

Computer Systems and Software

Hardware: Motherboard, CPU, RAM, Storage, Peripherals

Software: Firmware, System, Server-Side, Applications



SoftUni Team
Technical Trainers



SoftUni



Software University

<https://about.softuni.bg>

Have a Question?

sli.do

#qa-fund

1. Computer Systems and Software
2. Computer **Hardware**
 - Motherboard, CPU, RAM, Storage, Peripherals
3. Computer **Software**
 - Firmware, System Software and OS, Application Software
 - Web Apps, Desktop Apps, Mobile Apps





Computer Systems

Components and Functionality

What is a Computer System?

- **Computer system:** integrated bundle of hardware and software components, e. g. smartphone, POS terminal, laptop
 - Enables **efficient data input, processing, and output**
 - Comprises **interconnected software and hardware components**
 - **Human-computer interaction** for the end-users / APIs for **machine-to-machine interaction**
- **Key elements:**
 - **Hardware:** RAM, input/output devices, storage devices, CPU
 - **Software:** operating systems, drivers, apps, games

- **Early computing:** mechanical and electromechanical devices (e.g., Abacus, Babbage's Analytical Engine, ENIAC)
- **Advancements in technology:** transistors, integrated circuits, microprocessors (e.g., mainframe computers, minicomputers, personal computers)
- **Modern era:** pervasive computing, IoT, cloud computing, edge computing, rise of AI and machine learning

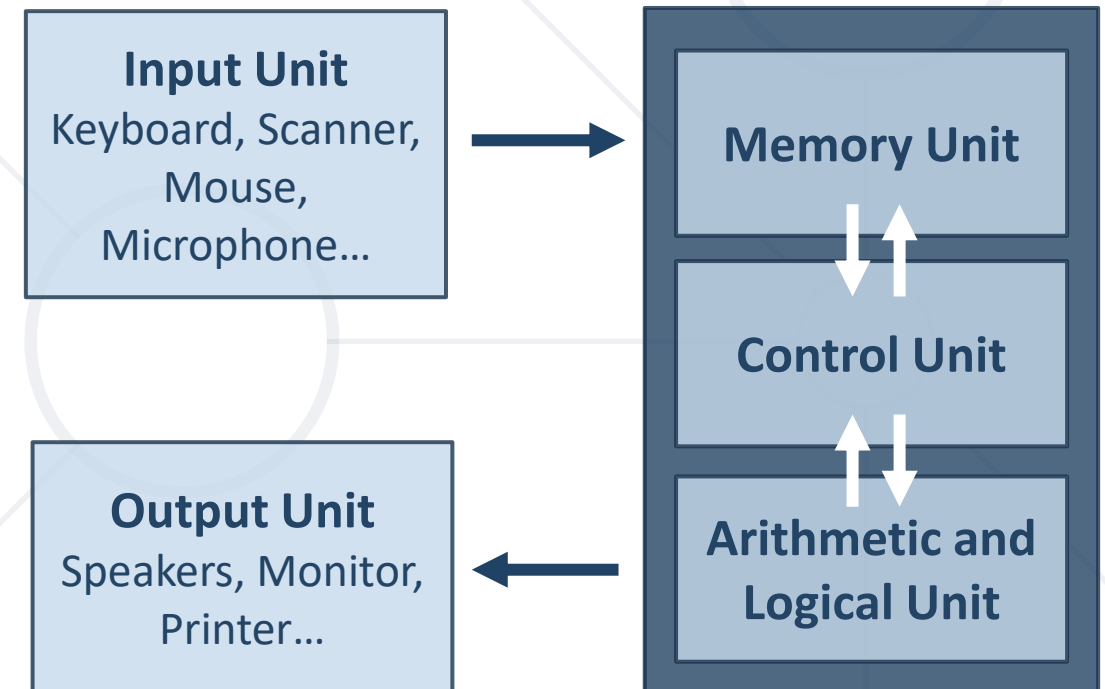




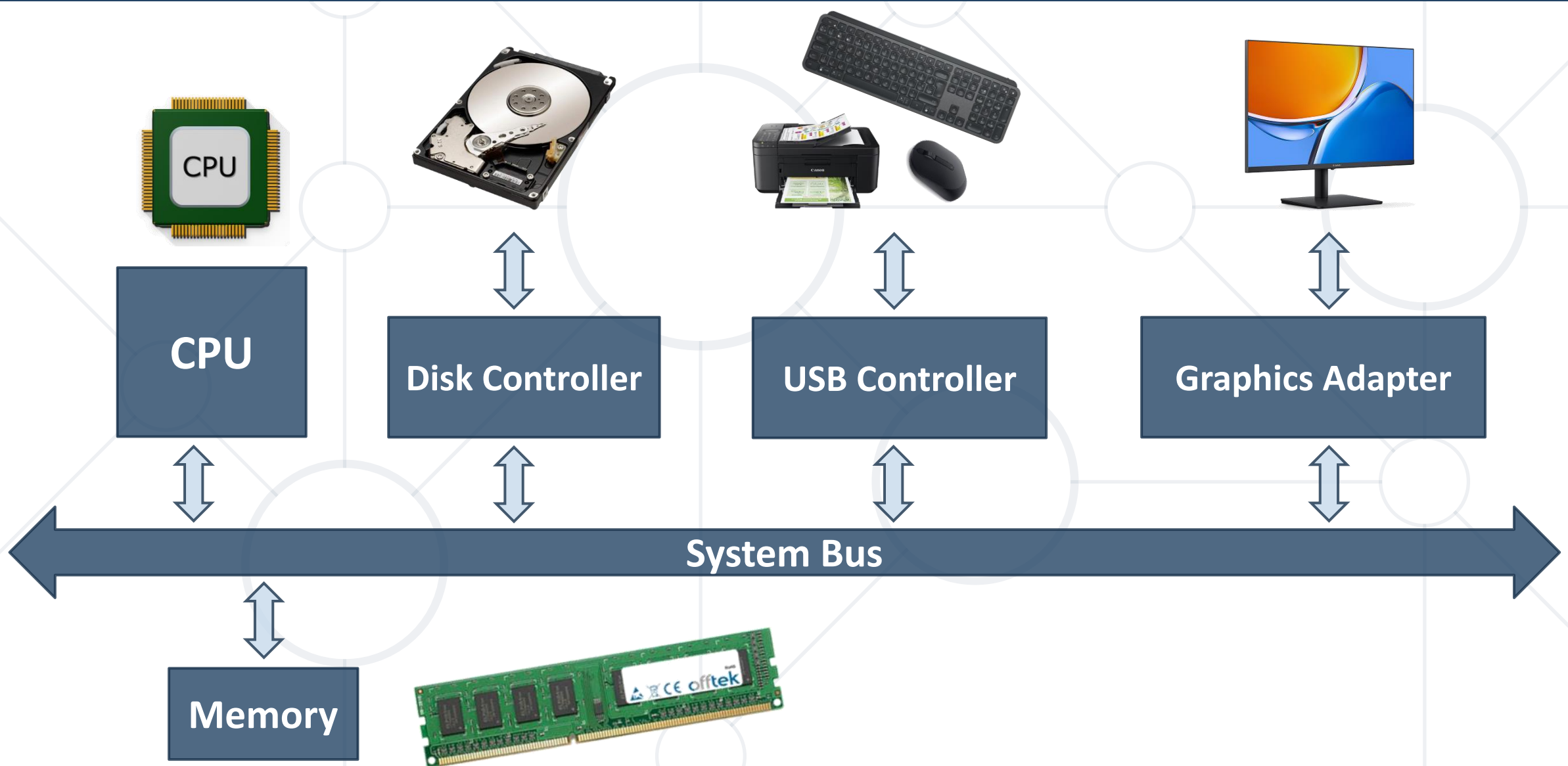
Computer Hardware

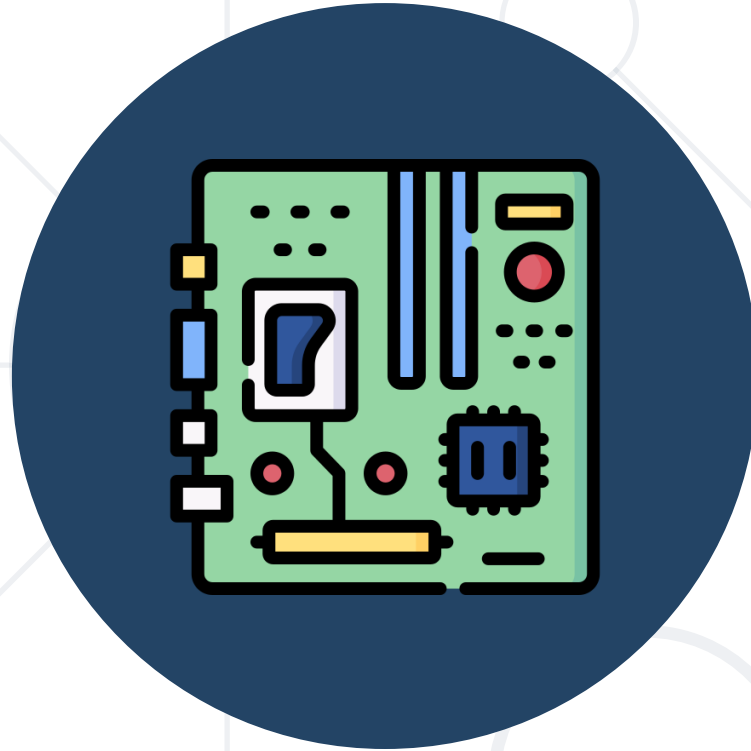
Motherboard, CPU, Memory, Storage, Peripherals

