



# 第1讲 分布式系统简介

§1.1 分布式系统简介

§1.2 分布式系统分类











# §1.1 分布式系统简介

## 中国技术又得世界第一

10月2日，权威机构国际事务处理性能委员会（TPC）发布最新测试结果：阿里巴巴关联公司蚂蚁金服的数据库OceanBase创造了新的世界纪录！此前该世界纪录由美国公司甲骨文（Oracle）创造，并保持了9年。

### 数据库TPC-C基准测试 全球TOP10

注：单位为tpmC，即每分钟内系统处理的新订单个数

01		阿里巴巴	6千万
02		Oracle	3千万
03		IBM	1千万
04		Oracle	8百万
05		Oracle	7百万
06		IBM	6百万
07		BULL	6百万
08		Oracle	5百万
09		Oracle	4百万
10		Hewlett Packard	4百万

来源：国际事务处理性能委员会（TPC）官网  
注：所属国家以进行测试的软件为准  
阿里巴巴参与测试的是其关联公司蚂蚁金服的数据库OceanBase  
BULL参与测试的是IBM的数据库DB2

OceanBase打破了由Oracle保持了9年之久的TPC-C基准性能测试的世界纪录，“这是中国基础软件取得的重大突破”。

——中国工程院院士、计算机专家李国杰

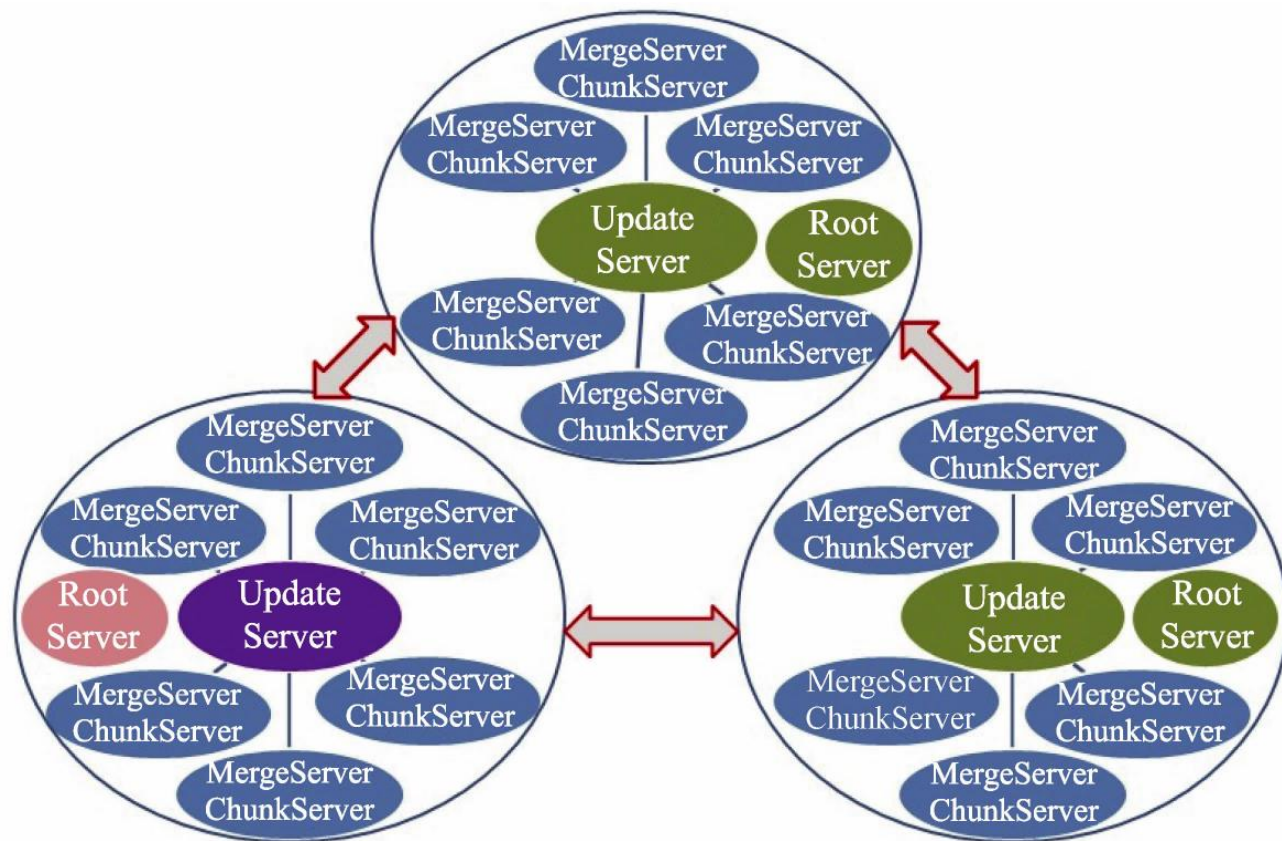
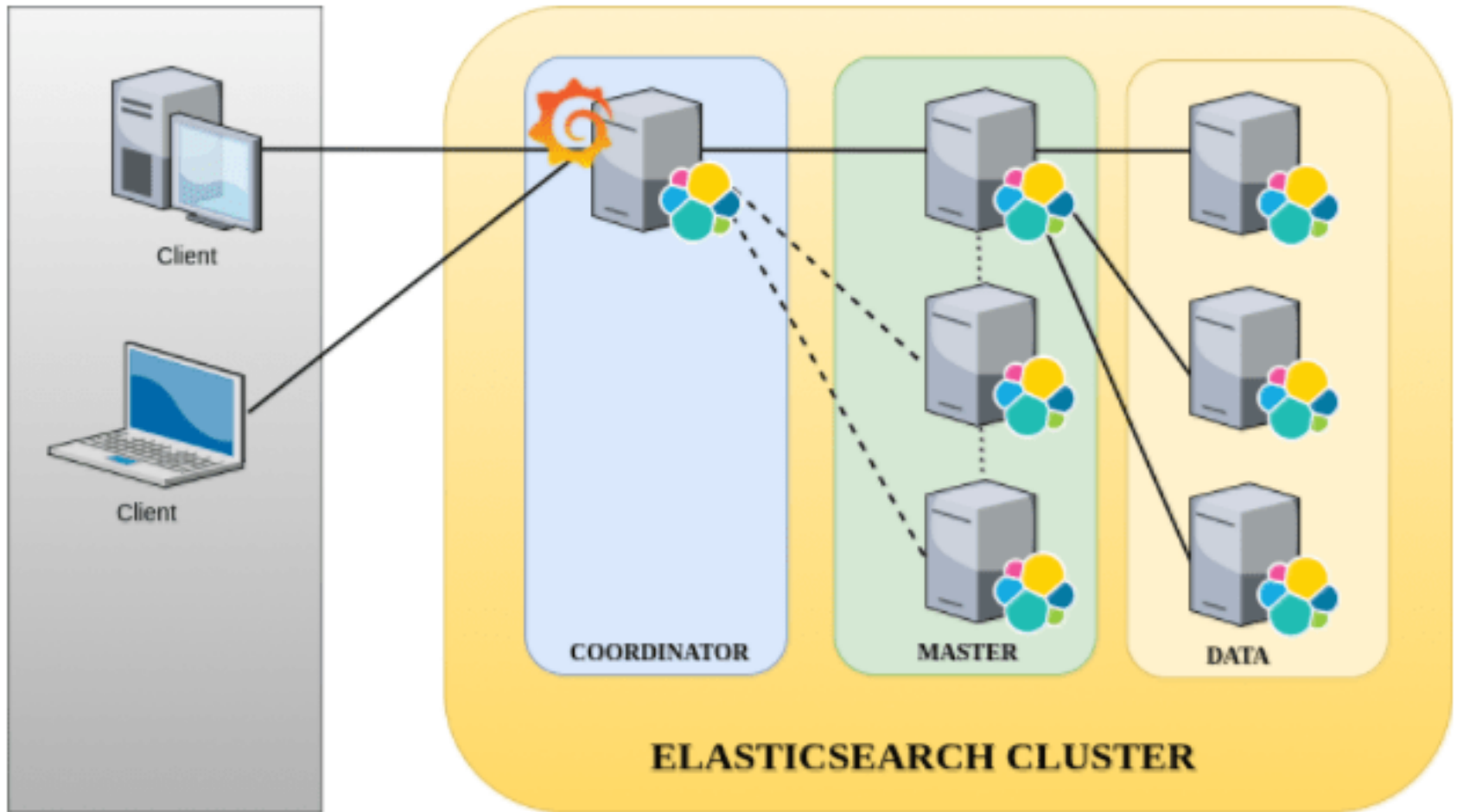


图 4 OceanBase 架构(三机群)

