Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Telecommunications and Media Informatics

Asset Tracking in Digital Industry

Scientific Students' Associations Conference Paper

Author
Attila Ernő Frankó

Supervisor
Pál Varga, PhD

October 29, 2018

Contents

Kivonat **4**

Abstract **5**

Introduction **6**

1 Related Works **8**

1.1 Logistics and Value Chains 8

1.1.1 Stakeholders and Systems Across Supply Chains 9

1.2 Digitization and IoT 10

1.2.1 Automation IoT 11

1.2.2 Industrial IoT 12

1.3 The Arrowhead Framework 13

1.3.1 Service Oriented Architectures and System of Systems 13

1.3.2 Governance and Interoperability Using Arrowhead 14

1.3.3 Inter-Cloud Service Interactions 15

1.4 Indoor Positioning and Radio-frequency Identification 15

1.4.1 Localization methods 16

1.4.2 UWB 18

1.4.3 Radio Frequency Identification 19

2 Service-oriented Asset Management Across the Supply Chain **20**

2.1 Use Case and Requirements 20

2.1.1 The Konacranes use case 20

2.1.2 Requirements and Long-term Goals 22

2.2 Architectural Considerations 24

2.2.1 The Big picture and General considerations 24

2.2.2 System-of-Systems as Local clouds 25

2.2.3 Service interaction between local clouds – the inter-cloud scenarios . 26

2.3 Main elements and their interworking: Digtal Twin and RTLS 26

2.3.1 RTLS architecture 27

2.3.2 The Digital Twin module 29

3 Hardware and Software Design **30**

3.1 RTLS Design Considerations 30

3.1.1 Infrastructure 30

3.1.2 Radio-frequency Design 31

3.1.3 Power Supply and Thermal Management 32

3.2 Software Design Considerations 34

3.2.1 Real-time Positioning Software 34

3.2.2 Communication Within The Overall System 35

3.3 Validation and Demonstrations 36

3.3.1 Length of the measurement cycles 36

3.3.2 Localization accuracy 36

3.3.3 Power consumption 37

3.3.4 Demonstrations 37

4 Summary **40**

4.1 Main Contributions 40

4.2 Roadmap for Future Developments 41

4.2.1 Smart Maintenance 41

4.2.2 Delivery Systems integration 42

Acknowledgement **43**

Bibliography **46**

Kivonat

Az ipar – és ezen belül főként a termelés, gyártás – digitalizációját célzó törekvés, közismert nevén az *Ipar 4.0*, az elmúlt évek egyik meghatározó trendje, gyakran kutatott területe. Népszerűsége ellenére, a digitalizáció megvalósítása a gyakorlatban továbbra is számos koncepcionális és implementálási kihívás elé állít minket, melyek lassítják a széleskörű elterjedést. Jelen dolgozat középpontjában egy kompley eszköz-követő (asset tracking) rendszer tervezése, megvalósítása és verifikációja áll, melynek során a fenti kihívásokra kívánok megoldásokat kínálni.

Egy gyártási és logisztikai folyamat digitalizációja során a legfőbb igény, hogy a folyamat egésze követhető és kontrollálható legyen, azonban az automatizált – egymástól független – részfolyamatokat hagyományosan emberi interakció köti össze.

Munkám első részének gerincét egy olyan koncepció tárgyalása adja, amelyben az egyes részfolyamatokat végző rendszerek alkotják az ellátási lánc hálózatát, melyen keresztül egy adott eszközhöz tartozó adatokat cserélhetik. Ezáltal egy adott eszközt leíró adatok birtokosa mindig az a folyamat, ahol az eszköz tartózkodik, így az adatok egy valós idejű reprezentációt valósítanak meg, az eszköz úgynevezett digitális ikerpárját.

A dolgozat második része egy RFID alapú eszköz-követő rendszer tervezését és implementációját ismerteti, amely – ipari igényeknek megfelelően – integrálható a már meglévő infrastruktúrába. Ennek során az elkészült rendszer és a fenti koncepció előnyeit egyaránt tárgyalva kívánom bemutatni, hogy miként küszöbölik ki az emberi interakcióból adódó bizonyos logisztikai hibákat. A rendszer működőképességét egy a gyakorlatban is megjelenő probléma szimulációján kívánom tesztelni, bizonyos, már megvalósult részfeladatok konkrét mérési eredményeivel alátámasztva.

Abstract

The efforts that are aimed at digitizing the industry – especially manufacturing and production – within *Industry 4.0* are one of the dominant and heavily researched trends of recent years. Despite its popularity, putting it into practice still has many challenges, both conceptual and practical, that inhibit the widespread adoption. This paper details the design, implementation and verification of a complex asset tracking system meanwhile discussing possible solutions to many of the actual challenges.

Digitized production and logistics processes offer controllability and traceability, in contrast to the traditional sub-processes are executed through human interactions.

The first part of this work examines a concept, in which autonomous systems that cooperate execute different sub-processes constituting *Supply Chain Networks*. This allows data exchanges between individual devices within and in between these systems. In this case, the owner of the data is always the process in which the given device is involved. Therefore, building on this data gives a real-time representation of an asset, i.e. digital twin of the device.

The second part of this paper introduces the design and implementation of an RFID-based asset tracking system that can retrofit into the legacy infrastructure of a company. During these steps, the advantages of such a system will be presented within the described concept, focusing on how certain logistics failures and bottlenecks due to human interaction can be eliminated. The feasibility of this system will be tried within a use case of a practical problem reasoned with measurement results of certain, implemented sub-tasks.

Introduction

The importance of digitization in production is well-known and there is no sector in industry that hasn't recognized it yet. Broadly speaking, digitization here means being able to gather all relevant data about manufacturing processes and use them to increase efficiency, productivity and control over them. This leads to factories where cyber-physical systems manage whole supply chains based on the collected information. Despite the vision of emerging *future factories*, the changes in certain segments of industry are not going as fast as it was predicted. Those companies that traditionally invest in R&D are also flagships of the digital transformation, while those who focus on mostly the products instead of production, are the slow adopters of such new technologies.

In most cases, companies consider digitization as a complete replacement of the existing infrastructure; and more or less that might even be the truth in some cases. Generally, supply chains operate as homogeneous systems, in which each part is connected to primarily two other parts: one that provides our input and one that utilize our output. These systems are designed to perform one special task, therefore they lack any flexibility. This then naturally means that these systems have to be treated as a whole and its parts cannot be replaced or upgraded individually.

In order to involve these companies in the movement of digitization, alternative solutions need to be offered that don't require legacy systems to be replaced, rather to have retrofitted with new devices. This requires appropriate design principles that have to ensure the standalone operation of the legacy system and yet provide high modularity to allow the integration of additional devices. Such systems have to meet requirements like their predecessors related to reliability, safety and security of both critical data and communication.

An important factor that has to be taken into account during the design and implementation of new "smart systems" is the ability of ensuring interoperability. Most of these devices must function autonomously, and follow the concept of cyber-physical systems: they must be able to exchange relevant data that can be used for executing and optimizing business tasks and processes. A key part of the design needs to be the *System of Systems approach* which enables the creation of large-scale, vendor lock-free, distributed industrial systems.

Having this in mind, this paper will present a complex asset tracking system that can digitalize and further aid whole supply chains. This system consists of:

- a self-designed and implemented hardware infrastructure providing real-time location information using Ultra-Wide Band transceivers (UWB) [1];
- with also self-designed and implemented hardware components that can provide additional identification information to establish digital product twins with radio frequency asset identification (RFID) technologies [2];
- working distributively in a a service-oriented architecture (SOA), provided by the Arrowhead framework [3];
- according to the requirements of a real-life use case of the Finnish company Konacranes [4];
- with having designed for working with legacy information sources of the company.

Outline of this work

This paper is presenting my work throughout Chapters 1-4. Chapter 1 presents the state of the art within digital industry and logistics. Firstly, key definitions and concepts are introduced, such as the general approach taken while modeling value chains, and how the world of industrial and automation Internet of Things (IoT) connects to it, is also presented.

Moreover, since parts of this work is related to Real-Time Location Systems (RTLS) and Radio Frequency Identification systems (RFID), the basics of this domain is also discussed; together with the Arrowhead framework which will be the integration platform to build a system of systems (SoS) out of the single modules of the architecture.

Chapter 2 lays down the exact use case and its requirements. Moreover, I present an architecture concept here that can satisfy said requirements with specialized workflows and communication sequences. Here, my systems and hardware modules can aid or even replace activities previously done by human workers, starting from the very first product order up to delivery to an end-user at the end of the value chain.

Chapter 3 further details the individual modules of the architecture, mostly focusing on exact hardware and software design considerations made when making the prototypes and demonstrator systems.

Meanwhile, Chapter 4 provides draws the conclusions and provides next steps, my future work.

Chapter 1

Related Works

The term *Industry 4.0* nowadays notes the digitization of industry in a wider sense, however, when this term was invented the, emphasis was on automation and data exchanges in manufacturing. This has invoked concepts and then technologies related to Cyber-Physical Systems (CPS).

This chapter explores the fundaments of production and logistics within state of the art and within the envisioned Industry 4.0.

1.1 Logistics and Value Chains

Logistics and asset handling is still one of the lesser automated and digitized corporate functions. Warehouse handling and administration is still a highly labour-intensive part of all companies across all industries.

Naturally, there are cutting edge solutions, i.e. fully automated warehouses, lights-out factories[1] and complex logistics providers. However, these well-known examples of highly automated and granulated tracking systems, like the one of Amazon, require a certain scale of economy in order to worth the investment.

Even slightly smaller or regional companies cannot afford these. Therefore, in most cases the levels of automation are very distinct – especially in manufacturing. Moreover, even if they do, these are proprietary solutions and highly closed systems, which is developed by themselves, for themselves. There is a reason why there are tasks that are automated and the are ones that still require human work.

The small oligopolic competition between the various vendors are one of the reasons behind this. If a company chooses one of these vendors on any of the levels within the ISA 95 production pyramid (see section 1.2), they are almost forever locked in with that given vendor. There is currently no, forced or naturally evolved, standard way of building production and logistics from heterogeneous suppliers.

[1] Production factories with absolutely no human presence required on site.

My work here is, however, aimed at making these systems loose-coupled and service oriented so that a certain level of modularity and interchangeability would be the result. What is more, this should happen with re-use rather than re-invent (i.e. relying on legacy).

1.1.1 Stakeholders and Systems Across Supply Chains

Generally, production, warehouses and assets are managed through an Enterprise Resource Planning (ERP) system. Such a system consists of quite a lot and various modules, but is mostly supplied by one vendor, such as SAP [5]. These solutions offer rich functionality but they may not fit well into a company's business model. Customization is nevertheless expensive since these solution barely handle unique demands.

The corporate functionality *Supply Chain Maganagement* (SCM) consolidates all tasks related to the flow of goods and services. This includes the relocation and storage of raw materials, work-in-process inventories and finished products. Companies aim to create highly optimized, efficient solutions here, including management methods, planning and control, organization structure, etc.

Many aspects of the SCM has the appropriate support, usually integrated into an ERP (Enterprise Resource Planning) system which enables a close, but almost fully manual cooperation between stakeholders and companies. However, this way of logistics has not changed a lot and the level of automation is low.

The term "supply chain" in itself here implies that various stakeholders and their conflicting interest are involved in the overall life-cycle of a product [6]. Logistics is one of the components within SCM, in which all the interested parties are involved in contrast to many other processes that can be handled fully in-house. A usual manufacturing Supply Chain is presented in Figure 1.1. Within this chain, logistics and asset tracking is essential, however often neglected and with no room for improvement, for budgetary reasons.

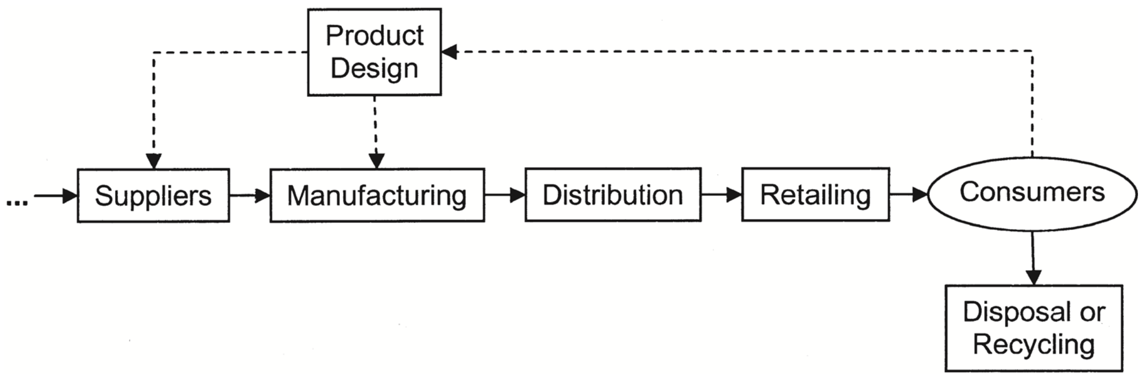

Figure 1.1: *A typical manufacturing Supply Chain [7]*

Using technologies that are often associated with Industry 4.0 in SCM has already begun [8]. However, these conceptions generally assume the existence of an already implemented CPS solution that enables the collection and the use of big data as it is presented in Figure 1.2.

