2021年
云原生行业研究报告(三)：容器技术

2021 Cloud Native Industry Research Report (3): Container
クラウドネイティブ産業調査レポート(3): コンテナ

报告标签：云计算、虚拟化、IaaS、Docker、Kubernetes

主笔人：胡竣杰

概览摘要

容器技术是虚拟化技术不断发展的必然产物，但正如微服务作为应用架构的前沿技术也不会止步。随着时间的推进，业务需求与技术栈对相关技术需求相互如影随形蛇形发展。

自2008年谷歌开创的cgroups容器部署并入Linux内核后，容器技术的发展逐年加速。Docker社区在2013年浮现，届时有30多家容器供应商，紧接着2014年Kubernetes容器编排项目宣布开启，仅用一年就对外发布，容器生态全面升级。时至今日，容器已经通过了十余年的发展，实现了快速迭代、高资源利用率、无限弹性扩容等特性，为云计算时代的全面开启提供了极为重要的底层技术支持。

Spring微服务框架+Docker容器+K8s集群管理被誉为在春天的货船上的盛世，云原生的大环境下，容器技术将不断迭代的同时成为广泛行业与企业IT架构建设的底层，全面赋能企业的敏捷与业务创新。

01 虚拟化技术与容器

虚拟化技术提高了IT敏捷性、灵活性和可扩展性，是云计算中最关键、最核心的技术原动力。容器技术将应用组件与依赖打包为一个标准、独立、轻量的环境，来部署分布式应用，比虚拟机以更小的粒度控制资源。

虚拟化技术与容器技术的发展体现了云计算背景下计算机底层架构正向低耦合、高灵活的整体技术趋势发展。

02 容器技术与容器编排

凭借镜像打包技术的轻量化和只读特性，容器实现快速更新迭代、提高资源利用率、快速复制弹性扩容。

在企业级应用中，大量的容器共同参与导致的运行复杂性与故障催生了容器编排管理工具的需求，Kubernetes应运而生。

03 容器的发展现况及趋势

容器凭借业务价值和技术价值满足了来自不同行业的企业的多层面需求，容器市场迎来创新与增长的快速发展。

容器在自身技术迭代发展的同时，与云原生架构的协同正进一步深化，与新兴技术架构融合升级并拓宽下游应用场景。

目录

CONTENTS

图表目录

01

虚拟化技术综述

- 虚拟化技术概述
- 容器虚拟化技术简述
- 虚拟化技术的演化路径

虚拟化技术概述

- 虚拟化技术提高了IT敏捷性、灵活性和可扩展性，是云计算中最关键、最核心的技术原动力

虚拟化的定义

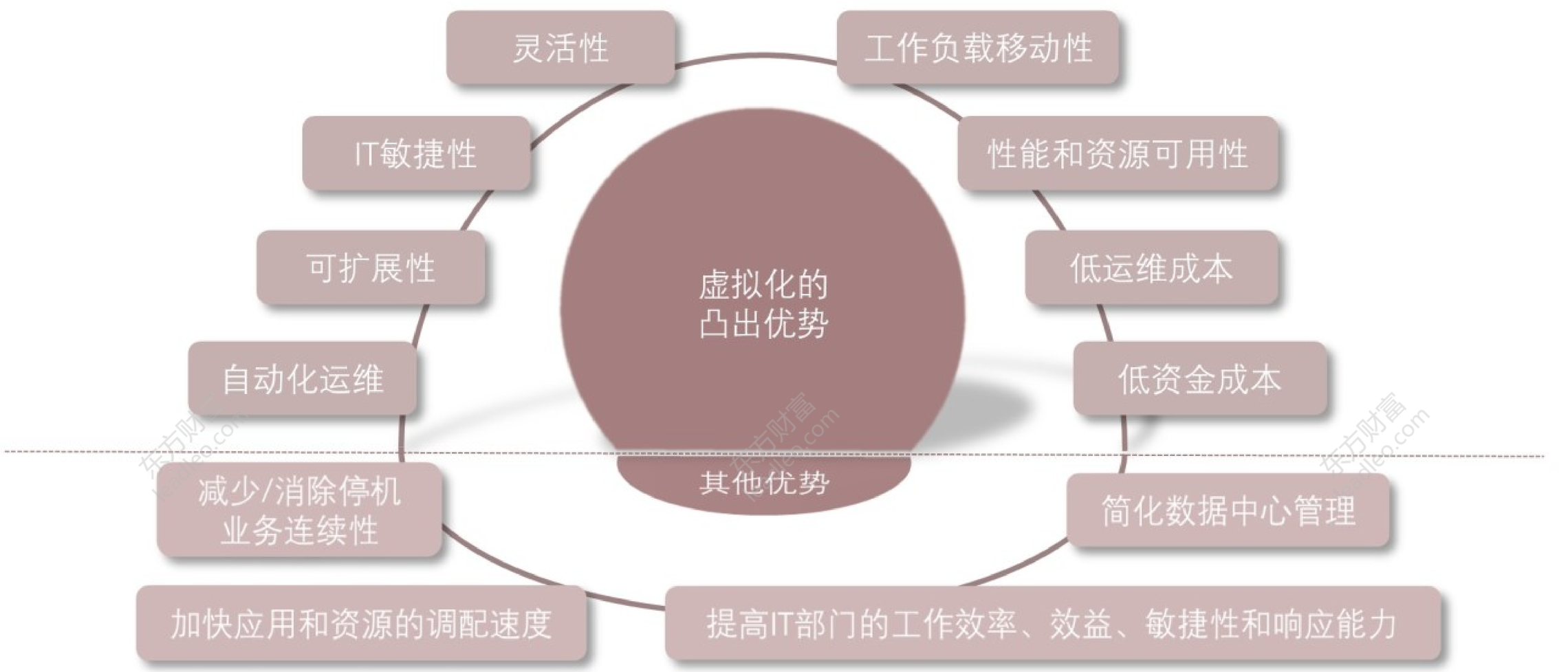
虚拟化（英语：Virtualization）是一种资源管理技术，是将计算机的各种实体资源，如服务器、网络、内存及存储等，予以抽象、转换后呈现出来打破实体结构间的不可切割的障碍，这些资源的新虚拟部分是不受现有资源的架设方式，地域或物理组态所限制。一般所指的虚拟化资源包括计算能力和资料存储。实现IT资源的动态分配、灵活调度和跨域共享，可以满足应用灵活多变的需求。

云计算与虚拟化技术

云计算的发展是虚拟化、分布式系统、分布式并发编程模式、面向对象的体系架构、软件即服务和信息安全等技术共同作用与发展的结果。云计算实现的关键突破在于资源使用方式的改变。

虚拟化技术实现了动态配置和扩展计算和存储的云计算资源，而逻辑上以单一整体的服务形式呈现给用户，成为了云计算中最关键、最核心的技术原动力。

虚拟化技术的优势



来源：头豹研究院

❑ 服务器虚拟化

通过虚拟化技术将计算机虚拟为多台逻辑计算机，通过硬件和操作系统之间引入虚拟化层实现硬件与操作系统的解耦，从而实现服务器的虚拟化。

虚拟化层的功能在于通过动态分区，实现一台物理服务器上同时运行多个操作系统实例，共享物理服务器资源，每个虚拟机从而得到一套独立的模拟出来的硬件设备，包含了CPU、内存、存储、主机、显卡等硬件资源。

服务器虚拟化的架构主要有：寄居架构和裸金属架构。

- **寄居架构 Hosted Architecture**

寄居架构将虚拟化层架在操作系统之上，当作一个应用运行。依赖于主机操作系统对设备的支持和物理资源的管理。

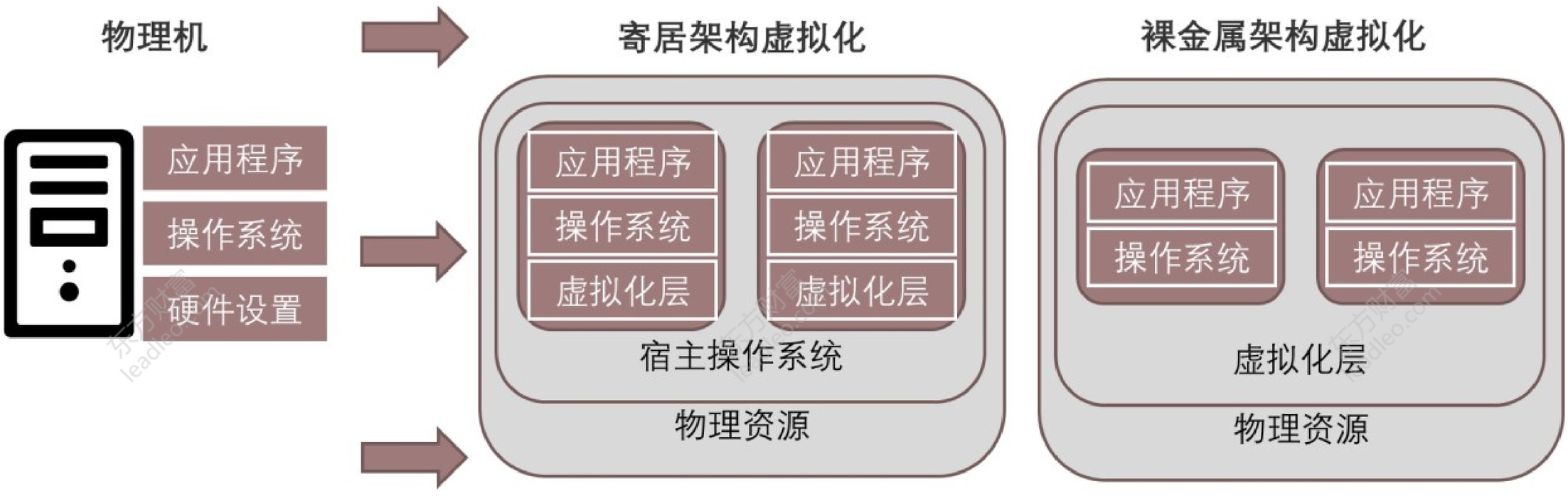
产品代表：VMware Server

- **裸金属架构 Bare Metal Architecture**

裸金属架构将虚拟化层直接运行在x86硬件系统上，在其之上安装操作系统和应用。通过解除操作系统与物理主机之间的紧耦合，使操作系统的部署更为轻便，可以直接访问硬件资源而不需要通过操作系统来实现对硬件的访问，具有更高的效率，工作负载的移动性显著增强。

