A redesigned system onboarding strategy for Smart Freight Matching

UNIVERSITY OF TWENTE.

CAPEGROEP
grensverleggend vooruit

eMagiz

Bachelor Thesis

Thijs Hammink

Publication date: 12-03-2019

Student
T. G. Hammink (Thijs)
Industrial Engineering and Management
University of Twente

First supervisor university	Second supervisor university	Supervisor eMagiz
Prof. Dr. M. E. Iacob (Maria)	Dr. L. O. Meertens (Lucas)	S. Kaya (Samet)
Adjunct Professor	Assistant Professor	Product Manager
University of Twente	University of Twente	eMagiz

Management summary

The research problem that is handled in this report is how a reduction of onboarding costs and onboarding throughput time could be realized. Therefore, a redesigned onboarding process is proposed and validated with a digital prototype. The research focuses on the reduction of contact moments between eMagiz services and the involved parties for the collection of necessary information during the onboarding process.

Research motivation

The company described as CargoMatcher in this research is a partner of eMagiz Services. The company is an online matching platform for Smart Freight Matching (SFM) between carriers and shippers. For shippers, there are two different options of how the order entry is done. The first is manual via their online platform, where orders are put in by hand. The second is via a system integration, where orders are put in the CargoMatcher platform directly from the shipper's own system. For carriers, there are two different options for order handling. Manual entry in CargoMatcher's platform or via a system integration. On the longer term, the system integrations are the best option for both parties. eMagiz services conduct these system integrations with the use of the eMagiz integration platform.

CargoMatcher requests a lot of small and repetitive integrations. This results in multiple running projects. These are projects with a relatively low level of technical difficulty, but with a high expected onboarding throughput time. This long onboarding throughput time is mostly caused by the waiting time between the multiple contact moments. From an eMagiz services perspective, it is hard to reduce the waiting time between the contact moments, because they are dependent on the readiness of the parties involved. Nevertheless, it is feasible to reduce the number of contact moments, because the contact moments are mostly requested by eMagiz services. This results in the following research question:

"How to reduce the expected throughput time for the onboarding process, by altering the onboarding strategy with the use of standard market connectors in a tenant onboarding tool?"

Methodology

In this research, a prototype is used for the validation of the proposed solution. Therefore, the methodology that is selected in this research is DSRM, because the development of a digital prototype is integrated into this methodology. BPMN is used, for the identification of the current situation and the mapping of the redesigned situation. Based on these BPMN models a PERT analysis is conducted for both the current and redesigned situation. PERT is a tool that is specially selected based on a literature study that is conducted in this research. Multiple filters are applied to the developed prototype. These filters are based on the theory of linguistic decision tree structures. This is a tool that is selected by a systematic literature review that is conducted during this research.

Evaluation

In this research, a norm is set for the chance that the onboarding process takes less or equal to 3 months. The norm that was set for this chance is 85 per cent. The PERT analysis is selected in this research as a measurement instrument to estimate this chance for the current and the redesigned situation. For the current situation, this chance is estimated to be 15,6 per cent. This chance is 69,4 per cent lower than the norm of 85 per cent. The estimated chance for the redesigned situation is 87,9 per cent. This change value is 2,9 per cent higher than the norm, and therefore the redesigned onboarding process is a suitable solution to solve the research problem.

Conclusion & recommendations

The expected onboarding throughput time could be reduced by using standard solutions (redesigned situation) instead of custom solutions (current situation). A standard solution is a reusable premade connector that is based on standards that are available in a market segment. In the redesigned situation the intake phase is altered and the development phase is replaced by an onboarding phase. Some issues which are not researched in this report were encountered during the validation of the redesigned situation. For these issues it is recommended that the following will be determined in further research:

- Determine if a tenant suits the tenant onboarding solution
- Determine when a standard solution is better than a custom solution
- Determine a revenue model
- Determine who is going to enter the information in the onboarding wizard

Table of contents

Management summary II

List of abbreviations IX

1. Introduction 1

1.1 Research stakeholders 1

1.1.1 CAPE Groep 1

1.1.2 eMagiz services 1

1.1.3 CargoMatcher 1

1.2 Problem identification 2

1.2.1 The Smart Freight Matching project 2

1.2.2 Problem (cluster) 3

1.2.3 Core problem 4

1.2.4 Research focus 5

1.2.5 Variables and indicators 5

1.2.6 Norm and reality 5

1.3 Research questions and Methodology 6

1.4 Report structure 7

2. Theoretical framework 8

2.1 Decision trees (systematic literature review) 8

2.1.1 Necessity of literature review 8

2.1.2 Systematic literature review 9

2.2 Quantitative analysis of business processes 11

2.2.1 Quantitative analysis techniques 11

2.2.2 Analytical approach 12

2.2.3 Critical Path Method (CPM) 13

2.2.4 Program Evaluation Review Technique (PERT) 13

2.2.5 Selection of analytical approach method 14

3. Current situation 15

3.1 Current onboarding process 15

3.2 PERT analysis of the current situation 16

3.3 Current architecture 18

4. Desired situation 20

4.1 Redesigned onboarding process for tenants 20

4.2 PERT redesigned the onboarding process 21
4.3 Desired architecture 23
5. Prototype 24
5.1 Mendix prototype 24
5.2 Prototype architecture 24
6. Validation 25
7. Evaluation 25
8. Conclusions 26
8.1 Research question and knowledge questions 26
8.2 Limitations 28
8.3 Recommendations and future work 28
8.3.1 Determine if a tenant suits the tenant onboarding solution 28
8.3.2 Determine when a standard solution is better as a custom solution 29
8.3.3 Determine a revenue model 29
8.3.4 Determine who is going to enter the information in the onboarding wizard 29
9.Bibliography 30
Appendices i
Appendix 1: BPMN CargoMatcher platform i
Appendix 2: Systematic literature review ii
Appendix 2A: Search strings ii
Appendix 2B: Concept matrix Linguistic decision tree ii
Appendix 2C: Concept matrix Label semantics ii
Appendix 2D: Concept matrix Hierarchies of linguistic decision trees iv
Appendix 2E: Concept matrix algorithm for generating linguistic decision trees v
Appendix 2F: Systematic Literature Review Protocol v
Appendix 3: BPMN models of current process ix
Appendix 4: PERT current situation xii
Appendix 4A: Extracting the PERT process from the BPMN models xii
Appendix 4B: Time estimates per action of the PERT process xv
Appendix 4C: PERT calculations current situation xxii
Appendix 5: User stories and development tasks xxiii
Appendix 6: Screenshots of prototype xxv
Appendix 6A: Home screen prototype xxv

Appendix 6B: First screen tenant onboarding intake wizard xxv
Appendix 6C: Organisation screen 1 tenant onboarding intake wizard.. xxvi
Appendix 6D: Organisation screen 2 tenant onboarding intake wizard xxvi
Appendix 6E: Process screen 1 tenant onboarding intake wizard xxvii
Appendix 6F: Process screen 2 tenant onboarding intake wizard xxvii
Appendix 6G: System screen 1 tenant onboarding intake wizard xxviii
Appendix 6H: System screen 2 tenant onboarding intake wizard xxviii
Appendix 6I: Confirm screen tenant onboarding intake wizard xxix
Appendix 6J: Tenant onboarding intake wizard completion notification xxix
Appendix 6K: Status dashboard after completing tenant onboarding intake wizard........................ xxx
Appendix 6L: Completed onboarding process xxx
Appendix 6M: Package download and installation screen... xxxi
Appendix 7: BPMN models desired situation.. xxxii
Appendix 8: Implementation of linguistic decision tree in the prototype xxxviii

Table of tables

Table 1 structure of the report 8
Table 2 articles for literature review .... **Fout! Bladwijzer niet gedefinieerd.**
Table 3 characteristics of analytical approach and quantitative simulation 12
Table 4 PERT estimations of individual actions 16
Table 5 estimations of individual action (redesigned) 22
Table 6 Search strings systematic literature review ii
Table 7 Concept matrix linguistic decision tree ii
Table 8 Concept matrix label semantics ii
Table 9 Concept matrix hierarchies of linguistic decision trees iv
Table 10 concept matrix algorithm for generating linguistic decision trees v

Table of figures

Figure 1 High-level matchmaking process CargoMatcher 2
Figure 2 Problem cluster 4
Figure 3 Six phases of DSRM 6
Figure 4 Example of LAH 10
Figure 5 Illustration difference between paths, actions and processes 12
Figure 6 Phases of onboarding process 15
Figure 7 AON project network of the current situation 16
Figure 8 The critical paths 17
Figure 9 Architecture current situation one customer 18
Figure 10 Architecture multiple customers with example message type GS1: POD 19
Figure 11 Desired tenant onboarding process 21
Figure 12 Redesigned AON project network 21
Figure 13 Critical paths redesigned process 22
Figure 14 Desired architecture with standard connectors 23
Figure 15 Onboarding wizard in architecture 24
Figure 16 BPMN CargoMatcher platform i
Figure 17 Search strings systematic literature review v
Figure 18 Initial papers systematic literature review vi
Figure 19 Papers for reading whole content vii
Figure 20 Last reading and concept matrices viii
Figure 21 BPMN Current onboarding process – presales and intake ix
Figure 22 BPMN Current onboarding process - sales and development x
Figure 23 BPMN Current onboarding process - acceptance and going live xi
Figure 24 PERT steps current situation - 1 xii
Figure 25 PERT steps current situation - 2 xiii
Figure 26 PERT steps current situation - 3 xiv
Figure 27 PERT Excel calculations current situation xxii
Figure 28 Prototype: home screen prototype xxv
Figure 29 Prototype: first screen tenant onboarding intake wizard xxv
Figure 30 Prototype: organisation screen 1 onboarding intake wizard xxvi

Figure 31 Prototype: organisation screen 2 tenant onboarding intake wizard xxvi
Figure 32 Prototype: process screen 1 tenant onboarding intake wizard xxvii
Figure 33 Prototype: process screen 2 tenant onboarding intake wizard xxvii
Figure 34 Prototype: system screen 1 tenant onboarding intake wizard xxviii
Figure 35 Prototype: system screen 2 tenant onboarding intake wizard xxviii
Figure 36 Prototype: confirm screen tenant onboarding intake wizard xxix
Figure 37 Prototype: tenant onboarding intake wizard completion notification xxix
Figure 38 Prototype: status dashboard after completing tenant onboarding intake wizard xxx
Figure 39 Prototype: completed onboarding process.. xxx
Figure 40 Prototype: package download and installation screen xxxi
Figure 41 BPMN desired onboarding process - presales and intake – implementation phase 1 xxxii
Figure 42 BPMN desired onboarding process - presales and intake – implementation phase 2 xxxiii
Figure 43 BPMN desired onboarding process - presales and intake – implementation phase 3 xxxiv
Figure 44 BPMN desired onboarding process - sales and configuration – implementation phase 1 xxxv
Figure 45 BPMN desired onboarding process - sales and configuration – implementation phase 2 xxxvi
Figure 46 BPMN desired onboarding process - acceptance and going live – implementation phase xxxvii
Figure 47 Implementation of a linguistic decision tree in the prototype xxxviii

List of abbreviations

AON	Activity on node
APS	Advanced planning system
BPMN	Business process model notation
COTS	Commercial of the shelf
CPM	Critical path method
DSRM	Design science research methodology
EF	Early finish
ERP	Enterprise resource planning
ES	Early start
iPaaS	Integration platform as a service
LAH	Linguistic attribute hierarchy
LDT	Linguistic decision tree
LF	Late finish
LT	Late start
MVP	Minimal viable product
PERT	Program evaluation review technique
POD	Proof of delivery
SMEs	Small and middle enterprises
T	Expected throughput time
TMS	Transport management system
var	Variance
WMS	Warehouse management system

This research is conducted for eMagiz services and one of their partners. To keep this partner anonymous a pseudonym is used to refer to this partner. The pseudonym that is given to the partner is CargoMatcher.

Definition of system integration

System integration: Mische (2001) defines four different states of system integration. Every state has its own definition of system integration. The most elementary state is state 1: interconnectivity. This definition is used as the definition of system integration in this research.

Interconnectivity involves making various pieces of often disparate equipment and technologies work together. This includes the sharing of peripherals, the simple transferring of files, and the creation of common pathways between different components. The basic applications, functionality, and uses all remain specific with respect to their technologies and users, with little or no integration at the functional levels.

1. Introduction

1.1 Research stakeholders

1.1.1 CAPE Groep

CAPE Groep is a software and integration consulting company located at the transport centre in Enschede (CAPE GROEP, 2018). Their target business is to offer 100% fit solutions for software development, connectivity, integrations, business intelligence, reports and cloud computing (CAPE Groep, n.d.). They offer long-term strategic partnerships to their customers. In this way, CAPE Groep wants to create synergy between people, methodology and technology to keep their customers agile and innovative.

1.1.2 eMagiz services

eMagiz is a company that delivers model-driven enterprise integration products and services (eMagiz, n.d.). One of their services is an Integration Platform as a Service (iPaaS) (Gartner, Inc., 2018), which they provide by an online portal (eMagiz, n.d.). The core value of the platform is that data integration can be modelled instead of that it needs to be hardcoded. This makes it easier for business consultants to realize application integration for their clients.

1.1.3 CargoMatcher

About 25 per cent of the trucks in Europe drive empty (Numann, 2017). CargoMatcher is a start-up, which focuses on these empty truck rides. CargoMatcher provides an online platform where they match empty kilometres of carriers with shippers, to fill up the empty rides. They mainly focus on small and middle enterprises (SMEs) which transport pallets of packed goods. Currently, they provide a platform which supports the Benelux.

1.2 Problem identification

1.2.1 The Smart Freight Matching project

eMagiz is a company that provides model-driven enterprise integration products and services to its customers (eMagiz, n.d.). CargoMatcher is one of those customers. CargoMatcher is a start-up that wants to minimize the empty kilometres of trucks on the road. Therefore, they want to realize an automatic real-time Cross Chain Control Centre (4C) (Topsector logistiek, 2018) for Smart Freight Matching (Himanen, Lee-Gosselin, & Perrels, 2007).

They are trying to realize this with the use of an online platform where they match orders of shippers with empty truck capacity of carriers. Their matching process consists of six phases. Which is briefly explained in this section to give a better understanding of the necessity for a solution.

The matching process starts with an order request of a shipper (global negotiator, n.d.), by entering the requirements and details of the shipment. In the second phase, CargoMatcher looks if there is a potential carrier (global negotiator, n.d.), which could fulfil this shipment. If CargoMatcher finds a potential carrier, which accepts the shipment, a match is made. In the third phase, CargoMatcher sends information about the shipper and the shipment to the carrier and vice versa. In the fourth phase, the carrier decides when it is going to pick up and deliver the shipment, taking the requirements of the shipper into account. During the fifth phase, the agreed shipment is conducted. During this shipment, the shipper could keep track of the process, due to track status updates. In phase 6 the shipment is finished, here the shipper must pay CargoMatcher for the shipment and CargoMatcher must pay the carrier for the service. The process is now completed. A high-level overview of this process is given in figure 1. In Appendix 1 a more detailed Business Process Model Notation model (BPMN) of this process is given (Object Management Group, n.d.). Here the colours represent the steps of the CargoMatcher platform process as given in figure 1.

