

Article

Smart Warehouse Management System: Architecture, Real-Time Implementation and Prototype Design

Muhammad Gufran Khan ^{1,*} , Noor Ul Huda ¹ and Uzair Khaleeq Uz Zaman ²

¹ Department of Electrical Engineering, National University of Computer and Emerging Sciences, Chiniot-Faisalabad 38000, Pakistan; f219752@cfed.nu.edu.pk

² College of Electrical & Mechanical Engineering, National University of Sciences and Technology, Islamabad 43701, Pakistan; uzair.khaleeq@ceme.nust.edu.pk

* Correspondence: m.gufran@nu.edu.pk

Abstract: The world has witnessed the digital transformation and Industry 4.0 technologies in the past decade. Nevertheless, there is still a lack of automation and digitalization in certain areas of the manufacturing industry; in particular, warehouse automation often has challenges in design and successful deployment. The effective management of the warehouse and inventory plays a pivotal role in the supply chain and production. In the literature, different architectures of Warehouse Management Systems (WMSs) and automation techniques have been proposed, but most of those have focused only on particular sections of warehouses and have lacked successful deployment. To achieve the goal of process automation, we propose an Internet-of-Things (IoT)-based architecture for real-time warehouse management by dividing the warehouse into multiple domains. Architecture viewpoints were used to present models based on the context diagram, functional view, and operational view specifically catering to the needs of the stakeholders. In addition, we present a generic IoT-based prototype system that enables efficient data collection and transmission in the proposed architecture. Finally, the developed IoT-based solution was deployed in the warehouse of a textile factory for validation testing, and the results are discussed. A comparison of the key performance parameters such as system resilience, efficiency, and latency rate showed the effectiveness of our proposed IoT-based WMS architecture.

Keywords: automation; Industrial IoT; Industry 4.0; warehouse management systems; architecture modeling; MQTT



Citation: Khan, M.G.; Huda, N.U.; Zaman, U.K. Smart Warehouse Management System: Architecture, Real-Time Implementation and Prototype Design. *Machines* **2022**, *10*, 150. <https://doi.org/10.3390/machines10020150>

Academic Editors: Krzysztof Żywicki and Jose L. Martinez Lastra

Received: 22 December 2021

Accepted: 8 February 2022

Published: 18 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As the world is advancing toward the future, technology is rapidly shifting its shape. Automation is taking over industries, and gone are the days when technology was merely used in hi-tech labs by scientists. Currently, almost every item in a household can be controlled with just one click, and nothing is untouched by the marvel of automation. Internet-of-Things (IoT)-based automation has opened up a huge number of applications and is the key technology for the Industry 4.0 revolution. The IoT can be defined as a network of automated systems that enable the user to control and coordinate devices in the mesh of the Internet. It helps in decentralizing control and management. In this era of the progression of industry into the digital world [1], the Industrial Internet of Things (IIoT) is transforming conventional industrial processes into Cyber-Physical Systems (CPSs) [2,3]. From the manual maneuvering and managing of heavy goods to automating processes, Industry 4.0 technologies aim to perform faster and more efficient work in less time [4].

Warehouses play an important role in the industry supply chain, as well as the production of any industrial unit. The whole flow of a company runs smoothly with respect to its warehouse, as it is used to store, manage, and track the goods. The incoming goods are received and stored in proper storage units by assigning a tracking ID. These tracking

IDs are also stored in a register or a computerized database against the product details. For the finished products that need to be delivered out of the warehouse, the information against their tracking IDs is updated in the database. In most warehouses, these tasks are mostly handled manually or with some partial automation [5,6]. As the company's trade volume increases, the number of supply chains of goods also increases. This takes the form of a complex and integrated supply chain network in which many storage units and racks of a warehouse can be linked to each other or work independently [7]. The maintenance and management of such a complex network are challenging, and even more tedious tasks are monitoring and tracking all the goods. The need is to effectively manage all the ongoing processes of a warehouse and design a dynamic storage facility that can be accessed easily [8,9]. Conventionally, the goods reaching the inventory are identified at the check-ins by either the Sales Order (S/O) or some ID, associated with them in the production department. An efficient way is to use automated tags or IDs; nonetheless, it is pertinent to tag every item or product reaching the inventory, which makes it easier to be tracked by the system.

Different types of warehouses have their own storage allocation mechanisms depending on their needs and requirements. The storage area of a warehouse can be located adjacent to the whole supply chain or in some separate area to manage the transport expenses while communicating with the upstream and downstream stakeholders.

Currently, for Warehouse Management Systems (WMSs), many tracking mechanisms are used. The authors of [10] proposed a system of barcodes to be implemented in every process of a WMS. Barcode scanners and barcode generators were used for this purpose. The efficiency was better compared to a manual system; however, the fact cannot be ignored that barcode stickers are less reliable and require replacing after some time. For relatively large warehouses, RFID tags have also been implemented as they are more reliable and last longer [11].

The integration of the IoT in recent years has been a game changer in terms of efficient warehouse management. Many new sensors and devices have been designed and deployed in various types of industries to facilitate the transition to the Industrial IoT [12,13]. The IIoT is still an emerging concept that needs to be developed. However, much research and many surveys are being conducted to comprehend the scope of this area and overcome the challenges it is facing [14–17]. To design efficient IoT-based architectures, the correct blending of the software and hardware modules of the system is necessary. IoT-based smart warehouses enable the transition to Industry 4.0 by catering to the challenges related to data collection, human and robot activity, and robust localization [18].

The architecture of a smart warehouse must be able to maintain a smooth workflow [19]. Several algorithms have been designed for smart management systems based on the IoT that are used for tasks such as product placement and product retrieving by visualizing and relating data in terms of space, position, and volumes [20]. Task optimization plays a vital role in a smart warehouse system [21]. From picking goods to arranging them on the shelves and pallets, all the tasks have to be optimized in a way that constructively improves the flow of the supply chain. Techniques such as adaptive task planning and path planning algorithms have been put into practice in order to manage large autonomous systems with limited resources [22].

Although there have been previous works in the domain of warehouse automation and inventory management systems [11,15,19], nonetheless, there is always room for improvement. The past research has mostly addressed partial automation on one or more sections of an inventory. There is also little to no research on the implementation of a complete architecture of a warehouse automation system, conforming to the modern standards of the Industrial IoT and providing practicality, feasibility, and efficiency. We address these issues and research gaps in this important domain, and the contributions of this work are mentioned below:

- We propose an architecture to automate the Warehouse Management Systems in smart warehouses for Industry 4.0 technology adoption. A domain-driven approach

was used to propose the architecture, which models the system using architecture viewpoints to cater to the needs of various stakeholders;

- The real-time implementation of the proposed model was achieved by designing prototypes according to the models for various sections and nodes in the architecture;
- A thorough description of the IoT-based system prototype design is presented, which can be customized according to the needs of any warehouse and inventory management system;
- A case study of a textile factory is discussed in which the proposed model was deployed for validation testing;
- Lastly, the customized hardware design and IoT-based implementation according to our proposed architecture were found to improve the latency and overall efficiency of the system.

The rest of the paper is organized as follows: Section 2 presents the background of this work, while Section 3 describes a review of the literature. Section 4 presents our methodology and the proposed architecture. In Section 5, we present a case study based on the proposed architecture. Further, the prototype design and deployment are presented in Section 6. The staggered transition is discussed in Section 7. Results and a comparison are given in Section 8, and finally, the conclusions are presented in Section 9.

2. Background

In this section, we present the background of the problem by giving an analysis of the market and domain under consideration.

2.1. Market Analysis

Warehouses and inventories are an integral part of the supply chain. All the trades in a supply chain are managed using a warehouse, which could be public, private, bonded, or governmental. All over the world, various multinational companies use different sorts of mechanisms in order to manage their supply chain and trades, either manually or automated. FedEx is one of the top 10 logistic companies in the world, having 127 warehouses all over North America, with an area of 35.4 million square feet. Similarly, DHL and UPS also have a presence all over the world, comprising large warehouses and a huge number of products to manage. Multinational E-commerce traders such as Amazon have warehouses all over the world, Amazon having revenue of up to USD 70 billion in the last quarter of 2020 [23]. The gigantic size of these companies calls for the need for updated and automated warehouse management systems.

2.2. Domain Analysis

A warehouse is basically an inventory that can be divided into different types like incoming goods, raw material store, and finished goods. Typically, a warehouse consists of racks, bins, and pallets to store goods. It comprises numerous sections, which are further sub-divided on the basis of the types of products, sales, and stakeholders. A brief description of various sections is presented below, and the overall structure is given in Figure 1.

2.2.1. Incoming Dock

The inflow of the warehouse starts at the incoming dock, and the inventory management process is usually initiated at this stage. The goods arriving here are entered into the system, either by manual check-ins or, in the case of a smart warehouse, using barcodes, RFID tags, or some other unique IDs [24]. Unique IDs for all the items, goods, or products make it easier to track them at later stages for production, packing, or dispatch. Barcodes are effective for short distances, but for longer distances, RFIDs are used, that efficiently scan the product even from a distance, hence saving time and hassle; the tags are long lasting as well, making them cost-efficient. From the incoming dock, the goods or products are sent to the storage area.

