Tampere University

Iita-Maria Salonen

PROCESS INDUSTRY 4.0 – EFFECT ON INTERFACES BETWEEN MES AND SHOP FLOOR INTEGRATIONS IN PULP AND PAPER INDUSTRY

Master of Science Thesis
Faculty of Engineering and Natural Sciences
October 2022

ABSTRACT

Iita-Maria Salonen: Process Industry 4.0 – Effect on Interfaces between MES and Shop floor Integrations in Pulp and Paper Industry
Master's Thesis
Tampere University
Automation Engineering
October 2022

The purpose of this thesis is to research how Industry 4.0 affects integrations between MES and shop floor in pulp and paper industry. Industry 4.0 is a generally used term for the fourth industrial revolution introducing modern technologies and producing principles into manufacturing. These modern technologies include big data and analytics, cloud computing, and IoT. Four Industry 4.0 design principles, interconnection, information transparency, decentralized decisions, and technical assistance, are handled as central principles when designing Industry 4.0 compliant factories. The impact of Industry 4.0 on factories covers the entire system, including system architecture, modern technologies on shop floor, and new communication methods and protocols.

The thesis can be divided into two parts, theoretical and practical part. First in the theoretical part, comprehensive literature review was conducted to find out Industry 4.0 related trends that affect the shop floor. As in pulp and paper industry little Industry 4.0 related research has been done, the scope of the literature review covered also research done in other industrial fields. Based on the findings of the literature review, Industry 4.0 compliant prototype was designed and implemented. The design and implementation of the prototype form the practical part of the thesis.

The most promising trends that are likely to be seen in factories when moving towards Industry 4.0 compliant smart factories, are smarter sensors, devices, and products, new wireless communication technologies and IoT messaging protocols, cloud and fog computing, service-oriented architecture, and decentralisation of decision making. As new communication technologies, such as IoT messaging protocols, seemed to be an important part of almost every finding in literature review, prototype was decided to be built based on communication using OPC UA PubSub over MQTT.

In this thesis it is concluded that investing in Industry 4.0 is crucial for business success in the future. In several articles and other sources introduced in this thesis the positive impact of Industry 4.0 solutions on factories has been shown. These benefits include e.g., increased profitability and productivity, more adaptability, and solutions for more complicated customer needs and scarce resources. Companies working with Manufacturing Execution Systems should be prepared to all the changes discussed in this thesis, although, as stated in the thesis, many of the recent technologies need to be tested more thoroughly in real production environment prior to concluding their suitability for pulp and paper industry. The implemented prototype gives promising results and indicates that communication using OPC UA PubSub over MQTT is relatively easy to implement.

Keywords: Industry 4.0, Manufacturing Execution System, Shop Floor Integrations

TIIVISTELMÄ

Iita-Maria Salonen: Process Industry 4.0 – Effect on Interfaces between MES and Shop floor Integrations in Pulp and Paper Industry
Diplomityö
Tampereen yliopisto
Automaatiotekniikka
Lokakuu 2022

Tämän diplomityön tarkoituksena on tutkia, kuinka Industry 4.0 vaikuttaa tuotannonohjausjärjestelmien ja lattiatason laitteiden välisiin integraatioihin paperi- ja selluteollisuudessa. Termiä Industry 4.0 käytetään yleisesti viittaamaan teollisuuden neljänteen vallankumoukseen, joka tuo mukanaan uusia teknologioita ja tuotantoperiaatteita teollisuuteen. Näit uusia teknologioita ovat massadata (engl. Big Data) ja analytiikka, pilvipalvelut ja esineiden internet (engl. Internet of Things). Neljää Industry 4.0 tuotantoperiaatetta, toisiinsa liittäminen, tiedon läpinäkyvyys, hajautettu päätöksenteko ja tekninen avustus, on käsitelty keskeisinä periaatteina suunnitellessa Industry 4.0 yhteensopivia tehtaita. Industry 4.0:n vaikutus teollisuuteen kattaa koko järjestelmän, mukaan lukien järjestelmäarkkitehtuurin, lattiatason uudet teknologiat ja uudenlaiset kommunikointimetodit ja -protokollat.

Tämä diplomityö voidaan jakaa kahteen osaan, teoreettiseen ja käytännön osuuteen. Ensin teoreettisessa osuudessa eli kattavassa kirjallisuuskatsauksessa selvitettiin Industry 4.0:n tuomia trendejä, joilla on vaikutusta lattiatasoon. Koska aiheesta on tehty vain vähän tutkimusta liittyen juuri paperi- ja selluteollisuuteen, myös muihin teollisuudenaloihin liittyvä tutkimus sisällytettiin kirjallisuuskatsaukseen. Perustuen kirjallisuuskatsauksen tuloksiin suunniteltiin ja toteutettiin Industry 4.0:n mukanaan tuomiin trendeihin perustuva prototyyppi. Prototyypin suunnittelu ja toteutus muodostavat yhdessä diplomityön käytännön osuuden.

Lupaavimmat trendit, joita tullaan näkemään siirryttäessä kohti Industry 4.0:n mukanaan tuomia älykkäitä tehtaita, ovat älykkäämmät sensorit, laitteet ja tuotteet, uudet langattomat viestintäteknologiat ja IoT-viestintäprotokollat, pilvipalvelut, palvelupohjainen arkkitehtuuri ja päätöksenteon hajauttaminen. Koska uudet viestintäteknologiat, kuten IoT-viestintäprotokollat, näyttivät olevan tärkeä osa lähes jokaista muuta kirjallisuuskatsauksessa esille tullutta trendiä, prototyyppi päätettiin rakentaa pohjautuen IoT-viestintäprotokollaan, josta käytetään englanniksi nimitystä OPC UA PubSub over MQTT.

Tässä diplomityössä todetaan, että panostaminen Industry 4.0:aan on erittäin tärkeää tulevaisuudessa yrityksen menestymisen kannalta. Useissa tässä diplomityössä esitellyissä artikkeleissa ja muissa lähteissä on osoitettu Industry 4.0:n myönteiset vaikutukset tuotantoon. Nämä hyödyt kattavat esimerkiksi lisääntyneen kannattavuuden ja tuottavuuden, enemmän joustavuutta ja ratkaisuja entistä monimutkaisempiin asiakkaiden tarpeisiin sekä niukkoihin resursseihin. Yritysten, jotka työskentelevät tuotannonohjausjärjestelmien parissa, tulisi valmistautua muutoksiin, joita esitellään tässä opinnäytetyössä, vaikkakin, kuten mainittu opinnäytetyössä, uusia teknologioita tulisi testata perusteellisemmin todellisessa tuotantoympäristössä ennen kuin niiden sopivuudesta paperi- ja selluteollisuuteen tehdään lopullisia päätelmiä. Toteutettu prototyyppi antoi lupaavan esimerkin OPC UA PubSub over MQTT protokollan käytöstä ja osoitti toteutuksen olevan melko yksinkertaista.

Avainsanat: Teollisuus 4.0, Tuotannonohjausjärjestelmä, Lattiatason integraatiot

PREFACE

This thesis process began in the spring 2022 when I was contacted by ABB, where I worked as a trainee during the previous summer. I was offered an opportunity to write a thesis and work simultaneously part-time for ABB. First, I would like to thank Anne Niemenmaa and Juha Karppanen from ABB for an interesting topic for my thesis and support and guidance through the whole writing process. I feel that they trusted me and my progress with the thesis. I would also like to thank Luis Gonzalez Moctezuma, who was my thesis instructor at Tampere University, for all the valuable feedback and guidance.

This thesis is the end of my studies at Tampere University, at least for now. I'm grateful for all the people I met during these years at the university, and for my friends and family who have supported me during this whole time.

Hyvinkää, 31st October 2022

Iita-Maria Salonen

CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

AI	Artificial Intelligence
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
BITKOM	Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V.
CoAP	Constrained Application Protocol
CPPS	Cyber Physical Production System
CPS	Cyber Physical System
CRL	Certificate Revocation List
DDS	Data Distribution Service
DSS	Decision Support System
DT	Digital Twin
DTLS	Datagram Transport Layer Security
EPIS	European Pulp Industry Sector
ESDA	Enterprise Scalable Data Architecture
GPS	Global Positioning System
gRPC	Google Remote Procedure Call
GSM	Global System for Mobile Communications
IBPP	Internal Business Process Performance
ICT	Information and Communication Technology
IIoT	Industrial Internet of Things
IIRA	Industrial Internet Reference Architecture
IoE	Information of Everything
IoT	Internet of Things
IP	Internet Protocol
IPsec	Internet Protocol Security Architecture
ISA	International Society of Automation
IT	Information Technology
JSON	JavaScript Object Notation
MES	Manufacturing Execution System
MESA	Manufacturing Enterprise Solutions Association
MIA	Middleware for Intelligent Automation
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
NFC	Near-field Communication
OPC UA	Open Platform Communications Unified Architecture
OPC UA FX	OPC Unified Architecture Field eXchange
OS	Operating System
OT	Operational Technology
PLC	Programmable Logic Control
QoS	Quality of Service
RAMI 4.0	Reference Architectural Model Industrie 4.0
REST	Representational State Transfer
RF	Radio Frequency
RFID	Radio Frequency Identification
RTP	Real-time Transport Protocol
SASL	Simple Authentication and Security Layer
SCP	Supply Chain Performance
SCTP	Stream Control Transmission Protocol
SHM	Shared Memory
SSH	Secure Shell
SSL	Secure Sockets Layer

TCP	Transmission Control Protocol
TLS	Transport Layer Security
TSN	Time Sensitive Networks
UDP	User Datagram Protocol
WSN	Wireless Sensor Network
XMPP	Extensible Messaging and Presence Protocol

1. INTRODUCTION

Rapid development of new technologies, such as artificial intelligence, IoT (Internet of Things), and cloud computing, have brought us to the point of The Fourth Industrial Revolution, often referred to as Industry 4.0. Industry 4.0 can be seen as a vision of the industrial production of the future which brings intelligence, flexibility, and efficiency to every field of industry. It will affect the whole product value chain from shop-floor devices of single factory to integrated cross-company networks and supply chains. [1] Manufacturing Execution Systems (MES) are key enablers of Industry 4.0 compliant manufacturing as they enable connection and coordination of new technologies included in Industry 4.0 [2]. Nevertheless, to meet the requirements of the new way of manufacturing, manufacturing execution systems must be developed to support more intelligent smart factories of the future. The goal of this thesis is to answer the following research questions:

Q1: What research has been done on the impact of Industry 4.0 on shop floor and shop floor devices in pulp and paper industry?

Q2: What qualitative and quantitative metrics are affected by Industry 4.0 implementations in the shop floor?

Q3: How changes in shop floor affect integrations between MES and shop floor devices?

Based on the answers to these questions, a prototype of Industry 4.0 compliant shop floor link is implemented.

Industry 4.0 is a rather new term, and it was first introduced as late as 2011 with its German name, Industrie 4.0 [3]. Since then, it has been a topic of major interest and a lot of research has been done on Industry 4.0 and related technologies. Many of these technologies affect the shop floor and change the whole way of manufacturing. In recent years also the role of MES as an important part of Industry 4.0 compliant factory has drawn attention. For example, in the article “The Industry 4.0 revolution and the future of Manufacturing Execution Systems (MES)” [4] Almada-Lobo presents four aspects that should be taken into consideration when developing MES for the fourth industrial revolution. Mantravadi et al. [5] wrote an overview of next-generation manufacturing execution systems and their importance for Industry 4.0, and in the article “Development of manufacturing execution systems in accordance with Industry 4.0 requirements: A review of standard- and ontology-based methodologies and tools” [6] Jaskó et al. presented recent Industry 4.0 trends and their effect on the development of MES. These papers show the importance of MES in Industry 4.0 and when considered all the changes coming to shop floor, the research related to integration between MES and shop floor is of great importance.

According to [7], process industry in comparison with other industries has already been closer to the vision of Industry 4.0 with almost fully automated production and data collection. In the beginning, the focus of Industry 4.0 was in technologies that enabled interconnection between higher-level functions and individual assets which has already been present for decades in process industry. Also, most use cases for Industry 4.0 were earlier in the automotive industry which poorly reflects the reality in process industry. Due to these reasons Industry 4.0 wasn’t initially able to show its potential in the process industry as well as in other industry fields. However, when Industry 4.0 is seen as a whole, covering the whole product value chain, and focusing also on people and processes instead of just technologies, its value for process industry is evident. More particularly, in pulp and paper industry Industry 4.0 provides tools and technologies to satisfy the increasing requirements related to environmental issues, resource scarcity, and customer needs.

In the MPI 2020 Industry 4.0 study [8] manufacturers across all industrial sectors were interviewed on their attitudes towards Industry 4.0 and on the results which they have gained applying Industry 4.0 solutions. Due to Industry 4.0, more than 5 percent increase was reported in productivity by 61% of the firms and in profitability by 43% of the firms in the previous year, and even greater percentages of companies are expecting increase both in the productivity and profitability in the future over the next five years. In [9], the impact of Industry 4.0 on corporate financial performance was examined. The study showed clearly that the Industry 4.0 maturity level has a significant effect on the internal business process performance (IBPP), and through supply chain performance (SCP), IBPP affects customer performance. The impact of IBPP, SCP, and customer performance on financial success have been consistent, and thus, the importance of Industry 4.0 for the growth of the company in the future is evident. These studies show how important it is for companies to invest in Industry 4.0 to gain competitive advantage.

In this thesis the focus is on the integration between MES and shop floor devices in Industry 4.0 compliant paper and pulp industry. The thesis has both theoretical part and practical part. In chapter 2, theoretical background is provided covering the basic concepts of Industry 4.0, MES, and shop floor integrations. In the third chapter, the research methodology is presented covering both the literature review and practical part of the thesis. In the fourth chapter the results of the literature review are presented, and it answers the research questions, what research has been done on the effect of Industry 4.0 on the shop floor and shop floor devices in pulp and paper industry, how these changes in the shop floor affect integrations between MES and shop floor, and what metrics are affected by these implementations. The literature review is the core of the theoretical part of the thesis. Chapter 5 introduces how the Industry 4.0 compliant prototype is designed and the sixth chapter handles the prototype implementation phase. The design and the implementation of the prototype form the practical part of the thesis. Before conclusions in the last chapter, results of the thesis are concluded and discussed in Chapter 7.

2. THEORETICAL BACKGROUND

This chapter presents the theoretical background for the thesis. First, Industry 4.0 is introduced including the key technologies and reference architectures. Next, manufacturing execution systems and their features and role in the future are discussed in chapter 2.2. In chapter 2.3, the third key part of the theoretical background, shop floor integrations, are explained more thoroughly.

2.1 Industry 4.0

The concept of Industry 4.0 is hard to define exactly. In the research published by BITKOM (Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V.) [10] over 100 definitions for Industry 4.0 were examined, and there was a lot of variation between definitions depending on the perspective. A few common factors were recognized, regardless of the industry field. These factors were intelligent networks between people, machines, products, and ICT (information and communication technology) systems, and implementation of some Industry 4.0 applications. Although there are some technologies that can be seen as part of Industry 4.0, their impact on the industry depends on the industry field and thus, the exact definition can’t be formed. However, some of the key technologies are presented in chapter 2.1.1, but factory can be seen Industry 4.0 compliant even if all the technologies are not applied in production.

Although there is no exact definition for Industry 4.0, some design principles have been introduced. Hermann et al. presented in their article “Design Principles for Industrie 4.0 Scenarios” [11] four main principles that are commonly referred in other literature as well. These principles are interconnection, information transparency, decentralized decisions, and technical assistance. First of these, interconnection, refers to the same concept as the first common factor recognized in the research published by BITKOM which is intelligent networks between people, machines, sensors, and devices. One of the technologies usually connected to Industry 4.0, IoT, is one of the Industry 4.0 key technologies since its importance for interconnection is significant. Sometimes the concept of IoT is expanded to IoE, Internet of Everything, which emphasizes the connections between people and machines, as well. However, the term IoT is a lot more often used than IoE, and in literature IoT is often used to refer, in addition to the connections between devices, also the connections between machines and people (e.g. [12], [13]).

The second design principle, information transparency, refers to the ability to exchange real-time information beyond enterprise boundaries [5] and to collecting, managing, and organizing huge amount of data from the factory [14]. This data can be further analyzed and used for, e.g., identifying gaps and improving processes [15]. Data from sensors can be linked with digital models of the plant to create digital twins (DT) which are, according to some sources, essential part of industry 4.0 (e.g. [14]).

