

SECTIONS 1.8 – 1.10

Distributed Systems, Kernel Data Structures & Computing Environments

PRESENTED BY

Prof. Mostafa Elhosseini

Professor of Smart Systems Engineering

Three Topics

1.8

Distributed Systems

الأنظمة الموزعة

How computers **cooperate**

1.9

Kernel Data Structures

هياكل بيانات النواة

How the OS **organizes**
information

1.10

Computing Environments

بيئات الحوسبة

Where operating systems run

أين وكيف تُستخدم أنظمة التشغيل

Same OS principles — from a tiny embedded controller to a huge public cloud.

SECTION 1.8

Distributed Systems

الأنظمة الموزعة

How several computers work together

What Is a Distributed System?

Physically separate computers, **connected by a network**, sharing resources. نظام موزع

Machines may be **heterogeneous** — different hardware, OS, or capabilities.

غير متجانسة

ANALOGY · A UNIVERSITY

Student records

Email

Learning system

Research computing

Physically **different machines** — experienced as **one environment**.

Four Benefits

Computation speed

Work shared across machines

More functionality

Each machine adds services

Server A → database
Server B → file storage
Server C → web service
Server D → authentication

Data availability

Data reachable from many places

Reliability الاعتمادية

No dependence on one machine

One computer = **local resources**

Distributed = **resources spread out**

Some operating systems **generalize network access as a form of file access**, hiding the details in the network interface's device driver

- The user may simply see: **/shared/report.pdf**
- But physically, the file may actually be on another computer across the network - **Abstraction — التجريد**

Network and Protocol

Network شبكة

A communication path between two or more systems

Laptop ↔ Wi-Fi router ↔ Server

Protocol بروتوكول

Rules for communicating — e.g. **TCP/IP**

لغة وقواعد الاتصال بين الأجهزة

- Network Adapter = NIC
- hardware interface that connects the computer to the network.
- How the computer connects to the network

Highway ramp = Network Adapter

Lets each person enter or leave the road.

ANALOGY

Network = **road**

Protocol = **traffic rules**

Adapter = **on/off ramp**

Connected but **no shared rules** → computers can't understand each other.

Networks by Distance

PAN Personal — phone ↔ Bluetooth headset

شبكة شخصية

LAN Local — room, building, campus

شبكة محلية

MAN Metropolitan — a city

شبكة على مستوى مدينة

WAN Wide — countries, continents · **the Internet**

شبكة واسعة

PAN person



LAN campus



MAN city



WAN world

Transport Media

The physical or wireless path the signals **travel through**. وسيط النقل

Copper wire

Fiber optic

Satellite

Microwave

Cellular

Radio

Infrared

Protocol → HOW

Medium → WHERE

Network OS vs Distributed OS

Network OS

نظام تشغيل شبكي

Computers **know they are separate** A knows it is A · B knows it is B Share files, send messages

Every computer is still an independent machine.

Distributed OS

نظام تشغيل موزع

Machine boundaries **hidden** Behaves like **one integrated system**

Many computers that try to look like one system

Keyword: **Autonomy** — high in Network OS, low in Distributed OS

الاستقلالية

SECTION 1.9

Kernel Data Structures

هياكل بيانات النواة

How the OS organizes information inside

The Kernel Must Track Everything

INFORMATION TO KEEP

Processes

Memory

Files

Devices

Scheduling

Free blocks

CONTAINERS IT USES

Lists

Stacks

Queues

Trees

Hash Maps

Bitmaps

Choose the container by how you **store, search, insert, or remove**.

Array vs List

Array مصفوفة

main memory itself is constructed as an array

Indexed positions **Very fast direct access**
Each element is **accessed directly**

`item[0]` `item[1]` `item[2]` ...

List قائمة

Items linked in sequence **Easy insert/delete** ·
search $O(n)$ Data + link/pointer to another node

The building blocks for stacks and queues

- Address = start address + (index × item size)
- It does not need to inspect item 0, then item 1, then item 2, etc.
- The access time does not grow significantly just because the array contains more elements.
- Less convenient insertion/deletion