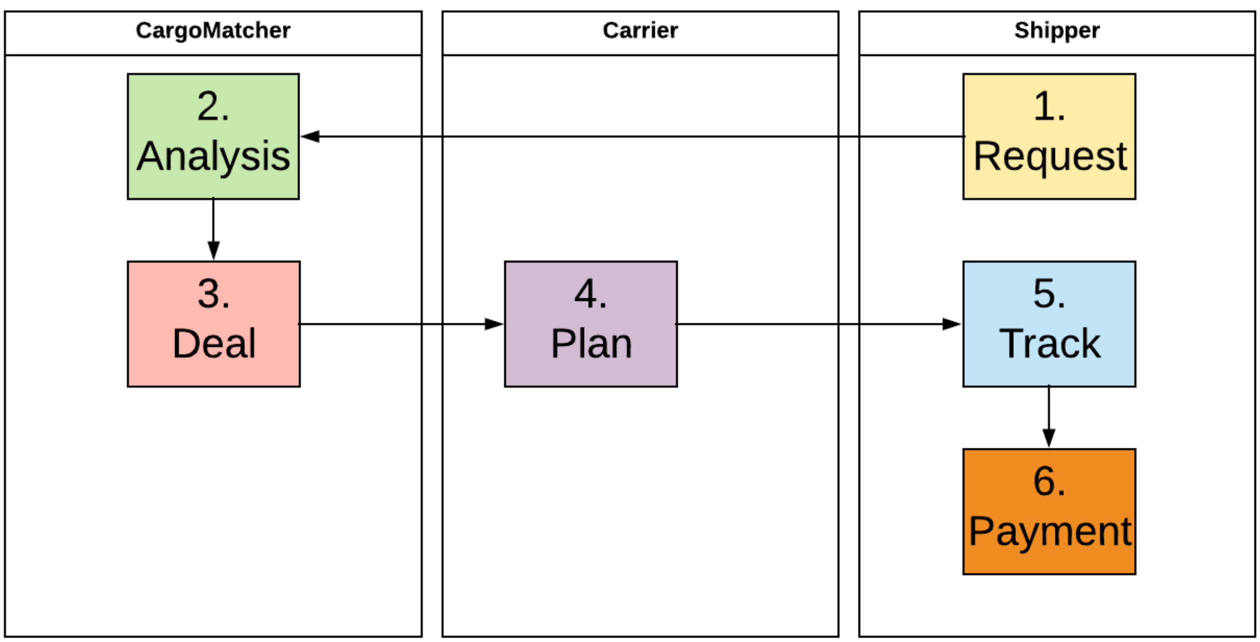

Figure 1 High-level matchmaking process CargoMatcher

CargoMatcher wants to automatize this matching process by making use of electronic messaging (Rouse, 2005) between customer systems, so orders could be requested and handled via the customers own system(s). This automation could be accomplished by a system integration to a message bus (Hohpe & Woolf, 2003). Integrating a new system to a message bus is called system onboarding. Which is further referred to in this research as onboarding. This onboarding process will be further explained in chapter 3.

Currently, already 300 companies make use of the CargoMatcher platform (Numann, 2017). Most of these companies request and handle orders manually via the CargoMatcher platform. CargoMatcher wants to onboard a lot of these companies, mainly the companies which (potentially) handle a lot of orders via the CargoMatcher platform. Therefore, CargoMatcher started a partnership with eMagiz services, an Integration Platform as a Service (iPaaS) provider, to conduct the onboarding of these companies (Gartner, Inc., 2018). The onboarding is a critical process for CargoMatcher because system integration is needed for automatic matching. In section 1.2.2 problems that CargoMatcher faces are described.

1.2.2 Problem (cluster)

In this chapter, an overview of the problems CargoMatcher faces is given. This is done to give an insight in which problem is the starting problem and how other related problems influence the core problem. This is represented in a problem cluster in figure 2 ((Heerkens & van Winden, 2012), where the problem on the right is the starting problem and the problem on the left is the core problem. The core problem, which will be explained in chapter 1.2.3, is chosen following the four rules of thumb of Heerkens and van den Winden (2012). Following Heerkens and van Winden (2012), the main problem is a problem that can be influenced and has no other cause (which can be influenced).

CargoMatcher introduced the problem with the fact that their matching costs are too high, due to too much manual matching. The manual matching is time-consuming which results in a high cost of employees. This manual matching is necessary, because the matching algorithm, which matches carriers and shippers automatically, could not be used.

In the situation of a shipper, which wants to place an order, the order details need to be put in manually to the CargoMatcher platform. If a shipper has a lot of orders, this manual order input could be very time consuming, while these orders are already in their own system(s). With a system integration, it will be possible to send these orders directly from their own systems.

Next to that, the matching algorithm could not be used due to a lack of input data. When the two meetings were conducted, the only input was a static working area. This working area is based on ZIP codes and is provided during the intake of a carrier. This information was the only information that was used for matchmaking by CargoMatcher at that moment. This is not the information the matching algorithm needs. The matching algorithm needs dynamic planning data from carriers. Carriers have this information available in their own systems. This information could also be made available with system integration.

System integration with both the shipper´s and the carrier´s system could eliminate the manual handlings and makes dynamic matching possible. This gives a big potential to reduce the matching time and to offer more accurate matches. Therefore, CargoMatcher and eMagiz already have conducted some system integrations. The experience that CargoMatcher has with these integrations is that the costs for those integrations are high and the throughput time is very long. The current throughput time varies from a week to multiple months. The most dominant cause of the big variation in the throughput time is the waiting time between the individual actions of the onboarding process.

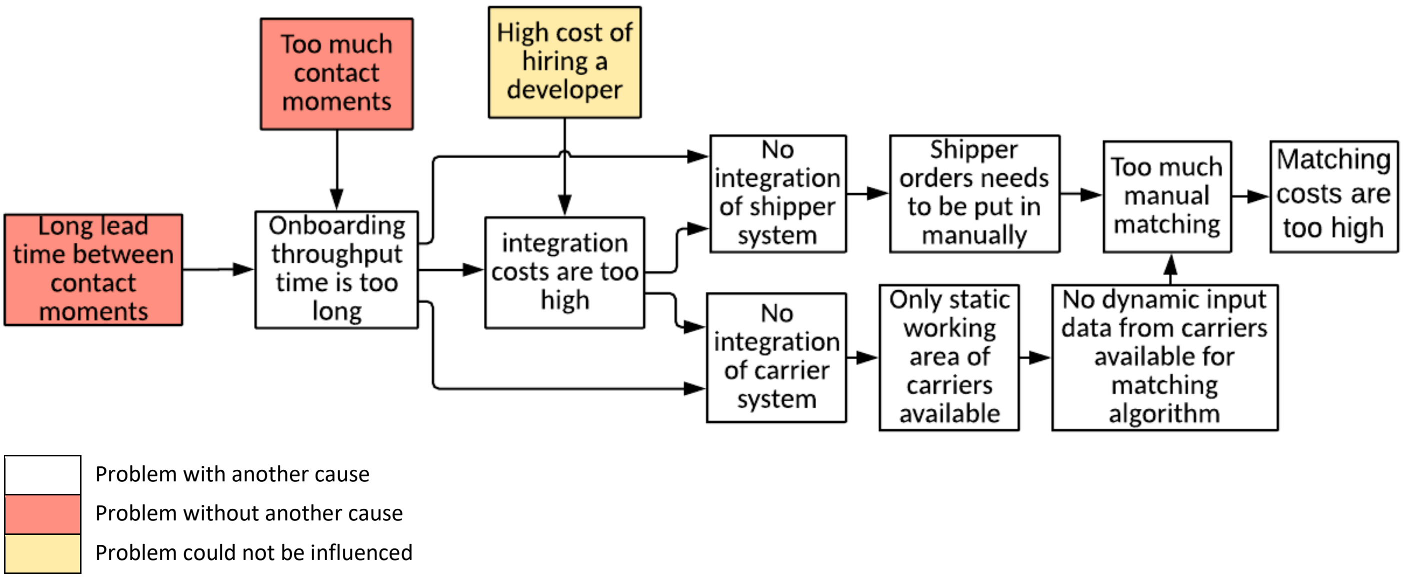

Figure 2 Problem cluster

1.2.3 Core problem

Heerkens and van Winden (2012) specify two types of problems, problems with another cause or problems without another cause. Problems without another cause are potential core problems if they could be influenced. In figure 2 the problems are modelled following these specifications.

In figure 2 two potential core problems are given. The first problem states that there are too many contact moments, with as a result that the onboarding throughput time is too long. The second problem states that the waiting time between contact moments is too long, which results in a long onboarding throughput time. In the case of a digital contact moment, the waiting time is the time between sending a message and receiving a response. In case of a physical contact moment, the waiting time is the time between when the appointment is made and the actual appointment.

So, both problems influence onboarding throughput time. As described in section 1.2.2 this throughput time varies from a week to multiple months.

Both problems are related to waiting time. As described in section 1.2.2 the onboarding throughput time varies from a week to multiple months. Where this throughput time is mainly the waiting time. Since the total waiting time for all the contact moments could be seen as the sum of the waiting times (W) between the contact moments 1,2,…, n-1, n. Where n is the last contact moment and the total amount of contact moments is n. This could be represented in the following equation.

$$W_1 + W_2 \ldots + W_{n-1} + W_n = W$$

W_i = waiting time (i = 1,2,…,n-1,n) Where n is the total amount of contact moments
W = total waiting time

Reducing the waiting time per contact moment or reducing the number of contact moments will reduce the total waiting time. From an eMagiz services perspective, it is hard to reduce the waiting time between contact moments because they are dependent on the readiness of other parties which are involved. These parties have their own agenda which are often full, especially on short time ranges. From the eMagiz services perspective, it is easier to reduce the number of contact moments. The contact moments are mostly requested by eMagiz services because they need *information* (further explained in chapter 3) from the involved parties. So, eMagiz services mostly determine the amount of contact moment themselves. Therefore, in consultation with the involved parties, it is decided that reducing the number of contact moments is a more feasible problem and has a bigger potential in reducing the waiting time. So, the core problem in this research is defined as follows:

There are too much contact moments between eMagiz services and the involved parties.

1.2.4 Research focus

This research is about the onboarding process for CargoMatcher. In this research, the possibility to offer a redesigned integration process is investigated, where integrations are not built from scratch (as described in chapter 3) but are made with prebuilt standard connectors which are constructed following market standards. In this research criteria and a prototype are established for a new eMagiz services application, which will support this new type of integration.

During the redesign, the focus is more on the contact moments between the involved parties as on the technical aspects of the onboarding. These contact moments are necessary for eMagiz services to get the necessary information to set up the system integration, but in the current situation, this causes a long waiting time. In the current situation, the long waiting time mainly causes the long onboarding throughput time. Next to that, the costs for system integration are too high for CargoMatcher, therefore a reduction of the onboarding costs is also an important aspect in this research.

1.2.5 Variables and indicators

After it is clear what the research problem is, a variable is needed to be able to measure the influence of the proposed solution. The variable in this research is the onboarding throughput time. This variable will be measured with three indicators. The first indicator is the expected amount of contact moments. The second indicator is the expected time for completing the Critical path and the third indicator is the time variance for the critical path. In this research, the current onboarding process is modelled in Business Process Mapping Notation (BPMN) models (section 3.1 and Appendix 3). These models are focused on identifying the contact moments. From these BPMN models, an AON process network for a PERT analysis (Winston, 2003) is extracted (section 3.2). This PERT analysis is conducted to give an insight in the current critical path and expected duration and variation of this path, which is further used to calculate the chance that the process is finished within a certain amount of time. In chapter 4 and Appendix 7 the desired process is mapped in BPMN and PERT models. From these models, a new measurement is conducted, which is compared to the zero measurement and the norm in chapter 7.

1.2.6 Norm and reality

The core problem is identified (section 1.2.2 and 1.2.3) and a variable with indicators is assigned to this core problem (section 1.2.5). For this variable, a norm and a reality value (the zero measurement) are needed (Heerkens & van Winden, 2012). This is needed to be able to see the difference between the reality and the norm, and to see if the proposed solution solves the problem (meets the norm) or not (does not meet the norm). In this research, the norm and reality are determined in terms of the chance that the throughput time of the onboarding process is less or equal to 3 months. In other words, what is the chance that the onboarding process is completed within 3 months from now if the tenant requests a system onboarding at CargoMatcher today? The value that is determined as the norm for this change is 85 per cent. So, the chance that the throughput time is less or equal to 3 months needs to be at least 85 per cent. The zero measurement (section 3.2) shows that in the current situation this chance is equal to 15,6 per cent. So, the reality value (15,6 per cent) is 69,4 per cent lower as the norm (85 per cent).

1.3 Research questions and Methodology

The goal of this research is to come up with an onboarding strategy that suits the needs of CargoMatcher described in section 1.2. For validation purposes, a prototype is made in chapter 5. A research methodology that is suitable for conducting research for the development of an artifact is the Design Science Research Methodology (DSRM).

DSRM is a method for conducting design science research in information systems (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007). In this research, the DSRM will be used for design, where the solution will be provided as a prototype artifact. Design science is of importance for the creation of successful artifacts. The DSRM consists of six phases. These six phases are explained below and shown in figure 3.

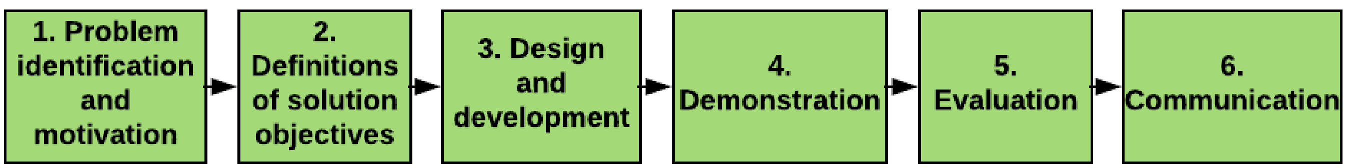

Figure 3 Six phases of DSRM

knowledge is required in some of the phases. This is represented in terms of knowledge questions. These knowledge questions are ordered according to the corresponding phase where the knowledge is required. These knowledge questions are supportive to the main research question which is answered in this research:

"How to reduce the expected throughput time for the onboarding process, by altering the onboarding strategy with the use of standard market connectors in a tenant onboarding tool?"

DSRM Phase 1: Problem identification and motivation

In this phase, the definition of the specific research question is given and a justification for the value of a solution is given. The importance of the problem and the relation with other problems becomes clear from this phase.

Knowledge questions

1. *What is the expected length of the critical path of the current onboarding process between eMagiz services, CargoMatcher, and CargoMatcher's customer?*
    1.1. *What does the current onboarding process of eMagiz services for the integrations to the CargoMatcher platform look like?*

1.2. *Which tool is suitable for measuring the expected throughput time of the critical path of the onboarding process?*

Phase 2: Define the objectives for a solution
Here the elements of a solution are derived from the problem definition from phase 1, with a list of feasible knowledge, different possibilities and a definition of what the solution should accomplish as a result.

Phase 3: Design and development
In this phase the artifact's desired functionality and its architecture are determined by choosing the best solution proposed in phase 2, followed by the creation of the actual artifact. In this research, this means the derivation of user stories from the new solution design followed by the realization of these user stories in a prototype.

Knowledge questions

2. *What is the expected throughput time length of the critical path for the desired onboarding process?*
   2.1. *What will the desired onboarding process of eMagiz services for the integrations to the CargoMatcher platform look like?*

3. What are the user requirements for the tenant onboarding prototype?
3.1 Which decision tree type could be used for the optimization of the selection of standard connectors for a redesigned tenant onboarding process?

Phase 4: Demonstration
In the demonstration phase, the use of the artifact to solve one or more instances of the problem is demonstrated and tested. In this research, the demonstration phase is used as a proof of concept of the newly designed process.