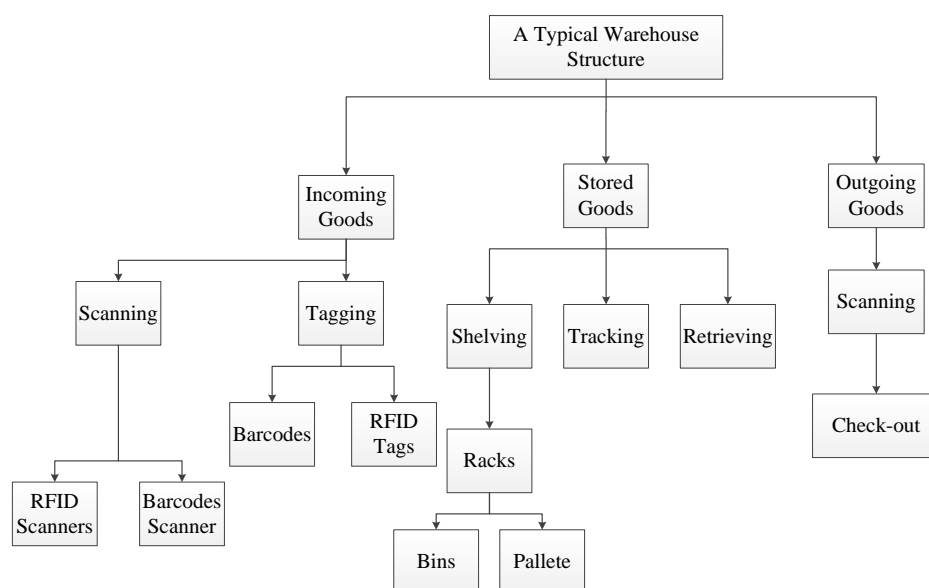


Figure 1. A generic display of the departments in a warehouse.

2.2.2. Storage and Retrieval

In a typical warehouse, the storage of the goods merely refers to placing the cartons on shelves or racks. There is no particular pattern or order designed for that purpose, which can not only decrease the efficiency but also make it difficult to track the cartons if they are not tagged. On the contrary, in a smart warehouse, several algorithms are designed to store the goods. From receiving the goods to putting them on the shelves, every action is performed based on a pre-defined set of instructions, to maintain the flow of the system [25]. The goods are transferred to shelves or pallets either manually or, in the case of a smart warehouse, using Automated Guided Vehicles (AGVs), conveyer belts, or line-follower robots [26]. Whenever any product needs to be exported, it is retrieved using the same functionality.

2.2.3. Tracking and Communication

The overall integration of departments and sections of a warehouse is necessary for efficient communication among departments. Traditionally, communication is maintained using manual pen and paper methods such as generating sales orders, demand orders, etc., and sending them to other departments through electronic mail or manually. As for smart warehouses, the IoT uses communication protocols to link every section and item to each other [27]. This ensures smooth communication within the warehouse and that the information is transmitted and received in time.

2.2.4. Outgoing Dock

The goods that need to be sent out of the warehouse are transported here; all the sales order details are cross-checked, and the product or bundles of products are checked out of the inventory, while the information is updated in the inventory records. The check-outs can be performed manually or with some partial automation; however, the use of scanners and RFID tags integrated with the software system can make the flow smoother and improve efficiency.

3. Literature Review

The idea of using a smart architecture with reference to the IoT and decentralization has been widely discussed in the literature. However, the design of a reference architecture for warehouse automation is a relatively emerging field. A summary of various works performed in this context is given below. The management of a warehouse may consist of

managing the goods or products coming into the inventory, keeping track of the location of the items, and handling the check-outs of the finished goods. Data collection or data entry is one of the important aspects of warehouse management. Data can be of any form such as IDs, prices, or time stamps, and the data collection units can be barcode scanners, RFID readers, sensors etc. An IoT-based WMS for industries operating in the development of customized products was proposed by [28]. In this WMS, all the parts are labeled with RFID tags through which the assembling of products is handled. RFID information from these parts is sent to the Electronic Product Code (EPC) information server from whence the inventory is monitored and modified and change requests and picking faults handled. However, in this proposed work, order picking, routing, and storage plans are not discussed. Mostafa et al. [29] described a system for inventory. When a product passes through the (in/out) gateway, its tags are read by scanners, providing information corresponding to the tag ID such as the location of the targeted products, category, product name, etc., to the forklift's screen. The driver obtains the information about the location of a particular product and picks it up, then the inventory system is updated in real time. Laxmi et al. [30] examined the use of GPS positioning for transport fleets along with RFID shipment tracking at the entrance and departure points of the warehouse. The focus of their approach was on supply chain management through the tracking and identification module. The position tracking module uses WiFi, as it requires a high data rate and more transmission power, while the information from the position tracking module is received through a base transceiver station. RFID has been very popular for the segmentation, tracking, and positioning of products [31,32]. The RFID reader's information is sent wirelessly to open-source hardware and the data are stored in a central server. All modules are connected over the same network for effective data communication. All the data from the warehouse are stored within a Raspberry Pi and displayed on the frontend. Similarly, a system was designed by [33] that included various environmental sensors along with RFIDs attached to the boxes. They demonstrated the use of the MQTT protocol for remote monitoring and maintenance of the storage conditions of short-lived stock items. Their proposed system also explored the efficiency of MQTT and COAP, and it was concluded that MQTT was superior to COAP in terms of data integrity, ease of implementation, and security. Laxmi et al. [11] also proposed a heterogeneous network architecture for shipment tracking using RFIDs and a Python-stationed gateway. The gateway provides subscriptions to topics through an MQTT broker, whenever a client requested them. The subscription request sent by a client is entertained and fetched by WLAN, which works efficiently due to its high transmission rate.

The use of vehicles for transportation within the warehouse is very common. Traditionally, manually driven vehicles are used for transportation; however, as the architecture transitions towards the IoT, unmanned vehicles, robots, and conveyer belts are also being introduced. A system was developed by [34] for inventory regulation by using Unmanned Aerial Vehicles (UAVs). These UAVs help in the counting and localization of inventory. UAVs have the capacity to avoid obstacles and collisions. This reduces the need for critically planning and maintaining inventory tracks in the warehouse. To use wireless connections for these UAVs, Ultra Wide Band (UWB) solutions using anchor nodes have been developed, whose range can be scaled using the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) depending on the requirements. They also designed a multi-technology, duty-cycling, time-slotted UWB MAC protocol for the optimization of power consumption by UWB radios. Nagendra et al. [35] proposed an IoT-based architecture for order picking that involved controlling and monitoring the inventory. The architecture included process specification and domain model specification using the product ID, location, rack, etc. Information model specification introduces the information about the position of the robots, the availability of the products, the position of the products, the size of the rack, etc. Service specification provides the service of operating auto and manual modes for orders received, the identification of the location of a rack, the movement of robots towards the specified rack, and so on. The authors also described an IoT-level specification including

measurement, automation, innovation, and cloud utilities, making the solution a Level-4 IoT system. Furthermore, the architecture involved device and component integration, where they mapped the functionalities to the sensors and actuators.

In any inventory management system, the location of goods is of great importance as it makes tracking easier and more efficient. A self-adaptive decision model for the inventory positioning, check-in, and check-out of the inventory and other event handling was proposed by Zhang et al. [36]. The model senses the environment, creates knowledge based on it, and trains a neural network to recognize the scenario to make a decision. The model improves itself by adjusting the knowledge base and decision-making is performed at the node level, which makes this system completely decentralized. Similarly, Liu et al. [37] developed a strategy for logistics management for which they devised a framework consisting of three parts: IoT-enabled vehicle terminals, resource management, and dynamic optimization services. Information is obtained by identification sensors, e.g., RFIDs, wireless connectivity, etc., and is updated to the Enterprise Information integrated System (EIS) and Geographic Information System (GIS), which further provides optimal routes; this also involves tracking of the inventory while loading, which is specifically based on RFIDs. This creates the first layer of the architecture, i.e., the IoT-based physical sensing layer. The second layer includes resource management in which the real-time data of vehicles is uploaded to the GIS, and the last layer involves providing optimal paths for the logistics. For parts' handling in warehouses, Schwarz et al. [38] developed a robotic arm with six Degrees of Freedom (DOF) by implementing the concepts of Industry 4.0. Object detection and semantic segmentation were used to classify the objects that were requested to complete an order. The system also involves 6D pose estimation using six degrees of freedom so that the items are picked up cautiously. This system was presented at the Amazon Picking Challenge and was highly acknowledged.

The automation of a part of a warehouse was performed in [39] by keeping the record of solid and liquid stock. The system uses an ultrasonic transducer to measure inventory, and the results are propagated to the Internet through a Raspberry Pi gateway device. However, it only focuses on the quantity of goods in the inventory, and it provides automatic order placement and lacks the check-in and check-out of the items based on their Unique Identifier (UID). The flow of goods can be drastically improved by Artificial Intelligence (AI) techniques [40], as AI can possibly help in decision-making for transportation within the warehouse, the loading and unloading of inventory, and the palletized goods operations. In these papers, the IoT architectures used were categorized into two types. One was where sensors and actuators directly communicate with the server over the Internet. This type is used when there are few sensors and/or the network is slow. The other one included a gateway (such as a Raspberry Pi), which further passes on the information to the server using some IoT connectivity.

Van Geest et al. [41] proposed the design of a reference architecture for a warehouse, based on architecture viewpoints. The architecture covers different sections of a warehouse and provides a detailed analysis of the subject. However, the architecture is more on the theoretical side, and the idea presented was very general, which makes it less viable for practical implementation.

As mentioned earlier, the architecture-based analysis of smart warehouses is an emerging category; hence, we addressed the shortcoming of the work by Van Geest et al. [41] and extended their work by adding more architecture viewpoints, practical testing, as well as the hardware implementation of the architecture.

