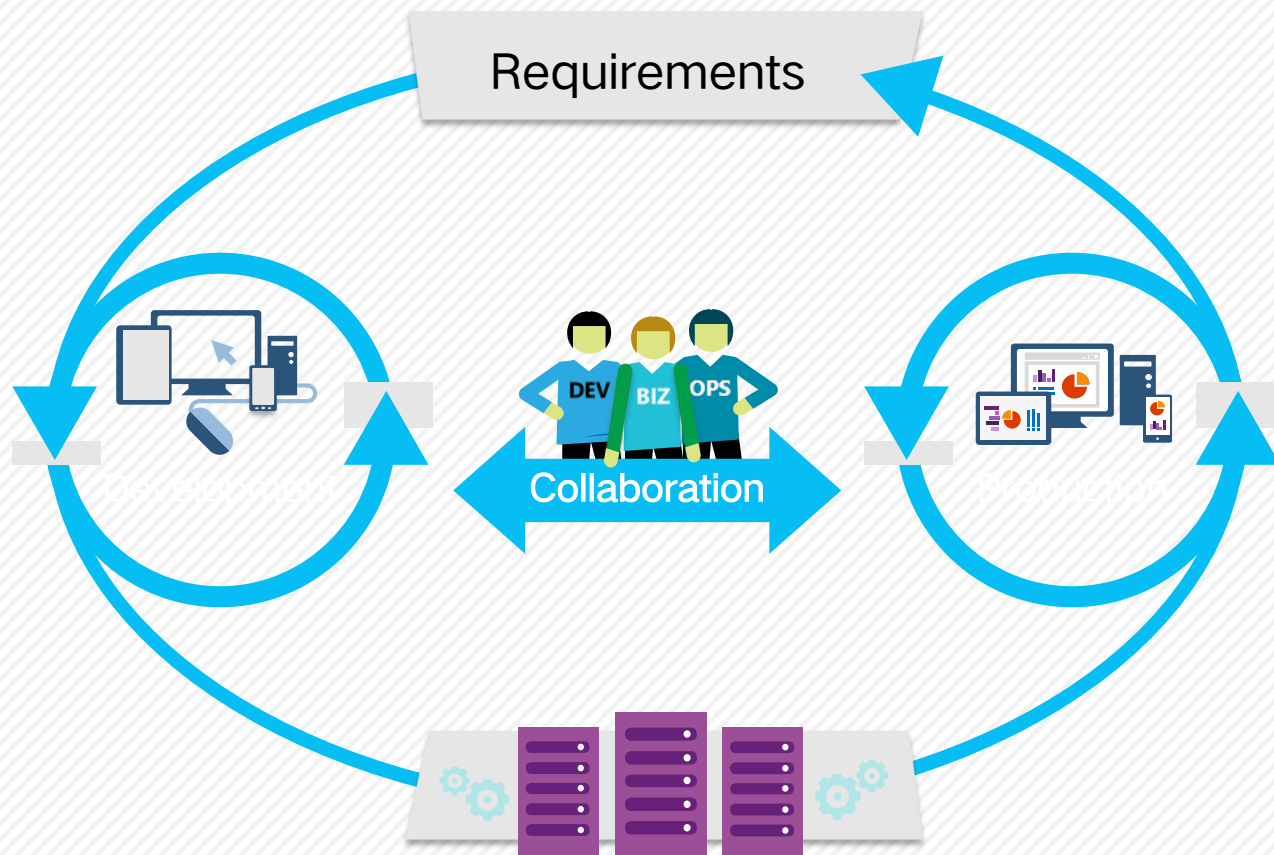


DevOps解决方案

DevOps on Cloud.

什么是DevOps?

