LECTURER: DR. TAI LE QUY

ANALYTICAL SOFTWARE AND FRAMEWORKS

TOPIC OUTLINE

UNIT 5

CLOUD COMPUTING PLATFORMS, ON-PREMISE SOLUTIONS, DISTRIBUTED COMPUTING

STUDY GOALS

- Understand concepts of cloud computing and most common service models
- Know differences between on-premise and edge computing
- Know different distributed computing models
- Understand the importance of virtualization in distributed computing models
- Know concepts of data streaming

EXPLAIN SIMPLY

1. What is Cloud Computing?
2. What does the acronym PaaS stand for? How does it differ from IaaS?
3. How does virtualization differ from containerization?

Cloud computing is the **on-demand delivery of IT resources**—applications, servers (physical or virtual), data storage, development tools, networking capabilities, and more—**hosted at a remote data center managed by a cloud services provider over the Internet** with **pay-as-you-go pricing**. Instead of buying, owning, and maintaining physical data centers and servers, you can **access technology services**, such as computing power, storage, and databases, **on an as-needed basis from a cloud provider**.

Source of the text: Amazon Web Services, Inc., 2021, IBM, 2021.

FIVE FUNDAMENTAL PROPERTIES OF CLOUD COMPUTING (NIST)

1. On-demand self-service

IT physical and virtual resources are dynamically assigned to, and released from, customers according to demand

2. Broad network access

Distributed IT resources are essentially provided through a powerful network that integrates all these resources

3. Measured service

The resource consumption is monitored, analyzed, and reported for both cloud providers and consumers to ensure transparency

4. Resource pooling

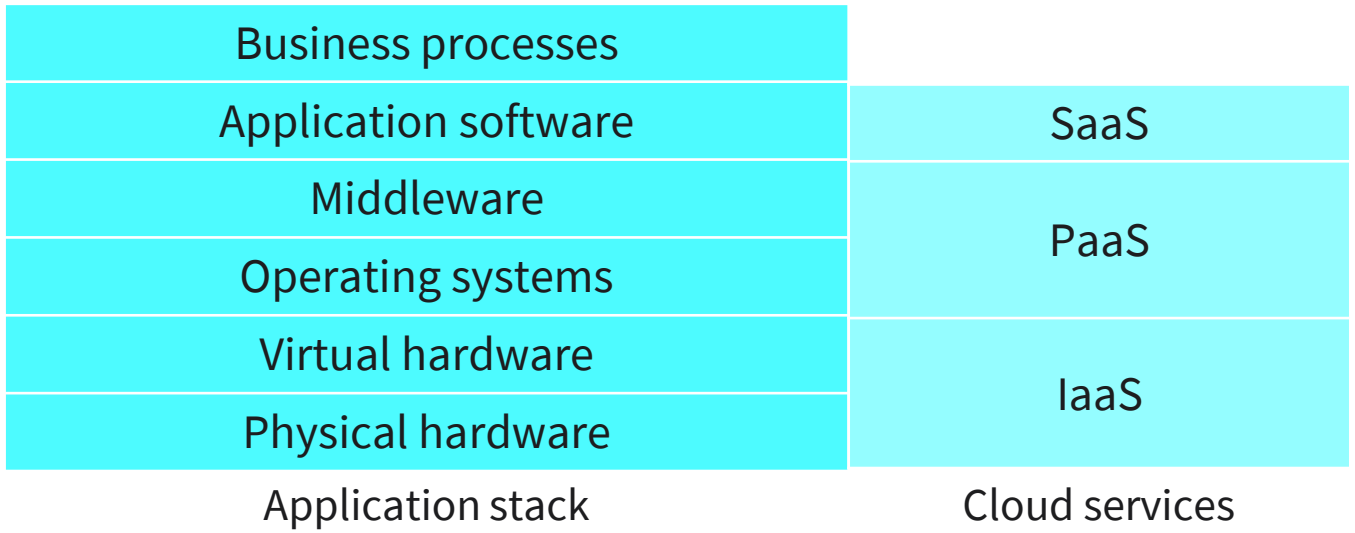
Resources are pooled in the cloud to allow for dynamic assignment, releasing, and reassigning of resources to multiple consumers

5. Rapid elasticity

Cloud providers offers resources to a very large number of consumers to reduce the cost to an individual consumer (economies of scale)

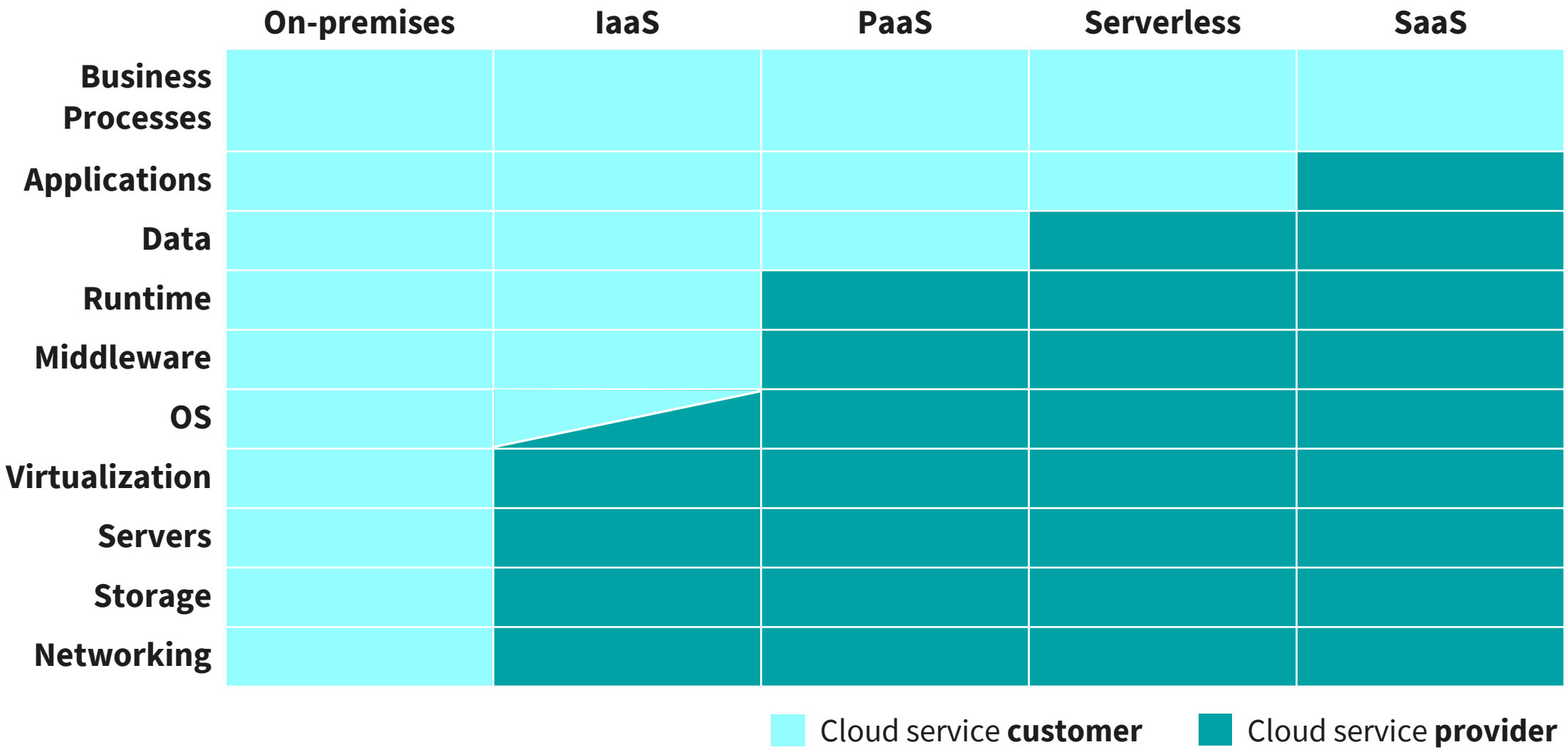
Source of the text: Course book DLMBDSA02, p. 109, Federal Office for Information Security, 2021.