产品代表：Xen、XenServer、VMware ESX Server、KVM

服务器虚拟化架构

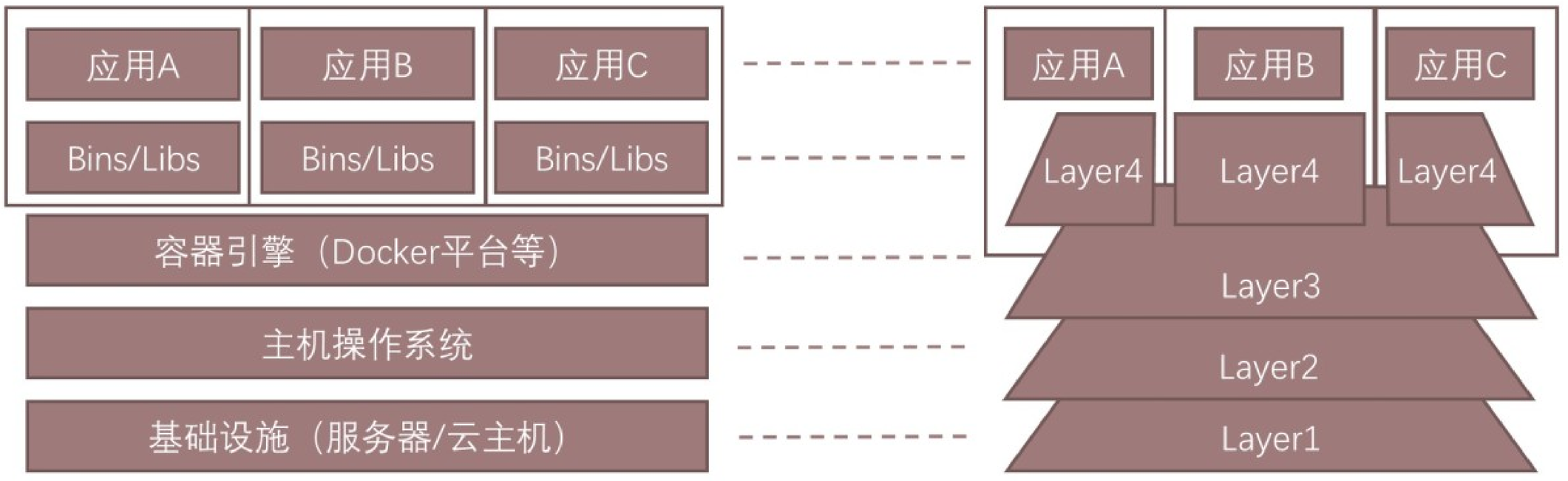


来源：成都信息工程大学，头豹研究院

容器虚拟化技术简述

- 容器技术将应用组件与依赖打包为一个标准、独立、轻量的环境，来部署分布式应用，比虚拟机以更小的粒度控制资源

容器虚拟化架构概述



来源：成都信息工程大学、Cloudman，头豹研究院

容器技术

容器是一种轻量级、可移植、自包含的软件打包技术，使应用程序可以在几乎任何地方以相同的方式运行。开发人员在自己笔记本上创建并测试好的容器，无须任何修改就能够在生产系统的虚拟机、物理服务器或公有云主机上运行。容器由两部分组成：

1. 应用程序本身
2. 依赖：各种依赖的二进制文件和库。每一个客户机操作系统都需要安装许多依赖。比如应用程序需要的库或其他软件容器在Host操作系统的用户空间中运行，与操作系统的其他进程隔离。

容器技术的低耦合、高灵活

虚拟机通过解除操作系统与物理主机之间的紧耦合，显著提升了操作系统的部署或工作负载的移动性。

但是当用户仅仅需要使用一小部分资源来运行一个简单的应用，却虚拟出一整台计算机来完成软件的发布却不仅仅浪费空闲的系统资源还需要等待启动虚拟机的运行时间。因此，容器以比虚拟机更小的资源分配粒度更好的满足这类轻量快速应用发布的需求。

如架构概述图显示，容器是用Layer来建立的，多个容器共享基础层以减少资源的使用或浪费，容器对共享资源进行隔离、限制、审计等，确保了容器只使用其必需的资源，为应用程序提供了一个隔离的运行空间。

虚拟化技术的演化路径

- 虚拟化技术与容器技术的发展体现了云计算背景下计算机底层架构正向低耦合、高灵活的整体技术趋势发展

计算机底层资源架构

	传统物理机架构	虚拟机架构	容器架构
定义理解	物理服务器 + 操作系统 + 环境配置 + 程序代码 = 应用	物理服务器 × [操作系统 + 环境配置 + 程序代码]（虚拟机封装） = 应用 应用 应用	(物理服务器 + 操作系统) × [环境配置 + 程序代码]（容器封装） = 应用 应用 应用 应用 应用
	独栋别墅 独立地基、独立大门 一栋楼一户人家	公寓大楼 共享地基、共享大门 一栋楼多套房、一套房一户人家 独立卫生间、独立厨房、独立宽带	胶囊旅馆 共享地基、共享大门 一套房多个隔间、一个隔间一位租户 共享卫生间、共享厨房、共享宽带
发展与改进		☐ 实现不同操作系统互通 ☐ 方便底层资源上云迁移 ☐ 按需给应用分配资源 ☐ 虚拟化出小的、独立的、随需随用的内核 ☐ 应用之间相互隔离方便管理 ☐ 虚拟机与物理机操作一致	☐ 比虚拟机模式以更快、更少资源的方式发布软件 ☐ 对底层资源占用更少 ☐ 应用服务通过更细的粒度进行分配和控制 ☐ 封装上仅包含程序代码与必要环境文件，移植性高
缺点与不足	☐ 只允许有一个操作系统 ☐ 操作系统之间不互通 ☐ 扩容周期长、成本高 ☐ 难以实现资源共享	☐ 封装包含操作系统，单个虚拟机占用底层资源大 ☐ 虚拟机启动等待时间长 ☐ 封装内容笨重，移植性能低	☐ 隔离性不彻底，安全性稍弱

来源：头豹研究院

IT基础架构演进历程

通用主机时代（1959年至今）
由执行集中处理的大型机组成，该集中处理机可以联网到数千个终端，最后使用联网的小型机进行一些分散和部门计算。

个人计算机时代（1981年至今）
以带有办公生产力工具的独立台式计算机的广泛使用为主导。

服务器时代（1983年至今）
由台式机或笔记本电脑客户端组成，这些客户端网络连接到处理大多数数据管理和处理的功能更强大的服务器计算机。

企业计算时代（1992年至今）
定义为大量个人计算机链接到局域网，并且越来越多地使用标准和软件将不同的网络和设备链接到企业范围的网络，以便信息可以在整个企业网络中自由流动。组织。

云计算时代（2000年至今）
一种计算模型，公司和个人可以通过Internet获得计算能力和软件应用程序，而不必购买自己的硬件和软件。

来源：头豹研究院

02

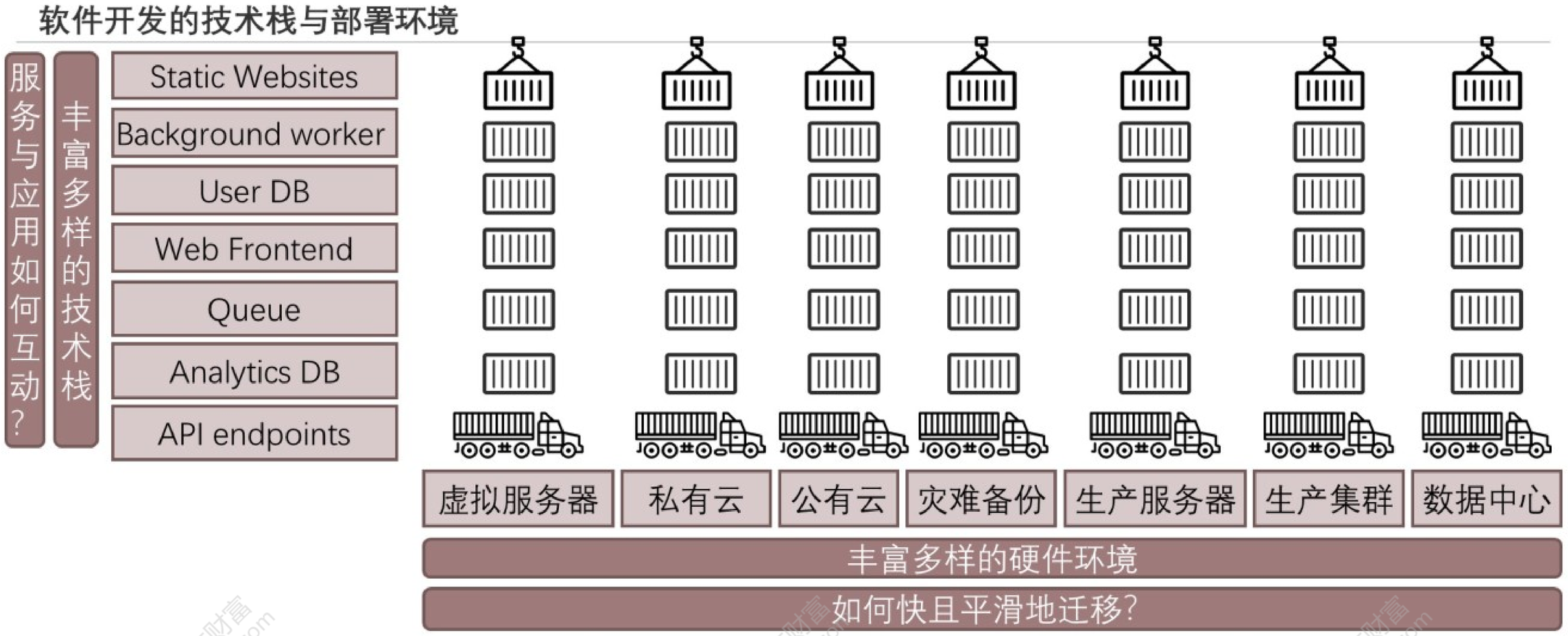
容器的技术概述

- 容器技术的必要性与优势
- 容器镜像架构
- 容器编排

容器技术的必要性与优势

- 容器技术使软件具备显著的可移植能力，满足开发人员与运维人员高效应对不同的技术栈与部署环境的需求

软件开发的技术栈与部署环境



来源：Cloudman，头豹研究院

❑ 软件开发面临的挑战

- 开发人员通常使用多种服务（比如MQ、Cache、DB）构建和组装应用，应用包含多种服务，这些服务有自己所依赖的库和软件包；
- 而应用很可能会部署到不同的环境，比如虚拟服务器、私有云和公有云，服务在运行时可能需要动态迁移到不同的部署环境中。

开发人员在编写代码时需要考虑不同的运行环境，运维人员则需要为不同的服务和平台配置环境。所以当前挑战在于如何让每种服务能够在所有的部署环境中顺利运行？

集装箱思想与容器技术

特性	集装箱	容器技术
打包对象	几乎任何货物	任何软件及其依赖
硬件依赖	标准形状和接口允许集装箱被装卸到各种交通工具，运输过程无须打开	容器封装内容无须修改即可运行在几乎所有部署环境的平台上
隔离性	装载着不同货物的集装箱可以重叠起来一起运输	资源、网络、库相互隔离，互不影响
自动化	标准大小和接口使集装箱很容易自动装卸和移动	提供run、start、stop等标准化操作，适合自动化
高效性	无须开箱，可在各种交通工具间快速切换搬运	轻量级，可快速启动和迁移
职责分工	货主只考虑放什么进集装箱；承运方只关心怎么运输集装箱	开发人员只考虑怎么写代码；运维人员只关心如何配置环境

