ThoughtWorks®

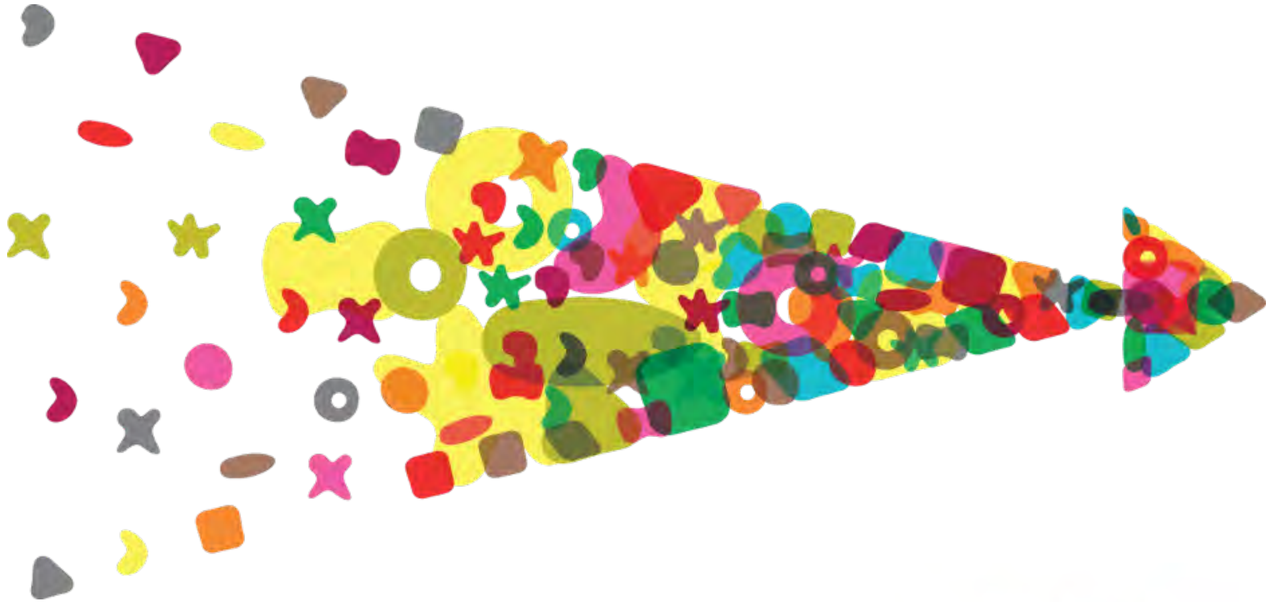
DOCKER
打造APP-CENTRIC交付

*ThoughtWorks*咨询师 钟健鑫

目录

- 现有交付模式下存在的棘手问题
- 用App-Centric交付模式优化持续交付
- 基于Docker实现App-Centric基础架构

现有交付模式下存在的棘手问题

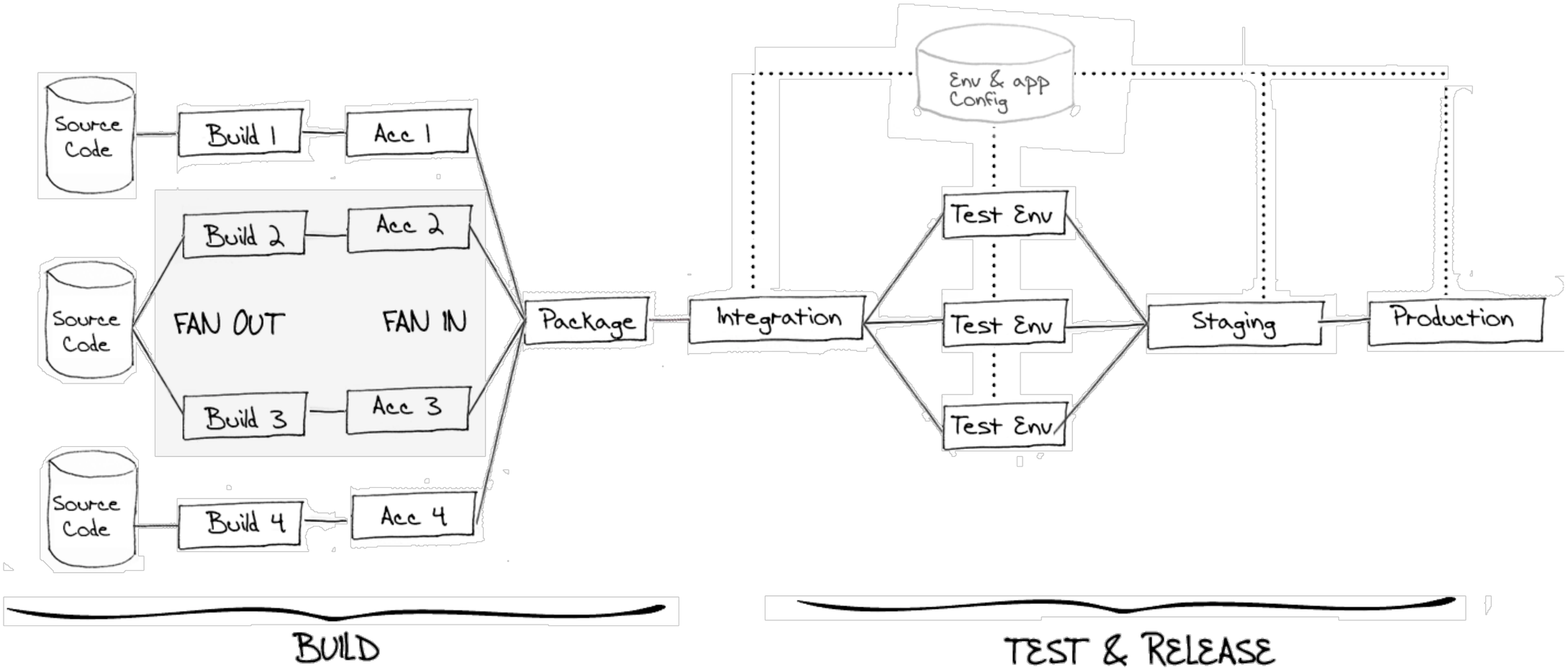


软件交付的复杂度逐步递增

投资回报率下降

现有交付模式下存在的棘手问题

持续交付一般视角



现有交付模式下存在的棘手问题

持续交付基础设施视角

项目启动

1.申请测试环境所需物理机或虚拟机
2.安装满足系统初始架构的运行环境和系统依赖

开发测试

1.架构调整，重新准备软件环境
2.运行环境和相关依赖升级或调整

发布上线

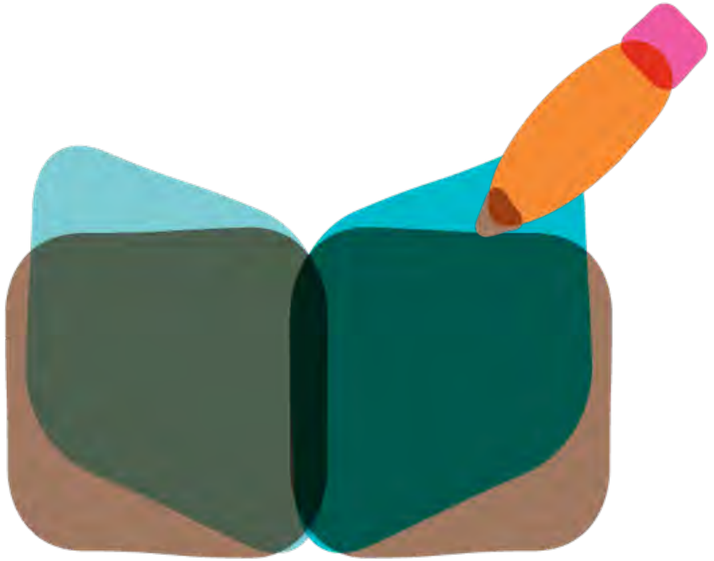
1.提前申请软硬件产品环境
2.尽量保证软件依赖和测试环境一致

维护升级