CLOUD SERVICE MODELS—APPLICATION STACK AND ASSOCIATED CLOUD SERVICE MODELS



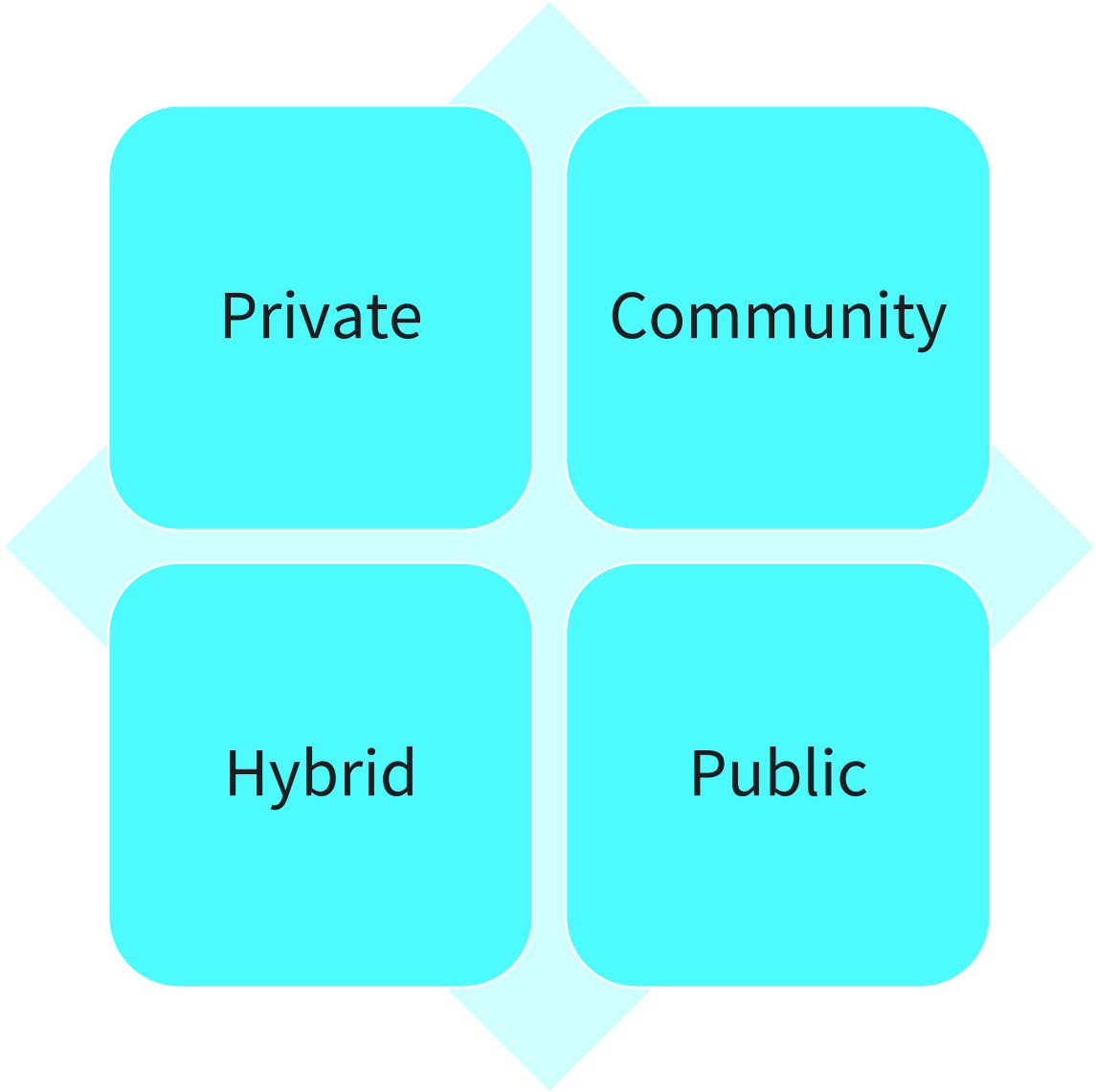
Source of the image: Course book DLMBDSA02, p. 110.

CLOUD SERVICE MODELS—WHO TAKES CARE OF WHAT?



Source of the image: Own creation based on IBM, 2021.

