境不一致
- 系统分散, 申请费力度高
-

持续交付技术债

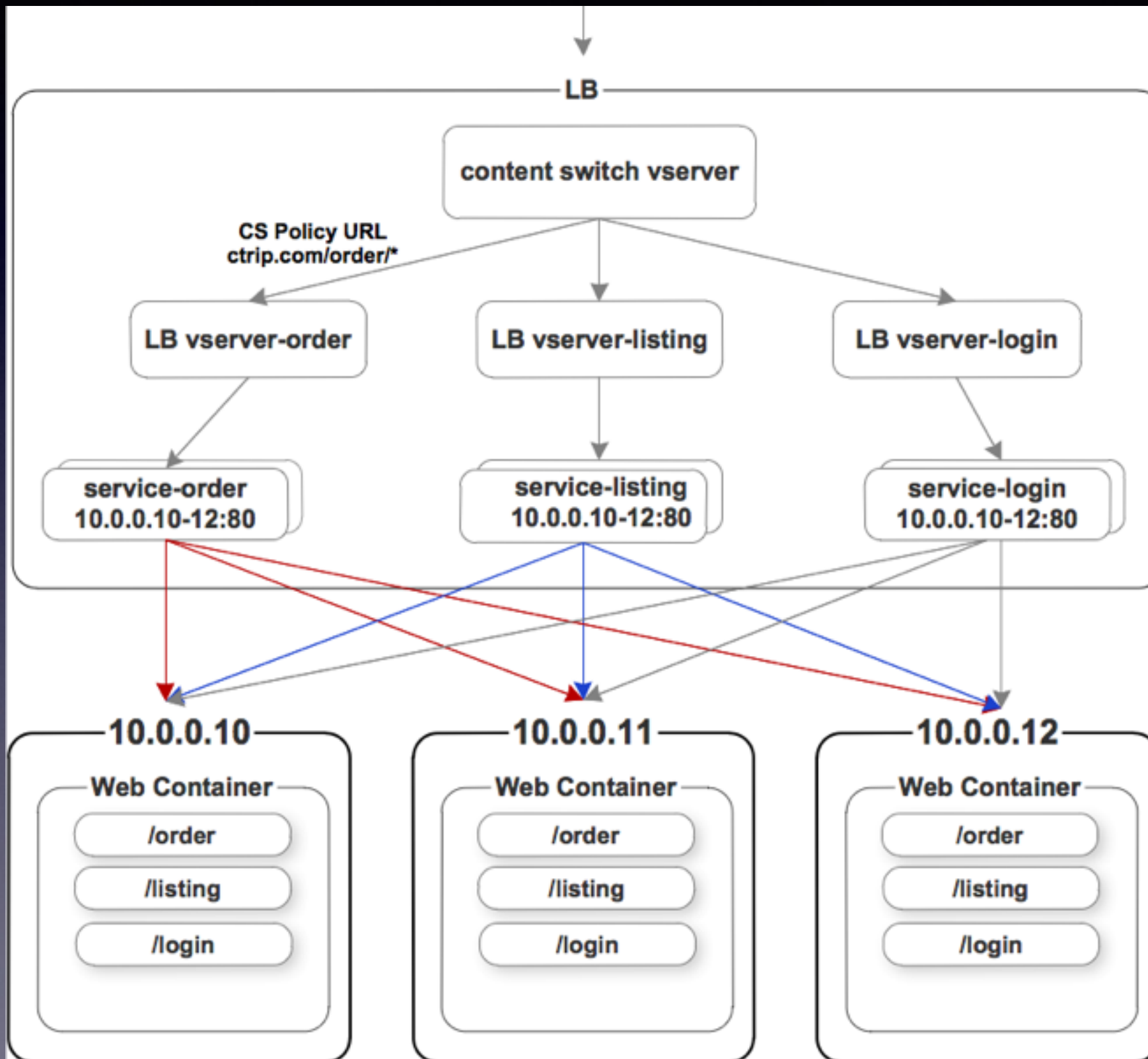
- V1/V2/V3
- 回滚、紧急回滚、超级紧急回滚
- LD发布
- 非Clean发布