1.数据中心或遗留系统迁移
2.硬件更替或OS升级补丁

现有交付模式下存在的棘手问题

浪费笔记



准备和维护各种环境浪费大量时间

- —开发人员无法全心投入到软件产品的开发

流程复杂、环境管理工作繁琐

- —新增修改环境流程较长且很难保持环境的一致性，增加管理成本

遗留系统升级和迁移困难

- —遗留系统的架构升级和调整往往会花费大量的时间，且很难预期和提前验证结果

基础设施的维护和扩展不够安全和高效

- —当运行环境需要升级或是横向扩展节点，总是小心翼翼的实施并反复验证确认

目录

- 现有交付模式下存在的棘手问题
- **用App-Centric交付模式优化持续交付**
- 基于Docker实现App-Centric基础架构

用APP-CENTRIC交付模式优化持续交付

1.基础设施代码或配置变更在产品环境生效称为“部署”

2.具有业务影响的功能变化对最终用户可见称为“ 发布”

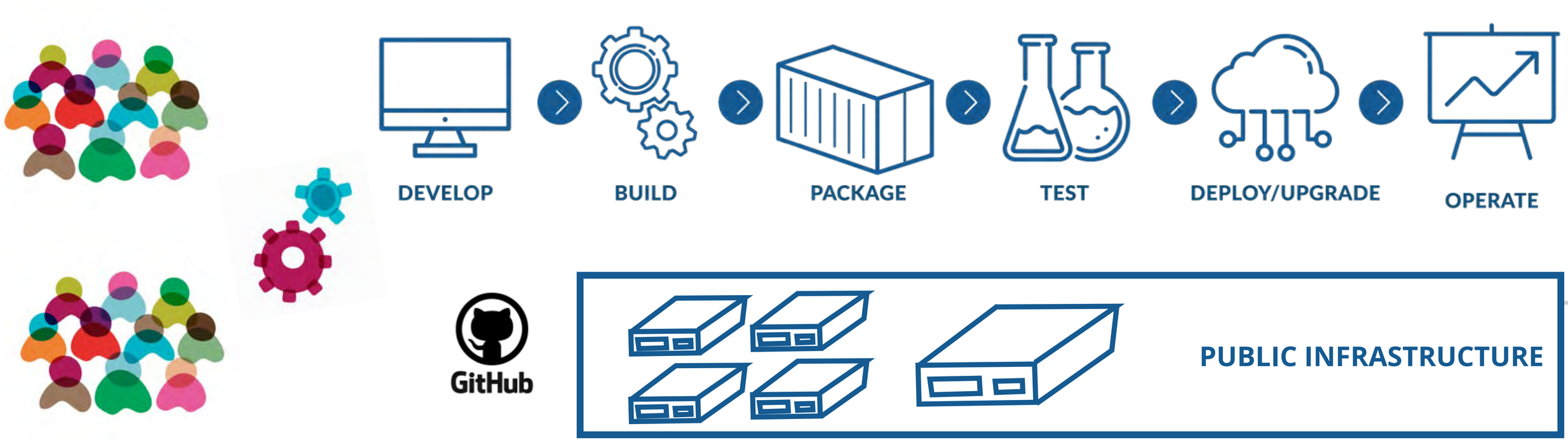
1.低耦合，可以被持续交付的基础设施

2.授权团队从无到有构建“服务”

