EBU7501: Cloud Computing
Week 1, Day 1: Cloud Computing Fundamentals

Dr. Gokop Goteng

Queen Mary
University of London

Lecture Aim and Outcome

- Aim
  - The aim of this lecture is to introduce the students to the basics of distributed and cloud computing
- Outcome
  - At the end of this lecture, students are expected to:
    - Define and identify types of distributed computing
    - Design and identify the components of cloud architecture
    - Have a good understanding of service-oriented architecture
    - Know the different commercial and open source cloud solutions available
    - Appreciate the advantages of using cloud computing as a technology

Lecture Outline

- **Cloud Fundamentals: Distributed Systems**
  - **Client/Server Architectures**
- **Cloud Fundamentals: Grid Computing**
  - **Grid Architecture**
- **Cloud Computing**
  - **Cloud Architecture**
- **Advantages of Cloud Computing**
- **Cloud Middleware**
- **Cloud Solutions**
  - **Commercial**
  - **Open Source**
- **Class Task**

Distributed Systems

- A distributed system consists of hardware, software and data components located at networked computers which communicate and coordinate their actions using protocols and passing messages [Coulours et al., 2005]
  - The components in distributed systems interact to accomplish a common goal or task, usually a computation with an output/result or a data process/analytics with its output for decision making.
- Distributed computing: This is a specialised area/field in computer science which is dedicated to the study of distributed systems.

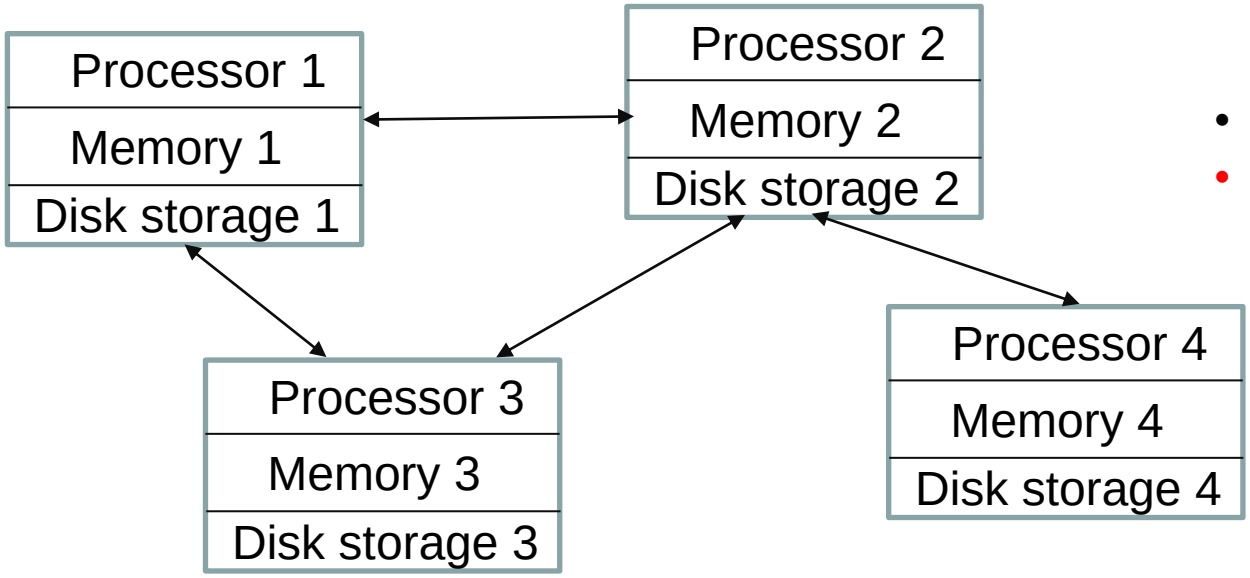
Characteristics of Distributed Systems

- Concurrency: Multiple components working on different parts of the same task at the same time
- No global clock for time system
  - Difficulty in all components located in different parts of the world to maintain the same time
    - The Network Time Protocol (NTP) is usually used to solve this problem in distributed systems
- Autonomous computational components/entities
  - Independent failure in components
    - Loosely coupled components such that if one fails, it should not affect the workings of the rest
- The components/entities communicate with one another by message passing

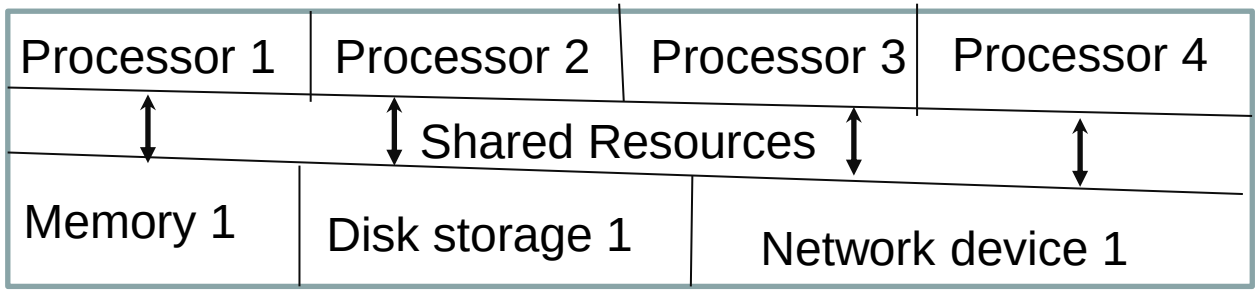
Parallel and Distributed Computing

- Parallel computing is the type of computation in which many computations and process executions are carried out simultaneously
  - This is done by dividing a large problem into smaller parts that are carried out by different components at the same time to speed up the computation or execution process
  - It uses shared resources such as memory
  - It exchanges information between processors
- So, it means parallel computing can be seen as a system whose processors use shared resources such as shared memory to execute processes by exchanging information while distributed computing has processors that have their own individual memory but communicate by passing messages

Parallel and Distributed Computing



- Distributed computing
- Loosely coupled



- Parallel computing
- Tightly coupled
- Resource contention

Distributed Programming

- Distributed programming is the art of writing programs that can run on distributed systems which allows communication and passing of messages among the distributed nodes and processors.
- Programs that run on distributed systems are called distributed programs