来源：Cloudman，头豹研究院

❑ 容器技术的优势 1：可移植性

- 对于开发人员：Build Once、Run Anywhere。

容器意味着环境隔离和可重复性。开发人员只需为应用创建一次运行环境，然后打包成容器便可在其他机器上运行。另外，容器环境与所在的Host环境是隔离的，就像虚拟机一样，但更快更简单。

- 对于运维人员：Configure Once、Run Anything。

只需要配置好标准的runtime环境，服务器就可以运行任何容器。这使得运维人员的工作变得更高效、一致和可重复。容器消除了开发、测试、生产环境的不一致性。

❑ 容器技术的优势 2：敏捷

轻量级的打包方式使其具有更好的性能和更小的规模，创建容器实例比创建虚拟机示例快得多，启动和停止容器达到毫秒极响应。同时相比虚拟机，容器能更方便地与主机共享数据。

容器适配提升企业IT架构敏捷性，快速迭代产品，以更低的成本进行业务开发试错，帮助企业把握业务快速增长的机遇。

❑ 容器技术的优势 3：弹性

由于容器单元间相互独立，由统一的编排工具管理，且编排工具具备发现容器节点的功能，所以容器的弹性扩容可以在短时间内自动完成；同时，由于每个容器均为独立的个体，容器调用的资源和容器的使用由编排工具管理，所以减少某一容器节点不影响整个容器系统的使用。

借助容器技术，企业可以通过部署容器，充分发挥云计算的弹性优势，降低运维成本的同时实现快速扩容的需求。

容器对比虚拟机

	虚拟机	容器
占用磁盘空间	GB级	轻量级打包，MB级甚至KB级
启动速度	分钟级	秒级/毫秒级
运行形态	运行于Hypervisor	运行在宿主机内核上
并发性	一台宿主机上最多几十个	上百个，甚至上千个
性能	逊于宿主机	接近宿主机本地进程
资源利用率	低	高
更新管理	推送安装补丁升级	快速迭代式
隔离和安全性	应用独立内核，完全隔离	共享宿主机内核，进程级隔离
管理平台成熟度	以OpenStack、vCenter等为代表，体系成熟	以k8s为代表，还在快速发展过程中

来源：头豹研究院

容器镜像架构

- 凭借镜像打包技术的轻量化和只读特性，容器实现快速更新迭代、提高资源利用率、快速复制弹性扩容

❑ **镜像（Image）-- 单元**

镜像是容器运行时的只读模板，每一个镜像由一系列的层 (layers) 组成。用文件系统FS来将这些层联合到单独的镜像中，允许独立文件系统中的文件和文件夹(称之为分支)被透明覆盖，形成一个单独连贯的文件系统。

当改变一个镜像，比如升级到某个程序到新的版本，一个新的层会被创建。因此，不用替换整个原先的镜像或者重新建立，只是一个新的层被添加或升级了。现在你不用重新发布整个镜像，只需要升级，层使得分发镜像变得简单和快速。

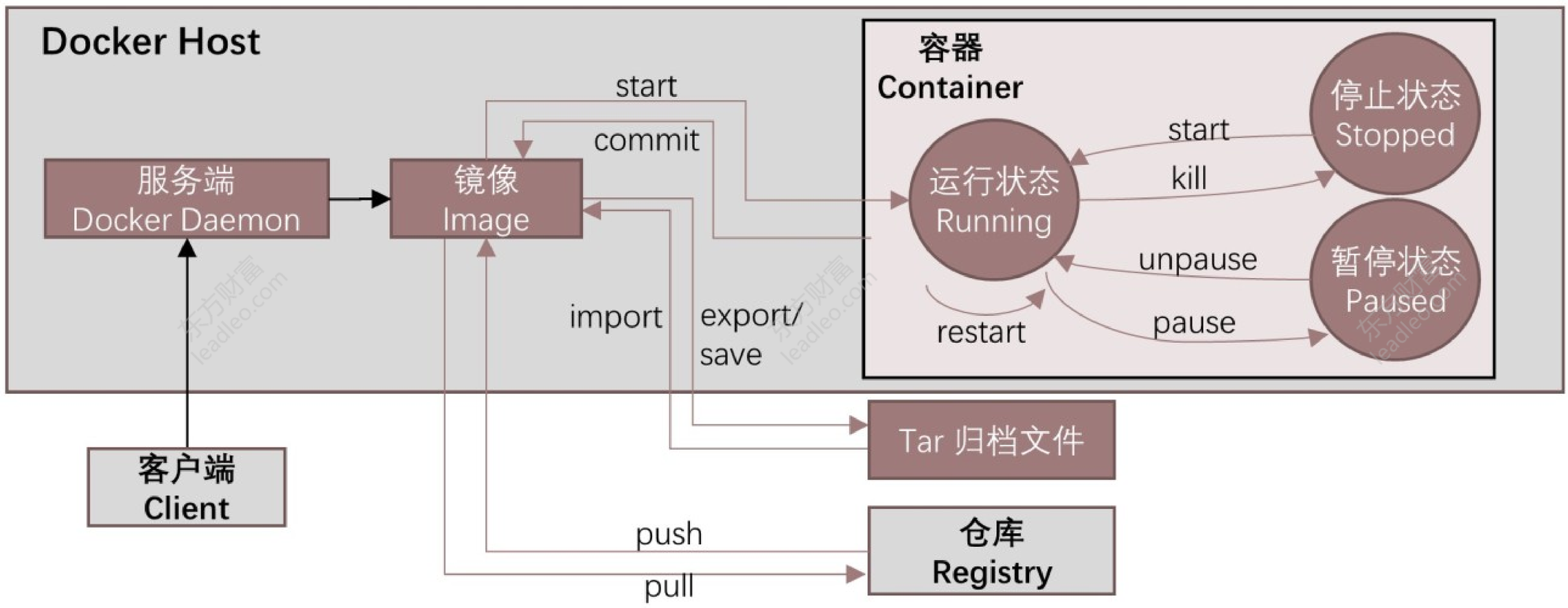
❑ **仓库 （Registry）-- 分发**

仓库用来保存镜像，可以理解为代码控制中的代码仓库。公有的仓库以Docker Hub为代表。Docker Hub 提供了庞大的镜像集合供使用。这些镜像可以是自己创建，或者在别人的镜像基础上创建。

❑ **容器 （Container）-- 运行**

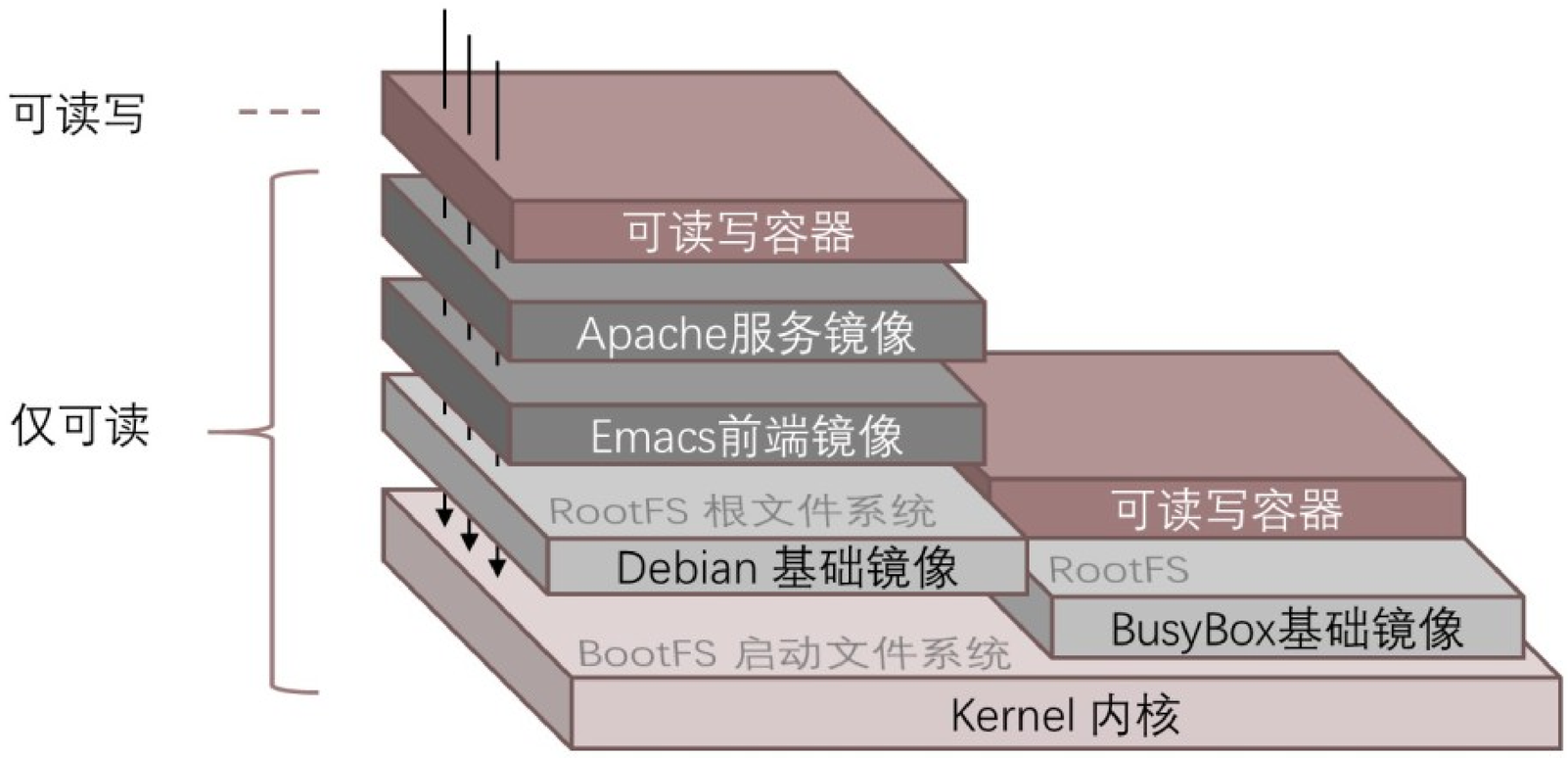
容器和文件夹很类似，一个容器包含了所有的某个应用运行所需要的环境。每一个容器都是从镜像创建的，不同的是容器带有额外的可写层。容器可以运行、开始、停止、移动和删除等操作。