- **Hardware** refers to the **physical components** of a computer
- Central Processing Unit (**CPU**) – microprocessor
 - **Executes the code** (programs)
 - All **data processing operations**
- **Input devices**
 - Enter data
- **Output devices**
 - Get information



Computer System Hardware





Motherboard

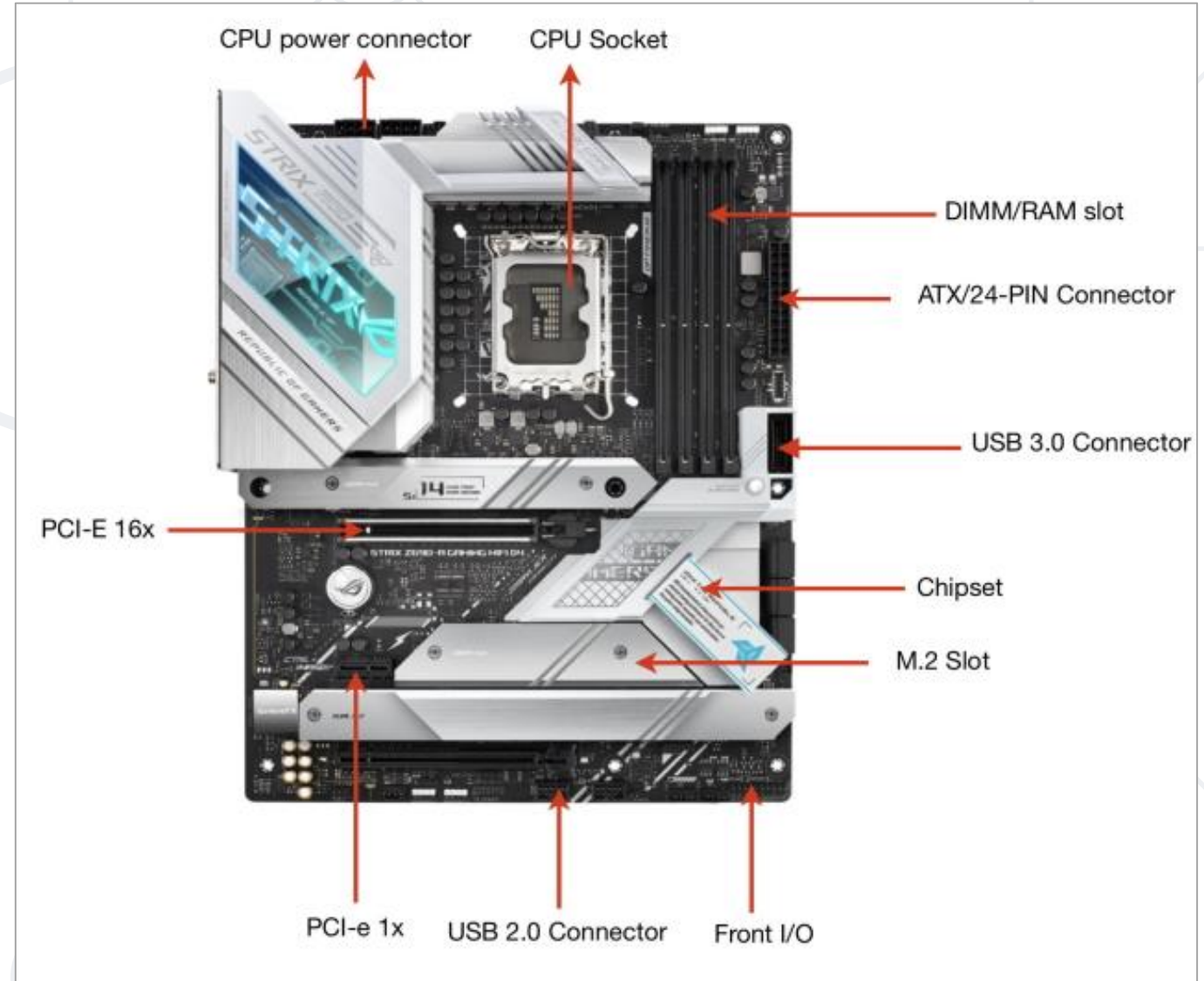
Backbone of a Computer System

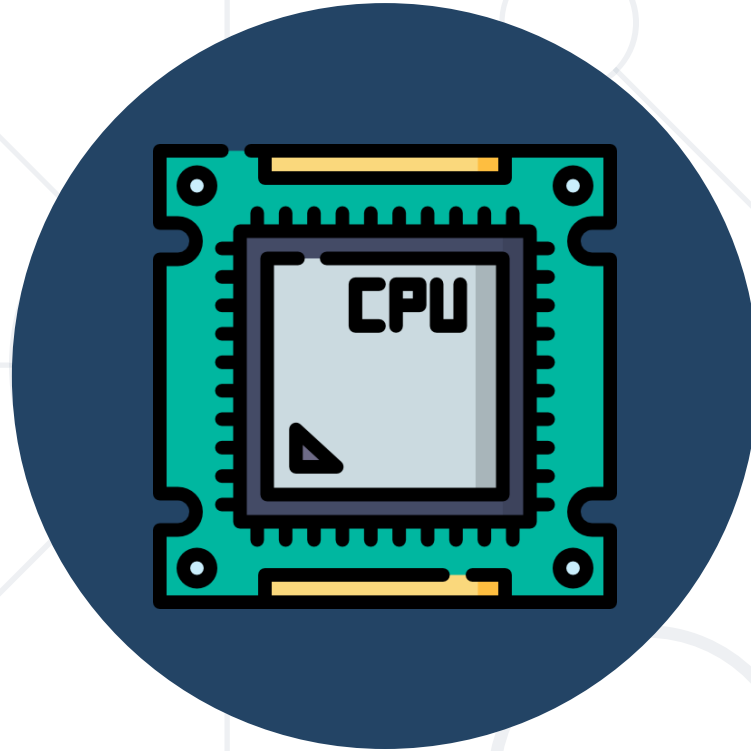
What is a Motherboard?

- **Motherboard** - central hub for **hardware connectivity**
 - **Communication** between all **hardware components**
- **Compatibility** considerations
 - Each motherboard is **designed to work with specific types** of **processors** and **memory**
- **Expansion slots** for enhanced functionality
 - **Video cards** for improved graphics performance
 - **Sound cards** for enhanced audio capabilities
 - **Network cards** for better internet connectivity

Motherboard Components

- CPU socket
- RAM slots
- Power connectors
- Chipset
- Expansion slots
- SATA connectors
- USB connectors
- Bluetooth module





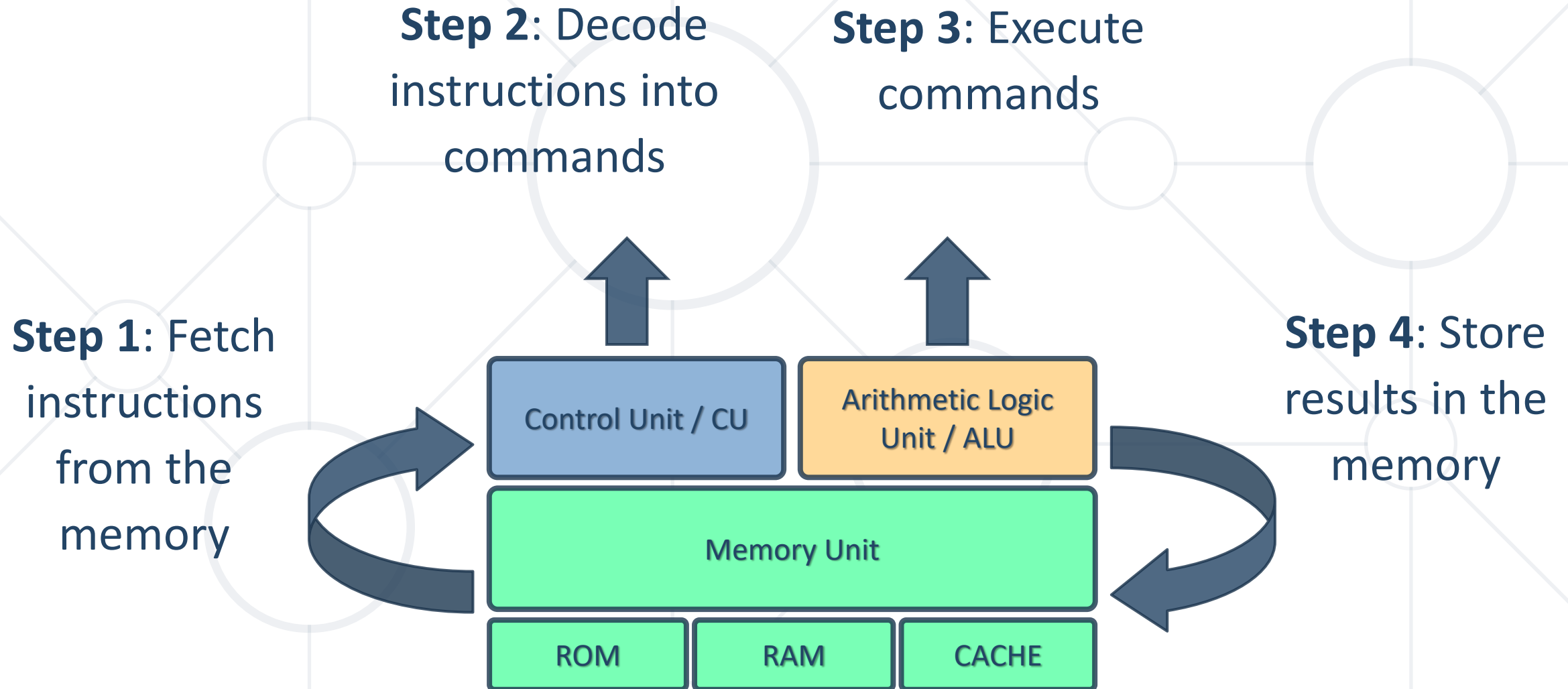
CPU (Microprocessor)

