POLITECNICO
MILANO 1863

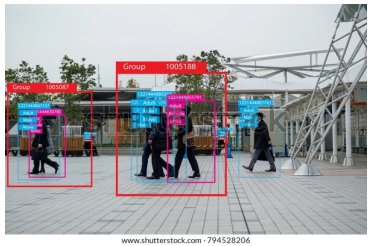
Hardware Architectures for Embedded and Edge AI

Prof Manuel Roveri – manuel.roveri@polimi.it

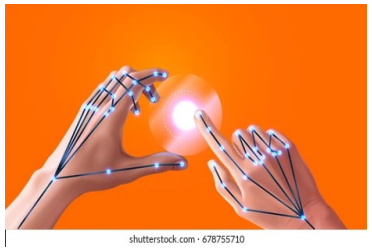
Lecture 2 – Embedded and Edge Hardware

An overview of an embedded and edge AI system

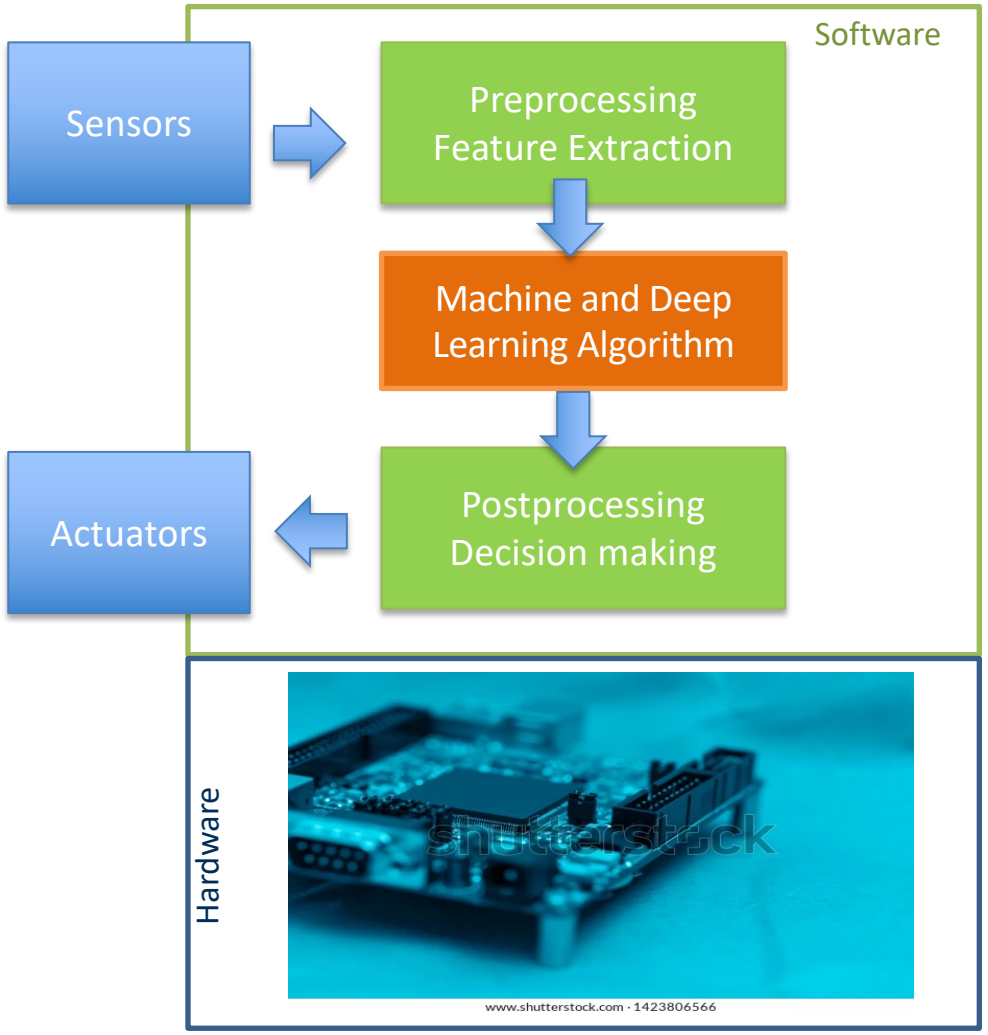
Wake-word detection



Person detection



Gesture recognition

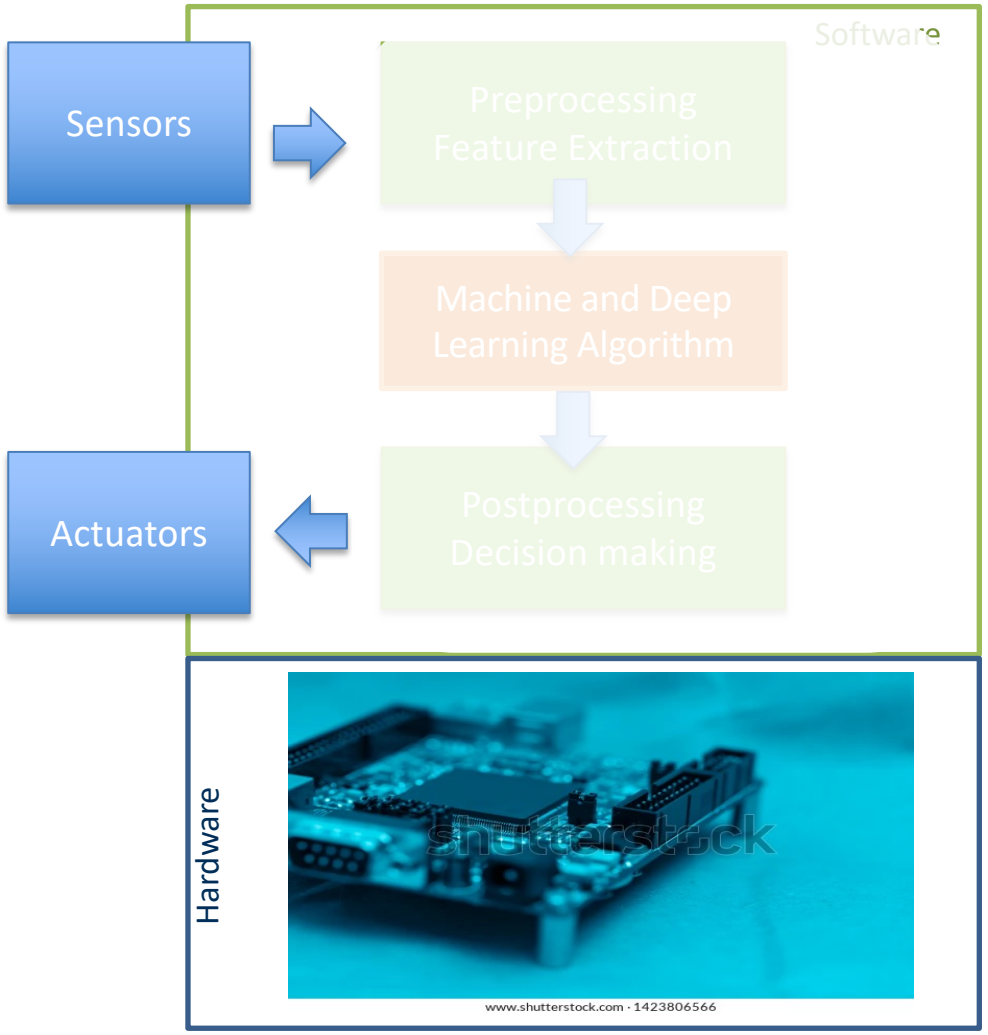


An overview of an embedded and edge AI system

Wake-word detection

Person detection

Gesture recognition



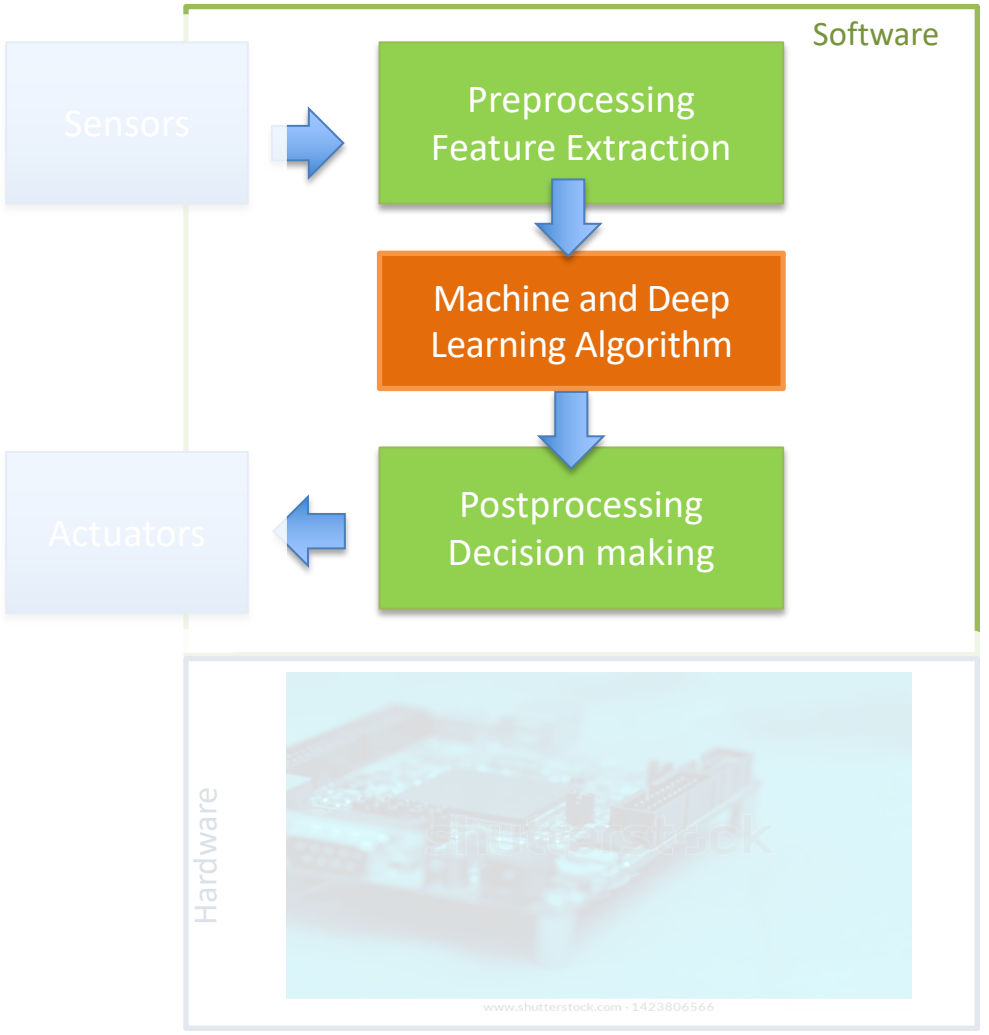
This lecture!

