DECENTER

Decentralised technologies
for orchestrated Cloud-to-Edge intelligence

D2.2

Final release of the DECENTER architecture specification and use cases

13/08/2020

Revision history

Administration and summary		
Project acronym:		DECENTER
Document identifier:		D2.2: Final release of the DECENTER architecture specification and use cases
Leading partner:		UL
Report version:		2.2
Report preparation date:		13.08.2020
Classification:		PU (Public Use)
Nature:		R (Report)
Author(s) and contributors:		Petar Kochovski, Vlado Stankovski, Janez Brežnik, Uroš Paščinski (UL), Andrea Leveghi (TN), Jaewon Moon, Seungwoo Kum (KETI), Silvio Cretti, Roberto Doriguzzi Corin (FBK), Guillem Gari (ROB) Ana Belen Gonzalez Mendez (ATOS), Sofia Kleisarchaki (KENTYOU)
Status:	-	Plan
	-	Draft
	-	Working
	X	Final

The DECENTER Consortium has addressed all comments received, making changes as necessary. Changes to this document are detailed in the change log table below.

Date	Edited by	Status	Changes made
20.02.2020	Petar Kochovski	Plan	Prepare Table of Contents
06.03.2020	Petar Kochovski	Working	Added UC3 description.
11.03.2020	Petar Kochovski	Working	Work on Section 3.
11.03.2020	Ana Belen Gonzalez Mendez	Working	Work on Section 5.
07.04.2020	Jaewon Moon	Working	Added UC4.
08.04.2020	Andrea Leveghi	Working	Added UC1 description.
15.04.2020	Silvio Cretti	Working	Added architecture description.
13.05.2020	Janez Brežnik	Working	Integrating partners' contributions.
26.05.2020	Guillem Gari	Working	Added UC2 description.
11.06.2020	Roberto Doriguzzi Corin	Working	Completed description of the architecture.
16.06.2020	Sofia Kleisarchaki	Working	Added Section 2.1.
16.06.2020	Ana Belen Gonzales Mendez	Working	Added initial business models per partners
17.06.2020	Uroš Paščinski	Working	Integrated partners' contributions
24.06.2020	Roberto Doriguzzi Corin	Working	Internal review
26.06.2020	Jaewon Moon	Working	Internal Review
28.06.2020	Janez Brežnik	Working	Improving document formatting
29.06.2020	Sofia Kleisarchaki	Working	Addressing comments in Section 2
09.07.2020	Ana Belén González	Working	Improvement of section 5 and subsection 5.3.2.3 added. Internal review comments of Section 5 addressed.
09.07.2020	Petar Kochovski	Working	Addressing internal review comments.
10.07.2020	Vlado Stankovski	Final	Finalisation of the deliverable.
13.08.2020	Vlado Stankovski	Final	Final remarks from the coordinator and the consortium have been addressed.

Notice that other documents may supersede this document. A list of latest *public* DECENTER deliverables can be found at the DECENTER Web page at https://www.decenter-project.eu/.

Copyright

Acknowledgements

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no. 815141 (DECENTER: Decentralised technologies for orchestrated Cloud-to-Edge intelligence)

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 1711075689, Decentralised cloud technologies for edge/IoT integration in support of AI applications).

The partners in the project are FONDAZIONE BRUNO KESSLER (FBK), ATOS (ATOS), COMMISSARIAT À L'ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES (CEA), COMUNE DI TRENTO (TN), ROBOTNIK (ROB), UNIVERZA V LJUBLJANI (UL), KOREA ELECTRONICS TECHNOLOGY INSTITUTE (KETI), GLUESYS (GLSYS), DALIWORKS (DW), LG U+ (LGUP), SEOUL NATIONAL UNIVERSITY (SNU).

The content of this document is the result of extensive discussions within the DECENTER © Consortium as a whole.

More information

Public DECENTER reports and other information pertaining to the project are available through DECENTER public Web site under http://www.decenter-project.eu.

Table of Contents

Executive Summary

Cloud computing as a key enablement technology involved in recent years from technologies that are used in single and heterogeneous data centres towards technologies that can be used to achieve virtualisation across the whole computing spectrum from existing data centres towards the Edge of the network. Fog computing as a new term encompasses a new computing concept that uses virtualisation across very dynamic and heterogeneous infrastructures across administrative domains and aims to address the necessary Quality of Service, dependability, privacy, security and other high-level requirements of advanced smart application scenarios. This development comes hands in hands with other technology trends such as the Internet of Things that are nowadays capable of streaming enormous amount of complex-structured data towards the computing infrastructures. The core processing requirements emerge from advanced Artificial Intelligence algorithms, such as Deep Learning Neural Networks and Digital Twins that almost always operate on very sensitive and private data that must be handled with great precision and care. The DECENTER project deals with many challenges where all these technologies meet (1) with the necessity to deliver network, memory and compute intensive Artificial Intelligence algorithms closer to the data sources, which requires the decentralisation of the Artificial Intelligence processes, on one side, and (2) with the necessity to adapt the core cloud computing services and protocols, such as the software engineering tools and the orchestration mechanisms in order to provide for that level of necessary dynamicity, dependability, privacy, security and trust, which are of enormous importance when a consortium, such as DECENTER has to provide an industry-strength Fog computing platform, (3) the decentralisation necessarily needs a gluing factor that would assist all participating entities in an ecosystem perform business together based on mutual dependability and trust. While the achievement of these high-level goals will most certainly remain to be research and innovation objective of more Horizon 2020 and Horizon Europe projects, DECENTER can be viewed as the first necessary step towards this direction that paves directions for future research and innovation.
Core part of DECENTER WP2 activities has been to develop our new visionary Fog Computing and Brokerage Platform architecture, which is presented in the present deliverable. With D2.2 our intention has been to provide a concise and up to the point rationale of the overall DECENTER architecture. As such, D2.2 is designed to complement (and not supersede) deliverable D2.1.

In particular, this deliverable provides the updates and finalized descriptions of our activities in relation to the design of the DECENTER use cases, Service Level Agreements' (SLAs) models and Resource Models (RMs), as well as the final version of the platform architecture. Furthermore, this deliverable describes the initial outputs from task T2.3 that started at the beginning of the second year.

The aforementioned activities resulted in four fundamental outcomes:

1. Complete definition of the four Use Cases. The deliverable provides a detailed description of each Use Case. Also, the deliverable highlights the requirements to the DECENTER platform that each use case has and describes in what way it benefits from the platform;
2. Outcome of the second year on Resource Models (RMs) and Service-Level Agreements (SLAs). It presents the final design and implementation applied to the use cases;

3. Design of the final DECENTER architecture. This deliverable provides the design of each component of the architecture;
4. Initial output of our business approach in the context of federated Cloud-to-Edge environments. The deliverable describes the output from the Business Model Workshop and the initial analysis of the platform business goals and models.

1. Introduction

The DECENTER project use cases pose significant requirements for the development of a new Fog Computing and Brokerage Platform.

The requirements for the Fog Computing and Brokerage platform can be summarised in the following few functional directions: (1) ability to flexibly combine Artificial Intelligence algorithms, methods and services in complex multitier smart applications with greatly varying QoS, dependability, privacy and security requirements (ease of use), (2) the ability to deploy and orchestrate such complex multitier smart applications across the Cloud-to-Edge computing continuum, (3) the ability to provide means for engagement of various participants in an ecosystem, such as software engineers, cloud infrastructure providers, and even, AI service developers, for example, by means of new Resource Models for the characterisation of resources and a blockchain-based Brokerage Platform. In non-functional terms the greatest focus has been on the provision of (1) algorithms and methods for the management of QoS, (2) Service Agreement Management, and (3) the development of mechanisms for cross-border data management.

In order to achieve these high-level goals of the DECENTER architecture, it is necessary that activities within WP3, which develops core methods of the Fog Computing and Brokerage Platform and WP4, which engineers the AI processes, so that they can be stretched from the Cloud towards the Edge of the network, while at the same time provide necessary AI-related properties (precision, recall, learning, privacy, security, Digital Twins). Hence, the main objectives of the DECENTER architecture are currently collectively addressed by both WP3 and WP4.

The whole purpose of D2.2 has been to provide a crystalized architecture that can be understood by computer scientists and engineers in technical terms, while the more philosophical, scientific and technical outcomes and experiences of this work will be disseminated towards the research audience in a DECENTER architecture-related paper, which is currently under preparation.

Deliverable D2.2 summarizes the work done within Work Package 2 (WP2) from month 13 until month 24 of the project. This document complements deliverable D2.1 [1] with the lessons learned from the implementation activities. In particular D2.2 provides: a detailed description and the requirements of the Use Cases (UC) that are used in the DECENTER project; representative implementation of the Service Level Agreements (SLAs) and Resource Models (RMs) for the UCs; a detailed description of the DECENTER's architecture; and preliminary business goals analysis of the platform and the UCs. The remaining of this document is as follows.

Section 2 presents the four use cases that are considered in DECENTER, which are: smart city crossing safety, robotic logistics, smart and safe construction, and ambient intelligence for safety at home. It reports our analysis of the requirements of the use cases and the platform, that led us to our architecture and system design. In addition, it describes the latest updates of each use case, and benefits obtained by using the DECENTER platform.

Section 3 updates the specification of Service quality Level Objectives (SLOs) for the DECENTER use cases. Moreover, it presents our final scheme for resource modelling of the main resources involved in the platform and their relationship to the SLAs. In addition, this section provides a description of the RMs and SLA models applied to each of our UCs.

Section 4 provides the DECENTER's architecture and the detailed design of the system components. The architecture is presented in three layers: Infrastructure, Platform and

Application layer. The components of each layer are described by providing information such as: subsystem they belong to, interfaces they provide and require, as well as description of their role in the system.

Section 5 delivers the initial business approach for federated Cloud-to-Edge environments. It analyses and describes the process for the establishment of the DECENTER Platform Business Model. In addition, this section describes the DECENTER Platform business goals and business models of the DECENTER new knowledge exploitation. Finally, it identifies the preliminary business models that are the outcomes of the Business Model Workshop and enlists our next steps that will be described in detail in D2.3.

2. DECENTER Use Cases

2.1 Use Cases

This section describes the four UCs used in DECENTER that exemplify the needs of real-life users of the platform from different yet complementary contexts.

The preliminary definition and description of the identified UCs is provided in D2.1 as part of our Y1 activities. Below are descriptions of the updates applied to the UCs during Y2 of the project, including final UC description and process workflow, it links the UCs requirements to DECENTER's innovation and it describes the benefits from using the DECENTER's platform.

2.1.1 Smart City Crossing Safety

2.1.1.1 Description

As reported in the deliverable D2.1 the objective of this use case is to increase the pedestrian crossing safety by leveraging the IoT and edge infrastructures of a smart city. This approach focuses on equipping a number of pedestrian crossings with devices existing on the market that enable monitoring of pedestrians intending to cross the road.

Following this path, we prepared a pedestrian crosswalk in Vela (suburb of Trento) with new poles, wires, cabinets, pipes and light signals to host all the IoT devices necessary to our use case.

The objective is to increase pedestrian crossing safety using DECENTER features:

- Vertical resource orchestration: for solving the problem of latency
- AI Model repository: to use AI services from pretrained model
- Hierarchical /distributed AI: to distribute the AI on edge or cloud
- Multi-tier fog computing platform: used to deploy the service in cloud or edge through QoS and SLA
- Privacy preserving AI to fuzzy sensible image and protect personal data
- Digital twin: to create a digital replica of the pedestrian crosswalk and communicate it to a dashboard

The first part of our goal consists in creating a pilot in one area, testing it and then extending this solution to other pedestrian crosswalks.

Won't be possible to test all the DECENTER features in this first part of the project but only those related with the platform and necessary that allow us to test the benefit of DECENTER.The pilot road infrastructure has been almost done (road excavation, poles and wires has been settled). Other devices will be installed after an internal test of the system that is ongoing in FBK's lab.

Currently the devices, necessary to our use case, have been prepared and their connectivity tested. They are able to communicate between them and work with a minimal number of AI microservices and recognize pedestrians, cycles, cars and vans.

For the end of year 3 these devices will be able to interact with the DECENTER platform and work fully (using all the features).

For year 2 a Minimal Viable Demo (MVD) is prepared. This demo allows us to test the functionality of all the parts of the pilot (image detection, sound detection, env. sensors data

analysis) and test the benefit of DECENTER, moving services toward edge or cloud to improve the latency:

1. Test
    a. Detect a vehicle approaching the crossing
    b. Detect a pedestrian crossing the street
    c. Get data from environment sensors
    d. Actuate one or more alerting device
2. Demo storyboard
    a. Start all application services
    b. Configure a Video AI service and start it (use recorded video stream as input source)
    c. Show generated events from UI and the video feed
    d. Open Rode-Ned and:
        i. set-up and deploy a new rule
    e. On new event light/buzzer should be triggered based on road status (show event and response time on UI)
    f. Decrease the application response time requirement
    g. The platform should adapt and move services (toward the edge)
    h. Check new performances

The process workflow is described in Figure 1.

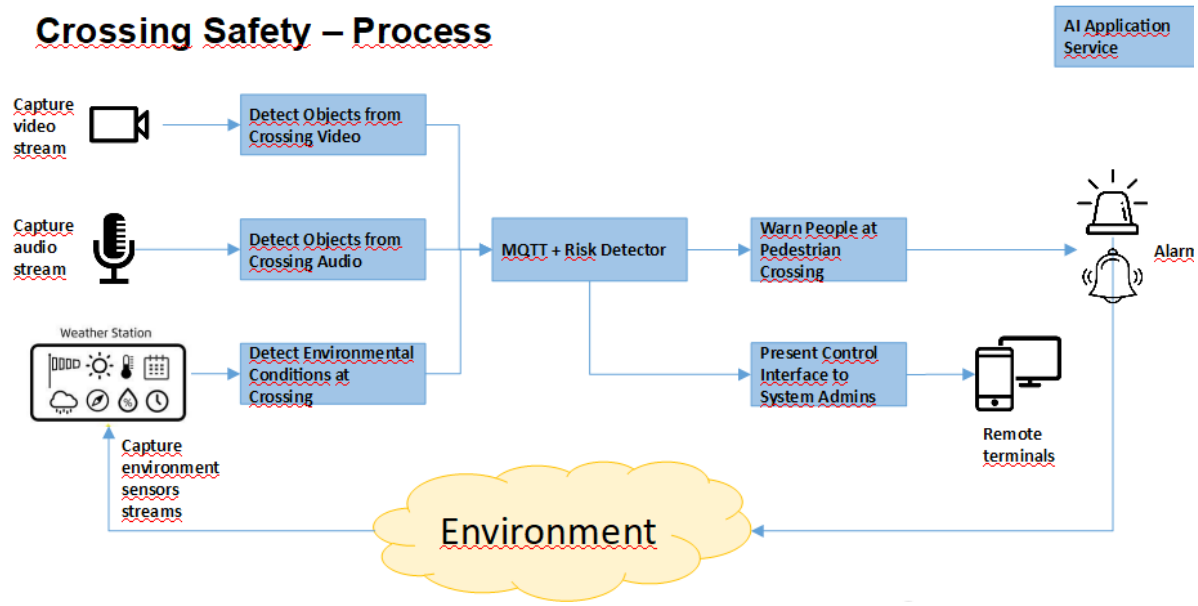

Figure 1. Smart City Crossing Safety workflow

The IoT devices collect data that will be sent through the MQTT broker at the risk detector. The risk detector, using digital twin, analyses the data creating a road status interpretation and if a risk is detected it will generate an alert. Also, one of its purposes, it's to send information to a control interface that will be able to show the data collected and the status of the pedestrian crosswalk.

For any of the AI applications services shown in Figure 1 there are one or more microservices associated with.

This table details the purpose of each AI application service.

AI Application	AI application service	Activity	Function
Crossing Safety	Detect Environmental Conditions at Pedestrian Crossing	Analysis	Detection of detailed environmental conditions (e.g. day/night, visibility, temperature, humidity, frost, etc.) in a pedestrian crossing from nearby IoT sensors
Crossing Safety	Detect Objects from Pedestrian Crossing Video	Analysis	Detect key objects (e.g. pedestrian, cyclist, vehicle, etc.) approaching the pedestrian crossing from video signal
Crossing Safety	Detect Objects from Pedestrian Crossing Audio	Analysis	Detect key objects (e.g. pedestrian, cyclist, vehicle, etc.) approaching the pedestrian crossing from audio signal
Crossing Safety	Determine Dangerous Situation at Pedestrian Crossing	Decision	Rule-based inference to determine dangerous situations given environmental conditions and the presence of certain objects and people.
Crossing Safety	Warn People at Pedestrian Crossing	Action	In case of a dangerous situation, send of sound and light signals to the people involved
Crossing Safety	Present Control Interface to System Admins	Action	A control interface (it may be a terminal interface) for the application or system administrator.

Table 1. AI applications services used in DECENTER UC-1

2.1.1.2 Requirements from the platform

Requirements from the DECENTER platform:

- Ability to deploy multiple existing microservices depending on the service scenario (FR04, FR06-FR12, FR18-FR24, FR13-FR16);
- Ability to use multiple existing AI models for recognize video stream, sound or environment data (FR37, FR38);
- Ability to communicate to each microservice (F25, F27);
- Ability to use the QoS-aware orchestration technology that respects the quality constraints set by the user (FR-10-12, FR-31-34).

2.1.1.3 Benefits from the platform

Benefits from the DECENTER platform:

- QoS-aware orchestration of resources (i.e. DECENTER Fog Platform);
- Use of AI model repository to manage AI model for an AI microservice (i.e. Model Manager and Model Repository components);
- Build an end-to-end AI service with a combination of existing AI models (i.e. Application Composer, Model Manager and Model Repository components).

2.1.2 Robotics Logistics

2.1.2.1 Description

Automated logistics using mobile robots is a very innovative field where in the last years the data that a robot can obtain from the environment has increased highly. The data processing sometimes is beyond the computational capabilities of the robot. Also new robotic algorithms or artificial intelligence applied to the field requires high computational capabilities, so in the real world most of them could not be applied due to the robot's limitations.