如何做到 想發就發？

App独立部署

- 单容器单应用
 - 50% Windows App如何解决?
- 单VM单应用
 - 应用粒度太细
 - vm数量太多，资源利用率低
- 容器 + 混合部署独立路由

App独立路由

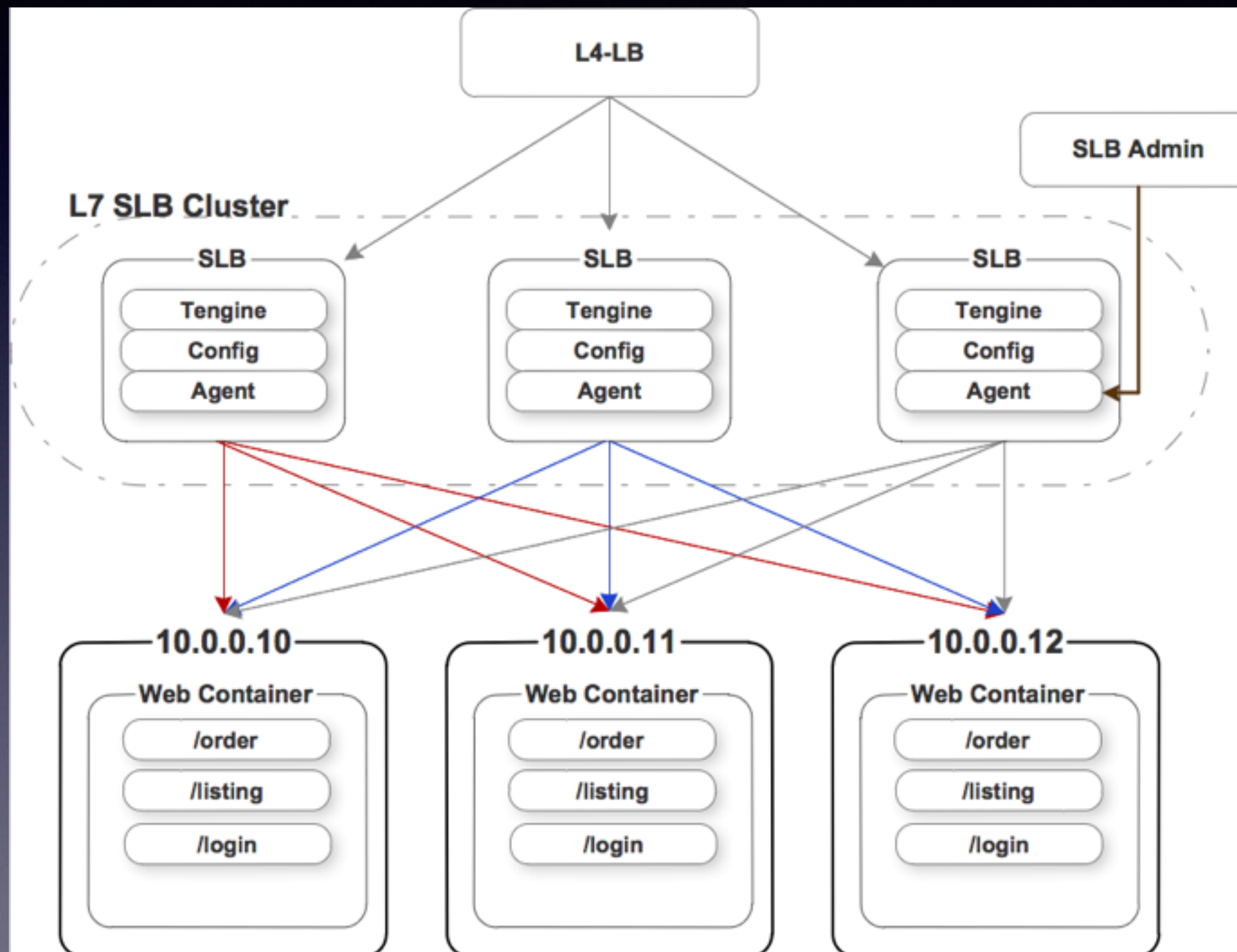


每个App独立cs vserver/