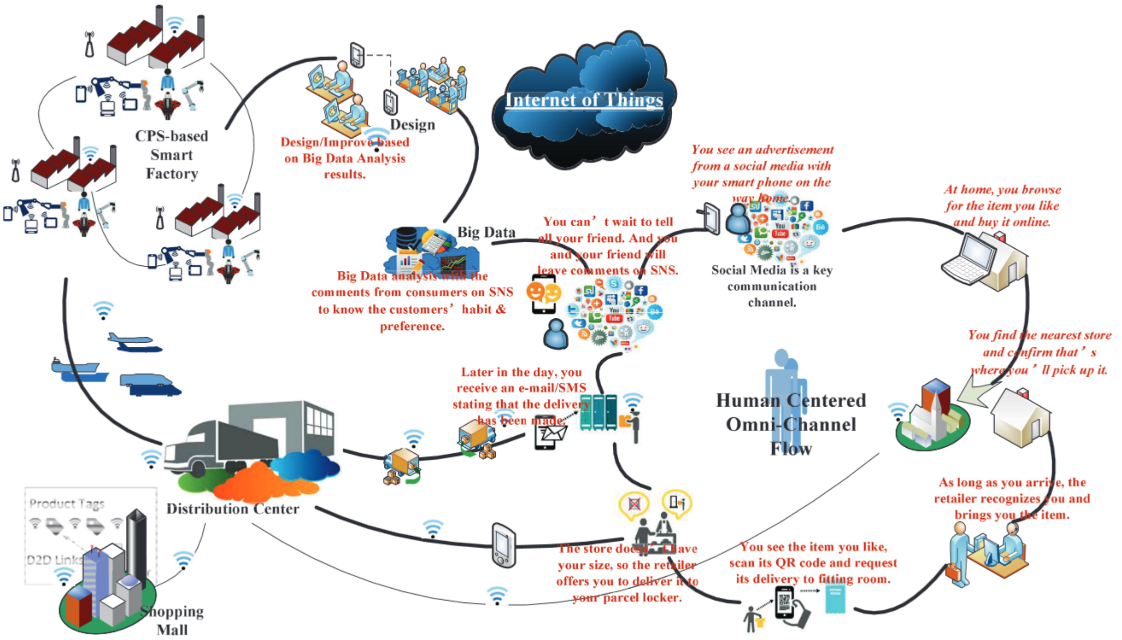

Figure 1.2: *A framework of human centered IoT-based logistic service for Omni-channel [8]*

However, the presumed quantity and quality of data is usually not available. This is due to the lack of real-time, high definition asset monitoring capabilities. Quality assurance and product traceability is not prevalent, however highly wished for.

Besides the warehouse and asset management, production itself is rarely monitored sufficiently either, let alone on a single item ("lot size of one") level. These type of data source can be the basis of high-level logistical and SCM solution. That is why the focus of this paper is on automatized asset tracking solutions implemented in such a way that fits into the Supply Chain Management approach.

1.2 Digitization and IoT

With the advent of affordable and easy-to-use hardware and software technologies, traditional industry is changing as well. There are many papers and case studies jumping on this bandwagon, presenting this field of research as straight forward and closed. However, besides the unclear corporate visions, there are issues on the technical concepts level as well. This work focuses how we can digitalize asset tracking over multiple process steps and even over multiple companies, and link it with digital product twins, in order to have full trace-ability.

Digital twins and smart products are concepts that could, in theory, facilitate the requirements of full trace-ability, and customization. A major issue is that these terms don't have a general definition – as many others in this field of research – and they vary between branches of science. For example, some definitions emphasize how to utilize it, while others focus on implementation opportunities.

In my understanding here, Digital Twin means a *"copy of the physical asset"* [9], which is achievable by gathering appropriate real-time information about the asset. Further along the line it can also include using machine learning and analytic tools on the collected data to estimate parameters that were not measured directly. Both approaches represent a certain level of digitalization, and they require software with different complexity and amounts of hardware support.

There are two major fields, in which IoT technologies can aid industrial production: within automation (in the shop floors) and throughout the supporting infrastructure.

1.2.1 Automation IoT

Industrial automation has always been highly regulated and standardized since it is a mission-critical field with serious security and reliability requirements. Traditionally, these solution was build for executing a specific tasks which resulted closed, rigid and monolithic systems. The ANSI/ISA-95 standard [10] that provides consistent terminology, information and operations models to clarify application functionality and how information is to be used. It defines the five levels to these systems as it is shown in Fig 1.3.

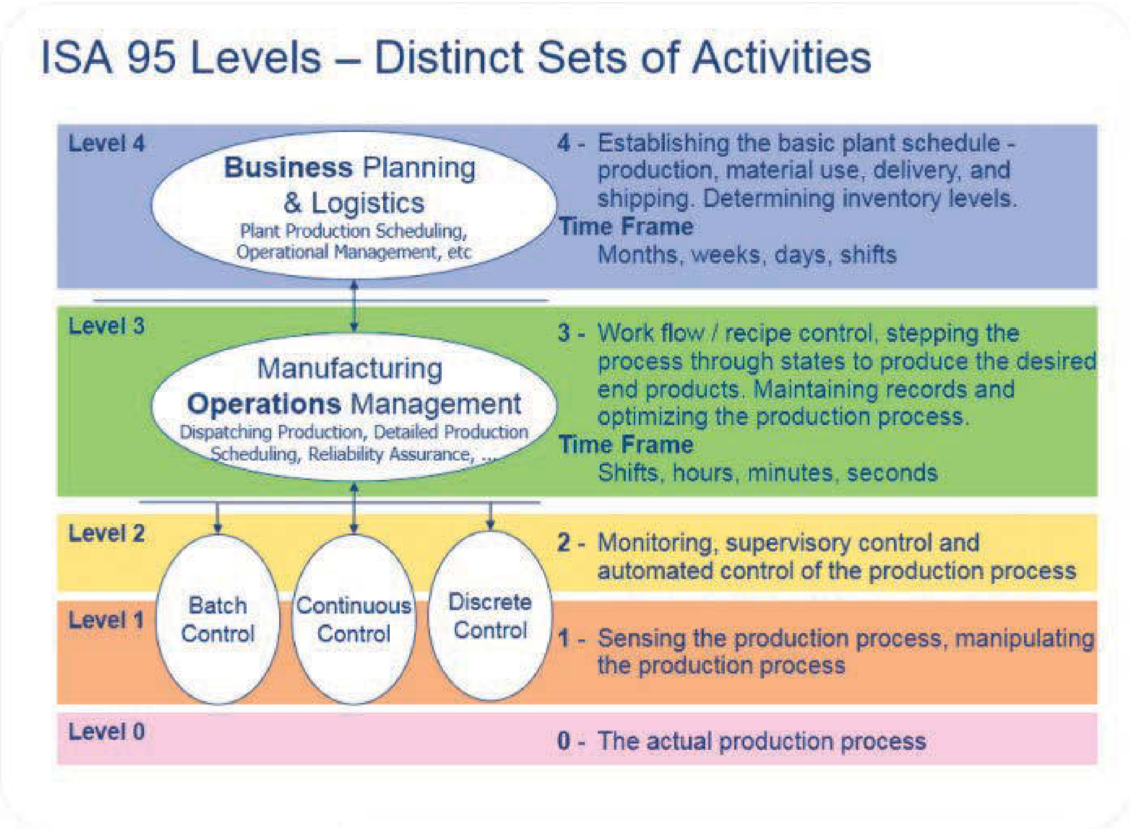

Figure 1.3: *The levels of automation in the ISA-95 standard [3]*

ISA 95 is accompanied by many related standards that define the general and communications architecture within and between the pyramid levels. These communication solutions, within levels and also between them are based on legacy protocols with a capability of hard real-time performance that fulfill jitter and latency requirements (millisecond range) e.g. one-way analog technologies, field buses, RS-232 or industrial Ethernet. Since all access to a level is centralized in the upper-level, there is no direct way between distinct levels, moreover the I/O ports of controller are fixed to peripherals. This creates a fairly rigid network setup, both on a physical level and in an accessibility sense. Any modification to the system requires that all levels are partially shut down and entered the new setup.

Smart manufacturing in Industry 4.0 means not only developing and deploying new systems, but making legacy ones flexible and easily reconfigurable. In order to do that, one of the main objective of Industry 4.0 is connecting these systems into the same network, where all devices i.e. controllers, sensor and other peripherals are IP endpoints, which are addressable and accessible on the network directly. That is what called Automation IoT [3].

This approach enables to use the traditional automation solutions as Cyber-Physical Systems in larger industrial environments. Since most of them are already monitored or they measures environmental variables these valuable information can be used in a wider scope. One of the possibilities is related to digital twins and real-time, highly granulated asset tracking by using legacy sensors.

1.2.2 Industrial IoT

Every use case requires building new systems in different degrees. These are based on current embedded technologies (e.g. microcontrollers, sensors) connected to Internet Protocol (IP) networks. However, it doesn't necessarily imply creating new automation solutions from scratch, because legacy systems can be extended with new devices. In many cases, this can be a cost-effective solution with the emergence of Wireless Sensor Networks (WSN).

WSN are commonly associated with cheap, smart sensors which have low processing capabilities and only used to gathering and forwarding raw data. It enables traditional systems to be equipped with these small, low-power devices. This can be taken into contrast with installing a general purpose, large scale ambient or embedded system. In this sense, all machines can be "things" that supply us with invaluable data on the single item level (way better resolution than what is usually available right now).

This concept – commonly known as Cyber-physical systems – offers numerous possibilities in automation, data exchange and manufacturing processes. Utilization of telemerty and data collected by monitoring pieces of equipment, devices, products and processes allows to establish a highly granulated, digitized, real-time functions e.g.: asset tracking and digital twin, distributed decision making and many others [11].

Primarily, CPS-s differ from traditional centralized systems in one major aspect: CPS-s are intended to be autonomous to some extent. They can incorporate ISA95 level 2-3 functionalities, and represent these resources to other level 3 or level 4 entities in the ISA95 sense. They shall pose (the cyber part) an interface towards other CPS-s that help with the monitoring, telemetry and even configuration of the incorporated physical resources, processes.

1.3 The Arrowhead Framework

1.3.1 Service Oriented Architectures and System of Systems

Dealing with large systems always requires using special design principles especially if we are talking about collaborative systems e.g. Cyber-physical systems in contrast to monolithic ones e.g. legacy automation systems. Collaborative systems or *System of Systems (SoS)* can be characterized by five main characteristics in relation to other very large and complex but monolithic systems [12]: (i) operational independence of its systems, (ii) management independence of the systems, (iii) evolutionary development, (iv) emergent behavior and (v) geographic distribution.

Design of such systems is based on the refinements of general heuristics which are not strict rules but only recommendations. These mostly pay attention to the stability of the overall system in any phase of its operation. Since collaborative systems defined by its interfaces, the inter-system connections really are the architecture of SoS [13]. During its lifetime, the inner structure of the SoS can change if the independent systems within it evolve. In each case the emphasis should be placed on the upper layer of the communication (the abstract components) in contrast to the actual physical interfaces which are given – in the case of autonomous systems – or hidden behind one or multiple abstract layers – e.g.: The Internet. Therefore the existence and stability of a SoS is heavily based on its ability of handling changes in the protocol stack and transferred data structure.[2] Traditionally, the inter-system connections are well-defined and laid during the design steps. However, in such an automation environment of CPS-s the need of making new collaborations leads us to systems in which its components change dynamically to serve new demands. This concept is pointing towards Service-Oriented Architectures (SOA) that use exactly such fundamentals.

The term Service Oriented Architecture (SOA) can refer to different things: it is commonly cited as a software architecture but it can mean only an approach or design paradigm. Basically, a SOA based system consists of services that are provided by physically independent, loosely coupled software programs with distinct design characteristics while they can also consume each others' services [14]. Its definition shares principles with the definition of modular programming: *"Different services can be used in conjunction to provide the functionality of a large software application [15]."*

The building blocks of the SOA systems are these services. There is no standardization on what characteristics services have, or how they can be implemented. The authors of [16] emphasized the following characteristics of service abstraction:

[2]For example: IP Standard's version numbering system allows both IPv4 and IPv6 packets to be transferred.

- Loose coupling: hardwired connections between entities are not permitted.
- Rather services are dynamically discoverable upon need (run-time).
- Services are self-contained and modular. A service supports a set of interfaces and these interfaces are logically cohesive (they implement the same functionality).
- Modular understandability: the user has to be able to use the service without having knowledge of any other underlying implementation details.
- Modular decomposability: a complex service can be created from simpler atomic services.
- Interoperability: systems using different platforms and programming languages should able to communicate with each other using services.

SOA is primarily associated with the Web Service stack. However, it is worth noting, that the SOA principles can be implemented using (nearly) any technology.

1.3.2 Governance and Interoperability Using Arrowhead

Arrowhead framework aims providing interoperability within and between closed or separated automation environments in a Service Oriented Architecture where the links between systems are the services themselves. It enables connecting modules or even different systems on demand and dynamically which allows high flexibility.

