

上交所容器技术微服务架构技术实践

上交所技术有限责任公司 孙长昊

零、内容

- 一. 我们
- 二. 目标
- 三. 方向
- 四. 实践
- 五. 思考
- 六. 展望

一、我们

上交所技术公司技术开发总部

- > 历史
- 职责
- 主业



一、我们

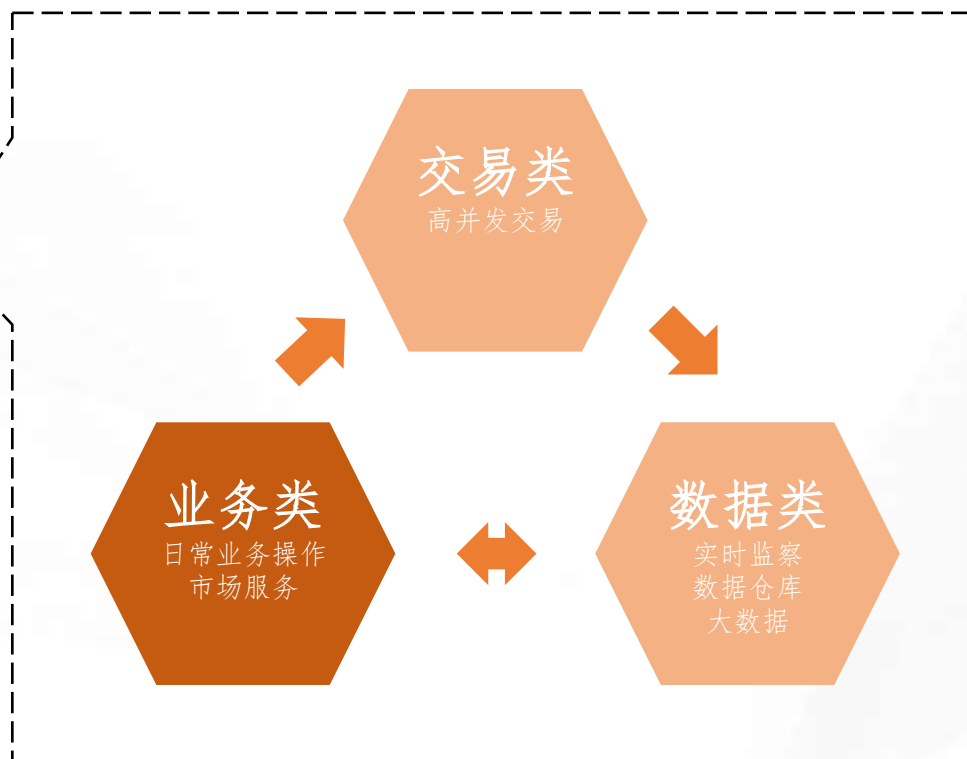
上交所技术公司技术开发总部

- 历史
- 职责
 - 为上交所提供核心技术服务
 - 为市场参与者提供技术和创新产品服务
- 主业

一、我们

上交所技术公司技术开发总部

- 历史
- 职责
- > 主业



二、目标

实现

> 技术体系完备化

- 软件过程标准化
- 服务发布高效化

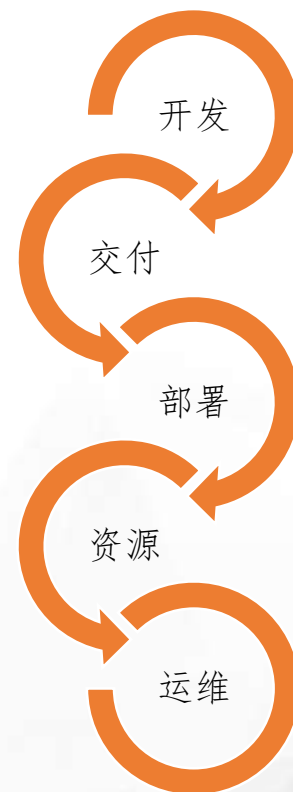
- 系统架构
- 技术储备

二、目标

实现

- 技术体系完备化
- > 软件过程标准化
- 服务发布高效化

- 方案
- 规范
- 工具



二、目标

实现

- 技术体系完备化