2019年

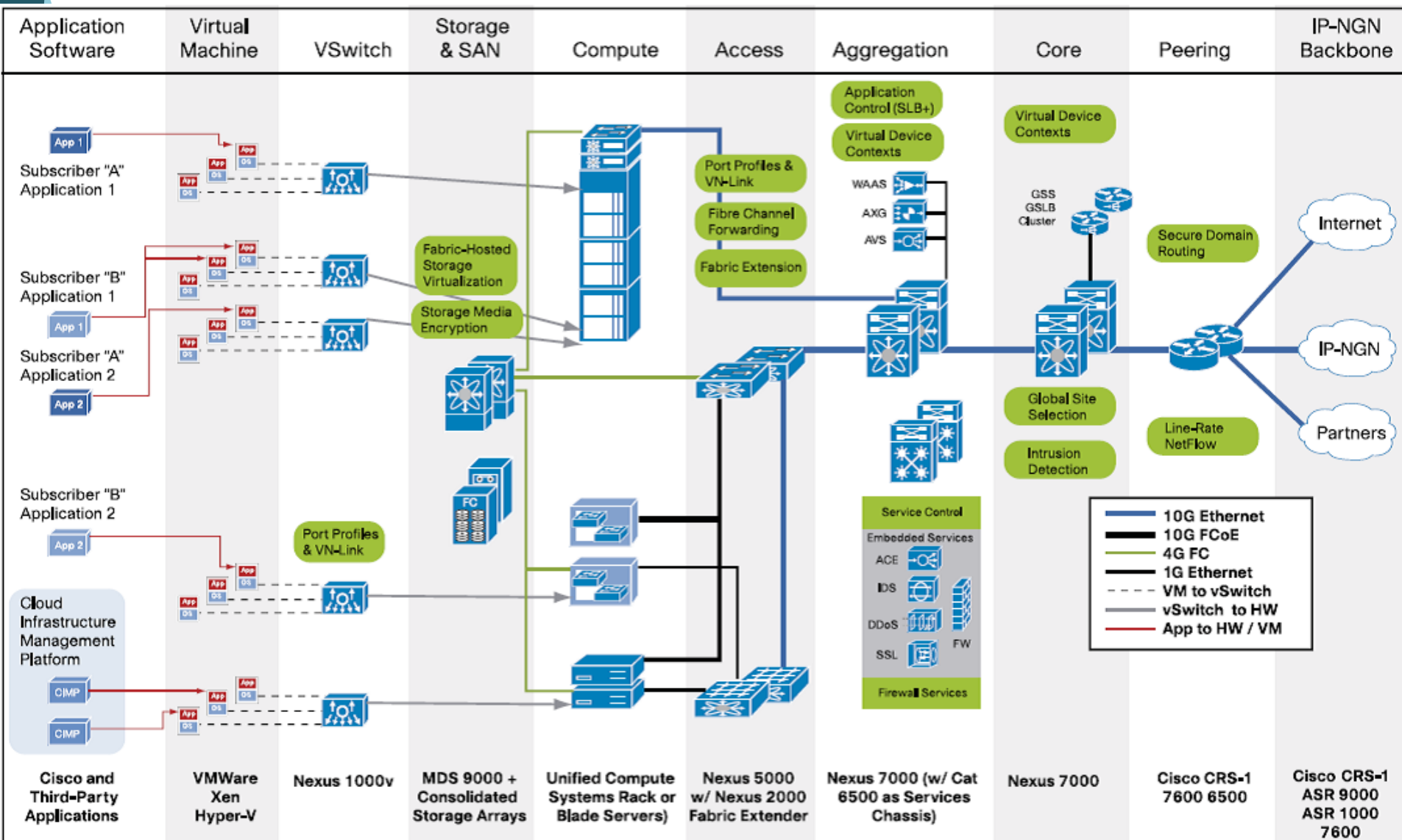
分布式数据库系统 -- 阿里巴巴OceanBase

# 什么是分布式系统？

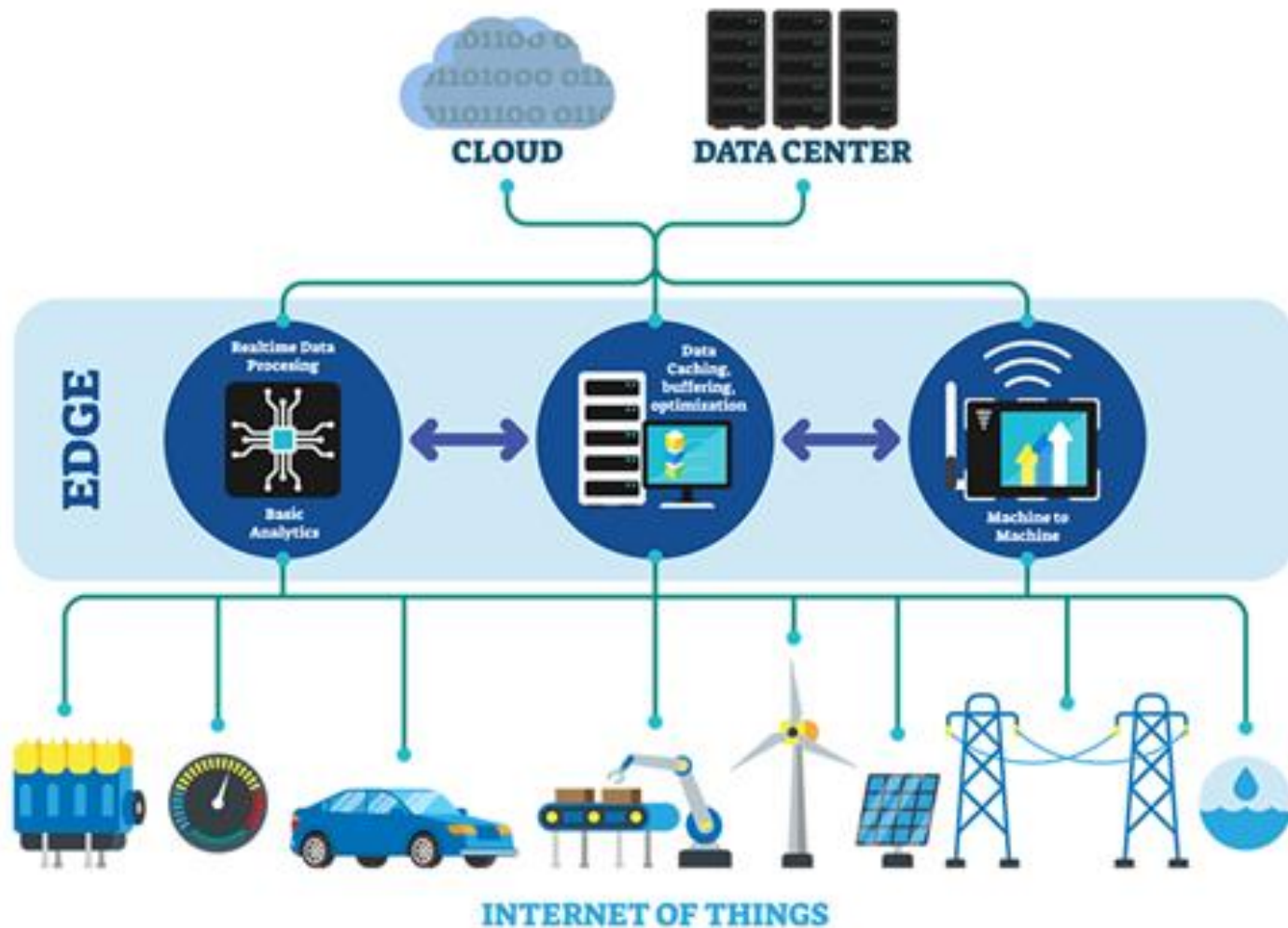


分布式、RESTful的搜索和数据分析引擎  
最受欢迎的企业搜索引擎

# 什么是分布式系统?



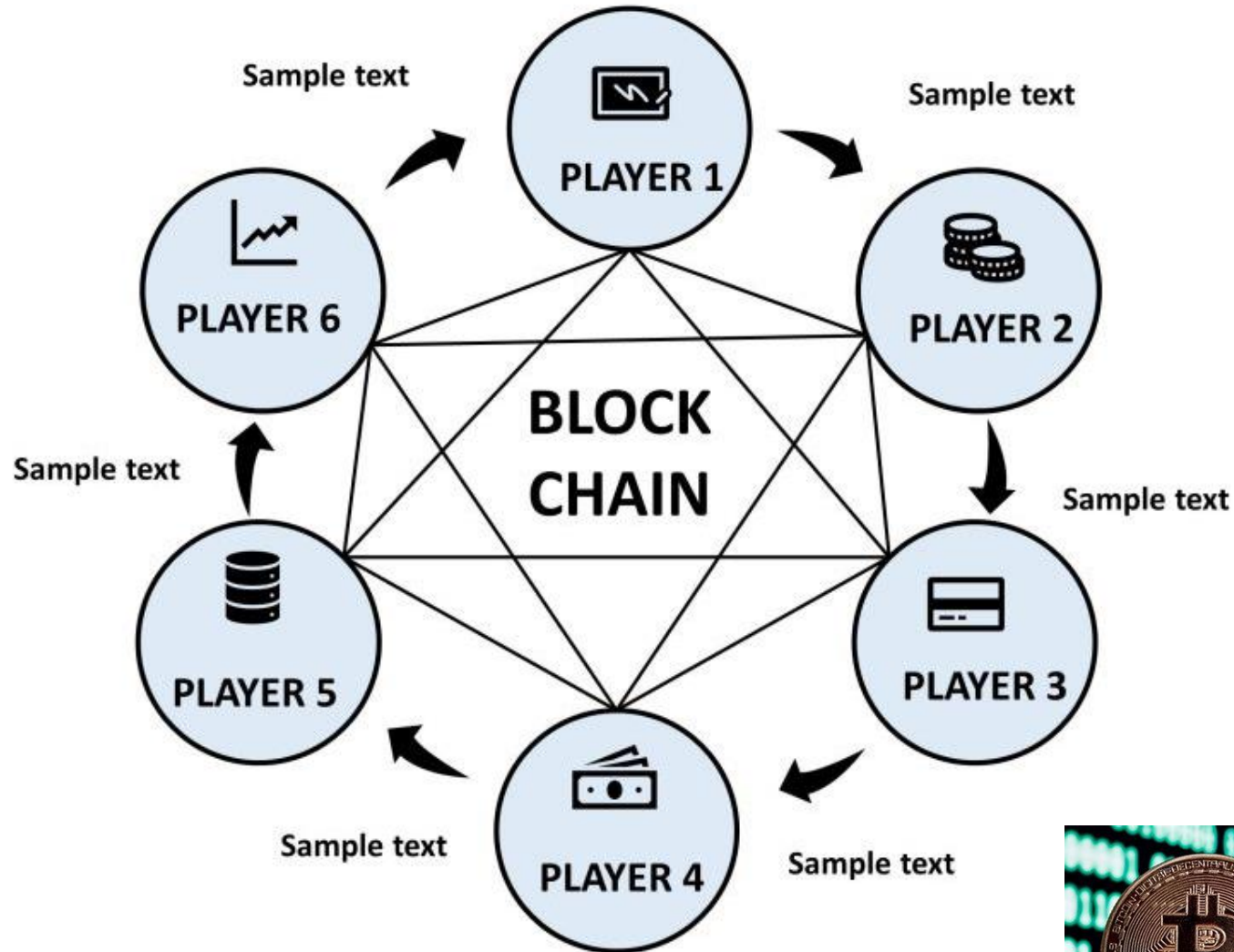
# 什么是分布式系统？



Cloud computing + Edge computing + IoT Systems



# 什么是分布式系统？



Bitcoin Blockchain Systems



# §1.1 分布式系统简介

- 分布式系统是技术和需求发展的结果

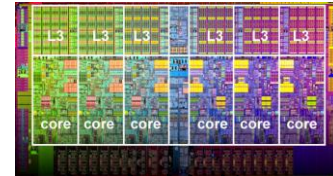
计算机的小型化以及计算性能的提升



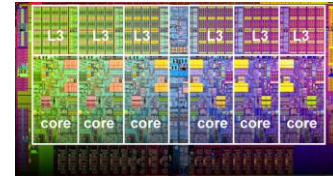
高性能主机



高性能服务器