The third principle, decentralized decisions, builds on the first two principles, interconnection, and information transparency. With transparent data flow from interconnected things, decisions can be made in a decentralized manner which means that, instead of one central system that makes all the decisions, individual entities can make their own decisions trying to optimize their own objectives [16]. The last principle, technical assistance, refers to systems capable to help humans in their work. For example, technical assistance can mean tools that help visualizing information or systems that are capable to perform tasks that are dangerous for people. [15]

2.1.1 Key Technologies

According to [15, p. 27], industrial revolution is a “process of transformation in technologies, which brings significant changes in the economic, cultural, and social structures of humans”. This definition highlights the importance of technological advancements in industrial revolution. The goal of this chapter is to present the most important technologies that act as a driving force for Industry 4.0. As stated in chapter 2.1, complete list of key technologies is hard to form, since not all technologies are relevant on ever industry field. However, three technologies are almost every time mentioned in the literature about Industry 4.0. These are IoT, cloud computing, and big data analytics (for example in [1][6][9][17]), and these are technologies that are introduced in this chapter more thoroughly. Cybersecurity, horizontal and vertical systems integration, artificial intelligence, and machine learning are technologies that are often mentioned as separate key technologies to enable Industry 4.0, but in this thesis, they are seen as enablers or parts of one of the three technologies mentioned earlier. Cyber physical systems (CPS), smart factories, and digital twins are often considered as key technologies, but in this thesis, they are seen as more wider concepts that are composed of other technologies, and hence, they are discussed separately in chapter 2.1.3. Other technologies that are sometimes mentioned in literature considering Industry 4.0 are augmented or virtual reality, blockchain, simulation, autonomous or collaborative robots, additive manufacturing, GPS (Global Positioning System), mobile technology, RFID (Radio Frequency Identification), and nanotechnology.[14][17][18][19][20] These technologies will have applications in Industry 4.0 compliant factories, but most likely not in all industrial fields. Therefore, these technologies are not handled in this thesis as key technologies.

Internet of Things

From the perspective of the scope of this thesis, IoT is perhaps the most relevant key technology in Industry 4.0 since its effect on changes in shop floor are remarkable. When focused on industrial domain, the abbreviation IIoT, industrial internet of things, is often used as well. Because this thesis focuses only on industrial domain, these abbreviations could be used interchangeably, but for the sake of clarity only IoT will be used in this thesis.

Apparently, the term Internet of Things was first used in year 1999 by Kevin Ashton [21]. Since then, the definition of the term has had many variations, but its main characteristics include connectivity, intelligence of objects and networks, and ubiquity [12]. The main idea is that interconnected things, such as sensors, actuators, humans, and appliances, form a network that enables more efficient, advanced, and accurate functioning thanks to improved communication and interaction between heterogenous network nodes [15].

The significance of IoT for Industry 4.0 can be better understood if the Industry 4.0 design principles introduced in Chapter 2.1 are examined. It can be noted that IoT has its role in each of them. The whole concept of IoT is all about interconnection, the first principle, but also to enable information transparency, decentralized decisions, and technical assistance, the intelligence of devices enabled by IoT is required. Actually, the role of the IoT in Industry 4.0 is so significant that Industrial Internet of Things is sometimes used as a synonym for Industry 4.0 (e.g. [22] [23]).

IoT implementation requires new technologies, such as new communication protocols and sensor technologies. One of the most relevant technologies is wireless communication, including protocols such as ZigBee and Wi-Fi. Another important aspect is new sensor technologies that enable effective and accurate data collection and thus data sharing among devices. IoT also emphasizes the importance of cyber-security since the connection of devices to internet exposes the devices to cyberattacks.

Cloud Computing

The second Industry 4.0 key technology is cloud computing which is often used as a general term also including the concepts of fog and edge computing. Cloud computing refers to the using of computing services, such as data storage and computing power, through internet. The usage is based on demand and thus, cloud computing enables scalability of the systems. Other important characteristics of cloud computing are its capability to process data fast and in great amounts, and reliability due to its security policies. [15]

Fog computing refers to distributed computing model which means that computing is done partly in cloud, and partly in the edge meaning devices between cloud and shop floor, such as routers, gateways, or specific fog devices. It combines the advantages of scalable and flexible cloud computing with the advantages of edge devices which offer smaller latency to data processing since the data is processed closer to the devices. Figure 1 represents the idea of fog computing. [12]

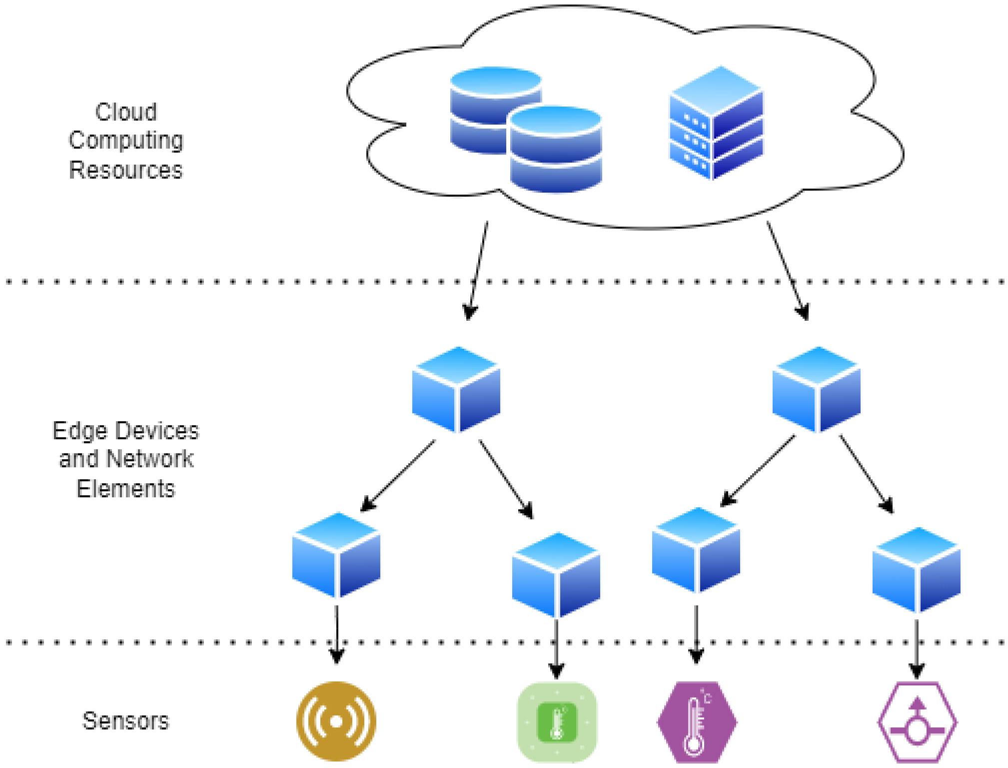


Figure 1 The basic idea of fog computing (based on [12])

The importance of fog computing is especially important for decentralized decision making when considered the Industry 4.0 design principles. When cloud computing is combined with edge computing, some of the decisions can be made in intelligent edge devices.

Big Data and Analytics

The third Industry 4.0 key technology is big data and analytics. Big data refers to huge volume of data from heterogeneous sources and in various formats. Data can be structured, un-structured, or semi-structured. Usually, data is generated fast which requires high processing speed. All these characteristics of data makes it complex to process.

The big data to be valuable, it must be processed and analyzed. This means processing data in a way that meaningful information can be extracted from the data. This meaningful information can be visualized to help concerned personnel to make data-based decisions even better. [15] AI (Artificial Intelligence) and ML (Machine Learning), both sometimes considered as Industry 4.0 key technologies, have huge role in Big Data analytics, since they enable advanced analytics. The role of Big Data, in terms of Industry 4.0 design principles, is especially significant when it comes to information transparency.

2.1.2 Reference Architectures

Since the concept of Industry 4.0 is somewhat unclear and complex, there has been attempts to design Industry 4.0 reference architectures to simplify the concept. The purpose of this chapter is to represent the two most often used architectures, RAMI 4.0 (Reference Architecture for Industrie 4.0) and IIRA (Industrial Internet Reference Architecture).

RAMI 4.0 was developed and published in 2015 as a result of cooperation between several institutions [24]. The reference architecture is represented in Figure 2.

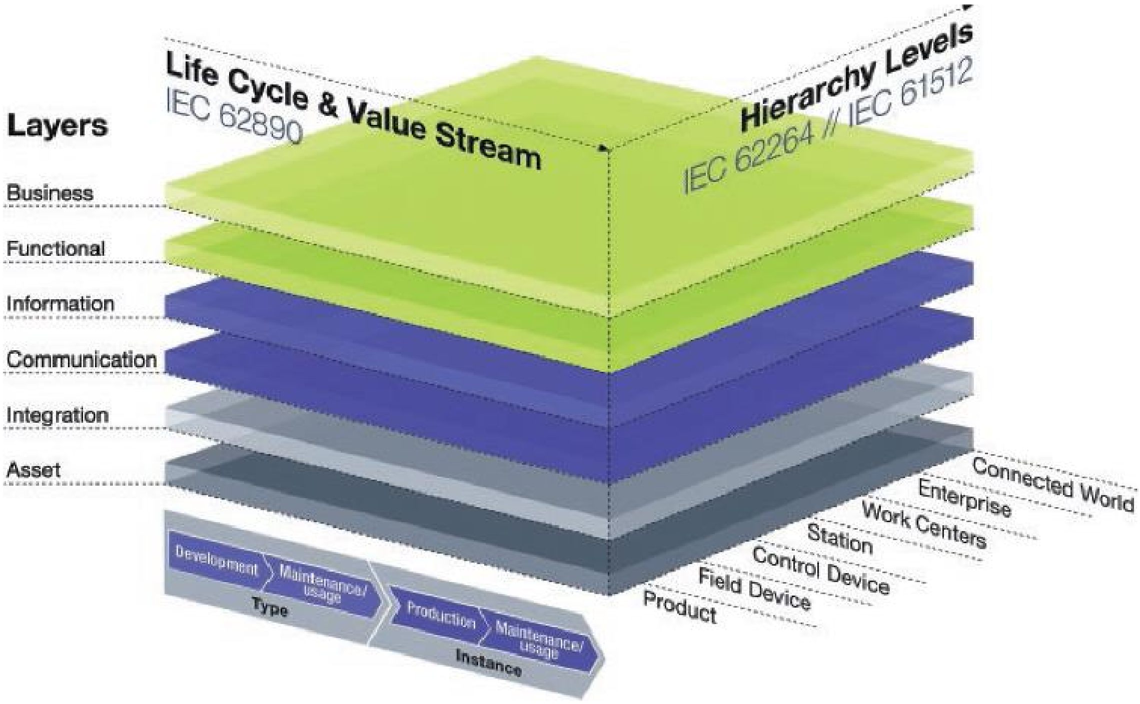


Figure 2 RAMI 4.0, Reference Architecture for Industrie 4.0 [24, p. 7]

The model consists of six layers that represent different perspectives on Industry 4.0. The first layer from top to bottom is business layer which concerns e.g., integrity between business functions and different business processes, and legal and regulatory frameworks. Below business layer is functional layer which serves as a run time environment for applications and services supporting business processes. In the middle are information layer and communication layer, first of which concerns data integration and integrity, data provision, and pre-processing, and second of which provides services for the next layer, integration layer, and controls the standardization of communication. Integration layer contains IT connected elements, such as sensors and HMI, and provides information for assets. The last layer at the bottom is asset layer which represents real world including, inter alia, physical components and people. [24]

RAMI 4.0 also considers hierarchy levels and the life cycle and value streams which are presented in other axis of the architecture model. Hierarchy levels are defined according to IEC 62264 and IEC 61512 standards, and they are considered more from functional than implementation perspective. IEC 62890 standard acts as a guideline for life cycle. Inclusion of life cycle and value stream in RAMI 4.0 emphasizes their role in Industry 4.0. [24]

IIRA is another commonly used reference architecture for Industry 4.0, and it was published as well in 2015 [25]. Figure 3 represents this framework.

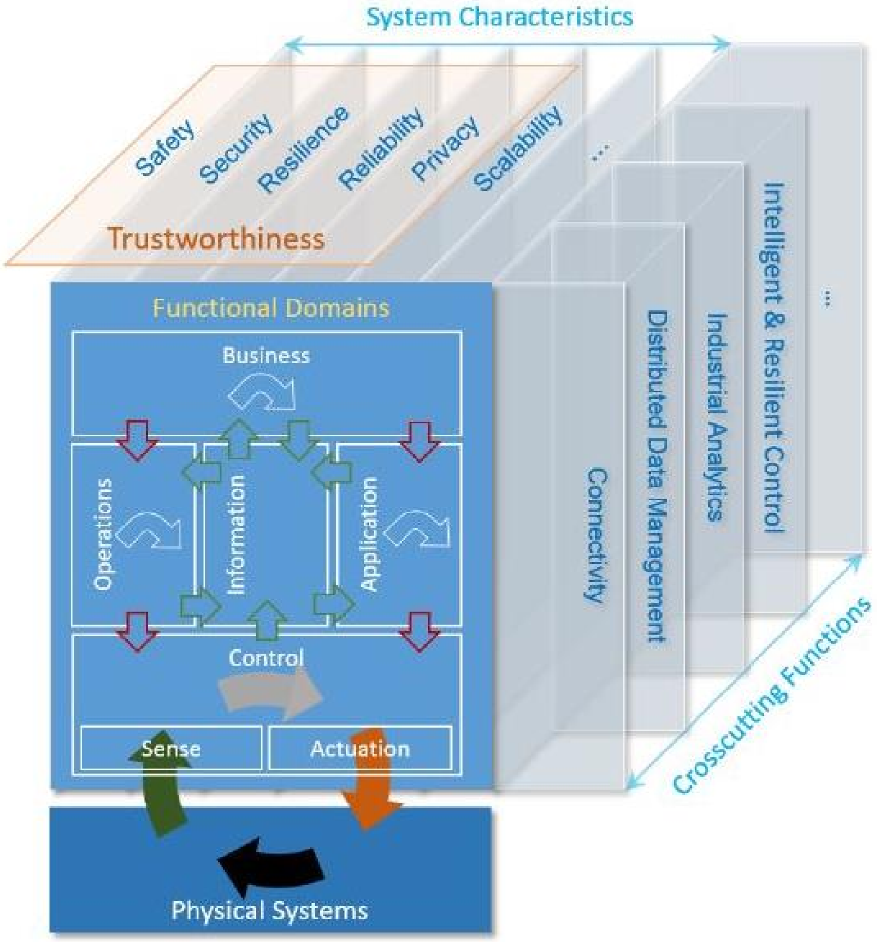


Figure 3 IIRA Industrial Internet Reference Architecture [25, p. 35]

In the center of the architecture framework are functional domains, namely business, operations, information, application, and control domain, which are decomposed from a typical industrial internet system. Each of them represents some distinct functionality of the whole system. On top are listed some key characteristics of systems that should be taken into consideration in functional domains and generally in the whole system, such as reliability and safety. On the right side some functions considering the whole system are listed, for example connectivity and industrial analytics. Just as system characteristics, these should be noted in the overall system implementation. System characteristics and crosscutting functions included in the architecture are based on the key system concerns identified in industrial internet systems. [25]

2.1.3 Cyber Physical Systems, Smart Factories, and Digital Twins

Terms “Cyber Physical System”, “Smart factories”, and “Digital Twins” are often used in the context of Industry 4.0. Some sources define these as Industry 4.0 key technologies, but since they are rather results of combining other technologies than just simple separate technologies, in this thesis they are discussed separately.

According to [26], there’s no unanimously accepted definition for CPS, and at its simplest CPS could be defined as a system that is used for monitoring and controlling physical world. Often CPS is defined as a combination of “cyber” systems with physical ones creating a system in which virtual and physical world interact together. Virtual components can interact together with physical ones through virtual copies of physical components that include digitalized data and information about the real-world asset. [17] In [26], three main components of CPS were defined as computation and control, communication, and monitoring and manipulation. Several CPS cooperating together via digital networks form a CPPS (cyber physical production system). [26]

Similar to CPS, there’s no exact definition for the term “Smart Factory”. In [27], the following definition was proposed:

“A Smart Factory is a manufacturing solution that provides such flexible and adaptive production processes that will solve problems arising on a production facility with dynamic and rapidly changing boundary conditions in a world of increasing complexity. This special solution could on the one hand be related to automation, understood as a combination of software, hardware and/or mechanics, which should lead to optimization of manufacturing resulting in reduction of unnecessary labor and waste of resource. On the other hand, it could be seen in a perspective of collaboration between different industrial and nonindustrial partners, where the smartness comes from forming a dynamic organization."

This definition gives quite wide framework for practical implementation of smart factory and avoids defining separate technologies or practices that smart factory must include. The definition is in line with the goals of the fourth industrial revolution aiming for more flexible production. Smart industry is often used as a synonym for Industry 4.0. Thus, smart factory as a part of smart industry is a component of industry 4.0 along with other components.[28] The relationship between CPS and Smart factory is somewhat unclear but often it is defined that CPS and CPPS combined with IoT form the base for Smart Factory.

Digital twin is a digital representation of some process, asset, or system. As explained in [14], DTs and IoT overlap in some respects and thus can't be completely separated, but the main idea is that IoT connects devices and enables data collecting whereas DTs manage and structure the data for later use and thus enable automation and optimization with the help of ML and AI. DTs are often seen as an essential part of Industry 4.0, and they are extremely often mentioned in literature related to Industry 4.0.