Parallel Programming

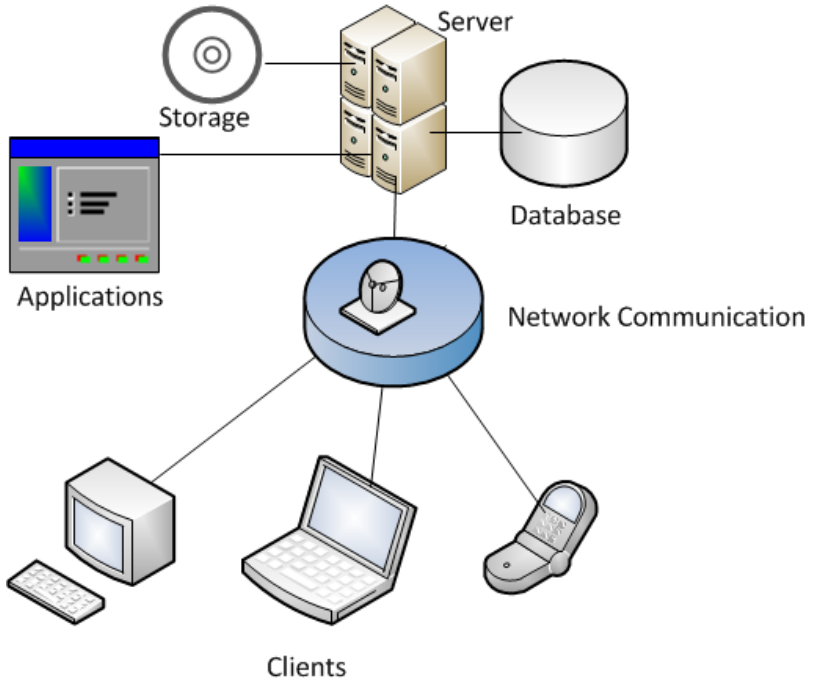
- Parallel programming is the art of writing programs that run on parallel computing systems which allows exchanging information and using shared resources such as memory and storage space to perform computations.
- Programs that run on parallel computing platforms are known as parallel programs

Examples of Distributed Systems

- Internet
- Intranets
- Grid computing
- Cloud computing

Architectures of Distributed Systems

- **Client-Server Architecture**:
  - 2-tier architecture
    - Server and clients
    - There is usually one server and one or more clients



Architectures of Distributed Systems

- **Client-Server Architecture**:
  - 3-tier architecture or Layered Architecture
    - Presentation layer
      - User interface where users interact with the system
    - Business logic or application layer
      - Where the functionalities of the application is implemented. This is where the intelligence reside
    - Data layer
      - Where the data management is done

Architectures of Distributed Systems

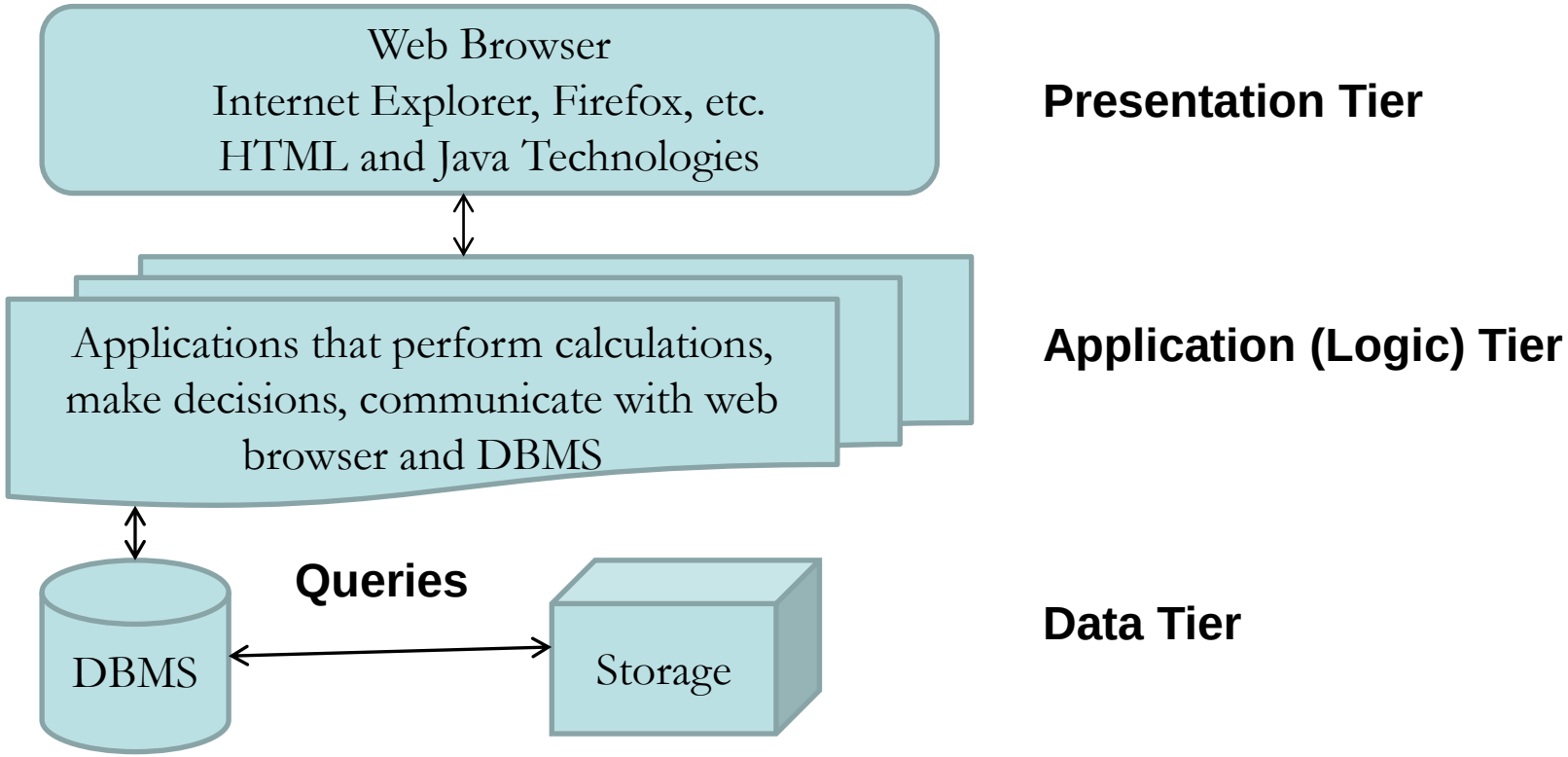
- **Client-Server Architecture**:
  - Web-based 3-tier architecture
    - Called electronic commerce websites
    - Web-server
      - Front-end which displays static and at times dynamic contents to the users from a browser
    - Application server
      - Dynamic content processing and generation of information
    - Database store
      - Back-end which consists of data sets and database management system (DBMS)