多核高性能CPU



多核高性能CPU



小规模低速网络

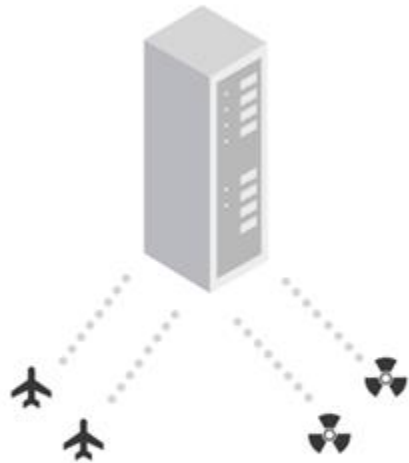


大规模高速网络

KB → MB → GB → 10GB → 100GB → ?

# 分布式系统的趋势

2000年之前



封闭集中的IT  
基础设施

2000—2020年



开放集中的云环境

2020年以后



开放分布式的普适计算环境



# 分布式系统定义

- 一个简单的定义

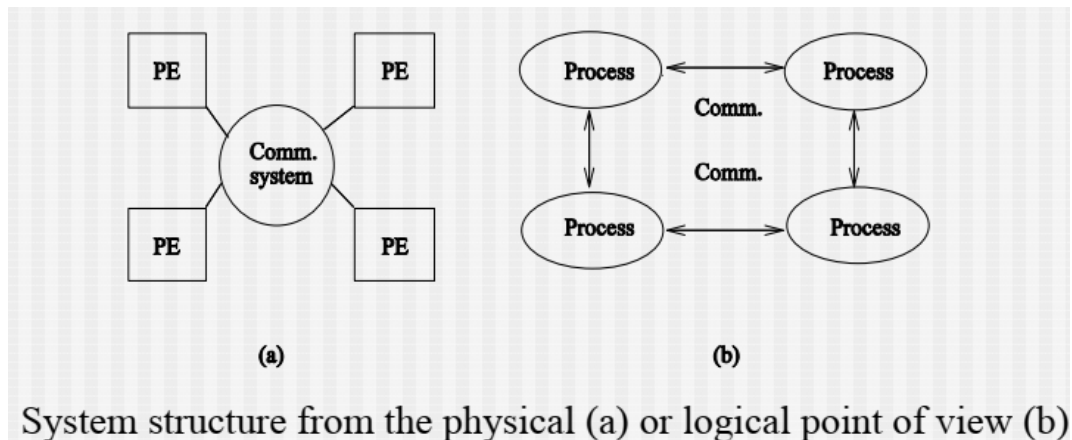
分布式系统是若干**独立自主计算机**的集合，这些计算机对于用户来说像是**单个耦合系统**。

- Leslie Lamport的定义

“A distributed system is one in which **the failure** of a computer **you didn't even know** existed can render your own computer unusable.”