 - 软件过程标准化
 - **> 服务发布高效化**
- 精益
 - 自动化

三、方向

方向

> 微服务

- Docker
- DevOps

优点

- 易开发
- 易运维
- 易扩展

局限

- 分布式
- 集成复杂
- 部署问题

三、方向

方向

- 微服务
- > **Docker**
- DevOps

优点

- 轻量级虚拟化
- 标准化运行时
- 易于测试集成

局限

- 运行环境限制
- 使用场景限制



三、方向

方向

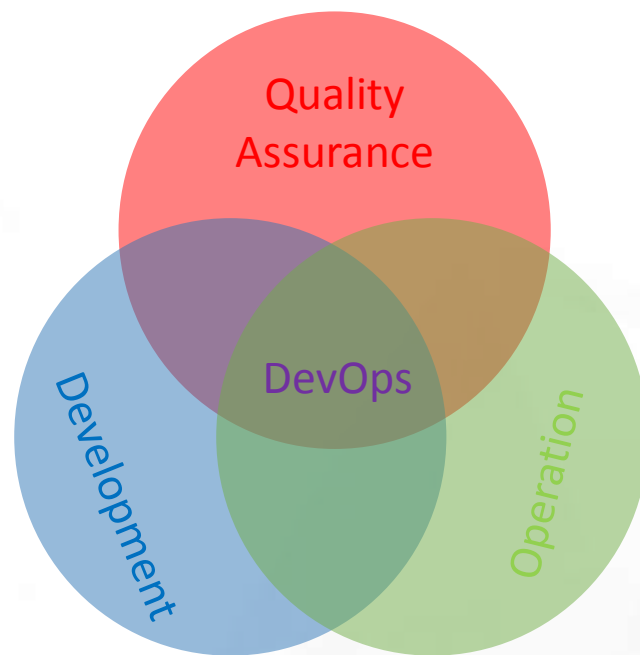
- 微服务
- Docker
- > DevOps

优点

- 降低风险
- 快速发布
- 强自动化

局限

- 协作方式
- 环境氛围



四、实践

技术体系完备化
软件过程标准化
服务发布高效化

四、实践

技术体系完备化

- 基于**Docker**的**PaaS**平台架构
- 微服务架构

四、实践

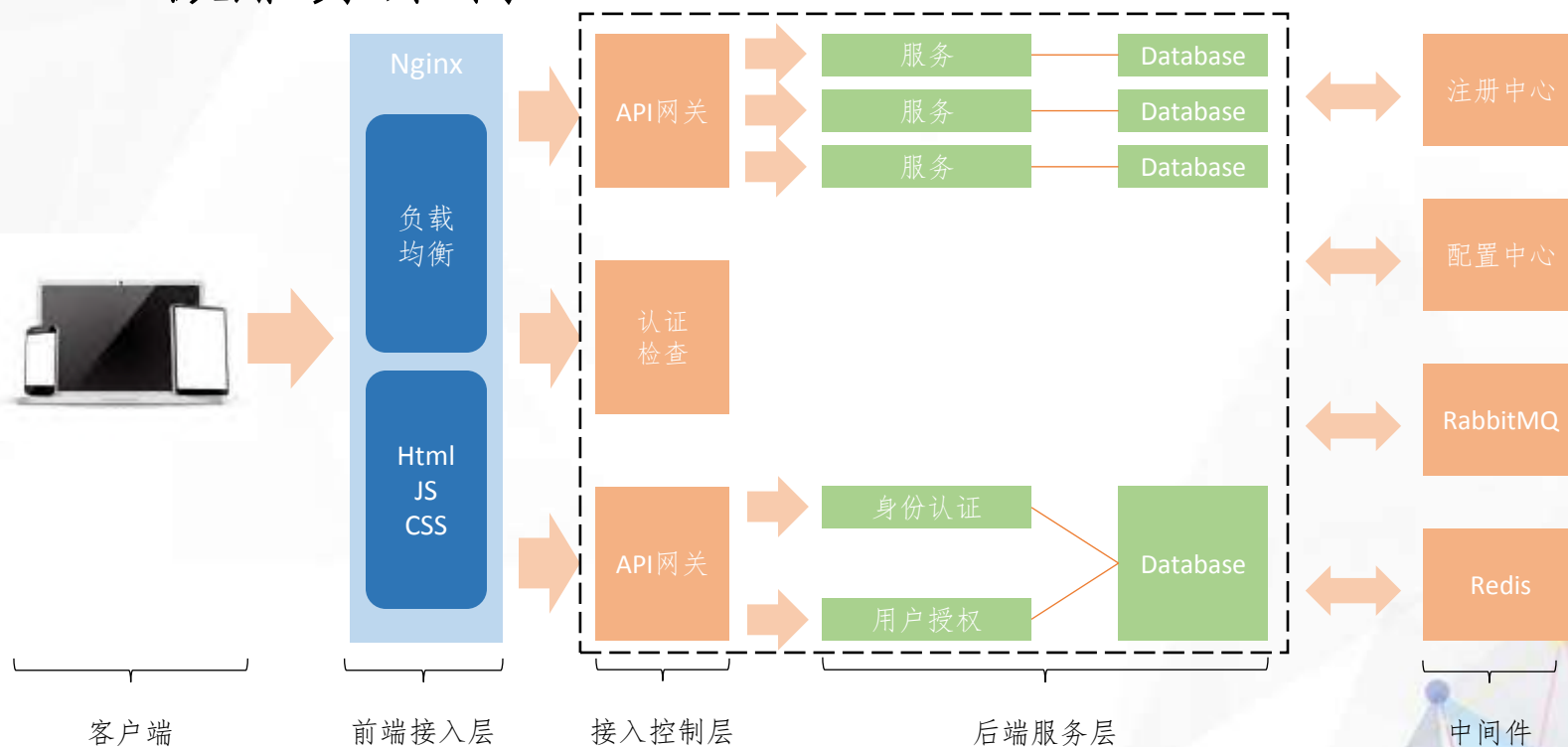