3.使整个交付螺旋透明无死角

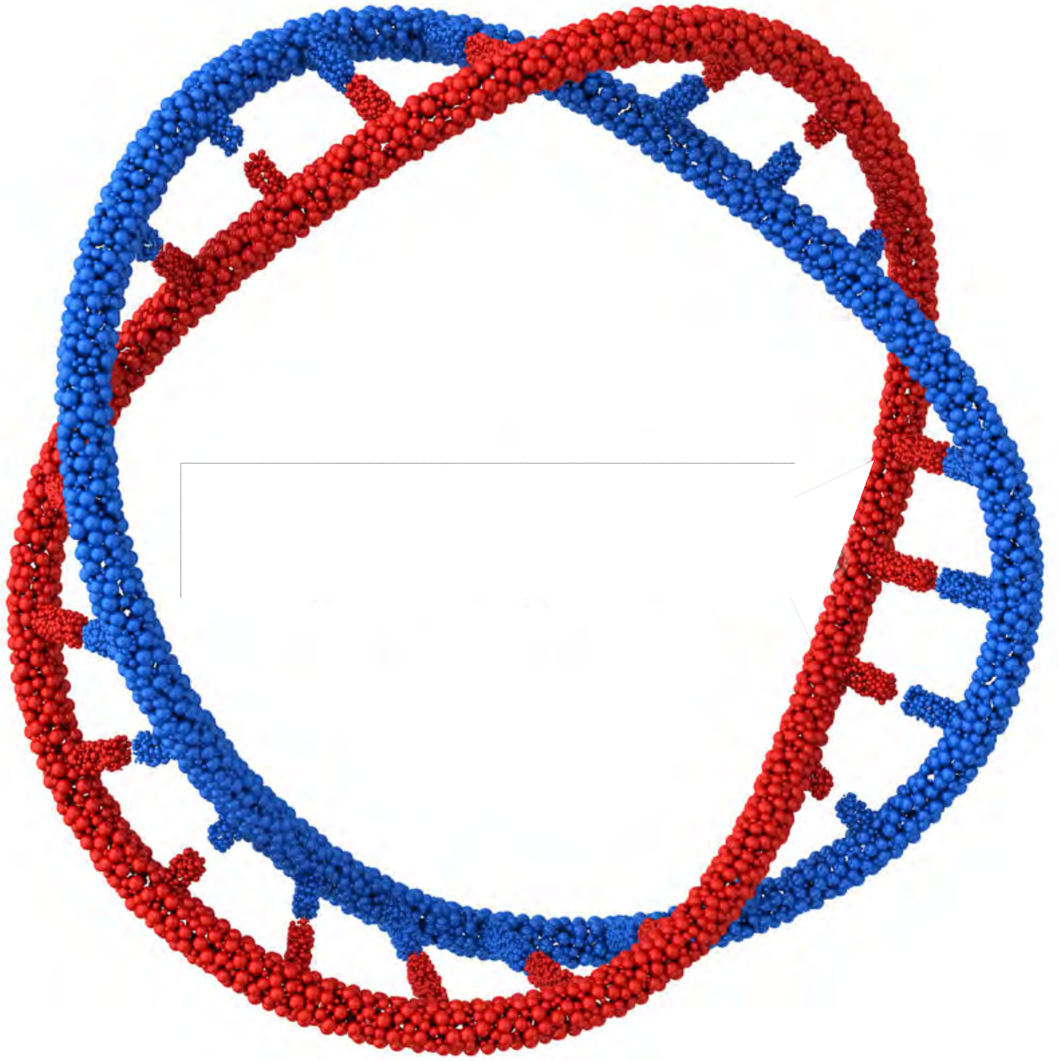
用APP-CENTRIC交付模式优化持续交付

APP-CENTRIC 交付过程



用APP-CENTRIC交付模式优化持续交付

APP-CENTRIC 交付过程



用APP-CENTRIC交付模式优化持续交付

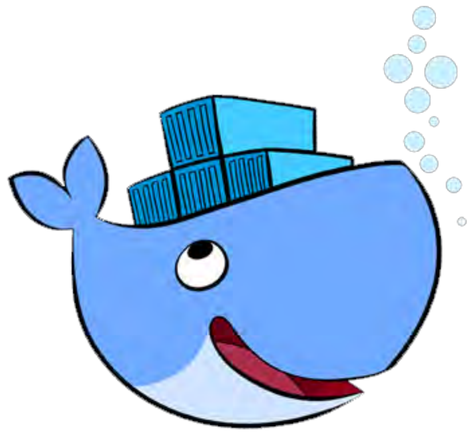
必备要素

App=软件+运行环境，底层可以是任何云平台，虚拟机或物理机

用APP-CENTRIC交付模式优化持续交付

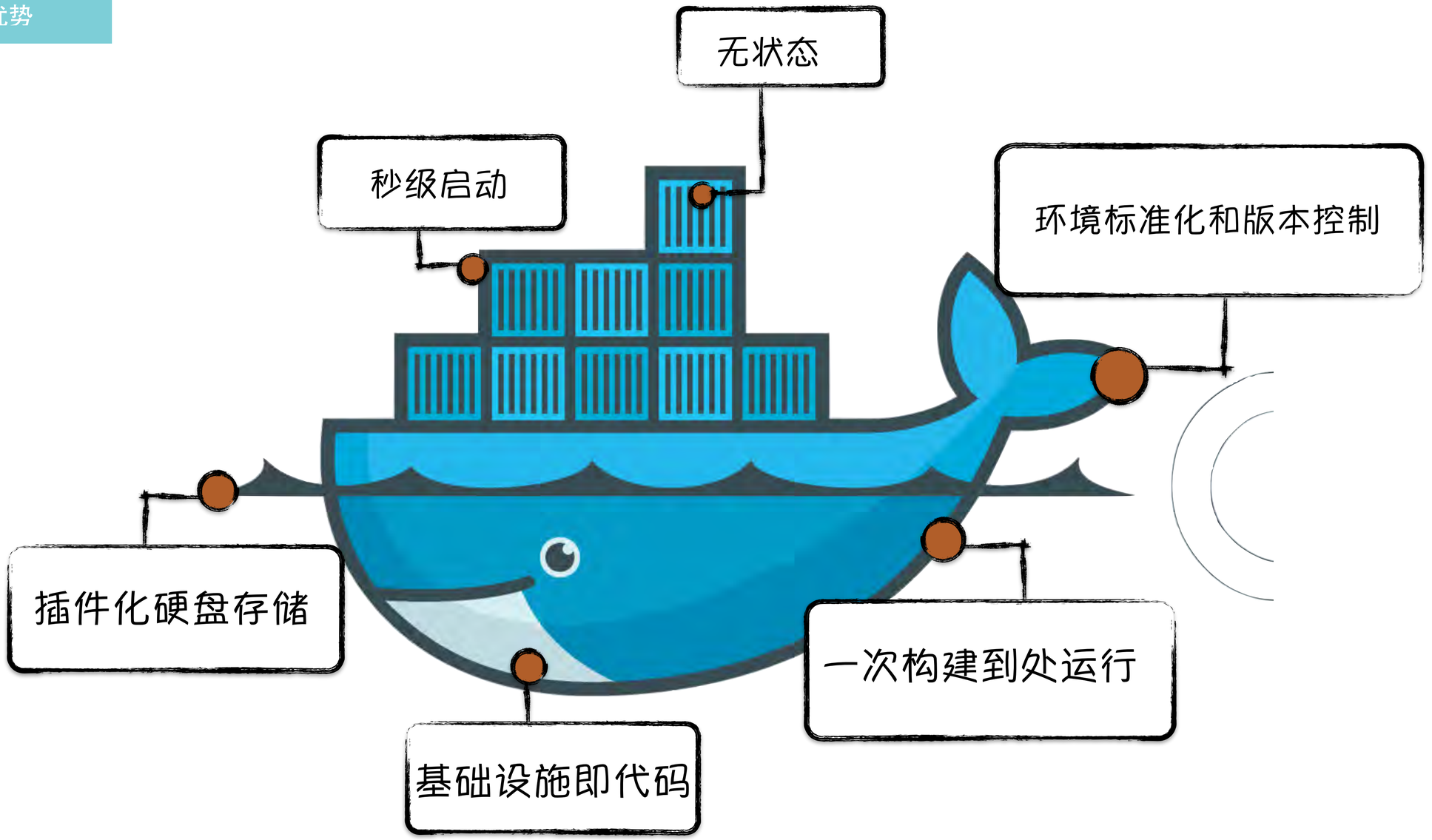