Phase 5: Evaluation
Phase 5 includes observation and measurement of how well the artifact supports a solution to the problem. This activity involves comparing the objectives of a solution to actual observed results from use of the artifact in the demonstration.

Knowledge question
4. Does the prototype fit the user requirements?

Phase 6: Communication
In the last phase the problem and its importance, the artifact with its utility and novelty is communicated to the audience. This is done by this research paper and a final presentation to eMagiz services.

1.4 Report structure

This report consists of eight chapters. Table 1 shows to which phase of the DSRM each chapter belongs.

Table 1 Structure of the report

Chapter	DSRM phase
1. Introduction	Problem identification and motivation
2. Theoretical framework	Design and development
3. Current situation	Problem identification and motivation
4. Desired situation	Design and development
5. Prototype	
6. Validation	Demonstration
7. Evaluation	Evaluation
8. Conclusions	

2. Theoretical framework

In section 1.3 eight knowledge questions are stated. For knowledge question 3 and 4 suitable tools need to be selected. Information about different tools is researched in this theoretical framework. In section 2.1 knowledge question 3 is answered and in section 2.2 research knowledge 4 is answered.

2.1 Decision trees (systematic literature review)

2.1.1 Necessity of literature review

In section 1.3 the research methodology and research questions for this research are described. One of those knowledge questions is sub knowledge question 3.1:

3.1 *Which decision tree type could be used for the optimization of the selection of standard connectors for a redesigned tenant onboarding process?*

For the prototype described in chapter 5, a decision tree will be implemented for the selection of standard connectors. This decision tree will be based on the information of this literature review. A decision tree is a map of the possible outcomes of a series of related choices (LucidChart, n.a.). It allows an individual or organization to weigh possible actions against one another. A decision tree typically starts with a single node, which branches into possible outcomes. Each of those outcomes leads to additional nodes, which branch off into other possibilities. In a decision tree, there are different types of nodes. The types of nodes are the decision node, the chance node and the endpoint/utility node. Putting all branches and nodes together gives it a tree-like shape.

In other words, a decision tree is a model that encodes the structure of the decision problem by representing all possible sequences of decisions and observations explicitly in the model (Jensen & Nielsen, 2007). Going through the tree will end up with a selection of endpoint/utility nodes. In this research, an endpoint/utility node represents a standard connector for a software system. The more systems eMagiz services will support, the bigger the list of standard connectors (Appendix 8) will be. Customers need to select the right standard connector for a successful system integration. Therefore, it is important to only provide standard connectors to the customer, which fulfil the requirements of the previously made decisions.

A decision tree is a clear method for modelling these decisions, but it is based on numerical branch values or clear grammatical values that are known beforehand. In this research, the grammatical values are sometimes vague, and concepts could change over time. In other words, the values are not certain. Therefore, the model needs to be updatable and agile. To find a solution to this problem a systematic literature review is conducted.

2.1.2 Systematic literature review

This literature review is based on the concept of linguistic decision trees. A linguistic decision tree is a framework for modelling with words (Qin & Wan, 2013). A linguistic decision tree is a variant of the decision tree. As described before, a decision tree is a model that encodes the structure of the decision problem by representing all possible sequences of decisions and observations explicitly in the model (Jensen & Nielsen, 2007). For this literature review a concept map, search strings with inclusion and exclusion criteria and a review protocol are made. This could be found in Appendix 2.

Table 2 shows the papers with their corresponding paper number, which are left after following the review protocol. The information of these papers is put into a concept matrix which is shown in Appendix 2B and 2C.

Table 2 Articles for literature review

Reference	**Paper number**
(He, Watson, Maple, Mehnen, & Tiwari, 2017)	paper 1
(He & Lawry, 2014)	paper 2
(Lawry, 2008)	paper 3
(McCulloh, Lawry, & Cluckie, 2008)	paper 4
(Qin & Wan, 2013)	paper 5
(Qin & Lawry, 2011)	paper 6
(Qin & Lawry, 2008)	paper 7
(Turnbull, Lawry, Lowenberg, & Richards, 2016)	paper 8

As could be derived from the concept map the following are the concepts used in this systematic literature review:
Concept 1: Linguistic decision trees
Concept 2: label semantics
Concept 3: Hierarchies of linguistic decision trees
Concept 4: Algorithm for generating linguistic decision trees

Linguistic decision trees
Tree induction learning models have received a great deal of attention over recent years in the field of machine learning and data mining, because of their simplicity and effectiveness (Qin & Lawry, 2008). So, does the linguistic decision tree (LDT). The LDT is a decision tree with attributes as nodes and linguistic descriptions of attributes as branches (Lawry, 2008). Within an LDT, each branch has an associated probability distribution on the different classes (Qin & Wan, 2013). Classes are different outcomes that a branch of an LDT could have. These classes are defined with label semantics. The model of the LDT is based on a framework for modelling with words (Qin & Wan, 2013). The LDT could be used as a classification model, or as a prediction model which is based on label semantics.

Label semantics
An LDT makes use of label semantics for defining the labels of the LDT. Label semantics is based on an

epistemic theory of vagueness (Lawry, 2008). This vagueness or impreciseness comes from the fact that human beings constantly use imprecise language to communicate with each other (Qin & Lawry, 2011).

Label Semantics is a random set-based framework for modelling with linguistic expressions based on this imprecise language which provides linguistic labels such as small, large, short, tall, young, old and so on (Qin & Wan, 2013). The fundamental question posed by label semantics is how to use linguistic expressions to label numerical values. The basic idea is that when individuals make assertions, such as 'The robber is tall', they are essentially providing the information that the label tall is appropriate for describing the robber's height. In this example, there could be an ordered preference between the labels where tall is more appropriate than medium which is in its turn more appropriate than short. The choice of words to describe the robber should surely then be based on these judgements about the appropriateness of labels. It is suggested that the vagueness of these description labels lies fundamentally in the uncertainty about when they are appropriate as governed by the rules and conventions of language use (Lawry, 2008). Labels may have different degrees of vagueness, for example when we say Peter is young and he is male, the label young is vaguer than the label male because people may have more widely different opinions on being young than being male. For a particular concept, there could be more than one label that is appropriate for describing this concept, and some labels could be more appropriate than others.

Label semantics provides two fundamental and interrelated measures on labels to describe an object in an underlying domain, the measure of appropriateness and the measure of mass assignments on labels (He, Watson, Maple, Mehnen, & Tiwari, 2017). A mass assignment on a set of labels x quantifies the belief that any special subset of labels contains all and only the labels, with which it is appropriate to describe x. Label semantics makes hereby use of a finite set of successive labels to describe an object in an underlying domain (He & Lawry, 2014).

Hierarchies of linguistic decision trees

Next to labels for the branches of the decision tree, a decision tree also consists of the order of nodes. In other words, the hierarchy of the decision tree. A hierarchy for an LDT is called a Linguistic Attribute Hierarchy (LAH) (He & Lawry, 2014). In a LAH the functional relationship between child and parent nodes are not defined precisely. Instead the labels for a parent attribute are defined in terms of the labels describing the attributes corresponding to its child nodes, by means of a linguistic decision tree (Lawry, 2008). An example of how such a LAH looks like is given in figure 4. Here the child nodes labels are defined in terms of x (x_1, x_2, x_3, x_4) and therefore the parent node is also labelled in terms of x (x_5).

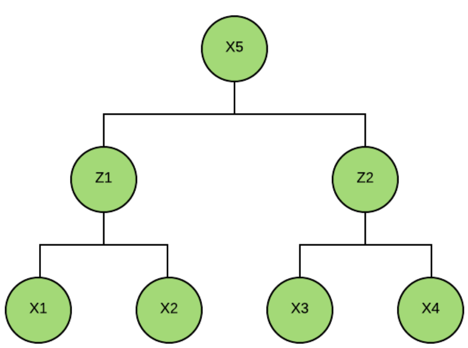

Figure 4 Example of LAH

It is not easy to define the labels for intermediate attributes in terms of their children, as the introduced intermediate attributes are not directly related to basic attributes (He, Watson, Maple, Mehnen, & Tiwari, 2017).

Algorithm for generating linguistic decision trees
As stated before, tree induction learning models and algorithms have received a great deal of attention over recent years in the field of machine learning and data mining (Qin & Lawry, 2008). There is also a specific algorithm for constructing LDT's, this algorithm is based on the ID3 algorithm and is called LID3 (He & Lawry, 2014). A LID3 works as follows, the most informative attribute is selected as the root of an LDT, and the tree will be expanded with branches associated with all possible focal elements of this attribute. For each branch, the attribute with maximal information gain in all free attributes will be the next node until the branch reaches the specified maximum length or the maximum class probability reaches the given threshold. The advantages of this LID3 are that it offers an inherent model of uncertainty, it offers good performance, robustness and generalization, and the branches of the tree could be interpreted as a set of linguistic rules based on the principals of label semantics (Turnbull, Lawry, Lowenberg, & Richards, 2016).

2.2 Quantitative analysis of business processes

In section 1.2.5 it is mentioned that the expected duration and the variation of the critical path of the onboarding process are measured. This is done by executing a quantitative analysis. A quantitative analysis is a mathematical analysis, based on a precise model of a business process supplemented with quantitative data (Berg, Franken, & Jonkers, 2008). Berg, Franken and Jonkers (2008) give three categories of the most common types of quantitative analysis. These three categories are:

1. Utilisation rate, Waiting time and stock
2. Critical path, alternative paths and throughput time
3. Costs

This theoretical framework had the purpose to provide the knowledge for a suitable analysis tool for the necessary measurements described in section 1.2.5. Therefore, this theoretical framework is only focused on the second category (critical path, alternative paths and throughput time). In a quantitative analysis of this category, different paths in a procedure are identified, visualized and an average throughput time for these paths are determined. An analysis in this category is based on the procedure of the different actions, where the individual allocation of actions to different actors has no influence on the results.

This section gives an answer to the following knowledge question, which is stated in section 1.3 as sub knowledge question 1.2.

Which tool is suitable for measuring the expected throughput time of the critical path of the onboarding process?

2.2.1 Quantitative analysis techniques

There are two general techniques to execute a quantitative analysis. These two techniques are the analytical approach and quantitative simulation. With an analytical approach, a formula or numerical method is used to come to one conclusive answer. Where a quantitative simulation makes use of probability distributions to come to multiple broader results (Berg, Franken, & Jonkers, 2008). Both techniques have their own characteristics as shown in table 3

Table 3 Characteristics of analytical approach and quantitative simulation

Characteristic	**Analytical approach**	**Quantitative simulation**
Time efficiency	High	low
Difficulty of use	Low	High
Getting results	Quick	Slow
Amount of results	One conclusive result	Many broad results
Amount of options in the model	Few options	Many options
time effects	Not visible	Visible
Is applicable to newly designed processes?	Yes	No

2.2.2 Analytical approach

This research proposes a new design for the onboarding process. From section 2.2.1 could be concluded that quantitative simulation is not applicable to analyse this newly designed process and the analytical approach is applicable to analyse the newly designed process. Therefore, no further information about quantitative simulation is provided.

A set of multiple actions that need to be executed in a certain order is called a process. A path is a sequence of actions in a process (Berg, Franken, & Jonkers, 2008). In figure 5 this difference is illustrated. In this figure, action 1 up to and including action 6 is called the process. The process consists of two paths, which are illustrated with blue and red arrows. The blue path is action 1 – action 2 – action 3 – action 4 and the red path is action 1 – action 5 action 6 – action 4. Each path has its own throughput time. The path with the longest throughput time is called the critical path. The critical path has the longest throughput time and therefore dominates the throughput time of the process. Influencing the throughput time of the critical path directly influences the throughput time of the whole process. Hence, a reduction of the throughput time of the critical path has the biggest opportunity for an optimisation of the throughput time of the process.

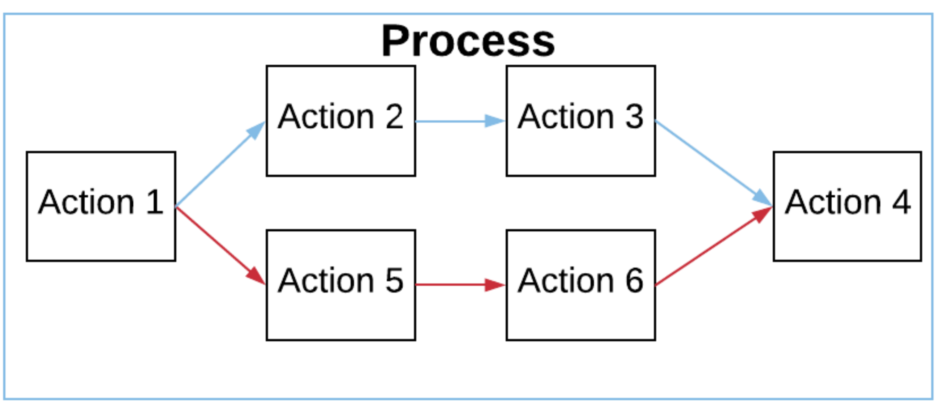

Figure 5 Illustration difference between paths, actions and processes

Since the biggest optimisation opportunity for the process throughput time lays at the critical path it is necessary to identify this critical path. Two traditional and often used methods for identifying the critical path are the Critical Path Method (CPM) and Program Evaluation Review Technique (PERT).

2.2.3 Critical Path Method (CPM)

CPM could be used if the duration of each action is known with certainty (Winston, 2003). With this method, it is possible to determine the length of time required to complete a process or to determine how long each action in the process could be delayed without delaying the completion of the process. The procedure for constructing a CPM model is as follows:

1. Make a list of all the actions that need to be completed to complete the whole process.
2. Determine the duration of each individual action. This is one fixed number (that is known with certainty).
3. Determine de predecessors of each action. For each action there is a set of actions that must be completed before the action begins, this is called the predecessors of the action.
4. Model al the actions in an activity on node (AON) project network.
5. Use a forward pass to determine the early start (ES) and the early finish (EF) of each action.
6. Use a backward pass to determine the late start (LT) and the late finish (LF) of each action.
7. Calculate the slack time.
8. Identify the critical path. The actions with zero slack time are called critical actions. The combinations of these critical actions are called the critical path.

The result of the CPM model is that it is possible to say what the duration will be of the process. This duration is equal to the length of the critical path and is known with certainty.

2.2.4 Program Evaluation Review Technique (PERT)

PERT could be used if the duration of each action is not known with certainty (Winston, 2003) to estimate the probability that the process will be completed by a given deadline. The procedure for constructing a PERT model is slightly different as for the CPM model. The procedure is as follows:

1. Make a list of all the actions that need to be completed to complete the whole process.
2. Determine the duration of each individual action. These are three individual time estimates (the optimistic, the pessimistic and the most likely time).
3. Calculate the expected time and the variance of each action.
4. Determine de predecessors of each action. For each action there is a set of actions that must be completed before the action begins, this is called the predecessors of the action.
5. Model al the actions in an activity on node (AON) project network.
6. Use a forward pass to determine the early start (ES) and the early finish (EF) of each action.
7. Use a backward pass to determine the late start (LT) and the late finish (LF) of each action.
8. Calculate the slack time.
9. Identify the critical path. The actions with zero slack time are called critical actions. The combinations of these critical actions are called the critical path.
10. Determine the process standard deviation. (square root of the sum of all the individual variances of the critical path)

2.2.5 Selection of analytical approach method

Knowledge question 4 states the following:

4. Which tool is suitable for measuring the expected throughput time of the critical path of the onboarding process?

Both CPM and PERT look like one another. The main difference between the two methods is the time estimation for the duration of individual actions. The CPM uses one duration time that needs to be known with certainty. PERT makes use of time estimations and the variance of this time estimate. In this research, the length of each action in the onboarding process is not known with certainty. Therefore, it is decided to use the PERT methodology since the PERT methodology is suitable to deal with this uncertainty.