Docker引擎与运行关系



来源：Docker, 头豹研究院

容器基础架构



来源：Docker, 头豹研究院

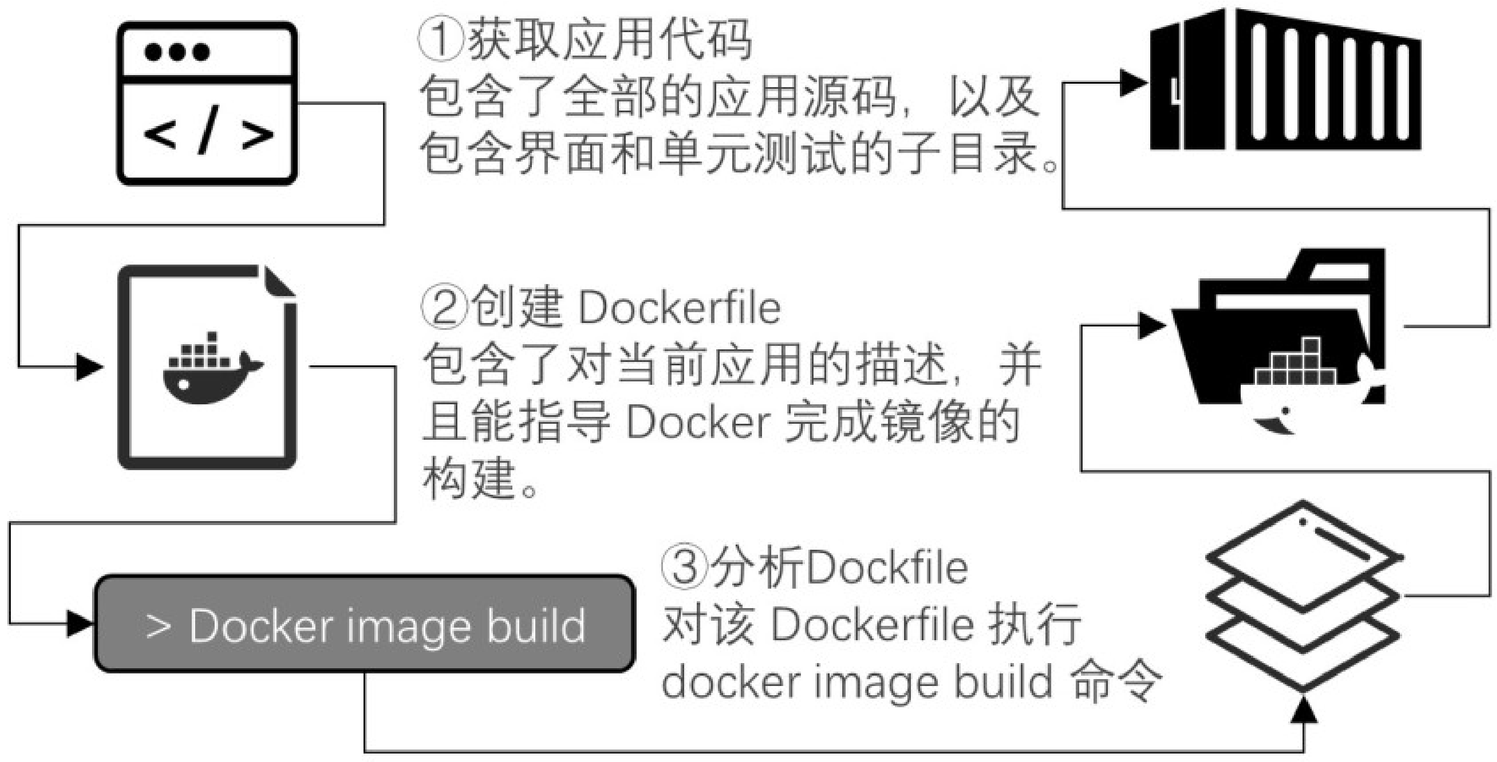
❑ **容器基础架构组成**

- **底层**为BootFS文件系统用于接入宿主机的服务器操作系统；
- **中层**镜像层，镜像层在程序运行过程中仅可读，主要包含上层程序的代码和运行程序所需的系统环境；
- **上层**为容器，容器是可改写的，镜像中代码的运行与结果的产生都发生在容器中，容器之间彼此独立。

❑ **容器文件读写处理简述**

- **读文件**
  1. 文件不在容器层，直接从镜像层读取；
  2. 文件只在容器层，直接从容器层读取；
  3. 文件在两层都有，读取容器层的文件，隐藏镜像层同名文件。
- **写文件**
  1. 首次写已有文件，文件不在容器层，需要镜像复制到容器层，再在容器层修改；
  2. 非首次修改已有文件，文件已在容器层，直接修改；
  3. 写新文件，直接在容器层写入。
- **其他**
  1. 如果容器被删除了，那么最上层的读写层也就删除了，改动也会随即丢失。镜像层的文件是只读的，并不会修改镜像的源文件。得以被上层复用，提高了资源利用率。
  2. 若要持久化改动，可以将容器保存为新的镜像。

应用容器化步骤



来源：Docker，头豹研究院

❑ 应用容器化

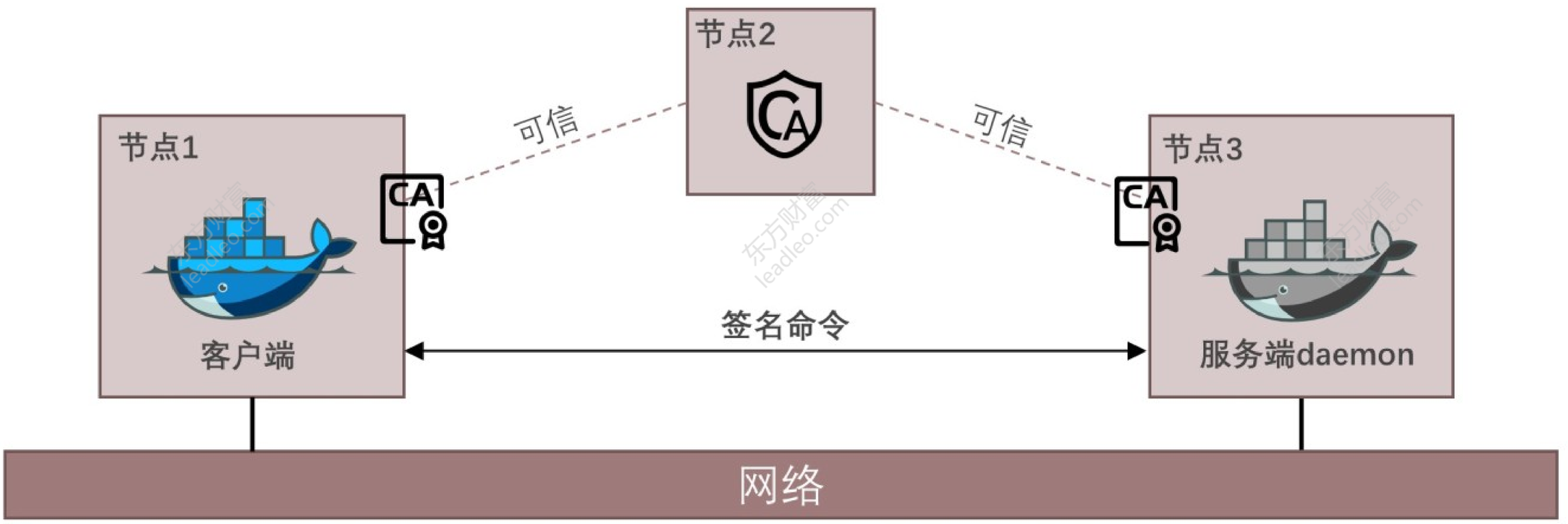
将应用整合到容器中并且运行起来的这个过程，称为“容器化”（Containerizing），有时也叫作“Docker化”（Dockerizing）。容器是为应用而生的，具体来说，容器能够简化应用的构建、部署和运行过程。

❑ 客户端与服务端

Docker 使用客户端-服务端模型。客户端使用 CLI，同时服务端（daemon）实现功能，并对外提供 REST API。Docker 为客户端与 daemon 间使用基于 TLS 的安全通信提供了两种模式。

- daemon 模式：Docker daemon 只接收认证客户端的链接。
- 客户端模式：Docker 客户端只接收拥有证书的 Docker daemon 发起的链接，其中证书需要由可信 CA 签发。