物理分布，逻辑集中  
个体独立，整体统一

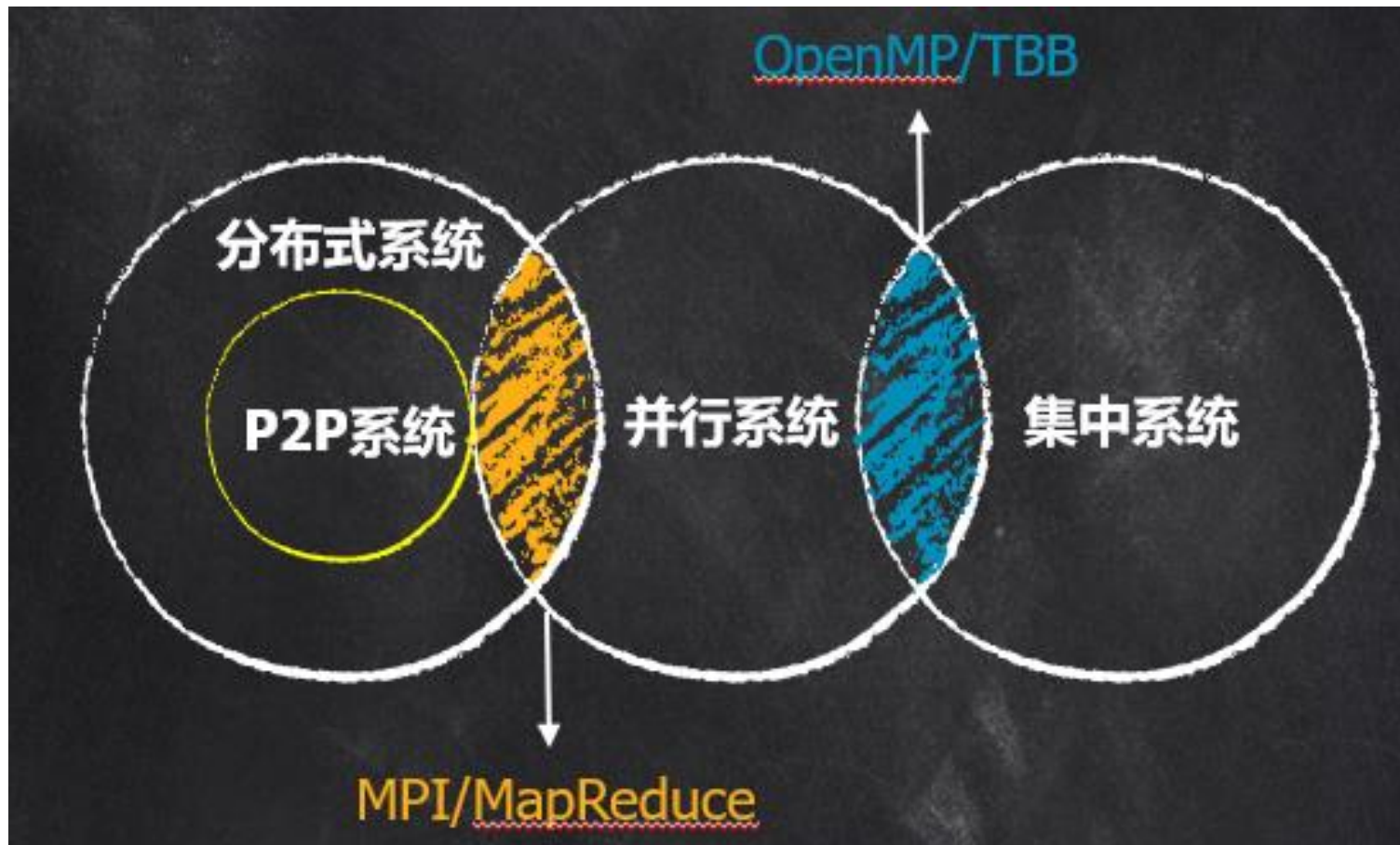


# 分布式系统的特征

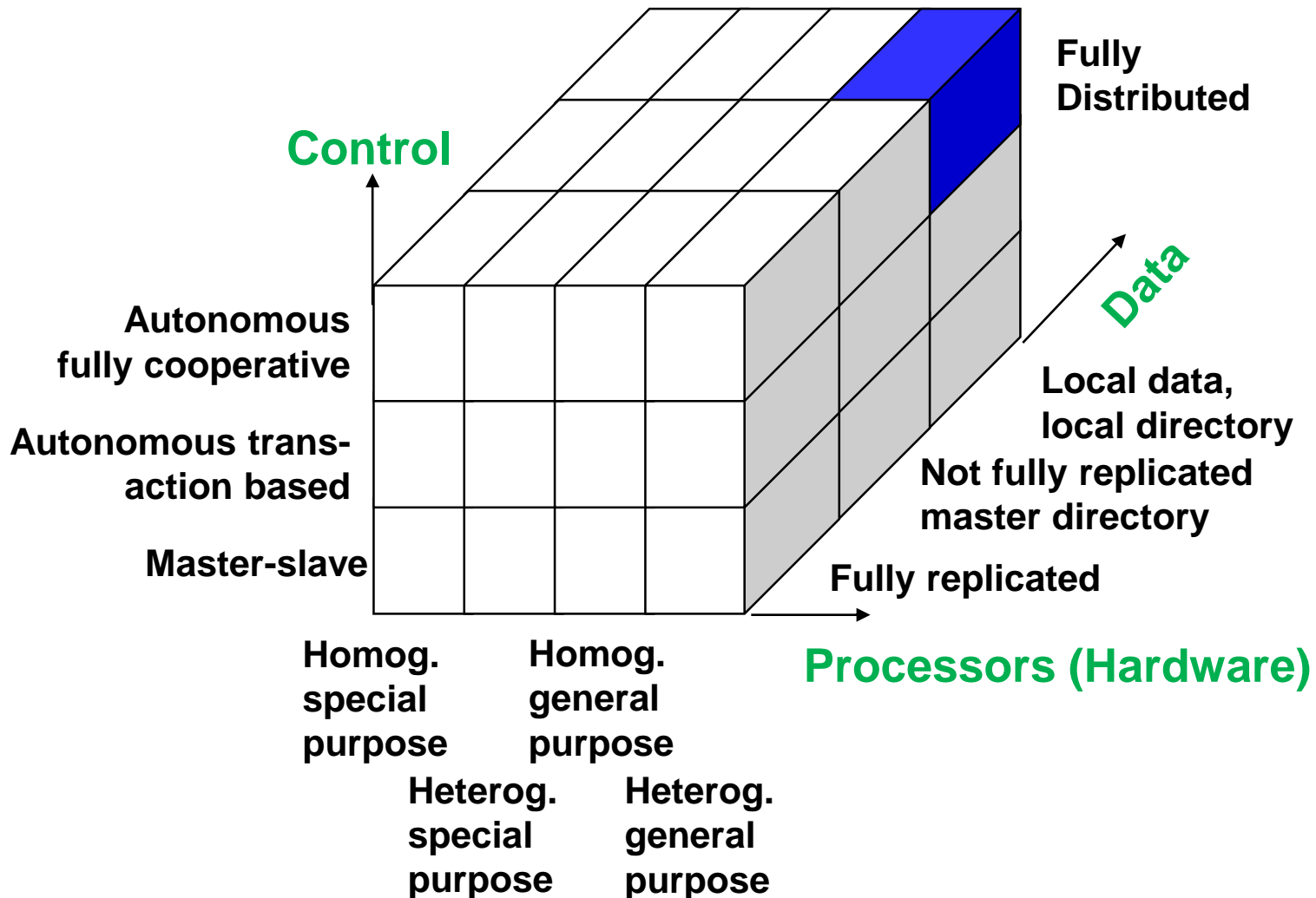
- 构成组件被所有用户共享;
- 系统资源可能不允许访问;
- 软件运行在计算单元甚至节点上;
- 允许多点控制;
- 允许多点失效;
- 没有全局时间。

# 分布式计算 vs. 并行计算

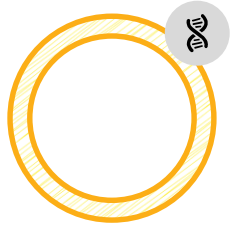
- 两者之间并无严格、确定性界限



# Enslow模型

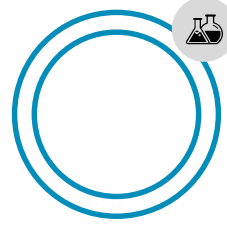


# 分布式系统的目标



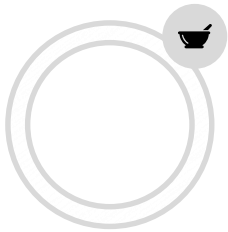
## 使资源可访问

让用户方便地访问资源



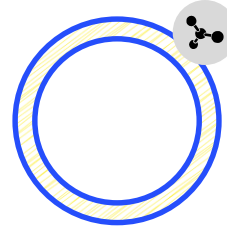
## 透明性

隐藏资源在网络上的分布



## 开放性

访问接口的标准化



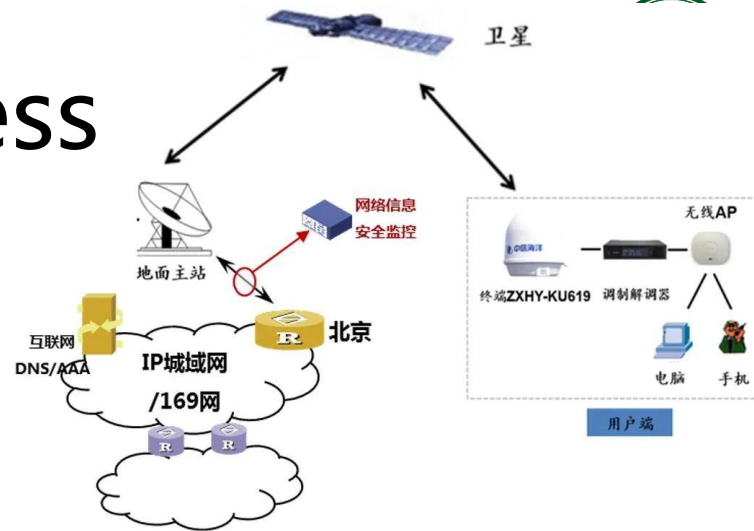
## 可扩展性

规模、地域、管理上可扩展



# 资源访问Resource Access

- 文件、数据、算力...
- 普适、随时访问



# 透明性Transparency

- 隐藏分布式系统的“分布性”

透明性	说明
访问	隐藏数据表示形式的不同以及资源访问方式的不同
位置	隐藏资源所在位置
迁移	异常资源是否移动到另一个位置
重定位	隐藏资源是否在使用过程中移动到另一个位置
复制	隐藏是否对资源进行复制
并发	隐藏资源是否由相互竞争的用户共享
故障	隐藏资源的故障和恢复
持久化	隐藏数据在主存和磁盘这一事实

# 透明性Transparency

- 透明性会影响系统性能
  - 完全透明难以实现的，而且并不可取
- 完全透明难实现：
  - 完全隐藏网络和节点的失效是不可能的
    - 不能区分失效和性能变慢的节点
    - 不能确定系统失效之前的操作是什么
  - 容错和一致性机制需要时间
    - 节点间的信息一致性需要时间
    - 容错性备份、复需要时间

# 透明性Transparency

- 不透明性有时有益：
  - 及时发现系统的性能问题
    - 如果网络通信性能问题、用户发现报告问题
  - 有时候需要暴露系统特征
    - 基于位置的服务、基于用户特征的个性化服务
    - 用户参与失效处理

透明性是一个较好的属性，但不是绝对的，需要具体分析确定。

# 开放性Openness

- 定义
  - 系统根据一系列准则来提供服务，这些准则描述了所提供服务的语法和语义（标准化）
- 具体要求
  - 良好定义的接口；
  - 良好的互操作性；
  - 支持可移植性；
  - 容易扩展（extensibility）



# 开放性Openness

- 策略

- 需要为客户端的缓冲数据设置什么级别的一致性?
- 我们允许下载的程序执行什么操作?
- 当出现网络带宽波动的时候如何调整QoS需求?
- 通信的安全水平设置多高?

- 机制

- 允许动态设定缓冲策略;
- 支持为移动代码设置不同的信任级别;
- 为每个数据流提供可调整的QoS参数;
- 提供不同的加密算法...