3. Current situation

3.1 Current onboarding process

The current onboarding process consists of different phases where communication between the different parties is required. This communication is a time-consuming process. To get a better understanding of the process insight into the different phases is provided. These different phases are given in figure 6, and BPMN models of this process are given in Appendix 3. In these BPMN models, the same colours as in figure 6 are used to indicate to which phase of figure 6 it belongs.

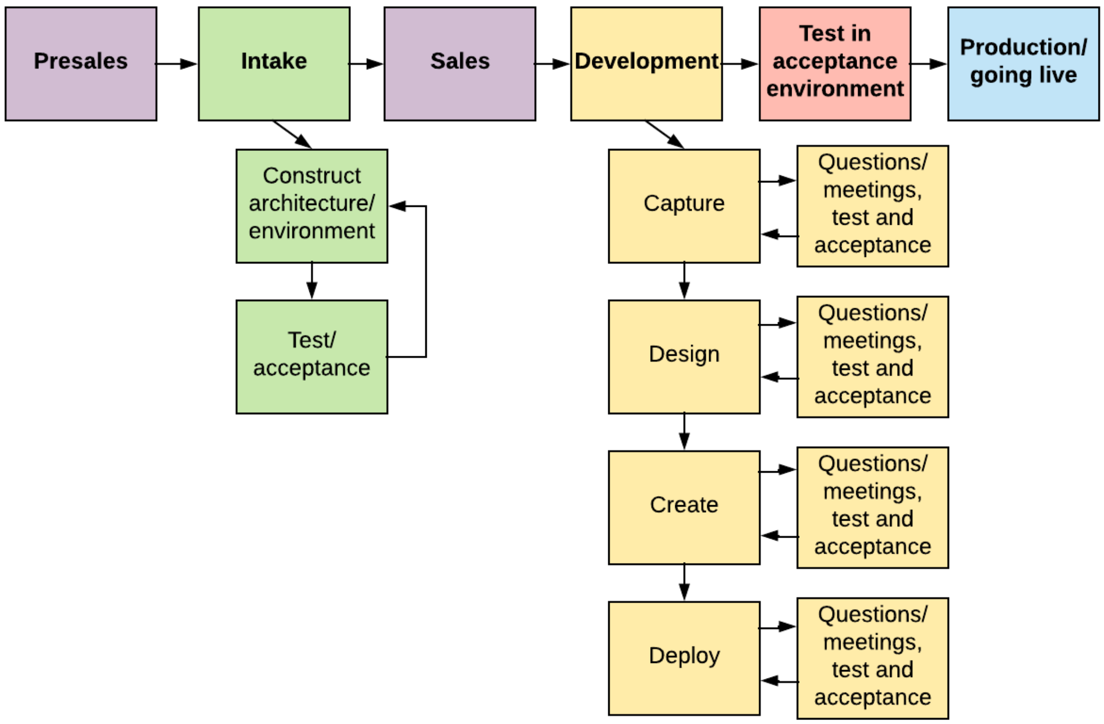

Figure 6 Phases of the onboarding process

The average onboarding process starts with the presales. Presales means that there is an opportunity for the onboarding of a customer. During presales, the customer discusses information about the services that eMagiz could provide. In this process, a customer could decide to get an intake for a solution design. If a customer decides to take an intake, then first an intake form is exchanged between the different parties. This is followed by a meeting between the different parties. From the intake form and the information that is gathered during a meeting, a solution design is constructed. In this solution design, the current architectures and processes of the customer are described. Next to that, it is given how the new architecture will look like, and on which processes the new solution has its influence. This solution design is provided to the customer, which then decides if it accepts or refuses the solution design. If the solution design is accepted, then the actual development starts. This development is executed by a developer which keeps contact with the other parties to gather the necessary details during the development process. After the development, the developer starts testing by exchanging test messages between systems over the message bus. If no errors occur during this testing, then the developer promotes the developed solution to the acceptance environment. In this environment, the developed solution is tested by CargoMatcher and their customer. If no errors occur, then the solution will be

prepared to be taken into practice.
This is called "going live and implementing in production".

3.2 PERT analysis of the current situation

In the previous section, the current onboarding process is visualized and modelled in multiple BPMN models. In this section, the estimated throughput time of the modelled process is determined with the use of a PERT analysis (Winston, 2003). This is a technique that is selected based on the found literature from the theoretical framework (section 2.2). In a PERT analysis, it is determined what the individual actions are and what the durations and predecessors are from these actions. In Appendix 4A 19 actions are extracted from the BPMN models. In this Appendix, it is determined what the description of the action and what the time span of each action is. For each of the individual actions, the predecessors are determined and put in an action on node (AON) project network (Winston, 2003). This AON project network is shown in figure 7.

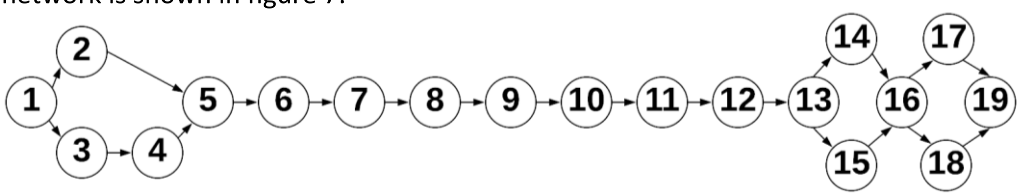

Figure 7 AON project network of the current situation

Each number in the AON project network refers to an action that is defined in Appendix 4A. In table 4 these action numbers (first column) and their description (second column) are given. The third, fourth and fifth column of table 4 show consecutive the best, most likely, and the worst time estimate for the given action. Appendix 4B describes how the values for these time estimates are determined.

Table 4 PERT estimations of individual actions

Action	Description	best	Most likely	worst	T	var	S
1.	Request system onboarding by tenant	0	0	0	0,000	0,000	0,000
2.	Inform eMagiz services about request	0,125	0,5	1	0,521	0,021	5,167
3.	Send intake form to tenant	0,125	0,5	1	0,521	0,021	0,000
4.	Complete and send intake form	1	5	10	5,1667	2,250	0,000
5.	Plan meeting	0,5	3	10	3,750	2,507	0,000
6.	Having a meeting	5	10	25	11,667	11,111	0,000
7.	Making and sending a quotation	0,125	1	5	1,521	0,660	0,000
8.	Accepting the quotation	1	5	25	7,667	16,000	0,000
9.	Capture phase	0,625	5	25	7,604	16,504	0,000
10.	Design phase	0,5	4	20	6,083	10,563	0,000
11.	Create phase	0,375	3	15	4,563	5,941	0,000
12.	Deploy phase	0,25	2	10	3,042	2,641	0,000
13.	Giving a test package for chain test	0,25	0,5	5	1,208	0,627	0,000
14.	Chain test by customer	1	5	10	5,167	2,250	0,000
15.	Chain test by tenant	1	5	10	5,167	2,250	0,000
16.	Accepting the test package	0,125	0,5	5	1,188	0,660	0,000
17.	Plan meeting to go live	0,5	3	10	3,750	2,507	2,542
18.	Prepare to go live	0,25	0,5	5	1,208	0,627	0,000
19.	Having a meeting/going live	5	10	25	11,667	11,111	0,000

PERT offers statistical tools, that uses these three time estimates as input, to calculate the expected throughput time (T) and the corresponding variance (var) per action (Winston, 2003). The T and the var value are shown in the sixth and seventh column of table 4. The total expected throughput time of the whole process is directly related to the length of the critical path as is described in section 2.2. Therefore, the critical path is determined by calculating the slack time (S) (eighth column). Every action where the slack time is equal to zero is considered as a critical action and lays on the critical path of the process. Step 14 and 15 both have zero slack time and could be executed at the same time. This results in two critical paths namely: 1,3,4,5,6,7,8,9,10,11,12,13,14,16,17,19 and 1,3,4,5,6,7,8,9,10,11,12,13,15,16,17,19. This is also shown in figure 8, where the colour green indicates the critical actions and the critical path. The expected throughput time (T) and the variance (var) of both the critical paths are equal, therefore these paths are considered equal and are further referred to as "critical path".

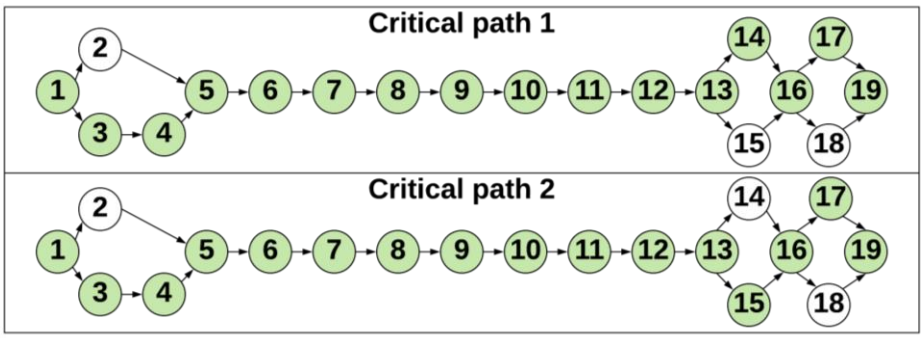

Figure 8 The critical paths

The total expected throughput time of the current onboarding process is 74,563 working days, which is the sum of all the expected throughput times of the individual critical actions. The standard deviation of the process is 9,239 working days, which is equal to the square root of the sum of the variances of the individual critical action. By assuming that the critical path is normally distributed, it becomes possible to calculate a z-value, this z-value determines the chance that the process is finished within a given due date, by extracting the chance value from a normal distribution table (Winston, 2003). For the zero measurement (section 1.2.6) the due date is set to 3 months. 2019 has 261 working days, and 3 months is one-fourth of one year. Therefore, the amount of working days, in 3 months, is set to 65,25. The z-value that belongs to this critical path is equal to -1,008. The normal distribution table gives a chance of 15,62 per cent for this z-value of -1,008. This means that in the current situation the chance that the process is finished within three months is equal to 15,62 per cent.

3.3 Current architecture

In section 3.1 and 3.2, the current onboarding process is described and the throughput time of this process is analysed. The result of the completion of the onboarding process is a system integration. This system integration is an integration to a message bus (Hohpe & Woolf, 2003). In figure 9 the architecture of such a message bus is shown. Here the big green rectangle represents the message bus, and the blue rectangles represent different systems. Both the systems as the message bus have their own format of how messages are structured. In this research Customer 1 represent CargoMatcher and Tenant 1, Tenant 2, and Tenant 3 represent different carriers and shippers. The green arrow represents a message type and points in the direction it is sent. In figure 9, Customer 1 has two outgoing message types (arrow points to the message bus) and three ingoing message types (arrow points to the system). The small green rectangles (with a white arrow) represent the connectors. A connector is either an onramp (connector points towards the message bus) or an offramp (connector points towards the system). Onramps and offramps are used to transform data from the system format to the message bus format or vice versa (eMagizChannel, 2018).

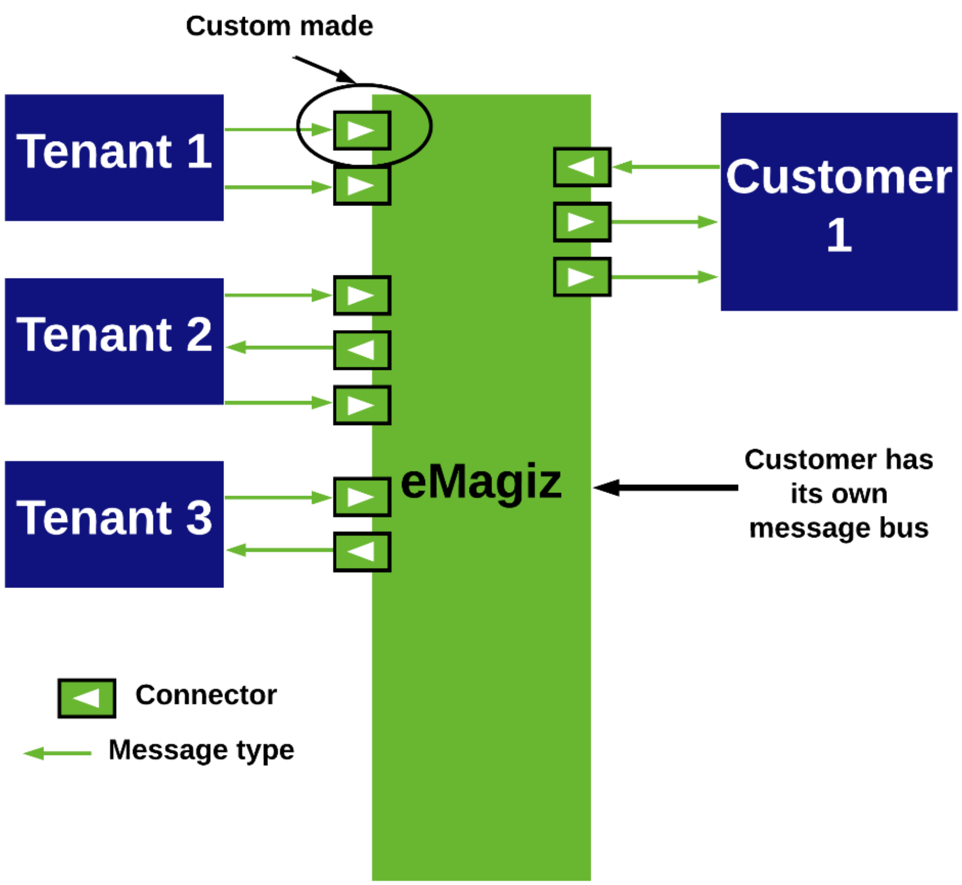

Figure 9 Architecture current situation one customer

In the current situation, each tenant or customer has for each message type an own custom-made connector. Figure 9 shows, six onramp connectors and four offramp connectors. These are ten custom-made connectors. In the case of CargoMatcher, there is only a small set of message types and a big amount of tenant systems. For example, CargoMatcher requests from every shipper a Proof of Delivery (POD), then there is an equal amount of entry connectors as the number of shippers needed for the message type POD. In this situation, a lot of entry connectors are made, which transforms the same POD format into the POD format of the message bus. Practice shows, that a lot of tenants use the same system or the different systems that are used use the same data format for a message type. In different market segments, such as the supply chain market, messages are standardized. An example of an organisation that makes such uniform international standards is GS1 (GS1, n.d.). Relating this back to the example of the PODs, this means that if a lot of the systems use this GS1 standard, then a lot of times the data transformation is mapped from the GS1 format to the message bus format.

There are more customers which have a comparable situation as CargoMatcher (a lot of systems with a small set of message types). Therefore, there are a lot of buses where the same transformation within the bus is made. The transformation for another customer, from the same message type to the message bus, could be different compared to CargoMatcher since each customer has its own message bus and therefore its own message bus format.

Figure 10 is illustrated to show this difference. Customer 1 has three tenants which have the ingoing standard message type GS1: POD. The transformation of the GS1: POD format to the message type for tenant 1 is equal to the transformation of tenant 2 and tenant 3. Customer 2 has three tenants (tenant 6, tenant 7, and tenant 8), with also three times the standard message type GS1: POD. For these three tenants, it also counts that the three transformations to the message bus are equal to one another. The message bus of Customer 2 has nevertheless another format as the message bus of Customer 1. Therefore, the transformations of the message type GS1: POD (to the message bus of customer 1) from tenant 1, 2, and 3 are not equal to the transformations of message type GS1: POD from tenant 6, 7 and 8 (to the message bus of customer 2).

