www.vip.com

唯品会“简易”DCOS 实践探讨

平台与架构部 - 基础架构

邱戈川(了哥)

duff.qiu@vipshop.com

微信：duffqiu

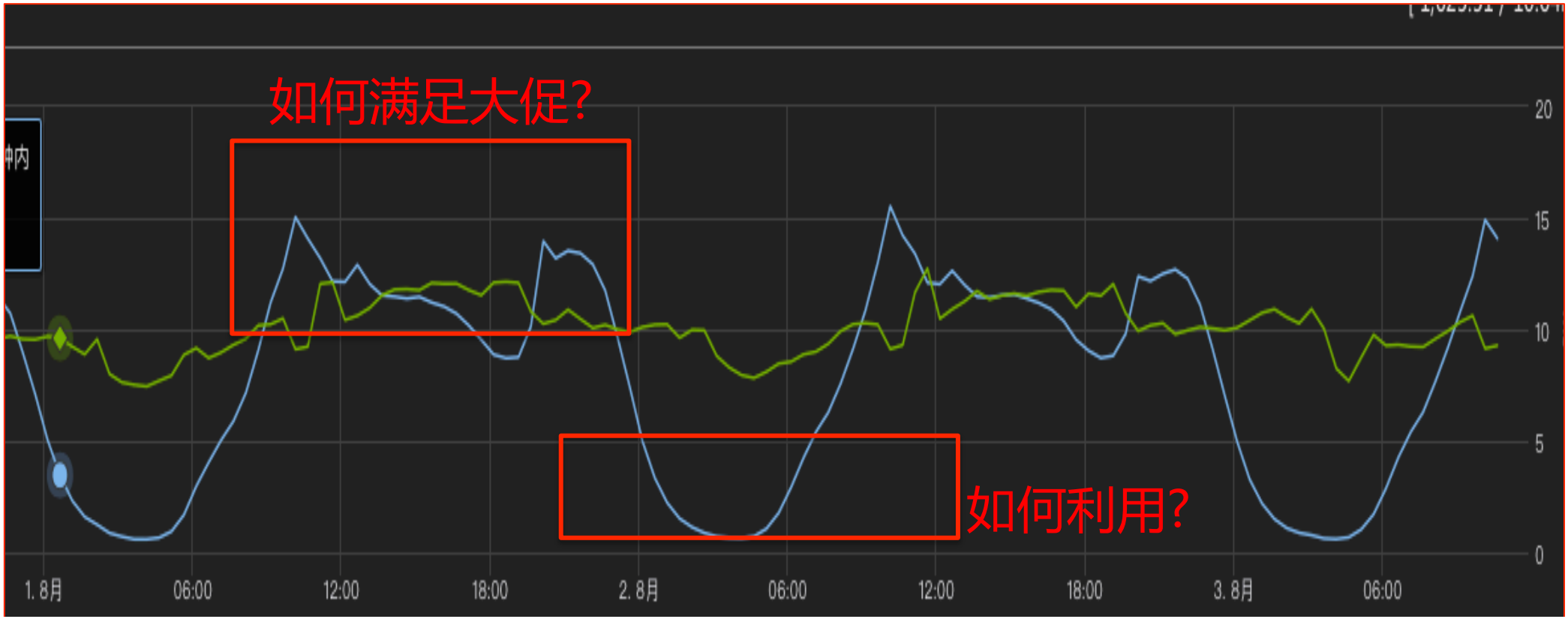
2016.8

唯品会 vip.com 一家专门做特卖的网站

唯品会
vip.com
一家专门做特卖的网站

Part I：为什么需要容器化平台与其概貌

唯品会经典的服务运行图



新特卖每日早10晚8上新

现今面临的问题

运维思路过于简单：单进程部署

物理机体系，运维难度大

资源利用率低

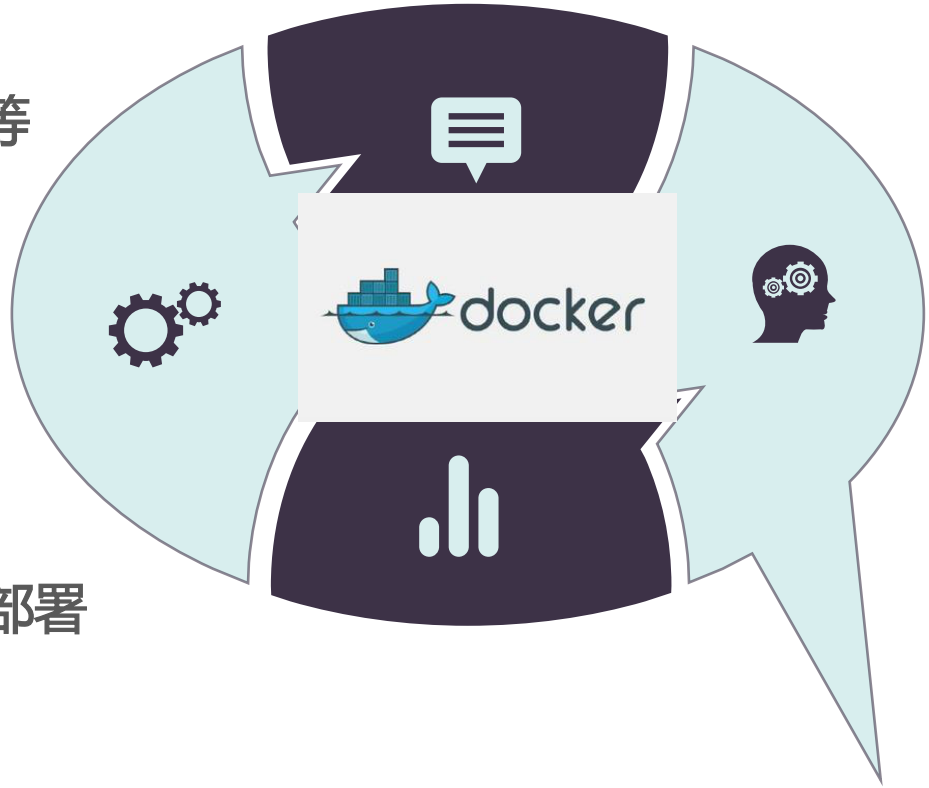
灵活性差，响应慢

效率低、成本高

什么是我们需要的管理平台?