DOCKER

*Docker*以*Linux*底层内核技术为基础，实现了轻量级的应用隔离和部署解决方案



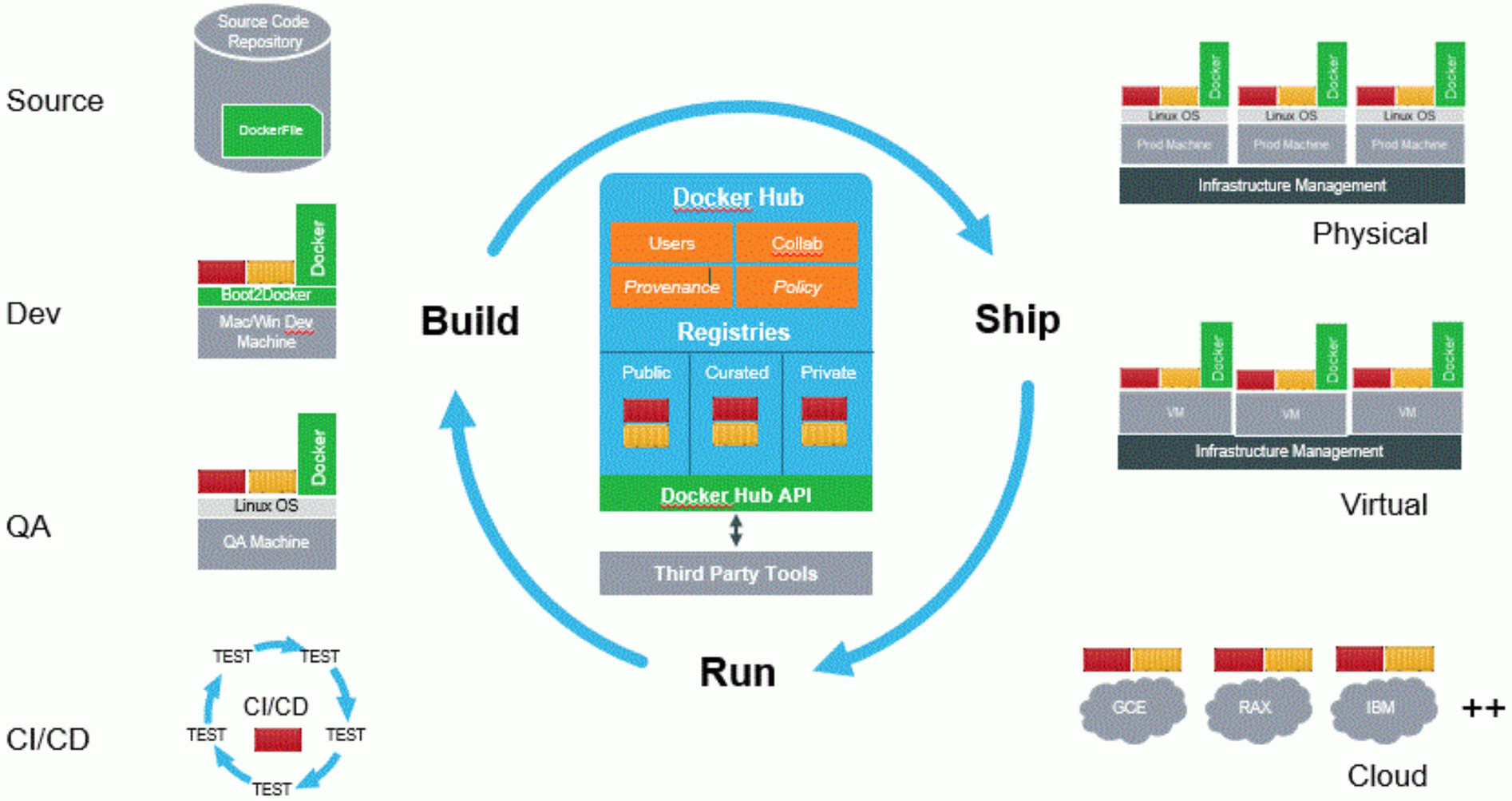
用APP-CENTRIC交付模式优化持续交付

DOCKER优势

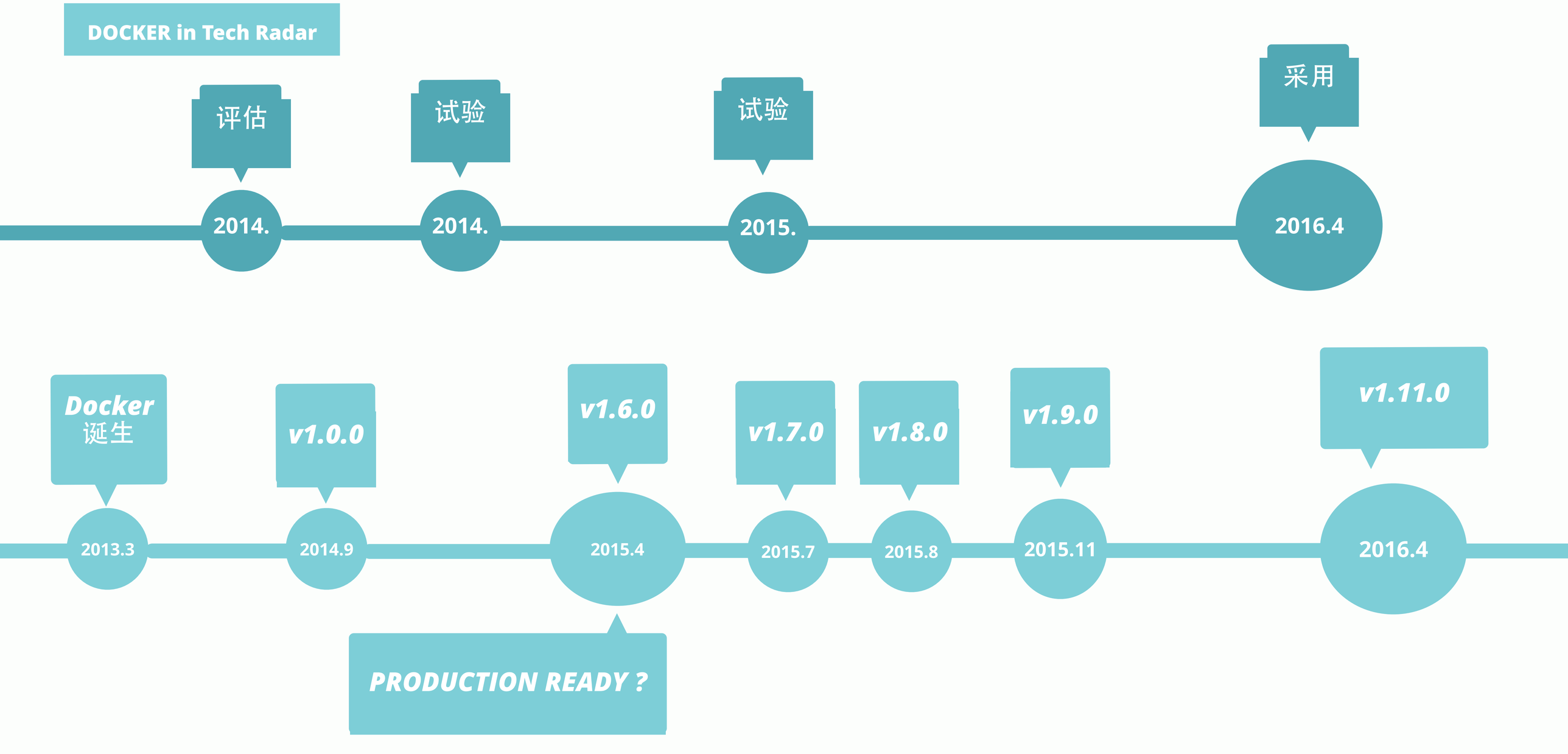


用APP-CENTRIC交付模式优化持续交付

DOCKER工作流



用APP-CENTRIC交付模式优化持续交付



目录

- 现有交付模式下存在的棘手问题
- 用App-Centric交付模式优化持续交付
- 基于Docker实现App-Centric基础架构