2.2 Manufacturing Execution Systems

According to MESA (Manufacturing Enterprise Solutions Association) [29, p. 1] the international definition for MES is:

"Manufacturing Execution Systems (MES) deliver information that enables the optimization of production activities from order launch to finished goods. Using current and accurate data, MES guides, initiates, responds to, and reports on plant activities as they occur. The resulting rapid response to changing conditions, coupled with a focus on reducing non-value-added activities, drives effective plant operations and processes. MES improves the return on operational assets as well as on-time delivery, inventory turns, gross margin, and cash flow performance. MES provides mission-critical information about production activities across the enterprise and supply chain via bi-directional communications"

Although this definition was published already in 1997, the idea behind it is like the idea behind Industry 4.0: vision of more effective manufacturing enabled by utilizing data and connecting activities across the enterprise. However, MES is a centralized system and one of the design requirements of Industry 4.0 was decentralized decision making. How will MES adapt to the new requirements and what will its role be in the Industry 4.0? This is a question that is handled in chapter 4. This chapter focuses on introducing the core MES features.

2.2.1 Core MES Features

In [29, pp.5-6], MESA presents MESA-11, eleven principal functions of MES which are:

1. Operations Scheduling: Based on priorities, attributes, and characteristics of production units, MES plans operation schedule to minimize, inter alia, waste, time, or energy.
2. Resource Allocation and Status: MES tracks resources such as machines, materials and tools, and their status. It gives instructions and reserves resources to meet the objectives of operation scheduling.
3. Dispatching Production Units: MES manages the flow of production unit by giving commands to send materials or orders to certain parts of the plant. Prescribed schedule can be modified, if needed.
4. Document Control: Documents, such as recipes, work instructions, or drawings, are managed and sent to the operations that need them.
5. Product Tracking and Genealogy: MES records the history of the product by tracking work on production units and information, such as serial number, component materials and their suppliers, or alarms during the production.
6. Performance Analysis: The actual results of manufacturing operations are provided in comparison with past results and expected business result.
7. Labor Management: Information related to work force such as status of personnel, time and attendance are reported.
8. Maintenance Management: MES ensures the availability of tools and equipment for manufacturing, and maintains history data of past problems and events
9. Process Management: Production is monitored and directed, decision support is provided, and process activities can be corrected automatically if needed.
10. Quality Management: Quality measurements are collected throughout the system and based on them quality analysis is provided
11. Data Collection / Acquisition: Data about processes and resources is collected, monitored, and organized

Many of these core functions support each other and work together to enable more effective and higher quality manufacturing. Since 1997, when these core functions were defined, MES systems have worked well based on these functions and fulfilled their purpose. Although some of these core functions will be still in the future critical part of the MES, now, at the edge of the next industrial revolution, re-designing of MES is critical to support the information needs of factories in the era of Industry 4.0 [30].

2.3 Shop Floor

According to Merriam-Webster dictionary [31], the official definition for shop floor is “the area where products are made in a factory” or “the workers in a factory”. In this thesis the term is used in its first context referring to the area where production machinery is. Other words used in the same context are ‘factory floor’ meaning the same as ‘shop floor’ and ‘field device’ meaning the automation devices on the shop floor. For clarity, only terms ‘shop floor’ and ‘shop floor devices’ are used in this thesis.

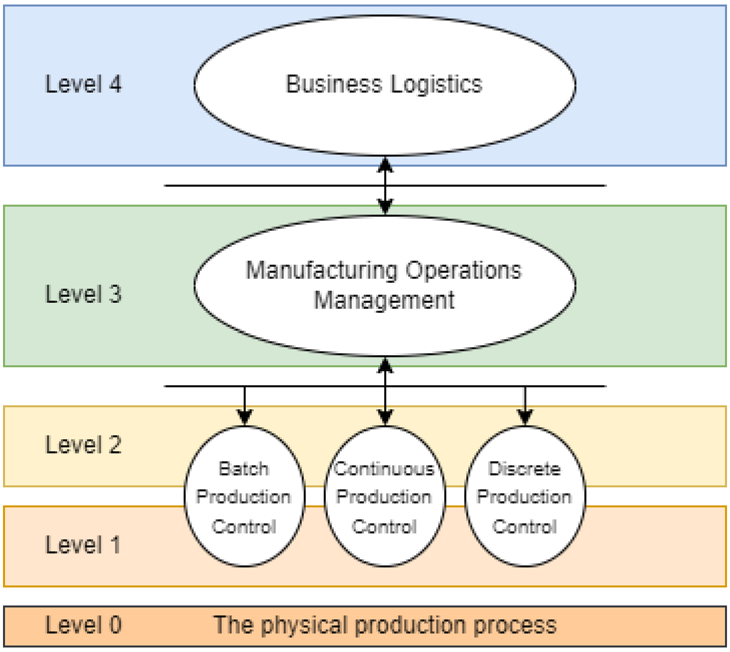


Figure 4 ANSI/ISA95 Automation control hierarchy (based on [32])

In traditional ANSI/ISA95 automation control hierarchy (see Figure 4) created by International Society of Automation (ISA) shop floor refers to the levels 0, 1, and 2. These levels consist of sensors, actuators, and shop floor control devices, such as PLCs. There are several protocols for communication between shop floor devices and more will come due to Industry 4.0. However, this thesis focuses on the integration between shop floor and higher levels of automation pyramid. There’s still one protocol that should be mentioned in this context, OPC UA.

OPC (Open Platform Communication) is used nowadays as a standardized interface between the automation pyramid levels and automation systems from shop floor to enterprise level systems. OPC UA (OPC Unified Architecture) is the latest specification which was developed to replace all the earlier OPC specifications. Some of its most important features are platform-independence, reliability, and interoperability. OPC UA has two fundamental components, data modelling and transports mechanisms. Data modelling refers to rules and practices required for the information model. It defines, e.g., base building blocks, base types, type hierarchy, and entry points. Transport mechanisms define the optimized transportation mechanisms for different kinds of use cases. [33] OPC UA specification was released in 2008 [34]. Since then, it has evolved to meet the requirements of Industry 4.0. These changes are discussed more in chapter 4.3.2.

3. METHODOLOGY

The purpose of this chapter is to present the research methodologies. In this thesis the design process represented in Figure 5 was followed.

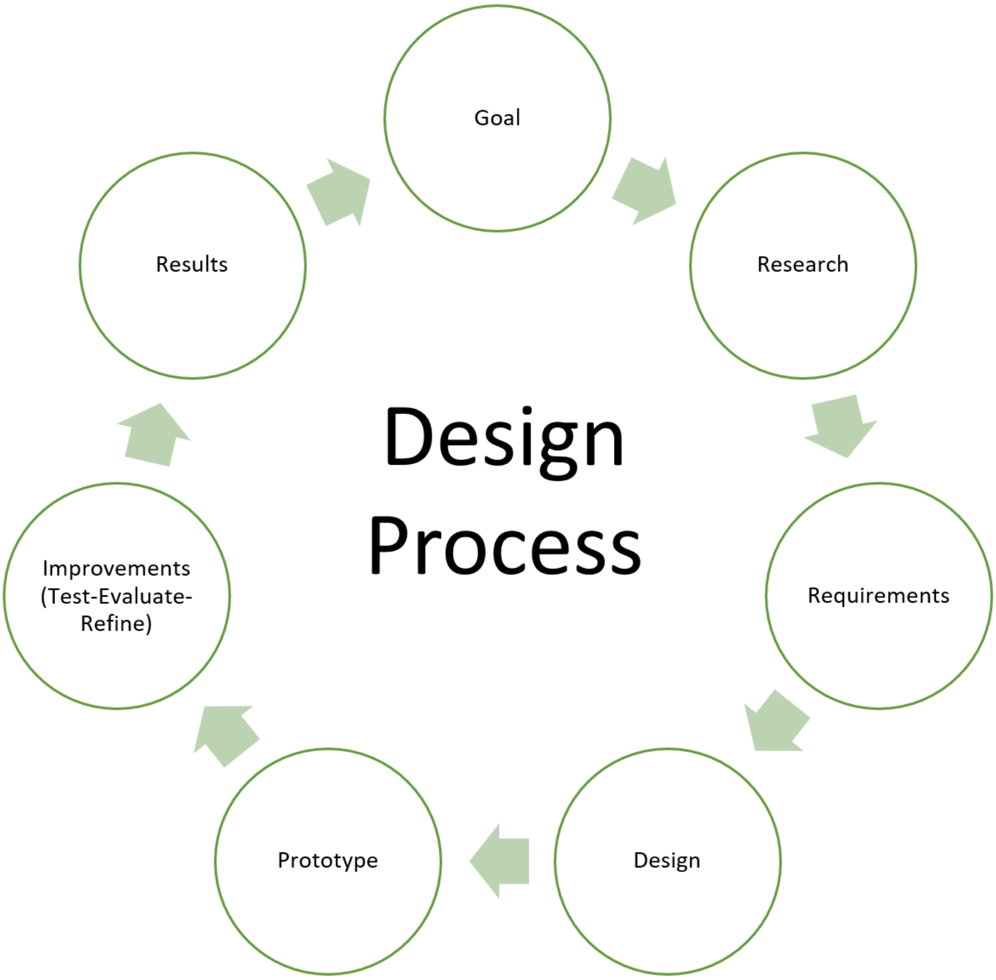


Figure 5 Phases of the design process

The first phase, goal, was already presented in introduction: to research Industry 4.0 compliant integrations between MES and shop floor and implement a prototype. The next phase, research, was done through a literature review, which is discussed more thoroughly in chapter 3.1 and 4. Chapters 3.2 and 5 introduce how the requirements for the prototype were defined and how the prototype is designed. The rest of the phases, prototype implementation, testing and improvements, and results are handled in chapters 3.3, 6, and 7.

3.1 Literature Review

The goal of the literature review was to research information related to the research questions which were

Q1: What research has been done on the impact of Industry 4.0 on shop floor and shop floor devices in pulp and paper industry?

Q2: What qualitative and quantitative metrics are affected by Industry 4.0 implementations in the shop floor?

Q3: How changes in shop floor affect integrations between MES and shop floor devices?

Articles were searched mainly in Scopus database. Also, Tampere University Library's discovery service Andor was used to find complementary information. Scopus was chosen as the main database because it offers wide covering of engineering articles and a possibility for title-abs-key search. The search was conducted with the following search terms:

Table 1 Literature review search terms

no.	Database	Keyword	Search method	Results
1	Scopus	(MES OR "manufacturing execution system") AND ("Industry 4.0" OR "Industrie 4.0")	Title-ABS-Key	221
2		("shop floor device" OR "factory floor device" OR "field device") AND ("Industry 4.0" OR "Industrie 4.0")	Title-ABS-Key	45
3		(pulp OR "paper industry") AND ("shop floor device" OR "factory floor device" OR "field device") AND ("Industry 4.0" OR "Industrie 4.0")	All fields	2
4		("IT/OT convergence" OR "OT/IT convergence") AND "industry 4.0"	All fields	36
5	Andor	(pulp OR "paper industry") AND ("Industry 4.0" OR "Industrie 4.0")	All fields	2 320

The articles found in Scopus were systematically researched. Of the articles found in Andor, only articles published after 2015 were examined since the goal of this thesis is to research new trends affecting the shop floor. Different combinations of thesis keywords were used as search terms. To maximize the search results, synonyms for shop floor device were used as alternative terms. Both MES and the term “manufacturing execution system” were used. Keywords also included the alternative term for Industry 4.0, its original German counterpart “Industrie 4.0”, which is sometimes used in research papers written in English instead of Industry 4.0. IT/OT or OT/IT convergence refers to integration between operational technology (OT) and information technology (IT). This was used as a search term (Table 1 row 4) since it is often used to refer to MES – shop floor integration.

Many of the articles included information which was related to industry 4.0 and its effect on MES and shop floor, but which still was out of the scope of this thesis. These articles handled e.g., other integrations than the integration between MES and shop floor, such as integrations between enterprises, between MES and ERP, or between different IoT devices, or only a specific topic related to Industry 4.0, such as cybersecurity. The content of these articles is not handled in this thesis. Only articles that offered valuable insight to research questions are handled more thoroughly in Chapter 4 which represents the findings of the literature review.

3.2 Requirements and Design

After the research, the next phase is to define requirements and restrictions for the prototype. This is done through requirement analysis. First, requirements are gathered based on the intentional usage and operating environment. Then, requirements are analyzed and defined clearly. Every requirement should be able to be verified. After analysis requirements are modelled and documented.

Based on the requirements, prototype can be designed. Design consists of logical and physical design. In logical design, data and process models are created. Data model represents information or data objects required by the prototype as entities and their relations. Process model is a representation of application processes and related inputs and outputs.

The goal of the physical design is to create the architecture of the system. The architecture must support the requirements and the logical design of the system. The design of the architecture includes software, hardware, and interfaces between different parts of the system. At the end of the design phase, it is ensured that the design of the system fulfils all the requirements.

3.3 Prototype Implementation, Improvements, and Results

The prototype is implemented as designed in the design phase. For the improvements the prototype must be tested. Before testing a test plan is created based on the requirements. As mentioned in chapter 3.2, each requirement should be able to be verified. Each requirement must be taken into consideration when creating a test plan so that it is ensured that every requirement is verified during the testing. If any faults or other problems are noticed during the testing phase, improvements must be implemented.

When all the tests have been conducted, improvements done, and the prototype seems to be ready, results are gathered based on the desired design, requirements, and the test plan. The functioning and the implementation of the prototype are evaluated taking into consideration the usability of the prototype and related technologies in a real manufacturing environment. Also, the complexness of the solution and implementation is evaluated. The most important part of the results is to evaluate if the original goal was fulfilled.

4. LITERATURE REVIEW RESULTS

The purpose of this chapter is to present the results of the literature review. About the literature review generally, it can be said that there is little research related to the effect of Industry 4.0 on the shop floor specifically in pulp and paper industry which can also be seen in the number of results in the third row in Table 1 in Chapter 3.1. Despite the great number of results when searched in Andor, only a few articles were found that gave valuable insight to Industry 4.0 generally in pulp and paper industry. In the article “An Industry 4.0 Research Review For The Pulp And Paper Industry” [35], 93 publications related to key performance practices of industry 4.0 in pulp and paper industry were reviewed. However, some of the reviewed publications were published before the term “Industry 4.0” even existed, and the article included also publications related to, for example, water or energy management. The article stated that there’s a gap in research related to CPS, IoT, and MES in pulp and paper industry. From the perspective of the scope of this thesis, these technologies would have given most relevant information related to the trends in shop floor and their effect on integrations between MES and shop floor. However, despite the lack of research in pulp and paper industry, it can be assumed that general trends in the shop floor, especially in other fields of process industry, reflect the situation in pulp and paper industry, as well. These changes are discussed in chapter 4.2. Before that, general changes in the automation hierarchy and architecture are handled in chapter 4.1. Finally, in chapter 4.3, it is discussed how these changes affect the integration between MES and shop floor.

4.1 Changes in the Architecture

Several articles researched in literature review showed significant changes in the architecture and hierarchy of automation. One of the key elements in Industry 4.0 is decentralized decision making which changes the relationships between different automation elements. ISA, the original creator of hierarchical automation pyramid, demonstrates the change as in Figure 6.

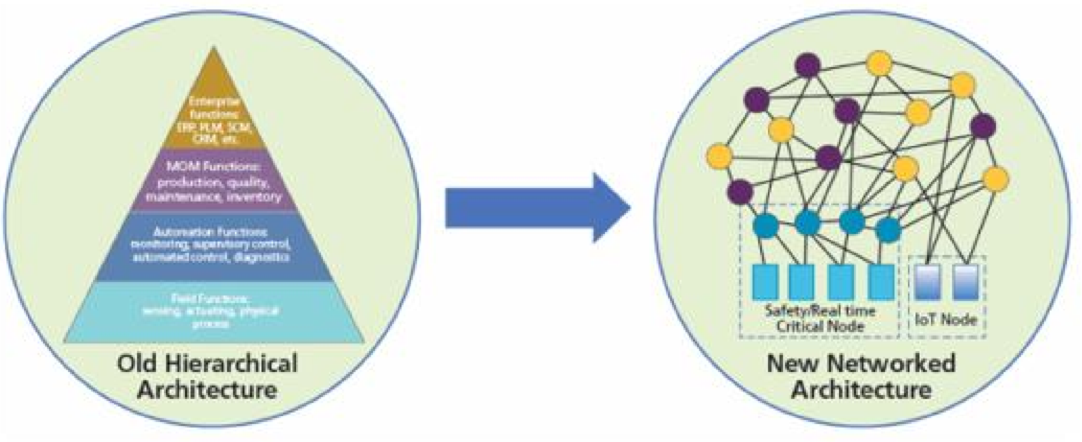


Figure 6 Old automation hierarchy changing towards distributed architecture [32]

As MES has originally been a system for centralized control, the trend towards decentralized decision making may invoke negative reactions among MES providers. However, decentralization doesn’t necessarily mean physical decentralization but instead logical one, which means that the MES can remain as one application but some of its functions are logically decentralized among shop floor assets. For example, CPS and smart products know their characteristics, state, and history, and based on these they can do some decisions themselves. This type of decentralization was one of the four aspects mentioned by Almada-Lobo [4] among vertical integration, connectivity and mobile, and cloud computing and analysis, for improving MES for Industry 4.0.

This kind of decentralization requires some changes also in physical aspect. How to enable flow of information between heterogenous assets? In several papers different kind of middleware are introduced to solve this problem. For example, Coito et al. have introduced Middleware for Intelligent Automation (MIA) in multiple research [36][37][38]. The basic idea is represented in Figure 7.