资源高效复用:
分时、分CPU等等

支持多数据中心
混合数据中心

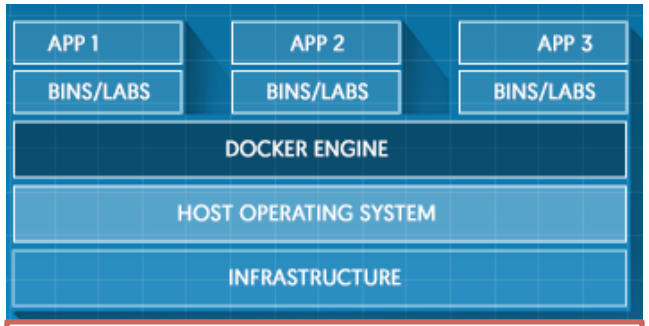


快速部署、自动部署
严格隔离

足够的弹性：
多应用类型、多主机类型
易于伸缩(实例伸缩、配置伸缩)
业务分组

Docker就够了？

什么是Docker容器、容器编排



Docker引擎

- 容器包含了应用本身以及所有相关的依赖在一个镜像中
- 共享一个Linux内核
- 运行在隔离的进程以及用户空间中

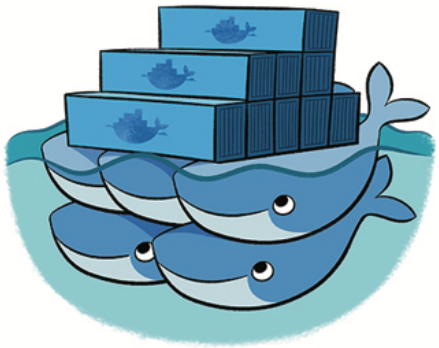
资源共享

可移植性

独立隔离

低损耗

环境一致性

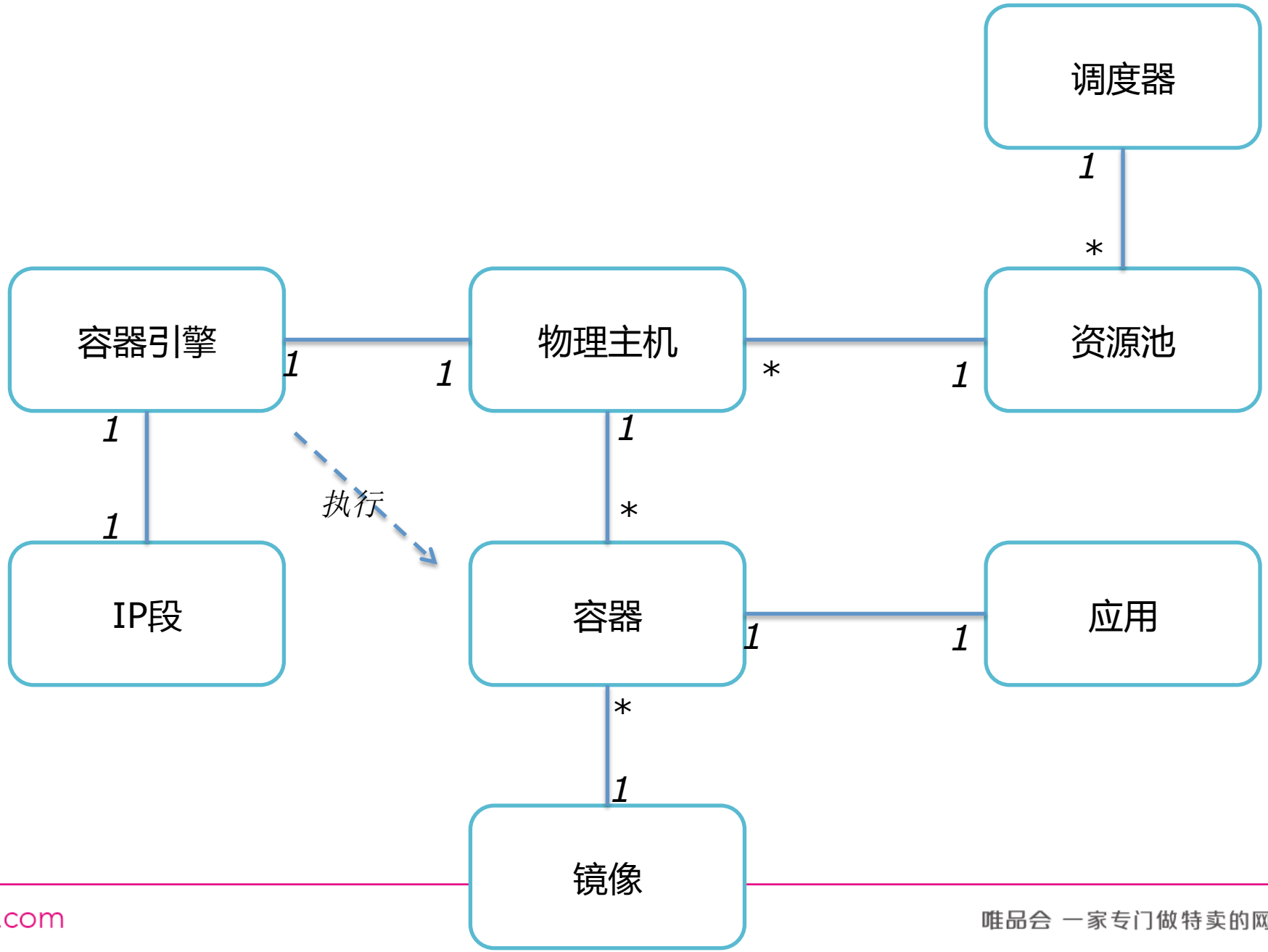


容器编排

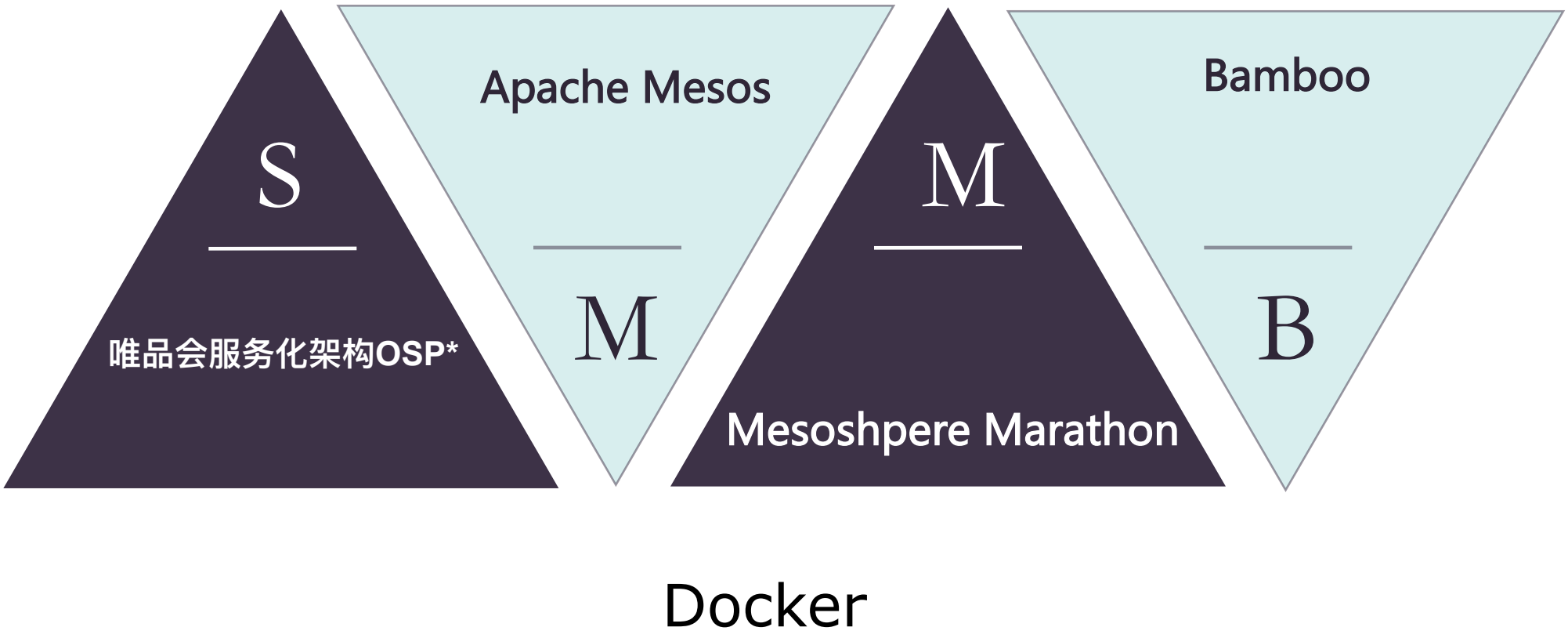
弹性伸缩

资源调度

容器化平台的概念模型



我们的容器化技术选型