An overview of an embedded and edge AI system

Wake-word detection

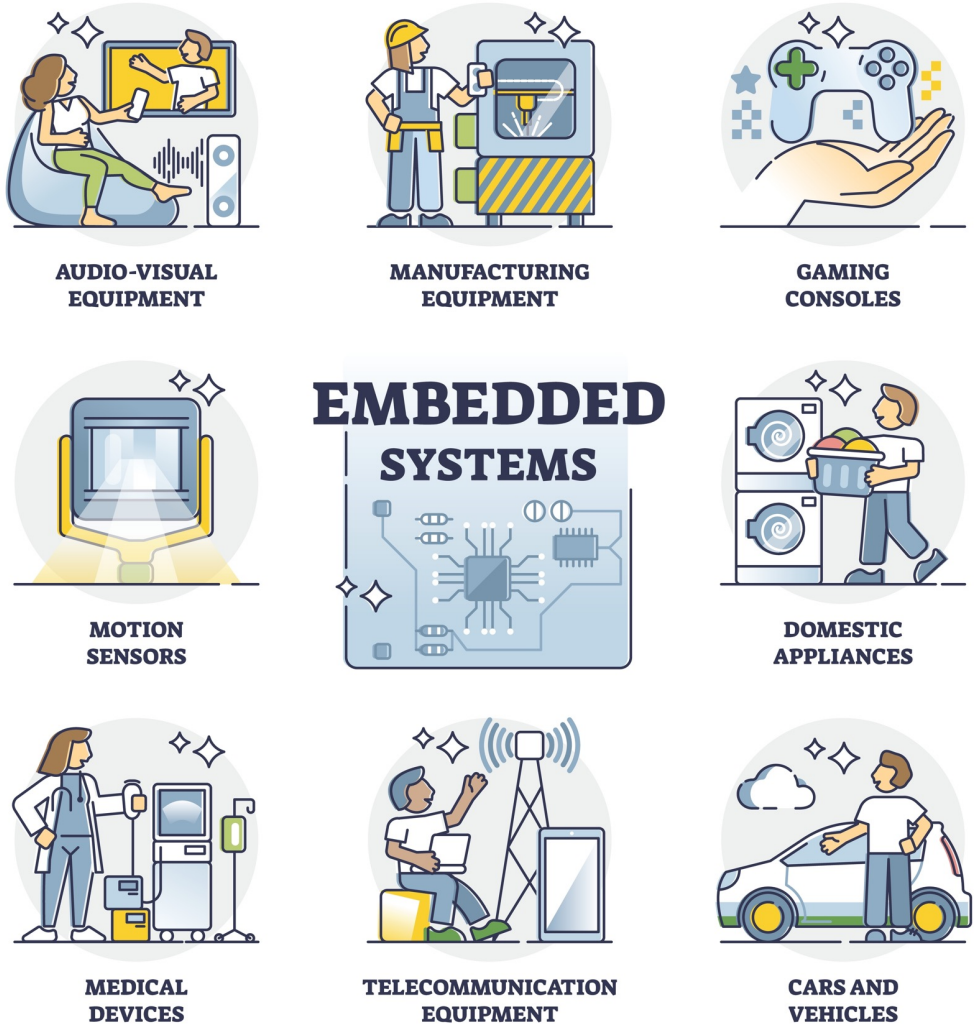
Person detection

Gesture recognition

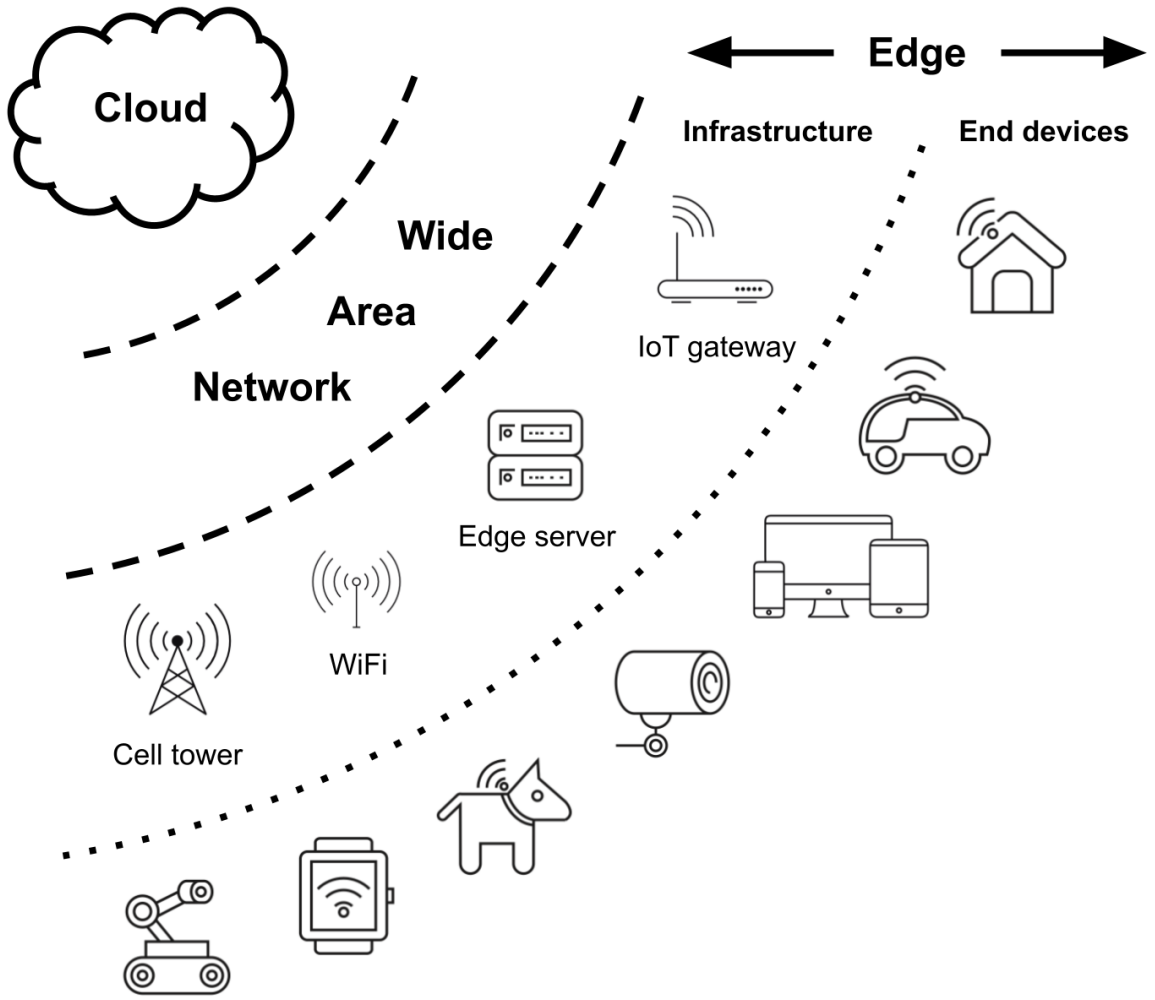


Next lecture!

The embedded perspective



- *Embedded systems* are the computers that control the electronics of all sorts of physical devices
- *Embedded software* is software that runs on embedded systems



Differently from general purpose computers (e.g., a laptop or smartphone) embedded systems are usually meant to perform **one specific, dedicated task.**

Embedded and Edge AI Hardware architecture

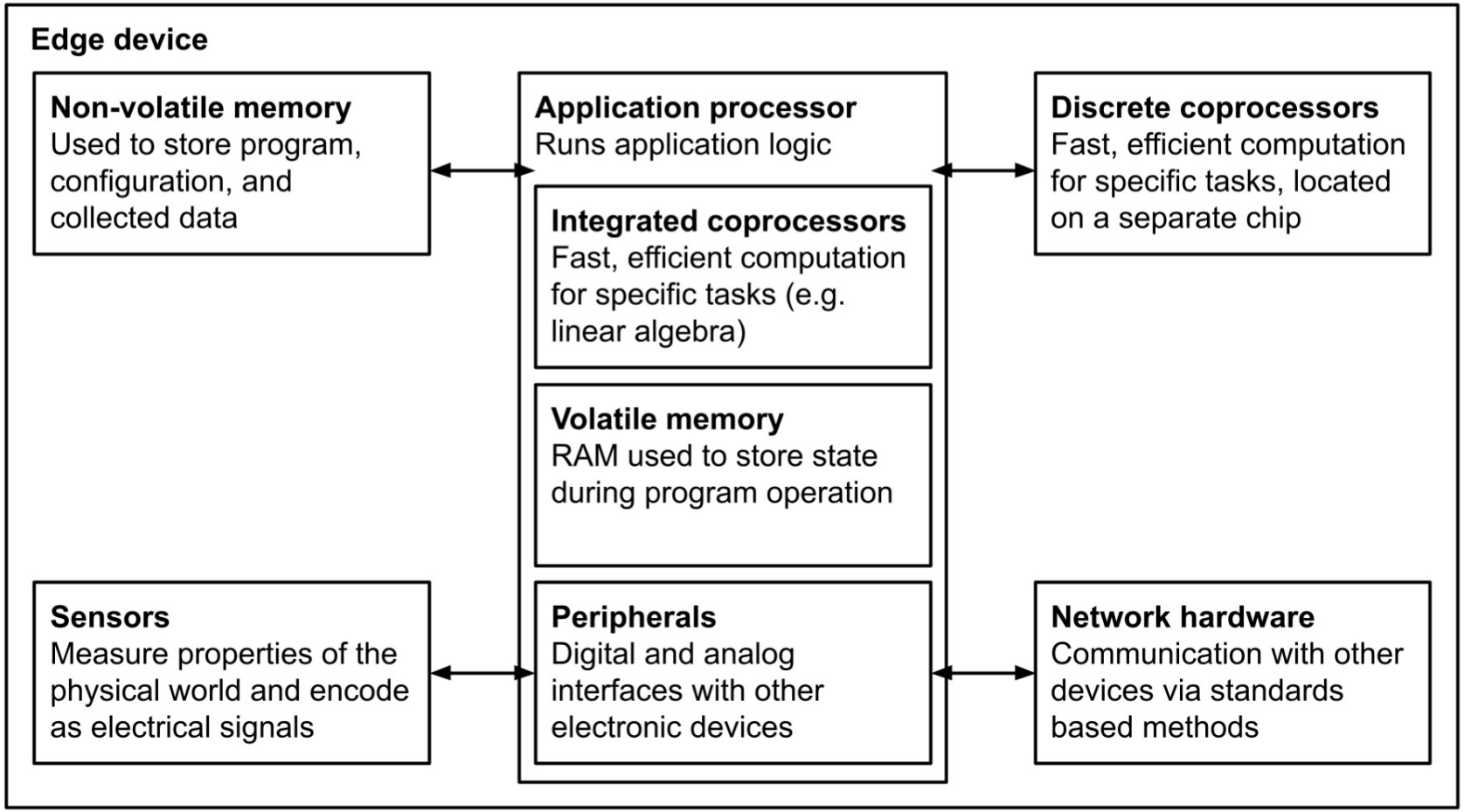
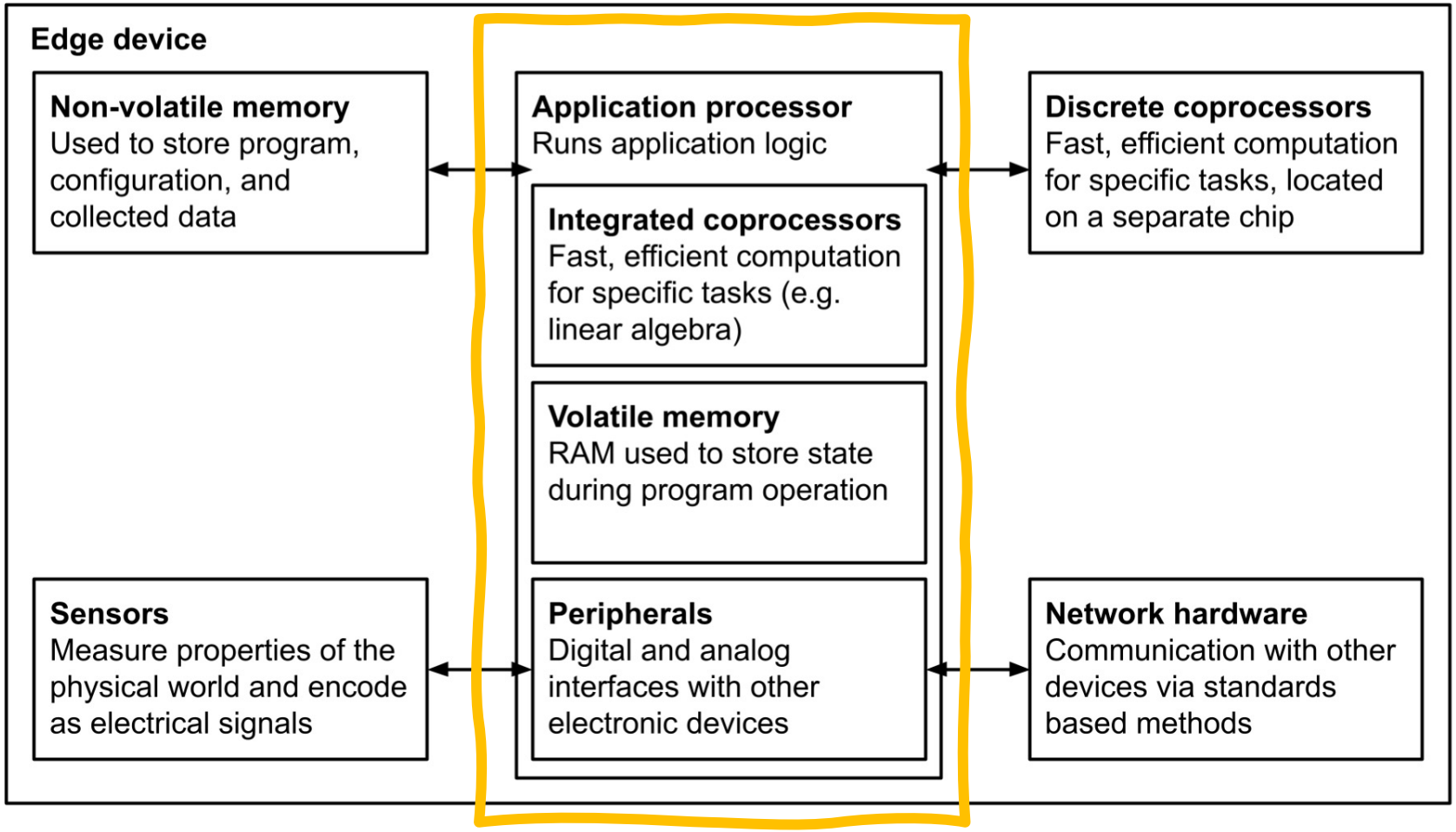