CLOUD CLASSIFICATION

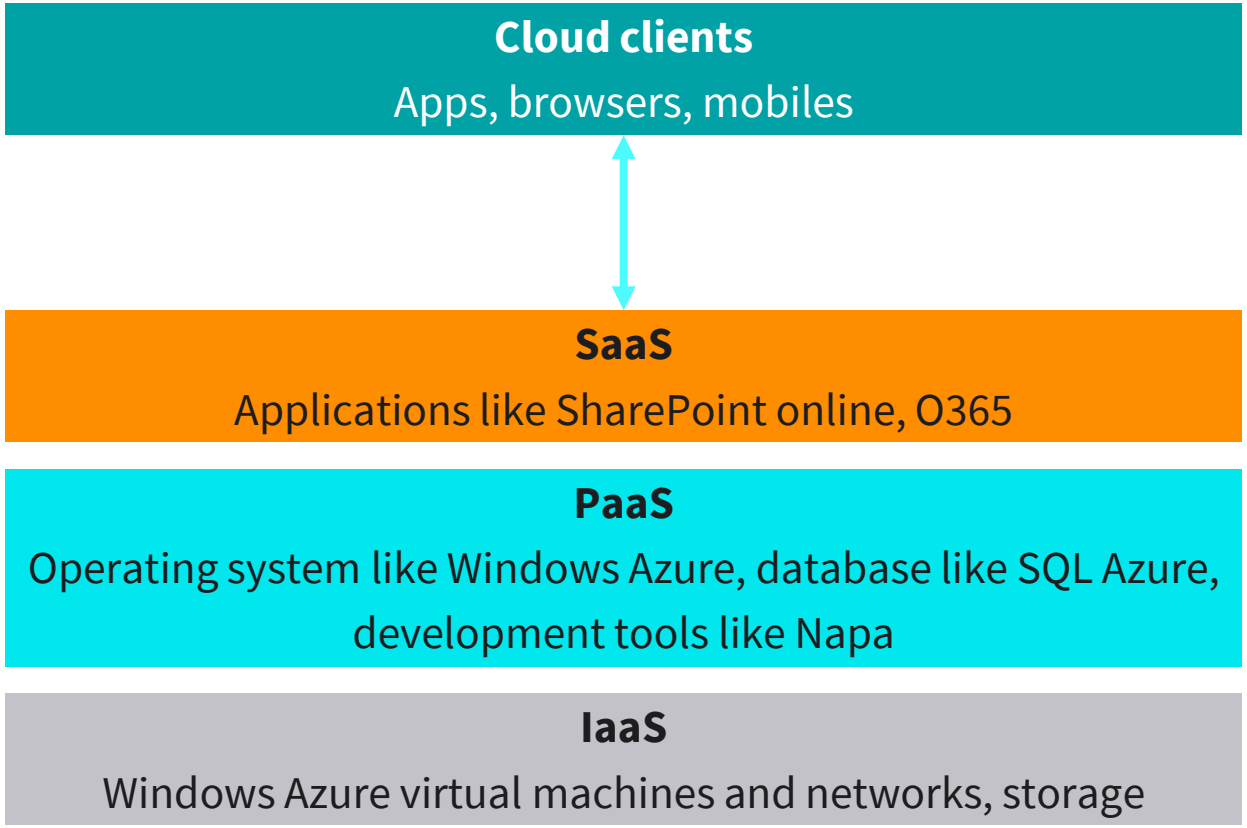


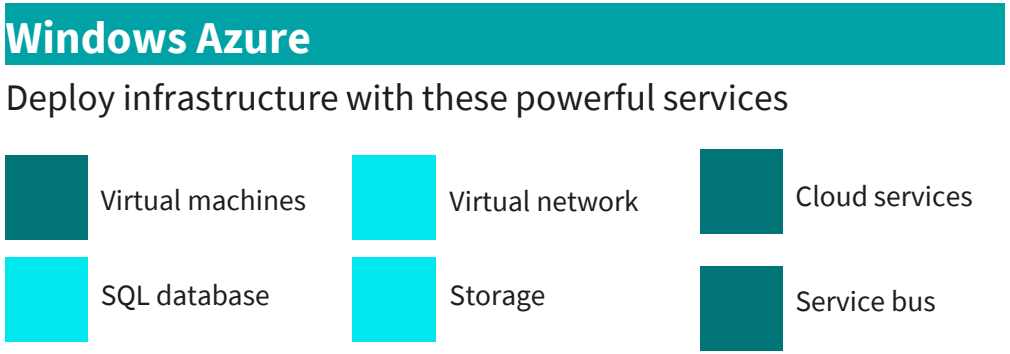
Source of the image: Course book DLMBDSA02, p. 113.

- Private Cloud
  - Dedicated setup for a single client or organization.
    - A private computing center but operated by someone else
    - Most secure approach – dedicated resources with dedicated access controls
    - Most expensive
  - Example: AppNexus platform
- Public Cloud
  - Infrastructure, platform, services, software, by cloud provider
    - Anyone can use the services, book and use what you need
    - Pay-per-use, subscription models
    - Used by end-users or small organizations
  - Example: Google AppEngine, IBM’s Blue, and Microsoft Azure

- Community Cloud
  - Allows several organization with common concerns or goals to share the infrastructure and related resources
    - Managed internally or by a 3rd party (hosted internally or externally)
    - Suited for organizations that work on joint projects
- Hybrid Cloud
  - Encompasses the best features of the public, private, and community deployment models
    - Dedicated own resources, e.g., private computer center operated by own enterprise IT
    - Use of private cloud (mission critical workloads)
    - Use of 3rd party/public cloud (less sensitive workloads)

MICROSOFT AZURE PROVIDES IaaS, PaaS, AND SaaS

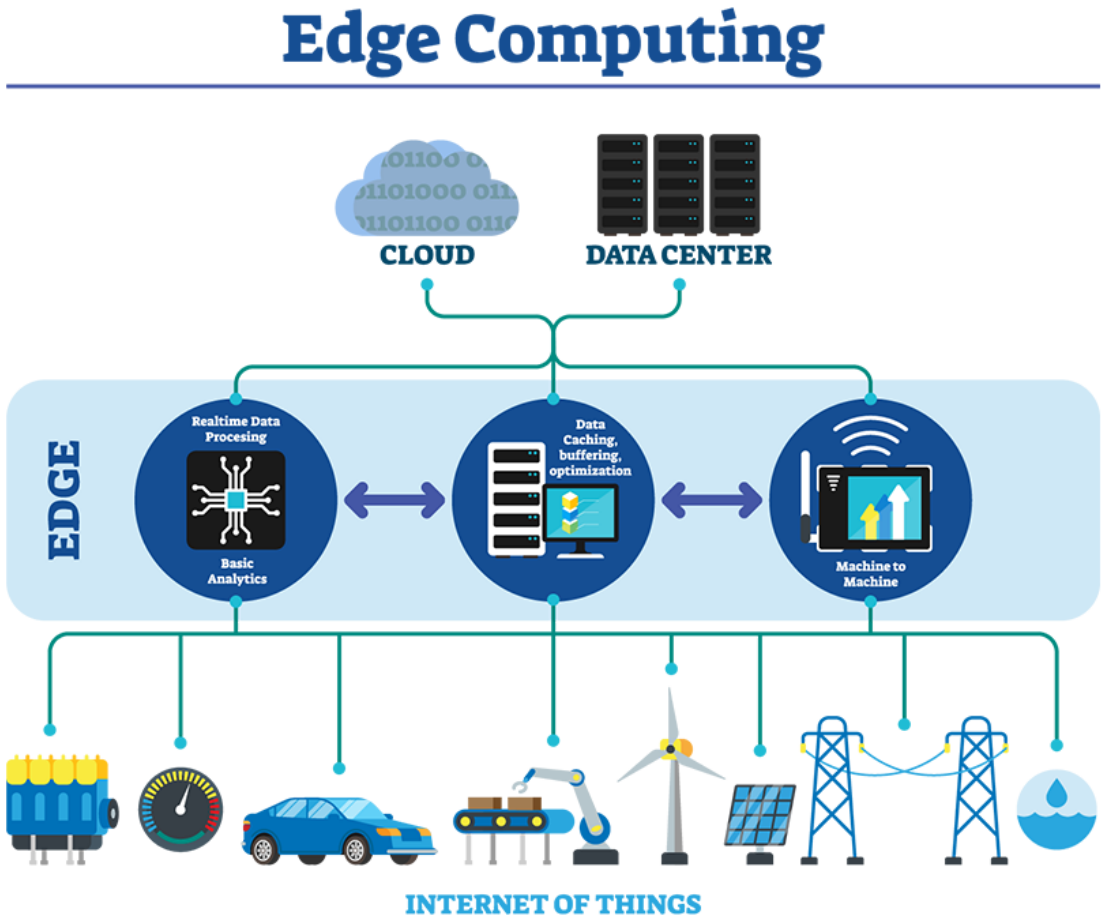




Source of the images: Course book DLMBDSA02, p. 116f.

