IoT-NGIN

D4.2
Enhancing IoT Ambient Intelligence

WORKPACKAGE	WP4	**PROGRAMME IDENTIFIER**	H2020-ICT-2020-1
DOCUMENT	D4.2	**GRANT AGREEMENT ID**	957246
REVISION	V2.0	**START DATE OF THE PROJECT**	01/10/2020
DELIVERY DATE	30/11/2021	**DURATION**	3 YEARS

(revised version:15/07/2022)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 957246

DISCLAIMER

ACKNOWLEDGEMENT

This document is a deliverable of IoT-NGIN project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 957246.

The opinions expressed in this document reflect only the author's view and in no way reflect the European Commission's opinions. The European Commission is not responsible for any use that may be made of the information it contains.

PROJECT ACRONYM	IoT-NGIN
PROJECT TITLE	Next Generation IoT as part of Next Generation Internet
CALL ID	H2020-ICT-2020-1
CALL NAME	Information and Communication Technologies
TOPIC	ICT-56-2020 - Next Generation Internet of Things
TYPE OF ACTION	Research and Innovation Action
COORDINATOR	Capgemini Technology Services (CAP)
PRINCIPAL CONTRACTORS	Atos Spain S.A. (ATOS), ERICSSON GmbH (EDD), ABB Oy (ABB), INTRASOFT International S.A. (INTRA), Engineering-Ingegneria Informatica SPA (ENG), Bosch Sistemas de Frenado S.L.U. (BOSCH), ASM Terni SpA (ASM), Forum Virium Helsinki (FVH), Optimum Technologies Pilroforikis S.A. (OPT), eBOS Technologies Ltd (EBOS), Privanova SAS (PRI), Synelixis Solutions S.A. (SYN), CUMUCORE Oy (CMC), Emotion s.r.l. (EMOT), AALTO-Korkeakoulusaatio (AALTO), i2CAT Foundation (I2CAT), Rheinisch-Westfälische Technische Hochschule Aachen (RWTH), Sorbonne Université (SU)
WORKPACKAGE	WP4
DELIVERABLE TYPE	REPORT
DISSEMINATION LEVEL	PUBLIC
DELIVERABLE STATE	FINAL
CONTRACTUAL DATE OF DELIVERY	30/11/2021
ACTUAL DATE OF DELIVERY	16/12/2021 (revised version: 15/07/2022)
DOCUMENT TITLE	Enhancing IoT Ambient Intelligence
AUTHOR(S)	Josep Escrig [I2CAT], Marisa Catalan [I2CAT], Mario Montagud [I2CAT], Terpsi Velivassaki (SYN), Artemis Voulkidis (SYN), Marios Sophocleous[EBOS],Antonios Gkonos[OPT],Antonello Corsi[ENG], Francesco Bellesini[EMOT]
REVIEWER(S)	T.Terpsi Velivassaki (SYN)), Antonello Corsi[ENG]
ABSTRACT	SEE EXECUTIVE SUMMARY
HISTORY	SEE DOCUMENT HISTORY
KEYWORDS	Ambient Intelligence, Tactile Internet, object recognition, AR, device identification, device indexing, device access control, IoT, localization, computer vision, UWB, VLP

Document History

Version	Date	Contributor(s)	Description
V0.1	4/08/2021		TOC
V0.2	22/09/2021	SYN	Updated ToC
V0.3	2/11/2021	I2CAT,SYN, eBOS, eMOTION, Optimum,	Contributions sections 2,3,4,5
V0.4	24/11/2021	I2CAT, SYN	Contributions sections 3,4
V0.5	10/12/2021	I2CAT	Document merges and contributions sections 1,3,4,6; Final document prior to internal review
V1.0	15/12/2021	I2CAT, CAP	Quality check; Final version
V2.0	15/07/2022	I2CAT	Reviewed version; Minor changes (section number mismatch; clarification about UC #2 and #3 Ambient Intelligence requirements) addressed.

Table of Contents

Document History 4
Table of Contents 5
List of Figures 7
List of Tables 8
List of Acronyms and Abbreviations 10
Executive Summary 12
1 Introduction 13
1.1 Intended Audience 13
1.2 Relations to other activities 14
1.3 Document Overview 15
2 Ambient Intelligence and Tactile Internet 16
2.1 Object recognition techniques 17
2.1.1 Computer vision 17
2.1.2 Identification of devices using RF technologies 19
2.2 Object positioning techniques 19
2.2.1 Global Navigation Satellite Systems (GNSS) 20
2.2.2 RF positioning 20
2.2.3 Visual Light Positioning (VLP) 22
2.2.4 Homography 23
2.3 Device authentication techniques 24
2.3.1 X.509 Certificates 25
2.3.2 Trusted Platform Modules 26
2.3.3 Symmetric Keys 26
2.4 Augmented/Mixed Reality 26
2.5 5G 27
2.6 Edge computing 29
2.6.1 MEC and 5G 31
3 IoT Ambient Intelligence requirements from the Living Labs 32
3.1 Human-Centred Twin Smart Cities Living Lab 33
3.1.1 Preconditions per application 34
3.1.2 Mapping to IoT-NGIN Ambient Intelligence tools 34
3.1.3 Use Case Requirements analysis 34

3.1.4 User Story ... 35
3.2 Smart Agriculture IoT Living Lab ... 36
3.2.1 Preconditions per application ... 36
3.2.2 Mapping to IoT-NGIN Ambient Intelligence tools ... 37
3.2.3 Use Case Requirements analysis ... 37
3.2.4 User Story ... 39
3.3 Industry 4.0 Living Lab ... 40
3.3.1 Preconditions per application ... 41
3.3.2 Mapping to IoT-NGIN Ambient Intelligence tools ... 42
3.3.3 Use Case Requirements analysis ... 43
3.3.4 User Story ... 46
3.4 Energy Grid Active Monitoring/Control Living Lab (EMOT) ... 46
3.4.1 Preconditions per application ... 46
3.4.2 Mapping to IoT-NGIN Ambient Intelligence tools ... 47
3.4.3 Use Case Requirements Analysis ... 47
3.4.4 User Story ... 48
4 Ambient Intelligence in IoT-NGIN ... 50
4.1 IoT Device Discovery ... 52
4.1.1 Description ... 53
4.1.2 Interfaces ... 62
4.2 IoT Device indexing ... 64
4.2.1 Description ... 65
4.2.2 Interfaces ... 67
4.3 IoT Devices access control ... 67
4.3.1 Description ... 68
4.3.2 Interfaces ... 70
4.4 AR/VR module ... 70
4.4.1 Description ... 71
4.4.2 Interfaces ... 71
4.5 IoT Device actuationion ... 72
5 Integration with the IoT-NGIN architecture ... 74
6 Conclusions ... 76
7 References ... 77

List of Figures

Figure 1: Work packages structure 14
Figure 2: R-CNN (Regions with CNN features) (Image source [12]) 18
Figure 3: Joseph Redmon's (YOLO) model (Image source [13]) 19
Figure 4: UWB location methods. ToF, TDOA and AoA [26]. 22
Figure 5: Architecture of a VLP indoor positioning system [28] 23
Figure 6: Example of an homography transformation of an image 24
Figure 7: Delay vs Bandwidth requirements for multiple applications [34]. 27
Figure 8: (left) Comparison of 5G bandwidth with other networks, (right) overall 5G technical capabilities [36] 28
Figure 9: Massive IoT vs Critical IoT [37]. 29
Figure 10: Overview of wireless connectivity requirements [38] 29
Figure 11: High-level architecture of the IoT-NGIN Ambient Intelligence platform 51
Figure 12: ML model training cycle. 53
Figure 13: Generic architecture of deep learning models. 54
Figure 14: Example of predicted bounding box compared to the ground-truth bounding box. 55
Figure 15: IoU definition and example. 56
Figure 16: VLP-based positioning system. Non-visual recognition. 57
Figure 17: Location Module for the VLP based positioning. Non-visual recognition. 58
Figure 18: The *Code Scanning* module of the *IoT Device Discovery* component 59
Figure 19: UWB-based positioning architecture elements. 60
Figure 20: Location Module for the UWB-based positioning. Non-visual recognition. 61
Figure 21: UWB initial deployment at i2CAT. 61
Figure 22: Positioning of the Device Indexing module in the project activities. 64
Figure 23: FIWARE-compliant high-level IoT architecture (source: [62]). 65
Figure 24: IoT Agent COAP example, internal interactions (source: [67]). 66
Figure 25: Device Indexing module architecture. 67
Figure 26: Positioning of the Devices Access Control module in the overall Ambient Intelligence-related components of IoT-NGIN. 68
Figure 27: High-level architecture of the Devices Access Control module. 69
Figure 28: The high-level architecture of IoT-NGIN. 74

List of Tables

Table 1: Relation of WP4 activities to other WPs and tasks 14
Table 2: Technologies enabling AmI and TIoT. 16
Table 3: Various CNN-based object recognition approaches. 17
Table 4: Comparison of RF positioning technologies 21
Table 5: Summary of the living lab relevance on the WP4 requirements 32
Table 6: Use cases requiring WP4 software modules 33
Table 7: Mapping of Human-Centred Twin Smart Cities Living Lab to IoT-NGIN Ambient Intelligence platform 34
Table 8: Requirements analysis for Use case #1 "Traffic Flow Prediction & Parking prediction". 34
Table 9: User story table for the "Traffic Flow Prediction & Parking prediction" use case. 35
Table 10: Mapping of Smart Agriculture IoT Living Lab to IoT-NGIN Ambient Intelligence platform. 37
Table 11: Requirements analysis for Use case #4 "Crop diseases prediction, smart irrigation and precision aerial spraying". 38
Table 12: Requirements analysis for Use case #5 "Sensor aided crop harvesting" use case 39
Table 13: User story table for the "Crop diseases prediction, smart irrigation and precision aerial spraying" use case 40
Table 14: User story table for the "Sensor aided crop harvesting" use case 40
Table 15: Mapping of Industry 4.0 Living Lab to IoT-NGIN Ambient Intelligence platform. 42
Table 16: Requirements analysis for Use case #6 "Human-centred safety in a self-aware indoor factory environment" 43
Table 17: Requirements analysis for Use case #7 "Human-centred augmented reality assisted build-to-order assembly" 44
Table 18: Requirements analysis for Use case #8 "Digital powertrain and condition monitoring " 45
Table 19: User story table for the "Human-centred safety in a self-aware indoor factory environment" use case. 46
Table 20: Mapping of Energy Grid Active Monitoring/Control Living Lab to IoT-NGIN Ambient Intelligence platform. 47
Table 21: Requirements analysis for Use case #10 "Driver-friendly dispatchable EV charging". 48
Table 22: User story table for the "Driver-friendly dispatchable EV charging" use case. 49
Table 23: IoT Device Discovery methods 52
Table 24: IoT Device Discovery interfaces 62

Table 25: IoT AR module interfaces 71

List of Acronyms and Abbreviations

AGLV	Automated Guided Land Vehicle
AGV	Automated Guided Vehicle
AI	Artificial Intelligence
AmI	Ambient Intelligence
AoA	Angle of Arrival
AP	Access Point
AR	Augmented Reality
BLE	Bluetooth Low Energy
CNN	Convolutional Neural Networks
DLT	Distributed Ledger Technology
ENISA	European Union Agency For Network And Information Security
FTM	Fine Time Measurement
GE	Generic Enabler
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
HOG	Histogram of Oriented Gradients
HSM	Hardware Security Module
IAR	IoT AR service
IDA	IoT Device Actuation
IDAC	IoT Device Access Control
IDD	IoT Device Discovery
IDI	IoT Device Indexing
IoT	Internet of Things
ITU	International Telecommunication Union
LL	Living Lab
LOS	Line of Sight

MR	Mixed Reality
NGSI	Next Generation Service Interfaces
NVF	Network Function Virtualization
POLP	Principle of Least Privilege
QR	Quick Response
RFID	Radio Frequency Identification
RoI	Region of Interest
RSSI	Received Signal Strength Indicator
RTK	Real Time Kinematic
SDN	Software Defined Networking
SE	Support Engine
SSD	Single Shot Detector
SSI	Self-Sovereign Identity
SVM	Support Vector Machine
TBD	To Be Defined
TDOA	Time Difference of Arrival
TIoT	Tactile IoT
TOF	Time of Flight
TPM	Trusted Platform Module
TSM	Tactile Service Manager
TWR	Two-way ranging
UC	Use Case
UWB	Ultra-Wide Band
VLP	Visual Light Positioning
VNF	Virtual Network Function
VR	Virtual Reality
WP	Work Package

Executive Summary

Ambient Intelligence in Internet of Things (IoT) enables the deployment of intelligent systems and environments with enhanced capabilities and services. IoT-NGIN aims to enable a set of innovative components and features for next-generation Ambient Intelligence IoT and demonstrate them via diverse living labs and use cases. This deliverable reports on initial contributions in that direction, within the scope of its WP4: Enhancing IoT Tactile & Contextual Sensing/Actuating.

Initially, the deliverable reviews in Section 2 key technological enablers for Ambient Intelligence IoT and Tactile Internet, by identifying those crucial to meet the objectives of the project. After that, in Section 3, the deliverable summarizes and categorizes the envisioned scenarios and requirements related to the Living Labs considered in the project, which serves to identify the technological contributions to be provided. In Section 4, the key components and modules being adopted and developed to enable the envisioned use cases are detailed, namely:

- IoT Device discovery: It will employ different mechanisms and technologies for the autonomous recognition of objects and people. Context awareness will be achieved in two ways: 1) the identification of objects (e.g. using visual methods) and 2) the spatial localization of moving devices (using visual and non-visual methods).
- IoT Device indexing: It will provide information related to IoT devices, such as their features, location or status. This information can be used to associate a physical device with its digital twin or can serve as input for advanced Augmented Reality or Device actuation services.
- IoT Device Access Control: It will supervise the access to IoT Devices, so that different features or actions will be available based on permissions.
- IoT Augmented Reality (AR): An IoT-AR assets repository will be implemented to be used in the use cases or by third parties allowing an enhanced interaction with IoT devices.
- IoT Device Actuation: It will allow to control or modify some features of IoT devices.

For each component/module, their planned sub-components, interfaces, involved protocols and data types, and related relevant information, like its interactions with other components, are detailed.

Finally, the integration of these components/modules within the complete end-to-end IoT-NGIN architecture is summarized in Section 5.

This deliverable reports on the initial version of technological contributions from WP4, so specific parts can report on work-in-progress or even on planned developments, and even current developments could evolve during the project's lifetime. This deliverable will be updated in two further iterations: D4.3 and D4.4, to be delivered in Month 22 and 31, respectively.

1 Introduction

Ambient Intelligence in the Internet of Things will allow to deploy intelligent environments and systems with enhanced functionalities to provide personalized services, anticipate the needs and desires of the users and simplify the interaction with digital services and their integration in everyday lives.

The IoT-NGIN project, and concretely WP4, aims to deploy a set of functionalities and components for enhanced ambient intelligence, which can be applied to deliver better services in the living labs and use cases proposed by the project. Based on the needs identified in WP1, IoT-NGIN focuses on the following features:

- IoT Device discovery: It will employ different mechanisms and technologies for the autonomous recognition of objects and people. Context awareness will be achieved in two ways: 1) the identification of objects (e.g. using visual methods) and 2) the spatial localization of moving devices (using visual and non-visual methods).
- IoT Device indexing: It will provide information related to IoT devices, such as their features, location or status. This information can be used to associate a physical device with its digital twin or can serve as input for advanced Augmented Reality or Device actuation services.
- IoT Device Access Control: It will supervise the access to IoT Devices, so that different features or actions will be available based on permissions.
- IoT Augmented Reality (AR): An IoT-AR assets repository will be implemented to be used in the use cases or by third parties allowing an enhanced interaction with IoT devices.
- IoT Device Actuation: It will allow to control or modify some features of IoT devices.

Based on the analysis of the requirements stated by the living labs, this document provides a first definition and design of the different IoT Ambient Intelligence components deployed in WP4 and summarizes the work performed in the initial stage of the project. It can be considered a living document which will be updated in D4.3 and D4.4 until the end of the project.

1.1 Intended Audience

The document can be considered as a reference for IoT-NGIN researchers, developers and participants in Open Calls to understand the functionalities of the IoT Ambient Intelligence components being deployed, to work on enhanced implementations and to facilitate the integration of the different modules in the IoT-NGIN architecture.

Also, the document can be useful to living labs and use cases owners to envision further necessities or requirements that could derive in future enhancements of the components and tools deployed in WP4.

Finally, the whole European AI and IoT communities will be potential readers of the present deliverable, as it includes information about employing AI to enable tactile internet in the next-generation IoT.

1.2 Relations to other activities

The following figure shows the position of WP4 in the full work package structure.

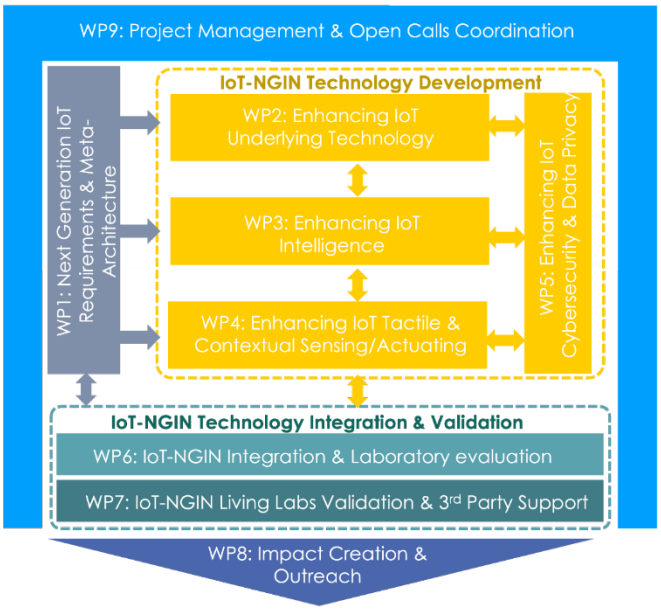

Figure 1: Work packages structure

Considering this, WP4 is related to the work packages and tasks described in Table 1.

Table 1: Relation of WP4 activities to other WPs and tasks

WP	Relation to WP4 and D4.2
WP1	The definition of the living labs and use cases requirements has served D4.2 for the definition of the IoT Ambient Intelligence components to be deployed in tasks T4.2, T4.3 and T4.4. Also, the role of the tools and modules deployed in WP4 shall be represented in the meta-architecture resulting by the activities in WP1.
WP2	No interaction is foreseen among WP4 and WP2.
WP3	The training of AI models through the Machine Learning as a Service (MLaaS) platform developed in WP3 may be useful for IoT Device Indexing module to recognize some objects and devices.
WP5	The IoT Device Indexing and the IoT Device Access Control modules deployed in WP4 will interact with different components developed within the scope of WP5. Concretely, there will be interactions with the IoT Vulnerability Crawler, the Digital Twins component and the Self-Sovereign Identities module.
WP6	WP4 components will be integrated with the rest of project's technologies and frameworks.

WP7	WP4 components will be implemented and used in several living labs and use cases as defined in Section 3. WP4 will support 3rd parties by offering a specific set of functionalities (such as the IoT-AR repository).
WP8	WP4 will provide notable outcomes and results for supporting impact creation activities. Moreover, it will consider feedback (e.g. from the market analysis and business modelling tasks) which could be relevant for the WP4 design and development.

1.3 Document Overview

The present deliverable is divided into seven sections:

- **Section 1** introduces the **motivation and general objectives** for the document, its intended audience, the relation to other project tasks and structure.
- **Section 2** analyses the main **enablers for Ambient Intelligence** (AmI) and Tactile Internet and identifies the technologies that will be the base for the work performed in WP4.
- **Section 3** analyses the **requirements of the living labs** and user stories in terms of IoT Ambient Intelligence. The resulting tables have been used to determine the IoT AmI enhancements that the IoT-NGIN project will address.
- **Section 4** introduces the **IoT Ambient Intelligence components** that are being deployed in WP4 as part of the IoT-NGIN architecture and provides an initial definition of the **functionalities and interfaces** of each module. Concretely, the following modules are described: IoT Device Discovery, IoT Device Access Control, IoT Device Indexing and IoT AR Service and Device Actuation.
- **Section 5** provides an overview of the **integration** of these components in the IoT-NGIN architecture.
- **Section 6** summarizes the **conclusions** of the document.

2 Ambient Intelligence and Tactile Internet

The concept of **Ambient Intelligence** was firstly introduced by the Information Society Technologies Advisory Groups of the European Commission in 2001 [1]. It refers to a digitalized environment that is sensitive and responsive and that proactively delivers support and services to the humans and machines living or working in it. Ambient intelligence relies on sensors embedded on the environment that can recognize people, objects and the situational context. From these observations, the ambient intelligent environment will produce a personalized response to the needs of the human that can even anticipate to his/her desires. Current Ambient Intelligence applications already cover different environments: smart homes, health monitoring and assistance, hospitals, transportations, emergency services, educations, work spaces, art, etc. [2].

On the other hand, **Tactile Internet** (or Tactile IoT) is defined by the International Telecommunication Union (ITU) as an internet with ultra-low latency, extremely availability, reliability and security [3]. It is designed for human-to-machine interactions and must provide to the humans a sense of touch, in particular the perception and manipulation of objects.

These two technologies, which are highly interrelated, will be enhanced in WP4, integrated into the IoT-NGIN meta-architecture and validated in the IoT-NGIN living labs. Ambient Intelligence (AmI) and Tactile IoT (TIoT) are supported in different degrees by several enabling technologies. Adopting or surpassing the state of the art of these enabling technologies and integrating them in AmI and TIoT environments will certainly enhance their capabilities.

Some of these enabling technologies that will be developed in WP4 "Enhancing IoT Tactile & Contextual Sensing/Actuating" are summarized in the following table and explained in the next sections.

Table 2: Technologies enabling AmI and TIoT.

Object recognition	Enhances context awareness
Object positioning	Enhances context/spatial awareness
Device authentication	Enhances security
Augmented and mixed reality	Enhances human-machine interaction
5G	Enhances ultra-low latency, reliability and massive IoT
Edge computing	Enhances ultra-low latency, reliability and security

Artificial Intelligence is another key technology that in this case would enhance the reasoning capabilities of the AmI. This technology is not included in WP4 since there is already a work package dedicated to this technology and its integration in the IoT-NGIN, which is WP3 "Enhancing IoT Intelligence".

2.1 Object recognition techniques

In this section, two types object recognitions methods that will be used in WP4 and in IoT-NGIN are explained. The first method is Computer Vision which is a visual method based on Artificial Intelligence and the second type of methods are based on radio frequency technologies which are non-visual alternatives.

2.1.1 Computer vision

As humans, we have the unique ability to comprehend and analyse a scenario captured in a picture. It isIt's not just about detection; it is also about describing each feature of an image and providing reasonable explanations for what is going on in the picture, as well as any potential motions or future transitions that could occur in the next second. Underneath the Artificial Intelligence technology, there is an entire research area called computer vision that studies this skill that we take for granted [4]. Given a two-dimensional image, a computer vision system's objective is to recognize the present objects and their properties. Face recognition, image analysis, visual searches like those performed by Google Images [5], and biometric identification methods are all problems that computer vision handles. The purpose of this work package is to provide state-of-the-art systems that can recognize various items in its living environment in real time while also being able to semantically index those objects to allow for intelligent queries.