The goal of this UC is to improve the responsiveness of robots in the real warehouse environment using the DECENTER platform. There are some key points that will be improved by using this approach:

- *Safety:* By improving the responsiveness of the robot the interactions with the humans, the robot will be less prone to failure and will make a safer environment for warehouse workers.
- *Better computational efficiency:* DECENTER will be able to provide a decrease of the use of the computational resources, so the robot will have a better battery life, more efficient routes and will have room to implement more advanced algorithms.

DECENTER provides the ability to place resources outside the robot effortless. we can use existing AI packages provided by DECENTER to identify objects. DECENTER allow to deploy them over an edge server, an idle robot, or the cloud without modifying the robot fleet configuration. Even though DECENTER allows reallocating of resources dynamically on Realtime improving the resource usage. Without DECENTER the robotic fleet could not benefit of services that require a high computational cost, like AI object recognition.

2.1.2.2 UC process workflow

The current objective deploys DECENTER on the robot fleet and adds artificial intelligence application (AIA) in order to improve robot route and workers safety.

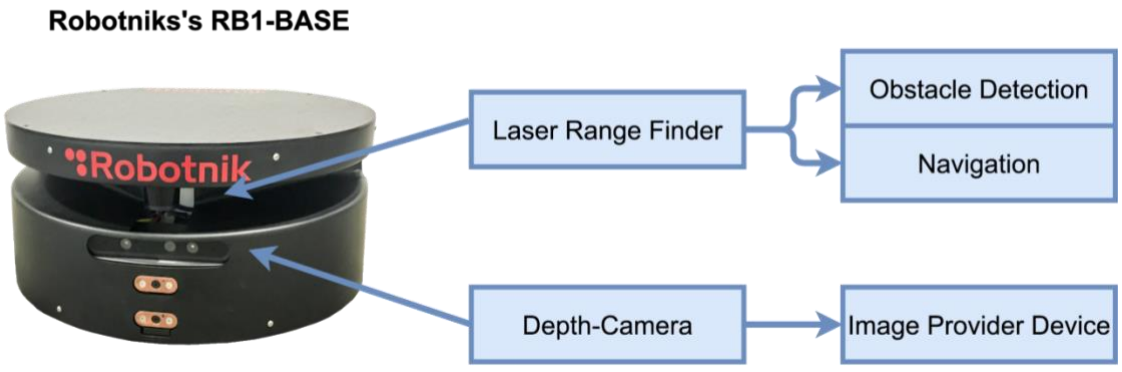

Figure 2. Robot sensors and process diagram

The robot that is used in this use case is Robotnik's RB1-BASE. As depicted in Figure 2, the robot is equipped with a depth camera and laser range finder, which are responsible to detect obstacles on the robot's way and take pictures of the obstacle.

Figure 3 depicts the communication process between three main entities, which are: robots (i.e. red squares), AIA (i.e. green squares) and Fleet management System (FMS) (i.e. blue squares). The FMS will assign missions to the robots and the robot will send updates. The AIA receives images from the robots and sends information to FMS.

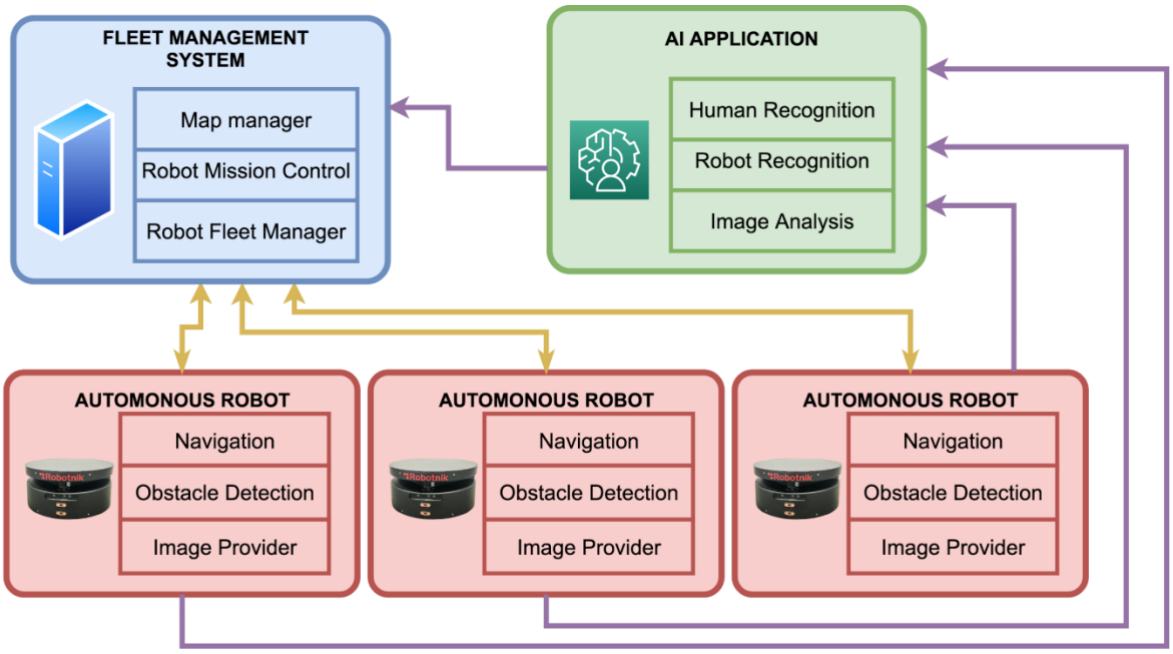

Figure 3. Communication and process diagram

The detailed workflow of the UC2 is depicted on Figure 4. In particular when the robot is moving and it detects an obstacle the front camera will take a picture of it to determine the type of the object (e.g. human/robot/other). The AIA will inform the Fleet Management System (FMS) which will update the robot behaviour (i.e. send visual and audible warnings, reroute, or wait).

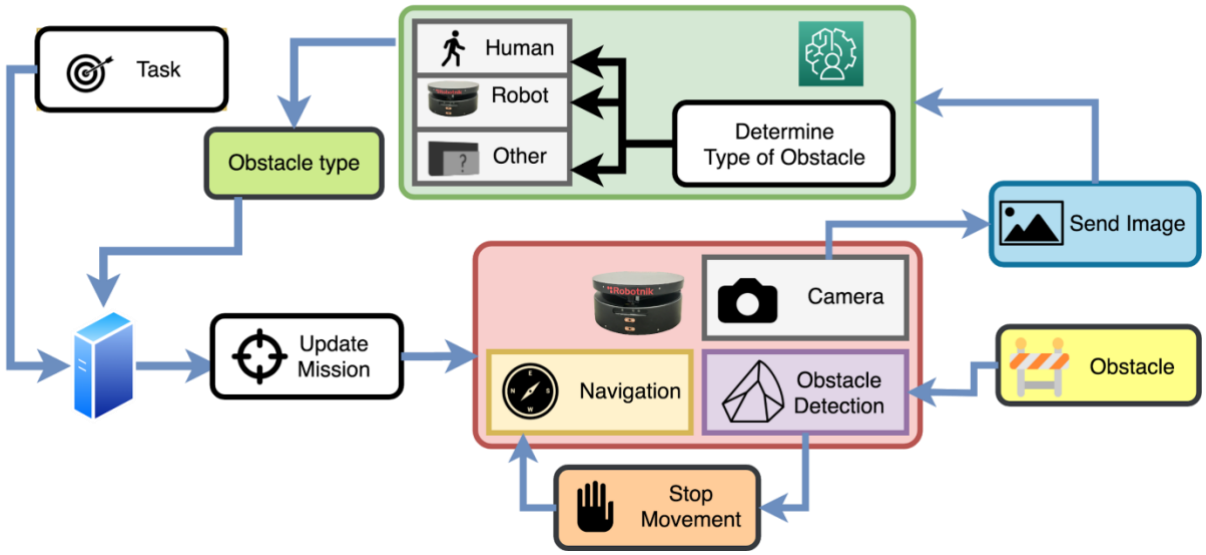

Figure 4. Robotics logistics workflow

The process workflow is as follows:

1. User assign task to the Fleet Management System;
2. The Fleet Manager Systems assigns a mission to the Robot;

3. The Robot starts the mission (Navigation);
4. The Robot finds an unavoidable obstacle (by Laser Range Finder);
5. The Robot stops movement (Safety);
6. The Robot takes a Picture;
7. The Robot sends to Edge Fog Device the image;
8. The Edge Fog Device analyses the image with an AI application;
9. The Edge Fog Device determines if the obstacle is either Human, another robot, or other;
10. The Edge Fog Device sends the obstacle type (Human/Robot/other) to Fleet Management System;
11. The Fleet Management System decides the new course of action (modify route/ Alert and wait / Mark as blocked alley, etc);
12. The Fleet Manager System sends the mission's updates to the Robot;
13. Go to step 3 until the goal is reached or the mission is cancelled.

2.1.2.3 Requirements from the platform

Requirements from the DECENTER platform:

- Place computational resources outside the robot to edge or the cloud and the orchestration of them. (FR-21, FR-26, FR-31)
- All the microservices deployed on the robots/edge/cloud will be containerized and the images are available on containers' repository (FR-18, FR-29)
- The use of different AI models and training sets to identify obstacles using the robot cameras. The existing models and trainings sets could be fetched from model repository (FR-36, FR-37)

2.1.2.4 Benefits from the platform

Benefits from the DECENTER platform:

- The ability to use different AI Packages and Models to detect accurate persons (and robots) and fast switch them without the need of local retraining (i.e. Model Manager and Model Repository components)
- The ability to offload the heavy process like to the cloud or edge and automatic switch (i.e. DECENTER Fog Platform)
- Use of digital twin application for the robot management for the users (i.e. Digital Twin)
- Use of the custom deployment algorithms of the microservices in order to improve the computational efficiency (i.e. DECENTER Fog Platform)

2.1.3 Smart and Safe Construction

2.1.3.1 Description

Construction is a dynamic process that requires constant information support. As a result, organising, monitoring, and implementing a construction project including its various safety, security, logistics, inspection and other aspects can be very challenging.

The goal of this UC is to address safety at smart construction sites. It improves safety by issuing notifications to the construction site manager when a safety violation is detected. In order to do so, the construction site is under constant video surveillance whose streaming data

is fed to AI methods for data processing. Before arriving at the current final scenario, four different scenarios were considered: vehicle identification, identification of people wearing personal protective equipment, identification of dangerous conditions, and counting waste and supply.

In order to fully exploit the benefits of the platform, the final scenario utilises AI methods that are supposed to perform two operations: 1) object detection (i.e. vehicle detection) and 2) member verification (i.e. detection of persons). In this scenario there is a separate container hosting a different AI method for each operation. Therefore, any vehicle that appears at the construction site will be detected. In case the vehicle is not permitted to enter that location, the construction site manager will be notified about the violation. Furthermore, if a person is detected at the construction site, a member verification analysis will take place in order to check if the person is a member of the group of people allowed at that location.

Throughout the analysis and research of the scenario, it had to be modified multiple times in order to achieve higher performance. For instance, initially it was planned to pull the AI models to a specific container that will continuously run an AI method on a host. However, this had to be changed, because the AI models required different AI methods. The second plan was to consecutively start/stop containers that contain different AI methods and pull the AI models each time. However, this approach was taking too much time, because the pull operation of the AI model could take up to several minutes, which could result in a late response in a real case. Finally, optimal results were achieved when the AI model pull operation was avoided by utilising containers that were composed of the AI method and AI model.

The final version of UC's workflow is composed of 12 consecutive steps (see Figure 5) and they are as follow:

1. The user (i.e. construction manager) uses DECENTER User Interface to select and AI model-1 (i.e. object detection) and a video streaming source from the available cameras at the construction site; The user also defines the QoS requirements (i.e. SLOs) and agrees upon the SLA agreement;
2. The Trusted Model Manager (TMM) requests access via the Blockchain Service (BS) to the AI method containing the chosen AI model;
3. BS triggers execution of Smart Contract (SC), which may also communicate with a Smart Oracle (SO) in order to obtain access to off-chain data, such as AI model regulations and policies;
4. Once the SC executes successfully, TMM requests the AI method from Data/Access Manager.
5. Data/Access Manager verifies the BC transaction. In case the verification is successful, Data/Access Manager grants access for TMM to access the AI method repository;
6. TMM pulls the AI method and utilises DECENTER Fog Platform to determine an optimal deployment option to run the method.
7. The container running AI model 1 begins continuously to receive a video stream from the camera, process the data and send context results to the Message Broker.
8. TMM continuously receives messages from the Message Broker and checks if they contain information on truck detection.
9. Once a message containing information of truck detection reaches TMM, it immediately notifies the construction manager and initiates the deployment of an AI method container containing AI model 2 for member verification.

10. The deployment of the second AI method is executed following steps 3-8. Once AI model 2 begins processing and streaming data to the Message Broker, the container running AI model-1 is stopped.
11. Each time a person passes in front of the camera, AI model 2 verifies if the person is a member of the group of workers allowed in that section of the construction site. If a violation is detected, TMM immediately notifies the construction manager.

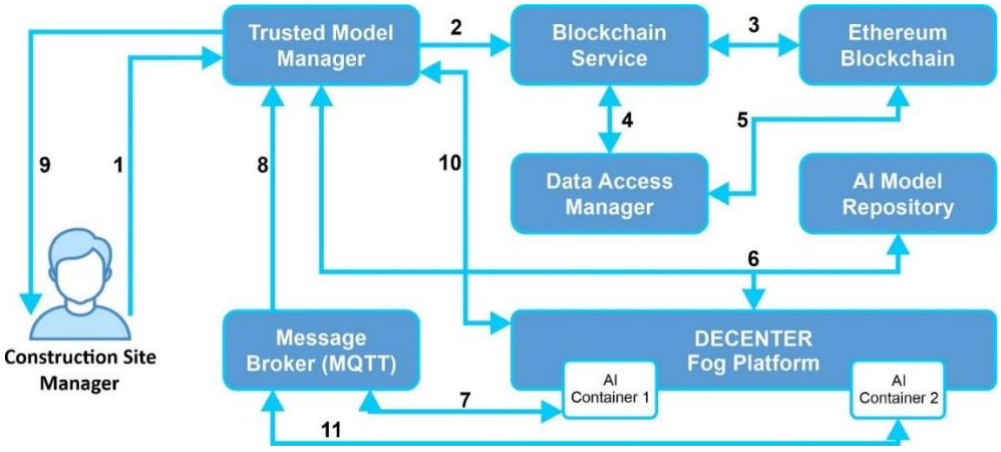

Figure 5. Sequence diagram of the orchestration process within the Smart and Safe Construction scenario

2.1.3.2 Requirements from the platform

Requirements from the DECENTER platform:

- Ability to use multiple existing AI models to recognize video stream, sound or environment data. The AI existing models will be fetched from an AI model repository (FR-36, FR-37).
- Ability to use the QoS-aware orchestration technology that respects the quality constraints set by the user (FR-10-12, FR-31-34).
- Cross-border data management technology that respects state regulations and certifications on privacy (FR-04, FR-05, FR-35).

2.1.3.3 Benefits from the platform

Benefits from the DECENTER platform:

- QoS-aware orchestration of resources (i.e. DECENTER Fog Platform)
- Select an AI model from the repository that is suitable for specific case (i.e. Model Manager and Model Repository components)
- Use Smart Contracts for accessing AI models in order to support privacy preservation through regulations and certification (i.e. Data Management subsystem)

Due to the dynamic nature of the construction site, which results with a high amount of alterations within time, resulting with a completely different scene from the time of development to the time of final demonstration, the intended scenario will be demonstrated by using pre-recorded video from a construction site in Slovenia.

2.1.4 Ambient intelligence for office environments[1]

2.1.4.1 Description

Recently, a lot of IoT devices have been deployed and used for a real-time safety monitoring. The current level of Indoor safety services is limited to monitoring status using devices such as cameras and cell phones. Therefore, it is not easy to respond promptly and appropriately to various situations without continuous supervision. In addition, most services are usually provided on cloud platforms, which can cause problems due to round trip delays. Privacy issues may also arise while uploading personal data such as images to the public cloud for analysis.

This use case will show the key features of DECENTER based on an AI application providing indoor ambient intelligent service. The detailed service scenario is as follows:

- This application checks the face of users visiting a certain space and verifies whether the person is authorised to consume certain content in that space or not. It will integrate two verifiers that can verify each group member: A group verifier, and B group verifier.
- We assume that only these two groups (A and B) are targeting to see specific content. A general content will be provided to visitors who do not belong to both Group A and Group B.
- Example: DECENTER members from Korea and Europe plan to meet in Trento. At this meeting, the organizer wants to provide important information such as meeting location and schedule. However, this information is useless for other Trento visitors. DECENTER can provide this service without additional personal information on edge devices.