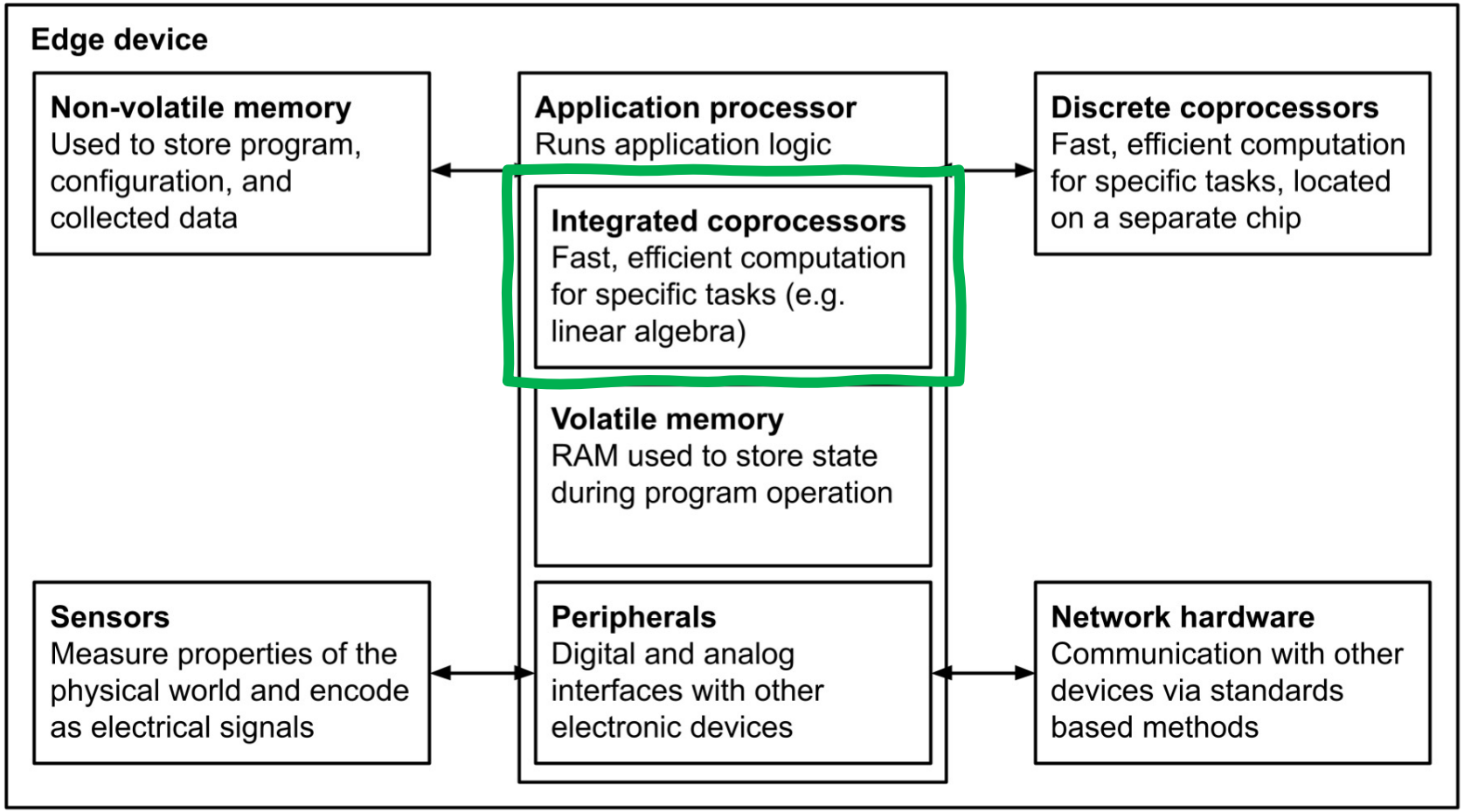
Image taken from [1]

Embedded and Edge AI Hardware architecture



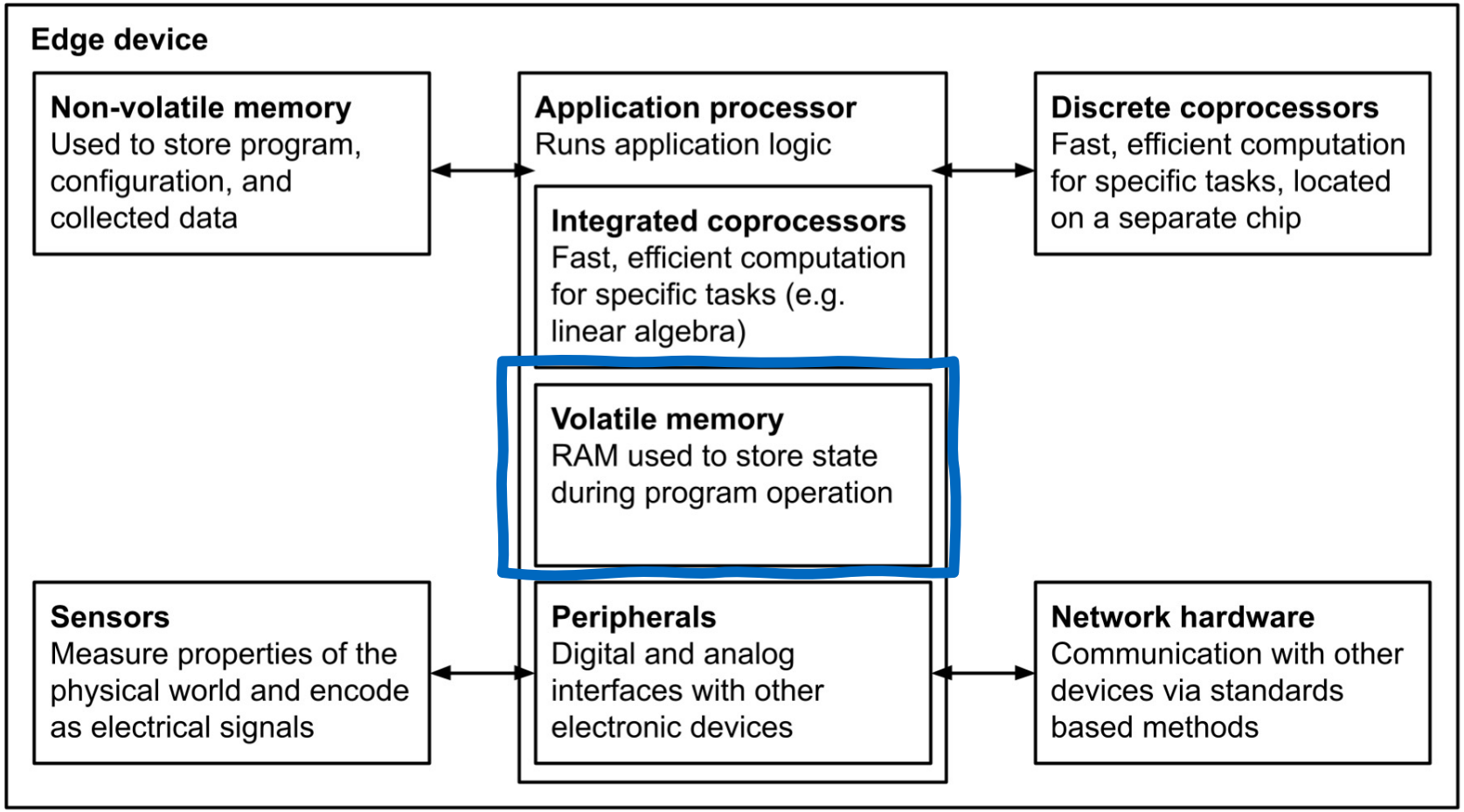
The general purpose processor running the embedded application

Embedded and Edge AI Hardware architecture



Built-in additional hardware meant to provide highly efficient computation on certain operation (e.g, a floating point unit –FPU- to perform FP operations)

