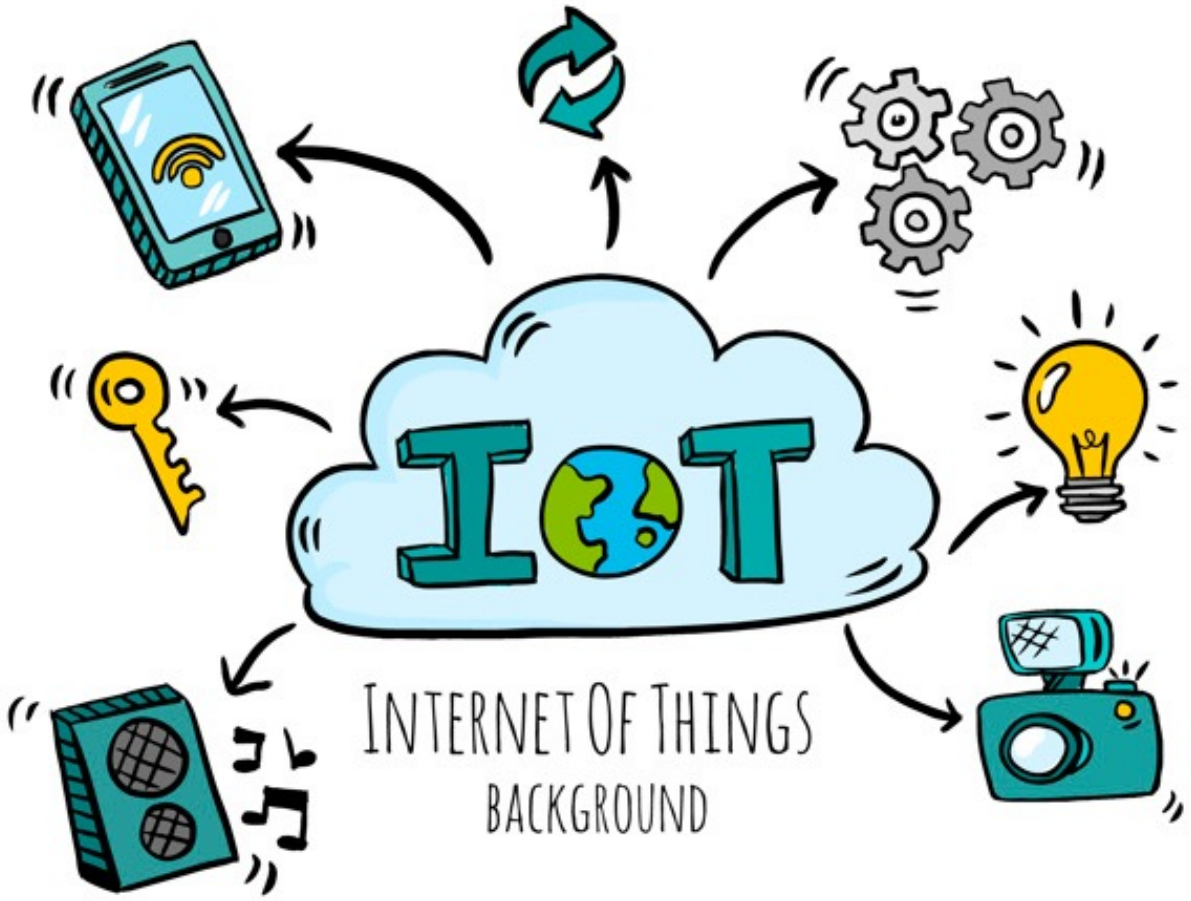
IoT Operating Systems Review



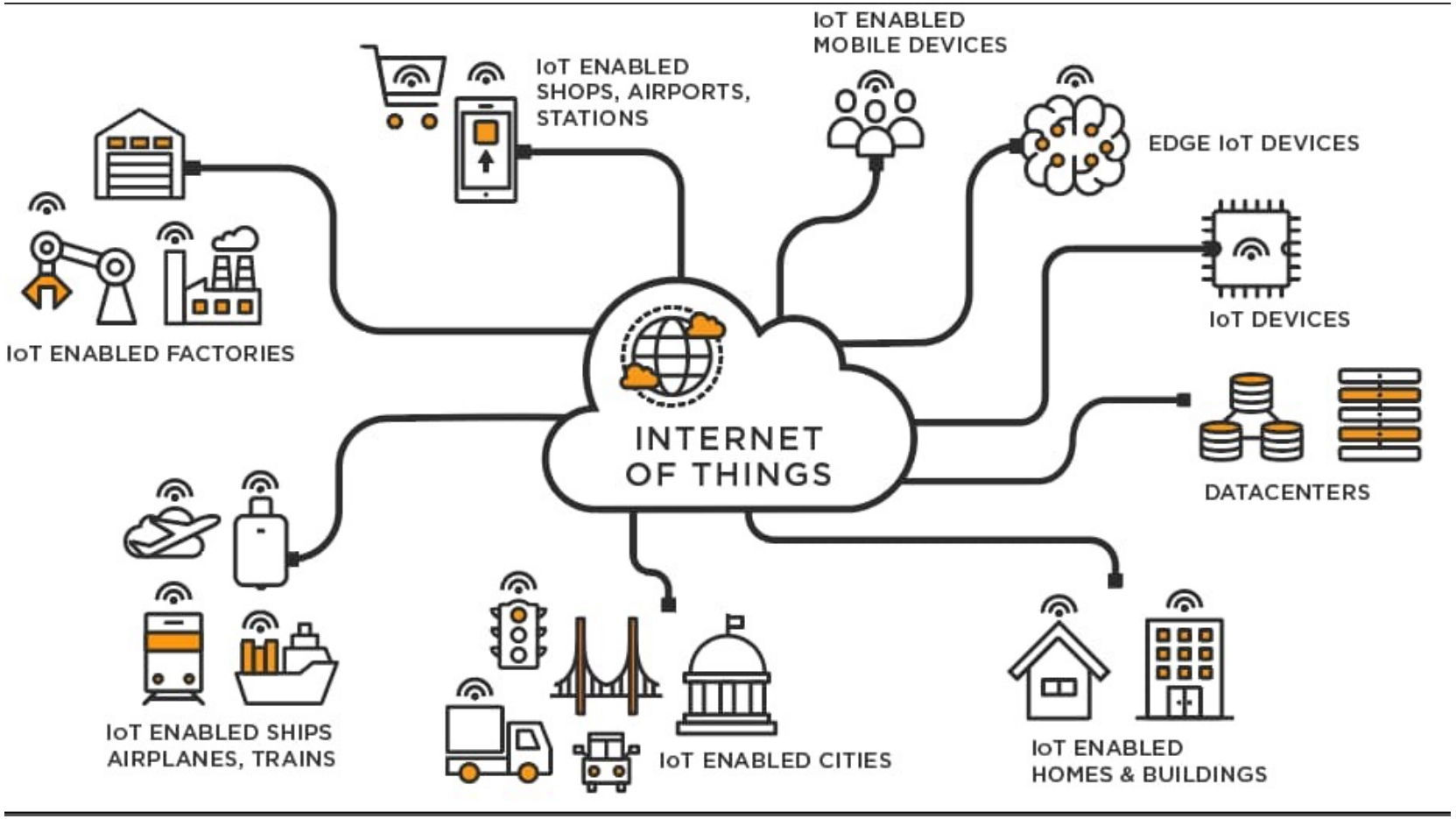
Luigi Capogrosso[1], Fabio Chiarani[1]

[1] name.surname@studenti.univr.it

Table of Contents

Ref: «*Internet of Things (IoT) Operating Systems Support, Networking Technologies, Applications, and Challenges: A Comparative Review*», Farhana Javed *et al.*, IEEE Communications Suyrveys & Tutorial, 2018.