lb vserver

硬件LB

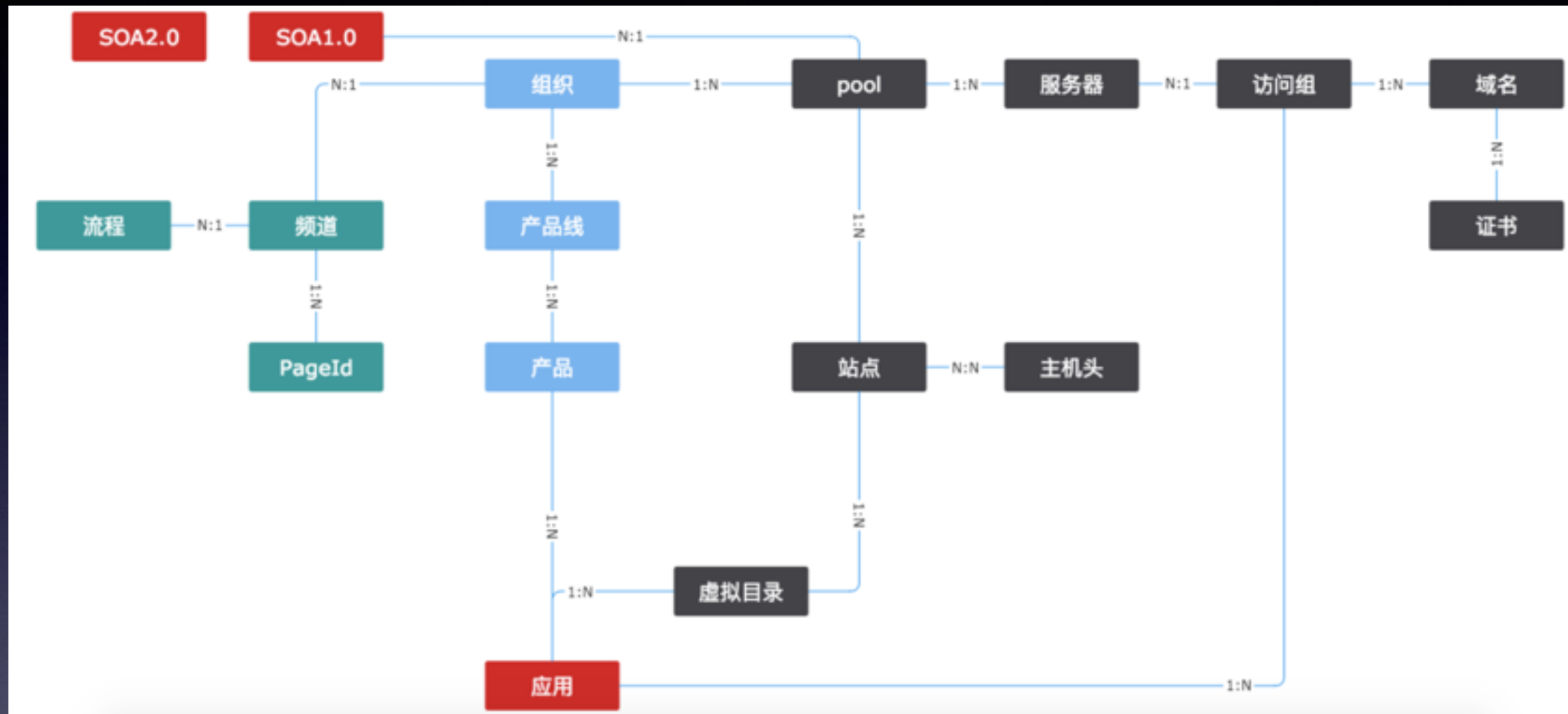
- API可靠性
- 扩展性
- 每种对象的最大上限

App独立路由



- L7 SLB
 - 水平扩展
 - 单集群扩容
 - Sharding
 - Restful API
 - Healthy check
- L4-LB -> LVS?