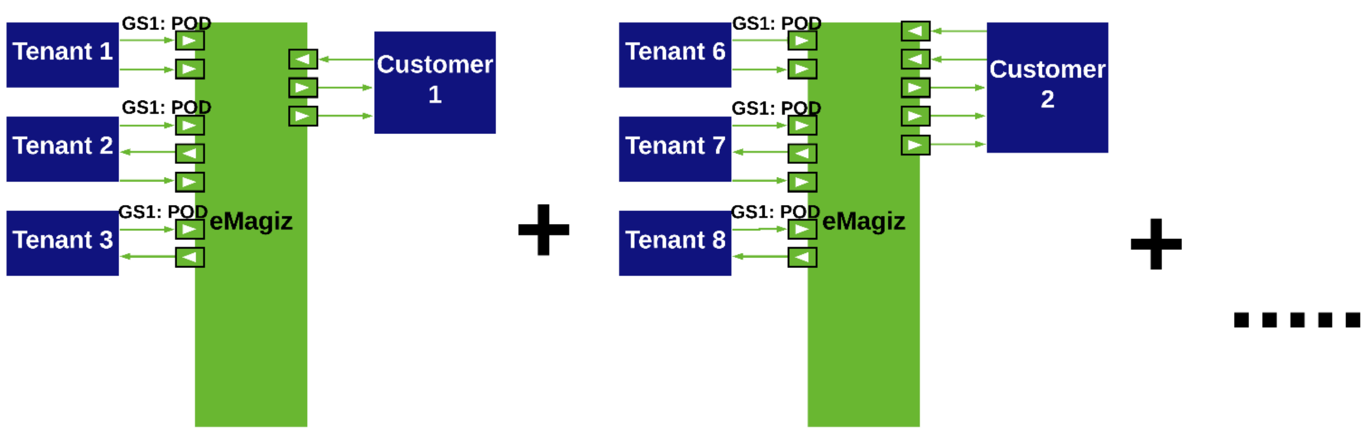

Figure 10 Architecture multiple customers with example message type GS1: POD

4. Desired situation

In this chapter, a redesigned onboarding process is proposed for the in chapter 3 analysed onboarding process. In section 4.1 the proposed onboarding process is explained and how it is extracted from the current situation. Section 4.2 contains a PERT analysis of the desired situation and section 4.3 shows the proposed architecture for this situation.

4.1 Redesigned onboarding process for tenants

In chapter 3 the current process is analysed. During this analysis, the following points for optimization of the onboarding process for CargoMatcher are found:

- The expected onboarding throughput time is 74,6 working days
- The expected throughput time for the development (Capture, Design, Create and Deploy) is 21,3 working days
- The number of tenants is big, but the set of message types and message type formats are small
- Tenants often use the same commercial off the shelf (COTS) software packages. Such as enterprise resource planning (ERP), transport management systems (TMS), advanced planning systems (APS), or warehouse management systems (WMS).

The combination of the current long development time per tenant, the use of standard message formats and the use of COTS software packages make the onboarding process interesting for a higher level of standardization. Instead of developing custom connectors (during the onboarding process) from scratch for every tenant individually, it is proposed to create connectors (beforehand) that are suitable for multiple tenants, who use the same message type formats. These connectors are further referred to as a standard connector.

The proposed onboarding process, where the standard connector is used, still starts with the presales. During the presales, there needs to be identified if the customer still needs a custom solution (current situation) or could make use of the proposed solution with the standard connector. If the customer is found suitable for the proposed solution, then an intake meeting needs to be set and conducted. During this meeting, an intake wizard is used to determine which standard connector the customer needs. The standard connectors are premade. Therefore, instead of making the connector fit to the customer and adapt to their message format (current situation), the customer needs to adapt to the requirements of the connector which are based on market standards. So, during the intake meeting it needs to become clear if the tenant meets all those requirements or not. If the tenant does not meet the requirements, the necessary information about which actions the tenant must undertake to meet the requirements could also be extracted from the intake wizard. The result of the intake meeting is a quotation. Just as in the current situation the tenant decides if it accepts the quotation or not. If the quotation is accepted, the tenant will onboard to the eMagiz service bus (see section 4.3). This is done by first processing all the information that is given in the intake wizard. Second, configuring the selected standard connector (Cambridge dictionary, n.d.). Third, composing a download package. This download package contains the configured standard connector and all the information that is necessary to install the standard connector. The tenant then receives this download package and conducts the installation of the download package (From here on the process stays the same as the current process). During acceptation, the solution is tested by CargoMatcher and their customer. If no errors occur, then the solution will be prepared to be taken into practice (going live and implementing in production).

Figure 11 shows an overview of the described process steps, where the same colours are used for each phase as in the current situation and orange borders are used to show that the step is different from the current situation. Detailed BPMN models of this process are given in Appendix 7.

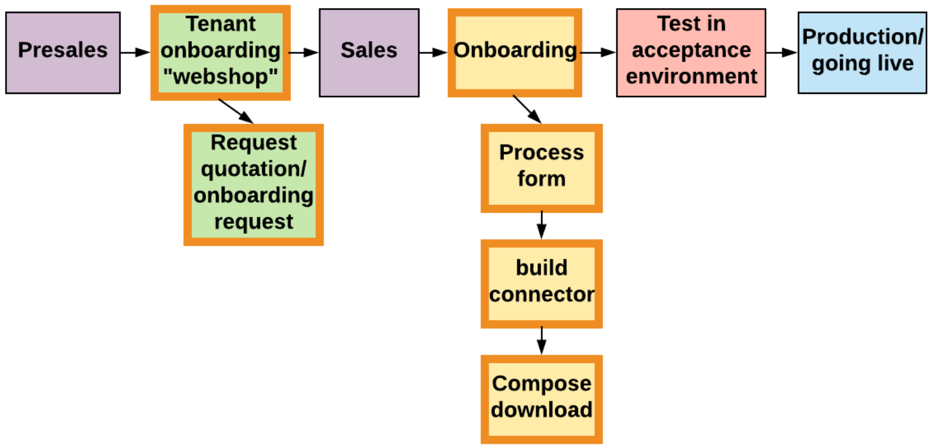

Figure 11 Desired tenant onboarding process

4.2 PERT redesigned the onboarding process

The redesign of the process has its influence on the expected onboarding throughput time. Figure 8 and table 4 of section 3.2 show 19 steps and their estimated values. The values for the actions that are not changed keep the same for the redesigned process. Based on a comparison between the BPMN models of Appendix 3 and 7 it is concluded that action 3 and 4 are removed from the model. Since no intake form is required anymore. In the current situation, there was a meeting (action 6) that differs from the redesigned intake meeting, but the throughput time is considered equal because the estimate was based on that is was a one-day meeting and not on the content of the meeting. Further action 9, 10, 11 and 12 are replaced by consecutive the processing of the form, building the connector, composing download, and the installation of the package. For these actions, new estimations and calculations are made. These values are shown in table 5 and a new AON project network is shown in figure 12. Action 9, action 10 and action 11 are actions that need to be executed by eMagiz services from without the eMagiz environment. Therefore, the time estimates of argument C6 of Appendix 4B are used for action 9, 10, and 11. Action 12 is compared to other estimations of actions that need to be executed by the tenant. Because the installation of the package requires multiple actions like the completing of an intake form (action 4 old model) or conducting a chain test (action 14 and 15) the estimations of these actions are used for action 12.

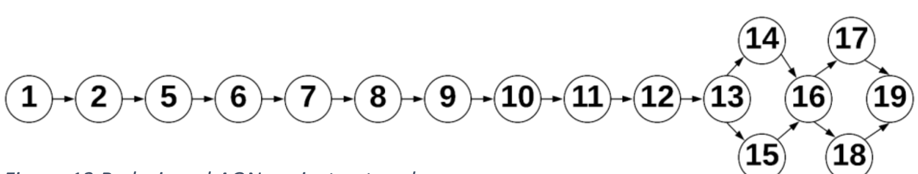

Figure 12 Redesigned AON project network

Table 5 Estimations of individual action (redesigned)

Action	Description	best	Most likely	worst	T	var	S
1.	Request system onboarding by tenant	0	0	0	0,000	0,000	0,000
2.	Inform eMagiz services about request	0,125	0,5	1	0,521	0,021	0,000
5.	Plan meeting	0,5	3	10	3,750	2,507	0,000
6.	Having a meeting	5	10	25	11,667	11,111	0,000
7.	Making and sending a quotation	0,125	1	5	1,521	0,660	0,000
8.	Accepting the quotation	1	5	25	7,667	16,000	0,000
9.	Processing form	0,125	0,5	5	1,188	0,660	0,000
10.	Building the connector	0,125	0,5	5	1,188	0,660	0,000
11.	Composing download	0,125	0,5	5	1,188	0,660	0,000
12.	Installation of the package	1	5	10	5,167	2,250	0,000
13.	Giving a test package for chain test	0,25	0,5	5	1,208	0,627	0,000
14.	Chain test by customer	1	5	10	5,167	2,250	0,000
15.	Chain test by tenant	1	5	10	5,167	2,250	0,000
16.	Accepting the test package	0,125	0,5	5	1,188	0,660	0,000
17.	Plan meeting to go live	0,5	3	10	3,750	2,507	2,542
18.	Prepare to go live	0,25	0,5	5	1,208	0,627	0,000
19.	Having a meeting/going live	5	10	25	11,667	11,111	0,000

Column eight of table 5 shows the slack time of each individual action. Every action where the slack time is equal to zero is considered as a critical action and lays on the critical path of the process. Step 14 and 15 both have zero slack time and could be executed at the same time. This results (just as in the current situation) in two critical paths namely: 1,2,5,6,7,8,9,10,11,12,13,14,16,17,19 and 1,2,5,6,7,8,9,10,11,12,13,15,16,17,19. This is also shown in figure 13, where the colour green indicates the critical actions and the critical path. The expected throughput time (T) and the variance (var) of both the critical paths are equal, therefore these paths are considered equal and are further referred to as "critical path".

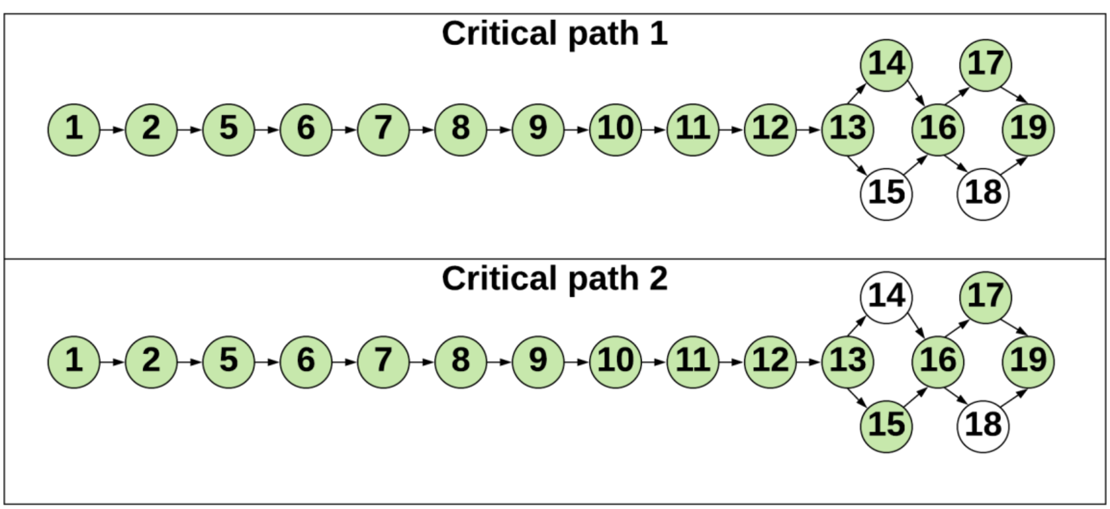

Figure 13 Critical paths redesigned process

The expected throughput time for the critical path is 56,833 working days with a standard deviation of 7,189 working days. By assuming that the redesigned process is normal distributed, the chance that the process is finished within 3 months from now (65,25 working days) could be determined by calculating the z-value. This z-value is equal to 1,17. Looking at the tables of normal distribution this results in a chance of 87,9% of finishing the redesigned onboarding process within 3 months.

4.3 Desired architecture

In section 3.3 the current architecture is shown, where the basic elements of an eMagiz message bus are explained. In the desired situation the custom connectors for tenants are replaced by standard connectors. These standard connectors are connectors that are made for an often used and standardized process. These connectors must be applicable to multiple tenants, with as goal that tenants, which use the same market standard and the same market process, could use the same connector (instead of the custom connectors as explained in section 3.3). What the effects are of the use of these standard connectors to the onboarding process is described in section 4.1.

The difference between the current and desired situation is, that in the desired situation, multiple customers could make use of one eMagiz message bus. Therefore, multiple tenants from multiple customers are also connected to the same bus. The desired architecture solves the problem described in section 3.3, that the buses of different customers have a different data format. Figure 14 shows this desired architecture, where it is shown that multiple tenants from different customers could make use of the same connector.

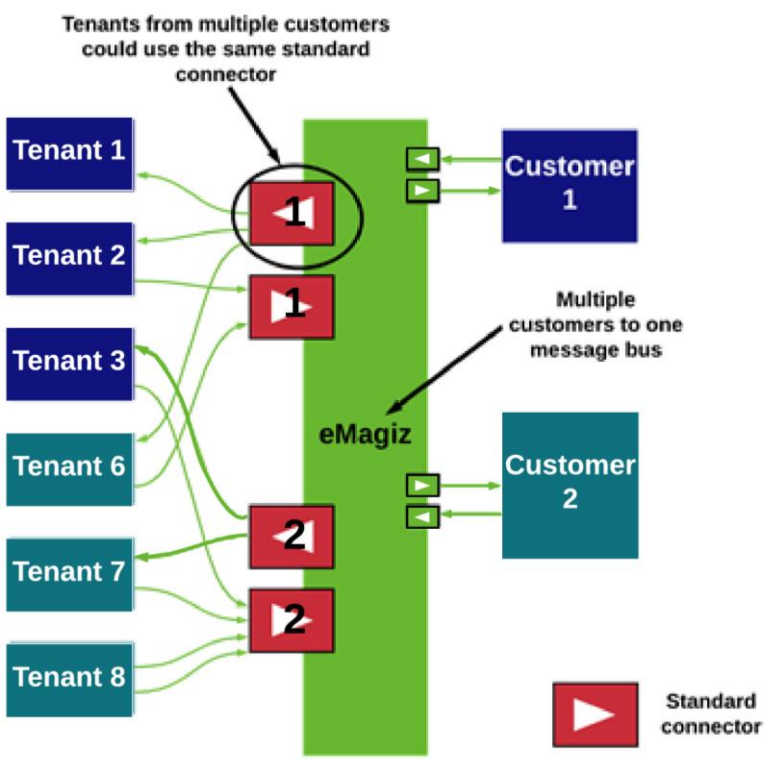

Figure 14 Desired architecture with standard connectors

5. Prototype

For this research, a prototype is constructed. This prototype is based on user stories and development tasks (Appendix 5) which are derived from the design of the desired process and architecture (chapter 4). Next to that, a mock-up is used, that is constructed by a UI/UX designer from eMagiz services for the layout of the application. Furthermore, the theory about linguistic decision trees (section 2.1) is used for the structure of the prototype.

