Microncontrollers and Communications in IoT

Internet of Things

CS5055 – 2025I
PROF.: CWILLIAMS@UTEC.EDU.PE
SRC: BUYYA&VAHID, U.MELBOURNE

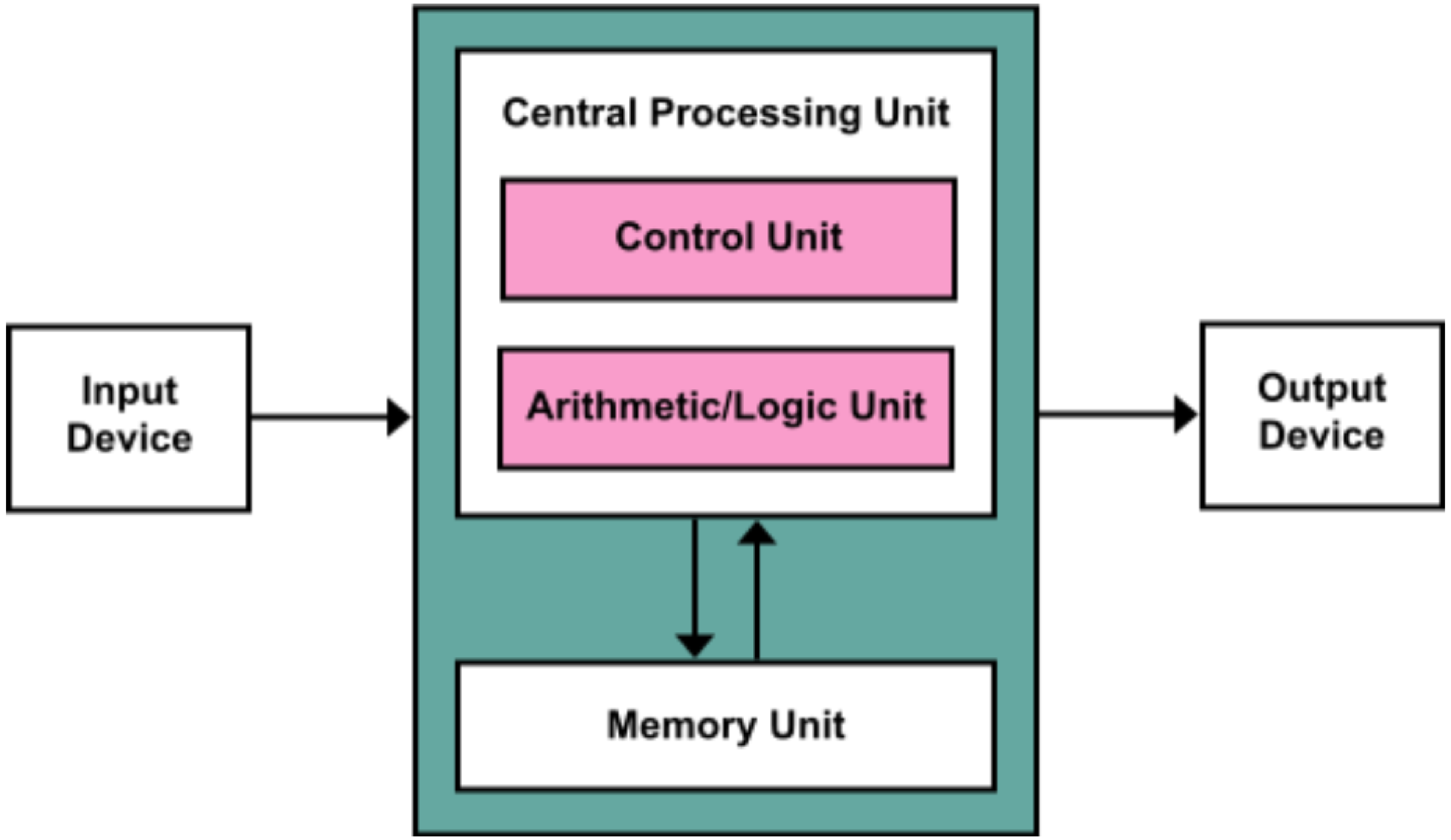
Executive Summary

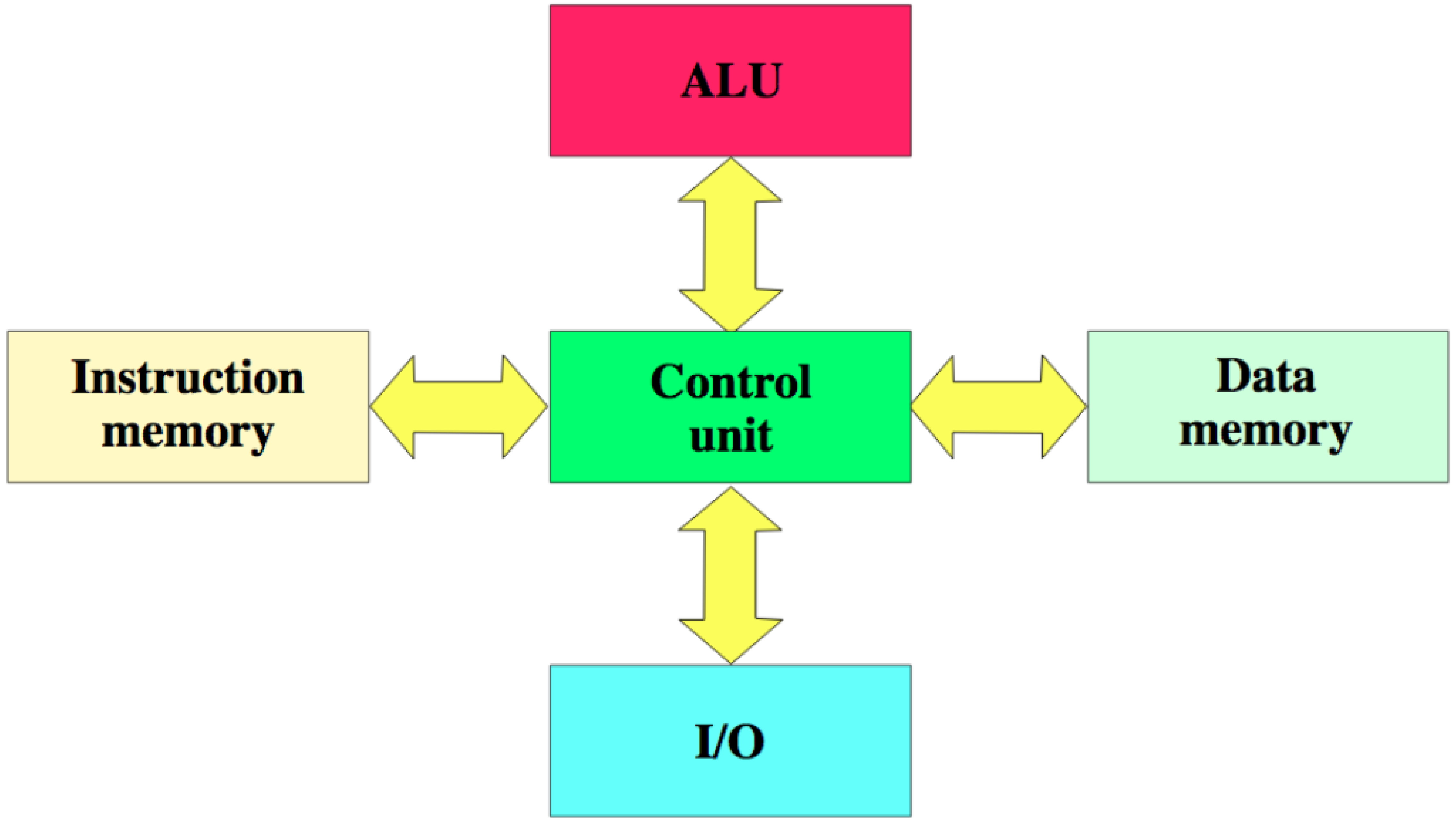
- **Motivation: IoT nodes include data processing and communications.**
- **Problem: How to implement efficient communications on small architectures**
- **Overview:**
- Recall: microprocessor and microcontrollers
- IoT Communications layer introduction
- **Conclusion: We extended our IoT model to include data processing and communications**

Outline

Recall: Microprocessors and Microcontrollers

- Von Neumann Architecture vs Harvard





Recall: Microprocessors and Microcontrollers

- **Von Neumann Architecture vs Harvard**

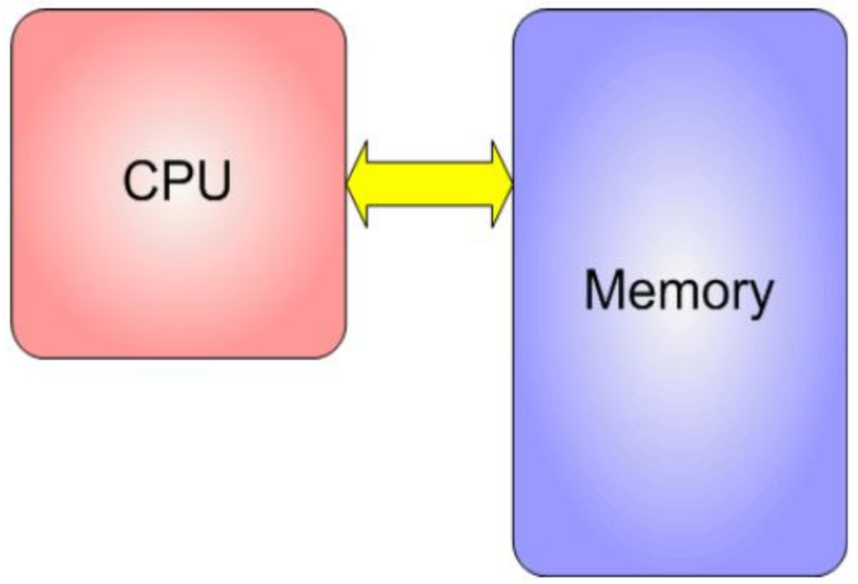


Fig.: Von Neumann

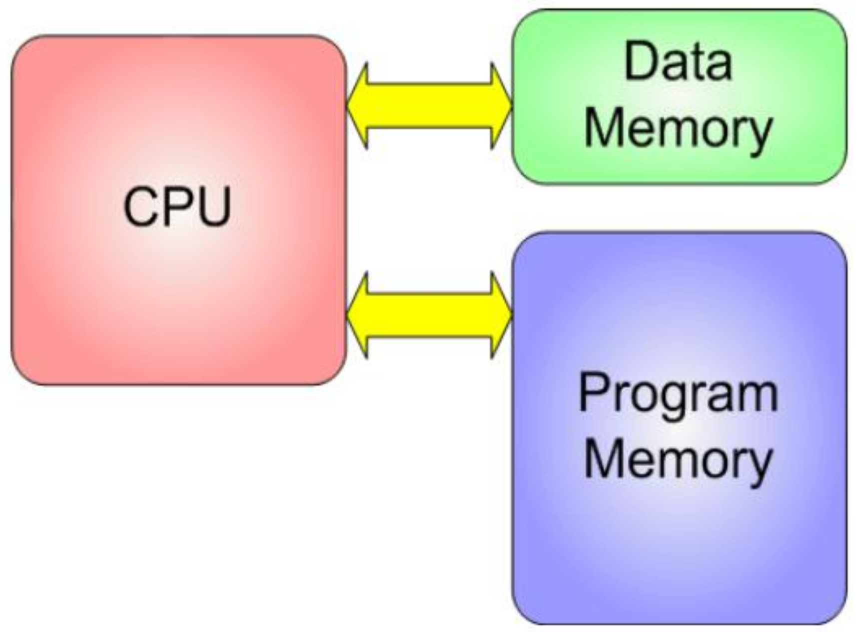


Fig.: Harvard

Feature	**Von Neumann Architecture**	**Harvard Architecture**
Memory	Shared memory for data and instructions	Separate memory for data and instructions
Data Paths	Single data bus for both instructions and data	Separate data and instruction buses
Performance	Slower due to shared memory and bus	Faster due to parallel data paths
Complexity	Simpler, lower cost	More complex, higher cost
Examples	General-purpose computers, personal computers, embedded systems	Digital Signal Processors (DSP), microcontrollers (e.g., ARM Cortex)

Outline

Recall: Microprocessors and Microcontrollers

- **Groups of Instructions:**
  - **CISC (Complex Instruction Set Computer)**
    - **Many clock cycles**
    - **Better use of RAM**
    - **Example: AMD and Intel x86**
  - **RISC (Reduced Instruction Set Computer)**
    - **Single clock cycle**
    - **Intense RAM Access**
    - **Example: ARM and SPARC**