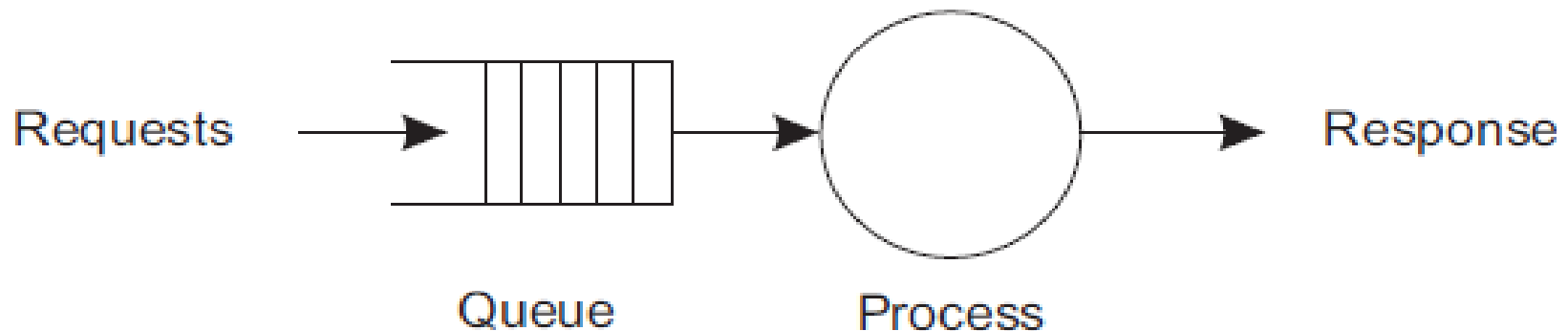
策略与机制分离：增加灵活性。

# 可扩展性Scalability

- 可以伸缩、增减的能力
- 三个维度的扩展性：
  - 规模Size: be able to easily add more users and resources to a system
  - 地理Geography: be able to handle users and resources that are far apart
  - 管理Administrative: be able to manage even if it spans independent administrative organizations
- 规模是最基本、最常用的扩展性
- 地理和管理扩展性日益重要

# 规模扩展性

- 本质是处理能力问题
  - 计算能力 (CPU、MEM性能)
  - 存储能力 (存储容量、存储IO性能等)
  - 网络能力 (带宽、延迟等)
- 理论基础：排队论



(M/M/1) 模型：无限长队列、到达率 $\lambda$ 、服务率 $\mu$ ，系统中有 $k$ 个请求的概率：

$$p_k = \left(1 - \frac{\lambda}{\mu}\right) \left(\frac{\lambda}{\mu}\right)^k$$

# 地理扩展性

- 地理分布的数据中心
- 地理分布的用户与供应商
- 不能简单从LAN扩展到WAN：
  - 广域环境中的延迟问题限制扩展性
    - 很多分布式系统假设客户端-服务器之间是同步交互
  - WAN中的连接常常是不可靠的
    - 简单地将流视频从LAN移动到WAN会导致失效
  - 缺少多点通信协议和算法
    - 譬如：搜索广播这样的功能无法执行

# 管理可扩展性

- 扩展管理边界、穿透管理边界
  - 使用方法、管理和安全策略冲突等问题
- 如：
  - 远程控制和管理设备（如IoT设备）
  - 地理分布的多数据中心协作
  - ...





# 扩展性技术Scaling Techniques

- Hiding communication latency
  - 将等待远程响应的时间与其他计算操作融合
  - 如：分离的消息处理、code at client （如JSP）
- Distributing components
  - Divide a component into smaller parts
  - E.g. dividing the DNS name space into zones
- Replicating components
  - Replicated servers, data caches...
    - 一致性是复制技术的关键难点

# *Centralized versus distributed*

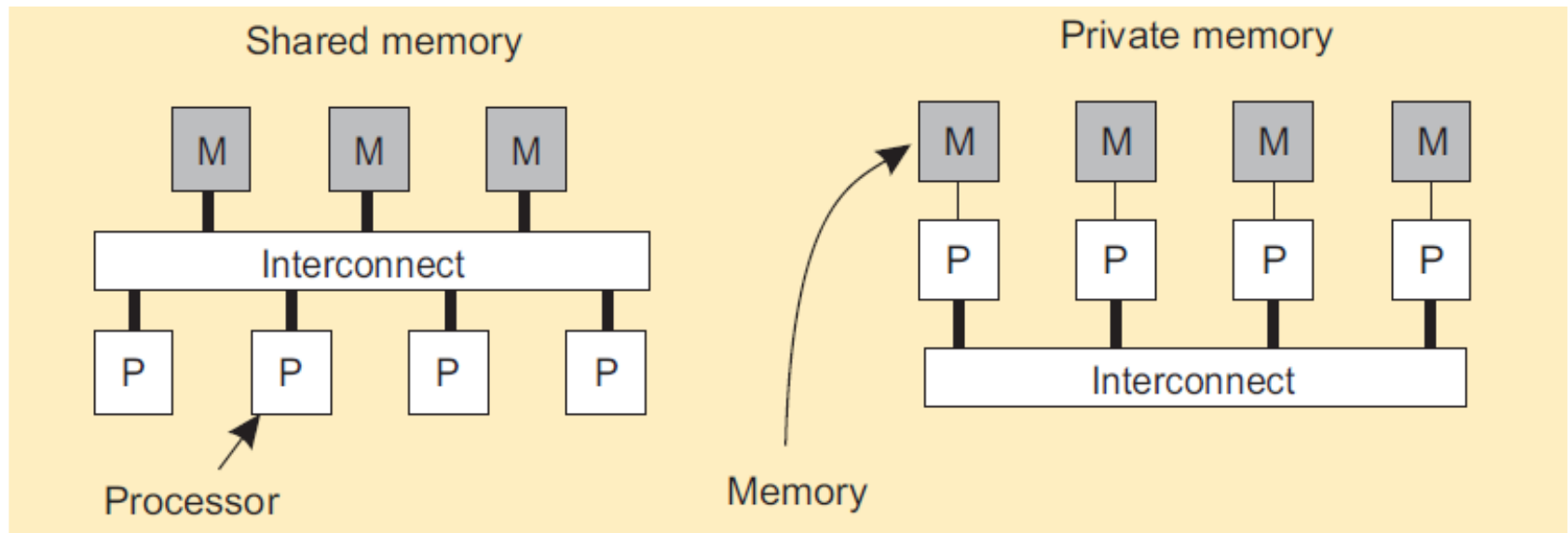
- Centralized approach

Concept	Example
Centralized services	A single server for all users
Centralized data	A single on-line telephone book
Centralized algorithms	Doing routing based on complete information

- Distributed approach
  - No global clock
  - No complete/global information
    - Decisions based only on local information
  - Failure locality

# 多处理器、分布式共享内存、多计算机

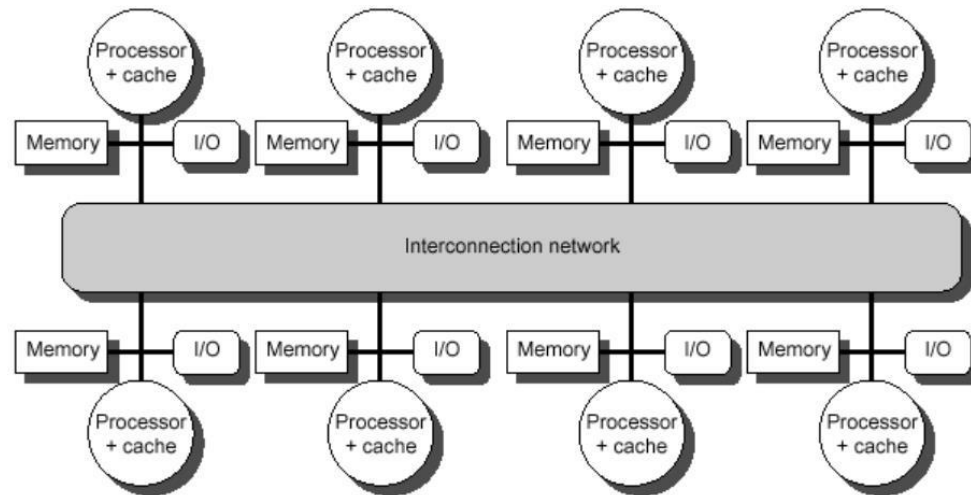
- 多处理器 (vs. 多计算机系统)
  - 编程相对简单，然而随着处理器或者核心数的增加也会遇到各种问题。