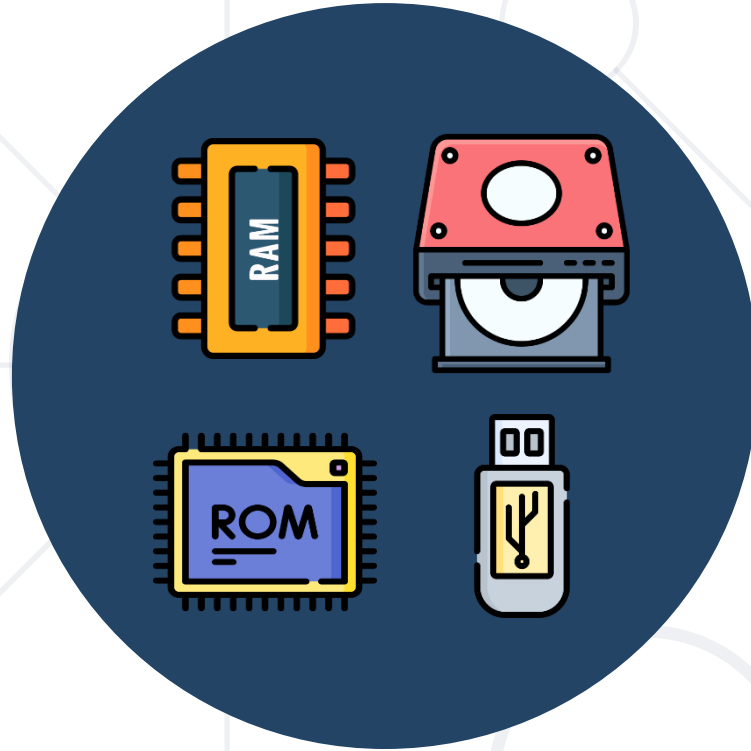
Central Processing Unit

What is CPU?

- **CPU** – the brain of the computer
 - Executes **calculations, actions, and runs programs**
 - Provides **processing power and instruction control**
- Three **core components**
 - **Control Unit (CU)**
 - Manages instruction flow and coordinates hardware functions
 - **Arithmetic and Logical Unit (ALU)**
 - Performs arithmetic and logic operations
 - **Memory Unit (MU)**
 - Stores data, programs, and information

CPU Parts Workflow



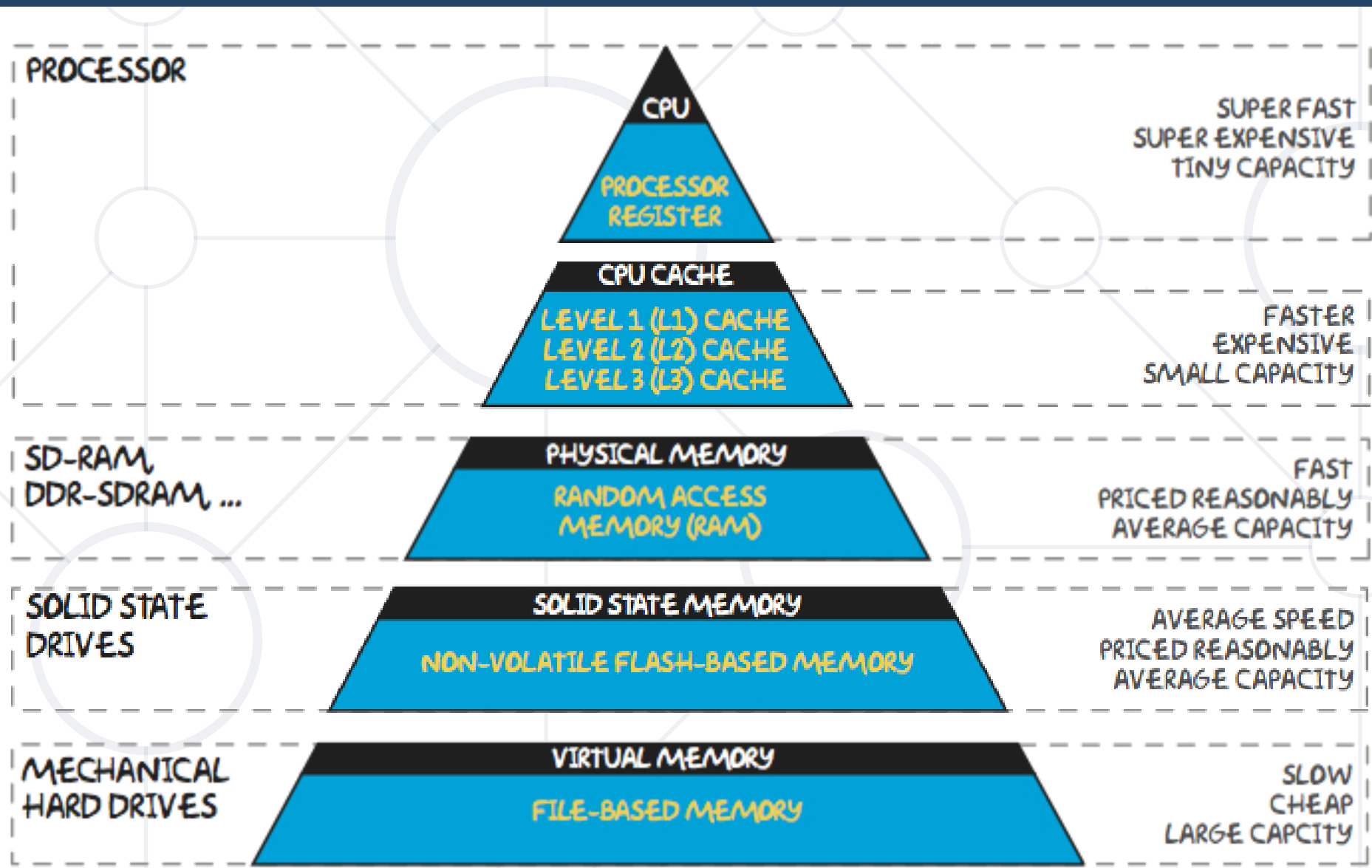


Memory and Storage

Storing Information in a Computer

- **Primary memory**
 - **RAM** – read / write: stores data, required by the CPU during the **execution of a program**
 - **ROM** – read-only: stores **crucial data for the system** to operate, like the essential program for the computer boot
- **Secondary memory**
 - Not accessed directly by the processor
 - Examples: hard drive, SSD, flash, optical drive, USB drive
- **Cache memory**
 - Part of the CPU, very fast: temporarily stores **frequently used instructions and data** to speed-up access

