



# 容器打造持续交付 和高可用架构的原力

开发者定义的云

@segmentfault  
@chenr6

DaoCloud™ - Your Container, Delivered

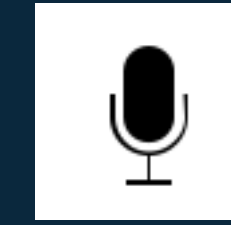
2015©

Google/EMC/VMware

提出第三平台概念，设计以应用及数据处理为目标的新技术与产品。



2009



2011



CloudFoundry

发布全球第一款商用PaaS平台技术，并公布其开源项目。

核心理念：基于IaaS构建PaaS，IaaS负责资源，PaaS负责应用。



Docker

出现在开源社区，并在短短十八个月里成为世界上最大的技术社区。



2013

核心理念：以应用为中心，提倡IaaS无关性，开放生态与标准。



2015



Docker PaaS


PaaS从一种技术，过渡为了一种规范，即互联网原生应用姿势。

核心理念：以应用和数据为中心，突出流程，调度，编排。并遵循开发标准。

# 创业的挑战



云加端的应用开发新趋势  
互联网+时代的生存法则



## 用户

开发者对用户体验的挑剔




## 业务

迭代速度是生命线



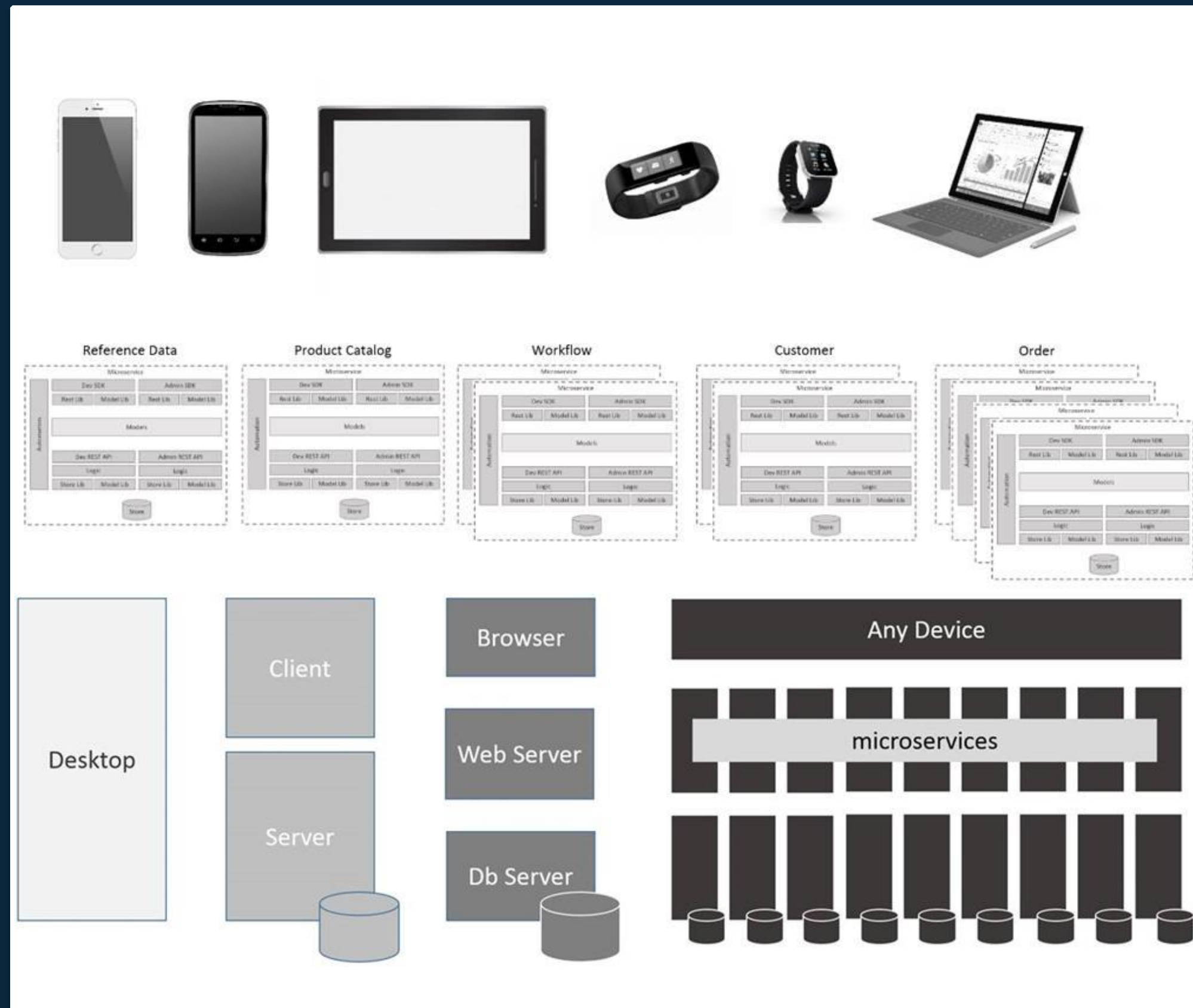
## 应用

多租户微服务

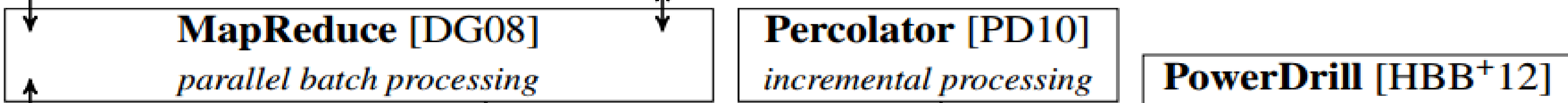
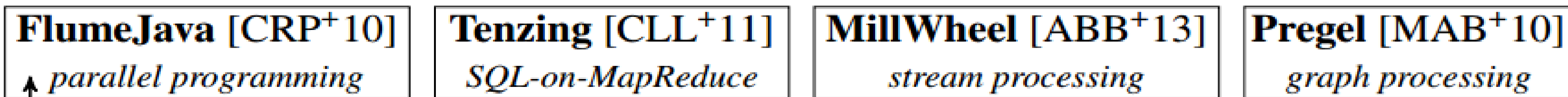


## 集群

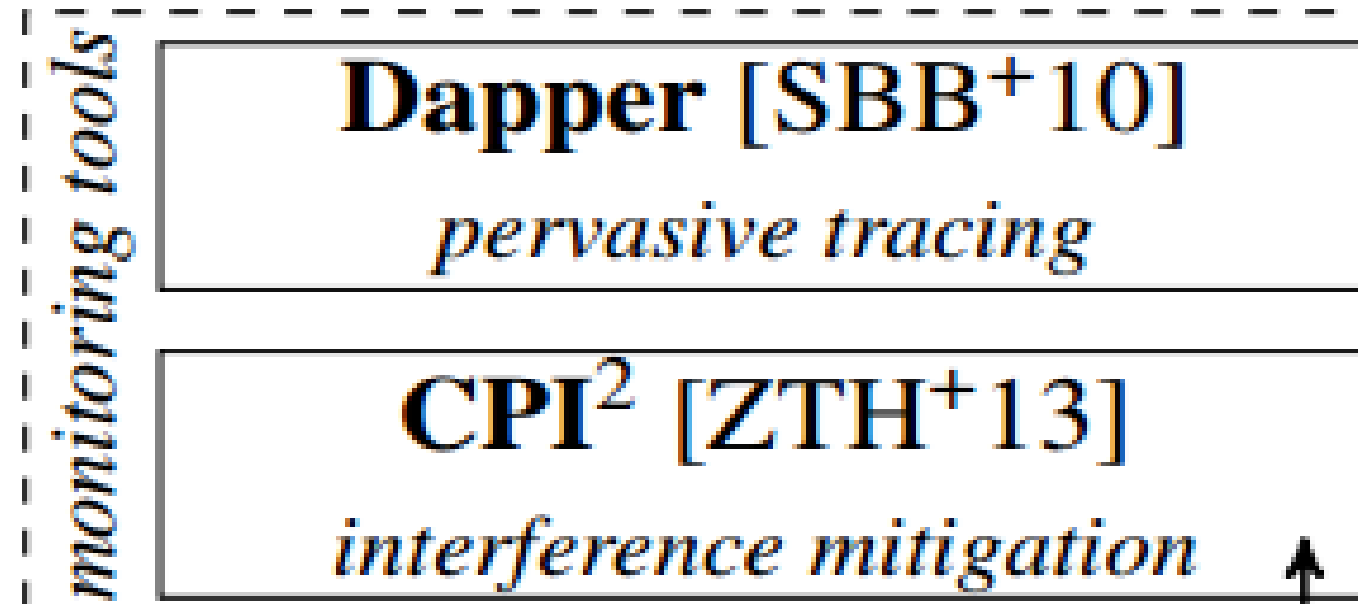
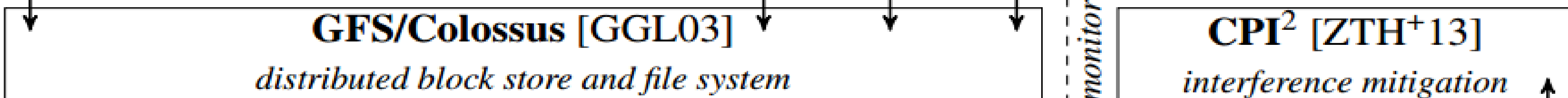
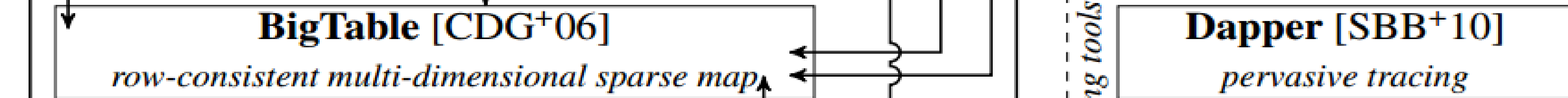
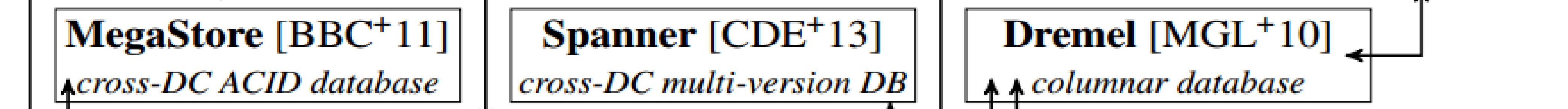
跨数据中心的混合部署