EDGE COMPUTING

- Aims to move computing power and storage resources closer to users or data sources.
- “edge” refers to local computing resources that are geographically far away from the origin server edge in the internet (cloud).
- Computations are done inside the network and a summary report is sent once to the server.
- Reduces latency in processing, used bandwidth, and cost.



Source of image: https://www.cdebyte.com/news/531

CLOUD COMPUTING VS. ON-PREMISES COMPUTING VS. EDGE COMPUTING

	Cloud Computing	On-Premises Computing	Edge Computing
Location of computing resources	At a **cloud provider**, **outside of the organization**	Locally **within the organization** without interference of a 3rd party	At **edge devices (IoT) and gateways** that need **local computing power**, potentially geographically (far) **away from organizations/cloud providers computing infrastructure**
Benefits	**pay only for what you consume**, **flexibility** and **elasticity** as needed **without big upfront investments and risks**, easy to **react to varying capacity requirements**, benefit from **economies of scale**, **freedom to experiment**	**Full control** (data privacy) **and customization** to organization's needs, **high-security** needs **under control of the organization**	**Less** (long-)distance **communication** to centralized computing infrastructure **needed**, **low latency**, **less dependency on network availability and bandwidth**
Challenges	**Less control** and potentially not customizable to all needs of an organization; **needs trust**, e.g., in terms of **data protection**, **operational costs** at rising demands	**High initial cost and associated risk** with investment for **aging infrastructure or to much/little capacity**; **needs more effort, time, and IT experts** for implementation and maintenance inside the organization	**Maintenance of edge infrastructure and** software on the edge **devices**, order of magnitude: **more devices and systems to** manage and **maintain**, new **security** threats

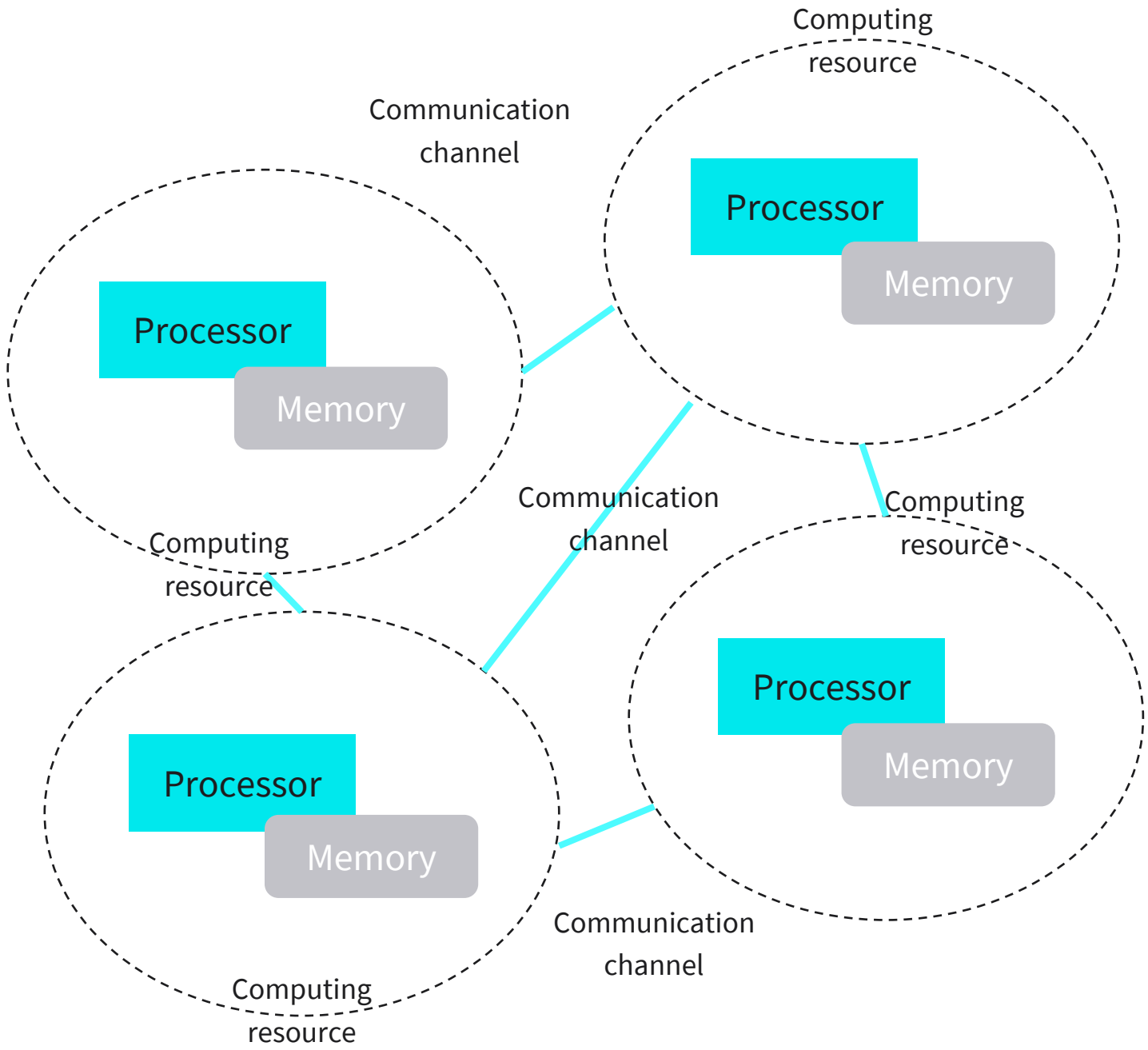
Source of the text: Course book DLMBDSA02, p. 115, Amazon Web Services, 2021, IBM, 2021, Microsoft, 2021.