Object detection in images and videos entails identifying and classifying a plethora of items in order to determine whether an object is, for example, a car, a person, or an IoT device [6]. This kind of technology, when supplemented with non-visual input from other sources such as sensors and/or radars, will allow the system to index the objects we wish to recognize in each of the aforementioned use cases. There will be also the option of registering IoT devices using a simple Quick Response (QR) code scanner for use cases that entail IoT device registration. Convolutional Neural Networks (CNN) will be employed as the core for the visual methods because they have been found to achieve outcomes that are beyond state-of-the-art when using typical computer vision techniques [7]. A dataset that contains the object the model must learn is required for this approach, known as supervised machine learning [8]. There are many approaches to follow for the object detection model that use CNNs with the combination of other Machine Learning algorithms like Support Vector Machines (SVMs). Below (Table 3) is a list of all the cutting-edge approaches that are currently being used for object detection purposes.

Table 3: Various CNN-based object recognition approaches.

No.	Description
1	Paul Viola and Michael Jones approach
2	Histogram of Oriented Gradients (HOG) with Support Vector Machines (SVM)
3	R-CCN (Regions with CNN features)
4	Fast R-CNN
5	Joseph Redmon's (YOLO) model
6	Single Shot Detector (SSD)

One of the most well-known approaches is one proposed by Paul Viola and Michael Jones in the paper "*Robust Real-time Object Detection*" [9]. This approach was first motivated by the objective of face detection but it can also be used to detect a diverse range of object classes. The advantages of using such an approach are that is fast and straightforward and also allows for real-time object detections with little processing power. The central feature of the approach is to train with a potentially large set of binary classifiers based on Haar Features [10]. These features represent edges and lines, and are extremely simple to compute when scanning an image. It is a supervised method that requires many positive and negative examples of the type of object to be discerned. Another traditional and similar method for classification is to use Histogram of Oriented Gradients (HOG) features and Support Vector Machine (SVM) [11]. It still necessitates a multi-scale sliding window, and while it outperforms Viola-Jones, it is much slower. Among the various deep learning approaches proposed for object detection, R-CNN (Regions with CNN features) is the most straightforward [12] and also boasted an almost 50% improvement on the object detection challenge of the previous approaches. This work's authors propose a three-stage process:

1) Break the input image into regions that contains possible objects and extract them.
2) Identify features in each region using a CNN.
3) Classify each region with the usage of SVMs.

The use of SVMs reduces the number of object candidates and reduces the computational cost of the method. It is important to note that each input image should have the same pixel dimensions; usually, this is checked with some image adjustments before inserting them into the model. While it produced excellent results, the training encountered a number of challenges, and the approach was eventually outperformed by others.

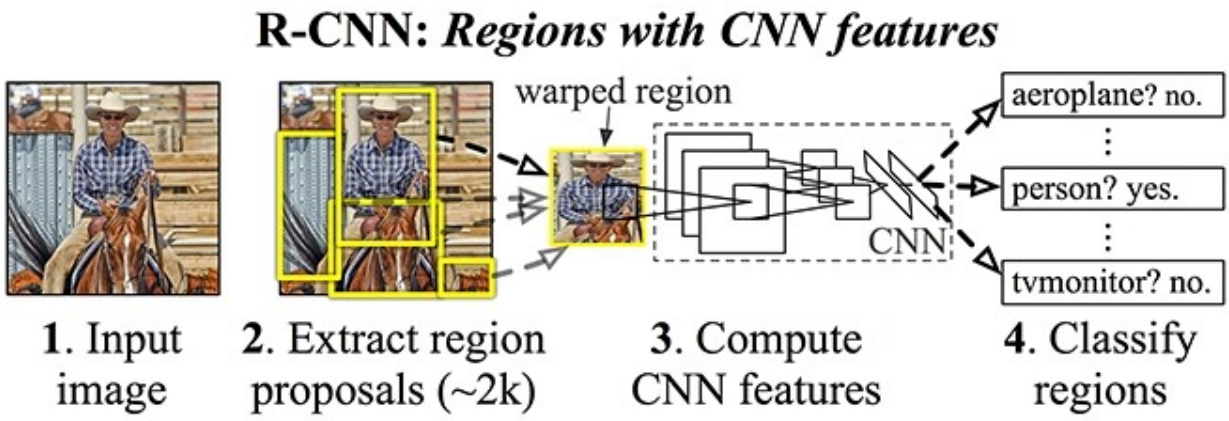

Figure 2: R-CNN (Regions with CNN features) (Image source [12]).

Fast R-CNN, an evolution of the aforementioned approach, was then proposed. Using Selective Search algorithm, it generates object proposals, but instead of using SVM classifiers that extract them all separately, it applied CNN to the entire image and then used Region of Interest (RoI) Pooling on the feature map with a final feed forward network for classification and regression. This approach was not only faster, but it also allowed the model to be end-to-end differentiable and easier to train due to the RoI Pooling layer and fully connected layers. The model's main disadvantage was that it still relied on a region proposal algorithm, which became a constraint when used for inference.

Following we have a more robust model which is based on Joseph Redmon's (YOLO) [13] model, that proposes a more global solution using a simple convolutional neural network. It

divides the image into regions and predicts bounding boxes and weighted probabilities for each region. Its global solution of using a single network evaluation makes it 1000x faster than R-CNN and 100x faster than Fast R-CNN. This approach is efficient and has outstanding results especially in COCO dataset [14] achieving one of the best mean average precisions (mAP, allowing for real-time object detection as well.

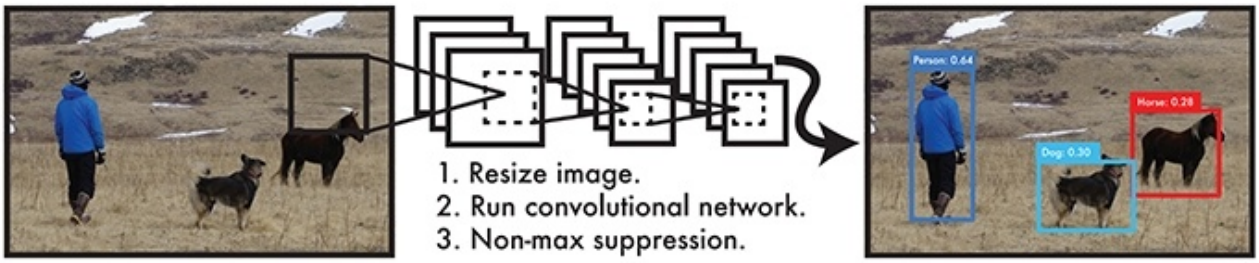

Figure 3: Joseph Redmon's (YOLO) model (Image source [13])

Finally, there are a notable approach, Single Shot Detector (SSD) [15] which takes on YOLO by using multiple sized convolutional feature maps achieving better results and speed. SSD divides the image into grids, with each grid cell responsible for detecting objects in that section of the image, rather than employing a typical sliding window technique. If no item is identified, the message "no object detected" is displayed. Anchor boxes are used in situations where several items are recognized in a single image. These are basic boxes that can identify several items in a picture and have various anchors set to them.

2.1.2 Identification of devices using RF technologies

Different RF technologies can be used to identify and recognize devices. On the one hand, RFID and NFC technologies are widely being used for identification purposes (for example, to manage access control directives or to detect/update good stocks in a warehouse based on proximity detection). On the other hand, UWB technology could be also used for identification purposes as it allows to determine the proximity/position to an object and, in addition, relevant identification data could be also shared through the wireless link. Further information about UWB technology is summarized in the next section.

2.2 Object positioning techniques

Object positioning is a key aspect for enabling ambient intelligence.

On the one hand, the capability to collect information about the position of mobile devices can be used for several aspects:

- Identifying features/properties of the object that can be used as a reference to infer what is happening in the scenario. The location information can be associated to a position, which can be absolute (e.g. UTM coordinates obtained from a GNSS system) or relative (e.g. 1D (distance), 2D (plane) or 3D coordinates in respect to a known reference). Furthermore, if the information includes a timestamp it can serve to locate the object referenced in time. Finally, if the object is an IoT connected device, other parameters such as the identity of the object might be communicated in each

update, making it possible to univocally associate the position with a well-known object.
- Deriving other features that might be interesting for ML/prediction purposes. Concretely, consecutive positioning updates could be used to infer aspects such as the speed and the trajectory of the object, e.g. an Automated Guided Vehicle (AGV) that moves around an industry site. This information, in conjunction with historical data, could be used to recognize mobility patterns, identify established routes, predict future movements or even detect abnormal behaviours.

On the other hand, the capacity to position an end user (e.g. a worker in an industry site) opens the door to advanced device recognition techniques. An application might be able to generate a virtual environment from the perspective of the user depicting all the objects, whose location is known, in a close proximity. This virtual environment could be linked with the real environment (user vision), allowing the user to recognize not only the objects that he/she can see, but also the objects that are not visible (for example, the location of an AGV that approaches in an aisle crossing) but are relevant for the user.

There exist different IoT-based technologies for location. Concretely, the project will distinguish between the ones described in the following subsections.

2.2.1 Global Navigation Satellite Systems (GNSS)

GNSS [16] technology has been widely adopted for geolocation in outdoor scenarios. It uses the information from different satellite constellations, such as GPS, Galileo or Glonass, to provide an absolute position. The localization is based on the calculation of the Time Difference of Arrival (TDOA) from the messages received from different satellites. The accuracy of the system depends on different aspects, such as the dilution of precision (geometry of the satellites), the number of satellites or the ionospheric conditions. In general, errors higher than 1.5 meters can be expected for autonomous GNSS systems. There exist different improvements or methods to optimize the precision of the location; for example, Real Time Kinematic (RTK) enhancements could reach precisions of tens of centimeters or less though they require communication to a base station or a server to share correction data. GNSS systems can be easily implemented in an IoT device with the integration of some hardware and they are already present in smartphones. The main drawbacks of these systems are their consumption and that they cannot be applied in indoor scenarios.

2.2.2 RF positioning

Wireless communications can also be used to provide positioning information. One of their advantages, compared to GNSS, is that they can be used in both, indoor and outdoor scenarios. In this respect, they are an interesting alternative for the implementation of indoor Real Time Location facilities. Furthermore, these wireless technologies can not only be used for positioning or tracking, but also to identify IoT devices or detect the presence of objects. Finally, these technologies can be also used to transmit data taking profit of the infrastructure deployed to provide location.

Table 4 summarizes the main aspects of the most common wireless technologies used for implementing real time location systems. As it can be seen, the obtained precision will depend on the technology used. Aspects such as power consumption and the range of the

technology are also relevant since they will impact in the durability of the battery of the IoT devices or the dimensions and cost of the needed infrastructure. The following subsections describe some of these technologies.

Table 4: Comparison of RF positioning technologies

Technology	Accuracy	Range (m)	Power consumption	Identification	Presence Detection
Wi-Fi	m	1-50	High	Yes	Yes
Bluetooth	m	1-20	Low	Yes	Yes
Ultra Wide Band (UWB)	cm-dm	1-50	Low (tags)	Yes	Yes
RFID	dm-m	Few meters	Very low (tags)	Yes	Yes

2.2.2.1 ISM band technologies

Wi-Fi ISM band technology can be used for positioning in both, indoor and outdoor, environments. Received Signal Strength Indicator (RSSI) levels received from existing Access Points (APs) can be used for this aim. This technique is, for example, used by Google to complement AGPS positioning in smartphones when a user enters and indoor environment. In a similar way, some companies deploy on-purpose APs to provide location services in indoor environments. The solution is cost-effective since only APs need to be mounted that can be used also for communication purposes, but the accuracy relies on the amount of APs deployed and it is in the range of meters. The IEEE802.11mc [17] amendment defines a positioning technique based in Time of Flight (ToF) calculation and the use of precise timestamps referred as Fine Time Measurement (FTM). Some companies, such as Aruba Networks, deploy this solution, which can achieve an accuracy between 1 and 2 meters in Line of Sight (LoS) conditions.

Bluetooth Low Energy (BLE) is another ISM band technology which is being applied in indoor environments. Initially, BLE location was based in the deployment of beacon devices and RSSI measurements. The accuracy obtained in complex indoor environments was in the order of few meters. BLE 5.1 [18] revision introduced location improvements based on Angle of Arrival (AoA) techniques. These mechanisms allow to reach accuracies between 0.5 and 1 meter depending on the scenario.

2.2.2.2 Ultra Wide Band technology

Ultra-Wide Band (UWB) technology is based on the use of spread spectrum. The application of UWB to location is defined in the IEEE802.15.4a [19] specification. Several improvements have been proposed in the IEEE802.15.4-2011 [20] and IEEE802.15.4z [21] amendments. Some of the features that make UWB interesting for its application in positioning solutions are that it is robust to interferences and that it allows to achieve accuracies lower than 30 centimetres

in LoS conditions. Furthermore, the technology can also be used for communication purposes. The technology can operate in the 3.1GHz to 9GHz band and is license-free. In order to assure that UWB does not disturb the rest of licensed services that coexist in the same frequency bands, its usage is regulated [22]. Concretely, the positioning devices need to respect a maximum value of the mean power spectral density (e.i.r.p) of -41.3 dBm/MHz (in the best case) and the defined duty cycle restrictions, which depend on the scenario (indoor or outdoor) and the type of devices and infrastructure (fixed or mobile). As an example, in indoor environments, devices have duty cycle limitations of 5% per second and 1.5% per minute. UWB positioning is based on the use of two different types of elements: 1) the anchors, which are the fixed infrastructure devices and act as the reference (well-known) positions; and 2) the tags, which are the devices that will be localized. Aspects such as the geometry of the infrastructure setup, the LoS conditions and the distance to anchors can impact the positioning. Several mechanisms can be used to get positioning information with UWB technology: 1) TWR (Two-way ranging), which is based in ToF measurements to determine the distance from the tag to the different anchors. The position is obtained through trilateration; 2) TDOA (Time difference of Arrival), which is based on the same approach as GNSS systems. This mechanism requires a more complex implementation since a precise time synchronization between the anchors must be assured. Synchronization can be performed in a wired or wireless manner; 3) AoA (Angle of Arrival). In this case, the anchors incorporate at least two antennas. Finally, UWB is entering the smartphone industry (Apple [23], Samsung [24]); it is to be expected that in the following years it will gain a higher penetration in the market and it will be adopted by other smartphone manufacturers. UWB technology has raised a special interest from the industry and the automotive sector [25]. From its features and accuracy performance, it has been considered an interesting candidate for the industrial use case in the IoT-NGIN project.

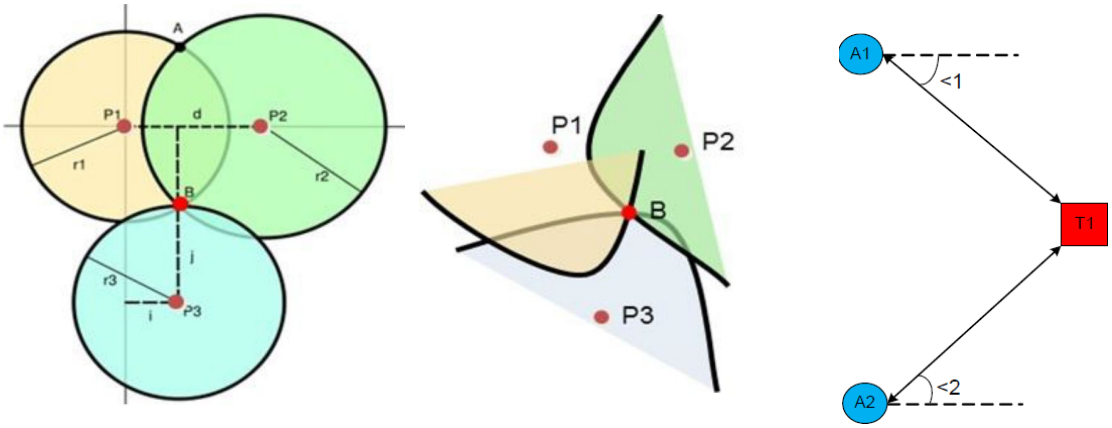

Figure 4: UWB location methods. ToF, TDOA and AoA [26].

2.2.3 Visual Light Positioning (VLP)

Visual Communications are considered an interesting technology for data transmissions. In addition to this, they can be applied to implement precise positioning solutions, referred as Visual Light Positioning. Using conventional LED devices, it is possible to modulate them to transmit information. This information can be captured using an embedded equipment or a

smartphone with a camera (Optical Camera Communications) [27]. Provided that the camera is able to capture at least four lights at the same time, as shown in the following figure, image processing mechanisms can be applied to identify the different light sources and calculate the position of the camera in relation to the lights. With this approach i2CAT has achieved to reach accuracies of less than 10 centimeters and few degrees in angle. It has also been considered an interested candidate for the industrial use case in the IoT-NGIN project.

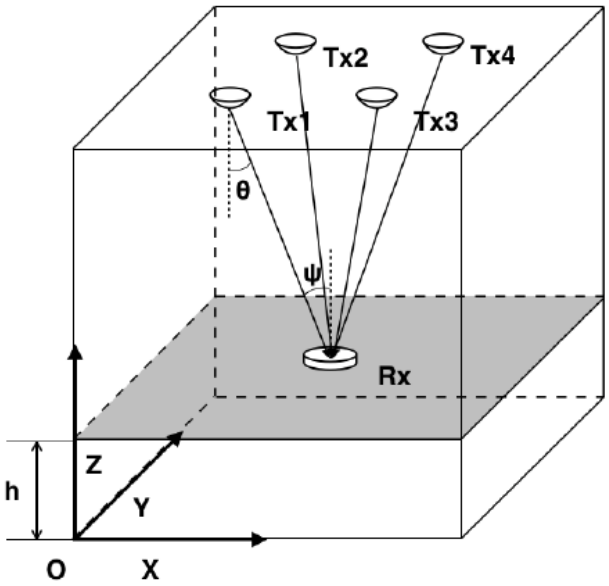

Figure 5: Architecture of a VLP indoor positioning system [28]

2.2.4 Homography

Homography is a mathematical relationship of any two images of the same planar surface in the space. This relationship has very practical applications, for example to get the real position of an object that appears in an image. It can relate the pixel position of an object which is on a known plane in the image with real position of that object. For example, knowing the pixel position (x,y) of the shoe of a person standing on the pavement in an image, it is possible to know the real GPS coordinates of that person (x',y').

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \mathbf{H} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

where H is the homography 3x3 matrix.

In Figure 6, an example indicating how this homography transformation would be applied to a full image is presented.

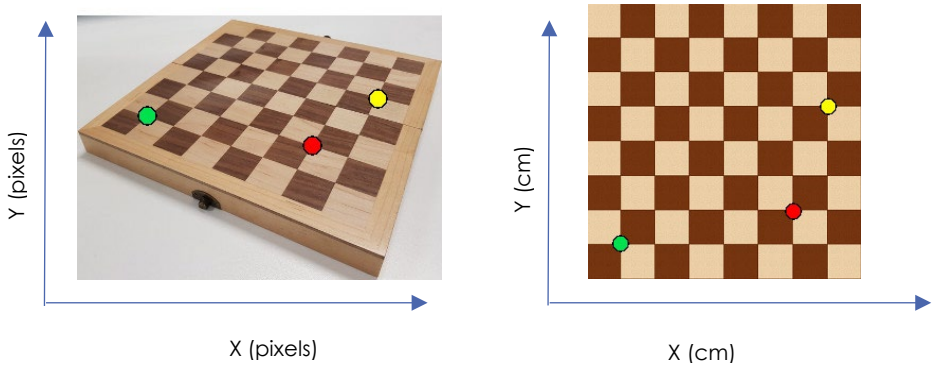

Figure 6: Example of an homography transformation of an image

2.3 Device authentication techniques

Authentication and Authorisation (Access Control) are core security considerations for the vast majority of networked applications and services. Explicitly inferring who is the caller of a service and what are the permissions that her identity is entitled to is essential to ensure privacy and data protection from external attacks. Indeed, Gartner, in their 2020 Gartner Hype Cycle for Identity and Access Management Technologies report [29], label IoT authentication as a "high benefit". Similarly, the European Union Agency For Network And Information Security (ENISA) in their 2017 published *Good practices for IoT and Smart Infrastructures Tool* [30] as a continuation of their *Baseline Security Recommendations for IoT in the context of Critical Information Infrastructure* report [31], enlist Authentication and Authorisation essential, baseline security factors to address when trying to secure an IoT system. At the same time, the same holds for Trust and Identity Management, effectively binding the identity of the service users to their trust status. Specifically, ENISA in [31] considers that the following technical measures related to authentication and authorisation, need to be taken into account in order to diminish the vulnerabilities of IoT:

- Authentication
  - GP-TM-21: Design the authentication and authorisation schemes (unique per device) based on the system-level threat models.
  - GP-TM-22: Ensure that default passwords and even default usernames are changed during the initial setup, and that weak, null or blank passwords are not allowed.
  - GP-TM-23: Authentication mechanisms must use strong passwords or personal identification numbers (PINs) and should consider using two-factor authentication (2FA) or multi-factor authentication (MFA) like Smartphones, Biometrics, etc., on top of certificates.
  - GP-TM-24: Authentication credentials shall be salted, hashed and/or encrypted.
  - GP-TM-25: Protect against 'brute force' and/or other abusive login attempts. This protection should also consider keys stored in devices.
  - GP-TM-26: Ensure password recovery or reset mechanism is robust and does not supply an attacker with information indicating a valid account. The same applies to key update and recovery mechanisms.

- Authorisation
  - GP-TM-27: Limit the actions allowed for a given system by implementing fine-grained authorisation mechanisms and using the principle of least privilege (POLP); applications must operate at the lowest privilege level possible.
  - GP-TM-28: Device firmware should be designed to isolate privileged code, processes and data from portions of the firmware that do not need access to them. Device hardware should provide isolation concepts to prevent unprivileged users from accessing security-sensitive code.

When it comes to device authentication, though, being lightly interactive, things tend to be different from baseline considerations, since authentication credentials should be bound to the device identity and characteristics rather than traditional credentials like username/password sets. There are several techniques and approaches that can be used to achieve strong authentication when it comes to IoT devices, such as [32]:

- One-way authentication (only the device authenticates itself against an authentication service)
- Two-way authentication (both the device and the authentication service authenticate to each other)
- Three-way authentication (a trusted third party authenticates both the device and the authentication service and catalyses their mutual authentication)
- Distributed authentication (a distributed algorithm is used among the different entities to authenticate)
- Centralized authentication (a trusted third party distributes and manages the authentication certificates).

In any case, a machine identity management is needed to build and manage trust against the identity of the devices that interact with a given IoT set of services. Usually, this is accomplished by synchronizing the identity of a device to a cryptographic key that is unique for each device, such as employing Trusted Platform Modules (TPMs) and X.509 certificates.