DevOps 是一种软件交付理念和方法，以协作测试、打包和部署软件为核心，目的是增强软件版本的发布规律和可靠性。



传统研发运维的问题和挑战

应用程序发行速度增加，需要快速进行生产环境部署

构建协作与自动化能力缺失，难以适应变更

应用监控缺失和管理过程不规范

实施更改要求的交付周期变短

开发团队，QA团队和IT运营团队关注点不同，协作不畅

不同环境缺乏标准化管理，以便适应更改

DevOps的优势

更快速地针对客户进行创新，适应不断变化的市场

业务
创新

规模

大规模运行和管理您的基础设施及开发流程

提高发布的频率和速度，更快速地进行创新并完善产品

交付
速度

合作
效率

开发人员和运营团队密切合作，流程相互融合

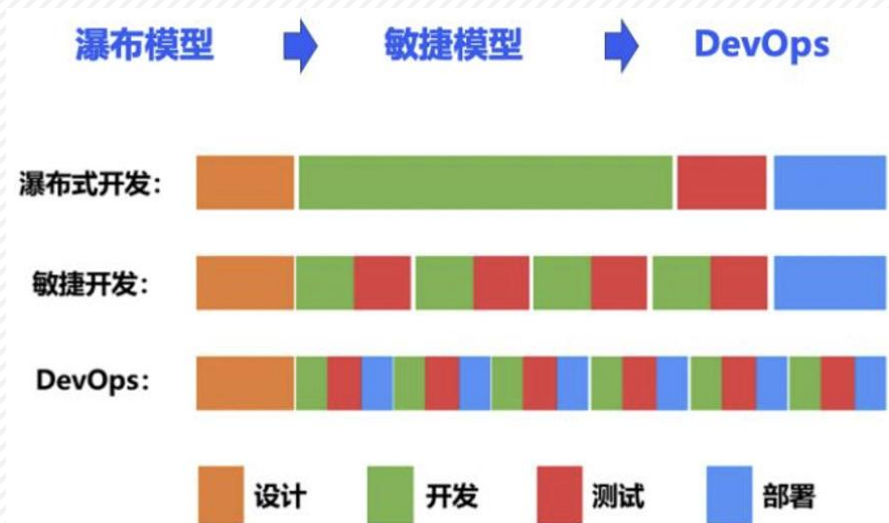
确保应用和基础设施的变更品质，更快速可靠交付

可靠性

安全

在快速运转的同时保持控制力和合规性

客户收益



• 相比瀑布式开发以及敏捷式开发更快的全流程迭代效率



改造前

- ❑ 业务创新: 现有几百个服务, 由于业务不断更新, 现有人员无法支持新增业务的快速迭代并推向市场
- ❑ 交互速度: 交付团队疲于应对现有的交付, 还要不断对接新业务交付, 交付能力下降
- ❑ 可靠性与安全: 多个交付团队参与实施, 交付标准不统一, 交付质量与安全得不到保证
- ❑ 团队合作: 需求快且急, 团队合作沟通成本上升, 相互推诿
- ❑ 规模: 团队和基础实施增加, 管理越发困难



改造后

- ❑ 业务创新: 近千条条流水线, 平均发布耗时3分钟/次, 平均每月构建6000分钟, 加速业务迭代更新
- ❑ 发布效率方面: 发布效率提升70%;
- ❑ 可靠性与安全: 标准的发布流水线, 统一了发布标准, 并通过代码规约, 代码评审保证安全
- ❑ 团队合作: 通过串联各系统平台, 流程与标准贯穿整个devops生命周期, 解决数据信息孤岛, 加强团队沟通
- ❑ 规模: 机器成本、人员成本降低20%

容器化改造/Devops服务内容

DEVOPS运维	运维服务	日常故障处理	服务月报	架构优化	事件响应
	监控内容	基础实施监控	应用性能监控	日志监控	全链路监控
DEVOPS建设	部署实施	DEVOPS环境构建	DEVOPS平台对接	Pipeline编写	
	方案设计	DEVOPS产品选型	DEVOPS方案设计	DEVOPS流程设计	
	项目管理	项目实施计划	项目实施跟踪	项目人员协调	
容器改造	应用构建	应用打包优化	Dockfile编写	基础镜像构建	
	应用改造	应用配置优化	应用架构优化	应用启动方式优化	
	应用调研	应用调研	应用改造方案	应用改造计划	