ANALOGY

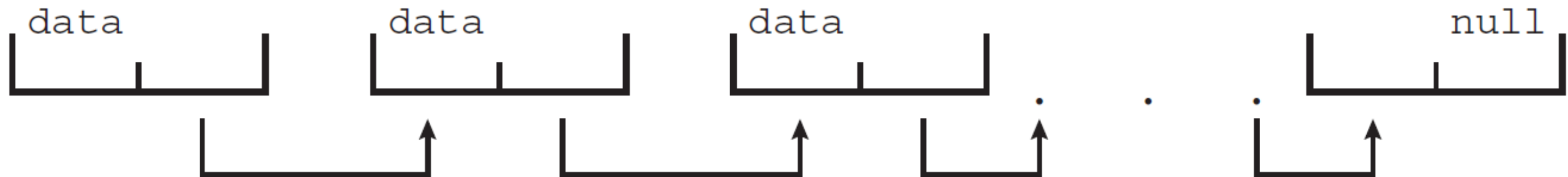
Cinema seats — seat 50 found directly

People holding hands — follow the chain to #50

Singly Linked List

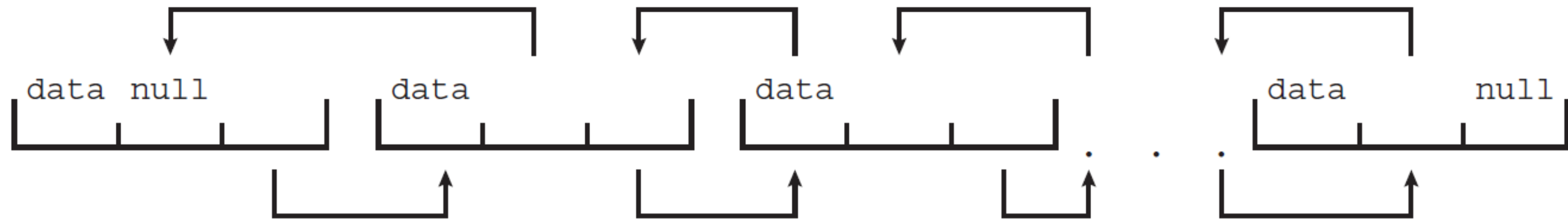
- Each element in the list is called a **Node** — عقدة.
- Every node has two parts:
- **Data** — البيانات the actual value stored in the node and
- **Next Pointer** — مؤشر للعقدة التالية stores the address of the next node.

Each item **points to its successor**; the **final pointer is null**.

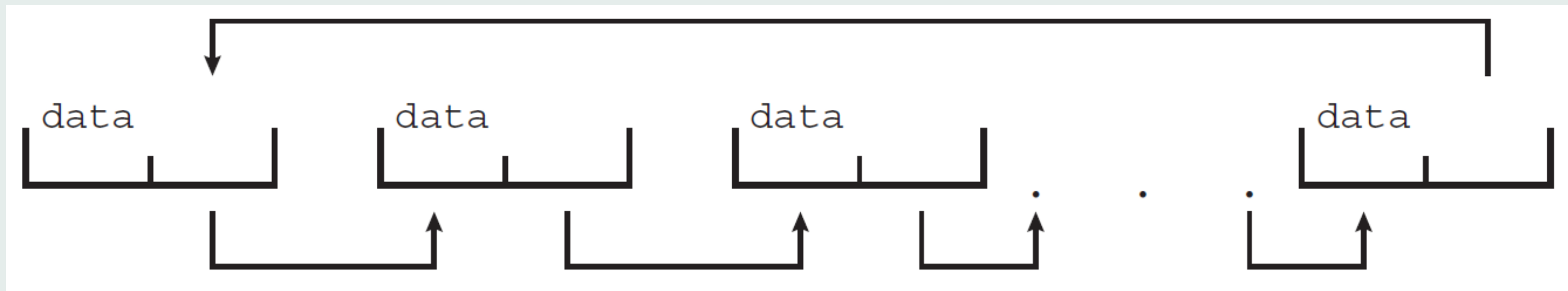


Doubly and Circularly Linked Lists

A given item can refer either to its **predecessor** or to its **successor**.



The **last element refers back to the first element** rather than to null.