The following paragraphs briefly overview some of the most usually employed methods and technologies to ensure reliable trust management.

2.3.1 X.509 Certificates

The X.509 certificates represent digital identities standardized under the IETF RFC 5280 [33] and are generally based on the certificate chain of trust model. There are several ways that the certificates may be authenticated, the most prominent ones being:

- Thumbprints (hexadecimal string unambiguously identifies a certificate generated by running a thumbprint algorithm against the certificate)
- CA authentication based on a full chain (effectively implying the identification of a trusted signing authority having signed the certificate chain)

The advantages of using X.509 certificates are that they uncover flexible identity management capabilities and are widely supported by existing systems and services.

Hardware Security Modules (HSM) and TPMs may be employed to securely store secrets X.509 certificates (or other relevant unique device identifiers), so that supply chain attacks may be avoided.

2.3.2 Trusted Platform Modules

TPMs are modules that enable both securely storing digital identity files (e.g. X.509 certificates) and generating relevant authentication elements (e.g. asymmetric key pairs for devices) that can be used to authenticate the devices (and, possibly, to encrypt the communication traffic). In general, TPMs are available in multiple modalities offering different levels of security guarantees, e.g. in the form of discrete hardware devices, embedded hardware equipment or even in the form of firmware/software. While typical TPMs provide several cryptographic capabilities, such as the ones already mentioned, they may enable other key features relevant to IoT authentication and authorization such as securing the devices booting and establishing the root of trust.

The main advantages of TPMs may be summarized in the hardware/software attestation services they may offer, being more secure than token-based shared access signature (SAS) symmetric key attestation, however, they are not always available in existing hardware and are generally difficult to use and manage.

2.3.3 Symmetric Keys

Symmetric keys are secrets known to both the devices and the associated authentication services and can be used to encrypt and decrypt messages between the two parties. Since they constitute delicate information, symmetric keys are usually stored in TPMs or HSMs.

The symmetric key attestation against the authentication services is carried out using the same security (SAS) tokens to identify the devices; SAS tokens come with signatures generated using symmetric keys. Then, the signatures may be recreated by the authentication services to verify whether the token is legitimate. When the device certifies with an individual enrolment, the device uses the symmetric key defined in the individual enrolment entry to create a hashed signature for the SAS token.

The advantages are summarized in the fact that they are easy to use and they should work in practically all IoT devices, however they are much less secure than the rest of the authentication methods (e.g. the X.509 certificates).

2.4 Augmented/Mixed Reality

Augmented Reality (AR) and Mixed Reality (MR) bring the promise of maximizing the potential benefits of IoT. They can provide effective visualization and interaction methods that are not unlocked to the limits of fixed screens, while still keeping the focus on the real scenarios.

IoT-NGIN will not only compile and categorize the existing AR/MR frameworks for IoT interaction, together with state-of-the-art contributions and available AR/MR headsets, in a unique repository, but most interestingly the project will provide a set of APIs and libraries to be able to interact with other components of the IoT-NGIN platform, like the device indexing, recognition, and authentication modules.

In conjunction, the newly developed libraries, APIs and algorithms will enable AR presentation and interaction methods that are not currently supported in existing frameworks

(e.g. presentation of data from IoT sensors detected by external video cameras, or via non-visible recognition methods), with the goal of effectively supporting the envisioned use cases.

Extra information about the planned AR frameworks and features to be used are provided in Section 3, while more detailed documentation and the progress on these tasks will be provided in subsequent iterations of WP4 deliverables.

2.5 5G

Applications of ambient intelligence often have technological requirements that cannot be met using legacy networks. These applications usually require extremely high bandwidth and ultra-low latencies as shown in Figure 7.

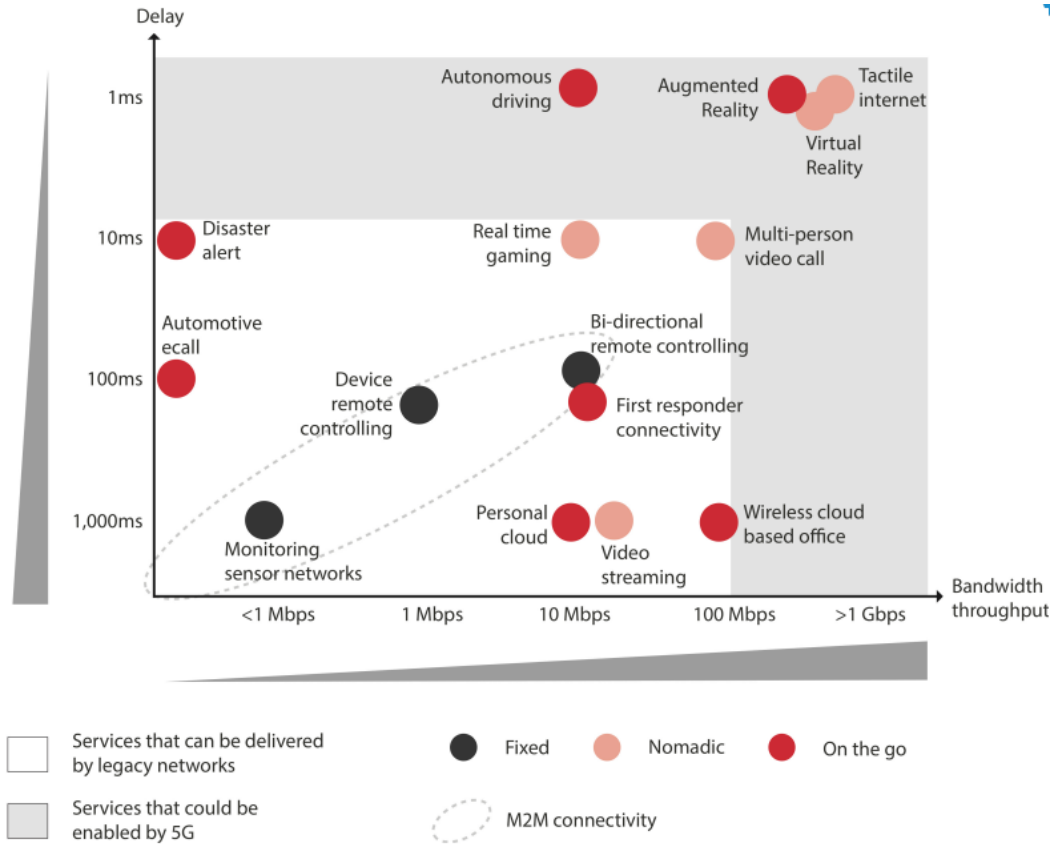

Figure 7: Delay vs Bandwidth requirements for multiple applications [34].

It is clear that ambient intelligence applications, such as Augmented Reality, Tactile Internet and other similar concepts cannot be realized using legacy networks. Both higher bandwidth and much lower latencies are required for example in autonomous driving. One of the most promising network technologies to overcome these obstacles is 5G [35].

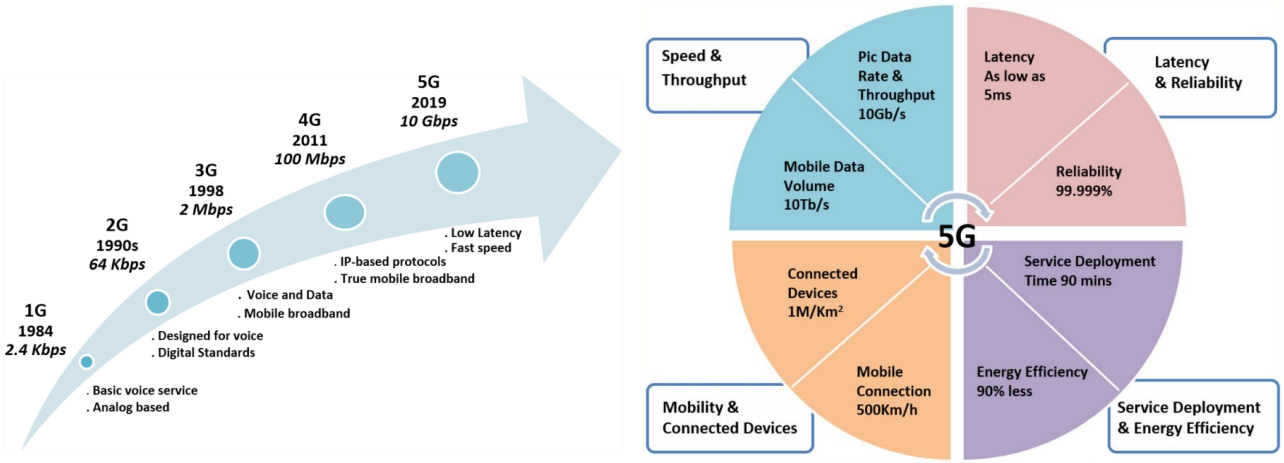

Figure 8: (left) Comparison of 5G bandwidth with other networks, (right) overall 5G technical capabilities [36].

Compared to previous network technologies, 5G can provide up to 10 Gbps bandwidth, 2 orders of magnitude higher than its predecessor 4G (Figure 8 (left)). Additionally, 5G possesses many more impressive features, such as impressive reliability (99.999%), ultra-low latency (5 ms), improved energy efficiency and a remarkable number of connected devices (1 M/km^2 & up to 500 km/h moving speeds), as shown in (Figure 8 (right)).

Several IoT applications have already been running today under 4G network such as massive and critical IoT; however, they are facing several difficulties. Massive IoT, aka massive Machine-Type Communications (mMTC), defines applications with lots of endpoints that continuously provide small amounts of data, mostly infrequently and to even remote locations. It mostly involves low-cost and low-energy applications with small data volumes in huge numbers that are regularly sent to the cloud. IoT sensors from billions of devices, objects, and machines communicate with one another, something that requires scalability and versatility. Ability to accommodate a massive number of devices is a core requirement for massive IoT along with network efficiency in order to connect potentially millions of devices. Massive IoT also requires long battery life and a wide coverage area. This is where 5G comes into the picture as Low Power Wide Area Networks (LPWANs) can best meet the needs of massive IoT devices [37].

On the other hand, another approach exists called Critical IoT, which involves fewer number of devices that handle massive levels of data. More specifically, critical IoT applications are described as Ultra-Reliable Low Latency Communications (URLLC). It represents the longer-term vision for high-bandwidth and low-latency applications and devices, going beyond just data collection and into more complex scenarios. Thus, it is easy to see how 5G will factor into the equation and allow critical IoT to become more of a reality going forward. Critical IoT will also require performance that can withstand harsh and remote environments, support for new manufacturing processes, scalability in order to support large-scale networks with thousands of controllers, robots, and machines, and security to protect end-point devices and networks against threats and attacks. A comparison between Massive IoT and Critical IoT is shown in Figure 9, whilst Figure 10 provides an overview of the wireless connectivity requirements for future factories implementing such technologies [38].

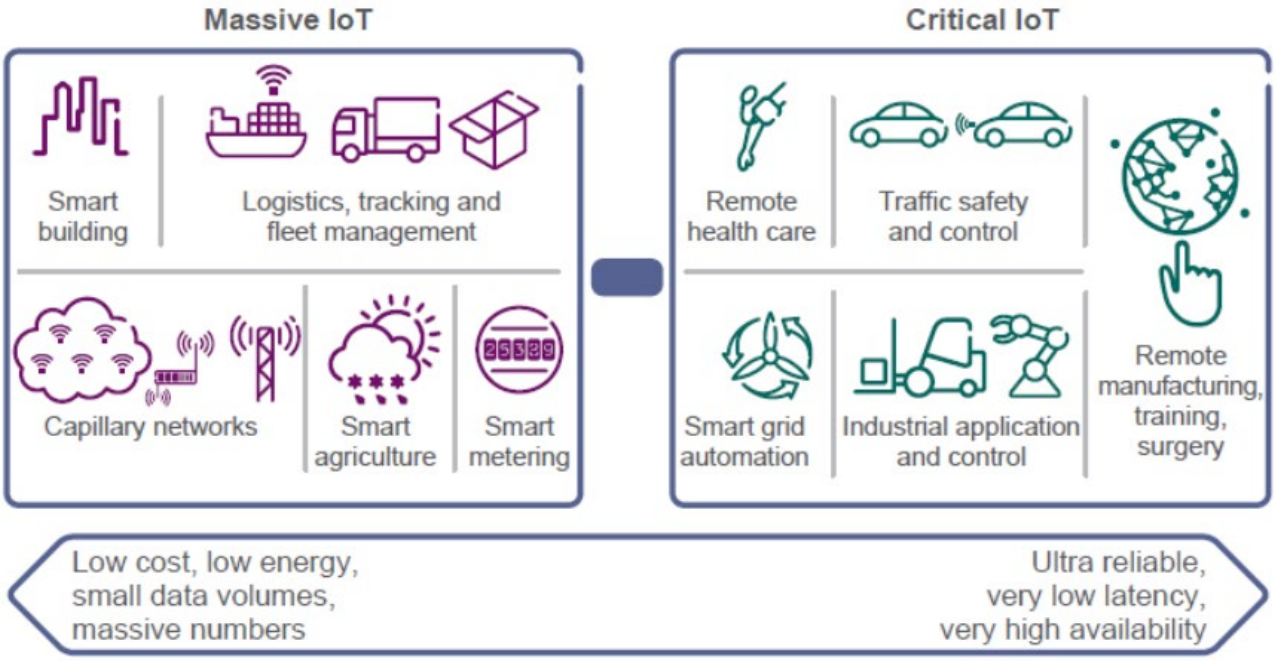

Figure 9: Massive IoT vs Critical IoT [37].

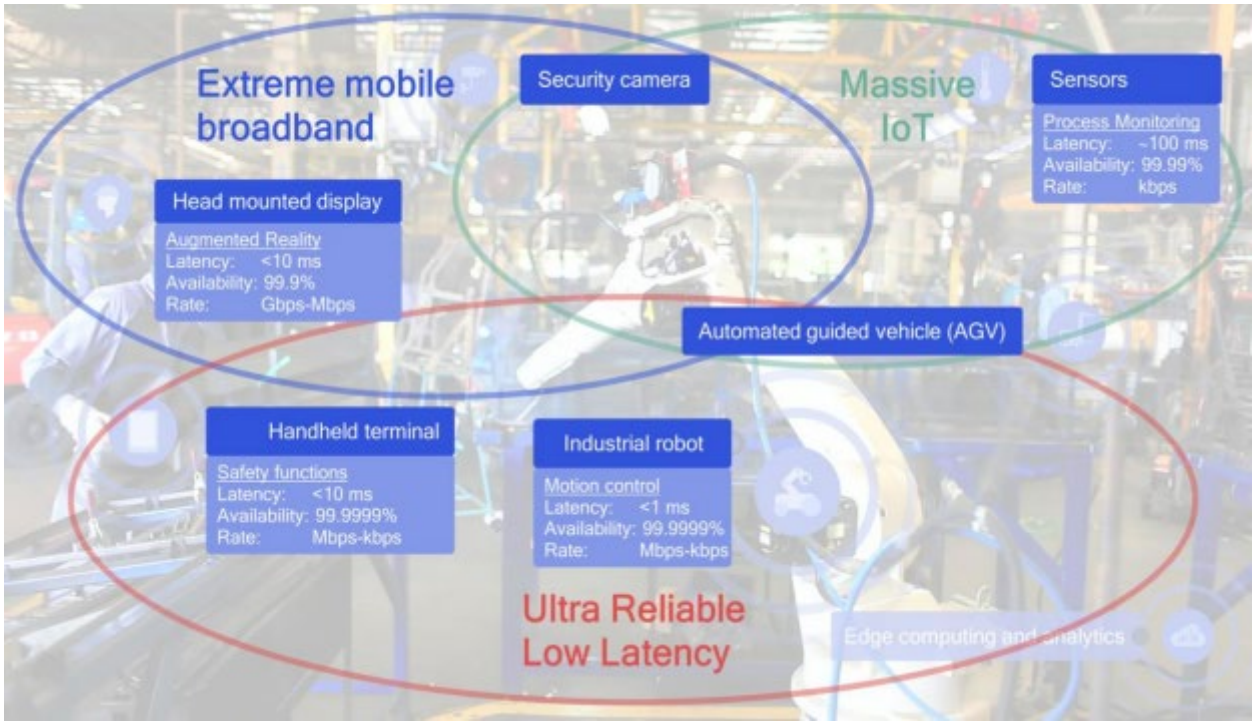

Figure 10: Overview of wireless connectivity requirements [38].

2.6 Edge computing

Cloud Computing has become a means of consolidating computing functions, storage and network management centrally but as the IoT and new telecommunication technologies arise, architectures based exclusively on a centralized cloud many difficulties

because they cannot meet the requirements of the IoT paradigm nor can they support the massive volume of data these applications have to handle.

The need of real-time experiences adds more burden to the networks, resulting in increased latency as well as storage and high bandwidth costs security and integrity of data coming from devices running in the periphery of the network.

Other well-known problems are the:

- IoT devices heterogeneity (sensors, actuators, smartphones, tablets, smart bracelets, laptops, etc.) with limited storage and processing resources;
- Uninterrupted, real-time service requests and responses.

These characteristics have been the motivation behind research which proposed Edge Computing as an alternative means of processing and filtering big data at the edge of the network before transmitting it to the cloud; the benefits of this include reduced bandwidth costs, storage and energy consumption. Therefore, by integrating an IoT-based Edge Computing architecture, applications and services provide users with faster, more efficient and secure responses.

Speaking of definition, we can start to contextualize the concept of Edge computing as follows.

Edge computing refers to an approach and an architecture designed to move computing and storage out of the remote cloud (public or private) and closer to the source of data. Edge computing brings application hosting from centralized data centres down to the network edge, closer to the end user as well as to the data generated by applications.

In addition to the broad definition posed above edge computing be also described by means of jargon related to the approach of edge computing design.

• Fog computing: Fog computing refers to an architecture of cloud services that span from central data centre cloud systems to near-edge and far-edge devices. The fog represents a single abstraction of a geographically disparate set of cloud and edge computers to behave and act as a single entity.

• Multi-access edge computing (MEC): Previously called mobile edge computing, MEC enables low-latency, high-bandwidth, and real-time applications to exist and be distributed at the edge of larger networks. MEC is defined by the European Telecommunications Standards Institute (ETSI) [39] and involves allowing developers to run applications within a telecommunication carrier's radio access network (RAN). Typically, the RAN physically exists with the radio network controller at a cellular base station. MEC could allow low-latency video streaming or cloud-based gaming.

• Cloudlets: A cloudlet is a small-scale cloud data centre. It may be perceived as a "cloud-in-a-box". In other words, it is a device to support resource-intensive use cases in client-server types of applications. This is similar to the MEC concept to facilitate better response times and lower latency but it does not necessarily associate with a telecommunication or carrier infrastructure.

In the following section we will refer to edge computing as MEC.

MEC is a natural development in the evolution of mobile base stations and the convergence of IT and telecommunications networking. Based on a virtualized platform, MEC has been

early recognized by the European 5G PPP research body as one of the key emerging technologies for 5G networks [40], together with Network Function Virtualization (NFV) and Software Defined Networking (SDN) [41]. In addition to defining more advanced air interface technologies, 5G networks leverage more programmable approaches to software networking and use IT virtualization technology extensively within the telecommunications infrastructure, functions, and applications. MEC is therefore an enabler of the evolution towards 5G, since it helps advance the transformation of the mobile broadband network into a programmable world and contributes to satisfying the demanding requirements of 5G from the point of view of throughput, latency, scalability, and automation.

MEC is mapped in a virtualized platform with an approach that has many similarities with the NFV. In fact, while NFV is focused on network functions, the MEC framework enables applications running at the edge of the network. The infrastructure that hosts MEC and NFV has many similarities and has the scope to enable the network operators to benefit as much as possible from their investment enabling the reusage of the infrastructure and infrastructure management of NFV to the largest extent possible.

It, it is useful to underline that the hosting of both Virtual Network Functions (VNFs) and MEC applications on the same platform results in low latency, proximity to the end-user, high bandwidth, and real-time insight into radio network information and location awareness.

This can be translated into value and can create opportunities for both mobile operators, and application and content providers, enabling them to play complementary and profitable roles within their respective business models and allowing them to better monetize the mobile broadband experience.

2.6.1 MEC and 5G

MEC standards make no assumptions on the underlying radio infrastructure, which makes MEC a highly flexible element in the 5G communications networks. This characteristic enables new levels of adaptability to different scenarios. Therefore, Service Providers (SPs) can use MEC without being forced to wait for full establishment of the 5G standard.

This approach allows software providers to offer third parties a cost-effective way to trial their applications. In fact, using edge cloud, an application can be hosted in a virtual retail space, test the revenue return, and scale-up to allow a smooth transition into the 5G network rollout.

Another focus area to improve the presence of 5G networks is about re-using the existing deployed systems in the process. Due to the MEC's virtualized characteristics, it is possible with low expenditure to monitor performance and resource needs of an application which, in turn, enables more accurate pricing for operators towards application providers for hosting the applications.

The common feature set of providing much-improved capabilities at the edge of the network, as well as improved intelligence about resources needed at the edge makes it easier to start the deployment in 5G test sites.

3 IoT Ambient Intelligence requirements from the Living Labs

This section analyses the four living labs of the IoT-NGIN project and their uses cases in order to gather some requirements for the four software modules and repositories that are going to be developed in WP4 (these modules are described in Section 4). The analysis of the Living Labs (LLs) use cases is primary based on the deliverable "D1.1 Definition analysis of use cases and GDPR Compliance" and a series of surveys and interviews that have been answered by the use case leaders and participants.

Table 5 shows which living labs will make use of the software modules that will be developed in WP4. The first column of the table lists the four living labs of the IoT-NGIN project and the next four columns correspond to the four software modules. The "tick mark" indicates that the module will be used in the LL, and the "question mark" indicates a potential use that still needs to be agreed.

The following sections 3.1, 3.2, 3.3 and 3.4 are specific to each living lab. All of them start identifying the preconditions of the use cases of the living lab that are relevant to Ambient Intelligence applications. These preconditions are also summarized and related to the Ambient Intelligence modules developed in WP4 in a mapping table. Table 5 already shows the reference of each mapping table and the uses cases of each LL that require the Ambient Intelligence tools.

Table 5: Summary of the living lab relevance on the WP4 requirements.

Living Lab	Device Discovery	Device Indexing	Device Access Control	AR/MR tools	Mapping table	Use cases
Human-Centred Twin Smart Cities		✓	?		Table 7	#1
Smart Agriculture IoT	✓	✓	✓	✓	Table 10	#4 #5
Employee Friendly Industry 4.0	✓	✓	?	✓	Table 15	#6 #7 #8
Smart Energy	✓	✓	✓	✓	Table 20	#10

Not all of the use cases of the LL will require Ambient Intelligence applications and make use of the software modules that will be developed in WP4. Concretely, at this stage of the project UC #2 and #3 have not identified Ambient Intelligence requirements. The UC that will require WP4 modules are listed in Table 6. The table also includes for each specific application of the UC the Ambient Intelligence modules that will be needed.