Arrowhead creates central governance with a minimalistic set of core components that take over certain configuration tasks from individual systems and components in such an automation environment. The systems in the Local cloud can provide and consume Services from one another: they create and finish servicing instances dynamically in run-time. Currently, there are three mandatory Core Systems – which will be mentioned in further chapters – as it is seen in Figure 1.4 [17]:

- The Service Registry stores all the Systems (that are currently available in the network) and their service offerings. Systems have to announce their presence, and the services they can offer. The registry takes note of this information when systems come on-line, and might have to revoke them when they go off-line.
- The Authorization System - as its name suggests - manages authentication and authorization (AA) tasks, however, it covers some other security-related issues as well (e.g. certificate handling)
- The Orchestrator is responsible for instrumenting each System in the Cloud: where to connect and what to consume. It instructs Systems so by pointing towards specific Service Providers to consume specific Service(s) from.

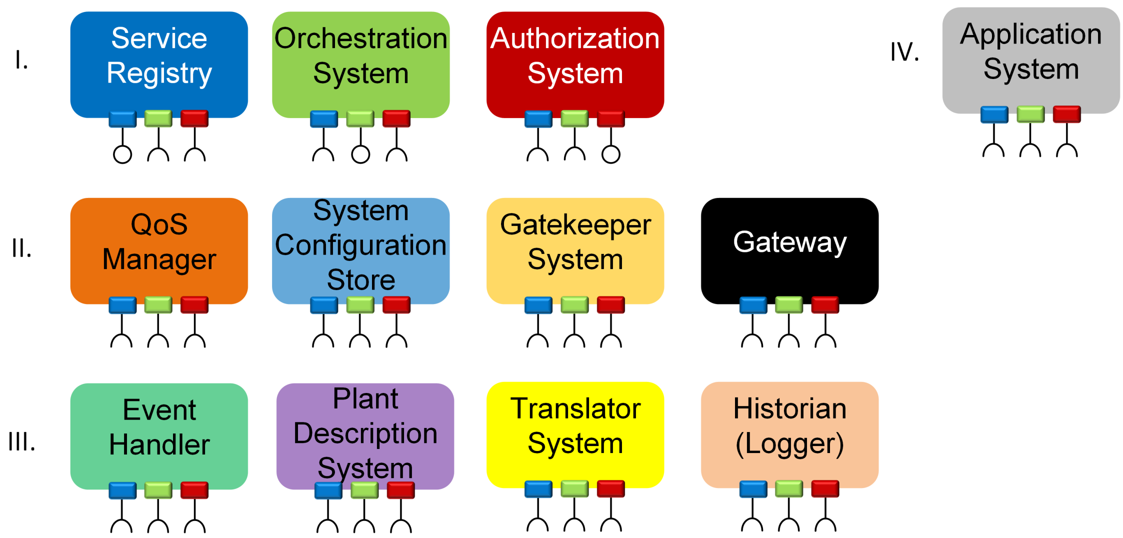

Figure 1.4: *The mandatory and supplementary core systems of the Arrowhead framework [17]*

1.3.3 Inter-Cloud Service Interactions

In many cases there is a need to make Local Clouds collaborative to share services between different System of Systems, e.g. automation environment of various stakeholders. Inter-cloud communication is built upon the local orchestration model, where Core Systems' handle and manage all connections within their own Local Cloud. In order to achieve global interoperability, supporting core systems, namely: the Gatekeeper [18] and Gateway [19] modules are used in inter-cloud orchestrating processes.

The Gatekeeper module provides essentially two services for the mandatory Core Systems: Global Service Discovery (GSD) and Inter-Cloud Negotiations (ICN). GSD is an extended service discovery mechanism that locates offered services in neighboring Clouds, while ICN is basically a matchmaking process which establish the connection between the endpoints, but it does not transfer actual data from one application system to another one.

Since the Gatekeeper is always visible from outside (Internet), it is vulnerable to security threats. Therefore it plays a part only in the control mechanism and exchanging the data between application systems is handled by the Gateway module. The Gateway provides a bare TCP or a secure Transport Layer Security tunneling service between Local Clouds, while it acts as a trusted agent: within its own Local Cloud application systems see their Gateway as the actual targeted system. A more detailed description of secure inter-cloud communication is available in [19], which I participated previously in.

1.4 Indoor Positioning and Radio-frequency Identification

Most of us use localization almost every day while navigating by a map application on our smart devices. Therefore the term "localization" has become one with GPS which stands for Global Positioning System. As its name says, it is a global navigation satellite system that provides geolocation and time information with an average accuracy of 5-15 meters. Its precision can be significantly increased, but it involves expensive investments. Moreover, GPS is generally not suitable to establish indoor locations, since microwaves will be attenuated and scattered by roofs, walls and other objects [20].

1.4.1 Localization methods

Due to the increasing demands of indoor localization, different solutions serving different needs came to light. There are two commonly used terms: (i) IPS (Indoor Positioning System) primarily concerns location-based services on mobile phones where GPS does not work, and (ii) RTLS (Real-Time Locating System) that is used to automatically identify and track the location of objects or people in real time. The boundary between the two terms is blurred and they are often interchangeable. Nevertheless, in this paper RTLS will be used consistently. The basics of such a system is usually built upon measuring distance between moving tags that are tracked and fixed (and known) reference points (referred to as Anchors in the RTLS nomenclature) to calculate the absolute position as it is shown in Fig 1.5. The concrete solutions vary depending on the use cases.

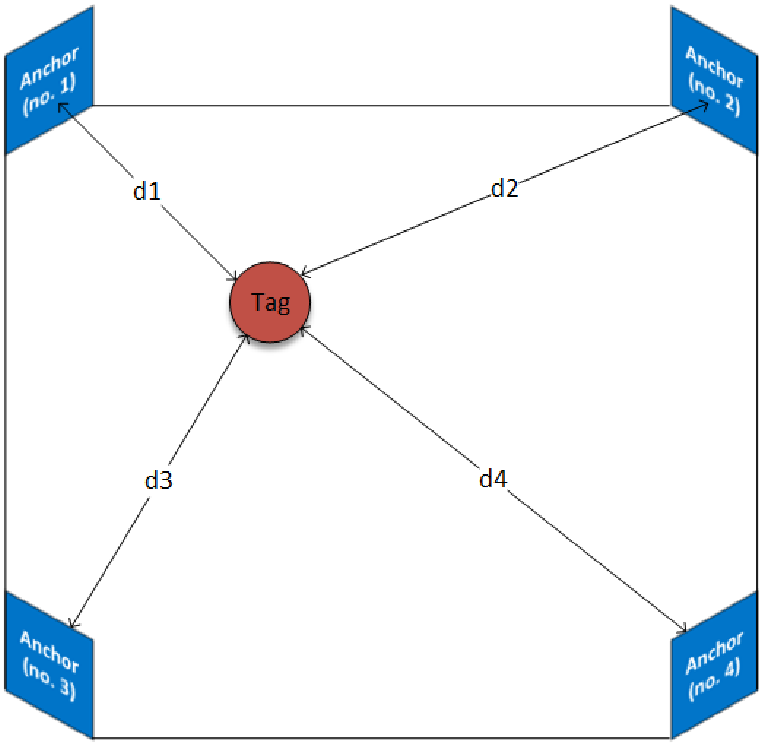

Figure 1.5: *A basic RTLS system that uses ToF method to estimate the position of tag*

Signal Strength Based Schemes [3]

These schemes involve measuring the signal strength of the arriving radio signal at the receiver. Knowing the power at which the signal was emitted from the transmitter, the propagation characteristics of that particular radio signal in air and with some *a priori* knowledge of the environment, it is possible to calculate approximately where the transmission originated, based on how attenuated it is at the receiver.

These schemes are perfectly adequate in certain circumstances, but, generally speaking, where high levels of accuracy are required, they need to be augmented with additional technologies to provide the necessary levels of accuracy. These schemes are often associated with Bluetooth and RFID (Radio Frequency Identification) technologies which, means RFID would handle both identification and positioning task. However, in this paper, UWB (Ultra wide-band) technology will be used for positing because of its better accuracy.

Time Difference of Arrival (TDoA) and Phase Difference of Arrival (PDoA)

Time Difference of Arrival method uses fixed and known reference points (readers) which are time synchronized. TDoA estimate gives the difference between the arrival times of two signals traveling from the transmitter to two readers (the reference point may be arbitrarily selected), which forms a hyperbola with foci at the two readers [22].

The position of moving tag can be determined by using a technique called Multilateration, which involves solving a non-linear equation system – where the solution is the intersection of hyperboloids. A huge disadvantage of this method is the necessity of synchronizing devices, which is achieved by wiring clock signal between them to provide an appropriate accuracy.

Phase Difference of Arrival (PDoA) method can be implemented different ways: based on high node density or array of antennas. In the first case, certain nodes are synchronized and placed in a straight line. Because of the constant speed of radio waves, the signal reaches the different nodes at different times. In the second case, an array of antennas replace the multiple number of nodes in one device. Assuming these arrival times can be measured accurately, then a measure of the phase difference arrival can derive the location of the tag. The disadvantages of this scheme is the high cost of an array of antennas or synchronization of nodes, moreover these method does not deal particularly well with multipath propagation between the transmitter and the receiver antenna array and so are best suited to Line of Sight scenarios [23][21].

Time of Flight (ToF)

In Time of Flight (ToF) method, estimating distance is based on the propagation time of radio signals from a transmitter to a receiver. The radio waves travel very close to the

[3] This is based on: [21]

speed of light, therefore the well-known $v = s/t$ equation can be used. The position of a tag can be determined by trilateration[4]: in this case, estimations are based only the three nearest reference points.

Since the measurements rely on the signal propagation time, the reference points do not have to be synchronized, which is major advantage of this scheme. However, it involves sending two messages, because the receiver cannot calculate the distance, due to the lack of synchronization. To handle this issue, the receiver returns the message to the transmitter which divides the propagation time by two (assuming that the time of flight is the same) – this is what's known as Two Way Ranging (TWR).

Nonetheless, this method is pretty inaccurate, since propagation time includes the turnaround time within the receiver. Additionally, there are various physical effects, such as clock drift, from which the overall propagation time suffers. To achieve precise measurement, Symmetrical Double-Sided Two-way Ranging (SDS-TWR) is used, which consists of four message instead of two. Here, the messages contain timing and delay related information (e.g. turnaround/process time). This method has a very accurate output, but it takes a lot of time to implement and needs adequate hardware support.

1.4.2 UWB

Ultra-wideband (UWB) is a radio technology that is traditionally used in non-cooperative radar imaging and recently in collecting sensor data, precision locating and tracking applications. It enables transmitting large amounts of digital data over a wide frequency spectrum using short-pulsed, low-powered radio signals. UWB commonly refers to a signal or system that either has a large relative bandwidth that exceeds 20% or a large absolute bandwidth of more than 500 MHz [1]. It is rarely used to transferring user's data in commercial products, because the cost of transceivers are relatively high and users' demands are can be fulfilled using other wireless or wired solutions e.g. WiFi or HDMI.[5]

Nowadays, UWB is frequently used in indoor RTLS systems, where high precision, real-time localization information is required. The high bandwidth and extremely short pulses waveforms help in reducing the effect of multipath interference and facilitate determination of TOA for burst transmission between the transmitter and corresponding receiver, which makes UWB a more desirable solution for indoor positioning than other technologies [24]. UWB technology, unlike other positioning technologies such as infra-red and ultrasound sensor, does not require a line-of-sight and is not affected by the existence of other communication devices or external noise due to its high bandwidth and signal modulation [25]. Furthermore, the cost of UWB equipment is relatively low and it consumes less power than other competitive solutions – BLE and WiFi not included.

[4]Trilateration is the process of determining absolute or relative locations of points by measurement of distances, using the geometry of circles, spheres or triangles.

[5]UWB was considered as a revolutionary solution to replace HDMI, but almost nobody required short-range transmission with such a high cost.

1.4.3 Radio Frequency Identification

This section is mostly based on: [2]. Radio-Frequency Identification (RFID) refers to the use of radio waves to read and capture information stored on a tag attached to an object. A tag can be read from up to several feet away and does not need to be within direct line-of-sight of the reader to be tracked, so it may be embedded in the tracked object. Although, RFID is generally used for identification, it is applicable in other fields e.g. localization.

Tags can be classified into two groups: *Active Tags* require a power source -— they're either connected to a powered infrastructure or use energy stored in an integrated battery. In the latter case, a tag's lifetime is limited by the stored energy, balanced against the number of read operations the device must undergo. *Passive Tags* also have an indefinite operational life and are small enough to fit into a practical adhesive label, but they don't require battery or maintenance because they use the radio energy transmitted by the reader.

Two fundamentally different RFID design approaches exist for transferring power from the reader to the tag: magnetic induction and electromagnetic (EM) wave capture. These two designs take advantage of the EM properties associated with an RF antenna – the near-field and the far-field.

The near-field coupling is based on Faraday's law of induction: a near-field antenna uses inductive coupling which means that it uses a magnetic field to energize the RFID tag. A magnetic field is created by alternating current in the near-field region that allows the RFID reader's antenna to energize the tag. The tag then responds by creating a disturbance in the magnetic field that the reader picks up and decodes. However, near-field RFID is the cheap and effective solution it has some physical limitations such as its range that can be a maximum of 1 meter.