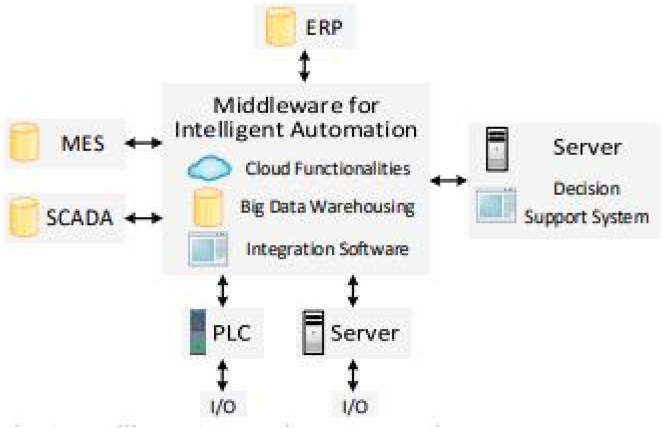


Figure 7 Middleware for Intelligent Automation [36]

The idea of the proposed middleware is to better enable real-time reaction capabilities by centralizing all the data sources into separate gateways or access points. MIA includes three toolsets which are cloud functionalities, big data warehousing, and integration software. Cloud functionalities implemented through fog computing enable decentralization of data processing and decrease latency when needed. With Big Data Warehousing real-time data acquisition and processing of unstructured data can be facilitated. The proposed model aims to provide middleware that enables simultaneous connections with all the automation layers, direct connections between endpoints, ability to handle different communication requirements, and ability to communicate with new systems by using communication adapters. The primary function of integration software is to enable these MIA features. In the article, OPC UA over TSN (time sensitive networks) was proposed as the communication approach due to its capabilities with horizontal and vertical interoperability.

IoT platforms are another solution that can act as a middleware. They enable better communication, device management, and data flow between assets. IoT platform itself is not an application, although framework for building applications, and its purpose is to link machines, applications, devices, and people to control and data centers, by comprising technologies, such as cloud computing and wireless communication. [39] According to [40], a good IoT platform should have capabilities for scalability, reliability, customization, and security. It should support different protocols and interfaces, and be cloud and hardware agnostic, and it should be built according to rigid architecture and efficient technology stack. In 2019, there were 620 publicly known IoT platforms on the market half of which were for industrial purposes which shows their significance for future manufacturing [41].

In the articles found in literature review, there were several examples of IoT platform solutions. One of these is OSPS (Open Scalable Production System) framework, represented in Figure 8, developed by European-funded R&D projects, H2020 FASTEN and H2020 ScalABLE 4.0 [42].

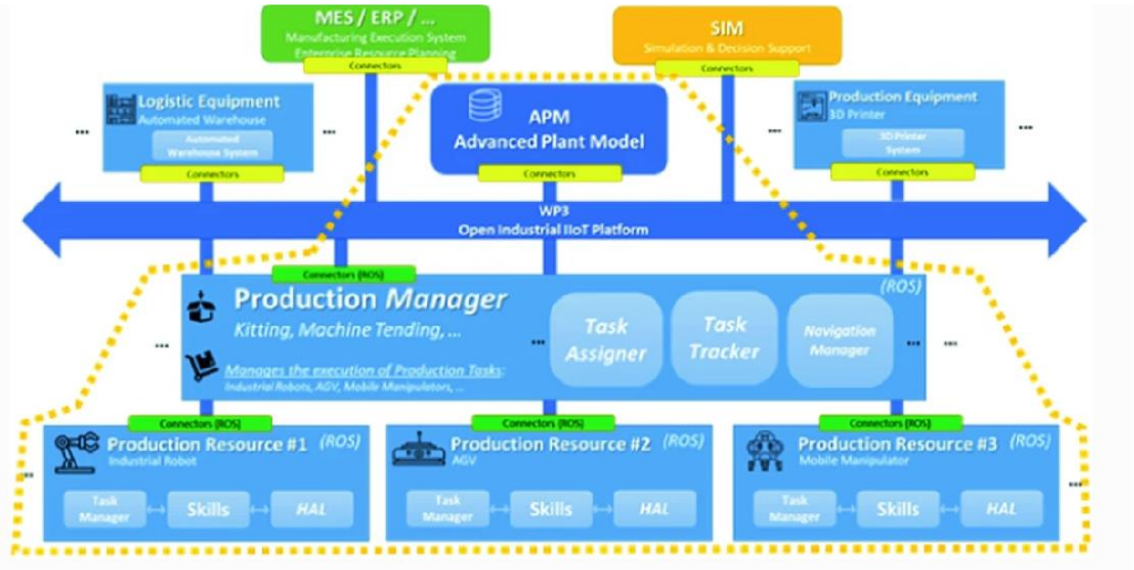


Figure 8 OSPS Framework [42]

The introduced framework aims to fully connected and scalable manufacturing, and integration of robotics, automation, simulation, and decision-support system (DSS) [42]. This framework was used in [43] to show its potential in applying simulation into manufacturing. Industrial IoT platform is used to support communication between different systems, including the communication between MES and shop floor. In the article, the role of the platform in communication was emphasized, although IoT platforms often offer solutions for, e.g., data handling and storing as well. Communication happens through IoT Message Broker either by “publish-subscribe” or “request-response” pattern.

“Request-response” communication pattern, or Client/Server communication, is more traditional way of communication between two devices. It is based on requests done by clients, and responses to requests done by servers. The importance of “Publish-subscribe”, or Pub/Sub pattern, has grown due to the requirements of more scalable information systems. Instead of one-to-one communication like in Client/Server communication, Pub/Sub has many-to-many approach. Between communicating devices there’s a message broker. Devices can “publish” and “subscribe” topics. For example, one sensor can publish a measurement as a topic to message broker. Message broker delivers the measurement to all the devices that have subscribed that particular topic. Figure 9 represents these two communication patterns.

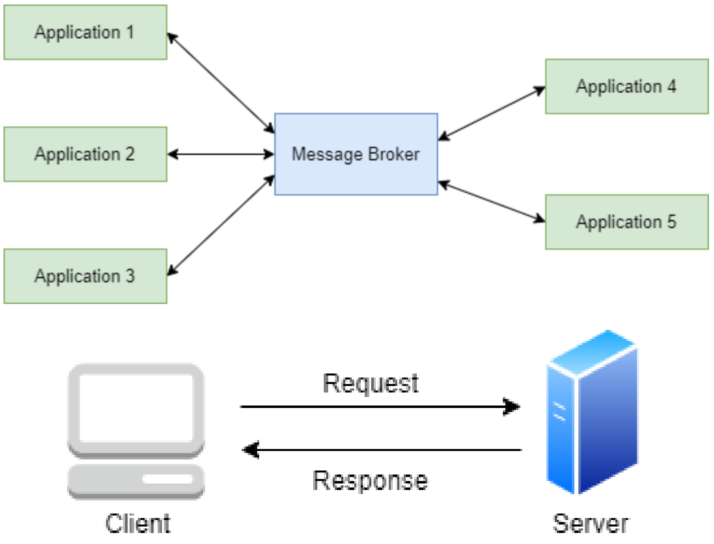


Figure 9 Comparison of Publish/Subscribe and Client/Server communication approaches

In the reviewed literature, message brokers were one of the most frequently mentioned approaches. In Figure 10 there's represented one implementation, Enterprise Scalable Data Architecture (ESDA), proposed in [44].

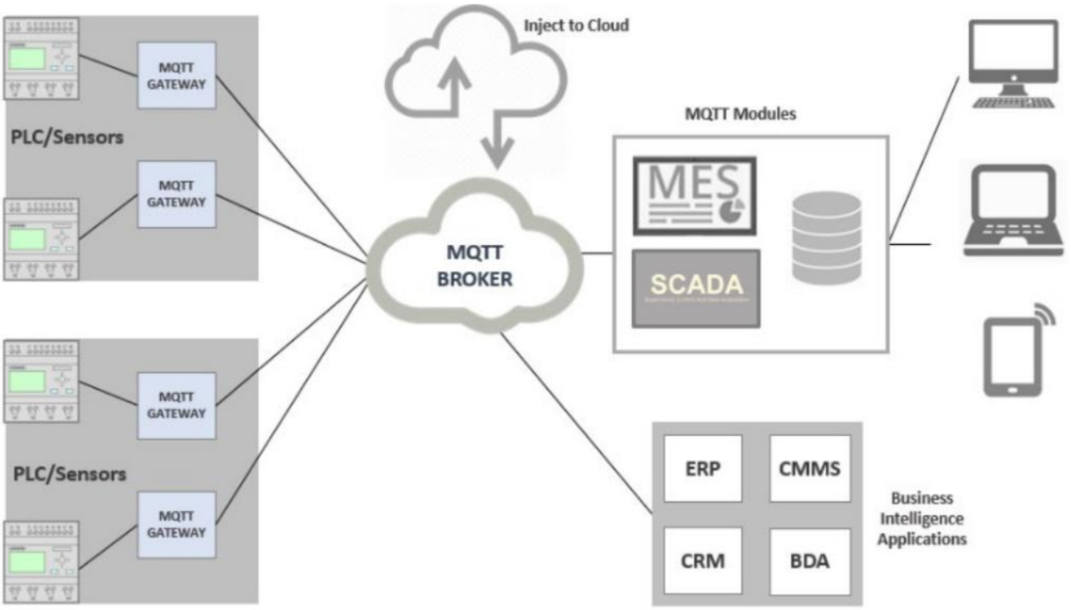


Figure 10 Enterprise Scalable Data Architecture [44]

The proposed architecture is based on MQTT (Message Queuing Telemetry Transport) broker. MQTT is a communication standard, about which there's further information included in chapter 4.3.2. Based on the literature, MQTT brokers seem to be extremely popular in IoT implementations, and thus they can be seen one of the most common trends affecting communication integrations in industry.

Although the idea of middleware seems promising for Industry 4.0, there might be some problems in the implementation. First and the most important aspect is security and privacy. When middleware is implemented as part of IoT, the system is exposed to threats related to internet such as cyberattacks. Fortunately, this challenge has been widely recognized and continuous work towards more secure protocols and practices is being done. Another problem related to a middleware is the lack of application support. As the amount of IoT devices and developers is constantly growing the integration to middleware should be standardized and simple so that integrations with new devices won’t cause problems in the future. As the amount of data will increase in tremendous pace in smart factories, the middleware should be able to process and handle incomplete data from heterogenous sources. Without this capability, the advantages coming with IoT will be wasted. [45]

In pulp and paper industry, where lifetime of production equipment is relatively long, also the IT systems should be able to work reliably for decades. From the middleware perspective this requires regular updates and technical support for long after the implementation. As time goes, the security threats evolve and change, as well, which must be considered before any IoT implementations. As the communication between shop floor and enterprise level is extremely time-critical especially in some parts of the pulp and paper production, the reliability of the connection must be secured. Internet connection time-outs or other problems in internet connection might cause serious economic damages in production.

4.2 Changes in the Shop Floor

One of the key characteristics of Industry 4.0 is “smartness”. In shop floor, this means smart devices. Based on systematic literature review, Silverio-Fernandez et al. [46] proposed following definition for the term “smart device”: “A smart device is a context-aware electronic device capable of performing autonomous computing and connecting to other devices wire or wirelessly for data exchange”. This is in line with the earlier introduced concept of CPS. Most importantly, smartness requires from shop floor devices capabilities for better communication and data acquisition, storing and processing. Some of the data processing previously done on higher levels of automation will be done already by the device itself.

Based on the literature reviewed, one of the most significant changes in shop floor due to Industry 4.0 is the increasing amount of data needed for implementing smart factory. For example, embedded digital twins and future predictive maintenance systems require more advanced measuring technologies. For this, smart sensors are one of the foundational enabling technologies. In [47], new sensor technologies for Industry 4.0 were discussed, among them smart sensors and their capabilities, nuclear sensors, and micro and nano sensors. According to Rahman et al. [48], in pulp and paper industry more advanced sensors have so far been used mainly for monitoring the process instead being integrated into control systems. In their article, they showed a solution based on near infrared (NIR) spectroscopy based soft-sensor and feedforward model predictive control to manage delignification process in pulp production.

One special case of a sensor is RFID sensor which was mentioned in several reviewed articles. RFID sensors are used for identification which is fundamental for better traceability required by smart factories. According to EPIS (European Pulp Industry Sector) [49], pulp producers want RFID to be the preferred identification technology for pulp products. For this, in 2020 GS1 RFID Identification of Pulp Products Guideline was published [50]. In 2019, Stora Enso introduced repulpable, water-soluble, and paper-based RFID tag, ECO Bale, which is available for all pulp producers [51]. These advances in RFID technology and some positive experiences in using RFID in pulp and paper industry [52] [53] indicate that the amount of RFID sensors is very likely to increase in the future.

To collect data from the increasing number of sensors, wireless sensor networks (WSNs) have gained popularity. WSNs are comprised of nodes cooperating in a network. Each node has processing capability, some sort of memory, RF (radio frequency) transceiver, power source, and one or more sensors and actuators. The data collected by the nodes is sent to the base station which is further connected to the Internet as represented in Figure 11. This is how acquisition of data is possible despite the location. [54]

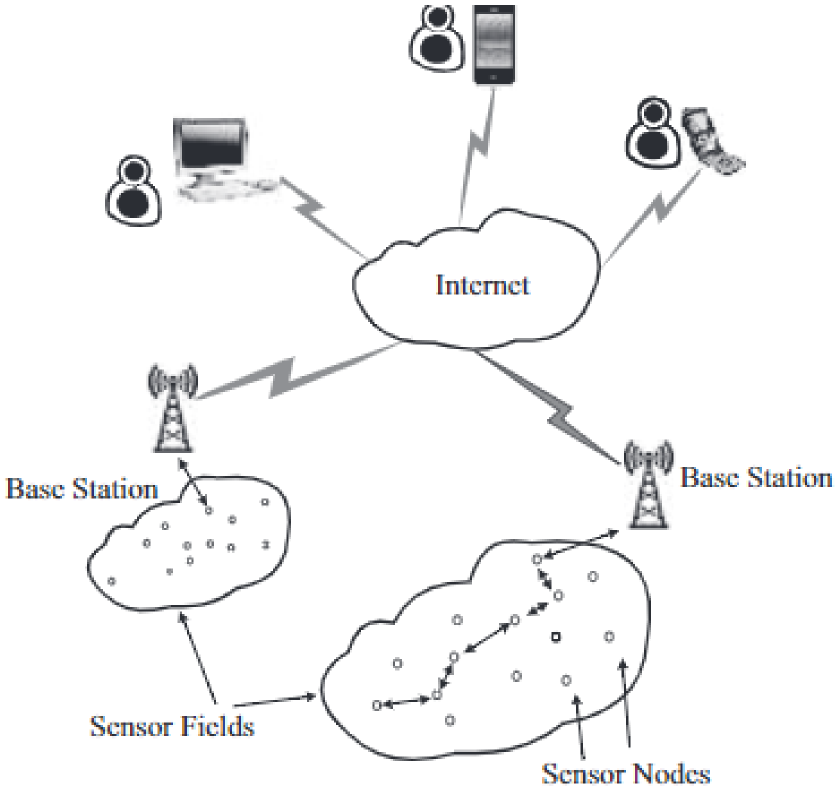


Figure 11 Functional principles of Wireless Sensor Networks [54]

In the article “Industrial IoT Monitoring: Technologies and Architecture Proposal” [55], it was stated that WSNs are fundamental to meet the Industry 4.0 requirements. The article presented main WSN standards, namely ISA100.11a, WIA-PA, WirelessHART, and ZigBeePRO, and proposed an architecture for monitoring the networks. Ahlen et al. [56] researched the potential of wireless control and WSN in process industry. A case study was conducted at Iggesund Paperboard mill, and the results indicated that despite the challenges wireless control systems work even in a complex industrial environment. However, the tests were conducted only on the starch cooker at the papermill, and the potential of the technology should be also tested in other parts of the pulp and paper production to see its usefulness.

In chapter 2.1.1 mentioned fog computing also affects the situation at the shop floor. Instead of direct links between MES and the devices, there’s edge devices between cloud-based services and IoT devices. The purpose of these devices is to act as software containers that can pre-process, spool, and compress data. Field devices can push collected data to and pull required data through edge devices. [57] Edge devices are also one type of middleware. If compared to cloud computing, fog computing might be more promising technology for real-time production like pulp and paper as it decreases latency. The implementation however requires more from design, since it must be decided what functions should be implemented on edge and what on cloud to achieve the optimal solution.

4.3 Changes in the Integration between MES and Shop Floor

The changes in the shop floor inevitably affect the integration between devices and MES. To support the functionalities of Industry 4.0 compliant factories MES and MES integrations must evolve. One of the trends affecting these integrations is service orientation. This means that MES is composed of services that include one or several functions. Services can be used through REST (Representational State Transfer) APIs (Application Programming Interface) by e.g., shop floor devices, and they have unambiguous input and output parameters. In addition to REST APIs, gRPC (Google Remote Procedure Call) and GraphQL APIs are other options but so far, they haven’t been as widely used. However, they might be more suitable for some use cases and thus should be taken into consideration. [58] Interoperability and Plug & Produce capabilities are another trend affecting the MES. Flexibility requirements of future factories insist on ability to recognize, manage, and implement changes automatically to the factory. [59]

Key enablers for future MES-shop floor integrations are new communication protocols which are required for effective communication with earlier mentioned technologies, such as WSNs, smart sensors, and middleware. Relevant communication protocols can be divided into wireless communication technologies and IoT messaging protocols. In the reviewed literature, several of these communication technologies were mentioned which shows their significance for future manufacturing.