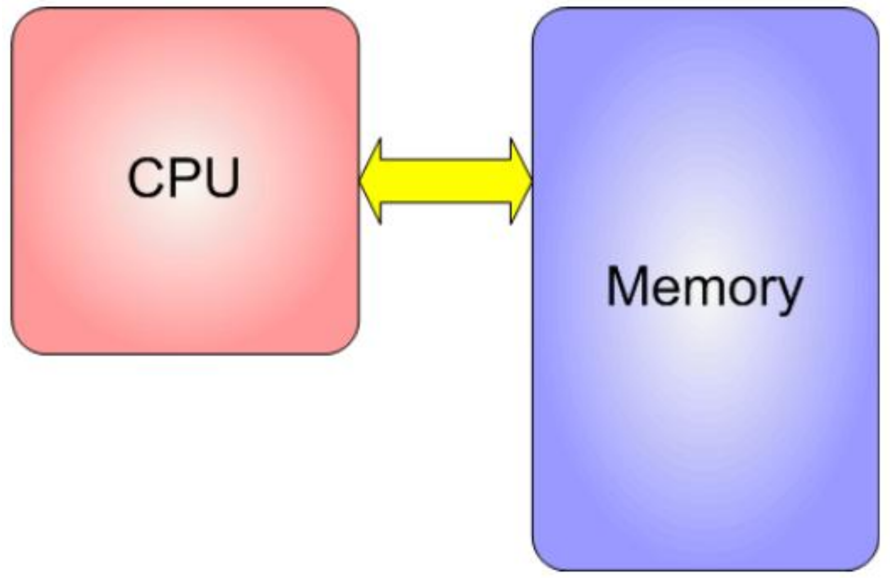


Fig.: Von Neumann

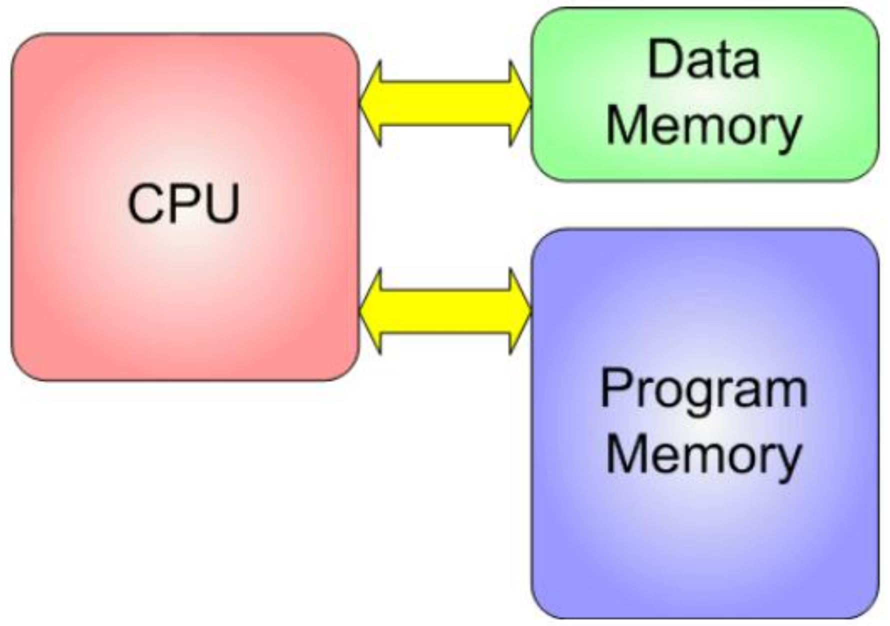


Fig.: Harvard

https://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/risccisc/

CISC	RISC
Emphasis on hardware	Emphasis on software
Includes multi-clock complex instructions	Single-clock, reduced instruction only
Memory-to-memory: "LOAD" and "STORE" incorporated in instructions	Register to register: "LOAD" and "STORE" are independent instructions
Small code sizes, high cycles per second	Low cycles per second, large code sizes
Transistors used for storing complex instructions	Spends more transistors on memory registers

```
LOAD A, 2:3
LOAD B, 5:2
PROD A, B
STORE 2:3, A
```

```
MULT 2:3, 5:2
```

Recall: Microprocessors and Microcontrollers

- Microprocessor
  - Intel 4004, year 1971
  - External: Memory and I/O
  - Von Neumann arch

- Microcontroller
  - TMS 1000, year 1971
  - Computer-on-a-chip concept
  - Many internal resources (memory and I/O)
  - Embedded systems applications
  - Harvard arch

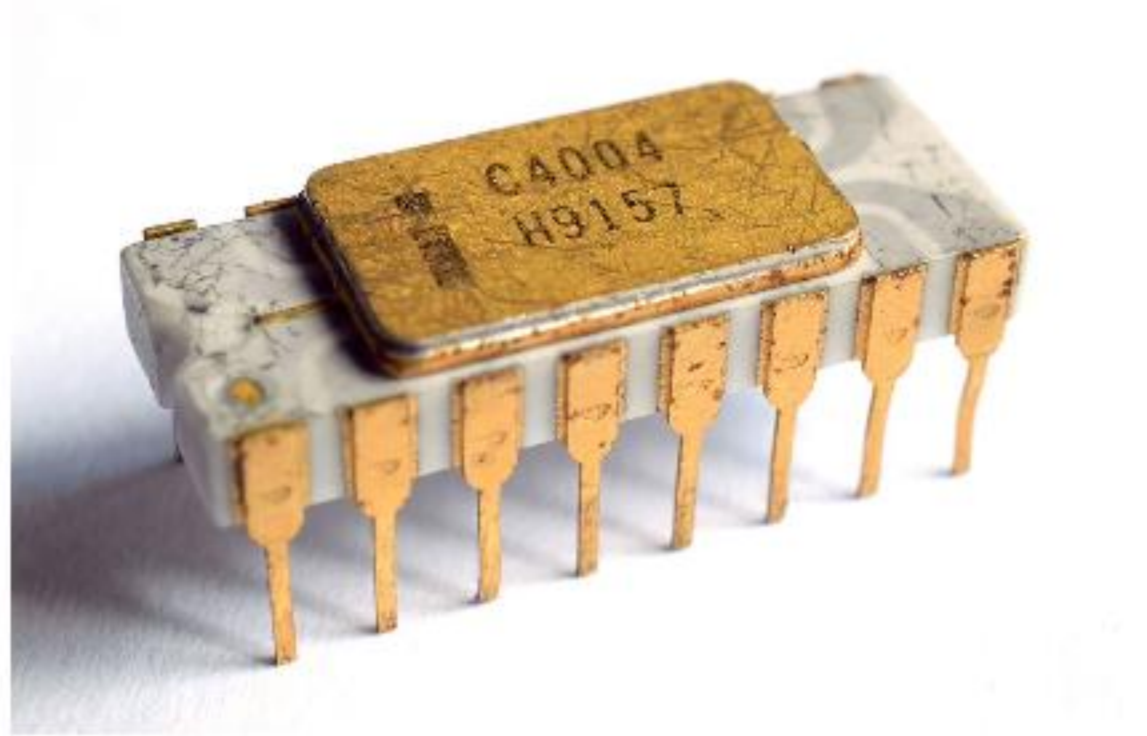


Fig.: Intel 4004 por fora

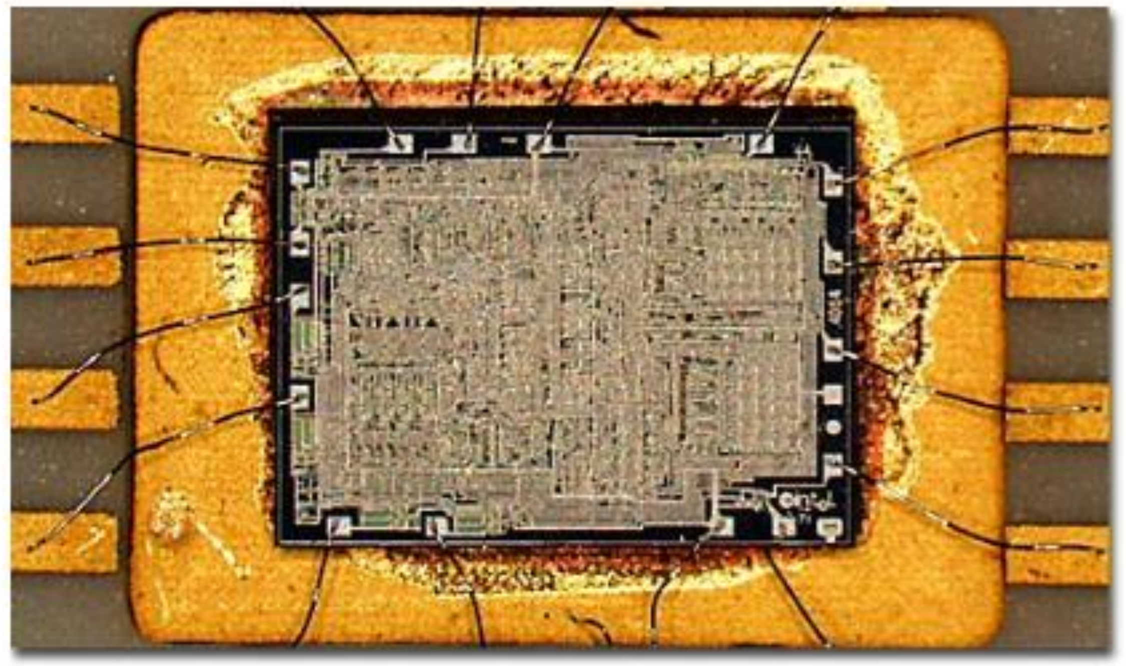


Fig.: Intel 4004 por dentro

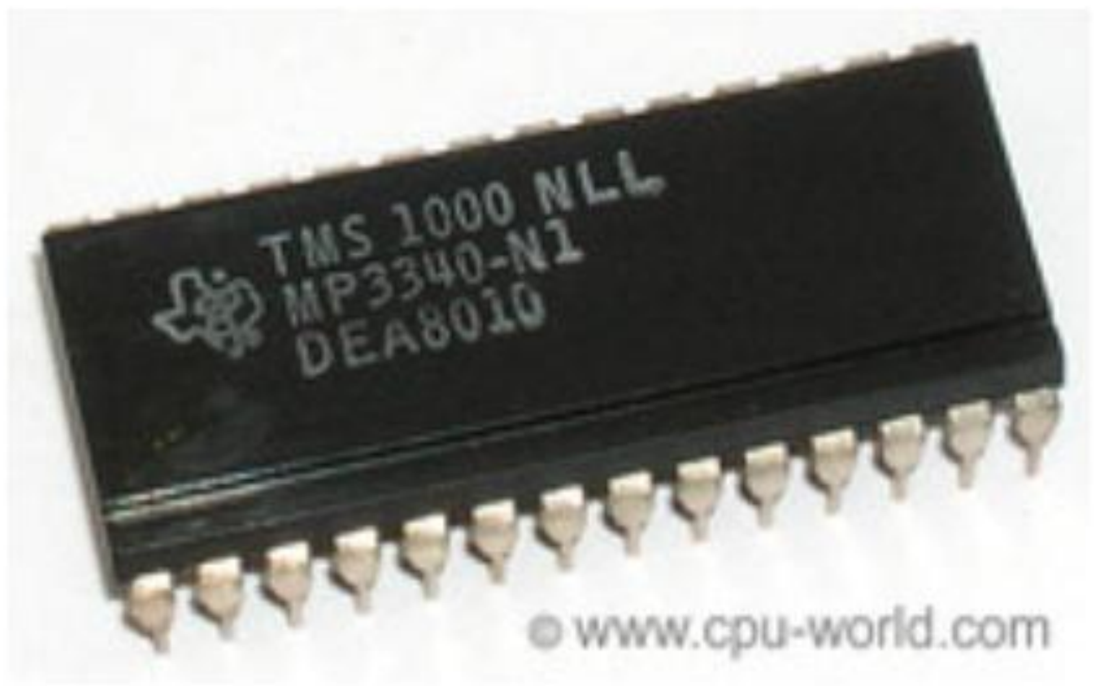


Fig.: TMS 1000 por fora

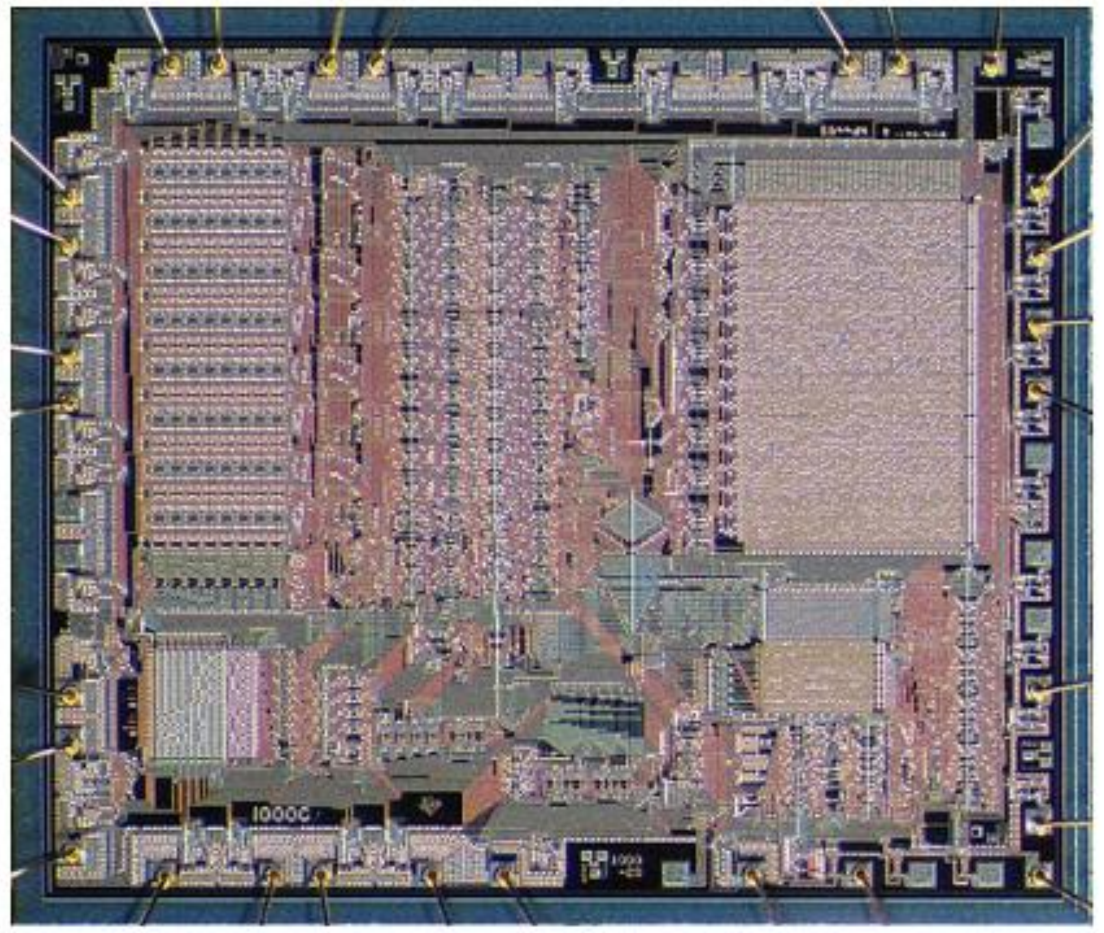


Fig.: TMS 1000 por dentro

Aspect	Microprocessor (MPU)	Microcontroller (MCU)
Definition	CPU only; needs external components (RAM, ROM, I/O)	Complete system on a chip (CPU + RAM + ROM + I/O)
Integration	Low (only CPU inside)	High (CPU, memory, peripherals inside)
Primary Use	Complex computing tasks (PCs, servers)	Embedded systems (appliances, robots)
Power Consumption	Higher	Lower
Processing Power	High-speed, powerful multitasking	Moderate speed, optimized for control tasks
Memory Management	External RAM/ROM	Internal RAM/ROM
Cost	More expensive	Cheaper
Typical Examples	Intel Core i7, AMD Ryzen	ATmega328P (Arduino), STM32, ESP32
Programming Complexity	Complex operating systems (Linux, Windows)	Simple firmware or real-time OS