Three Kinds of Linked List

Singly

قائمة أحادية الاتجاه



data + **next** pointer · forward only — like train cars

Doubly

قائمة ثنائية الاتجاه



next + **previous** · successor & predecessor

Circular

قائمة دائرية



last → **first** · for cyclic, repeated work

Stack vs Queue

Stack

مكدس

LIFO — Last In, First Out

Plates: A, B, C → **C** comes off first

push add top · **pop** remove top

OS use: **function calls** — parameters, locals, return addresses

Queue

طابور

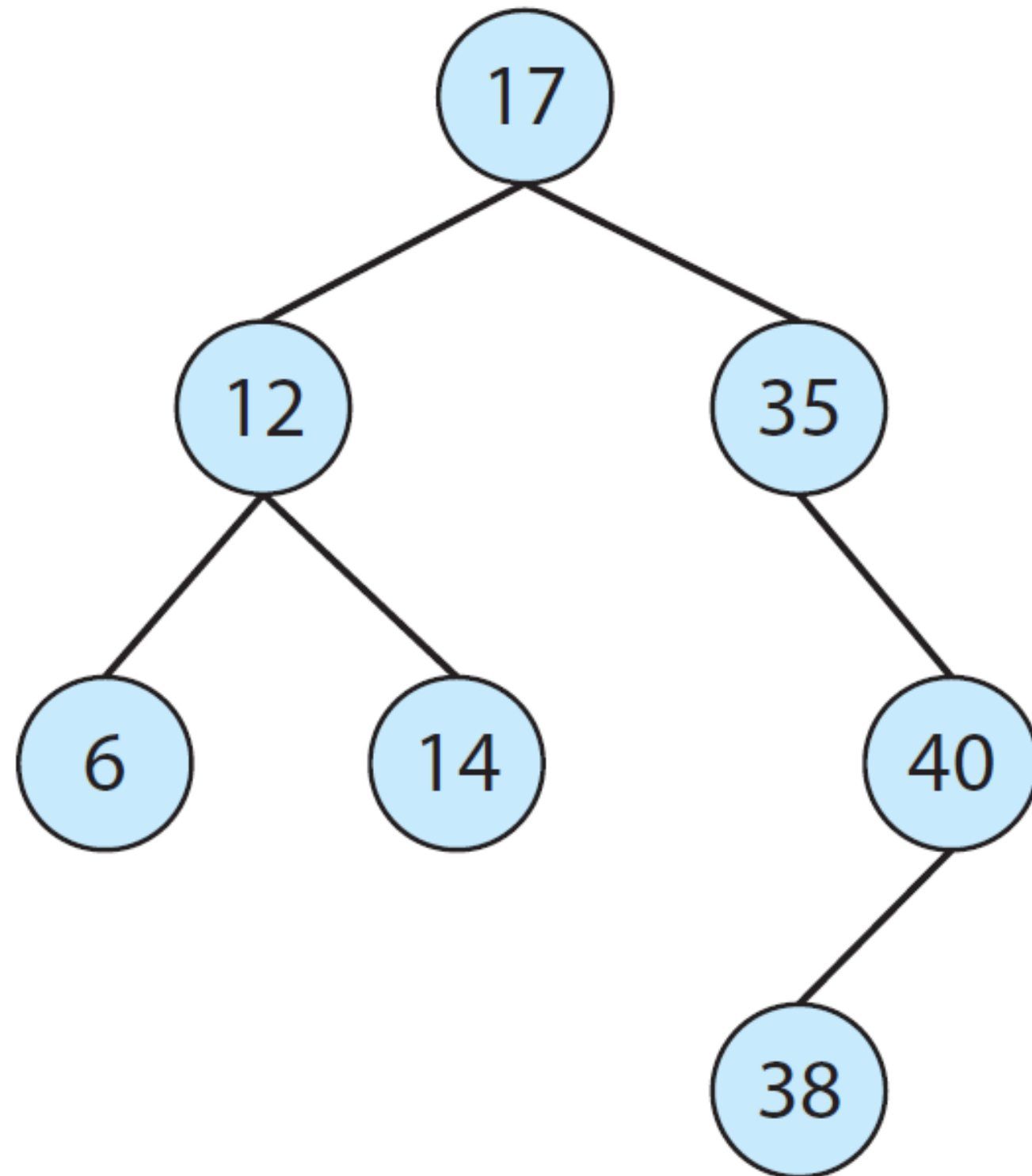
FIFO — First In, First Out

Checkout line: Ahmed, Sara, Ali → **Ahmed** leaves first

OS use: **print jobs** · processes **waiting for CPU**

Stack = **plates** · Queue = **people in line**

Trees and Binary Search Trees



A **tree** represents data **hierarchically**, linked through **parent–child relationships**. A **general tree** allows unlimited children; a **binary tree** allows at most two.

A **binary search tree** adds an ordering: **left child $\leq$ right child**.

Unbalanced, worst case

$O(n)$

Balanced, at most $\log n$ levels

$O(\log n)$

Linux uses a balanced binary search tree — a **red-black tree** — as part of its **CPU-scheduling algorithm** (Section 5.7.1).

Hash Functions (دوال التجزئة) and Hash Maps (خرائط التجزئة)

A **hash function** takes data as input, performs a **numeric operation** on it, and returns a value used as an **index into a table**.

A table with 10 locations:

0 1 2 3 4 5 6 7 8 9

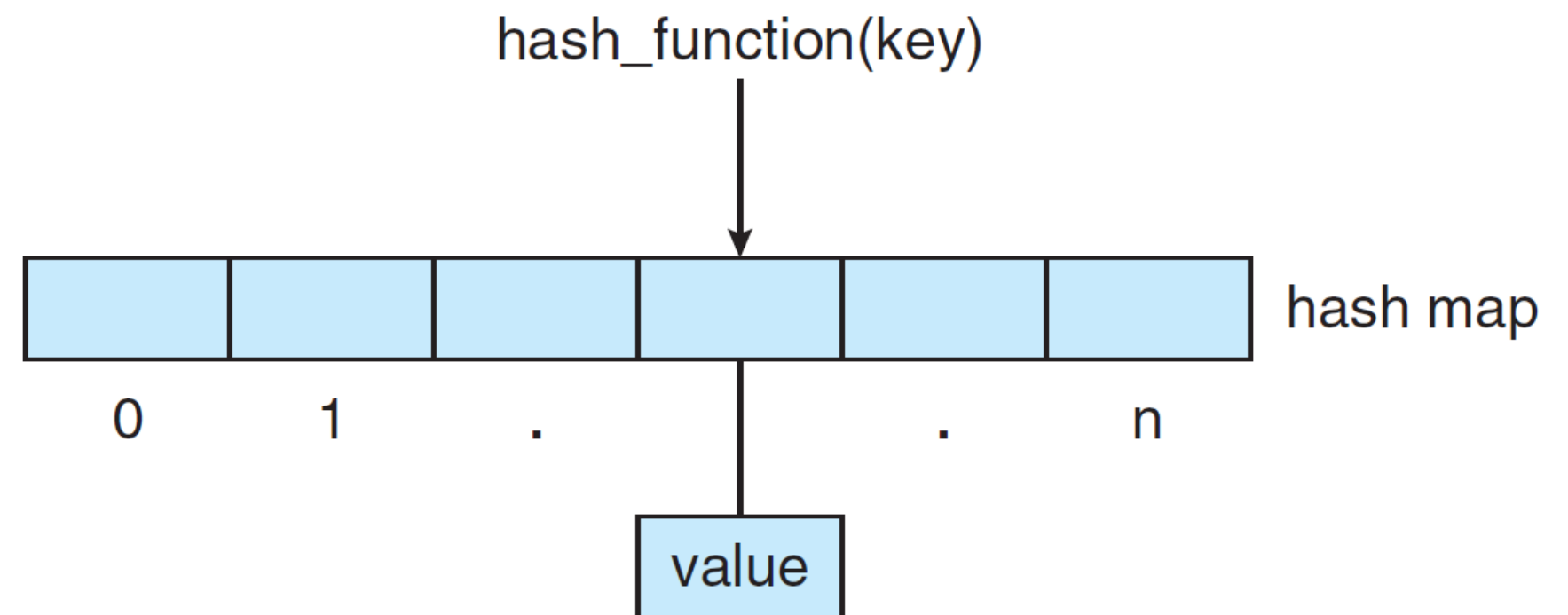
and our key is: Student ID = 12345

A very simple teaching example might be:

$\text{hash}(12345) = 5$

So instead of searching the entire table, we jump directly to: Index 5

Key $\rightarrow$ Hash Function $\rightarrow$ Index $\rightarrow$ Value



Searching a list of size n can cost **$O(n)$** comparisons; retrieval through a hash function can be as good as **$O(1)$** .

Hash Functions (خرائط التجزئة) and Hash Maps (دوال التجزئة)

Without hashing:

Search every guest name one by one.

With hashing:

Guest ID → calculate room/index → go directly there.

A **hash collision** — two inputs producing the same output — is accommodated by a **linked list at that table location** . More collisions, **less efficiency**.

A **hash map** associates **[key:value]** pairs: **password authentication** hashes the user name to retrieve the stored password for comparison.

Bitmap

A **bitmap** is a string of **n binary digits** representing the status of n items: **0 means the resource is available, 1 means it is unavailable**.

One bit per resource — 0 = available, 1 = unavailable.

خريطة البتات



CLASSIC USE

Disk management one bit per disk block

ANALOGY · PARKING LOT

0 = empty space 1 = occupied

Why not a full integer? It **wastes memory** — one bit answers “free or occupied?”

Linux Uses These — for Real

Linked lists

Queues

`kfifo`

Red-black trees

Data structures are **the tools the kernel uses** to organize its work.

SECTION 1.10

Computing Environments

بيئات الحوسبة

Where do operating systems actually run?

Traditional Computing

Batch

المعالجة الدفعية

Jobs processed in groups

1,000 payrolls overnight

كشف الرواتب أو رواتب الموظفين

little user interaction

Batch = collect jobs → process them
as a group → return results later

Interactive

الحوسبة التفاعلية

Waits for input and responds

user types → computer

responds

- text editor
- web browser
- ATM
- command line
- graphical applications

Time Sharing

تقاسم الوقت

CPU switches fast between

users $P1 \rightarrow P2 \rightarrow P3 \rightarrow P1 \dots$

Time sharing creates
concurrency (التزامن الظاهري),
not necessarily true parallelism.

Thin client — little local processing; relies on central servers

عميل خفيف

Time sharing creates the **feeling** that many tasks run at once.

Mobile Computing

SPECIAL HARDWARE

GPS Accelerometer مقياس التسارع

Gyroscope الجيروسكوب

location · orientation · motion · AR

CONSTRAINTS

Limited battery Small size · less cooling

Limited storage Wireless connectivity

AR = Augmented Reality — الواقع المعزز

Digital information is added on top of the real world.