# 多处理器、分布式共享内存、多计算机

- 分布式共享内存：
  - 基于多计算实现虚拟的共享内存
  - 性能难以达到多处理器水平，已经很少用

## Distributed Shared Memory





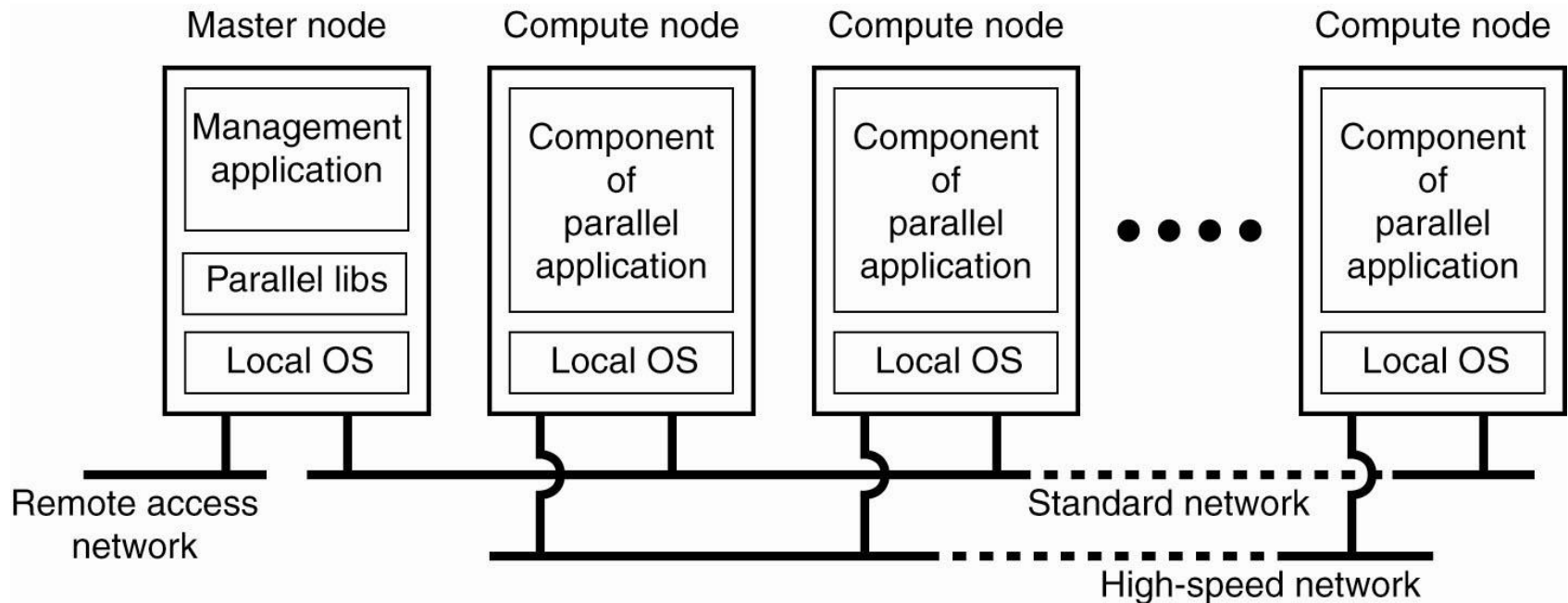




## §1.2 分布式系统分类

- Distributed (High Performance) Computing Systems
  - Cluster Computing Systems
  - Grid Computing Systems
  - Cloud Computing Systems
- Distributed Information Systems
  - Transaction Processing Systems
  - Enterprise Application Integration
- Distributed Pervasive Systems
  - Ubiquitous Computing Systems
  - Mobile Computing Systems
  - Wireless Sensor Networks