A far-field antenna uses capacitive coupling (or propagation coupling) to energize the RFID tag. Capacitive coupling occurs when the RFID reader's antenna propagates RF energy outward and that energy is used to energize the tag. The tag then sends back a portion of that RF energy to the reader's antenna as a response which is known as backscatter. As a rough design guide, tags that use far-field principles operate at greater than 100 MHz typically in the ultra high-frequency (UHF) band such as 2.45 GHz. A far-field system's range is limited by the amount of energy that reaches the tag from the reader and by how sensitive the reader's radio receiver is to the reflected signal. A typical far-field reader can successfully interrogate tags 3 m away, and some RFID companies claim their products have read ranges of up to 6 m.

Chapter 2

Service-oriented Asset Management Across the Supply Chain

This chapter details how the Arrowhead framework can aid extending a single production line into a smart process- and asset-management network to improve its role as the backbone of manufacturing. The whole concept aims to lay the basis for CPS-driven smart factories by digitizing the supply chain within and between production lines. This includes (i) monitoring each process and asset by autonomous systems that (ii) can interconnect with each other to serve dynamically changing needs (iii) through their ability to share and synchronize information in order to enhance management and logistics.

The following sections details my proof of concept implementation done within the Productive4.0 project [26]. Here, an actual industrial use case will be presented and investigated from different aspects in order to identify requirements.

2.1 Use Case and Requirements

2.1.1 The Konacranes use case

A Finnish company, Konecranes [4] is a world-leading group of Lifting Businesses™ serving a broad range of customers: manufacturing and process industries, shipyards, ports and terminals. Konecranes aims to integrate data from various sources across the value chain, especially related to collecting and integrating automated tracking information through the different steps of production. Their main goal is to utilize this information in various fields e.g. logistics, management and maintenance to optimize them.

One of their widely sold products is a chain-hoist, shown by Figure 2.1. This machine supports production in manufacturing areas by moving around large objects precisely. The assembly of product assets usually involves numerous, well-separated steps which are mostly done by human workers with the help of chain-hoists. Most of these steps are related to different workstations, where the work can be divided into phases as it is seen in Fig 2.2.

Figure 2.1: *A typical chain hoist that used in production line. It is hung down from a metal pipe by its hook so it can be easily moved.*

- Generally, the first phase covers moving the input of the actual step to the workstation. The input is usually the output of the previous step (in one word: the asset) and/or additional parts and base materials. The assembler has two jobs in this phase: (i) moving the input to the workstation by using a chain-hoist which lifts up the asset and (ii) tracking it by reading the attached bar-code. The bar-code reader in this case is a handheld device.

- The second phase includes the actual manufacturing processes, i.e. assembling the parts of the asset and fast-checking the product. The work itself are done on a work table, so the assets have to be put on the table. After finishing this step, they will be lifted up by a chain-host – which can be the same as before or another one. This phase of work usually takes less than one minute.

- The third phase covers moving the asset to the next step. Firstly, the asset has to be taken off the chain-hoist, then the assembler reads the bar-code. After this, the asset leaves the actual step of process and the worker returns to phase one.

Another steps include supplying and distributing materials, quality control processes and preparation for transportation are integral parts of manufacturing but, are not related to

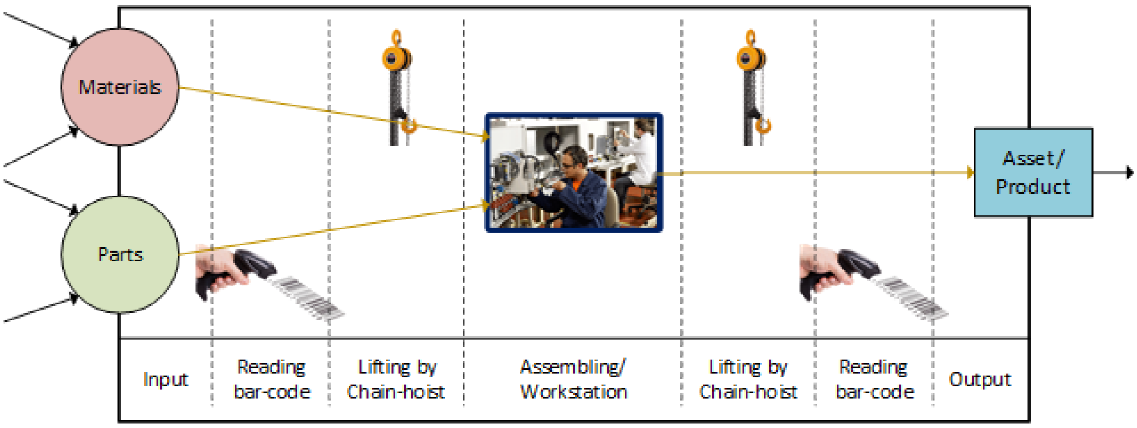

Figure 2.2: *Schematic figure of a workstation*

a specific workstation. All works in these steps, i.e. tracking and storing management, as well as logistical information, are also done manually at the moment.

2.1.2 Requirements and Long-term Goals

In this use case, all requirements are derived from the relatively low degree of automation in supplementary logistical tasks. However, there are no automated sub-processes during manufacturing, due to the fact that producing chain-hoists involves lifting heavy weights and positioning them precisely during assembly – therefore most of the jobs are done by manually controlled machines as it was described. A major problem with this production chain is the high amount of unnecessary human interaction: besides the assembly, moving the asset from one site to another, workers also have to deal with tracking products and materials (i.e., partly by reading their bar-codes).

Manual tracking affects the whole value chain, namely: (i) it slows down the assembling procedure by requiring the assembler to share the attention between two tasks and (ii) to handle tracking which is not related closely to the manufacturing process. Moreover, (iii) management and logistical steps are also bothered with this duty. This issues do not only decrease the efficiency of manufacturing, but increase the chance of possible logistical failures and bottlenecks due to human interactions involved.

Furthermore, Konecranes representatives raised opportunities to extend their ERP system, in order to improve post-production services such as maintenance and advisement. This involves monitoring the assets and store an extensive descriptive model of each product that contains relevant, measured factors related to production and real-time operation. In our point of view, it means that the planned system has to be expandable and modular enough to allow further functionalities to be added, or modules – e.g. a monitoring sensor network – to be attached. In addition, it has to be capable of receiving information about the asset from external systems (i.e. customers' ERP system). However, this topic is mostly out of the scope of the current paper, these ideas had to be considered further, because they affect architectural design.

The idea of sharing information between stakeholders of the supply chain – including the producer of the product, provider of the base materials and the customers – was presented in section 1.1. This type of interoperability enables opportunities of optimizing logistics, manufacturing and supplying the product e.g. service and maintenance. Designing a system such a way, i.e. making it capable of exchanging certain information with external systems, raises issues such as security and trustworthiness.

One of the key issues when a new piece of equipment are deployed, a new device is installed or a new method is utilized, is the compatibility with the legacy infrastructure. In many cases the majority of investment costs is spent on modifying the existing infrastructure to establish compatibility. In another scenario where the process and the production site are safety-critical or safety-related, the costs increase due to the qualification tests. In such a case where handling the needs does not involve safety-related solution, the best-practice is to develop a system that can be retrofit into the legacy infrastructure which is also cost-effective.

Based on the previous paragraphs, the requirements can be formulated as follows:

- Eliminating manual tracking – i.e. unnecessary human interaction – from each step of manufacturing by replacing it with automated tracking.
- Tracking has to provide full traceability of assets through the whole supply chain including assembling and logistical steps, as well. It is due to the fact that besides real-time positioning service, the path of an asset has to be traced back.
- All of the functions have to be implemented independently from the existing infrastructure without influencing its operation.
- These requirements (above) should be fulfilled as a part of an intelligent system, which has the main goal of tracking assets, but it is able to interconnect with existing ERP, CRM (Customer Relationship Management), CMMS (Computerized Maintenance Management System) and MES (Manufaturing Executing System) systems to provide management tasks and smart logistics as a supply chain network in the future as it presented in Fig 2.3.

From a more general prospective, RAMI4.0 [27] defines the Reference Architectural Model for Industry 4.0, which provides various high-level requirements – to be mapped for each use case in order to complete their thorough description regarding all aspects of Industry 4.0.

In the next section (2.2) requirements will be refined and detailed from the aspect of design and implementation.

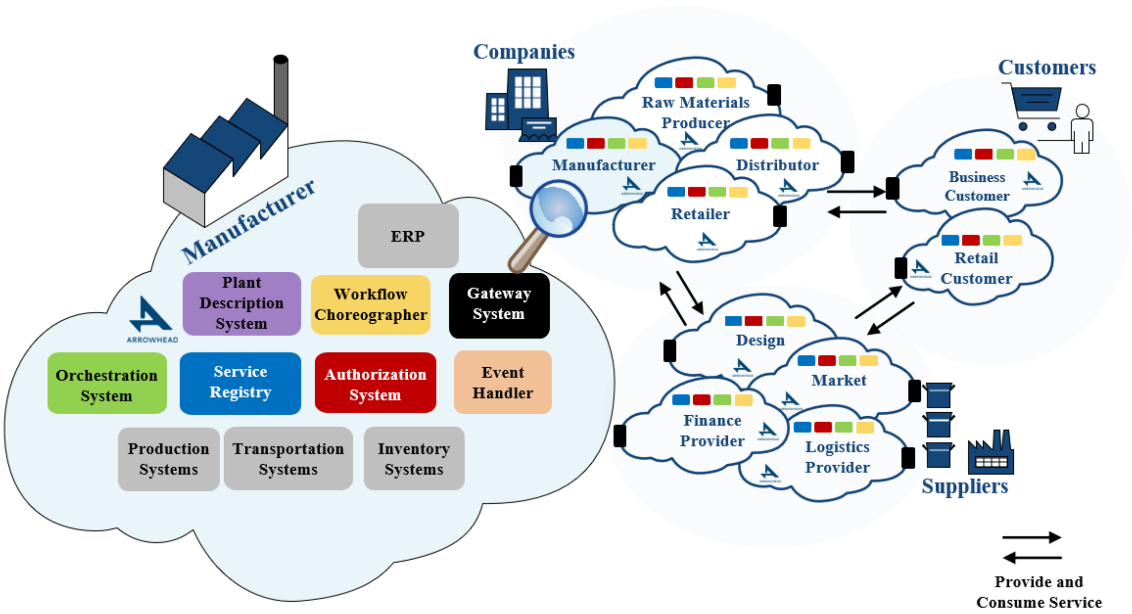

Figure 2.3: *Extended Supply Chain Management on Arrowhead basis*

2.2 Architectural Considerations

Since the planned system that fulfills the requirements is much more complex than most of the RTLS systems, its architecture will be discussed by using top-down approach. Therefore, in this section the emphasis is on the path and properties of data (i.e. connections between modules and systems, and detailing what can data be used for and how), in contrast to concrete implementation questions such as positioning.

2.2.1 The Big picture and General considerations

One the most important characteristics of the planned system that shapes its architecture, is the need for operating in a distributed manner. The way we realize the required functionalities must take into account that the systems and services must be separated clearly and connected loosely with each other while keeping the non-functional requirements also in mind (e.g. the interoperability with external and legacy management systems).

Modules must be able to run independently from each other, which means that every system is expandable with additional modules. In such a System of Systems architecture the links between subsystems are their interfaces which describe how and what type of data can be exchanged. It also implies that the concrete implementation of these modules is hidden so possible modifications do not have an impact on the operation of the overall system as long as the interface is invariable.

According to the requirements, the architecture has to provide not only the independence of modules but the ability of being interchangeable. This allows the system to follow the changes of the value chain – including the whole manufacturing process, management and logistics – without making significant changes in module-level. For example: the localization sub-systems can be replaced with another ones, or the sequence of manufacturing a product can change without affecting the operation of the overall system. Realizing it requires that the individual modules do not have hardwired connections.

Our inventory which the system relies on was introduced in Section 1.3 (and 1.4), so it was already revealed that the architecture is heavy built upon the Arrowhead framework. The capabilities of the framework was already detailed, so it can be clearly seen how it fits into this use case by providing collaborative IIOT based automation. To emphasize that the detailed system is a hands-on solution, during the further discussion of the architecture it will be shown how such an Arrowhead based system meets the requirements. However, in the first place we should look into how the mentioned design considerations can be realized by using Arrowhead.

2.2.2 System-of-Systems as Local clouds

The above described, independent systems compose a Local Cloud in the Arrowhead nomenclature. Different approaches can be used in this level, which means that a factory in itself might be a local cloud but it can divided into multiple ones. It mostly depends on the size of the site. In this use case each site is considered as one Local Cloud.

From this point of view the different modules act as Service providers and consumers, and can be interconnected with each other by Orchestration. While modules can be classified as Service Providers and Service Consumers, but in some cases a module produce a service as well as consume them. Since they do not really own the services – all services are registered in Service Registry – but, they implement one or multiple interfaces of a service the compatibility between them can be ensured. For example: Device 'A' and 'B' are connected with 'C' and they use an arbitrary protocol by which they exchange data in an appropriate format, in one word: their interfaces harmonize. If 'A' changes its interface to a new one – included protocol and data format – 'C' can be still connected with 'A' and 'B' in the same time as long as 'B' implements the new interface as well as the legacy interface. In this case there is no need to modify the Core System, device 'B' or any other element of the Local Cloud.