App元数据模型



统一模型

统一数据源

不同平台之间交互规范

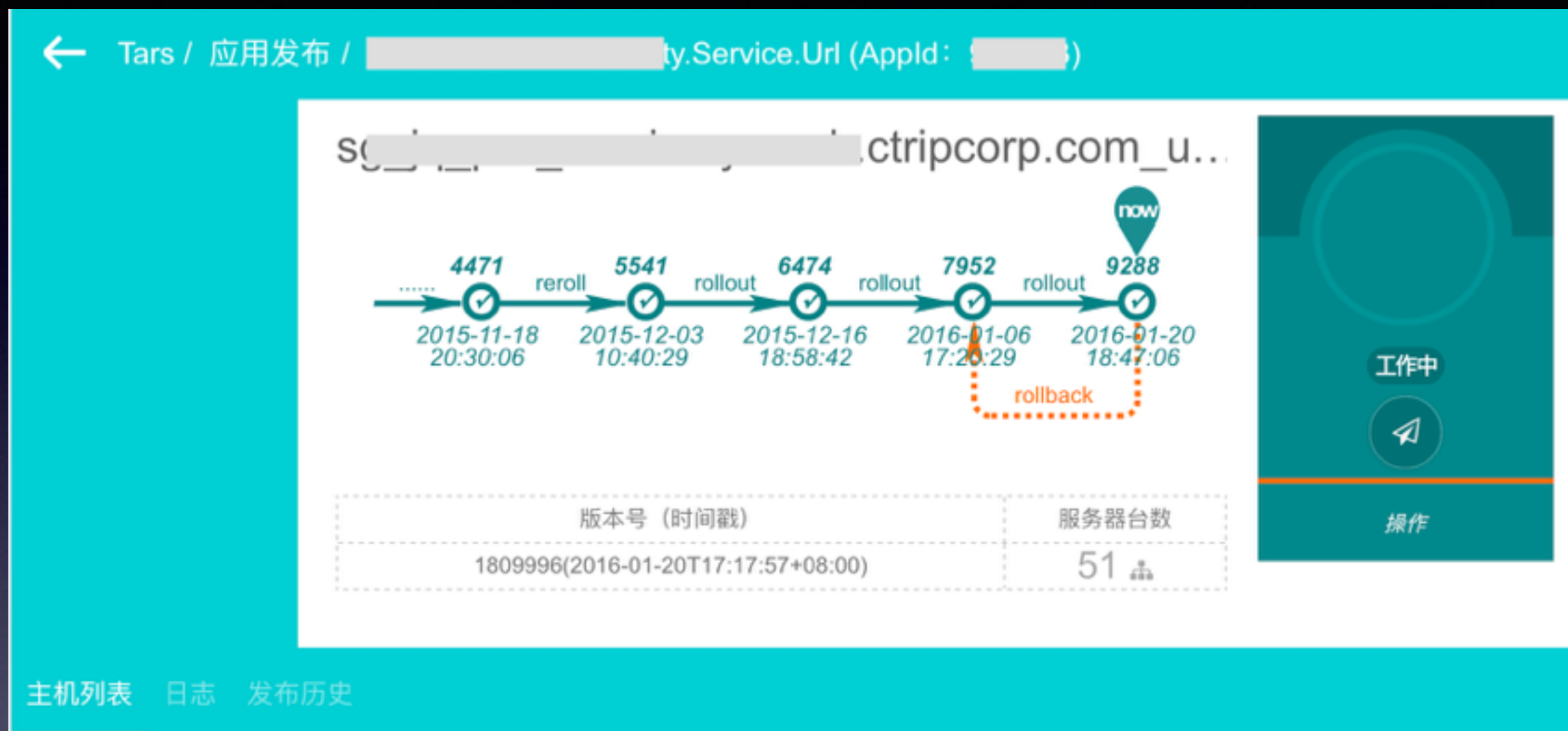
不同环境之间数据同步

明确元数据Owner

解耦

- 系统解耦
 - 不同的发布类型 由不同的系统/团队 去实现
 - 各系统之间遵循统一的元数据模型、数据源、协议
- 元数据解耦
 - cms 数据准确性 由哪个团队负责?
 - 哪些数据应该在cms, 哪些由各个平台自行维护?
- Agent解耦