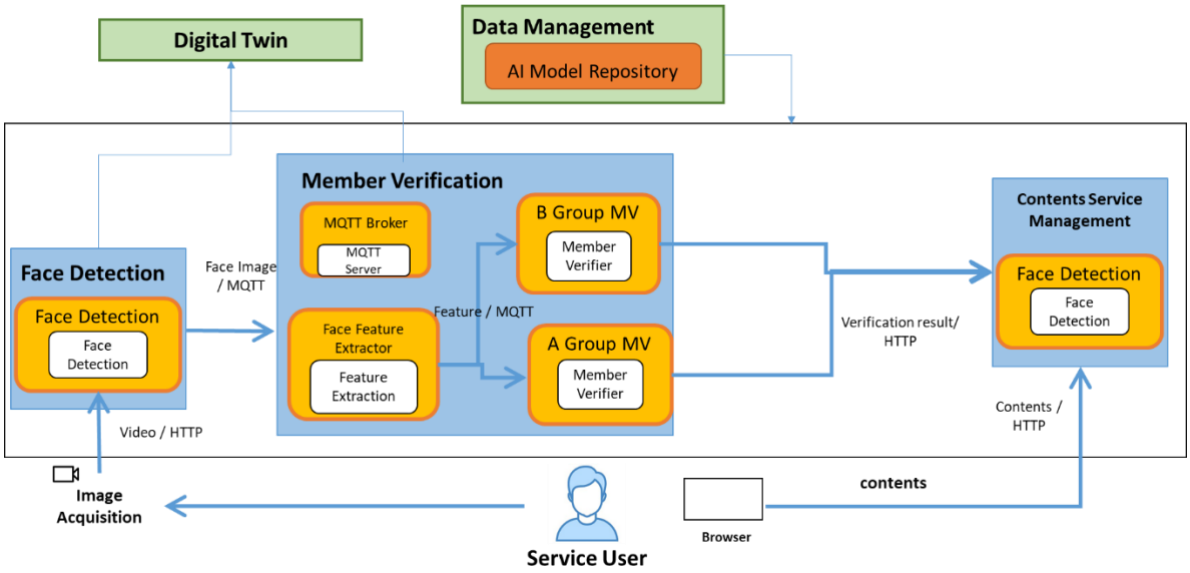

Figure 6. Data transfer between each microservice

[1] *Initially, three small scenarios were presented in UC4. However, given the revised and improved WPs' goals, a realistic use-case was set to provide the benefit of DECENTER. The second scenario was modified and extended and is presented as UC4. The name of UC4 has been changed from "Ambient intelligence for indoor safety" to "Ambient intelligence for office environments" to better express its meaning.*

This use-case in DECENTER does not collect any personal information. The DECENTER infrastructure needs to appropriately distribute resources and each micro-service only needs to get the proper AI model from the AI model repository. No personal information from users is stored in the cloud. It only manages pre-trained AI models. Through this use-case, it is possible to verify whether extended service can be easily provided without personal information.

Figure 6 shows the data transfer between each microservice of UC4. UC4-FC connected to the camera detects the face image of a visitor. UC4-FE extracts feature maps from the image input and UC4-MV verifies whether the visitor is a member or not. UC4-MV sends the result to the UC4-SC. Finally, UC4-SC provides appropriate content to visitors based on the final result. This table details the purpose of each AI application.

AI application service	**Activity kind**	**Function**
Face Detection (UC-FD)	Analysis	UC4-FD detects a face in the image from the camera. When it detects the face, it crops the face area and transfers it to UC4-FE.
Feature Extractor (UC4-FE)	Analysis	UC4-FE extracts a feature vector from the detected face image. Each feature vector contains discriminative face representations.
Member Verifier (UC4-MV)	Decision	UC4-MV utilizes face feature vectors to determine it belongs to the member group or not. The output includes a confidence score.
Contents Service Management (UC4-CS)	Decision	UC4-CS decides what content to provide based on the results of UC4-MV and provides appropriate content to the service front end.

Table 2. AI applications used in DECENTER UC-4

2.1.4.2 Requirements from the platform

Requirements from the DECENTER platform:

- Ability to deploy multiple existing microservices depending on the service scenario (SR_UC4_001, NFR_UC4_006)
- Ability to use multiple existing AI models for member verification (NFR_UC4_005)
- Ability to communicate each microservice (SR_UC4_001)
- Ability to use the QoS-aware orchestration technology that respects the quality constraints set by the user (NFR_UC4_004)

2.1.4.3 Benefits from the platform

Benefits from the DECENTER platform:

- QoS-aware orchestration of resources (i.e. DECENTER Fog Platform)
- Use of AI model repository to manage AI model for an AI microservice(i.e. Model Manager and Model Repository components)
- Build a new service without transferring private data to the cloud (i.e. Model based member verification)
- Build an end-to-end AI service with combination of existing AI model (i.e. Model deployment of DECENTER Fog Platform)

AI models are stored in the AI model repository before being used in microservices. The service application needs to choose the pre-defined AI models based on use-case scenario.

2.2 Use Case Requirements

The following subsections provide information on our final analysis of functional, non-functional and system requirements, corresponding to the DECENTER use cases. The following list of requirements acts as a catalyst for the architectural design, our design and development decisions and the business perspective.

2.2.1 Functional Requirements

ID	Description	Comments on how the requirement is (or will be) addressed
FR_UC1_001	Detect if a pedestrian is approaching to the crossing	A camera is positioned to view any movement on the sidewalk
FR_UC1_002	Detect if a cyclist is approaching to the crossing	A camera is positioned to view any movement on the sidewalk. The requirement will be addressed in year 3
FR_UC1_003	Detect if a vehicle is approaching to the crossing	A camera is positioned to view any movement on the sidewalk
FR_UC1_004	Detect if a disabled person is approaching to the crossing	A camera is positioned to view any movement on the sidewalk. The requirement will be addressed in year 3
FR_UC1_005	Detect if a person with a pet is approaching to the crossing	A camera is positioned to view any movement on the sidewalk. The requirement will be addressed in year 3
FR_UC1_006	Detect if a person with stroller is approaching to the crossing	A camera is positioned to view any movement on the sidewalk. The requirement will be addressed in year 3

FR_UC1_007	Collect information with respect to weather conditions such as temperature, humidity, rain, light, wind	A weather station will be installed near the crossing
FR_UC1_008	Elaborate a model of the current situation at the crossing	See FR_UC1_001, FR_UC1_002, FR_UC1_003, FR_UC1_004 and also obtain a model of the crossing
FR_UC1_009	Trigger actuators in case of dangerous situation (i.e. when a RED alert is ON)	It is tested on a simulation mock-up (based on crossing recorded video / IoT sensing)
FR_UC1_010	Trigger a sound alarm in case of dangerous situation	At least 1 speaker must be installed
FR_UC1_011	Flashing lights in case of dangerous situation (i.e. when a RED alert is ON)	At least 1 flashing light has to be installed
FR_UC1_012	Record/store data	A cloud backup is needed
FR_UC1_013	Improve its detection of possible dangerous situation	Periodically redeploy based on the logs
FR_UC1_014	Collect live data-streams from different sources / devices	Connected devices produce data streams fed to the alerting system
FR_UC2_001	Trigger a sound or visual alarms	At least 1 speaker or led lights must be installed in robots
FR_UC2_002	Allow the user to add missions	A graphical control interface is built for human interaction
FR_UC2_003	Allow the user to assign tasks to robots	A graphical control interface is built for human interaction

FR_UC2_004	Keep an updated map of the warehouse	A global knowledge about the environment at robots level. Communication between robots. Sensors to get information from the environment.
FR_UC2_005	Detect persons	Camera and AI methods to detect humans.
FR_UC2_006	Ensure the correct relation between providers through the use of smart contracts	The system will use safe and stables channels to use smart contracts
FR_UC2_007	Get and show the status of robots	A graphical control interface is designed
FR_UC2_008	Keep track of tasks	A graphical control interface is designed
FR_UC2_010	Provide statistical information	A graphical interface is needed to display the information
FR_UC3_001	Access and use existing pre-trained AI database models	Certain pre-trained AI models already exist, thus it is possible to retrieve them from a database/repository and use them. Currently the DECENTER AI Model Repository is being used.
FR_UC3_002	Train and additionally customize existing AI models (transfer learning)	Pre-trained model for identification of specific objects such as trucks already exists (e.g. AlexNet or YOLO), which are being used. It would be also possible to additionally customize the model, for example, to identify helmet in addition to a hat.
FR_UC3_003	Receive and process data from a video camera	Receive and ingest a video stream, and extract individual frames for processing and analysis
FR_UC3_004	Place bounding boxes at specific (interesting) images parts (object detection)	A specific algorithm for image segmentation is used, which results in specific bounding boxes of image segments.

FR_UC3_005	Trigger notifications for construction site engineer	All features identified will be sent as notifications to the mobile phone of the construction site manager, if certain conditions are met. (Rule based system for notifications will be implemented). Example: If a specific object is detected a notification will be generated.
FR_UC4_001	Extract features from images and detect moving objects	Edge will have an object detector, which is able to produce a feature map as an input of another engine.
FR_UC4_002	Have the ability to change video data input to image format and change the image data input to an interpretable size and shape by AI engine	The system will have an image pre-processor based on use-case scenario.
FR_UC4_003	Check whether the detected human has entered the restricted area	The system has a human detector which can detect the access of people in a specific space.
FR_UC4_004	Detect whether a visitor is an authorized member or not	The system has a member verifier based on registered images.
FR_UC4_005	Detect a face in the given input	The system has a face detector which detects the location of a face from the given input.
FR_UC4_006	Predict the future value of environmental factors	The system will have Indoor environment predictors, which can estimate the future value of Indoor factors (e.g., CO2, PM10).
FR_UC4_007	Store IoT sensor data (e.g., PM10) based on specific use-case scenario	Data storage will retain data for a specific period of time based on use-case scenario.

FR_UC4_008	Trigger alerts when a potentially dangerous situation is perceived or an error on a task occurs	The system will be connected to the service application which can trigger alerts.

2.2.2 Non-Functional Requirements

ID	**Description**	**Comments on how to the requirement is (or will be) addressed**
NFR_UC1_001	Work outside and to work in different situations in terms of temperature (-15°C to 40°C) and weather	Proper casing design
NFR_UC1_002	Operate even if specific hardware components fail to respond	The realised solution is not expected to be part of a critical system in its prototypical implementation, therefore hardware redundancy will not be required for the implementation of the use-case
NFR_UC1_003	Be robust in terms of number of devices and cameras available	In order to address some problems with some devices/cameras, the idea is to have more than the normal number of cameras/devices installed
NFR_UC1_004	Ensure robustness from internet connectivity	Ability to deploy ML trained algorithms as containers “at the edge”
NFR_UC1_005	Be used without an additional effort from the user	Almost by design. No additional effort will be requested by pedestrians (i.e. no need to install an app, etc)
NFR_UC1_006 [2]	Trigger alert in less than 200 milliseconds	Dangerous situations happen in a few milliseconds. The focus of the reactivity is the pedestrians with respect to drivers

NFR_UC1_007	Ensure privacy	No data (video) will be recorded except for the preparation of a demo
NFR_UC1_009	Ensure an AI model that adequately reacts to dangerous situations	Training data set from site during 3 months window has to be recorded to ensure designed AI model can be tested against playback. Moreover, possibility of reinforcement learning must be in place for refining the model
NFR_UC1_010	Guarantee security of data storage and processing	Secured and authorised access to system will be implemented.
NFR_UC1_011	Easily access, control and configure existing sensors or register new ones, in order to ensure their connectivity to the platform.	All the devices are connected to the cluster network, they can be reached and configured by using the HTTP web interface or command line access
NFR_UC2_001	Provide privacy of information (how many workers are working at each moment, number of boxes stored, ...)	Privacy of information
NFR_UC2_002	Delegate computational calculation to the Cloud	Vertical offloading
NFR_UC2_003	Delegate computational calculation to the Edge	Horizontal offloading AI algorithms running in the Edge.
NFR_UC2_004	Be reliable	Horizontal offloading
NFR_UC2_005	Be safe for humans	Security sensors.

NFR_UC3_001	The smart application shall be able to use more or less video cameras and shall be reused in different layouts with respect to different construction sites.	Scaling the number of available Fog nodes by using Smart Contracts to adapt for increasing number of video cameras.
NFR_UC3_002	Keep private the information processed on each construction site.	Smart Contracts are specified for individual Fog Nodes in order to allow access to sensitive information and control the access.
NFR_UC3_003	Provide a processing time of object detection 30 seconds.	System must detect if a person is wearing a helmet before the person can reach restricted area from the entrance or the office of the construction site; in a distance of at least 4.0 m (the width of manipulation intervention road) with a presumption that average walking speed is 1.4m /s.
NFR_UC3_004	Have the possibility to operate even if specific Fog Node fails to respond on time.	The system will use Smart Contracts to find out additional Fog Nodes that can be used for the process.
NFR_UC3_005	Perform correctly in different temperature/illumination conditions	This shall be achieved through proper set up of the sensors and video cameras.
NFR_UC4_001	Have a reliable member verification	The accuracy of member verification will be evaluated on face retrieval protocol, and the accuracy will be at least 70%.
NFR_UC4_002	Have one or more predictors that can infer the indoor future conditions	This system will predict more than two indoor future conditions.

NFR_UC4_003	Have a method for evaluating the performance of the human detector	The performance of the human detector will be evaluated by drawing precision-recall curve on public benchmark.
NFR_UC4_004	Get the face detection result during a proper time	The time response is highly dependent on the capabilities of each edge. The system will respond in less than 1 minute for the service. This system does not deploy AI solutions for some specific edge devices that are expected to take a longer time.
NFR_UC4_005	Select the appropriate model for a specific scenario that will run at the edge	In the case of indoor environment prediction, either cloud or edge will be able to select the most proper model based on information from edge and transfer it to edge.
NFR_UC4_006	Transfer the result of the AI engine from one edge to other edge	Each AI engine will connect to another engine through a local network.

2.2.3 System Requirements

ID	**Description**	**Comments on how the requirement is (or will be) addressed**
SR_UC1_001	Have enough Computational Power	To run the planning and AI methods independently
SR_UC1_002	Have connectivity to the Internet	To offload computation if needed but especially to offload interesting datasets that might cause retraining of algorithms
SR_UC1_003	Have enough HD capacity	To store and process the data coming from various data sources

SR_UC1_004	Have enough GPU capacity	It is likely that GPU capacity will be needed though the target to minimize as much as possible the use of expensive hardware. GPU is no more necessary for object detection. Will be evaluated if it could be useful using GPU in cloud for testing purpose.
SR_UC1_005	Have virtualization capacities	To run on the DECENTER platform, the system will be virtualized and redeployable in containers. Thus, the processing unit should have virtualization capabilities and be optimized for container virtualization.
SR_UC2_001	Provide robots which work autonomously	The robots work independently from the system. Once an order is sent, they will perform the task without contacting the system unless an alarm is triggered
SR_UC2_002	Have access to GPU resources	The system is composed of several servers, implementing GPUs for image recognition training.
SR_UC2_003	Have enough computational power	The system shall have enough computational power to run the planning and AI methods independently
SR_UC2_004	Have connectivity to the Internet	The system shall have connectivity to the Internet to offload computation both vertical and horizontally
SR_UC2_005	Have enough HD Capacity	The system shall have enough HD capacity to keep track of all the interactions in a year. After the closure of a year, information is archived and only the most important data and statistical information are kept
SR_UC2_006	Have virtualization capacities	To run the DECENTER platform, the system hardware has to be able to virtualize. Thus, the processing unit should have virtualization capabilities and be optimized for container virtualization

SR_UC3_001	Have sufficient computing resources necessary to run the application	The system will deploy AI solutions on Fog Nodes with sufficient memory and processing power.
SR_UC3_002	Have enough HD capacity to store the data of at least of monthly operation of the site	Cloud storage will be used.
SR_UC3_004	Use specific number of cameras	We assume to use 3 cameras for a specific construction site.
SR_UC3_005	Use GPU to run AI functionalities	In order to achieve stringent time limits with respect to AI inference time, the system must be able to utilise hardware accelerators, such as GPUs and TPUs.
SR_UC3_006	Provide data access which shall be controlled	Smart Contracts shall be used in order to allow or deny access to data (e.g. video frames), metadata and AI models. Consensus based access to data and similar other techniques shall be explored.
SR_UC3_007	Have a stable internet connection	The physical infrastructure will be of sufficient quality in order to allow for testing.
SR_UC4_001	Have proper AI models for edge	Cloud platform will generate AI models with training by itself or use stored pre-trained AI models. Cloud will be able to transfer the AI model to edge platform.
SR_UC4_002	Use the input data for inference	Edge platform will be able to store and read the data in a format such as database or a file.

2.2.4 UCs and Platform Requirements

The synthetic table below aims at explaining how the technical requirements of the UCs led to our decisions on the architecture. Particularly, the table provides a summary of the UCs requirements that are common requirements and maps them to common platform-related requirements (see more details in Section 3 of deliverable D3.1) as well as to the targeted architectural components. In addition, the alignment between KPI and UC requirements can be found in D5.1.

Common UCs requirement	**Platform-related requirement**	**Architectural component**
Ensure the correct relation between providers using smart contracts FR_UC2_006 NFR_UC3_001 NFR_UC3_002 NFR_UC3_004	Interoperability FR-01, FR-02, FR-03, FR-04, FR-05	Fog Platform, Brokerage Platform
Ability to respond even if specific nodes are not operating and recover from failure NFR_UC2_004 NFR_UC3_004	Reliability and Autonomy FR-06, FR-07, FR-08, FR-09	Fog Platform
Addition or reduction of computing resources SR_UC1_001 SR_UC2_003 NFR_UC3_001	Hierarchy and Scalability FR-10, FR-11, FR-12	Fog Platform, Management
Safe data processing and storage, confidentiality, authentication, and access control NFR_UC1_010 NFR_UC2_005 NFR_UC1_007 NFR_UC2_001 NFR_UC3_002	Security FR-13, FR-14, FR-15, FR-16	Management / Authentication
The processing unit should have virtualization capabilities and be optimized for container virtualization	Virtualisation Support FR-17	Fog Platform, Resource Management

SR_UC1_005 SR_UC2_006	Containerization Support FR-18	
Ability to deploy applications on the fog and stable internet connection NFR_UC2_002 NFR_UC2_003 NFR_UC4_005 NFR_UC4_006 SR_UC1_002 SR_UC2_004 SR_UC3_007 SR_UC4_004	System Management FR-19, FR-20, FR-21, FR-22, FR-23, FR-24	Management (Provisioning, Deployment, Alerting) Fog Platform (Stability)
Selection of appropriate AI models and response in a reasonable time NFR_UC1_006 NFR_UC1_009 NFR_UC3_003 NFR_UC4_004 NFR_UC4_005	Orchestration FR-25, FR-26, FR-27	Management (Inter-fog and Infra-fog Orchestration) Fog Platform
Repository of available application services / applications (e.g., Digital Twin, AI models) with easy installation and use FR_UC3_001 FR_UC3_002 SR_UC4_001 SR_UC4_002	Application and Service Management FR-28, FR-29, FR-30	Management (Application Catalogue, Repository, Service)
Allocate and share resources SR_UC1_001, NFR_UC2_002, NFR_UC2_003, SR_UC2_003, NFR_UC3_001,	Resource Management FR-31, FR-32, FR-33, FR-34	Management (resource allocation, orchestration, sharing, REB)

SR_UC3_001, NFR_UC4_006		
Provide authorized users with access to the management system NFR_UC1_005, NFR_UC1_011, FR_UC2_002, FR_UC2_003	User Management FR-35	Management
Use, customize and distribute pre-trained AI models NFR_UC1_009, FR_UC3_001, FR_UC3_002, NFR_UC3_003, FR_UC3_004, FR_UC4_001, FR_UC4_003 - FR_UC4_005, NFR_UC4_005	AI function Support FR-36, FR-37	Fog Platform Management (Hybrid decentralized AI models)
Collect / store data from heterogeneous sources and trigger alerts FR_UC1_007, FR_UC1_009- FR_UC1_012, FR_UC1_014, FR_UC3_003, FR_UC4_006 - FR_UC4_008	IoT function Support FR-38, FR-39	Fog Platform

3. Resource Models & Service-Level Agreements

This section summarizes the outcome of the second year of activities on task T2.2, which is related to Resource Models (RMs) and Service-Level Agreements (SLAs). In particular, this section presents their final design and implementation applied to each of the use cases. The initial design of the RMs and SLAs was presented in D2.1.