Recall: Microprocessors and Microcontrollers

- Microcontroller
  - Common PINS:
    - Vin – input voltaje
    - GND – Ground
    - RST – Reset
    - CLK – Clock
    - TX – Transmit
    - GPIO – General purpose Input/output
    - I2C – Inter-Integrated Circuit

Fig.: TMS 1000 por fora

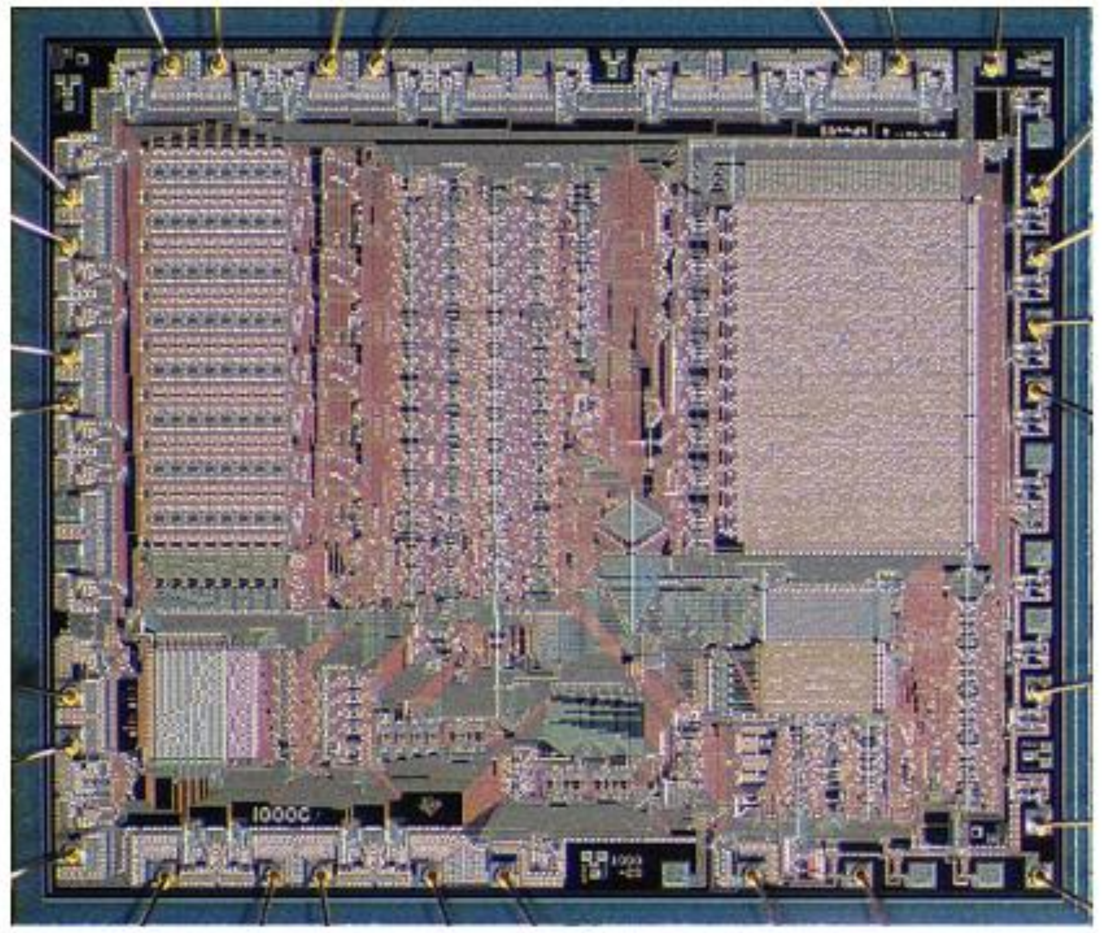
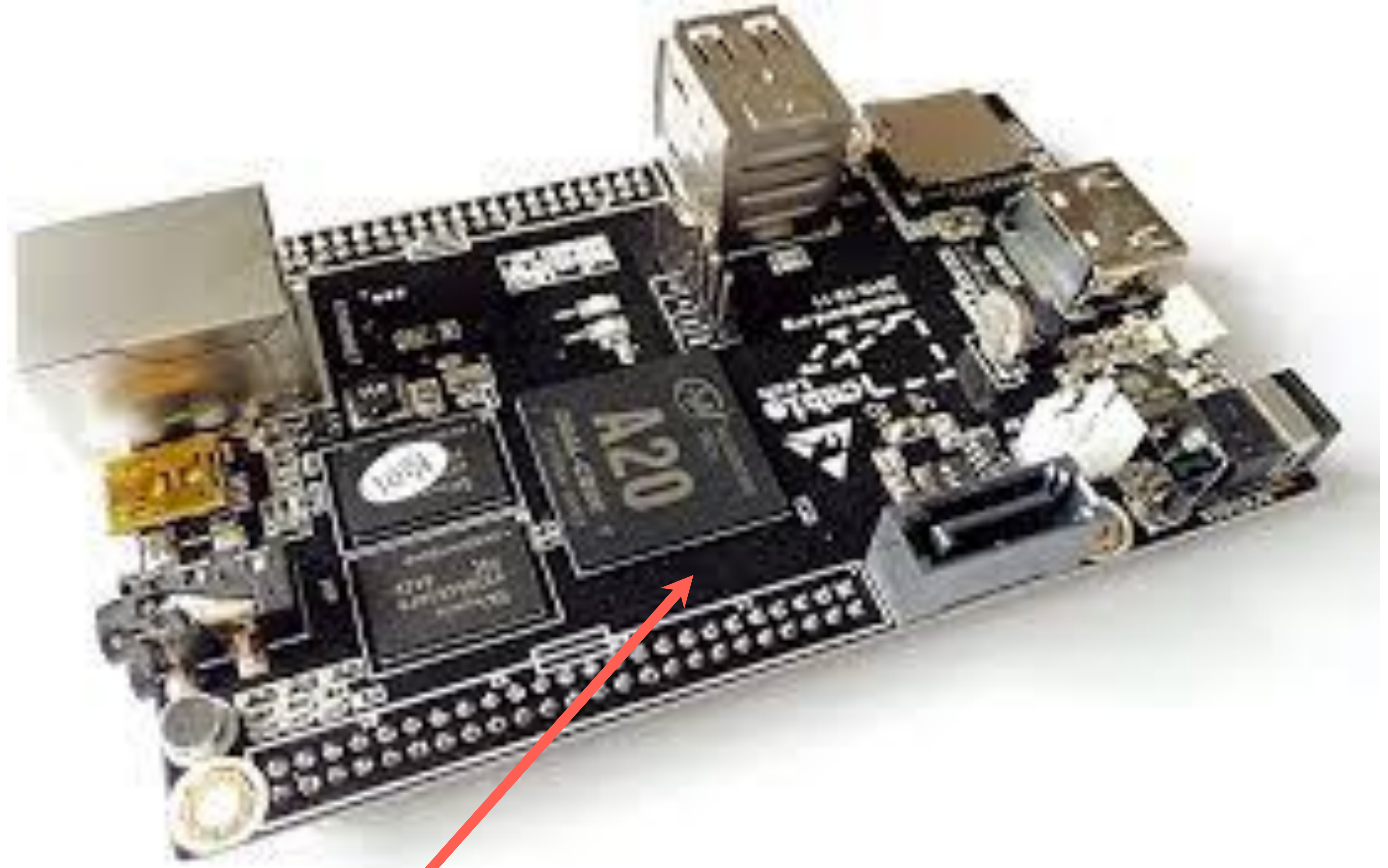


Fig.: TMS 1000 por dentro

This is a pin

Recall: Microprocessors and Microcontrollers

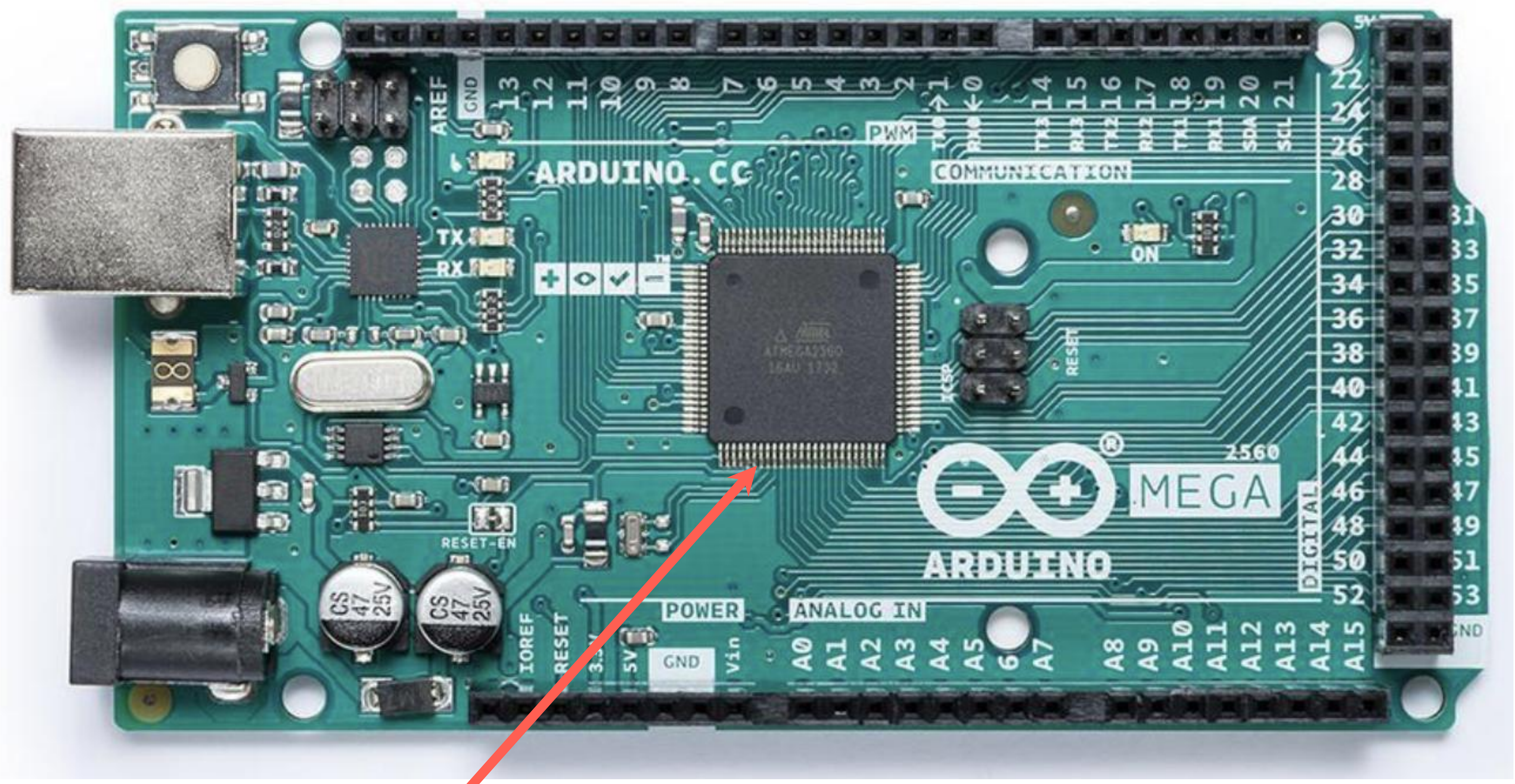
- Examples of Microprocessors:
  - Intel Quark SoC
  - Broadcom BCM 2835
  - ARM 11, Cortex A20



Pins are soldered to board for developing purposes!

Recall: Microprocessors and Microcontrollers

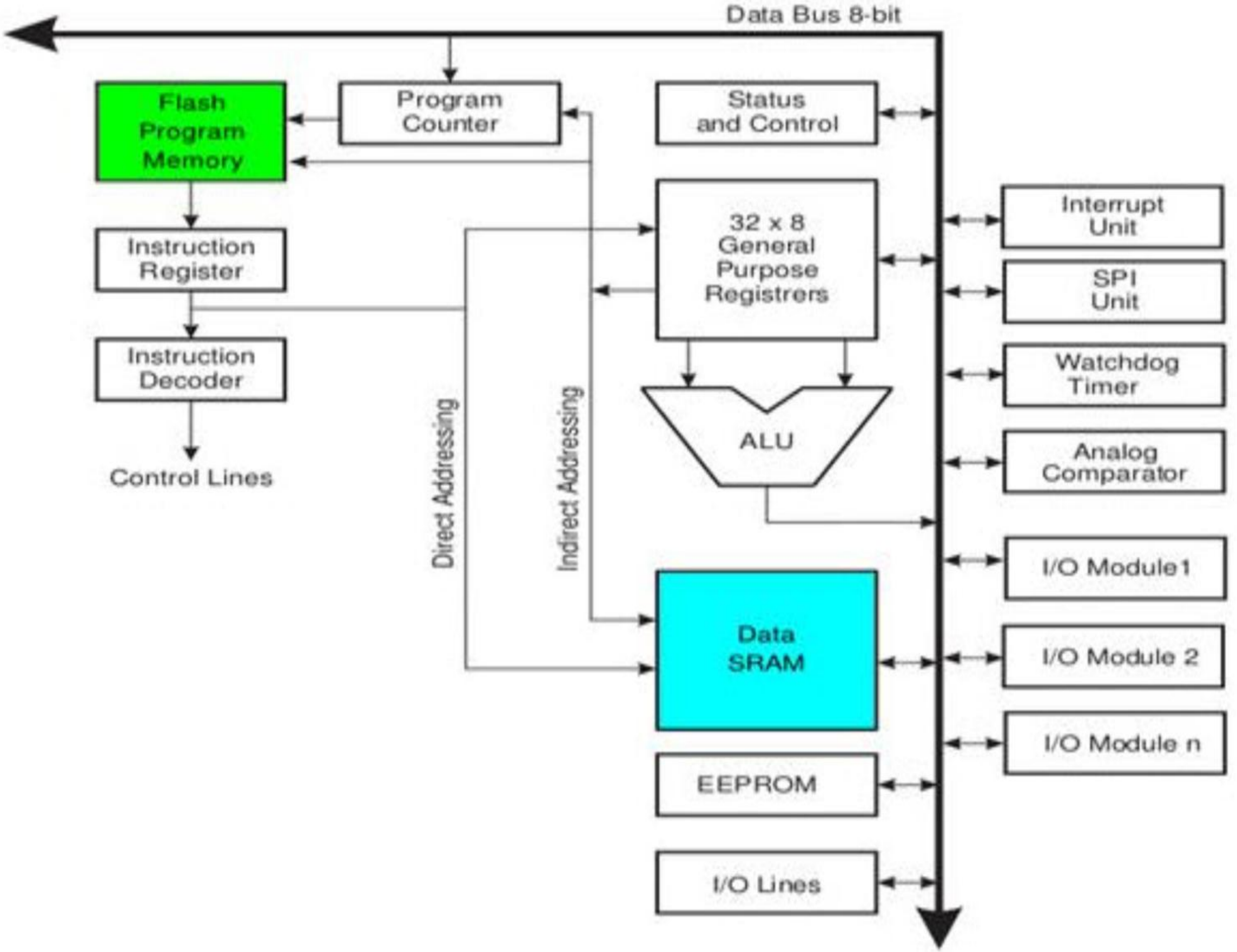
- Examples of Microcontrollers:
  - Arduino
  - Atmel AVR
  - PIC (Microchip Technology)