Embedded and Edge AI Hardware architecture

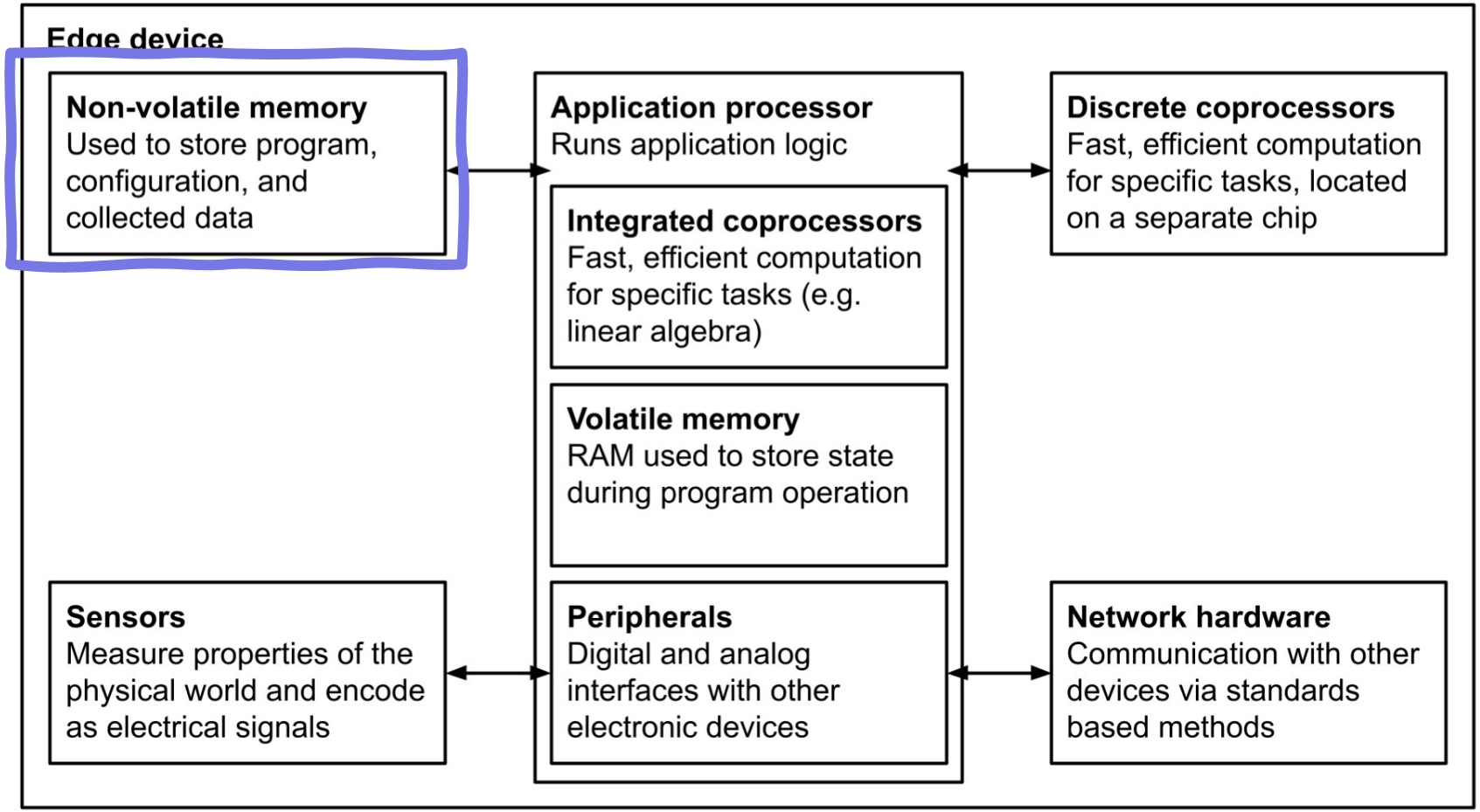


Working memory meant to support program execution
(additional external RAM could be available)

RAM is typically the bottleneck in Embedded and Edge AI

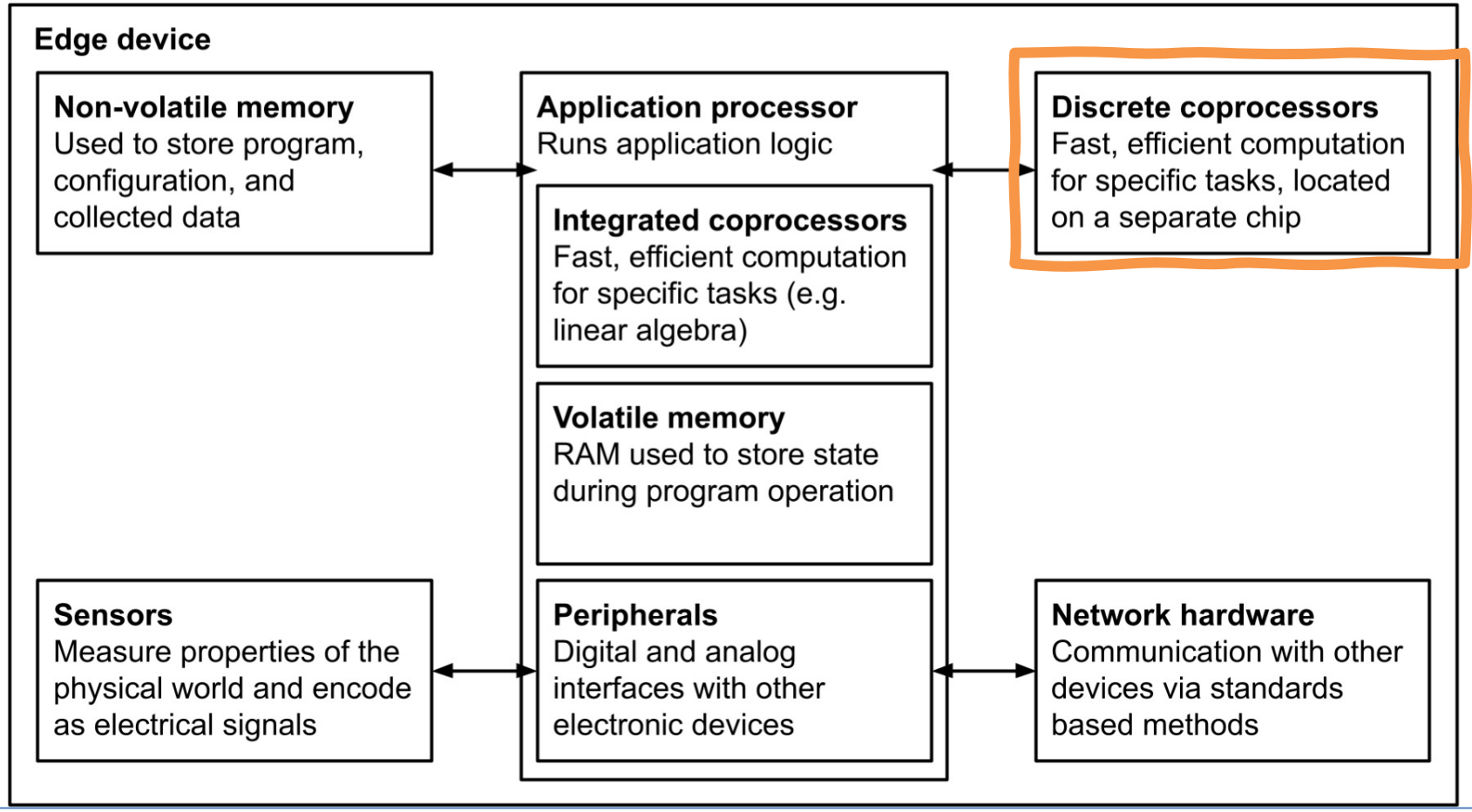
- ✓ Very fast memory, but it is very energy consuming
- ✓ It's volatile (content is lost when power shuts down)
- ✓ Cost
- ✓ Large physical space on the device

Embedded and Edge AI Hardware architecture



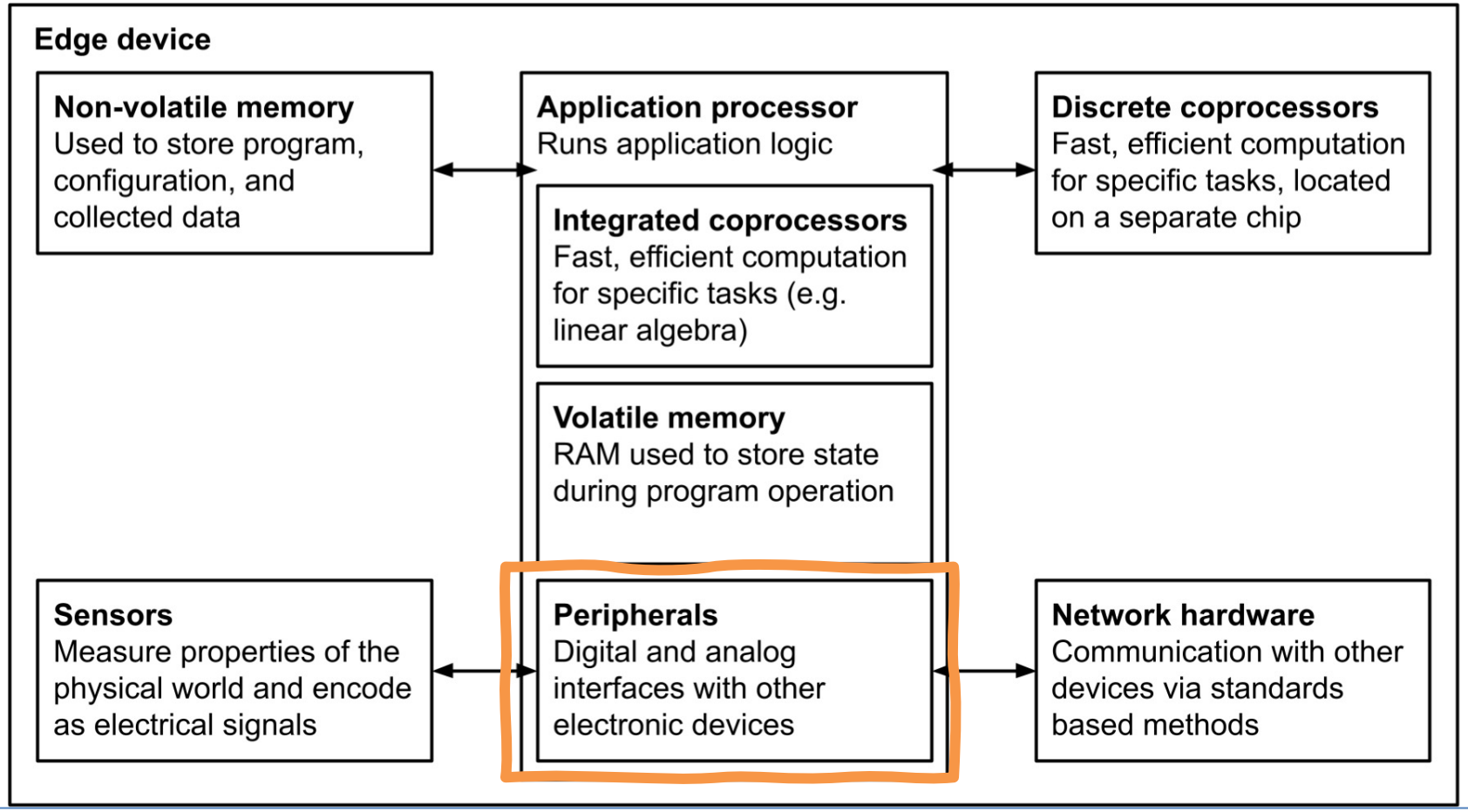
The **non-volatile memory** is the Flash memory. Used to store things that do not change often and that must preserved when the system shuts down (e.g., the software program, the device configuration). Slow to be read, and extremely slow to be written.

Embedded and Edge AI Hardware architecture



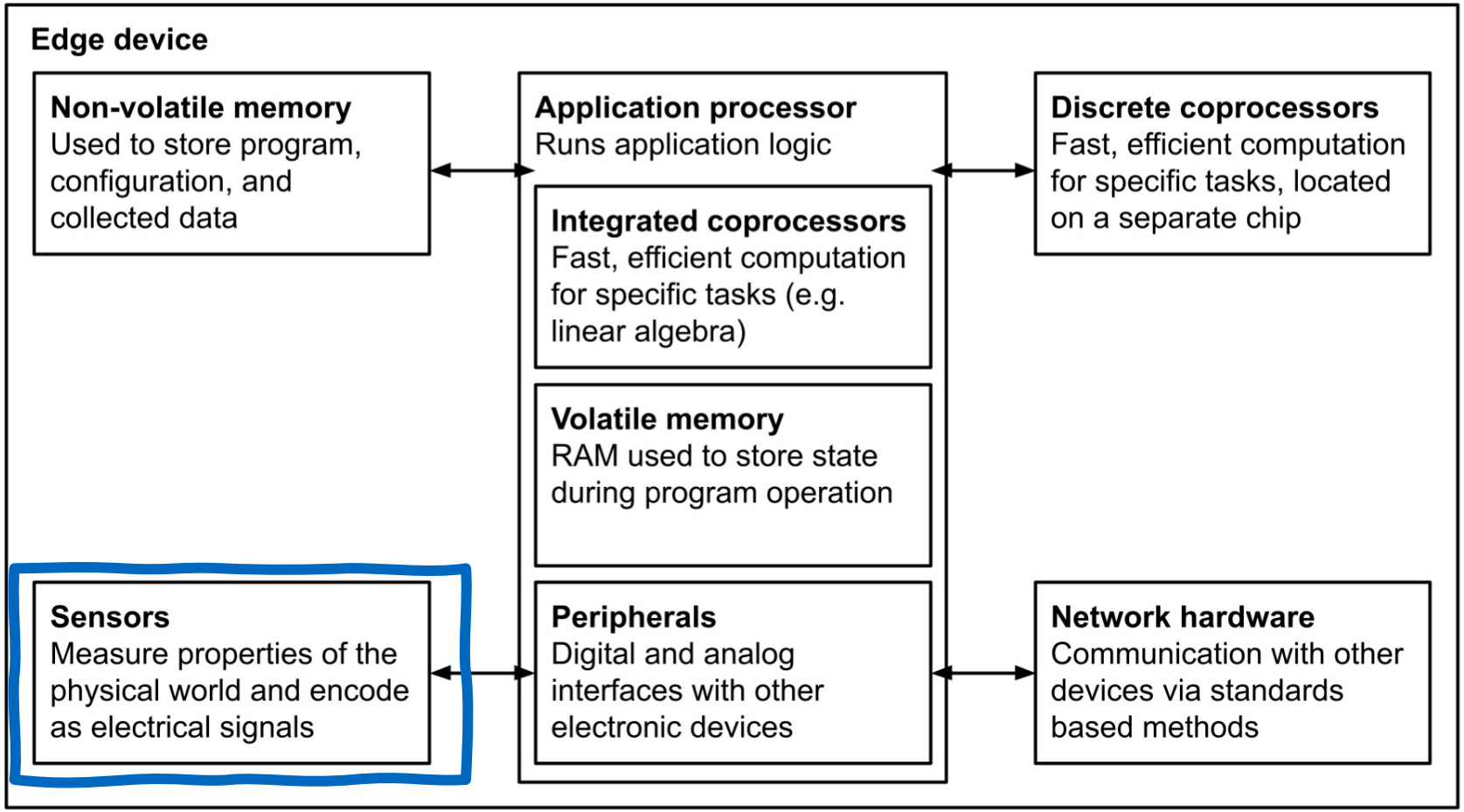
Discrete coprocessors are external co-processor supporting efficent and fast matemathical operations (e.g., a GPU or a TPU)

Embedded and Edge AI Hardware architecture



Provide an interface with the rest of the system (e.g., sensor and network hardware) through technological standardized solutions (e.g, GPIO, I2C, SPI, and UART)

Embedded and Edge AI Hardware architecture: the sensors



Let's see them in detail …

Sensors and signals: the edge AI perspective

Sensors and signals

- Sensors allow to acquire measurements from the environment or from humans
- Sensors generate streams of data
- Other sources of data are available (beyond sensors such as digital device logs, network packets and radio transmissions
- Sensors can provide different data formats: time series, audio, image, video, others…

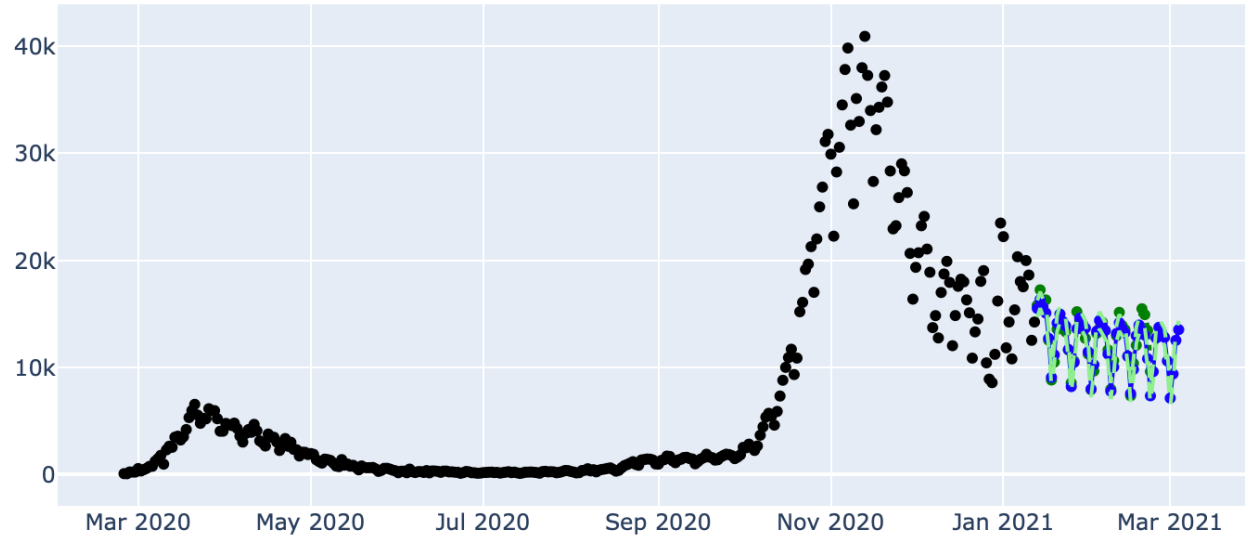
How to store data values?

- **Boolean** (1 bit): a number with 2 possible values
- **8-bit integer**: a non-decimal number with 256 possible values
- **16-bit integer**: a non-decimal number with 65536 possible values
- **32-bit floating point**: can represent a wide range of numbers with up to 7 decimal places, with a maximum of 3.4028235×10^{38}

Quantization techniques (see specific lecture) will allow to reduce the memory demand for each specific value

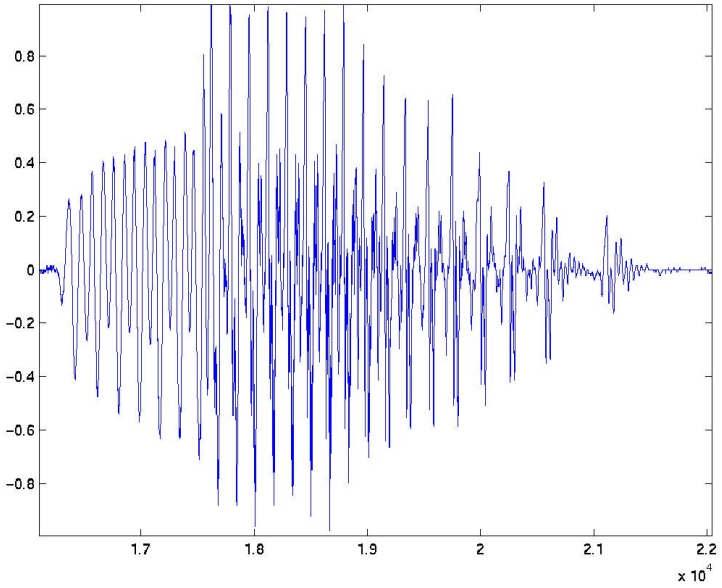
Time series data

- A time series is a sequence of data points indexed in chronological order: $X = (x_1, x_2, \ldots, x_N, \ldots)$
- In other words, a time series is a series of data points taken at successive equally spaced points in time.
- Sampling period (secs, min, hour, days, etc..) and number of bits (n) for the representation
- Memory requirements: n bit per sample (e.g., 4byte FP per sample)



Audio data

- A special case of time series data, audio signals represent the oscillation of sound waves as they travel through the air.
- Sampling (Hz) and quantization (number of bit), length of the signal (in s) and number of channels (e.g., mono or stereo) are key aspects



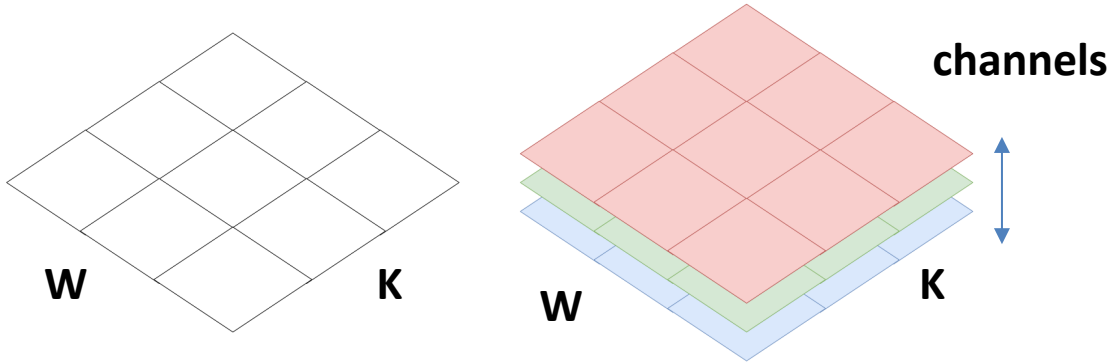
Memory requirements:
length x sampling x bit x channel

For example a 10s audio sampled at 100Hz quantized over 32bit and acquired in stereo manner requires:
10s x 100Hz x 4Byte x 2= 8Kb

320KB RAM Memory

Image data

- Images are data that represent the measurements taken by a sensor that captures an entire scene
- Images have two or more dimensions. In their typical form, they can be thought of as a grid of "pixels", where the value of each pixel represents some property of the scene at the corresponding point in space (stored in N bit)
- Memory requirements: W x K x N x channels



For example a 128 x 128 RGB image with 1byte per pixel requires:

128 x 128 x 1Byte x 3= 49Kb

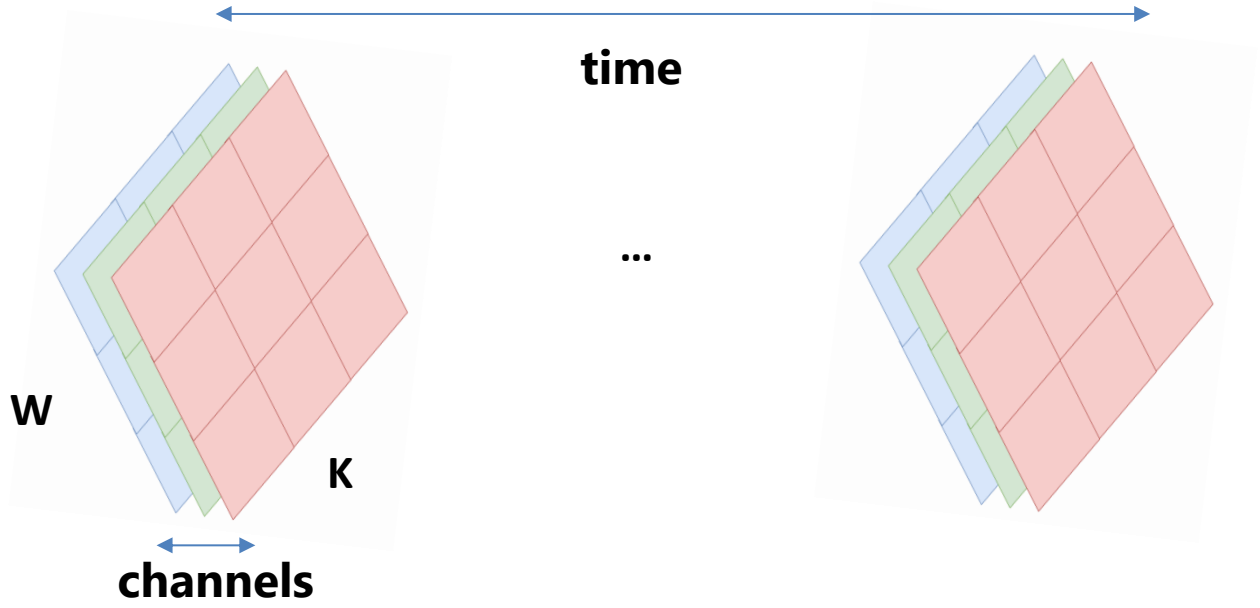
320KB RAM Memory

Video data

A sequence of (fixed) images reproduced with a suitably high frame rate

Similarly to image we have W, K, N and channels. In addition we have a frame rate (in frame/s) and the length of the video (in s)

- Memory requirements: W x K x N x channels x frame rate x length



For example a 10s video recorded at 30 fps at 128 x 128 RGB image with 1byte per pixel requires:

30 x 10 x 128 x 128 x 1Byte x 3= 14.7Mb (uncompressed)

320KB RAM Memory

14.7 MB …

Exercise slide

- Compute the memory demand for a 5s audio sampled at 44.1Khz over one channel (e.g., wake-word detection) with 2Byte per sample
- Compute the memory demand for a 28 x 28 grayscale image (1Byte), see MNIST
- Compute the memory demand for a 224 x 224 RGB image (1Byte per channel), see IMAGENET

For all the exercises consider un-compressed data!

Exercise slide (solution)

- Compute the memory demand for a 5s audio sampled at 44.1Khz over one channel (e.g., wake-word detection) with 2Byte per sample

  5s x 44.100 sample/s x 2 Byte/sample = 441 KB

- Compute the memory demand for a 28 x 28 grayscale image (1Byte), see MNIST

  28 x 28 x 1 Byte/pixel = 784 Byte

- Compute the memory demand for a 224 x 224 RGB image (1Byte per channel), see IMAGENET

  224 x 224 x 3 channel x 1 Byte/pixel = 150 KB

Types of sensors and signals

- There are thousands of different types of sensors on the market
- Families of sensors for embedded and edge AI:
  1. Acoustic and vibration
  2. Visual and scene
  3. Motion and position
  4. Force and tactile
  5. Optical, electromagnetic, and radiation
  6. Environmental and chemical

1) Acoustic and vibration

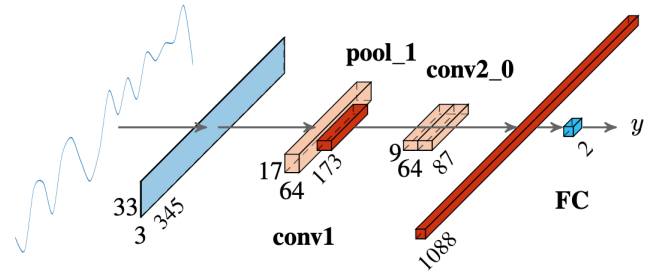
- "Hear" vibrations is a **crucial ability** in the field of embedded and edge AI
- Such an ability allows to **detect the effects** of movement, vibration, and human and animal communication at a distance.
- Acoustic sensors measure the effect of **vibrations travelling through a medium**:
  - air (microphones)
  - water (hydrophones) or
  - ground (geophones and seismometers)
- Information is distributed across **frequencies** (the sampling frequency is crucial for a given application scenario)

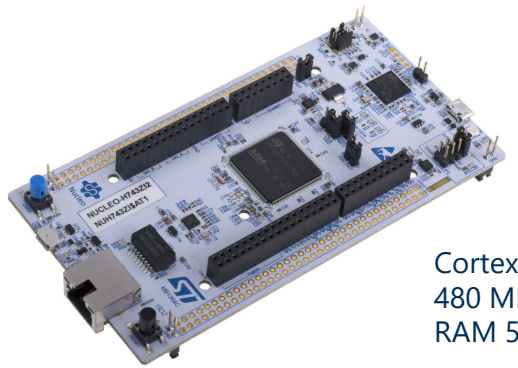
Acoustic sensors generally produce an audio data describing the variation of pressure in the medium

Birdsong detection and recognition (in the wild)

A TinyML-based IoT units endowed with CNNs to detect and recognize bird vocalizations

Sampling frequency 2kHz
Memory footprint 133Kb
Accuracy approx. 76%
Inference time 3.5s
Lifetime 12 days (3200 mAh battery)





Cortex M7
480 MHz
RAM 512Kb

S. Disabato, G. Canonaco, P. Flikkema, M. Roveri, C. Alippi, "Birdsong Detection at the Edge with Deep Learning", in Proc. 7th IEEE International Conference on Smart Computing (SMARTCOMP 2021)

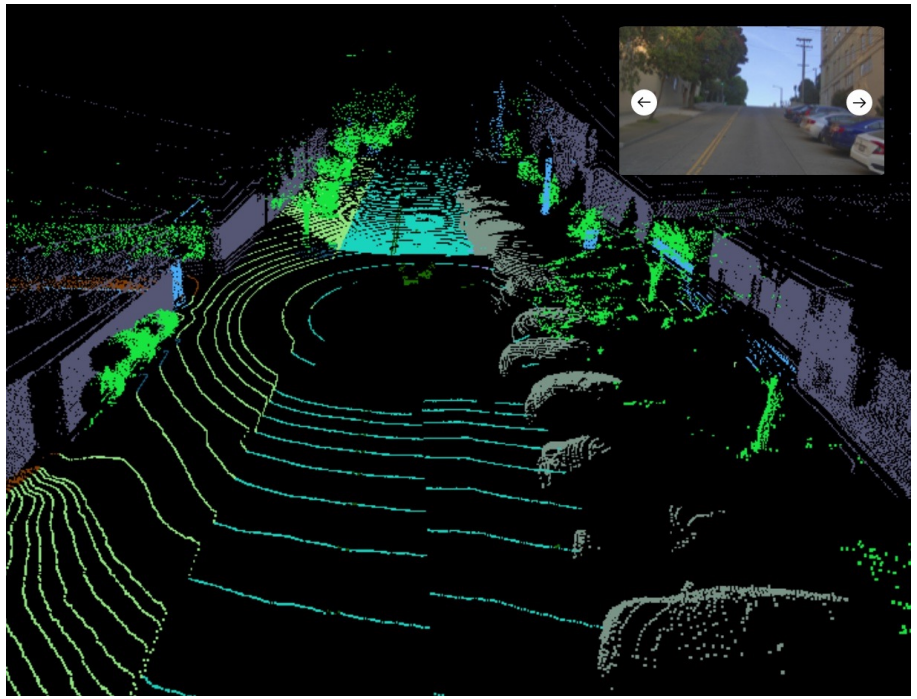
2) Visual and scene

Acquiring information without contact (passively):

- tiny, low-power cameras
- super high quality multi-megapixel sensors
- Image sensors capture light using a grid of sensor elements
- Features of cameras:
  - ✓ Color channels—grayscale or color (red, green, and blue, or RGB).
  - ✓ Spectral response—e.g., infrared radiation for thermal cameras to "see" heat.
  - ✓ Pixel size—larger sensors can capture more light per pixel, increasing their sensitivity.
  - ✓ Sensor resolution—the more elements on a sensor, the more details
  - ✓ Frame rate—how frequently a sensor can capture an image, typically in frames per second.

The output of cameras is an image (2D/3D) or a video

2) Visual and scene (part 2)



LIDAR

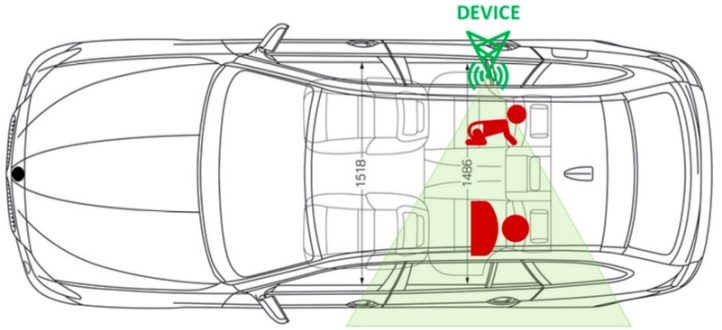
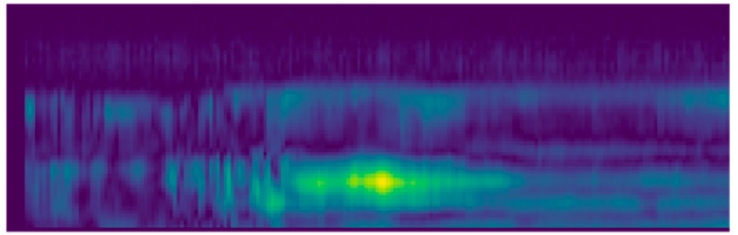
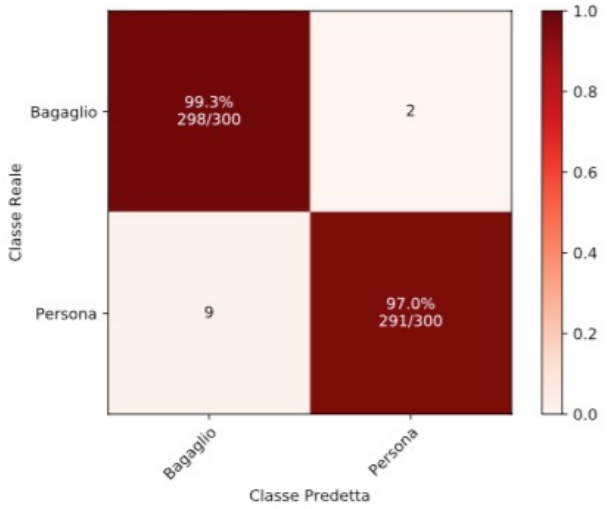


Fig. 5. The acquisition campaign for this experimental analysis



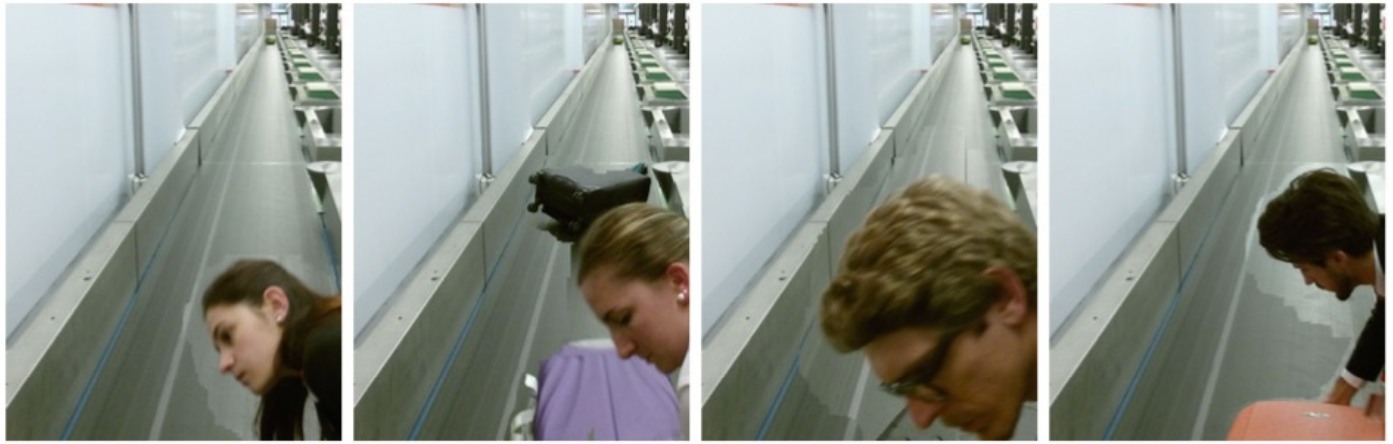
RADAR

People detection on the conveyor belt in Malpensa Airport (T2)



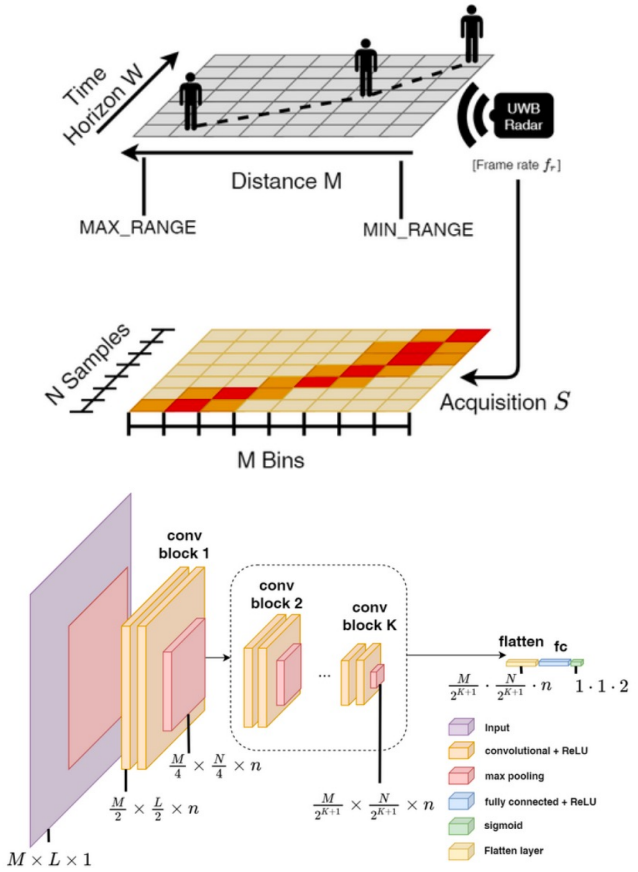
(a) Matrice di confusione

(b) Bagagli classificati come Persone



UWB-based smart sensors for retirement homes

Privacy-preserving behaviour recognition in patients



M. Pavan, A. Caltabiano and M. Roveri, "TinyML for UWB-radar based presence detection," accepted for publication in IEEE World Congress of Computational Intelligence 2022 (IEEE WCCI 2022)

3) Motion and position

Many different types of motion and position sensors:

- Tilt sensor—a mechanical switch measuring the orientation
- Accelerometer—measures the acceleration (the change in velocity over time) of an object across one or more axes (smart watches or predictive maintenance).
- Gyroscopes—measure the rate of rotation of an object.
- Time of flight—electromagnetic emission (light or radio) to measure the distance from a sensor to whatever object is directly in its line of sight.
- Real time locating systems—multiple transceivers in fixed locations around a building to track the position of individual objects
- Global Positioning System (GPS)—satellites to determine the location of a device

Measurements are typically represented as a time series

Building and critical infrastructure monitoring (Italy)

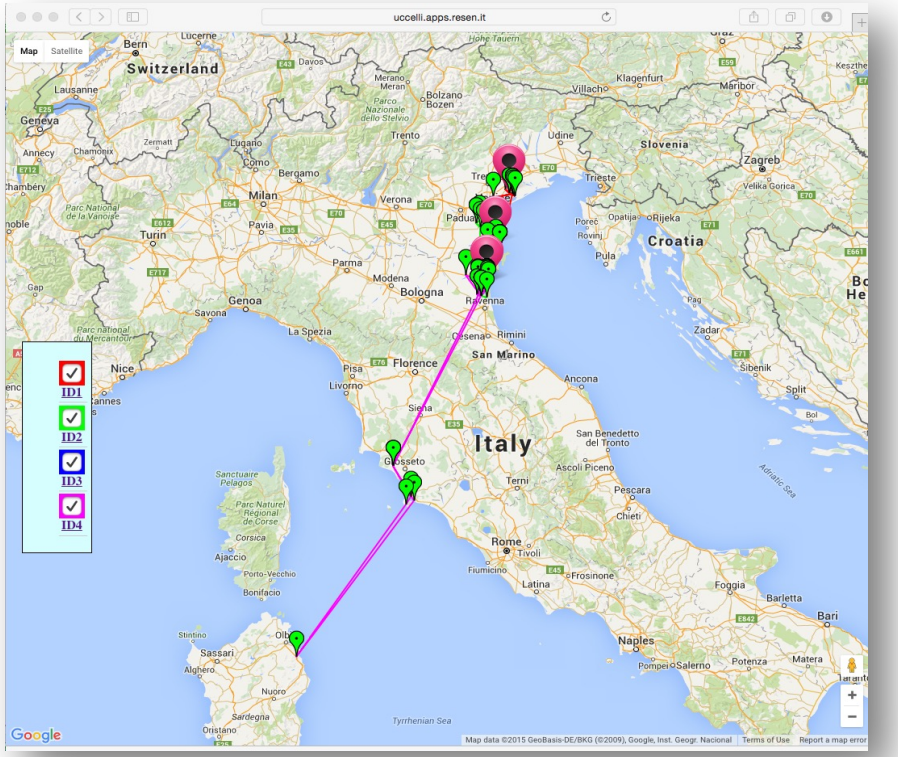
A network of TinyML-based IoT units for intelligent structural monitoring





Internet-of-Birds tracking system (Italy)

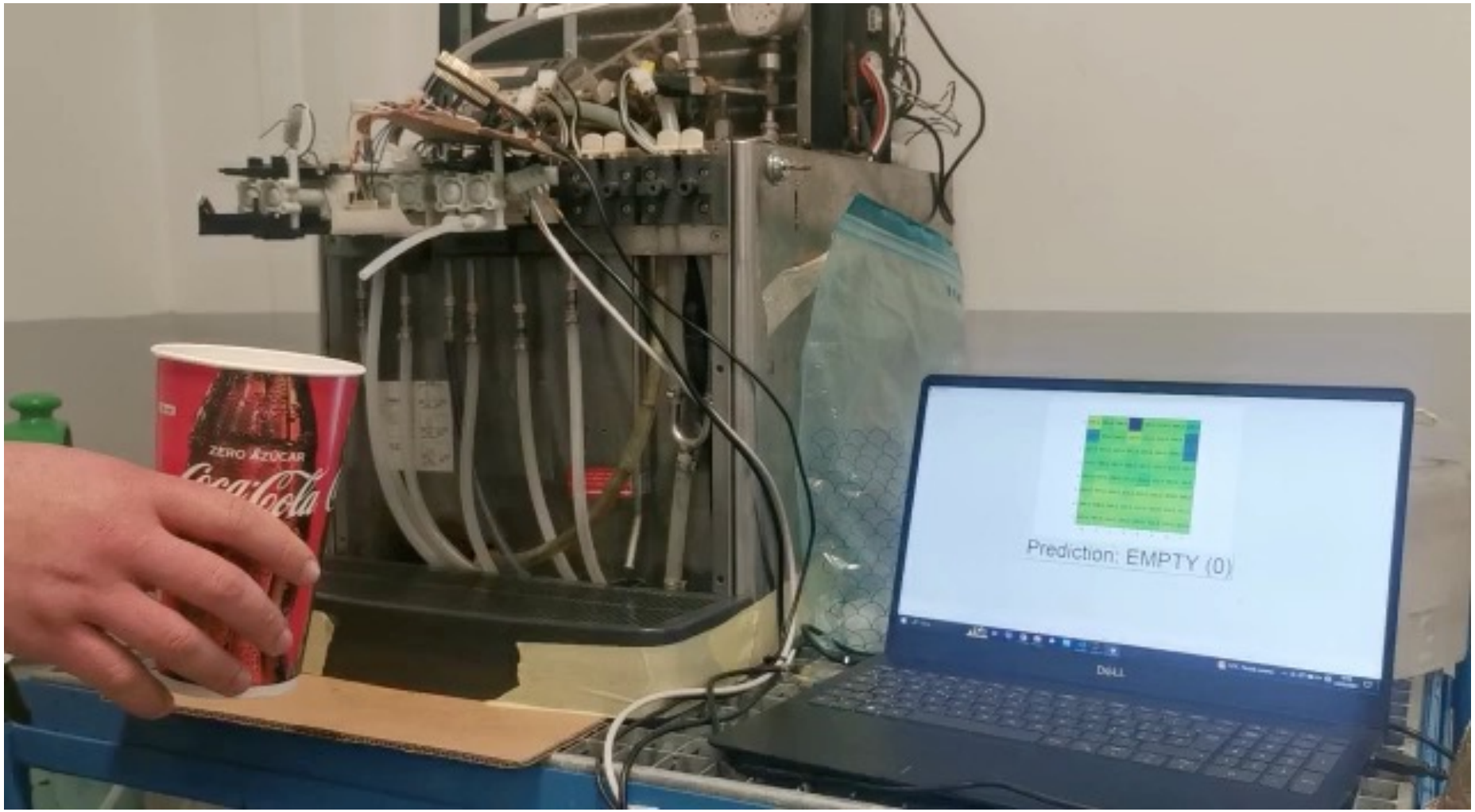
Tracking greater flamingos through ultra-lightweight smart IoT devices



1Y of working activities, 4 greater flamingos, sampling rate 20h, weight of the device < 20g

C. Alippi, R. Ambrosini, D. Cogliati, V. Longoni, M. Roveri, "A lightweight and energy-efficient Internet-of-Birds Tracking System", in Proc. IEEE International Conference on Pervasive Computing and Communications (PerCom 2017), Hawaii, USA, 13-17 March 2017.

Time-of-Flight Sensors for cup recognition



4) Force and tactile

Helpful in facilitating user interaction, understanding the flow of liquids and gases, or measuring the mechanical strain on an object:

- Buttons and switches— provide a binary signal
- Capacitative touch sensors—measure the amount that a surface is being touched by a conductive object, like a human finger.
- Strain gauges—measure how much an object is being deformed
- Load cells—these measure the amount of physical load that is applied
- Flow sensors—measure the rate of flow in liquids and gases
- Pressure sensors—used to measure pressure of a gas or liquid, either environmental or inside a system (e.g., inside a car tire).

Measurements are typically represented as a time series

Rock-collapse and landslide forecasting (Italy and Switzerland)

A TinyML-based monitoring system to identify critical situations

C. AIippi, R. Camplani, C. Galperti, A. Marullo, M. Roveri, “A high frequency sampling monitoring system for environmental and structural applications”, ACM Transactions on Sensor Networks, Vol. 9, No. 4, Art. 41, 32 pages, July 2013

5) Optical, electromagnetic, and radiation

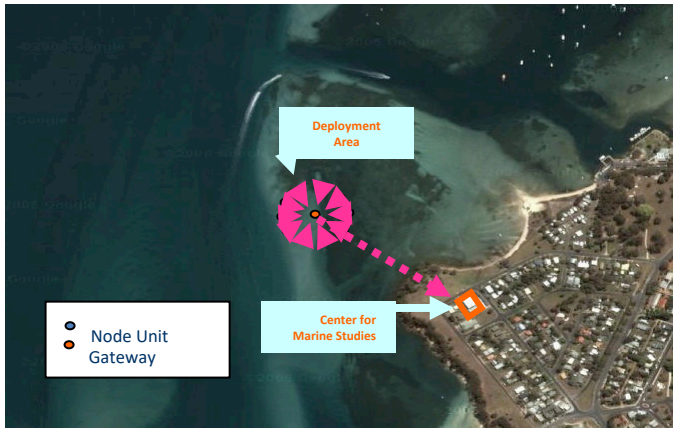
Sensors measuring electromagnetic radiation, magnetic fields as well as current and voltage:

- Photosensors—detect light at various wavelengths, both visible and invisible to the human eye.
- Color sensor—photosensors to measure the precise color of a surface, helpful for recognizing different types of objects.
- Spectroscopy sensors—measure the way that various wavelengths of light are absorbed and reflected by materials, giving an edge AI system insight into their composition.
- Magnetometer—measure the strength and direction of magnetic fields (a digital compass)
- Inductive proximity sensors—electromagnetic field to detect nearby metal (detect vehicles for traffic monitoring)
- Electro-magnetic field (EMF) meters—measure the strength of electromagnetic fields (e.g., emitted by industrial equipment)
- Current and voltage sensor

Measurements are typically represented as a time series

Acquatic monitoring system (Great Coral Reef, Australia)

A TinyML-based system to monitor the water pollution and forecast hurricanes



C. Alippi, R. Camplani, C. Galperti, M. Roveri, "A robust, adaptive, solar powered WSN framework for aquatic environmental monitoring", IEEE Sensors Journal, Vol. 11, No. 1, pp. 45-55, Jan. 2011.

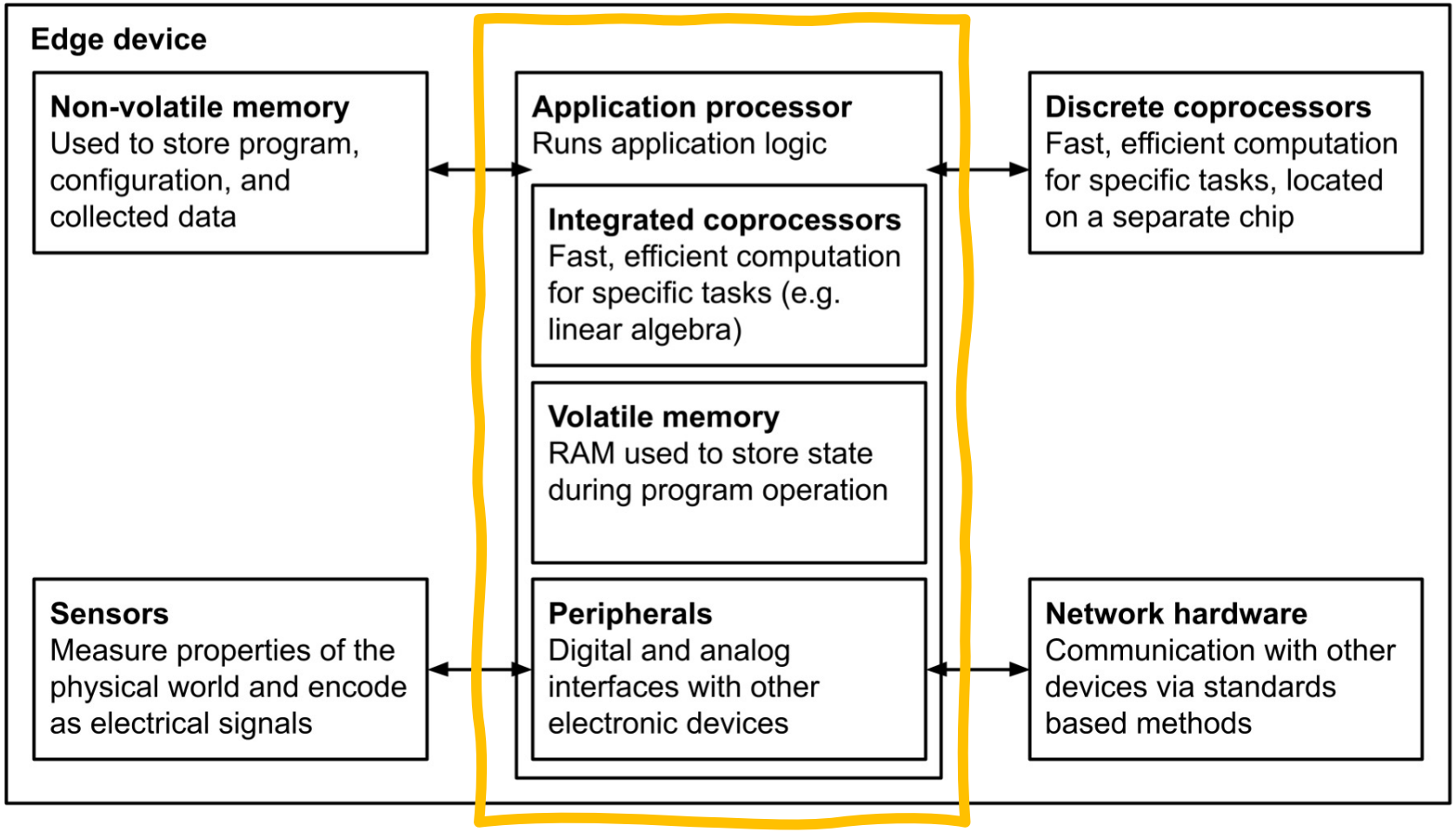
6) Environmental, biological, and chemical

- Temperature sensors
- Gas sensors—e.g., humidity or carbon dioxide sensors.
- Particulate matter sensor—monitor pollution levels.
- Biosignals sensors—e.g., measurement of electrical activity in the human heart (electrocardiography) and brain (electroencephalography).
- Chemical sensors—measure the presence or concentration of specific chemicals.

Measurements are typically represented as a time series

Rock collapse and landslide forecasting (Torrioni di Rialba)

Let's go back the application processors…



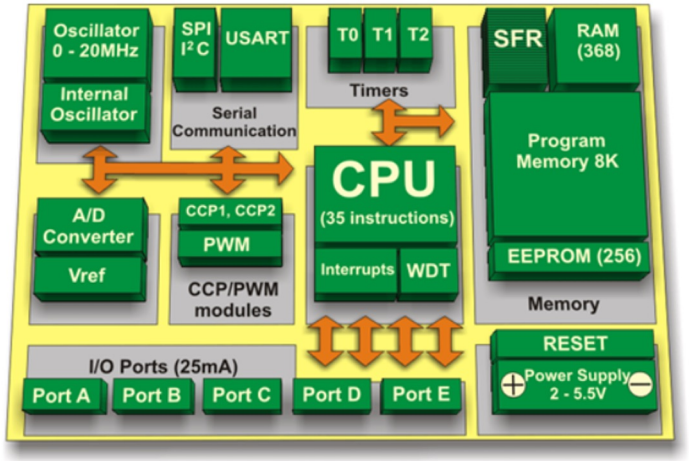
The general purpose processor running the embedded application

Microcontrollers and digital signal processors

- Microcontrollers are the basic technological block of our pervasive applications
- Tiny and cheap computers used for single purpose applications
- No operating system:
  - The "firmware", i.e., the software running on a microcontroller, is directly executed on the hardware and includes the low-level instructions to connect the peripherals
- But…. the key aspect of microcontrollers is the fact that they embed all the components in a single piece of silicon!

The (embedded) architecture of a micro-controller

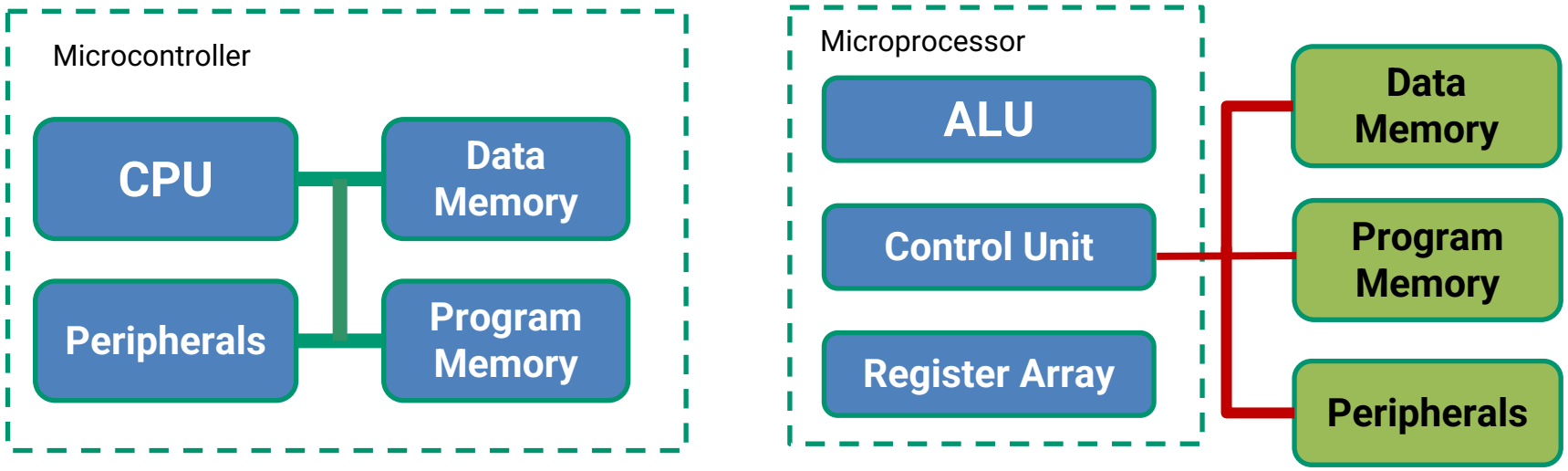
- Have a fixed hardware built around a central processing unit (CPU)
- The CPU controls a range of peripherals, which may provide both digital and analog functions such as timers and analog-to-digital converters.
- Small devices usually include both volatile and non-volatile memory on the chip but larger processors may need separate memory
- Their operation is usually programmed using a machine-language like Assembly or high-level language such as C



Microcontrollers are the best candidates for embedded systems, because their on-chip peripherals and memory enable the embedded system designers to save circuit space and overall dimensions, by not having to add these things on the board.

Microcontroller vs microprocessors

- A **microcontroller** contain a microprocessor (uP), Program Memory, Data Memory and a number of internal peripherals devices such as timers, serial ports, general purpose input/output (I/O) pins, counters, analog-to-digital (ADC) converters, etc. all inside a single silicon chip.
- A **microprocessor (uP)** is a silicon chip containing an advanced Central Processing Unit (CPU) that fetches program instructions from an external memory and execute them.



The wide range of microcontroller…

	Low-end MCUs	High-end MCUs
Description	low cost, small size, and energy efficiency.	Widely used microcontroller
Architecture	4-bit to 16-bit	32-bit
Clock speed	<100 MHz	<1000 MHz
Optional HW	-	Floating point unit (FPU), Single instruction, multiple data (SIMD) instructions
Flash memory	2-64 KB	16KB to 2MB
RAM memory	64 bytes to 2KB	2KB to 1MB
Input-output	Digital	Digital and Analog
Current draw	Running: Tens of mA at 1.5-5V Sleeping: uA	Running: Several tens of mA at 1.5-5V Sleeping: uA
Cost	1-2$	10$
Applications	Running conditional logic, acquire and transmit data	Signal processing and embedded machine learning

The wide range of microcontroller…

- Faster clock and 32-bit architecture
- SIMD to run several computations in parallel (very useful for Singnal Processing and ML)
- Larger RAM (that is always the bottleneck)
- Reasonably good energy efficiency

- Examples of high-end MCUs (based on ARM Cortex-M cores):
  - STM32
  - Nordic
  - Arduino

High-end MCUs
Widely used microcontroller
32-bit
<1000 MHz
Floating point unit (FPU), Single instruction, multiple data (SIMD) instructions
16KB to 2MB
2KB to 1MB
Digital and Analog
Running: Several tens of mA at 1.5-5V Sleeping: uA
10$
Signal processing and embedded machine learning

A different perspectives: System on Chip (SoC)

SoC devices aims at integrating all the functionalities of a traditional computing system into a single chip:

- Run Oss
- Tools and libraries to write server and desktop applications are used

Drawbacks:

- Low energy efficiency (w.r.t. microcontroller)
- Larger costs

System on Chip
Powerful edge computing system
64-bit architecture
>1 GHz clock speed
Hardware accelerator (GPU, TPUs,)
Flash in the order of GBs
RAM in the order of GBs
High performance Digital and Analog
Running: hundreds of mA at 5V Sleeping: uA
Tens of $ per unit
Technological scenario of mobile phones, smart cars, etc.. (e.g., Qualcomm Snapdragon)

A technological overview for embedded and edge AI

Device	Low frequency time series	High frequency time series	Audio	Low resolution image	High resolution image	Video
Low-end MCU	Limited	Limited	None	None	None	None
High-end MCU	Full	Full	Full	Full	Limited	Limited
High-end MCU with accelerator	Full	Full	Full	Full	Full	Limited
SoC	Full	Full	Full	Full	Full	Full
SoC with accelerator	Full	Full	Full	Full	Full	Full
Edge server	Full	Full	Full	Full	Full	Full
Cloud	Full	Full	Full	Full	Full	Full