Example:

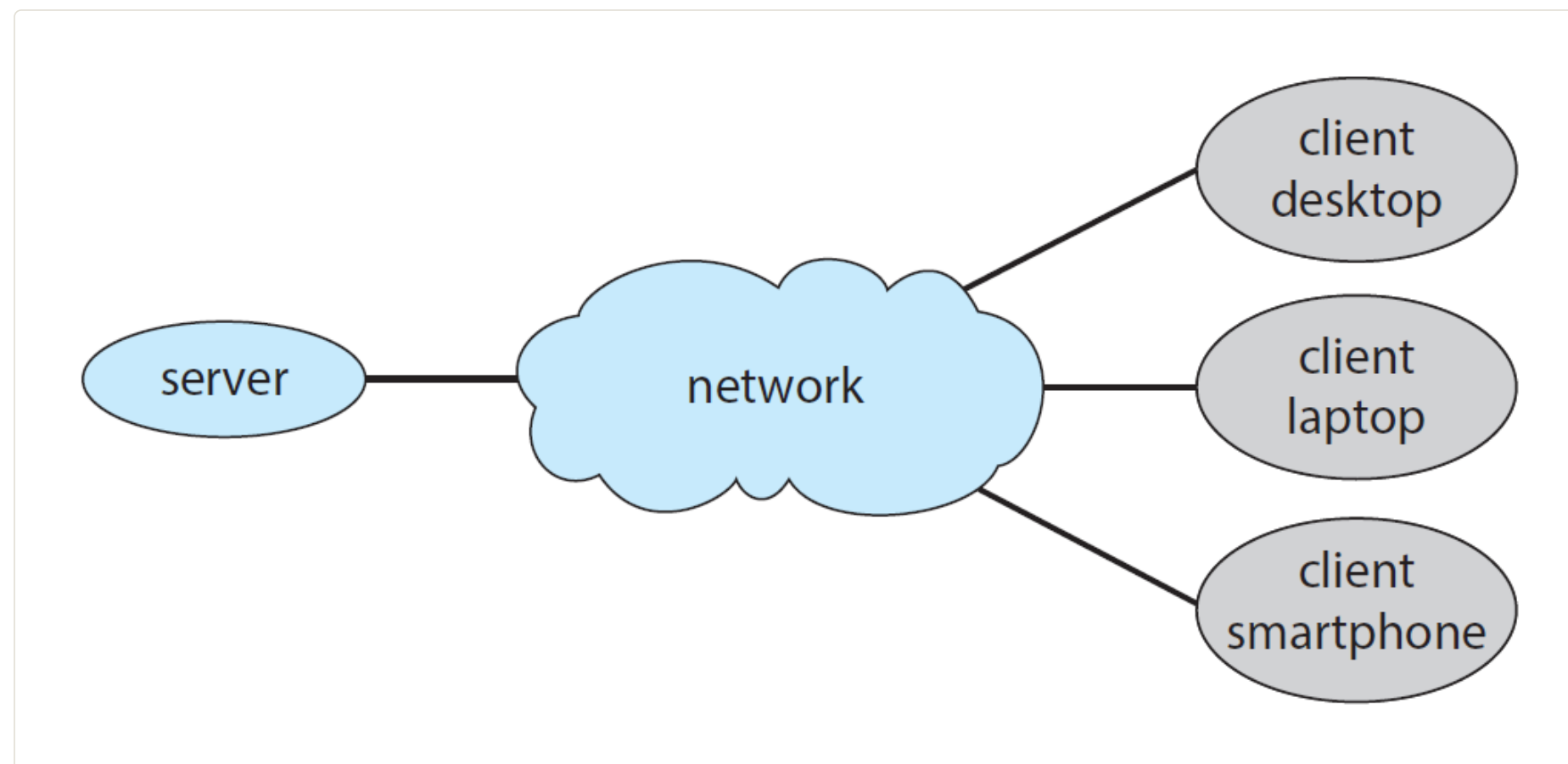
Camera shows the room → app adds a virtual chair or navigation arrow.

Key priority: **power efficiency** · Families: **iOS, Android**

كفاءة الطاقة

Client–Server Computing

Server systems satisfy requests generated by client systems — a specialized form of distributed system.



Compute-server system

Offers an interface for clients to **request an action** ; the server **executes it and returns the results** — for example, a **database** answering data requests.

File-server system

Offers a **file-system interface** for clients to **create, update, read, and delete** files — for example, a **web server** delivering pages and multimedia to browsers.

Client–Server Computing

Client

عميل

Requests a service · e.g. browser

Server

خادم

Provides the service · e.g. website



Compute server

“Find students with GPA > 3.5” → runs query → results

File server

Create · read · update · delete files

“Do **work** for me”

“Manage **files** for me”