Architectures of Distributed Systems

- **Client-Server**: Clients are used for input by users and display of output while servers are the location of data, data management and where intelligent processing happens
  - Server: Provides computing resources to clients
  - Client: Requests (uses) computing resources that are provided by the server
- Servers and clients communicate through computer networks and can reside on separate or the same machine

Architectures of Distributed Systems

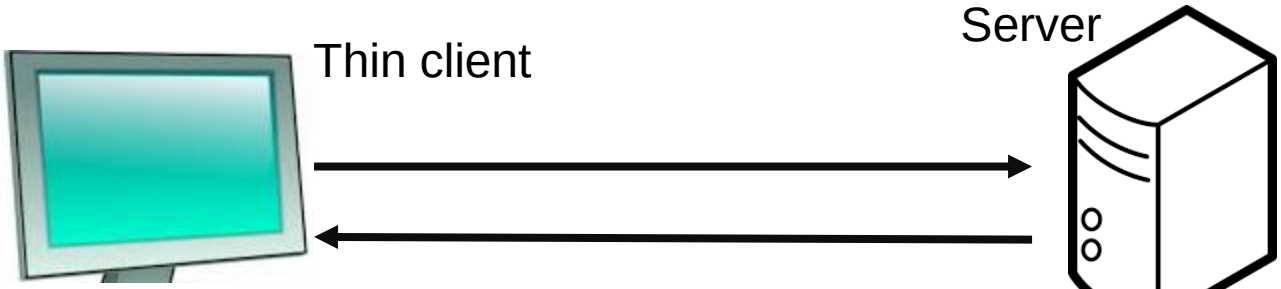
- **Client-Server Architecture**:
  - n-tier architecture or multi-tier architecture
    - A generalised name for client-server architecture



Thin and Fat Clients

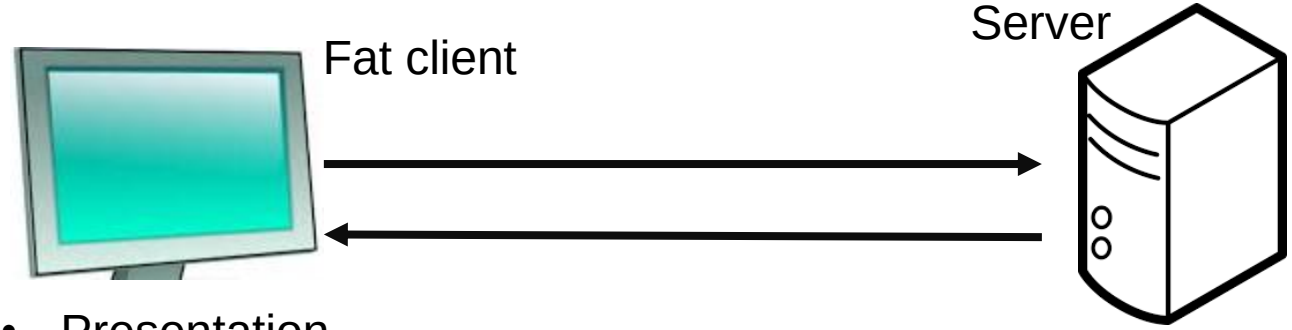
- Thin client:
  - The client is only the presentation used to make input and display output, it has no any client application processing capabilities or “intelligence”. All data storage, data processing and intelligence are provided by the server side
  - The heavy-lifting of data management and application processing is entirely on the server and network
- Fat client:
  - Some application processing capabilities are delegated to the client
  - Application processing can be executed locally at the client side
  - Application processing is also done at the server as well
  - More balanced distribution of application processing between the client and server

Thin and Fat Clients



- Presentation

- Data management
- Application processing



- Presentation
- Application processing

- Data management
- Little or no application processing

Peer-to-Peer (P2P) Systems

- A Peer-to-peer (P2P) system consists of nodes of computers that perform both client and server functionalities and capabilities.
  - This means that P2P is client-oriented (not server-oriented)
  - Peer machines are client computers connected to the internet
  - Peer machines autonomously join or leave the network
  - This means all machines are equal, no master-slave relationship
  - No central coordination or central database is required
  - No Peer machine has a global view of the entire P2P system
- The field of computing that deals with P2P systems is known as P2P computing

Cloud Fundamentals: Distributed Systems

- What is distributed computing?:
  - Is the field of computing systems in which multiple computer components communicate with each other via networks to solve a common problem by passing messages across to each other
  - The task is divided among the computers to save time and computer resources
  - There is concurrency (at the same time) and independent failure mode among the cooperating computers
- What is parallel computing?:
  - Is the use of multiple central processing units-CPUs (multi-core) of computers to process data or perform computations on the same task at the same time so as to speed up the process. Each CPU processes part of the whole task and the entire results are combined to produce the output.
- Advantages:
  - Modularity
  - Speed up computation and data processing
  - Ease of tracking system failure

Cloud Fundamentals: Distributed Systems

- Flynn's taxonomy
  - In 1966, Michael J. Flynn proposed 4 classifications for computer architectures known as the Flynn's taxonomy
    - SISD: Single Instruction Single Data stream
      - Serial computer
      - No parallelism
    - SIMD: Single Instruction Multiple Data streams
      - Exploits some parallelism which is simple to accomplish
    - MISD: Multiple Instruction Single Data stream
      - This is not a popular architecture in practise, though it is sometimes used in fault tolerance but almost never used in any application
    - MIMD: Multiple Instruction Multiple Data streams
      - Truly parallel computation and most widely used
      - Multiple independent processors concurrently executing different instructions on different data streams
      - Majority of distributed systems use MIMD architecture
      - The basis for present supercomputer architectures

Cloud Fundamentals: Historical Background

- Started with the concept of “Utility Computing” by John McCarthy in 1961
  - Computing of the future will be organised and used as public utilities such as telephone, electricity, water, etc
- Grid computing
  - Ian Foster and Kesselman started the concept of computational grid in 1995 to mimic the electricity grid that provides electric as a utility on demand basis which is based on how much you use the computational resources and be charged on a metering system
- The innovations around this concept gave birth to technology giants to develop cloud-based platforms such as search engines (Google, Yahoo), email services (Hotmail, Gmail), social open publishing services (Facebook, MySpace, YouTube), social media (Twitter, LinkedIn) in the mid 1990s
- Salesforce.com started the concept of remotely provisioned services on enterprise systems in the late 1990s
- Amazon.com launched its successful Amazon Web Services (AWS) platform that provides enterprise remotely provisioned storage, computing elements and business functionalities in 2002