运维管理
服务体系

SLA管理

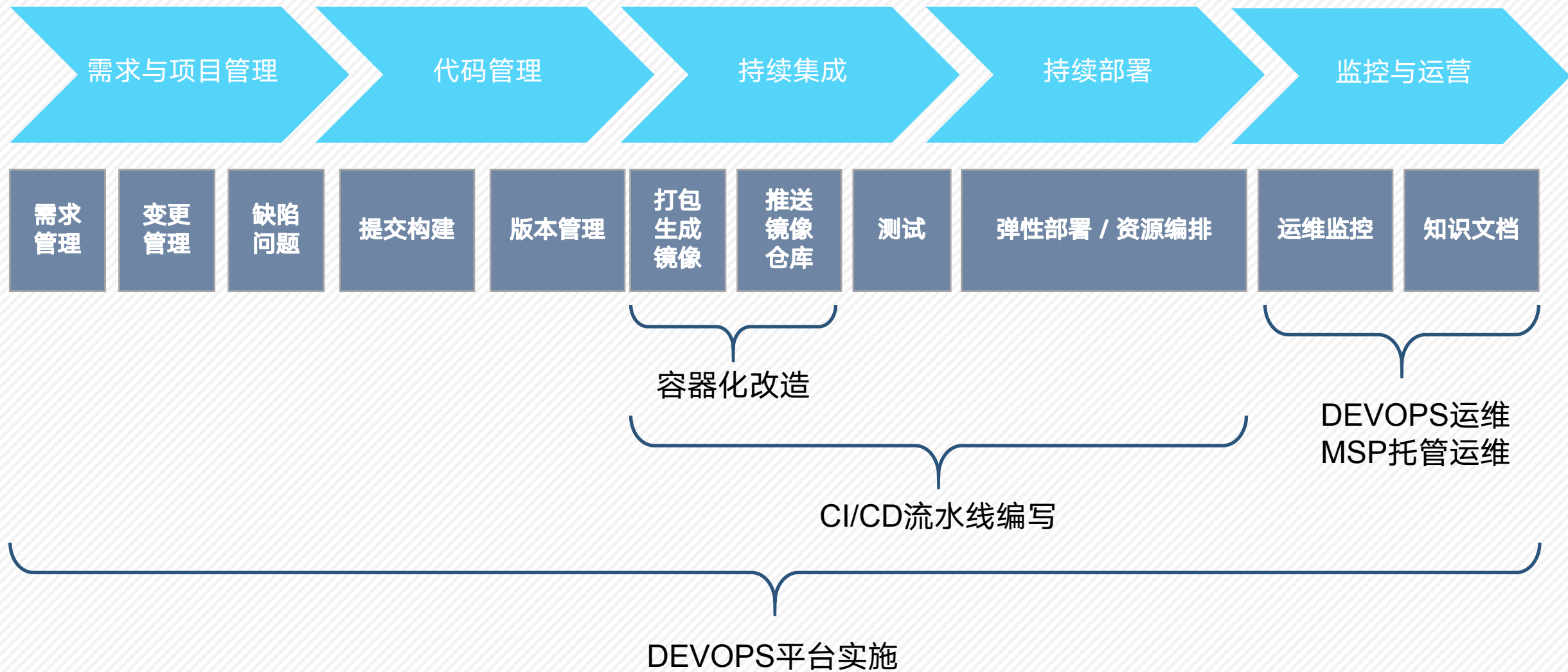
知识管理

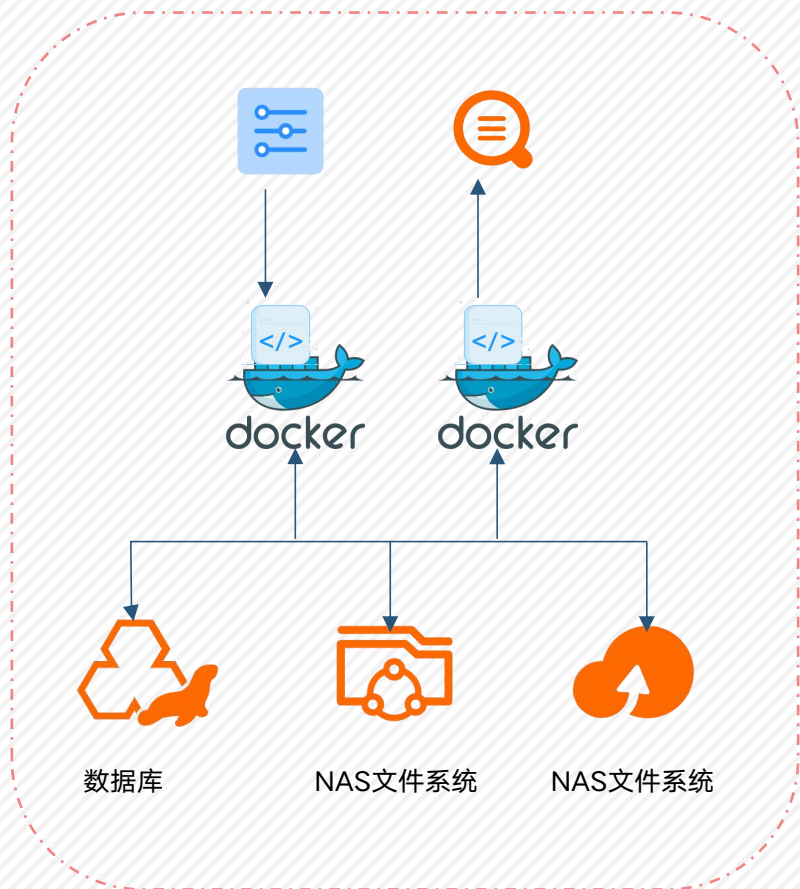
服务流程

服务团队

服务中台

DEVOPS服务输出单元





容器改造的目标

- 容器中只运行“无状态代码”