Peer-to-Peer Computing

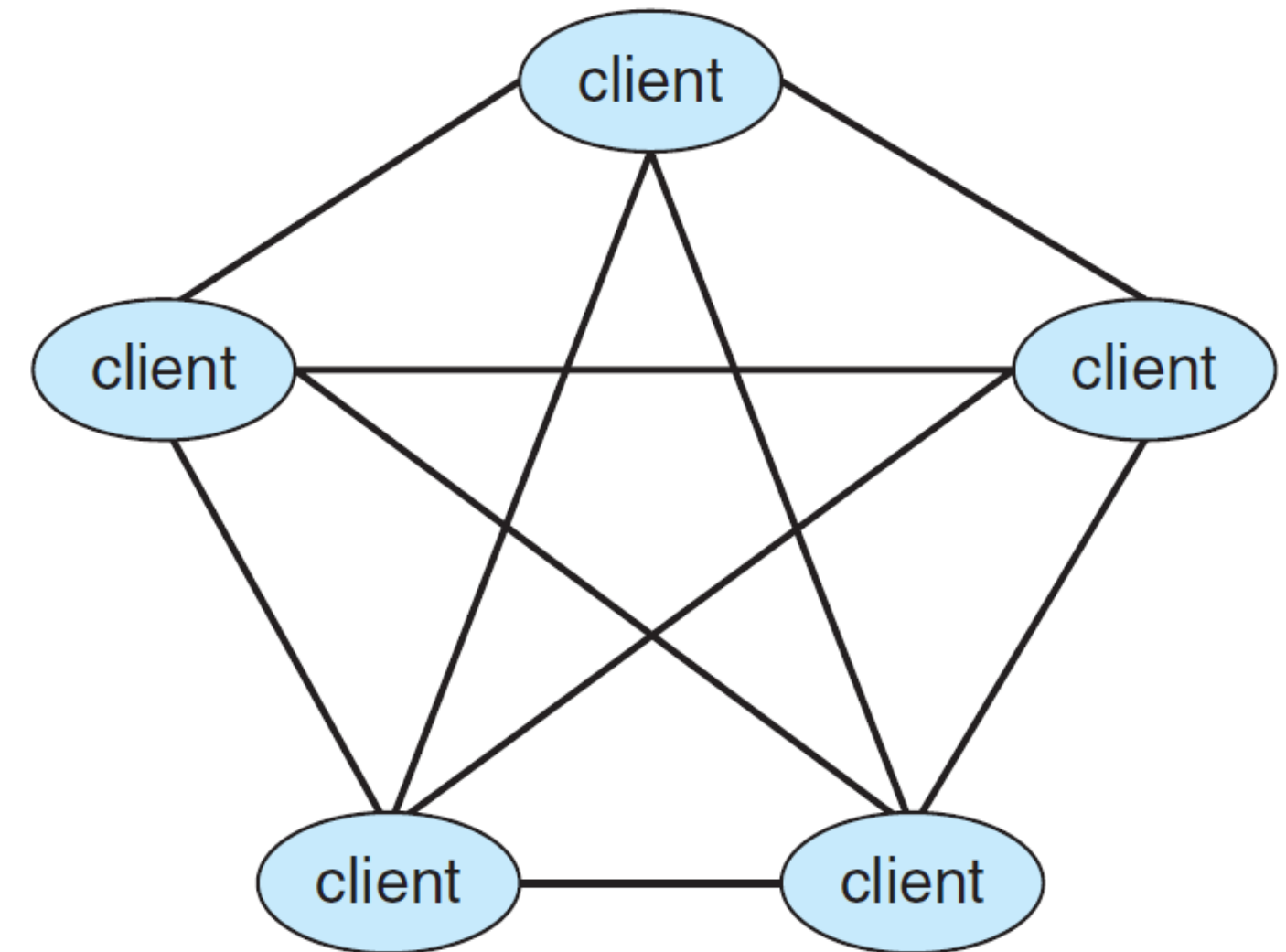
Clients and servers are **not distinguished**: **every node is a peer** and may act as client or server depending on whether it **requests or provides** a service. Where a client–server system has a **server bottleneck**, a P2P system can **spread services across many nodes** .

Centralized lookup service

A joining node **registers its service**; requesters **query the lookup service** , then **communicate directly**. Napster indexed peer files this way.

Discovery protocol

No central lookup: a peer **broadcasts its request to all nodes** , and providers **respond directly** — the **Gnutella** approach.



Peer-to-Peer (P2P)

Each **peer** is a client sometimes, a server sometimes.

الند للند

A downloads from B



A uploads to C



A = consumer + provider

Centralized lookup

خدمة بحث مركزية

Directory: “Who has file X?” → “Peer B” Then A ↔ B
directly · **Napster**

Discovery protocol

بروتوكول الاكتشاف

No directory — peer broadcasts Others answer directly
· **Gnutella**

No dependence on **one central server**.

Cloud Computing

Computing resources delivered **over a network as services** — an extension of virtualization.

الحوسبة السحابية

Processing

Storage

Virtual machines

Applications

Don't buy 100 servers — **rent 100 virtual servers** for 3 days, then release.

Public

سحابة عامة

Customers over the Internet

Private

سحابة خاصة

One organization

Hybrid

سحابة هجينة

Public + private combined

Types of Cloud Computing

Cloud computing delivers **computing, storage, and even applications as a service** across a network — a **logical extension of virtualization**, which it uses as its base. **Amazon EC2** offers thousands of servers, **millions of virtual machines**, and **petabytes of storage**, billed monthly by use.

Public cloud

Available via the Internet to **anyone willing to pay**.