The previous paragraph implies that if any modification happens in the production chain, the overall system can adapt to the changes. Ordinarily, the sequence of production is set in stone, but let's assume that if because of any purpose – e.g. malfunction of a workstation – it has to be modified. Since every workstation has its own tracking subsystem that is available through Service Registry, we do not have to modify the whole system, only sign – by modifying the default orchestration set – that the tracking information about that concrete assembling step will be provided by another service. In the case of using dynamical orchestration, the Arrowhead framework can match the appropriate systems based on the description of the Service they provide (or aim to consume).

2.2.3 Service interaction between local clouds – the inter-cloud scenarios

Before detailing the specific systems – related to asset tracking – within the architecture, our last requirement has to be discussed. The interconnection with existing and external systems are still defined blurry, but with the help of Arrowhead it can be clarified, at least as much as we have to deal with it in the Local Cloud. Basically, there are two possible classes of scenarios: When a connection has to be established (i) with another (similar) Local Cloud or (ii) a quasi-unknown outer system about which limited information is available. In the first case the other Local Cloud might belong to the another factory site or facility of our company with which we want to share information or resource. In the second case, one of the company's customers or suppliers runs the outer system, but a Local Cloud can belong to them as well.

Inter-connection with clients or partners allows us to track assets – even if it is within their factory site. The data collected this way can be used to aid maintenance service or getting information about the quality of base materials, that will be utilized during manufacturing.

Arrowhead handles both situations in different ways. It offers inter-cloud servicing via the Gatekeeper module, which means another Local Cloud can reach services in the other one. It is a well-defined scenario in Arrowhead, so nobody can consume services within a Local Cloud without permission, which is always checked during authorization. Moreover, Arrowhead offers secure communication between Local Clouds provided by the Gateway module. In the case of connecting with an outer entity, there is no appropriate method to receive data directly, since a Local Cloud is usually a closed environment.

2.3 Main elements and their interworking: Digtal Twin and RTLS

While the overall architecture meets some basic requirements through inheriting Arrowhead features, but to fulfill all of them, special modules of the system have to be discussed: Digital Twin and RTLS. These are only special in the sense that they provide the functionality of automated asset tracking, but they do not have a key role in the Local Cloud. It is important to emphasize that both of them are classes of services and not instances of them – i.e., multitude of RTLS and Digital Twin instances can exist in the same Local Cloud. The designed overall architecture which is presented in Figure 2.4 consists of:

- RTLS systems – This module provides real-time position information of assets that are identified by using RFID. Each workstation has its own RTLS system in this use case.
- Localization Core System (IPS)[1] – This stores the configuration of each RTLS systems – the absolute location of reference points – and transforms the measured position data into absolute location by using trilateration algorithm.

[1]The name IPS comes from the previous phase of the project, but it was kept.

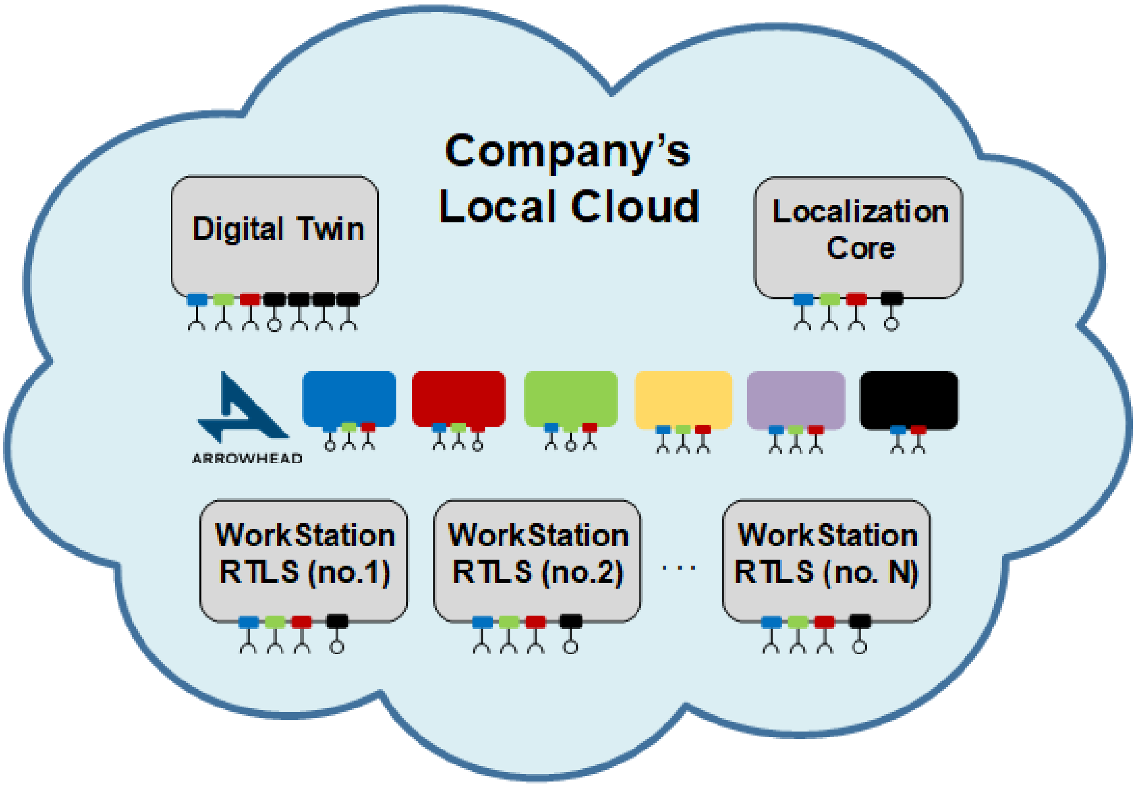

Figure 2.4: *Conceptual elements of the Local Cloud for the asset tracking use case*

- Digital Twin module – A real-time copy of the physical asset, in this case it is used for processing received data from RTLS systems.
- Mandatory cores systems: Orchestrator, Authorization System and Service Registry – These systems handle matchmaking in the Local Cloud and interconnect different systems that use each other's services in an authorized way,
- Supplementary core systems: Gatekeeper and Gateway – These modules enable inter-cloud orchestration between Local Clouds so they provide interoperability within the Supply Chain.

2.3.1 RTLS architecture

The first detailed module is the **RTLS** system that provides information about the location of the asset. In this use case, each workstation has its own RTLS system, therefore the asset tracking system do not cover the whole factory site – only areas where the assets can be found during manufacturing. This means that the whole production line is traceable by multiple numbers of RTLS systems. The operation of these systems can divided into three tasks, namely: (i) identifying the asset, (ii) calculate its relative position within the system, and (iii) determining its location by using localization core system (IPS). The existence of the third task is optional: each of them can have their of own localization core systems or they can use a common one in the Local Cloud. On one hand it is advantageous to use a common resource because all configuration (mostly the absolute position of reference points) is found in one place, on the other hand it can turn into a mess if there are to many workstations are to be handled. Furthermore, the chance of overloading increases if each RTLS uses the same service.

Positioning in these systems uses a ToF[2] based architecture – where measuring distance is based on propagation time of messages – i.e. the Anchors (fixed reference points) do not require time synchronization. Each tracking system of a workstation consists of four *Anchors* and one *Tag*, where the tag represents the asset. Since an asset moves between workstations and other locations, equipping each asset with a tag wouldn't be effective. Therefore the a Tag is attached to the windlass of a chain-hoist and the corresponding workstation. This solution does not have any effect on asset tracking, because within a workstation only the chain-hoist lifts and moves the asset – so its windlass is always above of the asset. A schematic figure of a workstation is shown in Fig 2.5.

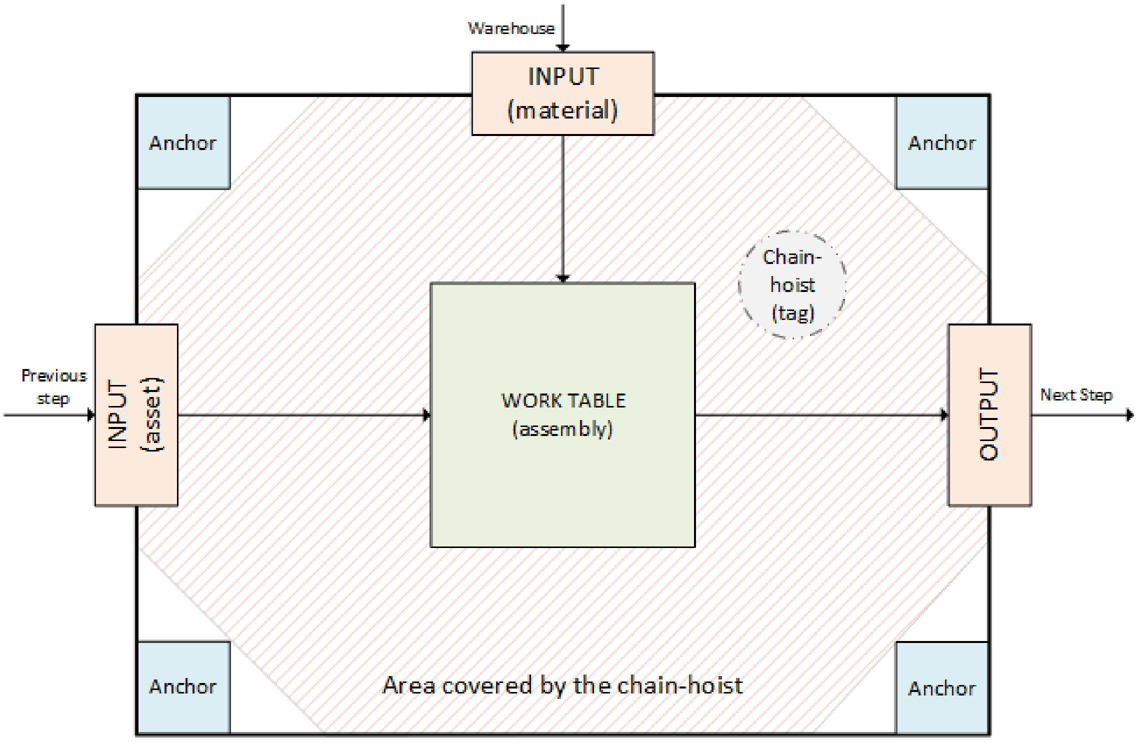

Figure 2.5: *Schematic figure of a workstation with RTLS*

The RFID-based identification process follows a scheme that is closely related to positioning. A piece of base material or an asset is equipped with an RFID Tag[3] and when it arrives at the workstation the RFID reader – which shares its controller device with the UWB Tag – polls the Tag. Since identification also has to be automated, reading should be triggered by an event. To realize this function, the so-called geo-fencing method will be used, which allows us to utilize location based services. In section 2.1 the layout of a workstation was introduced in a nutshell. As described previously, there are three priority points within one

[2]Time of Flight
[3]Not to be confused with UWB Tag

assembling station which the asset must visit: input stage, assembling and output stage. In these positions we have enough time to read RFID tags, moreover these are the only points where the composition of the asset can change, so it can be tracked, which parts are installed on the product in the different phases of the actual task. To trigger the reading by geo-fencing, the area (and boundaries) of different positions have to be pre-defined in the localization core system. Thus, the position information is attached to the individual assets because they go hand in hand.

2.3.2 The Digital Twin module

The other important module in this architecture is the so-called **Digital Twin**. The term Digital Twin has multiple different definitions in existing literature, some of them are related to a specific usage while others only emphasize its characteristics. A general definition introduced by [9] is: *"The concept of using a digital copy of the physical system to perform real-time optimization is often referred to as a Digital Twin."* and *"The vision of the Digital Twin itself refers to a comprehensive physical and functional description of a component, product or system, which includes more or less all information which could be useful in the current and subsequent lifecycle phases"*. In this case Digital Twin refers to a virtual copy of an asset based on gathered information. Nevertheless, this paper aims implementing only an Industry 4.0 compliant Asset Tracking system. The further vision of the Digital Twin was adapted as an extendable form of storing product-related dataset, i.e. it offers a flexible basis for further improvements within such a use case. This paper always emphasizes whether it is talking about the term *Digital Twin* or the *Digital Twin module* – the latter is a service in the Local Cloud.

From our point of view, the Digital Twin module is an integrated storage and processor that receives data from the RTLS systems and organizes them into an appropriate format to use it in logistics and product management. After further development it has to be able to dynamically update itself by consuming certain services the Local Cloud. For example: if an asset enters an area (workstation) it generates an (enter) event and sends it to the Digital Twin module. Based on the location of the asset, the Digital Twin module consumes the Service provided by the corresponding workstation[4], in order to update its copy about the asset.

[4] In this case, the service provides information about what part has been added to asset at that workstation.