The requirement of the UC for the Ambient Intelligence services are also detailed in the following sections (3.1, 3.2, 3.3 and 3.4). All of these requirements are organized in tabulated

tables for each use case. Finally, there are some tables including some user stories that capture the requirements for the perspective of the end user of each UC.

Table 6: Use cases requiring WP4 software modules

Use case	Ambient Intelligence Application	Requirements table	User story
UC#1 Traffic Flow Prediction & Parking prediction	Traffic Measurement System stations indexing	Table 8	Table 9
UC#4 Crop diseases prediction, smart irrigation and precision aerial spraying	IoT devices recognition, and indexing, Device Access Control, AR interaction	Table 11	Table 13
UC#5 Sensor aided crop harvesting	IoT devices recognition, and indexing, Device Access Control	Table 12	Table 14
UC#6 Human-centred safety in a self-aware indoor factory environment	AGV – Humans collisions prevention	Table 16	Table 19
UC#7 Human-centred augmented reality assisted build-to-order assembly	AR assistance on assembly steps	Table 17	
UC#8 Digital powertrain and condition monitoring	Access control to the powertrain	Table 18	
UC#10 Driver-friendly dispatchable EV charging	Charging station – AR interaction	Table 21	Table 22

3.1 Human-Centred Twin Smart Cities Living Lab

In this section, one relevant use case (UC #1) under the Human-Centred Twin Smart Cities is analysed towards identifying their requirements from the IoT devices discovery, indexing, access control and AR/MR interaction components. The other 2 use cases of the living lab (UC #2 and UC #3) do not require any IoT-NGIN component developed in WP4. The UC analysis includes preconditions of the use case from the ambient intelligence perspective, mapping of those preconditions to Ambient Intelligent tools, use case data analysis and use case Ambient Intelligence techniques analysis.

3.1.1 Preconditions per application

The preconditions identified as relevant for the Ambient Intelligence processes for the UC#1 "Traffic Flow Prediction & Parking prediction" are identified as follows.

1. The use case aims to predict the traffic flow and parking availability to decrease traffic jams and bottlenecks.
2. These predictions will be based on data gathered by different sensors and from external databases.
3. Some of these sensors are contained in **the Traffic Measurement System (TMS)** stations which collect data of traffic volume and average speed.
4. The TMS stations are **IoT Devices** that are installed in different points of a city Helsinki.
5. These TMS stations need to be properly **indexed** in the IoT-NGIN platform together with their ID, position and status to make them visible in a digital twin of the city.
6. The access to the data collected by the TMS stations might be controlled by an **access control** method.

3.1.2 Mapping to IoT-NGIN Ambient Intelligence tools

The preconditions collected in the previous section are mapped to IoT-NGIN's Ambient Intelligence platform in Table 7.

Table 7: Mapping of Human-Centred Twin Smart Cities Living Lab to IoT-NGIN Ambient Intelligence platform

Device discovery	Device indexing	Access control	AR/MR tools
Not needed	Indexing of TMS stations (ID, position, status)	*Potentially needed* For gaining access to the TMS stations	*Not needed*

3.1.3 Use Case Requirements analysis

The following table describes, from a generic point of view, the features of the data produced in the Human-Centred Twin Smart Cities Living Lab that is relevant for the development of the Ambient Intelligence tools.

Table 8: Requirements analysis for Use case #1 "Traffic Flow Prediction & Parking prediction".

Use Case	#1 Traffic Flow Prediction & Parking prediction
Ambient Intelligence Application	TMS stations indexing

Ambient Intelligence Service		
Device discovery	**Device recognized**	TMS station
	Registration method	Manual
	Recognition method	Manual
Device Indexing	**Data registered**	TMS stations ID, position [x,y] and status
	Data Type(s)	JSON files
	Database type	TBD (part of IoT-NGIN)
	Data size	TBD (Low)
Device Access Control	**User rights**	TBD
	Spatial proximity	TBD
Actuation – AR interaction	**Actuation**	Not needed
	AR interface	Not needed
	AR software	Not needed

3.1.4 User Story

The scope and use of the Ambient Intelligence services in the use cases of the Smart cities LL are revealed via the user story of in Table 9. The table describes simple statements on behalf of the Actor as to their desired use and scope of the use case.

Table 9: User story table for the "Traffic Flow Prediction & Parking prediction" use case.

As a	I want to	So
Solution provider	register a new TMS station	I can provide the data to users
App developer	know where the TMS stations are located	I can show them in a digital twin

3.2 Smart Agriculture IoT Living Lab

In this section, the two use cases under the Smart Agriculture Living Lab are analysed towards their Ambient Intelligence needs and specifically towards identifying their requirements from the IoT devices discovery, recognition, and indexing components. The UC analysis includes preconditions of each use case from the ambient intelligence perspective, mapping of those preconditions to Ambient Intelligent tools, use case data analysis and use case Ambient Intelligence techniques analysis.

3.2.1 Preconditions per application

The preconditions identified as relevant for the Ambient Intelligence processes for the "Crop diseases prediction, Smart irrigation and precision aerial spraying" use case are identified as follows.

1. Crop diseases' prediction is based on measurements, acquired via SYN **SynField** [42] precision agriculture IoT nodes, integrating a variety of sensor modules, as well as images acquired via **drones** flying over the field.
2. The SynField devices are desired to be identified in the context of tactile internet, using a **Quick Response (QR) code** attached.
3. The SynField devices may be recognized via computer vision.
4. The devices that are desired to be identified in the context of tactile internet are fixed.
5. Drones and SynField devices will be able to be registered in the IoT-NGIN platform through a **publish-subscribe mechanism**.
6. Drones and SynField devices will be able to be **localized using Global Positioning System (GPS) signals**.
7. Collected data and results from data analysis will be **visualized through AR/MR tools**, after recognition of SynField devices, assuming that the user holds the **appropriate access rights**.
8. **Actuation** controls of SynField devices will be possible via **AR tools**, the user holds the appropriate access rights.

In addition, the preconditions assumed for the "Sensor aided crop harvesting" use case, in relation to Ambient Intelligence, are the following.

1. **Automated Guided Land Vehicle (AGLV) serving as carrier machines** of crates, assisting the harvesting process, will be capable of calculating and following an appropriate trajectory from the harvesting location to the loading points.
2. **AGLV** will be able to **locate and avoid workers** (for safety reasons) **and trees** (for operating reasons) using novel computer vision (V-SLAM) techniques.
3. The crates identification at the **loading points** will be based on **code scanners** located there, as well as **QR codes attached to the crates.**
4. The AGLV will carry a **QR code**, which will allow the user to identify the device, assuming that the user holds the **appropriate access rights**.

3.2.2 Mapping to IoT-NGIN Ambient Intelligence tools

The Smart Agriculture LL will exploit ambient intelligence modules in both use cases. QR based device identification will be used to identify the SynField devices, drones, AGLV and crates. Moreover, device indexing will be exploited to perform Create Read Update Delete (CRUD) operations over the LL's devices repository. Both use cases will impose access control on the IoT devices, both for accessing device or monitoring data or for performing actuation controls. To this, different access rights would apply to different roles or types of users. In addition, AR functionality will be exploited in the "Crop diseases prediction, Smart irrigation and precision aerial spraying" use case to visualize information related to SynField devices. AR functionality can be also exploited to apply actuation control on the SynField devices. The exploitation of ambient intelligence modules in the Smart Agriculture Living Lab is summarized in Table 10.

Table 10: Mapping of Smart Agriculture IoT Living Lab to IoT-NGIN Ambient Intelligence platform.

Device discovery	Device indexing	Access control	AR/MR tools
RFID & QR based For drones, SynField devices and AGLVs	For drones, SynField devices and AGLVs	For both monitoring and actuation	For visualization, management tasks and actuation controls of SynField devices

3.2.3 Use Case Requirements analysis

The following tables describe specific requirements of the use cases in the Smart Agriculture IoT Living Lab from the Ambient Intelligence tools.

Table 11: Requirements analysis for Use case #4 "Crop diseases prediction, smart irrigation and precision aerial spraying".

Use Case		Crop diseases prediction, smart irrigation and precision aerial spraying
Ambient Intelligence Application		IoT devices recognition, and indexing
Ambient Intelligence Service		
Device discovery	**Device recognized**	SynField devices
	Registration method	Manual/programmatic
	Recognition method	Computer vision & QR based
Device Indexing	**Data registered**	TBD, including: Device type, ID, timestamp
	Data Type(s)	JSON files
	Database type	TBD (part of IoT-NGIN)
	Data size	Depending on the number of devices and how often they are detected <50 Mb per day
Device Access Control	**User rights**	Yes – for reading data and applying actuation (e.g. irrigation)
	Spatial proximity	N/A
Actuation – AR interaction	**Actuation**	Yes, for information visualization and activating the irrigation using AR gestures
	AR interface	Yes
	AR software	TBD (e.g. Unity [43])

Table 12: Requirements analysis for Use case #5 "Sensor aided crop harvesting" use case

Use Case		Sensor aided crop harvesting
Ambient Intelligence Application		IoT devices recognition, and indexing
Ambient Intelligence Service		
Device discovery	**Device recognized**	AGLVs
	Registration method	Manual/programmatic
	Recognition method	QR based
Device Indexing	**Data registered**	TBD, including: Device type, ID, timestamp
	Data Type(s)	JSON files
	Database type	TBD (part of IoT-NGIN)
	Data size	TBD (Low)
Device Access Control	**User rights**	Yes – for reading/visualizing data
	Spatial proximity	N/A
Actuation – AR interaction	**Actuation**	No
	AR interface	No
	AR software	N/A

3.2.4 User Story

The scope and use of the Ambient Intelligence services in both use cases of the Smart Agriculture LL are revealed via the user story of each use case. Table 13 and Table 14 describe simple statements on behalf of the Actor as to their desired use and scope of each use case, respectively.

Table 13: User story table for the "Crop diseases prediction, smart irrigation and precision aerial spraying" use case

As a	I want to	So
Smart Farmer	Access monitoring data collected by SynField and drone devices.	I can have a good view of my orchard state.
Smart Farmer	Access plant disease predictions	I can take measures to prevent them
Smart Farmer	Easily interact with my SynField devices via my mobile phone	I can easily trigger visualization and actuation tasks
Smart Farmer	Securely interact with my SynField devices via my mobile phone	Only I, as an authorized user with the appropriate access rights, can insert commands to my SynField devices.

Table 14: User story table for the "Sensor aided crop harvesting" use case

As a	I want to	So
Smart Farmer	Access current and historical information about my carrier AGLVs	I am aware of their history, logs and alerts, as well as for fruit management purposes
Smart Farmer	Easily interact with my carrier AGLVs via my mobile phone	I can easily trigger visualization and actuation tasks
Smart Farmer	Securely interact with my AGLVs via my mobile phone	Only I, as an authorized user with the appropriate access rights, can monitor or manage my AGLVs.

3.3 Industry 4.0 Living Lab

In this section, the three use cases under the Industry 4.0 Living Lab are analysed towards their Ambient Intelligence needs and specifically towards identifying their requirements from the IoT devices discovery, indexing, access control and AR interaction components. The UC analysis includes preconditions of each use case from the ambient intelligence perspective,

mapping of those preconditions to Ambient Intelligent tools, use case data analysis and use case Ambient Intelligence techniques analysis.

3.3.1 Preconditions per application

The preconditions for Use case #6 "Human-centred safety in a self-aware indoor factory environment" are given below.

a. The **Autonomous Guide Vehicles (AGVs)** working in an indoor area of the Bosch factory will need to be **recognized by visual and non-visual methods**. The visual recognition will be done by several CCTVCTTV cameras installed on the ceiling of the working area and a **computer vision algorithm** trained to detect the AGVs from the images. The non-visual recognition will be done based on radio communications between the AGVs and several beacons installed in the factory, which will allow to detect the presence and calculate an updated position for the tracked device. Furthermore, object detection and location methods using visual light communications and optical camera communications (OCC).
b. **Humans** present in the area also need to be **recognized by visual methods** in order to avoid the collision between humans and AGVs. Other objects such us human-guided vehicles or packaging units that can also be recognized by computer vision.
c. The **AGVs** detected using non-visual methods will be positioned using **Ultra wide-band technologies**. The UWB beacons will need to be installed in the factory. The calculations of the position (see section 2.2.2.2) based on multilateration can be implemented in the embedded devices installed in the AGV or in an edge server.
d. The position of the AGVs using **visual light positioning** and optical camera communications will be calculated in the embedded devices installed in the AGV.
e. The positioning of the objects detected by computer vision will be calculated by **camera calibrations and homographic transformations** (see section 2.2.4). These calculations will be run in an **edge server**.
f. The AGVs recognized by the recognition methods will be **indexed** in a database. It will also be necessary to register the position of the device.
g. The position of the humans recognized by the computer vision methods will also be registered in order to be able to avoid possible collisions with the AGVs.
h. There must be some fix screens on the aisles of the factory showing with AR the AGVs approaching from side aisles and that might hide by walls and be not visible from the position of the screen.
i. The **AR interface** might show additional information of the AGV such as the velocity, the status or the estimated time of arrival to the crossing point.
j. The human-guided vehicles must have a tablet or similar device showing information of potential collision with enough anticipation.

The preconditions for Use case #7 "AR Assisted build-to-order assembly" are identified as follows.

a. Assembly workers in ABB factory receive assembly instructions via an AR application, utilizing an **AR headset**, e.g. Microsoft's HoloLens.
b. The **AR application** must support the importing of **CAD-models**, the imported model will be used to visualize assembly instructions.

c. The AR application must be able to anchor the imported model in a sensible way into the mixed reality environment, e.g. overlay the 3D model on top of the physical object that is being worked on.
d. The AR application needs to communicate with existing software systems, to read the status of the assembly and to produce relevant instructions.

The preconditions for Use case #8 "Digital Powertrain and condition monitoring" are listed below.

e. **Powertrains** consist of common components, such as the controlling drive unit, which produce multiple signals that can be used for **condition monitoring**. The underlying methods and protocols used to gather these signals may vary, thus a flexible approach to handle large amounts of heterogeneous powertrain ensembles is required for the sake of application development.
f. The devices are operated in mostly industrial settings, thus are under strict security and privacy policies. Access to certain elements and/or interfaces of the powertrain must be restricted.
g. Applying **machine learning** methods is of interest, as data can be gathered from multiple similar powertrain ensembles. Especially the drive units used are often identical in their functionality.

3.3.2 Mapping to IoT-NGIN Ambient Intelligence tools

The preconditions collected in the previous section are mapped to IoT-NGIN's Ambient Intelligence platform in Table 15.

Table 15: Mapping of Industry 4.0 Living Lab to IoT-NGIN Ambient Intelligence platform.

Device recognition	Device indexing	Access control	AR/MR tools
Discovery of AGVs with computer vision and non-visual methods (UWB, VLP, LDIT)	Indexing of AGVs (ID, position, speed, status)	Access to the information of the AGV	Information of the AGVs delivered by AR

3.3.3 Use Case Requirements analysis

The following tables describe specific requirements of the use cases in the Industry 4.0 Living Lab from the Ambient Intelligence tools.

Table 16: Requirements analysis for Use case #6 "Human-centred safety in a self-aware indoor factory environment".

Use Case		**Human-centred safety in a self-aware indoor factory environment**
Ambient Intelligence Application		AGV – Humans collisions prevention
Ambient Intelligence Service		
Device discovery	**Device recognized**	Automated Guided Vehicles, human-guided vehicles (also humans)
	Registration method	Manual / Automatic recognition For UWB and VLP methods the first registration will be manual
	Recognition method	UWB - VLP– Computer Vision
Device Indexing	**Data registered**	Device type ['AGV'], ID [integer], Position [x, y], timestamp
	Data Type(s)	JSON files
	Database type	TBD (part of IoT-NGIN)
	Data size	Depending on the number of devices and how often they are detected <500 Mb per day
Device Access Control	**User rights**	Not needed
	Spatial proximity	Not needed
Actuation – AR interaction	**Actuation**	If there is a potential collision detected, the workers will receive a notification from the system. Other actuations are not needed
	AR interface	Fixed screen and tablets
	AR software	TBD

Table 17: Requirements analysis for Use case #7 "Human-centred augmented reality assisted build-to-order assembly".

Use Case		**Human-centred augmented reality assisted build-to-order assembly**
Ambient Intelligence Application		AR assistance on assembly steps
Ambient Intelligence Service		
Device discovery	**Device recognized**	Drive cabinet
	Registration method	Manual / Automatic recognition
	Recognition method	Computer Vision – AR Model target
Device Indexing	**Data registered**	List of assembly steps, status of assembly
	Data Type(s)	TBD
	Database type	Local
	Data size	TBD
Device Access Control	**User rights**	Not needed
	Spatial proximity	Not needed
Actuation – AR interaction	**Actuation**	Completion of assembly steps (wiring). Automatic recognition of completed step, or manual interaction by user
	AR interface	AR Headset, HoloLens
	AR software	Vuforia studio / Unity+Pixyz, TBD

Table 18: Requirements analysis for Use case #8 "Digital powertrain and condition monitoring "

Use Case		**Digital powertrain and condition monitoring**
Ambient Intelligence Application		Access control to the powertrain
Ambient Intelligence Service		
Device discovery	**Device recognized**	Powertrain ensemble (Drive + motor + sensors)
	Registration method	Manual
	Recognition method	Manual
Device Indexing	**Data registered**	Drive operating data (motor speed, voltage, torque, current …). External sensors (temperature, thermal camera data, accelerometer data).
	Data Type(s)	The various sensor and device signals are gathered to an intermediary gateway device running node-red. The gateway device serves/forwards mostly JSON payloads, but can adapt to other formats as well
	Database type	Local, timeseries database
	Data size	TBD
Device Access Control	**User rights**	Access is restricted by default, only authorized applications and users should have access to the data.
	Spatial proximity	Not needed
Actuation – AR interaction	**Actuation**	Not needed
	AR interface	Not needed
	AR software	Not needed

3.3.4 User Story

The scope and use of the Ambient Intelligence services in all of the use cases of the Industry LL are revealed via the user story.

Table 19 and Table 20 describe simple statements on behalf of the Actor as to their desired use and scope of each use case, respectively.

Table 19: User story table for the "Human-centred safety in a self-aware indoor factory environment" use case.

As a	I want to	So
Worker of the factory walking on one of the aisles	Check in a screen if there is an AGV coming from one of the transversal aisles that I cannot see	I can enter the aisle safely
Worker of the factory walking on one of the aisles	Receive a notification in a tablet when there is a risk of collision	I can avoid the collision
Forklift driver carrying packages on the factory aisles	Check in a screen if there is a AGV coming from one of the transversal aisles that I cannot see	I can stop or reduce the speed of the forklift
Forklift driver carrying packages on the factory aisles	Receive a notification in a tablet when there is a risk of collision	I can avoid the collision

3.4 Energy Grid Active Monitoring/Control Living Lab (EMOT)

In this section, one relevant use case (UC #10) under the Energy Grid Active Monitoring/Control Living Lab is analysed towards identifying their requirements from the IoT devices discovery, indexing, access control and AR/MR interaction components. The other use case of the living lab (UC #9) does not require any IoT-NGIN component developed in WP4. The UC analysis includes preconditions of the use case from the ambient intelligence perspective, mapping of those preconditions to Ambient Intelligent tools, use case data analysis and use case Ambient Intelligence techniques analysis.

3.4.1 Preconditions per application

The preconditions for Use case #10 "Driver-friendly dispatchable EV charging" are identified as follows.

a. **Charging stations** deployed in Smart Energy Grid Active Monitoring/Control Living Lab will need to be recognized by visual method. The **visual recognition** will be done by **cameras** installed on the electric vehicles fleet involved in the Italian pilot demonstration activities;
b. Charging stations recognized by the recognition method will be **indexed** in a database and a link with the management and monitoring platform of the charging stations will be created to enable interaction in AR mode;
c. **AR interface** might show additional information of the charging station such as the connector type, power output, energy retailers and Demand Response (DR) campaigns;
d. Electric Vehicle (EV) users must have a tablet or similar device for AR interaction with charging station. In addition to viewing the aforementioned information of the station, the user can also actuate in AR mode with the charging station; the user can authenticate themselves, set up the desired power output, select an energy retailer, and select to participate in a DR campaign, start the charging session and manage the payment.

3.4.2 Mapping to IoT-NGIN Ambient Intelligence tools

The preconditions collected in the previous section are mapped to IoT-NGIN's Ambient Intelligence platform in Table 20.

Table 20: Mapping of Energy Grid Active Monitoring/Control Living Lab to IoT-NGIN Ambient Intelligence platform.

Device recognition	Device indexing	Access control	AR/MR tools
Discovery of charging stations with visual method	Indexing of charging stations	Access to the information of the charging stations	Control of the charging stations delivered by AR

3.4.3 Use Case Requirements Analysis

The following tables describe, from a generic point of view, the features of the data produced in the Energy Grid Active Monitoring/Control Living Lab that is relevant for the development of the Ambient Intelligence tools.

Table 21: Requirements analysis for Use case #10 "Driver-friendly dispatchable EV charging".

Use Case		**Driver-friendly dispatchable EV charging**
Ambient Intelligence Application		Charging station – AR interaction
Ambient Intelligence Service		
Device discovery	**Device recognized**	Charging Stations
	Registration method	Automatic recognition
	Recognition method	Visual method - Computer Vision
Device Indexing	**Data registered**	Device type ['charging stations'], ID [integer], Position [x, y], timestamp
	Data Type(s)	JSON files
	Database type	TBD
	Data size	500 Mb per day
Device Access Control	**User rights**	Authentication with username and password
	Spatial proximity	Less than 2 meters
Actuation – AR interaction	**Actuation**	Charging session management
	AR interface	Tablets and similar devices
	AR software	TBD

3.4.4 User Story

WP4 researches to improve IoT devices discovery, recognition and indexing and to specify a new way to control IoT devices, using AR technology; in Smart Energy Grid Active Monitoring/Control Living Lab, UC#10 "Driver-friendly dispatchable EV charging" is considered for WP4 demonstration activities. In the UC#10, electric mobility plays the role of a flexibility provider for the stabilization of the electricity grid heavily penetrated by distributed renewable energy plants. In this scenario, the charging station is the IoT device to be recognized and to be enabled for the interaction using AR technology; the electric vehicle deployed in the Italian living lab, equipped with a camera, will be used to collect images of the charging stations at different times of the day and in different time conditions,

so as to be able to guarantee learning by artificial intelligence system and obtain a reliable discovery and recognition mechanism. After learning process, it will be possible to start the implementation of AR interaction mode with the charging station, using the APIs of charging station monitoring and management platform developed by EMOT. In this way, EV user will be able to view the details of the charging station, as well as access and control it in AR mode; in particular, EV user will be able to perform charging sessions setting up desired energy retailers and power output. In addition, EV user will have the possibility to select DR campaign participation and help stabilize the network by earning a discount on the cost of the charge, agreeing to a modulated charging session based on the Distribution System Operator (DSO) needs.