In IoT, heterogeneity is the key

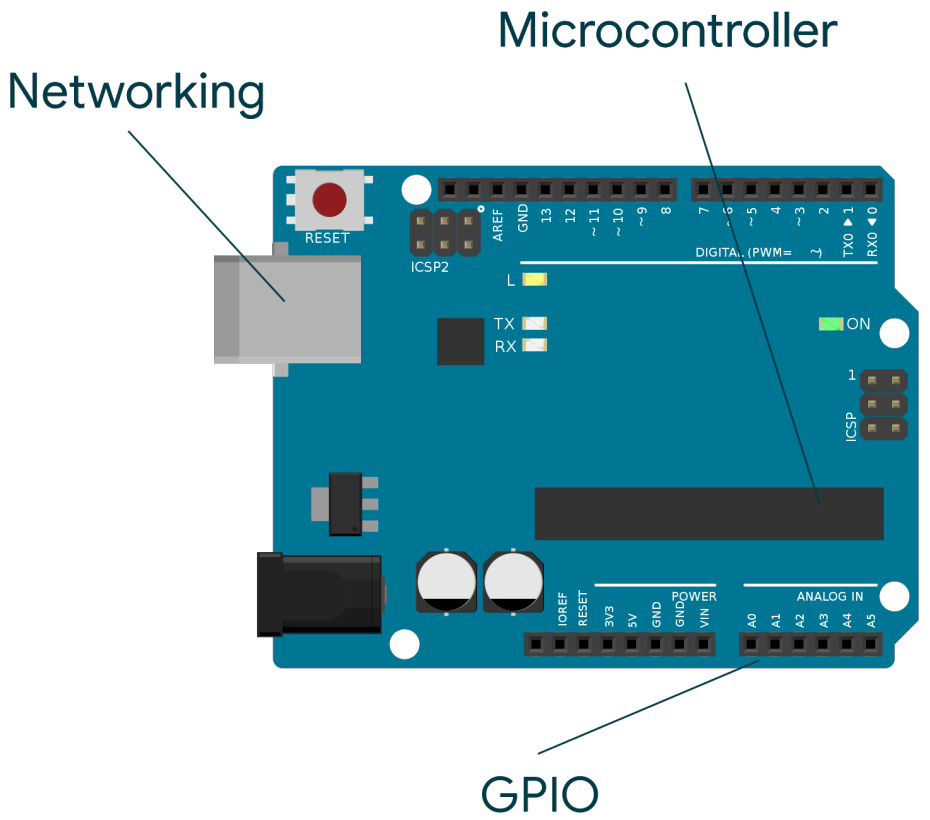


Goals

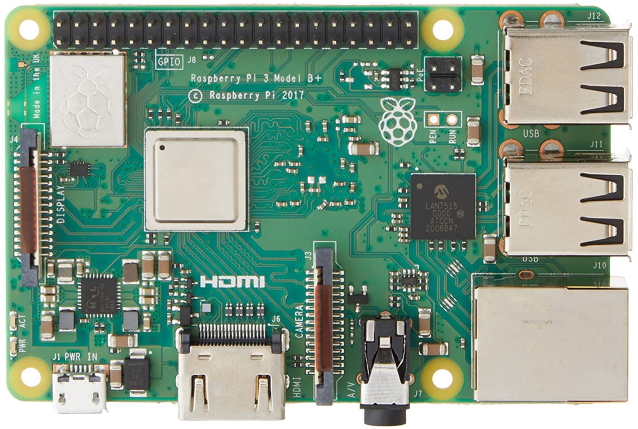
- Multitasking
- Networking
- Real-Time capabilities
- Data Storage
- Device management

How we obtain

- Adeguate kernel
- Scheduling
- Programming model



In the IoTs, the OS of a device determines its structure.
The architecture has an enormous influence on kernel size.