Cloud Fundamentals: Grid Computing

- Coined from electricity "Grid" of 1910
  - A concept that computational resources will be based subscription
  - Used on-demand basis
  - Did not achieve a "truly" on demand and subscription features of electricity
- Large scale distributed systems
  - Collection of computational resources from geographically distributed locations
  - To solve a common problem
  - Mostly used for scientific applications rather than for commercial purpose
- Uses general purpose middleware
  - Interacts with heterogeneous systems

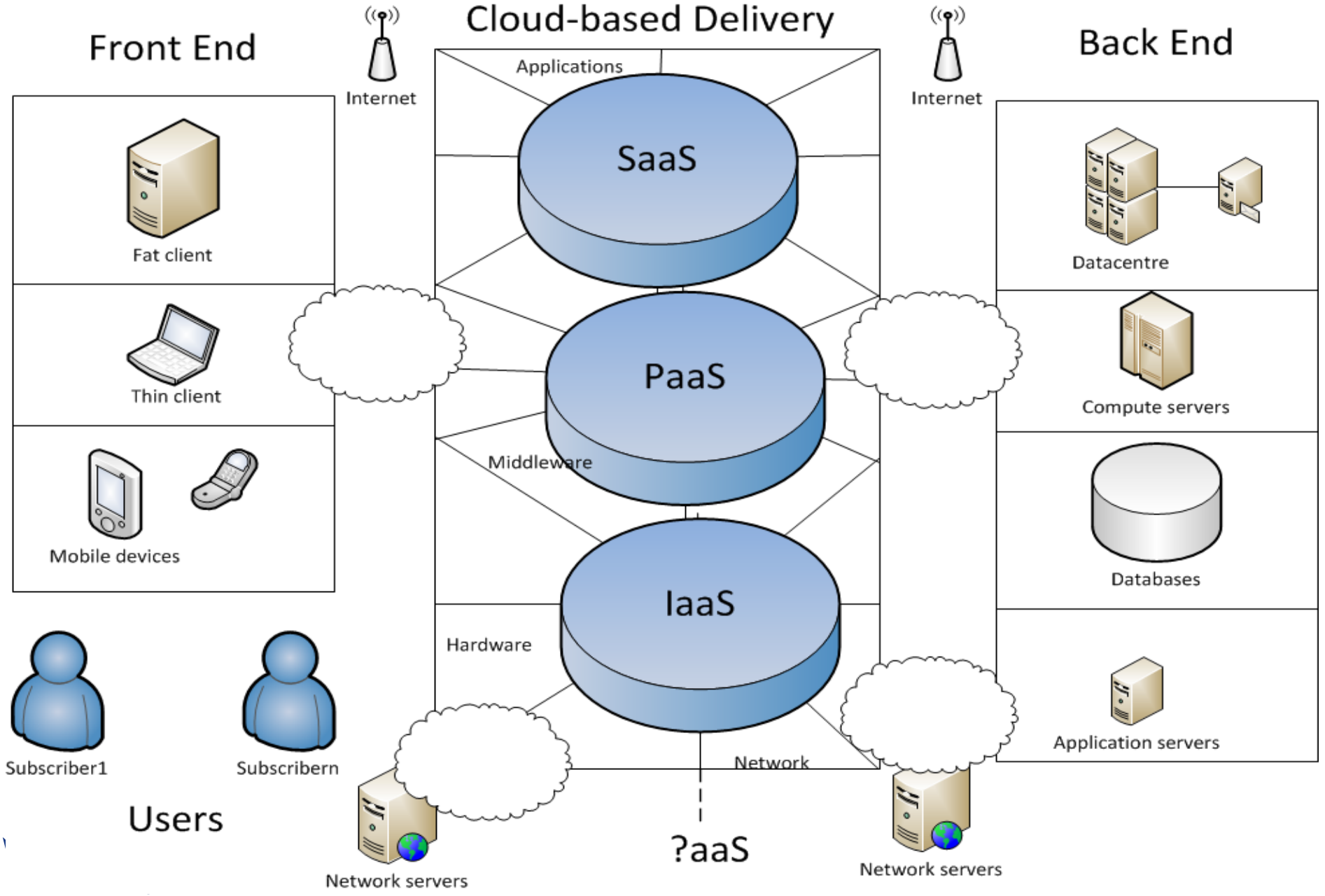
What is Cloud Computing?

- Cloud computing is a computing infrastructure which consists of shared pool of virtualised hardware, software and data resources that are provided as services at both small and large scale on a demand basis over the internet
  - Cloud computing is a form of distributed computing which enables computational resources to be provided as services by service providers to users known as service requestors or service consumers
  - The service requestors subscribes to the services and are charged based on the amount of usage on a demand-basis
  - The provisioning and management of the resources is usually done by the cloud service provider

Cloud Computing

- A “truly” utility computing facility that offers computing resources as services on a large scale
  - Enabled by the internet
  - Uses general purpose and specialised middleware to connect heterogeneous systems and platforms
  - Based on the analogy of electricity as a utility
    - Users do not own and maintain the resources, but subscribe for the computing resources
    - No standalone applications or hardware
    - Consists of shared computing resources offered by the provider
  - Usage is on demand basis
  - Elasticity of computing resources
    - Resources can scale up and down on the fly based on subscribers requirements and agreements
  - Based on service-oriented architecture (SOA)
    - SaaS: Software-as-a-Service
    - PaaS: Platform-as-a-Service
    - IaaS: Infrastructure-as-a-Service
  - Economies of scale
    - Cheaper and saves time