Table 22: User story table for the "Driver-friendly dispatchable EV charging" use case.

As a	I want to	So
Charging station operator	Enable AR mode for interaction between charging station and electric vehicle user	Electric vehicle users can use AR smart glasses to control charging station

4 Ambient Intelligence in IoT-NGIN

The emergence of IoT has introduced a high level of automation in monitoring and actuation in multiple application domains, via numerous and continuously emerging applications. The increased interest for digital services, remote interaction, as well as "gentle" integration of such digital services in our everyday lives and business environments has led to the increasing adoption of ambient intelligence in IoT systems. In parallel to this, the logical next step in the IoT domain is towards to the *Tactile Internet*, as underlined in an ITU report 2014 [44]. The same report states that "*the Tactile Internet will enable haptic interaction with visual feedback, with technical systems supporting not just audiovisual interaction, but also that involving robotic systems to be controlled with an imperceptible time-lag*".

IoT-NGIN aims to support ambient intelligence and tactile internet, as part of the next-generation IoT. In addition to the 5G enhancements, which are out of the scope of this deliverable, IoT-NGIN contributions are towards the following components of the tactile internet architecture considered in [45]:

- The tactile service manager (TSM), which primarily deals with disseminating service-specific information between the two tactile edges, as well as with registration and authentication functions.
- The Support Engine (SE), which provides both computing and storage resources for meeting the performance, delay and reliability requirements of the E2E communications at the tactile edges. SE also runs computationally heavy AI functionality, to support AI application in the tactile edges. In other words, heavier AI tasks are offloaded here, enabling the perception of real-time connectivity via predictive analytics and ensuring fast and reliable communication along the path between the source and destination tactile devices.

The IoT-NGIN contributions towards enhancing IoT with ambient intelligence and towards the tactile internet refer to:

- Automatic identification of "things"
- Access control in "things'" functionality
- Advanced interaction with the "things", based on Augmented Reality (AR) tools
- Actuation controls on the "things", applied either automatically in the context of ambient intelligence or locally/remotely as a result of AR/VR or haptic interaction.

These functionalities are delivered via the AmI components of IoT-NGIN, which are depicted in light blue and interact among them and with other IoT-NGIN components, as depicted in Figure 11. Specifically, the IoT-NGIN AmI components include:

- *IoT Device Discovery (IDD)*, which is responsible for identifying or recognizing an IoT device, based on conventional methods, such as QR scanning, or advanced visual methods such as combining computer vision based recognition with UWB based localization.
- *IoT Device Indexing (IDI)*, which provides information related to IoT devices' status and characteristics. IDI can be used to collect device information by the physical device and appropriately update its digital twin.
- *IoT Device Access Control (IDAC)*, which provides controlled access to the IoT devices, considering a permission- and role-based mechanism applied before any activity on the device by any user.

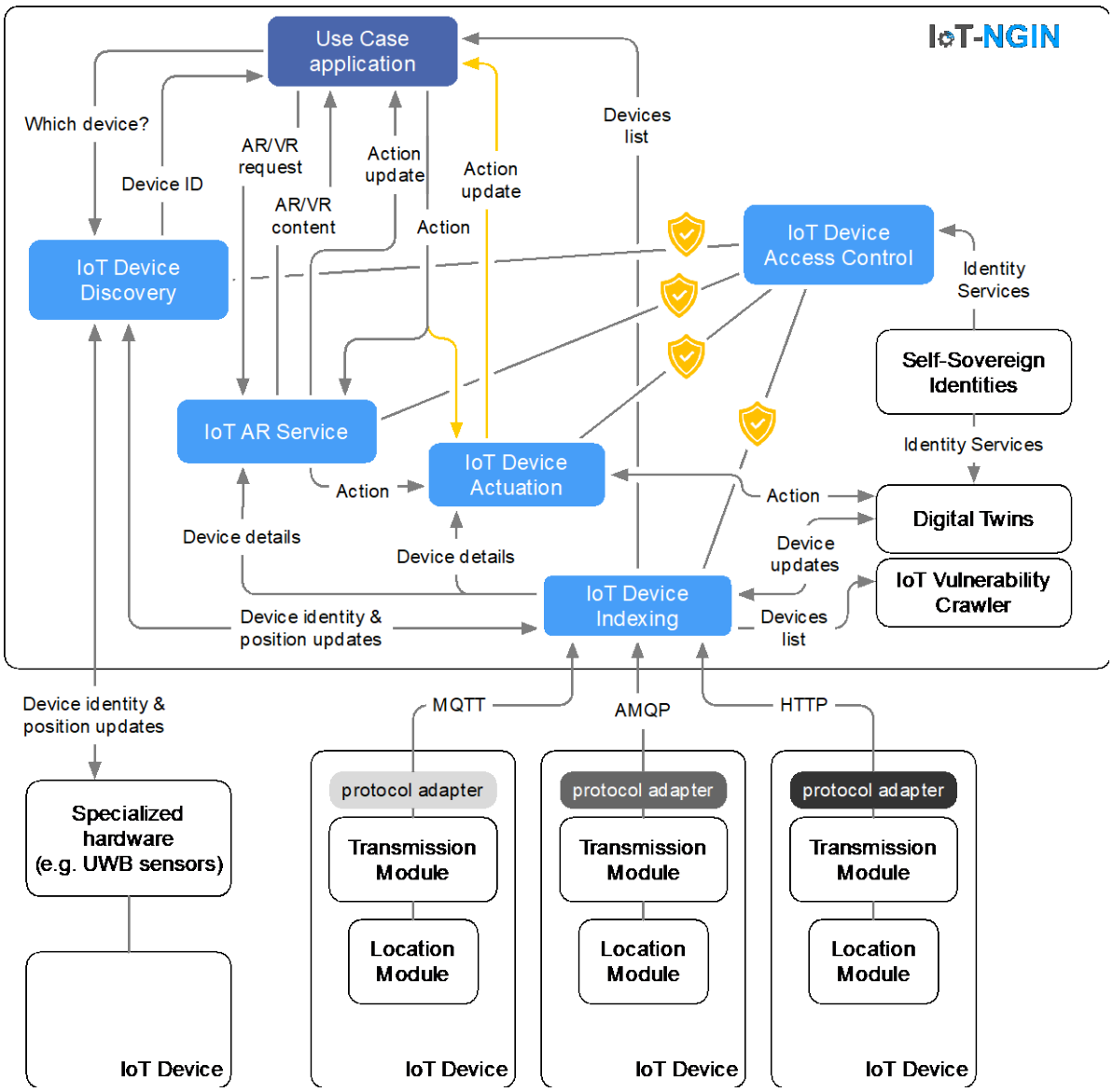

Figure 11: High-level architecture of the IoT-NGIN Ambient Intelligence platform

- *IoT AR service (IAR)*, which is implemented exploiting the *IoT AR assets' repository*, in order to provide application specific AR functionality. The IoT AR assets' repository will be implemented within IoT-NGIN and will include software components from widely accepted AR and gaming platforms.
- *IoT Device Actuation (IDA)*, which enables the actuation controls to be applied on the IoT devices automatically or as a result of (advanced) human interaction.

In the context of an AmI based use case (UC) application, the *IDD* service provides device identification for the IoT devices used in the UC, while it feeds device details and position updates to the *IDI* module. *IDD* may communicate with specialized hardware, such as UWB sensors, in order to determine device positioning, potentially needed for the device identification task. On the other hand, *IDI* receives device information via RESTful or pub/sub communication channels with the IoT devices and may provide device updates to a *Digital Twin* or the device list to the *IoT vulnerability crawler* (both components being developed within the scope of WP5). Then, the UC application may request some AR interaction with

the identified device, which could refer to monitoring or management tasks or even actuation on the device. This request, after proper access control, is made to the *IAR* service, which may potentially retrieve device details from *IDI* or request the device control actions to be made by the *IDA* service. Alternatively, device actuation may be requested directly by the UC application from the *IDA* service. Notably, the device actuation may be applied via a Digital Twin, as shown in the figure, and then applied to the physical device. In any case, the requests between the services involving access to the UC IoT devices must ensure that the requester service or user has received prior authorization to apply the requested action on the IoT device of interest, based on their permissions, by the IDAC service. The service may apply access control, exploiting self-sovereign identities' (SSI) functionality developed within WP5. Last, but not least, the AmI functionality may integrate Distributed Ledger Technologies (DLTs) which will ensure the integrity of information, such as hashes of images of devices, ML models supporting the recognition, access control information, etc.

In the following subsections, the technical specifications of the IoT-NGIN AmI components are provided.

4.1 IoT Device Discovery

In the Iot-NGIN project, the concept of **device discovery** has been defined as the result of two processes 1) device recognition and 2) device positioning. These two processes will be done manually for the discovery of certain devices in some of the use cases just by registering a new device and its position manually in a database. However, in the context of the Ambient Intelligence, the use cases of the IoT-NGIN project will show preferably how different IoT devices are discovered automatically with different recognition and positioning methods.

The IoT-NGIN architecture will support different types of IoT device recognition and positioning methods. In this way, the project will be able to cover a wide range of scenarios and use cases. The following table summarizes the technologies that will be considered for the project. Note that these technologies will require specific hardware to perform the recognition and positioning tasks.

Table 23: IoT Device Discovery methods

IoT Device Discovery		Recognition Method	Positioning Method	Specific Hardware Required
Visual Methods	Computer Vision	Convolutional Neural Network	Homography	Camera
	Visible light positioning	Visual Light Communication	Trilateration	LED Lamps Camera
	QR Codes	QR Code Scanner	GPS	Camera, GPS
Non Visual Methods	Ultra Wide Band positioning	Ultra Wide Band	Trilateration	UWB beacons UWB tag

4.1.1 Description

4.1.1.1 Computer Vision

As mentioned in section 2.1.1, **Convolutional Neural Networks (CNN)** have been found to achieve outcomes that are beyond state-of-the-art when using typical computer vision techniques. Thus, they will be employed as the core of our solution. For the whole structure of the model, we will use an approach that includes a continuous cycle of four steps until the model's accuracy meets and applies to our goals. As for the data, datasets are difficult and time-consuming to create in general, but they are essential for developing computer vision applications. If no specific dataset is available for each use case, fortunately, certain datasets are readily available, although datasets designed for specific object detection tasks usually offer better results. ImageNet is one of the most comprehensive and well-known, containing 14 million images, images contain bounding box annotations, manually annotated using WordNet concepts [46] [47]. Another well-known one is the Microsoft Common Objects in Context (COCO) dataset, loaded with 328,000 images including 91 objects types with 2.5 million labelled instances [14]. Lastly, Indoor Scene Recognition dataset could be useful for the use-cases that need to detect Indoor Scene devices [48].

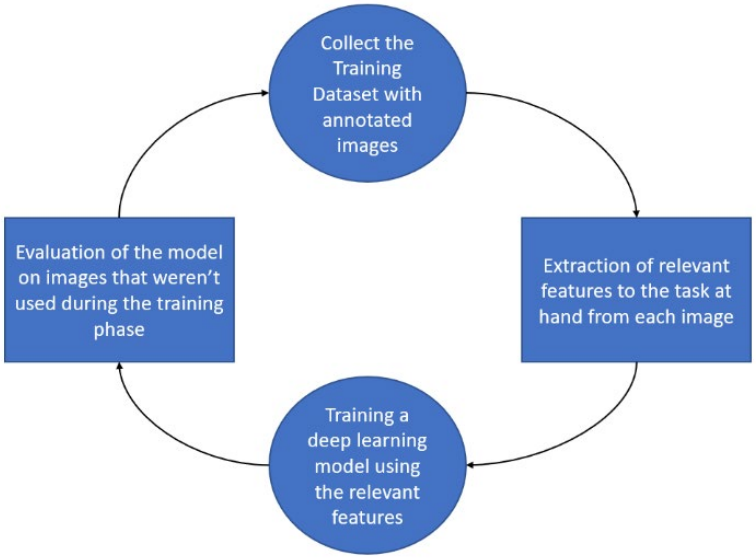

Figure 12: ML model training cycle.

The first step is to either create or use an existing dataset of annotated images. The devices and objects we want to detect in a scene can be annotated with pairs of bounding boxes. After obtaining this dataset, we must extract features relevant to the task at hand from each image. This is an important stage that contains the model's philosophy. The traits used to detect faces which are based on facial criteria, for example, are obviously not the same as those used to recognize IoT devices. The next stage is to train a deep learning model using the features that have been extracted. Training involves sending a large number of pictures to a machine learning model, which will then learn how to perform each use case's task based on those features. Following the previous stages, we will need an evaluation method that uses images that were not used during the training phase. This allows the training model's accuracy to be tested. The architecture of the model will not be predefined; it will be

decided by the nature of each use case, and it will leverage the state-of-the-art solutions described in Section 2.

The Convolutional Neural Network compresses data, resulting in a smaller vector that represents a million-pixel image. Unattended classification or models suited to the data can be constructed with relatively few samples using learnt representations. Models in traditional architecture have as a final layer a Dense layer, where the result is determined based on the type of problem. For instance, if it is pretended to classify vehicles and create a model that predicts which colour they are, the model will produce two results: the vehicle's class and colour. In our case, we want the model to be able to predict both the position and the class of an object detected for indexing purposes. The generic architecture of the deep learning model that will be employed for our approach is shown in Figure 13.

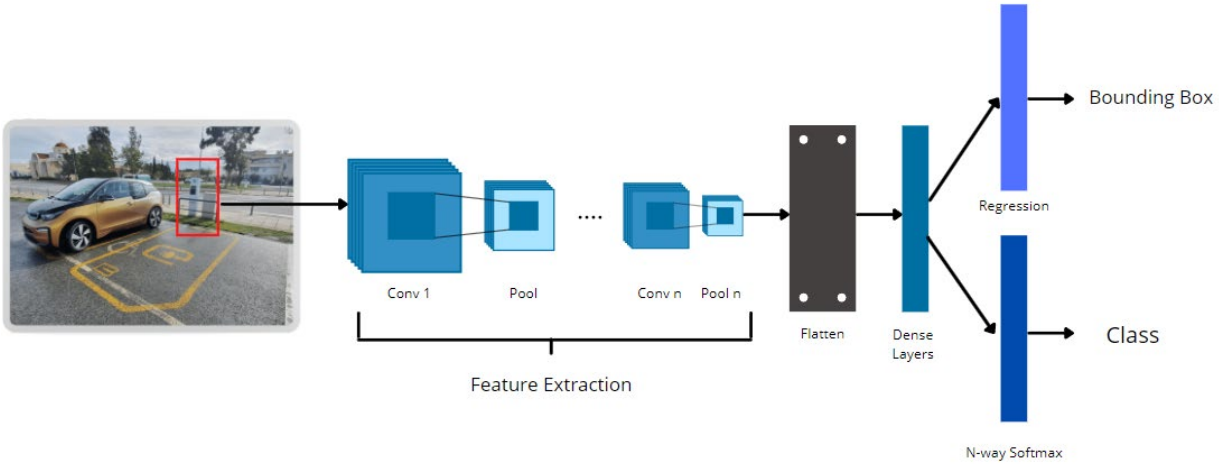

Figure 13: Generic architecture of deep learning models.

The novel aspect of our solution would be the usage of filter to the input image before parsing it to our network. Applying filters to the image helps in enhancing quality of the image (e.g., removing noise). There is a big variety of image filtering (Smoothing Filter, Gaussian Filter, Laplace) that can be used to help with the object detection. We will perform an investigation by testing a bunch to select the most suitable for the architecture. The usage of a Histogram Equalization of the picture is another aspect that will be investigated to check if it will contribute to better results for the model. In this algorithm a histogram of an image is represented by the intensity vs the number of pixels with that intensity. The histogram of an image that is excessively dark (or too bright) usually has all of the pixels concentrated on one side of the histogram. When the pixels are dispersed among the intensity values, the image has a lot of contrast, which makes it easier to see the object of the model. Essentially, the Histogram Equalization approach is used to improve the contrast in photographs. As a result, it can improve the efficiency of models that employ such photos, resulting in a more accurate output.

4.1.1.1.1 Evaluating an object detection model

We must assess a model's performance on images that were not included in the training once it has been successfully trained. We may assess whether our model is ready to be deployed in each use-case based on training failures and performance indicators. If the model fails to fulfil our objectives, we must either increase the size of the training dataset or alter the model's architecture by adding, removing, or changing some of its layers. We will

plot our metrics based on their accuracy and epochs to help us understand them better. This will also assist us in determining when to halt the model learning process and avoid issues like overfitting and underfitting (Figure 14).

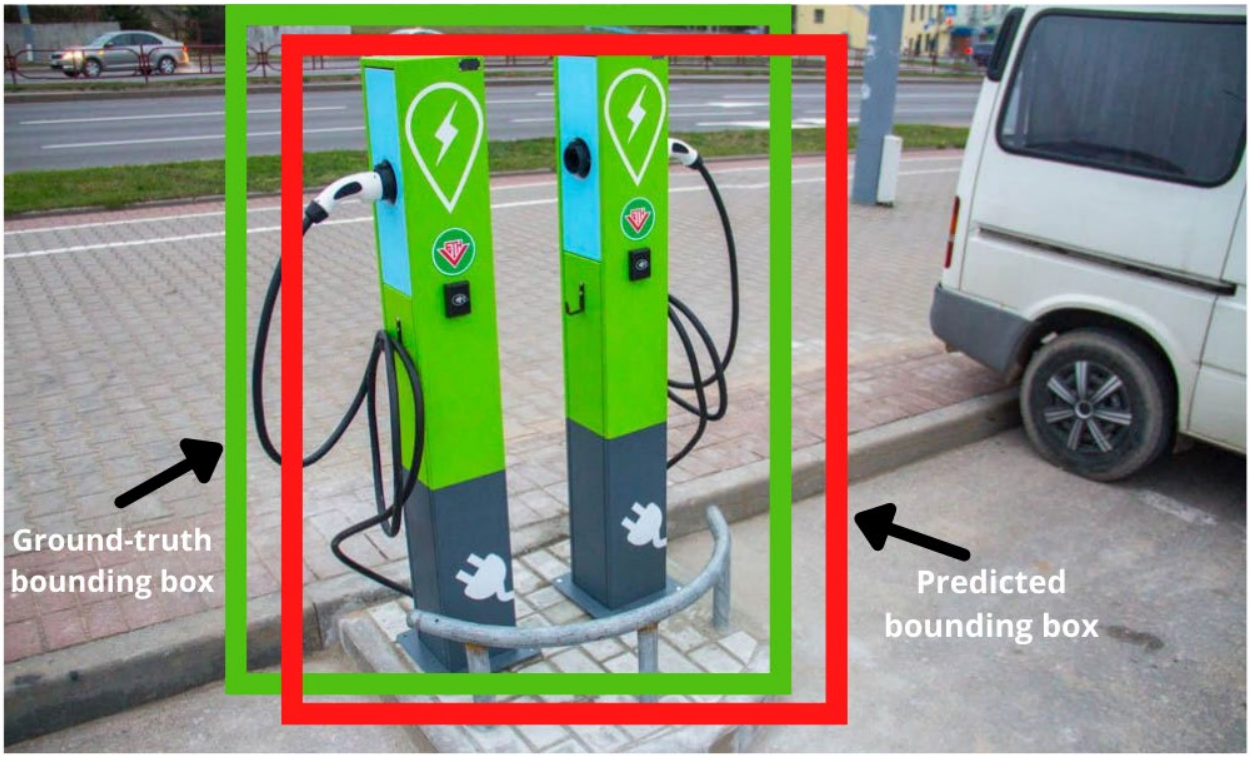

Figure 14: Example of predicted bounding box compared to the ground-truth bounding box.

4.1.1.1.2 Metrics

Evaluating object detection models is more difficult than evaluating a simple supervised learning model. The object detection problem has several unique challenges that are not encountered in typical classification tasks. This is because one image can contain a large number of objects, each of which can belong to various classes. As a result, our testing should be divided into two areas: the first is to see whether the model was able to find all the items in an image, and the second is to see if the model was able to classify them correctly. For these two goals, researchers devised a statistic known as the mean Average Precision, which integrates the two into a single metric (mAP). The bounding boxes are the boxes drawn to identify the object's location when you annotate it. The output of an object detection model is divided into three parts:

1. The bounding boxes, which are basically a set of 4 variables: X_1 and X_2 (the coordinates of the box in the photo), width and height of the box
2. The class of the bounding box, e.g. "Charging Station"
3. The probability score for that component of the prediction score

We utilize the Intersection over Union (IoU) as a similarity metric to determine if an object was located. It is calculated by dividing the overlap's area by the combined size of the two bounding boxes. Because the first step is to determine whether two bounding boxes refer to the same item, the IoU is required to calculate the Average Precision. To answer this question, we must first define a default IoU value threshold, which is commonly set to 0.5. The bounding

box coordinates and the probability score for the anticipated class are outputs from the model. You begin by sorting the probability scores, and then use an IoU threshold to determine if the bounding box is True Positive (TP) or False Positive (FP). If there are many detections for a single object, the detection with the highest IoU is designated as TP, while the others are designated as FP (Figure 15).

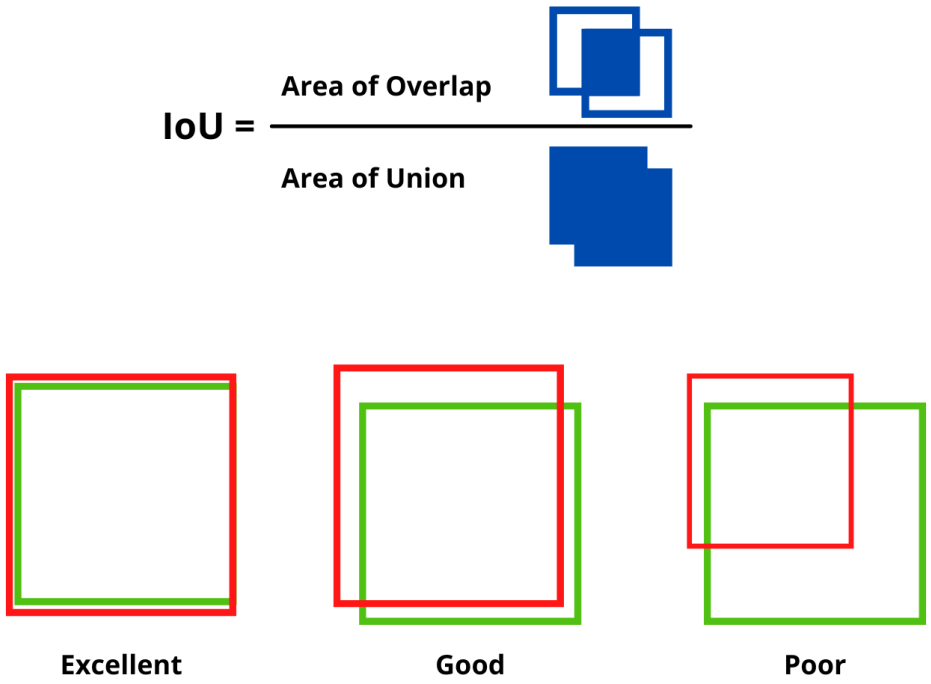

Figure 15: IoU definition and example.

