# D2.2. Specification of the Architectural Framework

This document serves as deliverable **D2.2. Specification of the Architectural Framework** in the CORNET-funded **Trusted IoT project**.

This deliverable includes the overall framework architecture to define the different components, which meet these requirements and define the intercommunication among these components in order to address all the requirements in an integrated way.

# USE CASE 1: Industry 4.0 (BTU)

*see document "**D2.1 Requirements Specification for the Use Cases**"*

# USE CASE 2: Mobile Robots (TUD)

In D2.1. Requirements Specification for the Use Cases the requirements for the TU Dresden use case in WP4 are analyzed. This document describes the architectural framework that will be used by giving the outline for the implementation that will start in WP3.

## Proposed Robotic System

The setup, with the use-case of WP4, will incorporate an ultra-low-powered FPGA and/or use hardware or software methods to be as energy efficient as possible.

Fig 1. shows the different components that are present in the mobile robot. Because FPGAs are rich in I/O interfacing, all components, namely sensors, actuators, processing components, and user interfacing, can be centralized on a single device.

From the I/O perspective, it is easy to constrain a low-power mobile robot. However, the challenge of incorporating such a device is to achieve the required performance with only a centralized processing architecture. The basis of this challenge is that, in an autonomous mobile robot, the inability to satisfy real-time performance requirements poses a fatal problem in terms of safety.

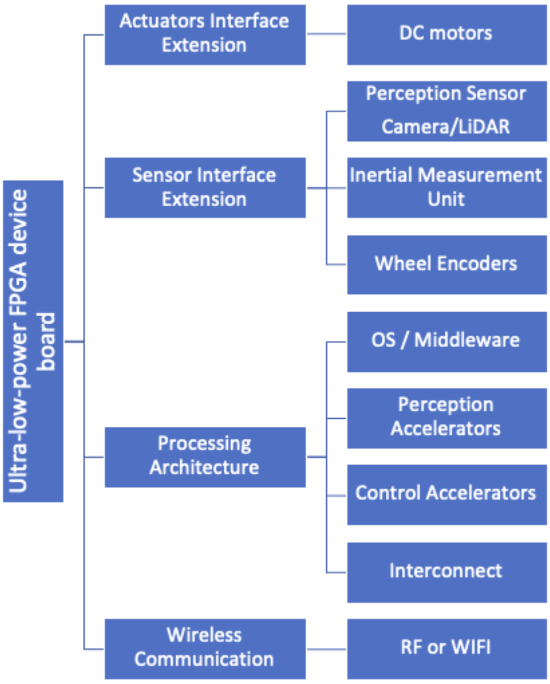


*Fig. 1: Block diagram of the components in the proposed ultra-low-power FPGA-based mobile robot*

Another challenge is that the tremendous amount of software and hardware components require modularity for further reusability, upgradability, ease of use, and performance. For this, a middleware is also proposed to be included in the platform.

# Components

As previously explained in the **Requirement Specification** document, the main components of a mobile robot can be split between Hardware and Software.

## 2.1. Sensors

The implemented sensors on the mobile robot, from the simplest, are:

- Wheel encoders: Usually enclosed with the actuators, they should have an excellent resolution to allow proper localization without too much loss in time.
- Inertial Measurement Unit (IMU): These components enclose a set of sensors useful for the assessment of the inertial state of the robot. The sensors are: at least an accelerometer, but, very often, also a gyroscope, magnetometer, barometer, and temperature. These sensors measure the acceleration of the object, the angular speed of the object, the direction of the magnetic field, and the atmospheric pressure (useful for altitude estimation). The former three can be combined in unique algorithms to very accurately estimate the pose of the robot, e.g., a Kalman Filter.
- Light detection and ranging (LiDAR) and/or Camera: These devices allow the robot's perception capability. By giving it a map composed of LiDAR or Camera measurements of landmarks, the robot could estimate its pose and localization within this map by comparing what it is sensing at the moment and the landmark map. For

scenarios without a prior map, localization is achieved by solving the simultaneous localization and mapping (SLAM) problem.