4. Methodology and Proposed Architecture

The addition of an automated inventory management system is the prime need of huge warehouses that deal with much stock and many goods on a daily basis. In order to design a methodology, the domain was studied in detail, and the needs were identified. The domain analysis is presented in Section 2. A detailed literature review and background study were used to pinpoint the problems. The major problems identified were the lack of efficient

communication between departments, keeping track of the sales orders, transporting the cartons, and the managing and tracking of prepared goods inside the inventory. The stakeholders that were involved in this domain were also identified so that their needs could be prioritized and managed accordingly.

To cater to these problems, we propose an architecture based on architecture viewpoints taken from [42]. The viewpoints are a template to construct models for information systems based on the needs of the system and its stakeholders' concerns. The views were used to describe the functional elements of a system and their interaction with each other and with the resources outside the systems. They also account for the system requirements regarding the software and hardware support, physical deployment, and communication between different components. They make use of a ubiquitous language in order to create models that are equally understandable to software developers, as well as the business personnel. This helps in modeling the architecture according to the interests of the stakeholders to attain the maximum business profit.

The warehouse management system was modeled as an information system connected to several internal and external elements. In this paper, we used the context diagram, functional, and operational viewpoints to propose an architecture. These were used to identify the problems in the modeling of the warehouse and then to solve the problems by proposing the models. Furthermore, we also designed and developed an IoT-based hardware prototype to implement the architecture in a textile factory's warehouse.

4.1. Context Diagram

The context viewpoint provides the overall description of the system, as well as the various sections working under it. It elaborates the relationships between people in the system and the environment in which the system is set, also showcasing the dependencies and interactions. The overall brief description of the proposed architecture is given in Figure 2. The context diagram has a clustered form, with the WMS in the center. One of the main problems that need to be catered to is how each and every department is related to each other. The context diagram handles this by showing the relationships, as well as the dependencies of various departments, the people, and the environment within the whole system with the help of directed arrows. The WMS controls and administrates the whole system in coordination with the managing department.

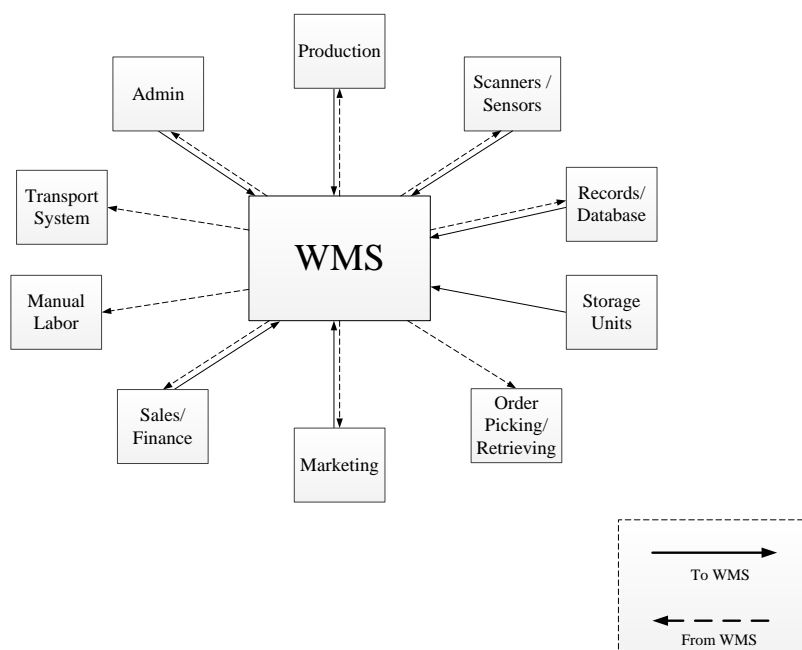


Figure 2. Context diagram.

4.2. Functional View

The functional viewpoint is a display of the major functional elements of the system, the responsibilities they hold, and the interactions they perform with other elements. In order to have the smooth functionality of the information system, the responsibilities of the elements are made clear. In an information system, various departments work in combination with some departments, as well as in parallel with some others. Every department has tasks assigned to it, which are performed in the interaction, based on some interfaces, with the other departments. The functional view is the most basic and important model for any information system. It consists of the functional elements, interfaces, connectors, and external and internal entities of a system. In addition to explaining the functions of the elements, it also defines the scope of each entity. The functional viewpoint is shown in Figure 3. The interfaces defined are the hardware, as well as the software, where each large block represents a functional element and the arrows indicate the interfacing with other functional elements.

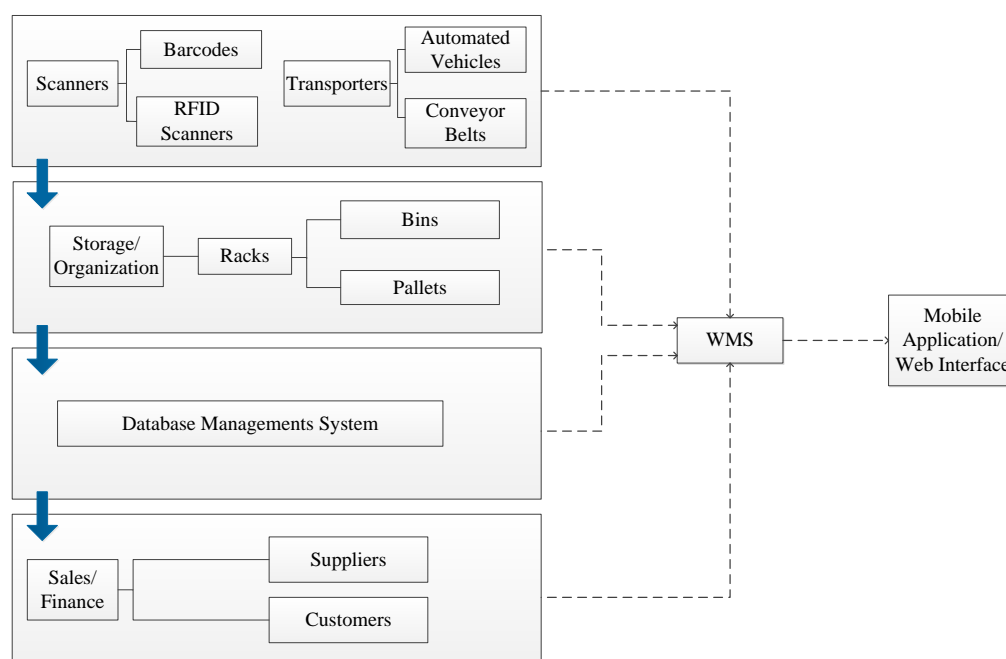


Figure 3. Functional view.

4.3. Operational View

The operational viewpoint is a road map of how smoothly the system will work once the architecture is deployed. It is mainly designed to address the problems regarding the effective running of the system. The main concerns during the deployment of any architecture include the installation and upgrading of the system entities (hardware and software), the migration of the data from the old databases to the new ones, the monitoring and control, and the configuration of the operations. The operational viewpoint in Figure 4 depicts the workflow of the system at every node after the deployment of the architecture, which is of great help while practically implementing the architecture. It also deals with the software-related issues such as the security of the system during operation and data transmission. The system is subdivided into divisions called “nodes”. Each node refers to a department in the warehouse. The arrows show the workflow. Any hardware such as sensors, etc., is suggested to be installed at the nodes as these are the entry points to every division. In this way, the minimum amount of hardware can help monitor a large premises.

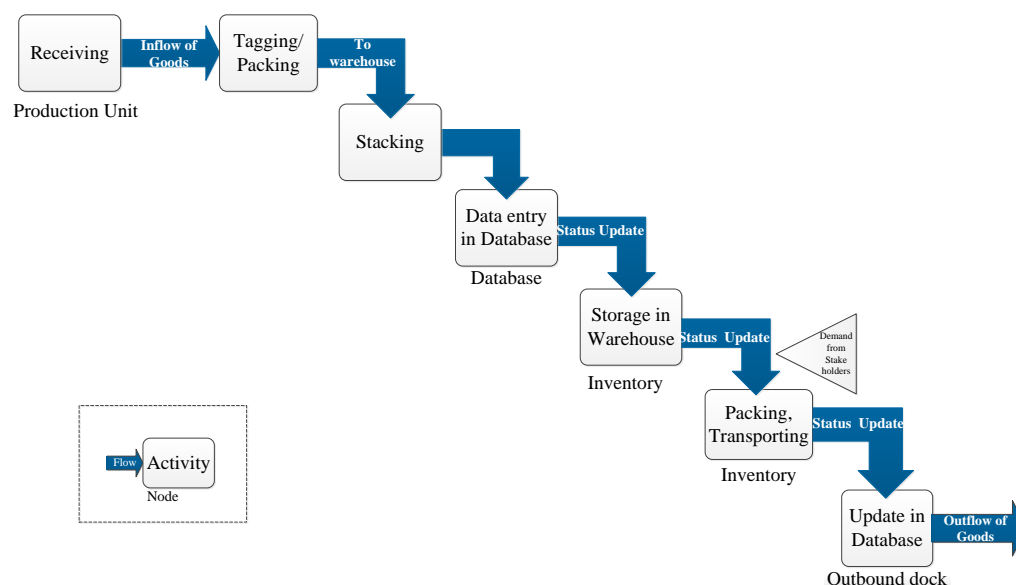


Figure 4. Operational view.

5. Case Study

In order to implement the proposed architecture, we present a case study based on a textile factory. The textile industry is presumably one of the biggest industries in the world, worth USD 1.5 trillion in 2020 [43]. In addition to the existing companies, many new ones are emerging, making this industry a potential market for growth and revenue. With the increase in the size of the market, the supply chain is becoming more and more complex, and there is a dire need for smart warehouses in this domain. Some of the bigger names in this domain are making a transition towards smart WMSs, but many companies are still working with the traditional pen and paper approach to keep track of their services. For the target case study, first of all, the data for the textile factory were collected using questionnaires, interviews, and tours of the premises of the factory, as shown in Table 1.