At each line, you compute the precision/recall by counting the accumulated TP and FP. The Mean Average Precision (mAP) is the averaged AP over all item categories, and it is the statistic that is most commonly used to assess an object detection model. One feature of object annotation tasks is that there is no consensus on how to represent the location and size of objects, and different metrics are used by different performance assessment systems.

The mAP is useful in a competition environment or to get a rapid assessment of how a model is functioning, but different metrics will be needed, if it is desired to understand what the model is doing. TIDE was created by the authors of [49] to overcome this problem. TIDE is a general toolbox for computing and evaluating the influence of object detection and instance segmentation on overall performance. We will use it to gain a better understanding of the mistakes our model makes so that we can adjust its structure properly.

4.1.1.1.3 Split of Training/Testing Dataset

One thing to keep in mind is that in a multiclass classification task, the dataset's distribution must be known, because each observation can only have one class. As a result, the dataset may be simply divided into training and testing. However, in object detection, one image can contain a variety of items from several classes. As a result, we must ensure that the population of each class in our training and test sets is adequately balanced.

4.1.1.2 Visual Light Positioning

The Visual Light Positioning system is based in two types of devices:

- An infrastructure of switching LEDs (which can be controlled and modulated to transmit information) installed in the ceiling. LEDs will be modulated with specific patterns so than consecutive lights could be distinguished and identified. For this, specific hardware needs to be integrated with the LEDs (that will be commercial COTS devices). The lights are located in well-known locations (in the figure 3D world coordinates). These locations are used to compute the position of the moving IoT device.
- A localization unit composed of an embedded computing device (such as a Raspberry Pi), a communication interface, a camera device and an external power source (e.g. a power bank or supplied by the object where the unit is attached). The camera will be used to capture images and video that will allow the positioning of the device.

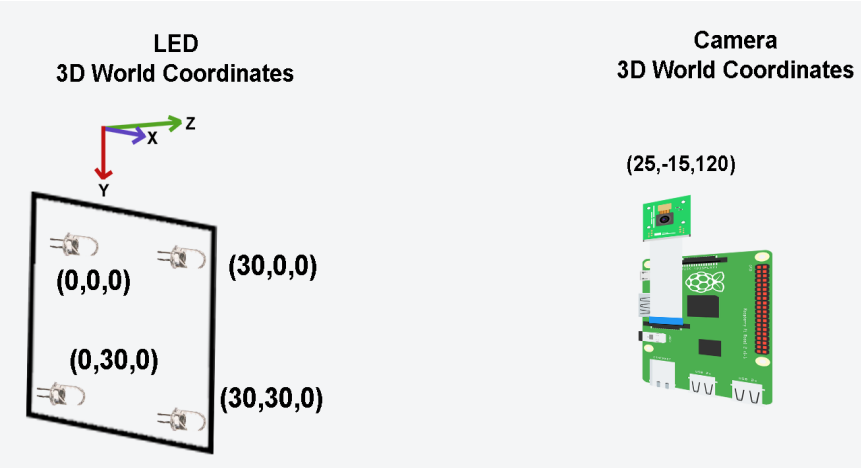

Figure 16: VLP-based positioning system. Non-visual recognition.

In order to locate the IoT device, different processes need to be implemented in the localization unit. OpenCV library will be used to analyze and process the information obtained with the camera. The full process needs to be fast enough to allow the tracking of a moving device in real time.

Figure 17 illustrates the different blocks that the Visual Light Positioning location module needs to implement.

First, the recognition of the lights needs to be performed. The image taken by the camera is processed with OpenCV software to recognize the light emitting sources. Concretely, the program will find the contours (perimetral pixels of the area), the area (how large is the area of white pixels) and the centre (centre of the area) of each LED.

Secondly, it is necessary to sample and demodulate/recover the information sent by each of the recognized LEDs. So, each LED will be identified and its position in the ceiling can be univocally determined (provided that the LED encoding scheme in the ceiling has been assigned adequately, so that each set of lights has a different pattern).

After this, a trilateration algorithm can be used to determine the position of the IoT object based on the distance from the camera to the different LEDs. For a 3-D position at least 4 LEDs need to be captured by the camera. If the object moves only in a plane (no height variation), only 3 lights would be necessary and the positioning estimation can be simplified.

Finally, once the position is calculated in the device, it will be sent to the platform (to the IoT Device Indexing module) through a wireless interface.

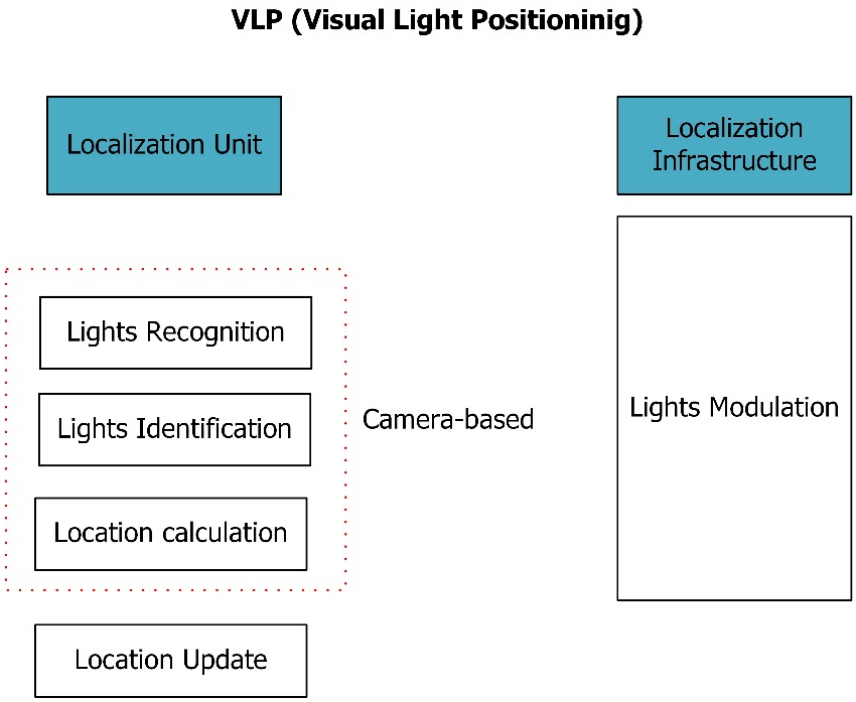

Figure 17: Location Module for the VLP based positioning. Non-visual recognition.

An initial HW/SW prototype with basic functionalities has been deployed and is being validated in the laboratory premises focusing on the requirements of the Industrial use case.

4.1.1.3 Code scanning

Barcodes and QR codes become an essential part of IoT towards the vision of the Internet of Everything (IoE) as the primary method of labelling products, items and, generally, things. Indeed, simpler machine-readable codes, such as linear and 2D codes, realize wider applicability than more complex ones, such as Radio-Frequency Identification (RFID), since they can be scanned with a simple scan of a smartphone or tablet. Two-dimensional codes, such as QR codes, can deliver a lot more information about the product or the company than regular barcodes, opening up the way for both the user and the manufacturer to receive information of their interest about the device, allowing to move from "marketing Big Data" to "after-sales feedback Big Data".

The Code Scanning module of IDD aims to enable code scanning primarily in mobile devices, such as smartphones and tables. This module may be called by an IoT-NGIN or a third-party service or via a mobile app, potentially interacting with IoT-NGIN services.
The component is responsible for activating the scanning operation, decoding the scanned code and providing it to the requester application or service as structured information. Based

on this information, the requester application' service will initiate the services associated with the scanned information. These could include visualization of information, accessing websites as indicated from scanned URLs, initiating services associated with the scanned device (e.g. AR services), etc. As depicted in Figure 18, an IoT-NGIN or third-party service or even a mobile application may call the Code Scanner through its API, which then captures images of the code via the mobile device's camera. Then, these are passed to the *Scan Service*, which detects and decodes the code in the captured images.
IDD supports on-device scanning of smart tags, covering standard formats of both linear (e.g. Code 39 [50], Code 128 [51], Code 16K [52], PDF417 [53], etc.) and 2D codes (such as QR code [54], Aztec [55], Data Matrix [56], etc.). Moreover, the component will consider JAB code (Just Another Bar Code), which is a high-capacity 2D colour bar code that can encode more data than traditional black/white codes and is currently under standardization [57] [58].
In addition, the Code Scanning module will leverage ML algorithms for automatic detection of the code patterns. Pre-trained ML models will allow the detection to take place on the device, automatically detecting the code format.

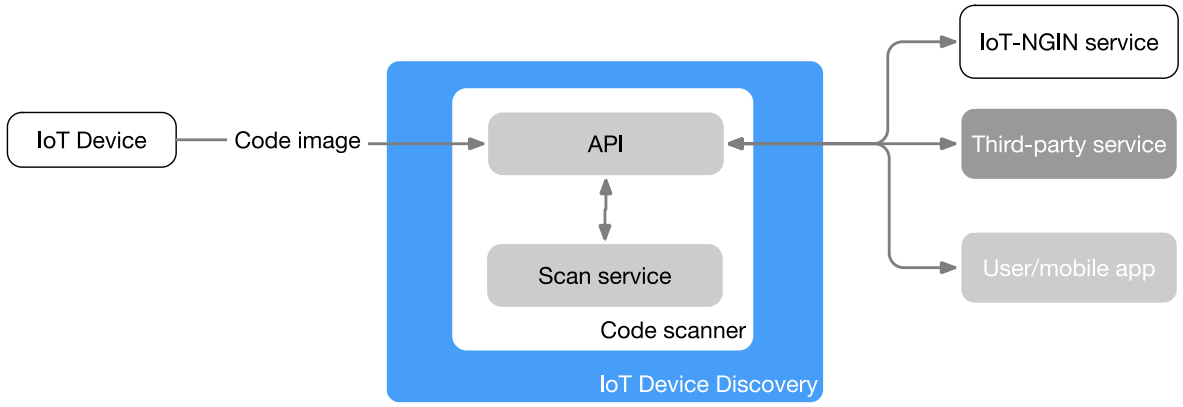

Figure 18: The *Code Scanning* module of the *IoT Device Discovery* component

4.1.1.4 Ultra Wide Band Positioning

As explained in Section 2, the architecture of the UWB-based positioning system relies on two types of devices:

- UWB anchors: Which act as the localization infrastructure. These are fixed and well-known references installed in the area where the IoT objects will be located.
- UWB tag: Hardware equipment attached to the IoT device to be located.

The following figure illustrates the hardware components that will be implemented for these two devices.

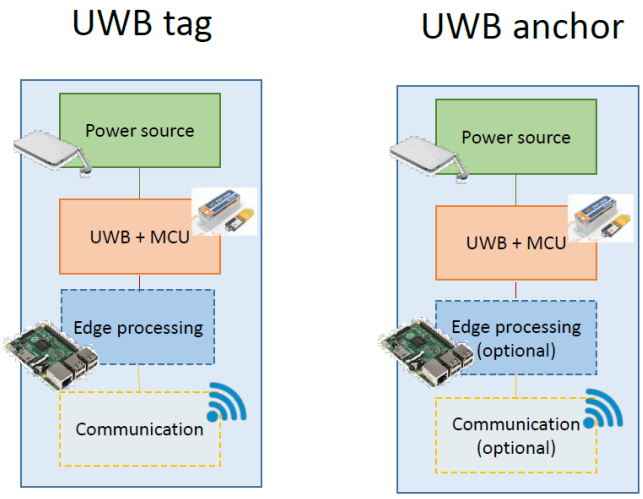

Figure 19: UWB-based positioning architecture elements.

UWB tags will be powered by an external power source (which could be a power bank or the supply provided by the object to be located, e.g. an AGV device). The tag will also integrate a computer unit with sufficient resources to run the localization algorithms (edge computing in the end-device). Finally, it will include a communication interface to transmit the localization updates to the IoT-NGIN platform (concretely, to the IoT Device Indexing module).

UWB anchors could also implement some edge processing and communication capabilities (e.g. for TDOA based approaches where the tag sends a message and the position is calculated based on the reception of the message in the different anchors). However, for the IoT-NGIN project a simplified anchor approach will be initially considered, which will alleviate the requirements for the deployment of the infrastructure (e.g. no additional communication interfaces will be required).

The following figure illustrates the different blocks that the UWB location module needs to implement. The localization will be triggered by the localization unit (tag). Initially, before performing a new localization, a tag needs to determine which anchors are the ones that will be used to calculate the position. These can be determined by listening to the beacons sent in the network. After this, the device will send ranging message (polls) to these anchors. With the responses the tag will be able to compute its distance from each of the anchor nodes (at least three anchors are needed). To minimize collisions a TDMA-based channel access will be used. Furthermore, a Two-Way Ranging scheme has been chosen to measure the distances between the tag and the different anchors. After this, the calculation of the position will be performed on the device using trilateration algorithms. Furthermore, optimization techniques will be investigated and implemented to maximize the accuracy of the system in the presence of non-ideal conditions (such as the impact of obstacles, no line of sight (NLOS) situations or the degradation caused by some antenna alignments) which will affect the performance of the system. For these, UWB signal quality parameters will be collected and analysed. Finally, a localization update will be sent to the system. This information can be also used as a notification of the presence/discovery of an IoT device in the tracked area (in use cases where a precise localization is not necessary).

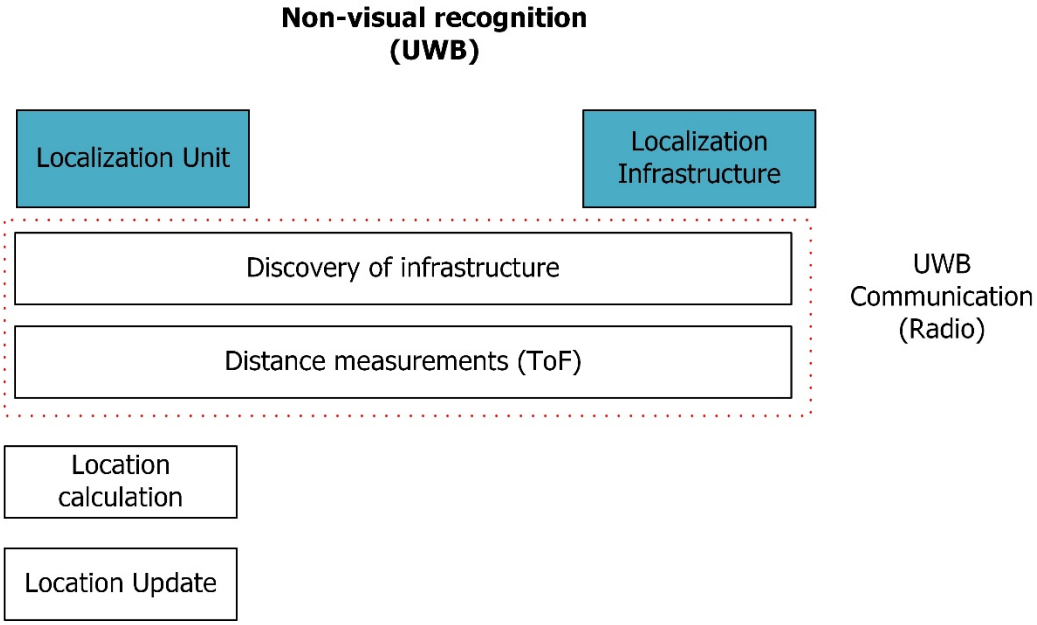

Figure 20: Location Module for the UWB-based positioning. Non-visual recognition.

An initial HW/SW prototype with basic functionalities has been deployed based on a DW1001 UWB module and an ESP32 development board as processing and communications unit. This is being validated in the laboratory premises to see the expected performance in terms of accuracy, object recognition frequency and delay, robustness and scalability. Figure 21 shows a detail of the localization unit and an initial deployment at i2CAT office.

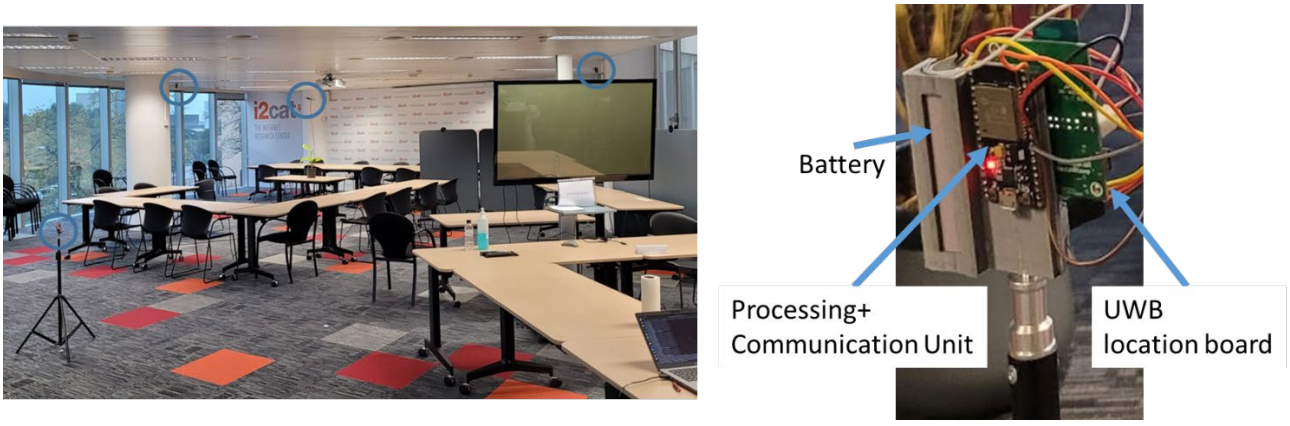

Figure 21: UWB initial deployment at i2CAT.

4.1.2 Interfaces

Table 24: IoT Device Discovery interfaces

IoT Device Discovery module		
Description	The module will enable the IoT-NGIN architecture with the capabilities to support a more dynamic and advanced recognition (identification and localization) of available IoT devices in the scenario. These capabilities will either run in the edge or in own IoT devices.	
Provided Interfaces	VLP and UWB positioning Interface	
	Description	The interface will provide positioning updates of the IoT registered devices. The location of the device can be performed either using Visual Light Positioning or Ultra-Wide Band technology, depending on the infrastructure deployed/available in the use case. Updates will be transmitted to the IoT Device Indexing module using a REST API. The positioning calculation and the message update will run on the IoT device that is located (periodic updates).
	End-point URL	None. The Interface acts as a client (publishes information to the End-Point URL provided by the IoT Device Indexing module)
	Protocol used	HTTP
	Methods	POST (will publish the information)
	Message	TBD. The message will include the following parameters: IoT device identifier, timestamp, localization (x,y coordinates), location method (UWB, VLP), reliability (can be used as an indication of the quality of the measurement)
	Computer Vision Interface	
	Description	The computer vision interface is being defined.
	End-point URL	TBD
	Protocol used	TBD
	Methods	TBD
	Message	TBD
	Code Scanning	
	Description	Through this interface, the user will be able to initiate a code scan operation through a mobile app.
	End-point URL	/codescanner/api/scan/

	Protocol used	HTTPS
	Methods	POST

	Message	{
		"scanner": 1,
		"command": "scan",
		“timestamp”: “2021-12-23T18:25:43.511Z”
		}

	Code scanning with known format	
	Description	Through this interface, the user will be able to initiate a code scan operation for a known code format through a mobile app.
	End-point URL	/codescanner/api/{code_format}/scan/
	Protocol used	HTTPS
	Methods	POST

	Message	{
		“scanner”: 1,
		“command”: “scan_qr”,
		“timestamp”: “2021-12-23T18:25:43.511Z”
		}

	Scanned information	
	Description	Through this interface, the user will be able to receive information about the latest scanned code
	End-point URL	/codescanner/api/scannedinfo/
	Protocol used	HTTPS
	Methods	GET

	Message	{
		"scanner": 1,
		"info": "[info in json format]",
		"timestamp": "2021-12-23T18:25:43.511Z"
		}

Required Interfaces	IoT Device Indexing interface: Will provide the REST API for communicating the positioning updates. IoT Device Access Control: Will provide the REST API for access control interactions. User Application: Interaction through HTTP.	

4.2 IoT Device indexing

The IoT Device Indexing (IDI) module complements the IoT device recognition and positioning module (IDD), effectively enabling the creation of a repository of IoT-NGIN supported devices and allowing quickly querying about their status and basic characteristics. An analysis of the associated requirements (see Section 3) reveals that the device indexing module should be able to support a wide range of status types, such as positions, speed as well as status (e.g. active/inactive, charging/not charging, moving/stationary). Since the LL device types are expected to be different and in real-life situations this heterogeneity is expected to be even more diverse, it is essential to adopt a design as flexible as possible. Further, since the IoT-NGIN heavily relies on the notion of digital twins and each device should be attributed with a digital twin instance, tight integration with the relevant framework, developed in the context of the project's Distributed Ledger Technology (DLT)-enabled Meta-Level Digital Twins activities is sought, so that the device indexing module and the list of digital twins are synchronized (effectively, the device indexing module indexes the physical part of the twins). Figure 22, below, presents the positioning of the Device Indexing module in the overall project activities.