5.1 Mendix prototype

Due to the limited amount of time, the prototype is designed as a minimal viable product (MVP). The prototype is considered MVP as all user stories (Appendix 5) are completed and it could function as an onboarding intake wizard, which could be used by customers from eMagiz, such as CargoMatcher. CargoMatcher should be able to use the prototype to do the intake for the onboarding of their customers, which make use of standard market process messages. These customers are further referred to as tenants.

The prototype is made within Mendix (Mendix, n.d.). Mendix is chosen because it is a low-code application development platform. Low-code means that it is a visual development approach to application development. This makes it easier to learn and use for non-technical developers than coding. Next to that, Mendix is often used by Cape consultants for their software development and the eMagiz integration platform is a Mendix application. Therefore, it is preferred to use Mendix.

Appendices 6A up to and including 6H show screenshots of the developed prototype. The prototype is made from a customer perspective. This means that the interface for the customer is fully functional and that other user roles are only used for input purposes. Appendix 8 shows the structure of the in the prototype implemented linguistic decision tree.

5.2 Prototype architecture

The prototype is a standalone Mendix app which is not connected to any current existing eMagiz services application (Mendix, n.d.). Figure 15 shows the place of the prototype in the desired architecture from section 4.3.

In section 4.3 the principle that multiple tenants could make use of the same connector is shown. Nevertheless, there are a lot of requirements a tenant needs to meet before it is suitable to make use of such a standard connector. Next to that, some information is necessary to configure the connector for the tenant (Cambridge dictionary, n.d.). The prototype is designed as a framework to check if there is a standard connector available and to have the possibility to ask for the necessary configuring information.

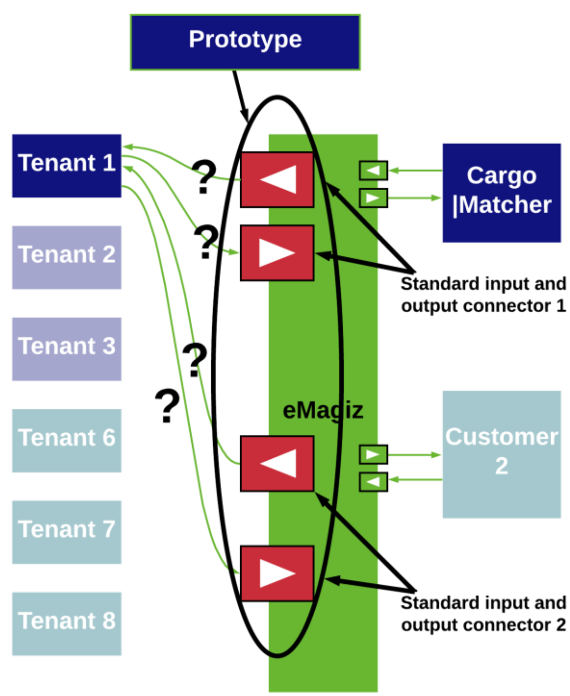

Figure 15 Onboarding wizard in architecture

6. Validation

In chapter 4 the current process is redesigned. The key change in the redesign is that custom made solutions are replaced by a premade solution, that is based on market standards. This restricts the applicability of the proposed solution because the tenant must adapt to the selected market standard of the available solution, instead of a tenant solution that adapts to the customer. Because the applicability is restricted it is necessary to identify in an early stage if a tenant could use such a solution or not. This is necessary information that needs to be known for sure before the quotation could be made because the quotation will state the type of solution and an agreement on the number of working hours. Making a custom solution will require way more working hours as configuring a standard solution that was premade. Therefore, if the quotation is accepted for a standard solution it is not justifiable to charge extra working hours. These extra hours will result in a reduction of profit or even huge costs instead of profit for eMagiz services.

The prototype has turned out to function well on the following points

- selection of different elements of the standard solution (process, process messages, system, etc.)
- providing information about where the solutions were based on
- asking for information from tenants and where to find this information

Nevertheless, the prototype turned out that this only functioned well for tenants that were suitable for a standard solution (the happy flow), but it was not useful for tenants that did not suit standard solutions. Completing the whole intake wizard takes a while. Questions that are asked at the end of the wizard could still determine if a tenant suits the solution or not. If it turns out at the end of the wizard that the tenant does not suit and the tenant does not want a custom solution, then the time used to give all the detailed information was not necessary, while identifying that the tenant was not suitable could have been identified in a more efficient way. Therefore, it is of high importance to know with high certainty that the customer is suitable for the standard solution or not.

7. Evaluation

The goal of this research was to reduce the expected throughput time and to increase the certainty that the onboarding process is finished within three months.

In section 3.2 the current situation is analysed with a PERT analysis. This analysis resulted in an expected throughput time of 74,6 working days and a standard deviation of 9,2 working days for the current onboarding process. Where there was a chance of 15,6 per cent that the onboarding process was finished within three months.

In section 4.2 there is also a PERT analysis conducted for the redesigned situation. Here the expected throughput time is 56,8 working days with a standard deviation of 7,2 working days. This resulted in a chance of 87,9 per cent that the onboarding process is finished within three months.

So, the expected number of working days for completing the onboarding process is reduced with 17,8 working days. Next to that, the standard deviation is decreased with 2 working days. This has as a result that the redesigned onboarding process has a 72,3 per cent higher chance to be completed within three months as the current onboarding process.

8. Conclusions

In the last chapter of this research document, conclusions concerning the research question and the multiple knowledge questions are given. Furthermore, the limitations of this research are discussed. Finally, recommendations for eMagiz services are described and which further work is necessary.

8.1 Research question and knowledge questions

In chapter 3 the following research question was stated:

How to reduce the expected throughput time for the onboarding process, by altering the onboarding strategy with the use of standard market connectors in a tenant onboarding tool?

To help to answer this research question, four knowledge questions with or without sub knowledge questions were stated (chapter 3). First, the conclusions for these four knowledge questions are stated in this section. Thereafter, these individual conclusions will be used to make a final conclusion on the research question.

The first knowledge question that was stated in chapter 3 is knowledge question 1:

1. *What is the expected length of the critical path of the current onboarding process between eMagiz services, CargoMatcher, and CargoMatcher's customer?*
    1.1. *What does the current onboarding process of eMagiz services for the integrations to the CargoMatcher platform look like?*
    1.2. *Which tool is suitable for measuring the expected throughput time of the critical path of the onboarding process?*

For the answer to sub knowledge question 1.2, a literature study is conducted in section 2.2 of chapter 2. In this literature study, a quantitative analysis tool is selected to measure the expected throughput time of the critical path of the onboarding process. The tool that is selected is a PERT analysis. During a PERT analysis time three time estimates are made for every individual action of the process. These time estimates are based on the worst case scenario, most likely scenario and best case scenario. With these three time estimates it is possible to calculate the expected throughput time for each individual action. By plotting all the actions in an Activity on Node (AON) project network, it is possible to determine the slack time of each individual action. Actions without slack time are called critical actions and lay on the critical path. The expected throughput time of the process is the sum of all the throughput times of the individual critical actions. Together with the process standard deviation, which could also be determined based on the three time estimations, it is possible to calculate the chance that a process is finished within a given time period.

Knowledge question 1.1 identified the current onboarding process between eMagiz services, CargoMatcher and CargoMatcher's customer. This is done in section 3.1 of chapter 3 by mapping the process in multiple BPMN models, which are shown in Appendix 3. From this BPMN process, it is concluded that it consists of 19 actions (Appendix 4A). This process includes actions which are related to the presales, intake, sales, development, test in acceptance environment and going live.

In Appendix 4B the individual throughput time of each action that is identified (knowledge question 1.1) is estimated following the PERT analysis (knowledge question 1.2). In section 3.2 of chapter 3, the critical path of the onboarding process is determined by mapping this in an AON project network. Here it is determined that the expected throughput time is 74,6 working days and that the standard deviation of current onboarding the process is equal to 9,2 working days. This results in a 15,6 per cent chance that the onboarding process takes 3 months or less.

The second knowledge question that is stated is knowledge question 2.

2. *What is the expected throughput time length of the critical path for the desired onboarding process?*
   2.1. *What will the desired onboarding process of eMagiz services for the integrations to the CargoMatcher platform look like?*

The answer to knowledge question 2 gives the expected throughput time for the redesigned onboarding process. To get the answer to this question, first the redesign of the process needed to be determined. This process is redesigned in section 4.1 of chapter 4.

In section 4.1 it is determined to change the strategy of making custom solutions for each individual tenant (current situation) to offering “standard solutions”, based on available standards in the market, which could be configured for multiple similar customers. This means that the customer has to adapt to the provided standard solution if the customer differ from the selected market standard, instead of that the solution adapts to the customer (custom solution).

The “standard solution” consists of a standard connector that is based on standards for message formats in the specific market segments and the configuration of this standard connector. In the redesigned situation the standard connector could already be made before a tenant requests onboarding, while in the current situation the custom connector is made during the onboarding process. This could be done, because all the necessary information for making a standard connector is available and the design of the connector is independent of the tenant.

This means that in the redesigned situation only the connector needs to be configured instead of developed. For configuration information is necessary. This information for configuration could be gathered during the intake. If the information could not be gathered during the intake, then there is a chance that the tenant is not suitable for the chosen connector. Gathering all the information during the intake is therefore a check if the tenant could make use of the “standard solution”. This needs to be identified before the quotation is stated because in the quotation the type of solution is stated. Therefore, the type of solution should not change after the quotation is accepted.

After the sales is completed, the configuration needs to be completed. The phase of completing the configuration is called the onboarding phase. After the onboarding phase is completed the process keeps the same as in the current situation.

The expected throughput time for the above described redesigned process is determined in section 4.2 of section 4. This expected throughput time is determined by a PERT analysis. The expected throughput time of the redesigned onboarding process is 56,8 working days and the standard deviation of this redesigned process is 7,2 working days. This results in a chance of 87,9 per cent for finishing the redesigned onboarding process within 3 months.

3. *What are the user requirements for the tenant onboarding prototype?*
   3.1 *Which decision tree type could be used for the optimization of the selection of standard connectors for a redesigned tenant onboarding process?*

In chapter 5 a prototype is constructed for the redesigned process of chapter 4. This prototype focuses on the altered phases of the redesigned process. These are the intake and the onboarding phase. In this prototype, filters are applied. These filters need a logical structure. This structure is determined with the use of specific type of decision trees. Sub research question 3.1 has determined this decision tree type. The selected decision tree type is the linguistic decision tree, which is suitable for modelling with words. The linguistic decision tree that is constructed for the prototype is shown in Appendix 8. The main requirement of the prototype is that needs to be able to select the relevant standard connector and to gather all the information necessary for the configuration of this connector. The specific user stories and the list of development tasks are shown in Appendix 5 and screenshots of the prototype are given in Appendix 6.

4. Does the prototype fit the user requirements?
Validation of the prototype is discussed in chapter 6. From this chapter, it could be concluded that the prototype functioned well on the following points for tenants that need a standard solution.

- Selection of different elements of the standard solution (process, process messages, system, etc.)
- Providing information about where the solutions were based on
- Asking for information from tenants and where to find this information.

Nevertheless, the prototype did not function well for tenants that did were not suitable for a standard solution, because the identification that a tenant is not suitable for the standard solution is not efficient.

From the answers to these four research questions, it could be concluded that the expected throughput time for the onboarding process could be reduced by using the proposed onboarding process that is stated in chapter 4. By using offering the standard connectors as proposed in this redesigned situation the expected throughput time is reduced with 17,8 working days compared to the current situation. Nevertheless, it could be concluded that the process works for tenants that are suitable for using a standard solution, but the prototype has turned out that more research is required for identifying if the solution is suitable or not for each individual tenant.

8.2 Limitations

As stated in section 1.2.4. this research is more focused on the process instead of on the technical aspects, due to a lack of knowledge. Nevertheless, it is confirmed by eMagiz services experts that using standard market connectors for multiple tenants is technically feasible. Next to that, in this research, the expected throughput time is calculated based on estimations. These estimations are based on interviews that were conducted during this research. Therefore, these values could differ from reality.

8.3 Recommendations and future work

Based on this research multiple recommendations and future work are proposed.

8.3.1 Determine if a tenant suits the tenant onboarding solution

From the research, it could be concluded that it is necessary to know if a tenant suits the proposed standard solution or does not suit this solution and still needs a custom solution. Therefore, it is recommended to do further research on this topic to be able to quickly identify if the tenant could make use of the proposed solution or not.

8.3.2 Determine when a standard solution is better as a custom solution

Building a standard solution requires more work as building a custom solution. The time profit of standard solutions is that after it is made it could be reused multiple times. For reusing no building time is required only some configuration time which is significantly lower as the building time of a customer solution. Therefore, the amount of sales needs to be determined. In other words, the amount of sales, when a standard solution is more beneficial as a custom solution, needs to be determined. This differs per situation. Therefore, research is required, to be able to determine beforehand if a standard solution will be beneficial in comparison to a custom solution.

8.3.3 Determine a revenue model

In the current situation, part of the revenue model of eMagiz services is selling a bus to a customer, where the customer pays for the specific bus. In the redesigned situation multiple tenants from and multiple customers will be connected to one eMagiz bus. Therefore, it needs to be determined what the costs for a customer will be and to have tenants connected to this eMagiz bus.

8.3.4 Determine who is going to enter the information in the onboarding wizard

In the redesigned process, going through the intake wizard in the intake phase is conducted during a meeting, where the eMagiz services consultant enters the information in the wizard. As identified the throughput time from the planning of the meeting up to including the meeting is expected 11,7 working days. This is 20,5 per cent of the expected throughput time of the whole onboarding process. If the tenant or customer could enter this information themselves without the need of having a meeting, then there is a significant potential for reducing the expected throughput time for the whole process. Nevertheless, in the redesigned situation there is chosen for the use of a meeting, because of the technical complexity. Therefore, research could be conducted to research if it is possible to let customers and/or tenants enter the information in the intake wizard and if so, how this could be realised.

9.Bibliography

Berg, H. v., Franken, H., & Jonkers, H. (2008). *Handboek business process engineering.* Enschede: Bizzdesign.

Cambridge dictionary. (n.d.). *configure*. Retrieved from dictionary.cambrigde.org: https://dictionary.cambridge.org/dictionary/english/configure

CAPE GROEP. (2018). *Over Cape*. Retrieved from https://www.capegroep.nl/over-cape/

CAPE Groep. (n.d.). *Uw Partner in wendbaarheid & innovatie*. Retrieved March 15, 2018, from https://www.capegroep.nl/

eMagiz. (n.d.). *About eMagiz*. Retrieved March 15, 2018, from https://www.emagiz.com/about/

eMagiz. (n.d.). *What we do?* Retrieved from https://www.emagiz.com/

eMagizChannel. (2018). entry- and exit connectors [Video]. Retrieved from https://www.youtube.com/watch?v=0hqlBFYY7zI&list=PLDR9Rd5_9t6jX-aX1hrmMj2mvFnAxwLSK&index=21

Gartner, Inc. (2018). *Integration Platform as a Service*. Retrieved from https://www.gartner.com/it-glossary/information-platform-as-a-service-ipaas/

global negotiator. (n.d.). *carrier*. Retrieved from dictionary of international trade: https://www.globalnegotiator.com/international-trade/dictionary/carrier/

global negotiator. (n.d.). *Dictionary of internal trade*. Retrieved from https://www.globalnegotiator.com/international-trade/dictionary/shipper/

GS1. (n.d.). *Welcome to gs1*. Retrieved from gs1: https://www.gs1.nl/en

He, H., & Lawry, J. (2014). The linguistic attribute hierarchy and its optimisation for classification. 1967-1984.