UNIT 6

DISTRIBUTED COMPUTING

GENERAL DISTRIBUTED SYSTEM OVERVIEW

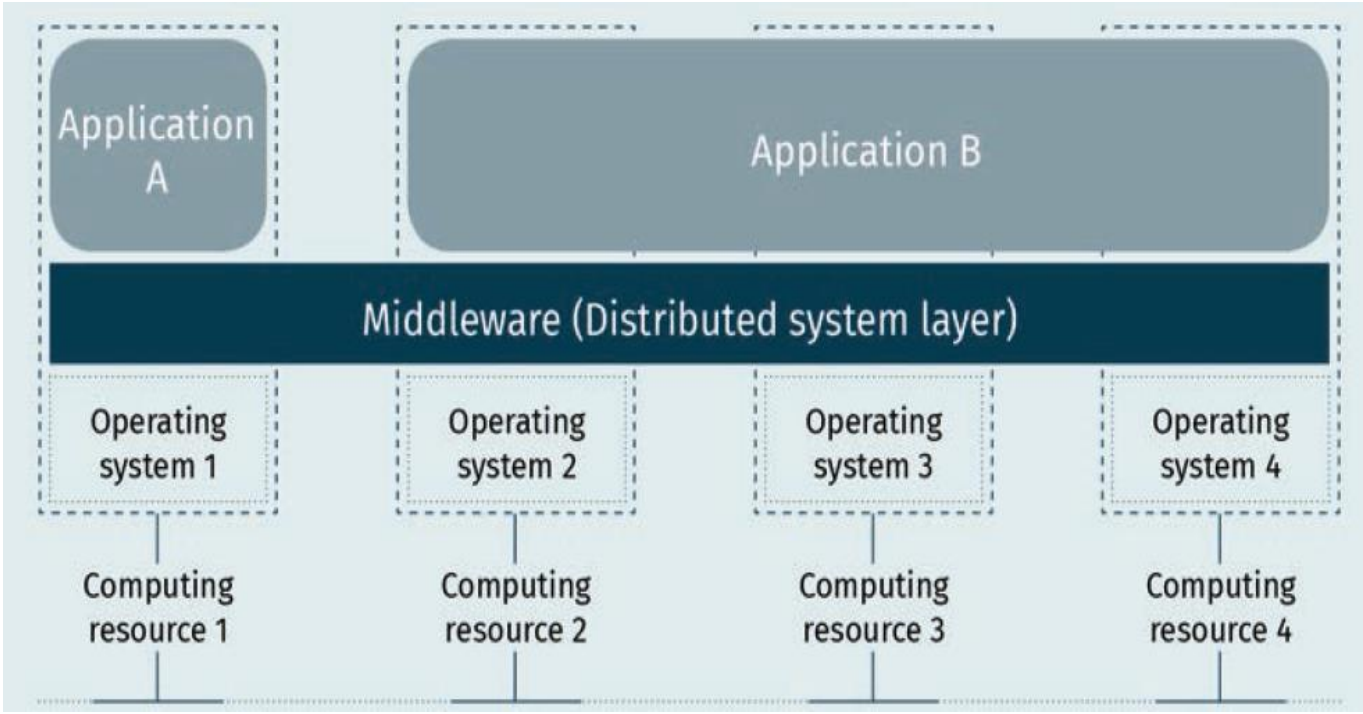
*„A distributed system is a collection of **autonomous computing elements that appears** to its users **as a single coherent system**.“*



Source of the text: Steen, M. van, & Tanenbaum, A. S., 2017.
Source of the image: Course book DLMBDSA02, p. 125.

EXECUTION OFF DISTRIBUTED APPLICATIONS

- Middleware is an intermediate software that lies in the middle between distributed components of a software application that run on different operating systems.
- The main role of middleware is to provide a uniform interface to the application software and to hide the heterogeneity of the various hardware components, operating systems, and communication protocols, and to hide the remote location of the computing resources.



- Communication between the computing resources is performed by message-passing methods like http request/response messages or remote procedure calls.
- CORBA (Common Object Request Broker Architecture) is a standard that enables applications, at different locations and developed by different vendors, to communicate in a network.

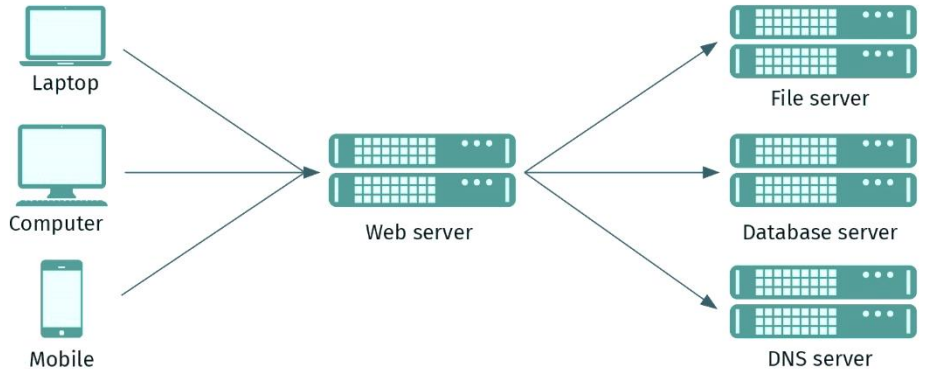
Source of the image: course book

ADVANTAGES OF DISTRIBUTED SYSTEM

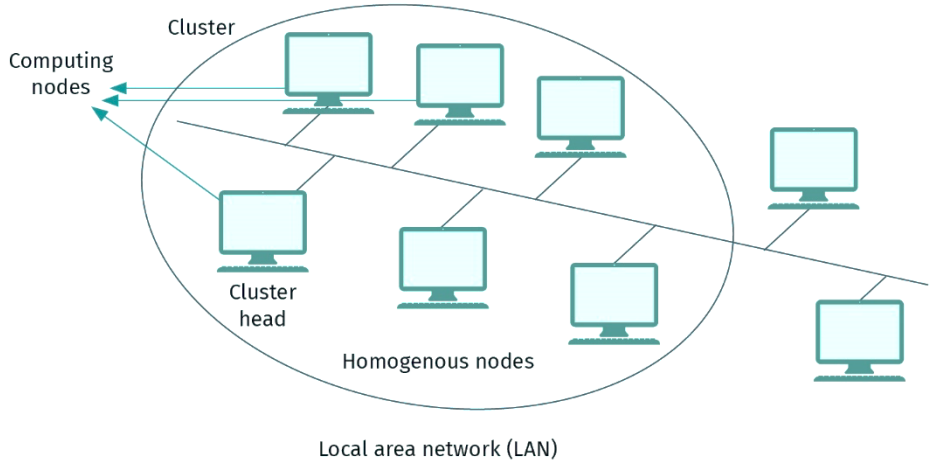
- **High performance**: boosting the computation performance of distributed applications.
- **Low cost**: organizations can perform tasks on their existing computers that have lower processing and storage capabilities.
- **High scalability**: The computational power of the distributed system can be extended easily by adding more resources.
- **High reliability** (fault tolerance): The impact of hardware and software failure is reduced by having redundant resources, such that if any resource fails in the distributed system, then another resource within the system continues to provide uninterrupted processing.

- A **webserver** is replying to requests by clients over a network and can itself be client to other servers.
- **Connected computers working together as a single integrated computing resource** act as a computer **cluster**. Nodes of a cluster use a homogenous network.