An important aspect of every SLA is to define the involved parties (i.e. stakeholders) that are supposed to reach an agreement upon a certain service. In our case, there are four fundamental stakeholders: end-user, Application Service Provider, Infrastructure provider, independent Monitoring Provider.

The end-user as a stakeholder can demand a specific AI application that may be hosted on a computing infrastructure in his/her close proximity or remotely. Often, the end-user does not have the knowledge to define SLA requirements. Thus, the requirements are delegated to the Application Service Provider in a natural language format.

The Application Service Provider has the role to compose, deploy and maintain the AI application by using the DECENTER platform. When using the DECENTER platform, this stakeholder defines the application's business requirements more formally in a form of functional and non-functional requirements. Additionally, the Application Service Provider is responsible for delivering and installing on-site equipment that may be required by the application.

The DECENTER platform provides services that allow applications to be deployed and run on an infrastructure or a federation of infrastructures. In other words, this stakeholder provides means for composing and deployment of AI applications. In addition, the DECENTER platform participates in the SLA, accepts functional and non-functional requirements, and exposes monitoring services for application-specific metrics and infrastructure-level metrics.

The Monitoring Provider is responsible for collecting the metrics from exporters and different interfaces through which those metrics can be assessed. This stakeholder is also part of the SLA contract, but however it is not a main party but it is a supporting party.

Figure 7 depicts the stakeholders that participate in the process of signing the SLA agreement.

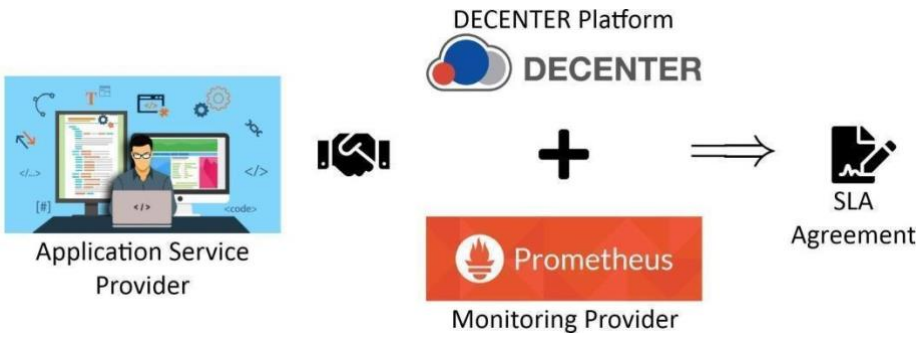

Figure 7. Stakeholders involved in SLA signing

In comparison to SLAs, which require at least two stakeholders to agree upon some terms, the RMs are done by a single stakeholder. In particular, the resource models are designed by the stakeholder that owns the hardware and wants to describe it. With respect to requirements and capabilities, an SLA would encompass the business requirements translated to some measurable assets, while resource models would express capabilities of some hardware resources.

The non-functional requirements can be classified into three types:

- **Service Preconditions**: define the region, which a computing pod should be deployed into and the tier (i.e. Cloud, Fog, Edge)
- **Resource Requests**: specify the capacity and types of requested resources and they can be seen as the resource requests in Kubernetes.
- **Service Level Objectives (SLO)**: QoS requirements to be met (i.e. values bounded by thresholds). Their respective metrics are monitored.

3.1 Use Case RM and SLA models

This subsection delivers examples of RM and SLA models implementation in each DECENTER use case.

3.1.1 Smart City Crossing Safety

The Application components are deployed as microservices on the DECENTER platform, as shown in Figure 8. Multiple instances of an AI service can be deployed if more than one data source is available (i.e. one Video AI service per video camera data source).

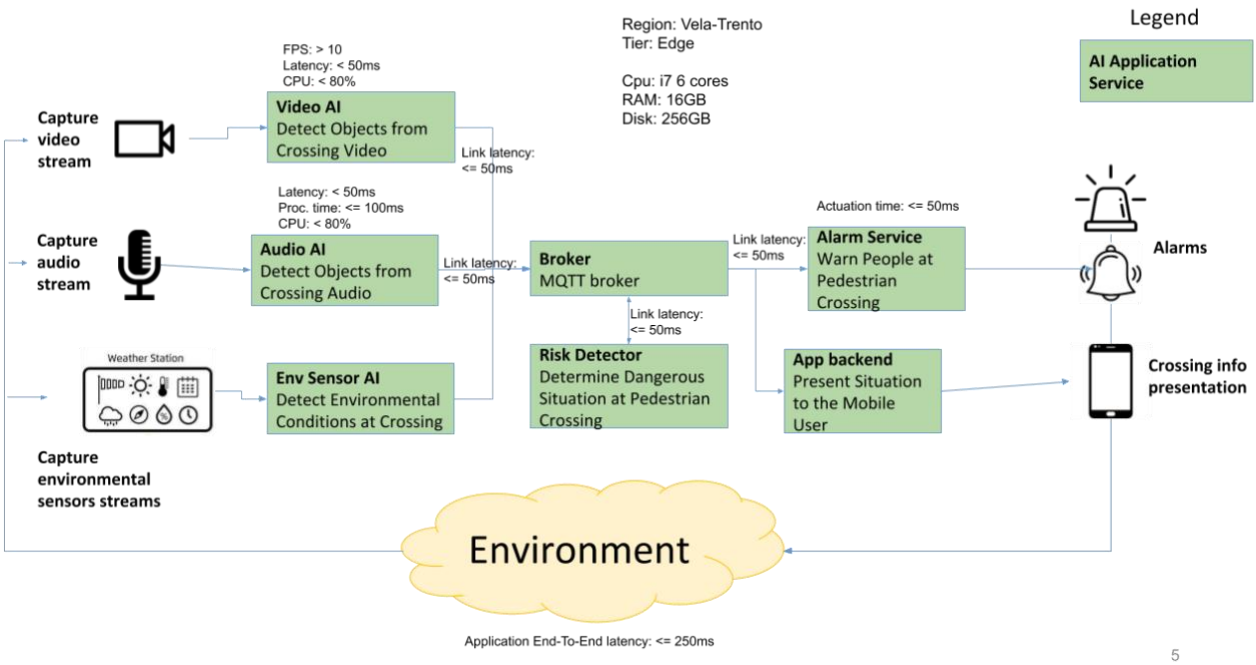

Figure 8. Application microservices for Smart City Safety Crossing use case

- **Service Preconditions**
  The deployment consists in two regions:
  - Trento (Cloud)
    - Location: Trento

        - Type: Cloud infrastructure
    - *Vela-Trento (Edge)*
        - *Location: Vela (TN)*
        - *Type: Barebone PC (CPU i7 6 cores, 16GB RAM, 256GB disk)*

- **Resource requests**

  Table 3 enlists relevant resource requests for the AI application microservices. Each resource is to be intended free on the node (i.e. not consumed by other microservices).

Service	**vCPU**	**Memory**	**Storage**
Video AI	3	2GB	--
Audio AI	2	1GB	--
Env Sensors AI	1	0.5 GB	--
Risk Detector	1	1GB	--
Alarm Service	1	0.5GB	--
Application backend	--	--	20Gb

Table 3. AI application's resource requests

- SLOs

  Each service QoS is described by one or more metrics, each of those could be retrieved automatically by the platform, like CPU usage, memory consumption, latency and bandwidth between microservices, and others are more specific and must be exported by the application.

  Table 4 defines the required QoS value thresholds for the metrics of each application service.

Entity	**Metrics**	**QoS Value**
Video AI service	FPS	> 10
	CPU utilization	<= 80 %
	Availability	>= 95%
Audio AI service	Processing time	<= 60 ms

	CPU utilization	<= 80 %
	Availability	>= 95%
Alarm Service	Actuation time	<= 20 ms
	Availability	>= 95%
Risk Detector	Availability	>= 95%
Link between Camera and Video AI	Network Latency	<= 20 ms
	Network bandwidth	>= 1000Mbps
Link between Microphone and Audio AI	Network Latency	<= 20 ms
	Network bandwidth	>= 1000Mbps
Link between Video AI and Broker	Network Latency	<= 20 ms
Link between Audio AI and Broker	Network Latency	<= 20 ms
Link between Evn Sensor AI and Broker	Network Latency	<= 3 sec
Link between Broker and Risk Situation Detector	Network Latency	<= 20 ms
Link between Broker and Alarm Service	Network Latency	<= 20 ms
Application delay (End-To-End)	Response time	<= 200 ms

Table 4. Service QoS metrics for DECENTER UC-1

Figure 9 depicts a Resource Model that describes a video camera owned by an Application Service Provider and located at the pedestrian crossing. The *Administrative Domain* object in the RM delivers information about who owns the video camera. The *Region Object* specifies the location, where the video camera is located and used. Finally, the *External Endpoint: Thing* describes the service or the edge device that is being included in the RM.

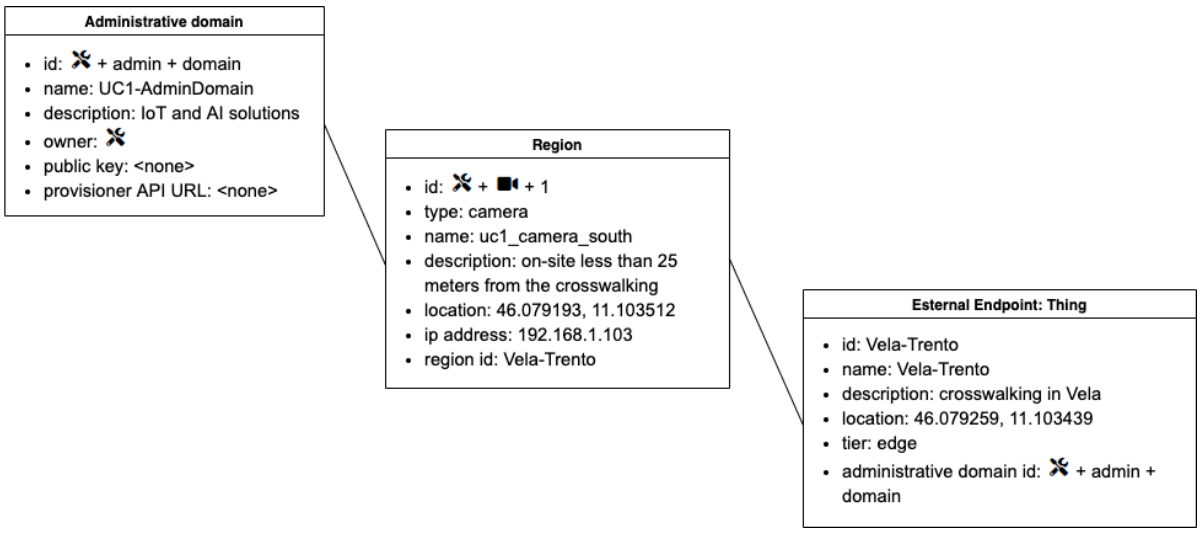

Figure 9. Example of RM for an IP camera used in Smart City Safety Crossing use case

3.1.2 Robotics Logistics

Figure 10 depicts the scenario that was used to deliver the RM and SLA model from the robotics logistic use case.

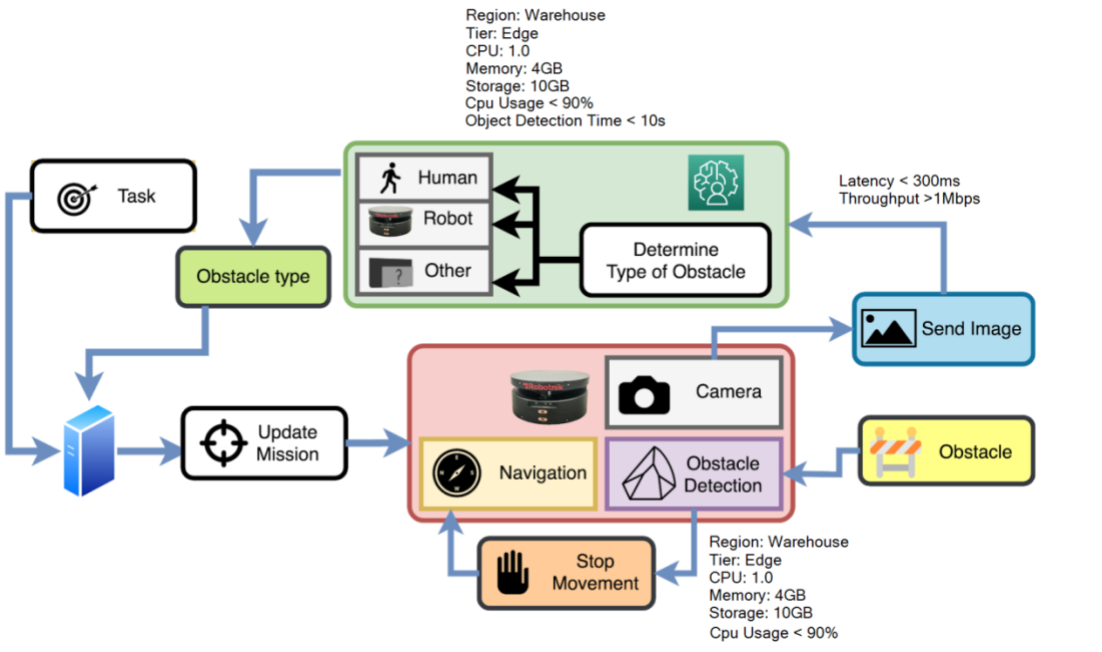

Figure 10. Functional and Non-Functional Requirements for robotics logistic

- **Service Preconditions:**
  - Region:
    - *Robots:* Local warehouse
    - *Fleet Management System:* Local warehouse

        - *AI Application:* Local warehouse
    - Tier:
        - *Robots:* Edge
        - *Fleet Management System:* Edge
        - *AI Application:* Edge
- **Resource Requests:**
    - vCPU:
        - *Robots:* >=1
        - *Fleet Management System:* >=1
        - *AI Application:* >=1
    - RAM:
        - *Robots:* >=4GB
        - *Fleet Management System:* >=4GB
        - *AI Application:* >=4GB
    - *Disk Space:*
        - *Robots:* >=10GB
        - *Fleet Management System:* >=10GB
        - *AI Application:* >=10GB
- **SLOs**
    - *Object Detection Time < 10s,* The AI Application should process the image and send the results to the Fleet Management System in less than 10 seconds
    - *Person Object Detection Confidence >70%,* The AI Application should be configured to only inform that the obstacle is a person if the detection algorithm returns an identification confidence level greater than 70%
    - *Robot Object Detection Confidence >75%,* The AI Application should be configured to only inform that the obstacle is another robot if the detection algorithm returns an identification confidence level greater than 75%
    - *CPU Utilisation < 90%*
    - *Latency < 300ms,* network latency between the robot and the IA Application should be at most 300 ms
    - *Throughput >1Mbps,* The Network throughput between the robot and IA Application should be at least 1Mbps

3.1.3 Smart and Safe Construction

Figure 11 depicts the scenario that was used to deliver the RM and SLA model from the Smart and Safe Construction, where the green coloured components represent the application microservices.

For this use case, it is necessary to assure: high network performance that will allow seamless data transfer; high AI service availability on infrastructure with stable workload, whilst a specific amount of FPS will allow reliable and precise AI processing of high-quality video stream.