客户端与服务端通信模型

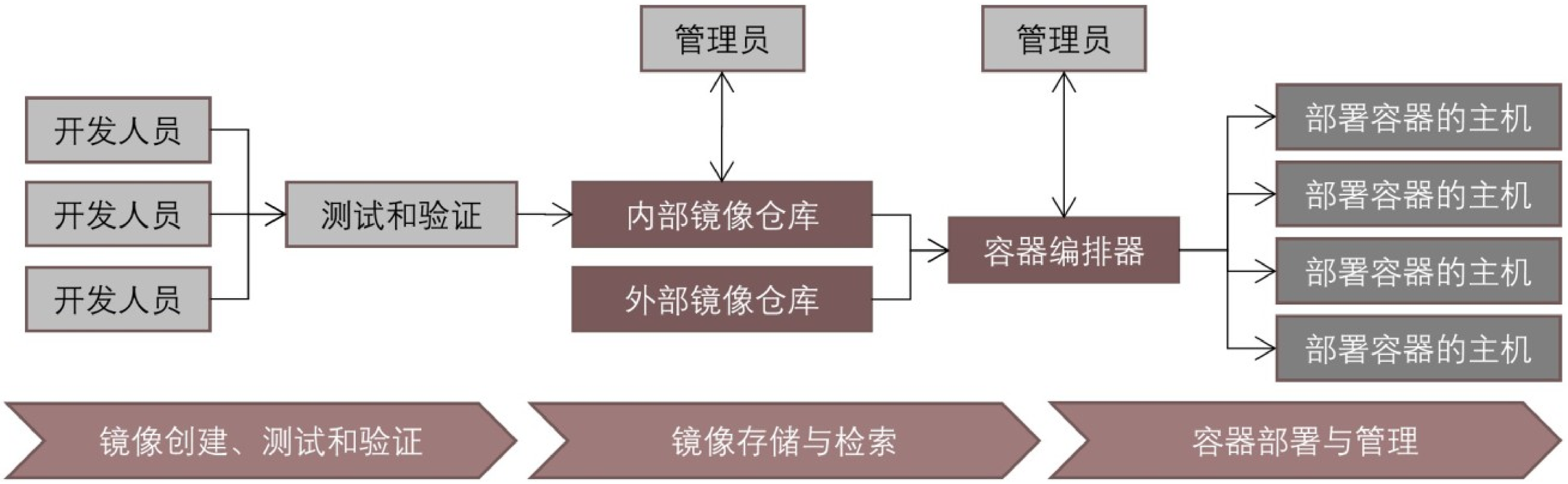


来源：Docker，头豹研究院

容器编排

- 在企业级应用中，大量的容器共同参与导致的运行复杂性与故障催生了容器编排管理工具的需求，Kubernetes应运而生

容器技术架构的生命周期



来源：青藤云安全，头豹研究院

容器部署与管理

编排工具让开发运维人员或自动化工具，能够从镜像仓库中获取镜像，将这些镜像部署到容器中，并管理正在运行的容器。

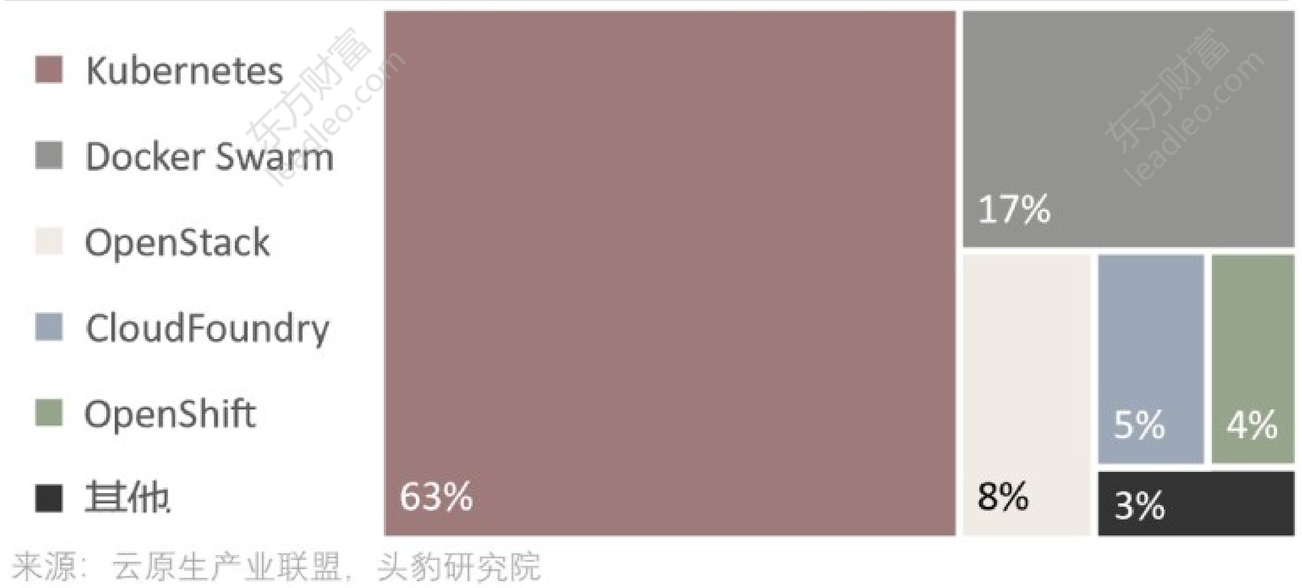
编排工具提供的抽象性让开发运维人员可以便捷地设定某个镜像运行所需的容器数量、以及每个容器需要分配的资源，如内存、处理能力和磁盘。

编排工具还负责监控跨主机的容器资源消耗、作业执行和机器健康状况。

Kubernetes（K8s）

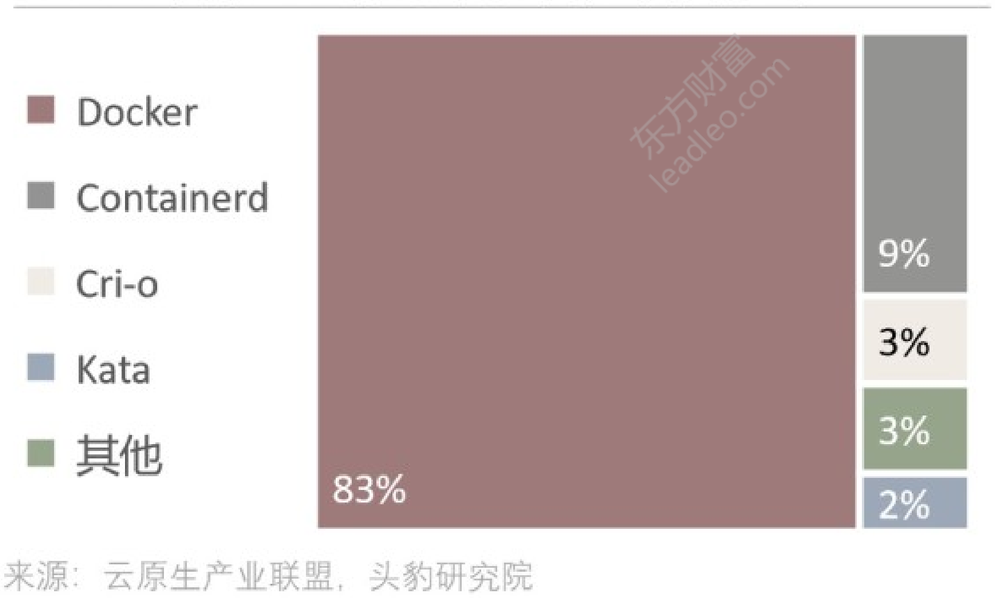
Kubernetes简称K8S 是 Google于2014年开源容器编排调度管理平台。相比与Swarm、Mesos，K8S引入Pod、Replica、Label、Service等机制简化了容器调度与管理，提供可靠性，增加了功能特性。与Docker情况类似，K8s目前成为最流行的容器编排平台，已成为容器编排的事实标准。

2020年容器编排中国用户使用分布



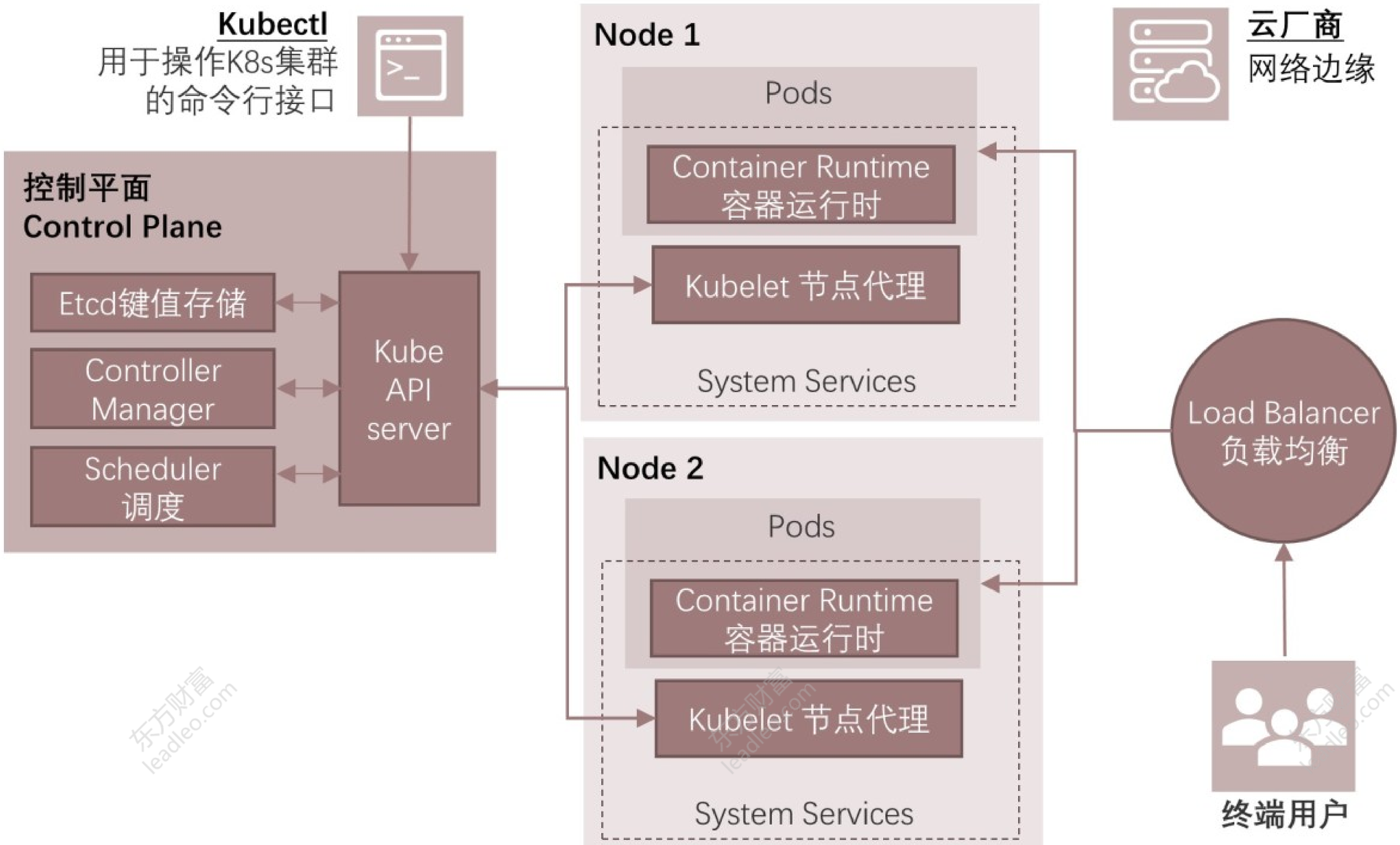
来源：云原生产业联盟，头豹研究院

2020年容器运行时中国用户使用分布



来源：云原生产业联盟，头豹研究院

Kubernetes原理



来源：Kubernetes、阿里云，头豹研究院

❑ K8s核心能力：分布式应用管理

- 资源调度：根据应用请求的资源量，在集群中选择适合的节点运行；
- 应用部署与管理：支持应用的自动发布与应用的回滚以及相关的配置管理；自动化存储卷的编排，让存储卷与容器应用的生命周期关联；
- 自动修复：K8s监测集群中的宿主机，如若宿主机或操作系统出现故障，节点健康检查会自动迁移应用，并支持应用的自愈；
- 服务发现与负载均衡：通过Service资源的各种应用服务，结合DNS喝多中负载均衡机制，支持容器化应用之间的通信；
- 弹性伸缩：K8s监测业务上的负载，如果业务本身的资源利用率过高，或者响应时间过长，对该业务自动扩容。

❑ K8s关键设计理念

- 声明式API：明确API对象，无感调谐到期望的状态，使开发者关注应用本身而非系统执行细节；
- 可扩展性架构：所有K8s组件基于一致的、开放的API实现和交互；
- 可移植性：通过负载均衡服务、CNI容器网络接口、CSI容器存储接口，帮助业务应用屏蔽底层基础设施的实现差异，实现灵活迁移的目标。