Client-Web-Server Architecture Overview



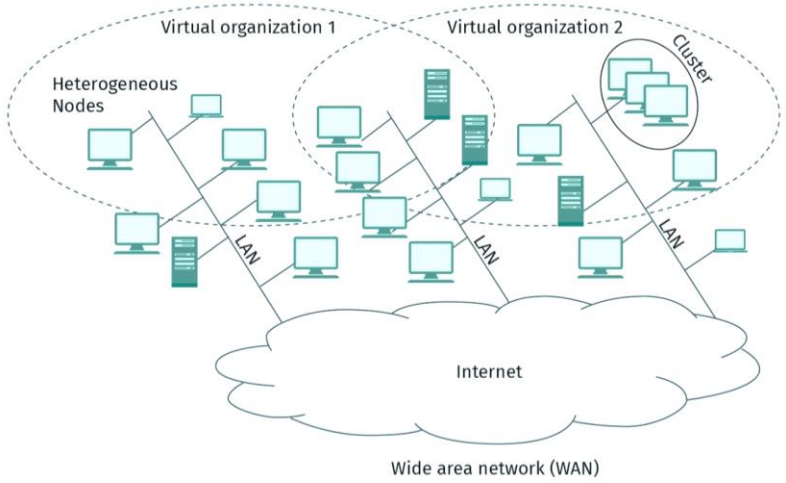
Cluster Architecture Overview



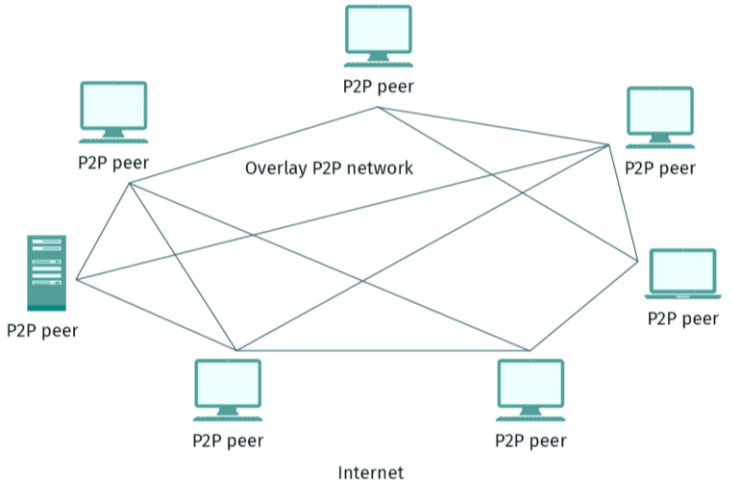
Source of the text: Course book DLMBDSA02, p. 126f.
Source of the images: Course book DLMBDSA02, p. 127f.

- A **grid computing** system is a set of **loosely connected heterogeneous resources** distributed across multiple administrative domains.
- A **Peer-to-Peer (P2P)** computing system is a network of computer nodes (peers) connected to each other through the internet that **allows** two or more nodes **to collaborate directly and spontaneously without the need of centralized coordination**.

Grid Architecture Overview

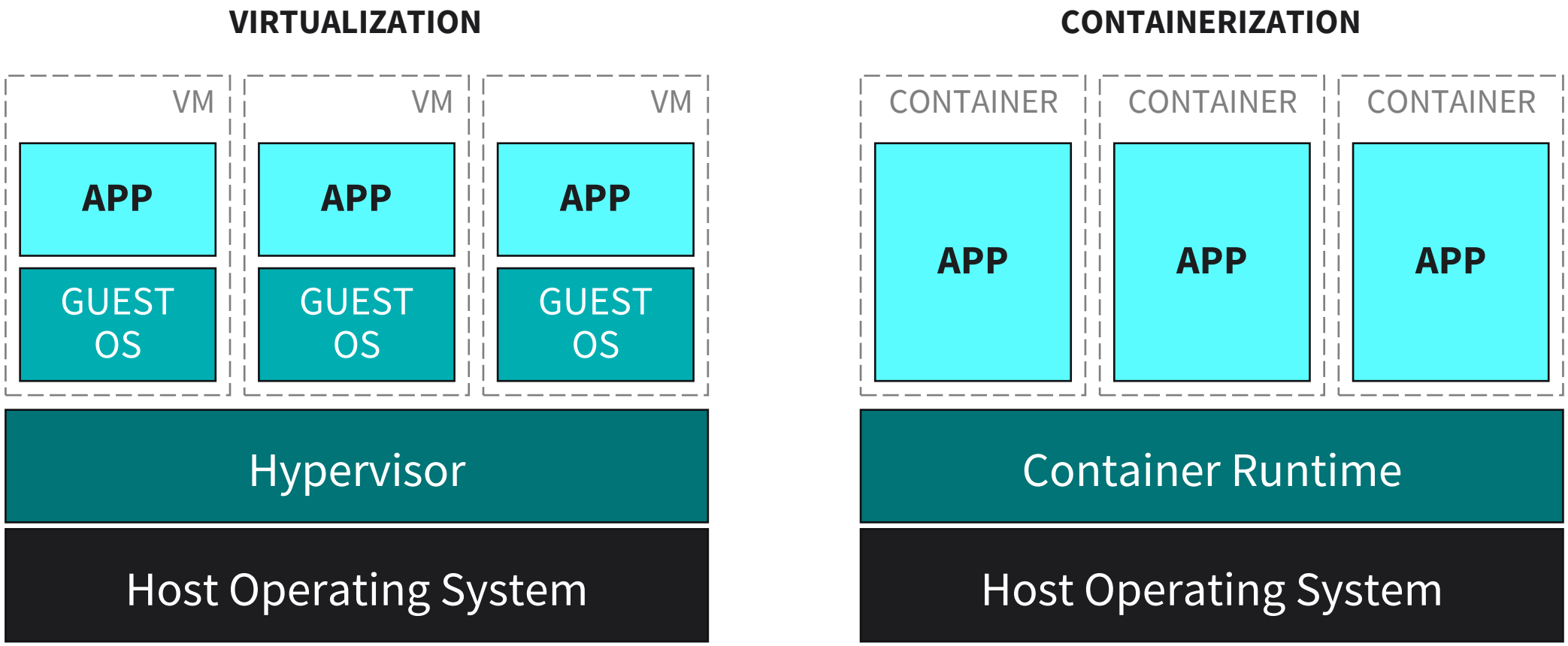


Peer-to-Peer Architecture Overview



Source of the text: Course book DLMBDSA02, p. 128f.
Source of the images: Course book DLMBDSA02, p. 129f.