基于DOCKER实现APP-CENTRIC基础架构

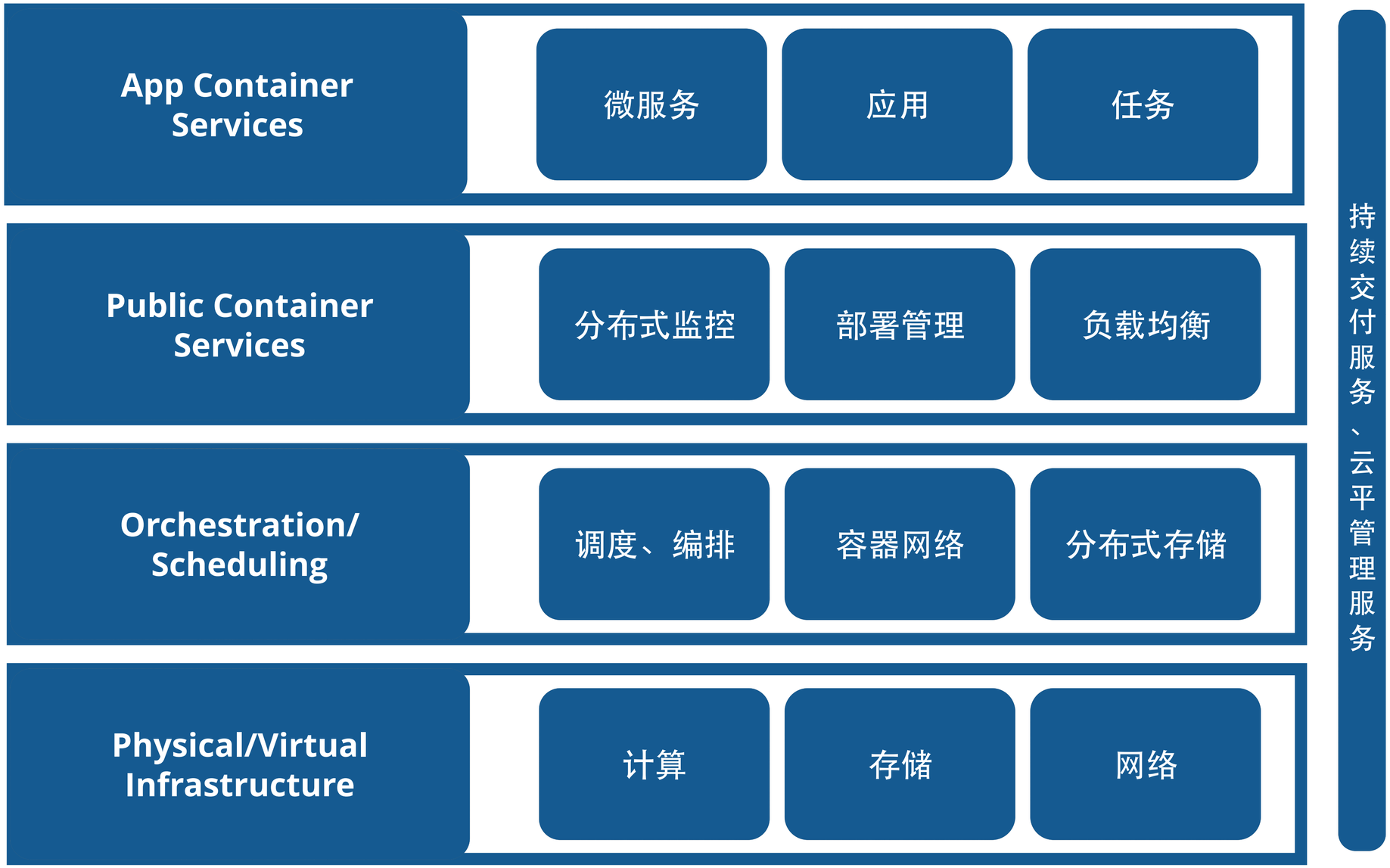
App-Centric简述

应用的部署、管理、发布、持续集成、健康检查 一切功能以应用为中心设计，隐藏了对底层资源的管理，架构本身不要求有任何具体业务依赖，

如果核心是管理计算、存储、网络的硬件资源，那就是*IaaS*

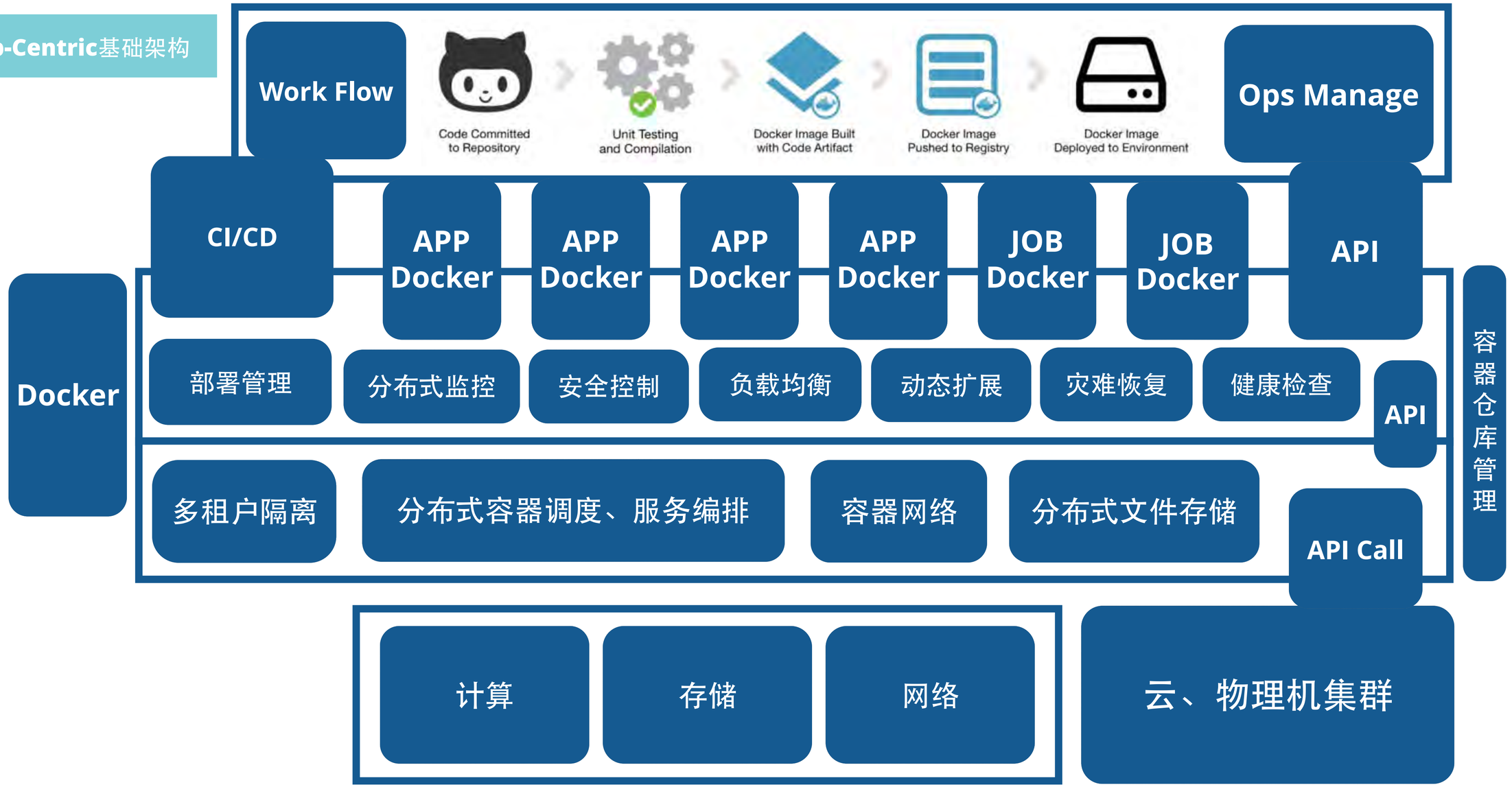
基于DOCKER实现APP-CENTRIC基础架构

App-Centric基础架构



基于DOCKER实现APP-CENTRIC基础架构