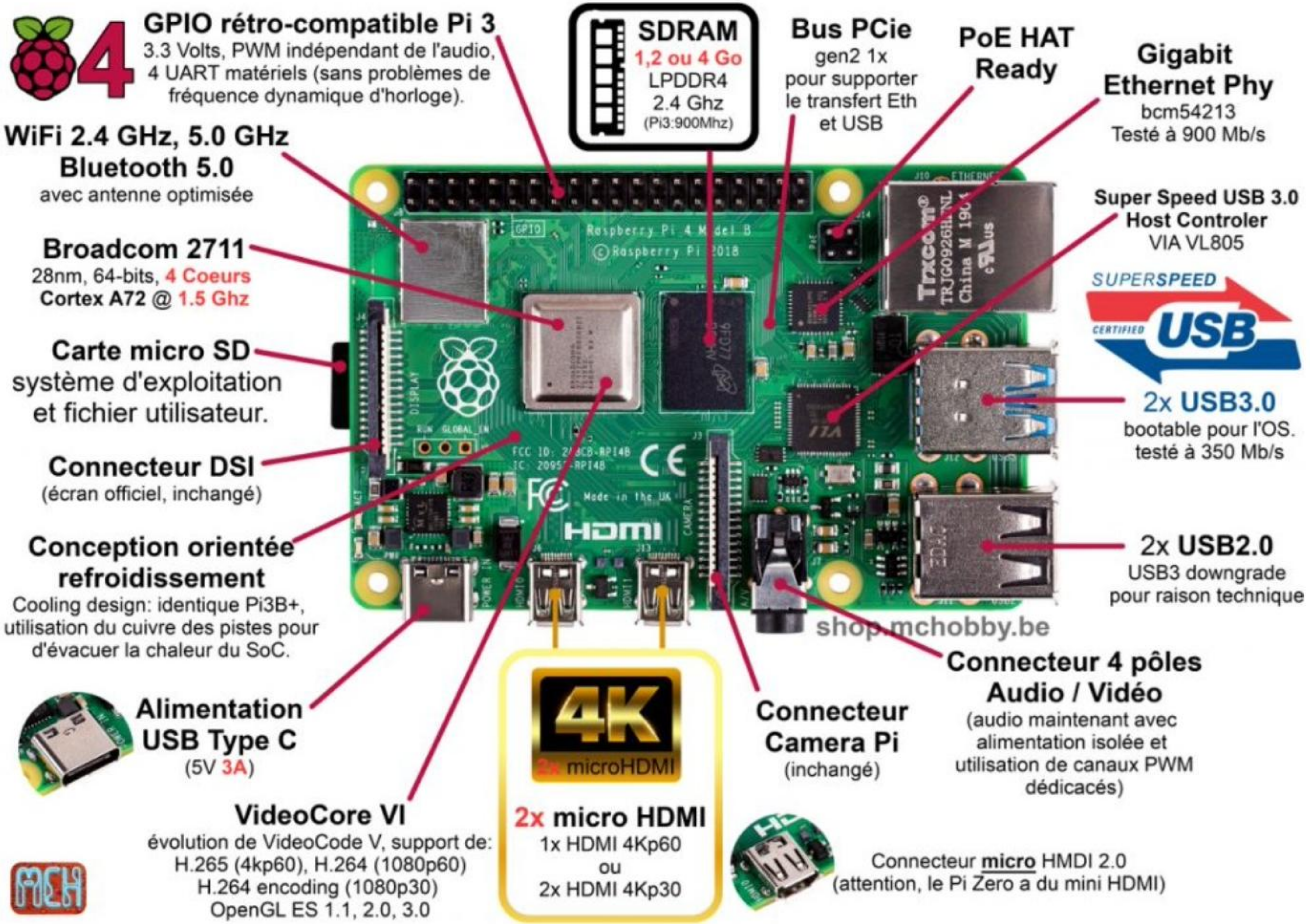
Pins are soldered to board for developing purposes!

Recall: Microprocessors and Microcontrollers

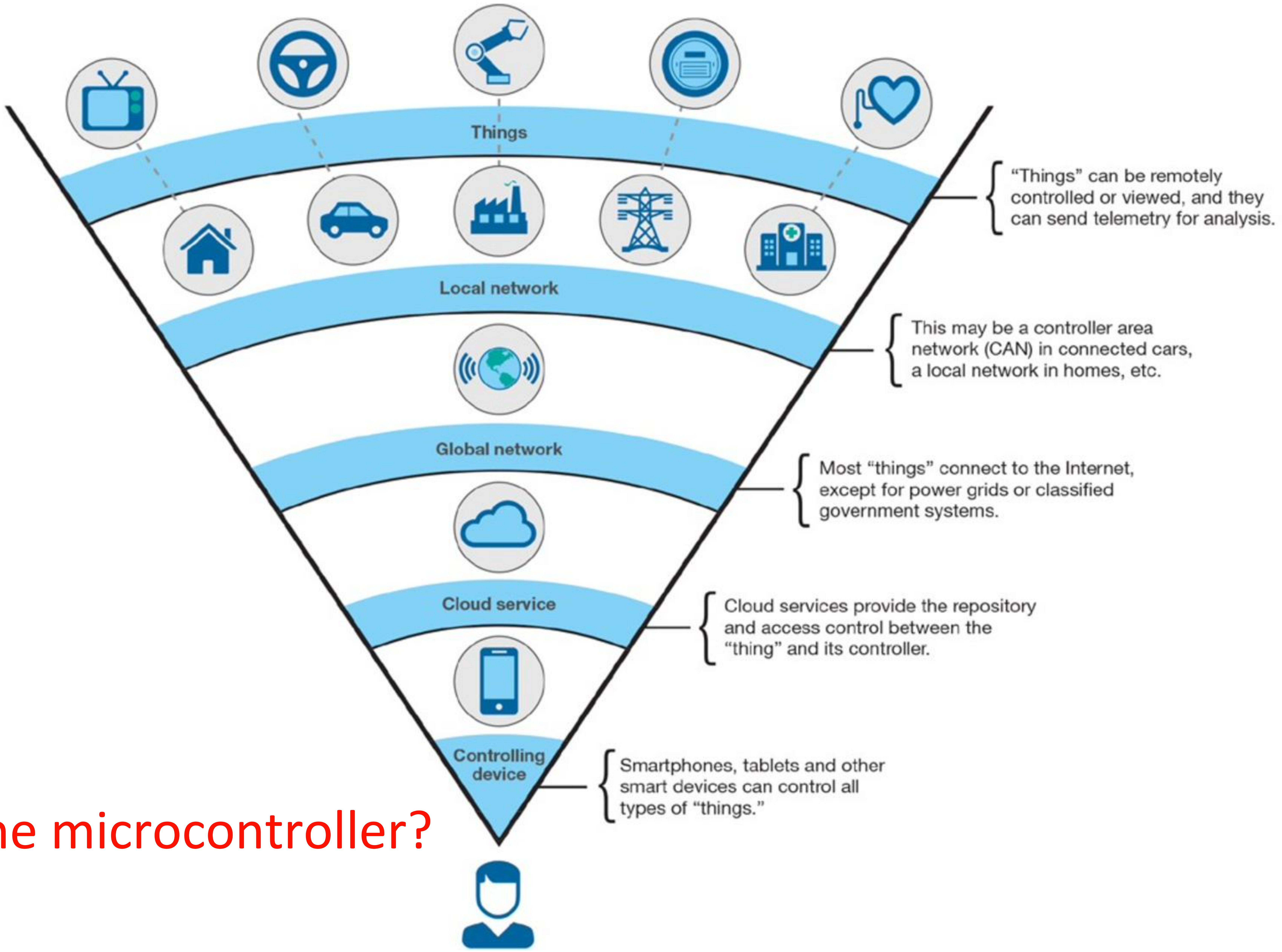
- Arduino Architecture



Development boards: Raspberry 4

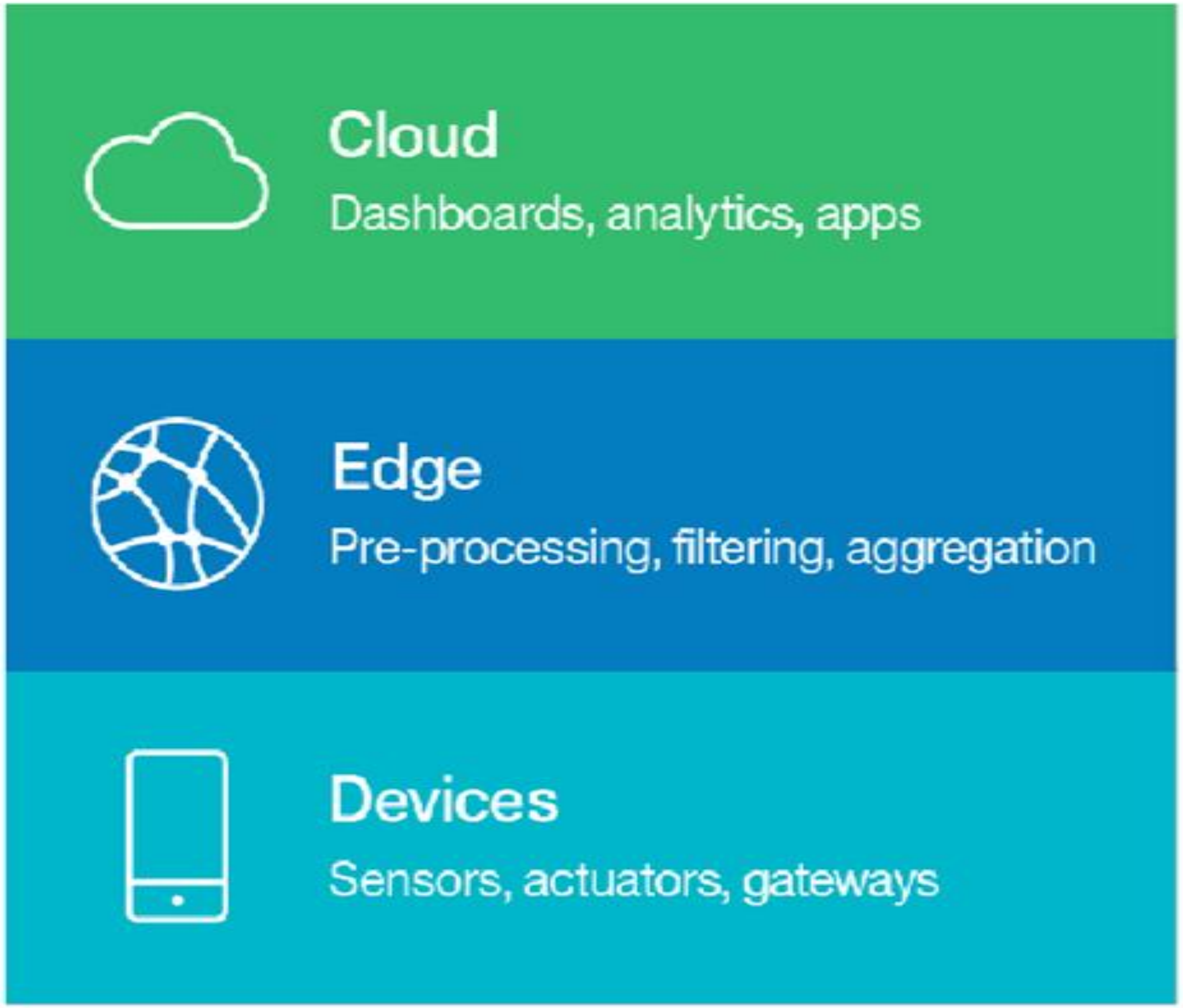


IoT Architecture



- Q. Where is the microcontroller?