Cloud Computing: High Level View Architecture



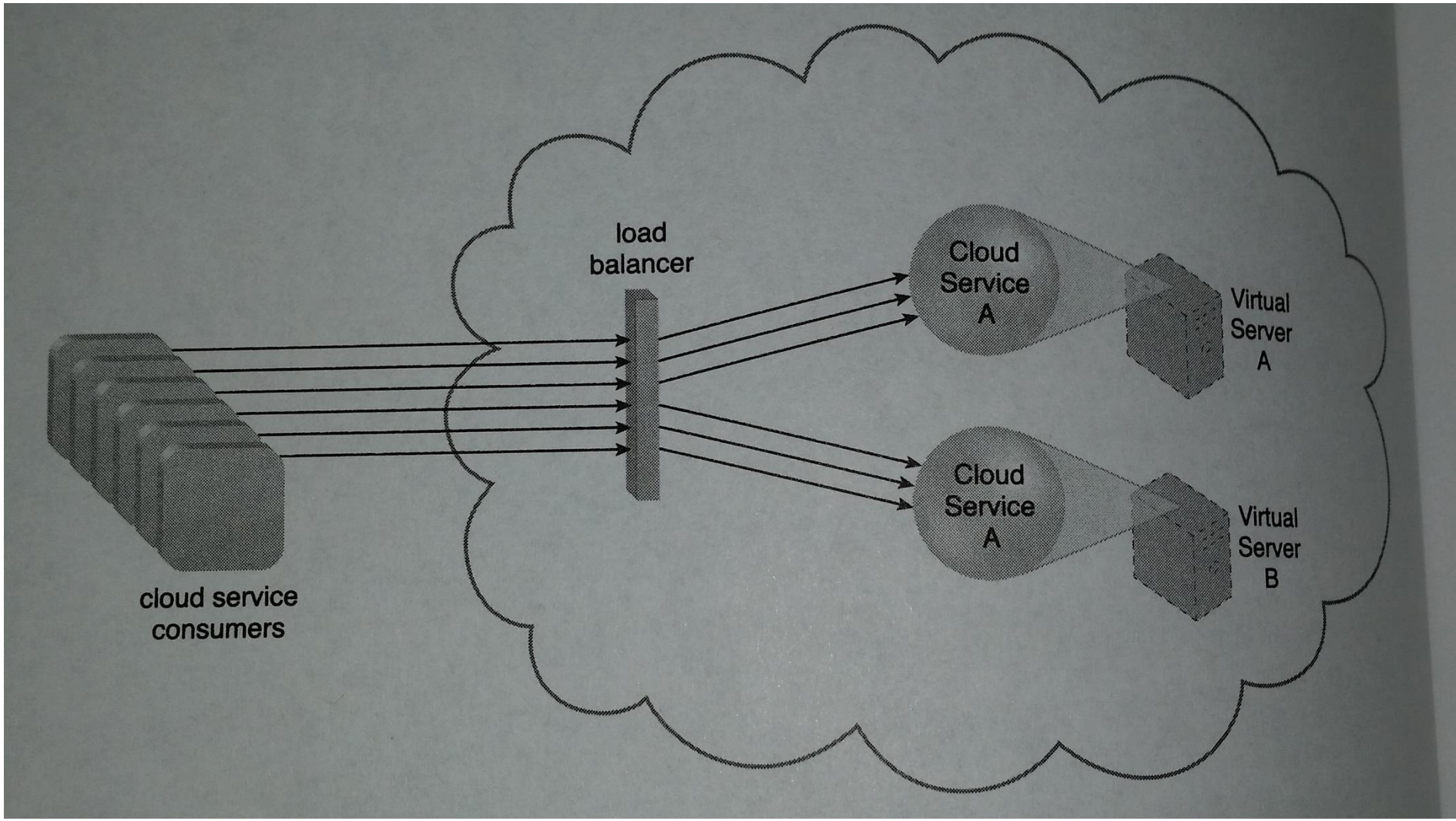
Some Cloud Architectures

- Workload distributed architecture
- Dynamic scalability architecture
- Elastic resource capacity architecture
- Service load balancing architecture
- Elastic disk provisioning architecture
- Redundant storage architecture

Workload Distributed Architecture

- Horizontal scaling by adding one or more identical IT resources and a load balancer
- This will evenly distribute the workload among the IT resources
- This reduces IT resource over-utilisation and underutilisation