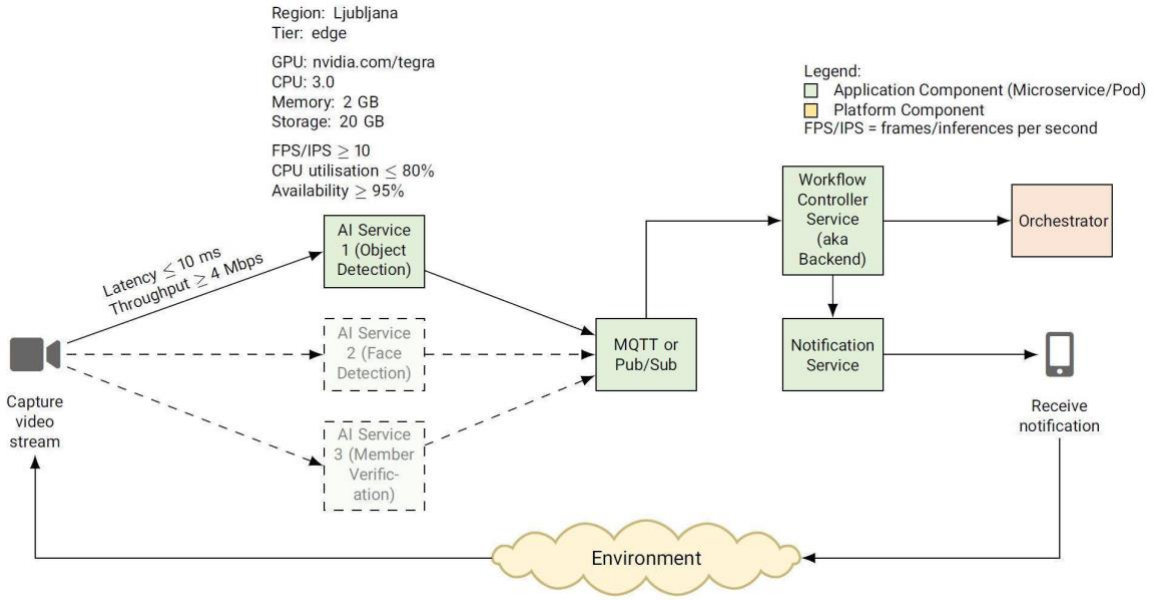

Figure 11. Functional and Non-Functional Requirements for Smart and Safe Construction

The detailed non-functional requirements that were chosen as part of the SLA agreement for the container AI Service1 are as follow:

- Service Preconditions:
  - Region: *Ljubljana*, limits deployment of AI Service1 onto an infrastructure in the Ljubljana region;
  - Tier: *Edge*, limits deployment of AI Service1 onto some edge infrastructure;
- Resource Requests:
  - GPU: *nvidia.com/tegra*, limits deployment of the AI Service1 onto a node that exposes Nvidia GPU with Tegra SoC to a microservice;
  - CPU: *3.0*, limits deployment of AI Service1 onto some node that has available at least 3 vCPUs (i.e. not consumed by other microservices);
  - Memory: *2 GB*, limits deployment of AI Service1 onto some node that has available at least 2 GB of memory;
  - Storage: *20 GB*, limits deployment of AI Service1 onto a node that has available at least 20 GB of storage.
- SLOs:
  - *FPS/IPS ≥ 10*, specifies that a service level for the number of frames/inferences per second of AI Service1 has to be at least 10. This is an application-level metric that requires a custom exporter;
  - *CPU Utilisation ≤ 80 %*, specifies the average CPU utilization from the AI Service1 has to be at most 80%;
  - *Availability ≥ 95 %*, specifies that the availability of the AI Service1 has to be at least 95%;
  - *Latency ≤ 10 ms*, specifies that the network latency between AI Service1 and the video camera should be at most 10ms;

- *Throughput ≥ 4 Mbps*, specifies that the network throughput between AI Service1 and the video camera should be at least 4 Mbps.

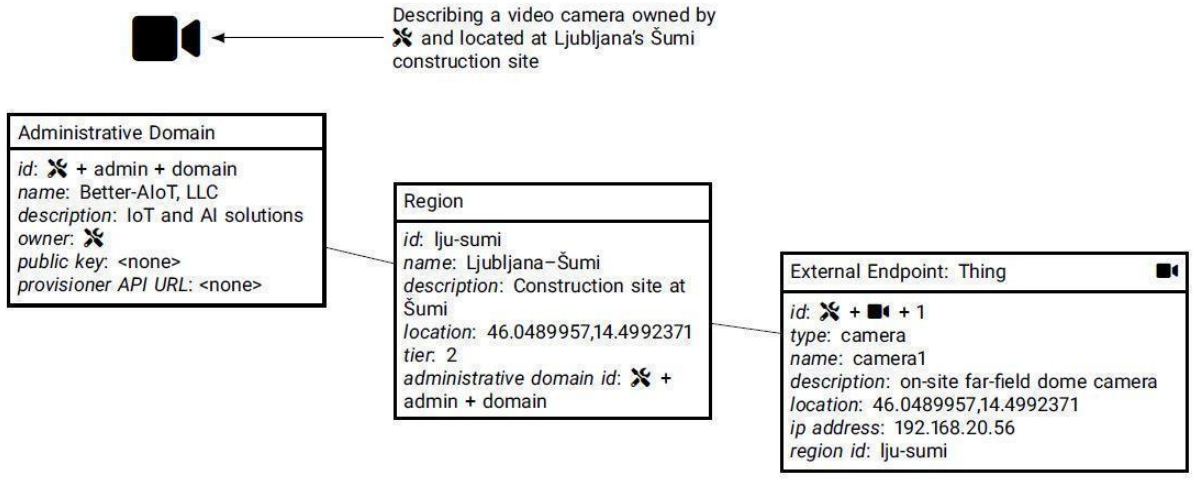

Figure 12. Resource Model Example of a video camera used in Smart and Safe Construction use case

Figure 12 depicts a RM that describes a video camera owned by an Application Service Provider and is located at Ljubljana's Šumi construction site. The RM was developed following the initial UML diagram of a generic Resource Model, which was presented in D2.1. The *Administrative Domain* object in the RM delivers information about who owns the video camera. The *Region Object* specifies the location, where the video camera is located and used. Finally, the *External Endpoint: Thing* describes the service or the edge device that is being included in the RM.

3.1.4 Ambient Intelligence for office environments

Figure 13 depicts the composition of microservices in UC4. It consists of two types of AI microservice, one MQTT broker and one service controller.

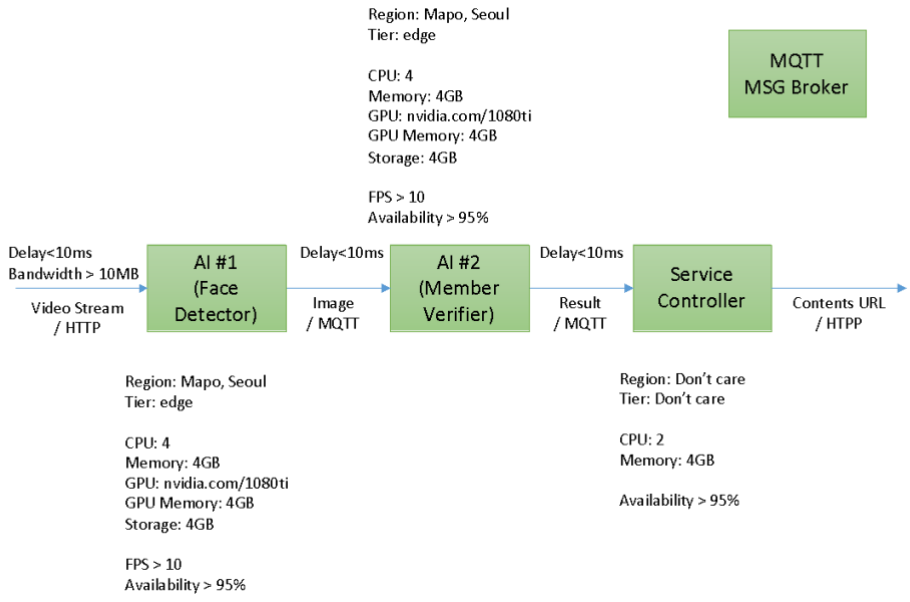

Figure 13. Functional and Non-Functional Requirements for Indoor Intelligence

The resource models and SLOs for UC-4 are as follows.

- Service Preconditions:
  - Seoul (Cloud)
    - Location: Seoul
    - Type: Cloud infrastructure
  - *Sangam-Cloud (Edge)*
    - *Location: Sangam*
    - *Type: Barebone PCs (CPU i5 quad-core, 16GB RAM, 256GB disk, with Nvidia GPU)*

Resource Requirements for each microservices are described in Table 2. Some microservices need GPU resources for faster execution of video-related deep learning models.

Service	**vCPU***	**GPU**	**Memory**	**Storage**
Face Detector	4	1, Nvidia	8GB	20Gb
Member Verifier	4	1, Nvidia	8GB	20Gb
Service Controller	4	-	--	20Gb

Table 5. UC-4 resource requirements

- SLOs:
  The following SLOs are defined as per-microservice-basis, to ensure that the end-to-end delay for this use case does not exceed three seconds.
  - For the video processing AIs (Face Detector and Member Verifier), FPS/IPS ≥ 10. It can be also translated to minimum processing latency of 100ms for each frame received on the microservice. If the input stream of Face Detector is higher than 10fps, then the frames will be dropped at the Face Detector microservice.
  - Availability ≥ 95 %, specifies that the availability of the AI Service-1 has to be at least 95%.
  - Latency ≤ 10 ms, specifies the network latency between each microservices.
  - For AI microservice which receives input from a video camera (Face Detector), Bandwidth > 10MB.

4. DECENTER Architecture

The main scope of DECENTER can be summarised as follows: create and operate AI-based workloads anywhere at any time. Indeed, the project is defining and developing a set of tools that combine and evolve solutions stemming from the Cloud Computing and Internet of Things technology domains in order to provide intelligence closer to IoT sensors and devices and to end-users, where data is actually produced and, in most of the modern IoT-based scenarios, consumed. This implies that DECENTER allows to capture information, to process and visualise it, and to trigger relevant actions at the edge of the infrastructure, in a timely, effective and privacy-preserving manner.

We are facing a particularly challenging task, since it involves actors and stakeholders that, until the recent past, have been pursuing completely diverse objectives: on the one hand the infrastructure manager (or the cloud provider), whose primarily interest is in the robustness, security and efficiency of the infrastructure, with the goal of serving as much customers as possible while minimising costs; on the other hand the AI application developer, who is exclusively interested on the business logic of the AI application and on the user experience. The two actors know very little about each other's context and challenges: the infrastructure manager has almost no visibility on the specific parameters of an AI application that are of interest for the developer (e.g. completion time, accuracy of the AI model, and more), while the developer has (commonly) little or no knowledge about cloud operations and about management and orchestration of a geographically distributed infrastructure. Indeed, while the training of a machine learning model based on some data sets requires knowledge and skills in the AI domain, transforming a model into a running microservice, which can effectively communicate with other processes by means of technology-agnostic protocols, and understanding how this can be optimally served by a computing infrastructure is a task that needs IT and cloud computing skills. At the same time, with the constant shift of services from the cloud to the edge, it has become quite clear that the behavior of a cloud-native application (i.e., an application composed of loosely-coupled microservices communicating among them) can be heavily influenced by the matching that is made to assign computational resources to the different modules of such application. As a matter of facts, considering one of the DECENTER use-cases as an example, it is clear that the microservices that process data to detect dangers for pedestrians in street crossings will perform as expected if they are placed in a node that has enough computation capacity to elaborate the sensors and video streams and is, at the same time, placed very closely to where these streams are captured.

DECENTER aims at filling the gap between these two types of stakeholders and knowledge sets, taking advantage of a relevant trend in which, following what has happened in the software development area, the continuous flow of activities to develop and operate an AI model is slowly converging into the hands into a single expert. This figure takes the name of MLOp (where ML stands for Machine Learning and Op for Operation), that is, someone that develops and operates machine learning workloads, thus recalling the successful experience of the DevOps, those figures that are developing software tools and operating them.

In this perspective, DECENTER developed an architecture that provides all the necessary tools to create and operate AI-based workloads in a heterogeneous, distributed and opportunistically created (when this is needed) fog computing infrastructure, covering the whole cloud-to-edge continuum. More specifically, the DECENTER architecture provides **platforms** and **services** (running on top of the former) to support the entire cycle of creation

and operation of AI applications. The services are introduced to support the creation of AI-based cloud-native applications starting from models that can be easily retrieved and managed and to manage and share models, data and already processed information (e.g. the outcome of the inference of an AI-model). The platforms are then used to capture data from sensors and devices (which is then fed to the applications), to orchestrate the fog computing resources according to the needs of the AI applications, to monitor their behaviour, and to find, acquire and manage resources from third parties (other cloud providers) which can be necessary to cater to the requirements of the applications (e.g. a fog node in a specific location).

4.1 High-level architecture design

A high-level view of DECENTER is provided in Figure 14. In the figure, the DECENTER platform is organised in three functional layers or blocks: Fog and Brokerage platforms, Application services and User interfaces. The *Application Services* layer has been designed to enable the deployment of AI-based applications in the form of microservices, while the resources are managed by the *Fog* and *Brokerage Platforms*.

The colour code in the figure highlights which components have been developed from scratch (yellow), which ones are off-the-shelf (OTS) components adopted in the project (red), and the OTS software that have been modified within the project (cyan). Labels **C** and **E** indicate which components are installed on cloud tier and which are distributed at the edge, respectively.

In the figure, the components are divided in three main systems, each further divided in sub-systems:

- **Application Services**: comprises components offering services for the AI applications.
- **Fog Platform**: comprises components that implement the DECENTER platform for managing heterogeneous and distributed computing infrastructures, which are essential to be able execute AI based services with greatly varying computational, memory and communication requirements across the highly heterogeneous Cloud-to-Edge computing continuum. The Fog Platform is further split into the IoT Platform, Front End, Back End, Monitoring, and Infrastructure Manager sub-systems.
- **Brokerage Platform**: comprises components for implementing the resource sharing among different infrastructure owners.

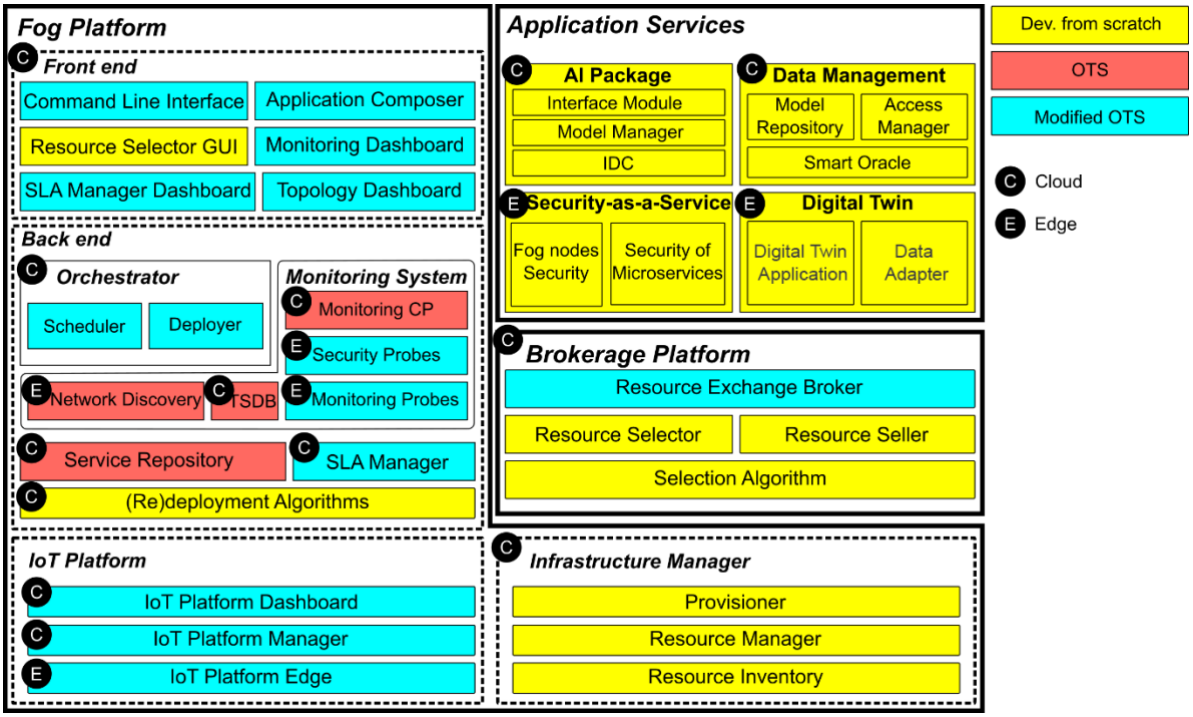

Figure 14. DECENTER high-level overview

In the following subsections each component is described in greater detail and linked to the project's task(s) in which it has been implemented.

4.1.1 Application Services

COMPONENT NAME	**Interface Module**
SUBSYSTEM	*AI Package*
DESCRIPTION	
This component provides network interfaces to the AI methods in a microservice. It supports HTTP and MQTT protocols to configure and control the AI methods.	
RELATIONSHIPS	
• *Model Repository (Data Management)*	
IMPLEMENTATION	
Built from scratch as a DECENTER AI package in Task T4.4.	

COMPONENT NAME	**Intermediate Data Compression (IDC) Module**

SUBSYSTEM	*AI Package*
DESCRIPTION	
This component implements intermediate data compression method inside DECENTER AI Package. The intermediate data compression method compresses the data generated by partial AI model to reduce network bandwidth usage.	
IMPLEMENTATION	
Built from scratch in Task T4.1.	

COMPONENT NAME	**Model Manager**
SUBSYSTEM	*AI Package*
DESCRIPTION	
This component implements model management for an AI service inside a container. It interacts with Model Repository to download an AI model of interest to a running AI microservices.	
RELATIONSHIPS	
• *Model Repository*	
IMPLEMENTATION	
Implemented from scratch in Task T4.3.	

COMPONENT NAME	**Model Repository**
SUBSYSTEM	*Data Management*
DESCRIPTION	

The Model Repository stores AI models. It implements methods that allow to persist and retrieve the models through RESTful APIs. Microservices that are granted access from the Access Manager can retrieve, download and run the AI models.
RELATIONSHIPS
• *Access Manager (Data Management)* • *Interface Module (AI Package)* • *Model Manager (AI Package)*
IMPLEMENTATION
Implemented from scratch in Task T4.3.

COMPONENT NAME	**Access Manager**
SUBSYSTEM	*Data Management*
DESCRIPTION	
This component implements access verification of data (AI model) with respect to the current deployment configuration. It compares the deployment configuration and running environments by Smart Oracle to grant the access of desired AI model.	
RELATIONSHIPS	
• *Smart Oracle* • *AI Package* • *Deployer (Orchestrator)*	
IMPLEMENTATION	
Data management with Blockchain technology is implemented to verify access to an AI model in T4.3.	

COMPONENT NAME	**Smart Oracle**

SUBSYSTEM	*Data Management*
DESCRIPTION	
This component implements innovative mechanisms for regulations, policies, certifications and permissions-based cross-border data management, with the main goal of preserving users' data privacy. It is installed on the cloud.	
IMPLEMENTATION	
Designed and implemented in T4.3.	

COMPONENT NAME	**Data Adapter**
SUBSYSTEM	*Digital Twin (DT)*
DESCRIPTION	
This component implements interfaces to store valuable features extracted from the AI methods, which can be used for Digital Twin implementation. It provides APIs to store features from the AI methods and to retrieve data for Digital twin application	
RELATIONSHIPS	
• *Interface Module (AI Package)*	
IMPLEMENTATION	
The interfaces are designed and implemented with respect to the features from use cases. This work is part of T4.2.	