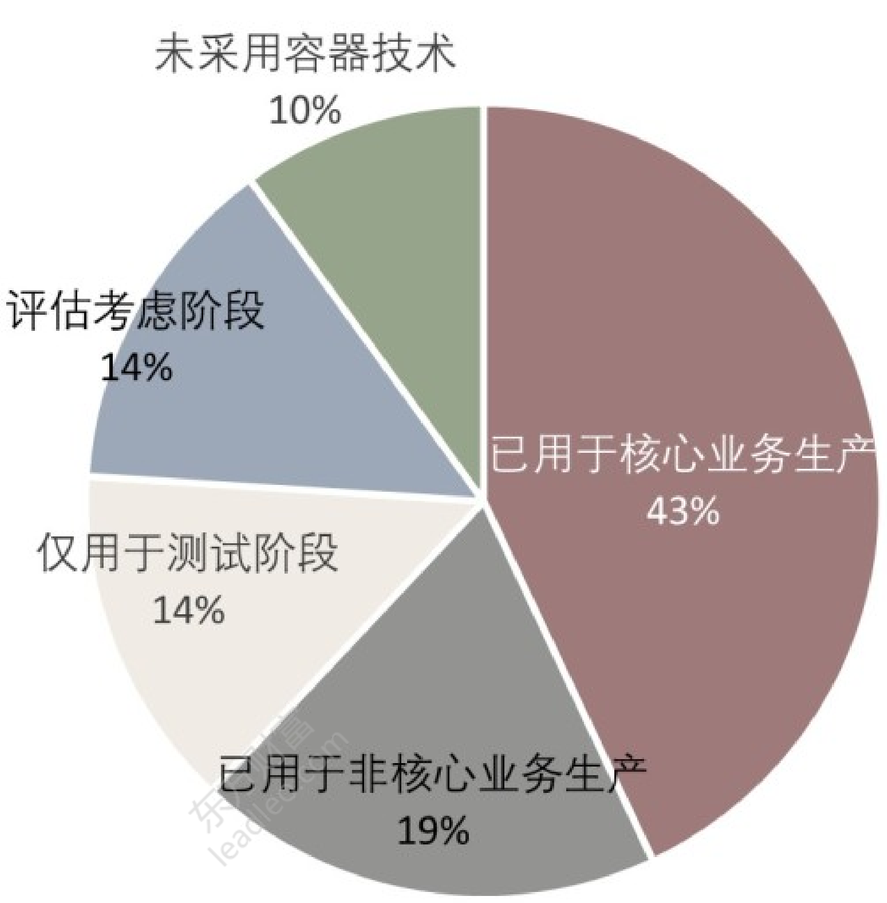
03

容器的发展现况及趋势

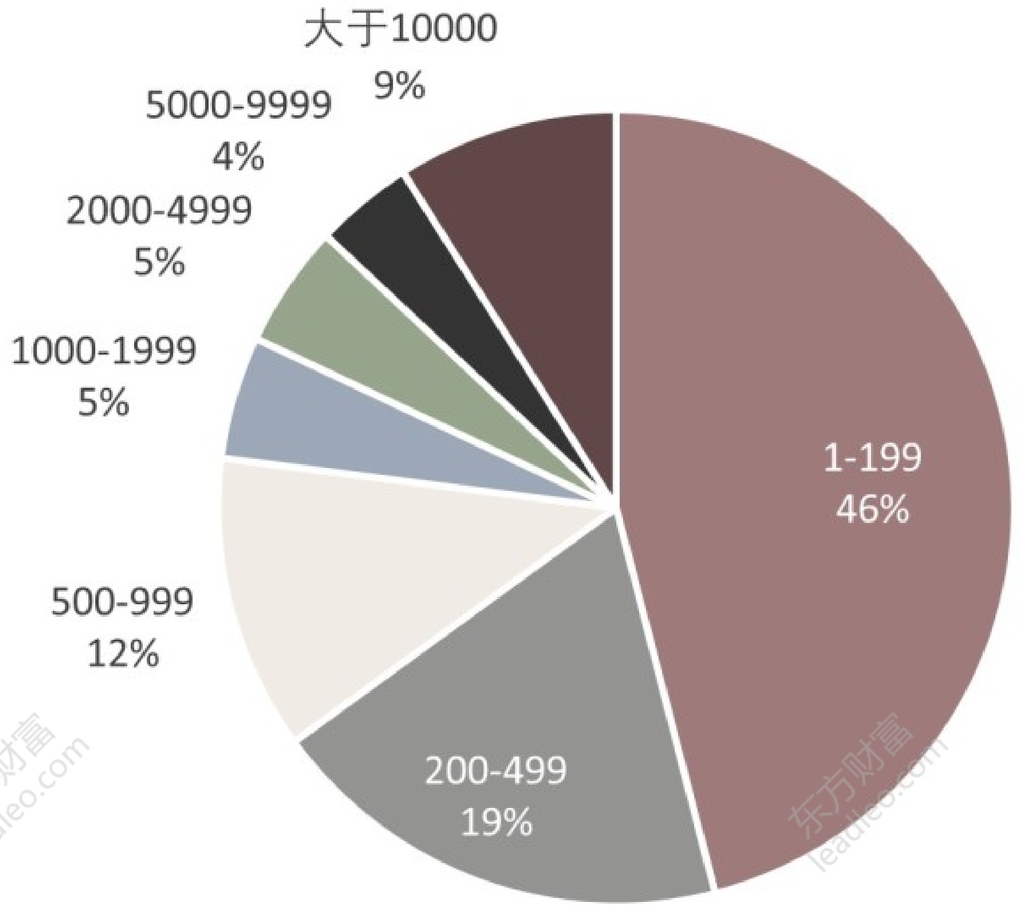
- 2020年容器技术的应用现况
- 容器市场的发展驱动因素分析
- 容器的发展趋势
- 容器相关厂商图谱

2020年容器技术的应用现况

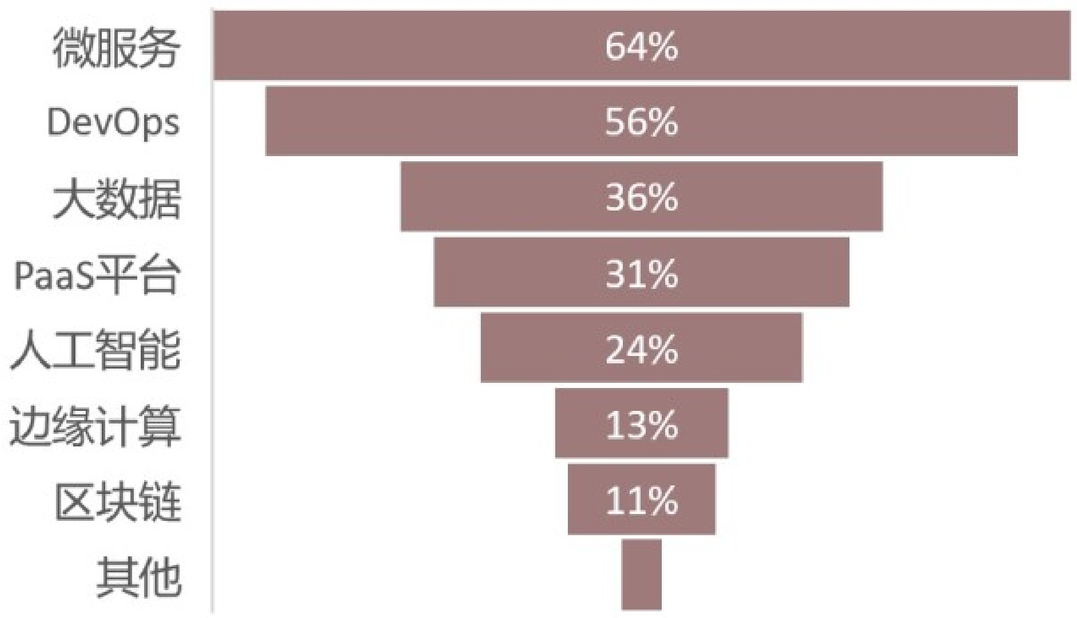
容器技术采纳现况



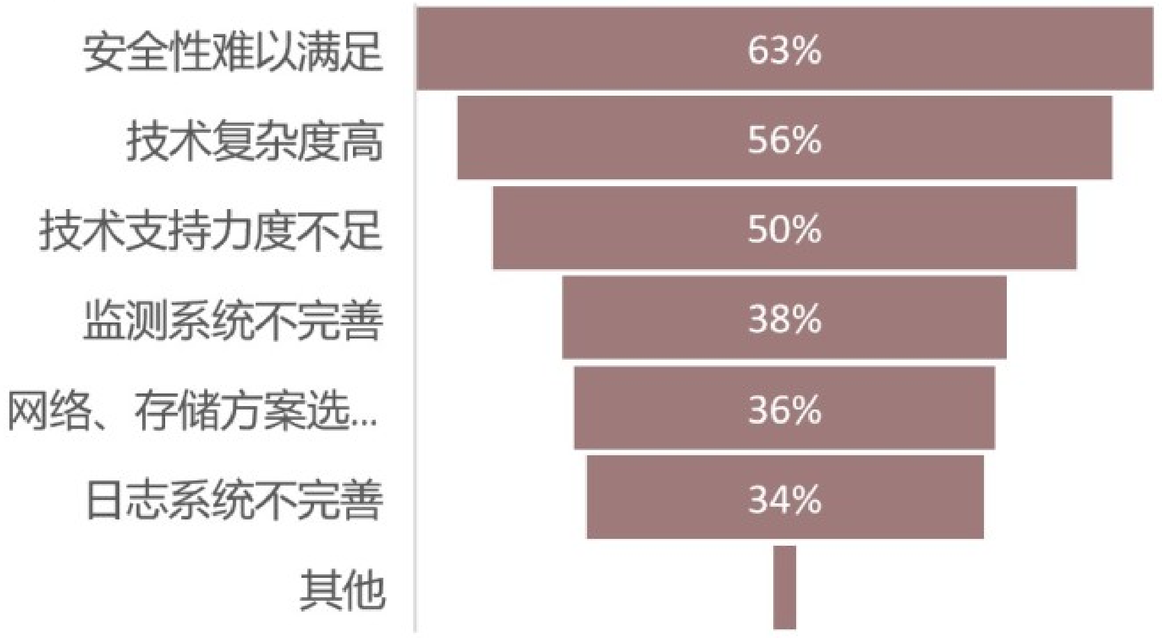
生产环境的容器集群规模



容器主要使用场景



容器技术使用中存在的问题



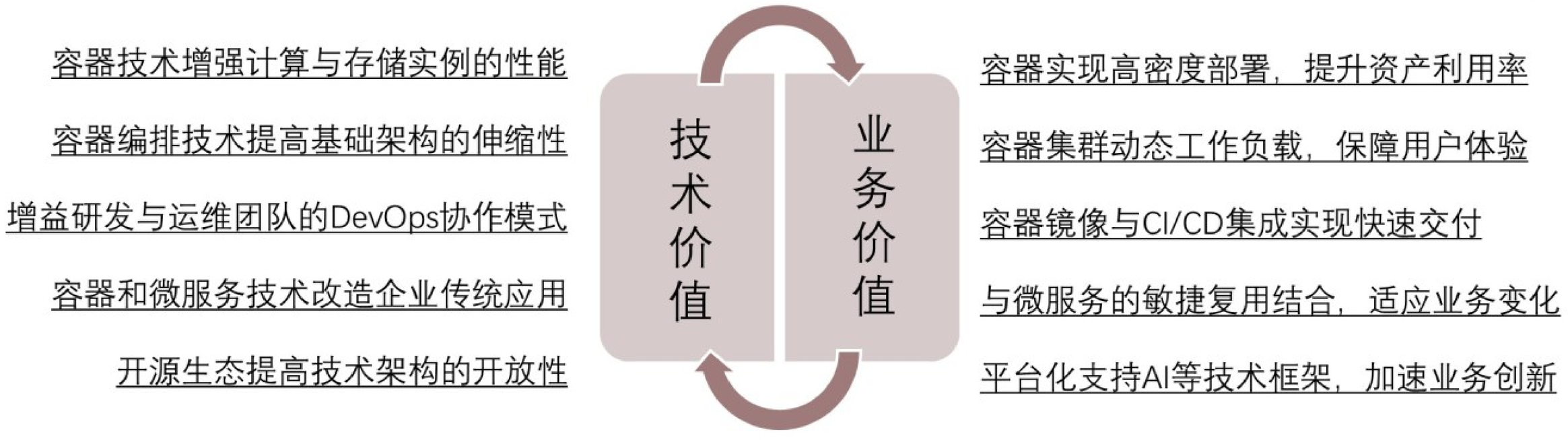
- 超过60%的云原生用户已经在生产环境中应用了容器技术，近30%的用户正在考虑甚至正在测试评估容器技术的应用；
- 不到1000节点的容器集群规模已经能够满足77%用户的生产需求；
- 微服务应用承载与DeveOps研发运维流程自动化是容器技术最为主要的应用场景；
- 容器技术的底层安全性是用户使用中最大的问题，技术复杂度与技术支持能力也是用户关注的重点。