Table 1. Data collection during the case study.

S. No.	Questions
1	What is the location of the inventory?
2	What are the incoming and outgoing goods?
3	How are the forecasting demands managed?
4	How efficiently is the space being used?
5	What happens if the system breaks down?

The textile factory under consideration has four warehouses, as shown in Figure 5, and the proposed architecture model is presented for the Finished Fabric Warehouse (FFW).

In the FFW, finished and sewn articles are stocked in the form of cartons for which the rolls of cloth are received from the rolling department. The warehouse consists of racks for storage purposes with three rows in each column, with a transportation system consisting of four-wheelers operated manually. Presently, there is no central database to maintain the record and track the orders. The Sales Order (S/O) is generated from the Sales Department, containing the sales order number and order information, which is printed and passed on to the Production Department. The Production Department generates Demand Orders (D/Os) against the specific S/Os and prepares the order, and the prepared order's items are packed into cartons. Each carton is weighed, and the information related to the articles inside is printed and pasted on the cartons. Sometimes, barcodes stickers are added during the rolling process on the uncut cloth, but there is no efficient use of them as the data are not maintained in the database. The cartons under a specific S/O are moved to the inventory

and then sent off to the check-out points. At the check-out, again, the carton's information is tallied with the S/O manually, and the cartons are sent out for delivery. A paper is sent back to the Sales Department that indicates that the order has been dispatched, and upon receiving it, the Sales Department closes that particular S/O and marks it complete. All the records are maintained manually. The process of receiving the paper from the check-out point back to the sales department takes 24 h, and there is no tracking of the orders during the whole process. The current workflow in the factory is shown in Figure 6.

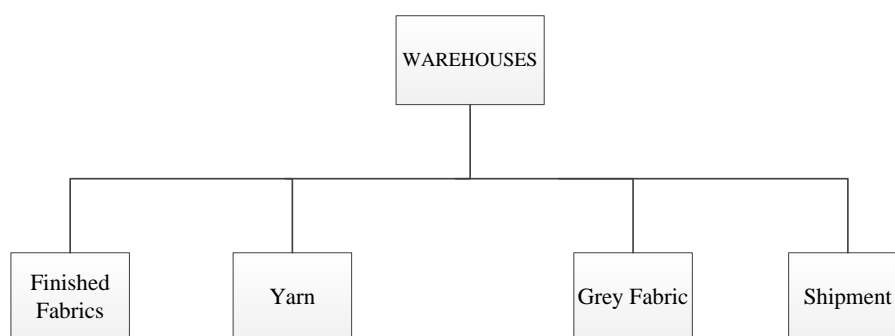


Figure 5. Division of the warehouse in the textile factory used for the case study.

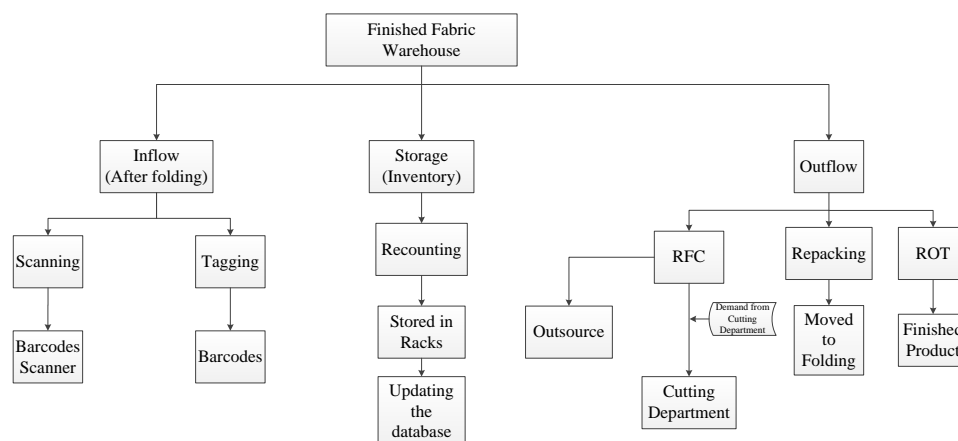


Figure 6. Existing workflow of the warehouse used for the case study.

5.1. Implementation of the Architecture Models for the Case Study

We modeled the existing architecture of the warehouse onto our proposed architecture, and the models were designed in reference to the proposed viewpoints, which are discussed below.

5.1.1. Context Diagram for the Case Study

The context diagram in Figure 7 shows the relationship of the warehouse management system with all the departments and sections in the factory. It presents the overall context of the system with the entities that are running the system. The Finished Fabric Warehouse (FFW) indicates that the WMS is being implemented for the FFW in the textile factory. The clustered form of the diagram shows how different departments work in relation to the management system. The Outsourcing Department was specifically incorporated for this particular warehouse as it was required by the stakeholders. The directional arrows show the dependencies of the departments on each other. They also show the flow commands in the whole system. The solid arrows show the commands and messages that are directed to the WMS to pass on, transmit to other departments, or perform some functionality according to it. The dashed arrows show the commands and messages that are transmitted

by the WMS to the rest of the departments. The model provides a clear and concise understanding of the information system.

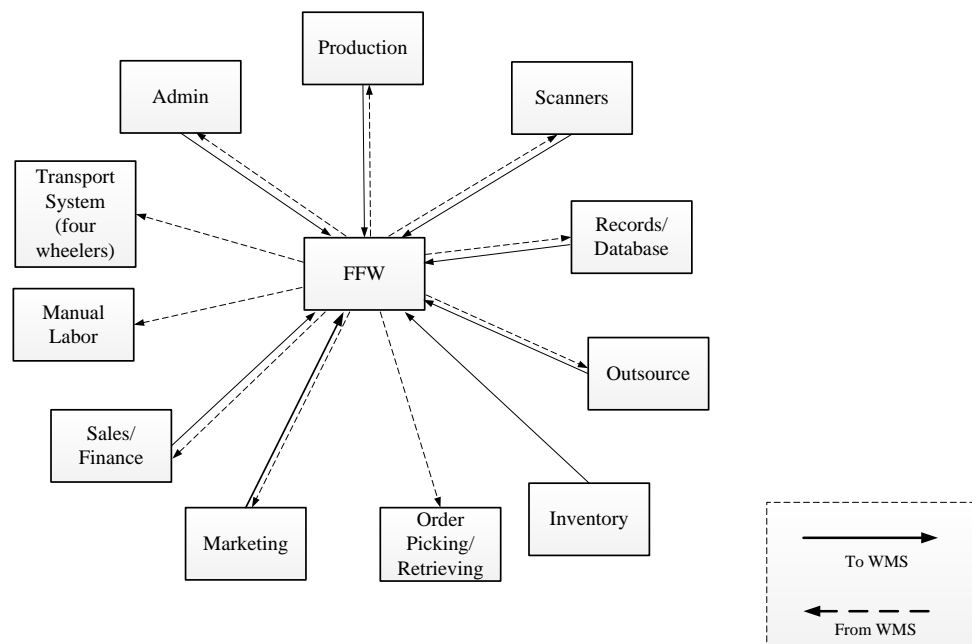


Figure 7. Context diagram (for the case study).

5.1.2. Functional View for the Case Study

As shown in Figure 8, the large blocks show the functional elements of the warehouse management system including scanning, storing, managing, and communication. The smaller blocks inside the functional elements define the entities used for those functional elements; for example, the transportation is carried out using the four-wheeler and conveyor belts. The interfaces are shown using arrows. The bold arrows show the connection using Ethernet plus WiFi based on the element, whereas the dashed arrows show the wireless connection. The wireless communication is based on the publisher–broker–subscriber model, as is explained in further subsections.

5.1.3. Operational View for the Case Study

For the case study, the operational view is the same as shown in Figure 4. Each node is equipped with scanners and connected to the other nodes by the communication model. The operations of the inventory start after receiving finished products from the production unit, which are then packed in the cartons and tagged with RFIDs, which are stored in the database under the S/O number. The cartons are kept in the warehouse until they are delivered and then moved to the transporting nodes, which are also equipped with the scanners. The records in the database are updated in real time, which makes it easier to track any carton using the S/O or RFID.

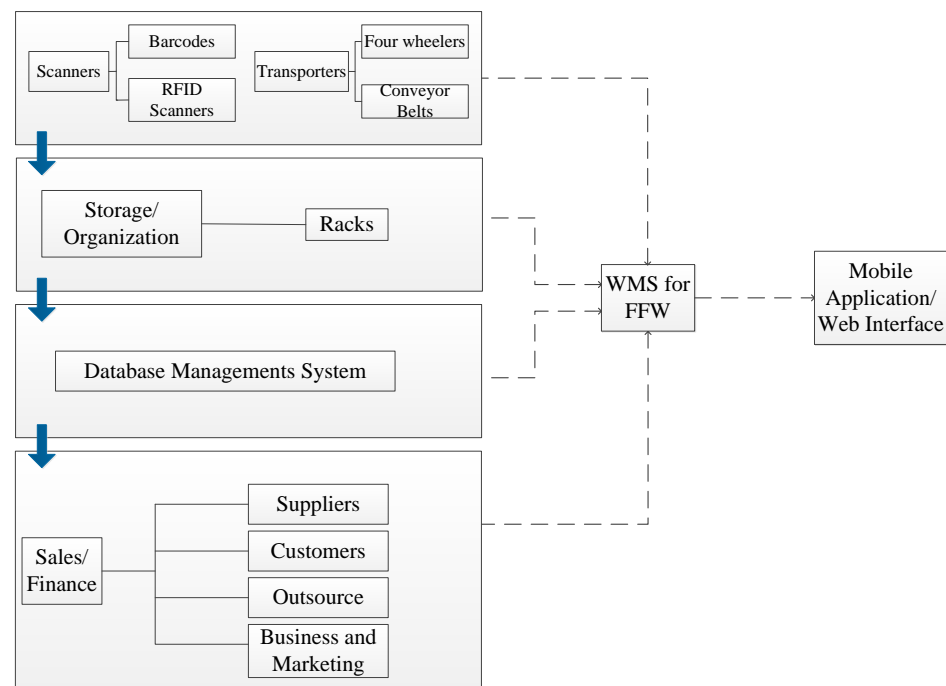


Figure 8. Functional view (for the case study).