Private cloud

Run by a company for **that company's own use**.

Hybrid cloud

Includes **both public and private** components.

SaaS

Software as a service: applications such as **word processors** available via the Internet.

PaaS

Platform as a service: a **software stack ready for application use**, such as a database server.

IaaS

Infrastructure as a service: **servers or storage** over the Internet, for example for **backups**.

Cloud Service Models

SaaS

Software as a Service

البرمجيات كخدمة

A finished app — no
infrastructure to manage

“Just give me the software.”

PaaS

Platform as a Service

المنصة كخدمة

A platform to deploy your apps

“Give me an environment.”

IaaS

Infrastructure as a Service

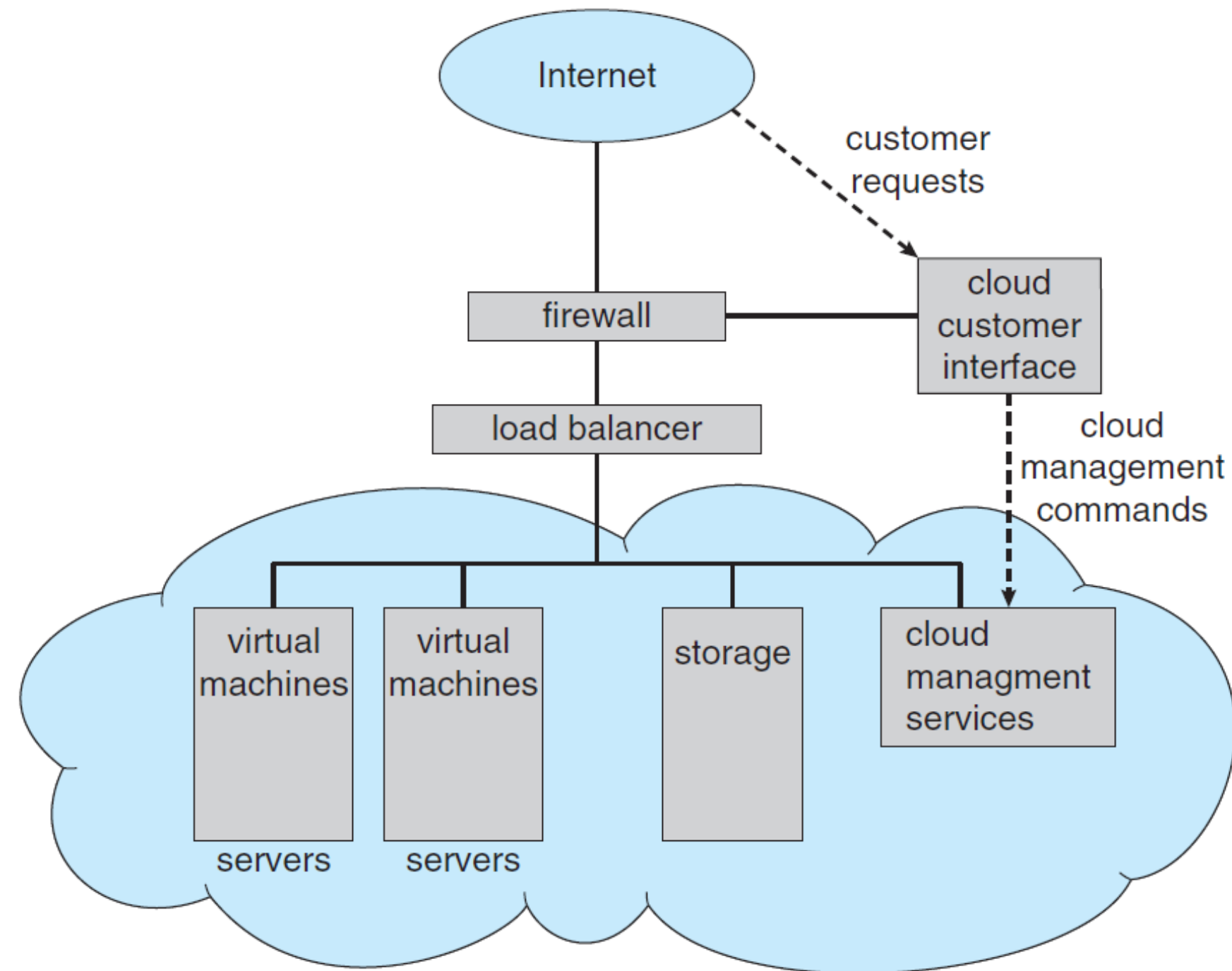
البنية التحتية كخدمة

Rent VMs, storage, networking

“Give me virtual hardware.”

SaaS = **use** software · PaaS = **build** software · IaaS = **rent** infrastructure

Inside a Public Cloud Providing IaaS

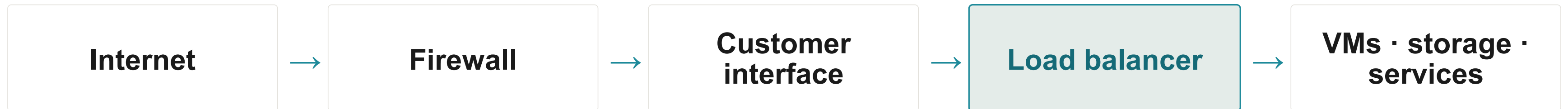


Traditional operating systems run inside the cloud infrastructure; above them, **VMMs manage the virtual machines** in which **user processes** run.

The VMMs themselves are managed by **cloud management tools** such as **VMware vCloud Director** and the open-source **Eucalyptus** toolset — a good argument for considering them **a new type of operating system**.

Both the **cloud services** and the **cloud user interface** are protected by a **firewall**.

Inside a Public Cloud



VMMs

Manage the virtual machines

Cloud management

Manages the larger environment

Load balancer — spreads requests so no machine gets all the work

موازن الأحمال

1,000 requests → **Server A** **Server B** **Server C** **Server D**

Real-Time Embedded Systems

Embedded computers are the **most prevalent form of computer in existence** — car engines, manufacturing robots, optical drives, microwave ovens — with **very specific tasks** and **little or no user interface**.

VARIANT 1

General-purpose computers running a **standard operating system** such as Linux, with **special-purpose applications**.

VARIANT 2

Hardware devices with a **special-purpose embedded operating system** providing **just the functionality desired** .

VARIANT 3

Hardware devices with **application-specific integrated circuits (ASICs)** that perform their tasks with **no operating system at all** .

A real-time system functions correctly only if it returns the correct result within its fixed time constraints .

Embedded Systems

A computer **built into another device** for one specific job.

نظام مضمن

Car engines

Robots

Optical drives

Microwaves

One purpose · little or no user interface

THREE WAYS TO BUILD IT

Standard OS

Linux + special application

Embedded OS

Special-purpose OS

Dedicated hardware

ASIC — no OS

Real-Time ≠ Fast

Result must be **correct AND before the deadline.**

نظام زمن حقيقي

Example · Car airbag

Correct decision after **10 seconds** → **useless**

NORMAL COMPUTING

Correct answer matters most

REAL-TIME COMPUTING

Correct answer + correct timing

Traditional & Mobile

ENVIRONMENT / MODEL	ARABIC	WHAT IT DESCRIBES	MAIN OS CONCERN
Traditional Computing	الحوسبة التقليدية	General desktop/workstation style computing	Run jobs, interact with users, share CPU
↳ Batch Processing	المعالجة الدفعية	Jobs collected and processed in groups	Throughput; little user interaction
↳ Interactive Computing	الحوسبة التفاعلية	User gives input and expects a response	Fast response time (زمن الاستجابة)
↳ Time Sharing	تقاسم الوقت	CPU rapidly switches among processes/users	Fair CPU sharing; creates concurrency
↳ Thin Client	عميل خفيف	Little local processing; relies on central servers	Network/server dependence
Mobile Computing	الحوسبة المتنقلة	Smartphones & tablets — GPS, accelerometer, gyroscope, wireless	Power efficiency (كفاءة الطاقة) , battery, cooling, storage

Client–Server & Peer-to-Peer

ENVIRONMENT / MODEL	ARABIC	WHAT IT DESCRIBES	MAIN OS CONCERN
Client–Server Computing	الخادم–حوسبة العميل	Clients request services; server provides them	Request → processing → response
↳ Compute Server	خادم حوسبة	Server performs a requested computation	“Do work for me”
↳ File Server	خادم ملفات	Server provides file operations	“Manage files for me”
Peer-to-Peer (P2P)	النند للنند	Every node can act as client or server	Services spread across peers; less dependence on one server

Cloud, Embedded & Real-Time

ENVIRONMENT / MODEL	ARABIC	WHAT IT DESCRIBES	MAIN OS CONCERN
Cloud Computing	الحوسبة السحابية	Computing, storage, VMs, and apps delivered as services over a network	Virtualization, scalability, resource sharing
↳ Public / Private / Hybrid	خاصة / سحابة عامة / هجينة	Who owns/uses the cloud infrastructure	Deployment model
↳ SaaS	البرمجيات كخدمة	Finished application	“Use the software”
↳ PaaS	المنصة كخدمة	Platform to build/deploy applications	“Give me an environment”
↳ IaaS	البنية التحتية كخدمة	VMs, storage, networking	“Give me virtual hardware”
Embedded Systems	الأنظمة المضمنة	Computer built inside another device for a specific job	Small, specialized, often little/no UI
Real-Time Embedded	الأنظمة المضمنة ذات الزمن الحقيقي	Embedded system where the result must arrive before a deadline	Correct result + correct timing

The Whole Story

Distributed Systems

Many computers **cooperating**

Kernel Data Structures

Efficient **organization** inside the OS

Computing Environments

The **places and models** where OSs work

The same OS principles apply from a **tiny embedded controller** to a **public cloud** of virtual machines.