Chapter 3

Hardware and Software Design

In this section the concrete design steps and implementation will be shown in contrast to the previous section, which detailed the scope of the overall architecture and the operation of different modules within the asset tracking system. While some of the requirements appear only at a conceptual level (e.g. providing interoperability with further modules), others are concerned with concrete tasks that have to be realized.

3.1 RTLS Design Considerations

3.1.1 Infrastructure

The infrastructure of the RTLS system meeting the described requirements consists of a few devices: namely, four Anchors and one Tag. As it was discussed, the Tag is placed on the windlass of a chain-hoist, which basically moves the asset to be tracked. The location of Anchors is almost arbitrary, but they have to enclose the tracked area of the Tag. Generally, where there is a direct line of sight between anchors and the Tag; this system can cover a $30^{[m]} \times 30^{[m]}$ area for sure,[1] however the assembly workstation areas are much smaller than that in this use case. Furthermore, the result of trilateration is insensitive to what shape the anchors form, but it is a de facto (unwritten) standard to use the vertices of a square if it is possible.[2]

Regardless its role in the infrastructure, each device has the same hardware: an RFID-enabled, UWB based, automated, wireless tracking module. Each of them are settled in a plastic case, which can be fixed to the wall by screws as it's presented in Figure 3.1.

The hardware itself consists of numerous components, which serves different purposes, namely:

[1]Before implementing the system I did multiple measurements with its prototype where the capabilities of the system were tested. The only question which this test could answer is whether the system is stable and provide accurate position in an environment which is similar to the use case – later I presented that test for Konecranes representatives too. Therefore, all presented values are verified, but not their possible maximums are shown.

[2]Most positioning systems operate with high efficiency in the case of forming a square.

- Espressif's ESP32 microcontroller unit [28] – This is the core of the hardware, which sends commands to other modules and handles the tasks in the system. Moreover it enables communication with other services in the Local Cloud via WiFi.
- ThingMagic Nano UHF-RFID reader module [29] – Its job is identification of assets by polling RFID Tags nearby.
- Decawave DWM1000 [30] – This module is an UWB transceiver that is responsible for determining the position of the asset.
- Power Supply modules: Low dropout voltage regulators to provide stable voltage levels and a Serial-to-USB converter for debugging and updating firmware.

Figure 3.1: *The plastic case that stores the hardware: the RFID-enabled, UWB based, automated, wireless tracking module*

3.1.2 Radio-frequency Design

Radio frequency-related considerations play an important role when designing RFID-enabled asset tracking devices that use wireless networking for both communication and positioning. Fortunately, each technology uses different radio bands, namely: UHF-RFID uses 865-868 MHz, WiFi[3] operates in the 2.4 GHz ISM band while UWB Supports 6 RF bands from 3.5 GHz to 6.5 GHz. It means that the used technologies are compatible, so they do not interfere with each other – which significantly eases the design process. It is worth to note that different RTLS systems will not interfere with each other either, since all modules have adjustable RF output power, so their range can reduced to an appropriate level where the corresponding modules can communicate without jamming other systems.

The module IC used for UWB communication is the Decawave's DWM1000, which integrates DW1000 UWB transceiver chip, antenna, power management and clock control [30].

[3]Most popular and widely used *IEEE* 802.11b/g/n are discussed

It simplifies RF design as the antenna and the associated analogue and RF components are already on the module, therefore the only restriction that affects the design is the size of keep-out area. Its rich functionality enables low power consumption (in sleep mode) fast-response time (communication with host controller via SPI) and compatibility with MAC layer of *IEEE* 802.15.4-2011 standard (LR-WPAN).

Embedded UHF-RFID module ThingMagic Nano is used for identifying assets equipped with RFID Tags. In this case, the RF design consideration are related to the antenna, since the modules does not have one. It requires an antenna with the impedance of 50 ohms for adequate VSWR (Voltage Standing Wave Ratio), which prevents the module to be damaged. The vendor of the module released a document that details how track between the antenna connector – which is an u.Fl connector in this case – and the module has to be designed for matching impedance [31]. These restrictions – that affect the length and the clearance of the track – was kept during the PCB design.

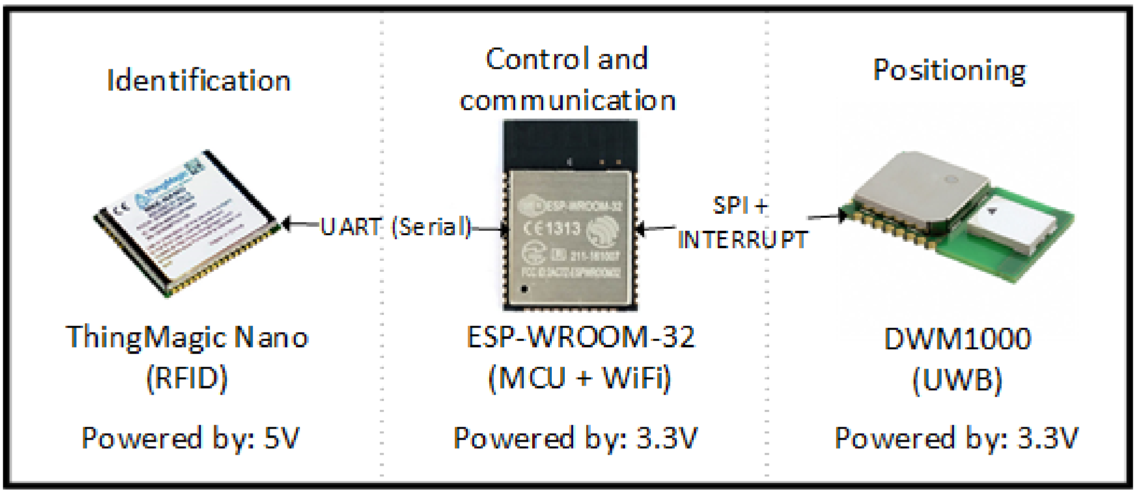

Figure 3.2: *ESP-Wroom-32, MagicThing Nano and DWM1000 – The most important parts of the hardware*

The host controller of the module that handles tasks and provides communication between the Local Cloud is an ESP32 Wroom microcontroller. From the aspect of RF design, it enables using WiFi – Bluetooth Low Energy can be used as well – but, it has the same advantage as DWM1000, i.e. we only have to deal with the keep-out of the module, because it integrates the MCU and the antenna.

3.1.3 Power Supply and Thermal Management

The **power supply** of the device is also an important topic – especially in industrial environment. Most of the modules and ICs require a stable 3.3V supply as usual. To ensure stable voltage an LDO (Low-Dropout Voltage Regulator) is used, which can serve the needs, since the maximum of the delivered current by the source is under 600mA, otherwise multiple LDO should be used. For powering the UHF-RFID module, the direct 5V source has to be used, which is also the input of the LDO. The external power supply that provides the 5V voltage level can be connected to the board via a micro-USB connector, so the device can supplied by the standard residential voltage of 220V. Since in an industrial environment it is not available in every case – for example, the Tag placed on the chain-hoist cannot by supplied this way – the device also has a MAX5033 step-down DC/DC converter, which can transform the input voltage in the range of 7.5V and 78V into 5V. Anchors also can use a LiPo or an Li-ion battery as an external power source, which can be handled and charged by a power path management module.

The designed **PCB** of the device contains a *Serial to USB* converter module as well, in order to ease debugging and programming. As it can be seen in Figure 3.3, the hardware has three mounting holes by which it can be settled and fixed into the plastic case. It is important to take these holes into account during RF design, since the fixing screws can be made of metal that affects the radio frequency communication.

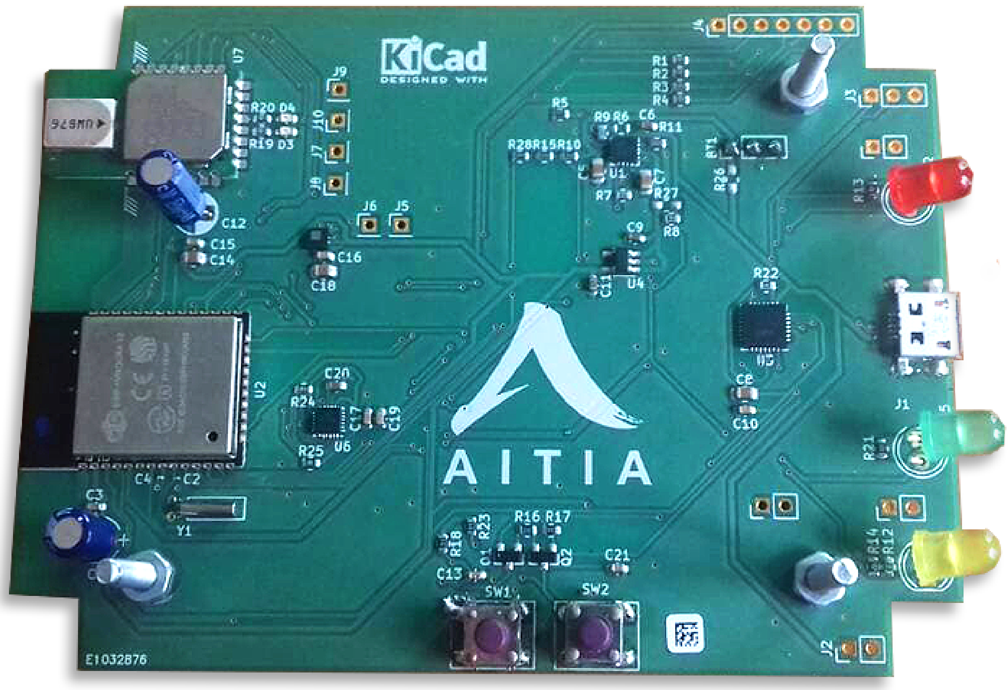

Figure 3.3: *A previous version of the hardware which was designed by using open-source tools*

This RFID reader module is special in its power supply needs, and in the topic of thermal management – hence it deserves a separate paragraph. According to its datasheet, the power supply can vary between 3.3 and 5.5 VDC, but the Output Power accuracy is questionable, and the current draw is much higher at lower voltages. Therefore, in this use case it is essential to have a stable power supply of at least 5V, because it can draw more than 500mA and consumes more than 2W during reading operation. It implies that the modules warms up really fast, which can hinder reading due to the inner temperature sensor that prevent transmission via the firmware if the temperature is out of operation conditions. However, in this use case the duty cycle is low, since the asset stays at least one minute within a workstation, while only three reading operations should be performed. Nonetheless, it is recommended to implement heat sink by thermal vias soldered to the thermal interface of the module, since it stabilizes the operation temperature that provides accurate output power.

3.2 Software Design Considerations

The overall software of the RTLS system is based on the ESP-IDF[4] which is FreeRTOS based high-level framework for ESP32 modules. This gives numerous libraries, easing the development of the firmware. The libraries include:

- lwIP, which is a full TCP/IP stack,
- a HTTP client library,
- cJSON, which is a lightweight JSON parser,
- upper-level GPIO, I^2C and SPI drivers

– just to mention the most important ones.

3.2.1 Real-time Positioning Software

The software architecture takes the advantages of the libraries and the operation systems so the different functions have been organized into tasks with different priorities. Since most of them are time-dependent – such as positioning –, a high accuracy timer is responsible for precise timing.

To reduce power consumption, the MCU can put some modules into sleep mode, e.g. the RFID reader, but with an additional IMU (Inertial Measurement unit) it can be made more efficient. The IMU is set to detect if the Tag has not been moving for a while, and informs the controller. The MCU commands all modules – including itself – to switch into sleep mode. The IMU also detects if the Tag is moving and wakes the controller up by sending an interrupt. This power management scheme can be very useful if the Tag has to be supplied by a battery for any reason.

The most important part of the firmware is the positioning task, which involves using the API of DW1000 that provides a high-level driver to the module since it receives commands and data via its SPI interface. The RTLS uses a ToF based architecture, so the Anchors and the Tag are not synchronized. The scheme is the following: (i) the Tag sends a beacon to discover the anchors which respond to it, (ii) the Tag stores the visible anchors and execute an SDS-TWR (Symmetrical Double-Sided Two-Way Ranging) with each of them,

[4]IoT Development Framework

(iii) finally they organize the measured distances into an appropriate format and pass them to another tasks which handles network communication. This task is executed in every second, periodically. To reduce the load of MCU, an effective MAC filtering provided by DW1000 is used in UWB communication. The transceiver only forwards the message to the MCU if the address field in the MAC header (defined in *IEEE* 802.15.4) is matching. In this case the DW1000 interrupts the microcontroller to handle this as the highest-priority task.