Workload Distributed Architecture

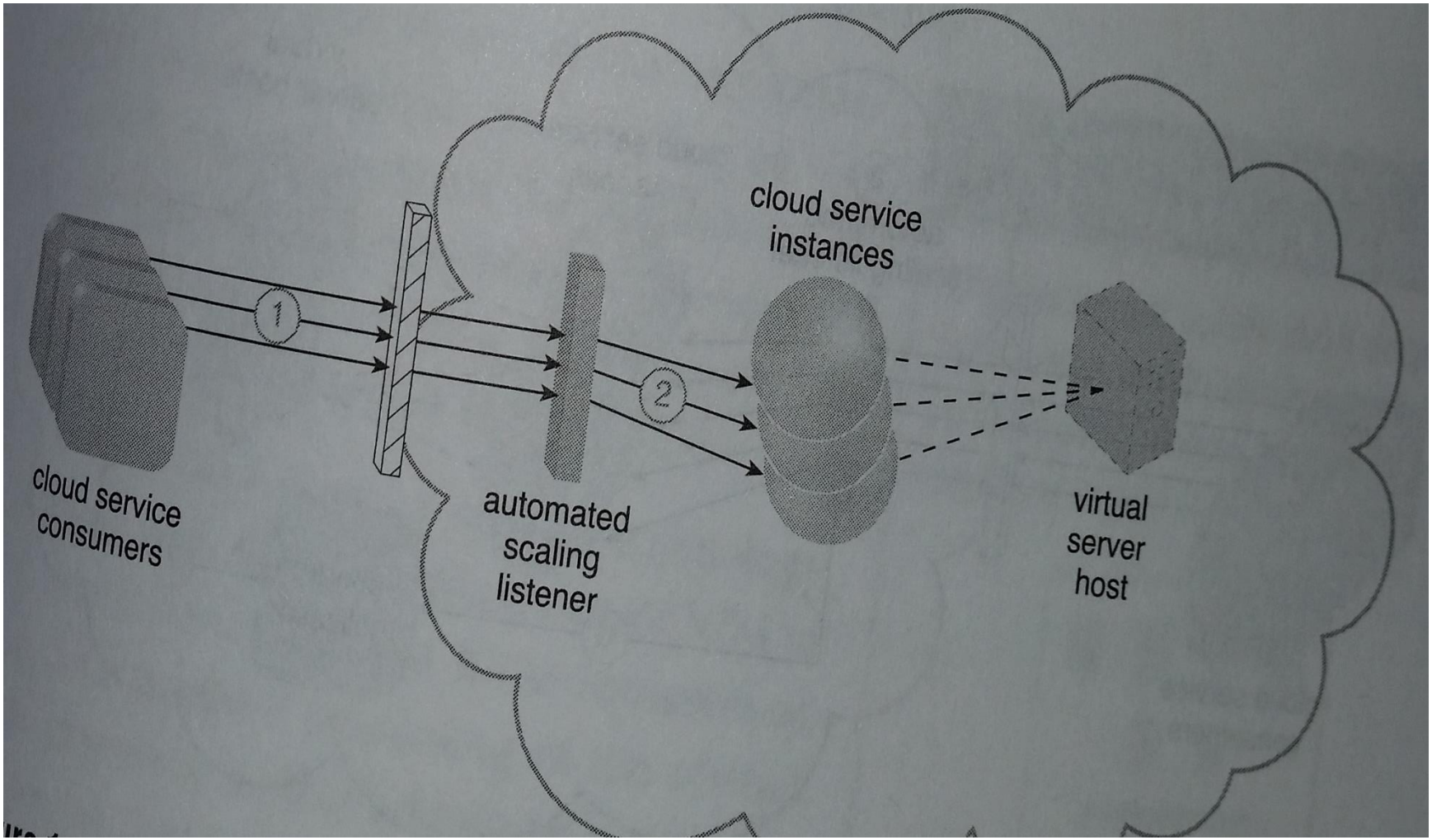


Source: Cloud Computing: by Thomas E., Zaigham M. and Ricardo P, 2013

Dynamic Scalability Architecture

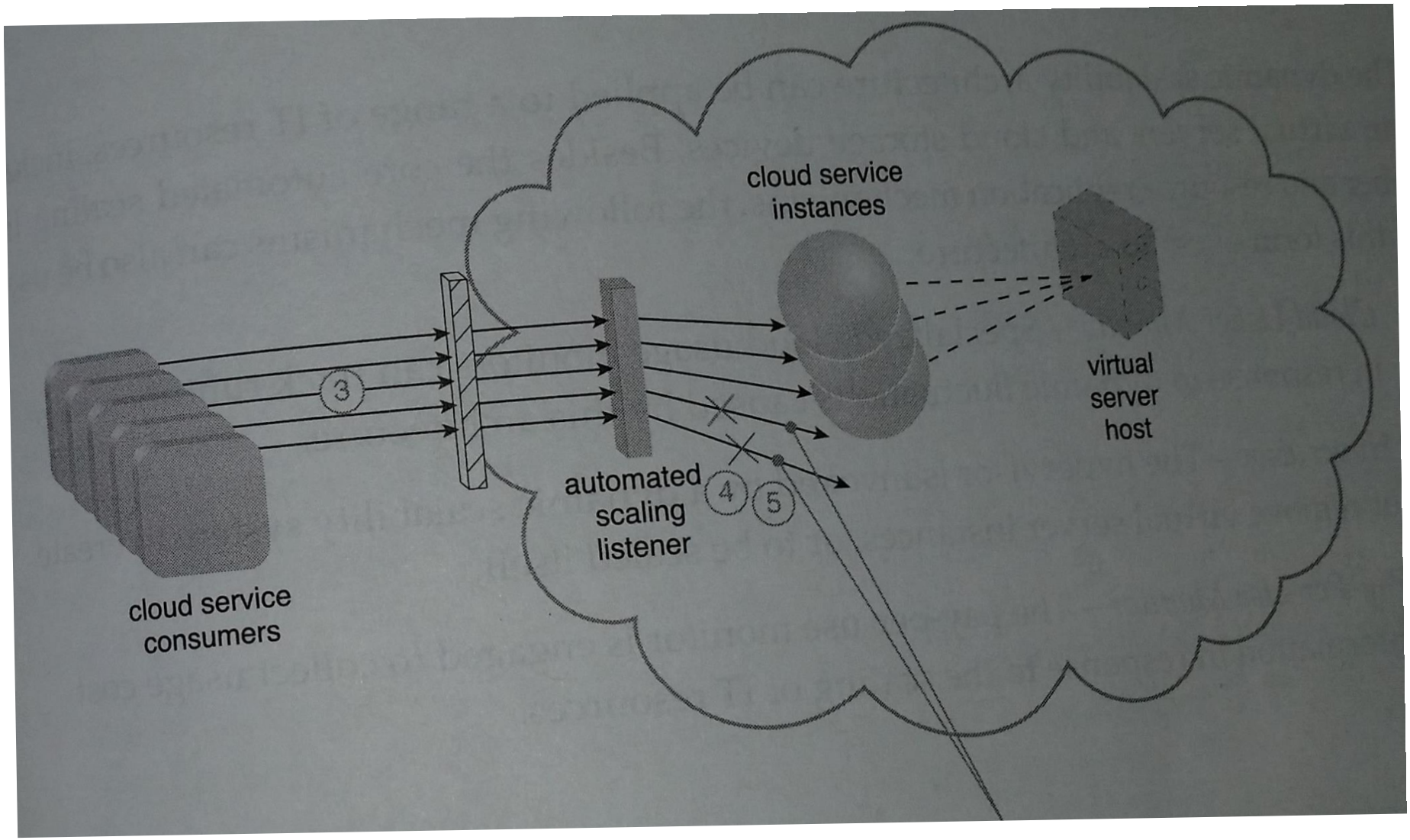
- This is based on pre-defined scaling conditions
- The automated scaling listener is configured with workload thresholds that dictates when new IT resources need to be added to the workload processing
- There is horizontal and vertical dynamic scaling and relocation