IoT Architecture



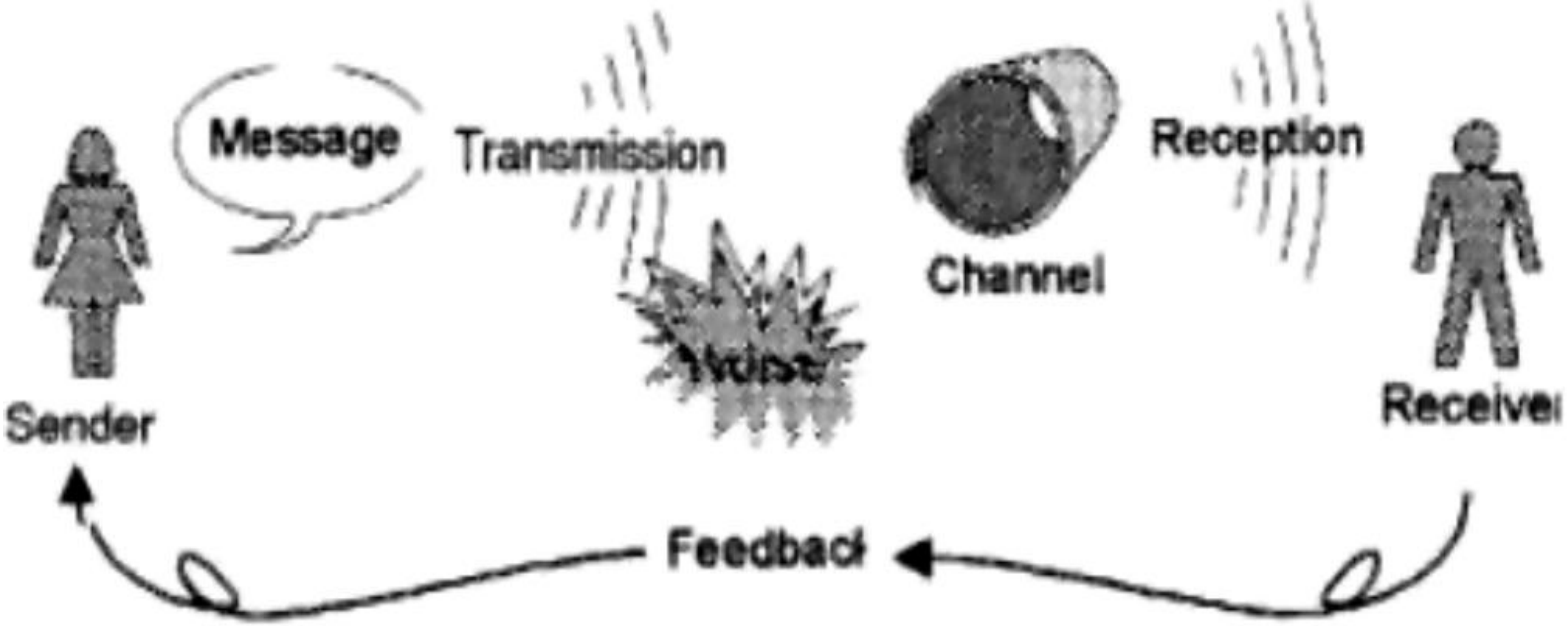
- Q. Where is the microcontroller?
- Q. Where can we find a microprocessor?

Outline

- Introduction
- Microcontrollers and Microprocessors
- Introduction to Communications and Routing
- Transmission Models in IoT
- Conclusions

Introduction to Communications

- Really old terminology:



Fonte: Claude Shannon e Warren Weaver, "The Mathematical Theory of Communication", University of Illinois Press, Urbana, IL, 1949

Introduction to Communications

- Transmission Technologies
  - Consider heterogeneous devices and subsystems
  - Consider power availability
- Each application has its own flavor
  - No silver bullets!
- Three main features:

Available energy

Transmission velocity

Distance between devices

Introduction to Communications

- What are we looking for in IoT regarding Communications?

  - Low energy consumption
  - High transmission velocity
  - Long distance

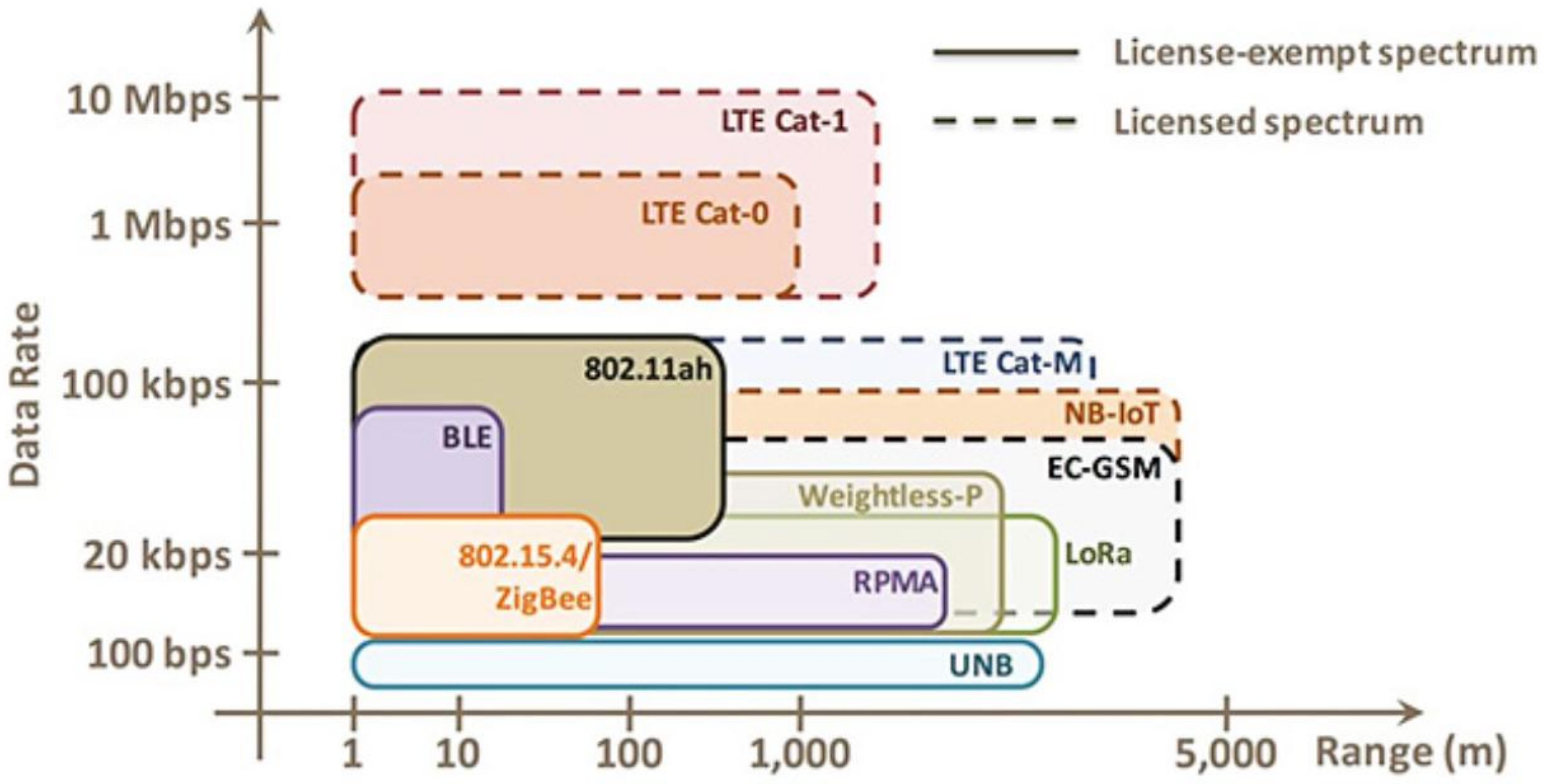
- Trade-offs:
  - Low energy consumption -> long distance
  - High velocity -> long distance

  - How to manage trade-offs? Multi-objective optimization?

Types of communication

- Distance
  - Short distance
  - Long distance
- Electromagnetic nature
  - Confined media (wired)
  - Radiation (Wireless)
- Architecture
  - LAN
  - WAN

Communication options: Examples



Communication technologies in IoT

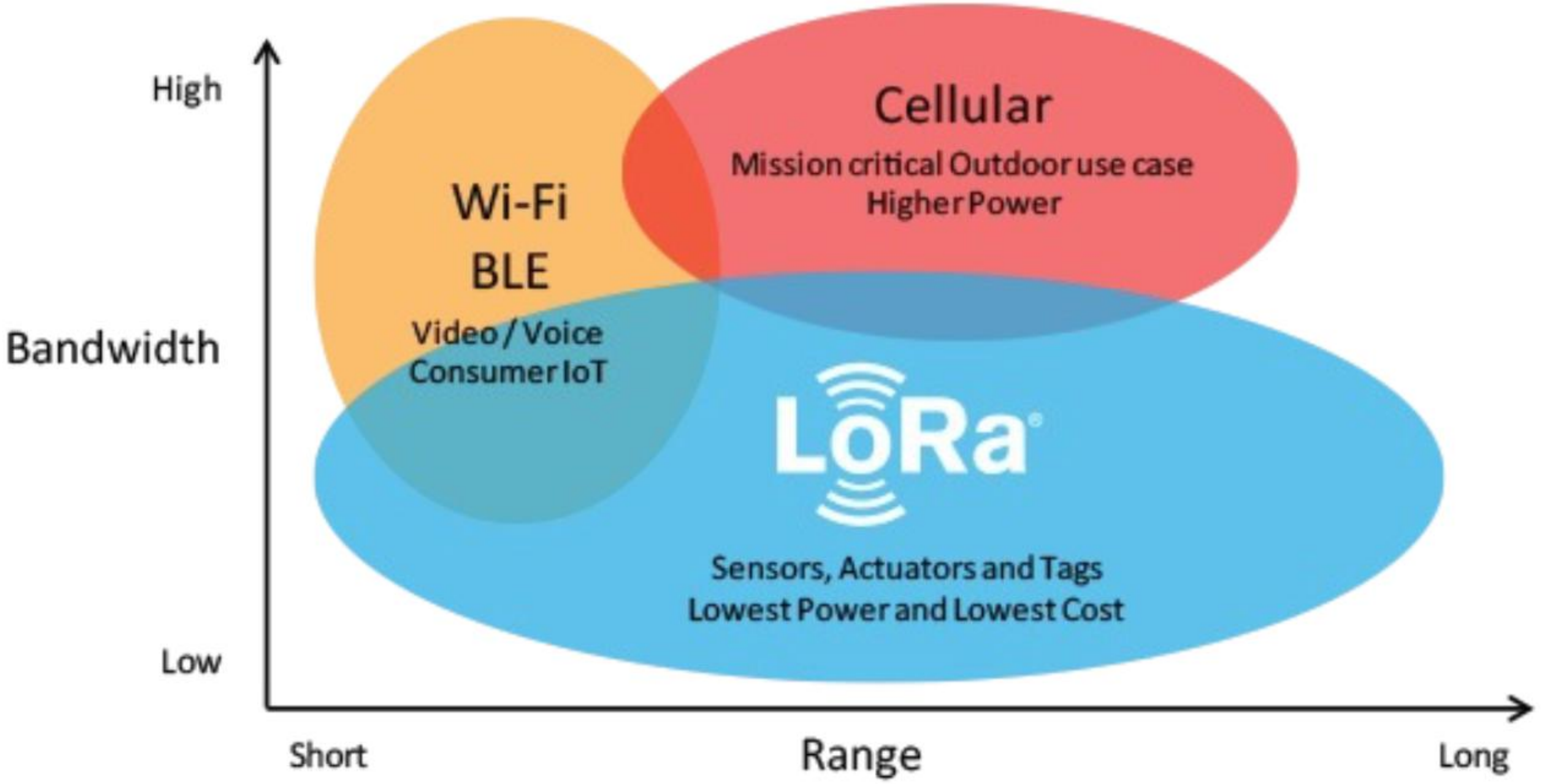
Technological solutions that are sort of standardized

Communication technologies in IoT

- **Wifi:** High velocity, high energy consumption
- Zigbee: low energy consumption, low bandwidth (1 Mbps), short distance (aprox. 10 m)
- **Bluetooth:** More energy consumption than Zigbee, short distance, higher velocity
- **SigFox:** (This is not an open technology) low energy consumption and long distance with more than 100 b/s
- **LoRa:** long distance (10km), low energy consumption, small bandwidth (50 Kb/s)
- **NBIoT:** long distance, low energy consumption and around 250 Kb/s
- **3G, 4G, LTE, 5G:** long distance, high energy consumption, manageable bandwidth… Problems with base station implementation.

Communication technologies in IoT

- **Other technologies:**
  - **RFID:** examples are cars, stores. Low energy consumption, magnetic induction
  - **NFC:** examples are cellphones and credit cards – Low distance (centimeters) acceptable bandwidth and low energy consumption.



Communication technologies in IoT

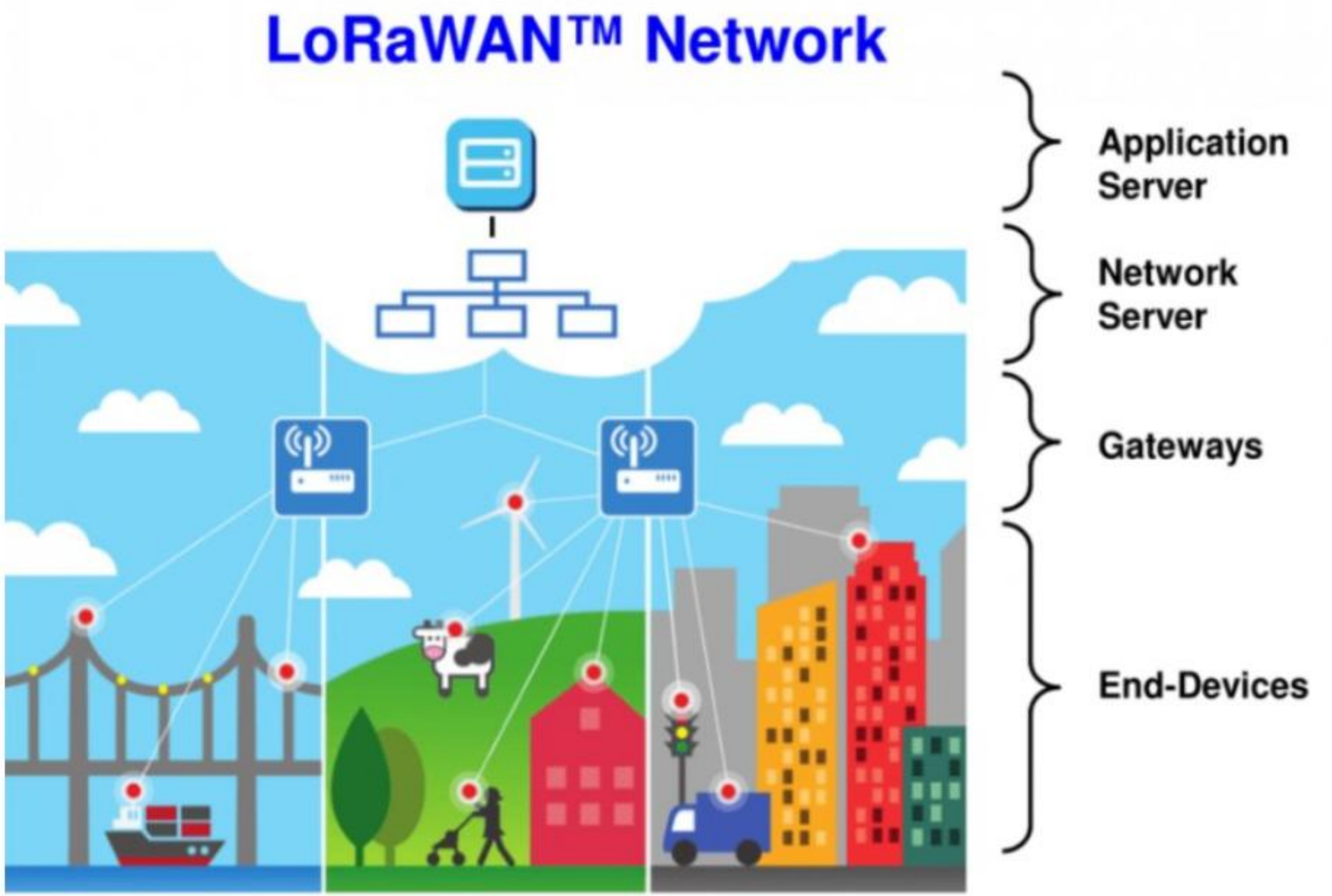
LoRaWAN:

Main Characteristics

- **Long range**: Up to 15 km in open fields.
- **Low energy consumption**: Ideal for devices powered by long-life batteries.
- **Low data rate**: Perfect for small data packets.
- **Use of unlicensed bands**: No license required (e.g., 868 MHz in Europe, 915 MHz in America).
- **High penetration**:
Good transmission even in dense urban environments.