Memory Hierarchy



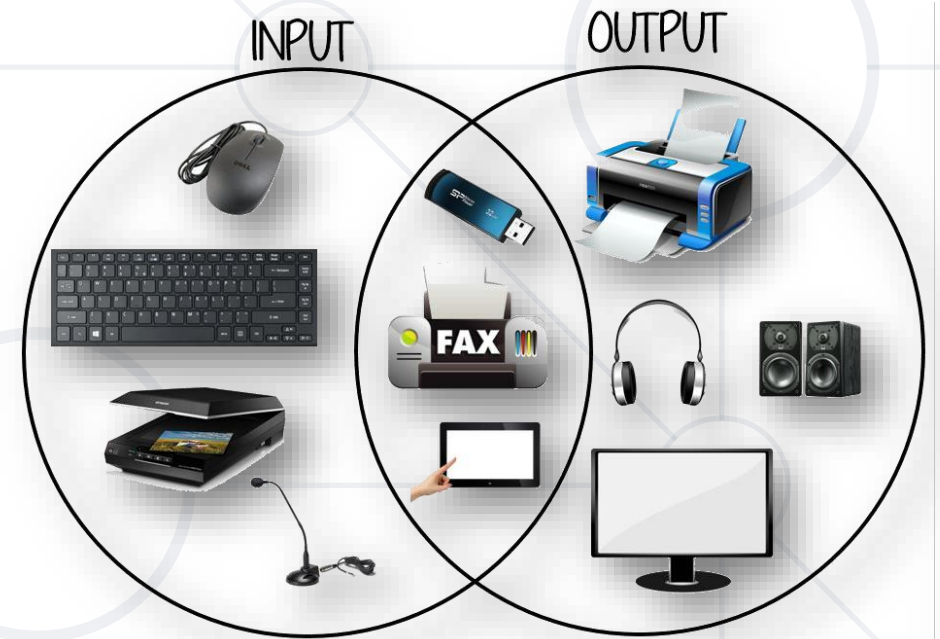


Peripheral Devices

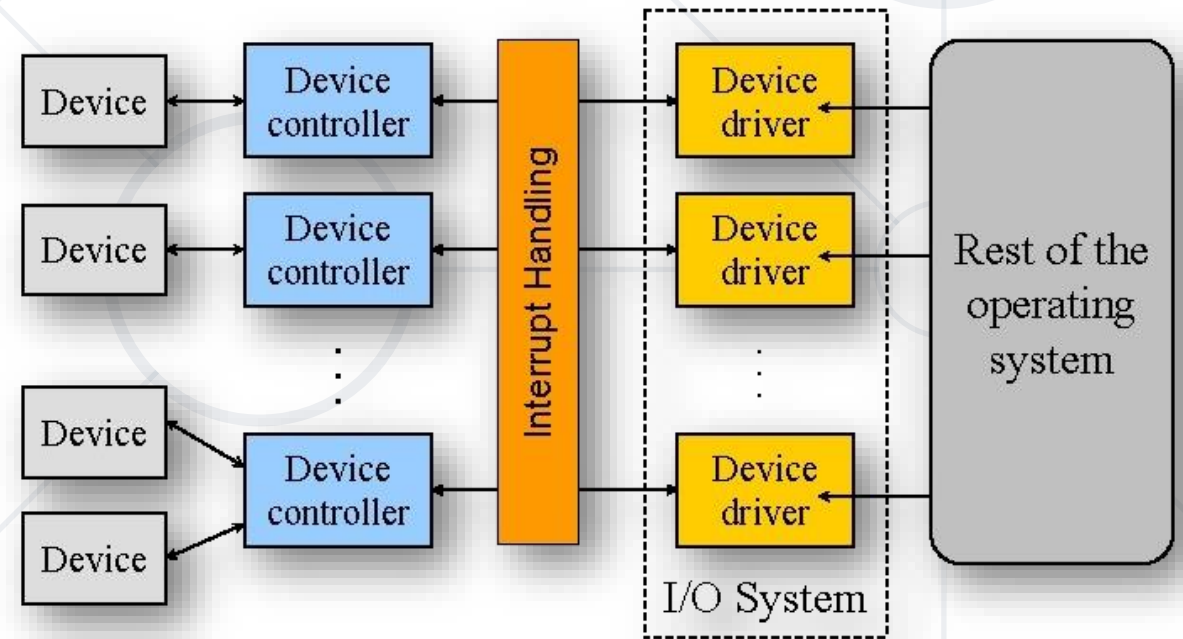
Expanding Computer's Functionality

What is a Peripheral Device?

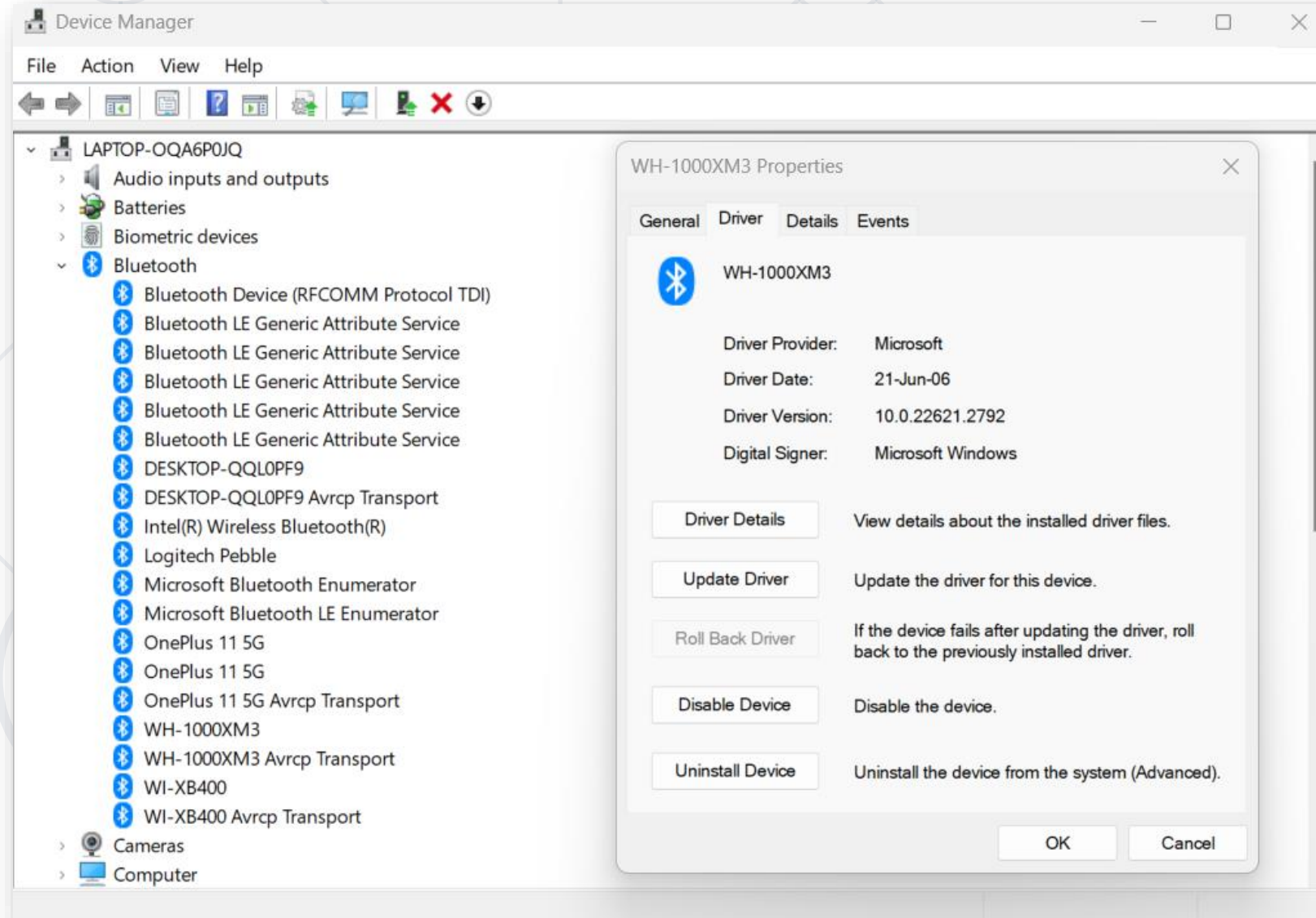
- Any connected device that expands computer's capabilities with additional functionality
- **Three main categories:**
 - **Input devices** → read data, e.g. keyboard, mouse, microphone
 - **Output devices** → write data, e. g. speakers, printer, monitor
 - **Input/output devices** → mixed, network card, hard drive, touchscreen monitor



- Device **controller**
 - A **physical device** for **connection** between a peripheral device and the computer, e. g. USB controller
- Device **driver**
 - **System software**, which enables the **communication and data transfer** between devices and the system



Device Manager in MS Windows



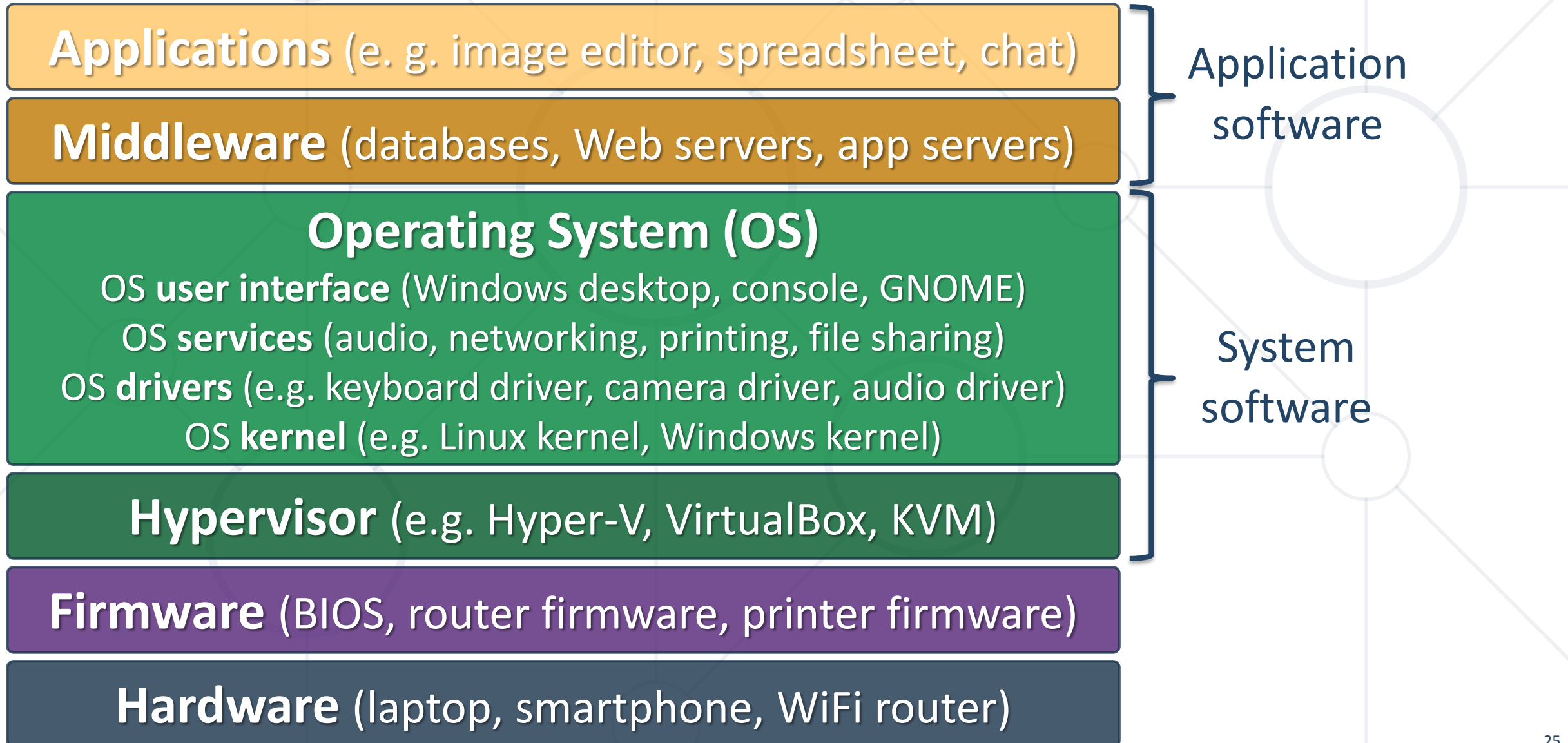


Computer Software

Firmware, System Software, Applications

- **Computer software**
 - Computer programs, instructions, and data that enable a computer system to **perform specific tasks**
- **Types** of software:
 - **Application software:** help the business to run, e.g. email software, spreadsheets, word processing, CRM systems, etc.
 - **System software:** interacts with and **manages the hardware**
- **Standalone apps** vs. **software systems** (client + server)