技术体系完备化

> 基于 Docker 的 PaaS 平台架构



四、实践

技术体系完备化 > 微服务架构



四、实践

软件过程标准化

- 基于**Docker**的环境管理
- 微服务架构规范
- 微服务与**Docker**的结合
- 基于微服务架构的持续集成

四、实践

软件过程标准化

> 基于**Docker**的环境管理



四、实践

软件过程标准化

> 微服务架构规范

a. 接口管理

b. 数据格式

c. 服务鉴权

d. 技术栈规约

- 路由-API网关
- 格式-RESTful

四、实践

软件过程标准化

> 微服务架构规范

a. 接口管理

b. 数据格式

c. 服务鉴权

d. 技术栈规约

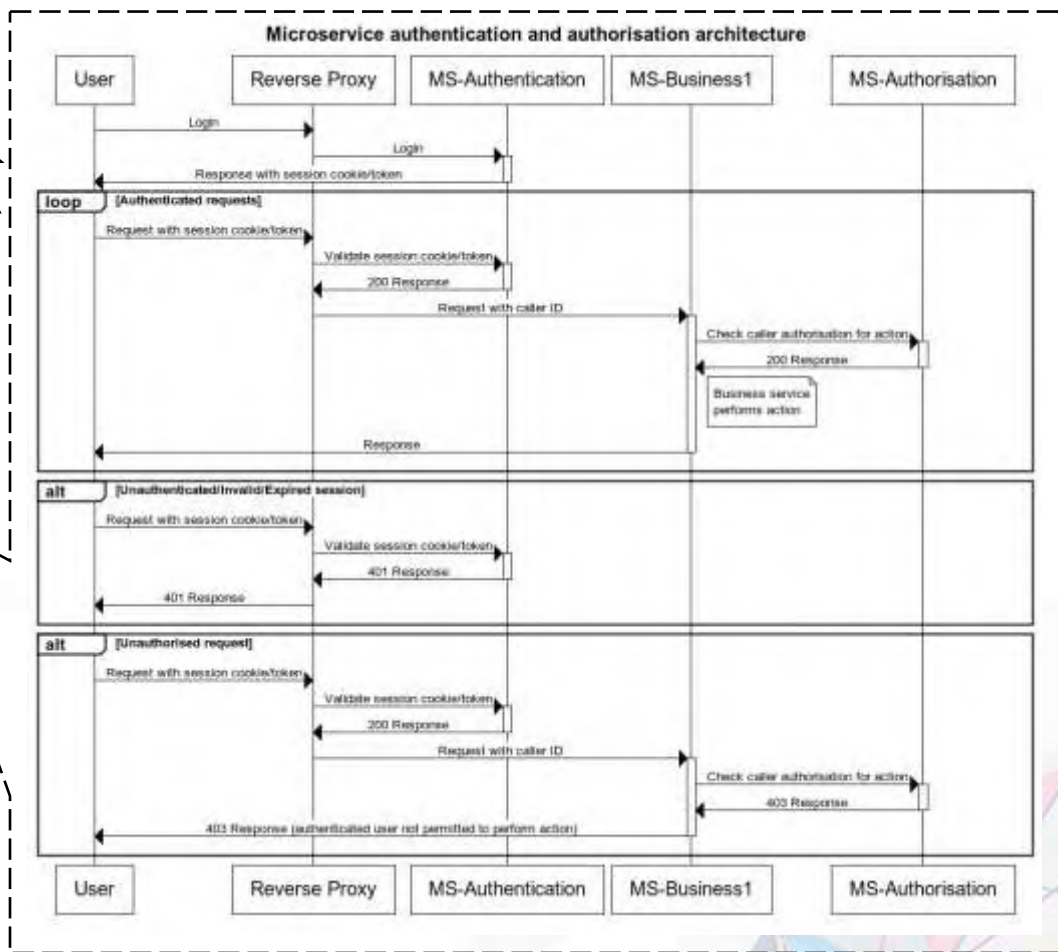
- HTTP Status + ResponseBody
- HTTP Status + BizCode
 - {code:12, message: 'xxxxx'}