4.3.1 Wireless Communication Technologies

Wireless communication technologies are key enablers for IoT and Industry 4.0. In reviewed literature some of them were mentioned. Table 2 represents the basic characteristics of each wireless communication technologies mentioned in the reviewed articles. The information in the table is based on the sources [60], [61], [62], [63], [64], [65], [66], [67], [68], [69], [70].

Table 2 Wireless communication technologies

	Range (m)	Data rate	Frequency (GHz)	Year
Wi-Fi	50-100	11-54 Mbps	2.4 / 5	1991
Bluetooth	10	125 Kbps -3 Mbps	2.4	1994
ZigBee	60-3200	250 Kbps -1 Mbps	2.4	2003
WirelessHART	250	250 Kbps	2.4	2007
6LoWPAN	10-200	250 Kbps	2.4	2007
GSM	500-120000	up to 200 Mbps	0.4-30	1991
ISA100.11a	50-500	250 Kbps-6.8 Mbps	2.4	2009
NFC	0.1	424 Kbps	0.01356	2004

Wireless communication technologies are listed in the table in order of prevalence in the reviewed literature. Absolutely the most often mentioned wireless communication technology was Wi-Fi. Bluetooth and ZigBee were used quite often and the same number of times. WirelessHART and 6LoWPAN were used in several articles, about half of the times in comparison with ZigBee and Bluetooth. GSM (Global System for Mobile Communications), ISA100.11a and NFC (Near-field Communication) were mentioned a few times.

In the context of pulp and paper industry, few articles considering wireless technologies were found. In [71], Wi-Fi based sensing technology was presented for measuring the moisture content of the woodchips. In [56], WirelessHART and ISA100 were mentioned as enabling technologies for implementing WSN at paper mill. However, generally related to process industry there are more implementations with wireless communication technologies. The problems with the connection issues in wireless networks might however cause problems in real-time processes. Their usefulness in all parts of pulp and paper production process is thus questionable and should be researched more.

4.3.2 IoT Messaging Protocols

There are five IoT messaging protocols, namely MQTT, AMQP (Advanced Message Queuing Protocol), DDS (Data Distribution Service), XMPP (Extensible Messaging and Presence Protocol), and CoAP (Constrained Application Protocol). In Table 3 basic characteristics of these protocols are represented. As with the wireless communication technologies, protocols are listed in the table in order of prevalence. The information in the table is based on the sources [72], [73], [74], [75].

Table 3 IoT communication protocols

	Transport Protocol	Communication Model	Security	Year
MQTT	TCP	Publish/Subscribe	TLS/SSL	1999
CoAP	UDP, SCTP	Request/Response, Publish/Subscribe	DTLS, IPsec	2010
AMQP	TCP, SCTP	Request/Response, Publish/Subscribe	TLS/SSL, IPsec, SASL	2003
DDS	TCP, UDP, SHM	Publish/Subscribe	TLS/SSL	2004
XMPP	TCP, UDP, RTP	Publish/Subscribe	TLS/SSL	2004

MQTT was by far the most used protocol in reviewed literature, and it appeared four times more often than CoAP and AMQP. DDS was mentioned almost as often as CoAP and AMQP, but XMPP appeared only once in the reviewed literature. To clarify the relationship between IoT messaging protocols, transportation protocols, and earlier introduced wireless communication technologies, they are placed in TCP/IP (Transmission Control Protocol / Internet Protocol) protocol stack in Figure 12.

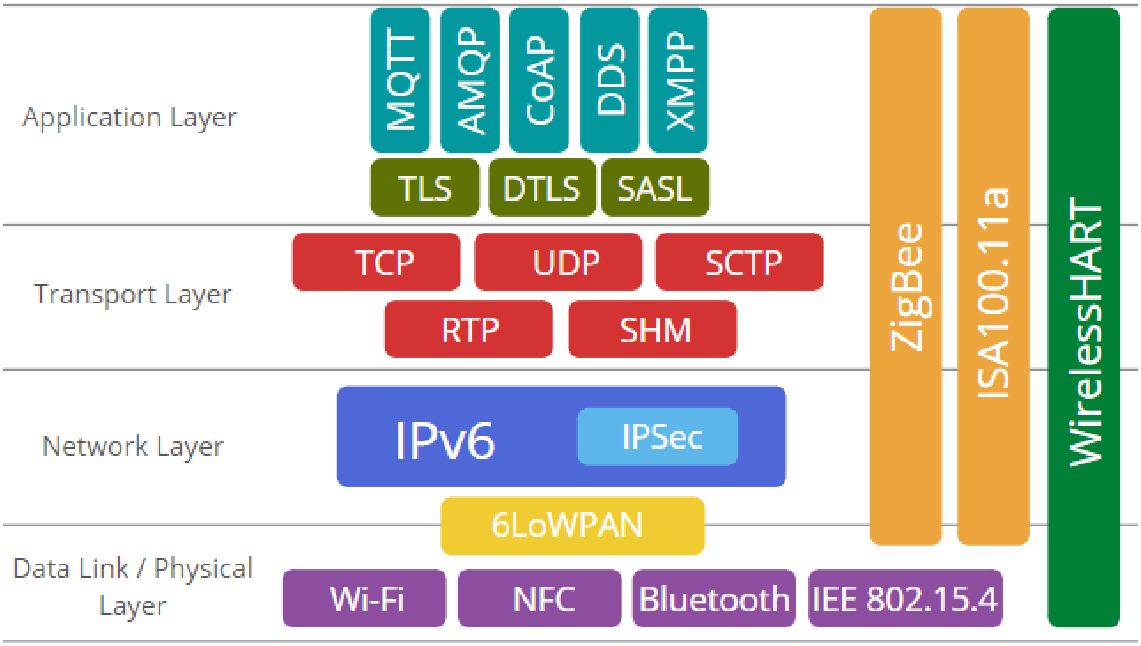


Figure 12 IoT protocol stack (based on [76] [77])

Although all five protocols are used in IoT, they have some differences as can be seen in Table 3. One of the differences is the transport protocol over which messaging protocols work. Different transport protocols include TCP, UDP (User Datagram Protocol), SCTP (Stream Control Transmission Protocol), RTP (Real-time Transport Protocol), and SHM (Shared Memory) transport protocol. TCP and SCTP are both reliable protocols with flow and congestion control mechanisms, unlike UDP which is connectionless and simpler protocol [78]. RTP and SHM are less common protocols. RTP is fast and simple stateless protocol used for real-time streaming [79], and SHM enables communication on the same machine through shared memory [80]. TLS (Transport Layer Security), DTLS (Datagram Transport Layer Security), SASL (Simple Authentication and Security Layer), and IPsec (Internet Protocol Security Architecture) enable more secure communication for IoT messaging. Another difference between different IoT messaging protocols is the communication pattern, either publish/subscribe or request/response. These were already introduced chapter 4.1.

As with the wireless communication technologies, few articles were found considering the IoT messaging protocols in pulp and paper industry. In [81], implementation of MQTT broker was suggested for more efficient data flow in pulping process, and in [82] MQTT implementation was used for remote monitoring system for papermaking equipment. In commercial sector, ANDRITZ has developed a platform for recovery boiler monitoring in pulp production, and the software of the platform uses AMQP messaging [83]. However, there are more articles related to IoT messaging protocols in other fields of process industry and process industry generally.

The usage of IoT protocols in pulp and paper industry hasn’t so far been researched enough to see its usefulness. The problems already discussed in chapter 4.1 related to internet-based connections cover the biggest problems also with the IoT messaging protocols. These problems include especially real-time capabilities and security.

In chapter 2.3, OPC UA was shortly introduced. Sometimes OPC UA is compared straight with IoT messaging protocols such as MQTT and AMQP. However, the comparison is not so straightforward. In 2018, OPC UA specification Part 14: PubSub was released. Before this, OPC UA communication was based on client/server communication pattern, but the new specification enables the use of publish/subscribe pattern which is more scalable and thus suits in some cases better for IoT communication. OPC UA PubSub can be used over UDP like some of the IoT messaging protocols, but in addition OPC UA PubSub can be used over other IoT messaging protocols, AMQP or MQTT. [84]

Using OPC UA over TSN, or OPC UA FX (OPC UA Field eXhange), was already mentioned in chapter 4.1. This approach for industry 4.0 communication is seen as one of the most prominent ones [85], and it was mentioned in several articles in the reviewed literature. TSN refers to mechanisms deployed in Ethernet standard that guarantee message delivery, provide real-time and redundancy features, and enable precise timing.

4.4 Summary of the Results

Table 4 represents the changes that Industry 4.0 brings to shop floor and integration between shop floor and MES based on the literature review. The second column lists most important performance metrics that are influenced by each change. Third column introduces some potential problems that might arise in the implementations.

Table 4 Summary of the changes

	Improved metrics	Potential problems
Smarter devices, sensors, and products	Lower latency, improved decision-making and real-time capabilities, better traceability, and flexibility	More complex and bigger devices with more processing capacity required
Middleware	Improved decisions through better real-time capabilities and data flow, adaptability and scalability, improved communication among heterogenous devices	If middleware requires internet connection, internet security and privacy threats and connectivity issues
Wireless sensor networks	More data that enables better decision making, improved traceability	Connectivity issues, probably requires redundancy to guarantee reliability
Fog computing	Better real-time capabilities and lower latency	Requires good architectural design to bring any advantages
Service Orientation	Better scalability, adaptability, and interoperability	In cloud implementations problems with internet
Wireless communication protocols	Required by other technologies. Improved flexibility and adaptability	Connection issues
IoT Messaging protocols	Lower processing capability required and simpler devices	Internet connection problems

The biggest problems seem to be caused by internet, including issues in the internet connection and security issues. If these can be overcome, other problems with implementations are minor and can be solved quite easily. More complex devices with more processing capacity are more expensive but advantages brought by smarter devices are likely to increase profits. Only the size of smarter sensors might be a problem in some cases. Fog computing requires a lot from design but if planned properly, considering scalability, adaptability, and the future needs of the system, it is likely to bring benefits.

Advantages and improved metrics caused by listed technologies and implementations enable increased profitability. Some of the metrics improve the lead time and efficiency, such as better real-time capabilities and lower latency. Adaptability, interoperability, and scalability enable manufacturing of more customized products, and improved traceability enables better quality and process control. It might also help to recognize problems in processes earlier.

5. PROTOTYPE DESIGN

Based on the literature review and company needs, it was decided that the practical part of the thesis handles integration using OPC UA over MQTT. Since OPC UA is widely used in pulp and paper MES projects, and MQTT seemed to be quite popular in Industry 4.0 literature, this implementation seemed most relevant. The goal of this chapter is to introduce the results of requirements analysis and design phase before the integration implementation.

5.1 Requirement analysis

Chapter 3.2 presented the methodology for requirement analysis which is done based on the intentional usage of solution and the operating environment. MQTT will likely be used mostly “from bottom to top”, i.e., from end devices to other systems such as MES, and for parallel communication, to deliver e.g., sensor data. As the goal of the practical part of the thesis was only to implement a prototype, operating environment is not critical aspect of the requirements. When implementing a solution for real paper and pulp production, several things, such as vibration and heat, should be taken into consideration during requirements analysis. However, as the prototype mirrors the real requirements of entrepreneurial environment, the communication should be secure. For company purposes it was also defined, that pure MQTT data from the broker could be subscribed also without OPC UA. Hence the requirements for the prototype are as represented in Table 5.

Table 5 Requirements

1	Communication	- Simple data can be sent from end device to other system and other end devices using OPC UA over MQTT - Data from the broker can be subscribed also without OPC UA
2	Security	- Communication through broker is secured so that the data can't be accessed or modified by non-authorized users

Figure 13 represents these requirements.

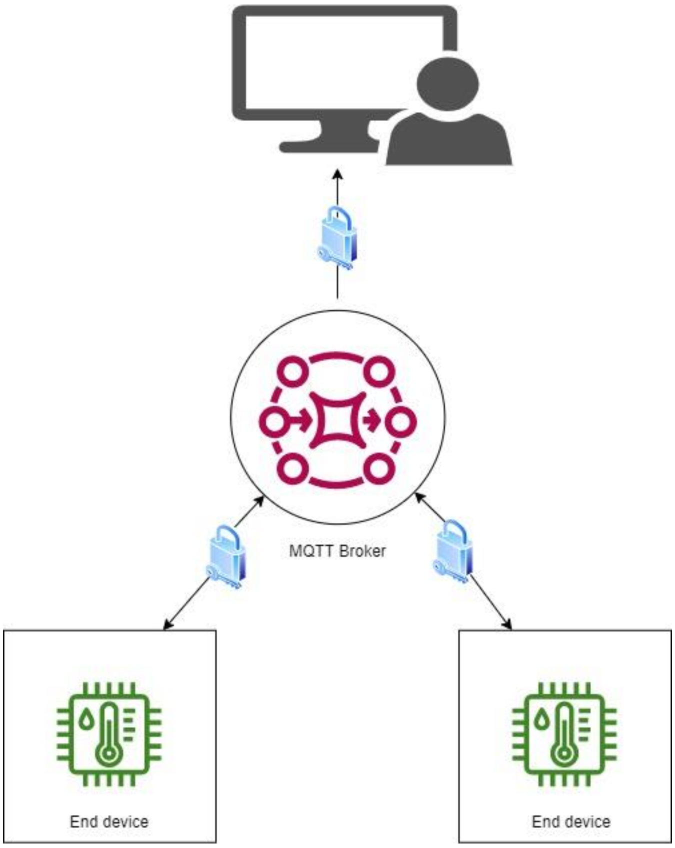


Figure 13 Requirements for the prototype

After the requirement analysis the prototype can be designed. Design phase consists of physical and logical design.

5.2 Physical Design

The goal of the physical design is to define used hardware and protocols. Raspberry Pi 3 model B+ is used as an end device, and it measures temperature and humidity with a simple DHT11 sensor. Both the Raspberry Pi and DHT11 sensor are easily available, and a lot of related source code can be found, and thus they were chosen to be used in the prototype. MQTT broker is installed on laptop with Linux Ubuntu operating system (OS), and as the top-level system works a laptop with Windows OS. OPC UA PubSub over MQTT is used as the communication protocol in the prototype and both Ethernet and Wi-Fi are used as link layer protocols. The code that subscribes topics from the MQTT broker without OPC UA is implemented on Red Hat Enterprise Linux virtual server.

5.3 Logical Design

The goal of the logical design is to define high-level abstract representation of processes and entities in the system. Figure 14 represents the class diagram of the system.

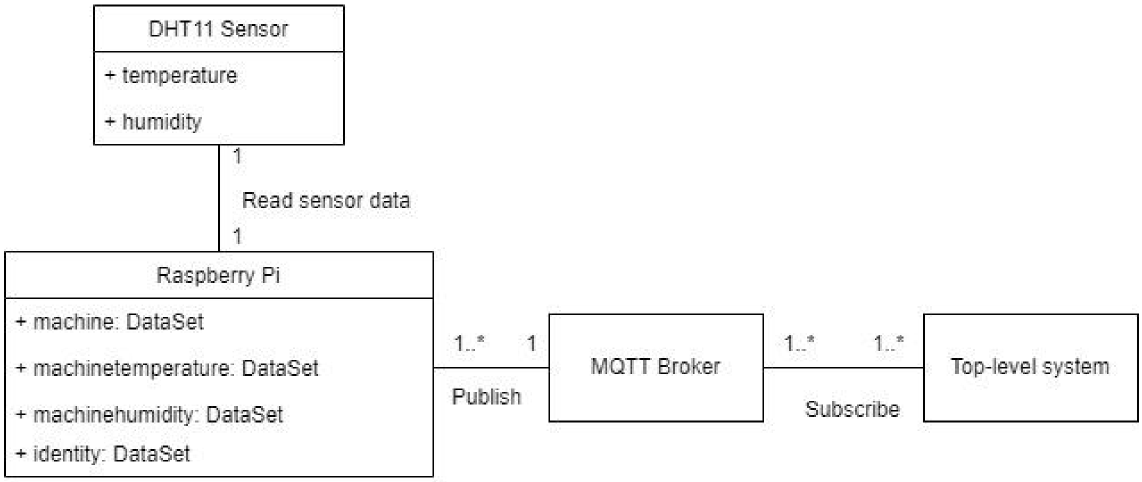


Figure 14 Class Diagram

The process is as follows: Raspberry Pi reads sensor data cyclically from DHT11 sensor. There are four different datasets that are sent to MQTT Broker: machinetemperature, machinehumidity, machine, which includes both the temperature and humidity, and identity, which includes information of the end device. These datasets are published to broker. Top-level system, in this case laptop with Windows OS, subscribes the data from MQTT broker.