3.2.2 Communication Within The Overall System

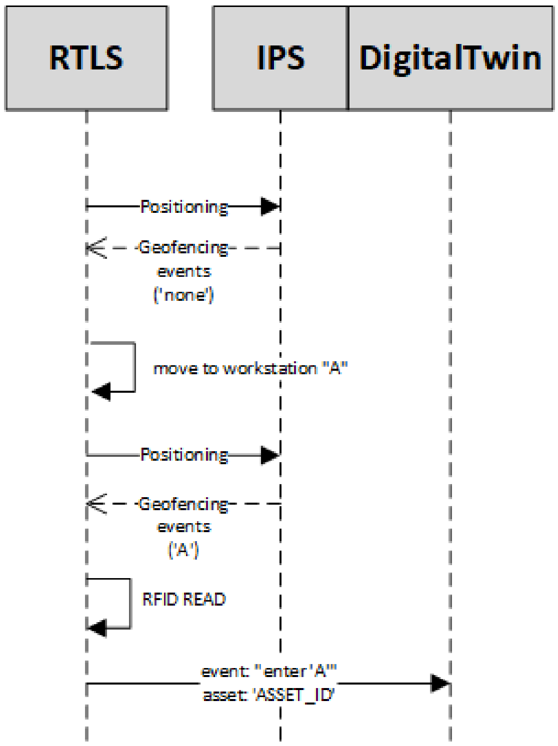

Figure 3.4: *A simplified sequence of messages between RTLS, IPS (Localization Core) and the Digital Twin module*

The network task sends the dataset provided by the positioning task to the localization core (IPS) via HTTP in JSON format. After trilateration calculated the location of asset, the localization core responds with a status code of 200 (HTTP OK). Its payload consists of the coordinates of the calculated location and if the asset is located in one of the predefined geofence area, the response additionally contains the identifier of the geofence and an action code that commands the device to preform a task – in this case only the "RFID READ" action is used. The Tag commands he RFID module to read the RFID Tags nearby and attach the result to the dataset. Finally, the dataset will be forwarded to the Digital Twin module for further processing. The whole sequence of message can be seen in Figure 3.4. In the demo version of this system module, an Arrowhead compliant EventHandler used a "standard" interface to the Digital Twin module.

The Digital Twin module parses the incoming dataset and updates the digital copy of the asset. All stored information have a timestamp so the location and progression of a product is fully traceable. In this phase of work the Digital Twin module does not have any graphical interface that can display the information about the assets: the relevant data can be queried into a log file.

3.3 Validation and Demonstrations

Although the complete validation of the use case is still in progress, this section presents the applicability of the developed system in the current state of the project. Since the Digital Twin module is still under development, that part of the system can not be tested in an integrated manner. Instead of this, we focus on hardware related issues e.g. scalablity and robustness of the RTLS system.

3.3.1 Length of the measurement cycles

Since the RTLS system uses a ToF based UWB architecture, during one measurement the Tag transmits two messages and also receives two of them. One measurement cycle contains four measurements (with the four anchors) which implies that the Tag has to process 16 messages within a measurement cycle. In this use case the measurement cycles are one second long, because it provides the required traceabilty. Therefore if the Tag does only measurements and sends them to the localization core, the process time of a message has to be less than 62,5 ms. The ESP32's CPU can be set 80, 160 and 240 MHz, which means that if it is set to 240 MHz, a message should be processed and forwarded within 15 000 000 CPU cycles. Since the DWM1000 UWB transceiver uses MAC filtering, the MCU does not process any unnecessary message. However, other tasks affects the length of processing, all of them are triggered and short (an RFID reading takes approx. 20-25 ms) – mostly when it enters a specific area, but in this case the asset does not move so the length of a measurement cycle can be expanded to two seconds.

3.3.2 Localization accuracy

The accuracy of the positioning method was measured in different circumstances, moreover a stress test was done, which simulated if the Tag is flooded with messages on purpose – e.g a DOS type attack in order to make localization impossible. Since it was a team effort, and the test case was organized and executed mostly by Gergely Vida, the results of the measurements are detailed in his paper [32].

The test environment was set outside, and the tracked area was larger than in the hoist-chain use case. The results of measurements are presented in Table 3.1. It is clearly shown that the accuracy of the system varies between 15 and 35 cm, which can be improved, but already fulfills the requirements of the use-case.

Table 3.1: *Accuracy of the RTLS system [32]*

Anchor ID	Distance from Tag [cm]	Average of measured distances [cm]	Average accuracy [cm]
Anchor no. 1	1143	1158,6	+15,6
Anchor no. 2	803	796,5	-16,5
Anchor no. 3	2641	2605	-36
Anchor no. 4	436,4	428,8	-7,6

3.3.3 Power consumption

Power consumption of the module was also measured, because the standard residential voltage is not available in every case. In such circumstances, the Anchor points has to be supplied by an external battery. Theoretically the Tag can also be supplied this way, but in this use case the chain-hoist can provide the appropriate power source. Table 3.2 presents the results of a short measurement, where the voltage across a shunt resistor of 1.1 ohms was measured. The state represents time intervals in which the voltage was constant.

Table 3.2: *Power consumption of the ESP32 during executing different tasks*

Voltage Stages	Measured interval [ms]	Measured voltage [mV]	Calculated current [mA]
State 1	620	60	54,54
State 2	3000	125	113,63
State 3	400	165	150,00
State 4	300	125	113,63
State 5	730	150	136,36
State 6	100	125	113,63
State 7	280	55	50,00
State 8	60	125	113,63
State 9	960	55	50,00
State 10	40	125	113,63
State 11	320	155	140,90
State 12	280	60	54,54
State 13	2000	130	118,18
State 14	40	160	145,45
State 15	100	130	118,18
State 16	40	60	54,54
State 17	80	130	118,18
State 18	160	60	54,54

It is worth to note that the measurements contain the power consumption of the ESP32 during using WiFi connection, but according to DecaWave's TWR Calculator, the consumption of DW1000 is negligible – in worst case it is 200 mA for 310 μs in every 62,5 ms. Based on this, the expected battery with a capacity of 2600 mAh is 25 hours. However, it implies that the battery has to be changed per a day by a worker, it is still a more efficient solution than manual tracking of assets.

3.3.4 Demonstrations

The localization part of the design and implementation is already working in Tenneco Inc. Kecskemét; and it also has been demonstrated during the Productive4.0 workshop in Helsinki, at the Konecranes sites.

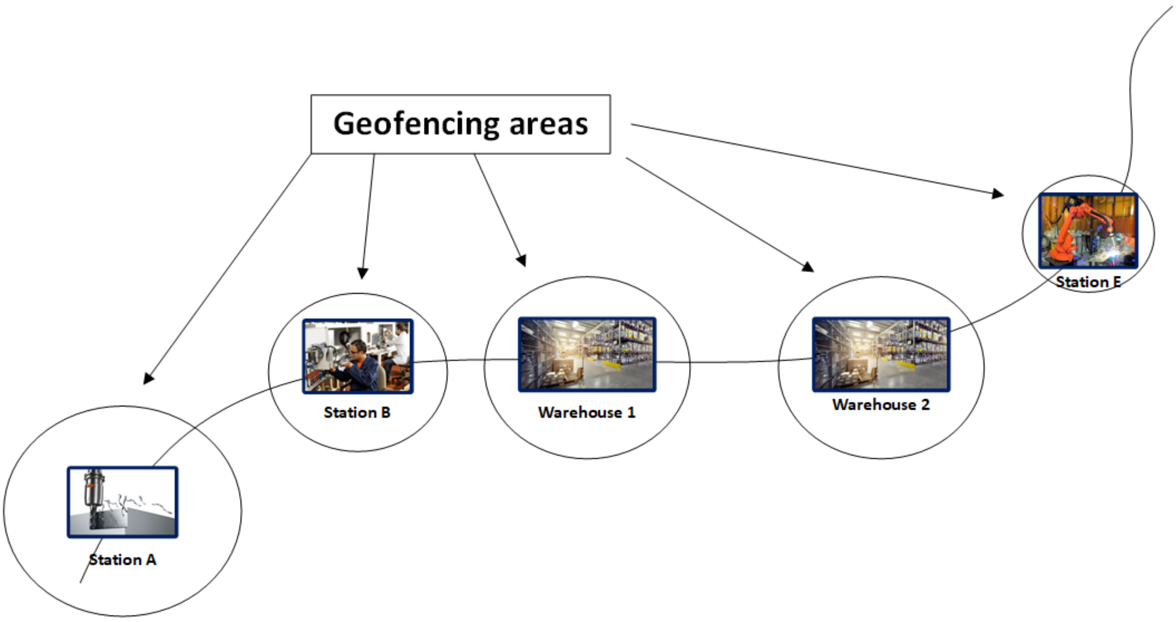

Figure 3.5: *Geofencing areas visited by the asset during the demo case*

The complete, integrated solution described here will be fully demonstrated to the international audience during the Bilbao meeting of the Productive4.0 project (27-29 November), related to the workstation tracking and value chain integration use-case.

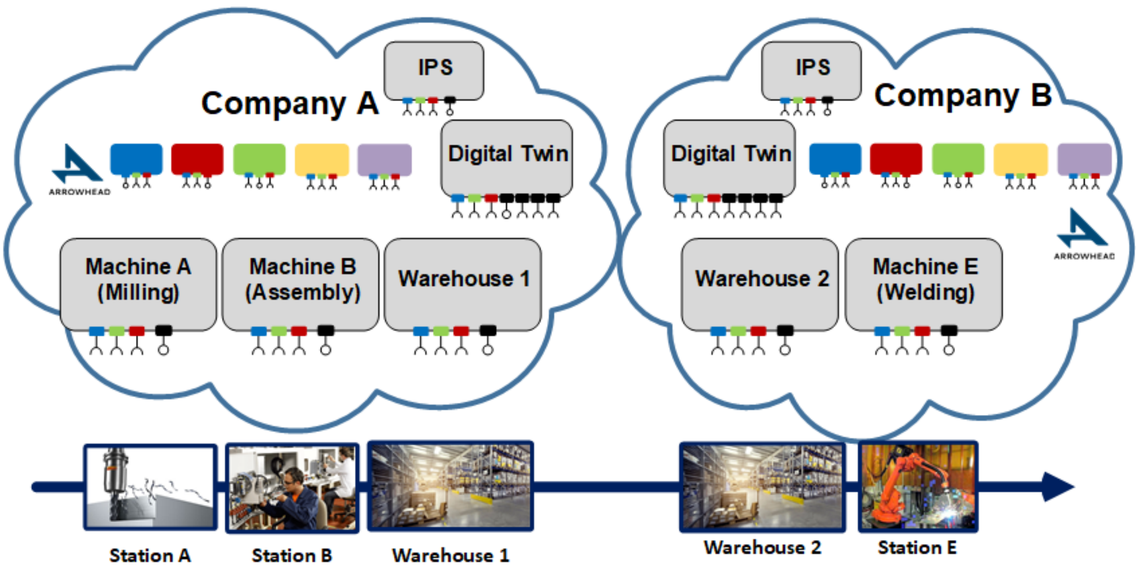

Figure 3.6: *The test scenario is based the inter-cloud communication between two Companies' Local Clouds*

In this demo session the asset will visit five geofancing areas (Station 'A', 'B', 'C' and Warehouse '1', '2') that are located in two RTLS systems as it is seen in Figure 3.5. These systems belong to different companies and both of them run a Local Cloud which is presented in Fig. 3.6. Within a Local Cloud a Digital Twin module can be found as well as an Arrowhead compliant EventHandler – which acts as an interface between RTLS systems and EventHandlers.

The asset starts its lifecycle at Station 'A' where its digital copy will created in the Digital Twin module. Each time the asset enters an area, it will send an "enteredArea" event to the EventHandler which forwards it to the Digital Twin module. The Digital Twin module consume a service provided by the current area and stores the log message. If the actual work-step is done, the asset leaves the area and generates an "leftArea" event, which will also be forwarded to the Digital Twin module by the EventHandler. When the asset leaves Company A and enters Company B the local Digital Twin module fetch the data from the other one by inter-cloud orchestration. The overall sequence of messages can be seen in Figure 3.7.