来源：云原生产业联盟，头豹研究院

容器市场的发展驱动因素分析

- 容器凭借业务价值和技术价值满足了来自不同行业的企业的多层面需求，容器市场迎来创新与增长的快速发展

容器技术的业务价值与技术价值

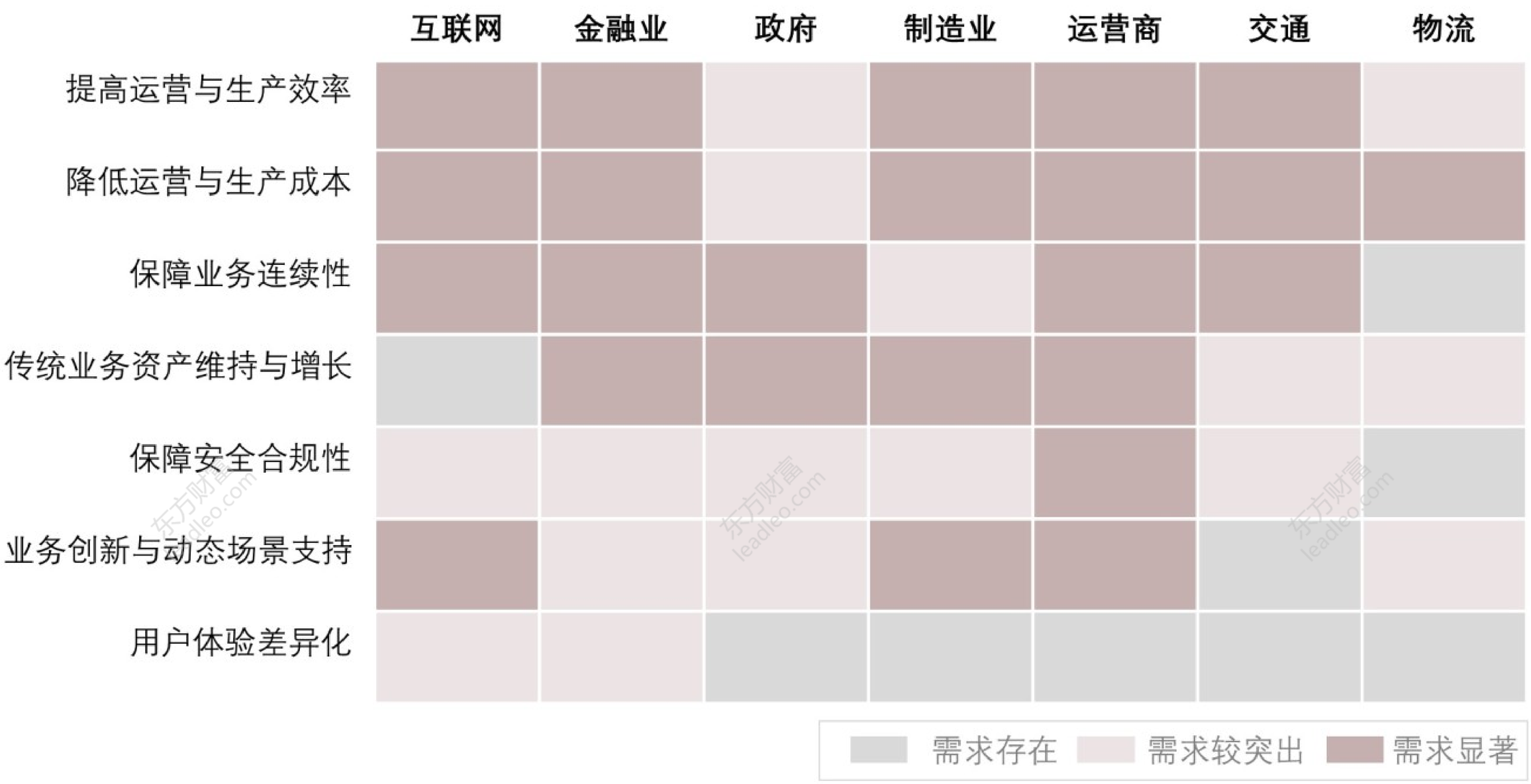


来源：头豹研究院

❑ 业务需求与技术需求共同驱动

容器行业的下游应用集中在互联网、金融、政府、制造业、运营商、交通和物流行业。企业提升企业数字化IT架构，借助容器、容器编排以及微服务、等云原生技术加速企业的应用研发、敏捷交付得以更好地服务市场及客户，面对当前企业的多层面需求。

云原生容器平台的行业需求对比



来源：头豹研究院

容器的发展趋势

- 容器在自身技术迭代发展的同时，与云原生架构的协同正进一步深化，与新兴技术架构融合升级并拓宽下游应用场景

容器的技术发展趋势

云端Serverless技术融入主流

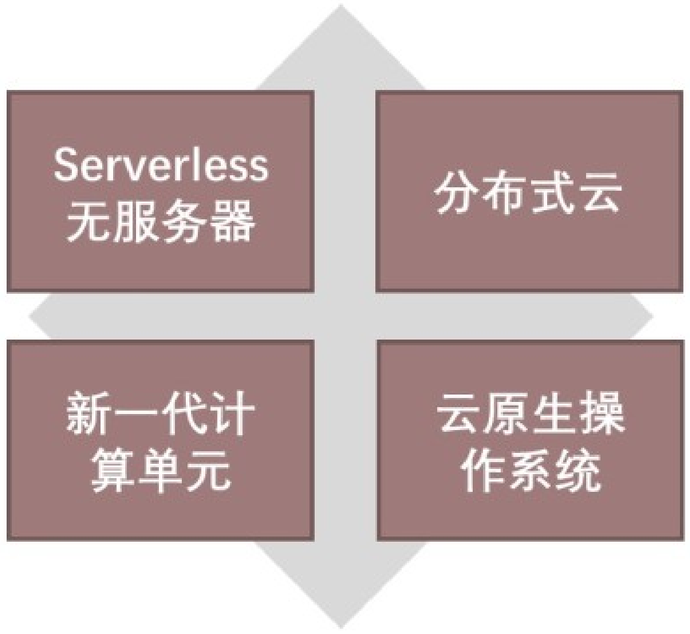
极致弹性、免运维、强安全、高效开发

- 与云场景深度融合，复杂性下沉，弹性
- 应用开发、交付范性的模式重新改写

无处不在的计算催生新一代容器

针对计算场景优化、安全、轻重、高效

- 基于MicroVM的安全容器
- 基于WebAssembly的可移植、轻量容器
- OS虚拟化创新，如cgroup v2提升隔离性



动态、混合、分布式的云环境新常态

统一技术栈、统一应用界面

- 公共云服务能力将延伸至边缘计算和IDC
- 云原生结构推进无边界云计算，促成云边端应用一体协调

云原生操作系统浮现

定义开放标准，封装资源，支撑应用

- 计算、存储、网络、安全的云原生进化
- 多种工作负载、海量计算任务，多种异构算力的高效调度和编排
- 标准化、自动化、可移植、安全可控的应用交付、管理体系

来源：阿里云，头豹研究院

容器的应用领域发展趋势

云原生架构协同深化	新兴技术架构融合升级	下游应用场景拓宽
微服务 容器的隔离性和无状态特性为微服务提供了运行环境	**混合云/多云管理平台** 容器化应用对环境的高度兼容性适应云管理平台	**企业云上数字化** 帮助企业将传统应用高效迁移上云，采用云计算降本增效
DevOps 容器的使用支撑了DevOps工作流、及时反馈、持续学习	**人工智能** 容器具备快速迭代发布、弹性伸缩支撑AI计算集群管理	**在线服务** 在疫情新常态的催化下，在线会议、远程办公、互联网医院、互动课堂、在线游戏、直播、视频等在线服务需求正在加速渗透经济生产和日常生活。容器作为底层架构满足在线服务的高并发、高弹性的计算需求。
无服务器架构 补充传统容器架构在偶发、非连续事件场景的应用需求	**边缘计算** 容器的轻量化、可移植性、快速部署特性契合边缘计算	

来源：头豹研究院

容器相关厂商图谱

IaaS

公有云：华为云（HUAWEI）、阿里云、腾讯云、亚马逊云科技、金山云、京东智联云

私有云：QINGCLOUD、易捷行云 EasyStack、锐捷 Ruijie Networks、紫光云 UNICLOUD.COM、浪潮 inspur

容器底层生态

容器引擎：docker、kata、cri-o、CLOUD FOUNDRY、openstack、iSulad、container d、Firecracker、Red Hat OpenShift、podman

容器编排：kubernetes、docker SWARM、RANCHER、minikube、AWS ECS、suse、VOLCANO Erupt with Computing Power、TKEStack

容器平台

综合云服务厂商：华为云（HUAWEI）、阿里云、百度智能云、亚马逊云科技、金山云、QINGCLOUD

云计算厂商：DaoCloud、博云 BoCloud、易捷行云 EasyStack、时速云、灵雀云 alauda.cn、UCLOUD 专业基础云计算服务商、才云 caicloud、谐云科技 HARMONYCLOUD、ghost cloud