He, H., Watson, T., Maple, C., Mehnen, J., & Tiwari, A. (2017). A new semantic attribute deep learning with a linguistic attribute hierarchy for spam detection. 3862-3869.

Heerkens, H., & van Winden, A. (2012). *Geen Probleem.* Buren: Business School Nederland.

Himanen, V., Lee-Gosselin, M., & Perrels, A. (2007). Transport Policy. In V. Himanen, M. Lee-Gosselin, & A. Perrels, *Building blocks for sustainable transport* (pp. 40-42). Emerald Group Publishing.

Hohpe, G., & Woolf, B. (2003). *Message bus*. Retrieved from enterprise integration patterns: https://www.enterpriseintegrationpatterns.com/patterns/messaging/MessageBus.html

Jensen, F. V., & Nielsen, T. D. (2007). Graphical Languages for Specification of Decision Problems. In F. V. Jensen, & T. D. Nielsen, *Bayesian Networks and Decision Graphs* (pp. 279-342). Springer.

lawry, J. (2008). Label semantics as a framework for granular modelling. 87-102.

LucidChart. (n.a.). *what is a decision tree diagram*. Retrieved from https://www.lucidchart.com/pages/decision-tree?a=1

McCulloh, D., Lawry, J., & Cluckie, I. (2008). Real-time flood forecasting using updateable linguistic decision trees. 1935-1942.

Mendix. (n.d.). *Why mendix is different*. Retrieved from https://www.mendix.com/

Mische, M. (2001). Defining systems integration. In J. Myerson, *Enterprise Systems Integration* (pp. 4-8). Taylor & Francis Ltd.

Numann, J. (2017, September 7). *Vrachtwagen volgeladen op weg*. Retrieved March 15, 2018, from https://www.telegraaf.nl/nieuws/336178/vrachtwagen-volgeladen-op-weg

Object Management Group. (n.d.). *Business Process Model and Notation*. Retrieved from http://www.bpmn.org/

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 45-77.

Qin, Z., & Lawry, J. (2008). Knowledge discovery in a framework for modelling with words. 241-276.

Qin, Z., & Lawry, J. (2011). Prediction and query evaluation using linguistic decision trees. 3916-3928.

Qin, Z., & Wan, T. (2013). Hybrid Bayesian estimation tree learning with discrete and fuzzy labels. 852-863.

Rouse, M. (2005). *messaging*. Retrieved from techtarget: https://whatis.techtarget.com/definition/messaging

Slack, N., Brandon-Jones, A., & Johnston, R. (2013). Operations performance. In *Operations Management* (pp. 46-61). Pearson.

Topsector logistiek. (2018). *cross chain control centers*. Retrieved from https://topsectorlogistiek.nl/cross-chain-control-centers/

Turnbull, O., Lawry, J., Lowenberg, M., & Richards, A. (2016). A cloned linguistic decision tree controller for real-time path planning in hostile environments. 1-29.

Winston, W. L. (2003). CPM and PERT. In W. L. Winston, *Operations Research* (pp. 431-450). Cengage Learning.

Appendices

Appendix 1: BPMN CargoMatcher platform

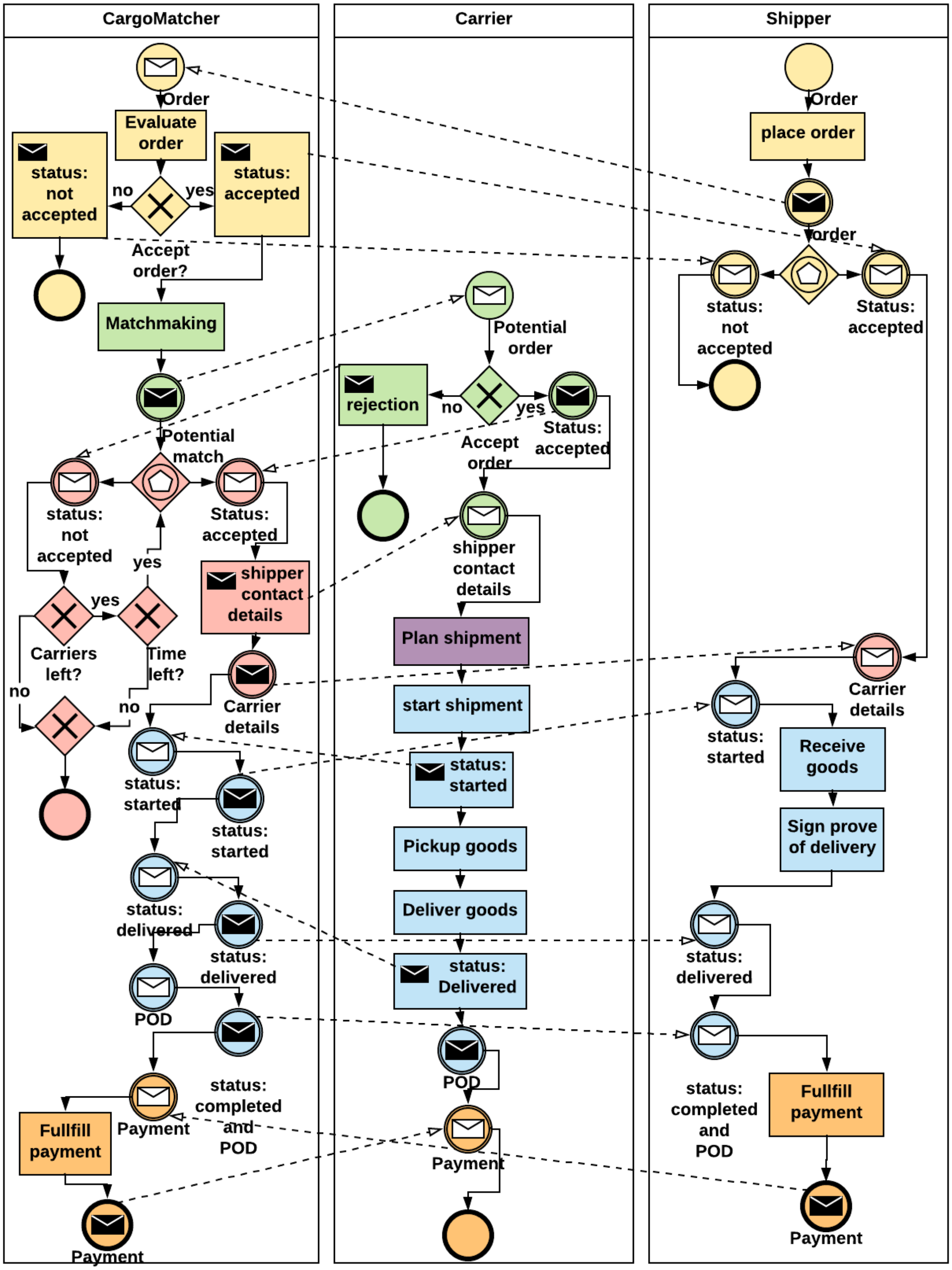

Figure 16 BPMN CargoMatcher platform

Appendix 2: Systematic literature review

Appendix 2A: Search strings

Table 6 Search strings systematic literature review

Search String	Scope	Date of search	Date range	Number of entries
Search protocol for Scopus				
"linguistic decision trees"	ALL	17-5-2018	ALL	45
Total found articles				**45**
Total articles before reading				**45**
Reading abstract and scanning of articles				-19
First reading of the articles				-7
Final reading of the articles				-11
Total articles after first scan				**8**

Appendix 2B: Concept matrix Linguistic decision tree

Table 7 Concept matrix linguistic decision tree

Concept 1: Linguistic decision trees	
paper 3	Linguistic Decision Trees A linguistic decision tree is a decision tree with attributes as nodes and linguistic descriptions of attributes as branches. Also associated with each branch, there is a mass function over the output focal sets.
paper 5	Linguistic decision tree (LDT) is a tree-structured model based on a framework for "Modelling with Words". Linguistic decision trees Linguistic decision tree is a transparent classification model based on label semantics. A linguistic decision tree is consisted by a set of branches with associated class probabilities.
paper 7	Linguistic decision tree Tree induction learning models have received a great deal of attention over recent years in the fields of machine learning and data mining because of their simplicity and effectiveness.

Appendix 2C: Concept matrix Label semantics

Table 8 Concept matrix label semantics

Concept 2: label semantics	
paper 1	Label semantics provides two fundamental and interrelated measures on labels to describe an object in an underlying domain. The measure of appropriateness and mass assignments on labels. A mass assignment on sets of labels x, quantifies our belief that any special subset of labels contains all and only the labels, with which

	it is appropriate to describe x.
paper 2	Label Semantics: Label Semantics uses a finite set of successive labels L to describe an object in an underlying domain Ω. Appropriateness measure and mass assignment are two fundamental and interrelated measures in Label Semantics.
paper 3	Label semantics is based on an epistemic theory of vagueness according to which the individual agents involved in communication believe in the existence of language conventions shared across the population of communicators. On the other hand, perhaps you have some ordered preferences between labels so that tall is more appropriate than medium which is in turn more appropriate than short. Your choice of words to describe the robber should surely then be based on these judgements about the appropriateness of labels. Furthermore, it is suggested that the vagueness of these description labels lies fundamentally in the uncertainty about when they are appropriate as governed by the rules and conventions of language use. The underlying assumption here is that some things can be correctly asserted while others cannot. Exactly where the dividing line lies between those labels that are and those that are not appropriate to use may be uncertain, but the assumption that such a division exists would be a natural precursor to any decision-making process of the kind just described. The ranking of available labels would seem to be an intuitive first step for an agent to take when faced with the decision problem about what to assert. Also, the direct allocation of probabilities to a range of complex compound expressions so that the values are internally consistent is a fundamentally difficult task. Hence, restricting such evaluations to only the basic labels would have significant practical advantages in terms of computational complexity.
paper 4	Label Semantics proposed by Lawry is a framework for modeling with linguistic expressions, or labels such as small, medium, large. Such labels are defined by overlapping fuzzy sets which cover the universe of continuous variables.
paper 5	Label Semantics is a random set based framework for modelling with linguistic expressions based on labels such as small, large, short, tall, young, old and so on. The fundamental question posed by label semantics is how to use linguistic expressions to label numerical values. The basic idea is that when individuals make assertions, such as ‘John is tall’, they are essentially providing the information that the label tall is appropriate for describing John’s height.
paper 6	Vague or imprecise concepts are fundamental to natural language. Human beings are constantly using imprecise language to communicate each other.

	We usually say \`Peter is tall and strong' but not \`Peter is exactly 1.85 meters in height and he can lift 100kg weights'. We may notice that labels are used in natural language to describe what we see, hear and feel. Such labels may have different degrees of vagueness (i.e., when we say Peter is young and he is male, the label young is more vague than the label male because people may have more widely different opinions on being young than being male. For a particular concept, there could be more than one label that is appropriate for describing this concept, and some labels could be more appropriate than others. Here, we will use a random set frame-work to interpret these facts.
paper 7	Label semantics is a random set-based framework for ‘‘Computing with Words’’ that captures the idea of computation on linguistic terms rather than numerical quantities.

Appendix 2D: Concept matrix Hierarchies of linguistic decision trees

Table 9 Concept matrix hierarchies of linguistic decision trees

Concept 3: Hierarchies of linguistic decision trees	
paper 1	In a linguistic attribute hierarchy (LAH), functional mappings between parent and child attribute nodes can be defined in terms of weighted linguistic rules that explicitly model both the uncertainty and vagueness, which often indicates our knowledge of such aggregation functions. It is not easy to define the labels for intermediate attributes in terms of their children, as the introduced intermediate attributes are not directly related to basic attributes in the system When a hierarchy is comprised of LDTs, the information is propagated through the LDTs from the lowest level of information sources (e.g. sensors). Therefore, the rules can be defined as conditional expressions in the label semantics framework weighted by conditional probabilities.
paper 2	Each successive layer uses the output from the previous layer as input, forming a hierarchy from low-level to high-level features. Assume each attribute is described with a set of labels, and subsequent label expressions are derived. In a linguistic attribute hierarchy (LAH), functional mappings between parent and child attribute nodes can be defined in terms of weighted linguistic rules that explicitly model both the uncertainty and vagueness, which often indicates our knowledge of such aggregation functions.
paper 3	In linguistic attribute hierarchies the functional relationship between child and parent nodes are not defined precisely. Instead the labels for a parent attribute are defined in terms of the labels describing the attributes corresponding to its child nodes, by means of a linguistic decision tree. To illustrate this idea consider the following simple linguistic attribute hierarchy as shown in figure 5.

Appendix 2E: Concept matrix algorithm for generating linguistic decision trees

Table 10 Concept matrix algorithm for generating linguistic decision trees

Concept 4: Algorithm for generating linguistic decision trees	
paper 2	In LID3, the most informative attribute is selected as the root of an LDT, and the tree will be expanded with branches associated with all possible focal elements of this attribute. For each branch, the attribute with maximal information gain in all free attributes will be the next node until the branch reaches the specified maximum length or the maximum class probability reaches the given threshold. The learning process forms a level order traversal based on the training data.
paper 8	rules generated by an LDT remain meaningful and useful to experts. To summarise, LID3 has the following advantages: 1.It offers an inherent model of uncertainty 2.The model offers good performance, robustness and generalisation 3.The branches of the tree can be interpreted as a set of linguistic rules based on label semantics, thereby improving the interpretability of the tree 4.The linguistic structure of the tree enables linguistic queries or explanations and information fusion.