# Cluster Computing Systems

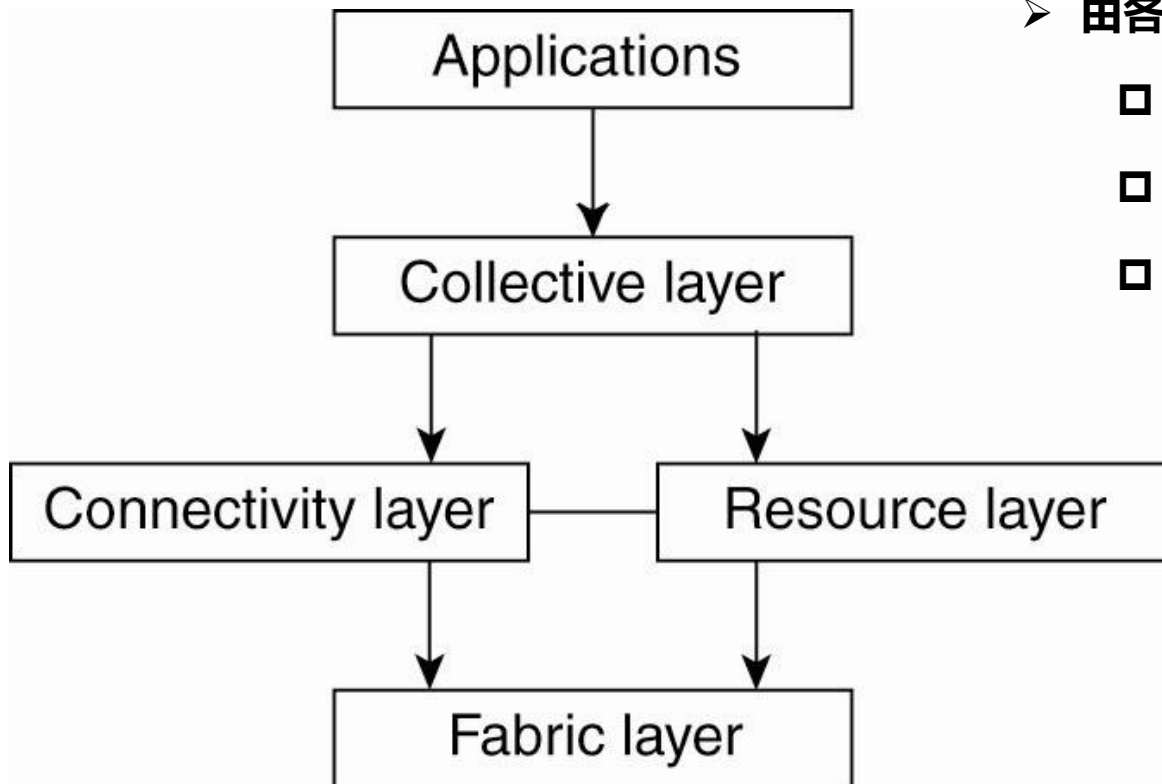


➤ 集群计算系统本质上是通过LAN连接起来的高端计算系统

□ 同构：相同的OS，近乎相似的硬件

□ 单个管理节点

# Grid Computing Systems

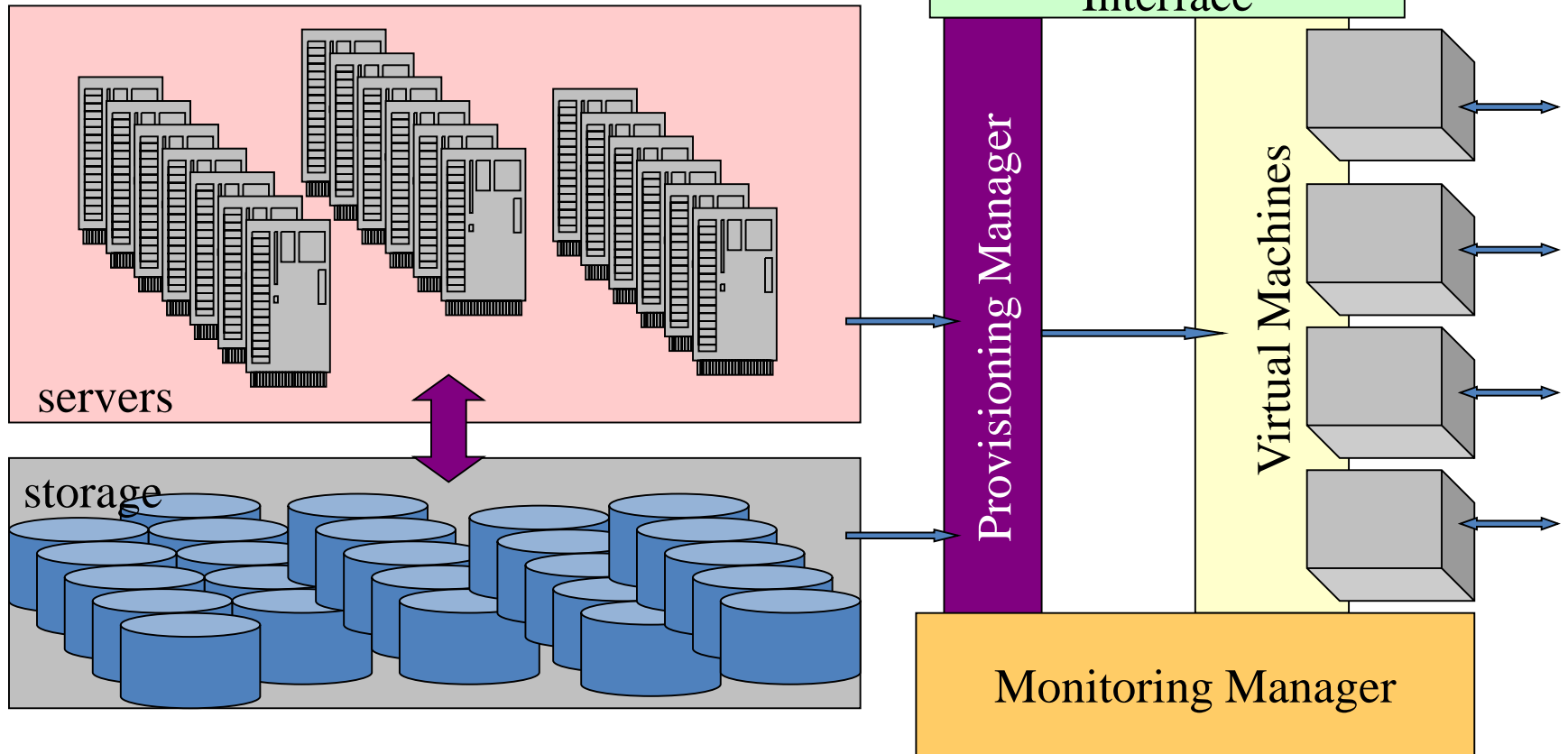


➤ 由各地的节点构成的系统

- 异构;
- 包含多个组织;
- 容易扩展到广域网的环境中

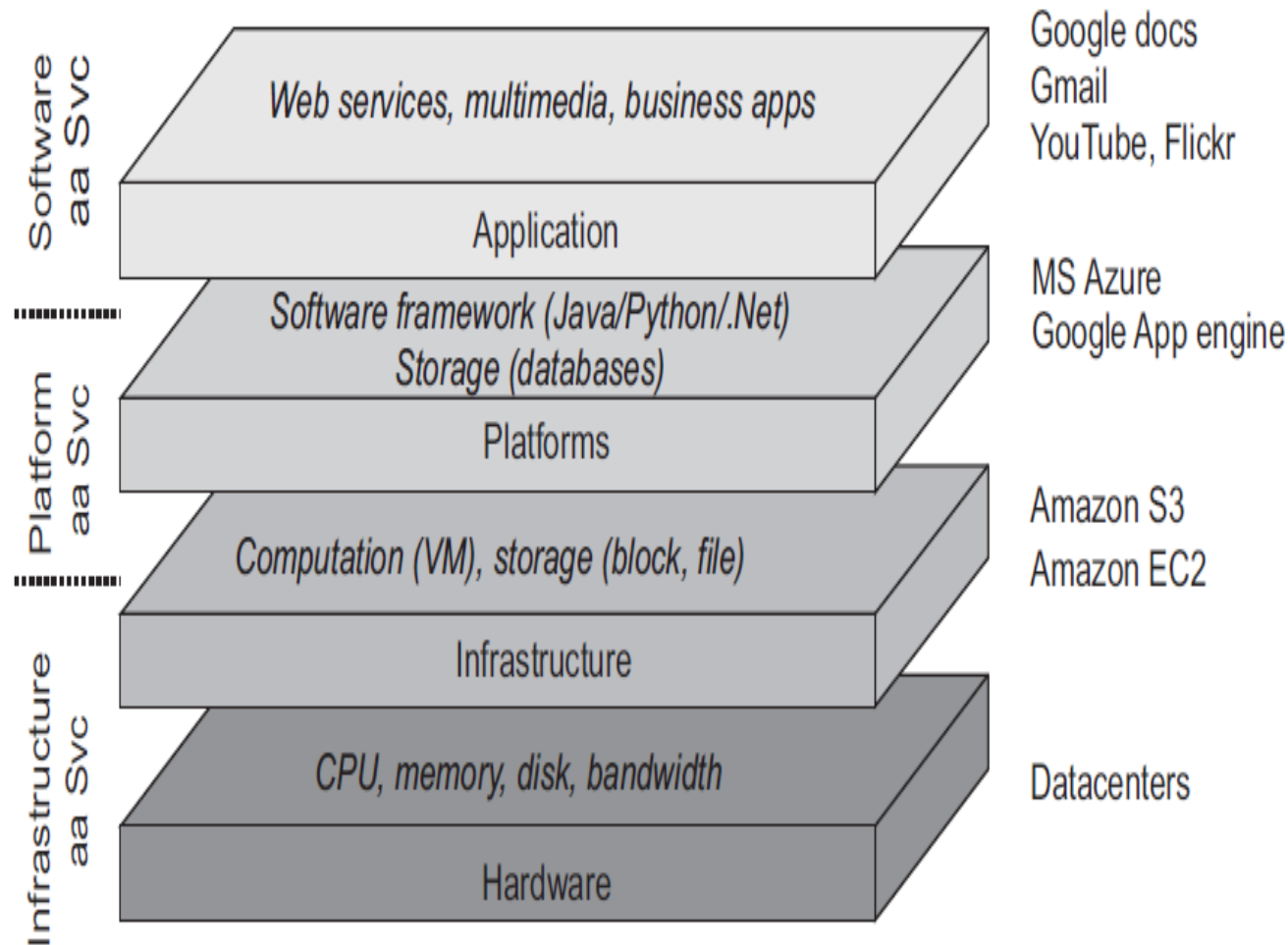
# Cloud Computing Systems

- 服务化的计算系统

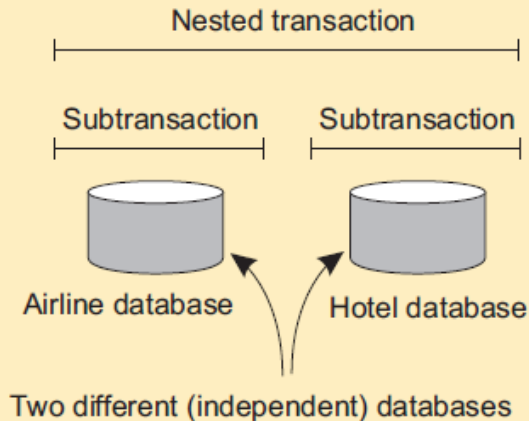


# Cloud Computing Systems

- 三个服务层次



# Transaction Processing Systems



- **Atomic:** happens indivisibly (seemingly)
- **Consistent:** does not violate system invariants
- **Isolated:** not mutual interference
- **Durable:** commit means changes are permanent

原函数	说明
BEGIN_TRANSACTION	标识一个事务处理的开始
END_TRANSACTION	终止事务处理并试图提交
ABORT_TRANSACTION	杀死事务处理并恢复旧值
READ	从文件、表或其他地方读取数据
WRITE	往文件、表或其他地方写入数据