传统IT服务厂商：中科曙光 Sugon、浪潮 inspur、网宿科技 ChinaNetCenter、SANGFOR 深信服科技

容器方案

容器存储：杉岩数据 SANDSTONE、焱融科技 YAN RONG TECH、portworx

容器安全：HYPER.SH、奇安信、gVisor

来源：GigaOm、各公司官网，头豹研究院

名词解释

- **云原生：**区别于从本地环境移植到云上的大部分程序，云原生强调最初的开发就是为了最终部署到云环境上。在公有云、私有云和混合云等新型动态环境中，赋能组织或企业去构建和部署可弹性扩展的应用。
- **持续交付：**(Continuous delivery, CD)，是一种软件工程手法，让软件产品的产出过程在一个短周期内完成，以保证软件可以稳定、持续的保持在随时可以发布的状况
- **微服务架构：**微服务架构 = 80% 的 SOA 服务架构思想 + 100% 的组件化架构思想 + 80% 的领域建模思想，系统中的各个微服务可被独立部署，各个微服务之间是松耦合的。每个微服务仅关注于完成一件任务并很好地完成该任务。
- **DevOps：**Development和Operations的组合词，是一组过程、方法与系统的统称，用于促进开发（应用程序/软件工程）、技术运营和质量保障（QA）部门之间的沟通、协作与整合。
- **容器技术：**有效的将单个操作系统的资源划分到孤立的组中，以便更好的在孤立的组之间平衡有冲突的资源使用需求，这种技术就是容器技术。
- **虚拟化技术：**Virtualization，是一种资源管理技术，是将计算机的各种实体资源，如服务器、网络、内存及存储等，予以抽象、转换后呈现出来。
- **面向对象：**Object Oriented，是软件开发方法，一种编程范式。面向对象是相对于面向过程来讲的，面向对象方法，把相关的数据和方法组织为一个整体来看待，从更高的层次来进行系统建模，更贴近事物的自然运行模式。
- **寄居架构：** Hosted Architecture，寄居架构将虚拟化层架在操作系统之上，当作一个应用运行。依赖于主机操作系统对设备的支持和物理资源的管理。
- **裸金属架构：** Bare Metal Architecture，裸金属架构将虚拟化层直接运行在x86硬件系统上，在其之上安装操作系统和应用。
- **镜像：**Image，镜像是容器运行时的只读模板，每一个镜像由一系列的层 (layers) 组成。
- **镜像仓库：**Registry，仓库用来保存镜像，可以理解为代码控制中的代码仓库。
- **容器编排：**让开发运维人员或自动化工具，能够从镜像仓库中获取镜像，将这些镜像部署到容器中，并管理正在运行的容器。
- **Kubernetes：**K8s，Google于2014年开源容器编排调度管理平台。相比与Swarm、Mesos，K8S引入Pod、Replica、Label、Service等机制简化了容器调度与管理，提供可靠性，增加了功能特性。
- **Serverless无服务器架构：**在无需管理服务器等底层资源的情况下完成应用的开发和运行，是云原生架构的核心组成部分。

方法论

- 头豹研究院布局中国市场，深入研究10大行业，54个垂直行业的市场变化，已经积累了近50万行业研究样本，完成近10,000多个独立的研究咨询项目。
- 研究院依托中国活跃的经济环境，从社会保险、人工智能、大数据等领域着手，研究内容覆盖整个行业的发展周期，伴随着行业中企业的创立，发展，扩张，到企业走向上市及上市后的成熟期，研究院的各行业研究员探索和评估行业中多变的产业模式，企业的商业模式和运营模式，以专业的视野解读行业的沿革。
- 研究院融合传统与新型的研究方法，采用自主研发的算法，结合行业交叉的大数据，以多元化的调研方法，挖掘定量数据背后的逻辑，分析定性内容背后的观点，客观和真实地阐述行业的现状，前瞻性地预测行业未来的发展趋势，在研究院的每一份研究报告中，完整地呈现行业的过去，现在和未来。
- 研究院密切关注行业发展最新动向，报告内容及数据会随着行业发展、技术革新、竞争格局变化、政策法规颁布、市场调研深入，保持不断更新与优化。
- 研究院秉承匠心研究，砥砺前行的宗旨，从战略的角度分析行业，从执行的层面阅读行业，为每一个行业的报告阅读者提供值得品鉴的研究报告。

法律声明

头豹研究院简介

- 头豹是中国领先的原创行企研究内容平台和新型企业服务提供商。围绕“协助企业加速资本价值的挖掘、提升、传播”这一核心目标，头豹打造了一系列产品及解决方案，包括：**报告/数据库服务**、**行企研报服务**、**微估值及微尽调自动化产品**、**财务顾问服务**、**PR及IR服务**，以及其他企业为基础，利用大数据、区块链和人工智能等技术，围绕产业焦点、热点问题，基于丰富案例和海量数据，通过开放合作的增长咨询服务等
- 头豹致力于以优质商业资源共享研究平台，汇集各界智慧，推动产业健康、有序、可持续发展

四大核心服务

研究咨询服务

为企业提供**定制化报告**服务、**管理咨询**、**战略**调整等服务

企业价值增长服务

为处于不同发展阶段的企业，提供与之推广需求相对应的“**内容+渠道投放**”一站式服务

行业排名、展会宣传

行业峰会策划、**奖项**评选、行业**白皮书**等服务

园区规划、产业规划

地方**产业规划**，**园区**企业孵化服务

报告阅读渠道

头豹官网 —— www.leadleo.com 阅读更多报告

头豹小程序 —— 微信小程序搜索“头豹”、手机扫上方二维码阅读研报

添加右侧头豹分析师微信，身份认证后邀您进入行研报告分享交流微信群

详情咨询

客服电话

400-072-5588

上海

王先生： 13611634866

李女士： 13061967127

深圳

李先生： 18916233114

李女士： 18049912451

南京

杨先生： 13120628075

唐先生： 18014813521

头豹 Project Navigator 领航者计划介绍

每个季度，头豹将于网站、公众号、各自媒体公开发布**季度招募令**，每季公开

125个

招募名额

头豹诚邀**政府及园区、金融及投资机构、顶流财经媒体及大V**推荐共建企业

头豹共建报告

2021年度特别策划

Project Navigator

领航者计划

头豹诚邀各行业**创造者、颠覆者**

领航者

知识共享、内容共建

头豹邀请**沙利文**担任计划**首席增长咨询官**、**江苏中科院智能院**担任计划**首席科创辅导官**、**财联社**担任计划**首席媒体助力官**、**无锋科技**担任计划**首席新媒体造势官**、**iDeals**担任计划**首席VDR技术支持官**、**友品荟**担任计划**首席生态合作官**

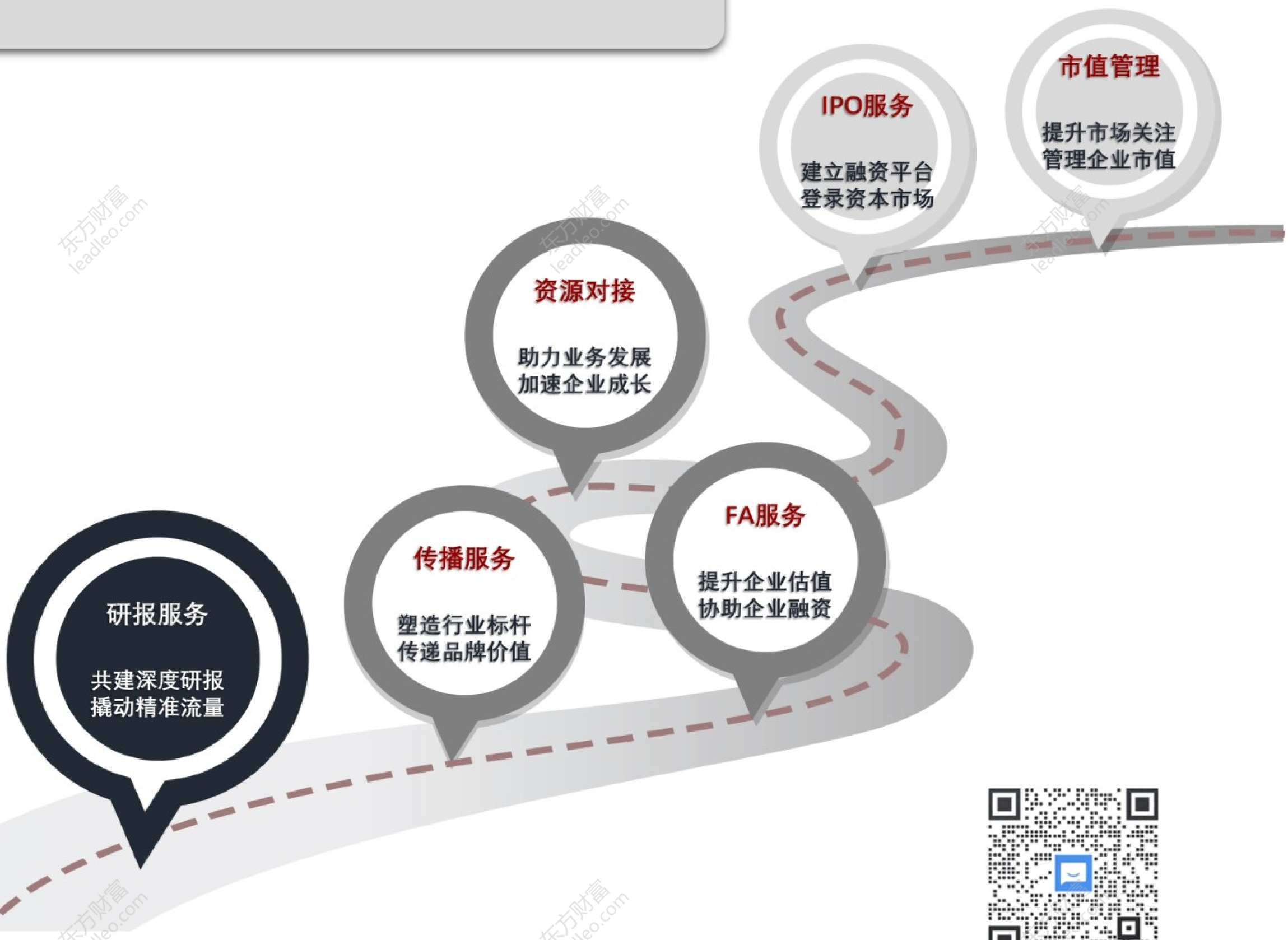
共建报告流程

企业申请共建 → 头豹审核资质 → 确定合作细项 → 报告发布投放 → 信息共享、内容共建

备注：活动解释权均归头豹所有，活动细则将根据实际情况作出调整。

头豹 Project Navigator 领航者计划与商业服务

- 头豹以**研报服务**为切入点，根据企业不同发展阶段的资本价值需求，以**传播服务**、**FA服务**、**资源对接**、**IPO服务**、**市值管理**为基础，提供适合的**商业管家服务解决方案**



扫描上方二维码

联系客服报名加入

备注：活动解释权均归头豹所有，活动细则将根据实际情况作出调整。

头豹 LeadLeo
www.leadleo.com
400-072-5588

读完报告有问题?

快，问头豹！你的智能随身专家

扫码二维码
即刻联系你的智能随身专家

千元预算的

高效率轻咨询服务

STEP01 智能拆解提问

人工智能NLP技术
精准拆解用户提问

STEP02 云研究院后援

云研究院7×24待命
随时评估解答方案

STEP03 解答方案生成

大数据×定制调研
迅速生成解答方案

STEP04 专业高效解答

书面反馈、分析师专访、专家专访等多元化反馈方式

头豹 LeadLeo
www.leadleo.com
400-072-5588