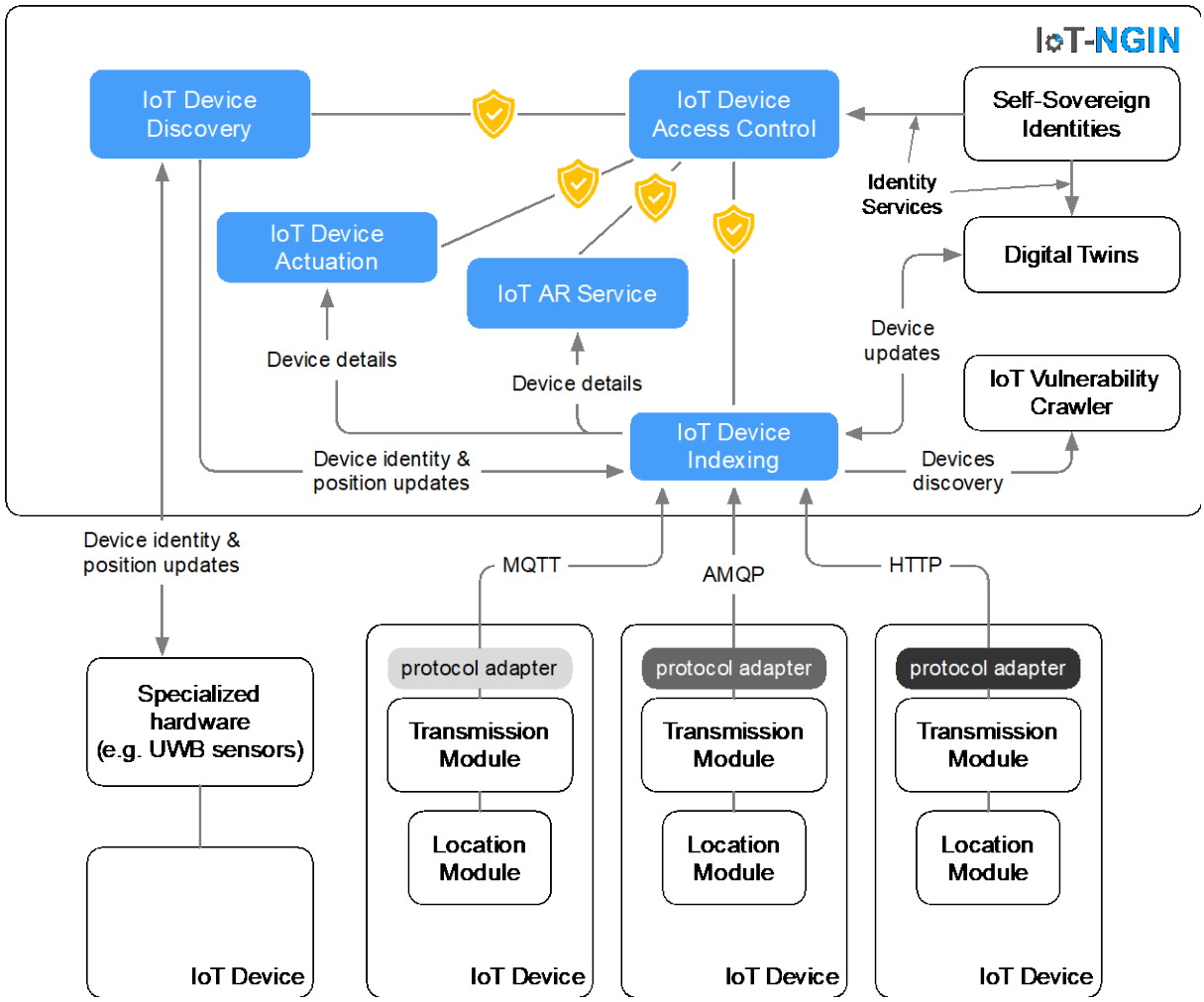

Figure 22: Positioning of the Device Indexing module in the project activities.

Based on Figure 22, the following paragraphs present the design of the Design indexing module, the technologies used, the exposed API as well as its main interactions with the rest of the project components and services.

4.2.1 Description

The Device Indexing module stands at the core of the IoT devices management, actively maintaining not only an index of the devices, but also holding several device-specific information (like location, status etc.). Granted the extremely fragmented device and protocol landscape of IoT and the need to support even event-driven develop (e.g. for integrating with the vulnerability crawler presented in deliverable D5.1 [59]), a flexible approach needs to be adopted for the design and implementation of the component.

FIWARE [60] offers a number of open API (usually open source as well) reusable components, called Generic Enablers (GEs), that can be used in a variety of context and applications, including IoT. Indeed, in [61] and [62], examples of building IoT platform solutions based on FIWARE are presented. In general, the architectural approach is to:

- Allow devices to connect to an IoT Agent which abstracts the complexity of the networking protocols and translates the relevant information into data models and protocols compliant with the OMA Next Generation Service Interfaces (NGSI) 9/10 specifications [63].
- Push the device-provided information to a context broker supporting the OMA NGSI 9/10 standard. From this context broker, external systems such as Complex Event Processing Engines or third-party applications, may get updates on the devices identities and status.

Figure 23, below, depicts a simplified FIWARE-compliant high-level IoT architecture.

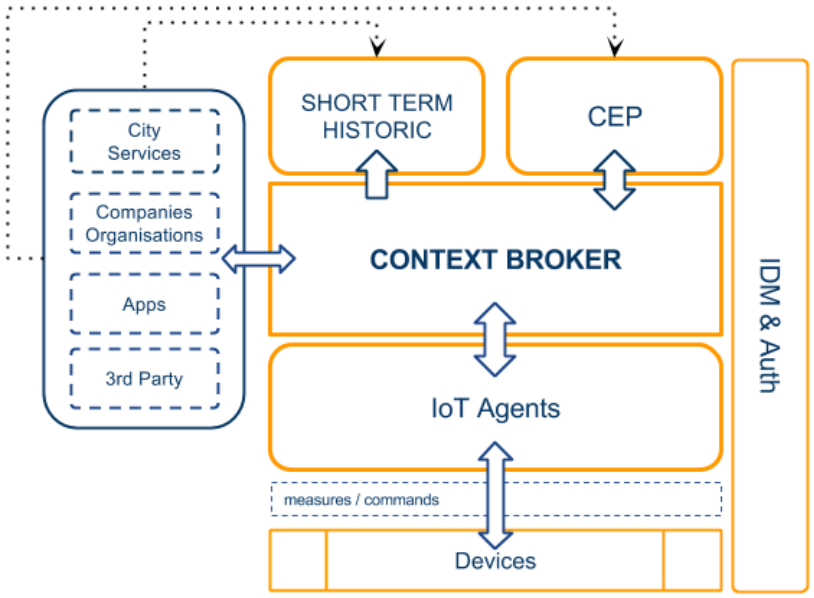

Figure 23: FIWARE-compliant high-level IoT architecture (source: [62]).

It is worth mentioning that the FIWARE context broker (named Orion), exposes both RESTful HTTP APIs (both northbound and southbound) [64] but also a publish-subscribe API, including popular services such as MQTT [65] and can be tuned to operate in the form of a high-availability cluster, allowing for vertical scaling of the service, in cases of increased traffic.

With respect to the IoT Agent, this acts as an implementation of the Backend Device Management GE, according to the FIWARE reference architecture, translating IoT-specific protocols into the NGSI context information protocol, that is the FIWARE standard data exchange model. In fact, as per [66], depending on the target technology and protocol, there are already available five different IoT Agent flavours (namely for JSON, Ultralight, LoRaWaN, Lightweight Machine2Machine and OPC UA), that can be used to support the communication between the devices on one hand, and the context broker, on the other. Notably, it is possible to also define custom IoT Agents. The core set of features supported by the FIWARE IoT Agents are as follows:

- Device registration
- Device information update
- Device command execution and value updates
- Device management
- Device provisioning
- Type configuration

Figure 24, below, graphically presents how a COAP IoT Agent would work (from an interface perspective), starting from the registration/indexing of the device, up to delivering a command to be executed on the device.

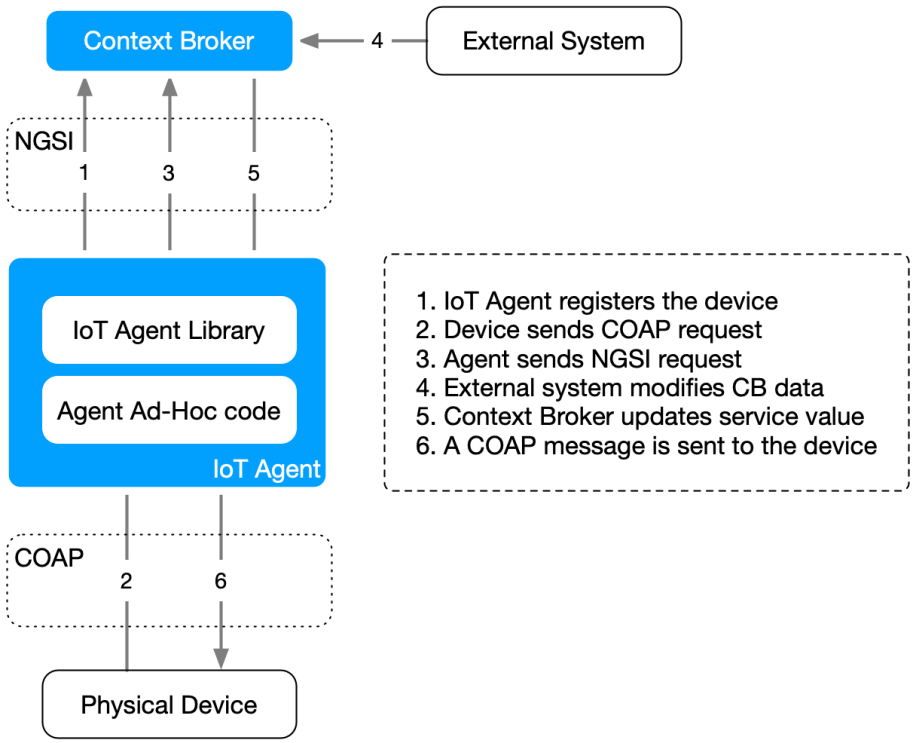

Figure 24: IoT Agent COAP example, internal interactions (source: [67]).

Essentially, as a first step, the IoT agent needs to register the device to the context broker, effectively indexing it. Next, the device sends a protocol-specific request to the IoT Agent (step 2) that gets translated into an NGSI-compatible request and gets pushed to the context broker (step 3). In step 4, we suppose that an external entity (e.g. a client application, or an IoT-NGIN component like the device recognition and positioning one detailed in paragraph 4.1) changes a certain part of the device-specific data. In turn, in step 5, the context broker would change the relevant service value and send it to the IoT Agent. Finally (step 6), the IoT Agent would send a protocol-specific message (in this example COAP) to the device.

Considering the above discussion, from an architectural perspective, the design of the IoT-NGIN indexing module as an integration of the FIWARE IoT Agent framework and the FIWARE Orion Context Broker is depicted in Figure 25.

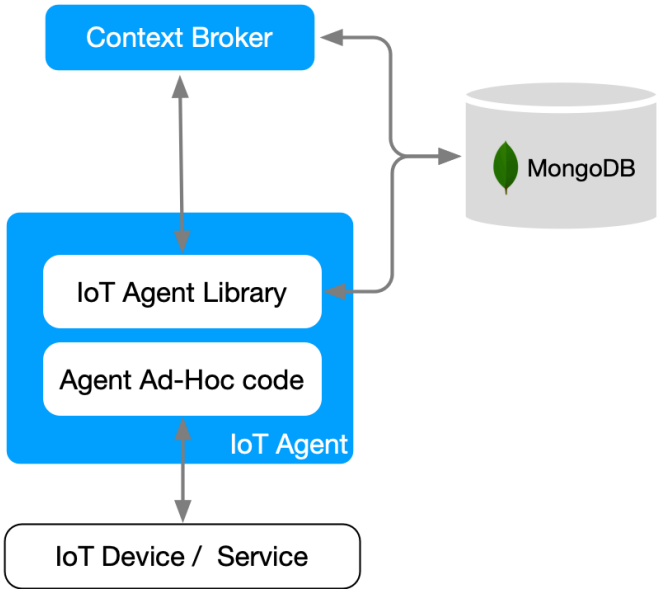

Figure 25: Device Indexing module architecture.

Notably, all the considered subcomponents are able to horizontally scale and operate in the context of relevant service clusters, effectively granting the above design with high-availability characteristics.

4.2.2 Interfaces

The interfaces of the Device Indexing component are the ones exposed by the FIWARE IoT Agent (southbound interface) and the FIWARE Orion Context Broker (northbound interface), documented in [68] and [69], respectively. The relevant specification is omitted in this document for reasons of brevity and guarantees of upstream API changes synchronization.

4.3 IoT Devices access control

Identity management and access control are essential security design concepts, applicable in both physical and cyber systems. Particularly when referring to the digital aspect of the IoT world, identity management and access control are of paramount importance (see

paragraph 2.3 for details and discussion regarding relevant state of the art approaches and EU security guidelines).

In the context of IoT-NGIN, access control upon the IoT devices is required to be based on dynamic, personalized access rights and ambient intelligence. Indicatively, conventional static access rights could be extended by managing user rights by physical proximity or visibility, as measured by a mixed reality headset to take "digital control". In this context, the devices access control component features a transparent permission architecture, evaluating access rights based on physical access (presence in the room, sensors/ glasses), user rights/groups (resident vs guest vs employee) or ownership (who owns the device, digital receipt).

In the following paragraphs, details on the module positioning within the IoT-NGIN context and on its technological architecture are provided.

4.3.1 Description

Considering the importance of security, privacy and trust in the IoT technology (hence also IoT-NGIN) landscape, it is reasonable that the devices access control module lies at the core of the project activities with respect to IoT Ambient Intelligence and Tactile internet enablement (see Figure 26).

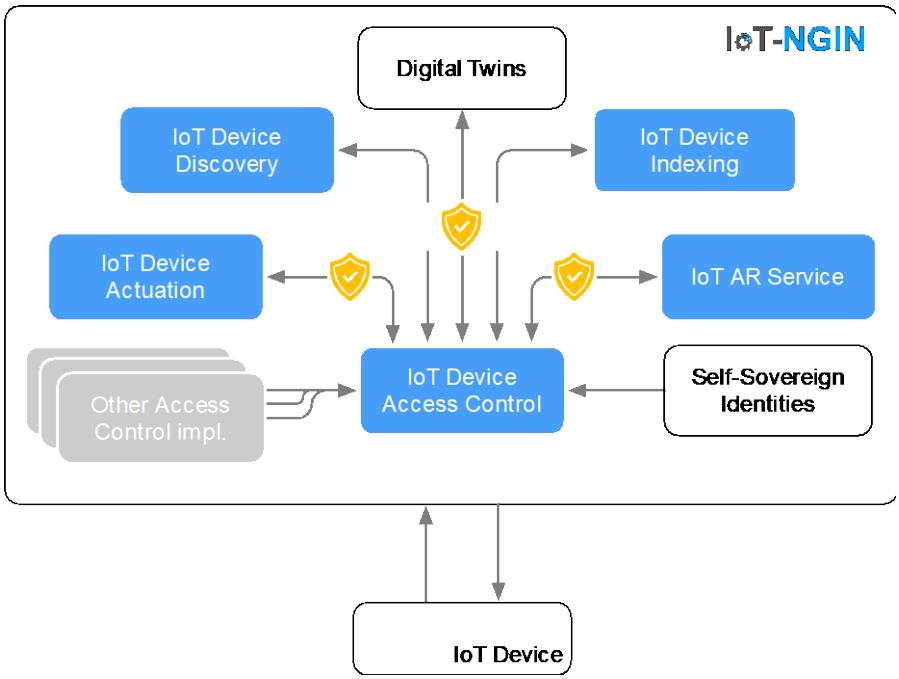

Figure 26: Positioning of the Devices Access Control module in the overall Ambient Intelligence-related components of IoT-NGIN.

As can be seen in Figure 26, the devices access control module should intercept all communications towards the IoT devices, to ensure that the services or, in any case, users are authorized to interact with the IoT devices.

Notably, though, there are numerous standardized authentication and authorization methods and protocols, either native to IoT devices (see paragraph 2.3) or more common in software services architectures (such as OAuth2.0, LDAP, Radius etc.). To this end, and in order to dynamically support arbitrary numbers of authorization interfaces, the complexity of access control management should be abstracted from the IoT devices, digital twins and supporting/interacting services and should be offloaded, instead, to the access control component, which would, then, act as a security middleware between the IoT devices on one hand and their (physical and digital) environment, on the other. The adoption of such an approach combined with the need for applying ambient intelligence access control e.g. based on the proximity of the IoT device and the interacting party, directly implies a requirement for not only simultaneously supporting multiple authentication and authorization schemes but, also, supporting authentication and authorization methods chaining.

Indeed, the devices access control module acts as a security and policy management gateway between the IoT devices (or, better, their corresponding digital twins), on one hand, and the various services (inherent to IoT-NGIN or not) and users that want to interact with them, on the other. Evidently, this interaction may refer to either simple monitoring (e.g. query the status of the IoT device, retrieve its details or measurements) or actuation upon them (e.g. force a status change).

In any case, the devices access control module is implemented as a flexible Ingress Gateway (effectively implemented as a standard API gateway enforcing diverse/chained access control methods) as depicted in Figure 27. In IoT-NGIN, the *kong* open-source API gateway [70] will be employed as an Ingress/API Gateway, backed up by a simple (PostgreSQL) database cluster mostly used to keep track of the API gateway configured routes, services and upstreams. Figure 27 presents the high-level architecture of the devices access control module.

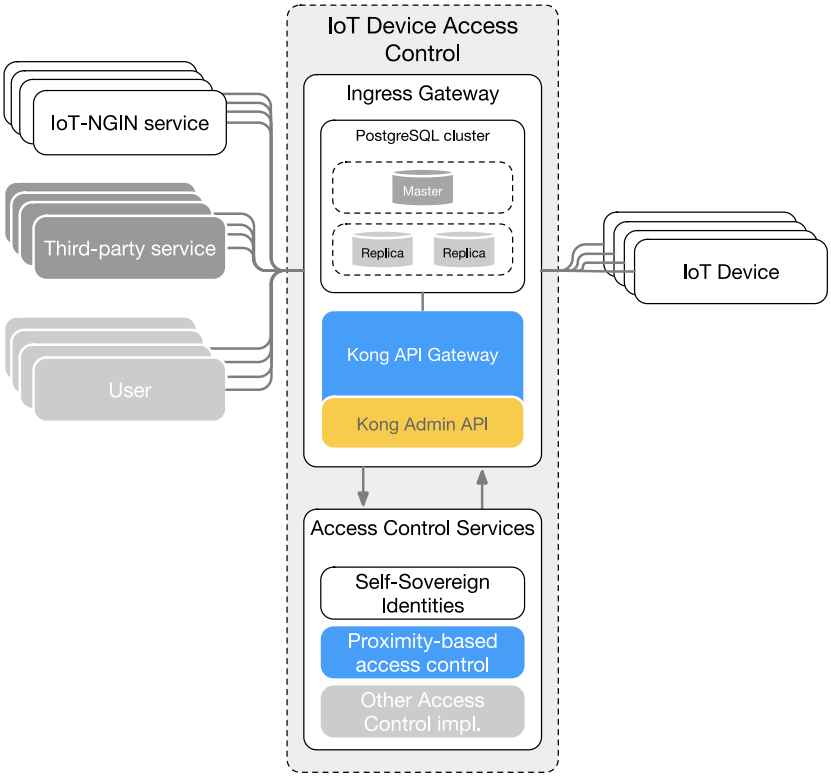

Figure 27: High-level architecture of the Devices Access Control module.

According to Figure 27, every request performed by an IoT-NGIN or third-party service or user, will be, first, evaluated by the gateway (see [71] for a comprehensive guide on how kong is architected, configured and administered) against a set of authentication and authorization plugins. In case this evaluation indicates that the calling user/service is authorized to use the IoT device resource (monitoring or actuation service), then permission will be granted and the request will be forwarded to the IoT device (or, better, its digital twin). In the context of IoT-NGIN, the authentication and authorization services exposed by the Self-Sovereign Identity services of the project (see deliverable D5.3 [72] for details and relevant discussion of the relevant exposed services) are going to be used as primary access control mechanisms. At the same time, other popular access control mechanisms (e.g. HMAC, JWT, Basic Auth, LDAP or OAuth2.0) may be supported.

Finally, in the context of the Ambient Intelligence access control concept, a proximity-based access control plugin will be implemented, effectively evaluating a request's validity by comparing the distance between a user and an IoT device with a pre-defined distance that would indicate acceptable proximity, from an access control point of view.

4.3.2 Interfaces

The module does not feature a single interface for exposing its services but, rather, exposes the interfaces of the underlying Authentication and Authorization services it employs.

4.4 AR/VR module

As discussed in Section 2.1.4, AR and MR become promising methods to visualize data from a rich variety of IoT sensors, while keeping the focus on the associated real scenarios, and even to interact and actuate such sensors.

Table 4 in Section 3 has provided an overview of the living labs and use cases that envision the use of AR for IoT interaction: Smart Agriculture IoT (UCs #4 #5), Employee Friendly Industry 4.0 (UCs #6 #7 #8) and Smart Energy (UCs #10). Each of these related use cases have been described with further details in the subsequent subsections of Section 3.

As many of the Living Labs and Use Cases require, or rely on, AR functionalities to achieve their targeted goals, it is necessary to firstly perform an analysis and categorization of state-of-the-art AR frameworks, tools and hardware solutions (e.g., mobile devices, headsets) to provide support for the envisioned scenarios and features. In addition, in order to overcome limitations of existing solutions and to fully adhere to the IoT-NGIN technological contributions, it is necessary to augment current AR solutions with additional software modules and APIs to effectively interact with the other related components and modules of the IoT-NGIN platform described in previous sections, namely: IoT Device Discovery (IDD), IoT Device Indexing (IDI) and IoT Device Access Control (IDAC). The preliminarily envisioned dependences and interactions between these components / modules are sketched in the high-level architecture diagram from Figure 12. In addition, the previous subsections of Section 4 provide tables and descriptions of the functionalities to be provided by each of these related modules, including the associated data and attributes and their types, the protocols to use for communication, associated databases and interfaces, etc.

4.4.1 Description

An analysis of the IoT-NGIN platform architecture diagram (Figure 12) and of the aforementioned tables from Section 4 for each of the WP4 components / modules provides a good starting point for the design of the required AR solutions for the Living Labs, and potentially for the Open Calls.

For doing so, relevant initial analysis tasks include, but are not limited to:

- Analysis of the type of scenarios and environments in which to apply AR (indoor, outdoor, lighting, size…)
- Analysis of potential restrictions or requirements regarding the use of specific hardware (e.g., detection cameras, fixed screens on which to present AR, mobile devices, AR headsets…). This will also determine the data that can be obtained from the AR presentation devices (e.g. its embedded sensors, like GPS, Wi-Fi interfaces, orientation sensors…) and the data that can be presented using them.
- Analysis of the types and number of sensors with which to interact and actuate.
- Analysis of the types and amount of information to display / present
- Analysis of the interaction mechanisms with other hardware components that will provide, or assist in, the discovery of IoT sensors (e.g., additional video cameras).
- Analysis of the required protocols and messages to receive notifications from IoT Device Discovery (IDD) modules.
- Analysis of the required protocols and messages to check the associate authorizations to retrieve data from sensors with the IoT Device Access Control (IDAC) modules.
- Analysis of the required protocols and messages to retrieve the information from the IoT Device Indexing (IDI), by also recognizing what sensor(s) the information relates to.
- Analysis of the required protocols and messages to actuate the IoT sensors via AR tools / gestures, and to reflect the updated information on the associated Digital Twins.

A task force will be initiated, in collaboration with the Living Lab owners, to determine the requirements, needs and desires for each of the above aspects, which will also determine the AR framework and hardware devices to select, as well as the extra developments needed to fulfil the requirements for the envisioned use cases.