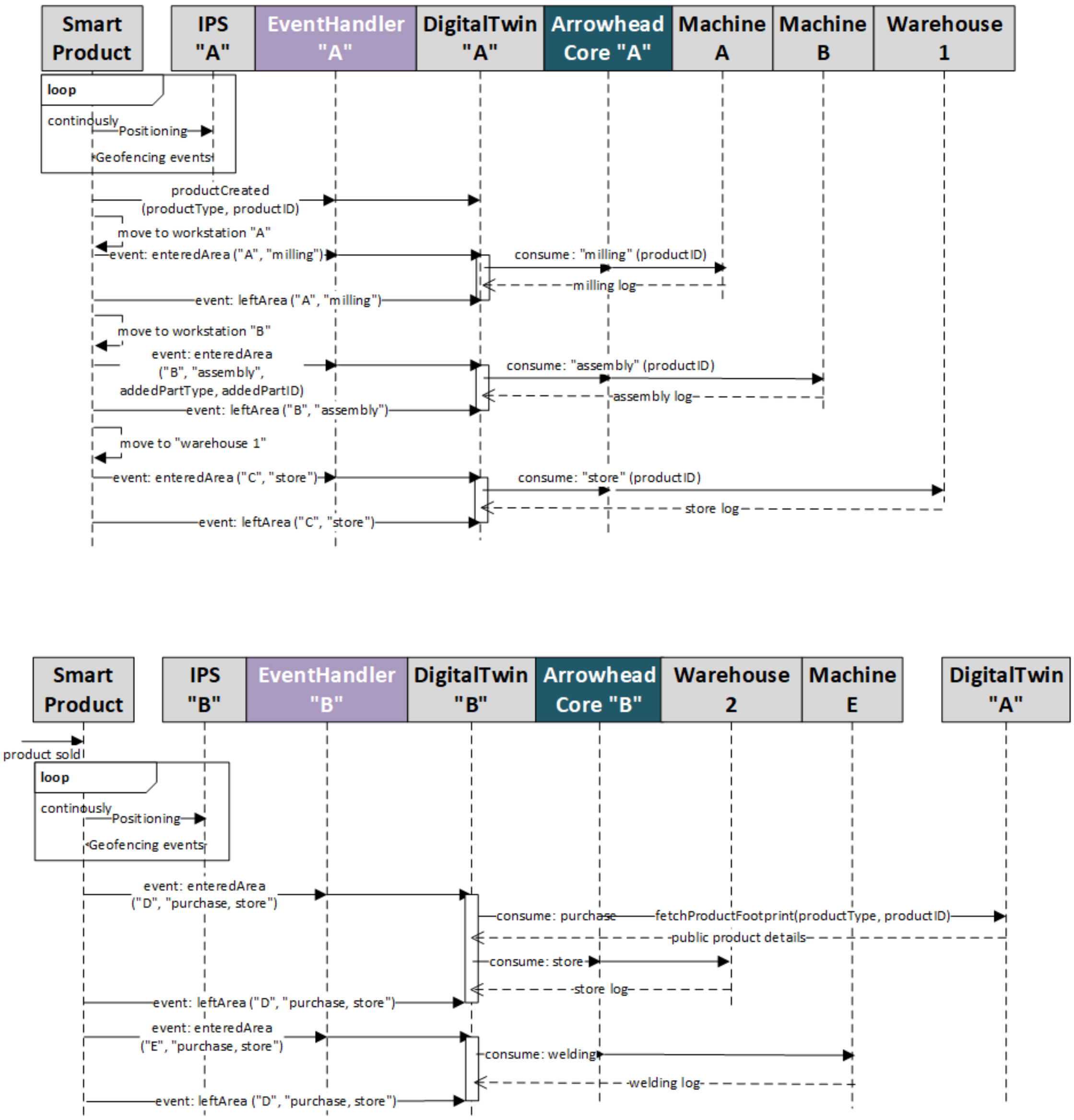

Figure 3.7: *Sequence of messages during the Demo case*

Chapter 4

Summary

Asset Tracking, Digital Twin and in general, the digitization of production lines pose numerous, great opportunities. The previous chapters detailed how these technologies improve Supply Chain Management and within that, Logistics. In practice there are many domains where full traceability of products – even based on this presented system – can be used for various purposes of optimization.

However, most of the requirements and objectives defined in Section 2.1.2 are fulfilled, some functions should be enhanced and a few ones have not been realized yet. The latter are mostly related to the Digital Twin module that is a simple receiver and data parser unit in the current state of project. In general, the Digital Twin concept is going to play a key role in digitized industry as a complex processing module providing highly granulated real-time information about assets. Therefore, beside the general summary of my main contributions to this field, an upcoming "roadmap" section presents concepts for future developments regarding the Digital Twin module as well as utilizing the data provided by the implemented RTLS systems.

4.1 Main Contributions

This section provides a conclusive summary of this paper, emphasizing the contributions and the possible impact. The main goal of the paper was to fulfill the visions that were described in the Introduction and create an asset tracking system which fits into the concept of Industry 4.0, namely: an automated, wireless and – relatively – cost-effective solution that is able to connect the CPPS (Cyber-Physical Production System) with the SCM, Logistics, and ERP systems. The concept of the Arrowhead Local Clouds, together with the inter-cloud communication enables these interactions; although certain Application Systems, such as the Digital Twin Module, and the RTLS had to be specifically planned and implemented.

My main contributions to this work were the following:

- I outlined the concept of an Asset Tracking system as a solution to minimize human interaction within an industrial use-case, namely production of chain-hoist. In order to have domain experts involved, I defined the requirements of these systems by involving the representatives of Konecranes. Moreover, a previous, demo version of the RTLS system (that does not yet full integration with the Digital Twin module) was presented.

- Based on the requirements and the original concept I designed the architecture of the overall system and the individual modules, mainly the RTLS systems which is responsible for real-time tracking. Later, this architecture was refined in an Arrowhead compliant manner.

- I presented the concrete and detailed design of the RTLS system including the methods of localization and identification, the infrastructure, the hardware and the embedded software. It was emphasized what design considerations were used in order to optimize radio-frequency operations and provide appropriate power supply and thermal management for the board.

- I designed and implemented the embedded software which is the core of the automated tracking system. This software controls external modules, handles UWB based ToF positioning method – which provides highly granulated traceablity – as well as WiFi communication. The software also provides automated identification of assets, by using the RFID reader module and geo-fencing areas – provided by the IPS Core System.

- In a joint work with G. Vida [32]: we tested the implemented RTLS system and measured the accuracy of the positioning, which is adequate – its average accuracy is well within 20 cm but in some cases it is less then 15cm. This accuracy makes it possible to use geo-fencing based identification efficiently.

4.2 Roadmap for Future Developments

In this paper a general definition of Digital Twin was used, however there are more specific ones, emphasizing its usage e.g.: *"(Digital Twin is) a dynamic virtual representation of a physical object or system across its lifecycle, using real-time data to enable understanding, learning and reasoning"* [33]. This sentence points out that Digital Twin is not only a real-time copy of asset, but also a huge dataset, which can be used as an input for analytics, machine learning and many others.

4.2.1 Smart Maintenance

A previous project that I participated in – namely, MANTIS [34] – aimed to create a proactive service maintenance platform build upon CPS-s. It offered a highly optimized maintenance management based on finding root causes of deterioration of assets and estimating their remaining useful life, but it was challenging to use some advanced solution – e.g. Artificial Neural Networks (ANN) based estimations – in practice, due to the lack of the otherwise required, huge amount of data. With the concept of Digital Twin, these data reliant tools can be used efficiently to provide highly optimized maintenance solutions in an IIOT based automated environment.

In this case the development of Digital Twin module includes preprocessing and storing the gathered information, forwarding it to ANN and storing the estimated output values too. In such a scenario the Arrowhead framework can also provide efficient, distributed resource management in the Local Cloud if the ANN is running locally instead of a cloud service (e.g. AWS[1]). Of course, this solution requires monitoring the asset, which can be resolved by modifying the existing RTLS system – which is straightforward – or equipping the assets with smart sensors that can provide higher level of granularity.

4.2.2 Delivery Systems integration

Arrowhead handles inter-cloud orchestration in a secure way to enable interoperability between different stakeholders of Supply Chain Network, although it does not mean that the participants fully trust each other. However, the existing solution inhibits unauthorized entities from accessing services – and indirectly data – but there is no guarantee of trustworthiness, i.e. validity of data. In this case the consumer of a product can provide false information about the asset that can make the Digital Twin less useful.

Smart Contracts – i.e. through Blockchain integration – offer an efficient solution to avoid these pitfalls [35]. In this case the "transaction" (data exchange) is stored in a distributed ledger permanently. Of course, a participant can provide false data, but it is permanently stored as the proof of falsification, which is a legal case. This concept is really simple and efficient in situations where the fact of falsification is obvious, e.g. delivery systems – in which every step from producer to customer is traceable. This can be adopted in other cases e.g. providing false monitoring information, but it requires further design considerations.

[1]Amazon Web Services

Acknowledgement

I would like to render my sincere thanks to my supervisor, Dr. Pál Varga for his guidance and suggestions. I would also wish to express my gratitude to Csaba Hegedűs for his expert advice which has been invaluable throughout all stages of work.

Bibliography

[1] R. S. Kshetrimayum, "An introduction to UWB communication systems," *IEEE Potentials*, vol. 28, no. 2, pp. 9–13, 2009.

[2] R. Want, "An introduction to rfid technology," *IEEE Pervasive Computing*, vol. 5, pp. 25–33, 2006.

[3] J. Delsing, *IoT Automation: Arrowhead Framework*. CRC Press, 2017.

[4] "About Konecranes." `https://www.konecranes.com/about-konecranes`, 2018.

[5] "Sap extended warehouse management." `https://www.sap.com/products/extended-warehouse-management.html`, 2018.

[6] M. Cao and Q. Zhang, *Supply Chain Collaboration: Roles of Interorganizational Systems, Trust, and Collaborative Culture*. Springer Science & Business Media, 2012.

[7] S. E. Sampson and C. M. Froehle, "Foundations and implications of a proposed unified services theory," *Production and operations management*, vol. 15, no. 2, pp. 329–343, 2006.

[8] Y. LV, L. TU, C. K. M. LEE, and X. Tang, "IoT based Omni-Channel Logistics Service in Industry 4.0," in *IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, 2018.

[9] R. Söderberg, K. Wärmefjord, J. S. Carlson, and L. Lindkvist, "Toward a digital twin for real-time geometry assurance in individualized production," *CIRP Annals. Manufacturing Technology*, vol. 66, no. 1, pp. 137–140, 2017.

[10] B. Scholten, *The Road to Integration: A Guide to Applying the ISA-95 Standard in Manufacturing*. International Society of Automation, 2007.

[11] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 3, 2014.

[12] F. Blomstedt, L. L. Ferreira, M. Klisics, C. Chrysoulas, I. M. de Soria, B. Morin, A. Zabasta, J. Eliasson, M. Johansson, and P. Varga, "The arrowhead approach for SOA application development and documentation," in *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, 2014.

[13] M. W. Maier, "Architecting principles for systems-of-systems," *Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998.

[14] T. Erl, *SOA: Principles of Service Design.* Prentice Hall, 2007.

[15] T. Velte, A. Velte, and R. C. Elsenpeter, *Cloud Computing, A Practical Approach.* McGraw-Hill Education, 2009.

[16] M. H. Valipour, B. Amirzafari, K. N. Maleki, and N. Daneshpour, "A brief survey of software architecture concepts and service oriented architecture," in *2nd IEEE International Conference on Computer Science and Information Technology*, 2009.

[17] C. Hegedűs, D. Kozma, G. Soós, and P. Varga, "Enhancements of the arrowhead framework to refine inter-cloud service interactions," in *Conference: The 42nd Annual Conference of IEEE Industrial Electronics Society (IECON)*, 2016.

[18] C. Hegedűs and P. Varga, "Service interaction through gateways for inter-cloud collaboration within the arrowhead framework," in *5th International Conference on Wireless Communications, Vehicular Technology, Information Theory, Aerospace and Electronic Systems (VITAE)*, 2015.

[19] C. Hegedűs, P. Varga, and A. Frankó, "Secure and trusted inter-cloud communications in the arrowhead framework," in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, 2018.

[20] W. M. Y. W. Bejuri, M. M. Mohamad, R. Zahilah, and R. M. Radzi, "Emergency rescue localization (ERL) using GPS, wireless LAN and camera," *International Journal of Software Engineering and Its Applications*, vol. 9, no. 9, pp. 217–232, 2015.

[21] Decawave, *REAL TIME LOCATION SYSTEMS: An Introduction*, 1.1 ed., 2014.

[22] R. Ye and H. Liu, "UWB TDOA localization system: Receiver configuration analysis," in *2010 International Symposium on Signals, Systems and Electronics*, 2010.

[23] B. Sackenreuter, N. Hadaschikl, M. Faßbinder, and C. Mutschler, "Low-complexity PDoA-based localization," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016.

[24] A. Alarifi, A. Al-Salman, M. A. A. Alnafessah, S. A.-H. M. A. Al-Ammar, and H. S. Al-Khalifa, "Ultra wideband indoor positioning technologies: Analysis and recent advances," *Sensors (Basel)*, vol. 5, 2016.

[25] S. Krishnan, P. Sharma, Z. Guoping, and O. Woon, "A uwb based localization system for indoor robot navigation," *Proceedings of the IEEE International Conference on Ultra-Wideband*, pp. 77 – 82, 2007.

[26] "Productive4.0 - opening the gates to the digital future." `https://productive40.eu`, 2018.

[27] DIN Std., *Reference Architecture Model Industrie 4.0 (RAMI4.0)*. DIN SPEC 91345, 04 2016.

[28] "Esp32 series datasheet." `https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf`.

[29] "Thingmagic nano embedded uhf rfid module." `https://cdn.sparkfun.com/datasheets/Sensors/ID/NANO-final-12Mar2015.pdf`.

[30] Decawave, *DWM1000 UWB transceiver module Datasheet*, 1.7 ed., 2014.

[31] "ThingMagic Nano Design Guide." `https://cdn.sparkfun.com/datasheets/Sensors/ID/Nano_Design_Guide_rev01E.pdf`, 2018.

[32] G. B. Vida, "Beltéri lokalizációs rendszerek kommunikációs és adatfeldolgozási architektúrája," in *BME-VIK Scientific Students' Associations Conference*, 2018.

[33] R. N. Bolton, J. R. McColl-Kennedy, L. Cheung, A. Gallan, C. Orsingher, L. Witell, and M. Zaki, "Customer experience challenges: bringing together digital, physical and social realms customer experience challenges," *Journal of Service Management*, 2018.

[34] "What is mantis?." http://www.mantis-project.eu/about-mantis/what-is-mantis/.

[35] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292 – 2303, 2016.