COMPONENT NAME	**Digital Twin Application**
SUBSYSTEM	*Digital Twin*
DESCRIPTION	

This component implements Digital Twin service by using features stored in DT Data Adapter. *This component by exploiting real-time data collected by IoT sensors and AI learning data, can create a Digital Twin representation, i.e., a digital replica of existing physical entities. Such a representation is useful to model how real situations might evolve and how a system might react based on a combination of inputs.*
RELATIONSHIPS
• *Data Adapter (Digital Twin)*
IMPLEMENTATION
This work is part of T4.2.

COMPONENT NAME	**Security as a Service (Security aaS)**
SUBSYSTEM	*None*
DESCRIPTION	
This is a set of containerized components that can be deployed throughout the distributed infrastructure in order to enforce security at infrastructure level (e.g. by detecting and mitigating attacks to the edge nodes) and/or at application level (e.g. by detecting abnormal application behaviour).	
RELATIONSHIPS	
• *Service Repository*	
IMPLEMENTATION	
Designed and implemented from scratch in Task 3.4. It includes a solution for protecting the fog nodes from DDoS, and a component for anomaly detection for microservices called L-ADS.	

4.1.2 Fog Platform

COMPONENT NAME	**IoT Platform Manager**
SUBSYSTEM	*IoT Platform*

DESCRIPTION
This component implements the control plane of the IoT Platform adopted. It is installed centrally on the cloud even though it could be distributed. *Currently two IoT Platforms are integrated in DECENTER: SensiNact and Things+. For a further description of these two platforms see D3.3 [33].*
RELATIONSHIPS
● *IoT Platform Edge*

IMPLEMENTATION
In Task 3.1 two solutions are provided, namely SensiNact and Things+.

COMPONENT NAME	**IoT Platform Edge**
SUBSYSTEM	*IoT Platform*
DESCRIPTION	
This component implements the edge part of the IoT Platform adopted. It is installed on the edge nodes and interacts with the IoT Platform Manager in order to offer IoT related services (e.g. access to sensor data). Moreover, it supports the Digital Twin component.	
RELATIONSHIPS	
● *IoT Platform Manager* ● *Digital Twin*	
IMPLEMENTATION	
In Task 3.1 two solutions are provided, namely SensiNact and Things+.	

COMPONENT NAME	**IoT Dashboard**
SUBSYSTEM	*IoT Platform*
DESCRIPTION	
This component acts as a graphical front end of the IoT Platform Manager.	
RELATIONSHIPS	
● *IoT Platform Manager* ● *Time Series DB*	
IMPLEMENTATION	
It is implemented in task T3.1, according to the two IoT solutions provided (SensiNact and Things+).	

COMPONENT NAME	**Monitoring Probe**
SUBSYSTEM	*Back End - Monitoring*
DESCRIPTION	
This component represents the remote agents (e.g. Prometheus exporters) responsible to collect metrics from different objects: infrastructure metrics, application metrics, etc. It is distributed on each resource that should be monitored.	
RELATIONSHIPS	
● *Monitor Control Plane*	
IMPLEMENTATION	
In Task 3.1, Monitor Probes, based on Prometheus exporters, are developed to support the monitoring system.	

COMPONENT NAME	**Security Probe**

SUBSYSTEM	*Back End - Monitoring*
DESCRIPTION	
This component collects metrics related to security issues. It is distributed on each resource that should be guarded against security attacks.	
RELATIONSHIPS	
• *Monitoring Control Plane*	
IMPLEMENTATION	
In Task 3.1 and Task 3.4, Security Probes are developed to support the monitoring system.	

COMPONENT NAME	**Network Discovery**
SUBSYSTEM	*Back End - Monitoring*
DESCRIPTION	
This component monitors the networks of a computing infrastructure and keeps the picture of network topology up to date. It is installed as a monitoring agent on each network resource.	
RELATIONSHIPS	
• *Monitoring Control Plane*	
IMPLEMENTATION	
In Task 3.1, Network Discovery is developed to support the monitoring system.	

COMPONENT NAME	**Time Series Data Base**
SUBSYSTEM	*Back End - Monitoring*

DESCRIPTION
It is the time series database where all the collected metrics are stored. It is implemented thanks to an off-the-shelf Open Source product (InfluxDB). It is installed centrally in the cloud even though it could be distributed for performance and availability purposes.
RELATIONSHIPS
• *Monitoring Control Plane*
IMPLEMENTATION
Developed in Task 3.1 and based on Prometheus and InfluxDB.

COMPONENT NAME	Monitoring Control Plane
SUBSYSTEM	*Back End - Monitoring*
DESCRIPTION	
This component coordinates the monitoring related activities collecting metrics from all the other components in the Monitoring sub-system and storing them into the time series database. Through this component or directly through the database, data can be consumed by other systems.	
RELATIONSHIPS	
• *Monitoring Probe* • *Security Probe* • *Time Series DB* • *Network Discovery* • *Other components that need to access monitoring data*	
IMPLEMENTATION	
Developed in Task 3.1, it is based on Prometheus and installed on the cloud.	

COMPONENT NAME	Service Repository

SUBSYSTEM	*Back End*
DESCRIPTION	
This component stores the images of the containerized microservices of AI applications. It could be implemented as a private or public container registry (i.e. Docker Hub). It is installed on the cloud.	
RELATIONSHIPS	
• *Application Service layer* • *Scheduler*	
IMPLEMENTATION	
Developed in T3.1 in order to allow all the DECENTER project to store and distribute the container images. It is based on GitLab registry.	

COMPONENT NAME	**SLA Manager**
SUBSYSTEM	*Back End*
DESCRIPTION	
It dynamically certifies SLAs stipulated with other cloud/fog providers when some resources are rented through the Brokerage Platform, by taking as input the needed monitoring data and notifying the Brokerage Platform if any SLA violation occurs. It is installed on the cloud.	
RELATIONSHIPS	
• *Monitoring Control Plane* • *SLA Manager Dashboard* • *Resource Exchange Broker*	
IMPLEMENTATION	

The SLA Manager is developed in Task T3.1 and it uses the JavaScript Object Notation (JSON) to interface with the other components such as the Monitoring System and the REB.

COMPONENT NAME	**Scheduler**
SUBSYSTEM	*Back End - Orchestrator*
DESCRIPTION	
It gathers deployment requests from the Application Composer (or Command Line Interface) and container images from Service Repository. Thanks to the Deployment Algorithm decides the best placement for the application microservices in the different regions/nodes of the infrastructure and finally deploys the applications using the Deployer component. *The Scheduler implements two different algorithms (MILP and MDP)[32], which have different orchestration goals. The MILP algorithm is used to determine an optimal region for the orchestration of multi-tier applications, whilst the MDP is used to determine an optimal node/set of nodes for orchestration of multi-tier applications. Because of the different output these algorithms have, two separate Kubernetes controllers that handle different Custom Resource Definitions (CRDs) were designed and implemented in Y2. A thorough description of the custom Kubernetes controllers is delivered in D3.3 [33]. This approach is designed to demonstrate the flexibility and extensibility of the DECENTER platform to be able to address greatly varying QoS requirements of different AI-based applications.*	
RELATIONSHIPS	
● *Deployment Algorithm* ● *Deployer* ● *Service Repository* ● *Command Line Interface* ● *Application Composer*	
IMPLEMENTATION	
In DECENTER we foresee two levels of scheduling: one at region level and one at node level. For the second one we exploit functionalities already offered by Kubernetes, while the first one is original to DECENTER and it is installed on the cloud. This activity is carried out in Task T3.1.	

COMPONENT NAME	**Deployment and Re-Deployments Algorithms**

SUBSYSTEM	*Back End - Orchestrator*
DESCRIPTION	
It implements an algorithm, pluggable inside the Scheduler, able to select the region(s)/node(s) where a given microservice should be deployed. It bases its decisions on the current status of the infrastructure and on microservices/data flows requirements set by the user. It is installed on the cloud.	
RELATIONSHIPS	
• *Monitoring Control Plane* • *Scheduler*	
IMPLEMENTATION	
These algorithms are studied in Task T3.3. They include algorithms for the initial deployment of applications in the Fog infrastructure. Instead, re-deployment algorithms have been studied and proposed to find an alternative deployment solution when the QoS standards of an application are no longer respected. The current implementation of the latter is based on a Markov Decision Process method.	

COMPONENT NAME	**Deployer**
SUBSYSTEM	*Back End - Orchestrator*
DESCRIPTION	
This component performs the actual deployment of the microservices on the distributed infrastructure. It is installed on the cloud but its agents are distributed on all the nodes of the infrastructure.	
RELATIONSHIPS	
• *Scheduler*	
IMPLEMENTATION	

The Deployer is part of the Orchestrator and it is the result of tasks T3.1 and T3.3.

COMPONENT NAME	**Provisioner**
SUBSYSTEM	*Back End – Infrastructure Manager*
DESCRIPTION	
This component oversees all the operations that are needed to commission or decommission resources (e.g. nodes) either added/removed directly to/from the infrastructure or rented through the Brokerage Platform. It is installed on the cloud.	
RELATIONSHIPS	
● *Resource Selector GUI*	
IMPLEMENTATION	
The Provisioner is implemented in tasks T3.1.	

COMPONENT NAME	**Resource Manager**
SUBSYSTEM	*Back End – Infrastructure Manager*
DESCRIPTION	
This component keeps under control the status of the infrastructure in terms of available vs allocated resources. It stores this information in the Resource Inventory (e.g. by updating the resource status whenever it changes). It is installed on the cloud.	
RELATIONSHIPS	
● *Resource Inventory* ● *Provisioner*	
IMPLEMENTATION	
The Resource Manager is the result of tasks T3.1.	

COMPONENT NAME	Resource Inventory
SUBSYSTEM	*Back End – Infrastructure Manager*
DESCRIPTION	
It is a database storing the infrastructure status in terms of spare, commissioned and rented resources. It is installed on the cloud.	
RELATIONSHIPS	
● *Resource Manager* ● *Resource Seller*	
IMPLEMENTATION	
The Resource Inventory is the result of tasks T3.1.	

COMPONENT NAME	Command Line Interface
SUBSYSTEM	*Front End*
DESCRIPTION	
It allows users to access the Scheduler APIs from a console. It is installed on the cloud.	
RELATIONSHIPS	
● *Scheduler*	
IMPLEMENTATION	
It is implemented in task T3.1 leveraging k8s command line (kubectl).	

COMPONENT NAME	Application Composer
SUBSYSTEM	*Front End*
DESCRIPTION	
It is used to model AI applications following the microservice architecture. Through the Application Composer, a user can draw the application graph, specify which microservices compose the application and impose computational and networking requirements for each microservice and data flow. It is installed on the cloud.	
RELATIONSHIPS	
• *Scheduler*	
IMPLEMENTATION	
It is implemented in task T3.1.	

COMPONENT NAME	SLA Manager Dashboard
SUBSYSTEM	*Front End*
DESCRIPTION	
This component acts as a graphical client of the SLA Manager component. Through it, it is possible to access the SLA Manager and to configure it. It is installed on the cloud.	
RELATIONSHIPS	
• *SLA Manager*	
IMPLEMENTATION	
It is implemented in task T3.1.	

COMPONENT NAME	Topology Dashboard

SUBSYSTEM	*Front End*
DESCRIPTION	
It provides a geographical and topological view of the distributed fog infrastructure. Moreover, it shows the locations/regions where applications have been deployed. It is installed on the cloud.	
RELATIONSHIPS	
• *Monitoring Control Plane*	
IMPLEMENTATION	
It is implemented in task T3.1.	

COMPONENT NAME	**Monitoring Dashboard**
SUBSYSTEM	*Front End*
DESCRIPTION	
This component acts as a graphical front end of the monitoring system. It shows in a graphical and/or tabular format the metrics collected. It also offers the possibility to set thresholds and alerts on specific metrics. It is installed on the cloud and based on Grafana.	
RELATIONSHIPS	
• *Monitoring Control Plane* • *Time Series DB*	
IMPLEMENTATION	
It is implemented in task T3.1 leveraging Grafana.	

4.1.3 Brokerage Platform

COMPONENT NAME	Resource Seller
SUBSYSTEM	*Back End – Brokerage Platform*
DESCRIPTION	
This component handles the process of selling spare resources (advertisement, reservation, confirmation) interacting with the Resource Exchange Broker. It is installed on the cloud.	
RELATIONSHIPS	
● *Resource Exchange Broker* ● *Resource Inventory*	
IMPLEMENTATION	
Software component developed in Task T3.1 as part of the Brokerage Platform.	

COMPONENT NAME	Resource Selector
SUBSYSTEM	*Back End – Brokerage Platform*
DESCRIPTION	
It selects, using a Selection Algorithm, the appropriate resources that have to be rented from the Resource Exchange Broker for the successful deployment of AI applications (e.g. in case some resources from a specific location, which are by from a different provider, are needed). It is installed on the cloud.	
RELATIONSHIPS	
● *Resource Selector GUI* ● *Resource Exchange Broker* ● *Selection Algorithm*	
IMPLEMENTATION	

Developed in Task T3.2, it ensures the agreement between provider and user, by automatically matching each user request to the provider that satisfies their needs.

COMPONENT NAME	**Resource Exchange Broker**
SUBSYSTEM	*Back End – Brokerage Platform*
DESCRIPTION	
The Resource Exchange Broker is a blockchain-based repository where each provider can advertise spare resources, which can then be rented for a certain amount of time by other providers. It is installed on the cloud.	
RELATIONSHIPS	
● *Resource Seller* ● *Resource Selector* ● *SLA Manager*	
IMPLEMENTATION	
Developed in Task T3.2, it is based on an Ethereum private blockchain	

COMPONENT NAME	**Selection Algorithm**
SUBSYSTEM	*Back End – Brokerage Platform*
DESCRIPTION	
This component represents an algorithm that can be plugged into the Resource Selector in order to automatically select resources to be rented. It is installed on the cloud.	
RELATIONSHIPS	
● *Resource Selector*	

IMPLEMENTATION
Developed in Task T3.2 and T3.3, it is based on a matchmaking procedure.

COMPONENT NAME	Resource Selector GUI
SUBSYSTEM	*Front End*
DESCRIPTION	
This component is the front end of the Resource Selector. Through this GUI, advertised resources can be shown, selected and reserved. It can replace the Selection Algorithm in the case a manual selection of resources is preferred. It is installed on the cloud.	
RELATIONSHIPS	
• *Resource Selector*	
IMPLEMENTATION	
Developed in Task T3.1.	

5. Business Model for Federated Cloud-to-Edge Environments

5.1 Business Model for Federated Cloud-to-Edge Environments

Task 2.3 aims at identifying and analysing different business scenarios for DECENTER solutions in the Cloud and Fog Computing markets. Our aim is to address the market needs for diverse AI-based applications deployed and orchestrated across the complete Cloud-to-Things computing continuum. Additionally, this still ongoing task examines how the business model proposed can be adopted in the DECENTER use cases, so a different adoption roadmap will be defined for each use case.

5.2 DECENTER Business Model Analysis Roadmap

This section presents the initial roadmap envisioned for Task 2.3. The roadmap establishes the activities to perform from the beginning of the task (M13) to the project end, as well as the relation with other tasks.

The following figure shows the initial task roadmap:

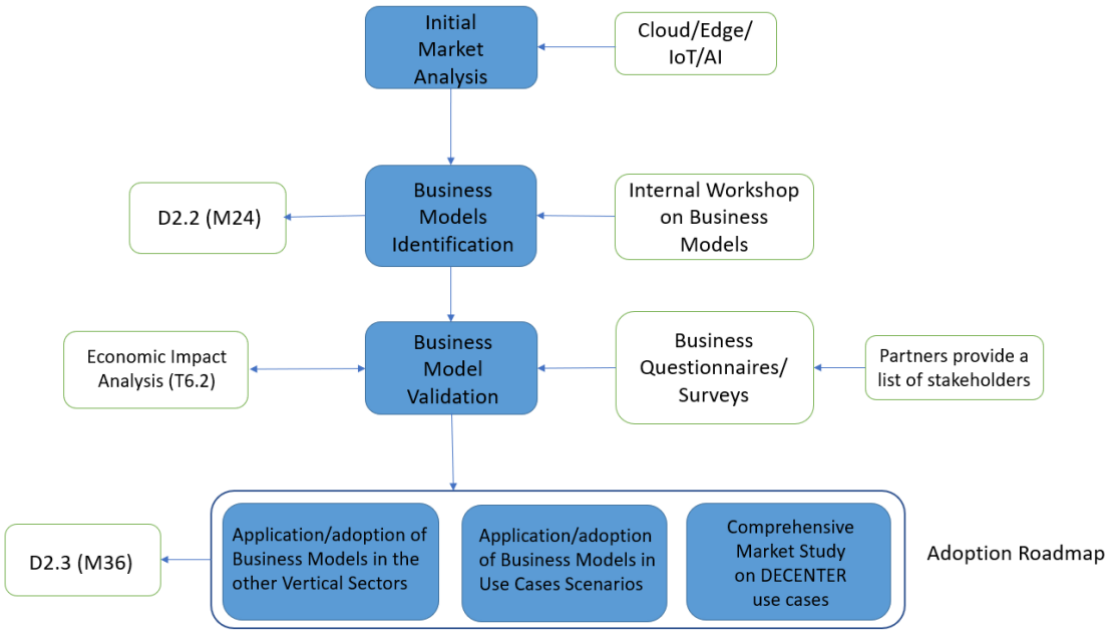

Figure 16. Task 2.3 Roadmap

The roadmap is split into 4 main activities, as defined below:

- **Market Analysis:** The first step to decide the DECENTER business models is to conduct an initial market analysis in the context of the technologies involved in the project. The first market analysis will be focused on the identification of competitors in the market. The market analysis helps to establish the value proposition of the DECENTER solutions, identify the market competitors and the business models usually used in the Cloud and Fog computing markets. A preliminary analysis of

competitors is included in Section 5.3 of D2.2. and a detailed market analysis will be reported in D2.3

- **Business Model Identification:** After analysing the market context of the project, different business models will be identified for DECENTER outcomes using the Business Model Canvas Methodology [3]. Moreover, an internal workshop among partners will be carried out to discuss the main elements of the Business Model Canvas (BMC) and finally set up a preliminary version of the BMs.
- **Business Model Validation:** Once the BMs identified and discussed by the consortium, it is needed to validate them. For that, we require that all the partners provide a list of stakeholders in the different domains to conduct a survey and several meetings to get feedback about the business aspects of DECENTER solutions. The survey's findings will also be used in Task 6.2 to perform the Economic Impact Analysis.
- **Adoption Roadmap:** Finally, the last step of the process is the creation of an adoption roadmap after analysing how the proposed BMs can be applied in the DECENTER use cases. In addition to the project use cases, the adoption roadmap will also examine other vertical sectors in which DECENTER solutions could be exploited.