6. Design and Development of the Prototype

In this section, we provide the details of the prototype, designed for the implementation of the architecture in the warehouse to complete the transition towards the smart warehouse management system integrated with the IoT. We divided the prototype development into four generalized modules. The modules are software, as well as hardware based and can be implemented using the details we give below:

- Sender module;
- Receiver module;
- Protocol stack;
- Database.

6.1. Sender Module

This is a generalized module for any type of information being provided to the WMS. It can scan the data from the barcode scanners/RFID scanners from the inventory, incoming dock, and outgoing dock. It can also include any data, such as the information of the employees entered into the system, the temperature and humidity of the Manufacturing Department, the financial records of the raw materials being bought, as well as the sales records of the products being outsourced. For every type of information being entered into the system, appropriate sensors or input devices can be installed into system and integrated with the network. The goods arriving at the inflow can be tagged with an RFID tag or with a barcode sticker, depending on their structure, degree of sustainability, and some other feasibility factors. RFID tags are deemed suitable for products that are non-perishable, as they can stay intact for a long time and can be used for tracking purposes. Various methods have been discussed in the literature regarding tagging goods and installing scanners for tracking [44,45].

For this case study, we designed hand-held and portable scanners using RFID reader modules and barcode readers, integrated with ESP controllers. ESP 32 is a high-speed microcontroller used for its very low latency, large number of GPIOs, and the availability of Bluetooth Low Energy (BLE), WiFi, and Ethernet connectivity, all at an affordable price. A number of sender modules were created for testing and validation and implemented all over the factory on the designated nodes. The scanners in close proximity to each other

were connected to the same broker to avoid using extra brokers. The RFID readers were connected with the controller using the Serial Peripheral Interface (SPI), and barcode readers were interfaced over BLE. The scanners are portable and work as standalone devices. They communicate with the central system over WiFi using the publisher–broker–subscriber model of MQTT, with the added option of Ethernet.

As mentioned earlier, each functional unit in the factory is called a “node”. The raw material (uncut cloth rolls) for production is identified through barcode stickers, and the finished goods are packed in a carton tagged with RFID tags. At each node, the goods are scanned and cross-matched with the S/O, which reduces the chances of errors and increases the system efficiency.

6.2. Receiver Module

This is a module designed to receive information from the WMS. The data being entered and updated in the communication network of the WMS are displayed here [46]. This can be a platform for user accessibility and interaction. Various platforms can be used for this purpose, such as a web or mobile application, a customized user interface, or a specifically designed software [47]. For the case study, the frontend of the web application was designed using the Angular framework along with a database developed using a SQL server and .NET backend. The web application can be used on all the devices that require access to the data. The scanned data reach the subscriber via the broker, then are displayed on the frontend using JavaScript and WebSockets. The frontend provides different functionalities such as user authentication, purchase and sales records, and inventory stock. A customized frontend software was designed for every department that catered to their needs. Different departments share tables in the database to exchange information so that the data are accessible to every element in the system at all times. A centralized database was also maintained in addition to the databases of each department.

6.3. Protocol Stack

The Internet Protocol Suite is the backbone of any IoT-based system. It is the bridge that connects all the modules together and exchanges data in between them. For any particular device or system, the protocol is chosen as per the requirements and needs, so that it can work efficiently in the given circumstances. Different protocols are available including wireless and wired [48]. Wired protocols such as SPI, I2C, and UART are useful in the absence of a wireless network or the Internet; nonetheless, these restrict the area of application as they have a limited wire length. A longer wire length is susceptible to transmission loss, as well as low transmission rates. Hence, wireless transmission protocols are vital to a fully automated smart warehouse management system [49]. As for the wireless networking protocols, there are different protocols working on different layers of the Internet Protocol Suite. These include HTTP, MQTT, SMTP, DHCP, etc., for the application layer, TCP, UDP, DCCP, etc., for the transport layer, and MAC (WiFi), ZigBee, OSPF etc., for the link layer. Conventionally, the client-server model using HTTP/s is used as the application layer protocol as it is easily configurable on many devices and gives good results. However, we replaced it with MQTT, which has provided promising results in real-time environments, as proven by the comparison at the end.

MQTT is a lightweight wireless networking protocol, with a small overhead. The greatest advantage it provides is the fast, real-time data transfer, which is vital in the case of a large warehouse and inventory. A typical architecture for the implementation of the MQTT protocol is shown in Figure 9. It is based on a broker, subscriber, and publisher model.

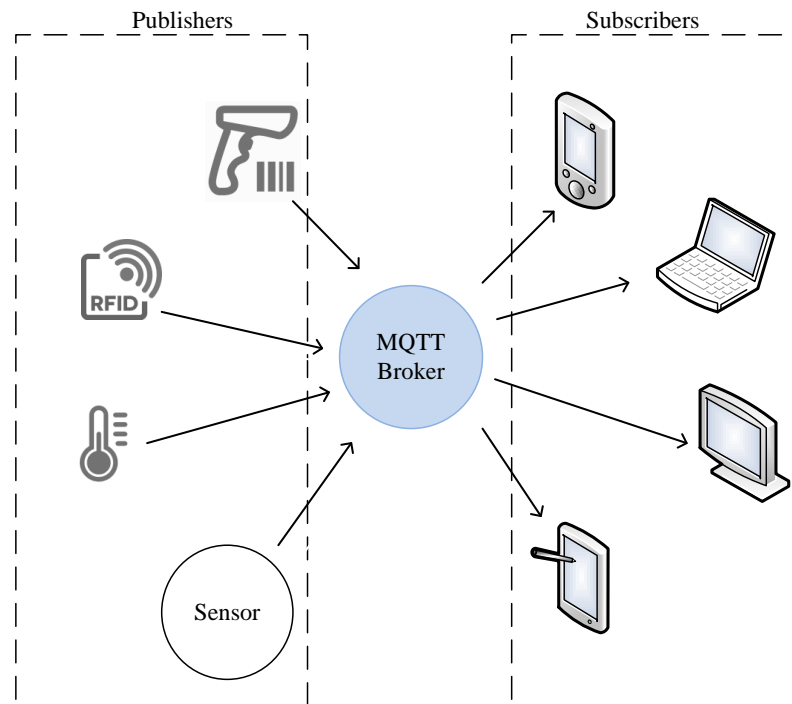


Figure 9. MQTT architecture for the Internet of Things.

To configure the broker, an open-source broker can be installed on any operating system. A large number of subscribers and publishers can connect to it using IP addresses. Publishers are the sender module, and subscribers are the receiver module. Publishers publish data (strings, integers, bits, etc.) to any “topic”, that acts as an address to publish and retrieve the data. Subscribers subscribe to the topic from which they want to receive data. There can be many topics for a single broker, with a large number of subscribers and publishers. Subscribers can also be publishers and vice versa. The small overhead size makes it possible for data to be transmitted in real time, making MQTT the perfect option for real-time data transfer and acquisition, which is vital for any real-time smart warehouse management system [50]. As mentioned earlier, any device with an operating system can be configured as a broker; in this case, the Raspberry Pi was used due to its portability and availability for an operating system, and the Eclipse Mosquitto broker was configured on it. The connection was secured using publisher authentication. The ESP scanners were connected to the Raspberry Pi wirelessly and published the IDs on the specific configured topics. A single Raspberry Pi with the broker installed on it can connect to many publishers and subscribers, hence covering a large area of the inventory. The IDs were retrieved at the backend using JavaScript, which acted as a subscriber. The block diagram of the prototype design is shown in Figure 10, and the overall working system is shown in Figure 11.