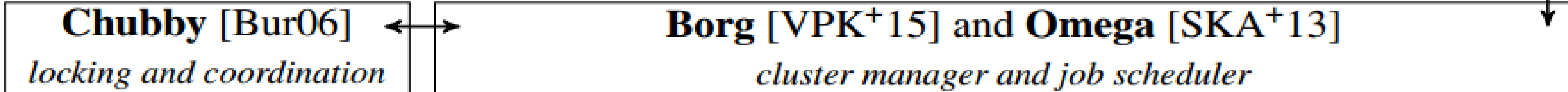
*data processing*



*data storage*



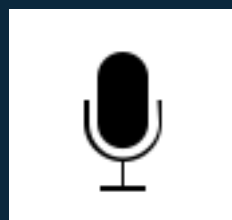
*coordination & cluster management*



# 理论指导



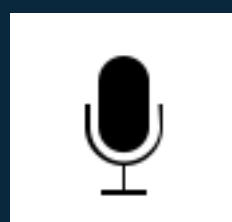
# 容器 vs. 虚拟机



容器技术不等同于轻量级虚拟化技术，适合传统IT转互联网架构的应用场景。

特性	虚拟机	容器
硬件接口	模拟	直接
操作系统	广泛	Linux为主
运行空间	用户级	内核级
隔离策略	Hypervisor	CGroups
系统资源	5-15%	0-5%
启动时间	分钟级	秒级
镜像存储	GB-TB	KB-MB
集群规模	近百	上万
高可用策略	备份，容灾，迁移	弹性，负载，动态