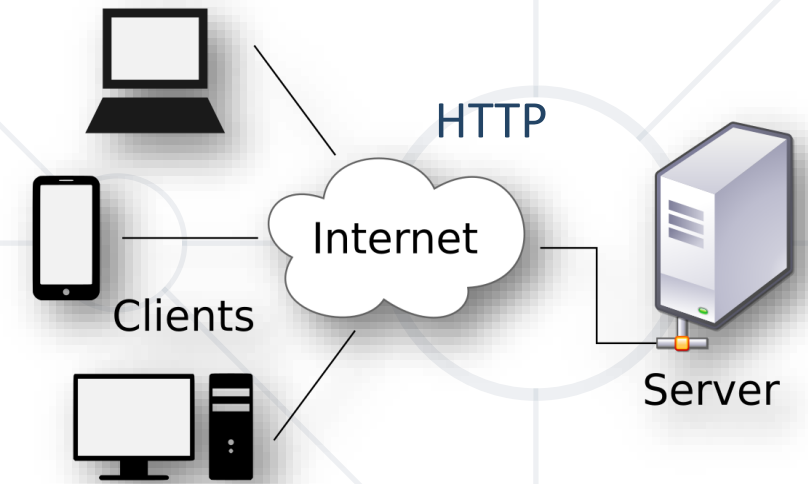
Software Stack



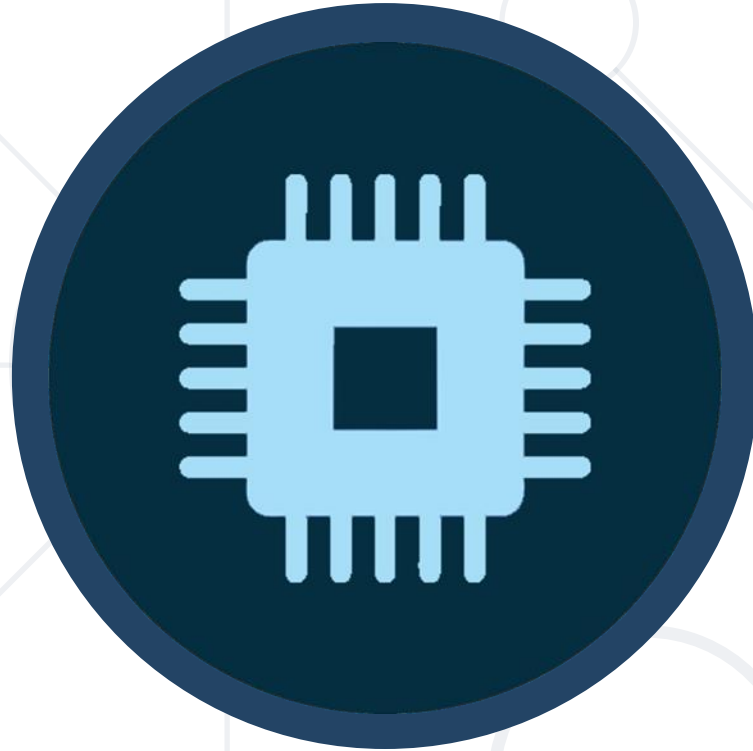
- **Firmware and embedded software**
 - **Low-level software** used to operate a hardware device
- **System software**
 - **Manages and controls hardware**, platform for applications
 - **Operating systems (OS)** – Windows, Linux, macOS, Android
 - **Hypervisors** – runs virtual machines (VMs) in the host OS
- **Application software**
 - Business applications, office apps, multimedia, communication
 - Several types: **Web apps, desktop apps, mobile apps**

- **Standalone apps**
 - Run **locally**, store their data locally, do not need Internet
 - Examples: Windows Calculator, Windows Explorer, Minesweeper
- **Software systems**
 - Consists of several **components** (e. g. client + server)
 - Example: mail server (remote) + mail client app (local)
 - **Cloud apps**: hold all user data in the cloud + local client
 - Example: Google Docs, Discord, Trello, Canva

- **Front-end** and **back-end** separate the modern apps into **client-side** (UI) and **server-side** (data) components
- **Front-end** == client-side components (Desktop / mobile app / Web browser)
 - Implement the **user interface** (UI)
- **Back-end** == server-side components (data and business logic APIs)
 - Implements **data storage and processing**



- **HTTP** connects front-end with back-end



Firmware

Bridge between Hardware and Software

What is Firmware?

- **Firmware** - **permanent**, low-level software, **embedded** in a device's read-only memory (ROM)
 - Controls device's basic functions and provides a **stable foundation** for higher-level software
 - Example: Wi-Fi router's firmware, coffee machine firmware
- **Functions** of firmware
 - Hardware **initialization** during the boot process
 - **Management** of low-level hardware operations (e. g. device **initialization**, hardware **diagnostics**, and system **booting**)

- Examples of **firmware applications**
 - BIOS / UEFI in laptops and desktop computers
 - Firmware in routers, printers, scanners
 - Embedded systems, such as IoT devices
- **Firmware updates**
 - Most devices allow firmware updates to improve functionality or fix issues
 - Can be critical for security and performance



System Software

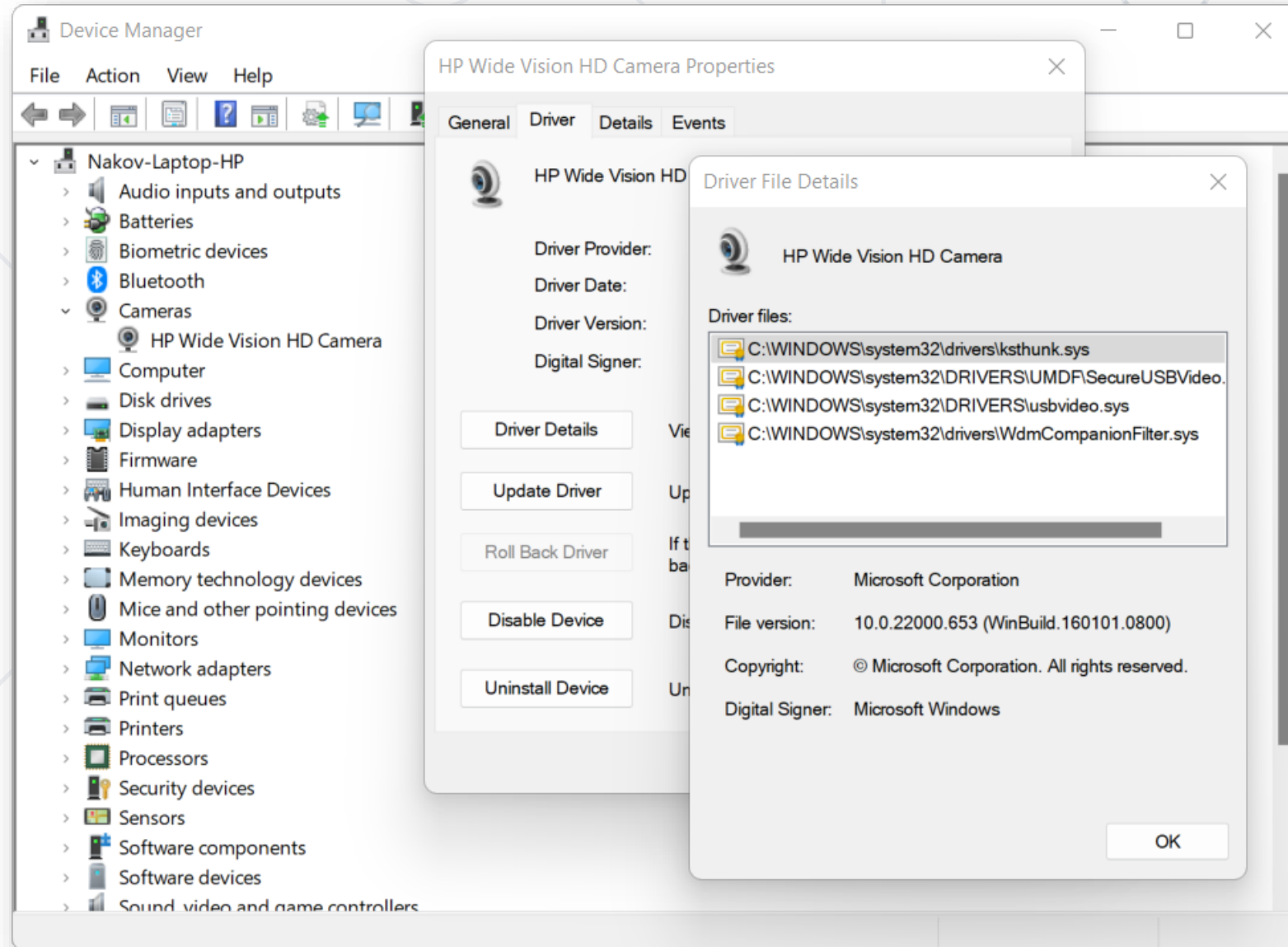
Foundation for Application Software

What is System Software?