四、实践

软件过程标准化

> 微服务架构规范

- a. 接口管理
- b. 数据格式
- c. 服务鉴权
- d. 技术栈规约



四、实践

软件过程标准化

➢ 微服务架构规范

a. 接口管理

b. 数据格式

c. 服务鉴权

d. 技术栈规约

- 组件名
- 组件版本
- 持续验证
- 持续更新

四、实践

软件过程标准化

> 微服务与**Docker**的结合

- a. 服务端口管理
- b. 镜像仓库管理
- c. 环境发布管理
- d. 服务依赖保护

- 服务端口可动态配置
- 组件端口可动态配置
- 端口清单

四、实践

软件过程标准化

> 微服务与**Docker**的结合

- a. 服务端口管理
 - b. 镜像仓库管理
 - c. 环境发布管理
 - d. 服务依赖保护
- 归档
 - 清理
 - 告警

四、实践

软件过程标准化

> 微服务与**Docker**的结合

- a. 服务端口管理
 - b. 镜像仓库管理
 - c. 环境发布管理
 - d. 服务依赖保护
- 一键发布
 - 环境分配
 - 环境回收

四、实践

软件过程标准化

> 微服务与**Docker**的结合

a. 服务端口管理

b. 镜像仓库管理

c. 环境发布管理

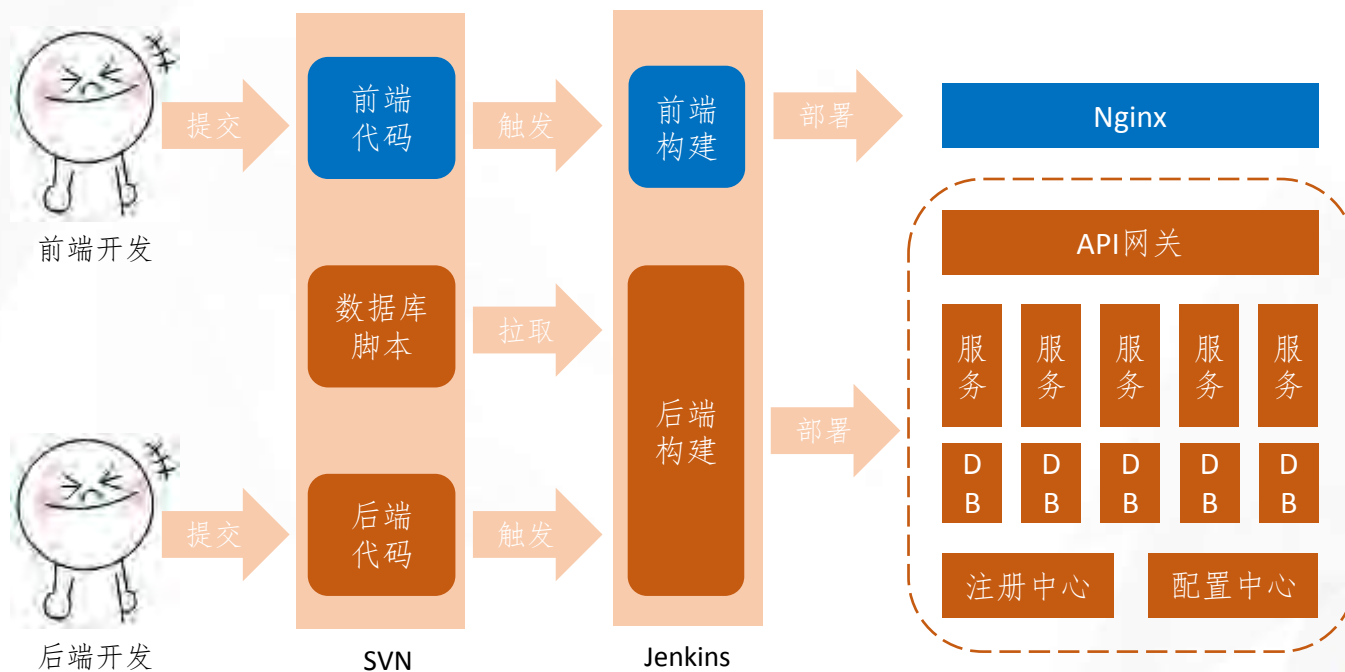
d. 服务依赖保护