Raspberry Pi

Raspian OS | Kernel size ~ 4 MB

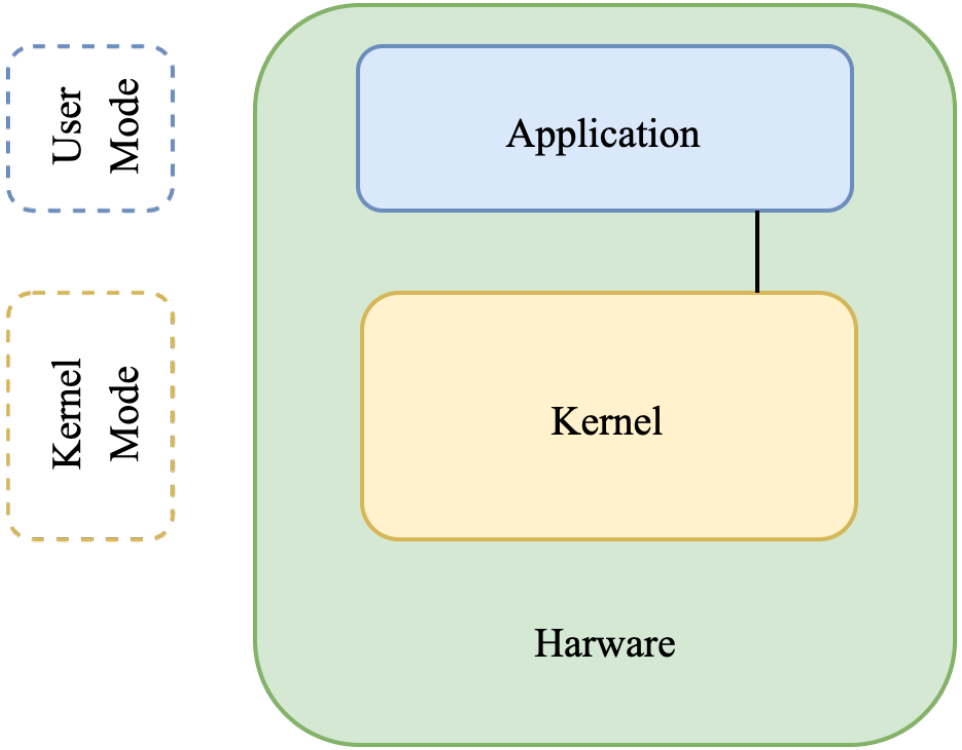
Zoletria Z1

Contiki OS | Kernel size ~ 1 KB

REF: http://www.dunkels.com/adam/dunkels04contiki.pdf

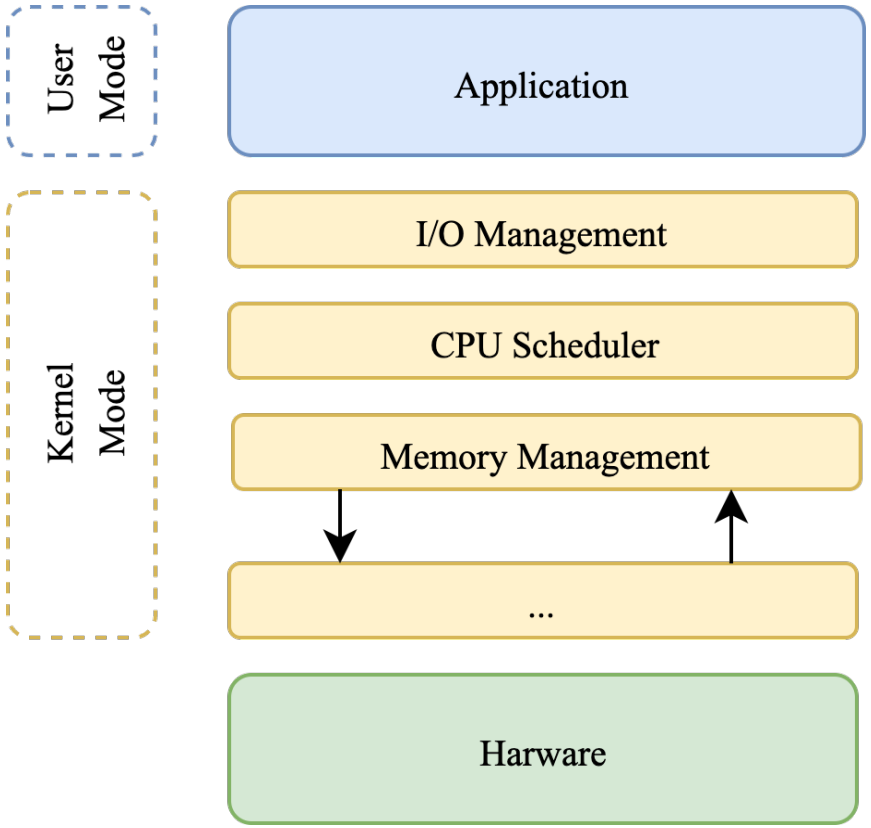
Monolithic kernel

A combination of necessary OS components and applications. **Services are implemented in a single binary file.**