容器改造的内容

- 列出存储数据的文件位置

改造后需要把数据文件存储到NAS或者对象存储（OSS）中供多个容器访问

- 列出代码的配置文件

改造后需要把代码的配置文件变更为环境变量，代码可以从环境变量中读取配置文件

- 日志输出

改造后需要定义日志输出格式，方便采集器采集到日志服务中

容器化改造计划

容器化改造

CI/CD编写

平台实施

平台运维

改造阶段

任务分解

时间预估

应用调研

- 序运行语言与版本
- 程序的打包环境及打包版本
- 程序配置（环境变量，配置中心，配置文件）
- 程序启动命令，否有优雅停机的逻辑？
- 微服务各个程序的依赖关系
- 是否有注册中心？
- 是否有网关服务？

应用改造

- 应用配置使用环境变量
- 应用注册servicename
- 编写启动脚本加入优雅关机

基础镜像构建

- 使用alpine构建
- 提前注入默认的环境变量
- 构建与运行环境版本相同的镜像
- 注入提前写好的启动脚本

应用构建

- 使用改造好的应用构建
- 使用相同的构建环境构建应用

T+2

T+12

T+14

T+16

CI/CD流水线编写计划

容器化改造

CI/CD编写

平台实施

平台运维

编写阶段

持续集成流水线

- 持续集成流程确认
- 根据自动化工具确定流水线的开发语法
- 编写流水线代码
- 流水线测试

任务分解

持续发布流水线

- 持续发布流程确认
- 根据自动化工具确定流水线的开发语法
- 编写流水线代码
- 流水线测试

K8S YAML文件

- 编写K8SYAML文件
- YAML文件运行测试
- YMAL文件集成到流水线

Helm Chart编写

- 使用YAML文件编写Helm Chart
- Helm Chart允许测试

时间预估

T+3

T+6

T+8

T+10

自建DEVOPS平台

容器化改造

CI/CD编写

平台实施

平台运维

需求与项目管理

代码管理

持续集成

持续部署

监控与运营

需求管理

变更管理

缺陷问题

提交构建

版本管理

打包生成镜像

推送镜像仓库

测试

弹性部署 / 资源编排

运维监控

知识文档



GitLab

Gitlab Runner



Docker Registry

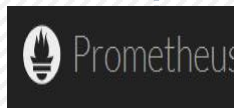


kubernetes



RANCHER

cucumber



DevOps的工具集列表(自建)

容器化改造

CI/CD编写

平台实施

平台运维

序号	功能	系统名称	系统介绍	授权	是否有原厂支持	可替代的产品
1	项目管理	Jira	JIRA是Atlassian公司出品的项目与事务跟踪工具，被广泛应用于缺陷跟踪、客户服务、需求收集、流程审批、任务跟踪、项目跟踪和敏捷管理等工作领域。	商业软件	有	Teambition
2	代码管理	GitLab	代码仓工具，提供代码存储、管理等工具	开源	无	GitHub、SVN
3	自动化工具	GitLab-Runner	一个CI插件，根据定义好的pipeline构建应用，必须与GitLab配合使用	开源	无	Jenkins
4	代码质量扫描	SonarQube	SonarQube是管理代码质量一个开放平台，可以快速的定位代码中潜在的或者明显的错误	开源	无	
5	镜像仓	Harbor	用于存储和分发Docker镜像的企业级Registry服务器	开源	无	
6	容器管理平台	Kubernetes	用于管理云平台中多个主机上的容器化的应用，Kubernetes的目标是让部署容器化的应用简单并且高效（powerful），Kubernetes提供了应用部署，规划，更新，维护的一种机制	开源	无	
7	Kubernetes管理平台	Rancher	用于管理K8S平台，提供WEB操作界面，应用商店和企业级权限管	开源/ 商业软件	无/有	ACK
8	监控平台	Prometheus	开源系统监控和警报工具，与K8S有较高的兼容性	开源	无	
9		Grafana	监控数据展示	开源	无	
10		skywalking	APM工具，包括了分布式追踪、性能指标分析、应用和服务依赖分析	开源	无	
11	自动化测试工具	Cucumber	Cucumber 是一个能够理解用普通语言描述的测试用例的支持行为驱动开发（BDD）的自动化测试工具，旨在提高软件质量并降低维护成本的软件开发过程	开源	无	
12	知识库	Confluence	Confluence是一个专业的企业知识管理与协同软件，也可以用于构建企业wiki	商业软件	无	Wiki