6. IMPLEMENTATION

Prototype implementation is based on the requirements analysis and design phase discussed in Chapter 5. The goal of this chapter is to present the implementation of the prototype, and it is divided into four parts, first of which introduces more thoroughly the implementation of the connection between DHT11 humidity and temperature sensor and Raspberry Pi. The second subchapter represents the MQTT broker implementation, and in the third subchapter the main aspects of the code implementation are introduced. The last chapter introduces the testing phase of the prototype.

6.1 DHT11 Sensor and Raspberry Pi

In the prototype the sensor data is acquired with DHT11 sensor which measures temperature and humidity. Figure 15 shows how the sensor is connected to the Raspberry Pi 3 model B+.

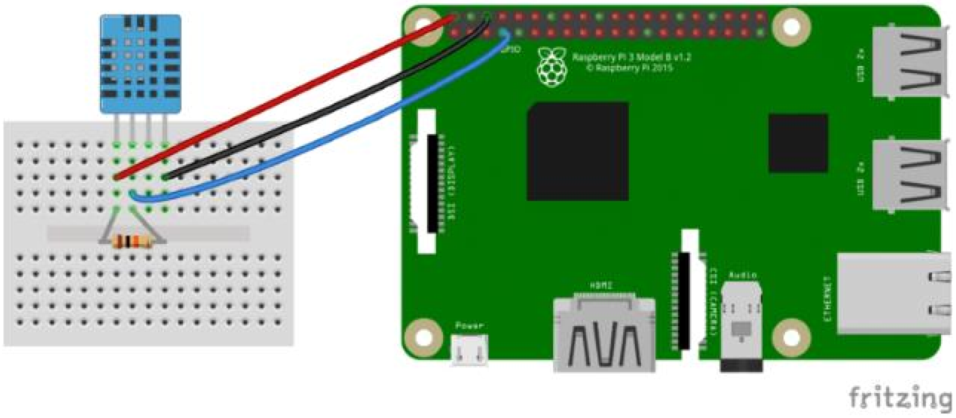


Figure 15 DHT11 connected to Raspberry Pi [86]

As the main implementation of the code is done with C# programming language, C# is also used to get the data from the DHT11 sensor. The implementation requires System.Device.GPIO and Iot.Device.Binding packages which provide tools for communication between microcontroller and different IoT devices.

6.2 MQTT Broker

There are several MQTT Brokers available, such as Mosquitto, RabbitMQ, EMQ, Azure IoT Hub, and HiveMQ. MQTT brokers can be either managed or self-hosted. To use a self-hosted broker the user needs to install the broker on their own server, unlike in the case of managed broker which can be used through managed broker services [87]. Mosquitto broker, installed on a laptop with Linux OS, is used in this prototype, as it is one of the most often-used open-source MQTT brokers and thus, there is a lot of related documentation available. It offers opportunities both for self-hosted and managed broker implementations, first of which was more convenient for the prototype.

Mosquitto broker also offers several different options to improve communication security, and thus the requirement of the secure communication is easy to implement. In the prototype, Mosquitto broker requires pre-defined usernames and passwords both from the publisher and the subscriber. Thus, the data can't be accessed or modified by external users. Mosquitto broker could also be used for TLS communication, where TLS security layer encrypts MQTT messages before transmission. Broker could also require SSL certificate for client authentication, and Certificate Revocation List (CRL) could be used for managing devices that no longer should have access to the broker. In addition to these, access control list and client-IDs could also be used for secure communication. [88]

6.3 Code Implementation

OPC foundation has published OPC UA IIoT Starter Kit [89] which is C# based GitHub repository for IoT communication based on MQTT and OPC UA PubSub using JSON (JavaScript Object Notation). This code was used as a basis for the code for implementing communication using OPC UA over MQTT. However, it should be noted that as mentioned in [89], the code is "not intended for immediate inclusion in a commercial product". It is covered under the MIT license, which grants a permission to, inter alia, use, copy, and sell copies of the software without limitations [90].

OPC UA IIoT StarterKit had quite comprehensive instructions to get started with the software, however, instead of using .NET 5.0 and Visual Studio 2019, .NET 6.0 and Visual Studio 2022 were used, as some methods used for acquiring data from DHT11 required using .NET 6.0. Hence, .NET 6.0 had to be set as a target framework in the project.

In chapter 6.3.3 it is explained how the data is subscribed from the MQTT broker without OPC UA. This code is based on mosquitto.h library and implemented with C programming language.

6.3.1 Creating and Modifying Datasets

The main problem in modifying the source code was to modify datasets that were published to MQTT broker. Originally, in the source code there was implementation for fictional gate that opens and closes. Published dataset included two variables, the state of the gate and counter that told the amount gate had opened or closed. The code also included identity dataset that was also used in the prototype. The goal was to implement three datasets, one of which includes the humidity, second of which includes the temperature, and third of which includes both the variables.

Before defining and creating the datasets, required variables needed to be defined. This was done in files in MqttAgent/config/sources folder. Datasets were defined in JSON files in MqttAgent/config/datasets. They both follow the guidelines of OPC UA Online Reference [91] Part 6, Mappings, in which chapter 5.4 defines how to construct messages using OPC UA JSON encoding, and Part 14 PubSub. In Figure 16, there’s an example of the structure of the JSON dataset file.

```
{
  "Name": "machinehumidity",
  "DataSetClassId": "15254774-8145-4DF7-92B5-789E5DCA9A0D",
  "Fields": [
    {
      "Name": "humidity",
      "Description": {
        "Text": "humidity"
      },
      "BuiltInType": 7,
      "DataSetFieldId": "e275f1ce-02eb-4fb7-af9c-e277711831db"
    }
  ],
  "ConfigurationVersion": {
    "MajorVersion": 675820800,
    "MinorVersion": 675820800
  }
}
```

Figure 16 OPC UA JSON Message encoding

The first “Name” field refers to the name of the dataset. DataSetClassId refers to universal DataSetClass, which is related to the metadata of the message. “Fields” is for the actual information that will be sent when messages are published, in this case for the information about the measured humidity. In this case, the name, description text, built-in type, and dataset field id are given for the field. Built-in type is the type of the variable.

All the built-in types and their IDs are defined in OPC UA Online Reference [91] in chapter 5.1.2. In this case the built-in type is 7 which refers to type UInt32. Dataset field id is a unique identifier for the field in the dataset.

Datasets were created in MqttConfigCreator/Program.cs file, where items included in the dataset are defined and created. For each item some parameters can or must be defined, such as name, attribute id, sampling interval, substitute value and possible metadata properties. For each dataset there is a method created which takes as a parameter the path to the related dataset configuration in MqttAgent/config/ folder.

To get the actual sensor data for variables, new C# file was created in the MqttAgent/Server folder. In this file, necessary methods were built using System.Device.GPIO and Iot.Device.Binding packages to get the data from DHT11 sensor. In this file two methods were created, both for humidity and temperature, to access the data. These methods are called in MqttAgent/Server/GPIOGateMonitor.cs file, where the values are given for variables.

6.3.2 Publisher and Subscriber Connections

In addition to datasets, publisher and subscriber connections needed to be modified. These connection configurations were in the publisher-connection.json and subscriber-connection.json files in the MqttAgent/config folder. The structure of these files is defined in the OPC UA Online Reference [91] in chapter 6.2.6.5.1. Figure 17 represents an example structure.

```
{
  "Name": "localhost",
  "Enabled": true,
  "PublisherId": {
    "Type": 12,
    "Body": "[ApplicationName]"
  },
  "TransportProfileUri": "http://opcfoundation.org/UA-Profile/Transport/pubsub-mqtt-json",
  "Address": {
    "TypeId": { "Id": 15510 },
    "Body": { "Url": "mqtt://localhost:1883/" }
  },
  "ReaderGroups": [
    {
      "Name": "machinehumidity",
      "Enabled": true,
      "SecurityMode": 1,
      "SecurityGroupId": "",
      "MaxNetworkMessageSize": 0,
      "DataSetReaders": [
        {
          "Name": "machinehumidity",
          "Enabled": true,
          "WriterGroupId": 1,
          "DataSetWriterId": 2,
          "DataSetMetaData": {
            "Name": "machine",
            "ConfigurationVersion": {
              "MajorVersion": 675820800,
              "MinorVersion": 675820800
            }
          },
          "DataSetFieldContentMask": 32,
          "MessageReceiveTimeout": 0,
          "KeyFrameCount": 10,
          "SecurityMode": 1,
          "SecurityGroupId": "",
          "TransportSettings": {
            "TypeId": { "Id": 15670 },
            "Body": {
              "QueueName": "opcua/[PublisherId]/machinehumidity",
              "RequestedDeliveryGuarantee": 2,
              "MetaDataQueueName": "opcua/[PublisherId]/machinehumidity/$Metadata"
            }
          },
          "MessageSettings": {
            "TypeId": { "Id": 15665 },
            "Body": {
              "NetworkMessageContentMask": 4,
              "DataSetMessageContentMask": 0
            }
          }
        }
      ]
    }
  ]
}
```

Figure 17 Structure of subscriber connection configuration file

The most relevant fields for the connection are TransportProfileUri and Address fields. The TransportProfileUri is for defining the used transport protocol and message mappings, in this case PubSub and MQTT for transport protocols and JSON as encoding. Address defines the address of the subscriber. TypeId in the address defines the datatype. In this case id is 15510 and it refers to NetworkAddressUrlDataType. In addition to these fields, Name, Enabled, and PublisherId fields are also same in publisher connection configuration. If needed, OPC UA PubSub enables defining additional connection properties and transport settings, as well, on top of already discussed configuration parameters.

The most important aspect from the point of this thesis, was to modify reader groups for the subscriber connection and writer groups for the publisher connection. The structure of the reader group and the dataset reader can be seen in Figure 17. Their structures are defined in OPC UA Online Reference in chapters 6.2.7.2.1 and 6.2.8.11.1 respectively. OPC UA PubSub enables defining security mode for the messages, and the value can be either 1, 2 or 3, meaning no security, signed but not encrypted messages, or signed and encrypted messages, respectively. If security mode has value 2 or 3, security group id must be given as it provides necessary information for securing the messages.

One reader group can include several dataset readers. WriterGroupId defines the identifier for the writer group, and DataSetWriterId defines the dataset selected in the publisher for the dataset reader. DataSetMetaData provides necessary information for decoding dataset messages. It has parameters for configuration versions which must be same as in corresponding dataset file (See Figure 16). DataSetFieldContentMask defines what information is sent in addition to the field value. It depends on the defined six-bit binary number, where each bit represents a flag, as defined in OPC UA Online Reference [91] chapter 6.2.3.2. In the example structure for example, the field value is 32 which transformed into binary form is 100000. This signifies “raw data” which means that the values of the dataset are encoded as structure and other flags related to the field are ignored. OPC UA PubSub enables defining maximum acceptable time between two dataset messages, MessageReceiveTimeout, to help acknowledging if messages aren’t received anymore or as often as required.

In OPC UA PubSub either delta frame or key frame dataset messages can be published. The delta frame messages contain only the changed values, unlike key frame messages that contain all the values. KeyFrameCount defines the maximum number of times the PublishingInterval expires before the key frame message is sent. Meanwhile only delta frame messages are sent.

Both the MessageSettings and the TransportSettings have a similar structure. As with the address, their datatypes are defined with id numbers, 15665 referring to JsonDataSetReaderMessageDataType and 15670 referring to BrokerDataSetReaderTransportDataType. In the body of the transport settings, QueueName refers to the queue in the broker where dataset reader can get the specific dataset. RequestedDeliveryGuarantee refers to the Quality of Service (QoS), which is used to manage network resources by prioritizing data on the network. In OPC UA PubSub there are five QoS levels. They are represented in Table 6.

Table 6 OPC UA PubSub QoS levels

ID	Value	Description
0	Not specified	The value of the parent object is used since the value is not specified
1	Best effort	Best effort is done, but in the worst-case data loss or duplication are possible
2	At least once	Guaranteed deliver at least once, but duplication is possible
3	At most once	Message is sent once, but in case of a lost it is not re-sent
4	Exactly once	Message is guaranteed to be sent and received exactly once

Metadata of the messages is also published to the broker. For this, MetaDataQueueName is required. NetworkMessageContentMask and DataSetMessageContentMask are defined similarly to DataSetFieldContentMask, which was discussed earlier. Their definitions can be found in OPC UA Online Reference in chapters 6.3.2.1.1 and 6.3.2.2.1, respectively. NetworkMessageContentMask defines the header fields that will be included in the network messages, and DataSetMessageContentMask defines the flags for the content of the header in message.

```
{
          "Name": "localhost",
          "Enabled": true,
          "PublisherId": {
            "Type": 12,
            "Body": "[ApplicationName]"
          },
          "TransportProfileUri": "http://opcfoundation.org/UA-Profile/Transport/pubsub-mqtt-json",
          "Address": {
            "TypeId": { "Id": 15510 },
            "Body": { "Url": "mqtt://localhost:1883/" }
          },
          "WriterGroups": [
            {
              "Name": "machinehumidity",
              "Enabled": true,
              "SecurityMode": 1,
              "SecurityGroupId": "",
              "MaxNetworkMessageSize": 0,
              "WriterGroupId": 1,
              "PublishingInterval": 10000,
              "KeepAliveTime": 0,
              "Priority": 0,
              "TransportSettings": {
                "TypeId": { "Id": 15667 },
                "Body": { "RequestedDeliveryGuarantee": 0 }
              },
              "MessageSettings": {
                "TypeId": { "Id": 15657 },
                "Body": { "NetworkMessageContentMask": 4 }
              },
              "DataSetWriters": [
                {
                  "Name": "machinehumidity",
                  "Enabled": true,
                  "DataSetWriterId": 2,
                  "DataSetFieldContentMask": 32,
                  "KeyFrameCount": 10,
                  "DataSetName": "machinehumidity",
                  "TransportSettings": {
                    "TypeId": { "Id": 15669 },
                    "Body": {
                      "QueueName": "opcua/[ApplicationName]/machinehumidity",
                      "RequestedDeliveryGuarantee": 2,
                      "MetaDataQueueName": "opcua/[ApplicationName]/machinehumidity/$Metadata",
                      "MetaDataUpdateTime": 30000
                    }
                  },
                  "MessageSettings": {
                    "TypeId": { "Id": 15664 },
                    "Body": { "DataSetMessageContentMask": 0 }
                  }
                }
              ]
            }
          ]
        }
```

Figure 18 Example structure of the publisher connection configuration

As can be seen when comparing Figure 17, which shows an example of subscriber connection configuration, and Figure 18, which represents similarly publisher connection configuration, structures are very similar. A few additional connection parameters, PublishingInterval, KeepAliveTime, Priority, and MetaDataUpdateTime, are included in the publisher connection configuration structure. First of these defines in milliseconds the frequency for publishing. KeepAliveTime can be set to define in milliseconds the time until the publisher sends a keep alive message if dataset message wasn't sent. Between writer groups of one publisher, more important writer groups can be defined as having higher priority, and thus they are processed before other writer groups. The priority can have a value between 0-255, where 0 is the lowest and 255 the highest priority. MetaDataUpdateTime is used to define the interval at which the metadata of the dataset is sent to metadata queue.

6.3.3 Subscribing from the Broker without OPC UA

As one of the requirements was to subscribe pure MQTT data from the broker without OPC UA, this chapter introduces the code implementation for that purpose. The implementation was done using C programming language and mosquitto.h library [92]. The code is represented in Appendix A. When running the code, the name of a .conf file including necessary connection information is given as a parameter. Example of the .conf file is represented in Figure 19.

```
subscribed_topic opcua/mydevice/machine

broker_ip 127.0.0.1

password 4321

username laptop

file_location /home/iita/files/
```

Figure 19 Example of file including connection information for subscriber

The idea of the code is to subscribe a topic from the broker and create a text file for the received messages. The subscribed topic, broker IP, password, username, and location for the created text files are defined in a separate .conf file as represented in Figure 19. In the code represented in Appendix A, first the information in the .conf file is stored in a config struct. The two functions of mosquitto library, mosquitto_connect_callback_set and mosquitto_message_callback_set, are used to define functions that are called either when connection is established, on_connect, or when messages are received, on_message. In case of an error, on_connect prints an error message, otherwise subscription is started based on the information stored in the config struct. When messages are received, on_message creates a text file, the name of which is based on the date, such as 29_8_2022, if such file doesn’t yet exist, and stores the received data in the file.

6.4 Testing the Prototype

Before testing, the test plan must be created. In table 7, there is listed tests that must be passed to verify the fulfillment of all the requirements.

Table 7 Test plan

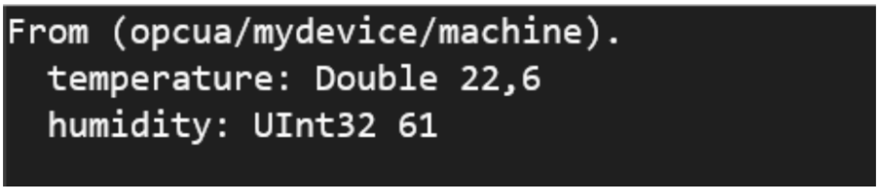
Test no.	
1	"machine" dataset can be published and subscribed
2	"machinetemperature" dataset can be published and subscribed
3	"machinehumidity" dataset can be published and subscribed
4	no access to data without valid username and password

The laptop with Windows OS was used as well as a development machine for modifying the code and it was connected to Raspberry Pi with SSH (Secure Shell) connection. As the code was built in the Visual Studio, the program was compiled for the Raspberry Pi, and the binaries were copied to the Raspberry Pi.