- Software designed to **manage** and **control** computer **hardware**, providing a **platform** for **application software**
- Examples of system software:
 - **Hypervisors** – runs virtual machines (VMs) in the host OS
 - **Operating systems** (OS) – Windows, macOS, Linux, Android
 - **Device drivers** – software that enables communication between hardware and operating system), e. g. mouse driver
 - **System utilities** – tools for system maintenance and optimization, e. g. anti-virus, task manager, print spooler

- Windows, macOS, Linux, Android, iOS
- Manage the **hardware** and **software** resources
- Manage **processes** (concurrently running apps)
 - Distribute the system resources between all processes
- Manage **file system** and **memory** (RAM)
- Manage **users**, **security** and **access control**
- System **updates** and **maintenance**





- In Windows, the **"Device Manager"** lists all devices, drivers, etc.

- Tools that help **maintain** and **optimize** a computer system
 - **Antivirus** and **malware** protection (e.g. Windows Defender)
 - System **backup** and **recovery** (e. g. Macrium Reflect)
 - Disk **cleanup** and **defragmentation** (e. g. CCleaner)
 - Performance **monitoring** and **diagnostics** (Task Manager)
 - Software **updates** and **patches** (e. g. Windows Update)
 - System hardware **information** (e. g. CPU-Z)
 - System **logs viewer** (e. g. Windows Events Viewer)

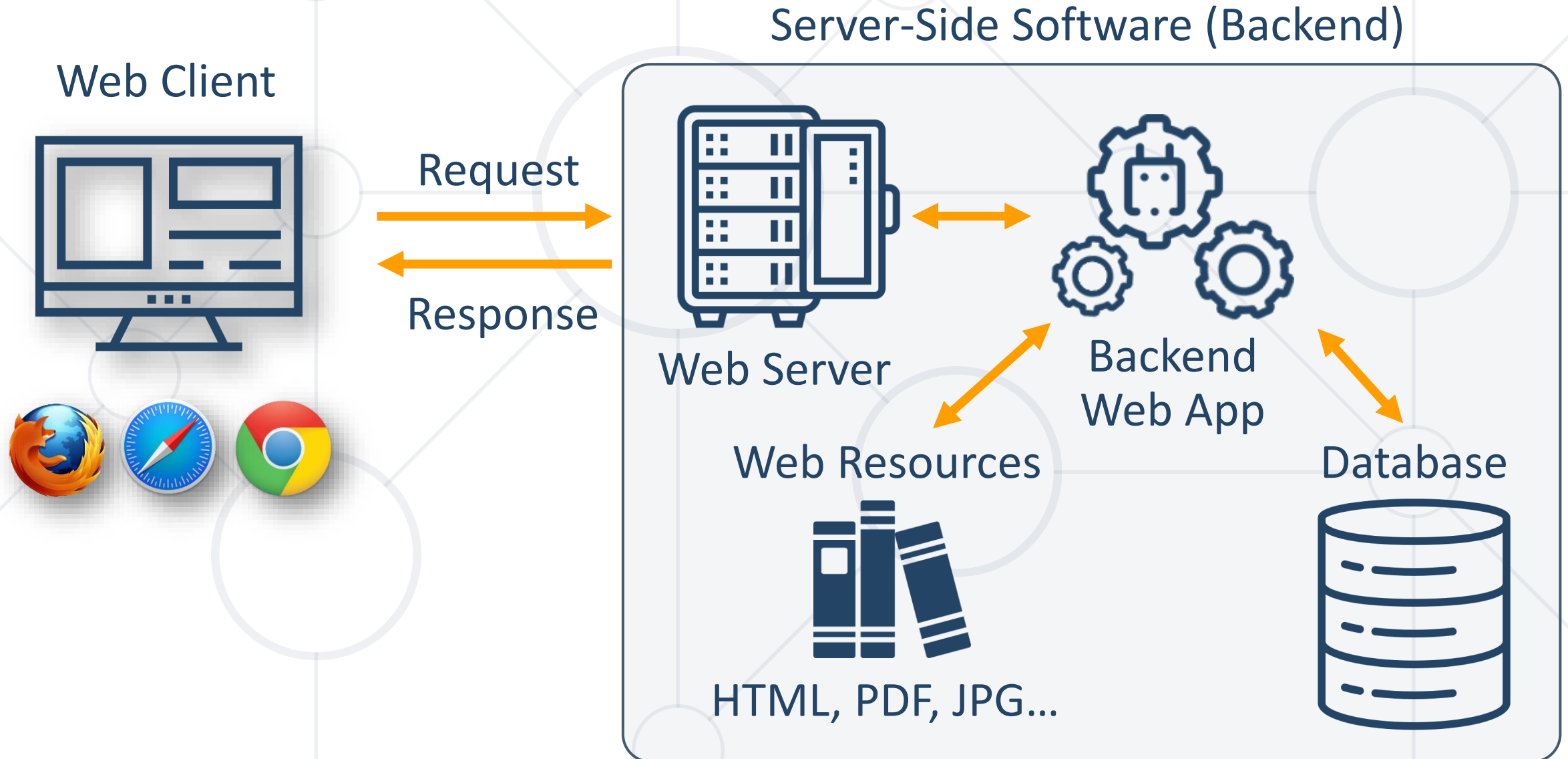


Server-Side Software (Backend)

Facilitating Backend Operations and Web Services

- **Server-side software** (backend software) runs on a remote server, processes requests and delivers data to client devices
- **Common types** of server-side software
 - **Web** servers (e. g. Apache, Nginx, IIS)
 - **Database** servers (e. g. MySQL, PostgreSQL, MongoDB)
 - **Application** servers / runtimes (e. g. Tomcat, Node.js, .NET Core)
 - **Mail** servers (e. g. Microsoft Exchange Server, Postfix)
 - **File** servers (e. g. Windows File Server, Samba)
 - **Authentication** servers (e. g. FreeIPA, Active Directory)

The Client-Server Model in Web Apps



- **Server-side** software (backend software):
 - **Executes** on a **remote server**, rather than on the user's device
 - Handles **data processing, storage, and retrieval**
 - Powers Web applications, backend APIs, cloud services, etc.
 - Requires **efficient resource management** for optimal performance
- Graphical User Interface (**GUI**) / **front-end** apps:
 - **Executes** on the **user's device** (desktop, mobile, or Web)
 - Providing seamless and visually **appealing user experience**
 - Can be **Web** apps, **desktop** apps, or **mobile** apps



Application Software

Apps for the End Users

What is Application Software?

- **Application software** is designed for users to perform **specific business tasks**, catered to their **individual needs**
- Examples of application software
 - **Productivity** tools (Microsoft Office, Google Workspace)
 - **Multimedia** software (Adobe Photoshop, VLC Media Player)
 - **Communication** apps (Zoom, WhatsApp, MS Teams)
 - **Web browsers** (Google Chrome, Mozilla Firefox, Safari)
 - **Games** (Fortnite, League of Legends)



Web Apps

Applications, Accessed from the Web Browser

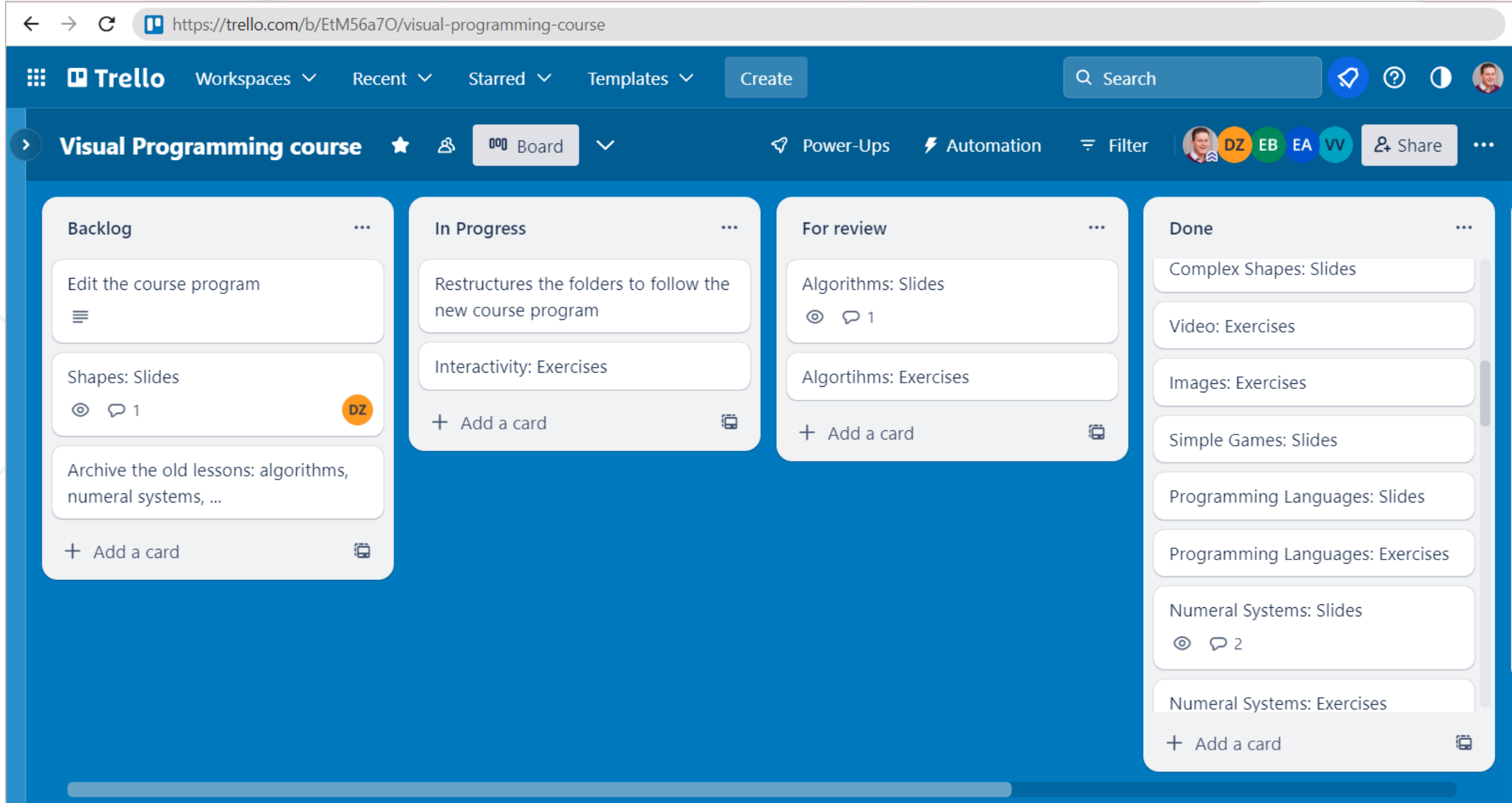
- What are **Web apps**?
 - Accessed through a **Web browser** with an **active Internet** connection
 - **Platform-independent**
 - Accessible on any device with a Web browser
 - Desktop / mobile Web browsers
 - **Automatic updates** (always up-to-date)
 - No need for manual installation or updating