App-Centric基础架构



基于DOCKER实现APP-CENTRIC基础架构

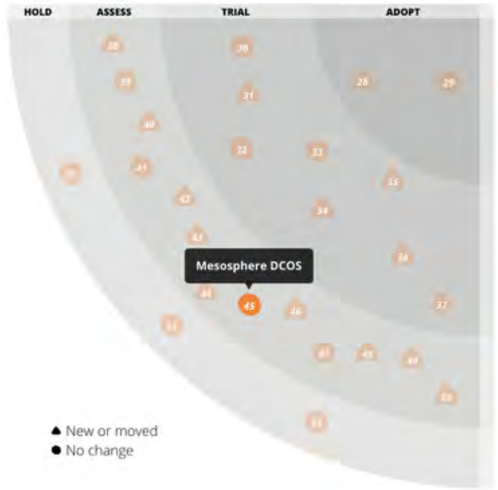
对比参考

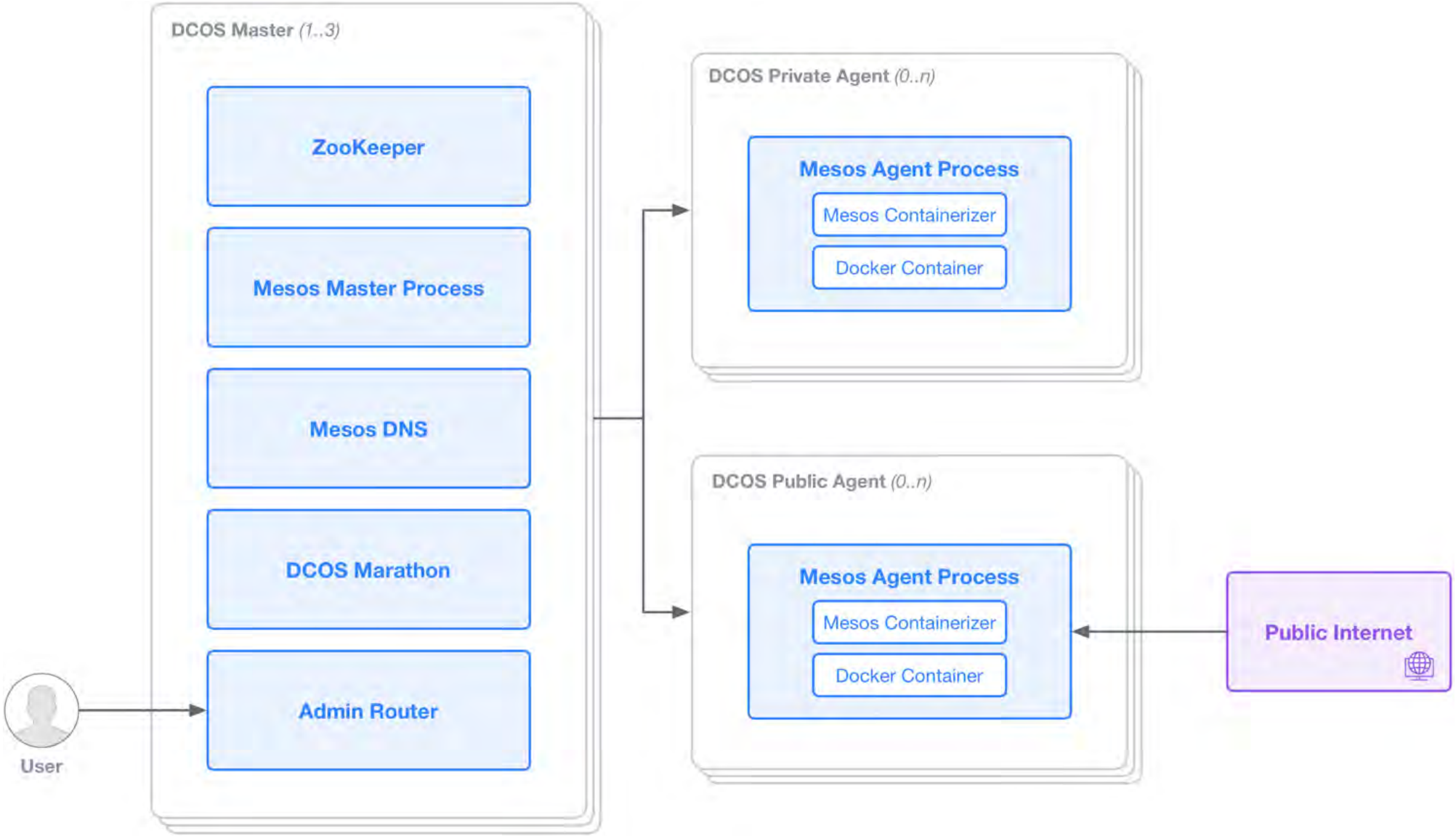
1.服务发现	✔	✔	✔
2.服务编排	✔	✔	✔
3.分布式存储	✔	✔	✔
4.滚动升级	✔	✔	✔
5.快速灾难恢复	✔	✔	✔
6.横向动态扩展	✔	✔	✔
7.集群资源统一调度	✔	✔	✔
8.健康检查	✔	✔	✔
9.负载均衡	✔	✔	✔
10.多级监控，中心日志收集	✔	✔	✔
11.持续交付服务	✘	✘	✘
12.安全	✔	✔	✔