可视化



主机列表

刷新

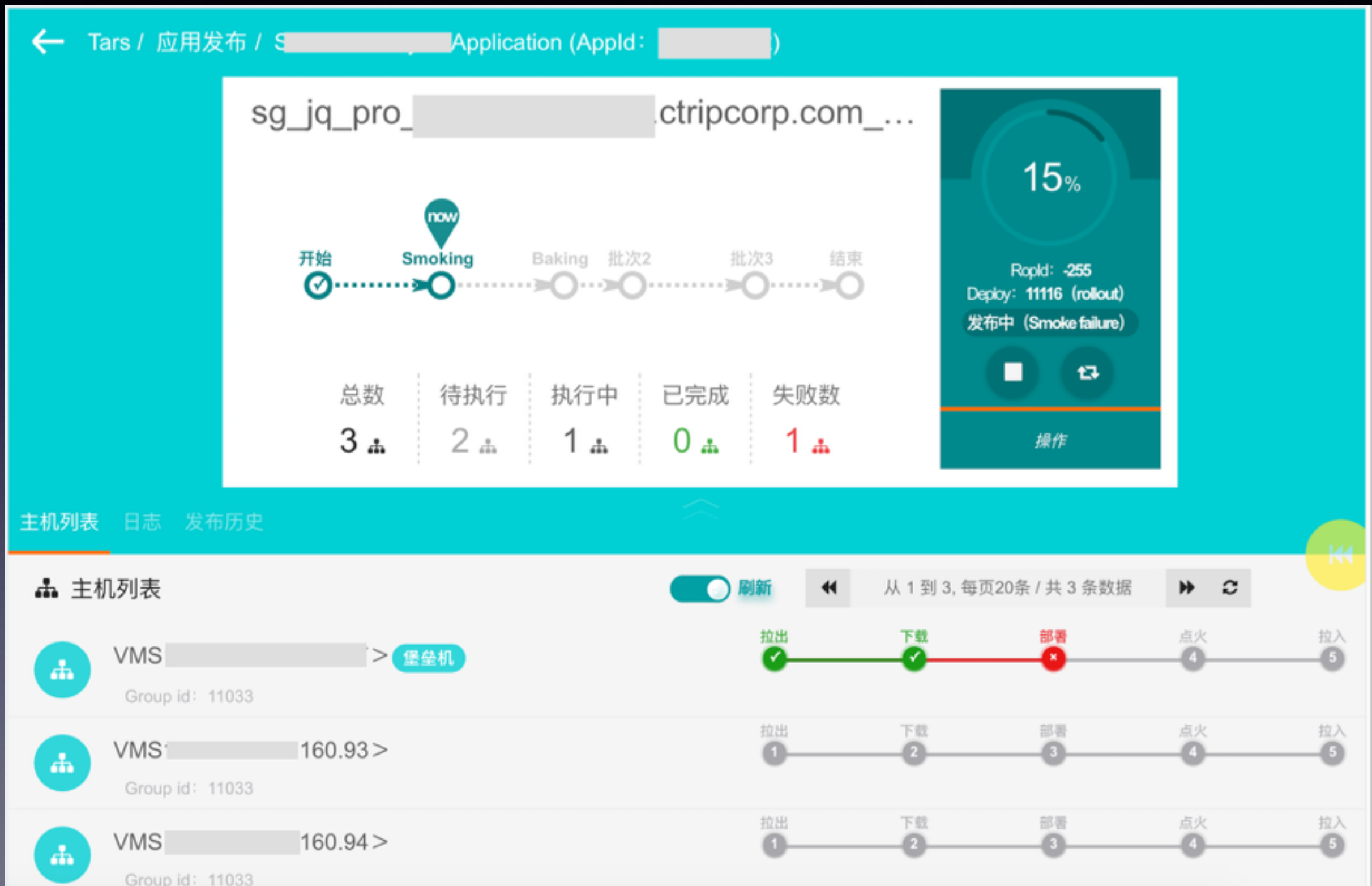
从 1 到 20, 每页20条 / 共 51 条数据

VM [redacted] 119.250 >
Group id: 394

VM [redacted] 119.1 >
Group id: 394

VM [redacted] 119.245 >
Group id: 394

可视化



携程持续交付理念

- 全发布类型/全流程/全环境、一站式持续交付平台
 - 全类型: 静态资源/DB/Android/iOS/Web/Service/Job/...
 - 全流程: 代码分支、版本控制、审批...
 - 全环境: 功能测试、集成测试、生产多IDC
 - 一站式: 涵盖整个App生命周期
- App独立发布(部署/路由)
- 建立App规范、梳理元数据
- 发布可视化

CD Portal

 CDPortal

[首页](#) [交付类型](#) [常用申请](#) [发布管理](#) [技术支持](#) [反馈](#) [帮助](#)

交付类型

Web应用

.Net >

Tomcat + Java >

Service交付

SOA 2.0 (.Net) >

SOA 2.0 (Java) >

Job Center

JobWS >

Linux Job >

Java应用

Java App >

静态资源 >

移动应用 >

SOA 2.0 (Java)

应用列表

基于Java平台开发的SOA 2.0应用



SOA Repository

SOA Repository是SOA服务的治理系统
您可以在SOA Repository站点里注册和管理您的SOA服务、管理文档、管理服务配置等



AppId申请

AppID是应用的唯一标识
申请AppID是应用发布上线前的必备步骤



Pool申请

Pool是指一组物理服务器的集合
申请Pool是应用发布上线前的必备步骤
如果您希望对Pool做服务器申请或扩容的操作，请点击“机器扩容”



应用上线申请

应用上线申请是应用发布之前的最后一步

大迁徙

- ~5000 + App/ ~3000 Dev/QA / 20 + BU
- 按新标准重新部署所有环境
- 10000+核 迁徙资源池
- 平滑迁徙，对业务无影响