D2.3 DECENTER Business Models will provide a detailed analysis of the BMs for DECENTER Platform as well as the BMs to be adopted by the use case partners. This analysis will include the BMs validation, the adoption roadmap and a comprehensive market analysis focused on the European and Korean market context and differences between them. Moreover, it will include the BMs for the project use cases considering the customers segments and the partners involved in each use case.

5.2.1 Methodologies

Two different methodologies will be used to analyse in-depth the BMs:

- The Business Model Canvas Methodology to define and analyse the BMs; and
- Business surveys, meetings and workshops, to validate the BMs following different validation criteria.

Next, we present an overview of the BMC methodology and an approach to the BMs validation. The validation criteria followed by the BM validation will be refined and updated in D2.3.

5.2.1.1 Business Model Canvas Methodology

To define the BMs for DECENTER Platform, we have used the BM Canvas Methodology [4] . This tool was created by Alexander Osterwalder [5] and it is widely used to define and analyse all the aspects to consider in a business model in a graphic way. The BM Canvas is a shared language for describing, visualizing, analysing, and assessing business models. It describes the rationale of how an organization creates, delivers, and captures value to their customers.

The BM Canvas template is depicted in Figure 17.

The nine elements described in the BM Canvas encompass the main part to understand a business: customer segments, offering, infrastructure and financial viability, and they provide the following information:

1. **Customer Segments:** The first step of a business is to identify who are the customers of the value created by the solution we offer. The customers need to be understood and segmented based on needs in order to define a proper commercialization strategy.
2. **Value Proposition:** The product or service offered must meet the customers' needs differently and uniquely from the competitors in the market.

3. **Channels:** The channels to contact with customers and deliver the value proposition can determine how successful a project is and must be efficient and cost-effective.
4. **Customer Relationships:** Each organization must build a relationship with its customers to retain them and make them grow. Based on the type of product or service and the customers segments, different relationship strategies can be chosen.
5. **Revenue Streams:** It describes how the organizations get revenues from its customers and how they are willing to spend on products or services.
6. **Key resources:** The resources needed to create the value offered to the customers. They can be financial, human, technological, etc.
7. **Key activities:** They are the crucial activities to create a product or provide a service.
8. **Key partnerships:** Organisations focus on the key activities and establish relations with other partners to optimize parts of the process or obtain resources.
9. **Cost Structure:** It describes the cost of the key activities and it is a determining factor to make a business profitable.

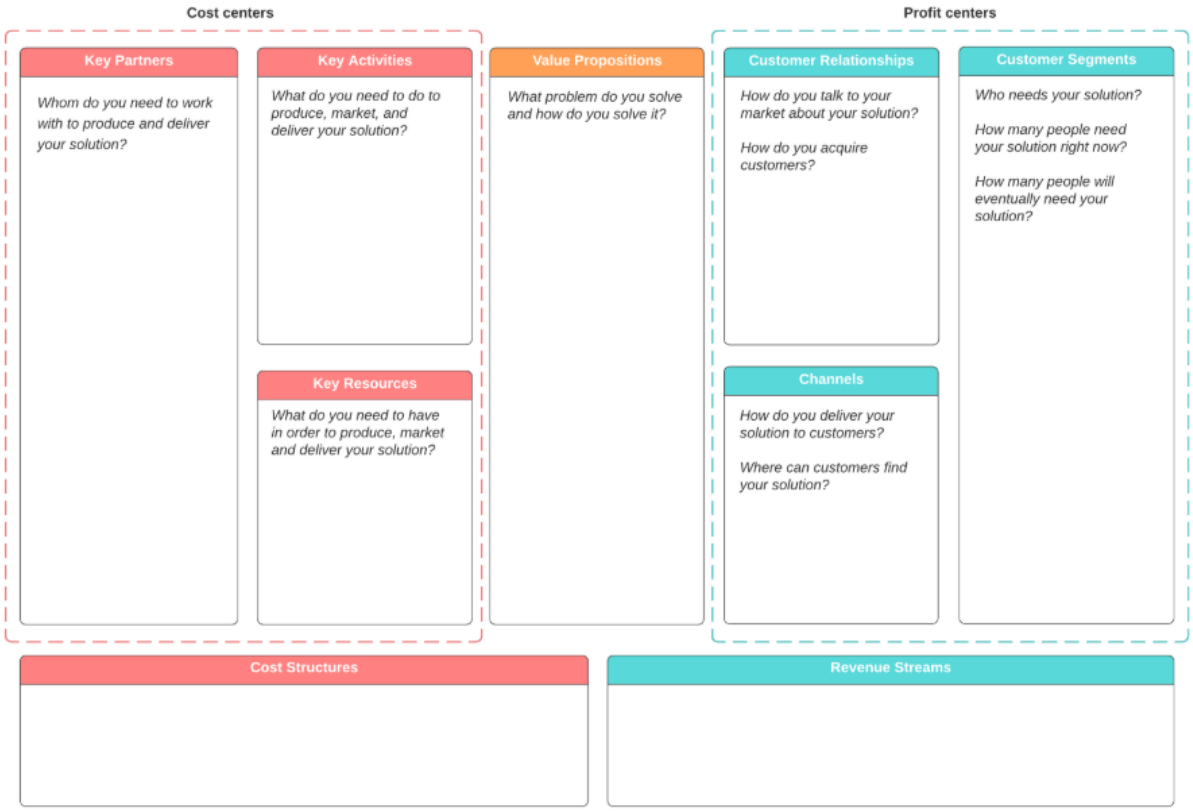

Figure 17. Business Model Canvas [6]

It is worth mentioning that this tool does not provide information regarding the profitability of a business; in that case, a business plan would have to be developed based on the BM and including financial statements.

5.2.1.2 Business Model Validation

The validation of the BMs will be decisive to choose the most promising one among the BMs identified for DECENTER Platform in the current market context. The validation process will be based on a previously methodology used successfully by ATOS in research projects within

the Cloud market [7], although modified to be used from the point of view of the customer instead of the provider of the DECENTER Platform.

Two activities will be performed to carry out the BM validation:

- **Research point of view**: Considering aspects of the market analysis, such as competitors, customers segment, value proposition, etc.
- **Interacting with stakeholders out of the project use cases**: A questionnaire will be circulated among potential customers of the platform previously selected by the partners to gather their thoughts about the potential of DECENTER applied to the different use cases and other vertical sectors. Additionally, several meetings will be carried out with stakeholders and potential key partners of the BM out of the project to get feedback about their expectations of the DECENTER Platform.

A set of criteria will be established in order to assess the questionnaire results and validate the BMs selected. A preliminary list of validation criteria is depicted in the next section.

Within D2.3, a more comprehensive analysis of the BM validation will be included, as well as the outcomes of the business questionnaire.

5.2.1.3 Validation Criteria

The validation will consider two different aspects of the BM related to the customers: the ***impact*** on the customers and the market, and the **ease of implementation** for customers of the DECENTER Platform through the BM proposed.

A set of criteria has been defined to validate the two dimensions, impact and ease of implementation. The range of the values assigned to each criterion is from 0 (lowest) to 5 (highest). The BMs identified will be evaluated considering the score of each criterion, which will help to compare BMs between them.

	Criterion	**Value**
		[0 - 1] lower, [1 - 2] low, [2 - 3] medium, [3 - 4] high, [4 - 5] higher
IMPACT	Revenue stream potential	Value the revenue stream proposed by the BM
	Customers' acceptance	Value the customers' acceptance of the product or service
	Differentiation	Value the novelty on the market comparing with other existing solutions
	Visibility	Value the potential to raise attention without dedicated marketing campaigns
	Customers' need assessment	Value how important for the customer is the need met by the solution proposed

	Impact on the customers' business	Value how the solution impact on the customers' business in terms of benefits
	Positioning	Value the expected positioning of the solution in the market compared with other competitors (value the size of the expected market)
EASE OF IMPLEMENTATION	Investment costs	Value investment costs for customers
	Learning Curve	Value the time invested in learning how the solution works
	Risks	Value the objective technical feasibility with today's knowledge
	Integration	Value the integration effort to use the solution together with customers' existing infrastructures and providers services
	Risks	Value the objective technical feasibility with today's knowledge
	Transparency	Value how detailed is the info provided to the customers

Table 6. Validation Criteria

5.3 Market Analysis Overview

5.3.1 Edge Computing and IoT Market

The global Edge Computing market is projected to grow from $1,756.5M in 2019 to $8,295M by the end of 2025 at a CAGR of 29.4% as predicted by Mordor Intelligence Report [8]. Over the forecast period, some large Telecom and manufacturing enterprises are expected to use Edge Computing related to IoT, such as Huawei that predict that by this year, 2020, over 50% of data may be stored, processed and analysed in the edge.

The main drivers of this market demand are the introduction of 5G, the increase of IoT software platforms with new capabilities, and the development of intensive applications involving Machine Learning (ML), AI and IoT.

IoT market analysts envision that Edge Computing will play a significant role in supporting IoT implementations going forward, as it enables IoT deployments more efficient and self-sustaining [9]. Edge for IoT brings multiple benefits for many IoT deployments, such as decreased response time together with increased communications efficiency, compared to using the Cloud to process and store data. Many new IoT applications such as drones, smart grids or connected vehicles rely on Edge computing. By the year 2016, Gartner recognised in

its “Hyper-Cycle for Infrastructure Strategies”, that Edge computing and IoT Edge Architecture were going to become the innovation trigger for market realisation in 2-5 years.

According to MordorIntelligence [10], the global IoT market is expected to reach a value of $1.256B by 2025 from $690B in 2019 at a CAGR of 10.53% during the forecast period. The positive growth rate is achieved due to several factors, such as the increase in Cloud platform, cost reduction in connected devices and the advent of advanced data analytics. Nevertheless, The Boston Consulting Group predicts significant differences in growth rates, which refers to different layers of IoT technology stack [11].

Next table shows the distribution of spending among different layers of IoT technologies.

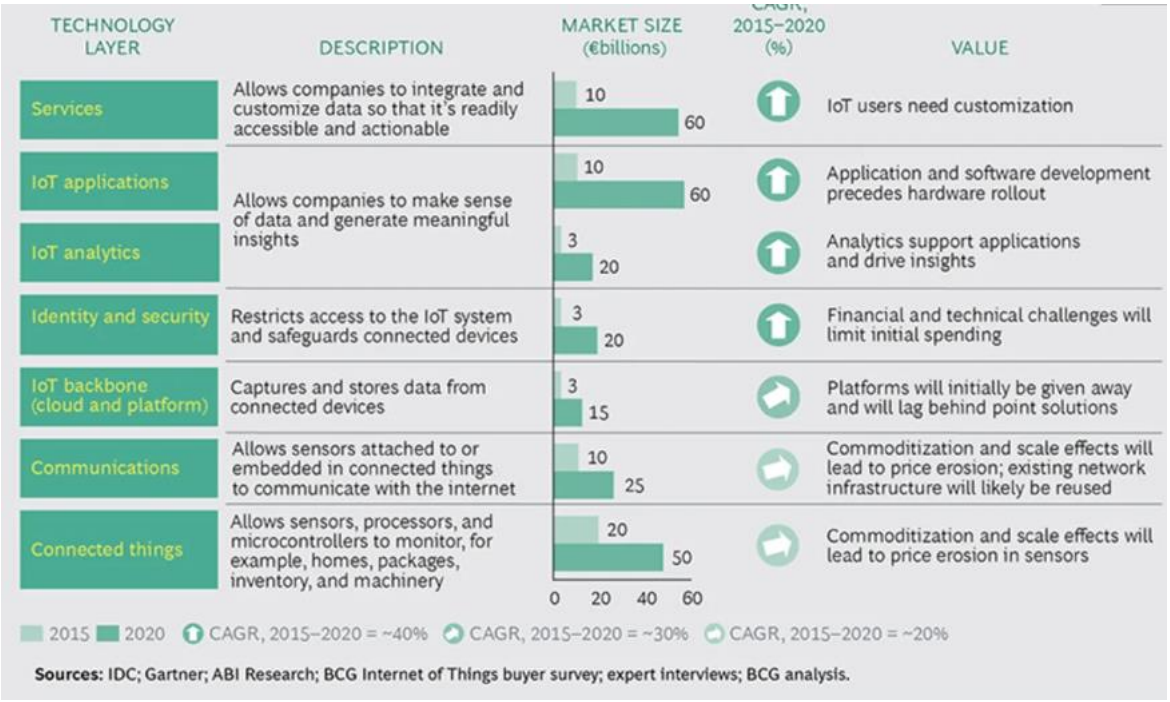

Figure 18. Distribution of Spending among different layers of IoT technology [12]

It is expected that by 2020, IoT analytics and applications layer have captured 60% of the growth from IoT.

5.3.2 Competitor Analysis

Considering the evolution of IoT and Edge markets, Cloud Vendors and traditional IT Equipment/Networks vendors are increasingly developing specific IoT and Edge computing solutions.

On the other hand, Cloud Vendors are fostering the partnership between systems integrators and professional services providers to enable offering full-scale end-to-end IoT solutions as reported by ABI Research, a global tech market advisory firm. ABI Research has analysed the partnerships and determined that “over 70% of the analysed partnerships provide data-enabled solutions, such as edge intelligence, streaming analytics, advanced analytics, and data science consulting services. So far in 2020, there is an increasing number of IoT edge suppliers partnering with Cloud providers and IT vendors, such as AWS, Cisco, SAP, IBM, and Microsoft. This indicates a departure from the traditional cloud-centric model and greater accessibility of edge technology in the IoT ecosystem”.

This section provides an overview of seven existing solutions in the market delivered by Traditional IT Equipment/Network Vendors and Traditional Cloud and Software providers.

After that, we will compare all the solutions with DECENTER Platform, following "customer-specific" purchasing criteria for IoT platforms [13].

5.3.2.1 Traditional IT Equipment/Network Vendors

- **CISCO**

CISCO IoT portfolio offers several essential IoT elements to deploy IoT applications in a secure connectivity environment:

- **Cisco IOx** enables developing IoT applications at the Edge. IOx combines IoT application execution within the Edge, secure connectivity with Cisco IOS Software, and powerful services for rapid, reliable integration within IoT sensors and the cloud [14]. This solution is composed of Cisco IOx Application Environment that combines Cisco IOS and the Linux OS, Fog Director that allows administrators to manage, administrate, monitor and troubleshoot edge applications, SDK and developments tools including methodology guidelines to help developers, and Fog applications ready for execution on IOx-enabled infrastructure.
- **Cisco Kinetic platform** allows to extract, compute, and move data from connected devices to several applications. This distributed system of software design the IoT operations by performing three key functions: extract data from disparate sources, compute data anywhere from Edge to destination to provide destination where it is needed, and move data to the right application at the right time for data distribution in multi-cloud and multi-locations situations [15]. Cisco Kinetic platform includes three modules: Gateway Management Module, Edge & Fog processing Module and Data Control Module.

- **DELL**
  - **Dell Boomi** is an Integration as a Service (IaaS) solution provided by Dell, which delivers a cloud service application, data, process, and Service-oriented-Architecture (SOA) integration scenario. It supports real-time integration and scales to meet the high-volume demands of mobile, extract, transform and load (ETL) and electronic data interchange (EDI) environments [16]. Additionally, it supports emerging use cases, including IoT, Edge and Blockchain.
  - **Dell Statistica** can be deployed at the Edge, using Dell hardware such as Dell Edge Gateway 5000 series, along with Dell Bloomi iPaaS.
- **INTEL**
  - **Intel Hardware** with Intel Core and Intel Atom processors [17] , to support many types of IoT devices, Intel Xeon Scalable Processors [18].
  - **Intel Software** with Intel Network Builders Edge Ecosystem [19] helps accelerate the development, adoption and deployment of edge-centric technologies, improving access to tested and optimized solutions for cloud and edge environments. In addition, Intel delivers an Intel IoT Market Ready solution [20] that provides end-to-end customized solutions for each industry.

5.3.2.2 Traditional Cloud and Software Vendors

- **Microsoft**
  - **Microsoft Azure IoT Suite:** This solution takes care of the work of deploying and orchestrating the various services to give the customer a complete end-to-end solution; it consists of the following services: IoT Hub, which serves to

communicate all devices to the Cloud and process massive volumes of data; Stream Analytics, which is a fully managed real-time analytics service that provides detection of anomalies from the IoT devices[21]; and Microsoft Azure IoT Suite, which also provides an Azure IoT device SDK that enables interoperability with gateways and facilitates programming.

- **Amazon**
  - **Amazon Web Services IoT** brings together data management and rich analytics in easy to use services designed for noisy IoT data. Amazon has structured its IoT offering in three blocks: Device Software, Connectivity & Control services and Analytics Services [22]. **AWS Greengrass** extends AWS to edge devices so they can act locally on the data they generate, while still using the cloud for management, analytics, and durable storage [23]**.**
- **Google IoT**
  - **Google Cloud IoT Core** allows fully managed services to easily and securely connect, manage, and ingest data from globally dispersed devices [24]. It also provides features to build and train ML models in the cloud. The main components of Cloud IoT Core are the device manager and the protocol bridges [25].
  - **Google Cloud Anthos** lets the customer build, deploy, and manage applications anywhere in a secure, consistent manner. The customer can modernize existing applications running on virtual machines while deploying cloud-native apps on containers in an increasingly hybrid and multi-cloud world. Google provides mainly this solution to Telecommunication companies, to let them run their applications wherever it makes the most sense [26].
- **SAP**
  - **SAP Cloud Platform IoT** provides scalable ingestion of sensor data as well as IoT device management and connectivity. It allows Cloud deployment and make decisions at the edge with a combination of real-time sensor and distributed data [27].
  - **Intelligent Edge Computing Software** deploys powerful microservices at the edge, specific insights and monitors real-time events [28].

After analysing the solutions, we can remark:

- Traditional IT/Network Equipment Vendors are trying to extend their long-established offering to provide software solutions to be able to provide end-to-end solutions. It is remarkable how Dell has developed Bloomi, an iPaaS platform that tackles interoperability issues.
- Cloud and Software Vendors are extending their offering portfolio with solutions to address applications in Edge and Cloud environments, although they continue to maintain the vendor-locking and interoperability limitations.

5.3.2.3 Comparison of platform vendors based on qualitative criteria

IoT Analytics [29]is a leading provider of market insights for the IoT Market, and recently has carried out a research and interviewed a number of customers of IoT/Edge platforms to know which are the criteria they follow when they choose their platform.

Eight common “customer-specific” purchasing criteria for IoT/Edge platforms have been identified [30], some of them are common to any technology decision-making and others are specific to IoT.

Next, we compare all the solutions described below with DECENTER, considering the 8 “customers-specific” purchasing criteria. The score will be L (low), M (medium), H (high), and Don’t know (DK).

Criteria	**Explanation**	**CISCO IoT**	**DELL Bloomi/ Statista**	**INTEL**	**Microsoft Azure IoT Suite**	**Amazon Web Services IoT**	**Google IoT**	**SAP**	**DECENTER**
Value for money	Expected Return on-Investment	M/L	M/L	M/L	M/L	M/L	M/L	M/L	H
Degree of Vendor lock-in	Flexibility to switch to another vendor if necessary	H	H	H	L	L	L	L	H
Platform Life Expectancy	To ensure the future availability of the platform	H	H	H	H	H	H	H	M
Size and Stability of the provider	To ensure the future availability of the provider	H	H	H	H	H	H	H	L
Support availability	Point of contact for support	H	H	H	H	H	H	H	L
Previous partnership with customer	Use existing relationships with positive experiences	DK	DK	DK	DK	DK	DK	DK	H
Strength of partnershi	Ensuring of Interoperability	H	H	H	L	L	L	L	H

p ecosyste m									
Complete ness of offering	End-to-end solution, covering all relevant functions	M	M	M	H	H	H	H	M

Table 7. Solutions comparison following the "customer-specific" purchasing criteria

5.4 Preliminary Identification of DECENTER Platform Business Models

During the first year of the project, an initial market analysis around Edge Computing was conducted and three BMs to exploit DECENTER outcomes were studied and reported within *D1.3. First Year Annual Report* [31]. Those BMs aimed at exploiting the DECENTER solutions and knowledge gained during the project by the partners. In this section, we focus on the BMs to exploit the DECENTER Platform, and we will include several BMs to exploit DECENTER knowledge in subsequent deliverable D2.3 DECENTER Business Models.

5.4.1 DECENTER Platform BM

During the second year of the project, DECENTER Consortium has discussed the BMs identified for DECENTER platform in the different project use cases. An internal remote workshop was carried out to talk about the different elements of the BMC and each partner provided its viewpoint regarding business, adoption barriers, and other factors affecting the commercialization of DECENTER solutions and also considering partners' profiles limitations.

An initial approach to the DECENTER Platform BM is as follows:

Key Partners	**Key Activities**	**Value Proposition**	**Customer Relationships**	**Customer Segments**
Services Providers (Cloud Providers, IoT providers, Infrastructure providers, etc.) Technical Partners	Operation of DECENTER platform Support Services	An open, secure and robust Fog Computing Platform to orchestrate cloud-to-edge resources Platform based on existing open-source frameworks working in Fog domain	Webinars and training courses Incorporation of new use cases demanding orchestrate of resources Strategic partnerships with services providers	Innovative SMEs developing Cloud/Fog/IoT solutions or services Innovative SMEs and companies working in AI Companies and SMEs working in

<table>
<tr><td></td><td>Key Resources

DECENTER components

Consortium Knowledge</td><td>Blockchain-based framework to allow cross-border dynamic binding of resources

AI models for cloud-to-edge computation</td><td>Channels

DECENTER project Website and online channels

Project Dissemination (Conferences, events, scientific publications)

Another related projects collaboration

Industrial events</td><td>vertical sectors with real time constraints

Services Providers

Software Developers

Municipalities

Robotics manufacturers

Construction companies

Home-services providers</td></tr>
<tr><td colspan="3">Cost Structure

Marketing

Maintenance of the platform

Personal costs

Services providers costs (Cloud providers, Blockchain costs)</td><td colspan="2">Revenue Streams

Platform-as-a-Service

Customized Revenue model based on projects</td></tr>
</table>

Table 8. Initial DECENTER Platform BM

This initial approach considers that DECENTER consortium or a group of partners will be willing to exploit DECENTER under the BM proposed. But there are still many open points regarding the BMs that will be refined in the final period of the project. To refine and select the most-suited DECENTER Platform BM, the information from the BMs validation, the stakeholders' feedback and partners' contributions will be considered.

Next, some considerations regarding the depicted BM:

- The BMC considers that DECENTER Platform is going to be commercialized by DECENTER Consortium or a group of partners. It could be considered to offer DECENTER Platform to be exploited by a third party, for example, a Cloud/IoT vendor. That approach delivers a different BMC that will be tackled in a subsequent deliverable.

- Key Partners: To exploit DECENTER Platform by a group of partners, a joint exploitation agreement will have to be signed by DECENTER consortium, which includes the role and responsibilities of each partner, some of them could play a key partner role if not interested in the commercialization due its profile. The BM needs to specify whether the Cloud Services are going to be offered along with the DECENTER Platform or not, as well as the IoT infrastructure.
- Customer Segment: In this BM are included all the potential customers for DECENTER Platform regardless of the project use case.
- Revenue Stream: After the workshop among partners, several questions regarding the revenue stream arose, considering that the revenue stream depended on the type of company that run the business to exploit DECENTER platform, for example, if Municipalities wants to offer services of crossing safety, those services has to be freemium for citizens. So, we need to consider several scenarios to exploit DECENTER Platform, in which different customers segments will be addressed with customized revenue streams.

The final deliverable D2.3 DECENTER Business Models will solve all the open points considered above and will also define different BMs for the vertical sectors related to the DECENTER use cases.

5.4.2 Business Models for DECENTER Use Case Partners

DECENTER solutions are being validated in four use cases of several industrial domains. These use cases are led by DECENTER partners with different profiles which limits the way they can exploit the project results. Also, each use case partner has identified a different customer segment and a revenue stream to address them.

Here, we outline a preliminary approach for the business models that may be adopted by the use case partners. We will analyse in-depth those BMs and adoption roadmaps for them, which will be reported in the D2.3.

UC1: Smart City Crossing Safety Business Model

The municipality of Trento expressed that its customers are the Citizens and the unique revenue stream envisioned is the Freemium services for citizens (Table 9).

Key Partners	Key Activities	Value Proposition	Customer Relationships	Customer Segments
Technical Partners (FBK) Infrastructure providers	Operation of DECENTER platform Support Services	An open, secure and robust Fog Computing Platform to orchestrate cloud-to-edge resources	Advertising	Citizens

<table>
<tr><td>(hardware, sensors, etc.)</td><td>Key Resources

DECENTER components

Consortium Knowledge</td><td>

Platform based on existing open-source frameworks working in Fog domain

Blockchain-based framework to allow cross-border dynamic binding of resources

AI models for cloud-to-edge computation</td><td>Channels

Advertising

City events</td><td></td></tr>
<tr><td colspan="3">Cost Structure

Marketing

Initial cost of devices, sensors, cameras, etc.

Maintenance of DECENTER platform (personal costs)

Maintenance of infrastructure (sensors, electric devices, electrical power, etc.)</td><td colspan="2">Revenue Streams

Freemium services for citizens</td></tr>
</table>

Table 9. Initial Smart City Crossing Safety Business Model

UC2: Robotics Logistics Business Model

Robotnik showed its interest in including some DECENTER features in the software development that they include in the robots they manufacture. They also revealed that its customers are mainly Logistics Robots' customers or Warehouses (Table 10).

Key Partners	**Key Activities**	**Value Proposition**	**Customer Relationships**	**Customer Segments**
Technical Partners	Support Services	An open, secure and robust Fog Computing Platform to orchestrate cloud-to-edge resources	Advertising Training courses	Logistics Robots' customers (Warehouses)

	Key Resources DECENTER components	Platform based on existing open-source frameworks working in Fog domain AI models for cloud-to-edge computation	**Channels** Same sales channels as used for other products (robots, etc.)	
Cost Structure Marketing Personal Costs (Software developers, hardware technicians) Others TBD			**Revenue Streams** Software development included in the robots. Cost added to Robots' prices	

Table 10. Initial Robotics Logistics Business Model

UC3: Smart and Safe Construction App Business Model

The University of Ljubljana is interested in providing Consultancy services to construction companies as advisory on the use of AI in construction (Table 11).

<table>
<tr><td rowspan="2">Key Partners

Infrastructure providers

Technical Partners</td><td>Key Activities

Operation of DECENTER platform

Support Services</td><td rowspan="2">Value Proposition

An open, secure and robust Fog Computing Platform to orchestrate cloud-to-edge resources

Platform based on existing open-source frameworks working in Fog domain

Blockchain-based framework to allow cross-border dynamic binding of resources

QoS models for AI applications deployed across the cloud-to-edge computing continuum</td><td>Customer Relationships

Webinars and training courses

Strategic partnerships with services providers</td><td rowspan="2">Customer Segments

Construction companies

Safety Services providers for Construction companies</td></tr>
<tr><td>Key Resources

DECENTER components

Consortium Knowledge</td><td>Channels

Promotional events in Construction sector</td></tr>
</table>

Cost Structure	Revenue Streams
Marketing	Customized pricing based on project
Maintenance of the platform	Consultancy services (in case of UL)
Maintenance of the electric devices (cameras, etc.)	
Personal costs	

Table 11. Initial Smart and Safe Construction app Business Model

UC4: Ambience Intelligence for Safety at Home and Around Business Model

KETI showed that its main customers could be innovative SMEs developing Cloud/Fog/IoT solutions or services targeted to intelligent services for smart office and that the best revenue stream would be subscription fees or customized pricing based on specific projects (Table 12).

<table>
<tr><th>Key Partners</th><th>Key Activities</th><th>Value Proposition</th><th>Customer Relationships</th><th>Customer Segments</th></tr>
<tr><td rowspan="2">IoT providers

Infrastructure providers

Technical Partners</td><td>Operation of DECENTER platform

Support Services</td><td rowspan="2">An open, secure and robust Fog Computing Platform to orchestrate cloud-to-edge resources

Platform based on existing open-source frameworks working in Fog domain

Blockchain-based framework to allow cross-border dynamic binding of resources

AI models for cloud-to-edge computation</td><td>Webinars and training courses

Strategic partnerships with services providers</td><td rowspan="2">People who want to rent a service to make their home safer

Schools/kindergartens

Nursing homes

SMEs that provides services of ambience intelligence/safety at home</td></tr>
<tr><td>Key Resources

DECENTER components

Consortium Knowledge</td><td>Channels

Promotional events</td></tr>
</table>

Cost Structure	Revenue Streams
Marketing	Subscription Fees
Maintenance of the platform	Customized pricing based on project
Maintenance of the electric devices (cameras, etc.)	Consultancy services
Personal costs	

Table 12. Initial Ambience Intelligence for Safety at Home and Around Business Model

6. Conclusions

Deliverable D2.2 is the centre masterpiece of the work performed in the context of WP2 activities. Its outcomes currently serve as guidance towards all other technology development and demonstration activities of the DECENTER project.

The consortium partners agreed that the presented architecture provides the means to position the project results in the overall landscape of cloud computing solutions with several novel features that particularly address the requirements for flexible and dynamic use of network, memory and compute intensive, highly decentralised AI services across the Cloud-to-Edge computing continuum. With this innovative architecture that is already becoming proven within WP5 activities, we believe that the DECENTER consortium has achieved a competitive edge which will serve as basis for the project's exploitation activities.

This deliverable concludes tasks T2.1 and T2.2 and presents the final results that were gained after the work performed during Y2 of the project. It provides our agreed and complete vision of the DECENTER architecture and detailed design of its components. As such it feeds towards the integration WP5. In addition, this deliverable provides information on the progress made in T2.3 (on business models) in year two of the project and provides indication of the work that will be carried out in the last year of the project. The work of WP2 is a fruitful outcome of the great collaboration between European and Korean partners the very exciting DECENTER project and is performed in a close collaboration with the activities of WP3, WP4 and WP5. Deliverables D2.1 and D2.2 therefore feed into the activities of these work packages and represent a central, well-thought and agreed masterpiece of the DECENTER project.

Section 2 gives the final description of the UCs and provides the final list of the DECENTER user and technical requirements. The outcome of this work is of interest for the activities of WP4, regarding the Digital Twin and for the activities of WP5, regarding the definition of demonstration KPIs.

Section 3 contributed to two main aspects: a) SLA model designs for the platform and b) the specification of Service quality Level Objectives (SLOs) for DECENTER. The outcome of this work is of interest for the architectural design and for the implementation activities taking place in WP3.

Section 4 describes the final version of the DECENTER architecture, offering architectures' component description and list of interfaces used by the components. The results of Section 4 are of interest for the activities of WP3, regarding the development and deployment of functionalities related to the Platform Layer, and the integration work package WP5.

In the final year of the project, the work on task T2.3 will continue. The outcome of T2.3 will be reported in D2.3, where in-depth analysis of the DECENTER business models and adoption roadmaps will be reported.

References

[1] DECENTER Deliverable D2.1; First release of the DECENTER architecture specification and use cases, 2019; Nascimento, J. Brežnik, S. Kleisarchaki, A. Leveghi, J. Moon, U. Paščinski, M. Savi and A. Soriano

[2] NFR_UC1_006 has been modified from 100ms in the D2.1 in 200ms because we believed that 100ms was too severe to be respected and at the same time 200ms is low enough to react fast to the problem

[3] https://www.wearemarketing.com/blog/business-value-proposition-canvas-methodology.html

[4] https://www.designabetterbusiness.tools/tools/business-model-canvas/

[5] http://alexosterwalder.com/

[6] https://www.strategyzer.com/canvas/business-model-canvas

[7] Eleni Agiatzidou, A. Lara Lopez, Ana Juan, Alexandros Kostopoulos, “Revisiting Business Models within Cloud Market”, 2015

[8] https://www.mordorintelligence.com/industry-reports/edge-computing-market-industry

[9] https://www.gsma.com/iot/wp-content/uploads/2018/11/IoT-Edge-Opportunities-c.pdf

[10] https://www.mordorintelligence.com/industry-reports/internet-of-things-moving-towards-a-smarter-tomorrow-market-industry

[11] https://www.bcg.com/publications/2017/hardware-software-energy-environment-winning-in-iot-all-about-winning-processes.aspx

[12] https://www.bcg.com/publications/2017/hardware-software-energy-environment-winning-in-iot-all-about-winning-processes.aspx

[13] https://iot-analytics.com/iot-platform-selection-3-best-practices/

[14] https://www.cisco.com/c/dam/en/us/products/collateral/cloud-systems-management/iox/at-a-glance-c45-737316.pdf

[15] https://www.cisco.com/c/dam/en/us/solutions/collateral/internet-of-things/kinetic-at-a-glance.pdf

[16] https://boomi.com/platform/what-is-ipaas/

[17] https://www.intel.es/content/www/es/es/products/processors/core.html

[18] https://www.intel.es/content/www/es/es/products/processors/xeon/scalable.html

[19] https://networkbuilders.intel.com/network-technologies/networkedgeecosystem

[20] https://www.intel.es/content/www/es/es/internet-of-things/market-ready-solutions/market-ready-solutions.html

[21] https://azure.microsoft.com/es-es/blog/microsoft-azure-iot-suite-connecting-your-things-to-the-cloud/

[22] https://aws.amazon.com/iot/

[23] https://aws.amazon.com/greengrass/?nc1=h_ls

[24] https://cloud.google.com/iot-core

[25] https://cloud.google.com/iot/docs/concepts/overview

[26] https://cloud.google.com/anthos

[27] https://www.sap.com/products/iot-platform-cloud.html

[28] https://www.sap.com/spain/products/edge-services.html

[29] https://iot-analytics.com/

[30] https://iot-analytics.com/iot-platform-selection-3-best-practices/

[31] DECENTER Deliverable; D1.3: First Year Annual Report [M12], 2019; D. Siracusa, M.Savi, M. Onofrio, R. Giaffreda, I. Cuadrado, A. Belen Gonzalez, S. Kleisarchaki, L. Gurgen, S.Lee, K. Kim, S. Kum, J. Moon, S.Lee, A. Leveghi, V. Stankovski, U. Pascinski, J. Breznik

[32] DECENTER Deliverable; D3.2: Resource Orchestration Strategies, 2020; U. Paščinski, P. Kochovski, V. Stankovski, F. Faticanti, M.Savi, O. Garcia, K. Kim

[33] DECENTER Deliverable; D3.3: Second release of the fog computing platform, 2020; I.Cordero, A. Garcia, R. Doriguzzi-Corin, S. Cretti, R. Giaffreda, S. Kum, U. Paščinski, P. Kochovski, V. Stankovski, C. Munilla, S. Lee

Abbreviations

AI:	Artificial Intelligence
GPU:	Graphics Processing Unit
IoT:	Internet of Things
ML:	Machine Learning
QoS:	Quality of Service
SLA:	Service Level Agreement
SLO:	Service Level Objectives
UC:	Use Case
WP:	Wok Package