- Global Positioning System (GPS): They provide the three translational DoF in a global coordinate. However, they have two significant drawbacks: they are very slow and don't provide the three rotational DoF if the satellite signals are blocked indoors.

## 2.2. Robot Locomotion Mechanism

The robot will rely on low-power DC motors combined with omnidirectional wheels for locomotion. These wheels are comprised of small discs (called rollers) around their circumference, which can be perpendicular or diagonal to the turning direction (depending on how the wheels are arranged). The capabilities these wheels provide are that they can be driven with full force in any order and even allow the robot to slide laterally.

The setup will be a four-wheeled skid-drive robot setup with Omni wheels that allows movements in any direction. A diagram of the system is portrayed in the figure below.

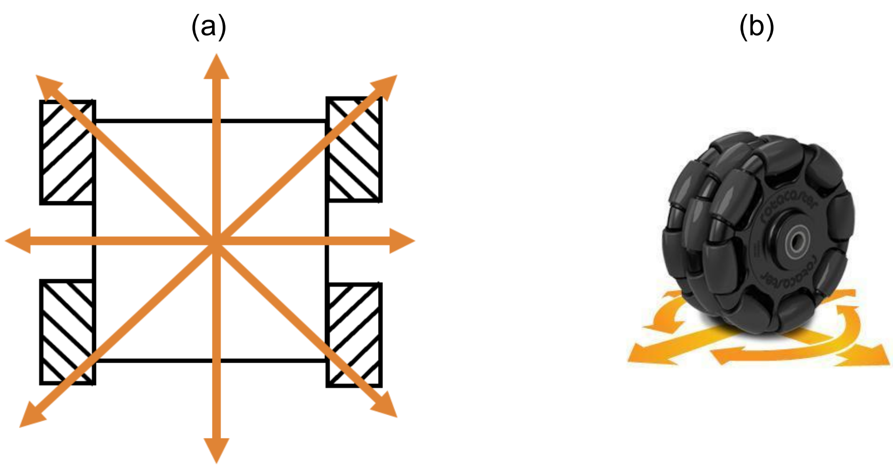


*Fig. 2: Mobile robot locomotion setup. (a) Omnidirectional skid-drive setup, and (b) Omnidirectional wheel [1].*

## 2.2. Batteries

The mobile robot should have increased autonomy and be energy-efficient. Mobile robots are usually battery-powered while requiring to be low-weight and have an excellent power-to-weight ratio. Li-ion batteries have one of the highest power-to-weight ratio of other battery technologies today (100-265 Wh/kg or 250-670 Wh/L). In addition, Li-ion battery cells can deliver up to 3.6 Volts, three times higher than technologies such as Ni-Cd or Ni-MH. This means they can deliver enormous amounts of current for high-power applications. At the same time, Li-ion batteries are also comparatively low maintenance and do not require scheduled cycling to maintain battery life [2].

## 2.3. Processing device

The mobile robot must be ultra-low-power while retaining high computation performance for localization and perception algorithms performed on edge. There are thus two proposed platforms that will be studied for the implementation of the proposed mobile robot platform:

1. **Ultra96-V2**: composed of a Xilinx Zynq UltraScale+™ MPSoC ZU3EG A484, 2GB LPDDR4 Memory, 802.11b/g/n Wi-Fi and Bluetooth, and two 40-pin and 60-pin 96Boards Low and High speed expansion headers. It also has a size of 85mm x 54mm.
2. **KR260 Robotics Starter Kit:** composed of a Zynq UltraScale+™ MPSoC EV (XCK26), 4GB DDR4, 4x RJ45 Ethernet Ports (10/100/1000), and x4 Pmod 12-pin interface and x1 Raspberry Pi HAT header with 26 I/Os. This board is 119mm x 140mm x 36mm.

Both devices have Xilinx's Ultrascale+ technology and have built-in TSMC's 16nm FinFET+ high-performance, low-power semiconductor process. They can deliver up to 60% overall device-level power savings over 7 series FPGAs and SoCs. The power-saving architectural enhancements include:

- Hardware-based clock gating
- Hardened block RAM cascading
- DSP block efficiencies
- Power-optimized transceivers

In addition to all the aforementioned, Zynq® UltraScale+ MPSoCs utilize multiple power islands and domains within the processing system for coarse-grained and fine-grain dynamic power gating to continually adjust power consumption to performance demands—lowering overall device power.