Modulation : CSS (Chirp Spread Spectrum)

LoRaWAN:

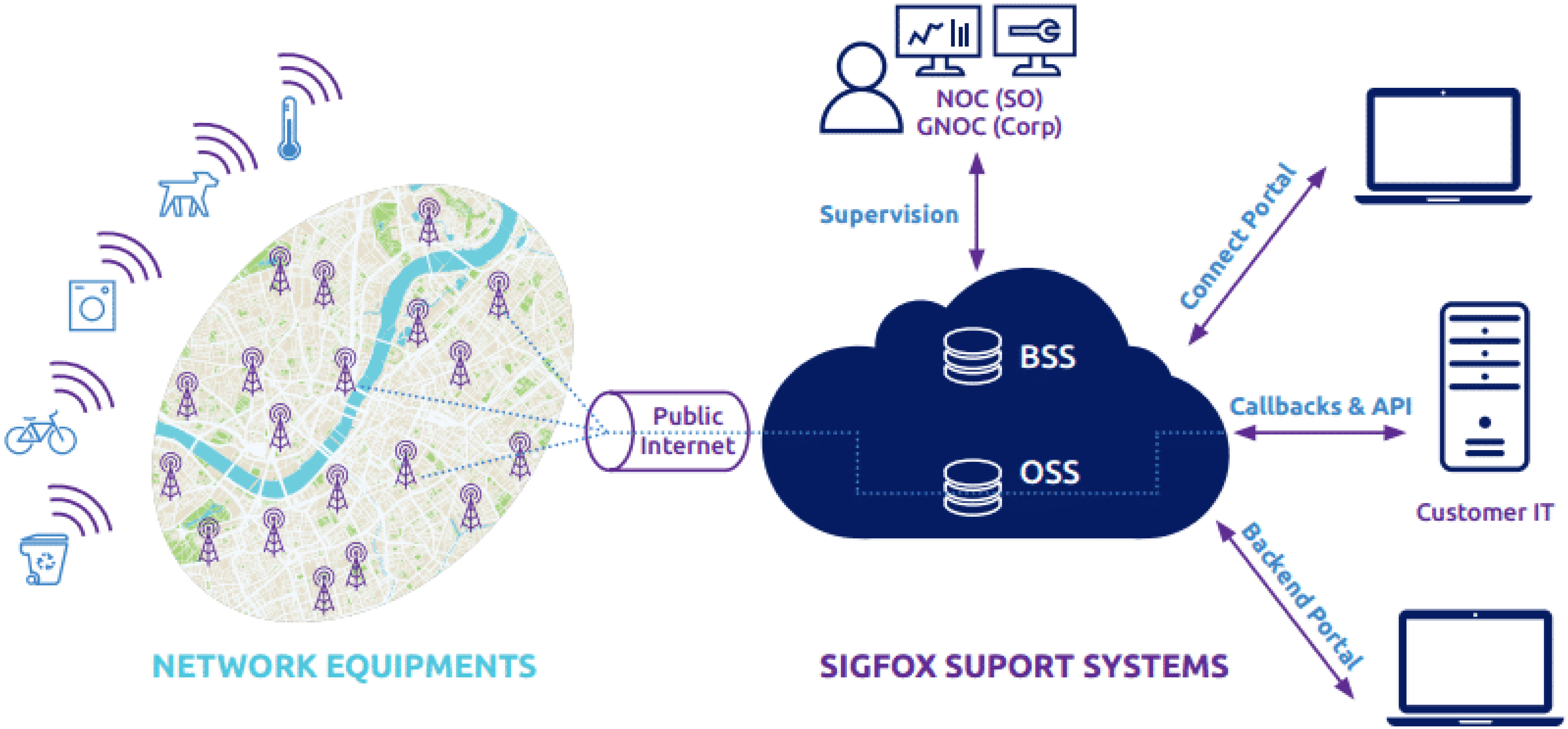

LoRaWAN™ Network
Application Server
Network Server
Gateways
End-Devices

Sigfox:

- **Ultra-low energy consumption**: Devices can last **years** on a small battery.
- **Long range**: Coverage of **up to 40–50 km** in rural areas and **3–10 km** in urban environments.
- **Small payloads**: Maximum of **12 bytes** per uplink message, and **8 bytes** per downlink message.
- **Limited daily messages**: Up to **140 uplink messages** and **4 downlink messages** per device per day.
- **Centralized network**: Devices connect to **public Sigfox base stations** managed by Sigfox or its partners.
- **Use of unlicensed ISM bands**: Operates mainly on **868 MHz (Europe)** and **915 MHz (Americas)**.
- **Low cost**: Affordable modules and subscription-based network access.
- **High penetration**: Good signal transmission through dense urban areas and indoors.

Communication technologies in IoT

- Sigfox arch

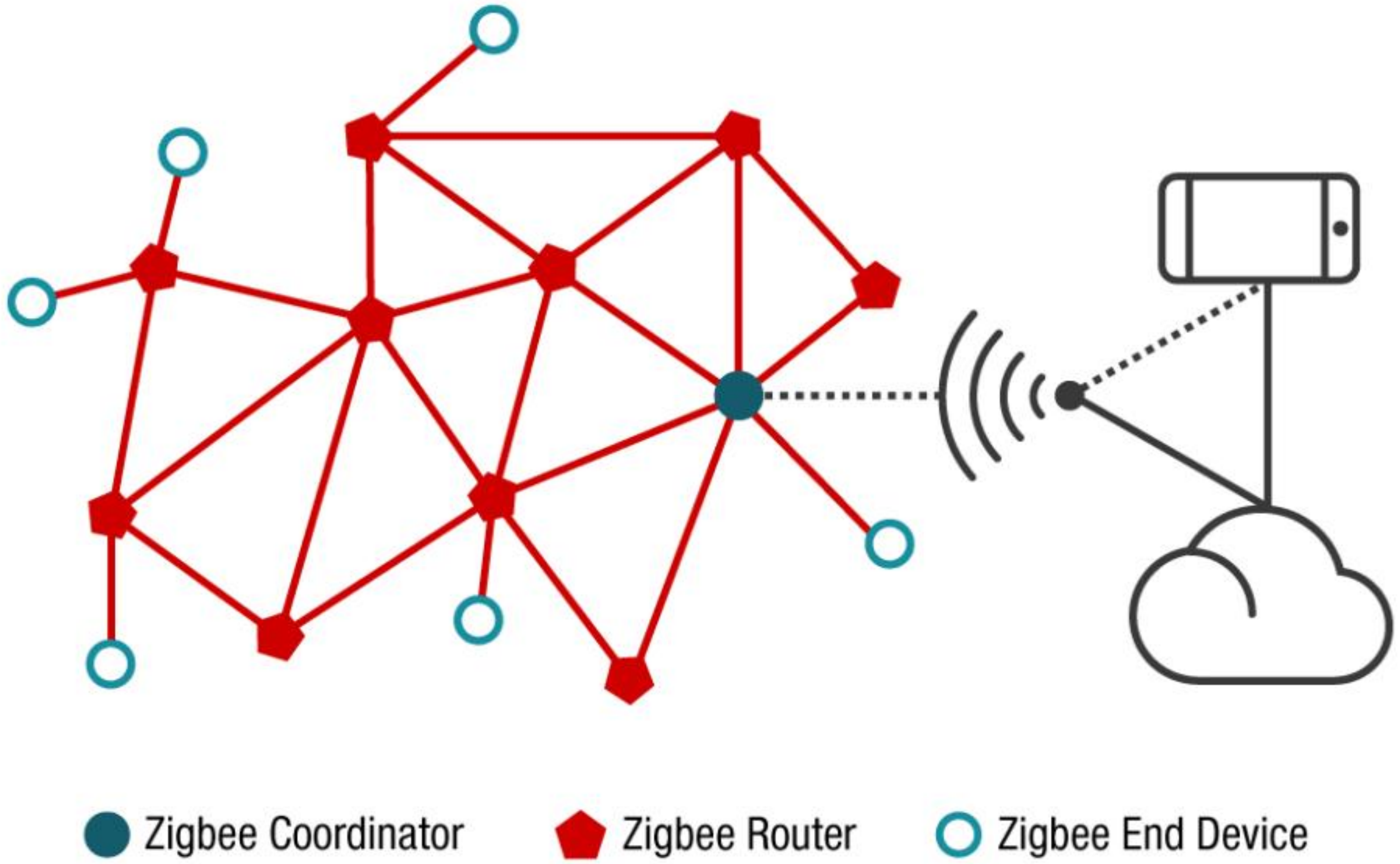


Zigbee:

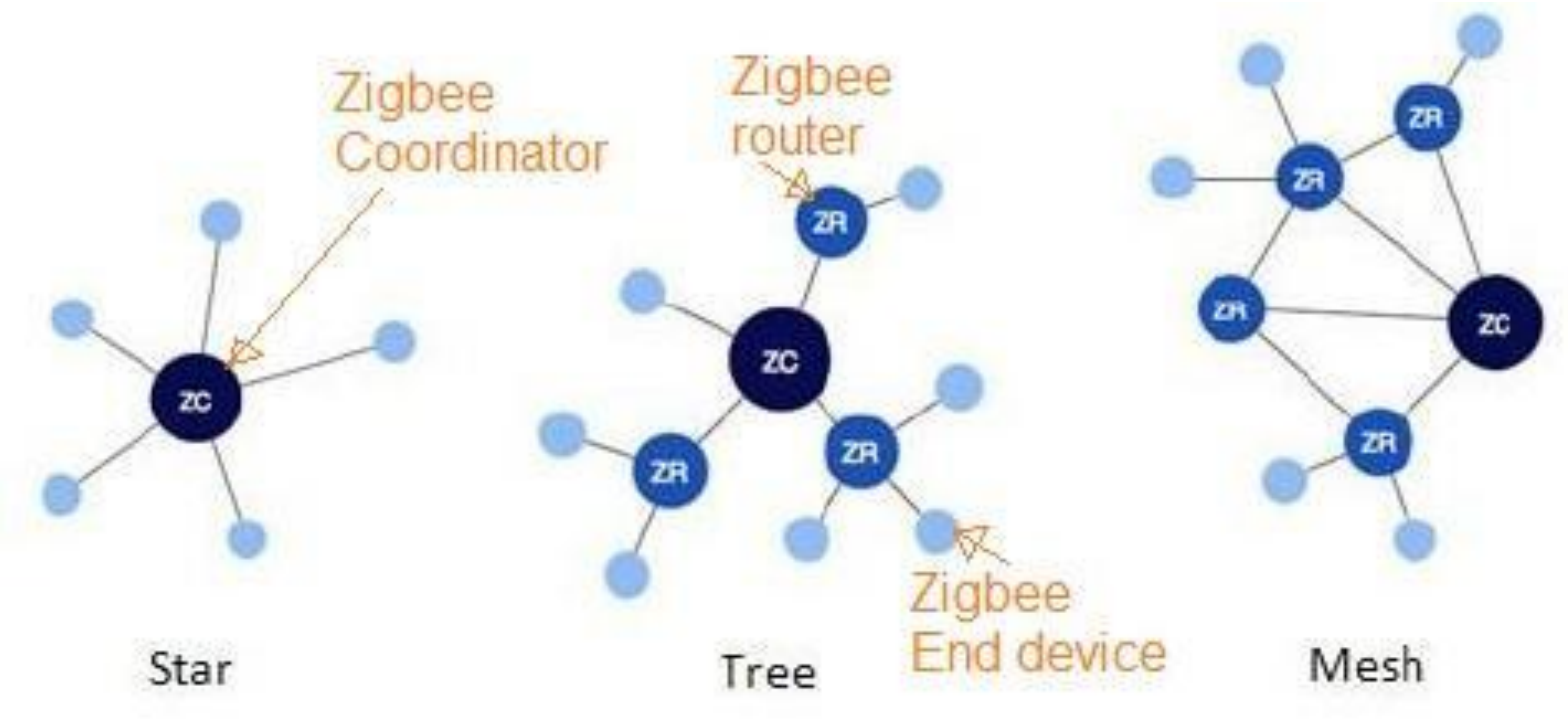
- **Ultra-low energy consumption**: Devices can run for years on small batteries due to low power requirements.
- **Short range**: Typically supports a communication range of 10 to 100 meters, with mesh networking extending range.
- **Small payloads**: Maximum of 250 kbps data rate, suitable for low-bandwidth applications like sensor data.
- **Limited daily messages**: Zigbee is optimized for low data rates but supports frequent communication with many devices, ideal for control and monitoring tasks.
- **Centralized or decentralized network**: Zigbee networks can be coordinated by a central coordinator, or they can operate with multiple routers in a mesh network.
- **Use of unlicensed ISM bands**: Operates mainly in the 2.4 GHz (globally), 900 MHz (North America), and 868 MHz (Europe) ISM bands.
- **Low cost**: Affordable for mass deployment, with low-cost modules and open standard protocols.
- **High penetration**: Zigbee's mesh networking allows good signal penetration even in dense urban areas and indoors, thanks to relay functionality between devices.