- **Benefits** of Web apps
 - **Scalability:** easily accommodate a growing user base
 - **Centralized data storage:** simplifies data management and backup
 - **Lower device requirements:** minimal hardware needed (processing is done on the server-side)
 - **Easier collaboration:** real-time collaboration
 - **Cross-platform compatibility:** works across various operating systems and devices

Testing Challenges for Web Apps

- **Compatibility** - if the app works consistently across **different Web browsers** and **different screen sizes** (responsive design)
- **Usability** - testing for **accessibility**, **intuitive use** on different devices, and ease of **navigation**
- **Network conditions** - Web apps rely on an active internet connection → testing under **different network conditions**
- **Security** - Web apps deal with sensitive data → testing for **vulnerabilities** such as XSS attacks and SQL injection
- **Performance** - performance can be affected by network speed / server load / browser capabilities → testing for **scalability / load capacity**

Trello Project Management Web App



The screenshot displays a Trello workspace for a "Visual Programming course". The interface includes a top navigation bar with options like "Workspaces", "Recent", "Starred", and "Templates", along with a search bar and user avatars. The board itself is organized into four columns:

- Backlog:** Contains three cards: "Edit the course program", "Shapes: Slides" (with 1 comment), and "Archive the old lessons: algorithms, numeral systems, ...".
- In Progress:** Contains two cards: "Restructures the folders to follow the new course program" and "Interactivity: Exercises".
- For review:** Contains two cards: "Algorithms: Slides" (with 1 comment) and "Algorithms: Exercises".
- Done:** Contains seven cards: "Complex Shapes: Slides", "Video: Exercises", "Images: Exercises", "Simple Games: Slides", "Programming Languages: Slides", "Programming Languages: Exercises", and "Numeral Systems: Slides" (with 2 comments). It also includes a card for "Numeral Systems: Exercises".

Each column has an "Add a card" button at the bottom. The board is titled "Visual Programming course" and includes a "Board" tab, "Power-Ups", "Automation", and "Filter" options.



Desktop Apps

Applications Running Locally on Your Laptop

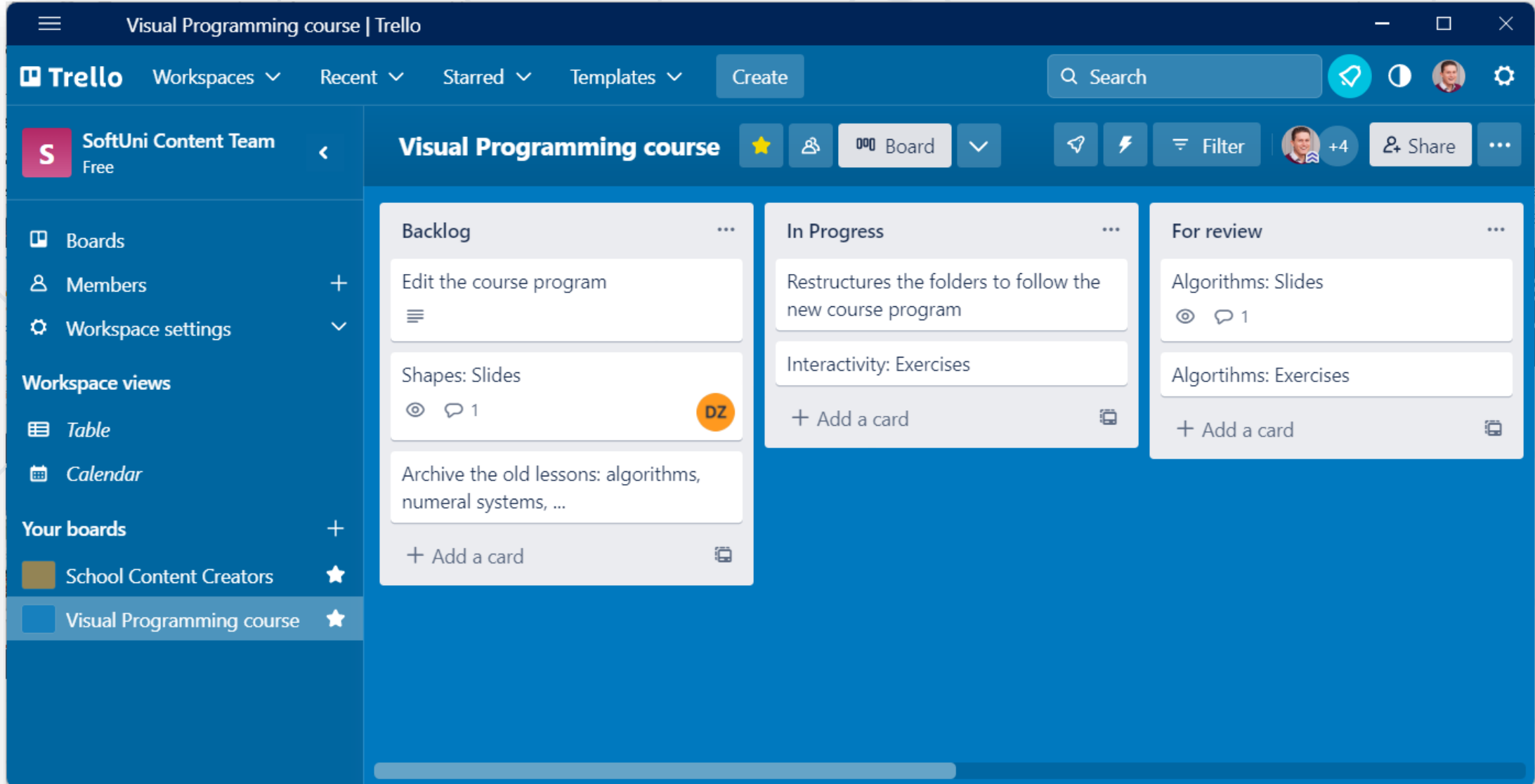
- What are **desktop apps**?
 - **Installed** and **run locally** on a user's computer
 - Store their data locally or remotely (depends)
 - **Offline access**
 - Can be used without an Internet connection
 - **More features**
 - Often more feature-rich than Web apps
 - Better integrated with the host OS

- **Benefits** of desktop apps
 - **Performance:** faster processing and response time, as tasks are executed locally
 - **Customization:** easily tailored to individual user preferences and needs
 - **Integration:** compatible with other locally installed software and hardware
 - **Cost-effective:** one-time purchase or licensing fees, instead of recurring subscription costs (depends)

Testing Challenges for Desktop Apps

- **Installation / uninstallation** including any dependencies or prerequisites
- **Performance testing on different hardware configurations** – processors, memory, and graphic cards
- **Compatibility testing** for different operating systems and their different versions
- **User interface testing** - desktop apps often have complex **UI** that need to be thoroughly tested
- **Integration testing** with other desktop applications

Trello Project Management Desktop App



The screenshot displays the Trello desktop application interface. At the top, a dark blue header bar contains the Trello logo, navigation links for Workspaces, Recent, Starred, and Templates, a 'Create' button, a search bar, and user profile icons. Below this, a sidebar on the left shows the 'SoftUni Content Team' workspace with options for Boards, Members, and Workspace settings. It also lists 'Workspace views' (Table, Calendar) and 'Your boards' (School Content Creators, Visual Programming course). The main area features a board titled 'Visual Programming course' with three columns: 'Backlog', 'In Progress', and 'For review'. The 'Backlog' column contains three cards: 'Edit the course program', 'Shapes: Slides' (with 1 comment and a 'DZ' label), and 'Archive the old lessons: algorithms, numeral systems, ...'. The 'In Progress' column has one card: 'Restructures the folders to follow the new course program'. The 'For review' column has two cards: 'Algorithms: Slides' (with 1 comment) and 'Algorithms: Exercises'. Each column has an 'Add a card' button at the bottom.