Dynamic Scalability Architecture



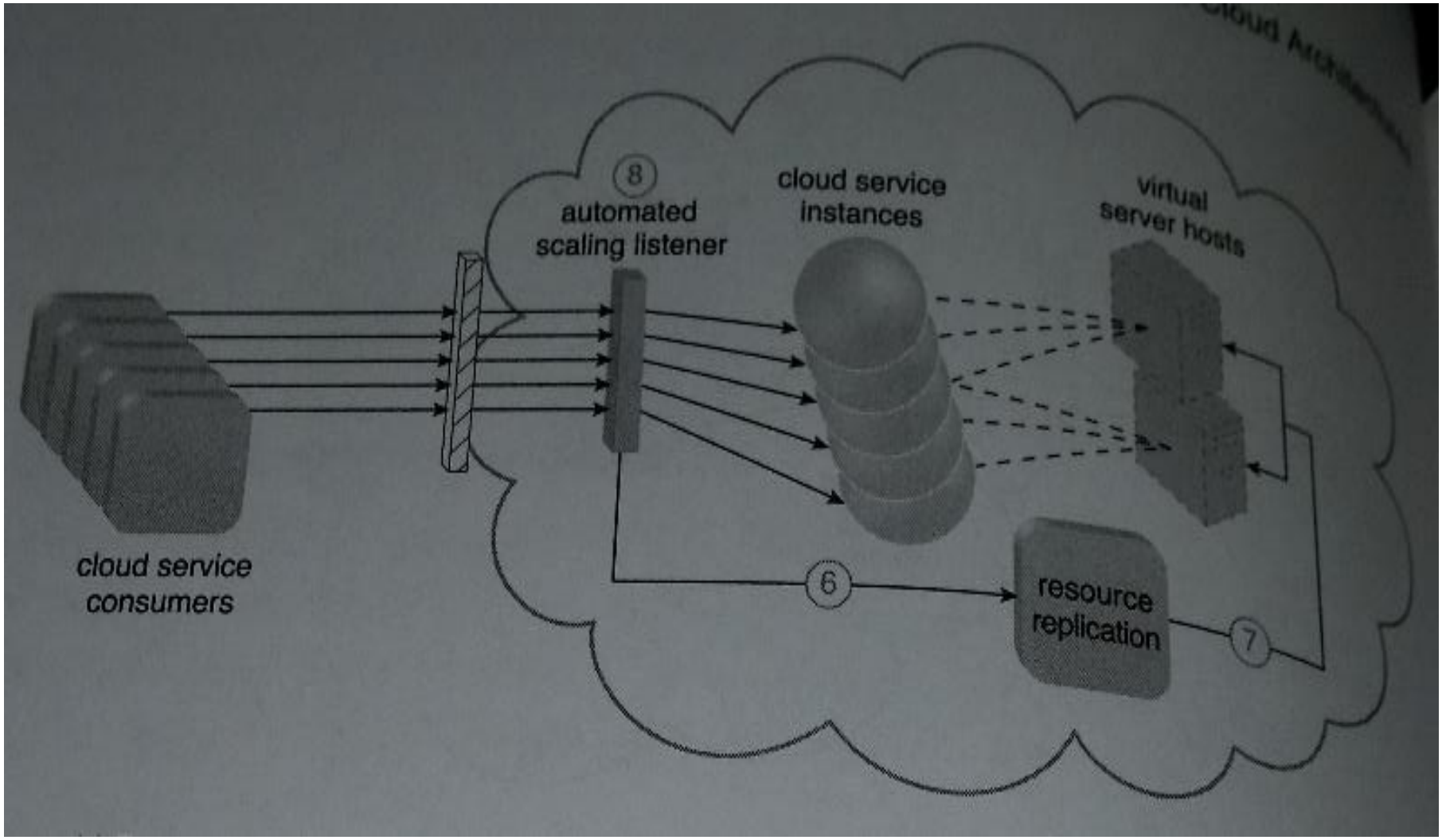
Source: Cloud Computing: by Thomas E., Zaigham M. and Ricardo P, 2013

Dynamic Scalability Architecture



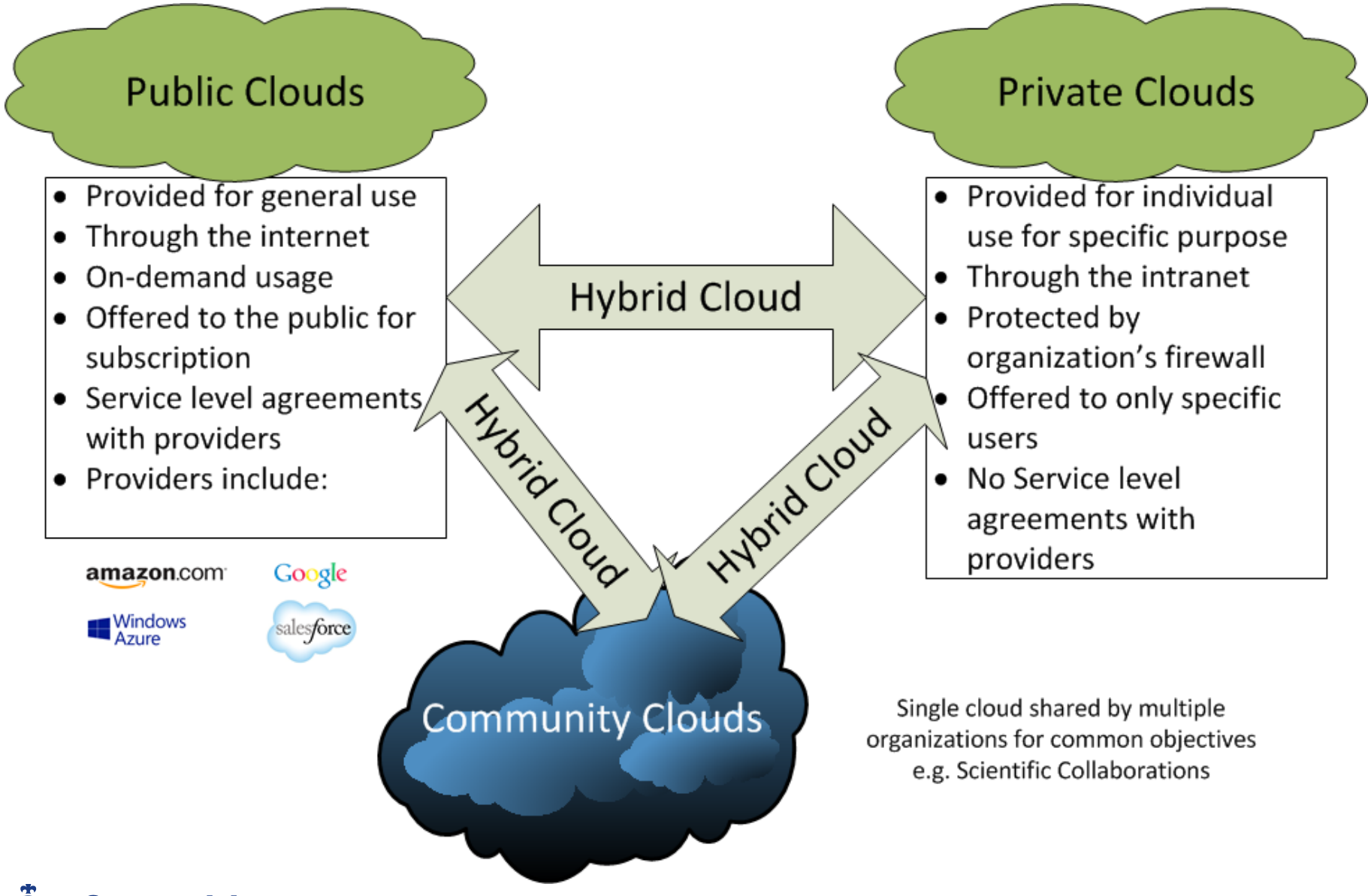
Source: Cloud Computing: by Thomas E., Zaigham M. and Ricardo P, 2013