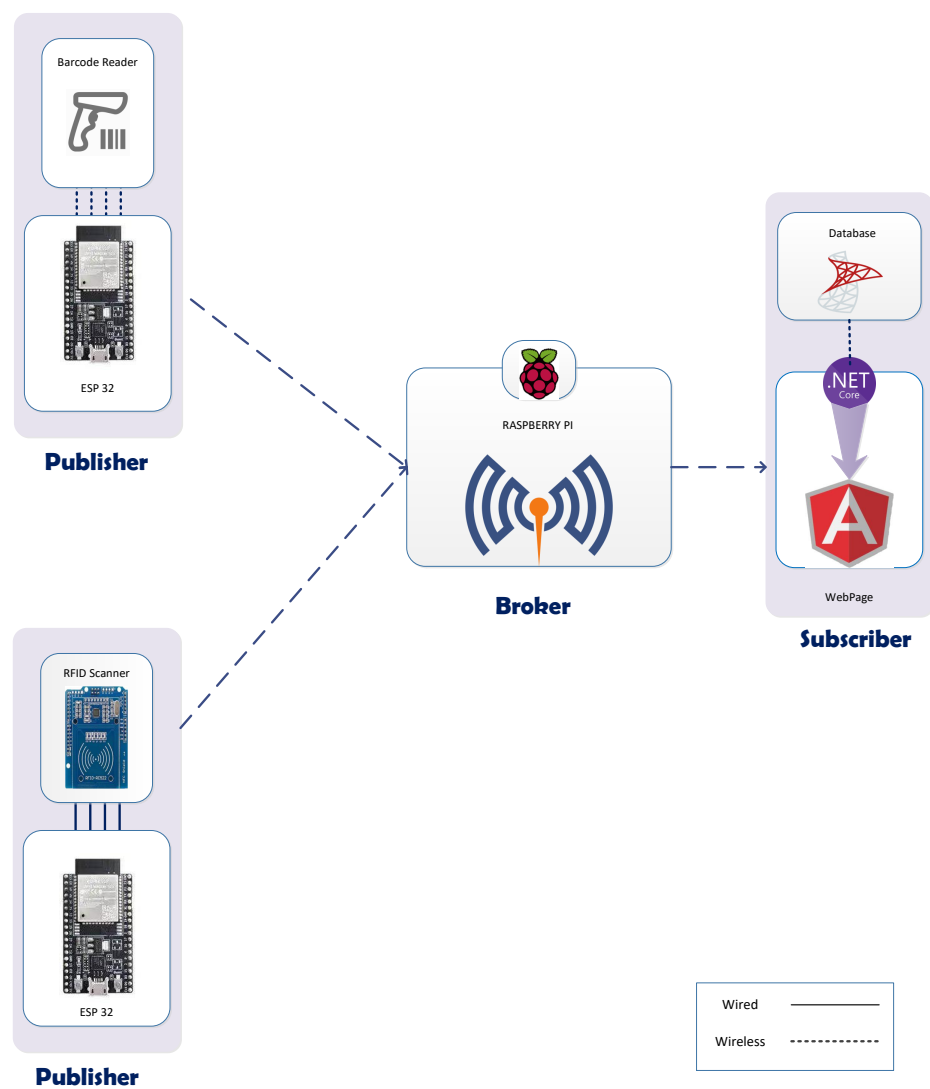


Figure 10. Block diagram of the prototype.

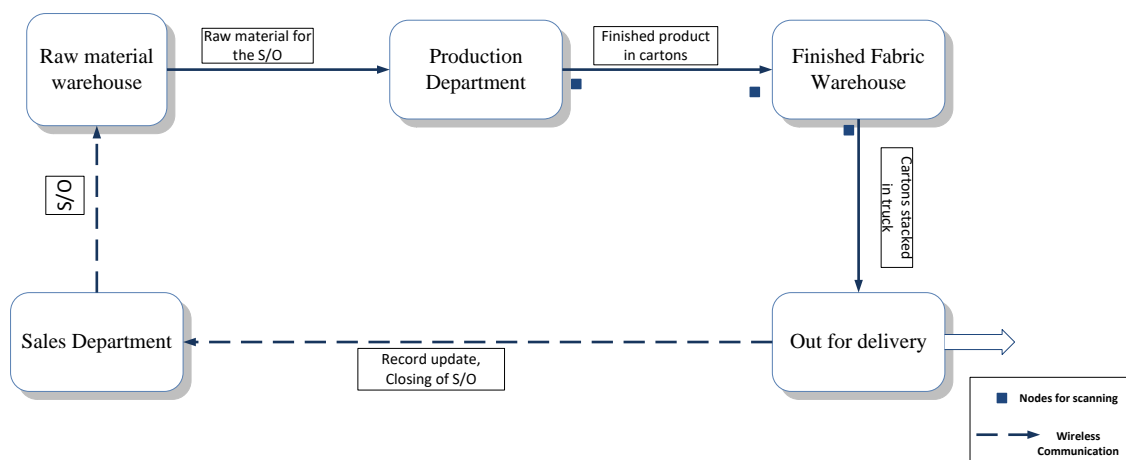


Figure 11. Block diagram of the working system.

7. Staggered Transition towards a Smart WMS

The above sections depicted the mapping of the proposed architecture onto the textile factory used for the case study. The hardware of the system was designed and then tested in the warehouse of the factory. For the sake of a practical implementation and successful start and running of the systems, a few areas needed to be addressed, which are discussed below in reference to the case study.

7.1. Modular Approach

As the discussed warehouse and any general warehouse already have existing management systems and databases, which work manually or on some automatic protocols, the transition towards smart warehouses is always staggered. A modular approach was used in this regard, and every department was tackled one by one. In the first phase, the data regarding the sales orders were accessed from the existing database of the factory, which was linked to the new database. The old database kept working in parallel. In the next phase, the information of every new incoming product was saved in the new database. After a course of a few months, the old database was completely discarded with the successful operation of the new system.

7.2. Mobility

The deployed system needed to be mobile and portable in order for it to be practical. To ensure that, the broker was installed on the Raspberry Pi, which were deployed in the warehouse. All the systems were on the same network and could be moved anywhere inside the premises.

7.3. Scalability

Scalability refers to the system's ability to be mapped onto an industrial infrastructure with a huge volume and size while still being able to work efficiently. To manage this, we used the MQTT protocol instead of the conventionally used HTTP/s, which was explained in detail in Sections 6 and 6.3. For the case study, the data that were transmitted in the system consisted of various sensor readings and, most importantly, the IDs on the goods. The articles were packed in the cartons, and each carton was tagged with a unique RFID, which was linked to the sales order of that particular order. At a certain time, the several scanning stations were working at the same time and transmitting the data. MQTT being light weight helped with the transmission of the data, as it has a very small payload. In Table 2, the data show the latency in the transmission of the payload bytes. To deal with the load of the system, more than one broker was added at a few places to enhance the system horizontally. This divided the load where there was network congestion and kept the transmission running at a smooth rate.

Table 2. MQTT vs. HTTP/s—latency.

Protocol	No. of Packets	Bytes Transmitted	Avg. Time for Complete Connection Cycle (ms)	Avg. Time for One Message (ms)
MQTT	1000	173,980	30,483	30.4
HTTP/S	1000	198,341	42,503	42.5

7.4. Concurrency

To make the system concurrent, so that it could be accessed by various managers at the same time, MQTT was again utilized. All the computers, as well as any devices that were used for the frontend display, etc., were made subscribers to the MQTT brokers. Different topics were setup that relayed different data from publishers to subscribers. The computers in every division of the factory were linked to the system using the same intra-network.

The data could be accessed by the authorized personnel using their login credentials, which made the whole system concurrent and accessible.

8. Results and Discussion

In this section, we present the results of overall proposal, as well as a comparison with the previous solutions. In order to prove the improved results, some key performance indicators are also presented, which show the efficiency of the proposed method quantitatively.

8.1. Key Performance Indicators

Below is a comparison of the proposed architecture with the previous one with respect to some performance indicators.

8.1.1. System Efficiency

The efficiency was measured in terms of time, before and after the implementation of the architecture and prototype. As mentioned in Section 4, before the implementation, the S/Os were manually sent to the next department. After production, the storing of each carton in the inventory required verifying the order contents. Similarly, the S/O completion was performed after one shift; till then, the order was out for delivery, and there was no way to track or make up for any mistakes made in the order. However, after the installation of the scanners at each node according to the model in Figure 4, there was a decrease of 79% in the time taken in storing the goods in the inventory, and the time to update the records after the completion of a S/O was reduced to a few minutes (Figure 12). Overall, while it increased the efficiency, it also helped reduce the complaints of stakeholders to almost zero, as the orders were checked and verified at every node and updated instantly.

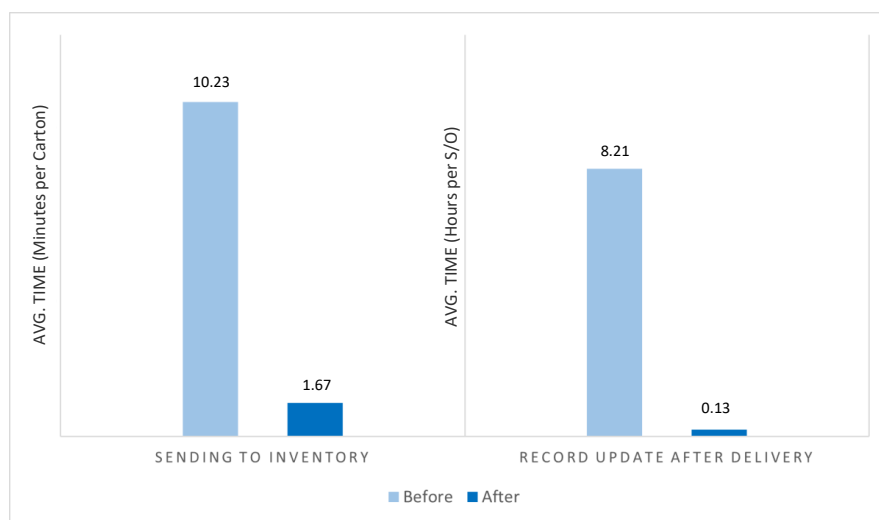


Figure 12. Comparison of the system efficiency before and after the implementation of the prototype.