阿里云云效-一站式DEVOPS平台

容器化改造

CI/CD编写

平台实施

平台运维

核心产品

限制条件：基础实施使用阿里公有云

项目协作

代码管理

制品仓库

流水线

测试管理

知识库

任务管理

需求管理

版本管理

迭代管理

里程碑管理

代码托管

代码评审

代码安全

代码规约扫描

敏感信息扫描

代码审计风控

Maven 仓库

NPM 仓库

制品管理

编译构建

制品打包

自动化部署

持续集成

代码静态扫描

发布回滚

测试用例

测试计划

缺陷管理

在线文档

知识协同

文档分享

阿里云私有化DEVOPS-AECP企业级容器平台

容器化改造

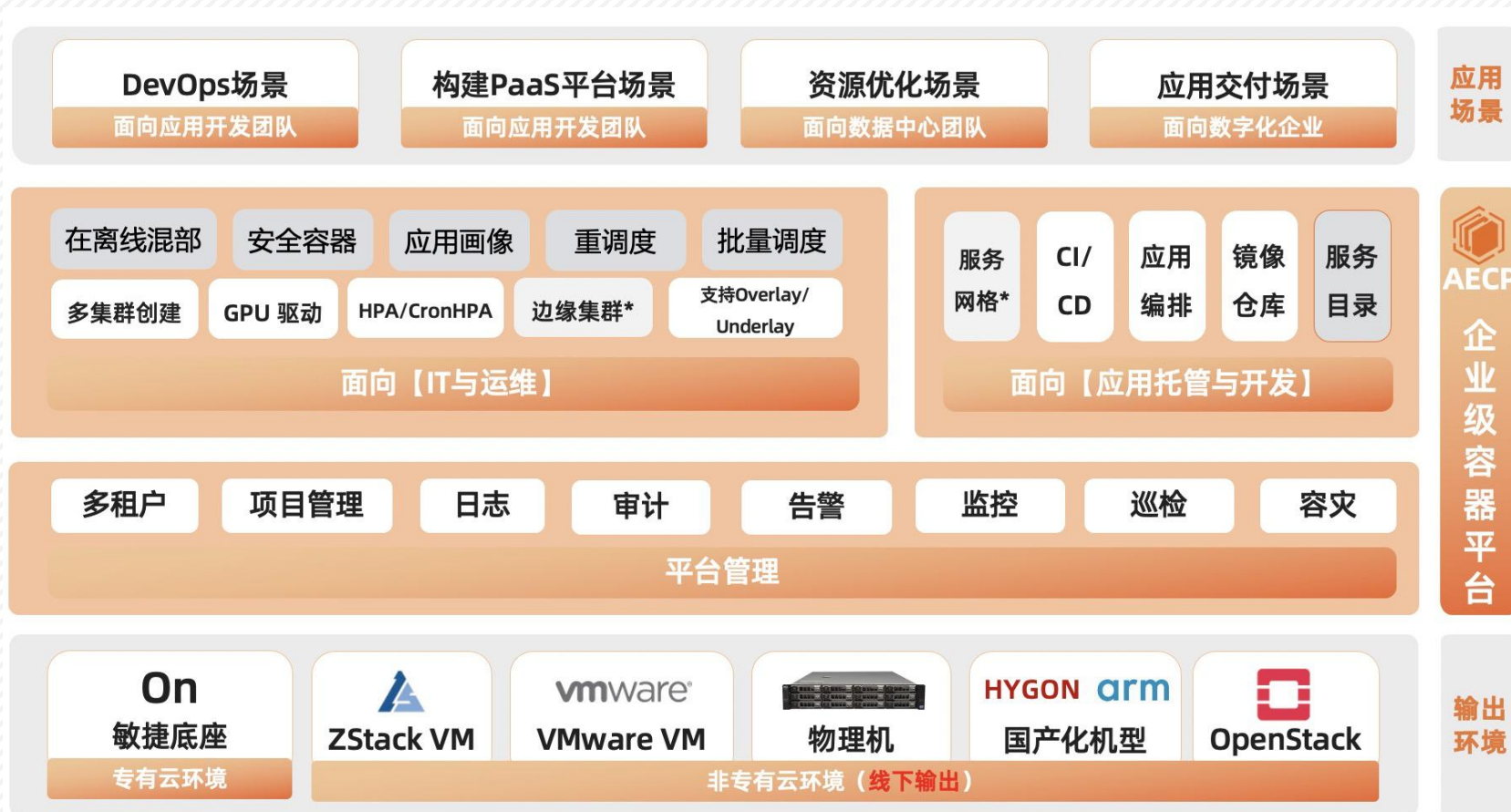
CI/CD编写

平台实施

平台运维



【产品架构】



【产品能力】

面向IT与运维的能力:

- 多租户、多集群能力，一键创建集群
- 故障隔离、集群备份能力，保障集群可靠运行
- 集成边缘管理能力，适应边缘云和IoT场景
- 重点支持面向资源利用率优化的能力栈

面向应用开发的能力:

- 支持Helm Chart应用编排
- 自带CICD工具，实现应用持续交付
- 丰富的应用可观测性

Kubernetes基础能力:

- 支持国产化机型，兼容主流IaaS虚拟化平台
- 自主的网络和存储插件，提供特色能力

额外付费项

建设中的能力

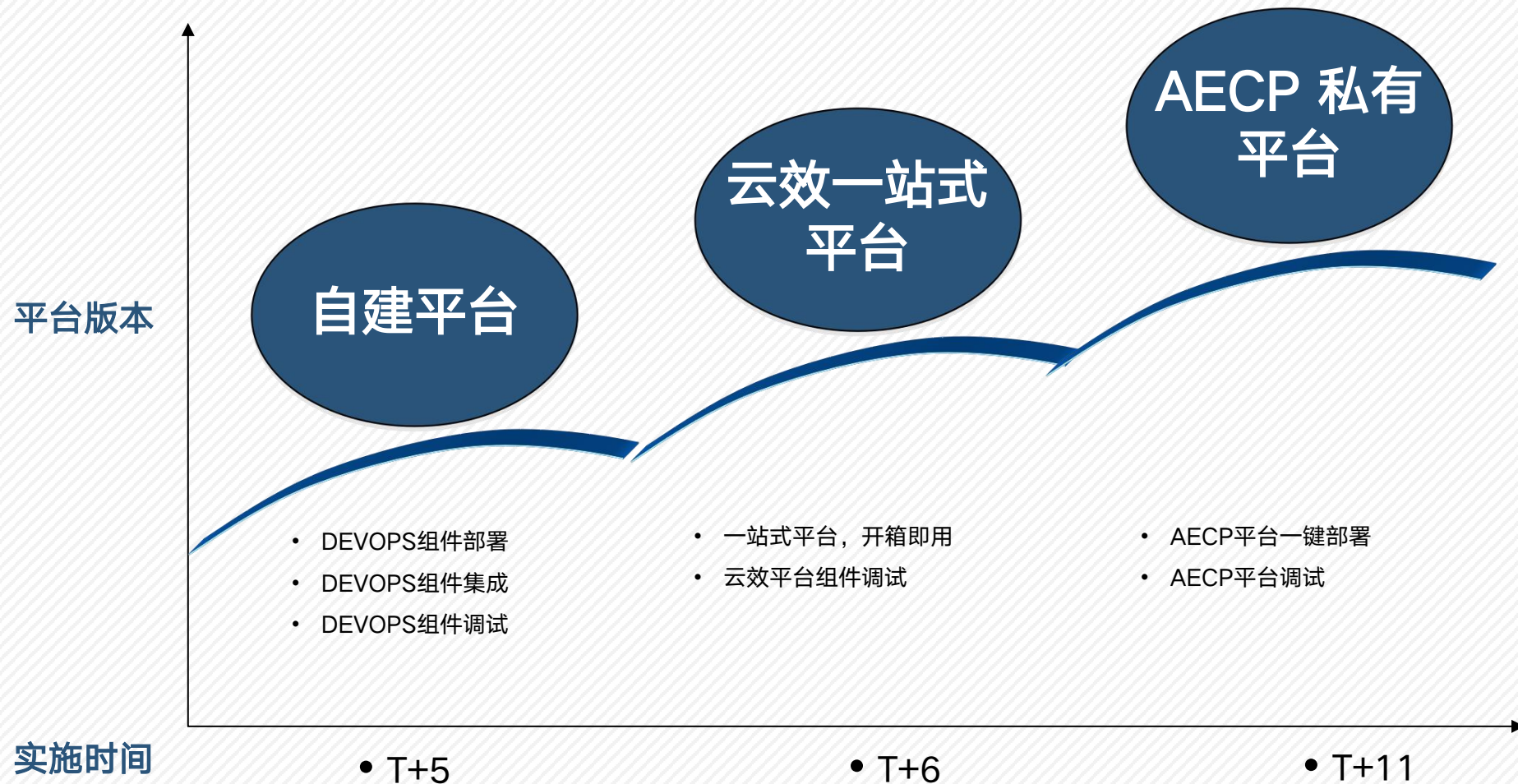
平台实施计划

容器化改造

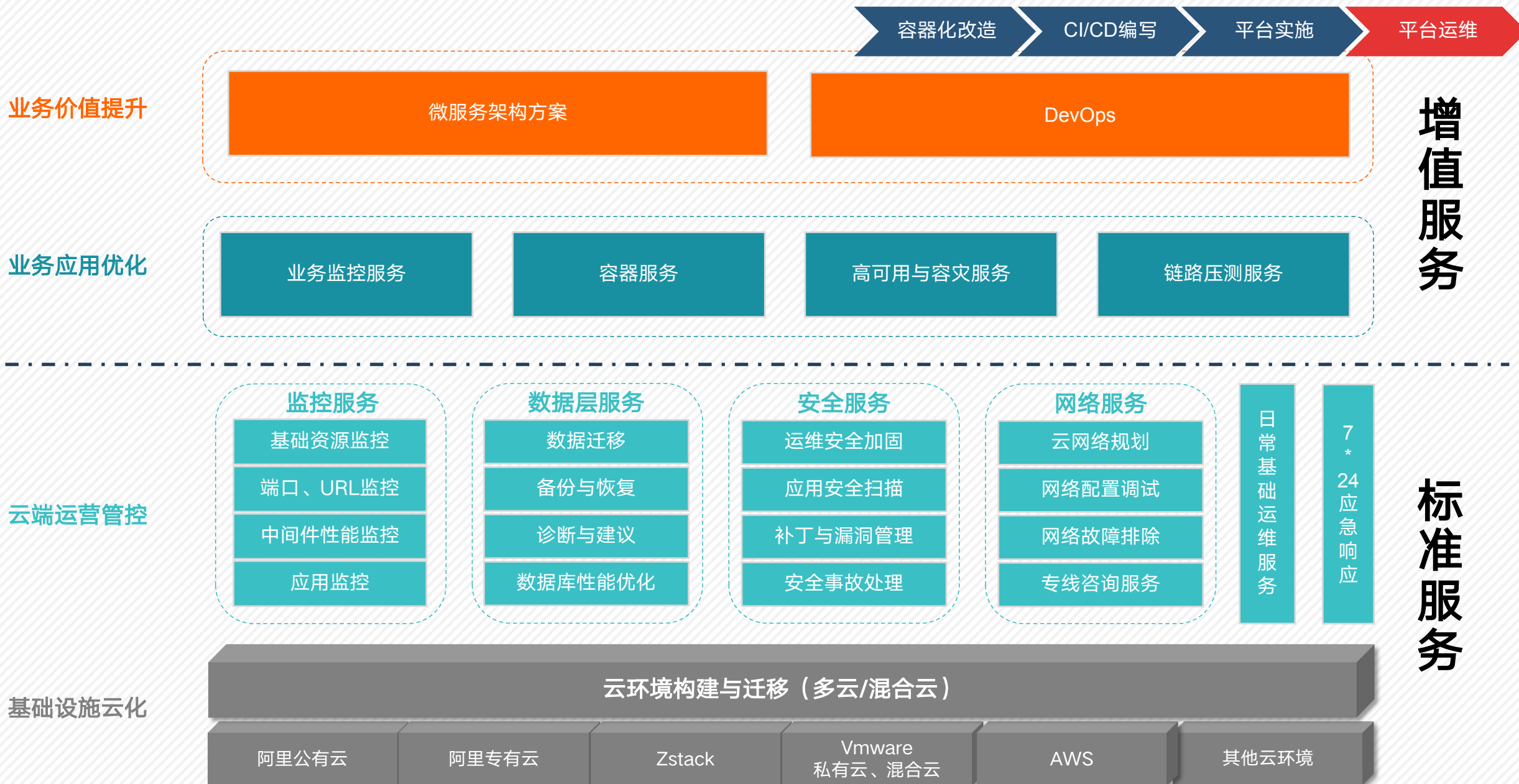
CI/CD编写

平台实施

平台运维



MSP服务组合



人员体制

客户

提瑞科技

项目责任者

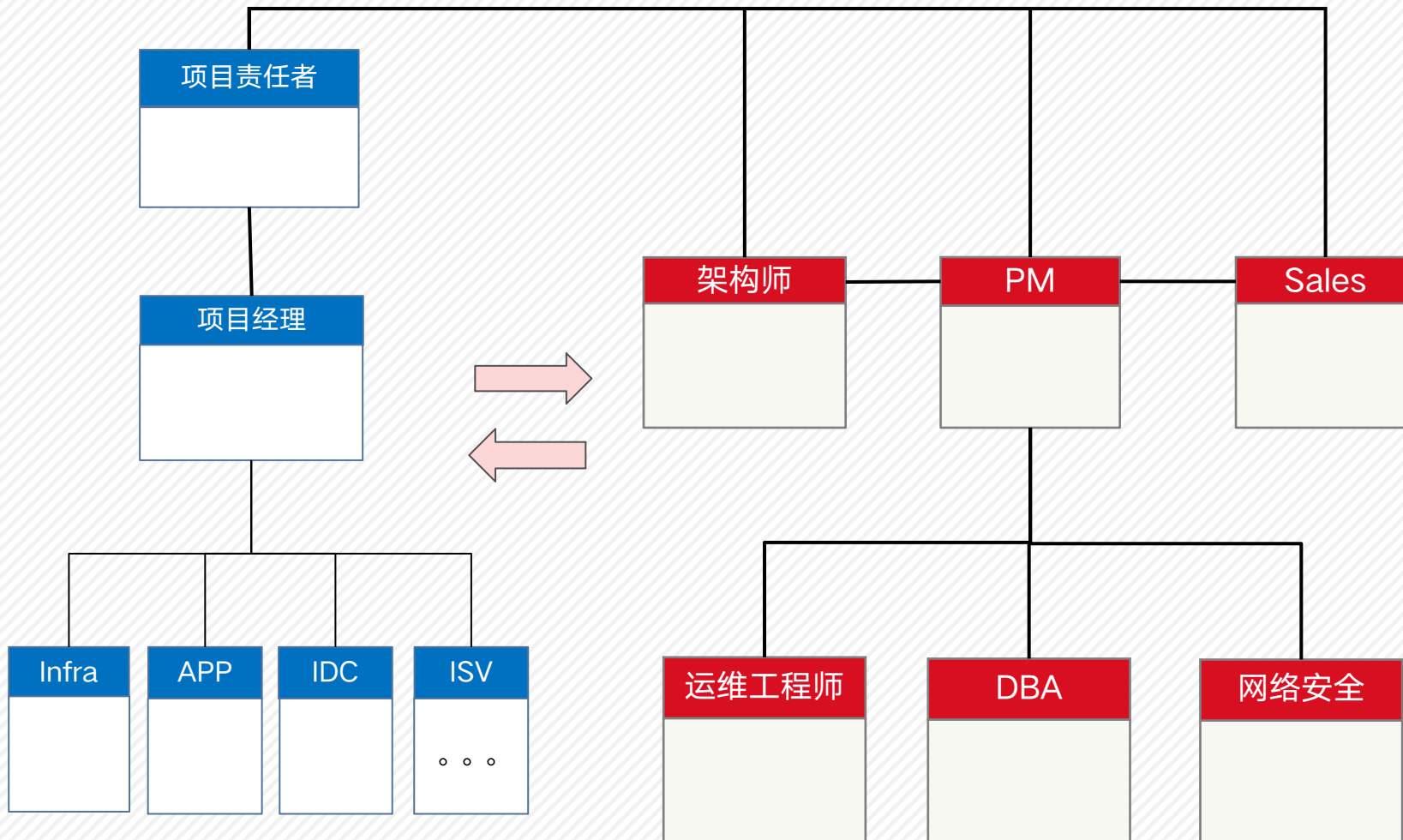
- 负责本次项目，具有决定权
- 成果物的最终确认

项目经理

- 管理整个项目
- 实际推进本次项目的实行
- 项目内外各种工作协调
- 处理对项目进展有影响的问题

项目成员

- 以本次项目为最优先进行工作。