The program was run both on the Raspberry Pi and the laptop, as the Raspberry Pi worked as a publisher and the laptop worked as a subscriber. MQTT broker was running at the same time on the laptop with Linux environment. The program was started with the following commands:

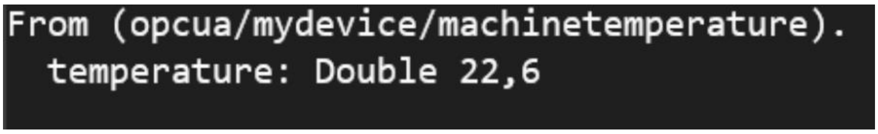
```
dotnet MqttAgent.dll publish -P=password -u=username
dotnet MqttAgent.dll subscribe -P=password -u=username -g=dataset
```

In this prototype phase passwords and usernames were given as parameters when running the program. In a real case scenario, this should be done in more secure manner. Parameter “-g” signifies the dataset that is subscribed. In the prototype this could be either machine, machinehumidity, or machinetemperature. Also, other parameters, such as broker address could be given with “-b” parameter, if the broker is not the same as defined in the original program. The messages received are represented in Figures 20, 21, and 22.



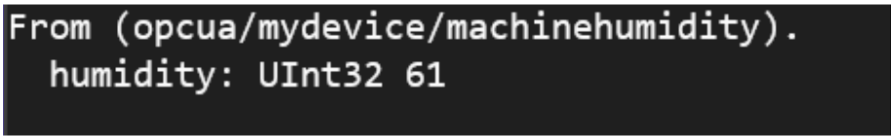
```
From (opcua/mydevice/machine).
  temperature: Double 22,6
  humidity: UInt32 61
```

Figure 20 Received message after subscription to "machine" dataset



```
From (opcua/mydevice/machinetemperature).
  temperature: Double 22,6
```

Figure 21 Received message after subscription to "machinetemperature" dataset



```
From (opcua/mydevice/machinehumidity).
  humidity: UInt32 61
```

Figure 22 Received message after subscription to "machinehumidity" dataset

As can be seen in Figures 19, 20, and 21, each message contains the information about the source of the message, the name of the parameter, the type of the parameter and the parameter value. In the code it was defined that each message will be published in 3 second time intervals, and this was also the time interval in which new message appeared on the screen. Thus, it can be concluded that test 1, 2, and 3 succeeded as expected. In addition to these datasets, there is also “identity” dataset that could be subscribed. It was also included in the original source code, and it wasn’t modified for the prototype.

The fourth test in the test plan was to ensure the data can’t be accessed without valid username and password. Both the publishing and subscribing were tested without valid user credentials, one at a time. Published messages were never received in the broker without password and username. When publishing was done with right credentials, subscriber couldn’t get the data without username and password. Thus, it can be concluded that also the fourth test was successful.

Same tests were also conducted with the code running on Red Hat Enterprise Linux based on receiving messages with MQTT without OPC UA. All the datasets were subscribed one at a time, and text files were created as defined with correct name and correct location.

7. RESULTS AND DISCUSSION

The goal of this thesis was to research the impact of Industry 4.0 on shop floor and shop floor devices in pulp and paper industry, and how these changes affect integrations between MES and shop floor devices. Based on the literature review, Industry 4.0 compliant prototype was built as an example of new type of integration between MES and shop floor. The purpose of this chapter is to summarize the most remarkable results of the thesis and discuss the significance of these findings. First, results of the literature review are discussed and after this, results of the prototype implementation are introduced.

As Industry 4.0 doesn't have an official definition, this thesis handled the topic from the perspective of four design principles, which were interconnection, information transparency, decentralized decisions, and technical assistance, and from the perspective of the main key technologies, which were IoT, big data and analytics, and cloud computing. In the literature review, impact of Industry 4.0 was divided into three categories: impact on the architecture, impact on the shop floor and impact on the integrations between MES and shop floor. The amount of research done on the impact of Industry 4.0 in especially pulp and paper industry was remarkably little, and as was noted in [35], there are gaps in research related to this topic. Thus, to be able to answer the research questions, it was inevitable that also other industry fields, especially other fields of process industry, had to be taken into the scope of the literature review. It was assumed that the situation in other industry fields also reflects the situation in pulp and paper industry.

The most remarkable architectural change according to the literature review seemed to be decentralization. Decisions are done in a decentralized manner, i.e., one system, such as MES, is not responsible for all the decision. Instead, decisions are made closer to the end devices, or end devices or products make some decisions regarding their own functionalities by themselves. To enable decisions made by devices or products, smarter devices and products are required, and as found out in the literature review, this "smartness" was one of the key aspects for changes in the shop floor. Smartness requires more data and new ways to collect, analyze, and deliver it. New sensor technologies, such as RFID sensors, and wireless sensor networks were thus other important part of Industry 4.0 compliant shop floor. Fog and cloud computing have also very important part in enabling decentralized decisions. The most remarkable changes in the integrations between MES and shop floor were service orientation, new wireless communication technologies, and IoT communication protocols.

As was already discussed in the introduction, investing in Industry 4.0 is crucial for companies to enable future growth. This thesis summarizes the key changes that will likely to be seen more and more on the factory shop floor when moving towards Industry 4.0 compliant smart factories, and how they will affect the integrations between shop floor and MES. With this information companies working on this field gain valuable insight on technologies that should be focused on to develop their solutions to meet the requirements of future factories. This increases company's competitive advantage on the markets and help them to serve customers also in the future.

After the literature review, Industry 4.0 compliant prototype was designed and implemented. The goal of the prototype was to test OPC UA PubSub over MQTT communication protocol to deliver simple sensor data from publisher to subscriber. As the prototype was ready, communication from the publisher to subscriber seemed to work reliably and work as designed. Biggest problems were related to the sensor reading the data, as there were some problems to acquire the data from the sensor. This however seemed to be due to poor hardware and when implemented in real environment the quality of the components would likely be remarkably better. The program itself seemed to work as anticipated. The results of the prototype implementation suggest that using OPC UA PubSub over MQTT seems relatively easy to implement, although real factory environment and commercial purposes set their own requirements for implementation. OPC UA Online Reference [91] provides comprehensive documentation to support implementation, although relatively little open-source code can be found. Similar solutions as the implemented prototype could be tested in pulp and paper industry for example in monitoring the dryer section or recovery boiler, where real-time requirements are not as strict as in other parts of the production and where a lot of sensor data is required and thus many-to-many approach would offer the most benefit. It should be noted, however, that although OPC UA PubSub over MQTT enables easy many-to-many approach to communication, this might not be the most suitable communication pattern to every occasion. As already stated earlier, pulp and paper industry have its own specific requirements for communication solutions that should be taken into consideration when thinking the adequacy of new communication protocols. Short-term testing doesn't necessarily provide enough information about the suitability of technology in factories, where the equipment is usually required to work almost non-stop for decades, and where real-time requirements are in some cases extremely strict. These aspects should be considered also with the other technologies mentioned in the results of literature review.

8. CONCLUSIONS

The purpose of this thesis was to answer the following research questions:

Q1: What research has been done on the impact of Industry 4.0 on shop floor and shop floor devices in pulp and paper industry?

Q2: What qualitative and quantitative metrics are affected by Industry 4.0 implementations in the shop floor?

Q3: How changes in shop floor affect integrations between MES and shop floor devices?

It was evident that the amount of research done regarding the impact of Industry 4.0 on shop floor devices particularly in pulp and paper industry was little, and there were clearly gaps especially in topics that would have been most beneficial for the purposes of this thesis. However, it was assumed that impact of Industry 4.0 on the other industrial fields reflect the impact also on the pulp and paper industry. These changes that Industry 4.0 brings to industries were more researched and their impact on qualitative and quantitative metrics were discussed in Chapter 4. The most important effects recognized were improved decision-making, better real-time capabilities, adaptability, traceability, and scalability. Most of the potential problems were related to the internet connection that dispose the system for new security and privacy issues and the problems with poor internet connection.

One of the main results of this thesis was the clarification of the impact of Industry 4.0 implementations on the integrations between MES and shop floor, as this brings value for companies working on this field. The key changes in the integrations handle service orientated approach to communication and new wireless technologies as well as new IoT communication protocols. Prototype to test new IoT messaging protocol, OPC UA PubSub over MQTT, was implemented to show its potential for industrial purposes. It was however stated that more testing on real production environment should be completed prior to making final conclusions on its suitability for the purposes of pulp and paper industry.

This thesis handled quite shortly and limitedly each separate change related to impact of Industry 4.0 on shop floor and shop floor integrations, and their effect on the whole production system. It would be beneficial to research more thoroughly these changes, such as WSNs or middleware implementations, separately and see their potential for different industrial fields such as pulp and paper industry. The implemented prototype was a good starting point for future implementations for real production environment, and potential of OPC UA PubSub should be further researched, as so far little research in industrial environment has been conducted.

REFERENCES

[1] Rüßmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engel, P., & Harnisch, M. (2015). *Industry 4.0: The future of productivity and growth in manufacturing industries*. The Boston Consulting Group. https://www.bcg.com/publications/2015/engineered_products_project_business_industry_4_future_productivity_growth_manufacturing_industries

[2] Shojaeinasab, A., Charter, T., Jalayer, M., Khadivi, M., Ogunfowora, O., Raiyani, N., Yaghoubi, M., & Najjaran, H. (2022). Intelligent manufacturing execution systems: A systematic review. *Journal of Manufacturing Systems*, *62*, 503–522. https://doi.org/10.1016/j.jmsy.2022.01.004

[3] Pfeiffer, S. (2017). The Vision of "Industrie 4.0" in the Making—a Case of Future Told, Tamed, and Traded. *Nanoethics*, *11*(1), 107–121. https://doi.org/10.1007/s11569-016-0280-3

[4] Almada-Lobo, F. (2015). The Industry 4.0 revolution and the future of Manufacturing Execution Systems (MES). *Journal of Innovation Management*, *3*(4), 16–21. https://doi.org/10.24840/2183-0606_003.004_0003

[5] Mantravadi, S. & Møller, C. (2019). An overview of next-generation manufacturing execution systems: How important is MES for industry 4.0? *Procedia Manufacturing*, *30*, 588–595. https://doi.org/10.1016/j.promfg.2019.02.083

[6] Jaskó, S., Skrop, A., Holczinger, T., Chován, T., & Abonyi, J. (2020). Development of manufacturing execution systems in accordance with Industry 4.0 requirements: A review of standard- and ontology-based methodologies and tools. Computers in Industry, 123, 103300–. https://doi.org/10.1016/j.compind.2020.103300

[7] Zippel, S. (2018). Process Industry 4.0: Transforming the process industry with Industry 4.0. *InTech*, *65*(7). https://www.isa.org/intech-home/2018/november-december/features/process-industry-4-0

[8] MPI Group. (2020). MPI 2020 Industry 4.0 Study Full Executive Summary. Industry 4.0 Drives Productivity – And Profitability – For Manufacturers. https://mpi-group.com/mpi-industry-4-0-study/

[9] Chen, H-L. (2021). Impact of Industry 4.0 on Corporate Financial Performance: A Moderated Mediation Model. *Sustainability (Basel, Switzerland)*, *13*(11), 6069–. https://doi.org/10.3390/su13116069

[10] Bauer, W. & Horváth, P. (2015). Industrie 4.0 - Volkswirtschaftliches Potenzial für Deutschland. *Controlling*, *27*(8-9), 515–517. https://doi.org/10.15358/0935-0381-2015-8-9-515

[11] Hermann, M., Pentek, T., & Otto, B. (2016). Design Principles for Industrie 4.0 Scenarios. *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 3928–3937. https://doi.org/10.1109/HICSS.2016.488

[12] Buyya, R. & Dastjerdi, A. V. (2016). *Internet of things: principles and paradigms* (1st edition). Morgan Kaufmann.

[13] Hammons, R. L. & Kovac, R. (2019). *Fundamentals of internet of things for non-engineers*. CRC Press.

[14] Jacoby, M. & Usländer, T. (2020). Digital twin and internet of things-Current standards landscape. *Applied Sciences*, *10*(18), 6519–. https://doi.org/10.3390/APP10186519

[15] Misra, S., Mukherjee, A., & Roy, C. (2021). *Introduction to industrial internet of things and industry 4.0*. CRC Press.

[16] Marques, M., Agostinho, C., Zacharewicz, G., & Jardim-Gonçalves, R. (2017). Decentralized decision support for intelligent manufacturing in Industry 4.0. *Journal of Ambient Intelligence and Smart Environments*, *9*(3), 299–313. https://doi.org/10.3233/AIS-170436

[17] Alcácer, V. & Cruz-Machado, V. (2019). Scanning the Industry 4.0: A Literature Review on Technologies for Manufacturing Systems. *Engineering Science and Technology, an International Journal*, *22*(3), 899–919. https://doi.org/10.1016/j.jestch.2019.01.006

[18] Bai, C., Dallasega, P., Orzes, G., & Sarkis, J. (2020). Industry 4.0 technologies assessment: A sustainability perspective. *International Journal of Production Economics*, *229*, 107776–. https://doi.org/10.1016/j.ijpe.2020.107776

[19] Ustundag, A. & Cevikcan, E. (2018). *Industry 4.0: managing the digital transformation*. Springer. https://doi.org/10.1007/978-3-319-57870-5

[20] Ralph, B. & Stockinger, M. (2020). Digitalization and digital transformation in metal forming: Key Technologies, Challenges and Current Developments of Industry 4.0 Applications. https://www.researchgate.net/publication/343826977_DIGITALIZATION_AND_DIGITAL_TRANSFORMATION_IN_METAL_FORMING_KEY_TECHNOLOGIES_CHALLENGES_AND_CURRENT_DEVELOPMENTS_OF_INDUSTRY_40_APPLICATIONS

[21] Ashton, K. That ‘internet of things’ thing. *RFiD J*. 2009;22(7):97–114. http://www.itrco.jp/libraries/RFIDjournal-That%20Internet%20of%20Things%20Thing.pdf

[22] What is Industry 4.0—the Industrial Internet of Things (IIoT)? Epicor. https://www.epicor.com/en/blog/learn/what-is-industry-4-0/

[23] Industrial IoT, I4.0, and Digital Transformation. ARC Advisory Group. https://www.arcweb.com/technologies/industrial-internet-of-things

[24] Adolphs, P., Bedenbender, H., Dirzus, D., Ehlich, M., Epple, U., Hankel, M., Heidel, R., Hoffmeister, M., Huhle, H., Kärcher, B., Koziolek, H., Pichler, R., Pollmeier, S., Schewe, F., Walter, A., Waser, B., & Wollschlaeger, M. (2015). Reference Architecture Model Industrie 4.0 (RAMI4.0). https://www.zvei.org/fileadmin/user_upload/Themen/Industrie_4.0/Das_Referenzarchitekturmodell_RAMI_4.0_und_die_Industrie_4.0-Komponente/pdf/5305_Publikation_GMA_Status_Report_ZVEI_Reference_Architecture_Model.pdf

[25] Industrial Internet Reference Architecture. (2015). Industrial Internet Consortium. https://www.iiconsortium.org/stay-informed-IIRA/

[26] Humayed, A., Lin, J., Li, F., & Luo, B. (2017). Cyber-Physical Systems Security-A Survey. *IEEE Internet of Things Journal*, *4*(6), 1802–1831. https://doi.org/10.1109/JIOT.2017.2703172

[27] Radziwon, A., Bilberg, A., Bogers, M., & Madsen, E. S. (2014). The Smart Factory: Exploring Adaptive and Flexible Manufacturing Solutions. *Procedia Engineering*, *69*, 1184–1190. https://doi.org/10.1016/j.proeng.2014.03.108

[28] Smart Manufacturing and Smart Industry in Context. i-Scoop. https://www.i-scoop.eu/industry-4-0/manufacturing-industry/

[29] Fraser, J. (1997). MES Explained: A High Level Vision. MESA International. http://alvarestech.com/temp/smar/www.delt.ufmg.br/seixas/PaginaII/Download/DownloadFiles/pap6.pdf

[30] Mantravadi, S., Møller, C., LI, C., & Schnyder, R. (2022). Design choices for next-generation IIoT-connected MES/MOM: An empirical study on smart factories. *Robotics and Computer-Integrated Manufacturing*, *73*, 102225–. https://doi.org/10.1016/j.rcim.2021.102225

[31] Merriam-Webster dictionary – Shop floor. https://www.merriam-webster.com/dictionary/shop%20floor

[32] Brandl, D. & Johnsson, C. (2021). Beyond the Pyramid: Using ISA95 for Industry 4.0 and Smart Manufacturing. https://www.isa.org/intech-home/2021/october-2021/features/beyond-the-pyramid-using-isa95-for-industry-4-0-an

[33] Damm, M., Leitner S-H., & Mahnke, W. *OPC Unified Architecture*. 1. Aufl. Berlin, Heidelberg: Springer-Verlag, 2009.