基于DOCKER实现APP-CENTRIC基础架构

MESOSPHERE

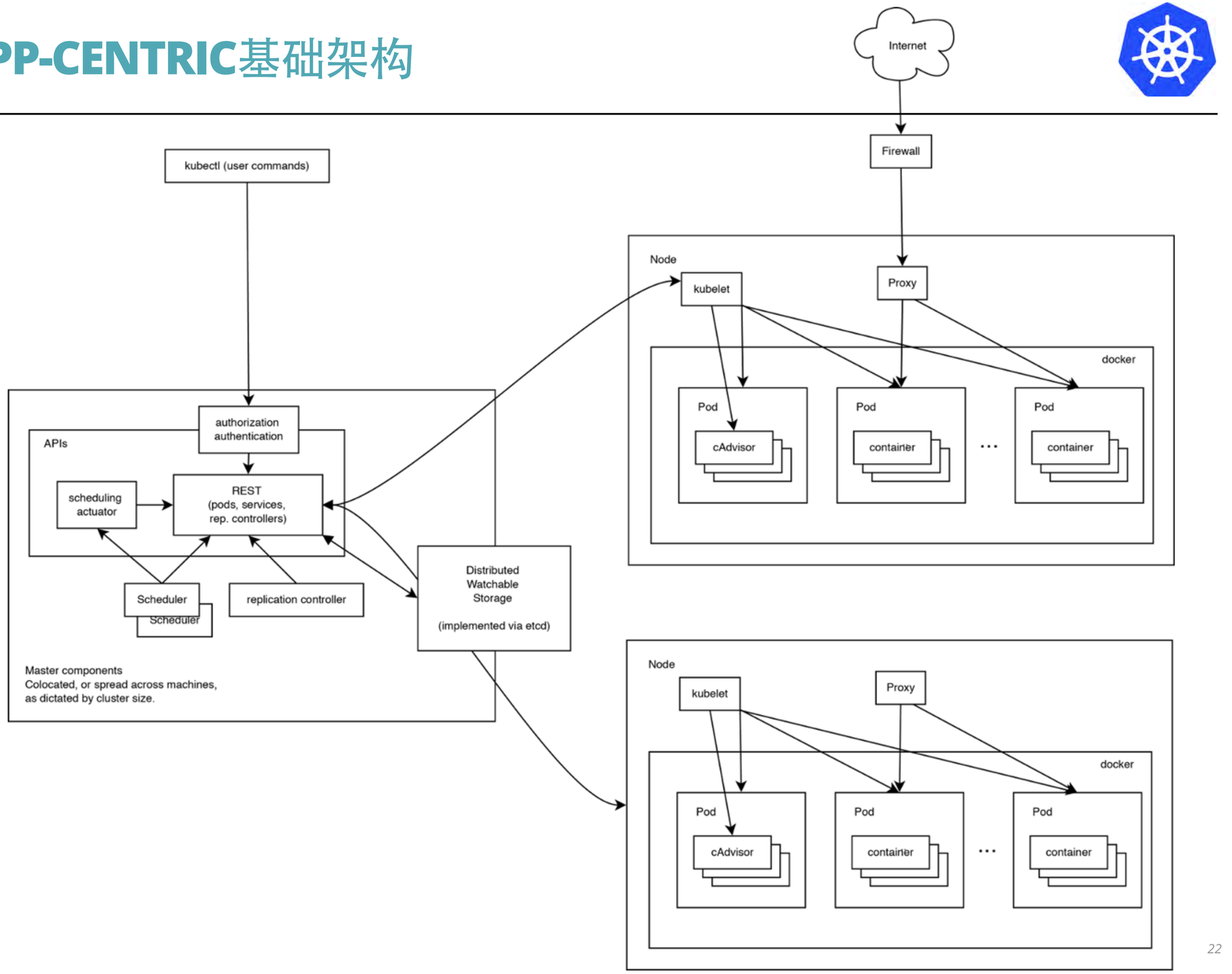
DCOS





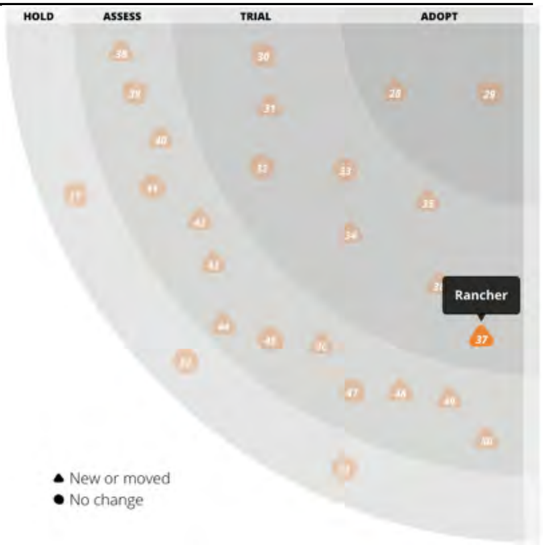
基于DOCKER实现APP-CENTRIC基础架构

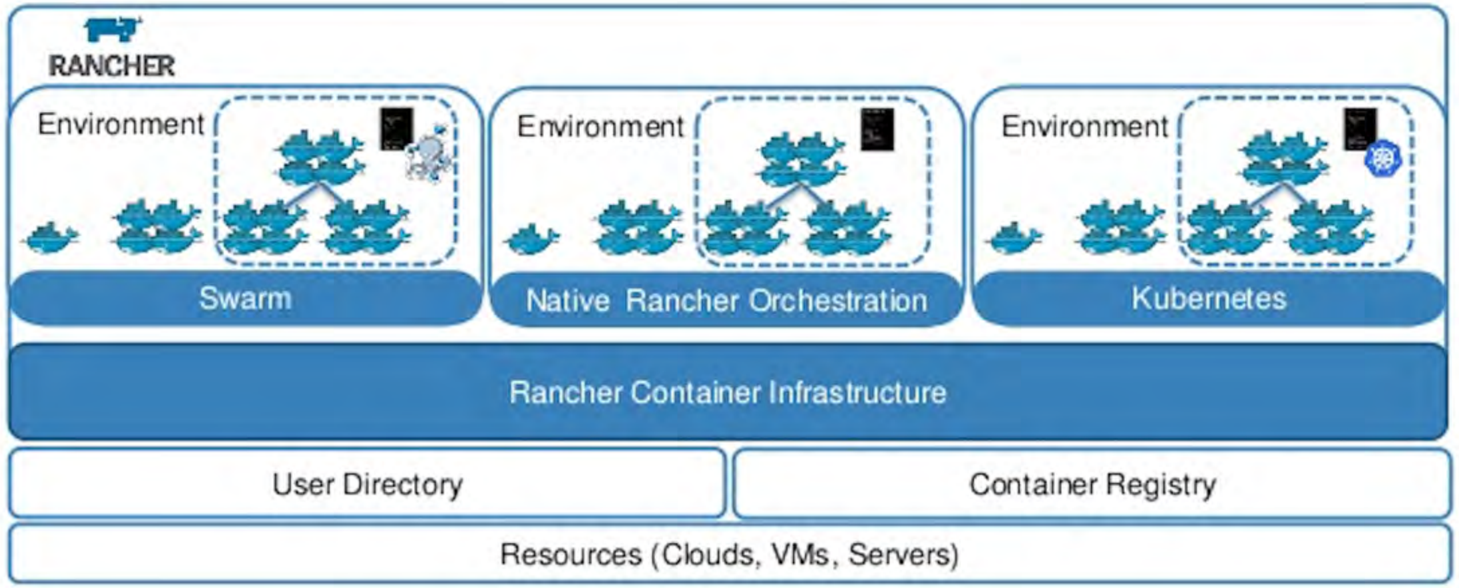
KUBERNETES

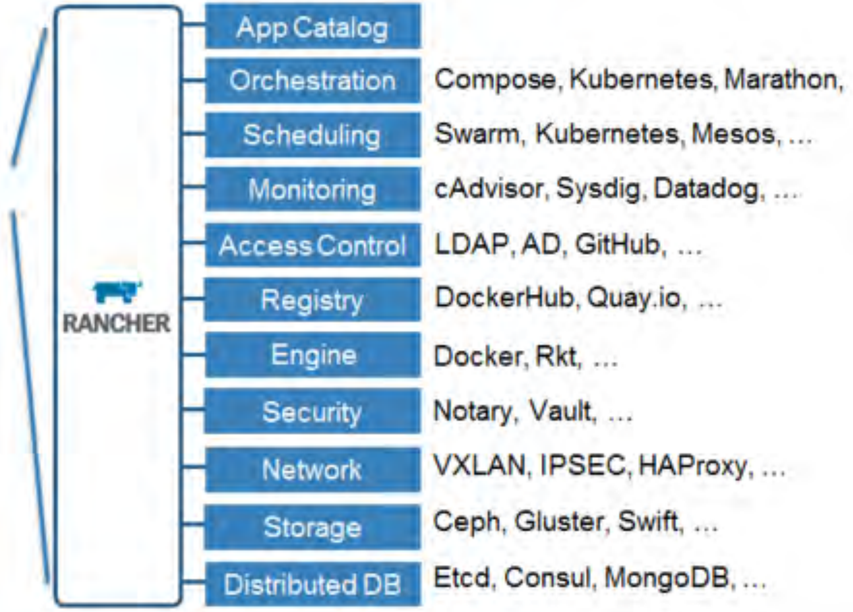


基于DOCKER实现APP-CENTRIC基础架构

RANCHER

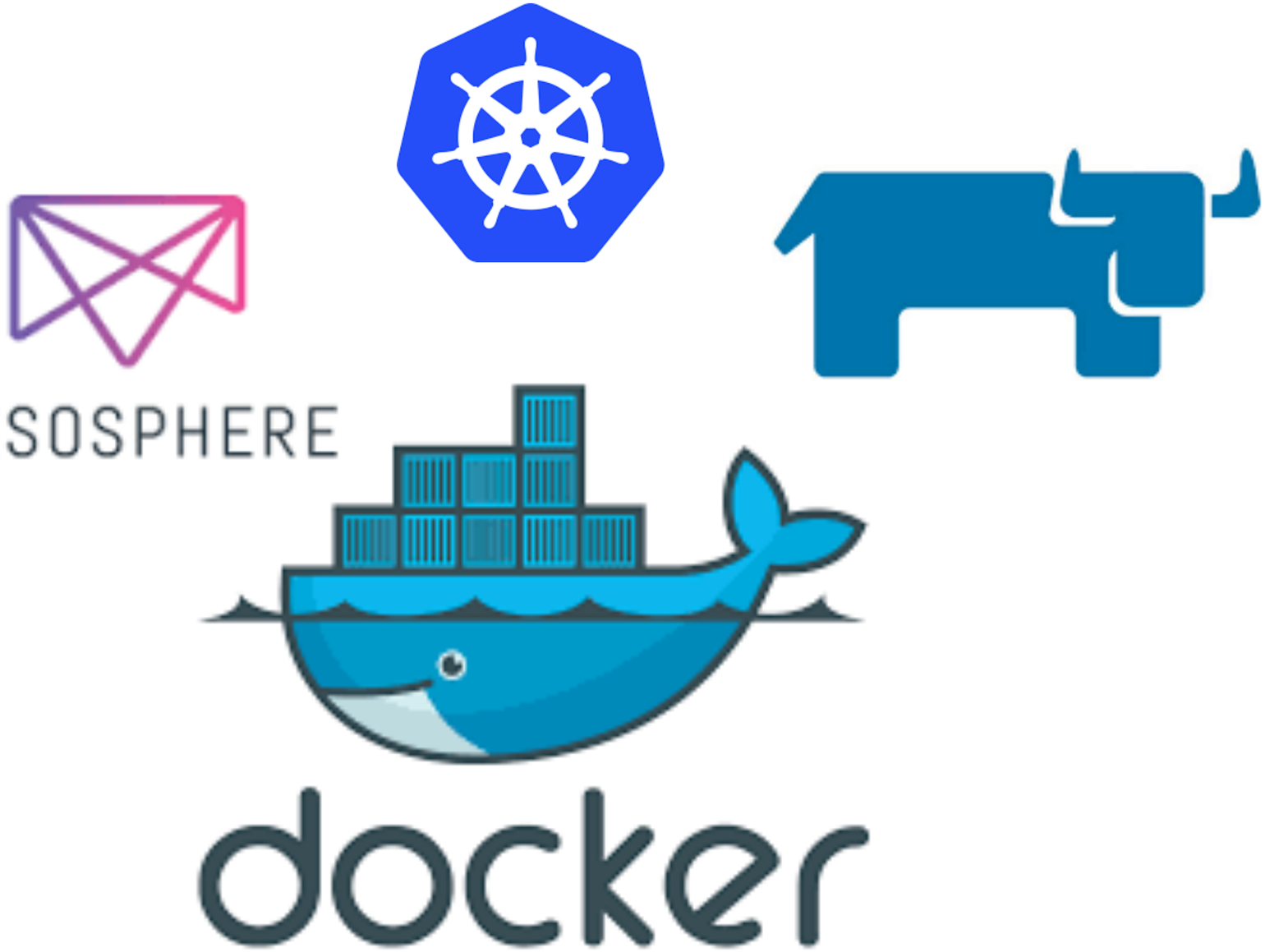






基于DOCKER实现APP-CENTRIC基础架构

Which one is the best?