Dynamic Scalability Architecture

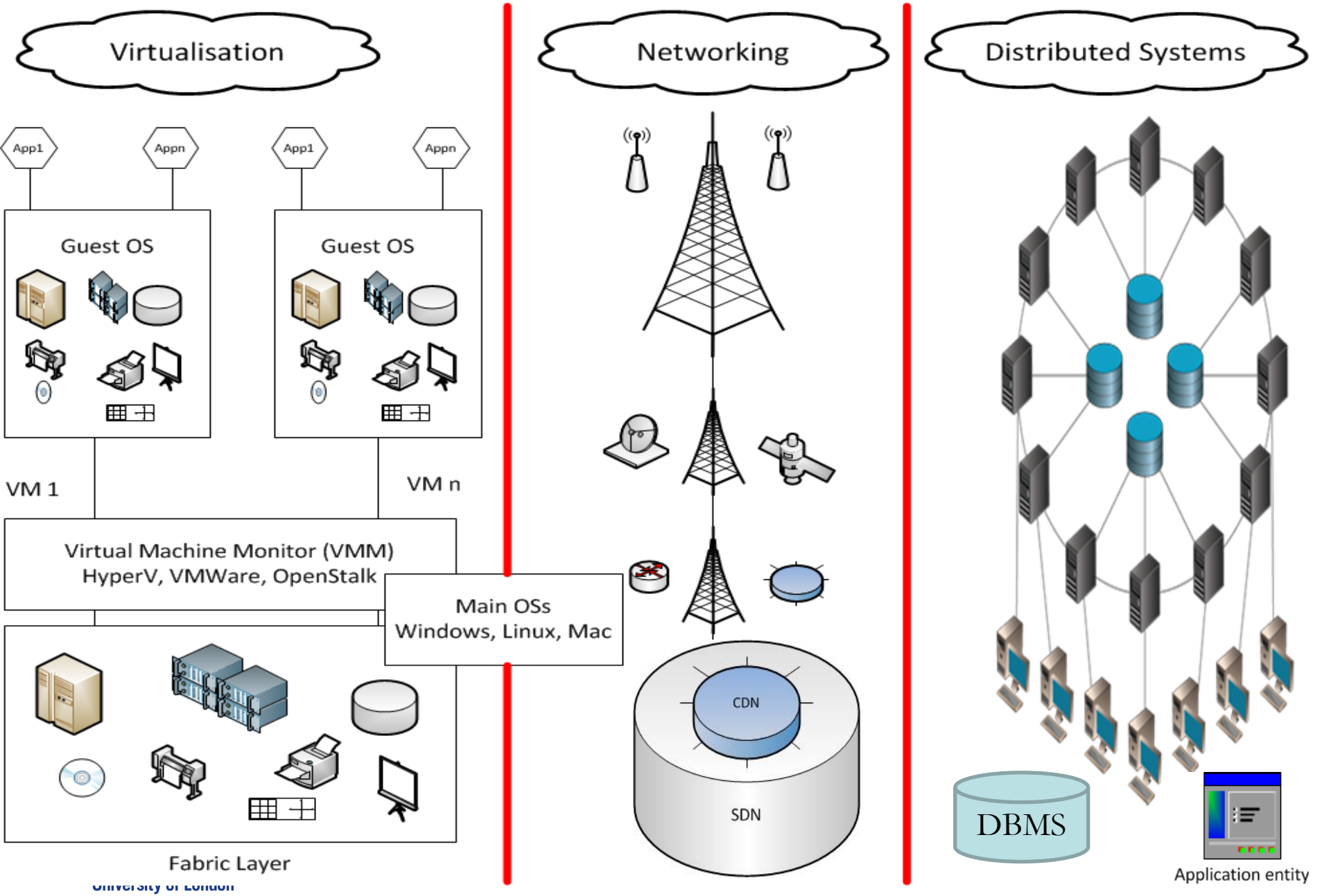


Source: Cloud Computing: by Thomas E., Zaigham M. and Ricardo P, 2013

Types of Cloud Computing Deployments



Cloud Computing Technologies



Cloud –Enabling Technologies

- These are technologies that facilitates the provisioning of computational resources as cloud-based services
- They are
  - Broadband networks provided under an internet architecture
  - Data centre technology
  - Virtualization technology
  - Web technology
  - Service-oriented technology
  - Multi-tenant technology
  - Resource broker and accounting technology
  - Information security technology

Cloud-based Computational Resources

- Physical servers
- Virtual servers
- Software applications
- Storage devices
- Network devices
- Services

Advantages of Cloud Computing

- Saves time of deployment
  - productivity
- Economies of scale
  - Cheaper than owning and maintaining a data centre
    - Cost of man power, electricity, administration, deployment
  - Unlimited storage and computing resources
- Reliability and availability
  - Redundancy, backups, recovery, snapshots of instance
  - Access to computing resources is location independent
  - support
- Scalability and resilience

Cloud Middleware

- Middleware is software that enables heterogeneous systems such as operating systems (Linux, Windows, Mac), applications, databases, web servers to interact and communicate with each
- It is the “middle” layer that facilitates hardware, operating systems and applications to communicate
- It provides security functionalities during this communication
- It supports running of complex distributed and cloud business applications using messaging services

Cloud Service Providers: Commercial

- Amazon Web Services (AWS)
  - EC2: Elastic Compute Cloud
  - S3: Simple Storage Service
- Microsoft Azure Cloud
  - Office 365
- Google Compute Engine and App Engine
  - Google Cloud Storage
- Oracle Cloud
  - Databases and database management systems
- Alibaba Cloud
  - Compute and Storage
- Tencent
  - Data streaming, Compute and Storage

Open Source Cloud Platforms and Technologies

- OpenStack
  - Started by Rackspace
  - Now supported by many IT companies
- Apache CloudStack
- Xen Hypervisor Cloud solution
- Microsoft HyperV Cloud solution
- OpenNebula

Class Task

- What do you think are some challenges with cloud computing implementations? Relate this to
  - Public clouds
  - Private clouds
  - Community clouds
- Name the biggest cloud provider in China and 3 other cloud providers in the world
  - What services do these cloud providers offer?
- What is your opinion about the future of cloud computing?
- What are the skills and knowledge a student needs to implement a cloud infrastructure?
- List as many as possible the technologies used for cloud implementations
- What is a virtual machine? What is virtualisation?