Mobile Apps

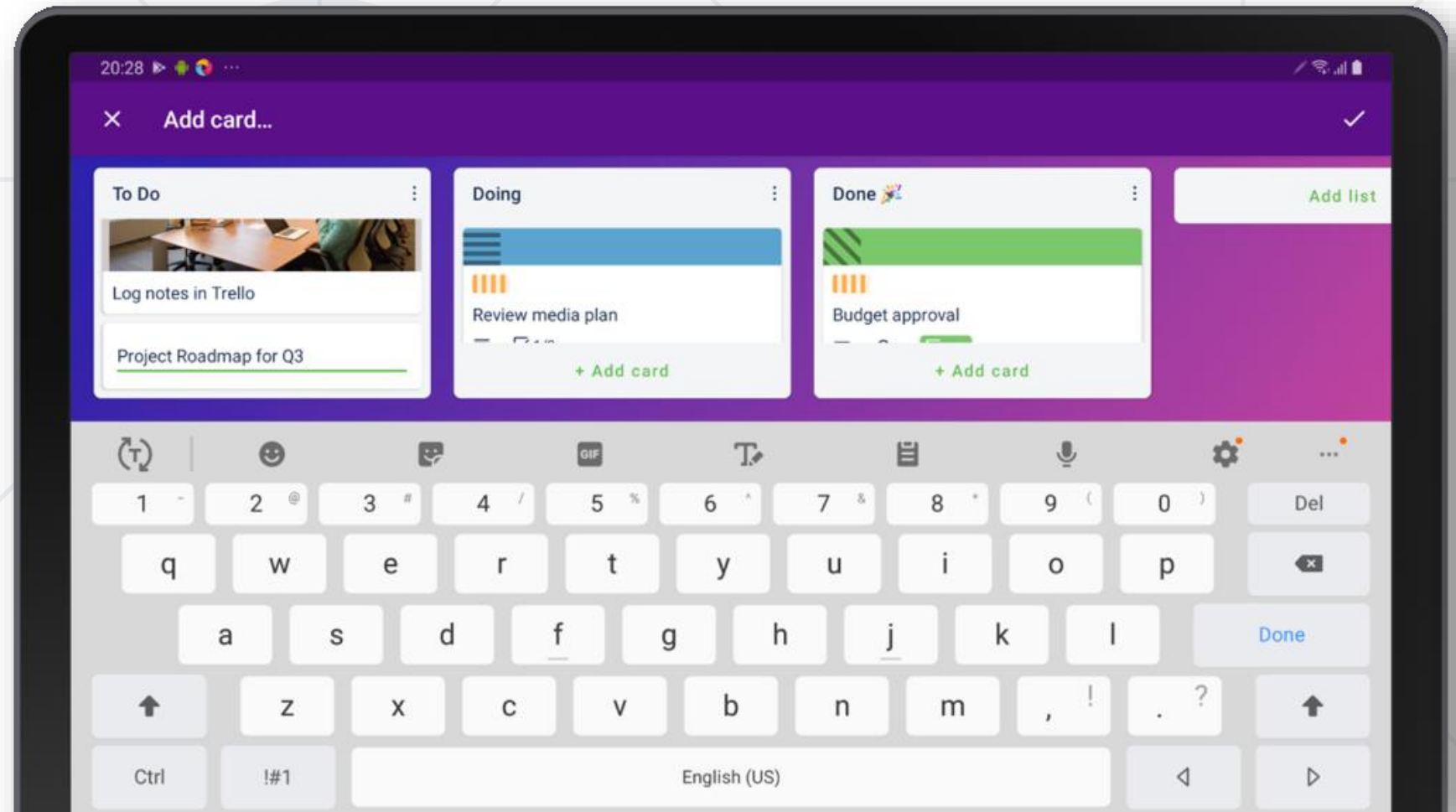
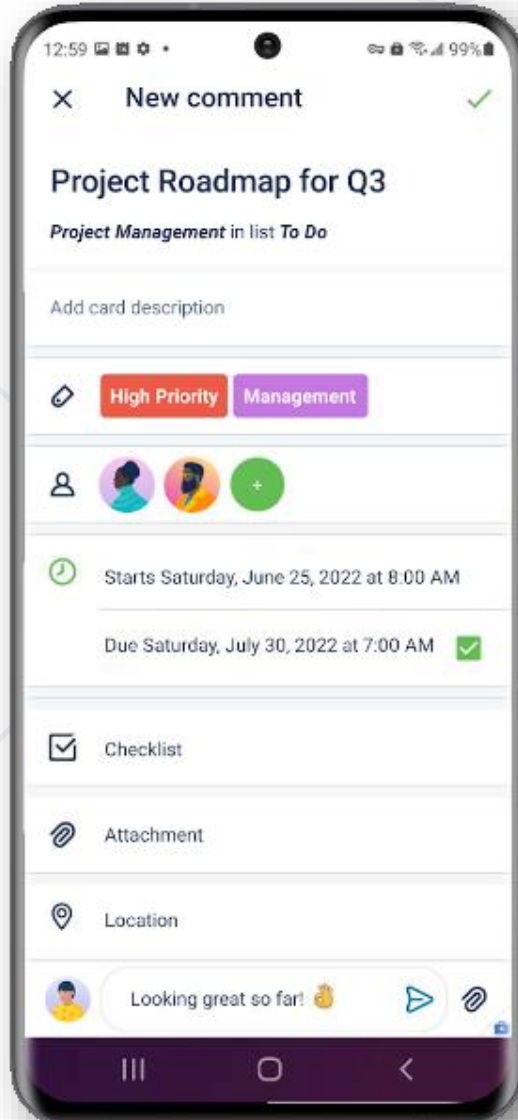
Applications Running Locally on Mobile Device

- What are **mobile apps**?
 - Designed specifically for **smartphones** and **tablets**
 - Accessible through **dedicated app stores** (e.g., Google Play, Apple App Store)
 - Optimized for **touchscreen interfaces** and mobile device features (adaptable UI design for different screen sizes)
 - Can work **offline, online** or mixed

- **Benefits** of mobile apps
 - **Portability:** access apps and data on-the-go, anytime, anywhere
 - **Push notifications:** real-time updates and alerts for improved user engagement
 - **Device-specific features:** leverage device capabilities like GPS, camera, and sensors
 - **Offline functionality:** some apps can operate without an Internet connection
 - **Streamlined user experience:** tailored for smaller screens and touch-based interactions

- **Compatibility** across different devices and OS versions is crucial for mobile apps (**many different devices and versions in use**)
- **User interface testing** – design and layout has significant impact on the user's experience on a **smaller screen**
- **Performance testing** – performance may be affected by **limited processing power and memory** on the user's device
- **Battery life testing** – to ensure that the app does not significantly **drain** the user's device **battery**

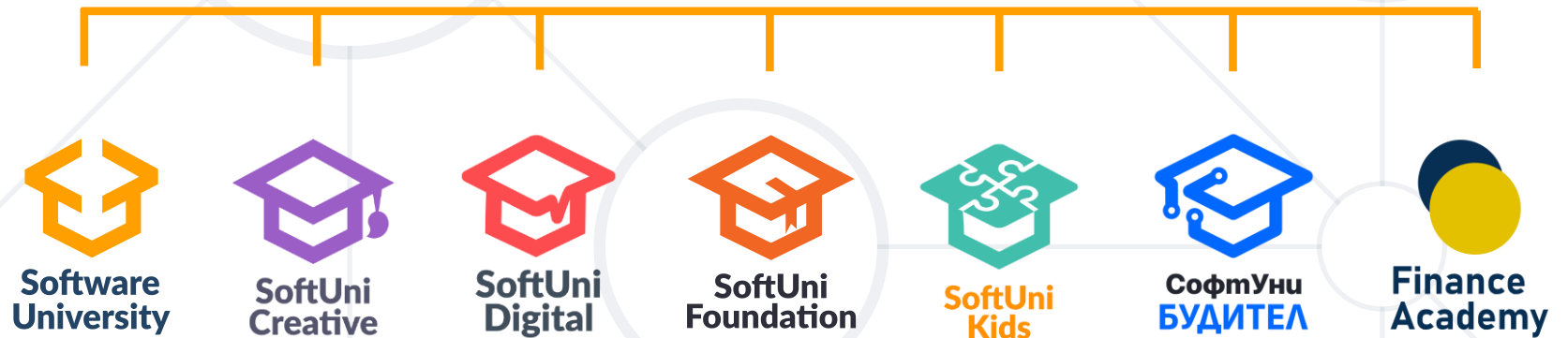
Trello Project Management Mobile App



- **Hardware** is the physical part
 - Main computer parts: **motherboard** (ties together all components), **CPU** (code execution), **input / output devices**
- **Software** – programs, running in the computer
 - **Firmware** and **system** software (OS, hypervisors)
 - **Server-side** software (back-end) vs. GUI / front-end apps
 - **Application** software (end-user apps): **Web, Desktop, Mobile** apps
 - **Software systems** (client + server) and **cloud apps**



Questions?



SoftUni Diamond Partners



- Software University – High-Quality Education, Profession and Job for Software Developers

- softuni.bg, about.softuni.bg

- Software University Foundation

- softuni.foundation

- Software University @ Facebook

- facebook.com/SoftwareUniversity



Software University



- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – <https://about.softuni.bg/>
- © Software University – <https://softuni.bg>