VIRTUALIZATION AND CONTAINERIZATION



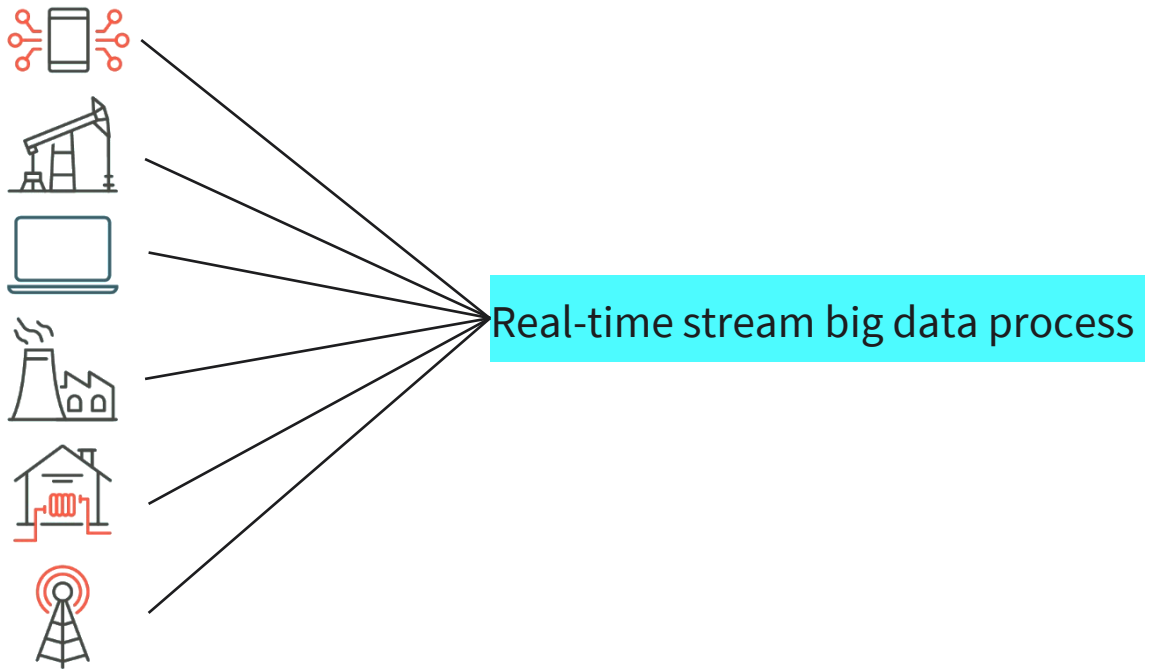
Source of the image: Own creation.

VIRTUALIZATION AND CONTAINERIZATION

Virtual Machine	Container
– "Hypervisor" separates virtual machine from hardware, eg., Vmware ESXi – Need to install full stack: • Operating system • Driver, addition software • Application – Feels like running entire computer • Windows inside OSX or Linux or Windows or… • OS + all applications	– Minimalist setup to run a single application – Aimed at "Micro-services" • Lightweight: just libraries, drivers for a single app • Uses the underlying host OS via Container software – Uses Container orchestration software to manage hundreds of specialized containers • Each for one Micro-Service, a single app (single purpose container)

Source of the text: Ulrich Kerzel, 2020

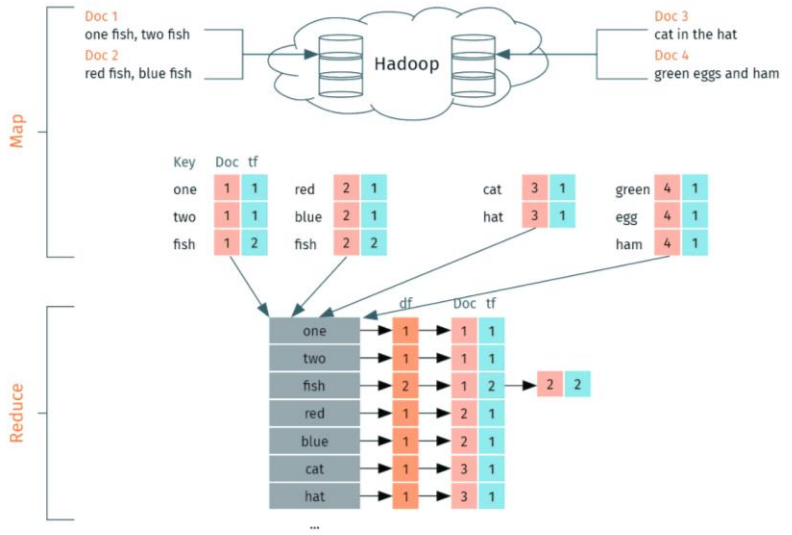
- **Instantly processing data** that gets **continuously generated as it is received** in (near) **real-time**
- Important where **fast decisions** must be made, e.g., handling sensor data from IoT devices, fraud detection, real-time alerts in healthcare, network attack alerts



Source of the text: Course book DLMBDSA02, p. 131f.
Source of the image: Course book DLMBDSA02, p. 132.

DATA STREAMING REPLACES BATCH PROCESSING WHERE REAL-TIME PROCESSING IS NEEDED

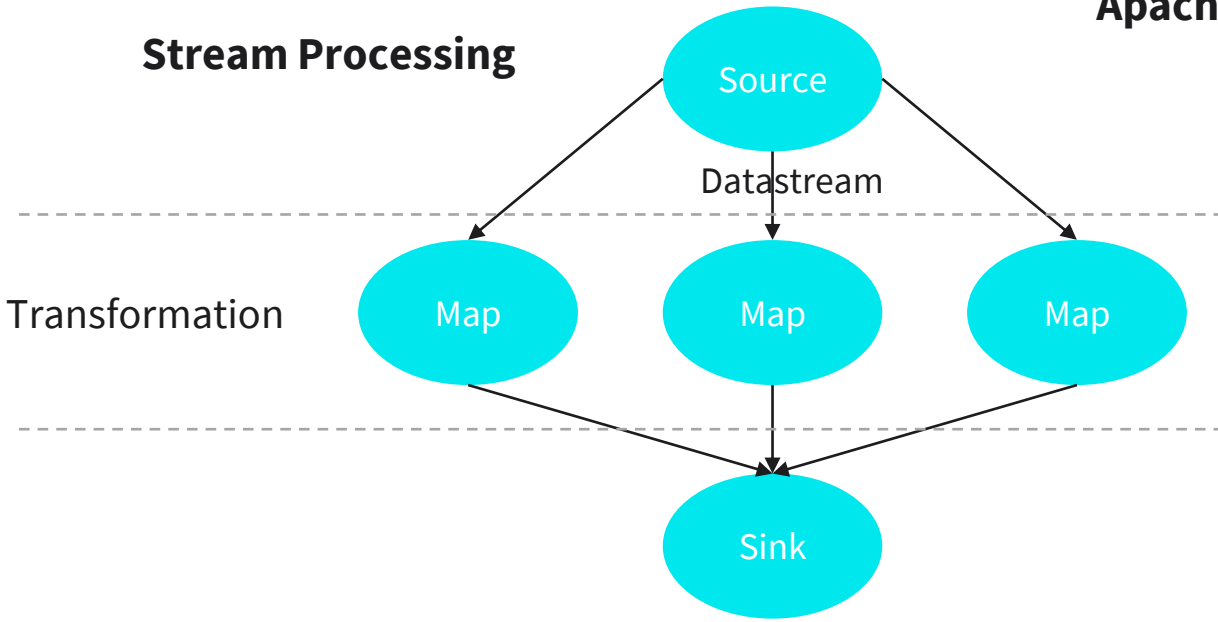
- In **batch processing** large volume of structured or unstructured **data collected over time gets processed all at once** (batch), typically completed simultaneously in non-stop sequential order.
- **Data streaming** handles an **infinite and continuous stream of data**, e.g., transactions of customers or sensors, **near-instantly** based on time- or data-driven windows (micro-batches).