(a) (b)

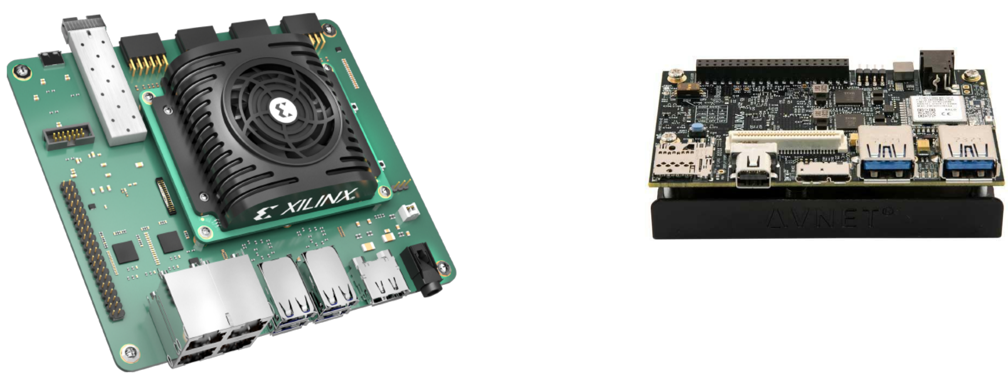


*Fig. 3: Proposed development boards for low power mobile robot. (a) KR260 Robotics Starter Kit, and (b) Ultra96-V2.*

The header expansion in both boards makes them perfect candidates for the sensors and actuators envisioned in the project.

## 2.4. Software and Middleware

For the system to have modularity, ease of use, easy integration, and easy upgradability, it is essential to implement it on top of middleware software. ROS (Robot Operating System) is an open-source software development kit for robotics applications [3].

ROS can be explained as the addition of plumbing + tools + capabilities + robust and constantly upgraded ecosystem.

- **Plumbing:** ROS provides a publishing-subscribing infrastructure based on messages that support the construction of distributed systems easily.
- **Tools:** It also provides tools for configuring, starting, introspecting, debugging, visualizing, logging, and testing distributed robotic systems.
- **Capabilities:** Provide a pervasive collection of libraries that implement proper state-of-the-art robot functionalities such as SLAM, motion planning, artificial intelligence, and perception.
- **Ecosystem:** ROS is supported and improved by a large community, with a strong focus on integration and documentation.

In addition, this project proposes using ROS 2, which uses the DDS standard for implementing the interfaces. It also supports real-time and more granular execution models, and custom executors can be implemented quickly.

[1] Rotacaster, https://www.rotacaster.com.au
[2] Lithium-Ion Battery,
http://www.cei.washington.edu/education/science-of-solar/battery-technology/
[3] Why ROS?, https://www.ros.org/blog/why-ros/

# USE CASE 3: Drones 4.4 (KU Leuven)

In D2.1. Requirements Specification for the Use Cases the requirements for the KU Leuven use case in WP4 are analyzed. This document describes the architectural framework that will be used for this use case and gives the outline for the implementation that will start in WP3.

## Proposed model

The setup, with the use-case of WP4 in the back of the mind, will incorporate **at least** one instance of the chosen RISC-V implementations.

The image below shows the different components that are generally present on a drone. It becomes clear that all the interaction happens in a star-shaped pattern with the **MCU** acting as the hub.

