CAPE GROEP

UNIVERSITY OF TWENTE.

THE DESIGN AND IMPLEMENTATION OF SITUATION AWARE SMART LOGISTICS IN PERISHABLE FOOD TRANSPORTATION

Author

Gilang Charismadiptya

Supervisor

Dr M.E. Iacob
Dr.ir. M.J. van Sinderen

Company Supervisor

Maik Wesselink

CAPE GROEP
Transportcentrum 14, 7547 RW Enschede

SUMMARY

An exception is “any deviation from an ‘ideal’ collaborative process that uses the available resources to achieve the task requirements in an optimal way” [2]. The exception handling process is slow. It may delay the delivery of the product. In consequences, this may cost the customers trust and their money. Also, because the products are perishable, it may get spoilt which will cost the company.

The food industry must consider food safety when transporting products. There are many hazards that can occur during transportation. For example, lack of security, improper handling, improper refrigeration, improper management of transportation units, improper loading practices, conditions or equipment [1]. The risk increases the chance of logistics exception

Prior research has shown many architectures for leveraging the internet of things into the logistics fields. However, most of the architecture purpose is to identify and record the location of the transported objects. However, there are not many architectures available for detecting logistics exception and provides useful information in respond. Existing architecture includes logistics exceptions management, described in [75] which focus on detecting the exception, and EURIDICE [43], which focus on gathering valuable information for logistics operation. Nevertheless, none of the mentioned architecture focuses on the integration of exception detection capabilities and presenting the information surrounding the exception into a real business process.

This thesis attempts to fill the gaps by designing an enterprise architecture for detecting and reacting to logistics exceptions. The enterprise architecture serves as a guide for implementing the concept of situation-aware logistics in real condition. Furthermore, I develop the architecture to serve as the basis for implementing artificial intelligence into logistics exception management for future research. I perform this study by using the design science methodology, which emphasize on stakeholder’s satisfaction.

The result of this design study is, the designed system can fulfil the stakeholder desire. However, some improvements are still possible. For example, the result suggests the addition of the scanning process have a measurable impact on the business process. Though, the scanning process is still within the accepted level, especially when compared to the real loading process that could take an hour to finish. The test subjects notify the customer faster compared to the previous business process.

For future research, the research should focus on how to incorporate artificial intelligence into the architecture. Artificial intelligence could have roles in two things: to tune the exception rules so that it produces more accurate notifications and to analyse the data after the exception happens. Another future work possibility is to perform more validation to the architecture. Thoroughly test the architecture in a real situation will provide new useful insight for improving the architecture. The final suggestion is, to improve the functionality of the architecture. For example, improve the scanning process by incorporating Identification technology such as RFID, which will eliminate the process of manually associating the pallets with the orderlines.

Acknowledgements

First, I would like thanks to my parents, that always supports me in my life and give me direction when I lost.

I would like to thank CAPE Groep, which gives me a place and facilities to work on my thesis. Especially Pieter verkoost that accept and believe in me that I can contributes my thesis at Cape Group. Also to Maik Wesselink and Sebastian Piest as my mentor from the company. They always gave me useful advice and feedback based on their experience and technical knowledge.

I would like to express my gratitude to Dr M.E. Iacob and Dr.ir. M.J. van Sinderen for being my thesis supervisor. Their guidance and feedback help me a lot during my time working on the thesis.

I also want to thank my cats, and all the internet cats I watched during my procrastination, for making me smile and giving me the energy to keep moving forward. Finally, I would like to thank all my friends and loved one for supporting me during my darkest hour working on this project.

Gilang Charismadiptya

Enschede,
Monday, 10 September 2018

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

1.1 SITUATION-AWARE SMART LOGISTICS

Perishable food logistics is the process of planning, implementing and controlling efficient, functional flow and storage of perishable goods, related services and information from one or more points of origin to the points of production, distribution and consumptions to meet customers' requirements. The food industry must consider food safety when transporting products. The top 5 food safety hazards during transportation are lack of security, improper handling, improper refrigeration, improper management of transportation units, improper loading practices, conditions or equipment [1]. Moreover, the company also need to consider the transportation risks, such as road accidents.

The risk increases the chance of logistics exception. An exception is "any deviation from an 'ideal' collaborative process that uses the available resources to achieve the task requirements in an optimal way" [2]. First, the company must identify the type and the reason for the exception. Then, the company must identify the affected products and its customers. Finally, the company can notify the customers and design a mitigation plan.

The exception handling process is slow. It may delay the delivery of the products, and the customers know about the possible delay too late. In consequences, this may cost the customers trust and their money. Also, unaffected products may get delayed by the exceptions. Because the products are perishable, it may get spoilt which will cost the company. Therefore, a perishable food company must react as quick as possible to an exception.

There are many points of optimisation for ensuring quick exception response. This study focuses on two things: automatic exception detection and improving traceability. Automatic exception detection is the capability to detect a possible exception automatically and informs the responsible transportation manager. Having automatic exception detecting capabilities will give the transportation manager time to react, and eliminates the time need to identify the exception type. Exception detection is made possible by leveraging real-time data from sensors, embedded into the products. The definition of traceability is the ability to trace and follow a food, feed, food-producing animal or substance intended to be, or expected to be incorporated into a food or feed, through all stages of production, processing and distribution [3]. By having traceability information, identification of the affected products and its customers can be quick.

Internet of things is a term first coined by Kevin Ashton, the director of the company that created the global standards for RFID [4]. Internet of things is defined more precisely by IEEE in 2015. Internet of Things is a network that connects uniquely identifiable "Things" to the Internet. The "Things" have sensing/actuation and potential programmability capabilities. Through the exploitation of unique identification and sensing, information about the "Thing" can be collected, and the state of the "Thing" can be changed from anywhere, anytime, by anything [5]. The advantage of using internet-connected devices is the user can monitor, and interact with the devices anytime, anywhere.

Internet of things devices now started to become ubiquitous in a wide range of area. For example, in everyday life, amazon echo can help buy everything Amazon offered, a smartwatch can measure the heart rate to detect illness. In the healthcare industry, internet of things devices successfully used to track and monitor the condition of the patient and the condition of medical equipment [6]. The agriculture industry uses IoT devices to monitor the environment and livestock. Factories use it to monitor the health of the production equipment.

I call the information system to provide exception detection and mitigation assistance by using IoT enabled sensors as situation-aware logistics systems. The problem is, currently only a handful of architecture has been described that incorporate the idea into an actual information system. A new system architecture needs to be devised. Thus, the goal of this study will be the design of the architecture of situation-aware logistics.

This thesis is part of the DATAREL project. The project objective is to expand the knowledge of employing data from Internet-connected sensor devices embedded into transportable items by utilising recently developed data processing techniques such as big data and artificial intelligence to detect emergence. The result of the project is expected to improve the resilience, real timeless, efficiency, and dynamics of logistics planning operation. NWO is funding the project [7]. The project is running for four years, starting in February 2018.

1.2 Research goals

The primary goal of this research is to define an architecture and its implementation for capturing and processing IoT data for improving logistics situation awareness. The study formulates the problems into the primary research questions:

What is the suitable design for situation aware logistics system?

To answer the primary research question, first, the study must define what the risks in perishable food transportation are and how they can be measured. The measure is essential to detect the exceptions as exceptions happen when the risks become concrete. Then, we need to know what the stakeholders want from such systems. I will design the architecture from the stakeholder's desire and the defined risks measures. Finally, I need to know how the current business process handles logistics exceptions. The knowledge of the current business process is essential to know which part of the business process needs to be modified to integrate the designed system.

The designed architecture may or may not produce expected effects. That is why the designed architecture must be validated; the study must define the measure of the system performance to validate the architecture.

As a result, the study decomposes the primary research questions into smaller research questions:

1. What are the requirements for smart logistic planning systems to manage the risk?
2. What is the design of the situation-aware logistics system?
3. What kind of alterations are required to integrate the system with the business process?
4. What is the performance of the architecture in reducing transportation risks?

1.3 METHODOLOGY

1.3.1 RESEARCH METHODOLOGY

The study will use the design science research methodology by Roel J. Wieringa [8]. The methodology is suitable for designing an artefact for solving a problem in a context. An artefact is a result that comes from the design activities. A context is where the artefact can effectively solve a problem. The methodology describes a workflow to do design science research, called the design cycle. The design cycle consists of three iterative phases: problem investigation, treatment design, and treatment validation. The diagram below describes the design cycle:

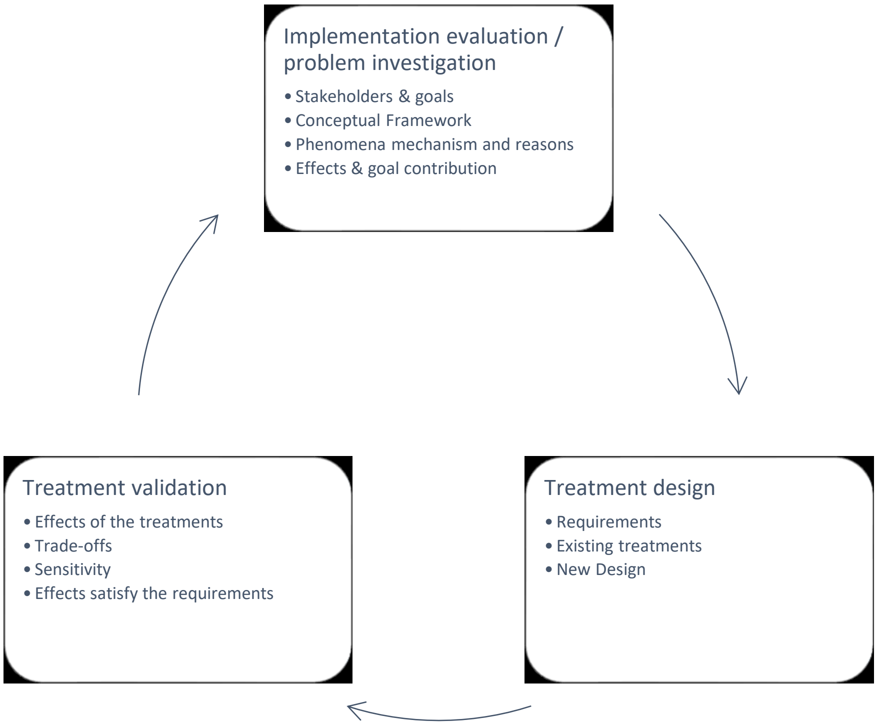

FIGURE 1 THE DESIGN CYCLE [8]

The workflow starts with problem investigation. In this phase, we will define the problems and the context of the problems. The first task is to define the stakeholders and their goals. The stakeholders are the primary user of the system. Therefore, it is essential to know what they want from the system. The next step is to create the conceptual framework. The conceptual framework defines the components that will interact with other components or the context.

The treatment design phase begins with defining the requirements. A requirement is the desire of the stakeholders that commit resources to realise their wish [8]. The stakeholders may not know their requirements. Therefore, the designer must cooperate with the stakeholders to define the requirements together. The result is the contribution argument, which is a prediction that the artefact will satisfy the stakeholder's requirements. There are two categories of requirements, functional requirement and non-functional requirement. The functional requirements are the desired function that an artefact must have. The non-functional requirements are the quality properties of artefacts.

The process of designing an artefact is a cycle because the product from an iteration may be proven unusable after validation. The proposed design comes from the cycle is not the best solution. However, it serves only as a candidate solution. In the second iteration, we may investigate the validation result, and design a better artefact. This study will only perform the first iteration of the design cycle. The design may require several iterations before it finished.

Existing treatments provide valuable insights into the ways the problems can be solved. After analysing the existing treatments by comparing it to the requirements, the designer can decide to improve existing treatments, reuse components of the existing treatments, or create an entirely new treatment design.

The treatment validation objective is to justify whether the treatment contributes to the stakeholder's goals when implemented in the problem context. The study must provide a validation model to validate the treatment. A validation model consists of a representation of the artefact (A prototype) and the representation of intended context. To study validation models, the literature provides a list of research methods: expert opinions, single-case mechanism experiments, technical action research and statistical difference-making experiments. The study chooses to use experts' opinions and single-case mechanism experiments to study the validation model. The study gathers expert opinions by presenting the treatment model (the prototype) to them and capture their prediction of the effects of the treatments.

Based on the methodology described above, the study will follow the steps below to perform the research:

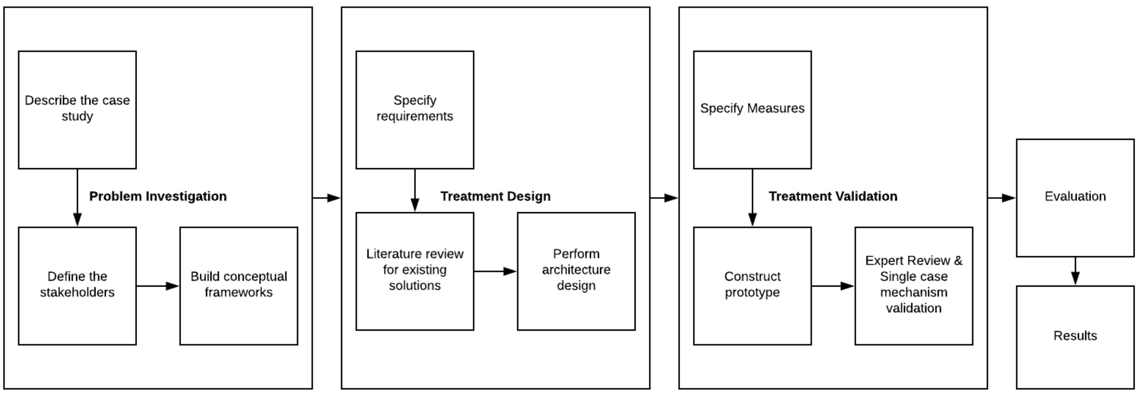

FIGURE 2 THE RESEARCH METHODOLOGY

1.3.2 PROTOTYPE DEVELOPMENT METHODOLOGY

The study must build a representation of the artefact to create the validation model. The study called this the prototype. The validation model artefact of this study is the prototype of the situation-aware logistics system. Situation-aware logistics system is an information system. An information system (IS) is an organised system for the collection, organisation, storage and communication of information. An information system consists of software, hardware and the organisation. This study will use the Scrum framework to develop the software part of the prototype.

Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value[9]. Scrum consists of scrum teams, roles, events, rules and artefacts. The scrum rules regulate the interaction and relationships between scrum components.

The stakeholders in the scrum framework are called the scrum team. It comprises of three stakeholders: a product owner, a development team and a scrum master. The primary role of a product owner is to write the product backlog and prioritise which features are the most important for them. The product backlog is the list of requirements, constructed as stories. A product owner is a single person who captures the desire of the stakeholders.

The development team are responsible for the incremental development of the product. The sprint team need to be small to remain nimble but enough to complete significant work in a sprint [9]. A sprint is an interval in which a product can be incrementally developed. The stakeholders in scrum perform sprint planning to determine the goal of the sprint. We must achieve the goal by implementing the backlog. A sprint goal can be adjusted according to factors such as the evaluation of the latest product increment, the capacity of the team and the past performance of the team.

The scrum master is responsible for promoting, coaching, and supporting the team to implement the Scrum method. The scrum master promotes the Scrum framework to a new member of the team that does not know about Scrum. The scrum master facilitates the product owner to build product backlog. The scrum master can help schedule a scrum meeting such as the scrum planning. The person is responsible for scrum implementation in the organisation [9].

The study is going to use the Scrum framework to build the prototype because it resembles the design cycle of the design science research methodology. Both methodologies exhibit the iterative nature of design development. Both start with the goals of the stakeholders and build the artefact from it. However, the purpose of the methodologies is different. The design science research methodology aimed at doing research, while the scrum aimed at building product. Hence, in this research, the study uses both, design science research methodology for doing the research and the Scrum framework for building the prototype. The diagram below presents the method for developing the prototype.

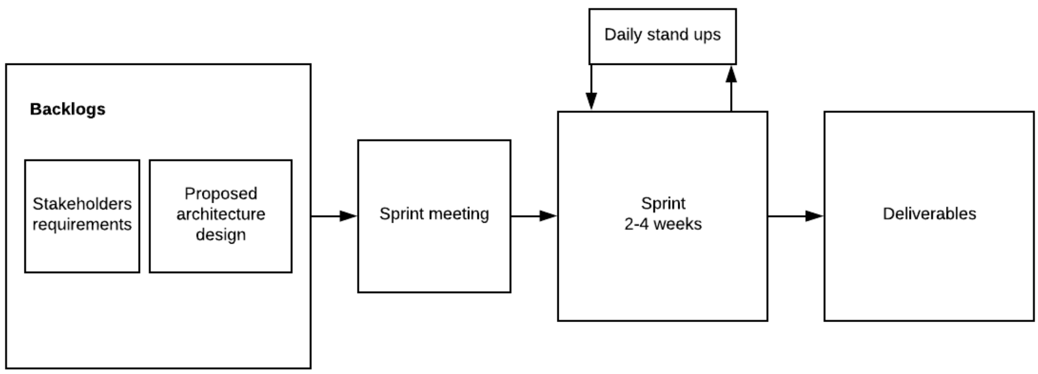

FIGURE 3 PROTOTYPE DEVELOPMENT METHODOLOGY

2 PROBLEM INVESTIGATION

2.1 THE CASE

2.1.1 ABOUT THE PERISHABLE FOOD COMPANY (PFC)

PFC is a food production company that supplies food for the healthcare industry. Their client is ranging from hospitals, nursing homes, and home care in the Netherlands. Aside from the fresh meat product, the company can also deliver groceries to a various healthcare institution.

PFC currently have a production centre branch in a city, in The Netherlands. They also have a new distribution centre in another district in the city. The new distribution centre has a large freezer with an area of 3.850m2 and an expedition hall with a total area of 5300m2. They must transport the product from their production centre to a distribution centre. Currently, they do not track the movement of the product between the locations. Their ERP systems treat both locations as one. An order line identifies the transported products.

Risk such as road accident, thief, unapproved transport of unrelated goods, a cooling system failure that can lead to food spoilage may happen. When that happens, tracing back to what are the affected order line, what causes the problems can be slow. The company needs a system that can provide tracking and risk mitigation can be developed to solve the problems.

2.1.2 PFC PRODUCT TRANSPORTATION OVERVIEW

PFC is composed of many business services. Examples of the services are the production service, customer service, financial service and the logistics service. The logistics service is further composed of many services. The study will focus on the product delivery service. The product delivery service is responsible for product movement from the production centre to the distribution centre, and from the distribution centre to customer.

product transportation function realised the product delivery service. The function consists of several processes and events. The first process is the production centre into distribution centre process. When that process completed, and the products transferred to the distribution centre, a transportation success event will be triggered. The system will notify the transportation manager. Then, the transportation manager will start the transportation process from the distribution centre to the customers.

During the transportation, an exception can occur. Examples of events that can trigger exceptions include road accidents, equipment failure, human error and vehicle failure. The exception will cause a delay in the completion of the process. Thus, the product transportation function has a process to handle the exception. The diagram below shows the overview of the product transportation process.

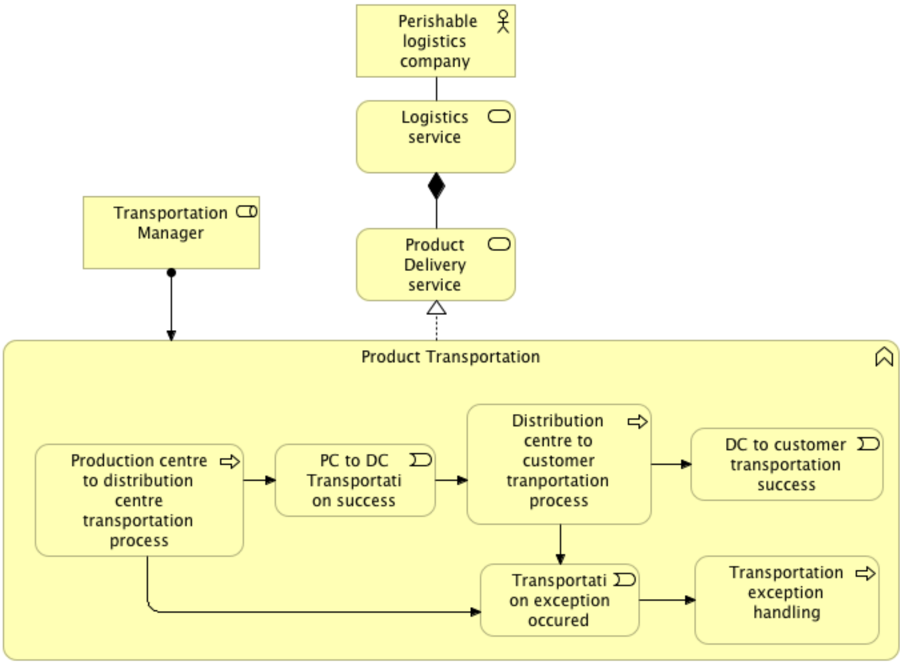

FIGURE 4 PFC PRODUCT TRANSPORTATION BUSINESS PROCESS OVERVIEW

This thesis will focus on the exception handling of the production centre to distribution centre transportation process. The process consists of scheduling, loading, transporting and loading perishable goods. The process is handled by multiple roles, from the transportation manager which responsible for the business function, loading/unloading employees and the vehicle operator. All roles must coordinate their action cooperatively to produce efficient transportation process.

The transportation process begins when the orderlines shipping ready events raised from the ERP systems. The transportation manager will schedule the order lines for shipping, using the picking systems. There are three different picking systems available. The picking systems will determine the loading time, the vehicle, the order lines locations inside the pallet and the pallet locations inside the vehicles. The scheduling process must consider the priority of the order lines, the expiration date, the type of perishable products (For example, the loading employee should place heavier products below lighter products inside the vehicles) and the cooling requirements.

The next step is to load the order lines into the pallets. Currently, the loading/unloading employee performs the process. The employee uses the scheduling data received from the picking systems to load the order lines into pallets correctly. After that, the loading/unloading employee will load the pallets into the vehicle. Then, the vehicle transports the pallets to the distribution centre. The unloading employees then unload the order lines. The production centre to distribution centre transportation process finished here. After the unloading process, the order lines are ready to go through the distribution centre to the customer transportation process. The diagram below depicts the PC to DC transportation process in more detail.

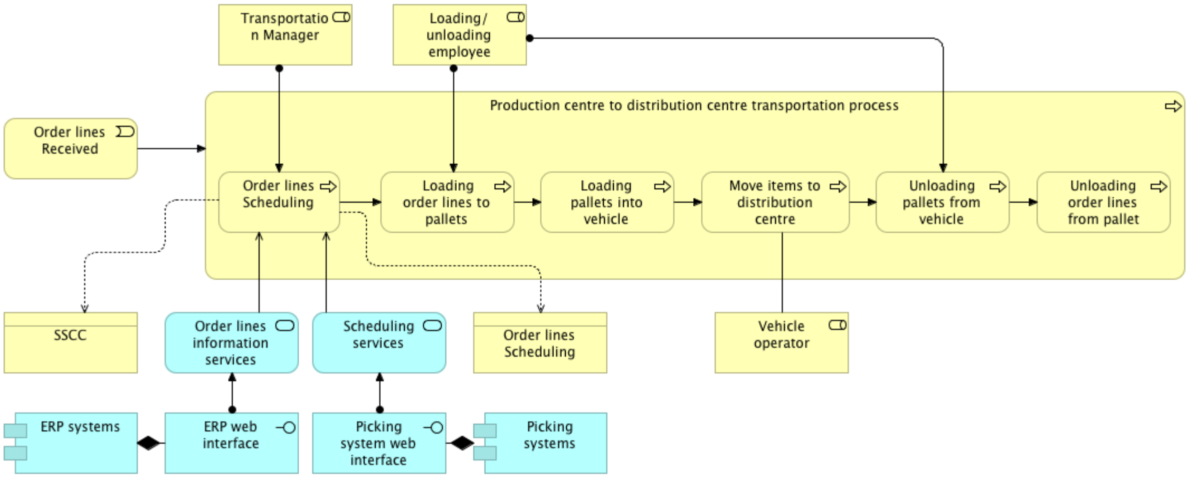

FIGURE 5 PRODUCTION CENTRE TO THE DISTRIBUTION CENTRE TRANSPORTATION PROCESS

PFC is currently doing a lot of manual works to handle process exceptions. Their transportation exception handling process starts by gathering information about the exceptions. The information includes the types of exceptions, the location of the exceptions, the reason the exceptions occurred, the standard process to handle the type of exceptions and the responsible stakeholder. Then, because the order lines inside the vehicle not tracked, manual workers must unload the pallets from the vehicle, then unload the order lines from the pallets. After that, the items conditions must be identified and checked. Also, the transportation manager must identify the customers. Then, the transportation manager can create a mitigation plan based on all the gathered information.

The mitigation plan includes the necessary information to solve the exceptions. The mitigation plan contains a list of saveable products. At this point, the transportation manager should inform both the unaffected and the affected customers of the delay. Figure 6 depicts the whole transportation exception handling process.

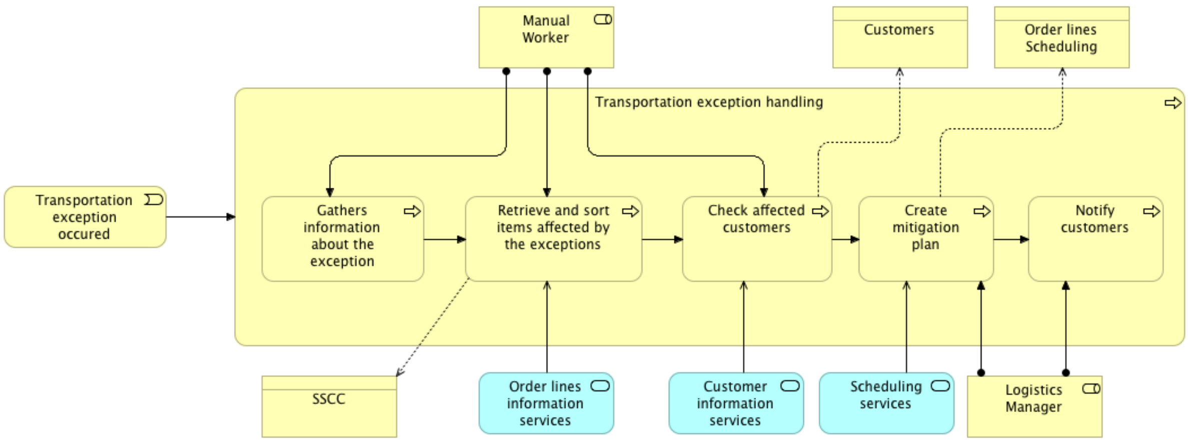

FIGURE 6 TRANSPORTATION EXCEPTION HANDLING PROCESS

From Figure 6, we can see that the PC to DC transportation process and the exception handling needs order lines data, scheduling data and customers data. The company use SSCC or serial shipping container code to identify order lines. The ERP systems manage the order lines and the customer's data. The ERP systems provide a web interface to access the data. For scheduling, the picking systems managed the information. The web interface provides logistics manager information to the systems. Figure 7 shows the various applications and data types used in the business process.

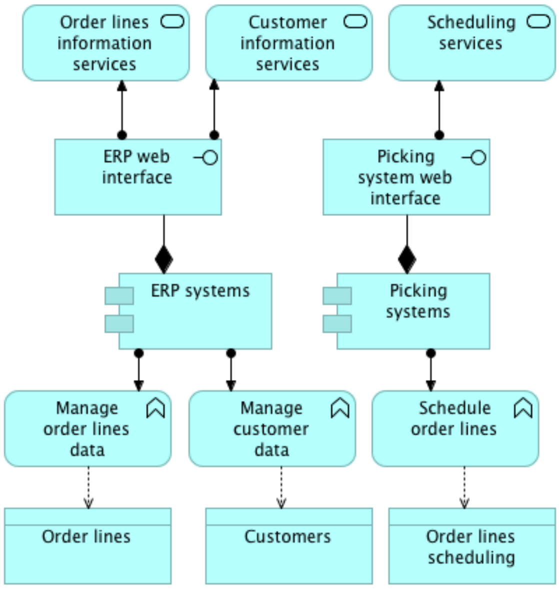

FIGURE 7 PFC APPLICATIONS LAYER

2.1.3 *PROBLEMS*

Currently, The ERP system that PFC use considers the production centre and the distribution centre as one. The condition possesses problems because of the untracked movements of their products. The untracked movements create problems described below:

2.1.3.1 Slow response to accidents

An accident is an unexpected event which causes loss, a decrease in the value of resources or an increase in liability. Examples of accidents are traffic accidents, thieves, loading/unloading accidents, equipment failure. Because the movements between the production centre and the distribution centre not tracked, the company cannot know which order lines are affected by an accident straightaway. The company needs to provide a worker to inspect and trace the products inside the vehicle sequentially. The manual tracing process is time-consuming. The tracing process cause mitigation decision to be delayed. The result is the timeliness of the product delivery may be affected. The delay may also cause end client to be informed too late about the accidents, affecting their business.

2.1.3.2 Partially transparent supply chain

Transparency is the information of product origin and transit before arriving at the customer's hands. The lack of information on the transportation between the production centre and distribution centre means the supply chain is not transparent. Risks can arise from such a situation. For example, a colony of bacteria infects one of the vehicles, and lethal food poisoning occurred. The company cannot know that the product is affected because of the vehicle. The company cannot correlate the vehicle with the products. If the company fails to find the source, the company risk the same thing happened again, can be faced with a legal challenge and customer loss.

3 REQUIREMENTS ANALYSIS

A requirement is a property of the treatment desired by some stakeholder, who has committed resources (time and money) to realise the property[8]. Therefore, first, we need to identify who the stakeholders are. Then, we must find their desire. I will identify the stakeholder's desire by performing a literature review and conducting an interview with the stakeholders. After that, the desire must be prioritised based on the importance of the person towards the projects. The stakeholder's desire is not the requirements for designing the artefacts. The desire is the goal of the system. The designer of the system must specify the requirements to achieve that goal.

To summarise, I use the following steps to determine the architecture requirements:

1. Stakeholders analysis.
2. Architecture requirements analysis.

3.1 STAKEHOLDERS ANALYSIS

In this section, I identify and describes the stakeholders of the case study. A stakeholder of a problem is a person, group of persons or institution who is affected by treating the problem [8]. A design is generated to solve the stakeholder’s problems. That is why the stakeholders’ analysis is vital. I use their desire as the basis of the requirements. All stakeholders have its desire and goal for using the system. The effect of treating a problem may not be beneficial to all stakeholders. There may be stakeholders whose interest is against one another. The system must make priorities based on the resource committed.

To identify the stakeholders, the design science research methodology book provides references to the onion model. The onion model is a classification system of stakeholder designed for determining stakeholders in a project [10]. The onion model proposes the concept of circles, slots and roles. The circles illustrate the relationship distance between the stakeholders and the actual system. The closer the ring to the centre means the closer its relation to the system. The circle contains slots, which predicts generic roles that may be needed by a project. The slots may not be used or filled with specific roles that interact with the projects.

There are four circles in the model: the product, our system, the containing system, and the wider environment. The product is the developed artefact. Our system contains stakeholders who directly interact with the system. The containing system contains stakeholders who get benefits from the operation of the system. Finally, the wider environment contains other related stakeholders, that do not directly interact with the product operation.

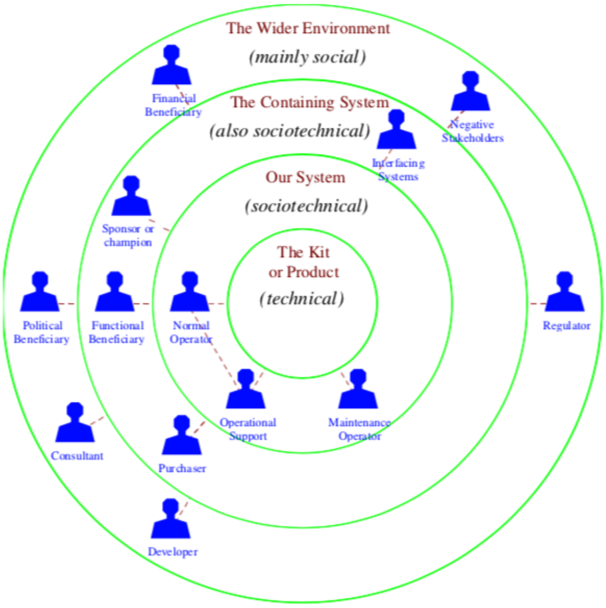

FIGURE 8 THE ONION MODEL

The circles contain the classification of possible stakeholder. In this study, I will not use all the available stakeholder's slot. I decided to fill the following slots: Normal operator, operational support, maintenance operator, functional beneficiary, developer, regulator, and the negative stakeholders. I omitted the rest because they do not commit resources to the system, or not relevant to the study. The regulator

The normal operator is the person that feeds inputs and command to the system. This person directly interacts with the system. The maintenance operator job is to ensure the system working as intended. The person is responsible for performing maintenance & troubleshooting. The operational support provides training on how to use the system to the normal operator. On the containing circle, we have the functional beneficiary, which is the person who uses the result from the system. In the wider environments, the developer is responsible for designing and realising the system. The regulator is responsible for ensuring the products comply with the law.

From the available slots, I decide the matching role based on the context of the case. A slot may contain multiple roles. The table below contains the slots and the matching roles.

Slots	Roles
Normal operator	Loading/unloading employees, Transportation operator
Operational support	System implementor
Maintenance operator	System administrator
Functional beneficiary	Transportation manager
Developer	System designer, programmer
Regulator	The government
Negative stakeholders	Hackers

TABLE 1 MAPPING BETWEEN THE STAKEHOLDER'S SLOTS AND ROLES

To know the desires of the stakeholders, I performed an interview with the identified person that have the roles in the case. The following section will describe the role of each stakeholder.

3.1.1 TRANSPORTATION MANAGER

The transportation manager is responsible for executing the transport process [11]. In this case, the transportation manager is responsible for scheduling the transportation between the factory warehouse and the distribution centre. The scheduling process is a decision-making problem. To make the decision, transportation managers need information that includes resource availability and delivery requirements so they can arrange shipments to take advantage of load/carrier consolidations or routing efficiencies [12]. We can improve the transportation efficiency by considering product flows, including volume, frequency, seasonality, physical characteristics, and special handling requirements [11].

Transportation manager has to ensure minimal customer service failures and unnecessary cost [12]. The transportation plan must consider the possibility of an exception to achieve minimal service failures. An exception is a deviation from the "optimal" (or acceptable) process execution that prevents the delivery of services with the desired (or agreed) quality[13]. Currently, the response time to an exception is slow. Some part of the supply chain causes the slowness when not tracked. It causes backtracking the order takes a long time to finish. There are 3 activities associated with improving service quality by exception handling: analysing, predicting, and preventing [13]. Hence, the logistics manager wants to know when an exception is happening, what causes the exception, who cause the exception, how to deal with the exception, and what is affected by the exception. The diagram below shows the motivation of the transportation manager.

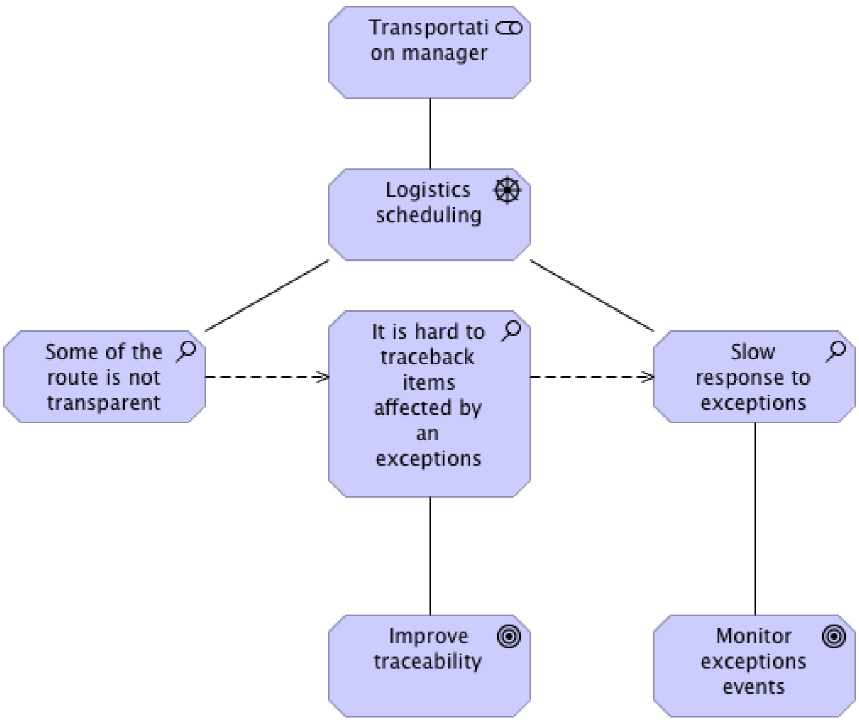

FIGURE 9 TRANSPORTATION MANAGER MOTIVATION MODEL

3.1.2 EMPLOYEES

There are two employee's roles in the case. The loading/unloading operator and the vehicle operator. The loading/unloading operator is responsible for loading or unloading the items from the vehicle. The loading/unloading employees are located on both the factory and the distribution centre. The employee's roles aside from loading or unloading the items are to give the transported items an identifier and submit the status of the items. The roles require the employee to interact daily with the systems. Therefore, usability is crucial for them. The definition of Usability is the effectiveness, efficiency, and satisfaction of the user [14]. Usability is important to minimise data entry mistake, lower the learning barrier and reduce the time for data entry, making the transportation process efficient.

The vehicle operator is responsible for delivering the items from the factory to the distribution centre promptly. The vehicle operator must report any anomalies during the transportation. The process is vital to monitor the items conditions. The report is necessary to ensure the speed of exception detection. For that reason, the vehicle operator desire usability. The diagram below depicts the desire of the employee

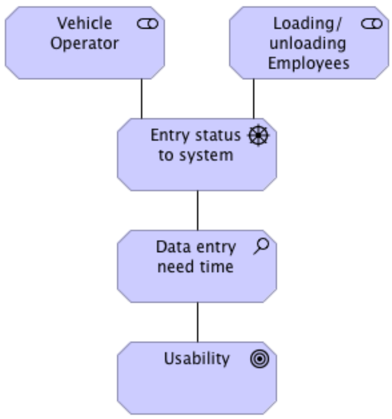

FIGURE 10 EMPLOYEES MOTIVATION MODEL

3.1.3 *CUSTOMERS*

There are five dimensions of service expectation: reliability, tangibles, responsiveness, assurance and empathy [15]. Reliability is the dependability of the service provider to uphold its promise. The Reliability correlates with the quality of service. Tangibles are the service appearance. The responsiveness is the ability of the company to help the customer in needs. The assurance is the ability to convey trust and confidence.

The dimension of reliability and tangibles correlates with the quality of service. One of the main tasks of the rapidly growing service sector is to ensure the quality of service to the customers [16]. The study suggests services that lack execution speed, flexibility and punctuality perceived as having low service quality [16]. These metrics depicts the performance of the logistical operation. Hence, logistics planning should concentrate on ensuring it.

The customer wants a relationship [15]. They want the company to care for them. One way to show care is to make available information about the customer orders, such as the status and current locations of the objects. Responsiveness, assurance and empathy are put to the test when the transportation process experience failure. The company is expected to notify the customer and assemble a follow-up plan when an exception happened. The notifications are essential, because the customer activities may be disturbed by the exceptions. Fast notification `gives the customers time to decide how to handle the effect of the exception.

In summary, the customers want the goods to the delivered in time, and intact. The customers also want to track where is the current location of their order, and when their order will have arrived. Finally, they want notifications when an accident happened as soon as possible.

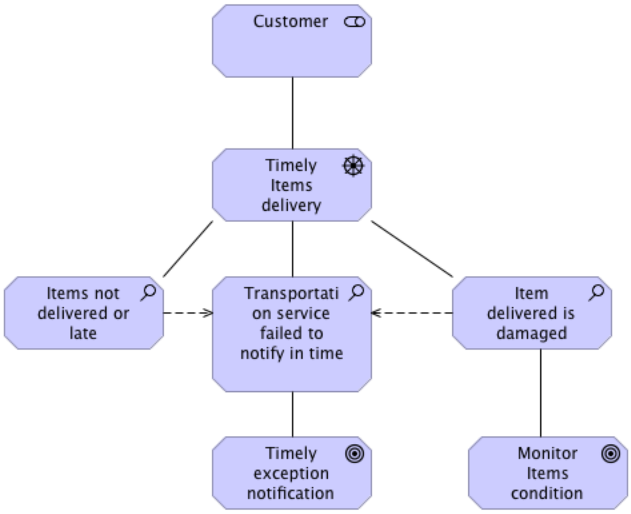

FIGURE 11 CUSTOMERS MOTIVATION MODEL

3.1.4 SYSTEM DESIGNER

The role of a system designer is to analyse and design the technical specification. The person will be held accountable for every decision that leads to system design. The person needs to communicate with the stakeholders to gather the requirements, determine the priority of the requirements and decide, and which requirements should the system designer implements.

A system designer wants a system that can cope with changes. The system designer wants that because the business environment is always changing. The company must continuously adapt to the new business trends and requirements. The changes will affect the systems. A system that could not adapt quickly will need significant rework by the system designer in the future. Moreover, if the cost of implementing the changes is enormous compared to creating new systems, the company may replace the systems. Both scenarios mean the company will lose time and money. Therefore, it is in the best interest of the system designer to design a system that can cope with changes.

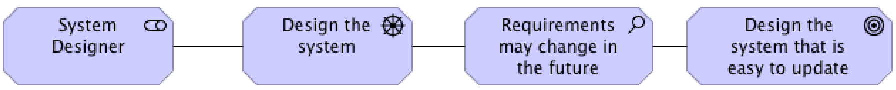

FIGURE 12 SYSTEM DESIGNER MOTIVATION MODEL

3.1.5 SYSTEM ADMINISTRATOR

The system administrator oversees managing the system. The responsibilities include: configuring the system's software and the hardware, managing user account, upgrading to the latest version, performing backup and recovery when a fatal error occurred. The system administrator wants the system to be scalable, secure, stable and easy to maintenance.

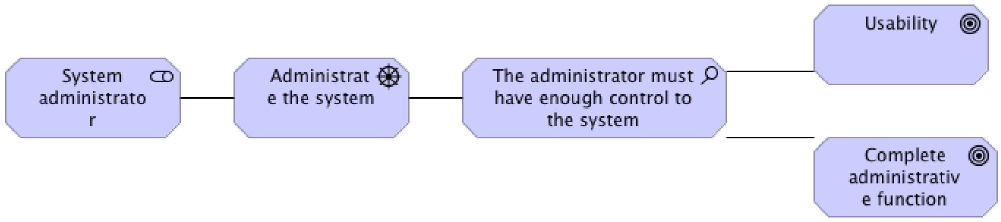

FIGURE 13 SYSTEM ADMINISTRATION MOTIVATION MODEL

3.1.6 *SYSTEM INTEGRATOR*

The system integrator is responsible for implementing the concrete application into the organisation. The person will train the involved stakeholders on how to use the system. The person also responsible for measuring the effect of the system, to validate whether the system meets the objective. The system implementor wants the system to be as easy to use as possible, so the organisation can accept the new system without too much effort to learn.

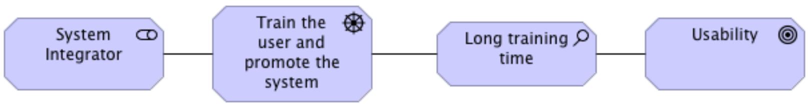

FIGURE 14 SYSTEM INTEGRATOR MOTIVATION

3.1.7 *THE GOVERNMENT*

The government is responsible for protecting its citizen. The government create policies and implement it. The government have the power to enforce the regulation. Therefore, the government may introduce regulatory risk for the business. Regulatory, legal and bureaucratic risks refer to the legal enforceability and execution of supply chain-relevant laws and policies (e.g., trade and transportation laws) as well as the degree and frequency of changes in these laws and policies [17]. One example of regulatory risk is the sustainability regulation. On 17 May 2018, the European Commission presented a legislative proposal setting the first-ever CO2 emission standards for heavy-duty vehicles in the EU [18]. The European Union government want to lower at least 15% of emission from heavy-duty vehicles [18]. Another regulation that the European Union implements is the General Data Protection Regulation. The GDPR states that organisations need to implement appropriate technological and operational safeguards for securing data, including putting in place strong privacy controls [19]. That means the system design requires information security and privacy as one of the main priorities.

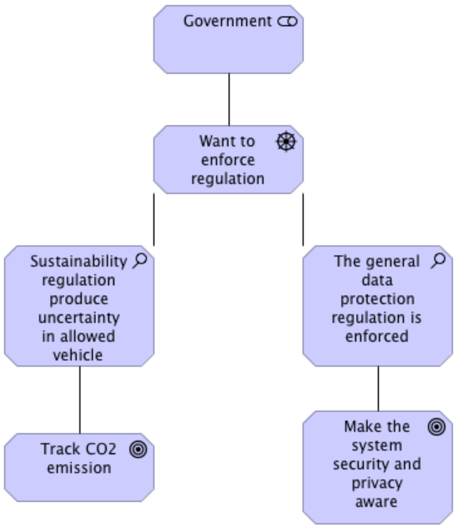

FIGURE 15 THE GOVERNMENT MOTIVATION MODEL

3.1.8 *THE CONSULTANT*

Cape Groep is an information technology consulting company established in the Netherlands. They mainly have an expertise in building information technology system for logistics operations. Examples of their clients are PostNL and HST Groep. Cape group is looking for a new technology solution to solve problems in logistics. That is why they partner with various universities and research group in the Netherlands. Cape Groep is part of the DATAREL project. They contribute to this research by providing resources and knowledge needed to finish the research. What cape Groep wants is a prototype implementation of the situation awareness logistics concept.

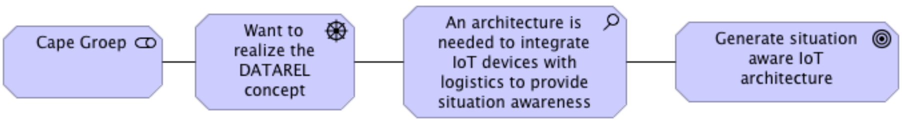

FIGURE 16 CAPE MOTIVATION MODEL

3.1.9 *THE SUPPLIER*

Ahrma is an information technology logistics company in the Netherlands. They serve clients in the field of foods & beverage, medicine and retail logistics industry. One of their products is the smart pallet. The pallets can be reused multiple times, and have sensors embedded in it. The sensor can detect temperature, weight and more. The Ahrma cloud store the data from the sensors. Ahrma is also part of the DATAREL project. They help this research by providing the IoT devices. Their desire from this project is to use smart pallets as the IoT device, as that will show their clients what the IoT devices can do.

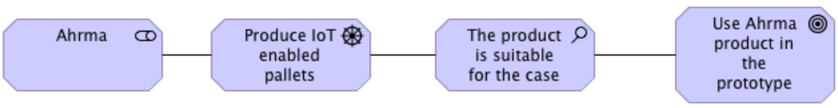

FIGURE 17 AHRMA MOTIVATION MODEL

The table below summarises the stakeholders and their desired goals for the system.

Roles	Desired Goals	References
Transportation manager	Want to improve traceability. Want to monitor exceptions occurrence.	[11][12][13]
Customer	Want timely exception notifications. Want to monitor items conditions.	[15][16]
The government	Want the company to adhere to the laws. (Information security & privacy)	[17][18]
The employees (Loading/unloading operator & Driver)	Want their work to be uninterrupted (Usability).	[14]
Ahrma (Supplier)	Want the system to use Ahrma products.	Interview
Cape Groep (Consultant)	Want to realise the DATAREL concept Want the system to be adaptable to a different company	Interview
System designer & programmers	Want the system to be simple and easy to update.	[14]
System integrator	Want usability.	[14]
System administrator	Want a powerful administrative function	[14]

TABLE 2 THE DESIRE OF THE STAKEHOLDERS

3.2 DESIGN REQUIREMENTS ANALYSIS

I derived the architecture requirements for situation aware logistics from the desire of stakeholders depicted in Table 2. First, I set a target for each of the desired goals. I use the targets as the basis to create the measure of stakeholder's goal achievement. The measure will be used to validate the design. I also use the targets to decide what kind of design principle should be used to build the design. I use the principle as the guideline to specify the exact architecture requirements. This way, the reasoning behind a requirement specification can be traced back to the stakeholders. I use it to make sure only essential requirements implemented in the systems.

The most important person responsible for PC to DC transportation business process is the transportation manager. He is the one that will use the output of the system. The transportation manager is also responsible for notifying the customers, which is related to the customer's desire. The customers desire to be notified as soon as possible when an exception that will affect their product delivery happened. Hence, I will discuss his requirements first. The transportation manager wants three things: improved transparency, traceability and monitor exceptions.

Exceptions monitoring depends on traceability and transparency. Supply chain transparency and traceability is a related concept. Traceability will only happen when the supply chain is transparent. A transparent supply chain means we know exactly where, when and why the items located at a place at a time. To improve transparency, we need to give identification to the transported items, then record the events that happen to surround the items. To improve transparency, I propose to embed IoT sensors into the pallets. That way, we can monitor the events surrounding the order lines. Embedding IoT sensors into pallets also aligned with the supplier (Ahrma) desire, to use the smart pallets into the solution. To use the smart pallet, we need to manage it. The designed systems must have the capabilities to register and deregister a pallet. It also must have the capabilities to manage the data from the smart pallets. The managing capabilities include the abilities to store and remove the data.

In Figure 7, we can see the applications and data entity that is used for PC to DC distribution process and the exception process. There are two applications: ERP systems and the scheduling systems. The ERP systems are used to store the order lines data. The scheduling systems are used to schedule the order lines into the vehicles. The designed system needs to handle the data from all the mentioned applications.

Furthermore, In the future, the applications might become different. For example, because of an update, changing vendors or maintenance. Therefore, the designed architecture requires data integration capabilities that are flexible to change.

The employee (Both the loading/unloading employee and the vehicle operator) are responsible for putting the association between the order lines with pallets and the pallets with the vehicle. Because they will use it daily, they need a data input method that is fast and easy to learn. This goal also aligned with the system integrator goal. The person is responsible for giving the employee training to use the systems. An easy to use systems will help to make the training process faster. Therefore, it is required for the design to adhere to the usability principles.

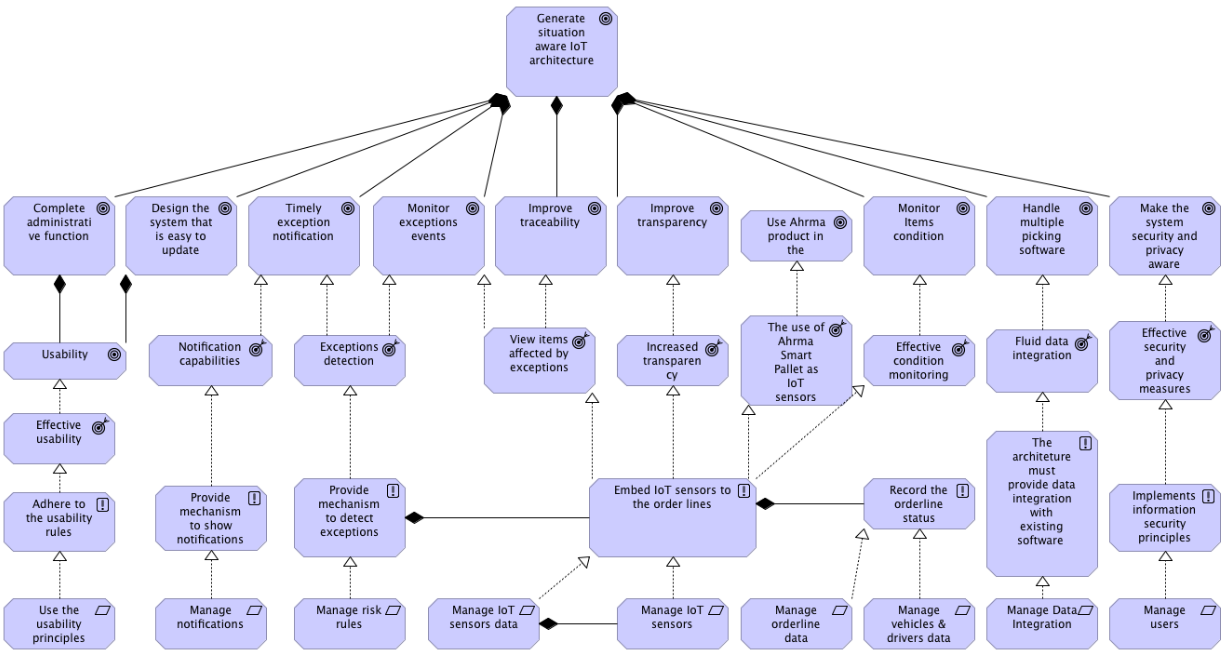

FIGURE 18 GOAL REALISATION MODEL

From Figure 8, we know there are nine systems requirements derived from the stakeholder's goals. I further categorised the requirements into functional requirements and non-functional requirements. Functional requirements are requirements for desired functions of an artefact. For example, the capability to manage users. Non-functional requirements are capabilities of the artefact that is not a function. For examples: Accuracy, efficiency, usability. The reason I categorised the requirements is that non-functional requirements need indicators to be validated. Indicators are variables that can be measured and that indicate the presence of the property. Therefore, I represent the functional and the non-functional requirements as a table below:

Ref	Functional requirements	Ref	Non-functional requirements
R1	Manage exceptions report	NFR1	Usability
R2	Manage notifications	NFR2	Information security
R3	Manage exceptions rules	NFR3	Accuracy
R4	Generate exception notification	NFR4	Effectiveness
R5	Manage IoT sensors		
R6	Manage IoT sensors data		
R7	Manage order lines associations		
R8	View exceptions report		
R9	Manage exceptions		
R10	Manage vehicles data		
R11	Manage driver's data		
R12	Manage data integrations		
R13	Manage users		

TABLE 3 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

What I mean by manage in Table 3 is the capability to create, read, update and delete the data managed by the system. I use the prefix *R* followed by a number to denote a functional requirement. Consequently, I use the prefix *NFR* to denote a non-functional requirement.

4 LITERATURE REVIEW

The next steps after finding the requirements are to review the existing solution in the literature. The literature review is essential to know the current best practice. I use the requirements determined in the previous section to scope the topic of the literature that I should explore.

The primary purpose of the situation-aware logistics information system is to process data that comes from IoT devices. There are many problems with managing the IoT data, and many ways to solve them. For example, where should the data be stored and processed, How the architecture should be structured, what kind of components are necessary to achieve the objective. Hence, the first thing that I decided to investigate is the reference architecture for managing data from IoT Sensors. I need determine which reference architecture is suitable for managing the data from the IoT sensors.

The transportation manager desire is to know when, and why an exception is happening. In that case, I need to know the type of exception that the system must detect. On top of that, the literature can provide me with the method to detect the exception. We should note that the detection method must be able to detect an exception, by considering the available information from the IoT sensor. I use the following limit to scope what an IoT sensor can provide: temperature, humidity, shock status, and location. The current and historical data will be available as an input.

After I know how to detect the exception, I need to know how to incorporate the exception detection method into the application. I need to know how to provide the necessary data for the detection method, and when to trigger the detection method. The complexity of the problem is high because the data may come from various IoT sensor at the same time. The data from the various sensor may contain the same information. Which means it is wasting processing resource if we perform the detection method over the same data set. The data that come from the sensor may not be relevant for the detection method. Moreover, there is a chance for data corruption. Hence, filters must be incorporated so that the detection method only process relevant data.

Moreover, Different IoT sensor may use a different data format from each other. Furthermore, if the detection method needs historical information, it may use a different data format as well. Therefore, the system needs an architecture to Filter, integrate and process the input data in real time.

Finally, because the user will use the system daily, it needs to have a good user interface. The user interface is especially important for the loading/unloading employee and the vehicle operator. The loading/unloading operator must input data to the system when the person is loading/unloading the items. The system must ensure that the usage of the system will not disrupt the loading/unloading process too much. Therefore, usability principles are essential.

I summarised the required information as research questions below:

1. What is the reference architecture for managing data from IoT sensors? **LRQ1**
2. What are the exceptions that the systems must detect? **LRQ2**
3. What is the architecture to detect an exception from the sensor's events? **LRQ3**
4. What is the architecture to integrate multiple data source into the applications? **LRQ4**
5. What are the usability principles in an information system? **LRQ5**

I use the systematic literature review methodology to do the literature review. A systematic literature review is a means of evaluating and interpreting all available research relevant to a particular research question, topic area, or phenomenon of interest [20]. The main differences between a systematic literature review and "normal" literature review are we need to generate a review protocol before performing the review. The protocol contains the search terms, the source database as well as the including and excluding criteria. The reason is so that the reader can trace where the result is come from, so the person can validate the logic behind the chosen search terms.

4.1 Search terms

The search terms are keywords for searching the literature database. I determine the keywords by decomposing the literature research questions. For example, in **LRQ1** we want to look for literature that describes a reference architecture for managing data from IoT sensors for logistics transportation. Accordingly, the keywords should contain at minimum "reference architecture", "IoT", "logistics" and "transportation". However, the results from the literature database may not be focused enough. Hence, I further refine the search terms by the searching result. If the results return keywords that is useful, I will append that to the search terms. If the results return unrelated results, I will exclude the keywords from the search terms. Table 4 contains the generated search terms.

Literature review research questions	Search terms
LRQ1	*("reference architecture" OR standards) AND transportation AND ("supply chain" OR "logistics") AND (IoT OR "internet of things") AND (smart OR intelligent)*

LRQ2	*(“classification” OR “taxonomy”) AND “Exception” AND “Logistics”*
LRQ3	*(“event” AND “processing”) AND (“IoT” OR “internet of things”) AND “Exception” AND “Logistics.”*
LRQ4	*“Data integration” AND (“IoT” OR “Internet of Things”)*
LRQ5	*“Usability” AND “principles” AND “Information systems.”*

TABLE 4 LITERATURE REVIEW SEARCH TERMS

I add *(“smart” OR “intelligent”)* to the search terms for LRQ1 because that keywords make the search results more relevant. Some papers may use different terms to convey the same concept. Hence, I use the operator “OR” with the variation of the terms to cover all papers. For example, (“classification” OR “Taxonomy”). Both concepts describe similar meaning.

4.2 LITERATURE DATABASES

To decide which literature databases to use, First I check which literature databases I can access. Then, I look for literature databases that contain paper relevant to the project. I do this by looking at the literature database description and whether the search terms return relevant results. The results are, I use the following academic databases to do the literature review:

- IEEE Xplore Digital Library
- ScienceDirect
- Scopus
- Google Scholar

I choose IEEE Xplore because it contains many technical papers about Internet of things, logistics and information systems. I also choose Science direct because the search result is accurate and relevant to the search terms.

Moreover, science direct also contains papers that are not in the IEEE database. I also use the Scopus database. The Scopus database gives a smaller number of papers compared to other databases. However, the results are usually more relevant. I use Google Scholar when I could not get relevant results from other literature databases. Google scholar served as a backup because the quality of the papers on google scholar is questionable. Google scholar find papers from all the internet, while in other literature databases, it only contains papers that have been approved or curated.

4.3 INCLUSION CRITERIA

I will include papers that matched the criteria. The criteria are chosen based on the research questions to make it focus. I will only include papers and standards from a maximum of seven years in the past. The inclusion criteria are selected to cope with the fast-paced technological progress in smart logistics. Listed below are the selected criteria for the paper:

- The paper is related to the architectural design of a logistics information system.
- The paper contains transportation exception taxonomy.
- The paper describes methods to process events stream from IoT devices.
- The paper describes methods to integrate data into information systems.
- The paper describes usability principles.
- The paper published between 2011 – 2018

4.4 EXCLUSION CRITERIA

I choose the exclusion criteria based on the irrelevant results of the search query. I excluded papers that possess the criteria described below:

- The papers only describe general IoT implication in logistics
- The papers only describe what exception in general is.
- The papers describe the offline processing of the data stream.
- The papers describe a security system for the harbour.
- The paper objective is to explain how IoT devices work at a technical level.

4.5 LITERATURE REVIEW RESULTS

4.5.1 LRQ1: REFERENCE ARCHITECTURE

I performed the search into the literature database using keywords defined in 4.1, to find existing architectures and standards currently used in logistics. I found 59 papers, describing IoT architecture in various context. However, three reference architecture described in Table 5 below are the most discussed in the literature.

Architecture	Literature
Layered	[21][22][23][24] [25][26][27][24][28][29][27][30][31][32] [33][34][32][35][36][37] [33]
IoT-A	[38][39] [37][40] [41]
EPCIS	[42][43][44][45][46][47][48][49][50][51][52]
Uncategorized	[53][54][55][56][57][58][59][60][61][62][63][64][65][66][67] [68][69][70][71][29][72][73]

TABLE 5 DISCOVERED REFERENCE ARCHITECTURE

Table 5 shows three approaches for reference architecture: Layered IoT architecture, IoT-A reference architecture and EPCIS. Some of the architecture in the literature database is uncategorized. The uncategorised happened because the literature talks about a specific function or the architecture only described in a few sources. Under the circumstances, I decided to focus on the three architectures: Layered, IoT-A and EPCIS.

4.5.1.1 Layered architecture

The layered Internet of things architecture is the most described architecture in the sources. It is designed to be generic. Hence it does not specify any standardised data model. As a result, the architecture is flexible enough to handle a wide range of problem context. Each of the architecture implementations may use different standards and components. The problem is, the lack of a clearly defined structure means best practice knowledge for architecture development is limited. Consequently, architecture designer may not use the optimal solution for solving a requirement.

In the layered architecture, the layers generally separated into three or five layers. The three layers model defines three layers: Perception Layer, Network Layer, and Application Layer. The perception layer is responsible for gathering data from the real world. The network layer oversees data transmission from the perception layer to the application layer. The application layer transforms the data into information. Figure 19 provides visualisation of how the layers stacked together.

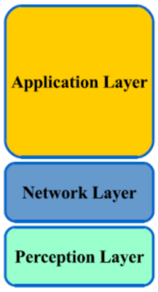

FIGURE 19 3-LAYER ARCHITECTURE [34]

The problem with 3-layer architecture presented above is it does not provide a complete picture of a production-ready architecture. The architecture requirements require the architecture to handle security. Regrettably, 3-layer architecture does not describe how the architecture could provide trust. On the other hands, it leaves room for architecture designer to improvise. Some of the literature studied describes the way to provide security or flexibility to the architecture.

The next iteration of the layered architecture is the 5-layer architecture.

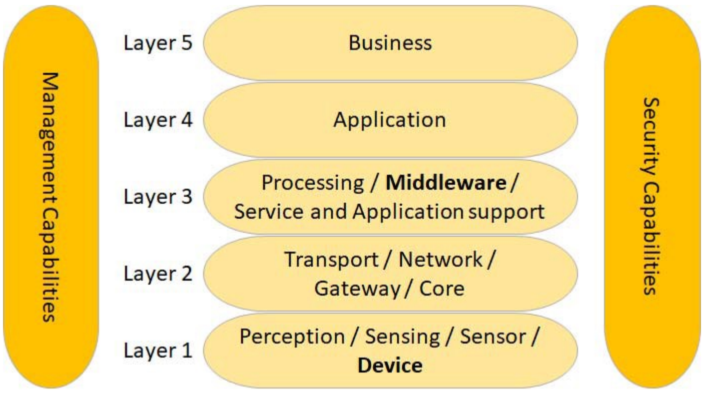

FIGURE 20 5-LAYER ARCHITECTURE [30]

The 5-layer architecture provides a more comprehensive components separation. The differences between the 3-layer architecture and the 5-layer architecture are the addition of business layer, processing layer, security capabilities and management capabilities. The processing layer is responsible for storing, analysing and processing the acquired data. The business layer represents the stakeholders' goals [30]. The management capabilities layer provides scalability, and the security capabilities layer provides security for all components.

The 5-layer architecture does not define the location of the processing layer and the layer above it. Several sources tried to remedy this deficiency by introducing cloud computing and fog computing into the architecture. In the cloud computing model, the data is processed using resources provided by a big data-centre, using on-demand pricing model. Cloud computing allows scalability by making the process of resource acquisition easy. The system administrator can click a button to increase a computing resource. The drawback of this approach is the centralisation of data processing. The centralisation means the data-centre must be able to process an unprecedented number of data. The number of data generated from IoT sensors is exploding each year. The power of the data centre and the financial budget of the architecture owner limits the scalability of cloud computing-based processing architecture.

Fog computing provides a solution to alleviate the problem with cloud computing. In this approach, the data are pre-processed close to where the sensors are gathering data, or in another word, on the edge of the system. By pre-processing the data, the system could filter unnecessary data, so the final computing resources only working with useful data. This final processing could happen in the cloud. Not only it will lessen the burden on the final processing layer, but it also means less data needs to be transferred by the network layer, possibly making the system faster and reliable. The disadvantage of this process is it increases the complexity on the edge of the system. The devices located on the edge need to be powerful enough to process data from the sensors. It also must be programmed to process the data.

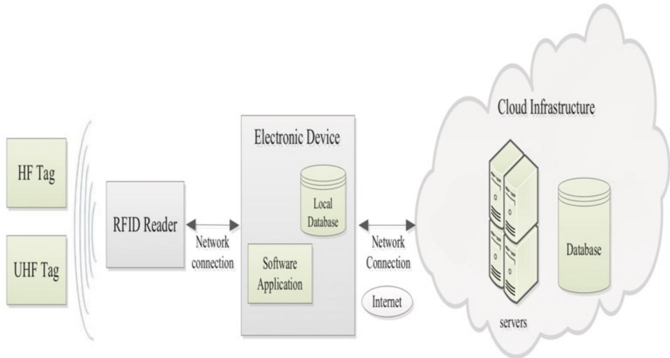

Figure 21 Cloud computing-based architecture [57]

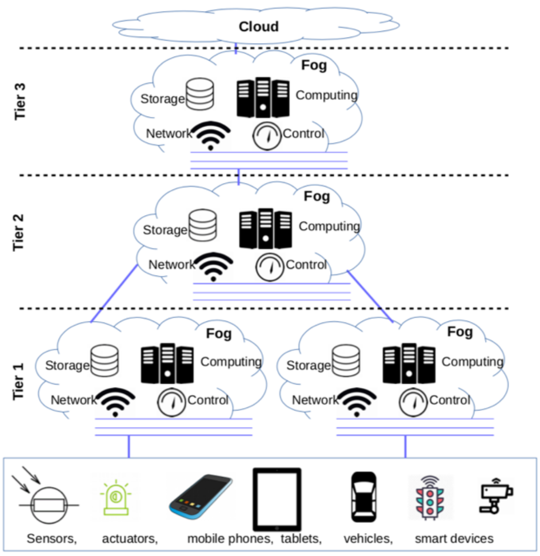

FIGURE 22 FOG COMPUTING-BASED ARCHITECTURE [60]

Figure 21 and Figure 22 show the differences between cloud computing-based architecture and fog computing-based architecture. In the cloud computing model, no processing happens before the data enter the cloud. While in Fog computing, the fog Layer already processes the data even before entering the cloud.

From the data preparation and processing perspective, the 5-layer architecture does not specify any specific methods. Data generated from the IoT device for the smart logistics planning is real-time and potentially massive. For example, to solve the sustainability business concerns, the system must be able to handle CO_2 emission sensor from the vehicle or ship. Another example, to track cold products, the system must handle the data from the temperature sensor. The volume of the data depends on how many tracked objects at one time. The velocity of the data depends on the resolution, how many data samples the user need per second.

The literature suggests using big data analytics techniques to process the data. Big data analytics defined the process of examining large data sets that contain a variety of data types to reveal hidden patterns, hidden correlations, market trends, customer preferences, and other useful business information [29]. Big data analytics defines three steps to do the process. The first step is the management of data sources. The second is the management of generated big data. The final step is the analysis of the data using analytical tools and data mining algorithms. Figure 23 contains a sample of this process in action.

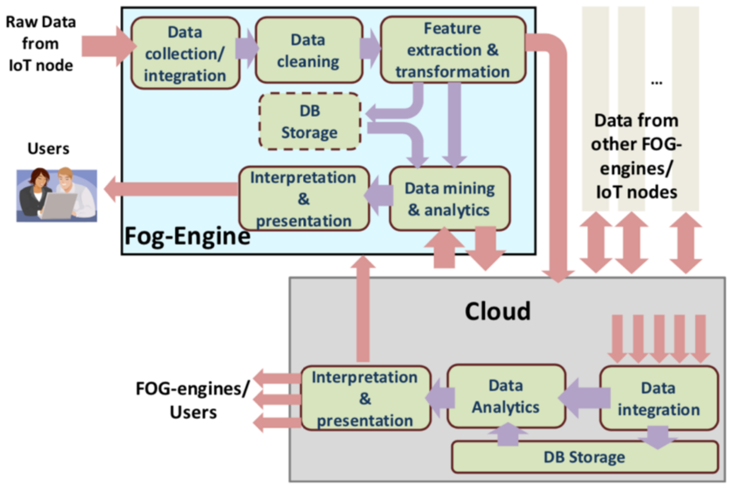

FIGURE 23 BIG DATA ANALYTICS IN FOG AND CLOUD ARCHITECTURE [62]

Figure 7 explains the use of fog architecture in conjunction with the cloud architecture to process data from the IoT sensors. The fog layer is used to pre-process the data by cleaning it from unnecessary noise and extract only the significant information. The data processed by the fog layer do not represent data from all the sensors. This pre-processed data needs to be integrated and analysed to get the complete picture. Cloud computing is suitable for the integration step because of its power and flexibility. The system needs the power to integrate and process data gathered from the fog layer, while must be flexible enough to handle changes to scale. In conclusion, the combination of fog and cloud architecture provide processing capabilities and scalability.

In summary, a layered architecture is a flexible architecture. The literature provides a solution to ordinary problems. It matched all requirements for smart logistics planning. However, this flexible approach means more time and resource is needed to design and validate an approach to solve a problem.

4.5.1.2 IoT-A

IoT-A is a complete reference architecture for Internet of Things. It is co-funded by the European Commission, industrial project partners, and universities in the EU. IoT-A is derived from the cutting-edge development of IoT architecture field in 2013 [41]. It composed from IoT reference model, IoT reference architecture and guidelines to use it. IoT-A is abstract enough to accommodate a wide range of context[41].

The first thing that IoT-A specify is the reference model. The IoT reference model aims at establishing a common grounding and a common language for IoT architectures and IoT systems [41]. It composed of several models: domain model, information model, functional model, communication model and trust, security and privacy model. The domain model describes the concepts related to the Internet of things such as the device and the service it provides. The information model defines the relations and attributes of IoT information. The functional model groups functional interaction of the concepts. The communication model describes how each functional component communicates with each other. Figure 24 describes the relationships between the reference models

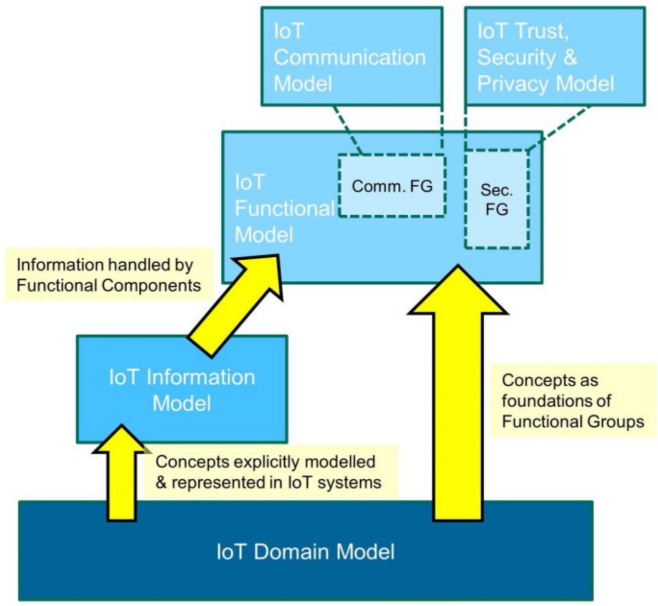

FIGURE 24 RELATIONSHIP BETWEEN IOT-A REFERENCE MODEL

The second part of the IoT-A is the reference architecture. The reference architecture is composed of several architecture views: functional view, information view, and deployment view. Architecture view is a way to structure architecture description. The motivation behind separating architecture into view is to make the derivation of architecture and its validation easier by providing focused description based on its functionality [41].

The functional model guides the development of the functional view, while the requirements steer it. IoT-A provides a list of requirements, called unified requirements. The requirements consist of IoT Process Management, service organisation, IoT services, virtual entity, communication, security, and management [41]. The unified requirements are generic, and some of the requirements align with the situation-aware logistics requirements. Table 6 below describes each requirement and mapping to situation aware logistics requirements.

IoT-A Unified Requirements	Situation-aware logistics requirements
Process modelling Process execution	Data processing
Service composition Service orchestration Service choreography	-
IoT service IoT service resolution	-
Virtual entity service Virtual entity resolution	-
End to end communication Network communication Hop to hop communication	Communication
Authorization Key Exchange & Management Trust & Reputation Identity Management Authentication	Security & Privacy
Configuration Fault Reporting Member State	Data source management

TABLE 6 IOT-A REQUIREMENTS, MAPPING TO THE SMART LOGISTICS PLANNING ARCHITECTURE REQUIREMENTS

I discover that some of the IoT-A requirements do not have a connection to the situation-aware logistics requirements such as the service composition, service orchestration and service orchestration. The mapping failure is because smart logistics planning does not have preferences for architecture style. I believe the IoT-A requirements is a derivative of the service-oriented architecture requirements. Nevertheless, I believe that the additional requirements proposed by IoT-A is useful and complete the smart logistics planning requirements. The exception to this is the reputation, which we do not believe is useful for situation aware logistics.

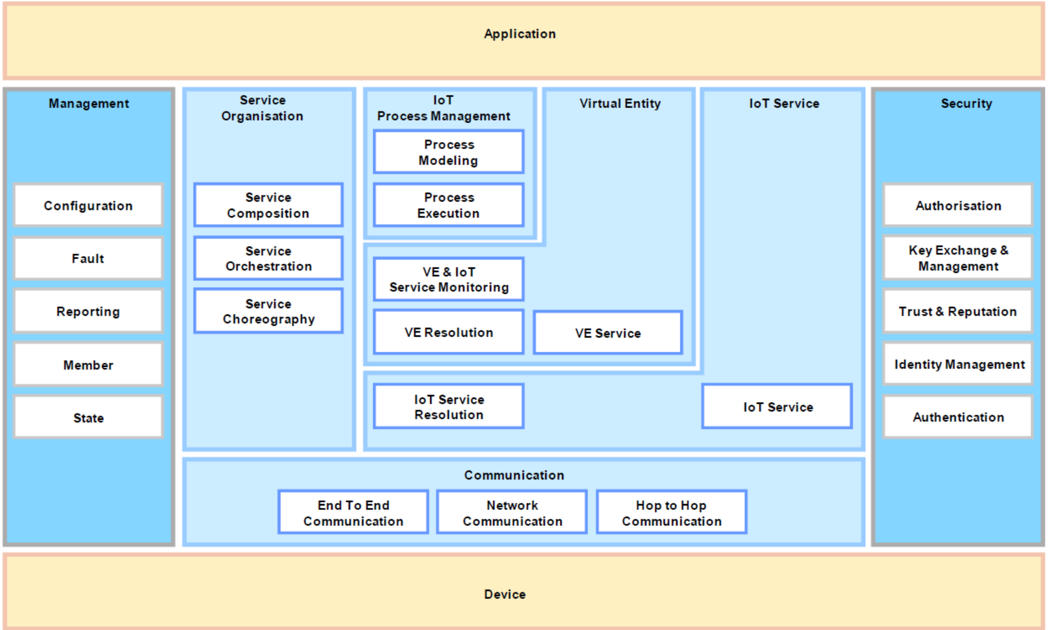

FIGURE 25 IoT-A FUNCTIONAL VIEW [41]

The functional view group each of the requirements into the function group and depict it in Figure 25 above. The purpose of this view is to describe the components. Hence, IoT-A architecture consists of 9 components: application, management, service organisation, IoT process management, virtual entity, IoT service security, communication and the device. The interaction between elements is not drawn inside the diagram to allow the existence of various context bounded implementation.

The purpose of the IoT process management is to provide integration between business function and information from the IoT devices. The service organisation function group works as the information router between services. Service organisation function group is responsible for arranging services, so it performed together to process data into the desired result. IoT-A uses the virtual entity to represents multiple sensor data as a single entity. For example, in a cold chain, items that are temperature monitored can be represented as a virtual entity, by associating the item identifier, such as a unique barcode with the temperature monitoring services. IoT service exposes a resource to the system.

The communication function group provide the means of communication between devices on the different network to the system and connecting one architecture components to the other. It is responsible for ensuring a reliable end to end communication of the system.

The security function group is responsible for managing the access right of the user, system components, and devices. The function group needs to manage identity and perform authentication. IoT-A also requires the architecture to record user reputation for building trust score. The management function group is responsible for managing configuration, do fault handling and reporting.

The last component of IoT-A is a guideline for system architect to make the reference architecture into real implementation. The guideline provides a generic methodology to build derivative architecture. The step is: (1) define the scope, i.e., the business goals; (2) create the Physical-Entity View and the IoT Context View; (3) define requirements; and (4) generate the remaining view [41].

Even though it seems complete, IoT-A does not explicitly mention where and how the data are stored and managed. We may store the data in a service. The downside is, the service becomes a single point of failure. For example, when the storage service is flooded with too much sensor data and fail, another system that depends on the storage service will also stop working. Another approach is to decentralise it. The system stores the data inside an IoT sensor service. That way, when one sensor service fails, only data from that sensor service is unavailable. The approach is a trade between complexity and availability. IoT-A should describe the best practice to do this, but it did not.

IoT-A also does not describe the need for standardisation. Standardization is essential for collaboration. So, IoT-A reference architecture is not useful for doing a collaboration, which is a concern for smart logistics planning stakeholders. It does not describe how to share the information gathered with the external party securely.

The European Union funds the IoT-A, so in theory, the world should widely use the system. However, the literature that describes a successful implementation based on IoT-A is fewer than expected. The specification includes links to web resource which contains IoT-A explanation. Unfortunately, the links are not working. The condition raised a question, whether the developer still maintains the reference architecture or abandoned. A quick search on the internet suggests the abandonment of IoT-A. The literature does not provide insight into why this happened. I think this is because of IoT-A complexity.

The conclusion is, IoT-A provide a comprehensive reference architecture for smart logistics planning, and it includes the guideline on how to implement it. It also specifies additional architecture requirements relevant to smart logistics planning architecture. However, IoT-A is also missing some essential information. It is not clear where to store data, how the architecture store data, the means of sharing information between different systems and little literature mention successful implementation found. Furthermore, I believe the architecture is abandoned.

4.5.1.3 EPC Information System (EPCIS)

EPC Information system or EPCIS is an architecture designed by GS1. GS1 is the official organisation responsible for barcode, which is ubiquitous in all products. EPCIS specifically engineered for traceability. Traceability is the ability to follow back the route a product takes, from the finished product at the hand of the end customer to the production of the raw material. Traceability capabilities can be used for example to track where food is contaminated when food poisoning is happening. Traceability is also essential to optimise logistics route. Traceability needs collaboration between partners inside the supply chain. It also needs a system that could track the position of an item in real time. EPCIS provide the architecture to do it.

To perform its function, EPCIS define an object identifier standard. The preferred identifier is the GS1 identification keys. It consists of many keys, which could be used by a company to relate objects to events and other data. EPCIS also define a data model for recording traceability data. EPCIS is responsible for capturing, recording and presenting the traceability data. Traceability data consist of four types: master data, transaction data, and visibility event data. Master data holds business facts, such as locations, product, clients. Transaction data contains data that describe transactions between business entities, and visibility events records save the “who,” what,” where,” and “when” an object is. “Who” is the object identifier that raises the event. “What” is what the object is doing in a place, “Where” is the location of the object, and when is the time of the generated event. Data storage in

EPCIS is general purpose. EPCIS can store any types of data because the data model is extensible. So, EPCIS can be extended to store any data from an IoT device. Because the event type is general purpose, GS1 provides a companion standard called the core business library. The core business library covers the definitions of data that can be used to populate the EPCIS data model. The diagram below depicts the generic type of data that EPCIS handle.

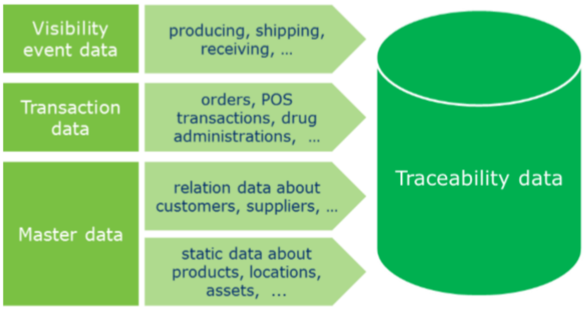

FIGURE 26 EPCIS GENERIC DATA TYPE [52]

To capture, identify and to enable sharing, EPCIS define an architecture based on the following principles: Layered, Extensible and Modular [52]. Layered means the separate architecture structure and the meaning of data from the real implementation. The arrangement is made to make reusability of the data structure and meaning in different architecture possible. Extensible means the core data types can be extended to include the needs from the application context. Other specification documents can replace modular, the components of the architecture. Figure 27 below depicts the components divided by layers.

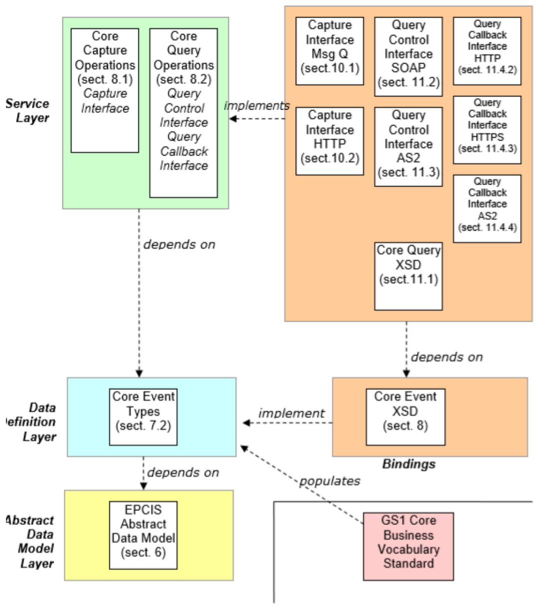

FIGURE 27 EPCIS ARCHITECTURE LAYERS [52]

The layers are the service layer, data definition layer and abstract data model layer. The abstract data layer specifies the requirements for defining data definitions at the data definition layer. The data definition layer specifies the type of data that EPCIS used. The GS1 common business vocabulary standards define data types. However, the use of other standards is possible if it fits into the EPCIS abstract data model requirements. The service layer contains component interfaces. These interfaces are responsible for implementing the core functionality of EPCIS: Capture, and Query. The capture interface is responsible for catching events raised by the objects. The query interface is responsible for handling user query for searching and filtering the captured events. Figure 28 below shows the concrete implementation of EPCIS.

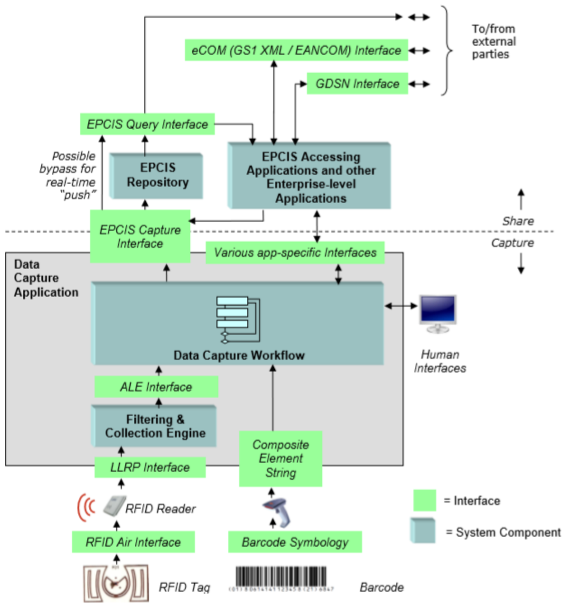

FIGURE 28 EPCIS ARCHITECTURE [52]

Figure 28 shows how the architecture can capture, query and share data retrieved from objects. First, when an event occurred, for example, "Object moved to a new location," the identity of the object is gathered using RFID or barcode. The data capture application processes the identity. This application is responsible for generating events that can be interpreted by EPCIS, through the capture interface. The events then are stored inside the EPCIS repository, ready to be queried. The user or other entities such as the third party can query the data stored inside the EPCIS repository using EPCIS query interface.

Based on the fact presented above, we believe EPCIS is mainly about specifying a standardised way to record, store and query information. The standard does not care about how the architecture communicates the data, or how the repository stores the data. EPCIS mainly cares about what type of data can enter the architecture, and how it can be presented using a standardised interface. The way EPCIS put emphasise on standardisation is useful for sharing, but not so useful for smart logistics planning. It is because smart logistics planning requires more than just standardisation, communication and data source management. It also needs specification for processing the data into useable information, ensuring scalability, and enforcing security and privacy which EPCIS do not indicate inside the specification. The good thing is, EPCIS is extensible. Architecture designer can extend the architecture to provide the missing requirements. Trying to extend the architecture is what some of the literature about EPCIS do.

The literature described extensions and implementations of EPCIS, for example, [47] propose to use MongoDB as the repository implementation for the events, addressing the scalability requirements, by using the capabilities of MongoDB to be horizontally scaled. [49] suggest creating a repository for knowledge, processed using fuzzy rules. [51] compares implementations of EPCIS architecture, from Microsoft BizTalk to Sun Java system RFID software architecture. The Microsoft implementation includes an event processing engine to process the data. We could not find literature that describes how to secure EPC based architecture. A source does discuss the challenge of securing EPCIS architecture, and which components need to be secured [46]. Security for EPCIS still needs more research.

In summary, EPCIS provide architecture specifications and standard data model, for capturing events from objects, store it and make it easy to search and filter the event data by the implementing company or partners. Unfortunately, EPCIS does not satisfy all the requirements for smart logistics planning, although it is usable as references for doing sharing between business partners

4.5.2 LRQ2: Logistics exceptions

An exception is "any deviation from an 'ideal' collaborative process that uses the available resources to achieve the task requirements in an optimal way" [2]. The primary purpose of the logistics process is to ensure the availability of the right product, in the right quantity, in the right condition, at the right place, at the right time, at the right cost, for the right customer. When the service provided deviates from the definition, we can consider it as having a logistics exception [74].

We can categorise Logistics exceptions can fall into three broad categories: Strategic, Tactical and operational [74]. It can be further breakdown into deviation or disruption. A deviation is said to have occurred when one or more parameters, such as cost, demand, lead time, within the supply chain system stray from their expected or mean value, without any changes to the underlying supply chain structure [74]. Disruption occurs when the non-availability of specific production radically transform the structure of the supply chain system, warehousing, and distribution facilities or transportation options due to unexpected events caused by human or natural factors [74]. In my opinion, a deviation is the type of exception that has minimal effect on the operation, while disruption will affect the entire operation. The table below contains examples of the classifications.

Planning level	Type of Event	Example
Strategic	Deviation	Capacity reduction
	Disruption	Bankruptcy
Tactical	Deviation	Scheduling delay
	Disruption	Scheduling failure
Operational	Deviation	Traffic jam
	Disruption	Traffic Accidents

Table 7 Classification of exceptions [74]

In this thesis, I will scope the research only to handle operational level exceptions. In the operational level, there are many possible causes of an exception. The supply side, transportation, facilities, contract breaches, communication failures or other fluctuations may cause the exception [74]. Another research also created the following classification for the type of exception: delivery exceptions, payment exceptions, system data exceptions, planning exceptions, document/paper works exceptions and communication exceptions [75].

The exceptions that this study will focus on is the delivery exception and facilities exception for perishable goods. Transportation or delivery exception is the delay or unavailability of either inbound and outbound transportation due to carrier breakdown or weather problems [74]. Factors that should be monitored to detect delivery exception includes weather condition, delivery milestones, customs clearance status, space availability, port progress status, warehouse status, transport availability, driver status and traffic status [75]. In summary, the method to detect the exceptions is by monitoring the status and location of the items. The status can be notified by the operational employee, while the location of items can be monitored using IoT sensor.

Facilities exception is the breakdown of machines leading to delay or unavailability. In the case of perishable goods, broken refrigeration unit can cause the transported items to degrade or perished. Regulation EC, No 852/2004 on the Hygiene of Foodstuffs, requires manufacturers to have suitable temperature controlled handling and storage facilities that can maintain food at appropriate temperatures and enable monitoring, recording and controlling the temperatures [76]. Every category of food has its specific temperature and humidity requirements [76]. For example in ice cream, the perishable condition depends on what is the ice cream is composed of and what kind of processing is done to make the ice cream [77]. Therefore, to detect this type of exception, the system needs to know the type of the transported items and the condition that can make the items perish. Then, the system can compare the baseline condition to the condition measured by the IoT sensor.

To summarise, the list below contains the exception that the situation-aware logistics information system should handle:

1. Transportation exception, which can be recognised by monitoring Weather condition, delivery milestones, customs clearance status, space availability, port progress status, warehouse status, transport availability, driver status, and traffic status.
2. Facilities exception, especially the climate control units, which can be detected by modelling the behaviour of the transported items, related to temperature and humidity.

4.5.3 LRQ3: The architecture for detecting exceptions

[75] describe a layered based architecture to do the event processing architecture to detect and suggest a possible solution to an exception. The paper calls it the real-time logistics exception decision model. The architecture consists of three layers: the event absorption layer, event processing, and filtering and event action, delivery and display. Figure 29 below depicts the three layers described in the paper.

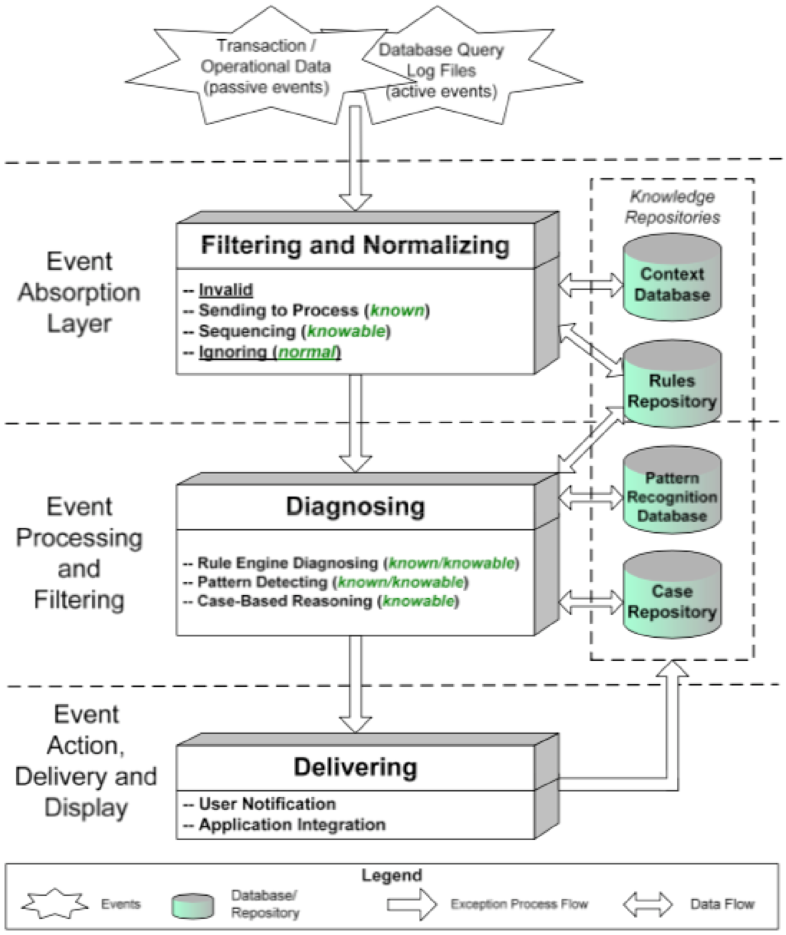

FIGURE 29 EVENT PROCESSING ARCHITECTURE [75]

In the event absorption layer, the architecture will collect the events through the data sources. Because the source may have a variety of format, the system must adapt the data into a common format. Then, the data must be validated and filtered for relevancy [75].

In the event processing layer, the event is processed using a rule-based system. First, the system categorises the filtered events by using a suitable pattern from the pattern recognition database or rules from the rule database. Both the rules and the pattern recognition database contain a model for an exception. However, the rules only contain simple threshold model, while the pattern recognition database contains information on how to correlate patterns from multiple events. When the events correlate with some pattern or rules, it investigates the case database, which contains the type of information that needs to be in the exception report. The exception report contains information needed by the user to create a plan for mitigating the effect of the exception.

In the event action, delivery and display layer, the components present the user with notifications. Maintaining the quality of the notification is essential[75]. The aspect of quality includes timeliness, the accuracy of distribution, context, and format [75]. Timeliness means the notification should be available immediately when an exception happened. The accuracy of distribution means the system should deliver the notification to the right person. Context means the notification should provide information that is necessary to decide. Format means the notification should be readable by the user.

The architecture described above is not complete. There are still many challenges, for detecting useful information from events raised by IoT sensors. For example, Internet of Things (IoT)-generated data are characterised by its continuous generation, large amount, and unstructured format [47]. Sensor vendor may use a different data format. Moreover, An uncertainty may occur in the event data or event pattern or both [64]. The uncertainty is from unreliable information sources such as the event source, event inference, and faulty reading instruments that produce incomplete and imperfect data [64].

[78] introduces the COLLECT architecture, which stands for collaborative context-aware service-oriented architecture. The focus of the architecture is to sense the context of incoming data and augment the incoming data with the correct contextual information. This architecture is like the real-time logistics exception decision model, which detect an exception (the context) and enrich the event with information related to the exception.

The main advantage of COLLECT architecture is considering the data generated from the IoT device, which as described in the previous paragraph as massive, perpetual and unstructured. COLLECT done this by leveraging service-oriented architecture, combined with enterprise service bus. COLLECT use service-oriented architecture to ensure the reliability and flexibility of the processing capabilities, While utilising the enterprise service bus to transform the data from the sensors into a generic format.

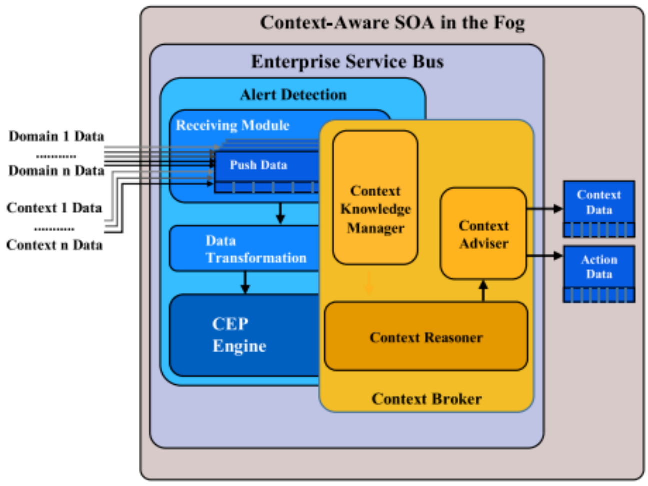

FIGURE 30 COLLECT ARCHITECTURE [78]

In a service-oriented architecture, all functionalities are exposed as web services and are accessible by other services, applications and agents via secured communication protocols [43]. The main idea of service-oriented architecture is to isolate shared resources [67]. In this architecture, architecture components that want to share their data provide web service description. This description provides necessary information for the data consumer to request the service. The fact a service hides the data makes the system flexible and more natural to maintain. The system owner can change or upgrade architecture components easily just by ensuring the new components comply with the operation description. Service-based also means the architecture designer can use middleware paradigm to process the data, orchestrate the process using an execution engine and route the data using an Enterprise service bus. Security and scalability can be implemented quickly, due to service-oriented architecture using the web standards, and the web has complete tools to do security and scalability.

An ESB is a middleware solution that uses the service-oriented model to promote and enable interoperability between heterogeneous environment[79]. ESB typically provides the invocation, routing, mediation, adapters, security, management, process orchestration, complex event processing and integration tooling [80]. Invocation is the capability to send request and response. Routing means the ability to decide the destination of a message. Mediation is the capability to translate the message into a different format. The adapter is the component responsible for providing connectivity with data from the outside world, which may use different standards to communicate. Security means the enterprise service bus provides the capability to be secure. Management means ESB provide logging facilities to monitor and administer the infrastructure. In process orchestration, ESB may be used to execute a business process automatically. ESB may as well include mechanisms for event interpretation, event correlation and event pattern matching which enable event-driven architectures [80]. Finally, ESB may provide the graphical tools to design the routing between services.

Complex event processing can be regarded as a service that receives and matches lower-level events and generates higher-level events, for example, a pair of buy and sell orders may be aggregated into a transaction [20]. Complex event processing can be done using various techniques. For example, automata-based algorithm, pattern query processing, automated rule generation, rule-based systems, formal semantic systems, and the probabilistic data stream systems [21].

From Figure 30, we can see that the data processing flow is as described in Figure 29. First, the architecture adapts and transform the data into a generic format. Then, it processes the data into information by comparing the data with patterns or rules stored inside a repository. The main difference is the COLLECT architecture further defined the techniques to integrate the IoT sensor and process them.

Based on the concept described above, the following list contains design choices for situation aware logistics, to detect the exception and delivers meaningful information:

1. Situation-aware logistics should have the capability to do event absorption, event processing and event action, delivery, and display.
2. The event absorption stage will consist of adapting, validating and filtering the incoming data.
3. Enterprise service bus and service-oriented architecture are useful technique during the absorption phase.
4. The system will use a rule-based detection method to identify exception and determine what kind of information is useful for solving the exception. The user can input the exception rules, based on a model for transportation normal condition.
5. When displaying a notification to the user, the quality of the notification is essential.

4.5.4 LRQ5: Usability principles

The ISO 9241-11 describe usability as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use. In the case of situation aware logistics information system, the usability concerns about humans interaction with the information system. To know what kind of user interface is valid for the user, we need to know about the users, the work they do and about the interaction that takes place when using a system [81].

[81] describe the method to describe a useful user interface from analysing the task that the human user is going to perform from an information system. The steps below describe the task-based user interface design methodology for building useful user interface:

1. Develop an essential conceptual model of the task world.
2. Identify the major tasks and objects that need to be part of the system Depending on the type of application structure the application based on a process or product metaphor.
3. Create navigational paths in the interface structure depending on the task structure
4. Design the presentation using a platform style.

The first step to design a useful user interface is to build the model of interaction between the task that the person needs to do by using the information system. Modelling diagrams such as UML's use case diagram, activity diagram, collaboration diagram, and sequence diagram are suitable tools to do the modelling. However, UML is not the only diagram languages that are usable. We can also use contextual modelling [82] or by adhering to the ontology described in the task-based user interface design books [81].

The next step is to prioritise the task that needs to be in the user interface. The following heuristics are provided in the methodology to prioritise the task:

- The rights a role or agent has for the objects used in the task they are responsible
- for or perform.
- The tasks frequency.
- The events frequency.
- The number of tasks a role is responsible.
- The number of sub-roles a role has.
- The number of levels in subtasks of a task.
- The number of subtasks on the same level as a task.
- The objects used in a task.
- The roles involved in the task.
- The utilised objects in a role.
- The delegated/mandated Tasks

The task prioritisation will give us an idea what to put inside the user interface. Action and information that the user will most likely to use should be put inside shallow navigation path. Mental actions take a considerable amount of time to perform and can make up a substantial portion of the total performance time of tasks. Shortening the time for mental actions can be facilitated by making the relevant information visible in the interface [81].

The final task is to design the presentation. This process includes choosing colours, typefaces, layout, sound and tactile feedback. Colour is useful for conveying a hint to the user. Colour is thought to influence human mood, emotions and expressed aggression [83]. For example, red is associated with negative, danger-bearing emotions, since it is the colour of fire, blood, anger, and sometimes of poisonous or dangerous animals [84]. Hence, in the user interface, red is useful for hinting unsafe action or error that need immediate user attention. Because colour can have a meaning, it is important to use colour appropriately. A user interface that uses too much colour or uses it with incorrect meaning can end up confusing the user and increasing his load. The typeface is the image of the letter. Using a readable typeface is necessary. The layout is the grouping of the user interface elements. The best practice is to group user interface that has relation to task together. That way, information can be retrieved and processed together.

Another way to produce useful user interface is by adhering to well-tested design principles. Schneiderman describe that there are eight golden rules for user interface design: Strive for consistency, enable frequent users to use shortcuts, Offer informative feedback, Design dialogs to yield closure, Offer error prevention and simple error handling, Permit easy reversal of actions, Support internal locus of control, and Reduce short-term memory load [85].

Strive for consistency means similar task or action should be represented inconsistent meaning and design so that the user does not have to learn again, to do a similar task or action, lessen the burden of learning. By providing shortcuts, experts user can do a task more quickly by reducing the interaction required with the user interface. Offer informative feedback is when the user does something, the user interface presents a notification to the person. Design dialogue to yield closure means the task presented by the user interface must have a starting and an end. By permitting reversal of action, the user does not have to worry when doing an action. Internal locus of control is the capability for the user to initiate action on the application, not the reverse. Finally, by reducing short-term memory load, the information displayed should be simple and relevant. Action should be short, and data on multiple pages can be combined.

The list below contains the design consideration for situation aware logistics information system based on the usability principles:

1. Create a user interface based on the task and information surrounding the task.
2. Prioritize frequently used user interface.
3. Use colors, layout, and fonts to give hints to the user.
4. Adhere to the best practices for user interface design.

5 THE PROPOSED DESIGN

This section describes the design of situation-aware logistics information system. I divide the chapter into three sections. In the first section, I will talk about the architecture of the application, physical architecture to gather events, and how it supports the business process. In the data model section, I will explain the type of data used by the systems. Finally, in the user interface section, I will explain the various roles inside the system, and what can they do with the system.

5.1 BUSINESS PROCESS

As discussed in section 2.1.2, the transportation manager needs to do two things when exception happened: create a mitigation plan and notify the affected customer. However, because there is no transported items record, the content of the vehicle needs to be checked first, by a manual worker before the company can take action.

This condition makes the whole exception handling process slow. The slowness may affect other scheduled shipments that use the same vehicle. It can make a sensitive product, such as food and beverages to perish. The process will result in financial loss and might affect the relationship between the company and the customer.

The goal of situation-aware logistics information system is to give the transportation manager the necessary information to do decision making when the transportation business process exception happened. By having the information, the transportation manager can notify the notification earlier and create a mitigation plan before the employee retrieves and sort the items. I believe this solution is quicker because checking the items one by one takes a long time compared to check an information system. The solution also means the transportation manager aware of the condition of the shipment and can respond with appropriate measure immediately. Figure 31 depicts the new business process that uses the situation-aware logistics information system.

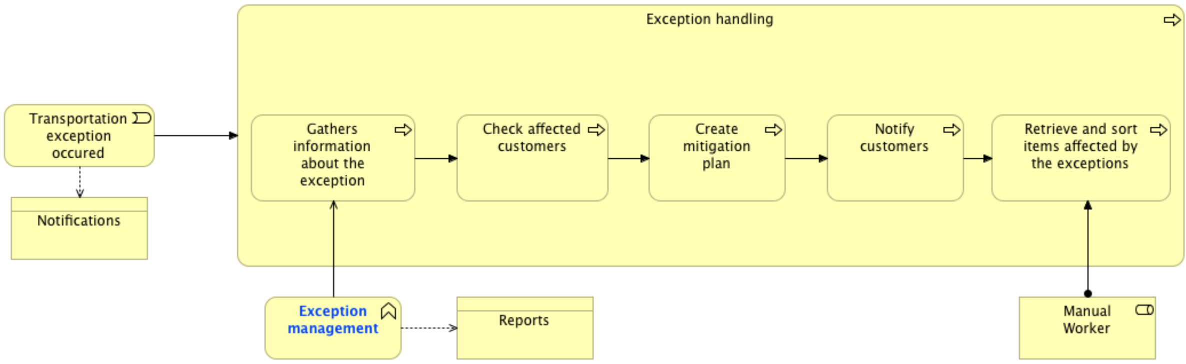

FIGURE 31 EXCEPTION HANDLING PROCESS

In Figure 31, we can see a new business function, exception management highlighted in blue. The transportation manager is responsible for performing this business function. The purpose of the exception management business function is for gathering information about the exception. The transportation manager does this by using the situation-aware logistics manager.

We can also observe some differences between from the baseline business process, depicted in Figure 6 compared to the new business process in Figure 31. The main differences are the order of the procedure to handle the exception. By using the system, the user will know the transported items and the current condition in the field. the transportation manager can notify the affected customer and create a mitigation process before manually inspect the condition of the items.

To gather information condition inside the truck, I need to choose the IoT sensor. The factor that influences me to choose a specific brand of IoT sensors is the type of data that the IoT sensor can provide, the refresh rate capabilities, how the manufacturer will support the IoT sensor and how to integrates the IoT sensor into the architecture.

The first factor is the type of data that the IoT sensor supports. As I discussed earlier in section 3.2, I want to know the current temperature, humidity, and location of the items. The IoT sensor refresh rate should be able to be changed dynamically. The dynamic refresh rate is critical because the temperature may depend on the insulation material, ambient temperature and humidity [86]. If the sensor data refresh rate is too slow, it may be too late when an exception is detected. If the sensor refreshes the data too fast, it may consume unnecessary computing power and communication bandwidth which will inflict cost. A dynamic refresh rate ensures the system can provide enough data resolution depending on the transportation condition while minimising the computing and communication cost. The location of the IoT sensor is essential. If we put it inside the container, then we cannot track the condition of the items when it is being loaded/unloaded. Another option is to attach it to every item transported. This way, we can track the condition of the items from the start of the loading process until the employee unloaded the items at the destination. However, this solution is expensive, because the company must provide IoT devices for every transported item. It also complicates the management of IoT device. The device must be stored, have maintenance and repaired if necessary. It will be tough to keep track of every IoT device as the company assets. As a middle-ground solution, I choose to use IoT sensors embedded in a pallet.

A pallet is a structure to support the movements of items inside the pallets by using forklift, jack or a crane. A pallet has a hole that a forklift can use to lift items in the pallets. Pallets are traditionally made from woods. However, a pallet that is made from woods is easily damaged. Many of the traditional transportation company only use a pallet once. This solution is expensive and not sustainable environmentally. Current pallet manufacturers are starting to use composite material, such as plastics, or covers the pallet with a special coating to make it more durable. Some pallets are also made from recyclable material to be environmentally friendly.

Instead use it just once, the industry is moving toward the concept of pallets pooling. In this concept, pallets are not owned by the transportation company, but by the pallets pooling company. The company rents the pallets and must return the pallets when the pallets are no longer in use or need maintenance. In other words, the pallet pooling company is the entity that manages the pallets life cycle.

By data integration, I mean how the situation-aware logistics system can retrieve the information. One thing for sure, a communication line must deliver the data. The communication line can be wireless or fixed line. The disadvantage of using a fixed line is we cannot monitor the condition of the items before reaching a checkpoint with the fixed line installation. Though, the cost of implementing such a system may be lower and more reliable. As in fixed checkpoint, the data come in batch. We do not have to implement a complex real-time algorithm to process the data

However, this is not what we want. What we want is to have real-time information about the environmental condition surrounding the items. Therefore, wireless communication is necessary. The truck equipped with a mobile gateway, which connects to the internet through a mobile internet connection. The data can be transferred directly to situation aware logistics information system or go processed through another system first, using fog and cloud computing paradigm, described in section 4.5.1.1.

In this thesis, I choose pallet with embedded IoT sensor made by Ahrma. The first reason is, their technology is already on the market, battle-tested. Their smart pallets solution can detect temperature, weight, location, and shock, which is what we want. Furthermore, the smart pallets are managed by Ahrma, using the pallets pooling concept described earlier in this section. The third party management makes pallets lifecycle management simpler. The data is available in real-time through their mobile gateway and can be accessed through their cloud platform using web services. Another important reason I choose Ahrma smart pallet is that Ahrma is part of the DATAREL project. They help this research by providing the IoT devices and documentation for their web services. However, I make the system adaptable, by leveraging an enterprise service bus, so that the system can use any sensor if the sensor provides necessary information to detect exception.

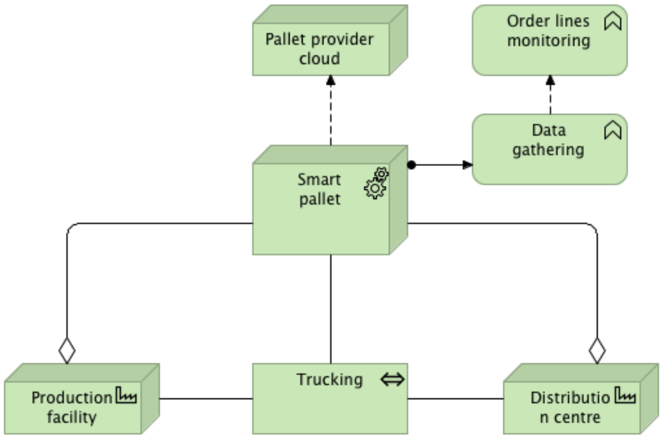

FIGURE 32 INFRASTRUCTURE FOR GATHERING DATA

In Figure 32, we can see how the information system incorporates the smart pallets. Based on the case study, we will use the smart pallet inside the production facility, the truck, and the distribution center. The production facility illustrates the origin, where the items are loaded into the smart pallets and then loaded into the truck. The truck depicts the vehicle, in which the pallets will monitor information for detecting the exception and the distribution center, which represent the destination of the items, and where the items will be unloaded and the pallets ready to be reused.

However, how we know which orderlines are monitored by which smart pallets? The information is essential for knowing which items are affected when an exception happened. Therefore, we need to associate the pallets with the orderlines and disassociate it afterwards. The association is made between the pallet’s identifier (GRAI or global returnable asset identifier) and the orderlines, which is a collection of items for the same customer. SSCC or serial shipping container code identify the orderlines. The smart pallet will transmit not only the measurement values but also the pallet identifier. That way, the system can indicate the condition surrounding the items.

The associating and disassociating steps are performed during the loading and unloading process. The reason is that at that stage, the employees will load the orderlines into the pallets, which means the identifier of the pallets and the orderlines is visible. The identifier is in the form of a scannable barcode.

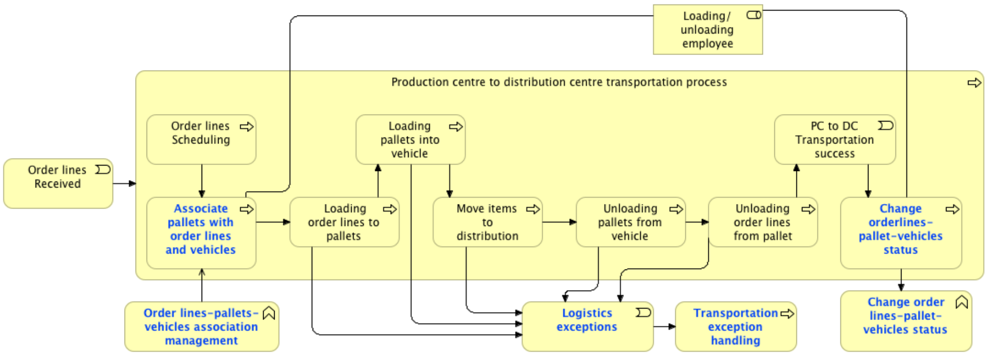

FIGURE 33 PRODUCTION CENTER TO THE DISTRIBUTION CENTER TRANSPORTATION PROCESS

Figure 33 depicts the new business process for doing the production center for distribution center transportation. We can view some addition to the business process. The first addition is the association between the orderlines and the pallets. A handheld scanner will scan the barcode of the pallets. The loading employee performs the process. The pallets start transmitting environment condition value immediately after associating the orderlines with the pallets. Thus, an exception can start being detected by the system. The rest of the process stays the same as the original business process depicted in Figure 5. Though, when the employees unload items, the unloading employee must scan the pallets to dissociate the pallets from the orderlines.

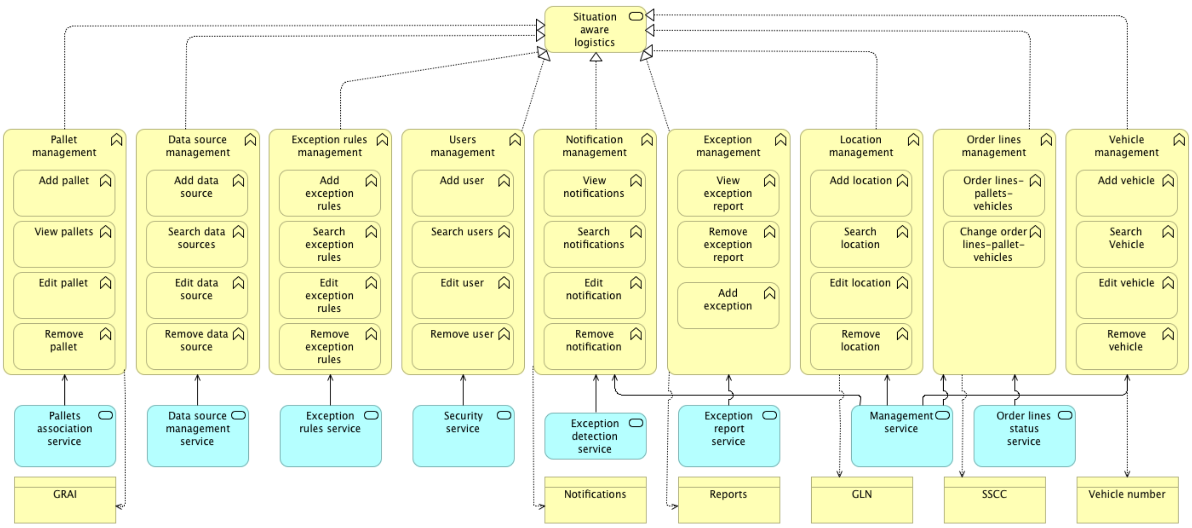

FIGURE 34 SITUATION AWARE LOGISTICS BUSINESS FUNCTION

Figure 34 describes the business function that situation aware logistics must implement to support the new production center to distribution center transportation business process and the exception handling business process. So, a business function is an executable action that can be performed by the user. Referring to the case, the user can be the transportation manager, loading/unloading employee and the driver.

The designed information system supports the business function, denoted blue in Figure 34. The first function is the pallet management. As required by **R5**. The pallet management function is responsible for managing the pallet inside the information system. When a pallet is new, it must be registered with the system before it can be associated or dissociated with orderlines as required by R7. The user must be able to know pallets which in the system, modify the pallet information and remove the pallets. The pallet management function is implemented inside the situation-aware logistics information system and exposed as the pallets service.

The next thing is the user must manage the IoT data, as required by R6. The smart pallet will generate a massive number of events that contains the measurement of the environment condition. This data will be stored automatically by the system. This stored data raised an opportunity to be analysed. Therefore, the architecture should provide data source management capabilities by allowing the user to search and delete the IoT Data.

After that, we need to manage the exception rules, as required by R3. The exception rules are used to detect an exception. The rules model the normal condition for an item category. A category can have multiple rules. Therefore, the user must be able to view, add, modify and remove the exception rules.

The system must have security, as required by NFR2. This is implemented as user management, as required by R13. Only an authorised and authenticated person should be able access various feature inside the system. This is important because the customer information is sensitive. Unauthorized access to the system may lead to data breach or damage to the system. Therefore, the system must have the capability to add, remove and modify users and its role.

The most critical part of the situation-aware logistics is the notification, as required by R2. The system must provide the necessary tools to navigate the notification. Notification must be able to be searched and filtered by the importance of the notification, the reason of the notification, the date and time, by the affected orderlines and by which sensors generated the notification. The user should be able to delete expired notification. Expired notification occurred for example when the mitigation operation completed, or the items successfully transported.

Apart from the notification, the application will generate reports. The type of report that the system can generate includes shipments detail reports, orderline detail reports, shipment notification report and the exception reports. The shipments detail reports contain all the information about the shipments, such as the shipment destination, current location, the vehicle, and the driver, the shipment status and the orderlines. The orderlines detail report includes the detail of the orderlines, such as the type of items, the customer, the orderlines status and the quantity. The shipment notification report will contain the shipment notification. The notification can be for notifying location and shipment status, as well as notification when an exception is detected. The exception reports include the detail of the exception, the affected orderlines, and the customer. This report will be useful for the transportation manager to notify the customer and deciding how to mitigate the exception. The user should be able to generate the reports, view the reports and delete a report that no longer has a purpose as required by R1 and R8.

The rest of the function is for managing the master data. Master data represents is commonly used information. Usually, the master data is static, unchanging. In Situation aware logistics, the master data comprises of the location, vehicles, and the drivers. The master data will be explained further in section 5.3. The user should be able to add, search, edit and remove the master data from the system, as required by R10, R11.

5.2 SYSTEM ARCHITECTURE

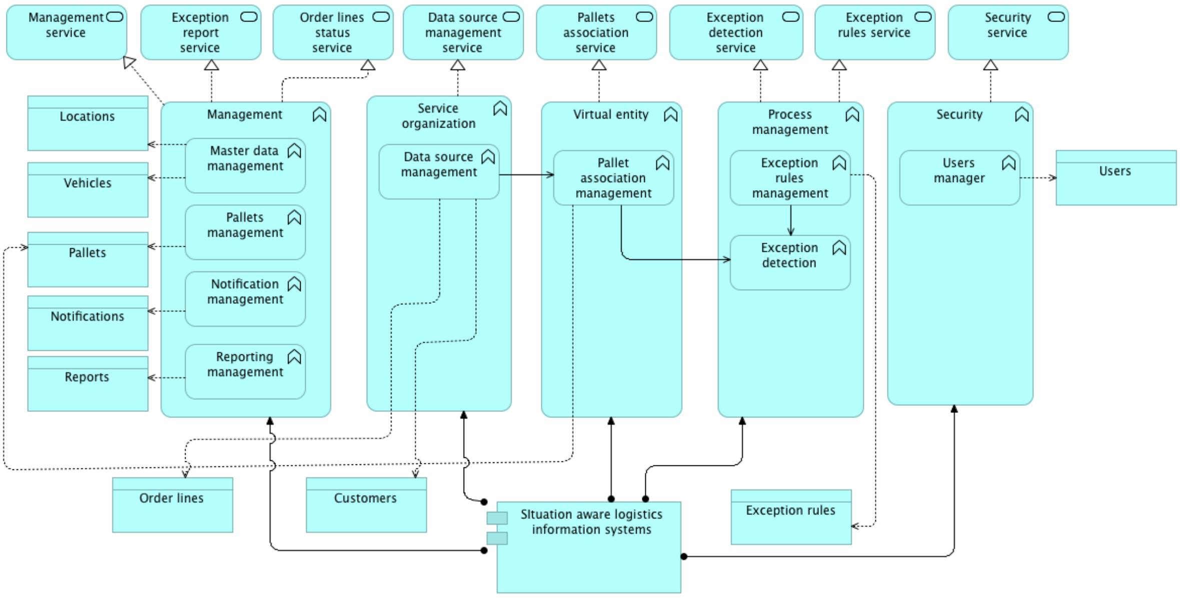

FIGURE 35 THE SYSTEM ARCHITECTURE

To support all the described functions in section 5.1, I design an information system. I based the information system on IoT-A reference architecture explained in section 4.5.1.2. I organised the architecture into five modules. I took inspiration from by the IoT-A functional view depicted in Figure 25 for the division of the module. The first module is the management. This module is composed of the function necessary to support the many management functions described in Figure 34. The management module will connect to the database to provide functions such as querying, viewing, modifying and deleting data object. The data object that the management module comprises the locations, vehicles, pallets, notifications. Additionally, this module also manages the reports.

The service organisation module will provide the capability to manage the data that come from the IoT sensors. The service organisation will perform as the event absorption layer, explained in 4.5.3. The module will validate, clean, enrich and filter the data from IoT sensor. The data then will be stored inside the database.

A virtual entity is the virtual representation of the real objects that is the being monitored. The object that the application monitored is the items and the pallets. Therefore, this module contains functions to manage the association between the pallets and the orderlines as required by R7. Another essential task of this module is for querying and selecting the appropriate exception rules, according to the type of items being transported. In summary, the virtual entity module is for inferring how IoT data should be evaluated based on the pattern of the data.

The exception management module contains the algorithm to evaluate the data from the IoT sensor against the exception rules. This module also responsible for generating the notification when the evaluation process completed, and an exception is detected as required by R4. This module also contains the function to view, add, delete, and remove exception rules. I also implement the association between the items and the exception rules in this module.

The last module that the system has is the security module. The module contains the management pages for adding, modifying and edit the user of the system. The security module also contains the pages for managing the user role as required by **R13**. The user roles are used to limit the user so that the person can only view what the user is authorized.

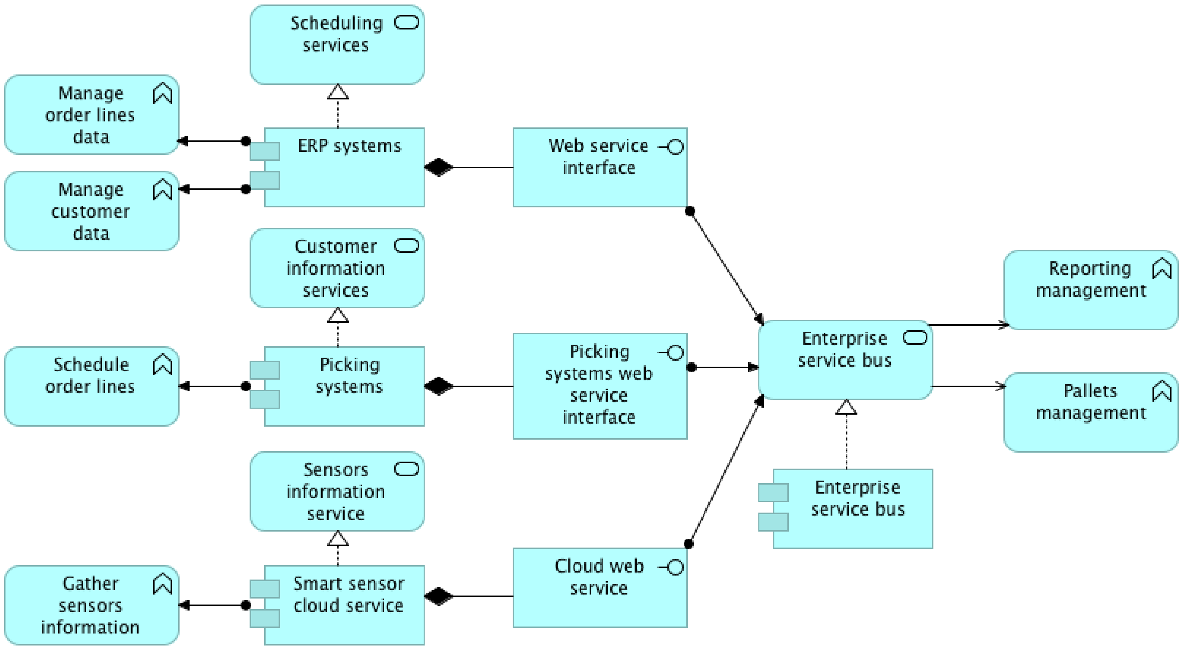

FIGURE 36 COMMUNICATING WITH EXTERNAL SYSTEMS

The system will communicate with the external system. The data retrieved from the external system must be integrated, as required by **R12**. The first external system is the ERP system. The ERP system stores the items, and the customer master data, as well as the orderlines detail. Situation-aware logistics information systems will use ERP systems to retrieve the customer's data and the items data that is not in the internal database. The retrieval process is done so that the user does not have to input the master data manually.

The second external system is the picking system. The picking system will plan when and how the orderlines will be transported. The picking system will send the shipment information to the designed system. This shipment information is essential because it contains information about the orderlines. The transportation manager will use the data for detecting the exception and generating the reports.

The last external system is the smart sensor. In this case, the system will integrate the sensor data from Ahrma smart pallet. The smart pallet will gather the data from the environment, and then upload it to the Ahrma Cloud for pre-processing. The Ahrma Cloud have a web service interface to interact with the external systems. The Ahrma cloud will push the data to the designed system.

From the figure 36, we can see that all data will pass through the enterprise service bus before routed to a specific function inside the application. The reason why I use the enterprise service bus is that the external application may change in the future. For example, by changing the provider of the ERP system or the IoT sensor, the new application may have different data format, that is incompatible with the system. Therefore, I use the enterprise service bus. The enterprise service bus has three main components, the adaptor, the transformer, and the routing.

The web service may use different communication language when communicating with the systems. Some web service prefers to use the SOAP protocol, while the other use the REST protocol. There might make a difference in how the external services establish the communication channel, and how the system provides authentication and authorisation. Examples of communication protocols include TLS, SFTP, SSH. The adaptor has the necessary tools to communicate via the various protocols

The transformation components contain the transformation algorithm defined by the system designer to transform the data. The algorithms will transform the incoming data into a common data format and perform validation. The system must validate it against the expected data type and its constraints. The enterprise service bus contains the common data format definition, in this case, the format for the customer, sensor data, shipment, orderlines, items, vehicles, drivers and locations. The study describes the data format in more detail in section 5.3.

After the data is in the common format, the data must be delivered to the correct function of the application. The routing components do this function. First, the ESB will investigate the message. Then, it will check the defined pattern for the incoming data. The master data from the ERP system and the picking system will be directed into the reporting function of the application. The sensor data will be redirected to the exception rules evaluation function. In this case, the service organization will function as the data recipient.

5.3 DATA MODEL

To assist in decision making when an exception happens, the user needs information. The person must know what items are affected by the exceptions. From the case study, we know that a truck contains a shipment. A shipment contains a group of orderlines that transported together. A shipment has several attributes: Destination, delivery times, vehicles, driver and the status of the shipment. GSIN or global Shipment identification number identifies a shipment. GSIN is an international standard for identifying a shipment.

The shipment status can be new, loading, in transit, unloaded and finished. The status is new when the planning system plans the shipment. When the shipment is ready to be loaded, the loading employee will change the shipment status to loading. When the items are loaded and ready to leave to the destination, the driver can change the status to in transit. After the shipments arrived at the destination, the unloading employee must unload the items. After all of the items are unloaded, the employee can change the status of the shipment to finished.

An orderlines represents a group of items that a customer ordered. An orderlines have the following attributes: Identifier, customers, status, and the list of items. The list of items contains the name of the item and the quantity of the items. Identifier for an orderlines is SSCC or serial shipment container code. SSCC is another global standard by GS1. Like the shipment's status, orderlines status can be new, loaded, in transit and unloading. The orderlines should save the location history for every movement of the orderliness to satisfy the traceability requirements.

The employee will place the orderlines inside a pallet, so the orderlines should have references to the assigned pallets. Similar with the shipment's status, the orderlines state can be new, loaded, in transit or unloaded. GS1 standard GRAI, of global returnable asset identifier, identifies a pallet. Aside from the identifier, a pallet has status as well: operational, in maintenance or out of order. Operational means the pallet is in excellent condition and ready to carry orderlines. In maintenance imply the pallet is undergone condition check or in repair. Out of order means the pallet is broken and need maintenance.

To globally identify a product, the system will use another GS1 standard, the GTIN which is an abbreviation for global trade item number. The item entity consists of the identifier (GTIN), the name of the items, description of the items, and physical properties, such as the weight, height, width, and depth. The physical properties are useful for designing mitigation operation when an exception happened. For example, in the case of vehicle breakdown, the exception manager can decide which truck size is appropriate to be dispatched to replace the broken truck.

The GS1Prefix identifies the customers. The properties inside the customer objects are the company name, contact name, phone number, email address, and address. The driver entity stores the name of the driver, address and contact detail. GS1 identifier GIAI or global individual asset identifier identifies the vehicles. The system will store the vehicle plate number and the vehicle type. The type can be a vehicle equipped with a refrigerated ISO container, or non-refrigerated ISO container.

There are two types of the location stored in the system. The first type is the location by address. The GS1 Global location number identifies the location. The system uses it to represent a known location, for example, the production center, distribution center and customer delivery address. The second type is a location by coordinate. This location type store latitude & longitude and used for recording the exact current position of the vehicle.

The system must store the event gathered from the sensors. I took inspiration from the EPCIS architecture described in section 4.5.1.3. The system will capture the "when," "who," "what" and "why" an event is generated by the IoT sensors. The attribute "when" indicating the date and time of the event. The who will contain information about which sensor raises the event. Because in this thesis I am going to use a smart pallet, the attribute "who" will be filled with GRAI, the GS1 identifier for pallets. The value captured by the sensor fills the "What" column. The "why" will be filled with the type of data that is returned by the sensor. Based on the explanation in section 4.5.2, the type of data that the system will accept will be temperature, humidity, shock, and location. For example, if the sensor is for capturing environment temperature, then the "what" attribute will be filled with the measured temperature value and the "why" attribute will be filled with the keyword "temperature."

Situation-aware logistics information system will detect an exception by using rule-based, complex event processing techniques described in section 4.5.3. Therefore, there is a need to store the rules to detect the exception. The exception category models the normal condition of an item. Multiple exception rules are in an exception category. For example, ice cream must be kept at the standard temperature for transporting: -18°C. It may not experience shock above 9G. Multiple exception rules are necessary to describe the items normal condition. Ice cream on the market has many variety and brand. However, the normal condition of transportation is generally the same. Hence, it is essential to have the capability to assign the same exception category to product variety. Nonetheless, if a similar product has different handling condition, the user needs to add a new exception category with different rules.

The rules comprise of the name of the rules, threshold value, comparison operator, message, event rates, and aggregation function. The comparison operator is used to compare the value of the sensor to the threshold value. The comparison operator is a standard mathematical operator such as less than, more than, equal, less than equal and more than equal. I define the event rates so that multiple sensor event data can be aggregated and compared together against the threshold. Event rates of one mean every incoming event will be compared with the threshold. The system will aggregate the events, according to the selection of aggregation function: minimum, maximum, average, sum and count.

When the events are below or above the normal condition, the system will generate a notification, and it will be displayed to the user. A notification is used not only for alerting exception, but also can be used for the user to monitor location or status changes. This information is stored inside the notification entity by the attribute notification category. A notification entity also consists of the observed date and time, which indicates the creation time of the notification. Then, a notification contains a reason why the notification exists. For example, the system may fill the notification with the message defined in an exception rule. The next thing that a notification has is the priorities. A notification may have "information," Warning" or "Immediate attention." Notification that has "immediate attention" level should be the first to be presented to the user.

Along with the "Warning" level, it usually triggered by violating the normal condition of items, expressed in exception rules. Example of "information" level notification is location and status change notification. When the notification is raised because of an exception, it is essential to know the triggers of the. Hence, the system must store it inside the notification entity. A notification also contains the identifier for orderlines or shipments. The reason is that a notification is for looking the current condition of an orderline or shipment.

Figure 37 shows the entity relationship diagram for the situation-aware logistics information system.

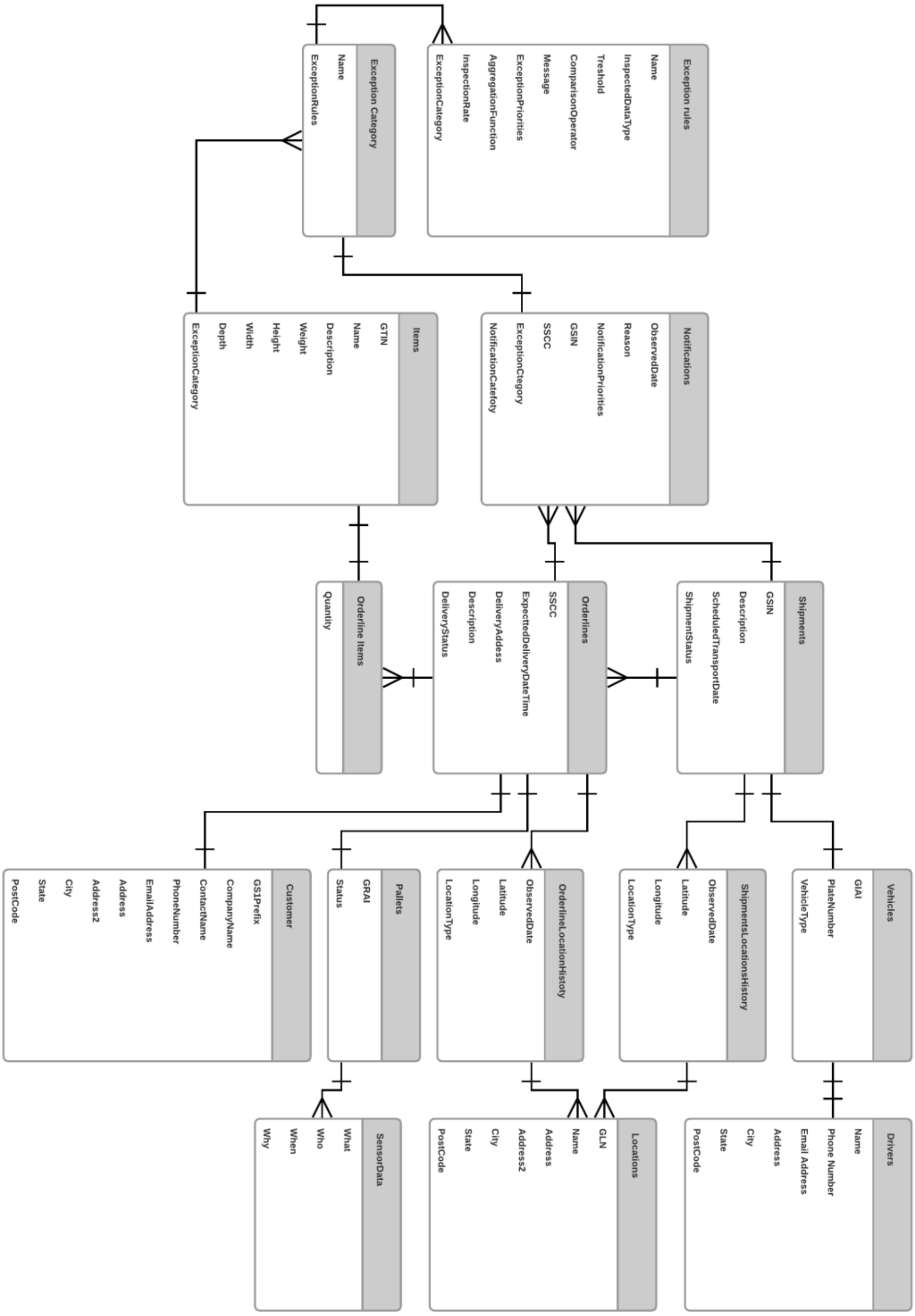

Figure 37 Entity relationship diagram

5.4 USER INTERFACE

As required by **NFR1**, the user interface is one of the critical aspects of the application. It is vital because the user interface serves as the bridge between the human user and the system. To design a useful user interface, I will use the task-based user interface design methodology discussed in section 4.5.4.

The first step in the methodology is to list the task that the user can execute inside the application. Figure 34 describes the tasks, represented as a business function. Then, I classify which roles are responsible for the task. I decide the role by analysing, which stakeholders are directly interacting with the system, based on their job description defined in section 3.1. Based on the stakeholder analysis, three stakeholders fit the requirements. The loading/unloading operator, the system administrator, and the transportation manager. Figure 38 describes the relationships between the user roles and the task.

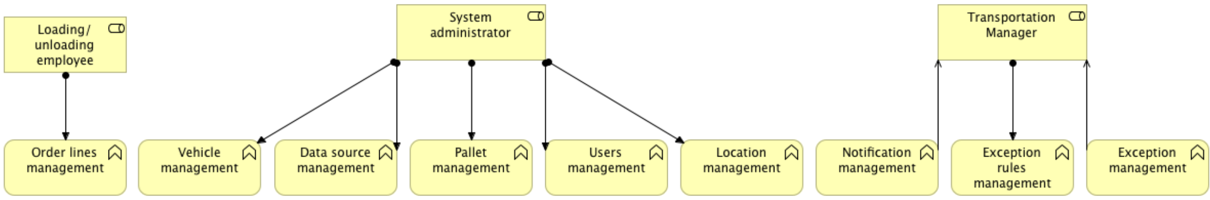

FIGURE 38 USER ROLE IN SITUATION AWARE LOGISTICS

To associate and dissociate the orderlines with the pallets, the user must input the identifier for the pallets and the orderlines. Because the loading/unloading employee is going to load the orderlines into the pallets, it is the perfect moment to do the association of the orderlines and the pallets. The employee must scan the barcode inside the pallets and the orderlines. The user interface must show feedback whether its success or a failure.

The system administrator will be responsible for administrating the master data gathered from the external application. The role can add, modify and remove the master data such as the vehicle, pallets, users, and the location.

The transportation manager is responsible for deciding the correct response to an exception. First, the person must know when an exception is happening. The system will achieve it by generating a notification. Thus, the user must be able to manage the exception.

As described in the previous section, for the system to detect an exception, there must be a user that input the model for items normal condition. I choose the transportation manager to do the task because the transportation manager handles the exception daily. The person can tweak the detection parameter based on the experience quickly.

To create a mitigation operation, the transportation manager needs to know the surrounding condition. The user interface must provide a searchable user interface that contains information about the current location of the shipment, the reason of the exception, the affected items, and the affected customers.

After defining the task and the user, the next step is to create the navigational paths and design the presentation. I separate the user interface into three primary user interfaces. The user interfaces for associating/dissociating the pallets, dashboard for exception management and administrative pages. The following section will discuss the user interface design.

5.4.1 THE LOADING/UNLOADING EMPLOYEE ROLE

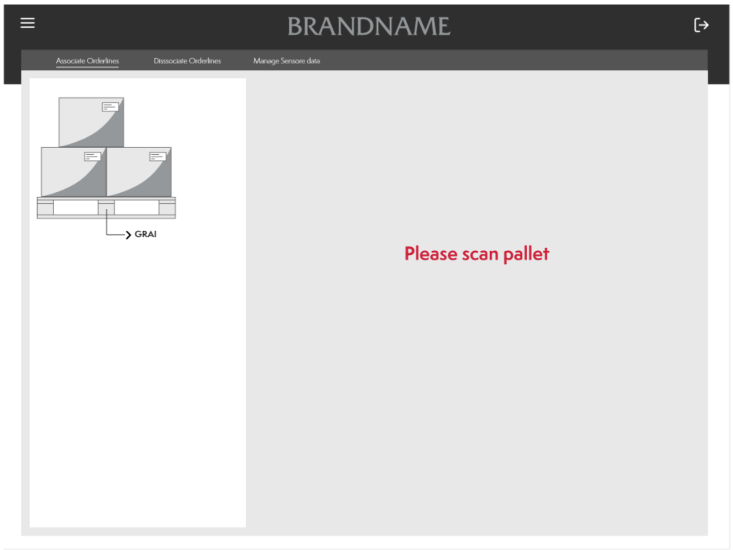

FIGURE 39 SCAN PALLET USER INTERFACE

In figure 40, we can see the user interface for scanning the pallets. The pallets will be scanned using a handheld barcode scanner. On the top of the user interface, we can see the navigation bar. There are two tasks that the role can do: associating and dissociating orderlines. The column on the left gives the loading/unloading employee guidance on the location of the barcode. The right column provides the user with feedback, and advice on what to do.

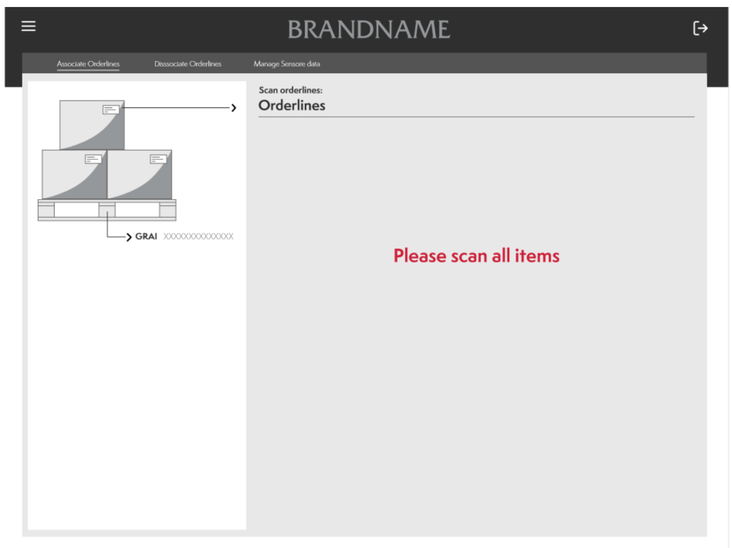

FIGURE 40 SCAN ITEMS USER INTERFACE

Right after the loading/unloading employee scan the pallet, the user will be presented with the second screen. In this second screen, the user must scan the orderlines inside the previously scanned pallet. On the left column, the user can see the scanned pallet identifier (GRAI). Moreover, the user can see the location of the orderlines barcode. On the right column, we can see the instruction, and buttons to finish the item scanning process, to cancel all the association processes and to go back to the pallet scanning user interface. The employee will also get feedback when the person scans the orderlines, complying with the eight golden rules user interface heuristics.

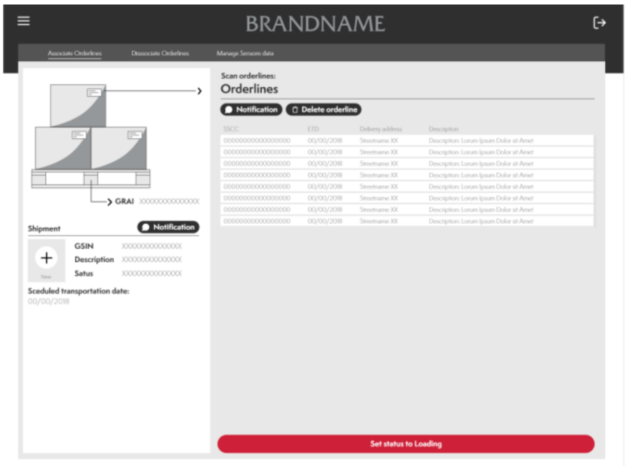

FIGURE 41 PALLET-ORDERLINES ASSOCIATION USER INTERFACE

After the user finished scanning all the orderlines, the system will present the user with a review screen. On the left column, the user can see the details of the shipments. On the right column, the user can review the scanned orderlines, and delete incorrect orderlines. The user can also manually raise exception notification when something is happening during the loading process as described in Figure 33. After the user is done reviewing the orderlines, the user can go back to the previous screen to scan more orderlines or click "set status to loading" button to finish the orderlines-pallets association process. The user will be brought back to the scan pallets barcodes pages as described in Figure 39.

5.4.2 TRANSPORTATION MANAGER DASHBOARD

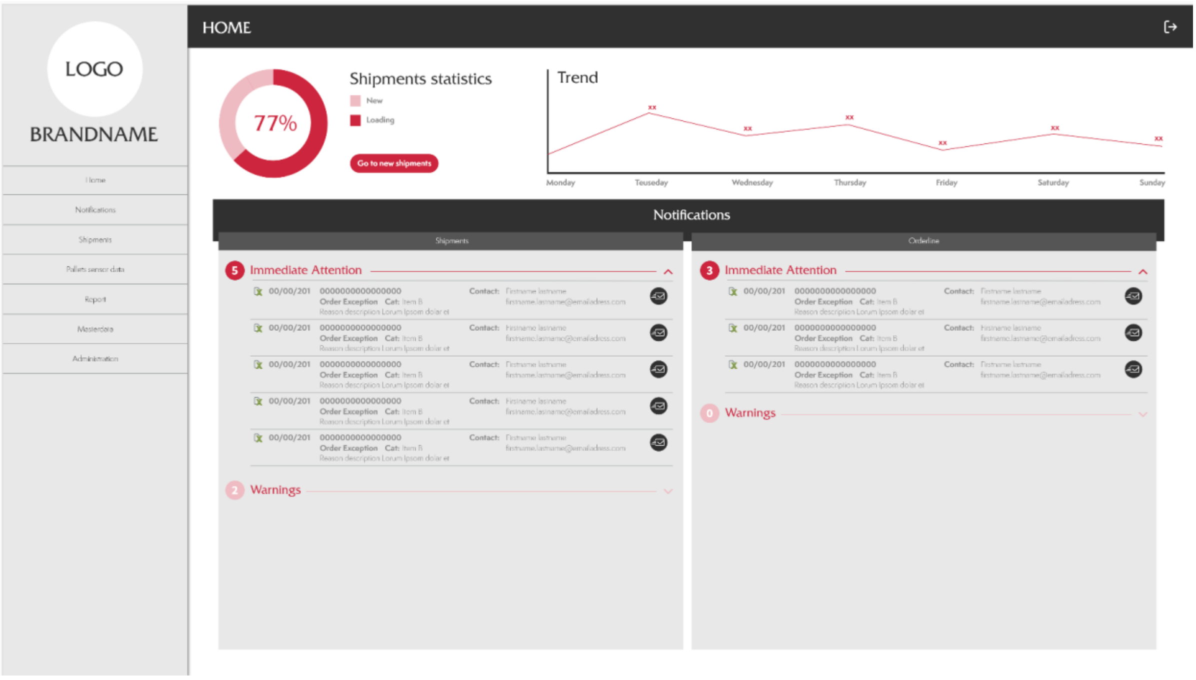

FIGURE 42 TRANSPORTATION MANAGER DASHBOARD

The transportation manager is responsible for mitigating exception. Consequently, the person needs to know what kind of exception is happening at the current time and location. The system provides the s capability as a dashboard for the transportation manager. Figure 42 contains the mock-up for the dashboard. The dashboard contains a list of notifications, that can be raised by an exception, status changes or object movements. The user interface presents the notification in three lists, based on the priority of the notifications. There is three level of priorities: Immediate attention, Warnings, and information. The system shows the notifications at the immediate attention list when the system detects a severe problem that needs a fast response. The system will present notifications on the warning list when the system detects an abnormal condition that is not affecting the transportation process. The system will show notifications in the information list when the orderline or shipment status is changes or the location of the orderlines changes.

On the top of the dashboard, we can see statistics of the shipments. The left column contains the percentage of today's shipments. On the left of the user interface, there will be a line graph that shows the trends for notifications that is in the "Immediate attention" categories. This graph can show the transportation manager the performance of the trans day. If the amount of exception on the day is abnormally high, the transportation manager can assume that something terrible is going on and do an investigation on the operation.

The navigation is on the left side of the user interface. The navigation contains another task that the transportation manager can do. The task includes: Managing the notifications, View shipment detail, View reports, and administering the master data. I will describe these tasks will in the next section.

When the transportation manager clicks the notification, or click the shipments navigation menu, the system will present the shipments detail page. The shipments detail pages the pages comprise of multiple tabs. The first tab is the shipment detail pages. It contains the current location, shipments destination, the vehicle and the driver of the shipments. The next tab is the orderlines tab. The tab contains the orderlines that are in the shipment. When one of the orderlines clicked, the user can view further detail of the orderlines, such as the items inside the orderlines, the customers, the orderline history, and the pallet association. Figure 43 below describe the mock-up for the shipment detail.

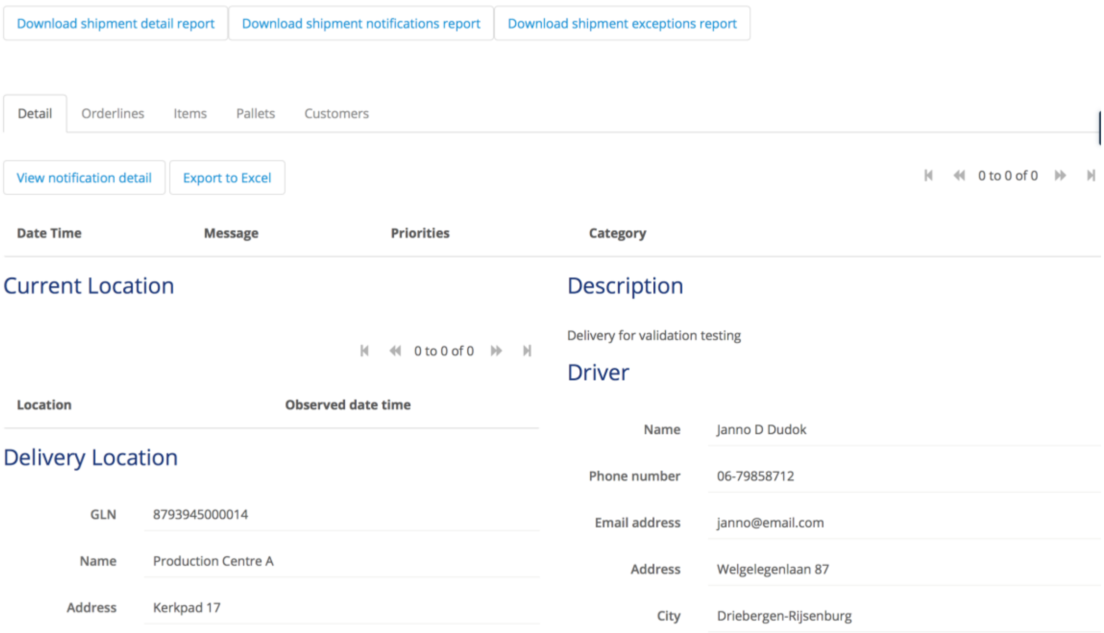

Figure 43 Shipment Detail

The items tab contains a list of the items inside the customer; the list provides information about the name of the item, the barcode, which orderlines the items belong to, and various dimension measurements, such as the weight, height, width, and depth. This information may be necessary for the transportation manager to create mitigation operation.

The next tab is the pallets. In the tab, the user can view every pallet in the shipments. We can also view the associations between the pallets and the orderlines. The tab contains a table with three columns: GSIN, SSCC and the GRAI. The user can download the data by clicking export to excel button.

The final tab on the shipment detail page contains the list of customers. In this tab, the applications present the user with a list of the customers. The list contains the customer's contact detail and the associated orderlines. Like in the previous tab, the user can download the information into excel. The transportation manager may use this feature to process the data further.

6 Prototype implementation

I use the Scrum methodology to build the prototype. I have explained the scrum methodology in section 1.3.2. The first thing I did is to create the user stories and generate sprint from it. I generate the user stories from the user requirements and the described target architecture. A user story is a means to capture a description of a software feature from an end-user perspective. A user story has the following format “*As a < type of user >, I want < some goal > so that < some reason >*.” When building the prototype, I did two sprint iterations.

As a loading/unloading employee, I want to report the location of the shipment, to let the transportation manager, know the location of the orderlines.
As a system developer, I want to simulate the scheduling system to test the system capability to retrieve order lines schedule.
As a system developer, I want to create a data model, so that the application can handle the data used for the exception business process.
As a system developer, I want to incorporate IoT sensors to provide exception detection for the transportation manager.
As a transportation manager, I want to view an exception report, so that I can create a mitigation plan and notify the affected customers.
As a system administrator, I want to be able to remove old data from the sensors, so that the system is not burdened by it.
As a loading/unloading employee, I want to associate order lines with pallets, so that the transportation manager can track the status of the order lines.
As a transportation manager, I want the system to detect the exception and generate a notification so that I can know when something happens with the delivery as soon as possible.
As a transportation manager, I want to add, remove and update exception rules, so that the system can detect exceptions, based on my rules.
As a transportation manager, I want to view, search and remove notifications, so that I can view the notification history and delete unnecessary notification.
As a vehicle operator, I want to report an exception, so that the transportation manager can create a mitigation plan and notify the customer.
As a system administrator, I want to be able to search data from the IoT sensors, so that I can check whether the sensor is working correctly.
As a system administrator, I want to the capability to search, add and remove IoT sensors from the applications so I can manage the sensors.
As a system developer, I want the functionality to add, search, modify and remove users from the application so that only authorised users can access the system.

TABLE 8 FIRST SPRINT USER STORIES

As you can see from Table 8, The first sprint focuses towards building the functionality of the application. I further break down each of the user stories so that I can create a workable daily task. From the perspective of time, I finished the first sprint in two weeks, working around 8 hours a day (Excluding the weekend) from the middle of July 2018 until the beginning of August 2018.

After the first sprint finished, I request for feedback from the product owner, in a meeting called the sprint meeting. In this meeting, I get a various opinion from the experts. I use the opinion for building the second spring. While the first sprint focuses on building the prototype, the focus of the second sprint is further refinement of the prototype and preparation for the validation phase. The second sprint is executed for two weeks, during the beginning of August until the middle of August 2018. I describe the user stories in table 9.

As the user, I want to have easy to use user interface, so that I can work more effectively.
As the developer, I want the system to have a clear data model so that I can maintain the application thoroughly.

As the developer, I want the components to be separated into modules so that I can reuse the components effortlessly.
As the developer, I want to add a data cleaning function, so that I can test the system from a known data state.
As the developer, I want to add data injecting function, so that I can test the system from a known data state.

TABLE 9 SECOND SPRINT USER STORIES

Situation-aware logistics information system is built on top of the physical infrastructure. I use cloud services from various vendor to provide the computing resources, data storage, data presentation, and IoT device management services. Ahrma provides the first service. The services provide the management capabilities for the smart pallets. The Ahrma cloud has the feature to track the usage patterns of the smart pallets, shows the current location of the pallets and provide reports about the condition of the smart pallets. This data is available as a web service.

The next cloud service that I use is the Mendix cloud. Mendix is a platform for tools to build, test, deploy and iterate applications. It provides complete application construction lifecycle, based on the Scrum methodology. The cloud services provide data storage services, processing capabilities and present the result it as a webpage on a desktop or mobile platform. I choose Mendix because of the low-code capabilities and integrated lifecycle, making developing and maintaining the application easier. The low-code capabilities mean we can build the application by defining an algorithm with a diagram, which they claimed to be faster than to write codes.

The last cloud service that I use is the Enterprise service bus provided by eMagiz. eMagiz is an enterprise service bus implementation, with a twist. Like Mendix, eMagiz offers the capability to manage the lifecycle of the based-on the scrum methodology. eMagiz provides the tools to communicate with various web services protocols, along with design tools for defining common data format, validation, data transformation, and routing. I choose eMagiz not only because it is suitable to the architecture requirements, but also because eMagiz is one of the stakeholders of the DATAREL project. Figure 45 below describes the infrastructure for situation-aware logistics.

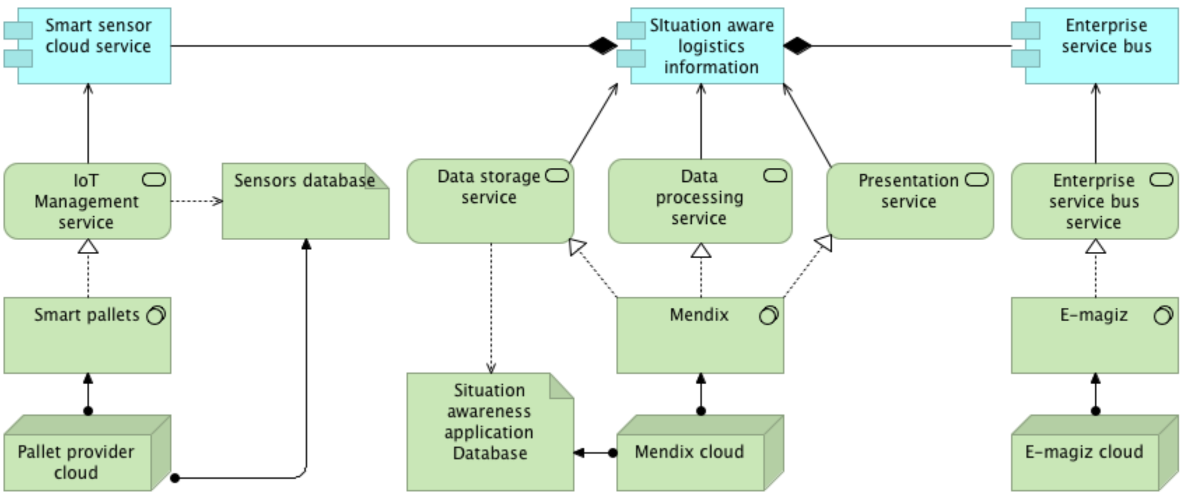

FIGURE 44 INFRASTRUCTURE FOR SITUATION-AWARE LOGISTICS

7 VALIDATION

7.1 HYPOTHESES

The next step after building the artifact in the design science methodology is to validate the result. This step is necessary to measure the effectiveness of the artifact to solve a problem. In this thesis, the research question that is related to the validation process is the Research Question 4: *How the system performs reducing risk?* This question is important to satisfy **NFR3** and **NFR4**.

The research question is too broad to be answered. So, I break down the question further. First, we need to define, what is the measurement of performance? To answer the question, we need to investigate the purpose of this thesis. The goal of this thesis is for designing an architecture for situation aware logistics information system.

Enterprise architecture applies architecture principles and practices to guide organisations through the business, information, process, and technology changes necessary to execute their strategies [87]. Therefore, we can investigate measuring two things: The completeness of the architecture to satisfy the stakeholder's desire, and the actual performance of the architecture implementation to solve the problem context. The completeness of the architecture can be measured by looking at the stakeholder's desire, create a scenario that represents the desire and feeds the scenario to the prototype implementation of the architecture. I formulate the hypothesis as: *By using the system, The Stakeholders' desire will be fulfilled **[H1]***

The performance of the implementation can be measured by looking into the changes that the architecture made into the business process. There are three significant changes to the business process in the proposed architecture, described in section 5. The first change is in the loading and unloading business process. In the proposed design, the new process adds the scanning process during the loading and unloading process. The addition of the scanning process can make the business process slower. The slowdown may cause delay and potentially incur a financial loss. In my assumption, the scanning process should not or only minimally disrupt the loading business process. For validating the assumption, I believe this assumption should be validated. I formulate the assumption as the following hypothesis: *By using the system, time to load the items should increase, but it will be negligible [**H2**].*

The next changes happen to the exception handling process. Instead of manually checking the transported items, the transportation manager uses the system to check the transported items. Using the information system is quicker than manually check the items. I designed the information system to provide the necessary information to make the decision. This assumption must be validated. I use the following hypothesis to model the assumption: *By using the system, the exception can be detected faster, leads to faster customer awareness [**H3**].*

In my assumption, the effect of faster decision making may lead to the saving of the items that encounter an exception. I firmly believe the assumption, especially in the case where the vehicle transported multiple types of items, which have different temperature requirements. The transportation manager can prioritise the objects that will perish If not handled just in time. To validate the assumption, I devise the following hypothesis: *The knowledge of which pallets contains sensitive* *items during transportation exception can minimise the items loss because the pallet can be prioritised during the mitigation operation* ***[H4].***

In summary, I use the following hypotheses to validate the system:

- *By using the system, the Stakeholders' desire will be fulfilled* ***[H1]***
- *By using the system, time to load the items should increase, but it will be negligible.* ***[H2]***
- *By using the system, the exception can be detected faster, leads to faster customer awareness.* ***[H3]***
- *The knowledge of which pallets contains sensitive items during transportation exception can minimise the items loss because the pallet can be prioritised during the mitigation operation* ***[H4]***

7.2 EXPERIMENT DESIGN

To test the hypotheses, I design various experiments and the tools to help to perform the experiments. The design science methodology describes various method to do validation. In this thesis, I will use two methods: The expert opinion and the single case mechanism. In the expert opinion, I submit the design of the artifact to a panel of experts, who imagine how such an artifact will interact with problem contexts imagined by them and then predict what effects they think this would have. The experts are working at Cape Groep and have many experiences working on the logistics information system. Furthermore, one of the experts is directly in communication with the end client.

If the predicted effects do not satisfy the requirements, this is a reason to redesign the artifact [8]. In the single-case mechanism, the researcher must build a prototype of a program, build a model of its intended context, and feed its test scenarios to observe its responses. The responses, good or bad, are explained regarding the mechanisms internal to the program or the environment [8].

I will use expert opinion method to evaluate [H1]: *By using the system, the stakeholder's desire will be fulfilled.* I use the expert opinion method because of the feasibility. I gather the experts on the stakeholders that directly interact with the system in one room and presents them the application. Then, they comment on the features of the situation-aware logistics Information system, what is acceptable and what needs to be improved. Additionally, I use the results of the motivation analysis and presents the interaction between the components that can satisfy their desire.

For the rest of the hypotheses, I use the single case mechanism validation method. To actualise it I develop a variety of experiment protocols. The experiment protocols contain the role of the test subjects, the measured variables and the procedure to validate the results. I define the experiment protocols so that the experiment can be repeated. The repetition capability is essential, as scientific theories require the experiment to be repeatable [88].

From the hypothesis, *by using the system, time to load the items should increase, but it will be negligible*, we can determine the experiment subject, independent variables, and the dependent variables. Because the hypothesis is about the loading/unloading process, then the experiment subject is the person who is responsible for the process, the loading/unloading employee. Independent variable is a variable that will be changed during the experiment, while the dependent variable is the variable change that is anticipated to change as the effect of changing the independent variable. Based on the hypothesis, we want to know whether the time to load the items is negligible or not. So, the independent variable should be the loading process before using the system and after using the system and the dependent variable should be the time to load the items. In other words, in the experiments for validating **[H2]**, I must compare the time before using the system, and after using the system.

I decided to experiment by simulating the process of loading the items, instead of doing it in the real world. The reason is to isolate the variable so that the experiment will not be contaminated with variables that may affect the result of the validation. To accomplish the experiment, I designed a tool to help to simulate the loading process, called the loading simulator. I made a validation tool using HTML and JavaScript. Figure 45 contains the screenshot of the application.

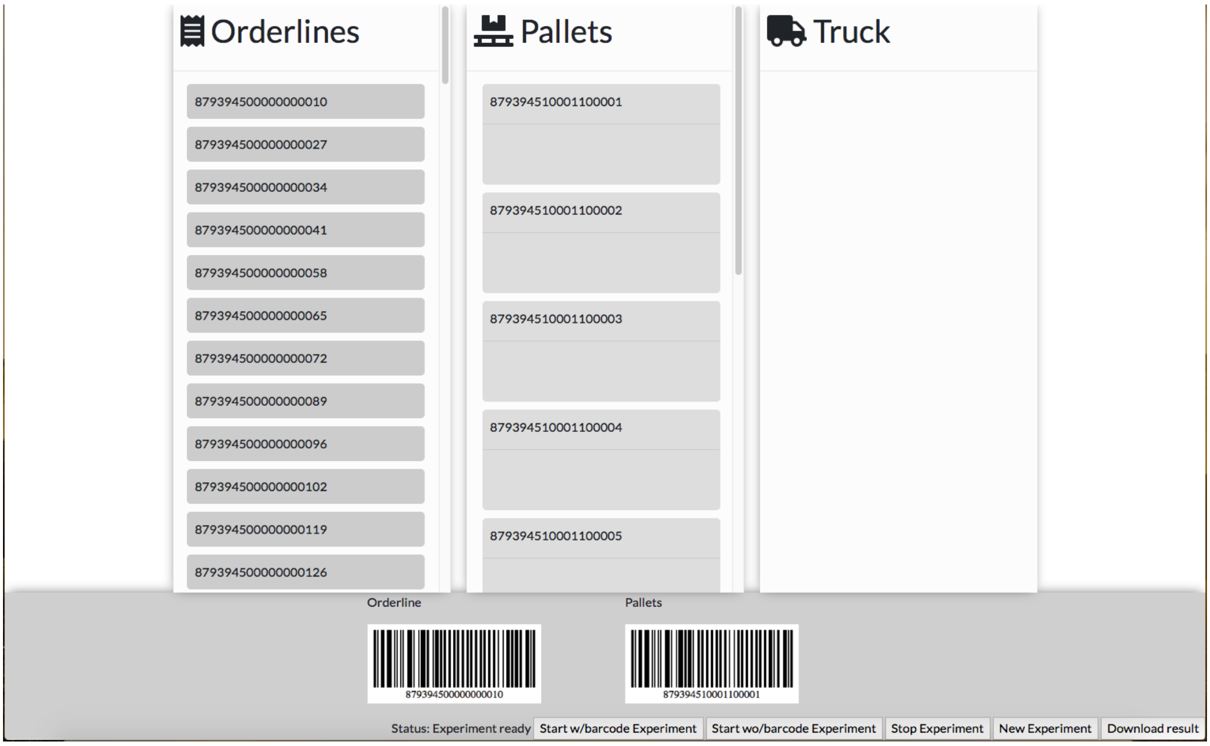

FIGURE 45 LOADING SIMULATOR

From the figure 25, We can observe the loading simulator comprised of three columns. The first column contains simulated orderlines; the second column holds the pallets. The third column represents the truck. The first thing that the test subject is to click start experiment. The system will automatically record the starting time. There are two types of the start button. With barcode scanning and without barcode scanning, representing the old loading process and the new loading process.

After the test subject clicks the “start w/barcode experiment” button, When the test subject clicks the pallet, the barcode for that pallet will be shown on the bottom of the simulator. The same thing will happen when the test subject clicks one of the orderline. The test subject must scan both barcodes into the system. After that, the test subject must drag the scanned orderlines into the pallet column. I use the dragging process to simulate the action of loading the orderlines into a pallet. One pallet will contain nine orderlines. When the pallet is full, the test subject must move the pallet into the vehicle column. The test subject can only click stop experiment when all the orderlines are in the pallets, and all the pallets are in the vehicle column. When the test subject clicks the stop button, the tool will record the end time.

Based on the starting time and the end time of the experiment, I can determine the duration of the loading process. The loading duration is the variable that I am trying to measure. I also observe how the test subject performs the task, like where the test subject most likely make a mistake or perform slowly. Table 10 below contains the experiment protocol for **[H2].**

Hypothesis	*By using the system, time to load the items should increase, but it will be negligible.*
Independent variables	The orderlines scanning process
Dependent variables	The time to scan the whole process
Subject	Loading/unloading employee
The baseline procedure	1. Open the loading/unloading simulator 2. Click start w/o experiment. 3. Drag nine orderlines into the pallet's column. 4. Drag the pallets into the truck column. 5. After the pallets loaded into the truck, click stop experiment.
The procedure	1. Open the loading/unloading simulator 2. Login as loading/unloading operator in situation aware Logistics IS prototype. 3. Open the associate orderlines page in situation aware Logistics IS. 4. Click start w/ barcode experiment. 5. Click a pallet. The system will show the pallet barcode on screen. 6. Click the orderline. The system will show the orderline barcode. 7. Scan pallet 8. Scan orderline 9. Drag orderline into the pallet's column 10. Drag the pallet into the truck column 11. After the test subjects load, the pallets into the truck, click stop experiment.
Assumptions	1. The truck contains 11 pallets 2. There are 99 orderlines to be transported. 3. One pallet can contain at maximum nine orderlines. 4. The truck is full. 5. The items transported is the same. 6. The employee will use standard 1-dimension laser barcode scanner.

TABLE 10 EXPERIMENT PROTOCOL FOR [H2]

For examining the hypothesis: *By using the system, the exception can be detected faster, which leads to better customer experience and faster in exception mitigation decision making,* we need to investigate the time to notify the customer when an exception happened. To do that, we need to simulate the exception handling process, explained in section 5.1 and record the time from the point of exception detection until the transportation manager notifies all the affected. The exception handling process requires several inputs: the exception and the complete information about the transported items.

As described in section 2.1.2, to decide a mitigation operation in the exception handling process, the transportation manager needs to know the current location of the vehicle, list of affected items inside the vehicle, person in charge for transporting the items and the customers. To provide the information, I decided to generate some of the required data. Compared to real data, I have more control without performing steps such as sorting, filtering, and anonymising the data. For data with personal information, such as customers, drivers, and users, I generate it by using personal information generator that is available freely on the internet. For time essential data, such as the shipment information, the pallets, and the orderlines, I generate it by hand, to make sure the shipment date and time synchronised with the date and time when I experimented. For the experiment, I only need one shipment. I also generate other data such as the pallets and the orderlines. A standard 20 feet container (width 235cm & height 589 cm) can hold 11 Euro pallets (width 80 cm & height 14.4 cm). Therefore, with the assumption the container is fully loaded, the shipment will contain 11 pallets. To ease the testing, I assume the shipment will contain 99 orderlines. The orderlines will contain randomised customer. A customer may have multiple orderlines in the shipment. An orderline contains multiple quantities of items. From an interview with logistics experts, usually, a container will be filled with items of the same type. The reason is, each of the items has a specific condition requirement. For example, the required condition for transporting milk is +6°C while the required condition for transporting meat is -10°C [76]. Based on that knowledge, I assume the vehicle will transport the product with the same type.

To simulate an exception, we need two things. The first one is a normal model of an item. The second one is the sensor event. The normal model will be used to detect an exception. A real item may have complex conditions that can cause an item to perish. For example, let's consider an ice cream. Ice cream should be transported with a temperature below -25°C [76]. However, even though the temperature was kept in the acceptable range, the ice cream may still become damaged from other things such as temperature fluctuation, drop in humidity and shock. Furthermore, the perish condition might also depend on the type of the ice cream and the ingredients of the ice cream [77].

Because of the variation of exception that can happen, using a real item will make the experiment unpredictable. I cannot control what happened to the items. Therefore, I choose to create a fictional item. I model the fictional items based on the ice cream previously described. The difference is, the item will only have one perishable condition, which is when exposed to a temperature above -25°C for at least 1 hour. I choose one hour based on the melting and hardness model of ice cream, described in [77].

To satisfy the second condition, which is to simulate the sensor event, I need to have some basis for generating the data. [89] provide some useable information. The paper describes the rate of temperature increase in a cold room when a refrigeration unit failed, in an insulated room. Figure 46 below contains the rate of increase.

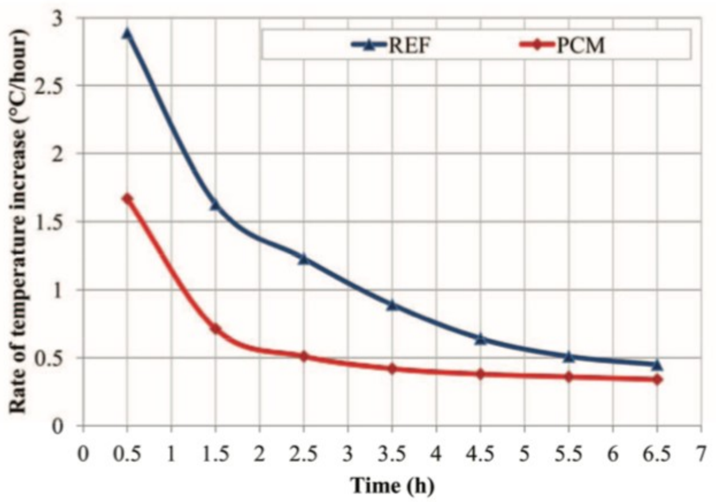

FIGURE 46 RATE OF TEMPERATURE INCREASE [89]

Figure 46, explains the differences in the rate of temperature between a box with traditional insulation material and a new material called PCM or phase change material. For the experiment, if we use the same type of material during the experiment iteration, the insulation material is not essential. I choose the reference material because it is what usually used in the real world.

For the reference material, we can see that in the first half hour, the temperature will increase by around 3°C. The rate of temperature increase is dropping quickly to 2°C after one hour. Because I set the items to become degraded when exposed to a temperature above -25°C for one hour, and based on the reference material, the temperature will increase by around 5°C in one hour, I create an assumption that the fictional items will defect when the temperature reach -20°C.

I create a sensor event generator that directly connects to situation aware logistics information system. The sensor simulator It will simulate the normal condition, followed by the temperature increase as described in Figure 46. Figure 47 below shows the event generator application.

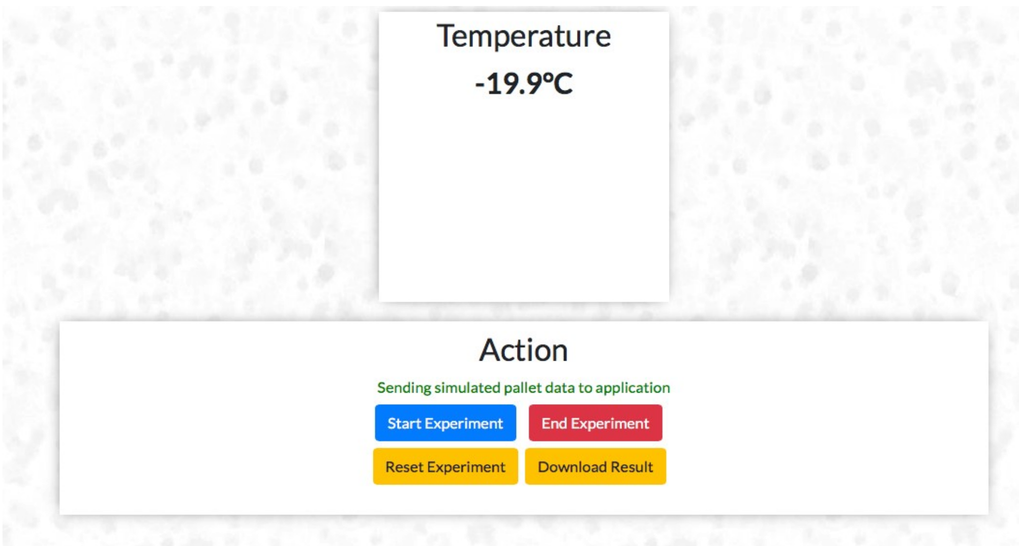

FIGURE 47 EVENT SIMULATOR

On the top of the event simulator application, we can see the transmitted simulated temperature into the situation-aware logistics information system. On the bottom of the page, we can see four buttons. The first button is the start experiments button. It will start simulating the temperature and send the result to the application. The end button will stop the simulation. The download result button is for downloading the experiment run time.

I also create another simulation tool to validate the customers that need to be notified and record the completion time. The tool contains a list of customers. The test subject must input the customer's email address to the system and give them the reason for the notification. The tool then will record the time after the test subject notifies all the customers. Figure 48 below shows the simulation tools.

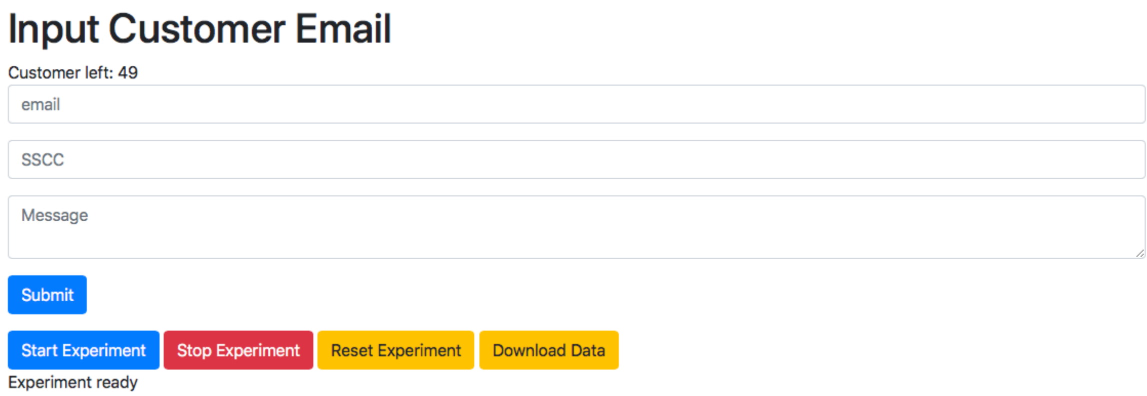

FIGURE 48 NOTIFY CUSTOMER SIMULATION TOOL

The customer simulation tools consist of a form and buttons. The form contains three fields, the email, the identifier for an orderline (SSCC) and the message. The test subject must fill all the fields and then click the submit button. On the bottom of the tool, there are four buttons. The start experiment button will record the start time of the experiment. The stop experiment button will record the finish time of the experiment. However, the application will disable the stop experiment button until the test subject notifies all the customers. When the download data button clicked, it will download a JSON file that holds the experiment time.

It is essential to know that in the previous business process; the transportation manager only knows which objects that are affected by an exception after the employee manually checked the items inside the vehicle. The condition means we can overlook the testing of the baseline business process and assume the customer will be notified at some time after the transportation completed. In this case, I assume the transportation process will take 3 hours to execute, and 1 hour to manually check the pallet. I assumed after I did a discussion with an expert from the client company.

In summary, to test the hypothesis, the test subject must perform a simulation of the exception handling process. First, the test subject must use the event simulator to simulate an exception. Then, the test subject must use the situation-aware logistics information system to identify the affected customer by the exception. Finally, the test subject must put the identified affected customers into the customer notification simulator. I also take note of the behavior of the test subject during the experiment. Table 11 below contains the test protocol for testing [H3].

Hypothesis	*By using the system, the exception can be detected faster, leads to faster customer awareness.*
Independent variables	The exception handling process
Dependent variables	The time to notify the customer
Subject	Transportation manager
The baseline procedure	-
The procedure	1. Open the situation-aware logistics IS. 2. Open palletsimulator.html 3. Open customernotification.html. 4. Click start experiment at palletsimulator.html. 5. Click start experiment at customernotification.html. 6. Transportation manager must determine which customer is affected by using the Situation aware logistics IS. 7. Transportation manager must put the affected customer into customer notification.html 8. When the test subject notifies all customers, click stop experiment on both the pallets simulator and the customer notification simulator.
Assumptions	1. The truck contains 11 pallets 2. There are 99 orderlines to be transported. 3. There are 49 customers in a shipment. 4. One pallet can contain at maximum nine orderlines. 5. The truck is full. 6. The normal temperature inside the container is -25°C. 7. The temperature will increase to -15°C in 1 hour when the refrigeration unit fails. 8. The item is considered damaged when the temperature reached -15°C.

TABLE 11 TEST PROTOCOL FOR [H3]

For validating the last hypothesis, *the knowledge of which pallets contains sensitive items during transportation exception can minimise the items loss because the pallet can be prioritized during mitigation operation.,* we need to decompose the hypothesis. The effect is item loss while the hypothesised cause is faster mitigation decision. Therefore, the independent variable is the time to mitigate the exception and the dependent variable is the item loss count. To know the item loss count, like the previous experiment, I need a model of an item condition. I reuse the same item model. However, in this experiment, I introduce the second item. The second item is necessary because the hypothesis talks about prioritisation. If the container only contains one type of item, then nothing can be prioritised, all the items will either be damaged or safe. If we have two types of items that defect at a different rate, then the items that will defect faster can be prioritised to be saved.

The second type of item still based on the previous items. However, instead of defecting when the temperature reached -15°C, the second item will defect when the temperature reaches -17°C. This condition means, the transportation manager must prioritise the second item type during the mitigation operation. I create the tool described below to simulate the truck and the item prioritisation.

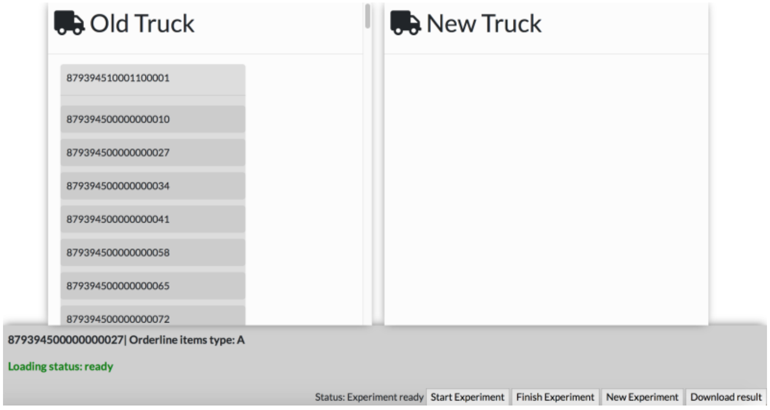

FIGURE 49 TOOL FOR SIMULATING MITIGATION OPERATION

The goal of the test subject in the tool is to move the orderlines from the old truck to the new truck. When the test subject clicks on the orderline, the tool will present them with information about the type of the items. The clicking action represents the process of manually inspecting an item. The test subject should inspect the orderlines and prioritise moving the orderliness that contains items with type B to obtain the baseline data. For the experiment; the transportation manager should use the situation-aware logistics information system to determine which orderlines should be prioritised to be moved into the new truck.

On the bottom of the application, we can see four buttons. The start experiment button will record the starting time and allow the test subject to move the orderlines from the old truck column to the new truck column. The finish experiment button can only be clicked when all the orderlines successfully moved into the new truck column. The finish experiment button will record the finish time of the experiment. The new experiment will reset the experiment, and the download result button will download the recorded time into the computer. Table 12 below describes the test protocol for **[H4]**.

Hypothesis	*The knowledge of which pallets contains sensitive items during transportation exception can minimise the items loss because the pallet can be prioritised during the mitigation operation.*
Independent variables	Current exception handling business process
Dependent variables	New exception handling business process
Subject	• Transportation manager • Loading/unloading operator
The baseline procedure	1. Open MitigationOperation.html. 2. Click start experiment. 3. Loading/Unloading operator must determine which pallet should be prioritised (Item B should be prioritised). The person can choose to put the pallet into the new truck immediately. 4. When all pallets have been moved, click stop experiment.
The procedure	1. Open the Situation Aware Logistics IS. 2. Open MitigationOperation.html. 3. Click start experiment 4. Logistics manager will use the situation-aware logistics IS to look which pallets should be prioritised (Item type B should be prioritised).

	5. Logistic Manager put the prioritised pallets through a communication channel such as email 6. Loading/unloading operator must put the Pallets from the old truck into the new truck, according to the logistic manager instruction 7. When all pallets have been moved, click stop experiment.
Assumptions	1. The truck contains 11 pallets 2. There are 99 orderlines to be transported. 3. There are 49 customers in a shipment. 4. One pallet can contain at maximum nine orderlines. 5. The truck is full. 6. The normal temperature inside the container is -25°C. 7. the temperature will increase to -15°C in 1 hour when the refrigeration unit fails. It will continue to increase at the rate of 1°C 8. Item type A is considered damaged when the temperature reached -15°C. 9. Item type B is considered damaged when the temperature reached -17°C

TABLE 12 TEST PROTOCOL FOR [H4]

7.3 EXPERIMENT RESULT

7.3.1 *[H1] BY USING THE SYSTEM, THE STAKEHOLDER'S DESIRE WILL BE FULFILLED*

To analyse how the system, fulfill the desire of the stakeholder, I am validating the hypothesis by submitting the artifacts with a panel of experts. I performed the validation on the 14th of August 2018 with three logistics information technology experts from CAPE Groep. I submit the prototype by giving them a demo of how the prototype will function according to the architecture specification. Then, they must give their opinion and give a score. I use a 1 to 5 scale. 1 indicates the desire is not satisfied while five means the system realised the desire.

Stakeholder	Desire	Mechanism	Opinion	Score
Transportation manager	Monitor Exception Occurrence	Gather data from IoT sensor Rule-based exception detection. Notification modules	**+** IoT sensor gives the user more information about the condition surrounding the items **+** The rule-based system is extensible. The rule for the new item can be added quite easily. **+** The rules can be adjusted on the fly. Useful to adjusting the sensitivity. **+** The notifications are sorted by importance, and provide necessary information to act	4

			- The manager must create the exception rules for each of the items. Imagine if there are thousands of items? - The rules based on the experience of the transportation manager. It is not scientific and may result in false positive. - There is no well-tested rule currently available to use. - The current prototype does not have a function to mark the notification as read / handled	
	Improve Traceability	Add scanning process before the items transported. Track the item location using Smart Pallet.	+ The scanning process happen at a specific location. We can record the location of the item when scanned. + The GPS enabled Smart Pallet to provide real-time & location of an item. Maybe Useful for deciding mitigation operation - On the road, the system tracked the items on the pallet level. An item can be removed from the pallet on the way, untraced.	4
Customer	Timely exception notifications	Provide Exception detection Provide information about the affected customers	+ The exception detection working as intended + The exception notification provide information about the affected customer. - The prototype does not support automatic customer notification.	4
	Monitor items conditions	Gather data from IoT sensor and present it to the user.	+ The exception notification provides sufficient information about the affected customer. - The system does not have a page for the user to use. Every information is disclosed through the transportation manager	4

The government	Adhere to security & privacy laws	Provide role-based security	**+** The role-based access right limit what the user can view. Only privileges role can access private information such as customer information. **-** The developed architecture only mentions sparsely about security. It does not mention things such as how to secure the communication between the IoT Sensor to the system, External application to the system or from the system to the user.	2
The employees	Undisturbed work	Scan the items using Laser barcode scanner Easy to use user interface	**+** The prototype user interface is simple and supports continuous scanning. **-** There might be another, undisruptive way, such as the RFID gateway to provide the association between the smart pallets and the orderlines.	4
Ahrma	Use Ahrma smart pallets	Provide the capabilities to interface with their devices through enterprise service bus	**+** The enterprise service bus provides the capabilities to adapt to various sensor implementation in the market.	5
Cape Groep	Realizing the DATAREL concept	Provide components as the basis for AI-based exception detection	**+** The rule-based exception detection is dynamic. The AI can focus on tuning the rules parameter. **-** The designed architecture does not provide information on what kind of information is available to the AI, what parameter the AI can change and how to communicate the AI with the system.	3
	Adaptable to a different company	Provide adaptable services through an enterprise service bus. The use of industry standards.	**+** The information system uses global standards, especially on the identifier of transported objects, such as the shipments, orderlines and the items. **+** The enterprise service bus provides the capabilities to adapt	5

			to different system provider that might be used by the different company.	
System designer	Simple to implement & updates	Provide clear separation between the components	**+** The system module is logical **+** The prototype is built with Mendix, which make the prototype easier to be developed further. Primarily because of its low code and continuous integration capability. **-** The prototype sometimes naming inconsistent, because it is not specified the designed architecture	4
System Implementor	Easy to learn	Use UX design principles	**+** The navigation of the user interface follows the design principles **-** Some of the user interfaces are too crowded with data, and complex	3
System Administrator	Powerful administrative function	Have the capability to create/read/update/delete all the data managed by the system. Have the capability to scale the system. Have the capability to back up the system.	**+** The system administrator has access and the necessary interface to manually manipulate the data inside the system. + Because the prototype built on top of a cloud platform, it supports on-demand scalability. **-** The architecture does not specify any backup method,	3
Average				3.75

TABLE 13 EXPERT OPINION ANALYSIS

In Table 13, we can see five columns. The first column and the second column contain the stakeholders and their desire, as described in section 3.1. The third column describes how the information system provides services that can satisfy the desire of the stakeholders. The fourth column contains the opinion of the experts. Finally, the last column holds the stakeholders desire satisfaction rating according to the experts.

From the Table 13, we can see that from the scale of 1 to 5, the situation-aware logistics information logistics score on average 3.75. The score means in general; the situation-aware logistics performs well in fulfilling the stakeholder desire. However, many improvements are still possible. The privacy & security gets the worst point. I believe that is because I focus on the main functionality for the direct user of the architecture, which are the transportation manager, the employee and the customer. Security is inherent to all information system. However, security also depends on the country and the company policy. Forcing the architecture to use a specific method for security make the system rigid and may make the architecture obsolete when the security method proven as insecure. That is why I only mandate the role-based access. Also, I derive the architecture from IoT-A. Therefore, the implementor of the system can also refer to the IoT-A standards for general security guidance.

Another desire that needs improvement is the administrative function. Even though the architecture provides a powerful tool to create/read/update/delete, the architecture does not include the mechanism to perform backup or rollback a transaction. The exclusion means, a dangerous operation, such as deleting a master data can be terrifying. One wrong move and it can render the whole system useless. In my defense, in every cloud provider I look, they provide a tool to back up the system up entirely. Moreover, the architecture should run on a cloud platform to provide scalability. However, it may be a good idea to expose these functionalities directly from the information system user interface.

The experts believe the user interface for the transportation manager to gathers information regarding an exception is too crowded with information. The eight golden rules specify that the design should reduce short-term memory load. The information system should split the information shown on the screen depends on the exception that is currently handled. Even though the experts perceive the mock-up interface, as described in section 0 as good, the prototype that I build experience lacks refinement, due to the time constraint. I believe that is what makes them overwhelmed with the user interface.

A good score is coming from the adaptability of the architecture. The architecture requires the implementation to use the industry data format standards. It adapts a various system that uses the same standard easier. Moreover, the architecture communicates with external applications, such as the sensors and the external application through an enterprise service bus. An external application that uses different data format or communication protocol can be adapted to the common format by the enterprise service bus.

The experts satisfied with the detection capabilities, as you can see from their opinion on monitor exception occurrence desire. However, the test subjects raised questions during the discussion, such as, is there any existing model of a normal condition available, as currently, the transportation manager is responsible for generating the item model based on the person experience. The problem is, the number of the transported items type may vary, and the experience-based model is unscientific and might result in many false alarms. However, I believe many research on the topic of perishable food condition is available. For example, I find many reference papers about the standard condition model of transporting an ice cream.

Furthermore, based on an interview for deciding the architecture requirements, I discover that the client company only deals with a limited number of item type, which can be managed by a person. As for the false alarm, the system provides the capability to change the exception rules on the fly. The transportation manager can dynamically tune the rules.

Because this thesis is a part of the DATAREL project, it is necessary to discuss the feasibility of using artificial intelligence into the architecture. I have the plan to have the artificial intelligence to tune the exception rules so that it can detect the exception more reliably and reduce false alarm. However, in this research, the focus is to build the situation-aware logistics system, and not using artificial intelligence to tune the detection parameter. Therefore, some questions are left open for future research. The questions are: What kind of information that the artificial intelligence need to have to tune the parameter? What is the preferred mechanism to tune the parameter? How to integrate the artificial intelligence into the situation-aware logistics application? I hypothesise the information needed is the exception rules, and the generated events as well as information on whether the notification is true or false. The parameter to be tuned is the threshold of the exception rules, while the integration can be provided by the same method to integrate external applications: through the enterprise service bus. However, the statement is only an opinion, and we must validate it in future research.

To conclude, the answer to the hypothesis: *By using the system, the stakeholder's desires will be fulfilled* is yes, the architecture fulfilled the desires of the stakeholders, especially the desires of the stakeholder that directly utilise the system. However, not all desires are entirely realised by architecture, as I prioritise on the daily user of the architecture. Some extension to the architecture may be required to fulfill all the desire. Examples include security, exception rules model tuning and backup capabilities.

7.3.2 [H2] By using the system, time to load the items should increase, but it will be negligible.

To test the hypothesis, I perform the test protocol described in Table 10. The test happens on the 14th of August and the 15th of August, with three test subjects. I repeat the experiment three times for each of the test subject and the type of experiment. The experiment yields 9 data points. The experiment lasts for at least 1 hour per person, per experiment. Figure 50 below shows the results of the experiment.

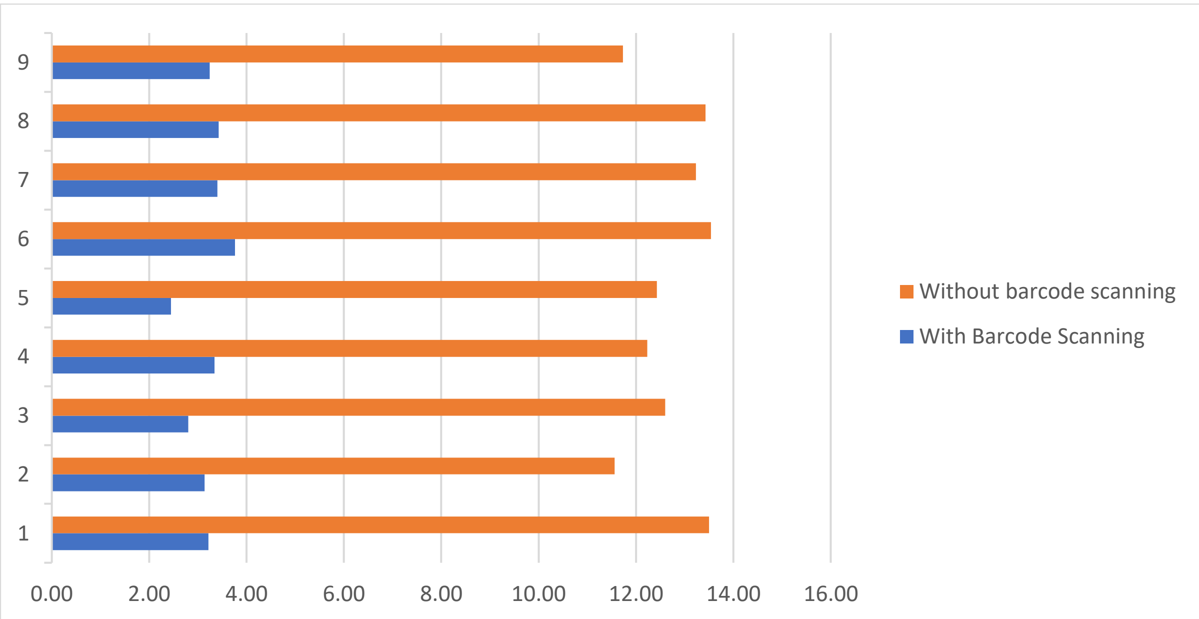

FIGURE 50 BARCODE SCANNING EXPERIMENT RESULTS

In figure 50, we can see two series of data, compared side to side. The X-axis corresponds to the minutes and seconds taken to finish the loading process. The Y axis is the iteration of the experiment. We can immediately see on the figure that the scanning process slow down the loading process. The simulated loading process without the addition of scanning process has the average completing time of 3 minutes and 20 seconds while the simulated loading process with the addition of scanning process has the average in 12 minutes 49 seconds. The results mean the new business process delays the loading process by approximately 8 minutes or an increase of 62.5%. I should give a warning that the simulated loading process does not represent the loading process in the real world, while the simulation does represent how the scanning process works. The mismatch might impact the result of the experiment.

I observe the user took a long time to scan the items because of the difficulties of using the 1-dimensional barcode scanner. The barcode scanner only works from a certain distance from the barcode. Because I print the barcode using a laser printer, the black level in the barcode line varies. The difference in the blackness may affect the scanning process. However, I did test the barcode booklet before experimenting, and it works flawlessly.

Another reason that might affect the performance of the loading process is the context switch. The test subject seems confused when challenged with a lengthy process involving a diverse variety of activities. In the experiment, the test subject must find the location of the barcode in the pallets and orientate the barcode scanner correctly. Then, the test subject must find the correct distance for the barcode scanner the register. After that, the test subject must check the feedback on the screen whether the association process ends up in success or failure. Afterward, the employee repeats the scanning process for all the orderlines and pallets. Finally, the employee must load the pallets into the truck.

Even though the scanning process is slow, the experts say that the loading process will take an even longer time than the simulated one. The loading process will typically take 1 hour, while in the simulation, it only takes around 3 minutes to load everything. Furthermore, improvement in the pallets and orderlines identification process, such as using RFID gateway scanner, or add more employee for the scanning process could help minimise the impact of the scanning process. Hence, the experts believe the scanning process will not have a significant impact on the loading process.

In summary, the experiment shows that the scanning process has a measurable impact on the scanning process. The cause of the impact is the time required for the scanner to recognise the barcode, and the context switch to perform the various task necessary to complete the scanning and loading process. However, the experts believe the impact is tiny compared to the benefit of the system and can be improved. The improvement includes employing new automatic identification techniques such as RFID or adds more employee to reduce the person in charge load.

7.3.3 [H3] By using the system, the exception can be detected faster, leads to faster customer awareness.

To test the hypothesis, I perform the test protocol described in Table 11. I execute the test on the 14th of August and the 15th of August, with three test subjects. After the test was over, I asked their opinion about their thoughts on the experiment. Figure 51 below shows the results of the experiment.

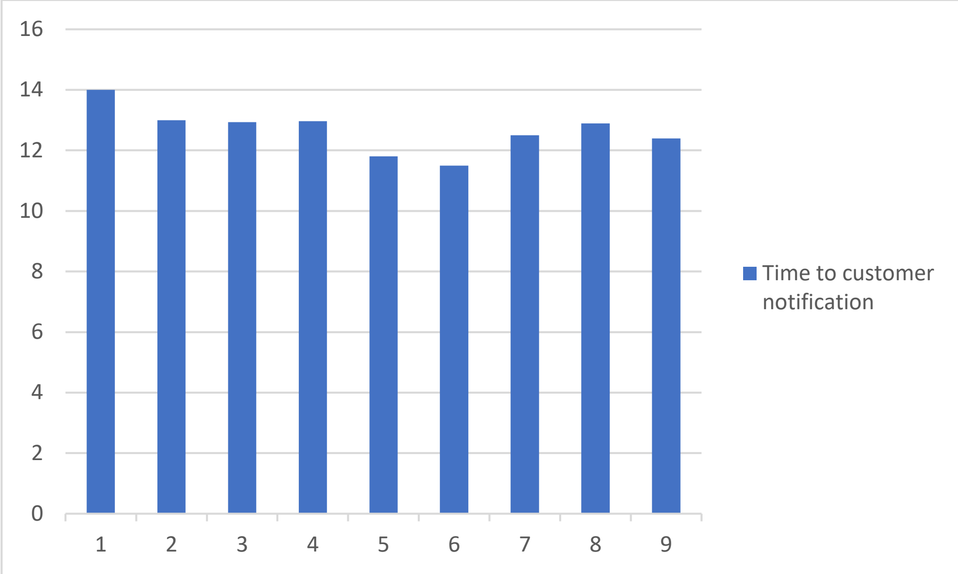

FIGURE 51 TIME TO CUSTOMER NOTIFICATION

In Figure 51 above, we can see a bar chart representing the time to notify the customer. The Y-axis indicates the time in minutes, while the Y-axis shows the iteration of the experiment. Based on the experiment results above, we can see that the test subject can process the exception notification and notify the affected customers in less than 14 minutes for 49 customers. The time includes the process necessary to compile the customer information. Therefore, I prove the process is proven to be faster than must wait for the defected items checked after the vehicle arrived at the destination. Based on the address of the case study and transit time information provided by google maps, the transportation between the production center and the distribution center takes around 2 hours to complete and might need one more hour to complete the manual check of the condition of the items. The experiment results prove that the new business process is significantly faster, almost 77% increase in performance.

Though, the discussion also leads to a fascinating insight. First, the designed situation aware logistics information system pushes the information surrounding the exception at once. The behaviour led to information overload for the test subject, making the person confused and lower the processing speed. The reason I present all the information is that different exception needs different kind information, and it is the transportation manager jobs to filter and use the essential information. The architecture should provide a way to store the type of information presented on the screen, based on the detected exception.

Another insight is about the process of notifying the customer. I observe that typing the customer email one by one from the situation-aware logistics information system to the email program took most of the time. The test subject even comments why the information system cannot send an email to the customer automatically. My reason is that some exception may be quick and easy to resolve or a false positive. Moreover, the company may have a policy on the severity level of the exception, before notifying the customer. That is why I require the transportation manager to be the final decision maker. Based on the feedback, I recommend adding a button to send email to the customer, so that the transportation manager still control the notification, while providing that person more straightforward way to notify the customer.

I also noticed that when the exception detected early, and the vehicle chooses to go back to be repaired, then the difference between using the situation-aware logistics information system and without the system is only 1 hour, plus 15 minutes for processing the customer information. The fact raises a discussion: does the improvement of customer awareness, which at maximum only improve the speed of notification by several hours, can make a difference for the customer? The experts believe it may have a positive benefit in critical fields, such as food transportation for the hospital. Furthermore, the company may reduce labor cost as manually check the content of the truck requires the on-field employee, which may better be assigned elsewhere.

The conclusion is, the situation-aware logistics information system, does make the customers receive notification about the status of their items faster, compared to the current business process. However, many improvements are still possible. The test subject may be overwhelmed by the information surrounding the exception because the current prototype does not filter the information based on the exception type. The overwhelming information slows the person capability to process the exception. Another bottleneck of the operation is the lack of a button to send mail to the customer. Both problems can be trivially solved.

7.3.4 [H4] THE KNOWLEDGE OF WHICH PALLETS CONTAINS SENSITIVE ITEMS DURING TRANSPORTATION EXCEPTION CAN MINIMISE THE ITEMS LOSS BECAUSE THE PALLET CAN BE PRIORITISED DURING THE MITIGATION OPERATION.

I run the validation process for hypothesis 4 on the 14th of August and the 15th of August. The experiment involves three test subjects. The test subjects perform the testing as described in Table 12. After the test was over, I asked their opinion about their thoughts on the experiment.

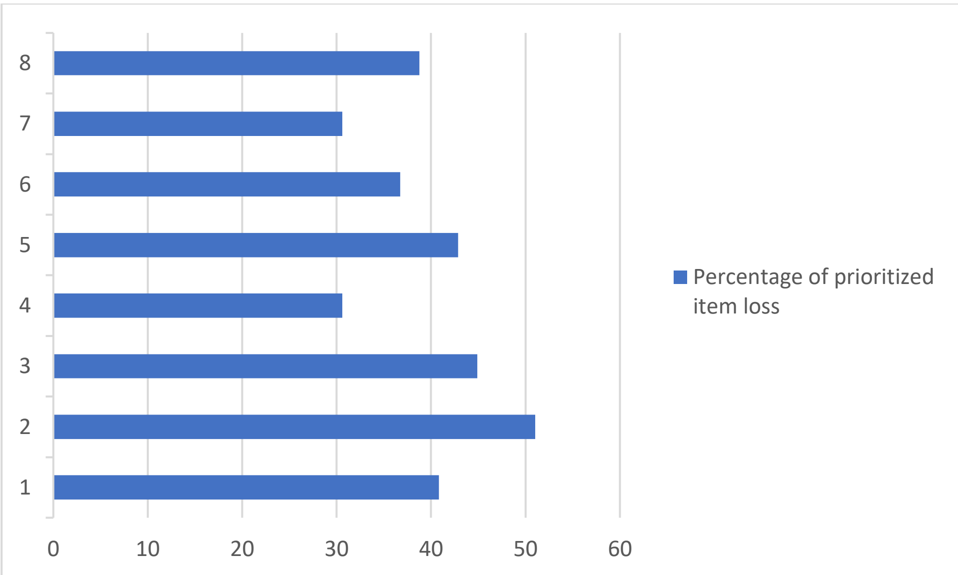

FIGURE 52 ITEM LOSS EXPERIMENT RESULT

The Figure 52 above displays the item loss for the prioritised items. The X axis shows the percentage of item loss while the Y-axis presents the iteration of the experiment. The experiments resulted in average, 39% of the prioritised items will be damaged, due to inability to handle the items in time. It is important to note that the 39% is the loss of the prioritised items (Item type B), which is the items that will perish slower than the rest of the transported item. Therefore, the system successfully saves 61% of the prioritised items or around 30% of all the items transported by the simulated truck.

Therefore, the experiment result shows that yes, the prioritisation of items during mitigation operation can reduce items loss. However, the condition for the usage of the information system to be able to save the items is quite restrictive. First, the situation-aware information systems provide the most savings when the transit time is longer than 1 hour. The reason because in the experiment, the insulation inside the vehicle keeps the temperature inside the truck cold enough for sufficient time to survive the transit.

The condition for the items is also specific. The truck must contain items that can withstand an exception for the time until the transportation manager decides how to react to the situation. Furthermore, the items must withstand another time until the technician or a new truck arrive at the scene. The reparation time or moving time, in case of switching the truck, increase the time that the items must withstand. Trying to perform mitigation operation when the items already perished will increase the cost to the exception. It may be better to continue the transportation process and deal with the exception after the truck arrives at the destination. The flowchart below describes the various condition where a mitigation operation might save some of the items.