大迁徙

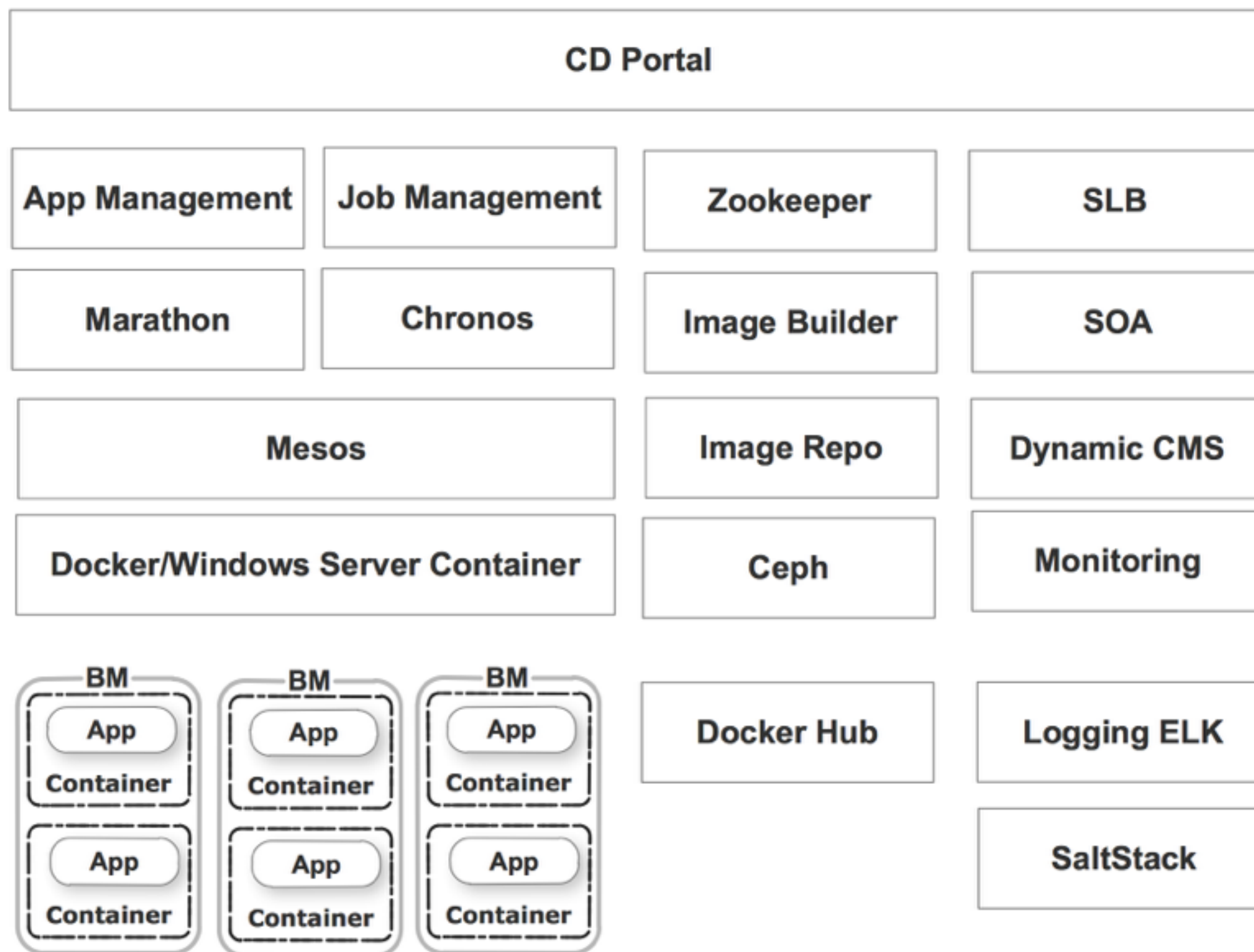
大迁徙



基于容器的持续交付

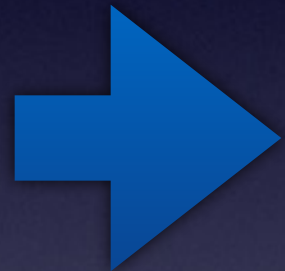
- 不可修改的基础设施
- 基于镜像发布&动态调度
 - e.g. Linux Job
- 配置中心
- 快速交付:即申请即用

基于容器的持续交付



携程持续交付之路

LD发布
Pool部署
无版本
缺乏规范
发布负责人
强流程
静态部署
工作日级交付



App独立发布
App逻辑隔离
包版本
建立规范
PD发布
轻流程
超分/迁移
小时级交付



容器镜像发布
App容器隔离
镜像版本
规范+高度抽象
PD自助
化流程于无形
动态调度
分钟级交付

Q&A



Ctrip
Continuous Delivery

说走就走，想發就發！