Communication technologies in IoT

- **ZigBee architecture**
  - End devices can be sensors or actuators
  - Topologies can be degined using low energy consumption router
    - Low energy sensor network



Network Topologies: Star, Tree, and Mesh

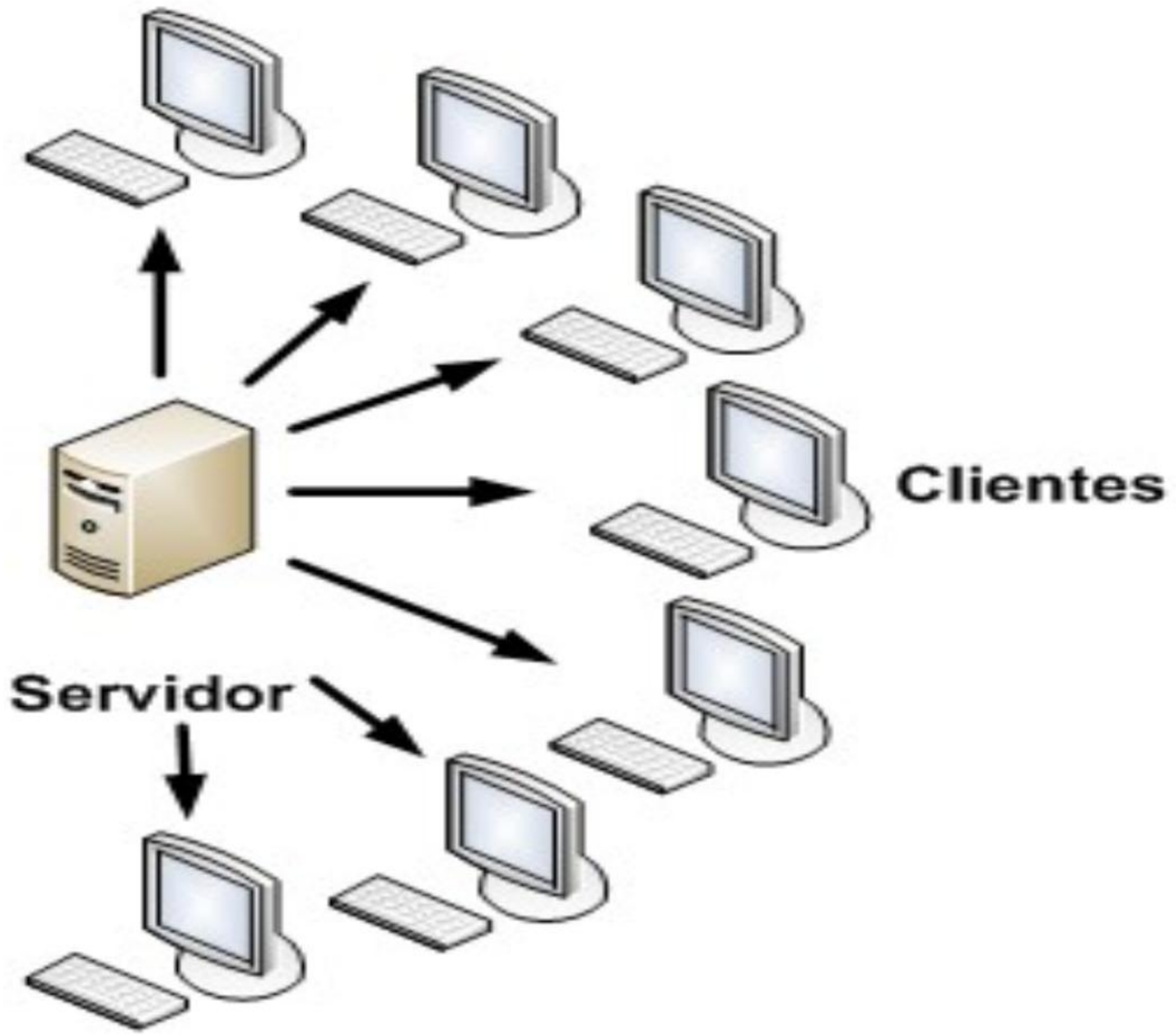


Outline

- Introduction
- Microcontrollers and Microprocessors
- Introduction to Communications and Routing
- Transmission Models in IoT
- Conclusions

Transmission Models in IoT

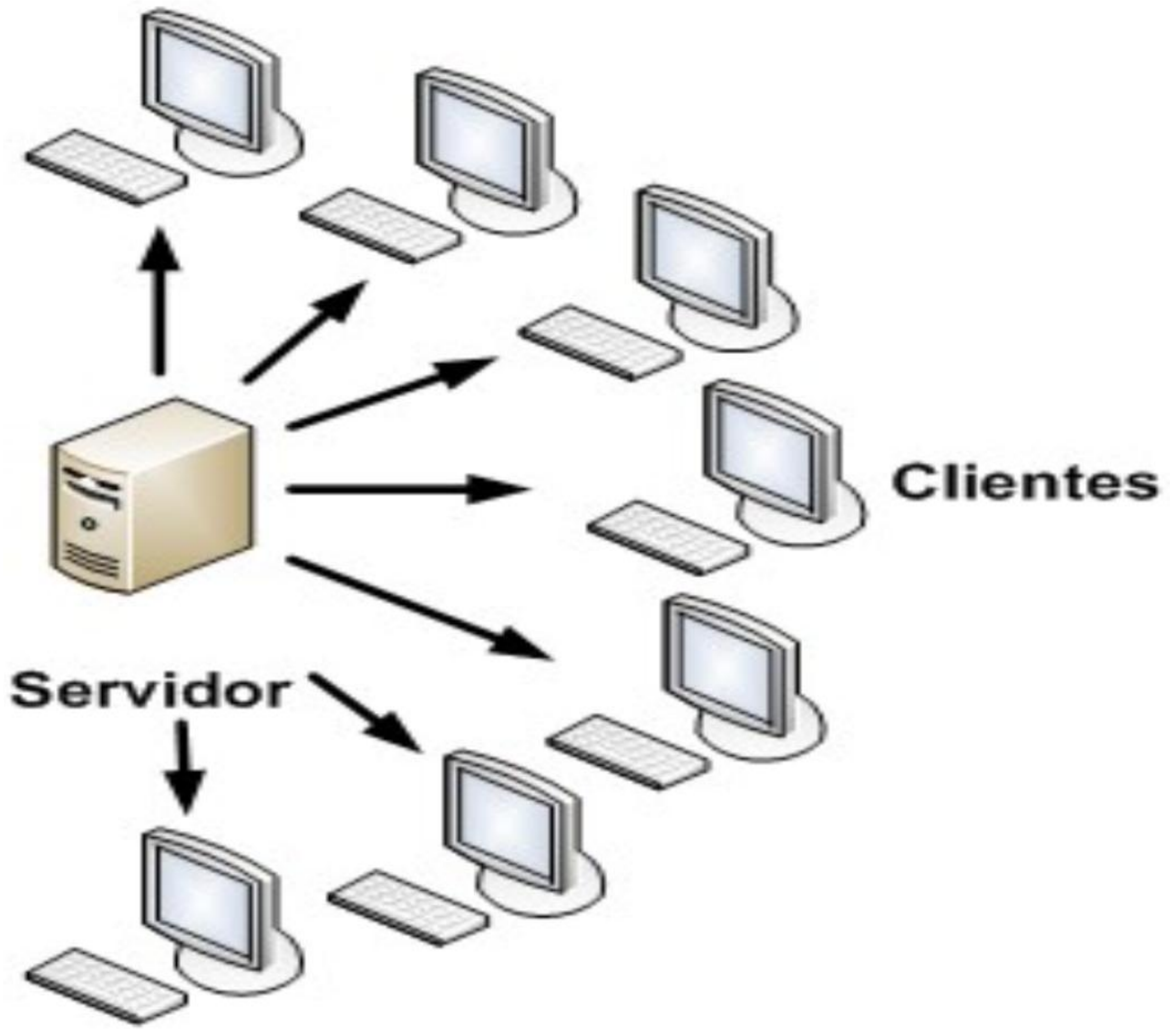
- Client-Server
  - Distributed application structure that distributes workloads between servers and clients
  - Client initiates communication
  - HTTP:
    - Using HTTP methods: GET, POST, PUT, DELETE
    - Use HTTP methods with other methods to construct services (RESTFull Services REST API)



Transmission Models in IoT

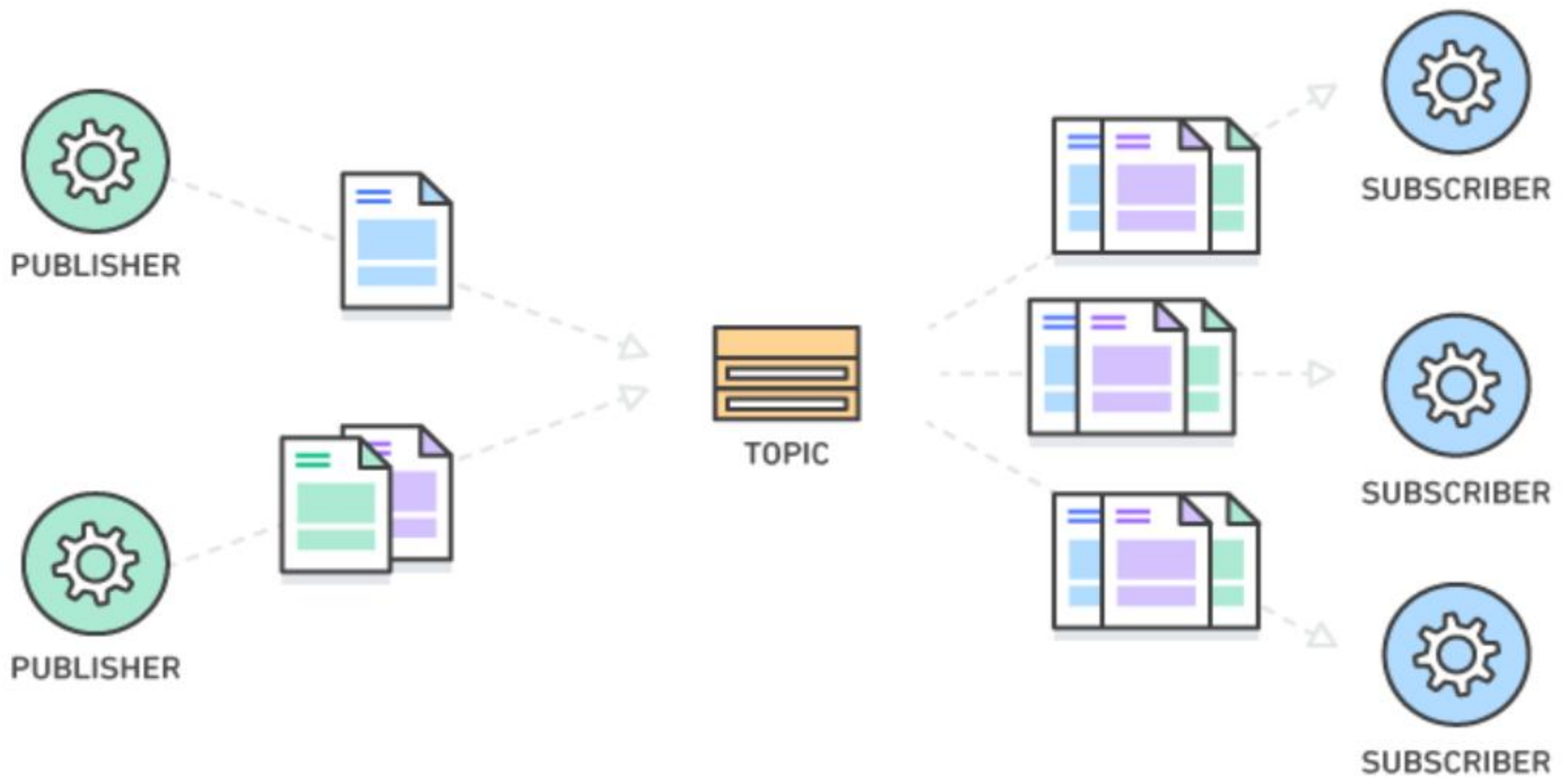
- **Client-Server**
  - Pros: Focuses processing on server. Clients can talk with one or more trusted servers
  - Cons: Heterogeneous “things” and scalability in IoT generate overhead. The server should be able to mantain simultaneous connections.

  - Scaling scenario (more sensors, actuators, data): Q. What happens with this paradigm?