- 接受项目经理或者项目责任者的指示，在业务确认的会议前，整理提供相关业务的情报。（需精通现在业务知识）
- 参加业务流程相关的会议，和自己部门相关的今后的业务流程进行审查和确认。
- 对项目任务进行引导和调整
- 对相关业务的重要成果物进行审查和承认



■ 上記体制图为参考方案。根据项目实施状况可能进行调整。

运维各阶段交付文档

运维阶段	相关文档
体系搭建	运维管理办法与制度文件 运维组织结构、岗位职责及人员配备建议 运维核心流程设计及相关文档模板 服务台操作手册
运维交接	交接测试报告 上线后支持阶段遗留问题清单 运维交接验收单，其中需要确认接收到的系统实施文档包括：流程蓝图、配置手册、功能开发说明书、权限设计、测试报告、系统架构、服务器清单、管理员相关用户名及密码、培训文档、用户手册、上线切换数据、上线后支持阶段问题清单、第一次月结报告等
运维前期	除了存放在运维系统中用户可以查看到的事件处理和用户交互信息外，提瑞会定期向客户提供各类运维绩效报表，比如：服务水平报告/能力水平报告等等
运维后期	在逐步将运维交接给客户内部团队独立运作的过程中，会整理提供： 运维遗留问题清单 运维期间系统变更、传输清单 新功能开发说明书、测试文档、操作手册清单 其他运维期间变更的系统实施文档清单 这些文档具体内容届时应该以一定形式存放在运维系统中

A dark, moody photograph of a city skyline, likely Chicago, reflected in water. The word "THANKS" is overlaid in white, bold, sans-serif capital letters in the center of the image. The sky is dark and cloudy, and the buildings are silhouetted against the light. The water in the foreground is calm, creating a clear reflection of the city.

THANKS