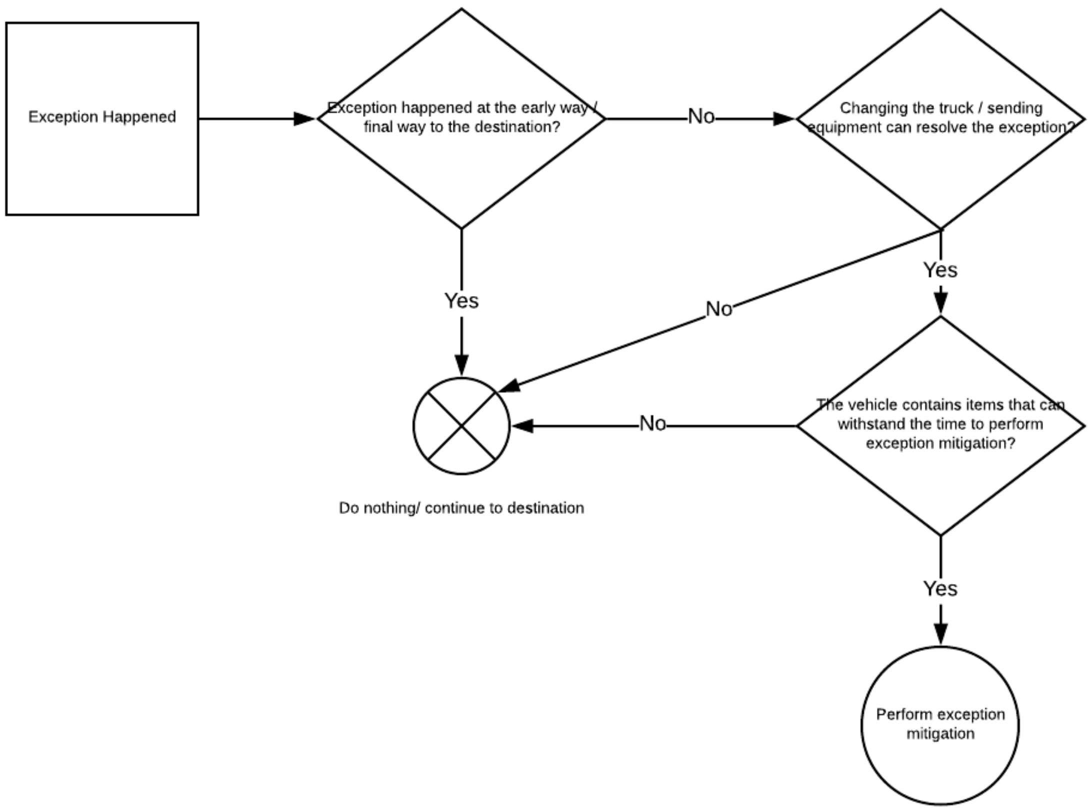

FIGURE 53 EXCEPTION MITIGATION FLOWCHART

In summary, if the condition is right, the system can potentially reduce the number of items loss during an exception. The condition right condition depends on the types of the transported items, the transit time, the cause of the exception and where the exception is happening. The restrictive condition suggests that the situation-aware logistics information system might not be the ultimate tool against item loss. In my opinion, the application is more suitable for prevention. Because the information system records every information that comes to the system, the system can analyse the past exception data, and the transportation manager can develop a prevention measure.

8 CONCLUSION

In this thesis, I successfully develop the architecture for the situation-aware logistics information system. The architecture consists of four parts, the business process, the application component model, the data model, and the physical model. To generate the architecture, I use the design science research methodology, described in section 1.3.1. The methodology starts with problem investigation, treatment design, and treatment validation. I use the methodology as guidance to build the research questions.

8.1 RESEARCH QUESTIONS

RQ1 *What are the requirements for smart logistic planning systems to manage the risk?*

To identify the requirements, first, I did a stakeholder analysis. The reason is, a requirement is the desire of stakeholders where the person willing to put resources. During the analysis, I found that there are nine stakeholders roles involved with the system. The stakeholders are the transportation manager, the customer, the government, the employees, Ahrma, Cape group, the programmers, the system implementor and the system administrator. I describe the stakeholders in more detail in section 3.1. Each of the stakeholders has their own desire. Therefore, I prioritise the stakeholders that interact directly with the systems.

Based on the prioritised stakeholder's desire, I build the architecture requirements. There are two types of requirements, functional requirements and non-functional requirements. The functional requirements denote the functionality of the design, while the non-functional requirements are how the functionality implemented in the architecture, must perform. I document the process of deducting the architecture requirements from the stakeholder desire in Figure 18. Table 3 provides the complete list of the requirements for the architecture. From that table, we can see that most of the requirements are about managing and presenting the variety of data that the system process. What I mean by managing is the capability to add, view, update and delete the data.

The requirements are essential to know what kind of components and interaction is needed to satisfy the stakeholder's desire. In my opinion, one of the essential requirements of the system is the capability to detect the exception. The system must provide a method to feed the data from the sensor into the detection algorithm, the design of the detection algorithm and a mean to present the results to the user to detect the exception.

RQ2 *What is the design of the situation-aware logistics system?*

To create the design, first, I look for existing architectures in the literature. I analyse their plus and minus compared to the architecture requirements. I describe the process in section Literature review results. I based my architecture on the IoT-A architecture. The reason is, IoT-A architecture provides most of the components, tools, and methods that the requirements specified. However, I also use components from another architecture. I use the data model for describing an event from IoT sensor from the EPCIS architecture. I did that because IoT-A does not provide the specification for it. Another architecture component that I use is the exception detection method. I use the event processing architecture defined in Figure 29. The processing architecture defines the components necessary to build a rule-based detection engine.

To provide communication with the external data sources, I use an enterprise service bus along with the web service paradigm. I use an enterprise service bus to adapt various data format from the IoT sensor manufacturer and company that are going to implement the architecture. Furthermore, the enterprise service bus also provides tools to secure the communication between the external systems.

I use ArchiMate to model the architecture. ArchiMate is a design language for modeling an enterprise architecture. I describe the architecture from multiple perspectives. Section 5.1 describes the architecture from the business process perspective. Section 5.2 describes the architecture from the component's functionality and interaction point of view while Section 0 describes the user interface for the application.

RQ3 *What kind of alterations are required to integrate the system with the business process?*

Section 2 describes how the old transportation process and mitigation operation process. While Section 5.1 describes the new business process that uses the designed system. For the transportation process, the main difference is the addition of scanning process, for associating the transported items with the IoT sensors. For the mitigation operation process, The I changed and reordered some of the action inside the business process. Instead of inspecting the content of the truck, by using the system, the transportation manager can immediately know which items are affected by the exception. The information can be directly used to notify the customers and formulate a mitigation operation. Therefore, the change to the business process, I switch the manual transported item check, into the end of the process, and prioritise the customer notification and preparing a mitigation operation. This way, the time for the customer to receive the notification should decrease, and there is a chance to reduce item loss.

RQ4 *What is the performance of the architecture in reducing transportation risks?*

I answer the research question by performing validations. Performance is a broad term. Therefore, the first thing I did this to define the terms performance further. I define the performance based on the goal of the thesis, the functionality of the system and the expected effect of the architecture. I formulated the following hypotheses: 1. *By using the system, the stakeholder's desire will be fulfilled. 2. By using the system, time to load the items should increase, but it will be negligible. 3. By using the system, the exception can be detected faster, which leads to better customer experience and faster in exception mitigation decision making. 4. By using the system, Items loss during the exception can be minimised, because of the faster mitigation decision.* I explained the hypothesis further in section 7.2.

To validate the hypothesis, I create some synthetic experiment. What I mean by synthetic is, I test the system by feeding some generated data into the system. The experiment only executes part of the business process, rather than test everything from start to finish. The reason is, to control the variable that is used for the experiment. Moreover, to save the time required to experiment. The time saving is crucial because the test subject only has limited time to do the testing. I develop three experiments to validate the hypothesis. I documented the experiments as three experiment protocol, described in Table 10, Table 11 and Table 12. I also develop a prototype implementation of the system, and a variety of tools to help to perform the experiments. I describe the tools in section 7.2.

For the first hypothesis, the answer is yes; the system can fulfill the stakeholder desire. However, some improvements are possible, primarily in the field of providing security, and user interfaces usability. For the second hypothesis, the result suggests that the addition of the scanning process have a measurable impact on the business process. Though, the experts say that the scanning process is still within the accepted level, especially when compared to the real loading process that could take an hour to finish. For the third experiment, the test subjects perform the customer notification faster than the previous business process. I found the bottleneck on the new business process is at the action of notifying the customer. It took a long time and error prone. It may be solved by providing a button to email the customer. The last experiment is about item loss. I discover that the faster generation of mitigation plan could result in a reduction of item loss due to the exception. However, the conditions that are required to make that happens is plenty. Therefore, I believe the system is more useful for exception prevention rather than exception mitigation. The statement comes from the fact that the application records the history of every event that leads to an exception, ready to be analysed.

8.2 LIMITATION

I test the architecture using self-generated data. It means the test result may not reflect how the architecture will behave in the real condition. However, I tried my best to make the generated data as valid as possible. All the master data practically based on anonymised real data. For example, the number of pallets and orderlines that need to be handled is based on the dimension of a real container.

Another example is from the event generator and the exception rules. I generate the event based on a paper that describes the temperature drop inside an insulated container. For the exception rules, I created a normal model based on ice cream, because I found papers that contains the definition of the normal condition.

The next limitation is the validation. In this thesis, I focus on the validation of the architecture performance. There are many aspects of the architecture that still needs validation. For example, validating the assumption that the architecture is secure, scalable and adaptable.

8.3 CONTRIBUTION

Prior research has shown us many architectures for leveraging the internet of things into the logistics fields. However, most of the architectures purpose is to identify, record the location of the transported objects, which is vital for traceability. [90] listed the architectures. Example of this type of architecture includes EPCIS [52], which is a GS1 standard. However, there are not many architectures already available for detecting logistics exception and provides useful information to react. Existing architecture for the purpose includes logistics exceptions management, described in [75] which focus on detecting the exception, and EURIDICE [43], which focus on gathering valuable information for logistics operation. Nevertheless, none of the mentioned architectures focuses on the integration of exception detection capabilities and presenting the information surrounding the exception into a real business process. This thesis attempts to fill the gaps by defining the roles and presenting the changes required to the business process for reacting to the exception. All to reduce the risk associated with logistics exception. In this thesis, I have developed an enterprise architecture to serve as guidance for implementing the concept of situation-aware logistics in real condition. Furthermore, I have designed the architecture so that it can serve as the basis for implementing artificial intelligence into logistics exception management for future research.

8.4 FUTURE WORKS RECOMMENDATION

Because this thesis is part of the DATAREL project, I suggest the next step is to research how to incorporate artificial intelligence into the architecture. This thesis only provide the architecture for logistics situation awareness. In my opinion, artificial intelligence could have roles in two things: to tune the exception rules, so that it produces more accurate notifications and to analyse the data after the exception happens. There are many questions for integrating artificial intelligence into the architecture. First, what kind of data is necessary for the artificial intelligence? What kind of AI algorithm is suitable for the role? How the AI presents the results?

Another future work that I suggest is to perform more validation to the architecture. As described in section 8.2, currently the architecture is only validated by synthetic data. The experiment I did slice the transportation process and the exception mitigation process into a workable chunk, to create a repeatable, controlled experiment. Nevertheless, thoroughly test the architecture in a real situation will provide new useful insight for improving the architecture. I formulate this as *what is the performance of the architecture in a real situation?*

Furthermore, some aspects of the architecture still require validation. The statement is true for both functional requirements and non-functional requirements. My recommendation for the next research is to build a new experiment protocol for each of the requirements and perform it.

My final recommendation for the next research is to improve the functionality of the architecture. For example, we can still improve the scanning process. We can incorporate Identification technology such as RFID into the architecture, which will eliminate the process of manually associating the pallets with the orderlines. The formulation for this problem is: What improvement is possible for the architecture? How the improvement increases the architecture performance? What is the performance of the improved architecture?

9 REFERENCES

[1] N. Ackerley, A. Sertkaya, and R. Lange, “Food Transportation Safety: Characterizing Risks and Controls by Use of Expert Opinion,” *Food Prot. Trends*, vol. 30, no. 4, pp. 212–222, 2010.

[2] M. Klein and C. Dellarocas, “A Knowledge-based Approach to Handling Exceptions in Workflow Systems,” *Comput. Support. Coop. Work*, 2000.

[3] “EU Regulation No 178/2002,” *Off. J. Eur. Communities*, vol. 1, no. 311, 2002.

[4] P. Suresh, J. V. Daniel, V. Parthasarathy, and R. H. Aswathy, “A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment,” in *2014 International Conference on Science Engineering and Management Research (ICSEMR)*, 2014.

[5] R. Minerva, A. Biru, and D. Rotondi, “Towards a definition of the Internet of Things (IoT),” *IEEE Internet Things*, no. 1, p. 86, 2015.

[6] L. D. Xu, W. He, and S. Li, “Internet of Things in Industries: A Survey,” *IEEE Trans. Ind. Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[7] “Five projects awarded within compartment 2 of ‘Big Data: real time ICT for logistics.’” [Online]. Available: https://www.nwo.nl/actueel/nieuws/2017/ew/vijf-toekenningen-voor-big-data-real-time-ict-for-logistics’---compartiment-2.html. [Accessed: 03-Jul-2018].

[8] R. J. Wieringa, *Design science methodology for information systems and software engineering*. Springer, 2014.

[9] K. Schwaber and J. Sutherland, “The Scrum Guide,” *Scrum.Org and ScrumInc*, 2013.

[10] I. F. Alexander, “A Taxonomy of Stakeholders,” *Int. J. Technol. Hum. Interact.*, 2005.

[11] J. Grabara, M. Kolcun, and S. Kot, “The role of information systems in transport logistics,” *Int. J. Educ. Res.*, 2014.

[12] T. P. Stank and T. J. Goldsby, “A framework for transportation decision making in an integrated supply chain,” *Supply Chain Management*. 2000.

[13] D. Grigori, F. Casati, U. Dayal, and M.-C. Shan, “Improving Business Process Quality through Exception Understanding, Prediction, and Prevention,” in *Proceedings of the 27th VLDB Conference*, 2001.

[14] ISO, “Iso 9241-11,” *Ergon. Requir. Off. Work with Vis. Disp. Termin. -- Part 11 Guid. usability*, 1998.

[15] A. Parasuraman, L. L. Berry, and V. A. Zeithaml, “Understanding Customer Expectations of Service,” *Sloan Manage. Rev.*, 1991.

[16] I. Meidutė-Kavaliauskienė, A. Aranskis, and M. Litvinenko, “Consumer Satisfaction with the Quality of Logistics Services,” *Procedia - Soc. Behav. Sci.*, 2014.

[17] S. M. Wagner and C. Bode, “AN EMPIRICAL EXAMINATION OF SUPPLY CHAIN PERFORMANCE ALONG SEVERAL DIMENSIONS OF RISK,” *J. Bus. Logist.*, 2008.

[18] “Reducing CO2 emissions from heavy-duty vehicles | Climate Action.” [Online]. Available: https://ec.europa.eu/clima/policies/transport/vehicles/heavy_en. [Accessed: 24-Jun-2018].

[19] C. Tankard, “What the GDPR means for businesses,” *Netw. Secur.*, 2016.

[20] B. Kitchenham, “Procedures for performing systematic reviews,” *Keele, UK, Keele Univ.*, 2004.

[21] B. Li, R. Yang, and Y. Hu, “An Experimental Study for Intelligent Logistics: A Middleware Approach,” *Chinese J. Electron.*, vol. 25, no. 3, pp. 561–569, 2016.

[22] C. W. Tsai, C. F. Lai, M. C. Chiang, and L. T. Yang, “Data mining for internet of things: A survey,” *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 77–97, 2014.

[23] S. Lu and X. Wang, “Toward an intelligent solution for perishable food cold chain management,” in *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, 2017, pp. 852–856.

[24] J.-Q. Li, F. R. Yu, G. Deng, C. Luo, Z. Ming, and Q. Yan, “Industrial Internet: A Survey on the Enabling Technologies, Applications, and Challenges,” *IEEE Commun. Surv. Tutorials*, vol. 19, no. 3, pp. 1504–1526, 2017.

[25] A. Moreno *et al.*, “Telematics system for the intelligent transport and distribution of medicines,” *IET Intell. Transp. Syst.*, vol. 7, no. 1, pp. 131–137, 2013.

[26] G.-X. Liu and F. Liu, “IoT-based TPL whole supply chain logistics information system model,” in *2013 International Conference on Machine Learning and Cybernetics*, 2013, vol. 04, pp. 1758–1762.

[27] B. W. Liu and H. Zhou, “Research on the construction of the logistics information platform in large-scale coal enterprises,” in *Proceedings - 2011 4th International Conference on Information Management, Innovation Management and Industrial Engineering, ICIII 2011*, 2011, vol. 2, pp. 377–381.

[28] M. P. Fanti, G. Iacobellis, A. M. Mangini, I. Precchiazzi, and W. Ukovich, “A flexible platform for intermodal transportation and integrated logistics,” in *2017 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, 2017, pp. 224–229.

[29] M. Marjani *et al.*, “Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges,” *IEEE Access*, vol. 5, pp. 5247–5261, 2017.

[30] C. Wunck and S. Baumann, “Towards a process reference model for the information value chain in IoT applications,” in *2017 IEEE European Technology and Engineering Management Summit (E-TEMS)*, 2017, pp. 1–6.

[31] A. J. Dweekat *et al.*, “Internet of things-enabled supply chain performance measurement model,” *MIT Press Essent. Knowl.*, vol. 54, no. 2015, p. xiv, 159 p, 2010.

[32] L. Feng, “Intelligent logistics and distribution system based on Internet of Things,” in *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 2016, pp. 228–231.

[33] E. Cavalcante *et al.*, “On the interplay of Internet of Things and Cloud Computing: A systematic mapping study,” *Comput. Commun.*, vol. 89–90, pp. 17–33, 2016.

[34] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[35] C. N. Verdouw, J. Wolfert, A. J. M. Beulens, and A. Rialland, “Virtualization of food supply chains with the internet of things,” *J. Food Eng.*, vol. 176, pp. 128–136, 2016.

[36] S. Venticinque and A. Amato, “Chapter 6 - Smart Sensor and Big Data Security and Resilience,” in *Security and Resilience in Intelligent Data-Centric Systems and Communication Networks*, 2018, pp. 123–141.

[37] C. N. Verdouw, A. J. M. Beulens, and J. G. A. J. van der Vorst, “Virtualisation of floricultural supply chains: A review from an internet of things perspective,” *Comput. Electron. Agric.*, vol. 99, pp. 160–175, 2013.

[38] A. Riahi Sfar, E. Natalizio, Y. Challal, and Z. Chtourou, “A Roadmap for Security Challenges in Internet of Things,” *Digit. Commun. Networks*, 2017.

[39] S. Li, “Chapter 1 - Introduction: Securing the Internet of Things BT - Securing the Internet of Things,” Boston: Syngress, 2017, pp. 1–25.

[40] C. N. Verdouw, A. J. M. Beulens, H. A. Reijers, and J. G. A. J. van der Vorst, “A control model for object virtualization in supply chain management,” *Comput. Ind.*, vol. 68, pp. 116–131, 2015.

[41] M. Bauer *et al.*, *Internet of Things – Architecture IoT-A Deliverable D1.5 – Final architectural reference model for the IoT v3.0*. 2013.

[42] S. Wang, “Internet of things based on EPC technology and its application in logistics,” in *2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce, AIMSEC 2011 - Proceedings*, 2011, pp. 3077–3080.

[43] M. Forcolin, E. Fracasso, F. Tumanischvili, and P. Lupieri, “EURIDICE - IoT applied to logistics using the Intelligent Cargo concept,” in *Concurrent Enterprising (ICE), 2011 17th International Conference on*, 2011, pp. 1–9.

[44] M. Chamekh, S. El Asmi, M. Hamdi, and T. H. Kim, “Context aware middleware for RFID based pharmaceutical supply chain,” in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2017, pp. 1915–1920.

[45] S. Amrioui, N. Malhene, and J. C. Deschamps, “Traceability in collaborative logistics: How to use EPCglobal solution in transport reconfiguration,” *IEEE Int. Conf. Digit. Ecosyst. Technol.*, 2012.

[46] S. Evdokimov, “RFID and the Internet of Things: Technology, Applications, and Security Challenges,” *Found. Trends® Technol. Inf. Oper. Manag.*, vol. 4, no. 2, pp. 105–185, 2010.

[47] Y. S. Kang, I. H. Park, J. Rhee, and Y. H. Lee, “MongoDB-Based Repository Design for IoT-Generated RFID/Sensor Big Data,” *IEEE Sens. J.*, vol. 16, no. 2, pp. 485–497, 2016.

[48] A. Bougdira, A. Ahaitouf, and I. Akharraz, “Towards an intelligent traceability system,” in *2016 International Conference on Information Technology for Organizations Development, IT4OD 2016*, 2016.

[49] R. Y. Chen, “Autonomous tracing system for backward design in food supply chain,” *Food Control*, vol. 51, pp. 70–84, 2015.

[50] A. K. Evizal, S. K. A. Rahim, T. A. Rahman, and S. L. Rosa, “Traceability Software for the Food Industry,” in *Advances in Food Traceability Techniques and Technologies: Improving Quality*

Throughout the Food Chain, 2016, pp. 191–206.

[51] A. Musa, A. Gunasekaran, and Y. Yusuf, "Supply chain product visibility: Methods, systems and impacts," *Expert Syst. Appl.*, vol. 41, no. 1, pp. 176–194, 2014.

[52] GS1, "EPC Information Services (EPCIS) Standard," pp. 1–138, 2016.

[53] S. Chen, R. Shi, Z. Ren, J. Yan, Y. Shi, and J. Zhang, "A Blockchain-Based Supply Chain Quality Management Framework," in *2017 IEEE 14th International Conference on e-Business Engineering (ICEBE)*, 2017, pp. 172–176.

[54] D. Vouyioukas *et al.*, "EPIKOUROS — Virtualized platforms using heterogeneous sensor services in cloud computing environment," in *2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE)*, 2014, pp. 1–5.

[55] M. El-hajj, M. Chamoun, A. Fadlallah, and A. Serhrouchni, "Taxonomy of authentication techniques in Internet of Things (IoT)," in *2017 IEEE 15th Student Conference on Research and Development (SCOReD)*, 2017, pp. 67–71.

[56] C. C. Byers, "Architectural Imperatives for Fog Computing: Use Cases, Requirements, and Architectural Techniques for Fog-Enabled IoT Networks," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 14–20, 2017.

[57] N. El Madhoun and F. A. Guenane, "An innovative cloud-based RFID traceability architecture and service," in *2014 International Conference on the Network of the Future, NOF 2014*, 2014.

[58] N. Primeau, R. Falcon, R. Abielmona, and E. Petriu, "Continuous Risk-Aware Response Generation for Maritime Supply Chain Disruption Mitigation," in *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2017, pp. 170–174.

[59] M. Blackstock, N. Kaviani, R. Lea, and A. Friday, "MAGIC Broker 2: An open and extensible platform for the internet of things," in *2010 Internet of Things, IoT 2010*, 2010.

[60] T. S. J. Darwish and K. A. Bakar, "Fog Based Intelligent Transportation Big Data Analytics in The Internet of Vehicles Environment: Motivations, Architecture, Challenges and Critical Issues," *IEEE Access*, 2018.

[61] S. Nechifor, A. Petrescu, D. Damian, D. Puiu, and B. Târnaucă, "Predictive analytics based on CEP for logistic of sensitive goods," in *2014 International Conference on Optimization of Electrical and Electronic Equipment, OPTIM 2014*, 2014, pp. 817–822.

[62] F. Mehdipour, B. Javadi, and A. Mahanti, "FOG-Engine: Towards Big Data Analytics in the Fog," in *Proceedings - 2016 IEEE 14th International Conference on Dependable, Autonomic and Secure Computing, DASC 2016, 2016 IEEE 14th International Conference on Pervasive Intelligence and Computing, PICom 2016, 2016 IEEE 2nd International Conference on Big Data*, 2016, pp. 640–646.

[63] S. Saddiq, N. A. Zafar, and F. Ullah, "Formal modeling of smart logistics monitoring," in *2017 1st International Conference on Electronics, Materials Engineering and Nano-Technology, IEMENTech 2017*, 2017.

[64] V. Akila, V. Govindasamy, and S. Sandosh, "Complex event processing over uncertain events: Techniques, challenges, and future directions," in *2016 International Conference on*

Computation of Power, Energy, Information and Communication, ICCPEIC 2016, 2016, pp. 204–221.

[65] C. Stergiou, K. E. Psannis, B. G. Kim, and B. Gupta, "Secure integration of IoT and Cloud Computing," *Futur. Gener. Comput. Syst.*, vol. 78, pp. 964–975, 2018.

[66] P. P. Ray, "A survey on Internet of Things architectures," *Journal of King Saud University - Computer and Information Sciences*, Elsevier, 08-Oct-2016.

[67] S. Vashi, J. Ram, J. Modi, S. Verma, and C. Prakash, "Internet of Things (IoT): A vision, architectural elements, and security issues," in *Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017*, 2017, pp. 492–496.

[68] M. Díaz, C. Martín, and B. Rubio, "State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing," *J. Netw. Comput. Appl.*, vol. 67, pp. 99–117, 2016.

[69] Z. Musa and K. Vidyasankar, "A Fog Computing Framework for Blackberry Supply Chain Management," in *Procedia Computer Science*, 2017, vol. 113, pp. 178–185.

[70] S. Sicari, A. Rizzardi, D. Miorandi, C. Cappiello, and A. Coen-Porisini, "A secure and quality-aware prototypical architecture for the Internet of Things," *Inf. Syst.*, vol. 58, pp. 43–55, 2016.

[71] M. Borkowski, W. Fdhila, M. Nardelli, S. Rinderle-Ma, and S. Schulte, "Event-based failure prediction in distributed business processes," *Information Systems*, 2018.

[72] Y. Zhang, S. Liu, Y. Liu, and R. Li, "Smart box-enabled product–service system for cloud logistics," *Int. J. Prod. Res.*, vol. 54, no. 22, pp. 6693–6706, 2016.

[73] A. Jede and F. Teuteberg, "Towards cloud-based supply chain processes: Designing a reference model and elements of a research agenda," *Int. J. Logist. Manag.*, vol. 27, no. 2, pp. 438–462, 2016.

[74] R. S. Gaonkar and N. Viswanadham, "Analytical framework for the management of risk in supply chains," *IEEE Trans. Autom. Sci. Eng.*, 2007.

[75] S. Gao, D. Xu, and P. Green, "Logistics Exception Management: A Theory-Driven Real-Time Decision Model," *Proceedings of The First AIS-Journals Joint Author Workshop in PACIS – Table #4: Decision Systems – Section 4-4*. pp. 1–34, 2013.

[76] S. A. Tassou, G. De_Lille, and Y. T. Ge, "Food transport refrigeration-Approaches to reduce energy consumption and environmental impacts of road transport," *Appl. Therm. Eng.*, no. 853, pp. 1467–1477, 2009.

[77] M. R. Muse and R. W. Hartel, "Ice Cream Structural Elements that Affect Melting Rate and Hardness," *J. Dairy Sci.*, 2004.

[78] A. Garcia-de-Prado, G. Ortiz, and J. Boubeta-Puig, "COLLECT: COLLaborativE ConText-aware service oriented architecture for intelligent decision-making in the Internet of Things," *Expert Syst. Appl.*, vol. 85, pp. 231–248, 2017.

[79] J. Kress *et al.*, "Enterprise service bus," *Ind. SOA*, pp. 1–10, 2013.

[80] F. Menge, "Enterprise service bus," *Free Open Source Softw. Conf.*, 2007.

[81] M. Van Welie, "Task-based user interface design," *Evaluation*, p. 217, 2001.

[82] K. Holtzblatt and H. Beyer, "Contextual design," in *CHI '99 extended abstracts on Human factors in computing systems - CHI '99*, 1999.

[83] R. A. Hill and R. A. Barton, "Psychology: Red enhances human performance in contests," *Nature*, 2005.

[84] M. A. Maier, P. Barchfeld, A. J. Elliot, and R. Pekrun, "Context Specificity of Implicit Preferences: The Case of Human Preference for Red," *Emotion*, 2009.

[85] B. Shneiderman, "The eight golden rules of interface design," *Department of Computer Science, University of Maryland*, 2016. .

[86] L. Huang and U. Piontek, "Improving Performance of Cold-Chain Insulated Container with Phase Change Material: An Experimental Investigation," *Appl. Sci.*, 2017.

[87] Federation of Enterprise Architecture Professional Organizations, "A Common Perspective on Enterprise Architecture," *Archit. Gov. Mag.*, 2013.

[88] T. S. Kuhn, *The Structure of Scientific Revolutions*. 1970.

[89] B. Copertaro, R. Fioretti, and P. Principi, "Thermal analysis on a phase change material latent heat storage in a cold room in case of power outage," in *EEEIC 2016 - International Conference on Environment and Electrical Engineering*, 2016.

[90] P. P. Ray, "A survey on Internet of Things architectures," *Journal of King Saud University - Computer and Information Sciences*, 2016.