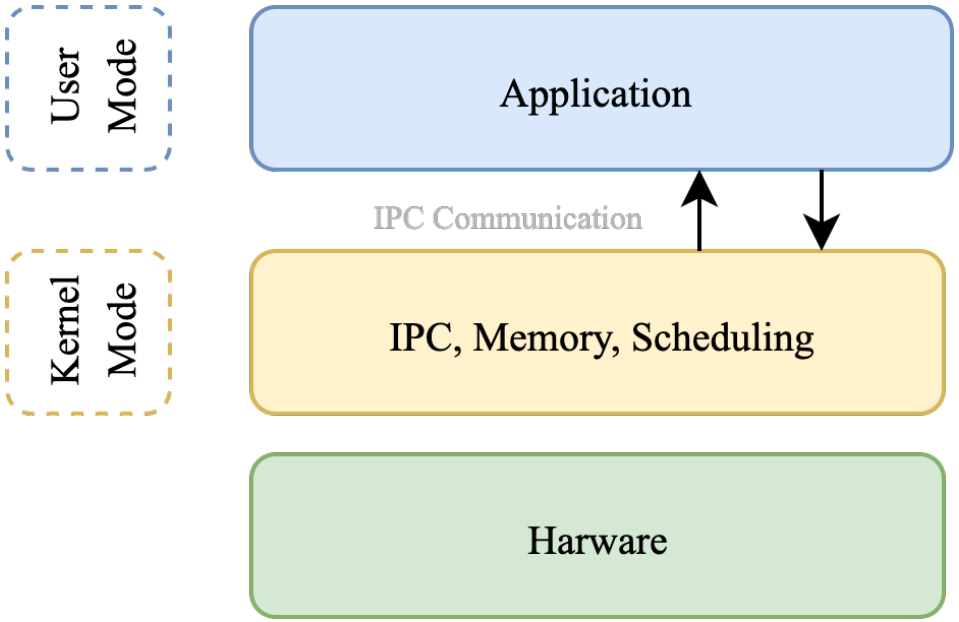
Layered kernel

Each layer represents a functionality. **The performance is not optimal, but it permits achieving high modulability.**