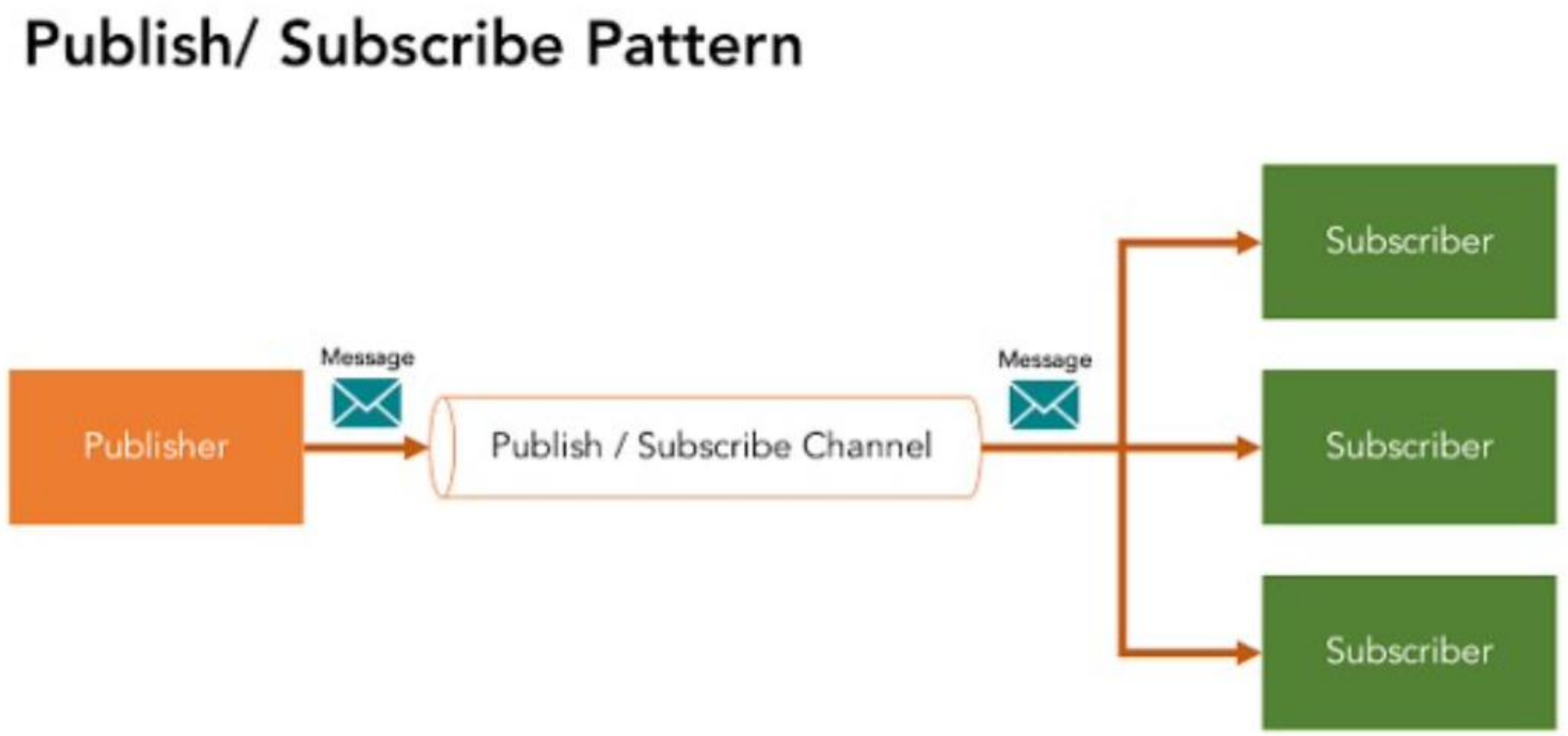
Transmission Models in IoT

- **Publish-Subscribe**
  - It's a message where editors do not program messages to specific clients, called subscribers, but they categorize published messages in clases without inputs from the subscribers.



Transmission Models in IoT

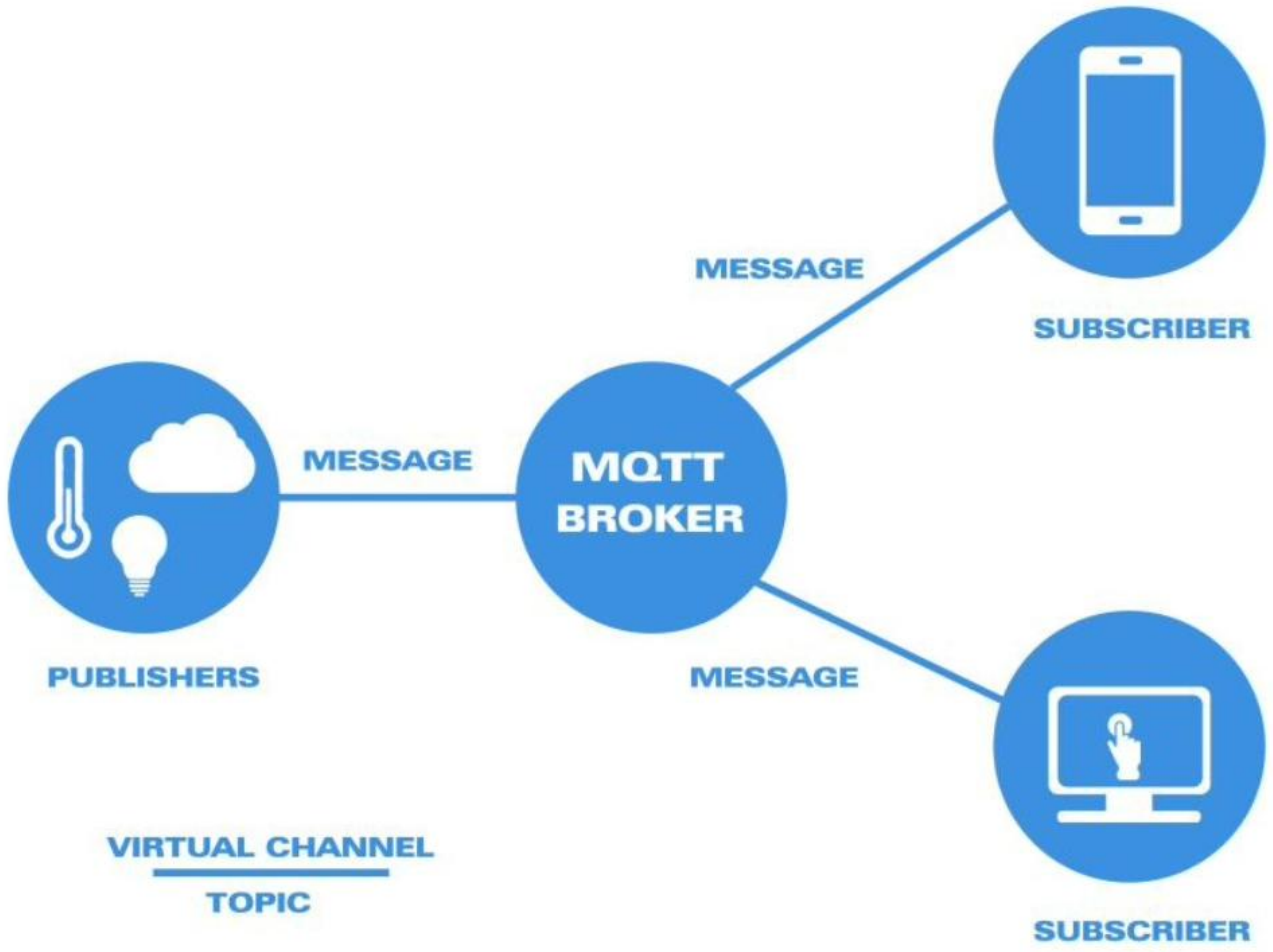
- **Publish-Subscribe**
  - An event is when the Publisher posts a message so that the subscribers are notified of new content.
    - A Publisher uses a communication channel and the message is copyed to each substscriber



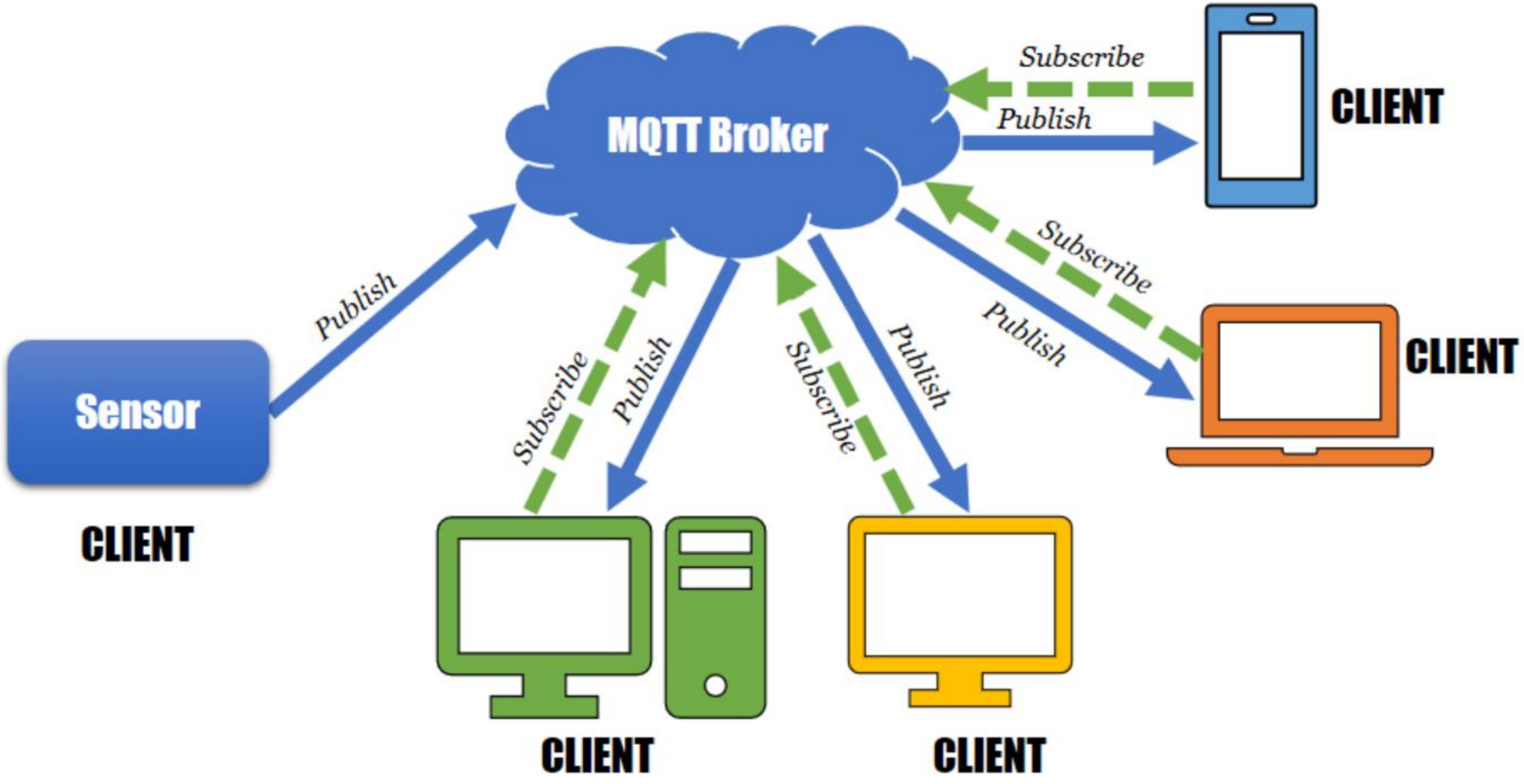
Transmission Models in IoT

- On an IoT Project:
  - **Broker:** Can be a function of the local Gateway ou be located in the cloud. It receives information from clients and redirects or processes.
  - **Publisher:** Connects to the broker to publish data
  - **Subscriber:** devices, applications, services that need data. They connect to the broker to be notified about new data.

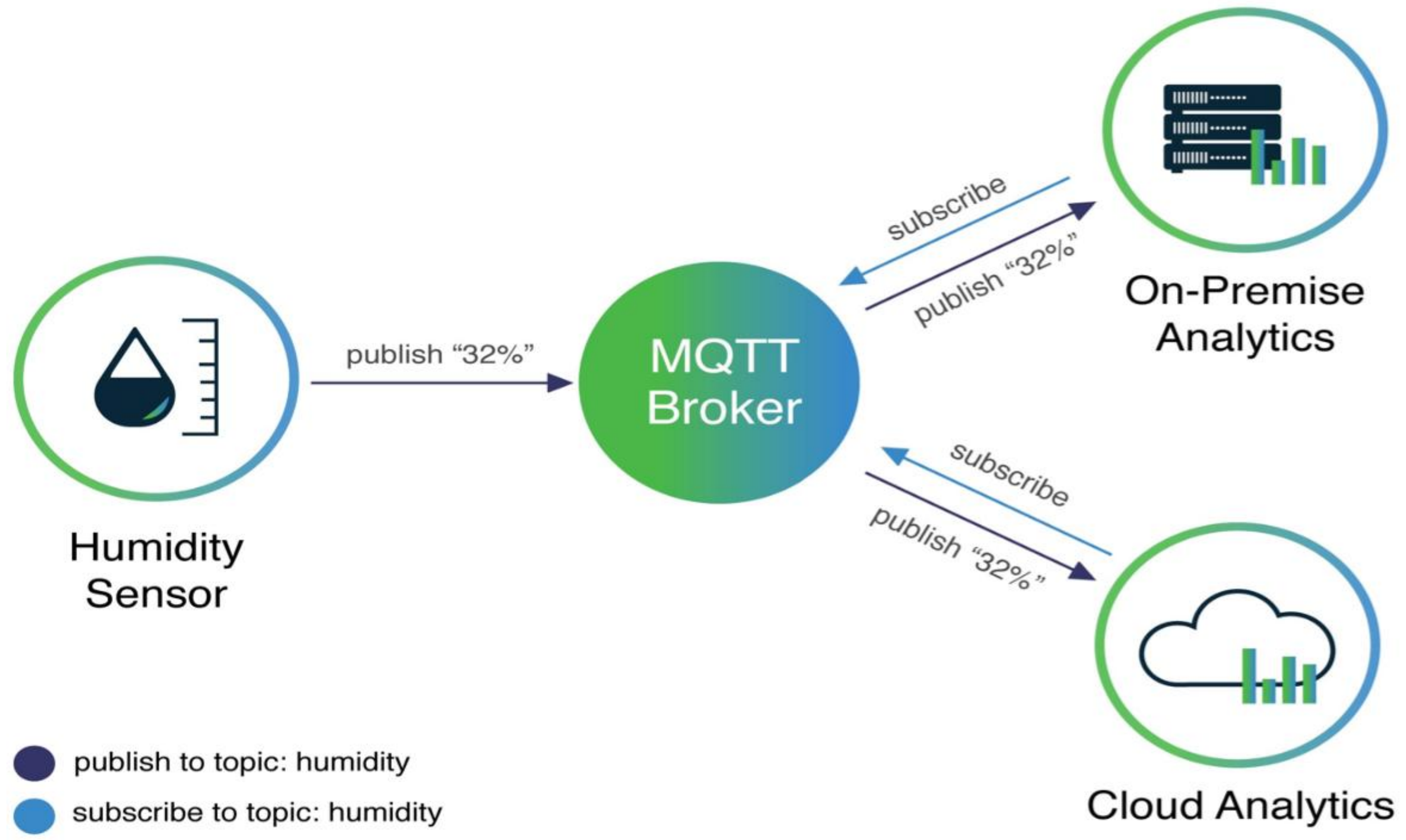
Transmission Models in IoT



Transmission Models in IoT



Transmission Models in IoT



Outline

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA