A distributed Embedded Systems IoT platform and Associated services Supporting Shopping Cart for Disabled People

Konstantinos Antonopoulos
Department of Electrical and Computer Engineering
University of Peloponnese
Patra, Greece
k.antonopoulos@go.uop.gr

Dimitris Karadimas
Department of Electrical and Computer Engineering
University of Peloponnese
Patra, Greece
d.karadimas@esdalab.ece.uop.gr

Alexandros Spournias
Department of Electrical and Computer Engineering
University of Peloponnese
Patra, Greece
a.spournias@esdalab.ece.uop.gr

Christos Panagiotou
Department of Electrical and Computer Engineering
University of Peloponnese
Patra, Greece
ch.panagiotou@esdalab.ece.uop.gr

Ignantios Fwtiou
TOBEA SMLLC Old National Road
Patras-Athens 289
Patra, Greece
fotiou@tobea.gr

Ioannis Symeonidis
Hellenic Institute of Transport
Center for Research & Technology Hellas
Thessaloniki, Greece
ioannis.sym@certh.gr

Christos Antonopoulos
Department of Electrical and Computer Engineering
University of Peloponnese
Patra, Greece
ch.antonop@uop.gr

Michael Hübner
Brandenburg, University of Technology
Cottbus–Senftenberg, Germany
michael.huebner@b-tu.de

Nikolaos Voros
Department of Electrical and Computer Engineering
University of Peloponnese
Patra, Greece
voros@uop.gr

***Abstract*—Over the last few years advancements in technological domains such as embedded system, IoT and distributed communication have greatly propelled the development of platforms and systems enabling people with disabilities to equally participate in everyday life activities. The added value of such platforms is even more pronounced when respective activities are of outmost importance like going to the supermarket and performing all the functions and services that a person without impairments can enjoy. In this context this paper presents and analyzes an end-to-end architecture where, distributed embedded systems intelligence, distributed data communication techniques and open data interfaces approaches are leveraged so that people with mobility impairments can be fully independent and functional when visiting a supermarket. In-depth analysis is offered on critical components and services such as, the specialized cart movement control unit, multifaceted localization techniques and automated pricing and merchandize management services. It is noted that this work is the outcome of a Greek National Research project supported by one of the biggest supermarket companies in Greece.**

Keywords—Distributed Systems, Networking, Internet of Things, Disabled People, Embedded Systems, Event-driven architectures.

I. Introduction

The inclusion of people with mobility impairments in daily life activities that they can perform independently and autonomously is a critical milestone for these people to participate in the society as equal members and thus in improving their quality of life [1]. Over the last years, with the rapid development of technology, many of these activities are becoming easier for an increasing portion of people with mobility difficulties. One of the main activities of every people is the purchase of goods that is mandatory for them to live (food, water etc..). These activities can take part in many places, but a common place where people can find all these goods gathered, are the supermarkets.

However, inside a supermarket a high number of challenges must be faced by the targeted social groups resulting from the accumulation of many moving people (and obstacles), especially during rush hours, narrow corridors, and the difficulty to navigate specialized supermarket carts. Additionally, a particular difficulty concerns the easy overview and management of the context of the supermarket basket and final the billing of their purchases.

The mobility problem can be addressed by the construction of specially designed strollers/carts [2]. However specialized movement control infrastructure and services are required to offer a holistic and practical solution as presented in this paper. In addition, through localization services these people can navigate easily to such places. Using their smart devices (tablets/smartphones), can observe their location in the supermarket, along with possible obstacles between the corridors, in order to avoid them. Finally, the location of the goods that need to acquire will be available to their smart device monitor. Additionally, applications that allow people to create their own shopping lists with their desired products, before reaching the area of supermarket, is another way to help them for an efficient navigation in the supermarkets. Finally, automated purchase management tools, are critical, allowing these portions of people to avoid staying in the cashier queues.

A key challenge of such applications is the establishment of an efficient collaboration mechanism in order to provide a complete end-to-end solution to end-users. The event-based nature of such systems, drives their architecture to follow an event-driven communication paradigm, utilizing protocols such as MQTT[1], AMQP[2]. Movement entities (human, objects, carts etc..) are changing their positions in a non-deterministic way, thus the sub-systems must observe all these changes, and

[1] https://mqtt.org/

[2] https://www.amqp.org/

978-1-6654-8128-1/22/$31.00 ©2022 European Union

respond by broadcasting events, for sharing this information and commands. Finally, all these events must comply to different constraints in terms of latency and reliability.

The rest of the paper is as follows. Section II presents the end-to-end architecture of the proposed system along with the corresponding services. Section III, provide a brief analysis of the localization services provided by the proposed system. Section IV and V, analyze the movement control service, and the exposed APIs for the end-users to purchase their products, respectively. Finally, Section VI, concludes the paper.

II. End to End Architecture & Services

A critical metric of such systems is their adaptability to different requirements posed at different levels of the system. This leads their design into more hierarchical architectures, where each layer is implemented independently, focusing on the constraints that must fulfill. Each layer is operating autonomously, applying different utilization of resources, hardware platforms, traffic communication patterns, security policies without jeopardizing the adaptability and extendibility of the system. The proposed hierarchical architecture, comprised of three layers, as shown in Fig 1.

At the top level, is the ***Public Layer***, where all the necessary APIs are exposed to the end-users for integrating with the system. Moving downwards, to the ***Private Layer***, where the most computational tasks are executed. The main functionality of this layer is to process, transform, and forward the information between the corresponding layers, as well as to apply rules to the overall communication flow. Additionally, it maintains all the necessary information about the Edge layer. Finally, at the bottom of the architecture is the ***Edge Layer***, which comprises mainly by the devices installed in the physical environment, utilized for observing, and acting to different type of events. A critical part inside the Edge layer are the local gateways that handle the inter-connection and orchestration of all devices. Due to the event-driven nature of the targeted system, the use of publish/subscribe messaging protocols are ideal for inter-exchanging information. In that respect, we utilize the MQTT protocol as the core messaging protocol.

More specifically, at each layer of the system a MQTT cluster is maintained, with load balanced MQTT topics, where different components of the system use to publish intra-layer messages. Additionally, specialized bridged MQTT topics are utilized, focusing on broadcast events to different layers of the system, allowing inter-layer communication. Our MQTT clusters are based on the EMQX broker[3].

A. Public Layer

End-user applications are only allowed to integrate through the public layer with the system. In this layer, MQTT APIs are exposed, for generating shopping-related information ("***eshop/#***"), along with authentication services. Furthermore, when the users are inside the surrounding area of the supermarket, location-related information are available, through "***locations/{locationId}/positions***" topic patterns. These broadcasting events contain the location of the users and the products, that want to purchase. At the same time, users can still view their shopping lists, along with the purchased products.

B. Private Layer

The private layer is the aggregating and processing layer of the proposed system. Its main responsibility is to process data streams originated from Edge layer and generated events that are to be forwarded to the corresponding layers. Location-related data streams are processed, originated from edge layer, (topic patterns "***carts/{cartId}/location***", "***obstacles/{cartId}/rssi***" and "***carts/{cartId}/ultrasonic***"), and the output is forwarded to the public layer, using the "***locations/{locationId}/positions***" topic pattern. Finally, data fusion techniques are applied to detect obstacles in supermarket, which in turn are forwarded back to the Edge layer (topic "***events/obstacles***"), informing movement control services for any obstacle presence.

C. Edge Layer

Finally, at the bottom of the architecture is the *Edge Layer*, where the utilized embedded systems operate. The connection point of the edge layer with the upper layers is accomplished through the referenced IoT gateway [3]. Besides the gateway, sense and actuation devices are installed in this layer, providing information consumed by location and assisting shopping services.

III. Localization Services

Localization Services play a crucial role in the daily activities and even more when focusing on persons with mobility impairments [4]. These services are mandatory in the context of assisted shopping because it provides valuable information for calculating optimal routes, allowing end-users to navigate in the super-market by avoiding areas congested with people or objects/obstacles. The proposed system implements two different types of localization services exploiting different types of embedded devices and serving different application scenarios. The first type of localization is related to the location of the end-user and the shopping cart that is in use in the area of the supermarket. The second type is related to the identification of obstacles (mobile or not) inside the supermarket. We choose to implement two types of localization in order to fulfill different application requirements in the localization services. Regarding the first type of localization, we need an accurate localization prediction service, thus, we apply the Simultaneous Localization and Mapping (SLAM) method [5]. The second type offers, a flexible, extendible, and easily deployable in any embedded system localization prediction service by utilizing BLE localization methods [6].

A. Cart Localization based on Lidar Laser

Simultaneous Localization and Mapping (SLAM) [5], is a core capability for an object to explore and understand its environment. There are several studies and implementations of SLAM methods, which consider odometer-related information. However, in our case this is not applicable due to lack of support from the utilized platform.

[3] https://www.emqx.io

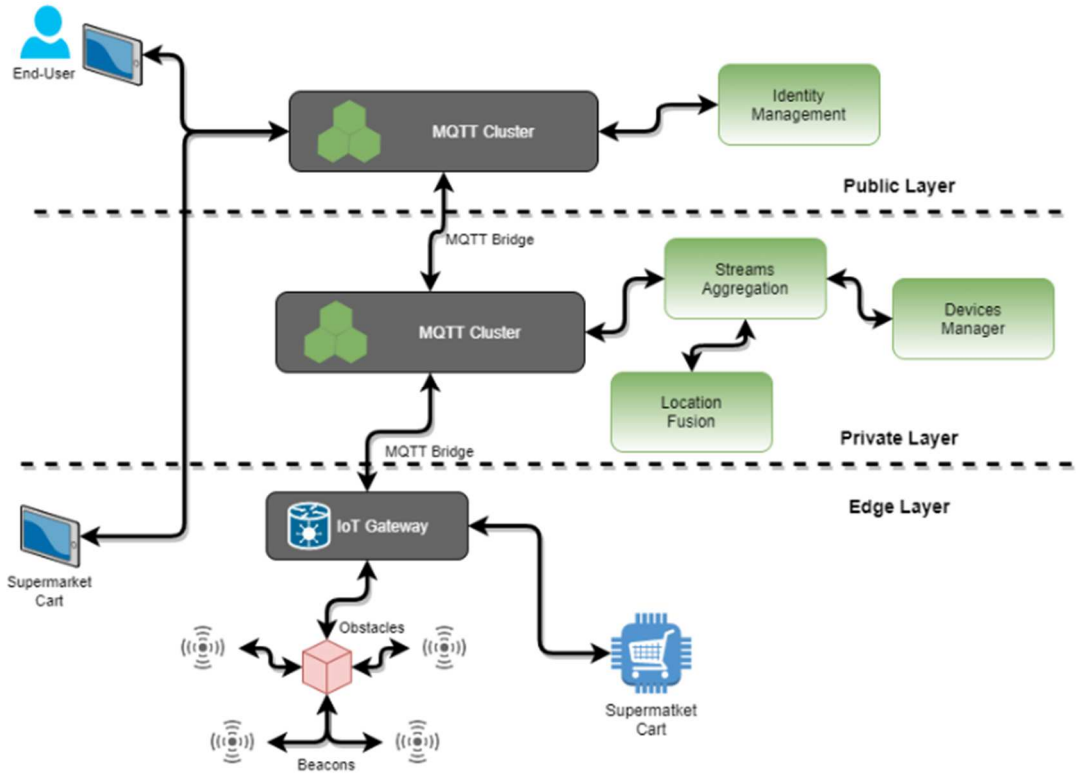

Fig 1. End-To-End Architecture

In contrast, a SLAM method uses solely the information from the laser scanner sensor in order to acquire all the necessary information.

The proposed method is based on a lidar device to generate a grid map of the environment that needs to be mapped, while the basic Hector SLAM method performs the actual mapping [7]. The scan matching algorithm adopted in our methodology is based on the Gauss-Newton approach [8]. Concerning the hardware requirements, the Hector method uses a 2D LiDAR LMS100 sensor [9], as well as an Intel Atom Z530-based processor board [10]. For the conversion of laser's data into odometer information, the *TF-ROS*[4] *package*, along with *Scan Matching* method [11] are adopted. The transformation is based on the conversion of information between the laser_link frame and the combination of base_link – base_footprint frames [12]. These results to the calculation of the cart's position which is accessible through ROS topic "**/pose2d**". This is the first step for the ROS map transformation flow, as shown in Fig 2.

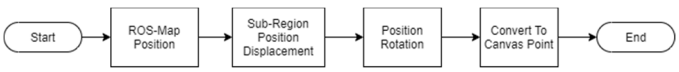

Fig 2. Map Transformation Flow

The next step is to acquire the cartesian position as exposed from ROS [5] topic ("**/pose2d**"). Thus, we have implemented a specific ROS package (namely, ros-map-bridge), that running to the movement control unit (MCU) and implements all the steps of the proposed transformation flow.

A challenge here is the definition of the ROS map. A ROS map has a fixed, predefined width and height, where during the SLAM process the valuable map is generated. This generated map is the actual map (supermarket area) that we want to visualize to the end-users. Based-on this, we annotate and define a sub-region on the predefined ROS map, where the actual transformation process takes place, when the cart is located inside this sub-region. Next, we perform the displacement of the original ROS cartesian point, to be matched to the cartesian system of our sub-region, that complies with the following formula:

$$(x, y) \in [A \times B] \Longrightarrow (x', y') \in [A' \times B'] \tag{1}$$

where, $[A \times B]$ is the ROS map, and $[A' \times B']$ is our sub-region. Moving to the next step of the flow, we perform the rotation of the point (x', y') for θ degrees, if necessary. The rotation degrees are calculated based on the map generated from the SLAM process and the Canvas map of the corresponding supermarket, that shown in the end-user application. The rotation formula is the following:

$$x_r = (\cos\theta \times x') - (\sin\theta \times y') \tag{2}$$

$$y_r = (\sin\theta \times x') + (\cos\theta \times y') \tag{3}$$

Finally, the calculated position (x_r, y_r) transformed to a canvas-based point using the following equations:

$$P_x = \left({}^{C_w}/_{2}\right) + \left(x_r \times \left(\left({}^{1}/_{R_{res}}\right)/(Cw_{res})\right)\right) \tag{4}$$

[4] http://wiki.ros.org/tf2

[5] https://www.ros.org/

$$P_y = \left({}^{C_h}/_{2}\right) + \left(y_r \times \left(\left({}^{1}/_{R_{res}}\right)/(Ch_{res})\right)\right) \quad (5)$$

where, C_w and C_h are canvas width and height (end-user application), R_{res} is the ROS map resolution (default 0.05), and Cw_{res} , Ch_{res}, are width, height map scaling between the 1:1 canvas map with the sub-region we define and the actual canvas map shown in the end-user application.

B. Obstacles Localization based on BLE sensors

As indicated, the main objective of the proposed localization subsystem is to identify any possible obstacles located inside the area of the supermarket and inform the relevant entities, about their presence. In that respect, we design and implement an BLE-based localization system [6], for estimating the position of any obstacles.

Starting from the deployment of the localization system, we install in the corridors of the supermarket, TI-Sensor Tag BLE beacons [13], at a maximum distance of 5m between them. The BLE beacons are configured to advertise their presence every 100ms. Next, on each obstacle that we want to know its position, we attach an ESP32 hardware device. ESP32 device is listening for BLE advertisements and every 2sec it switches to Wi-Fi and transmits the monitored RSSIs to the gateway. Sequentially, the gateway forwards this information to the upper layers for further processing and the generation of corresponding location-related events. The overall flow shown, in Fig 3 below.

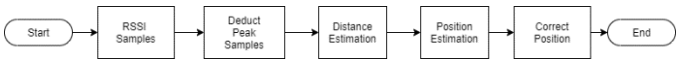

Fig 3. BLE Localization Flow

The first steps are related to the collection of RSSI values and the deduction of the weak RSSIs in favor of the strong ones. Here, we make a simple assumption, that high RSSIs are correct observations since they cannot be observed in-far distances. On the other hand, weak RSSIs can be results of physical phenomena, such as fluctuations or obstacles between the transmitter and receiver that weaken the signals. By observing strong RSSIs, we can conclude that the tracking objects are in-close distance from the transmitter. Based-on this assumption, we proceed with the following steps.

First, we "*clean*" the signal by removing the outliers (low RSSIs). This is accomplished by comparing the lowest and highest RSSIs. In case the deviation between these RSSIs is greater than a predefined threshold, then the lowest RSSIs will be excluded from the evaluated samples. This threshold is determined empirically during the installation of the system and determined according to the peculiarities of the environment. Given that A= $\{-70, -65, -45, -25, -33\ldots\}$ is the original RSSI set, applying the outlier step will generate a RSSI sample $B \subseteq A$, with the lowest RSSI values excluded, B= $\{-65, -45, -25, -33 \ldots\}$.

Next, we apply the RSSI rating mechanism to B set. The scope of the rating mechanism is to isolate a subset $C \subseteq B$, that contains at least three RSSI samples, with as much as possible high RSSI samples, which we name it "***envelope***". The first routine of the rating mechanism extracts a set of possible envelopes, where the second routing is rating each envelope, using the following equation:

$$rate = \frac{\sum_{i \in C}^{n} x_i > mean}{\sum_{i \in C}^{n} x_i < mean} \quad (6)$$

The "***envelope***" with the highest rate is used by the next step of the proposed flow, to determine the actual distance between the beacons and the receivers. In case of two or more envelopes with same rating, the envelope with more samples is selected. Even in a clear area, without any obstacles, the RSSI can be very noisy, making the normalization of the signal a critical part of the localization systems. Thus, we apply the Kalman Filter [14] to smooth the collected RSSIs.

The final calculated RSSI is used to determine the actual distance between transmitters and receivers, using the Log-Normal Shadowing Model [15], solved in term of distances as shown,

$$d = d_0 * 10^{A - RSSI - {}^{X_g}/_{10_n}} \quad (7)$$

Moving forward, the next step is to estimate the position of the obstacle in the corridors. The position can be calculated using trilateration algorithms [16]. To solve the trilateration problem, we selected the non-linear least squares solver [17]. In the trilateration problem, the accuracy is related to the total reference points we use as input. In-reality trilateration error arises due to the inaccuracy in measuring distances and mapping reference points and is largely affected by the geometrical arrangement of the reference points and the object. As a result, the involved circles or spheres may not intersect at the actual position of the object, or even may not intersect at all. Thus, it is necessary to determine a "near optimal approximation" [18].

Zone	***Distance (m)***	***Selection***
Immediate	≤ 1.5	Used
Near	>1.5 & ≤ 4	May Used
Far	> 4	Should Rejected

Table 1. Distance Classification Zones

In that respect we have defined three classification zones based on the estimated distance, as shown in Table 1. These zones were used to support the decision of whether to include or not the distance to the position estimation algorithm. Given the fact that the trilateration algorithm needs at least three reference points to estimate the position, we must find three reference points, that belong to any classification zone, in favor of the *Immediate* zone. Finally, previous positions estimations are used to correct the latest estimation.

IV. Movement Control Service

One of main components of the IoT platform is the Movement Control Unit (MCU). MCU is responsible for:

(a) Controlling the motors of the platform. As the structure is constructed to host a wheelchair mounted to a properly configured shopping cart; its movement relies on two hub motors for forward and backward movement and two step motors for clockwise and counterclockwise turning the axis of the hub motors by 90 degrees.

(b) Operating the motors with user controls for direction, speed, handbrake as well as selection of mode

of movement (either forward/backward or left/right). User controls encapsulate an analog two-axis joystick for speed, push buttons for selection of mode of movement and a switch for handbrake.

(c) Obstacles detection and avoidance based on three input streams aggregated in the MCU middleware via a local MQTT broker. Input streams for obstacle detection originate from a set of ultrasound sensors mounted in the perimeter of the structure, signals from the camera mounted in the shopping cart, responsible for goods detection and a lidar sensor mounted in the front of the structure.

(d) Aggregating and periodically upload the status of the structure to the upper components of the platform for maintenance, analytics, etc. The status upstream is also implemented by subscribing to an MQTT broker which is available and visible to the WLAN hosting the IoT platform.

The core part of the MCU is realized by a set of Raspberry Pi which are locally networked and communicate via MQTT. One of these Raspberry Pi plays the role of the master, providing the local communication layer, aggregating all input streams from various hardware parts and executing a simple Finite State Machine (FSM) for movement control and obstacle avoidance. Master Raspberry Pi provides the local MQTT broker and is also responsible for upstreaming MCU state to public MQTT topics. Motors and human controls (joystick and movement mode selection) is accomplished using extra components, wired with the master Raspberry Pi as depicted in Fig 4.

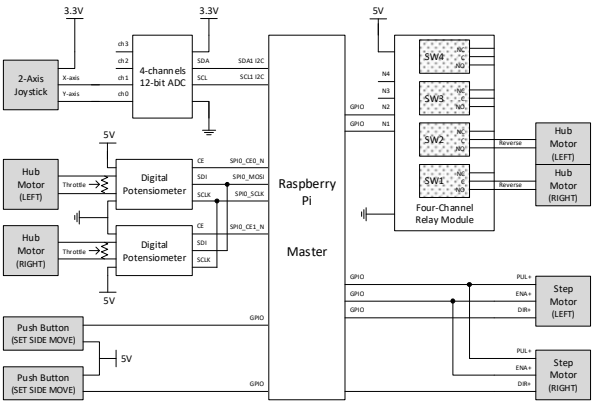

Fig 4. Wiring Diagram of MCU master

Specifically, the extra components that have been used are: ADS1015 4-channel ADC, 2 digital potentiometers MCP4151 and a 4-channel 5V relay module. Finally, master Raspberry Pi aggregates obstacle avoidance sensor values realized from ultrasound sensors, as described in Fig 5,

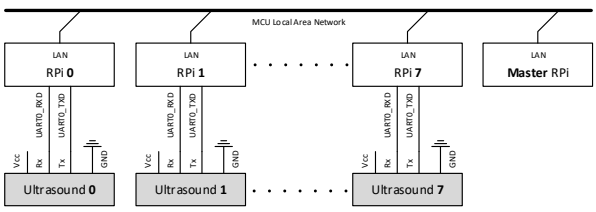

Fig 5. MCU Local Area Network Realization

Fig 6, presents a photo from a real structure implementation in lab environment. Red arrows annotate hub and step motors as well as position of obstacle avoidance sensors (8-ultrasound sensors and 1-lidar). Special compartments within the structure also exist and are dedicated for batteries, electronics as well as cabling around the structure.

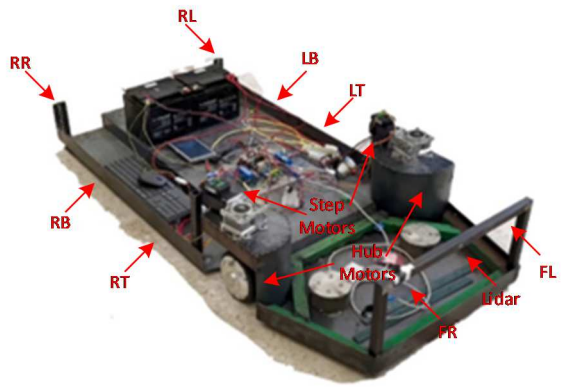

Fig 6. MCU Implementation

V. Communication Interfaces

In this section, auxiliary APIs are presented, focusing on functionalities that do not only apply to people with mobility impairments, but for the general convenience in the context of an easy and fast purchase of products. The interfaces provided to the end-users, comprise critical parts of such systems, and more precisely the ease of use and their extendibility

A. User Application

Starting from the user application, which is the graphical user interface of the proposed system, and it used by the end-users installed on a smartphone or tablet. This application is operating in two modes, *user-mode*, and *cart-mode*. In the user-mode the application is installed on users' devices, allowing the users to create/update their shopping lists, and view their purchases, shopping lists (historical data) etc. In the cart-mode, the application contains all the functionalities from user-mode plus interfaces that visualizes the surrounding area of the supermarket. The communication of the application with the platform relies solely on the MQTT protocol. The integration with shopping-related functionalities (topic pattern "***eshop/#***") is following the Request/Reply paradigm, while the location-related functionality is following an event-based paradigm (topic pattern "***locations/{locationId}/positions***").

B. Smart Shopping Basket

In the context of the proposed system, a smart shopping basket have been implemented, and installed in the shopping cart. This functionality allows the system to track the acquired products from the user, and this information can be used later from the automating pricing services. The scope of this functionality is to identify the products that picked from the user and dropped in or dropped out to/from the smart basket and publish the correlated events (topic "***eshop/shopping/basket***"). In turn the system will

put/remove the product to/from the electronic basket. Finally, the user application will be updated, showing the related information.

C. Shopping List & Automated Pricing Management

Finally, when the users have been acquired all the products that have in their shopping lists, they have to proceed to supermarket cashiers in order to pay for the acquired products. For this functionality, the system provides the option to the users to pay directly from their applications, without having to wait to the supermarket cashier queues. The integration with the system is following the Request/Reply communication paradigm, using the following topic "***eshop/purchases***".