8.1.2. Latency

Latency is the end-to-end delay in data transfer. In large and complex networks with many nodes, latency can increase by many folds. Low data transfer rates hinder all the communications and choke the network. MQTT is a protocol with a very small overhead, which is ideal for use in circumstances where the payload is a small string of numbers, as required in the proposed case. A comparison of the latency of MQTT and HTTP is given in Table 2. It shows the data collected by sending 1000 messages via HTTP and MQTT from distant telemetry devices, where the packets were collected by sniffing tools. The system was concurrently sending and receiving packets from multiple publishers, and the data show the packets sent by one of the publishers. The test was repeated multiple times

during different times of the day with variable network traffic to have average values. The results showed significantly faster data transmission rates from MQTT with low latency. This result was more prominent in the case of multiple nodes, as MQTT reuses the same connection, whereas HTTP/s requires a three-way handshake before every re-connection. The results may vary by changing the size of the payload, which was the same in this particular case as the data consisted of the ID of the cartons.

8.1.3. Power Dissipation

As the number of nodes increases in a system, the power intake increases as well. With the use of MQTT, we made sure that the power dissipation was kept to a minimum as compared to HTTP. MQTT uses power only during transmission, whereas HTTP remains awake after the initial handshake. A comparison of the power dissipation of MQTT and HTTP in Figure 13 shows that at the beginning, MQTT used up some power to connect and set up; however, once connected, it had a very minimum power dissipation for every subsequent message, as compared to HTTP. On the contrary, HTTP first sets up the three-way handshake, then keeps using power continuously, regardless of the message transmissions. This, overall, makes MQTT an efficient option for long-term, lightweight data transmissions with devices that are portable and battery powered.

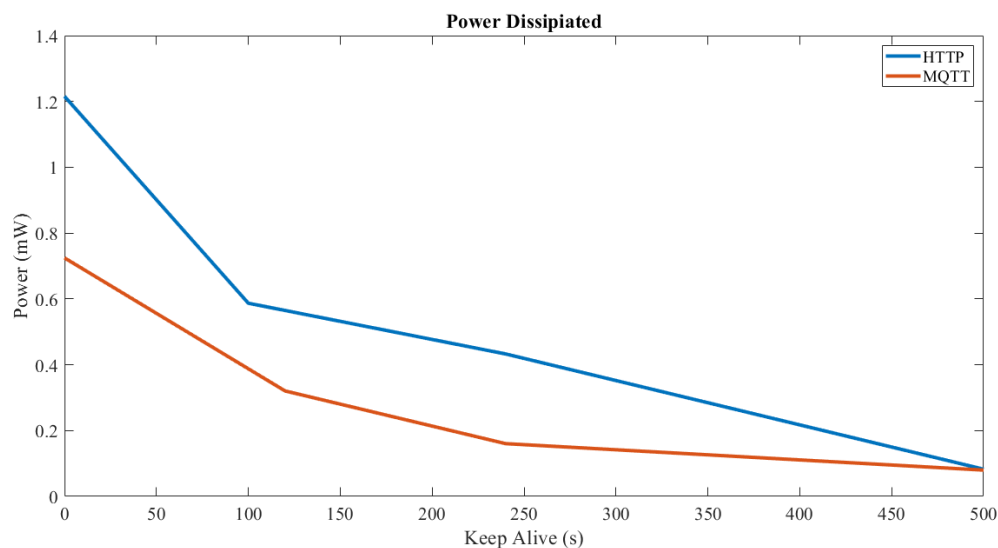


Figure 13. MQTT vs. HTTP—power dissipation.

8.2. Qualitative Metrics

In addition to the quantitative performance metrics mentioned above, qualitative analysis was performed on the system, which yielded the following results.

8.2.1. Resilient System

For any information system architecture to be successful, the reliability and resilience should be its key properties. Our proposed architecture and deployed prototype are reliable and resilient. In the case of a network connectivity problem, the database is maintained at a local server on the systems at each node. When the connectivity is re-established, the stored data are pulled into the main database and respective tables. In the case of a power failure, the power backup is maintained; however, if it is unavailable, the record is maintained manually and then inserted into the database.

8.2.2. Tracking and Visibility

As the proposed solution is integrated with the IoT, the stakeholders and customers can keep track of their orders and shipments at any time during the order booking till the checking out of the product from the inventory. This gives it an edge over traditional

warehouses with no IoT integrated. Various sensors transmit their data and readings to the server, which is within reach of the stakeholders at all times.

9. Conclusions

In designing a smart warehouse management system, the architecture and underlying data transmission technology play a vital role. This article paved new ways toward the less-explored domain of warehouse automation and presented a practical architecture design implementation. The proposed IoT-based WMS can manage a complex and integrated supply chain network by modeling it into simpler architectures that are equally understandable by the developers, as well as the business stakeholders. Domain analysis and viewpoints were used from the software systems architecture paradigm for this purpose. In addition, a detailed description of the prototype design and development with respect to the software and hardware requirements was performed according to the presented architecture. The prototype system integrated with the IoT was successfully deployed within a textile factory's warehouse. The successful deployment and the performance parameter results showed an overall efficiency of up to 88% compared to the previous architecture and validated the effectiveness of the proposed solution. For future research, the architecture can be extended to include external stakeholders (i.e., sellers and buyers), integrating them into the system to provide efficient tracking and visibility. The system security can be improved by adding authentication and decryption algorithms. Overall, the system is one of the stepping stones on the road towards complete automation and strives to fulfill the gaps in this domain.

Author Contributions: Conceptualization, M.G.K.; methodology, M.G.K. and N.U.H.; software, N.U.H.; validation, M.G.K., N.U.H. and U.K.u.Z.; formal analysis, M.G.K. and U.K.u.Z.; investigation, M.G.K., N.U.H. and U.K.u.Z.; resources, M.G.K.; data curation, M.G.K. and N.U.H.; writing—original draft preparation, M.G.K. and N.U.H.; writing—review and editing, U.K.u.Z.; visualization, N.U.H.; supervision, M.G.K. and U.K.u.Z.; project administration, M.G.K.; funding acquisition, M.G.K. and U.K.u.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Higher Education Commission (HEC) National Center of Robotics and Automation (NCRA) under Grant Number NCRA-RF-038.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
WMS	Warehouse Management System
FFW	Finished Fabric Warehouse
S/O	Sales Order
D/O	Demand Order
ROT	Roll In Tube
RFC	Ready For Cutting
HTTP	Hypertext Transfer Protocol
COAP	Constrained Application Protocol
MQTT	MQ Telemetry Transport
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
UART	Universal Asynchronous Receiver–Transmitter
BLE	Bluetooth Low Energy