# 容器 vs. 容器

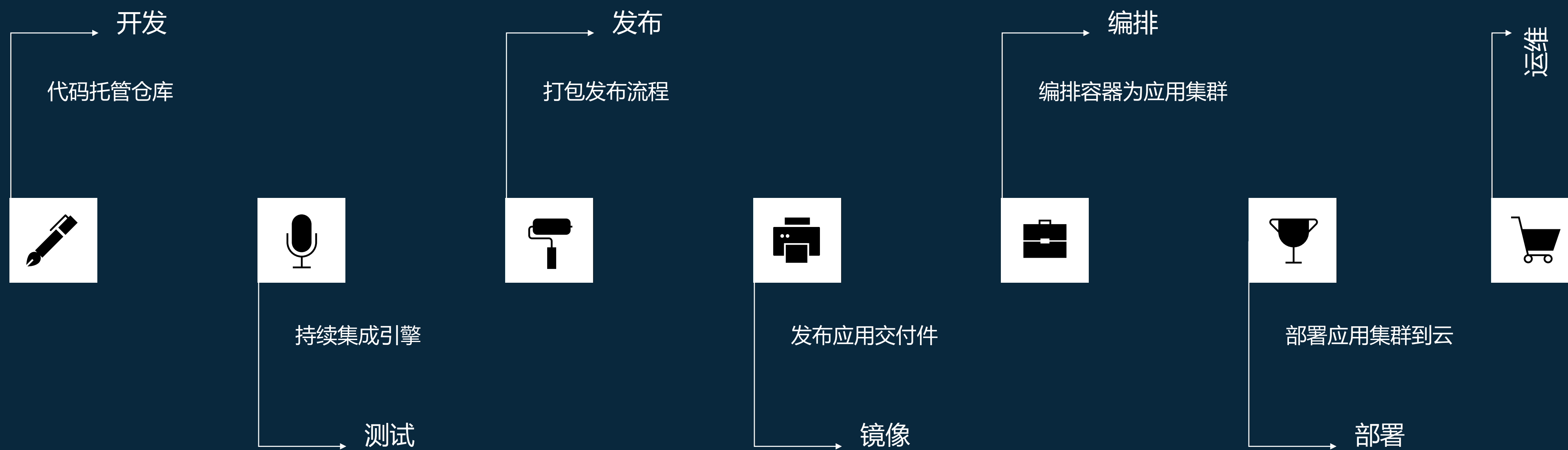


容器技术的发展，从系统向应用靠近，从功能向服务转换，从单点到集群为管理目标。

特性	LXC 2009	Warden 2011	Docker 2013
API	CLI	None	REST
管理界面	几个	一个	许多
跨平台	Linux	不限	不限
发布规范	无	Buildpack	Image
维护者	Linux开发者	Pivotal公司	Docker社区

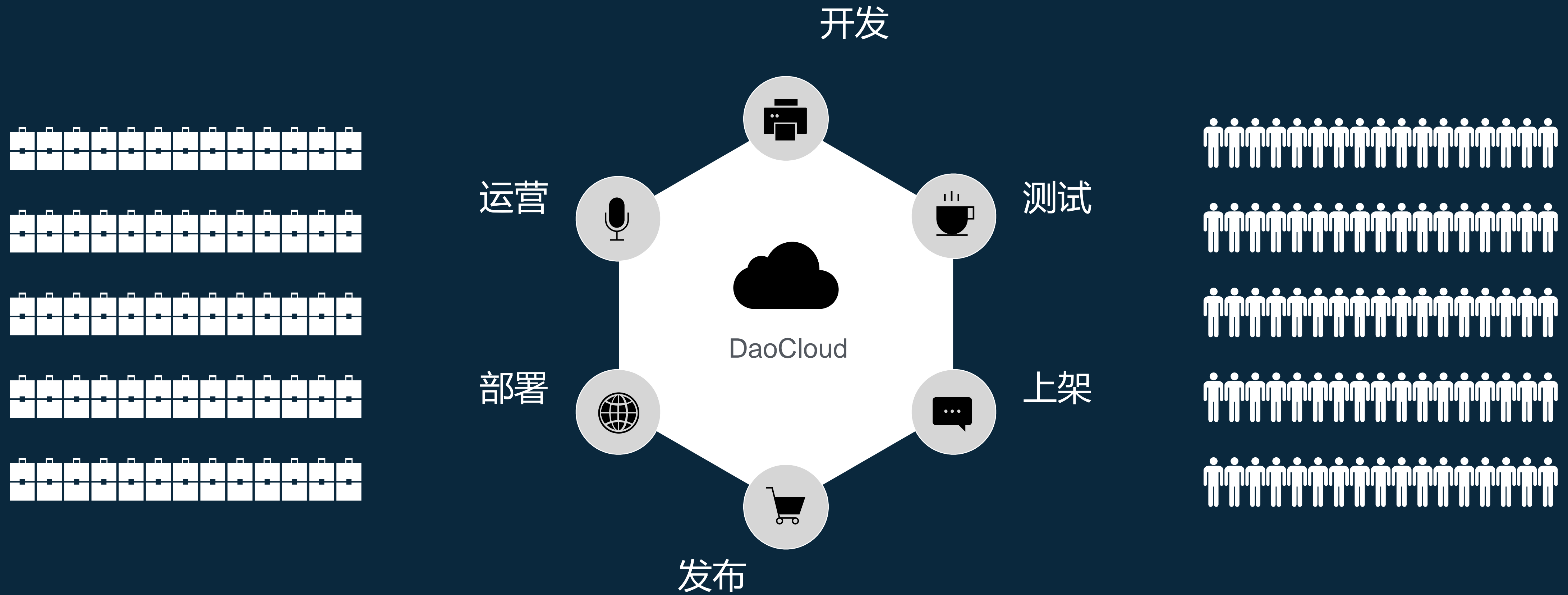


# 原力流水线 测试 构建 交付





# 建造目标容器的「Minecraft」世界



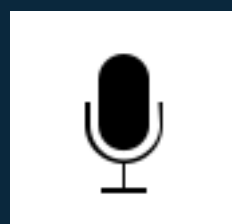
# 欢迎来到「Dao」的世界

| 开发者驱动的组织肌体，其能力不再是制作软件，而是交付服务，软件即服务是内生的肌体运转方式，并决定了我们的商业模式。

## Dao 法一：透明

| 显示并可声明的方式，来完成协作工作，并使得这一过程对所有的参与者可见，可追溯。

# 声明式的交付流程



从公有仓库获得镜像模板，注入可追溯的代码版本，封装成为应用镜像，并授权私有化存放。

## Dao 法二：契约

| 形成明确的契约，在系统架构上，分层是前提而不是目标。越早形成分层的共识，并协同维护好分层间的契约。

# 分层架构

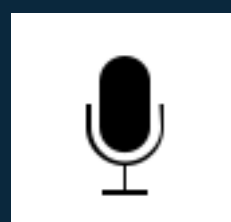


## Dao 法三：效率

| 让开发定义基础架构，任何层面访问基础架构的过程都是影响效率的事情，一旦发生就要优先想办法自动化并契约它。

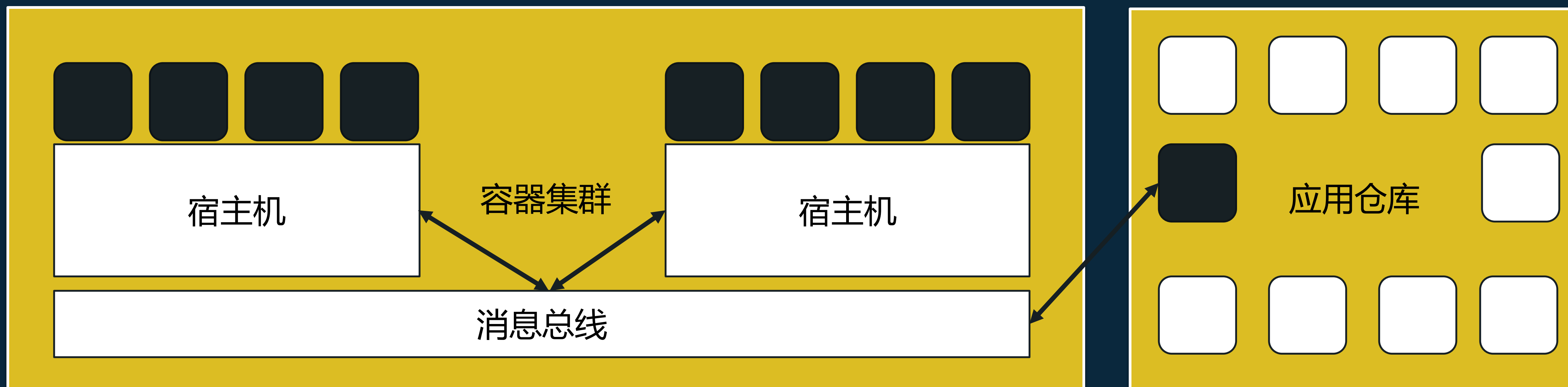


# 容器化交付



应用仓库作为起点，应用镜像从此出发，按照调度策略被部署至对应的宿主机上，启动后成为容器实例。消息总线负责容器集群的指令传递，并实时上报容器集群的运行状况。容器集群的访问入口，由动态负载均衡统一分配。

## 负载均衡



## Dao 法四：持续

| 持续的交付是业务生命线，互联网的能力描述就是迭代速度。不仅要交付组件，也要交付测试。交付无关基础架构。

# 交付流水线



镜像  
Image

1

构建  
Build

2

← 前

交付  
Ship

3

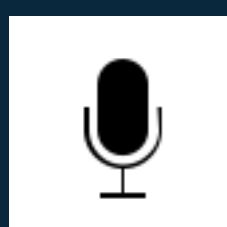
后 →

编排  
Compose

4

运行  
Run

5

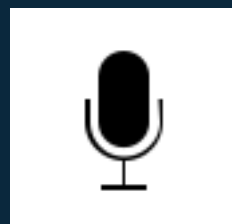
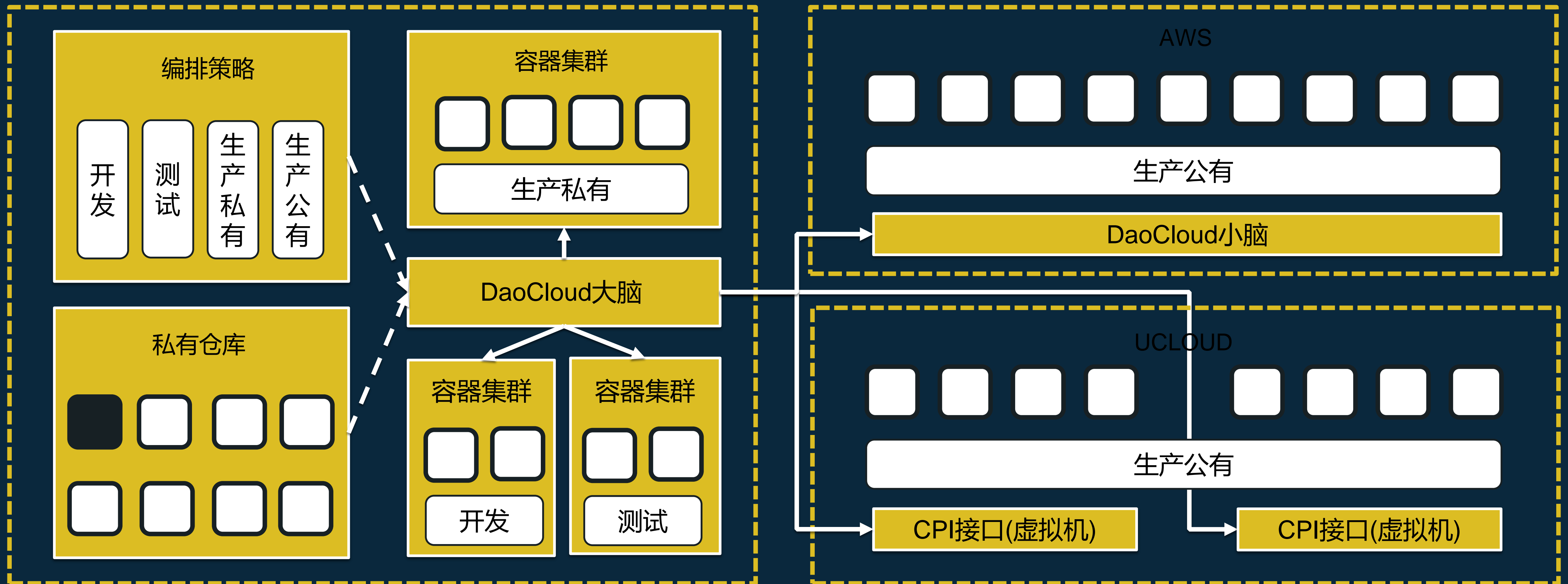


将应用交付的标准统一到容器上，以标准交付件的形式，实现跨阶段的标准化发布流程。  
以容器为核心的交付和运维，确保一致的交付件，贯穿开发、测试、运维场景。

## Dao 法五：弹性

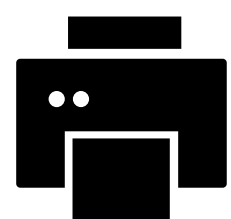
| 弹性扩展是多粒度的，不仅仅是架构能支持弹性，组件也能弹性，最后实现集群弹性。前提是自动化策略具备弹性。

# 混合部署



同一配置，全云通用。

# 容器管理平台



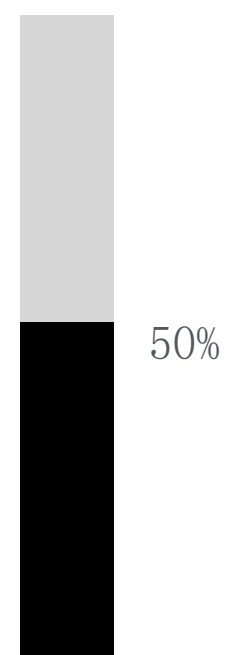
镜像仓库



应用监控



数据备份



微服务



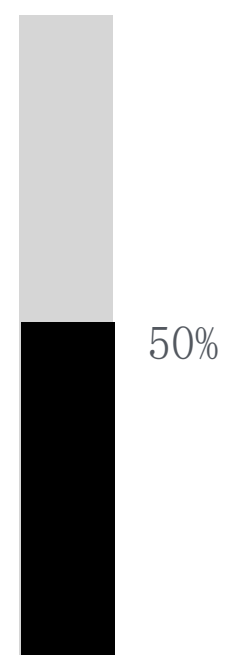
弹性架构



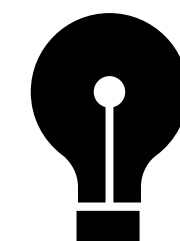
动态绑定



自动构建



灰度发布

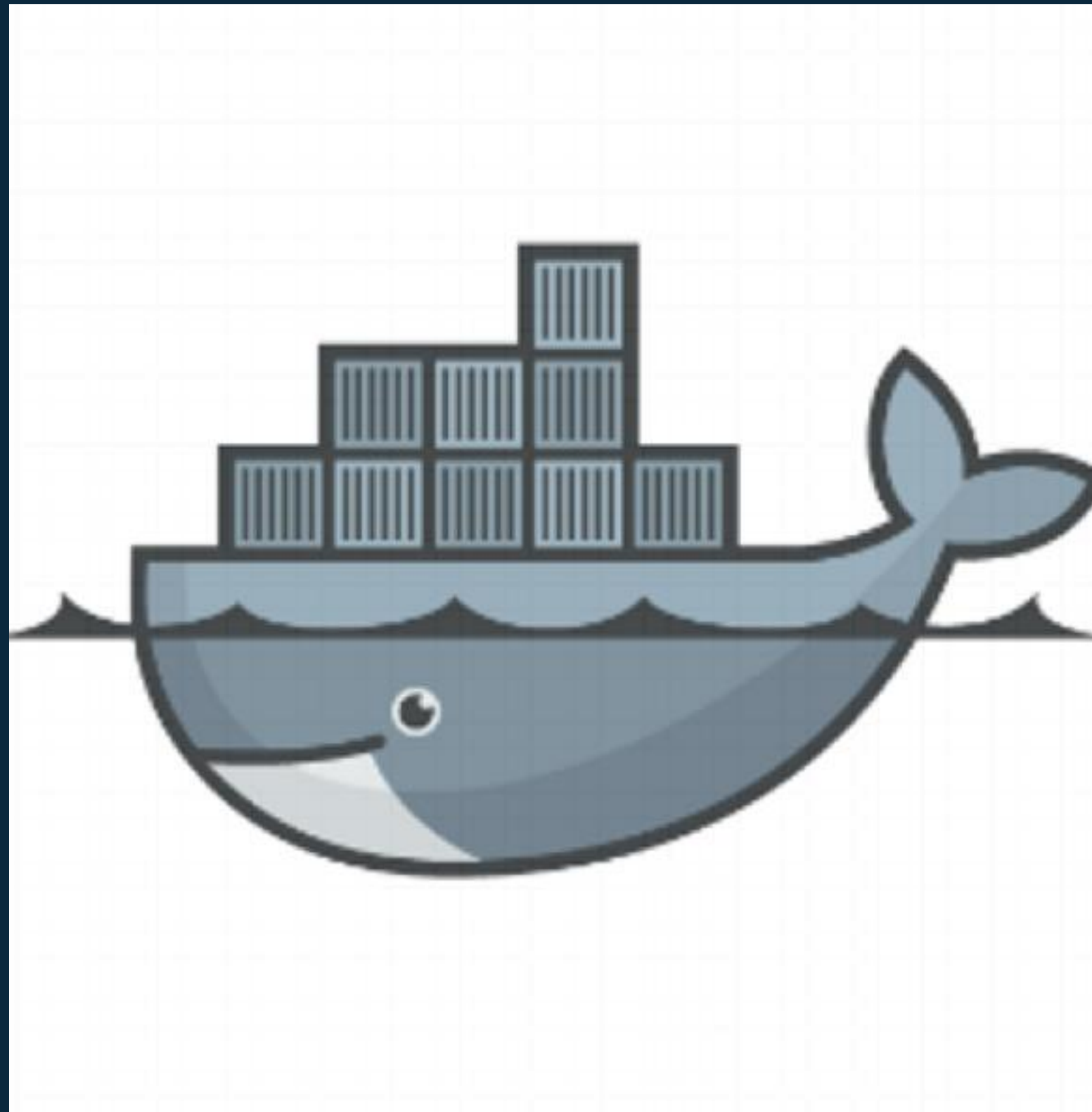


监控分析



应用商店

# 把容器...



制作容器的过程就是你把你的业务分裂开，将一部分藏在变化外的某个部署环境中。这样，即使你的局部遭变化或者重构，业务死不了。因为还有一部分核心在镜像中不受损害。



## …变成魂器



制作魂器的过程就是你把你的灵魂分裂开，将一部分藏在身体外的某个物体中。这样，即使你的身体遭袭击或者毁坏，你也死不了。因为还有一部分灵魂在世间未受损害。

容器打造原力：开发驱动，自动策略，混合部署

# 托 付



# 谢谢

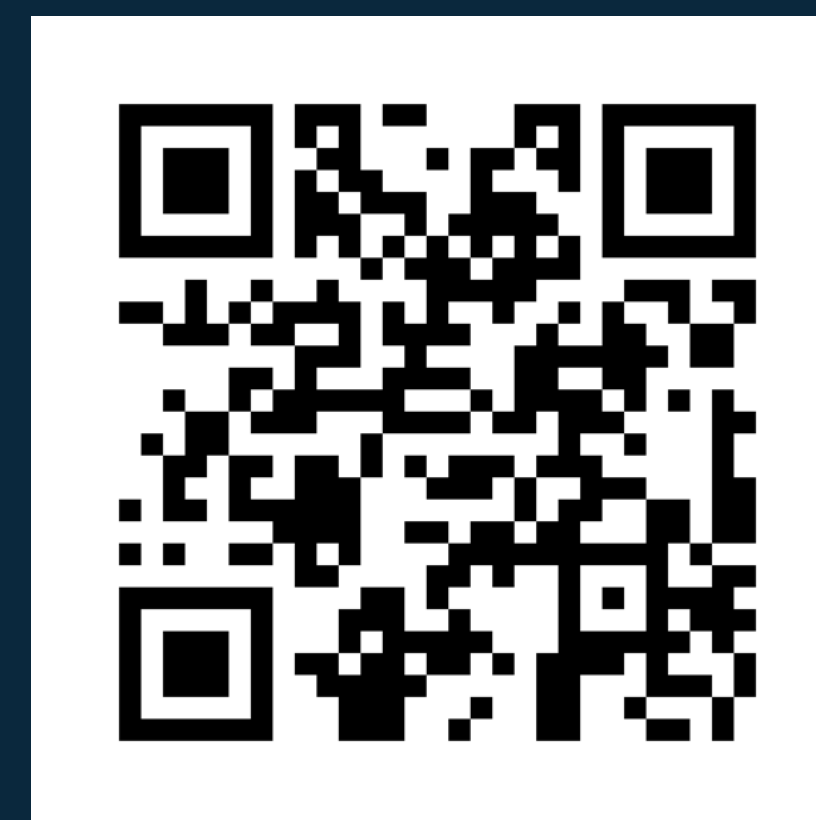
[www.daocloud.io](http://www.daocloud.io)

上海

杨浦区政府路18号波司登大厦901室

[info@daocloud.io](mailto:info@daocloud.io)

「 Container+ 联盟 」





Thanks, Q&A

@chenr6 | daocloud.io