VI. CONCLUSION

Designing and developing platforms enabling persons with mobility impairments to participate as equals to everyday life activities is a critical challenge for present and future societies. Leveraging advancements in a wide range of ICT technologies such as embedded systems, IoT and distributed control can make this vision as reality nowadays. In this paper we present a respective methodological approach concerning an end-to-end super maker platform for the specific sensitive social group. Our presentation focuses on challenges and respective solution on key components such as movement control unit of specialized carts, localization services and backend APIs for 3rd party software services under a holistic architecture.

ACKNOWLEDGMENT

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH CREATE INNOVATE (project code: T1EDK- 04183 EQUAL).

REFERENCES

[1] J. Carver, A. Ganus, J. Ivey, T. Plummer and A. Eubank, "The impact of mobility assistive technology devices on participation for individuals with disabilities.," in *Disability and Rehabilitation: Assistive Technology. 11.*, 2015.

[2] C. Antonopoulos, P. Kelverkloglou, N. Voros and I. Fotiou, "Developing autonomous cart equipment and associated services for supporting people with moving disabilities in Supermarkets: The EQUAL Approach," in *Information, Intelligence, Systems and Applications*, 2020.

[3] C. Antonopoulos, K. Antonopoulos, C. Panagiotou and N. Voros, "Tackling Wireless Sensor Network Heterogeneity Through Novel Reconfigurable Gateway Approach," in *Applied Reconfigurable Computing. Architectures, Tools, and Applications - 14th International Symposium*, Santorini, May 2-4, 2018.

[4] F. Routhier, W. B. Mortenson, L. Demers, A. Mahmood, H. Chaudhury, K. A. Martin Ginis and W. C. Miller, "Mobility and Participation of People With Disabilities Using Mobility Assistive Technologies: Protocol for a Mixed-Methods Study," *JMIR research protocols,* 2019.

[5] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. D. Reid and J. J. Leonard, "Simultaneous Localization And Mapping: Present, Future, and the Robust-Perception Age," *CoRR,* vol. abs/1606.05830, 2016.

[6] L. Bai, F. Ciravegna, R. Bond and M. Mulvenna, "A Low Cost Indoor Positioning System Using Bluetooth Low Energy," *IEEE Access,* vol. 8, pp. 136858-136871, 2020.

[7] S. Kohlbrecher, O. v. Stryk, J. Meyer and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2011.

[8] S. Gratton, A. Lawless and N. Nichols, "Approximate Gauss–Newton Methods for Nonlinear Least Squares Problems," *SIAM Journal on Optimization,* vol. 18, pp. 106-132, 2007.

[9] S. S. Intelligence, "2D LiDAR sensors LMS1xx / Indoor," [Online]. Available: https://www.sick.com/in/en/detection-and-ranging-solutions/2d-lidar-sensors/lms1xx/lms100-10000/p/p109841.

[10] Z.-I. A. Processor. [Online]. Available: https://ark.intel.com/content/www/us/en/ark/products/35463/intel-atom-processor-z530-512k-cache-1-60-ghz-533-mhz-fsb.html.

[11] L. V. Jixin, "Scan Matching and SLAM for Mobile Robot in Indoor Environment," 2016.

[12] R. S. Setup. [Online]. Available: http://wiki.ros.org/hector_slam/Tutorials/SettingUpForYourRobot.

[13] "Texas Instrument. Multi-Standard CC2650 SensorTag Design Guide.," [Online]. Available: https://www.ti.com/lit/ug/tidu862/tidu862.pdf. [Accessed 2022].

[14] Y. Sung, "RSSI-Based Distance Estimation Framework Using a Kalman Filter for Sustainable Indoor Computing Environments," vol. 8, no. 11, 2016.

[15] R. Desimone, B. M. Brito and J. Baston, "Model of indoor signal propagation using log-normal shadowing," in *Published in: Systems, Applications and Technology Conference (LISAT),*, Long Island, 2015.

[16] M.-S. Huang and R. Narayanan, "Trilateration-Based Localization Algorithm Using the Lemoine Point Formulation," *IETE Journal of Research.,* pp. 60-73, 2014.

[17] C. T. Kelley, Frontiers in Applied Mathematics: Iterative Methods for Optimization, 199.

[18] Y. Zhou, "An Efficient Least-Squares Trilateration Algorithm for Mobile Robot Localization," in *Published on IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis USA., 2009 .