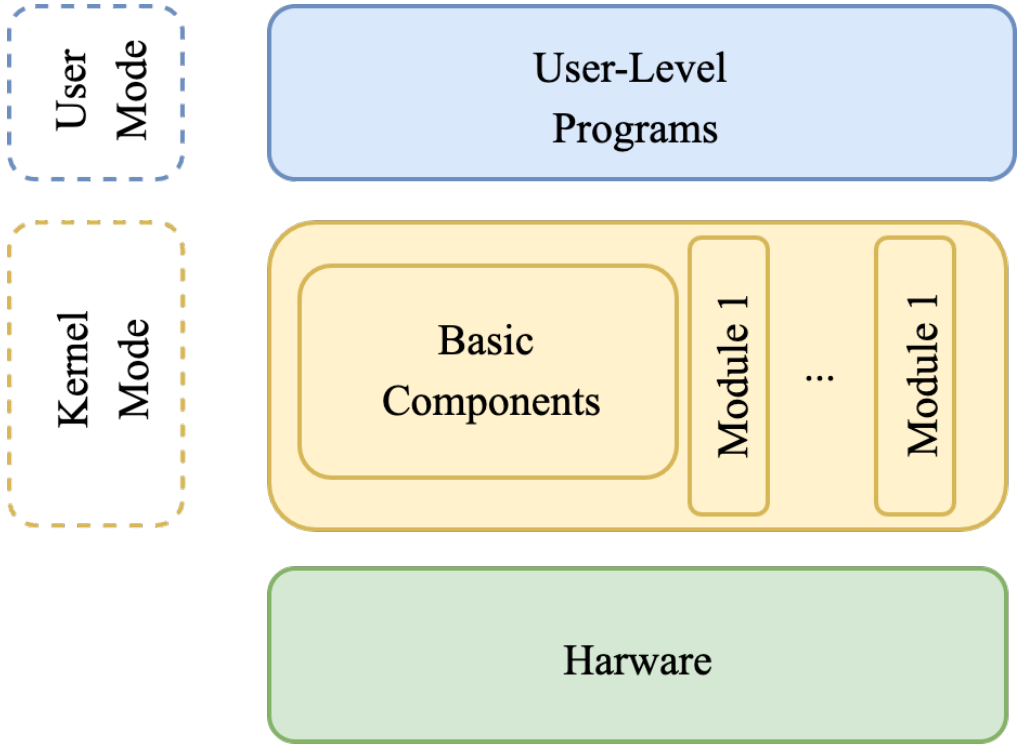
Micro kernel

Just a few essential services are implemented inside the kernel. **Other services are created as a user process.**



Loadable kernel

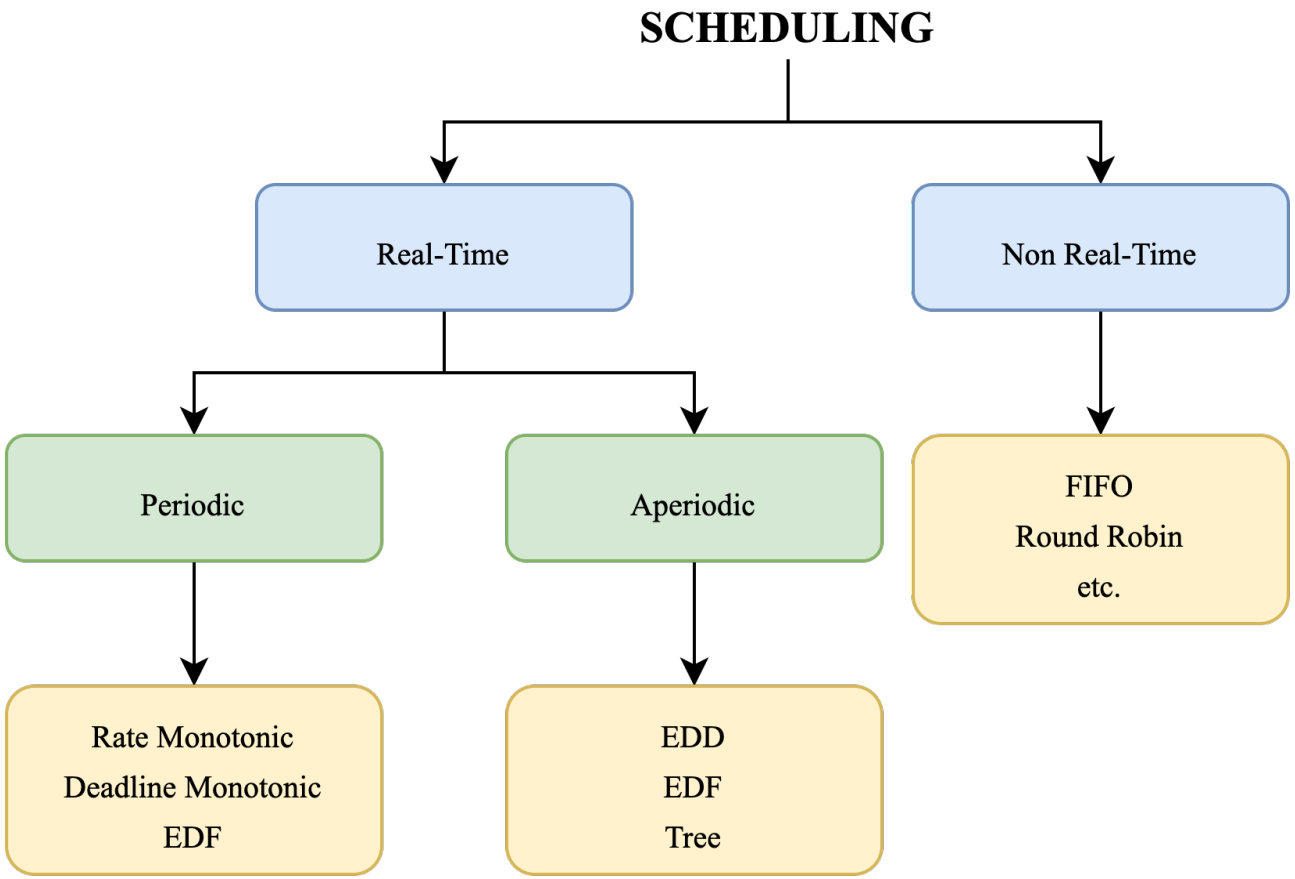
The kernel is composed only of essentials components, which results in a kernel size reduction. **Other services are mounted ad execution time.**

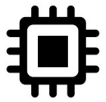


Each architecture has its benefits and lack behind.