- 熔断
- 隔离
- 限流

四、实践

软件过程标准化

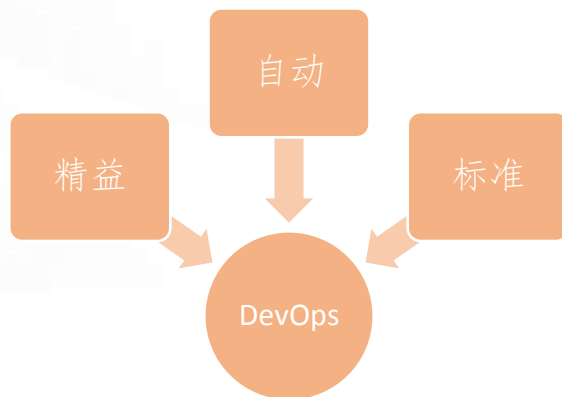
> 基于微服务架构的持续集成



四、实践

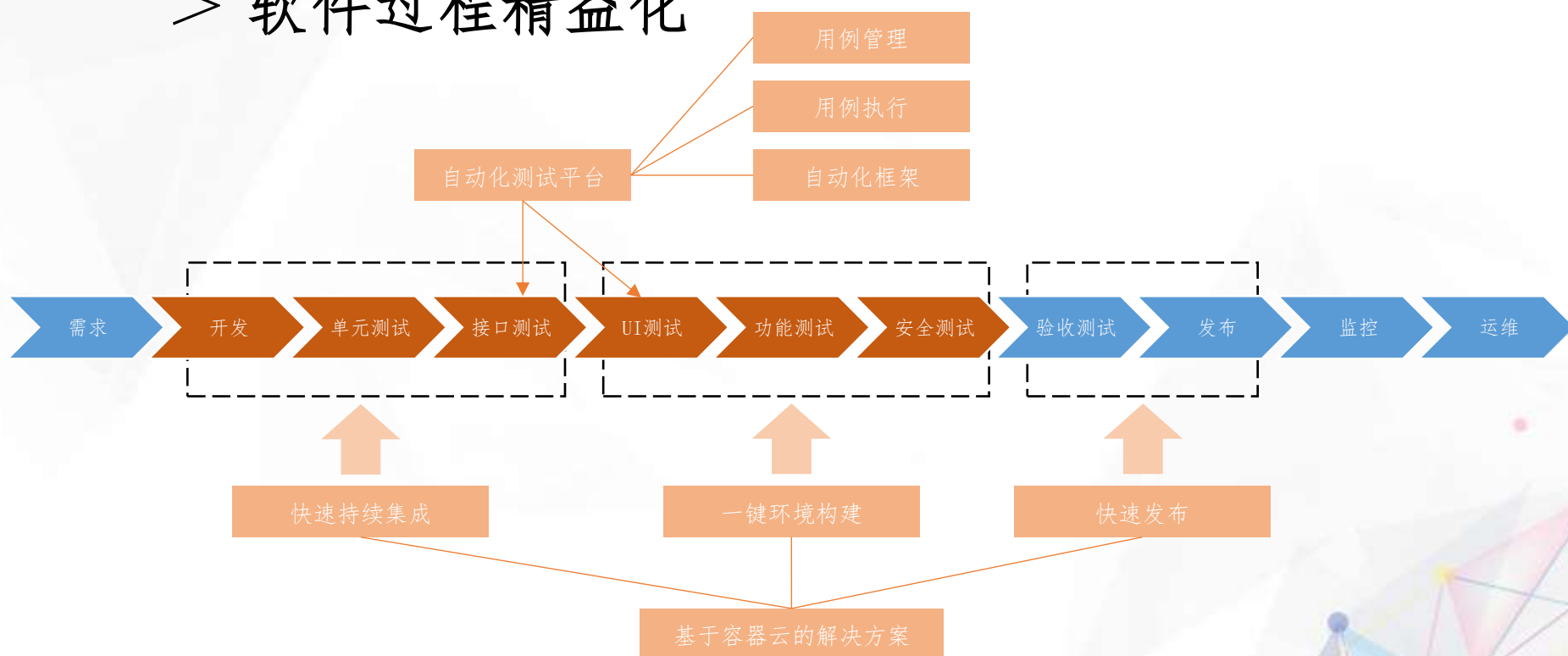
服务发布高效化

- 软件过程精益化
- 软件过程自动化
- 基于微服务与**Docker**的持续交付平台



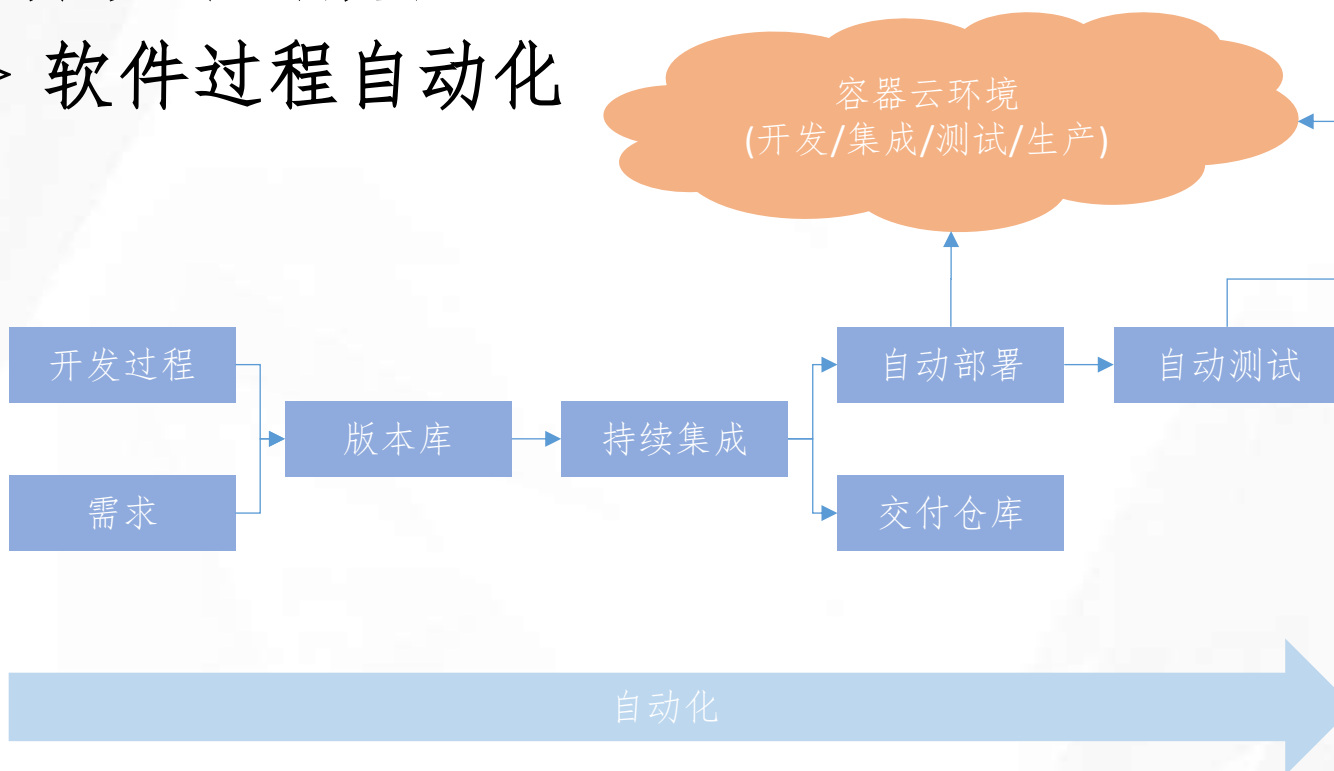
四、实践

服务发布高效化 > 软件过程精益化



四、实践

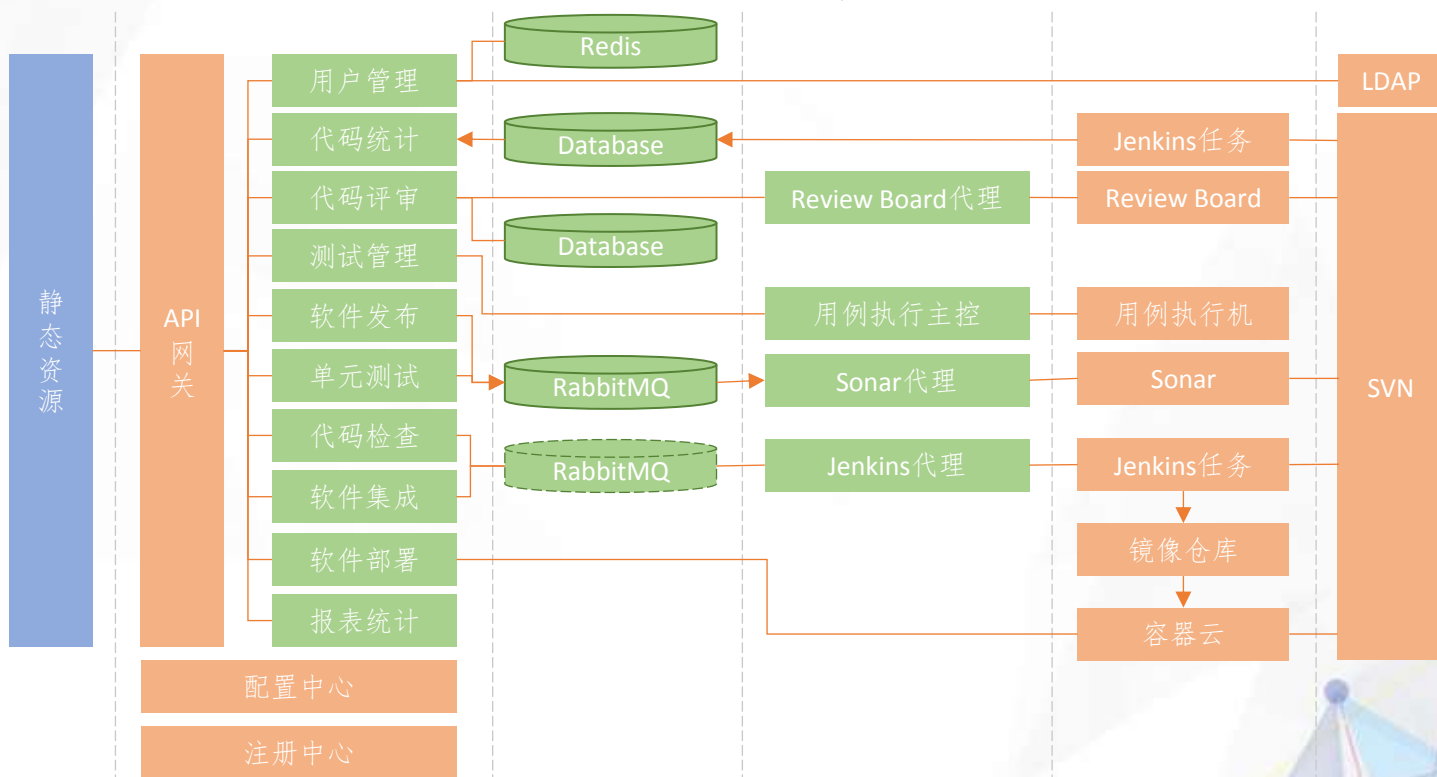
服务发布高效化 > 软件过程自动化



四、实践

服务发布高效化

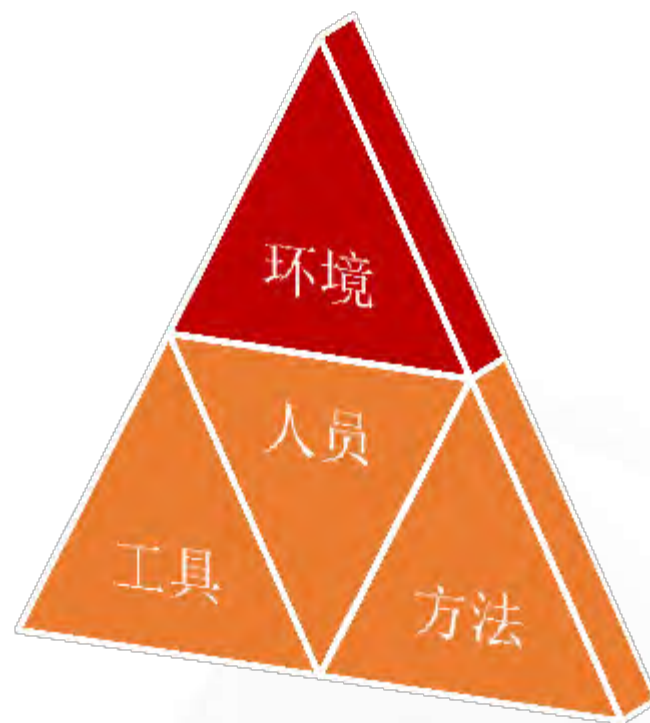
> 基于微服务与Docker的持续交付平台



五、思考

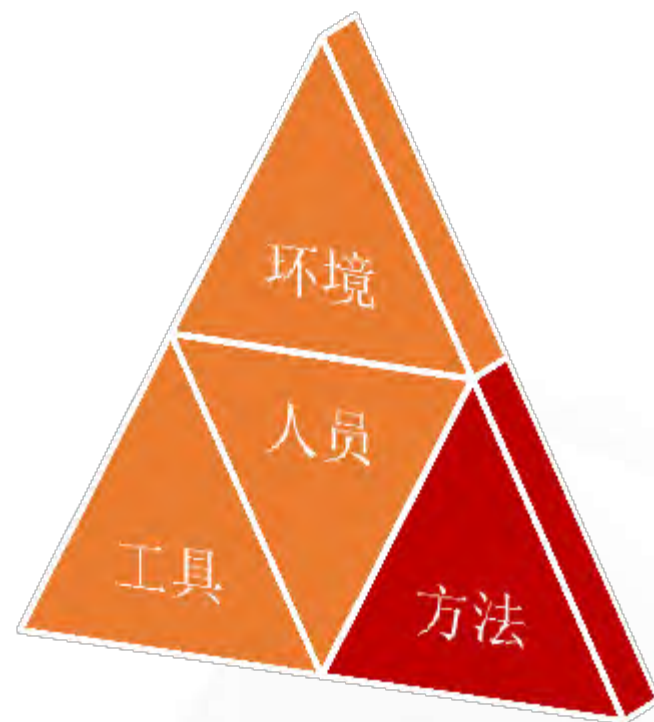
> 敏捷的组织架构-环境

- a. 产品导向
- b. 快速敏捷
- 微服务最佳实践-方法
- 开发运维工程师-人员
- DevOps工具储备-工具



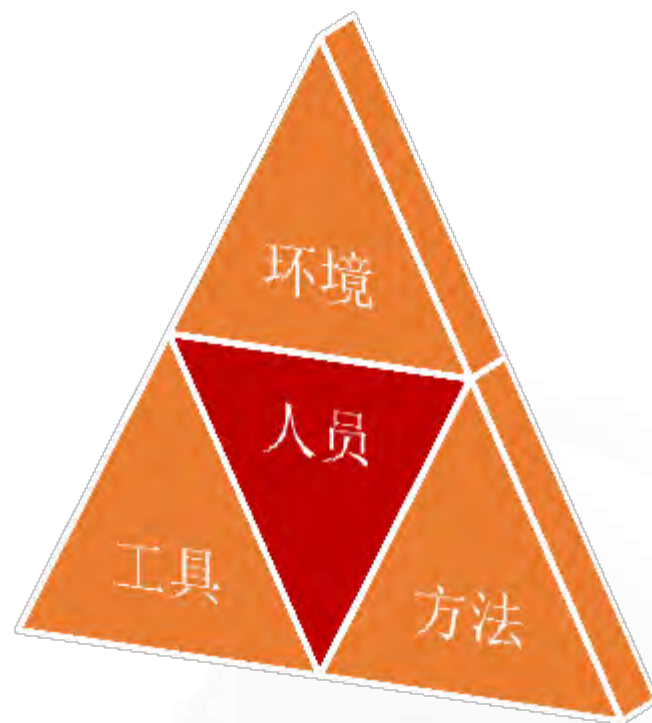
五、思考

- 敏捷的组织架构-环境
 > 微服务最佳实践-方法