Apache Hadoop simple example

Apache Flink

Stream Processing



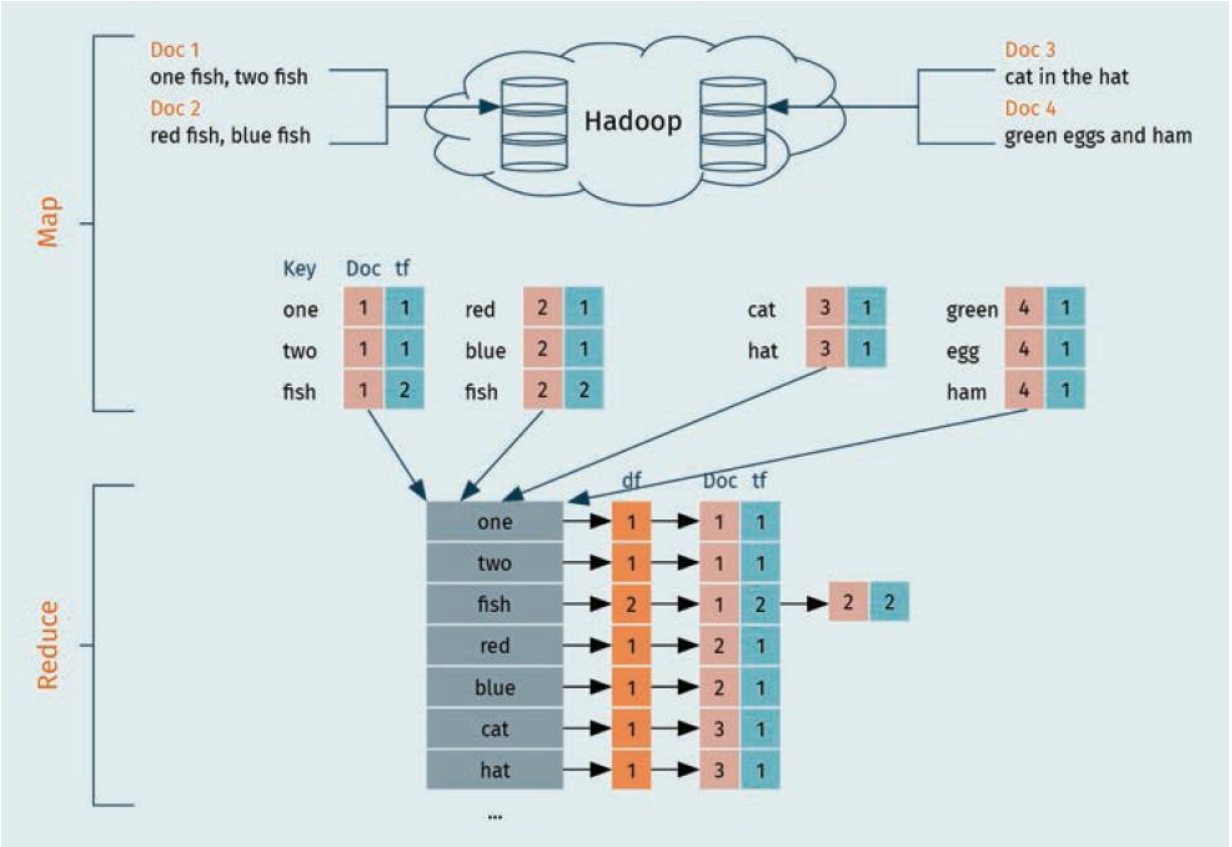
Source of the text: Course book DLMBDSA02, pp. 131–136.
Source of the images: Course book DLMBDSA02, pp. 132, 134.

APACHE HADOOP

- Apache Hadoop is a software framework for writing applications that rapidly process a large amount of data parallel to a large set of computing resources.
- Hadoop uses the MapReduce paradigm to process distributed data and documents over the cloud compute nodes.
- The two fundamental services of Hadoop are Hadoop Distributed File System (HDFS) and MapReduce framework.
- Hadoop breaks down input files into smaller blocks of equal size and distributes these blocks over multiple nodes in the Hadoop cluster.
- The Map- Reduce paradigm applies parallel processing of data distributed on multiple computing resources.
- The **map** function generates an (intermediate) set of pairs **<Key, Value>** for each computing resource
- **Reduce** function merge of all **Values** associated with the same **Key** in the (intermediate) sets.

Example: The requirement is to count

- term-frequency (tf), which is the number of repetitions of each word in each document (tf)
- document-frequency (df), which is the number of documents containing each word.



Source of the image: course book

REVIEW STUDY GOALS

- Understand concepts of cloud computing and most common service models
- Know differences between on-premise and edge computing
- Know different distributed computing models
- Understand the importance of virtualization in distributed computing models
- Know concepts of data streaming

SESSION 5

TRANSFER TASK

TRANSFER TASK

Your company is building a new retail system gradually replacing a legacy mainframe system to support new demands of your customers. The new system is planned to be cloud-native based on the Microservice architecture. It needs the ability to scale as necessary to adapt to varying demand.

What cloud model , **virtualization** or **containerization approach**, and which **data processing model** (batch or streaming) could support your approach? Why?

TRANSFER TASK
PRESENTATION OF THE RESULTS

Please present your results.

The results will be discussed in plenary.

1. Suppose you have two types of applications: legacy applications that require specialized mainframe hardware and newer applications that can run on commodity hardware. Which cloud deployment model would be best for you?
   a) Hybrid cloud
   b) On-demand self-service
   c) Private cloud
   d) Public cloud

2. According to the NIST, cloud computing provides five fundamental properties. Which of these properties ensures transparency between the cloud provider and the cloud consumer and enables pay-per-use pricing?
   a) Resource pooling
   b) Broad network access
   c) Rapid elasticity
   d) Measured service

3. You are developing an application and want to focus on building, testing, and deploying. You do not want to worry about managing the underlying hardware or software. Which cloud service type is best for you?

   a) Infrastructure as a service (IaaS)
   b) Platform as a service (PaaS)
   c) Software as a service (SaaS)
   d) Both (IaaS) and (SaaS)

LIST OF SOURCES

Amazon Web Services, Inc. (2021). *What is Cloud Computing*. Amazon Web Services, Inc. https://aws.amazon.com/what-is-cloud-computing/

Federal Office for Information Security (2021). *Cloud Computing Basics*. Federal Office for Information Security. https://www.bsi.bund.de/EN/Topics/CloudComputing/Basics/CloudComputing_Basics.html

IBM (2021). *What is Cloud Computing?* https://www.ibm.com/cloud/learn/cloud-computing

Microsoft (2021). *What Is Cloud Computing? A Beginner's Guide | Microsoft Azure*. https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/

Steen, M. van, & Tanenbaum, A. S. (2017). *Distributed systems* (Third edition, Version 3.01). https://www.distributed-systems.net/index.php/books/ds3/