[34] Unified Architecture. OPC Foundation. https://opcfoundation.org/about/opc-technologies/opc-ua/

[35] Telukdarie, A. & Kumar, A. (2018). An Industry 4.0 Research Review For The Pulp And Paper Industry. Proceedings of the International Conference on Industrial Engineering and Operations Management. http://ieomsociety.org/southafrica2018/papers/359.pdf

[36] Coito, T., Viegas, J. L., Martins, M. S. E., Cunha, M. M., Figueiredo, J., Vieira, S. M., & Sousa, J. M. C. (2019). A Novel Framework for Intelligent Automation. *IFAC PapersOnLine*, *52*(13), 1825–1830. https://doi.org/10.1016/j.ifacol.2019.11.501

[37] Coito, T., Viegas, J. L., Martins, M. S. E., Firme, B., Figueiredo, J., Vieira, S. M., & Sousa, J. M. D. C. (2021). The impact of intelligent automation in internal supply chains. *International Journal of Integrated Supply Management*, *14*(1), 1–27. https://doi.org/10.1504/IJISM.2021.113563

[38] Coito, T., Martins, M. S. E., Viegas, J. L., Firme, B., Figueiredo, J., Vieira, S. M., & Sousa, J. M. C. (2020). A Middleware Platform for Intelligent Automation: An Industrial Prototype Implementation. *Computers in Industry*, *123*, 103329–. https://doi.org/10.1016/j.compind.2020.103329

[39] Perry, M. J. (2016). *Evaluating and Choosing an IoT Platform*. O’Reilly Media, Inc.

[40] Tamboli, A. (2019). *Build Your Own IoT Platform: Develop a Fully Flexible and Scalable Internet of Things Platform in 24 Hours*. Apress L. P.

[41] Lueth, K. (2019). IoT Platform Companies Landscape 2019/2020: 620 IoT Platforms globally. IoT Analytics. https://iot-analytics.com/iot-platform-companies-landscape-2020/

[42] Arrais, R., Veiga, G., Ribeiro, T. T., Oliveira, D., Fernandes, R., Conceição, A. G. S., & Farias, P. C. M. A. (2019). Application of the Open Scalable Production System to Machine Tending of Additive Manufacturing Operations by a Mobile Manipulator. In *Progress in Artificial Intelligence* (pp. 345–356). Springer International Publishing. https://doi.org/10.1007/978-3-030-30244-3_29

[43] Santos, R., Toscano, C., & de Sousa, J. P. (2021). A new Simulation-Based Approach in the Design of Manufacturing Systems and Real-Time Decision Making. *IFAC PapersOnLine*, *54*(1), 282–287. https://doi.org/10.1016/j.ifacol.2021.08.033

[44] Castillo, A. C., Patsavellas, J., Salonitis, K., & Emmanouilidis, C. (2021). The productivity impact of the digitally connected 5 – layer stack in manufacturing enterprises. *Procedia CIRP*, *104*, 342–350. https://doi.org/10.1016/j.procir.2021.11.058

[45] Gregorio, M., Santos, R., Barros, C., & Silva, G. (2016). Problems in Adopting Middleware for IoT: A Survey. UBICOMM 2016: The Tenth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies.

[46] Silverio-Fernández, M., Renukappa, S., & Suresh, S. (2018). What is a smart device? - a conceptualisation within the paradigm of the internet of things. *Visualization in Engineering*, *6*(1), 1–10. https://doi.org/10.1186/s40327-018-0063-8

[47] Kalsoom, T., Ramzan, N., Ahmed, S., & Ur-Rehman, M. (2020). Advances in Sensor Technologies in the Era of Smart Factory and Industry 4.0. *Sensors (Basel, Switzerland)*, *20*(23), 6783–. https://doi.org/10.3390/s20236783

[48] Rahman, M., Kyprianidis, K., Avelin, A., & Dahlquist, E. (2017). An Approach for Feedforward Model Predictive Control for Pulp and Paper Applications: Challenges and the Way Forward. https://www.researchgate.net/publication/316515772_An_Approach_For_Feedforward_Model_Predictive_Control_For_Pulp_and_Paper_Applications_Challenges_And_The_Way_Forward

[49] EPIS. Technical Standards. https://epis.org/technical-standards

[50] GS1 RFID Identification of Pulp Products Guideline. (2020). https://www.gs1.org/standards/gs1-rfid-identification-pulp-products-guideline/current-standard#1-Introduction

[51] New Solution Release: Intelligent Pulp Solution. (2019). https://www.storaenso.com/en/newsroom/news/2019/11/new-solution-release-intelligent-pulp-solution

[52] Mesä Fibre. (2020). RFID Tracking system for pulp units improves traceability. https://www.metsafibre.com/en/media/Stories/Pages/RFID-tracking-system-for-pulp-units-improves-traceability.aspx

[53] SICK Sensor Intelligence. (2017). Paper per running meter: RFID technology ensures an accurate production sequence. https://www.sick.com/be/fr/paper-per-running-meter-rfid-technology-ensures-an-accurate-production-sequence/w/blog-rfid-technology-ensures-accurate-paper-production/

[54] Vijayalakshmi, S. R. & Muruganand, S. *(2018). Wireless Sensor Networks*. Mercury Learning & Information.

[55] Raposo, D., Rodrigues, A., Sinche, S., Silva, J. S., & Boavida, F. (2018). Industrial IoT Monitoring: Technologies and Architecture Proposal. *Sensors (Basel, Switzerland)*, *18*(10), 3568–. https://doi.org/10.3390/s18103568

[56] Ahlen, A., Akerberg, J., Eriksson, M., Isaksson, A. J., Iwaki, T., Johansson, K. H., Knorn, S., Lindh, T., & Sandberg, H. (2019). Toward Wireless Control in Industrial Process Automation: A Case Study at a Paper Mill. *IEEE Control Systems*, *39*(5), 36–57. https://doi.org/10.1109/MCS.2019.2925226

[57] Pallasch, C., Wein, S., Hoffmann, N., Obdenbusch, M., Buchner, T., Waltl, J., & Brecher, C. (2018). Edge Powered Industrial Control: Concept for Combining Cloud and Automation Technologies. *2018 IEEE International Conference on Edge Computing (EDGE)*, 130–134. https://doi.org/10.1109/EDGE.2018.00026

[58] Rocha, R. (2021). APIs REST, GraphQL or gRPC – Who wins this game? https://www.sensedia.com/post/apis-rest-graphql-or-grpc-who-wins-this-game

[59] Sauer, O. (2014). Developments and trends in shopfloor-related ICT systems. *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, 1352–1356. https://doi.org/10.1109/IEEM.2014.7058859

[60] Atanasov, S. (2013). An overview of wireless communication technologies used in wireless sensor networks. International Scientific Conference eRA-8. ISSN-

1791-1133. 11-18. https://www.researchgate.net/publication/262565993_An_Overview_of_Wireless_Communication_Technologies_Used_in_Wireless_Sensor_Networks

[61] Bluetooth Technology Overview. (2022). Bluetooth. https://www.bluetooth.com/learn-about-bluetooth/tech-overview/

[62] Different Wi-Fi Protocols and Data Rates. Intel. https://www.intel.com/content/www/us/en/support/articles/000005725/wireless/legacy-intel-wireless-products.html

[63] Digi XBee 3 ZigBee 3.0 Datasheet. https://www.digi.com/resources/library/data-sheets/ds_xbee-3-zigbee-3

[64] Petersen, S. & Carlsen, S. (2009). Performance evaluation of WirelessHART for factory automation. *2009 IEEE Conference on Emerging Technologies & Factory Automation*, 1–9. https://doi.org/10.1109/ETFA.2009.5346996

[65] WirelessHART. PhoenixContact. http://www.phoenixcontact.net/microsites/wireless-hart/895.htm

[66] Touati, F., Erdene-Ochir, O., Mehmood, W., Hassan, A., Mnaouer, A. B., Gaabab, B., Rasid, M. F. A., & Khriji, L. (2015). An Experimental Performance Evaluation and Compatibility Study of the Bluetooth Low Energy Based Platform for ECG Monitoring in WBANs. *International Journal of Distributed Sensor Networks*, *2015*(9), 645781–. https://doi.org/10.1155/2015/645781

[67] Ghayas, A. (2020) Mobile Data Speed With 2G, 3G, 4G And 5G Cellular Networks. Commsbrief. https://commsbrief.com/mobile-data-speed-with-2g-3g-4g-and-5g-cellular-networks/

[68] Minihold, R. (2011) Near Field Communication (NFC) Technology and Measurements. Rohde & Schwarz. https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_application/application_notes/1ma182/1MA182_5E_NFC_WHITE_PAPER.pdf

[69] Key Features of the Field Wireless System. vigilantplant. https://web-material3.yokogawa.com/fld-field-test-report-vol01.pdf

[70] ISA100 Wireless for Control Applications. (2016). https://isa100wci.org/en-US/Documents/Presentations/End-User-Conference-September-2016/ISA100-for-control-applications-Rotterdam.aspx

[71] Suthar, K. & He, Q. P. (2021). Multiclass moisture classification in woodchips using IIoT Wi-Fi and machine learning techniques. *Computers & Chemical Engineering*, *154*, 107445–. https://doi.org/10.1016/j.compchemeng.2021.107445

[72] Naik, N. (2017). Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. *2017 IEEE International Systems Engineering Symposium (ISSE)*, 1–7. https://doi.org/10.1109/SysEng.2017.8088251

[73] eProsima Fast DDS Documentation. https://fast-dds.docs.eprosima.com/en/latest/index.html

[74] About the Data Distribution Service Specification Version 1.0. (2004). https://www.omg.org/spec/DDS/1.0

[75] An Overview of XMPP. https://xmpp.org/about/technology-overview/

[76] Rullo, A., Bertino, E., & Sacca, D. (2018). PAST: Protocol-Adaptable Security Tool for Heterogeneous IoT Ecosystems. *2018 IEEE Conference on Dependable and Secure Computing (DSC)*, 1–8. https://doi.org/10.1109/DESEC.2018.8625143

[77] Bayou, L. (2018). *Assessment and enforcement of wireless sensor network-based SCADA systems security*. https://www.researchgate.net/publication/331587323_Assessment_and_enforcement_of_wireless_sensor_network-based_SCADA_systems_security

[78] Gangurde, P., Waware, S., & Sarwade, N. (2022). Simulation of TCP, UDP and SCTP with constant traffic for VOIP services. https://www.semanticscholar.org/paper/Simulation-of-TCP-%2C-UDP-and-SCTP-with-constant-for-Gangurde-Waware/8c6ba91e213016b25cb67767a8ac0321cc25360c

[79] Difference between UDP and RTP. (2020). GeeksforGeeks. https://www.geeksforgeeks.org/difference-between-udp-and-rtp/

[80] Shared Memory Protocol. https://docs.quanser.com/rcp/documentation/shmem_protocol.html

[81] Entzminger, R. A., Kay, J. A., & Peterson, C. A. “Analysis and Overview of Message Queueing Telemetry Transport (MQTT) as Applied to Forest Products Applications.” In *The Institute of Electrical and Electronics Engineers, Inc. (IEEE) Conference Proceedings*. Piscataway: The Institute of Electrical and Electronics Engineers, Inc. (IEEE), 2021.

[82] Zhang, Y., Li, Z., & Liu, Q. Design and Implementation of Remote Monitoring System for High Frequency Vibrator. DOI: 10.11980/j.issn.0254-508X.2018.12.010

[83] Picture this: Critical Data from Live Feed in the Recovery Boiler. (2020). Spectrum. ANDRITZ English Magazine of Pulp & Paper. No. 41. 2-2020. https://www.andritz.com/resource/blob/385942/5e70c0ebc1b82b19fa8bae2fcdbff9dc/andritz-spectrum41-en-web-data.pdf

[84] OPC 10000-14: OPC Unified Architecture. Part 14: PubSub. https://reference.opcfoundation.org/v104/Core/docs/Part14/#7.3.3

[85] Time Sensitive Networking and OPC UA (OPC UA FX). https://www.tttech-industrial.com/technologies/opc-ua-fx/

[86] Campbell, S. How to Set Up the DHT11 Humidity Sensor on the Raspberry Pi. Circuit Basics. https://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-the-raspberry-pi/

[87] MQTT Broker. Network Admin Guide. https://www.catchpoint.com/network-admin-guide/mqtt-broker

[88] IoT Security -Part 12 (MQTT Broke Security – 101). (2020). https://payatu.com/blog/dattatray/iot-security-%E2%80%93-part-12-mqtt-broker-security---101

[89] OPC UA IIoT StarterKit. OPC Foundation. GitHub. https://github.com/OPCFoundation/UA-IIoT-StarterKit

[90] MIT License. https://opcfoundation.org/license/mit.html

[91] OPC UA Online Reference. https://reference.opcfoundation.org/

[92] mosquitto.h Library Documentation. https://mosquitto.org/api/files/mosquitto-h.html

APPENDIX A

```
#include <stdio.h>
#include <stdlib.h>
#include <mosquitto.h>
#include <string.h>
#include <time.h>

#define MAX_CONFIG_VARIABLE_LEN 100
#define CONFIG_LINE_BUFFER_SIZE 100

struct configuration_struct {
         char subscribed_topic[MAX_CONFIG_VARIABLE_LEN];
         char broker_ip[MAX_CONFIG_VARIABLE_LEN];
         char password[MAX_CONFIG_VARIABLE_LEN];
         char username[MAX_CONFIG_VARIABLE_LEN];
         char file_location[MAX_CONFIG_VARIABLE_LEN];
};

FILE *data;
char data_location[MAX_CONFIG_VARIABLE_LEN];

struct configuration_struct read_conf_file(char* filename, struct configuration_struct config){
         FILE *f;
         char buffer[CONFIG_LINE_BUFFER_SIZE];

         if ((f=fopen(filename, "r")) == NULL) {
                  fprintf(stderr, "File can’t be opened\n");
                  exit(EXIT_FAILURE);
         }
         while(!feof(f)){
                  fgets(buffer, CONFIG_LINE_BUFFER_SIZE, f);
                  if (buffer[0] == '#' || strlen(buffer) < 2) {
                           continue;
                  }
                  if (strstr(buffer, "broker_ip ")){
                           char name[MAX_CONFIG_VARIABLE_LEN];
                           sscanf(buffer, "%s %s\n", name, config.broker_ip);
                  }
                  if (strstr(buffer, "subscribed_topic ")){
                           char [MAX_CONFIG_VARIABLE_LEN];
                           sscanf(buffer, "%s %s\n", name, config.subscribed_topic);
                  }
                  if (strstr(buffer, "password ")){
                           char name[MAX_CONFIG_VARIABLE_LEN];
                           sscanf(buffer, "%s %s\n", name, config.password);
                  }
                  if (strstr(buffer, "username ")){
                           char name[MAX_CONFIG_VARIABLE_LEN];
                           sscanf(buffer, "%s %s\n", name, config.username);
                  }
                  if (strstr(buffer, "file_location ")){
                           char name[MAX_CONFIG_VARIABLE_LEN];
                           sscanf(buffer, "%s %s\n", name, config.file_location);
                  }
         }
         return config;
}

void connected(struct mosquitto *msq, void *obj, int returned) {
         if(returned) {
                  printf("Connection failed. Error code: %d\n", returned);
                  exit(-1);
         }
}
```

```
void remove_char(char *s, char c){
	int i = 0;
	int j = 0;

	while (s[j]){
		if (s[j] != c){
			s[i++] = s[j];
		}
		j++;
	}
	s[i]=0;
}

char* handle_payload(char* payload){
	remove_char(payload, '{');
	remove_char(payload, '}');
	remove_char(payload, '"');
	return payload;
}

void message_rcv(struct mosquitto *msq, void *obj, const struct mosquitto_message *msg) {
	time_t now = time(NULL);
	char new_location[100];
	char date_string[100];
	char payload[100];
	struct tm *t = localtime(&now);
	strftime(date_string, sizeof(date_string), "%d_%m_%Y", t);
	strcpy(new_location, data_location);
	strncat(new_location, date_string, 10);
	strcpy(payload, msg->payload);
	strcpy(payload, handle_payload(payload));
	data = fopen(new_location, "a");

	fprintf(data, "%s\n", (char *) payload);
	fclose(data);
}

int main(int argc, char *argv[]) {
	struct configuration_struct config;

	config = read_conf_file(argv[1], config);
	const char* subscribed_topic = config.subscribed_topic;
	const char* broker_ip = config.broker_ip;
	const char* username = config.username;
	const char* password = config.password;
	const char* file_location = config.file_location;
	strcpy(data_location,file_location);

	int returned;
	idf=1;
	mosquitto_lib_init();

	struct mosquitto *msq;

	msq = mosquitto_new("subscriber", true, &idf);
	mosquitto_connect_callback_set(msq, connected);
	mosquitto_message_callback_set(msq, message_rcv);
	mosquitto_username_pw_set(msq, username, password);

	returned = mosquitto_connect(msq, broker_ip, 1883, 10);
	mosquitto_subscribe(msq, NULL, subscribed_topic, 0);
	if(returned) {
		printf("Broker connection failed. Error code %d\n", returned);
		return -1;
	}

	mosquitto_loop_start(msq);
	while(true){
	}

	mosquitto_loop_stop(msq, true);
	mosquitto_disconnect(msq);
	mosquitto_destroy(msq);
	mosquitto_lib_cleanup();

	return 0;
}
```