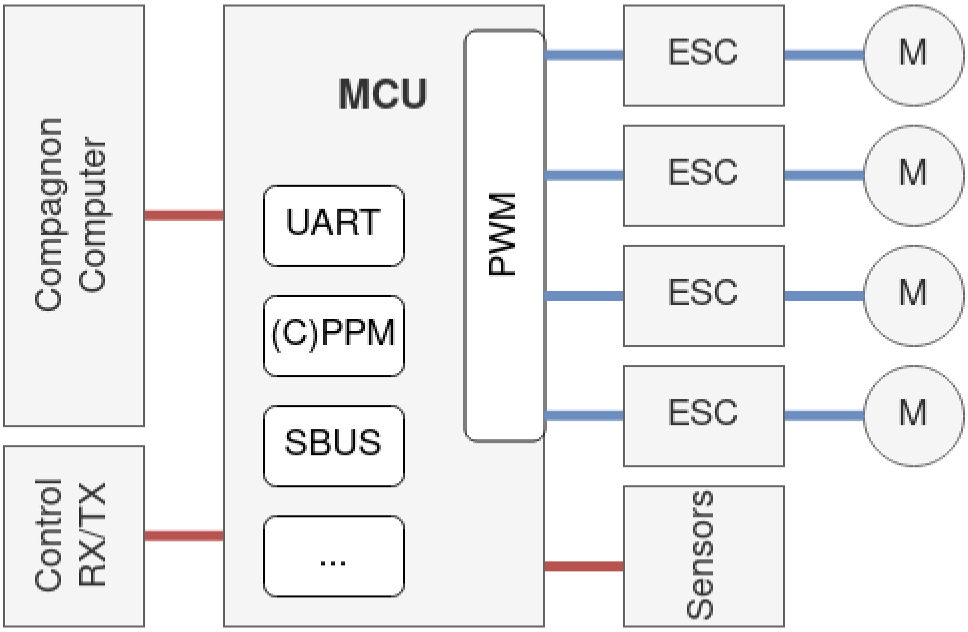


*Fig. 4: Block diagram of the components on a single drone*

## Components

The central component is a microcontroller labeled **MCU** in Fig. 4. A more prevailing name that is used for the MCU is the **Flight Computer** (FC). The main goal of the FC is to gather all inputs and commands; and use the obtained information to direct the rotors that steer the drone.

To be capable of actual flight, the FC needs to get instructions and sensor data. These are labeled as **Control RX/TX** and **Sensors** in Fig. 4, respectively. In a text-book setting, the instructions in which direction the drone should move, come from the user through a remote control. Using wireless communication technology, these instructions are sent from the user or operator to the drone.

Next to the inputs of the user, the drone also gathers input from sensors. The main sensors that serve this purpose are those that are present in the **Inertial Measurement Unit** (IMU), which are:

- accelerometers in 3 axis which measure the accelerations along each axes
- gyroscopes in 3 axis which measure the rotational velocity around each axes
- magnetometers in 3 axis which measure the local magnetic field components along its axes

The input of these 9 sensors allow the FC to make adjustments in the motor steering to achieve the desired behavior. Next to the more basic sensors that make up the IMU, other sensors could be added, for example GPS, ultrasonic range finders, air pressure sensors, lidars, or camera's.

Cameras are a classical example of a drone's payload. When a drone flys a mission, it typically has a purpose, e.g. make aerial video footage of a landscape. The payload that is required for a mission is often referred to as the payload. This can be one or more cameras, a parcel or first-aid kits. Sometimes it is required, for example in the case of cameras, that data processing is required. This can be offloaded to a **Compagnon Computer**, thusly labeled in Fig. 4

Finally, to achieve flight, a drone needs to generate lift. For the classical quadcopter this can be achieved by using 4 propellers that are placed on 4 motors (labeled **M** in Fig. 4). To lighten the computational burden of the FC, each motor is controlled by an Electronic Speed Controller (labeled **ESC** in Fig. 4).

When all these aforementioned components work in harmony, drone aviation can be achieved.

## The model

The proposed model for implementation is shown from an architectural point of view in Fig. 5.

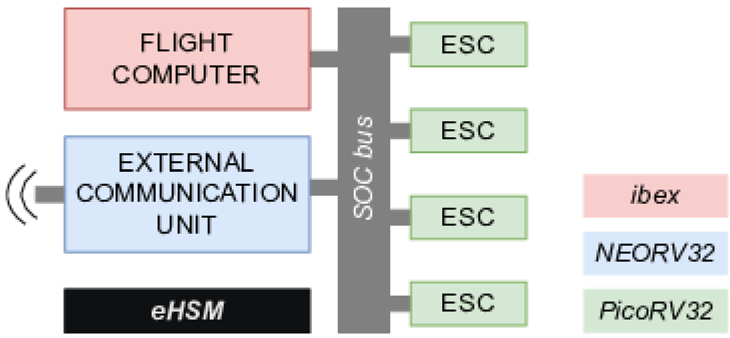


*Fig. 5: Block diagram of the components on a single drone*

In the assumed model the 3 selected RISC-V implementations are present.

The **flight computer** will have to be able to run PX4 [2]. Although actually running PX4 is no hard requirement, this will nonetheless define the requirements of the ibex core.

The **electronic speed controllers** (ESC) will run custom firmware.

The **external communication unit** will act as a placeholder for "any other core". As external communication is required for attestation, this component is chosen to be the placeholder.

Finally the **embedded Hardware Security Module (eHSM)** will contain all the required co-processors to achieve attestation and secure communication. This module will be customly developed.

The following components are defined:
- firmware on the FC
- firmware on the ESCs
- firmware on the external communication unit
- hardware and firmware on the eHSM

Every single processor will run its firmware.

## Component interaction protocols

To allow the different components to interact with each other and/or the outside world, different communication protocols are used. Below is a(n incomplete) list of used protocols in drones and systems-on-chip (SOC) more generally.

| ***Protocol*** | ***Brief description*** |
|---|---|
| RS-232 (UART) | A low bandwidth, serial communication standard |
| PPM | A one-to-many modulation for low bandwidth communication, based on the position of a pulse in a periodic signal |
| SBUS | A one-to-many protocol for low bandwidth communication |
| PWM | A one-to-one modulation for low bandwidth communication, based on the width of a pulse |
| 3G/4G/5G | Cellular network technology, best know for GSM |
| WiFi | Wireless network protocol, standardized in IEEE802.11 |
| Wishbone | An open source computer bus. It is part of the public domain and can therefore be used at no cost nor with any royalty obligations. |
| AXI | Advanced eXtensible Interface (AXI) is a on-chip communication bus protocol, developed by ARM. The use of the AXI bus is without royalties and is made freely available by ARM. |
| STBus | STMicroelectronics best solution for interconnecting IPs, of any data width, clock frequency and complexity, in a system-on-chip. Can be interfaced to APB/AHB/AXI IP Cores and peripherals |
| TileLink | Free and open bus architecture from CHIPS Alliance |

## Intended platform

All three RISC-V implementations roughly take between 2’000 and 10’000 Look-Up Tables (LUTs). Making a system-on-chip with these implementations can easily double or triple this amount.

*Fig. 6: The VC707 development board, by Xilinx*

The VC707 development board (shown in Fig. 6) is made by Xilinx and will be used for the development of the use case. The used FPGA (a Virtex-7 XC7VX485T-2FFG1761C) is a higher end product of the most recent 7-series. It roughly packs half a million LUTs so it should easily support the implementation.

With its 2 FPGA Mezzanine Card (FMC) connectors, there should also be sufficient Input/Output-pins available.

# USE CASE 4: Environmental Monitoring (VUB)

In D2.1. Requirements Specification for the Use Cases the requirements for the VUB use cases in WP4 are analyzed. This document describes the architectural framework that will be used for this use case and gives the outline for the implementation that will start in WP3.

# Proposed systems

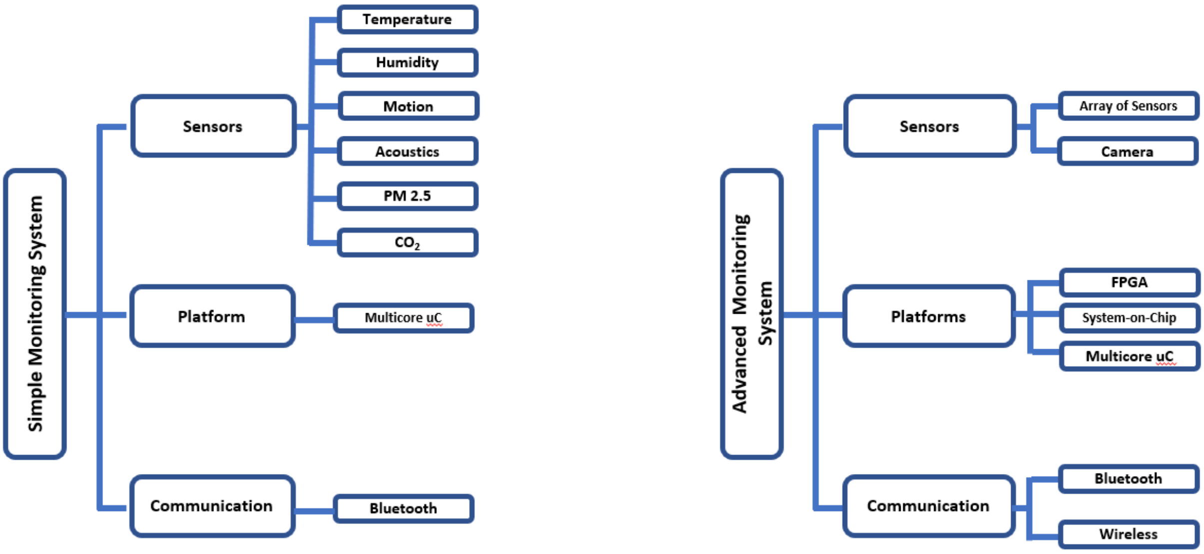


*Fig. 7: Block diagram of the components in the proposed environmental monitoring systems*

The setups considered in this use case present different levels of complexity based on their components and target platforms. Fig. 7 depicts the different components of the two environmental monitoring systems. The level of complexity is determined by the type of sensors used and, as a consequence, the capabilities of the platform to provide enough computational resources for data acquisition. Both monitoring systems present different challenges, either to interface an array of different sensors or to process the gathered data. More complex sensors such as cameras or microphone arrays lead to additional computational, memory and communication capabilities in order to process large quantities of data in real-time. Both systems are representative cases of existing monitoring systems.

The simple monitoring system is composed of single and low-data rate sensors used for environmental monitoring. The platform, mainly based on a multi-core microcontroller, needs to provide enough I/Os to interface all the sensors. A relatively low data rate RF wireless standard such as Bluetooth or Bluetooth Low Energy will be used to send the gathered data to a server.

The advanced monitoring system presents higher demands in terms of data processing. The system comprises at least one camera and an array of microphones. Due to the nature of the sensors, a large amount of data is collected, increasing the memory and the computational demands to provide a real-time response. In this system, the data is locally processed before being sent to a server.

# Components - Monitoring sensors (Laurent and Bruno)

## Simple monitoring system

The components used for the simple monitoring system mainly consist of low throughput and commercial of the shelf (COTS) components. The selection of the components is chosen to cover the most typical environmental parameters.

- Local temperature and humidity measurements allow to assess the impact of local arrangement on temperature. It is known that more green areas have a positive impact on both parameters.
- Motion measurements can be acquired by means of an inertial movement unit (IMU). Such sensors are typically composed of an accelerometer and a gyroscope. The purpose of this sensor is to measure vibration induced by traffic, crowd movement during manifestations, industrial infrastructure, etc.
- Acoustics help to detect activity or acoustic pollution. The sound pressure level (SPL) is a typical parameter recorded on platforms with low processing capabilities.
- A PM 2.5 sensor – particle sensor allows to evaluate the air quality and the live-ability of a given area. Transportation (i.e. braking systems, malfunctioning filters, etc.), heating of buildings, industrial areas, etc. are known to cause respiratory problems.
- The CO2 sensor helps to assess the amount of locally emitted CO2 by traffic, industries, etc. Urban areas are more prone to present higher levels of CO2, resulting in higher levels of health risks for the local residents.

Most sensors listed above can be interface in an analog and digital fashion. Analog sensors require additional analog to digital converters while sensors with digital interfacing typically transmit the sensor data via SPI, I2C and UART. Digital acoustic sensors (microphone) typically communicate via I2S or PDM.

A GPS module is added to ensure that the gathered environmental data can be matched with a given location. GPS modules typically compute the location of the module internally before transmitting it to the microcontroller or system. Some GPS modules also offer the possibility to obtain a very fine-grained timestamp along with a pulse per second tick (PPS). The transmission of the location and time is usually done via UART/I2C while the a general purpose IO pin (GPIO) is used for the PPS signal.

## Advanced monitoring system

This system focusses on more advanced acoustic, image and video monitoring. An array of microphones together with a camera provide a more fine-grained analysis of traffic, crowd formation, (urban) wildlife detection, etc. of the direct environment of the deployed system. The array of microphones and the camera require protocols with higher bandwidths compared to the sensors listed for the “Simple monitoring system”.

- Microphones with digital interfacing like PDM and I2S do not require addition analog hardware and can be directly interfaced with the processing platform, reducing overall complexity.
- Cameras come in different forms, flavors and communication protocols. Many desktop/laptop webcams with USB (type 2.0 A or type C) can nowadays be found. While USB 2.0 and USB-C can be found on most advanced devices, only few devices do support USB-C. Alternative cameras with SPI (for image data) and I2C

(configuration) are available and specifically target embedded systems. Although the throughput is typically lower compared to the USB counterparts, the advanced monitoring is required to have a sufficient amounts of memory and processing capabilities to process the images.

## Functionality

The acquired data is intended to be locally processed before being sent to a local server or to the cloud. In both systems, the data can be gathered and compressed to reduce the bandwidth needed. That implies that decisions can be made locally (e.g., alarm activation), but also that the platforms need to be powerful enough to processed the collected data. Nonetheless, the systems must be power efficient in order to operate in a battery mode if needed.

## Platforms

Based on the systems' requirements, different type of platforms are considered:

- Simple monitoring system: simple multi-core microcontrollers are considered for this system. Existing boards from STM, NXP, Microchip vendors will be considered and selected based on their security capabilities (Fig. 8).

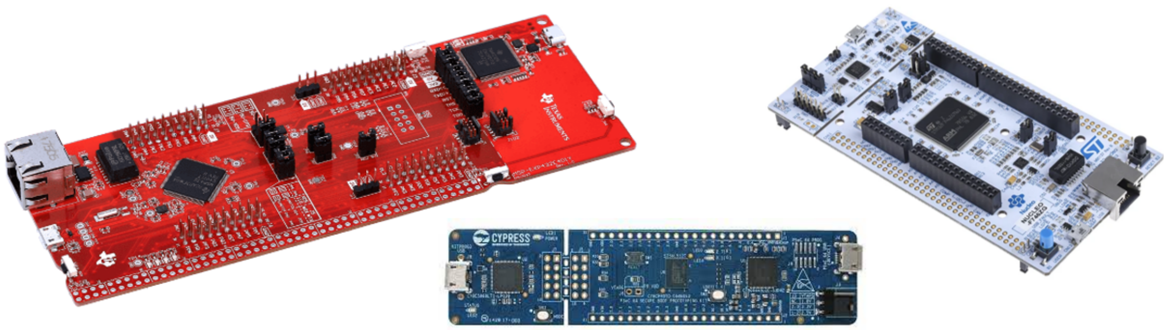

*Fig. 8: Development boards from Texas Instruments, Infineon and STMicroelectronics comprising a multi-core microcontroller with embedded security features.*

- Complex monitoring system: Due to the characteristics of the sensors, advanced embedded devices must be considered. Firstly, to provide the computational, memory, I/Os and communication capabilities demanded for real-time image/video/audio monitoring. Secondly, to provide security features, either in terms of embedded security components or through the use of novel AI-dedicated processing units. Fig.9 depicts some of the considered platforms, including heterogeneous SoC, FPGAs or TPUs.

*Fig. 9: Development boards comprising FPGA, Heterogeneous SoC or TPUs for the advance monitoring system.*

# USE CASE 5: Cooperative robotics (GFaI)

*see document "**D2.1 Requirements Specification for the Use Cases**"*