References

- Sethi, R.; Bhushan, B.; Sharma, N.; Kumar, R.; Kaushik, I. Applicability of Industrial IoT in Diversified Sectors: Evolution, Applications and Challenges. In *Multimedia Technologies in the Internet of Things Environment*; Springer: Singapore, 2021; pp. 45–67. [CrossRef]
- Boyes, H.; Hallaq, B.; Cunningham, J.; Watson, T. The industrial Internet of things (IIoT): An analysis framework. *Comput. Ind.* **2018**, *101*, 1–12. [CrossRef]
- Mourtzis, D.; Vlachou, E. A cloud-based cyber-physical system for adaptive shop-floor scheduling and condition-based maintenance. *J. Manuf. Syst.* **2018**, *47*, 179–198. [CrossRef]
- Lasi, H.; Fettke, P.; Kemper, H.G.; Feld, T.; Hoffmann, M. Industry 4.0. *Bus. Inf. Syst. Eng.* **2014**, *6*, 239–242. [CrossRef]
- Atieh, A.M.; Kaylani, H.; Al-abdallat, Y.; Qaderi, A.; Ghoul, L.; Jaradat, L.; Hdairis, I. Performance Improvement of Inventory Management System Processes by an Automated Warehouse Management System. *Procedia CIRP* **2016**, *41*, 568–572. [CrossRef]
- ElMaraghy, H.; Monostori, L.; Schuh, G.; ElMaraghy, W. Evolution and future of manufacturing systems. *CIRP Ann.* **2021**, *70*, 635–658. [CrossRef]
- Shashidharan, M.; Al, E. Importance of an Efficient Warehouse Management System. *Turk. J. Comput. Math. Educ. TURCOMAT* **2021**, *12*, 1185–1188. [CrossRef]
- Torabizadeh, M.; Yusof, N.M.; Ma'aram, A.; Shaharoun, A.M. Identifying sustainable warehouse management system indicators and proposing new weighting method. *J. Clean. Prod.* **2020**, *248*, 119190. [CrossRef]
- Sisinni, E.; Saifullah, A.; Han, S.; Jennehag, U.; Gidlund, M. Industrial Internet of things: Challenges, opportunities, and directions. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4724–4734. [CrossRef]
- Istiqomah, N.A.; Sansabilla, P.F.; Himawan, D.; Rifni, M. The Implementation of Barcode on Warehouse Management System for Warehouse Efficiency. *J. Phys. Conf. Ser.* **2020**, *1573*, 012038. [CrossRef]
- Laxmi, A.R.; Mishra, A. RFID based Logistic Management System using Internet of Things (IoT). In Proceedings of the 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 29–31 March 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 556–559. [CrossRef]
- Xu, L.D.; He, W.; Li, S. Internet of Things in Industries: A Survey. *IEEE Trans. Ind. Inform.* **2014**, *10*, 2233–2243. [CrossRef]
- Mostafa, N.; Hamdy, W.; Elawady, H. An Intelligent Warehouse Management System Using the Internet of Things. *Egypt. J. Eng. Sci. Technol.* **2020**, *32*, 59–65. [CrossRef]
- Zhang, C.; Chen, Y.; Chen, H.; Chong, D. Industry 4.0 and its Implementation: A Review. *Inf. Syst. Front.* **2021**. [CrossRef]
- Song, Y.; Yu, F.R.; Zhou, L.; Yang, X.; He, Z. Applications of the Internet of Things (IoT) in Smart Logistics: A Comprehensive Survey. *IEEE Internet Things J.* **2021**, *8*, 4250–4274. [CrossRef]
- Bellini, P.; Geest, M.V.; Tekinerdogan, B.; Catal, C. Smart Warehouses: Rationale, Challenges and Solution Directions. *Appl. Sci.* **2022**, *12*, 219. [CrossRef]
- Sun, X.; Yu, H.; Solvang, W.D.; Wang, Y.; Wang, K. The application of Industry 4.0 technologies in sustainable logistics: A systematic literature review (2012–2020) to explore future research opportunities. *Environ. Sci. Pollut. Res.* **2021**, *29*, 9560–9591. [CrossRef] [PubMed]
- Liu, X.; Cao, J.; Yang, Y.; Jiang, S. CPS-Based Smart Warehouse for Industry 4.0: A Survey of the Underlying Technologies. *Computers* **2018**, *7*, 13. [CrossRef]
- Sapry, H.R.M.; Ali, N.A.; Ahmad, A.R. Warehouse design and operation optimization. *J. Crit. Rev.* **2020**, *7*, 76–83.
- Cogo, E.; Žunić, E.; Beširević, A.; Delalić, S.; Hodžić, K. Position based visualization of real world warehouse data in a smart warehouse management system. In Proceedings of the 2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH), East Sarajevo, Bosnia and Herzegovina, 18–20 March 2020; pp. 1–6. [CrossRef]
- Lorenc, A.; Szkoda, M.; Szarata, A.; Jacyna-Golda, I. Evaluation of the effectiveness of methods and criteria for product classification in the warehouse. *Eur. J. Ind. Eng.* **2020**, *14*, 147–164. [CrossRef]
- Bolu, A.; Korçak, Ö. Adaptive Task Planning for Multi-Robot Smart Warehouse. *IEEE Access* **2021**, *9*, 27346–27358. [CrossRef]
- Davis, D. Amazon sales, Amazon revenue and Amazon annual profits. *Digital Commerce 360*. Available online: <https://www.digitalcommerce360.com/article/amazon-sales/> (accessed on 10 February 2022).
- Lundberg, M. Identifying the different viewpoints and key elements of digital transformation. School of Innovation, Design and Engineering, Malardalen University Sweden. Available online: <http://www.diva-portal.org/smash/get/diva2:1561538/FULLTEXT01.pdf> (accessed on 9 December 2021)
- Lorenc, A.; Lerher, T. PickupSimulo—Prototype of Intelligent Software to Support Warehouse Managers Decisions for Product Allocation Problem. *Appl. Sci.* **2020**, *10*, 8683. [CrossRef]
- Javaid, D.M.; Haleem, A.; Singh, R.P.; Suman, R. Substantial Capabilities of Robotics in Enhancing Industry 4.0 implementation. *Cogn. Robot.* **2021**, *1*, 58–75. [CrossRef]
- Aamer, A.M.; Sahara, C.R. Real-time data integration of an Internet-of-things-based smart warehouse: A case study. *Int. J. Pervasive Comput. Commun.* **2021**, ahead-of-print. [CrossRef]
- Lv, Y.; Lee, C.K.M.; Chan, H.K.; Ip, W.H. RFID-based colored Petri net applied for quality monitoring in manufacturing system. *Int. J. Adv. Manuf. Technol.* **2012**, *60*, 225–236. [CrossRef]
- Mostafa, N.; Hamdy, W.; Alawady, H. Impacts of Internet of Things on Supply Chains: A Framework for Warehousing. *Soc. Sci.* **2019**, *8*, 84. [CrossRef]

30. Laxmi, A.R.; Mishra, A. Automation in supply chain management system using Internet of Things (IoT). *Int. J. Eng. Technol.* **2018**, *7*, 777–783. [\[CrossRef\]](#)
31. Tejesh, B.S.S.; Neeraja, S. Warehouse inventory management system using IoT and open source framework. *Alex. Eng. J.* **2018**, *57*, 3817–3823. [\[CrossRef\]](#)
32. Sung, W.T.; Lu, C.Y. Smart Warehouse Management Based on IoT Architecture. In Proceedings of the 2018 International Symposium on Computer, Consumer and Control (IS3C), Taichung, Taiwan, 6–8 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 169–172. [\[CrossRef\]](#)
33. Deshmukh, P.R.; Bhalerao, D. An implementation of MQTT through the application of warehouse management system for climacteric fruits and vegetables. In Proceedings of the 2017 2nd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 19–20 October 2017; pp. 844–849. [\[CrossRef\]](#)
34. Macoir, N.; Bauwens, J.; Jooris, B.; Van Herbruggen, B.; Rossey, J.; Hoebeke, J.; De Poorter, E. UWB Localization with Battery-Powered Wireless Backbone for Drone-Based Inventory Management. *Sensors* **2019**, *19*, 467. [\[CrossRef\]](#)
35. Nagendra Guptha, C.; Bhaskar, M.; Meghasree, V. Design of IoT Architecture for order picking in a typical warehouse. In Proceedings of the 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), Bengaluru, India, 20–22 December 2018; pp. 50–53. [\[CrossRef\]](#)
36. Zhang, L.; Alharbe, N.; Atkins, A.S. An IoT Application for Inventory Management with a Self-Adaptive Decision Model. In Proceedings of the 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Chengdu, China, 15–18 December 2016; pp. 317–322. [\[CrossRef\]](#)
37. Liu, S.; Zhang, G.; Wang, L. IoT-enabled Dynamic Optimisation for Sustainable Reverse Logistics. *Procedia CIRP* **2018**, *69*, 662–667. [\[CrossRef\]](#)
38. Schwarz, M.; Milan, A.; Lenz, C.; Muñoz, A.; Periyasamy, A.S.; Schreiber, M.; Schüller, S.; Behnke, S. NimbRo picking: Versatile part handling for warehouse automation. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3032–3039. [\[CrossRef\]](#)
39. Jayanth, S.; Poorvi, M.B.; Sunil, M.P. Inventory Management System Using IOT. In Proceedings of the First International Conference on Computational Intelligence and Informatics, Hyderabad, India, 28–30 May 2016; pp. 201–210; Satapathy, S.C., Prasad, V.K., Rani, B.P., Udgata, S.K., Raju, K.S., Eds.; Advances in Intelligent Systems and Computing; Springer: Singapore, 2017; pp. 201–210.
40. Buntak, K.; Kovačić, M.; Mutavdžija, M. Internet of things and smart warehouses as the future of logistics. *Teh. Glas.* **2019**, *13*, 248–253. [\[CrossRef\]](#)
41. van Geest, M.; Tekinerdogan, B.; Catal, C. Design of a reference architecture for developing smart warehouses in Industry 4.0. *Comput. Ind.* **2021**, *124*, 103343. [\[CrossRef\]](#)
42. Rozanski, N.; Woods, E. *Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives*; Addison-Wesley: Boston, MA, USA, 2012.
43. Shabandeh, M. Topic: Apparel Market Worldwide. Available online: <https://www.statista.com/topics/5091/apparel-market-worldwide/> (accessed on 12 February 2022).
44. Panigrahi, R.R.; Jena, D.; Jena, A. Deployment of RFID Technology in Steel Manufacturing Industry—An Inventory Management Prospective. In *Advances in Machine Learning and Computational Intelligence*; Patnaik, S., Yang, X.S., Sethi, I.K., Eds.; Algorithms for Intelligent Systems; Springer: Singapore, 2021; pp. 705–719. [\[CrossRef\]](#)
45. Doss, R.; Trujillo-Rasua, R.; Piramuthu, S. Secure attribute-based search in RFID-based inventory control systems. *Decis. Support Syst.* **2020**, *132*, 113270. [\[CrossRef\]](#)
46. Dennert, K.; Friedrich, L.; Kumar, R. Creating an Affordable, User-Friendly Electronic Inventory System for Lab Samples. *SLAS Technol. Transl. Life Sci. Innov.* **2021**, *26*, 300–310. [\[CrossRef\]](#) [\[PubMed\]](#)
47. Yuvaraj, K.; Oorappan, G.M.; Megavarthini, K.K.; Pravin, M.C.; Adharsh, R.; Kumaran, M.A. Design and Development of an Application for Database Maintenance in Inventory Management System Using Tkinter and Sqlite Platform. In Proceedings of the IOP Conference Series: Materials Science and Engineering, International Conference on Mechatronics in Energy and Environment Protection (ICMEEP 2020), Erode, India, 16–17 October 2020; IOP Publishing: Bristol, UK, 2020, Volume 995, p. 012012. [\[CrossRef\]](#)
48. Kanagachidambaresan, G.R. Introduction to Wired and Wireless IoT Protocols in SBC. In *Role of Single Board Computers (SBCs) in Rapid IoT Prototyping*; Kanagachidambaresan, G.R., Ed.; Internet of Things; Springer International Publishing: Cham, Switzerland, 2021; pp. 99–120.
49. Sameon, S.S.B.; Yussof, S.; Jorgensen, B.N. Comparison between Communication Technology used in Smart Building. In Proceedings of the 2020 8th International Conference on Information Technology and Multimedia, ICIMU 2020, Selangor, Malaysia, 24–25 August 2020; pp. 212–217. [\[CrossRef\]](#)
50. Masykur, F.; Prasetyo, A.; Widaningrum, I.; Cobantoro, A.F.; Setyawan, M.B. Application of Message Queuing Telemetry Transport (MQTT) Protocol in the Internet of Things to Monitor Mushroom Cultivation. In Proceedings of the 2020 7th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), Semarang, Indonesia, 24–25 September 2020; pp. 135–139. [\[CrossRef\]](#)