4.4.2 Interfaces

Table 25: IoT AR module interfaces

IoT AR Module		
Description	The module will enable the IoT-NGIN architecture with the capabilities to present the information from related IoT sensors (e.g., the surrounding or requested ones) via AR and even to actuate them.	
Provided Interfaces	Presentation	
	Description	Interfaces to present information from associated IoT sensors, based on notifications from the IoT Discovery module

	End-point URL	To be agreed with the associate REST component
	Protocol used	Potentially HTTP, but depending on selected IoT Framework and database
	Methods	At least GET
	Message	TBD
	Actuation	
	Description	Interfaces to update information of associated IoT sensors, based on information retrieve from the IoT Indexing modules
	End-point URL	To be agreed with the associate REST component, and with the associated database, or middleware.
	Protocol used	Potentially HTTP, but depending on selected IoT Framework and database
	Methods	At least [POST/PUT/ DELETE and GET], to confirm (REST), but to be agreed upon the selection of sensors, databased and IoT frameworks
	Message	TBD
	Discovery	
	Description	In some cases, the IoT sensors may be discovered thanks to image processing or other detection methods directly provided by the AR devices. This interface will provide basic features to enable this.
	End-point URL	TBD, in coordination with T4.2 (IoT Discovery modules)
	Protocol used	TBD, in coordination with T4.2 (IoT Discovery modules)
	Methods	TBD, in coordination with T4.2 (IoT Discovery modules)
Required Interfaces	To be agreed with the associated IoT-NGIN modules, but potentially REST API for interaction (information retrieval and update)	

4.5 IoT Device actuationion

As discussed along the deliverable, IoT-NGIN envisions not only information retrieval from a wide variety of IoT sensors, but also their actuation to modify their status, behaviour, or any other related and relevant information. For doing so, it is also necessary to devise the

appropriate interfaces, protocols, messages and formats to be used, in an aligned manner with the other associated components and modules of the IoT-NGIN platform.

Even though different and effective actuation methods can be provided, like the use of either native or web-based dashboards and apps, it is also possible to actuate the sensors via AR. The related requirements and aspects to be determined for achieving these purposed via AR interfaces have been identified (although not yet determined) in the previous subsection (4.4) just to provide a unified view of all requirements related to AR.

5 Integration with the IoT-NGIN architecture

IoT-NGIN supports the edge computing paradigm, allowing the exploitation of distributed computing resources which are close to the data generation. The high-level architecture of IoT-NGIN, presented in deliverable document D1.2 "IoT meta-architecture, components, and benchmarking" [73] and depicted in Figure 28, indicates the distribution of components between the IoT devices and the -more capable- edge/cloud nodes.

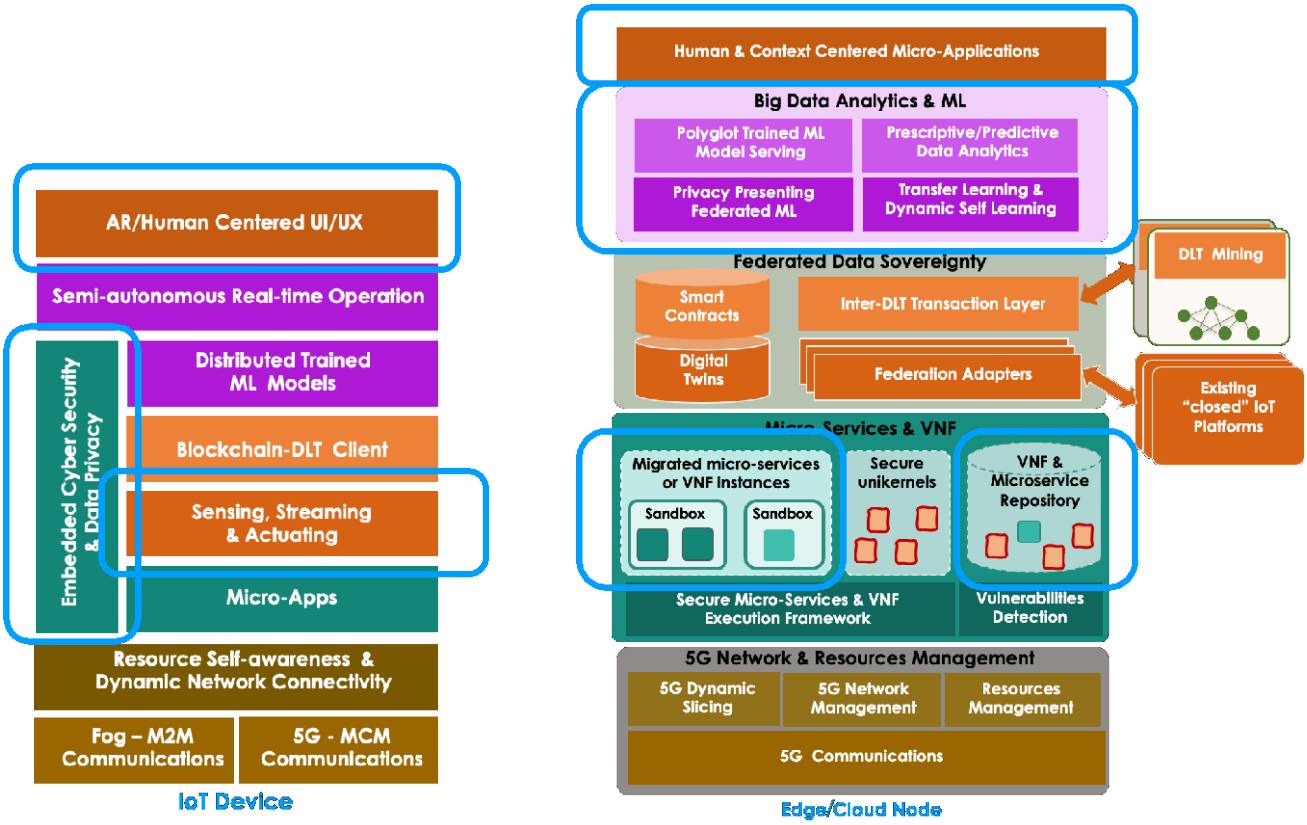

Figure 28: The high-level architecture of IoT-NGIN.

The logical architecture of the IoT-NGIN AmI tools is fully compatible with the high-level architecture of IoT-NGIN. The architectural building blocks which host the AmI tools of IoT-NGIN are highlighted in blue in the figure. In more detail:

- The AmI based *UC application* is on top of both nodes, namely the "AR/Human centered UI/UX" for the IoT device and the "Human & Context centred micro-application" for the edge/cloud node.
- The *IoT Device Discovery (IDD)*, *IoT Device Indexing (IDI)* and *IoT Device Access Control (IDAC)* services are understood as microservices or VNF instances in the relevant block of the edge/cloud node. Moreover, a counterpart of the IDAC service lies on the device to prevent unauthorized (direct) access, as part of the "Embedded cybersecurity & data privacy" components.
- Any ML functionality supporting the *IDD* service is provided by the "Big Data & ML" components of the edge/cloud node.

- The *IoT AR service (IAR)* is implemented using tools of the *IoT AR assets' repository*, which is included in the "VNF and Microservice Repository" of the edge/cloud node.
- The *IoT Device Actuation (IDA)* is realized as a component of "Sensing, streaming & actuating" on the IoT device, as well as a microservice on the edge/cloud node.

As a result, the AmI tools have a significant presence in both the IoT device and the edge/cloud nodes, which indicates the crucial role of AmI in future IoT systems, and also as an essential step towards the tactile internet. Moreover, the mapping of IoT-NGIN AmI tools to the high-level architecture of the project indicate the path towards developing use cases incorporating AmI and integrating it with other IoT-NGIN functionalities.

6 Conclusions

This document summarizes the work performed to enhance IoT Devices discovery, recognition and indexing and to improve pervasive security and ambient intelligence based control. It also identifies the initial steps on AR enhanced personalized IoT sensing and actuating. In this respect, it can be considered a first version that will be updated and completed in deliverables D4.3 and D4.4.

This initial version is devoted to the identification of the IoT Ambient Intelligence requirements of the living labs and to the definition and design of the necessary components of the architecture to fulfil the requirements.

Concretely, an initial description and design of the Device Discovery, Device Access Control and Device Indexing modules, their interfaces and their interaction has been provided. These modules are being deployed to validate its functionalities focusing on the IoT-NGIN use cases. Also, a preliminary description of AR and device actuation functionalities is provided.

Finally, the document serves as a reference of the functionalities, tools and Ambient IoT Enhancements implemented in WP4 and of how these features will be integrated in the IoT-NGIN architecture and can interact with the rest of services, tools and components developed in other work packages (e.g. ML and digital twins).

As next steps, the project will continue the tasks for the implementation of the modules for enhanced IoT Ambient Intelligence and their interfaces to be integrated with the IoT-architecture. Also, it will develop further enhancements for IoT device discovery, indexing, access control, AR and actuation focusing on the project living labs and use cases.

7 References

[1] Ducatel, K., Union européenne. Technologies de la société de l'information, Union européenne. Institut d'études de prospectives technologiques, & Union européenne. Société de l'information conviviale, Scenarios for ambient intelligence in 2010., Office for official publications of the European Communities Luxembourg, 2001.

[2] D. J. Cook, J. C. Augusto and V. R. Jakkula, "Ambient intelligence: Technologies, applications, and opportunities," *Pervasive and Mobile Computing,* vol. 5, no. 4, pp. 277-298, 2009.

[3] G. Fettweis, H. Boche, T. Wiegand, E. Zielinski, H. Schotten, P. Merz, S. Hirche, A. Festag, W. Häffner and M. Meyer, "The tactile internet-itu-t technology watch report," *Int. Telecom. Union (ITU), Geneva,* 2014.

[4] R. Szeliski, Computer Vision: Algorithms and Applicaitons, Texts in Computer Science, 2011.

[5] G. Images, "Google Images," Alphabet, 2001. [Online]. Available: https://images.google.com/.

[6] C. Papageorgiou, A general framework for object detection, Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271), 1998.

[7] S. a. M. Albawi, Understanding of a convolutional neural network, 2017 International Conference on Engineering and Technology (ICET), 2017.

[8] A. a. T. Singh, A review of supervised machine learning algorithms, 2016.

[9] P. V. M. J. Jones, Robust Real-time Object Detection, Cambridge Research Laboratory, 2001.

[10] Wikipedia, "Haar-like feature," [Online]. Available: https://en.wikipedia.org/wiki/Haar-like_feature#cite_note-Viola_2001-1.

[11] R. A. M. L. Auria, "Support Vector Machines (SVM) as a Technique for Solvency Analysis," *DIW Berlin Discussion Paper Nr. 811,* 2008.

[12] R. G. a. J. Donahue, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014.

[13] J. R. a. S. Divvala, "You Only Look Once: Unified, Real-Time Object Detection," 2016.

[14] M. M. Tsung-Yi Lin, Microsoft COCO: Common Objects in Context, 2015.

[15] W. Liu, "SSD: Single Shot MultiBox Detector," *Springer International Publishing,* pp. 21-37, 2016.

[16] "What is GNSS technology," [Online]. Available: https://www.euspa.europa.eu/european-space/eu-space-programme/what-gnss.

[17] "IEEE, 802.11-2016 (Revision of IEEE Std 802.11-2012), "Telecommunications and information exchange between systems Local and metropolitan area networks. Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Spe," 2016.

[18] M. Woolley, "Bluetooth Core Specification V5.1. Feature Overview. Version 1.0.1," 2020.

[19] IEEE, "IEEE 802.15.4a-2007 - IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Net," IEEE, 2007.

[20] IEEE, "IEEE 802.15.4-2011 - IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," IEEE, 2011.

[21] IEEE, "IEEE 802.15.4z-2020 - IEEE Standard for Low-Rate Wireless Networks--Amendment 1: Enhanced Ultra Wideband (UWB) Physical Layers (PHYs) and Associated Ranging Techniques," IEEE, 2020.

[22] ETSI, "ETSI EN 302 065-2 V2.1.1 (2016-11). Short Range Devices (SRD) using Ultra Wide Band technology (UWB); Harmonised Standard covering the essential requirements of article 3.2 of the Directive 2014/53/EU; Part 2: Requirements for UWB location tracking," 2016.

[23] Apple Developer Documentation, "Nearby Interaction," 2020. [Online]. Available: https://developer.apple.com/documentation/nearbyinteraction.

[24] M. Stone, "What is ultra-wideband, and how does it work?," 2021. [Online]. Available: https://insights.samsung.com/2021/08/25/what-is-ultra-wideband-and-how-does-it-work-3/.

[25] B. press, "What's the deal with Ultra Wideband?," 2021. [Online]. Available: https://www.bmw.com/en/innovation/bmw-digital-key-plus-ultra-wideband.html.

[26] Decawave, "APS003 Application Note. Real Time Location Systems. An Introduction," 2014.

[27] Y. H. C. Z. G. N. B. H. Willy Anugrah Cahyadi, "Optical Camera Communications: Principles, Modulations, Potential and Challenges," *MDPI Electronics,* p. 44, 2020.

[28] L. H. L. Q. J. Y. e. a. Yuan Zhuang, "A survey of Positioning Systems Using Visual LED Lights," *IEEE Communications Surveys & Tutorials,* vol. 20, 2018.

[29] Gartner Inc., "Hype Cycle for Identity and Access Management Technologies, 2020," 16 Jul. 2020. [Online]. Available: https://www.gartner.com/en/documents/3987655/hype-cycle-for-identity-and-access-management-technologi. [Accessed Nov. 2021].

[30] European Union Agency For Network And Information Security, "ENISA Good practices for IoT and Smart Infrastructures Tool," [Online]. Available: https://www.enisa.europa.eu/topics/iot-and-smart-infrastructures/iot/good-practices-for-iot-and-smart-infrastructures-tool/results#IoT. [Accessed Nov. 2021].

[31] European Union Agency For Network And Information Security, "Baseline Security Recommendations for IoT in the context of Critical Information Infrastructure," Nov. 2017. [Online]. Available: file:///Users/artem/Downloads/WP2017%20O-1-1-2%201%20Baseline%20Security%20Recommendations%20for%20IoT%20in%20the%20context%20of%20CII_FINAL.pdf. [Accessed Nov. 2021].

[32] Keyfactor, "The Top IoT Authentication Methods and Options," 29 Sep. 2020. [Online]. Available: https://www.keyfactor.com/blog/the-top-iot-authentication-methods-and-options/. [Accessed Nov. 2021].

[33] "RFC 5280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," , . [Online]. Available: http://tools.ietf.org/html/rfc5280#section-5.2.4. [Accessed 9 11 2021].

[34] M. Benjillali, 9 10 2017. [Online]. Available: https://www.itu.int/en/ITU-D/Regional-Presence/ArabStates/Documents/events/2017/IoTSMW/Presentations-IoT/Session1/IoT4SSC_Session_1_Benjillali.pdf.

[35] J. Nadeem, S. Arshad, N. Hina and G. Nadra, "Intelligence in IoT-Based 5G Networks: Opportunities and Challenges," *IEEE Communications Magazine* , vol. 56, no. 10, pp. 94-100, 2018.

[36] M. Attaran, "The impact of 5G on the evolution of intelligent automation," *Journal of Ambient Intelligence and Humanized Computing* , 2021.

[37] NOKIA, "Critical IoT vs. Massive IoT: How to spot the difference | Nokia," [Online]. Available: https://www.nokia.com/networks/insights/critical-massive-iot/.

[38] Qualcomm, *Ultra-Reliable Low-Latency 5G for Industrial Automation,* Qualcomm.

[39] D. Sabella, "MEC Standards on Edge Platforms." Multi-access Edge Computing: Software Development at the Network Edge," *Springer,* pp. 59-87, 2021.

[40] e. a. A. Reznik, "ETSI WhitePaper nr. 30. MEC in an Enterprise Setting: A Solution Outline," 2018. [Online]. Available:

https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp30_MEC_Enterprise_FINAL.pdf.

[41] Y. Z. L. X. L. Bing, "An MEC and NFV integrated network architecture," *ZTE COMMUNICATIONS,* 2017.

[42] Synelixis, "SynField," [Online]. Available: https://www.synfield.gr/about/.

[43] Unity Technologies, "Unuity Augmented reality," 2021, [Online]. Available: https://unity.com/unity/features/ar.

[44] ITU, "The Tactile Internet - ITU-T Technology Watch Report," 2014. [Online]. Available: https://www.itu.int/dms_pub/itu-t/oth/23/01/T23010000230001PDFE.pdf.

[45] O. Holland, E. Steinbach, R. V. Prasad, Q. Liu, Z. Dawy, A. Aijaz, N. Pappas, K. Chandra, V. S. Rao, S. Oteafy, M. Eid, M. Luden, A. Bhardwaj, X. Liu, J. Sachs and J. Araújo, "The IEEE 1918.1 "Tactile Internet" Standards Working Group and its Standards," *Proceedings of the IEEE,* vol. 107, no. 2, pp. 256-279, 2019.

[46] S. U. P. U. Stanford Vision Lab, "ImageNet," 2020 . [Online]. Available: https://image-net.org/.

[47] C. Fellbaum, "WordNet and wordnets," 2005. [Online]. Available: https://wordnet.princeton.edu/.

[48] a. A. A. Quattoni, Recognizing Indoor Scenes, 2009.

[49] D. Bolya, "TIDE: A General Toolbox for Identifying Object Detection Errors," in *Computer Vision -- ECCV 2020*, 2020, pp. 558--573.

[50] European Standards, "BS ISO/IEC 16388:2007 Information technology — Automatic identification and data capture techniques — Code 39 bar code symbology specification," 2007. [Online]. Available: https://www.en-standard.eu/bs-iso-iec-16388-2007-information-technology-automatic-identification-and-data-capture-techniques-code-39-bar-code-symbology-specification/. [Accessed 2021].

[51] European Standards, "ISO/IEC 15417 Information technology — Automatic identification and data capture techniques — Code 128 bar code symbology specification," 2007. [Online]. Available: https://www.en-standard.eu/iso-iec-15417-information-technology-automatic-identification-and-data-capture-techniques-code-128-bar-code-symbology-specification/. [Accessed 2021].

[52] European Standards, "CSN EN 12323 AIDC technologies - Symbology specifications - Code 16K," 2005, [Online]. Available: https://www.en-standard.eu/csn-en-12323-aidc-technologies-symbology-specifications-code-16k/. [Accessed 2021].

[53] European Standards, "BS ISO/IEC 15438:2015 Information technology — Automatic identification and data capture techniques — PDF417 bar code symbology specification," 2015. [Online]. Available: https://www.en-standard.eu/bs-iso-iec-

15438-2015-information-technology-automatic-identification-and-data-capture-techniques-pdf417-bar-code-symbology-specification/. [Accessed 2021].

[54] European Standards, "BS ISO/IEC 18004:2015 Information technology — Automatic identification and data capture techniques — QR Code bar code symbology specification," 2015. [Online]. Available: https://www.en-standard.eu/bs-iso-iec-18004-2015-information-technology-automatic-identification-and-data-capture-techniques-qr-code-bar-code-symbology-specification/. [Accessed 2021].

[55] European Standards, "BS ISO/IEC 24778:2008 Information technology — Automatic identification and data capture techniques — Aztec Code bar code symbology specification," 2008. [Online]. Available: https://www.en-standard.eu/bs-iso-iec-24778-2008-information-technology-automatic-identification-and-data-capture-techniques-aztec-code-bar-code-symbology-specification/. [Accessed 2021].

[56] European Standards, "ISO/IEC 16022:2006 Information technology — Automatic identification and data capture techniques — Data Matrix bar code symbology specification," 2006. [Online]. Available: https://www.en-standard.eu/iso-iec-16022-information-technology-automatic-identification-and-data-capture-techniques-data-matrix-bar-code-symbology-specification/. [Accessed 2021].

[57] European Standards, "ISO/IEC DIS 23634 Information technology — Automatic identification and data capture techniques — JAB Code polychrome bar code symbology specification," [Online]. Available: https://www.en-standard.eu/iso-iec-dis-23634-information-technology-automatic-identification-and-data-capture-techniques-jab-code-polychrome-bar-code-symbology-specification/. [Accessed 2021].

[58] GitHub, "jabcode," 2021. [Online]. Available: https://github.com/jabcode/jabcode. [Accessed 2021].

[59] H2020 IoT-NGIN consortium, "Deliverable D5.1: Enhancing IoT Cybersecurity," 2021.

[60] FIWARE, "FIWARE - Open APIs for open minds," [Online]. Available: https://www.fiware.org/. [Accessed 10 Nov. 2021].

[61] S. . Sotiriadis, K. . Stravoskoufos and E. G. M. Petrakis, "Future Internet Systems Design and Implementation: Cloud and IoT Services Based on IoT-A and FIWARE," , 2017. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-44924-1_10. [Accessed 10 11 2021].

[62] FIWARE, "Build your own IoT platform with FIWARE enablers," 27 Mar. 2015. [Online]. Available: https://www.fiware.org/2015/03/27/build-your-own-iot-platform-with-fiware-enablers/. [Accessed 09 Nov. 2021].

[63] "KNOWAGE AND NGSI," [Online]. Available: https://knowage.readthedocs.io/en/6.1.1/user/NGSI/README/index.html. [Accessed 10 Nov. 2021].

[64] FIWARE, "FIWARE-NGSI v2 Specification 1.0," [Online]. Available: https://swagger.lab.fiware.org/. [Accessed 10 Nov. 2021].

[65] FIWARE, "Welcome to Orion Context Broker," [Online]. Available: https://fiware-orion.rtfd.io. [Accessed 10 Nov. 2021].

[66] FIWARE, "FIWARE Academy: IoT Agents," [Online]. Available: https://fiware-academy.readthedocs.io/en/latest/iot-agents/idas/index.html. [Accessed 10 Nov. 2021].

[67] FIWARE, "FIWARE IoTAgent Node Lib - Architecture," [Online]. Available: https://iotagent-node-lib.readthedocs.io/en/latest/architecture/index.html. [Accessed 09 Nov. 2021].

[68] FIWARE, "IoT Agent API," [Online]. Available: https://iotagent-node-lib.readthedocs.io/en/latest/api/index.html. [Accessed 10 Nov. 2021].

[69] FIWARE, "FIWARE NGSI APIv2 Walkthrough," [Online]. Available: https://github.com/telefonicaid/fiware-orion/blob/master/doc/manuals/user/walkthrough_apiv2.md. [Accessed 10 Nov. 2021].

[70] Kong Inc., "Kong Gateway," 2021. [Online]. Available: https://docs.konghq.com/gateway/. [Accessed Nov. 2021].

[71] Kong Inc., "Comprehensive Getting Started Guide," 2021. [Online]. Available: https://docs.konghq.com/gateway/2.6.x/get-started/comprehensive/. [Accessed Nov. 2021].

[72] H2020 IoT-NGIN Consortium, "Deliverable D5.3: Enhancing IoT Data Privacy & Trust," 2021.

[73] IoT-NGIN, "D1.2 - IoT meta-architecture, components, and benchmarking," H2020 957246 - IoT-NGIN Deliverable Report, 2021.

[74] IoT-NGIN, "D9.1 - Project Handbook," H2020-957246 IoT-NGIN Deliverable Report, 2020.

[75] "RTLS technology comparison," [Online]. Available: https://www.sewio.net/uwb-technology/rtls-technology-comparison/.

[77] G. Fettweis, H. Boche, T. Wiegand, E. Zielinski, H. Schotten, P. Merz, S. Hirche, A. Festag, W. Häffner and M. Meyer, "The tactile internet-itu-t technology watch report," *Int. Telecom. Union (ITU), Geneva,* 2014.

[78] FIWARE, "Tutorials: Custom IoT Agent," 19 Oct. 2021. [Online]. Available: https://github.com/FIWARE/tutorials.Custom-IoT-Agent. [Accessed 09 Nov. 2021].