Scheduler Types and Real-Time Support

There are multiple suggested algorithms for IoT.





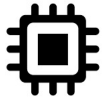
Scheduler Types and Real-Time Support

IoT applications are **highly required** to have a Real-time capabilities.

- Preemptive scheduling is needed
- The algorithm performance has an impact on the power usage

Mostly used:

Priority event-driven algorithms: they are **efficient** and **simple.**



Programming Models & Languages

Programming model can be classified into:

- **Multithreading** programming is the development paradigm that allows several sub-processes to be executed in parallel.
- The programs written using the **event-driven** technique, the flow of the program is determined by the occurrence of external events.

Mostly used:

There is no model that is more used than the other, indeed both are used based on context. Also, the most commonly used programming language in OSs is the **C language.**

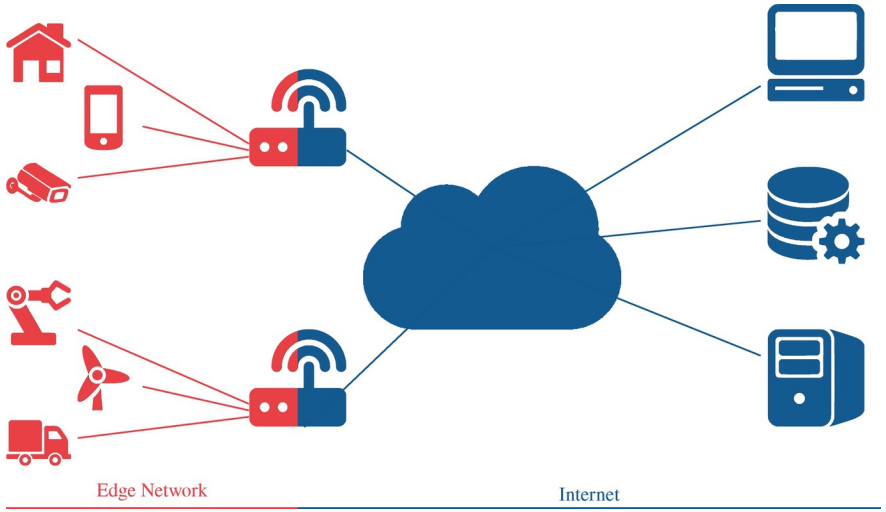
Reprogramming Techniques

Reprogramming is the capability to **change the software functionality of devices at run-time,** because these devices are inaccessible after deployment.

- **Dynamic reprogramming** scheme is highly required
- Each OS has its own methods for reprogramming and uses a different protocol for this purpose

Mostly used:

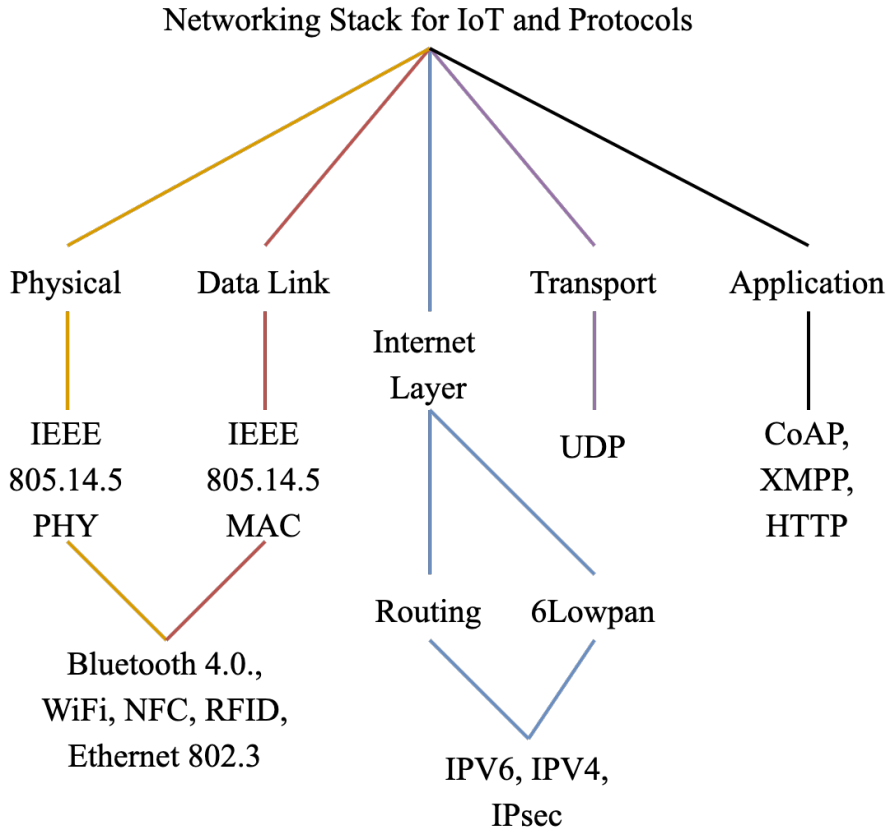
One of the famous protocol is **Trickle**.



Networking Technologies

In IoTs, **networking** is a **major element.**
Indeed, the essential elements of these devices are: **device**, **local network**, and the **Internet**.

- Some recent technologies are **6LoWPN** and **IPv6.**



Mostly used:

One of the most used technologies is still **IPv4.**

Testing and **debugging** is a **major part of software development**.

Therefore, while development **simulation** is an important concern.

Major OSs for IoTs: An Overview

Nano-RK: A Wireless Sensor Networking Real-Time Operating System

Major OSs for IoTs: An Overview

Features	Contiki	TinyOS	RIOT	Nano-RK	LiteOS	MantisOS	SOS	RETOS
Pubblication	2004	2000	2013	2005	2008	2005	2005	2007
Open/Closed	Open	Open	Open	Open	Open	Open	Open	Open
Architecture	Modular	Monolithic	Microkernel	Monolithic	Modular	Layered	Modular	Modular
Scheduling algorithm	RR	FIFO	Priority based	Priority based	RR	Priority based	Priority based	Priority based
Programming model	Multi-threading	Eavent-driven	Multi-threading	Multi-threading	Eavent-driven	Threads	Eavent-driver	Threads
Programming language	C	NesC	C, C++	C	LiteC++	C	C	C
Reprogramming	No	Yes	No	No	Yes	No	Yes	Yes
Testing	Yes	Yes	Yes	No	Yes	Yes	No	Yes
Debugging	Yes	Yes	Yes	No	Yes	Yes	No	Yes
Documentation	Yes	Yes	No	No	Yes	Yes	No	No

Major OSs for IoTs: An Overview

Features	Contiki	TinyOS	RIOT	Nano-RK	LiteOS	MantisOS	SOS	RETOS
Smart Homes	Yes	Yes	Yes	No	Yes	Unknown	Unknown	Unknown
Wearables	No	No	Yes	No	Yes	Unknown	Unknown	Unknown
Smart City	Yes	Yes	Yes	No	Yes	Unknown	Unknown	Unknown
Smart Grid	Yes	Yes	Yes	No	Yes	Unknown	Unknown	Unknown
Industries	Yes	Yes	Yes	No	Yes	Unknown	Unknown	Unknown
Smart Traffic	Yes	Yes	Yes	No	Yes	Unknown	Unknown	Unknown
Medical	Yes	Yes	Yes	Yes	Yes	Unknown	Unknown	Unknown
Smart Retail	No	No	No	No	No	Unknown	Unknown	Unknown
Smart Supply Chain	No	No	No	No	No	Unknown	Unknown	Unknown
Smart Agricultural	No	No	No	No	No	Unknown	Unknown	Unknown

Thank you!

Luigi Capogrosso[1], Fabio Chiarani[1]

[1] name.surname@studenti.univr.it