<<唯品会RPC服务框架与容器化演进>>

Mesos+Marathon Vs. Kubernetes

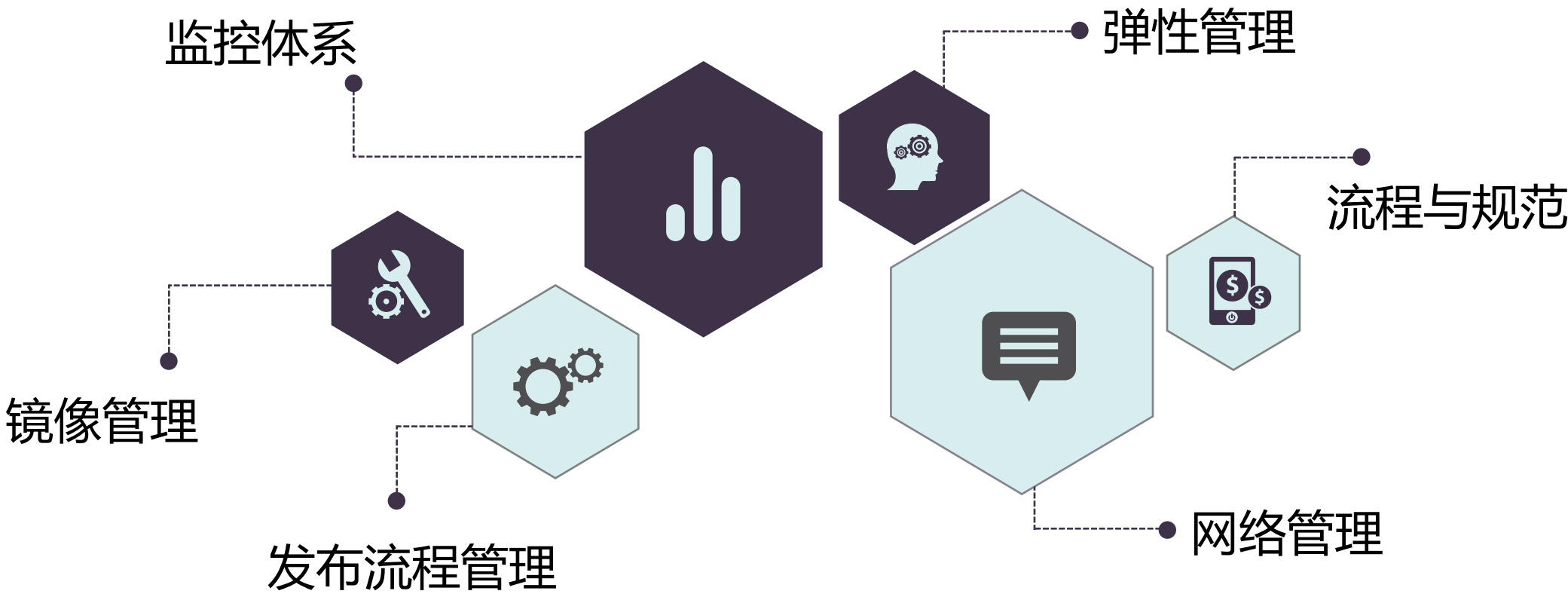
Mesos+Marathon

- ✓ 物理资源管理与调度，轻量级IaaS
- ✓ 完美适配现有服务化体系，以及多种分布式系统
- ✗ Web类应用需要额外框架支撑
- ✓ 多层次灵活调度，可自定义扩展
- ✓ 支持约束性调度，适合业务分组和隔离
- ✓ 容易搭建混合云模式弹性伸缩
- ✗ 容器编排/依赖比较困难
- ✓ 已经有成功案例
- ✓ 成熟度高

Kubernetes

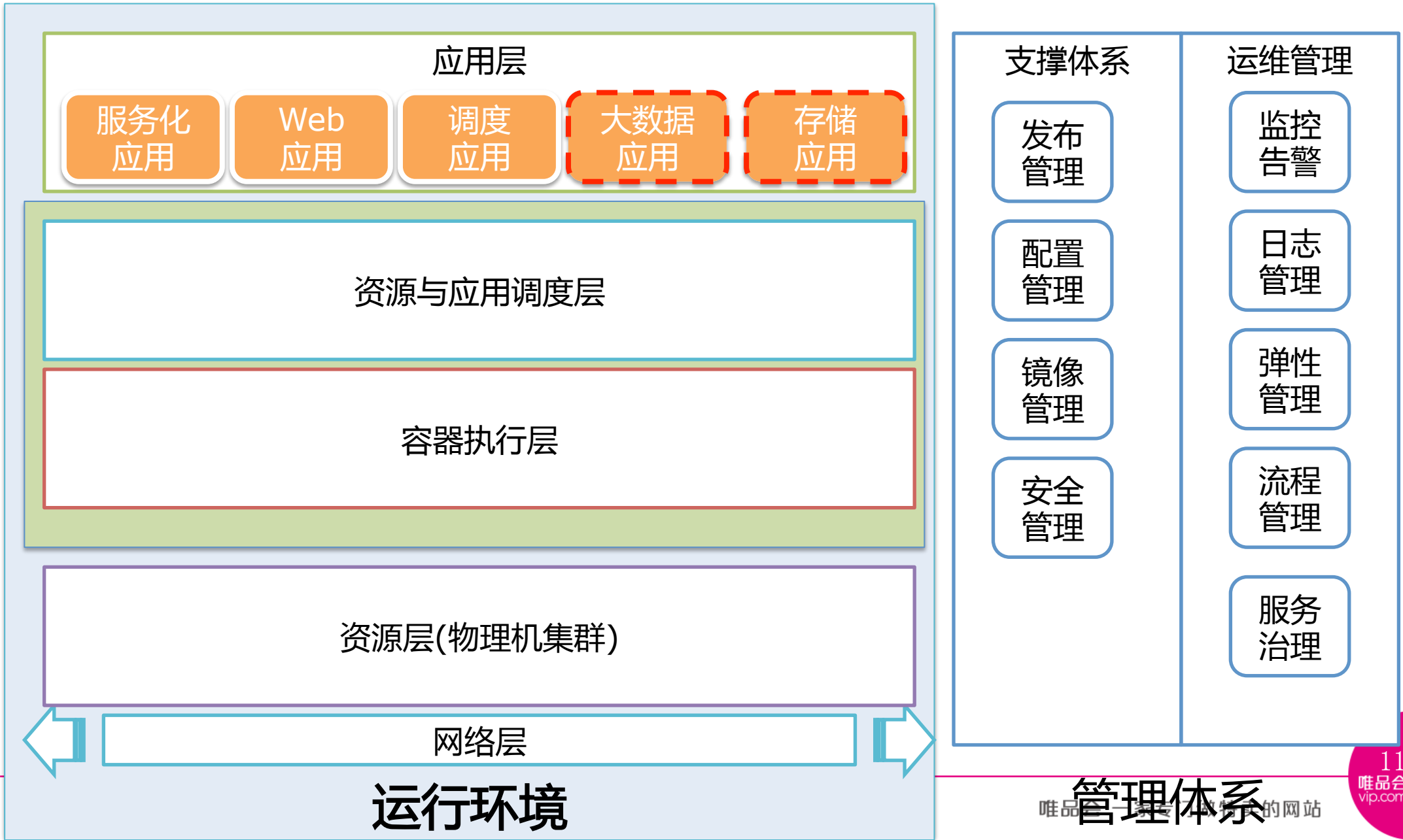
- ✗ 缺乏IaaS层的抽象
- ✗ 与服务化体系功能重合度大，支撑应用类型有限
- ✓ 预置Web类应用的服务注册与发现
- ✗ 调度可扩展性差
- ✗ 无业务分组功能
- ✗ 单一云模式
- ✓ 灵活的容器编排
- ✗ 没有生产案例
- ✓ 社区活跃度高

还欠缺什么？



容器化平台是一个体系

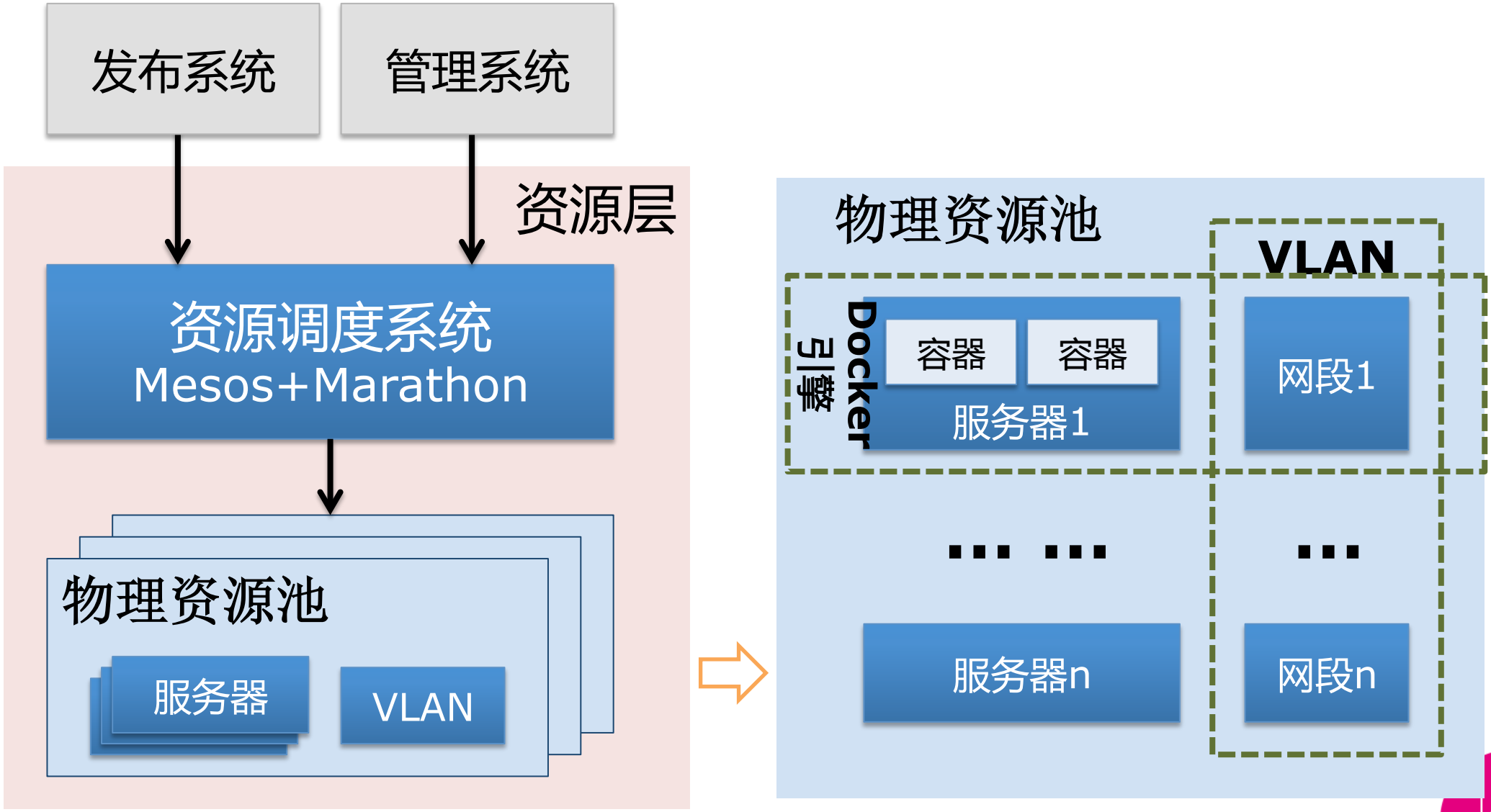
整体架构架构



唯品会
vip.com
一家专门做特卖的网站

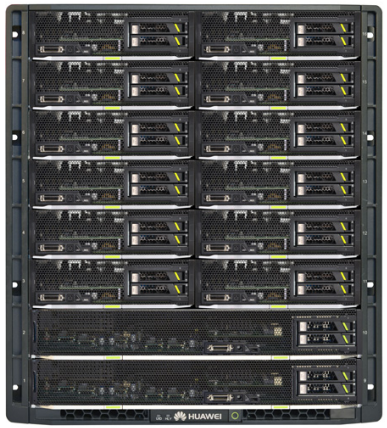
Part II：关键技术点

资源层模型

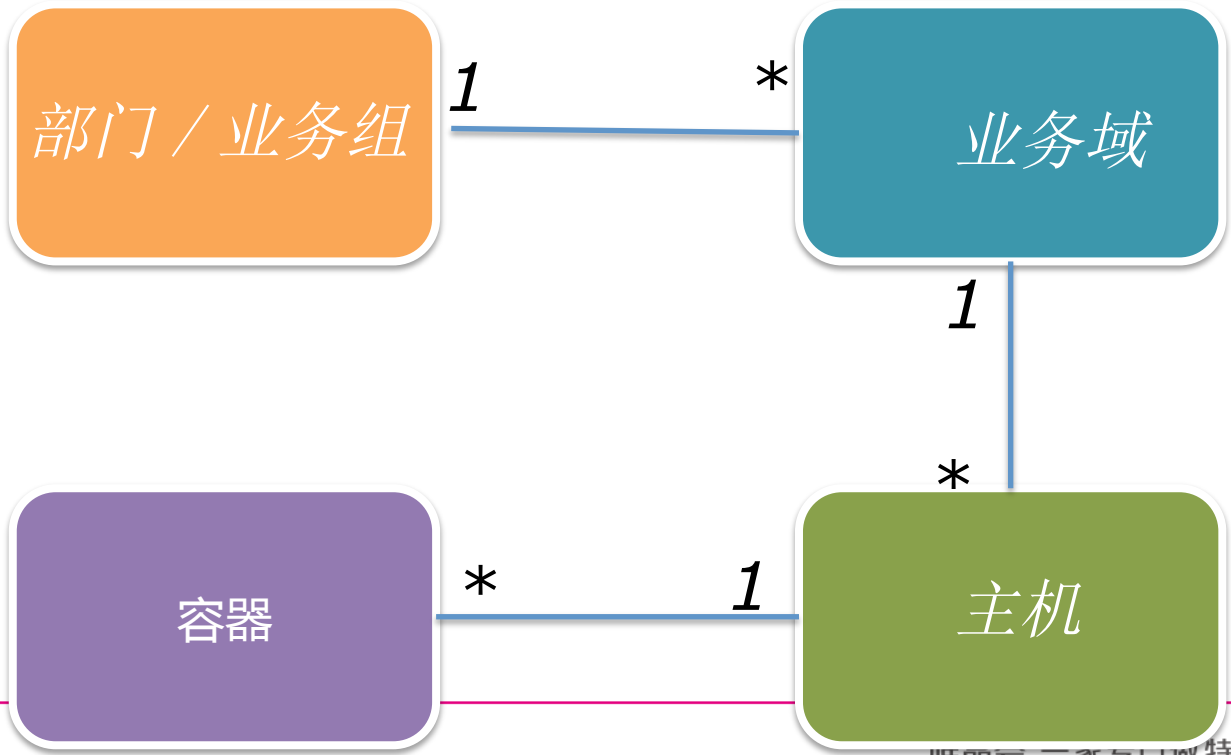

发布系统
管理系统
资源层
资源调度系统
Mesos+Marathon
物理资源池
服务器
VLAN
物理资源池
VLAN
Docker
引擎
容器
容器
服务器1
网段1
服务器n
网段n

物理资源

- 沿用现有的PC服务器管理机制
- 通过CMDB管理相关资源
- 上线流程、回收流程不变
- CMDB管理物理机与容器的对应关系

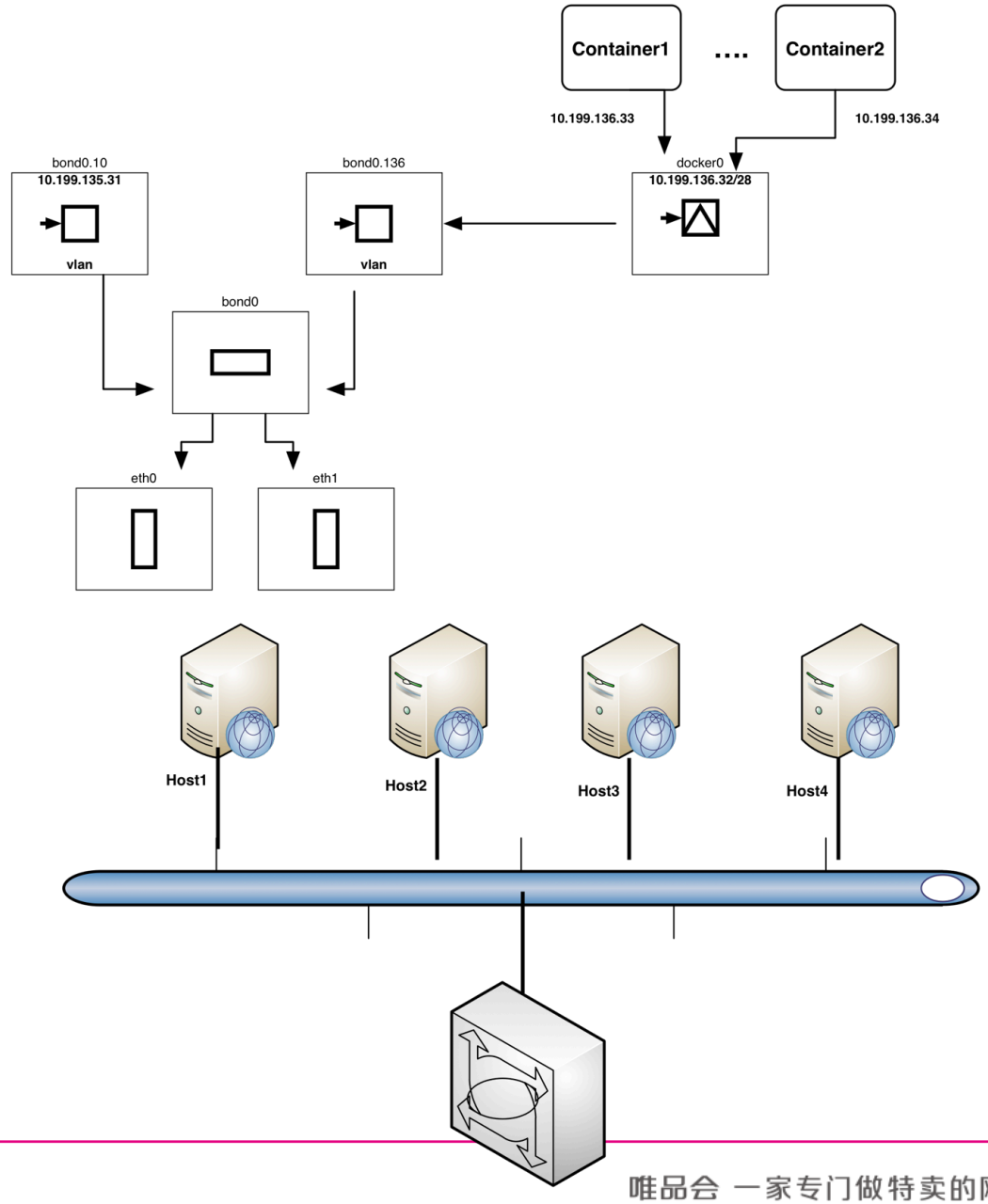


PC服务器

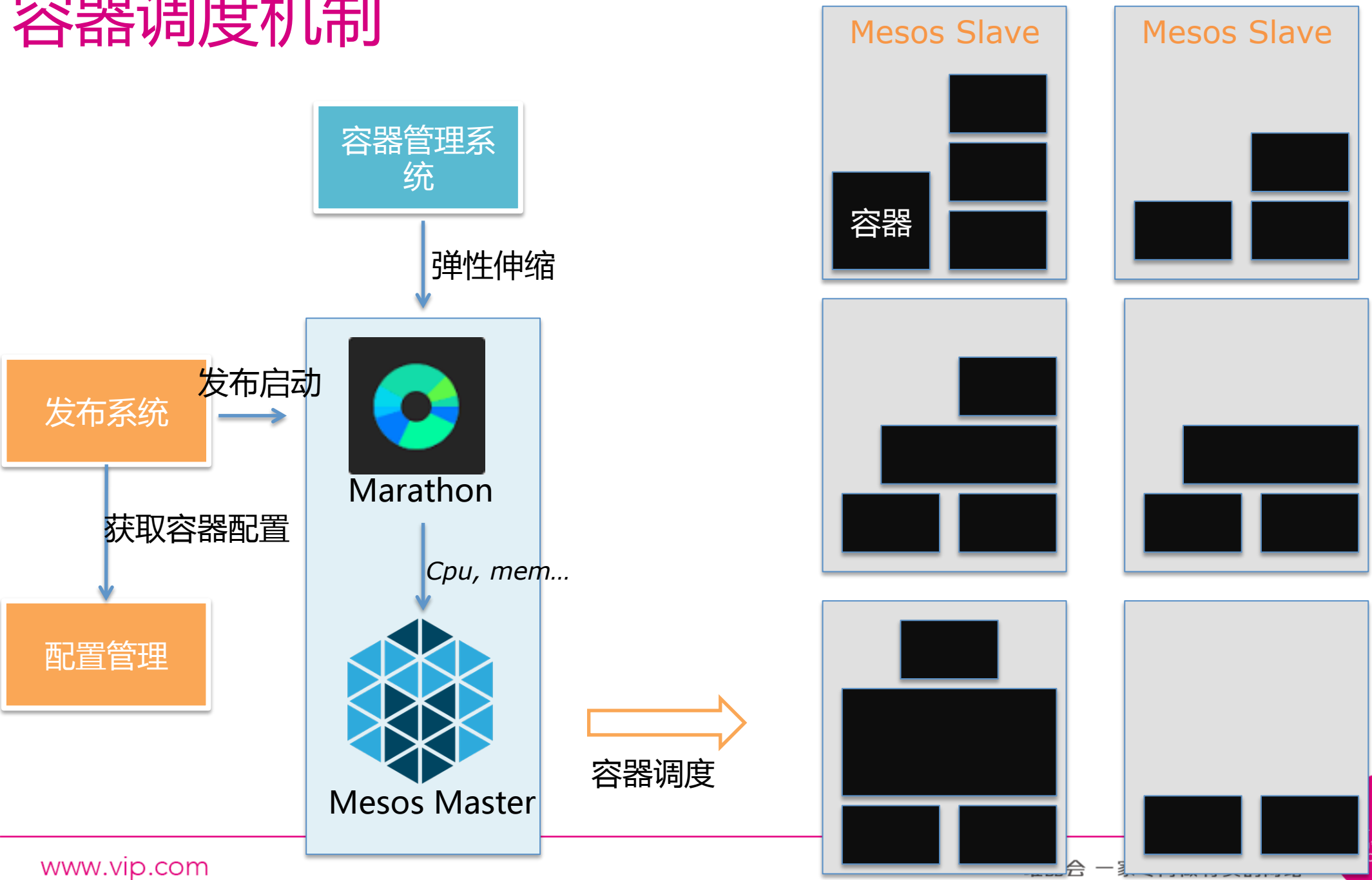


网络模型: Layer 2 - VLAN

- 使用Linux Native Vlan的方式，创建一个独立的Vlan作为Docker的Bridge
- 每台Mesos Slave机器配置一个独立IP段，由Docker Engine来自动分配和回收
- 不同域间网络隔离通过Vlan实现
- 每个容器分配一个独立IP
- 容器销毁IP会被回收，不可以在不同主机间漂移
- 网络监控使用网络设备配套监控系统



容器调度机制


容器管理系统
弹性伸缩
发布系统
发布启动
Marathon
获取容器配置
Cpu, mem...
配置管理
Mesos Master
容器调度
Mesos Slave
Mesos Slave
容器

容器化的执行

执行引擎

单容器单进程管理

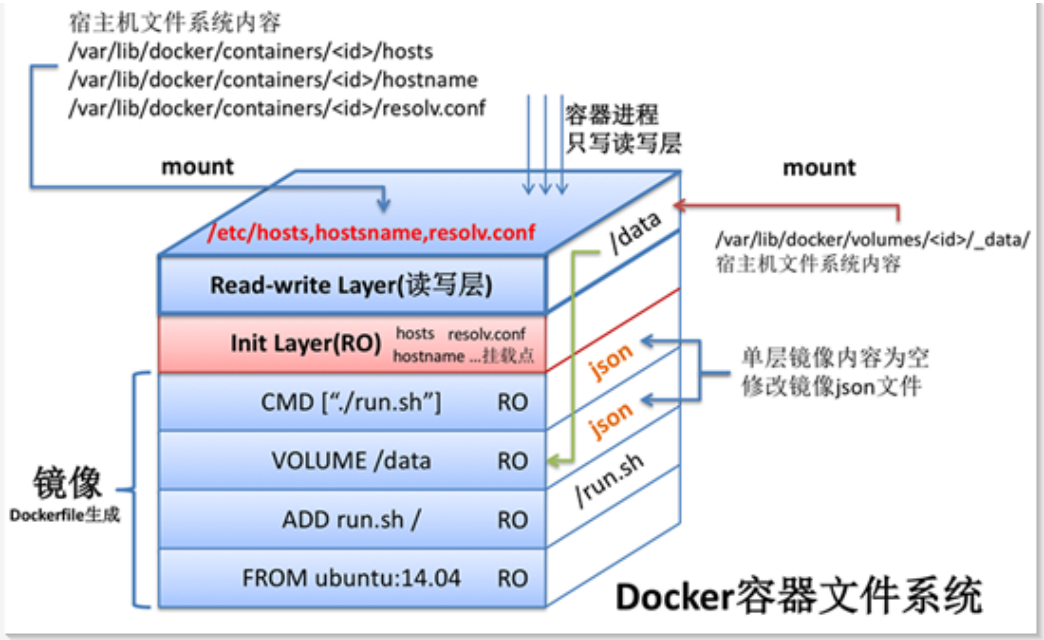
资源隔离

面向服务化

版本管理

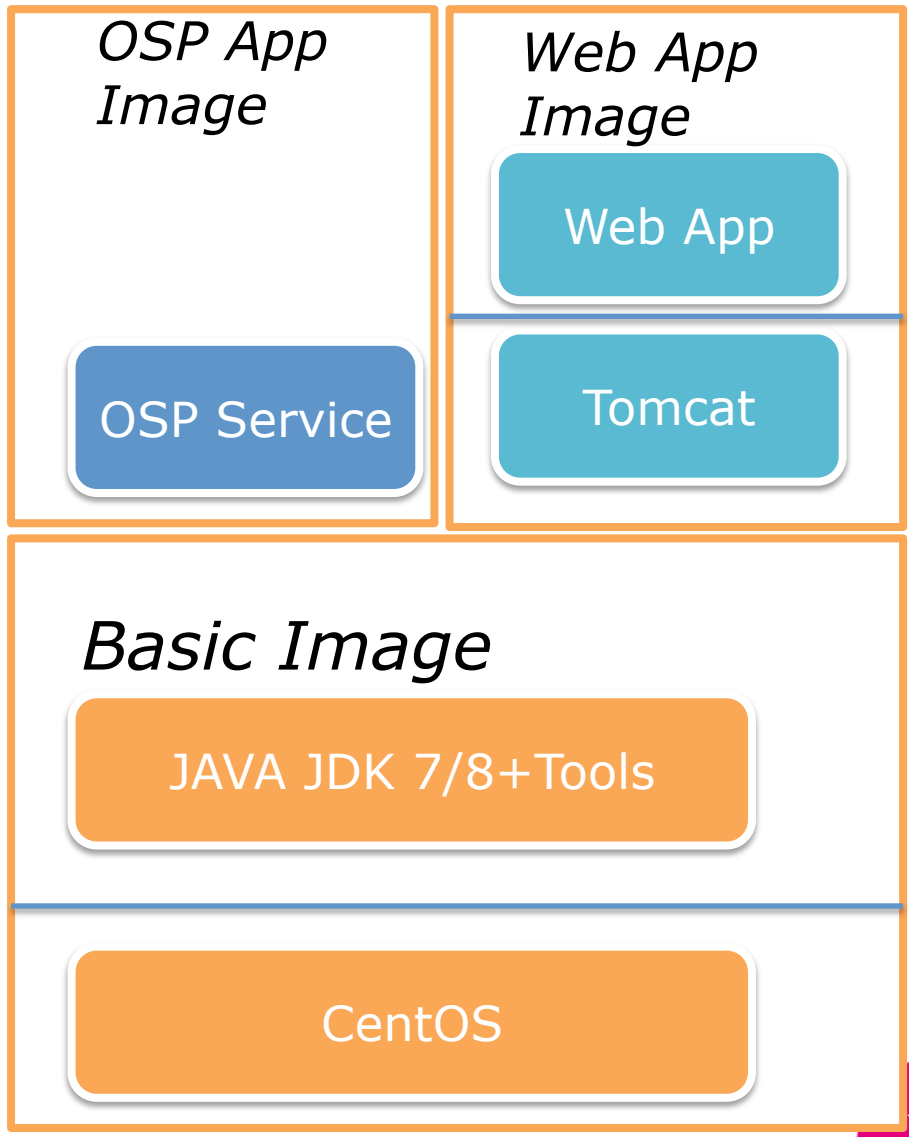
组件重用

镜像机制

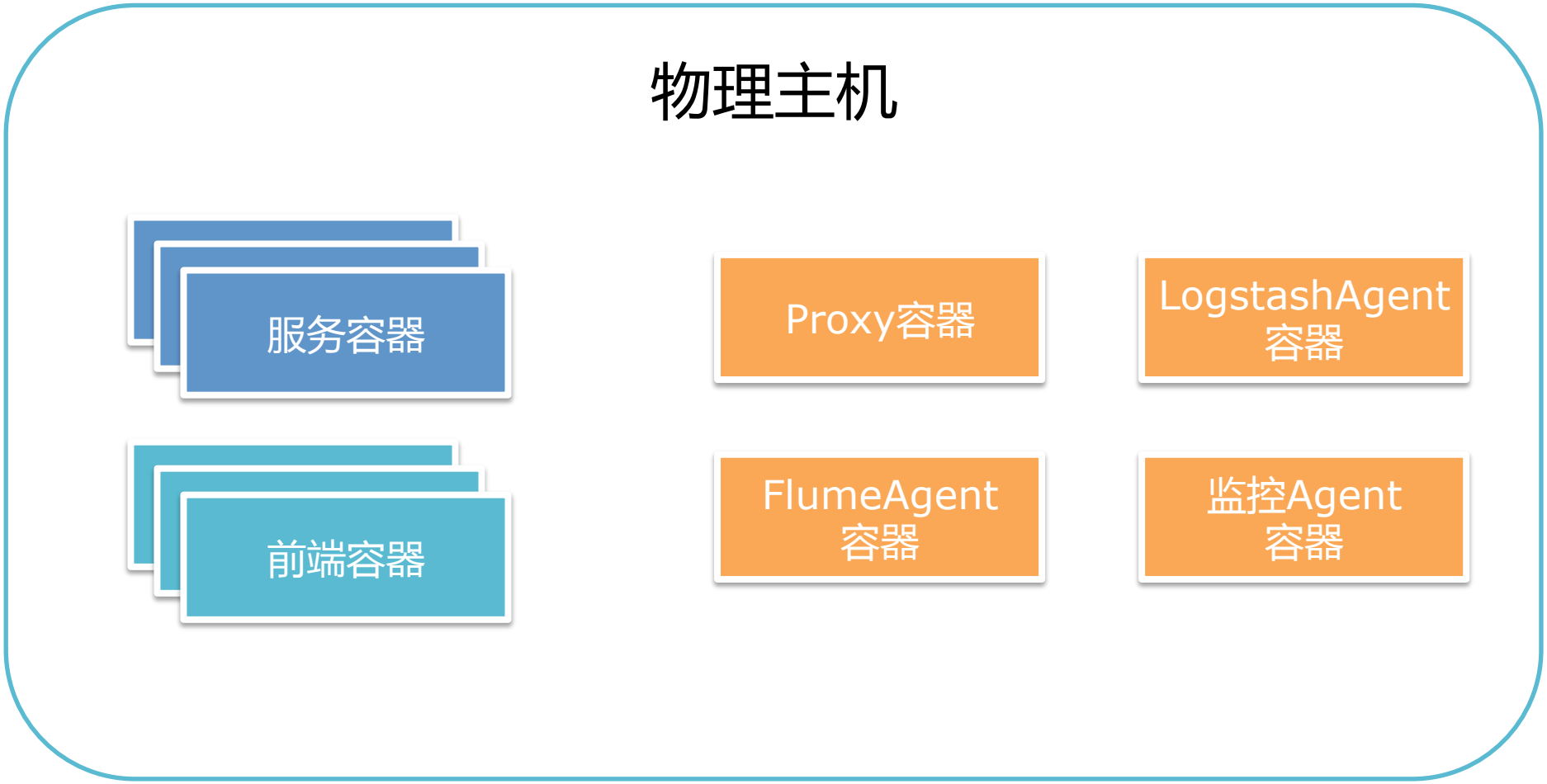


- 多层结构，逐层叠加
- 不可改变的特性
- 无状态性

环境一致性



应用容器化

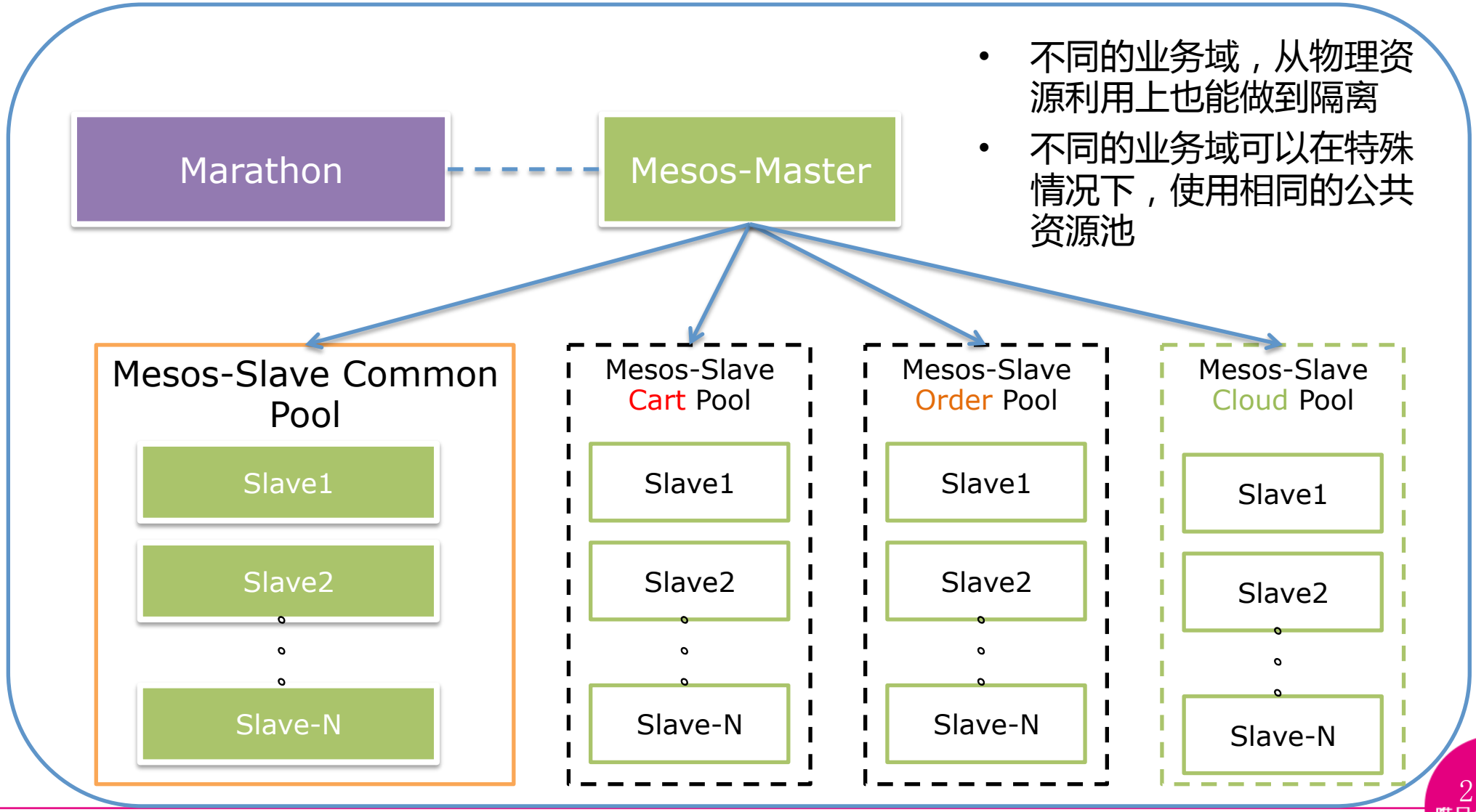


- 单容器独立IP
- 单容器单进程
- 通过Proxy容器实现服务注册、发现、路由和治理

容器存储

- 使用容器本地存储
- 通过卷挂载方式使用物理机的本地存储

业务分区与资源共享



- 不同的业务域，从物理资源利用上也能做到隔离
- 不同的业务域可以在特殊情况下，使用相同的公共资源池

生态体系监控与告警

Telescope

Dragonfly
Mercury

容器管理系统

Zabbix

原产设备系统

容器监控项：

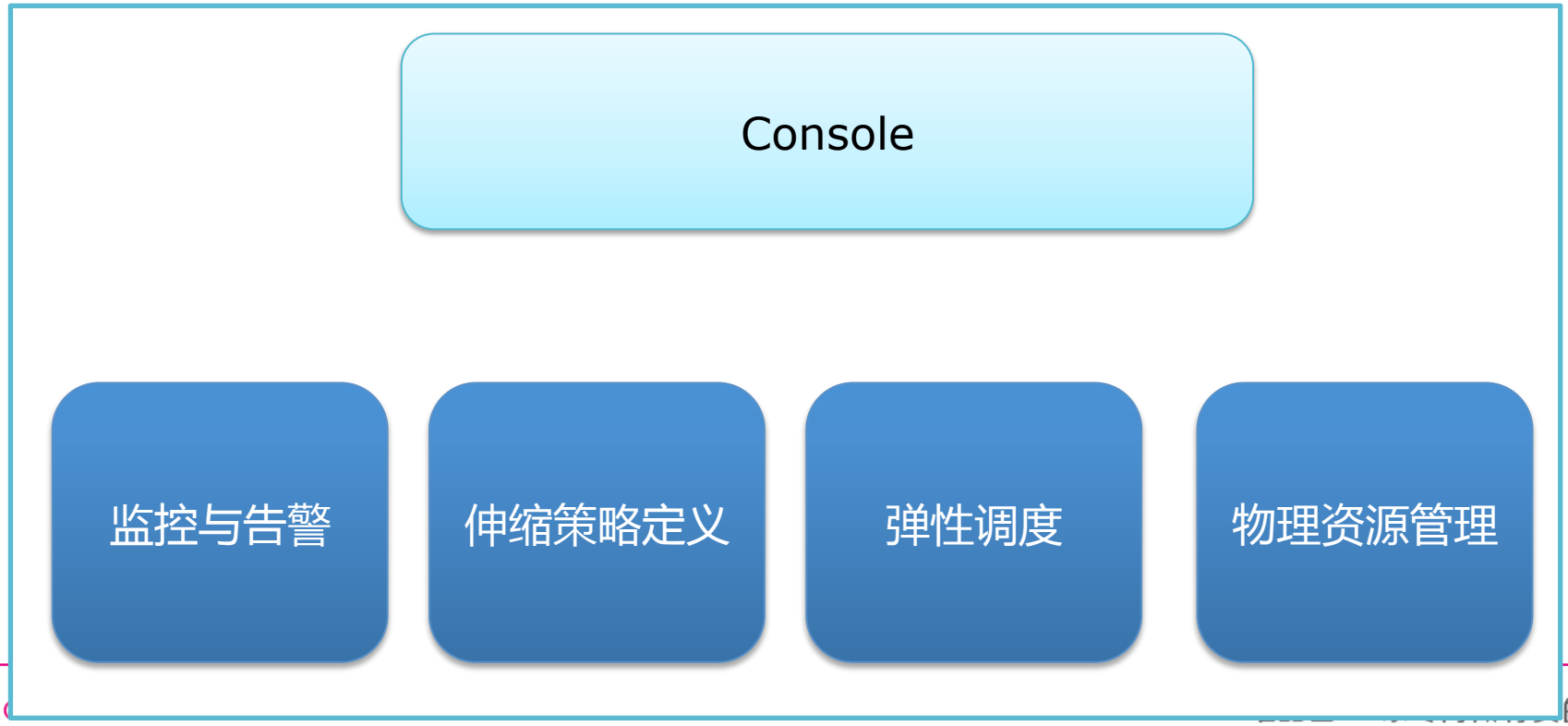
- CPU
- Disk IO
- Net IO
- Process
- Memory
- JVM监控
- …

其它系统可能需要配合容器化改造

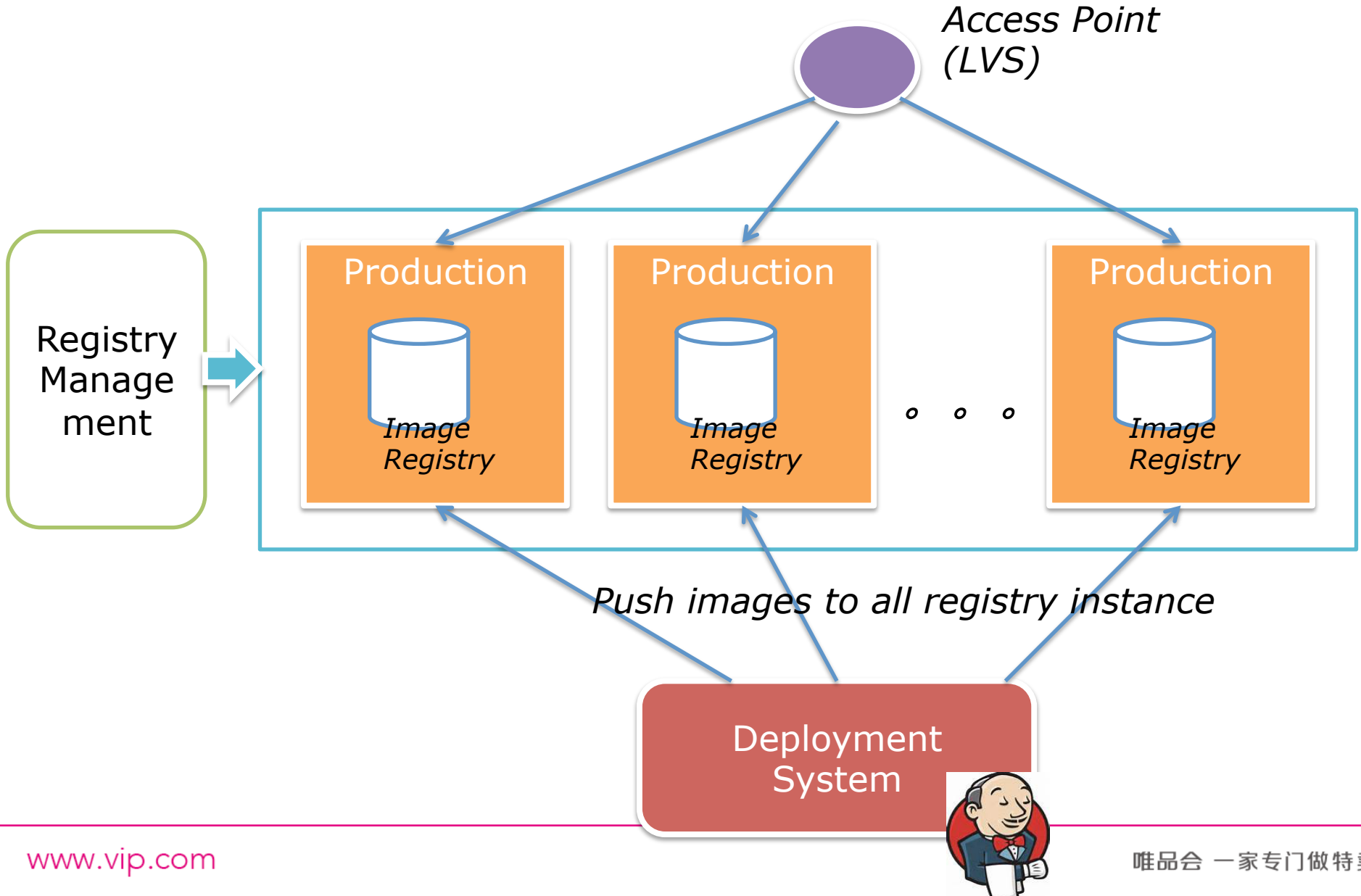
容器化管理系统:

系统设计目的

- 物理资源上线与回收
- 容器的监控与告警
- 容器的伸缩策略管理和弹性调度



Registry HA 方案



配套流程与规范

应用 / 服务开发流程、打包流程

持续集成、持续测试流程

监控告警处理流程

镜像发布维护流程

物理资源上线、回收流程

“黑科技” - 1

- 文件系统/硬盘相关
  - 修改swappiness的默认值，默认值是60
    - echo vm.swappiness=10 >> /etc/sysctl.conf
  - 通过数据预读并且记载到随机访问内存方式提高磁盘操作,默认值128
    - echo "8192" > /sys/block/sda/queue/read_ahead_kb
- Docker Engine的启动参数
  - --fixed-cidr=172.20.56.16/28 --default-gateway=172.20.56.1 --storage-driver=overlay --ip-forward=false --iptables=false --log-driver=journald

“黑科技” - 2

- docker容器里面的时间与主机时间不一致
  - 用docker run启动容器时加上“TZ”环境变量，例如：-e "TZ=Asia/Shanghai"。
- 当启动mesos-slave时出现类似错误信息时：Failed to perform recovery: Incompatible slave info detected
  - 可以通过清空work_dir目录下的meta文件夹内容，work_dir的默认安装路径为：/var/lib/mesos，例如：
  - 执行命令：rm -rf /var/lib/mesos/meta
  - 重启mesos-slave即可：systemctl restart mesos-slave
- 日志文件存储
  - 挂载主机目录到容器
  - 以容器ip作子目录

后续技术探讨

- 网络：
  - Contiv
  - Calico
- 存储：
  - Ceph
- Registry HA / 管理
  - VMware Harbor

广告：

欢迎加入唯品会！

*内推请发简历：**duff.qiu@vipshop.com***

谢谢

Thank you

www.VIP.com