 - a. 不同场景
 - b. 配套方案
- 开发运维工程师-人员
- DevOps工具储备-工具



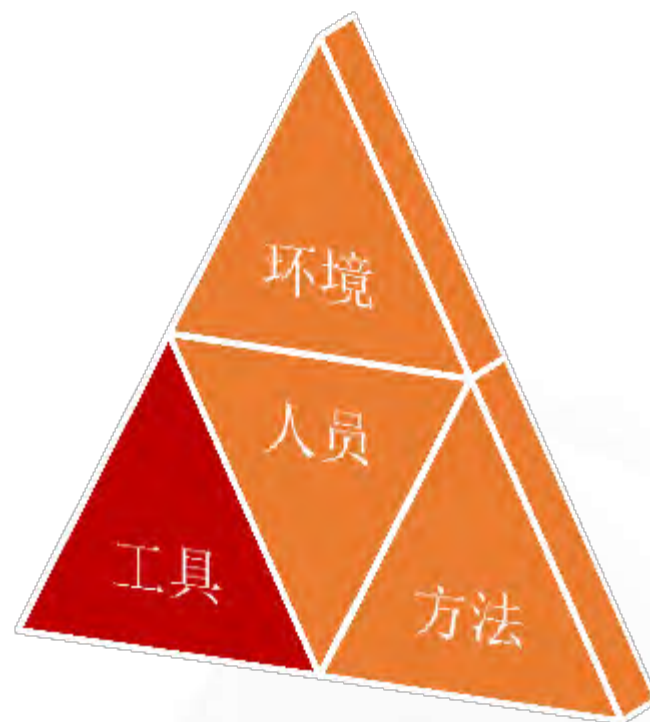
五、思考

- 敏捷的组织架构-环境
- 微服务最佳实践-方法
- > 开发运维工程师-人员
- DevOps工具储备-工具



五、思考

- 敏捷的组织架构-环境
 - 微服务最佳实践-方法
 - 开发运维工程师-人员
- > **DevOps工具储备-工具**



THANKS

SequeMedia
网络传媒

IT168
IT168.COM

ChinaUnix

ITPUB
www.itpub.net