现有交付模式下存在的棘手问题

浪费笔记

准备和维护各种环境浪费大量时间

- —开发人员无法全心投入到软件产品的开发

流程复杂、环境管理工作繁琐

- —新增修改环境流程较长且很难保持环境的一致性，增加管理成本

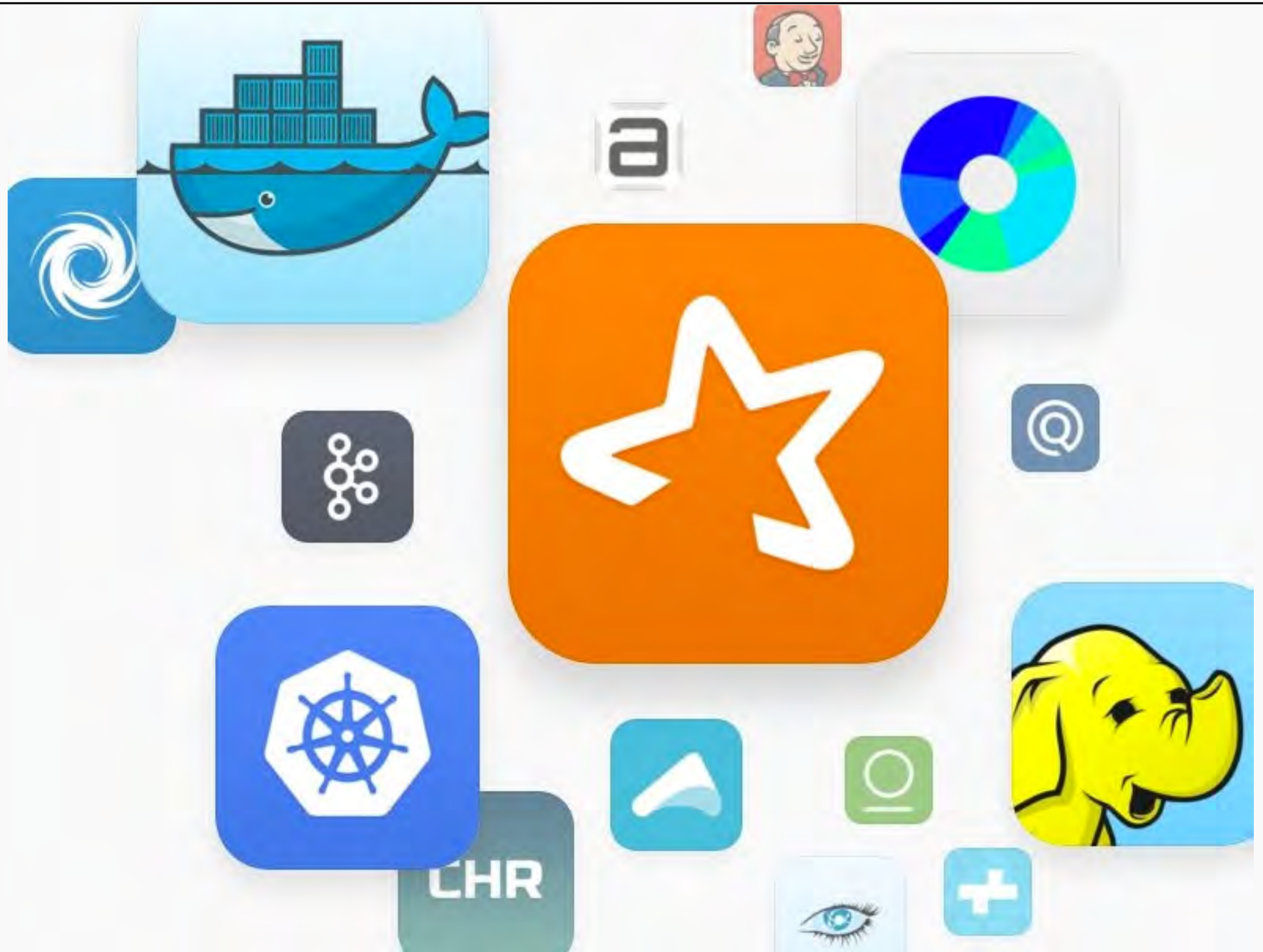
遗留系统升级和迁移困难

- —遗留系统的架构升级和调整往往会花费大量的时间，且很难预期和提前验证结果

基础设施的维护和扩展不够安全和高效

- —当运行环境需要升级或是横向扩展节点，总是小心翼翼的实施并反复验证确认

基于DOCKER实现APP-CENTRIC基础架构



THANK YOU

For questions or suggestions:

Go to TW Tech Radar
https://www.thoughtworks.com/radar

ThoughtWorks®