# TPM

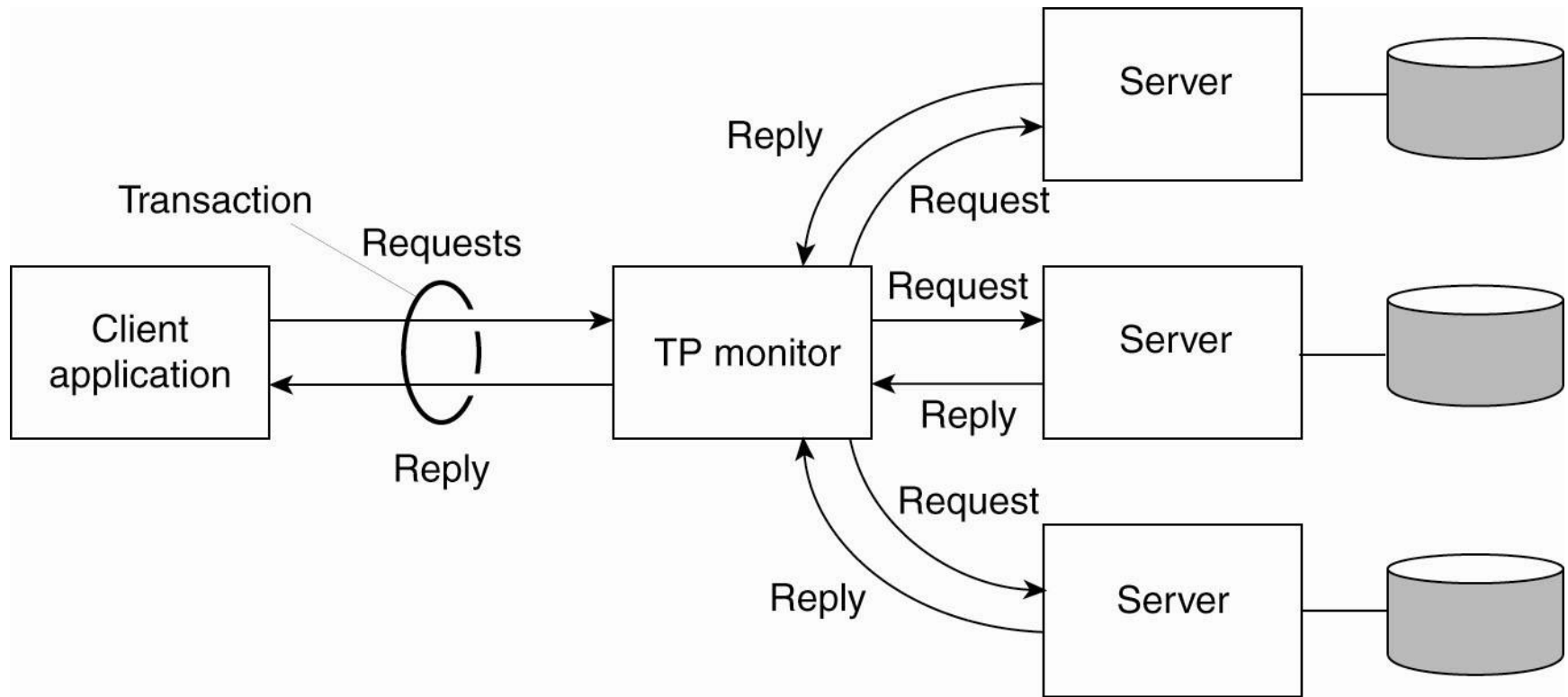


Figure 1-10. The role of a TP monitor

# Enterprise Application Integration

- 企业应用越来越复杂
  - 构成部件多
  - 功能多
  - 但是完成互操作却很复杂

## □应用集成：

- 客户端合并请求，收集请求结果
  - 允许应用程序之间进行通信



# Enterprise Application Integration

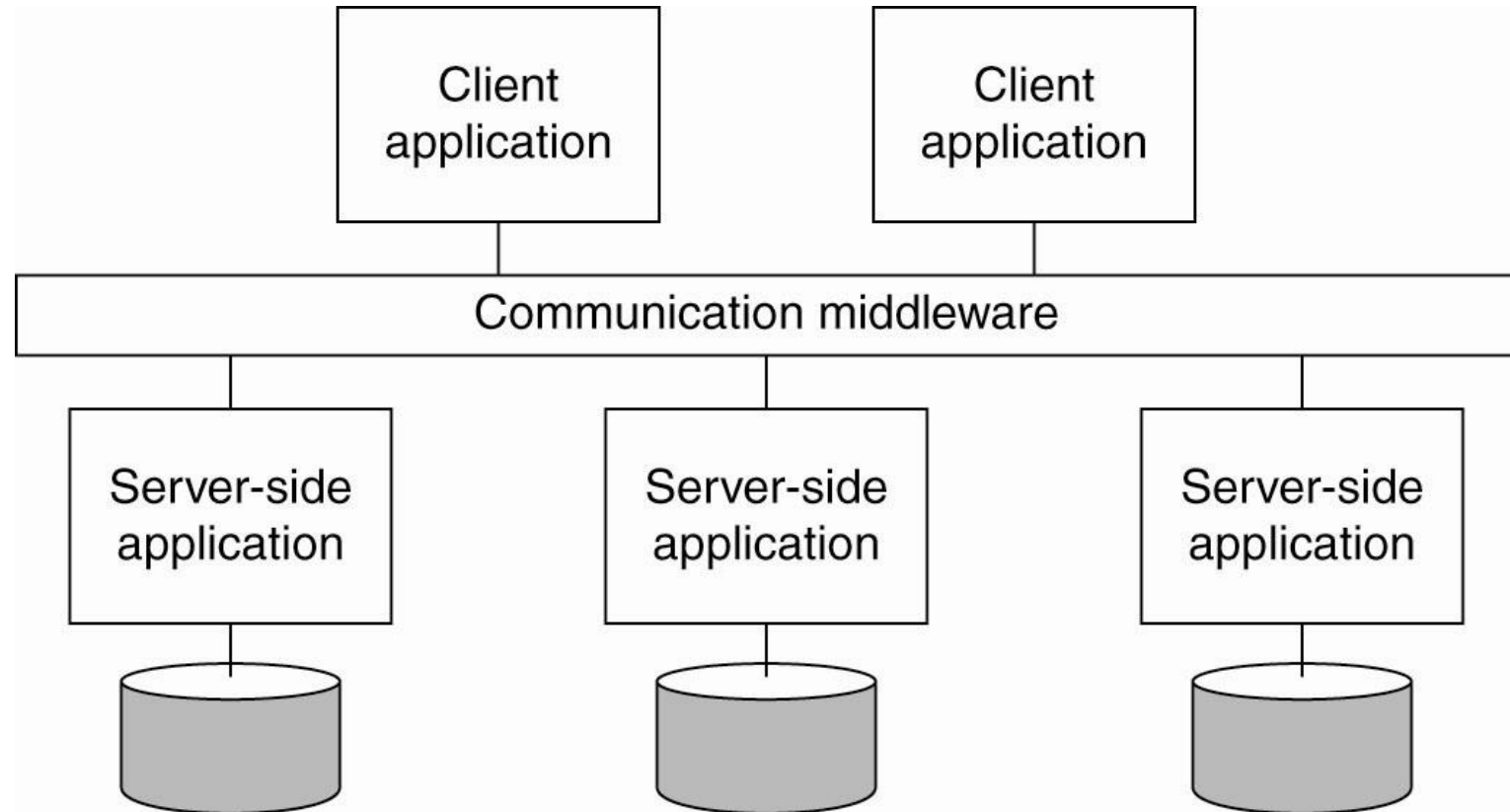


Figure 1-11. Middleware as a communication facilitator in enterprise application integration. RPC, RMI and MOM are examples.

# EAI技术方式

- 文件传输：
  - 技术实现简单，但是不够灵活；需要了解文件的格式和部署方式，了解文件的管理方法，更新传播和更新通知；
- 共享数据库：
  - 更加灵活，但是仍然需要通用的模式，导致出现瓶颈；
- 远程过程调用：
  - 当需要执行一系列的行为时非常有效，但是需要caller和callee同时在线；
- 消息传递：
  - 允许caller和callee在时间和空间上解耦。



# Distributed Pervasive Systems

- 分布式普适系统、分布式嵌入式系统
- Based on IoT
- 普适计算系统
  - 普适、连续计算，与用户连续交互
- 移动计算系统
  - 普适、计算设备是移动的
- 传感器网络系统
  - 普适、强调与环境的感知和作用

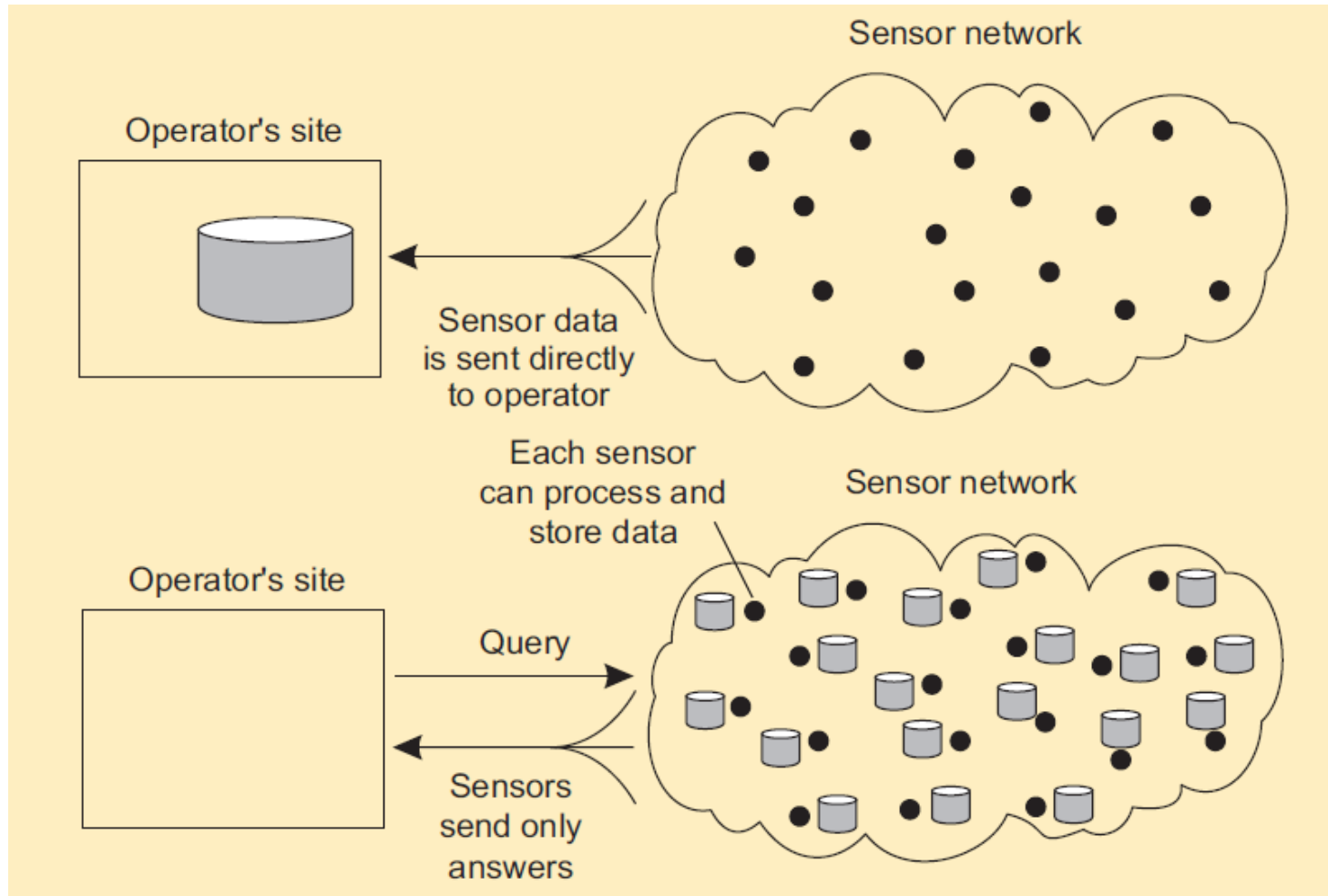
# 普适计算系统

- 分布式
  - 设备是通过网络连接、分布并且是透明访问的;
- 交互
  - 用户和设备之间的交互是高度隐蔽的;
- 上下文可感知
  - 系统知晓用户的上下文以便于优化交互行为;
- 自主性
  - 设备自主运行, 不需要人为干预, 因此具有高度的自我管理能力;
- 智能
  - 系统作为一个整体可以处理一系列的动态行为和交互;

# 移动计算系统

- 大量的不同的移动设备
  - 智能手机、平板、GPS设备、遥控器等
- 设备的位置随着时间的变化而变化
  - 本地服务、可连接性的变化
  - 关键词：“发现”
- 设备之间的通信变得很困难：
  - 没有稳定的路由，而且没有可保证的连接性，这也就要求网络连接可容错
  - 机会路由方式

# 传感器网络



# 传感器网络

- 节点数量众多：成百上千，甚至更多
- 节点简单：内存小、计算能力低、通信效率低
- 发展：从单纯感知到智能控制、智能物联网



# Summary

- 分布式系统定义
- 分布式系统的目标
  - Making resources accessible
  - Distribution transparency
  - Openness
  - Scalability
- 分布式系统的类型
  - Distributed computing system
  - Distributed information system
  - Distributed pervasive system





谢谢!

wuweig@mail.sysu.edu.cn