Appendix 2F: Systematic Literature Review Protocol

Step 1: Scopus search decision trees

Result: over 200.000 papers

Step 2: Research on types of decision trees

Result: different types of decision trees

Step 3: select a type of decision tree

Result: Linguistic decision tree

Step 4: Scopus search linguistic decision tree

Result: 45 papers

Search String	Scope	Date of search	Date range	Number of entries
Search protocol for Scopus				
"linguistic decision trees"	ALL	17-5-2018	ALL	45
Total found articles				**45**
Total articles before reading				**45**
Reading abstract and scanning of articles				-19
First reading of the articles				-7
Final reading of the articles				-11
Total articles after first scan				**8**

Figure 17 Search strings systematic literature review

Step 5: scan articles and define concepts

Result: 4 concepts

- Linguistic decision trees
- Label semantics
- Hierarchies of linguistic decision trees
- Algorithm for generating linguistic decision trees

Step 6: Read abstracts and check on accessibility

Result: -19 papers

Authors	Title	delete?	extra bron	reason
[No author name available]	Intelligent Automation and Computer Engineering	yes		article not available
Baldwin J.F., Lawry J.	A new approach to learning linguistic control rules	yes		article not available
Crockett K., Bandar Z., O'Shea J.	Fuzzification of discrete attributes from financial data in fuzzy classification trees	maybe		information about fuzzy decision trees
Dehghan S., Rahmani A.M.	A new intelligent link state routing algorithm based on Credit Base-CMAC neural network	yes		article is about routing algorithm and neural net
Diao R., Chao F., Peng T., Snooke N., Shen Q.	Feature selection inspired classifier ensemble reduction	yes		not about decision trees
Erechtchoukova M.G., Khaiter P.A.	The effect of data granularity on prediction of extreme hydrological events in highly urbanized watersheds: A supervised classification approach	yes		not about decision trees
Erechtchoukova M.G., Khaiter P.A., Saffarpour S.	Short-Term Predictions of Hydrological Events on an Urbanized Watershed Using Supervised Classification	maybe		little information about linguistic decision trees
Gasir F., Crockett K., Bandar Z.	Inducing fuzzy regression tree forests using artificial immune systems	yes		no access
He H., Lawry J.	The linguistic attribute hierarchy and its optimisation for classification	no		information about linguistic decision trees (LDT)
He H., Lawry J.	A linguistic CMAC vs. a linguistic decision tree for decision making	yes		DOI not found
He H., Lawry J.	A kind of cascade linguistic attribute hierarchies for the two-way information propagation and its optimisation	yes		DOI not found
He H., Lawry J.	A linguistic CMAC equivalent to a Linguistic Decision Tree for classification	maybe		more about CMAC
He H., Lawry J.	Optimal cascade linguistic attribute hierarchies for information propagation	yes		article not available
He H., Lawry J.	Linguistic attribute hierarchies for downwards propagation of information	no		information about linguistic decision trees (LDT)
He H., McGinnity T.M., Coleman S., Gardiner B.	Linguistic decision making for robot route learning	no		information about linguistic decision trees (LDT)
He H., Watson T., Maple C., Mehnen J., Tiwari A.	A new semantic attribute deep learning with a linguistic attribute hierarchy for spam detection	no		information about linguistic decision trees (LDT)
He H., Zhu Z., Mäkinen E.	Task-oriented distributed data fusion in autonomous wireless sensor networks	no		information about linguistic decision trees (LDT)
He H., Zhu Z., Mäkinen E.	Task-oriented distributed information fusion in wireless sensor networks	no		information about linguistic decision trees (LDT)
He H., Zhu Z., Tiwari A., Mills A.	A cascade of linguistic CMAC neural networks for decision making	no		information about linguistic decision trees (LDT)
Lawry J.	Label semantics as a framework for granular modelling	no		information about linguistic decision trees (LDT)
Lawry J.	Learning linguistic models from data	no		constructing LDT's
Lawry J., He H.	Linguistic decision trees for fusing tidal surge forecasting models	yes		DOI not found
Lawry J., He H.	Multi-attribute decision making based on label semantics	yes		no access
Lawry J., McCulloch D.R., Randon N.J., Cluckie I.D.	Artificial Intelligence Techniques for Real-Time Flood FORECASTING	yes		no access
Lawry J., Tang Y.	Granular knowledge representation and inference using labels and label expressions	no		information about linguistic decision trees (LDT)
Li Y., Chen H., Lv M., Li Y., Li Y.	Extracting semantic event information from distributed sensing devices using fuzzy sets	no		information about linguistic decision trees (LDT)
McCulloch D.R., Lawry J., Cluckie I.D.	Real-time flood forecasting using updateable linguistic decision trees	no		information about linguistic decision trees (LDT)
McCulloch D.R., Lawry J., Rico-Ramirez M.A., Cluckie I.D.	Classification of weather radar images using linguistic decision trees with conditional labelling	no		information about linguistic decision trees (LDT)
Qin Z., Lawry J.	Prediction and query evaluation using linguistic decision trees	no		ID-3 and information about linguistic decision tr
Qin Z., Lawry J.	Knowledge discovery in a framework for modelling with words	no		labelling and algorithms
Qin Z., Lawry J.	Fuzziness and performance: An empirical study with linguistic decision trees	no		clear definition of linguistic decision tree
Qin Z., Lawry J.	Hybrid Bayesian estimation trees based on label semantics	no		clear how labels are given (label semantics)
Qin Z., Lawry J.	Decision tree learning with fuzzy labels	no		fuzzy labels
Qin Z., Wan T.	Hybrid Bayesian estimation tree learning with discrete and fuzzy labels	no		discrete and fuzzy labels
Rico-Ramirez M.A., Cluckie I.D.	Classification of ground clutter and anomalous propagation using dual-polarization weather radar	maybe		bayes vs fuzzy classifiers

Figure 18 Initial papers systematic literature review

Step 7: Reading whole papers and check content for information about one of the four concepts

Result: - 7 papers

	author	Title	Delete?	Reason	article number
7	Crockett K., Bandar Z., O'Shea J.	Fuzzification of discrete attributes from financial data in fuzzy classification trees	no	generating fuzzy trees with discrete attributes	paper 1
8	He H., Lawry J.	The linguistic attribute hierarchy and its optimisation for classification	no	Linguistic decision tree clear information. A specific Linguistic ID-3 is given, label semantics. Next to that hierarchies of different LDT's	paper 3
9	He H., Lawry J.	A linguistic CMAC equivalent to a Linguistic Decision Tree for classification	no	label semantics and CMAC comparison to LDT's	paper 4
10	He H., Watson T., Maple C., Mehnen J., Tiwari A.	A new semantic attribute deep learning with a linguistic attribute hierarchy for spam detection	no	clear definition of LDT and LID-3 and Hierarchy of LDTs mentioned. However a lot of overlap with paper 3 and paper 6	paper 7
11	He H., Zhu Z., Mäkinen E.	Task-oriented distributed data fusion in autonomous wireless sensor networks	no	clear definition of LDT and LID-3 and Hierarchy of LDTs mentioned. However a lot of overlap with paper 3 and paper 6 and paper 7	paper 8
12	Lawry J.	Learning linguistic models from data	no	very good article	paper 12
13	Lawry J., Tang Y.	Granular knowledge representation and inference using labels and label expressions	no	good example about everyday use of language, and label semantics	paper 13
14	Qin Z., Lawry J.	Prediction and query evaluation using linguistic decision trees	no	good article about label semantics and modelling with words	paper 17
15	Qin Z., Lawry J.	Knowledge discovery in a framework for modelling with words	no	good also bayes mentioned	paper 18
16	Qin Z., Lawry J.	Fuzziness and performance: An empirical study with linguistic decision trees	no	label semantics ID-3 / LID -3 LDT	paper 19
17	Qin Z., Lawry J.	Decision tree learning with fuzzy labels	no	also data mining and data analysis given	paper 21
18	Qin Z., Wan T.	Hybrid Bayesian estimation tree learning with discrete and fuzzy labels	no	aslo discrete labels and fuzzy labels	paper 22
19	Royston S., Lawry J., Horsburgh K.	A linguistic decision tree approach to predicting storm surge	no	tree structure and rules	paper 24
20	Turnbull O., Lawry J., Lowenberg M., Richards A.	A cloned linguistic decision tree controller for real-time path planning in hostile environments	no	clear fuzzy trees and sets and descriptions of labelling	paper 27
21	Erechtchoukova M.G., Khaiter P.A., Saffarpour S.	Short-Term Predictions of Hydrological Events on an Urbanized Watershed Using Supervised Classification	yes	no specific information on the different concepts	paper 2
22	He H., Lawry J.	Linguistic attribute hierarchies for downwards propagation of information	yes	a lot of mathematic equations for a case study. Educative information is given shortly, but in paper 3 this same information is given in more detail	paper 5
23	He H., Zhu Z., Mäkinen E.	Task-oriented distributed information fusion in wireless sensor networks	yes	no new information compared to paper 8 and wireless sensor networks is not the research	paper 9
24	He H., Zhu Z., Tiwari A., Mills A.	A cascade of linguistic CMAC neural networks for decision making	yes	only about the CMAC part. A good comparison is made in paper 4. Therefore this paper is not needed.	paper 10
25	Li Y., Chen H., Lv M., Li Y., Li Y.	Extracting semantic event information from distributed sensing devices using fuzzy sets	yes	paper is more about deriving label semantics and LDT's from sensor data. Which is not the scope for the research	paper 14
26	Rico-Ramirez M.A., Cluckie I.D.	Classification of ground clutter and anomalous propagation using dual-polarization weather radar	yes	only data analysis on weather forecasting	paper 23

Figure 19 Papers for reading whole content

Step 8: Define papers to pick information from for concept matrices
Result: -11 papers

author	Title	Delete?	Reason	article number
Crockett K., Bandar Z., O'Shea J.	Fuzzification of discrete attributes from financial data in fuzzy classification trees	yes	Not applicable for concept matrix	paper 1
He H., Lawry J.	The linguistic attribute hierarchy and its optimisation for classification	no	Linguistic decision tree clear information. A specific Linguistic ID-3 is given, label semantics. Next to that hierarchies of different LDT's	paper 3
He H., Lawry J.	A linguistic CMAC equivalent to a Linguistic Decision Tree for classification	yes	label semantics information is the same as in paper 7 and not as clear as in paper 7. Next to that paper is about CMAC which is not the scope of this literature review. So paper can be deleted	paper 4
He H., McGinnity T.M., Coleman S., Gardiner B.	Linguistic decision making for robot route learning	yes	Section 2 is informative, but comparable to paper 3 after clear comparison of section 2 it is exact the same information and text as paper 3, and the rest of the paper is about a robotic route learning which is not the scope of this literature review. so this paper can be deleted.	paper 6
He H., Watson T., Maple C., Mehnen J., Tiwari A.	A new semantic attribute deep learning with a linguistic attribute hierarchy for spam detection	no	clear definition of LDT and LID-3 and Hierarchy of LDTs mentioned. However a lot of overlap with paper 3 and paper 6	paper 7
He H., Zhu Z., Mäkinen E.	Task-oriented distributed data fusion in autonomous wireless sensor networks	yes	clear definition of LDT and LID-3 and Hierarchy of LDTs mentioned, However it is the same as paper 3 and paper 6 and paper 7 and the rest of the paper is not in the scope of the literature review therefore it can be deleted.	paper 8
Lawry J.	Label semantics as a framework for granular modelling	no	mass relations and ldt's as types of granular models	paper 11
Lawry J.	Learning linguistic models from data	no	very good article	paper 12
Lawry J., Tang Y.	Granular knowledge representation and inference using labels and label expressions	yes	label semantics is same as in paper 3 , 7 6, and 8	paper 13
McCulloch D.R., Lawry J., Cluckie I.D.	Real-time flood forecasting using updateable linguistic decision trees	no	LID-3 and lable semantics	paper 15
McCulloch D.R., Lawry J., Rico-Ramirez M.A., Cluckie I.D.	Classification of weather radar images using linguistic decision trees with conditional labelling	yes	no new information and use casae on radar images	paper 16
Qin Z., Lawry J.	Prediction and query evaluation using linguistic decision trees	no	good article about label semantics and modelling with words	paper 17
Qin Z., Lawry J.	Knowledge discovery in a framework for modelling with words	no	good also bayes mentioned	paper 18
Qin Z., Lawry J.	Fuzziness and performance: An empirical study with linguistic decision trees	yes	no new information and some experiments which are not the literature reviews scope	paper 19
Qin Z., Lawry J.	Hybrid Bayesian estimation trees based on label semantics	yes	no new information or not the literature reviews scope	paper 20
Qin Z., Lawry J.	Decision tree learning with fuzzy labels	no	also data mining and data analysis given	paper 21
Qin Z., Wan T.	Hybrid Bayesian estimation tree learning with discrete and fuzzy labels	no	aslo discrete labels and fuzzy labels	paper 22
Royston S., Lawry J., Horsburgh K.	A linguistic decision tree approach to predicting storm surge	yes	tree structure and rules	paper 24
Turnbull O., Lawry J., Lowenberg M., Richards A.	A cloned linguistic decision tree controller for real-time path planning in hostile environments	no	clear fuzzy trees and sets and descriptions of labelling	paper 27

Figure 20 Last reading and concept matrices

Step 9: complete concept matrices

Result: see Appendix 8B t/m 8E.

Step 10: write literature review

Result: see systematic literature review in stage 3 decision models

Appendix 3: BPMN models of current process

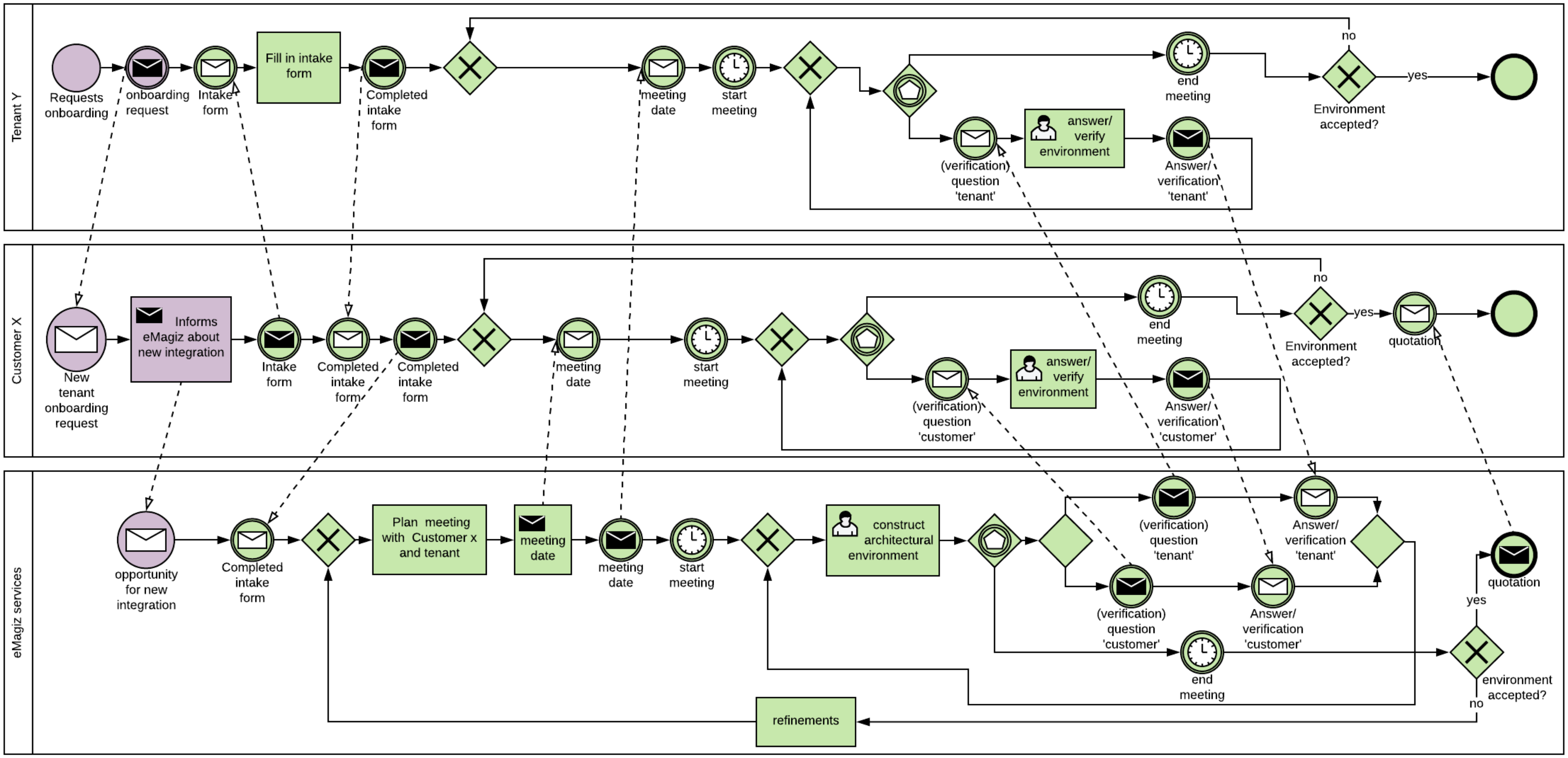

Figure 21 BPMN Current onboarding process – presales and intake

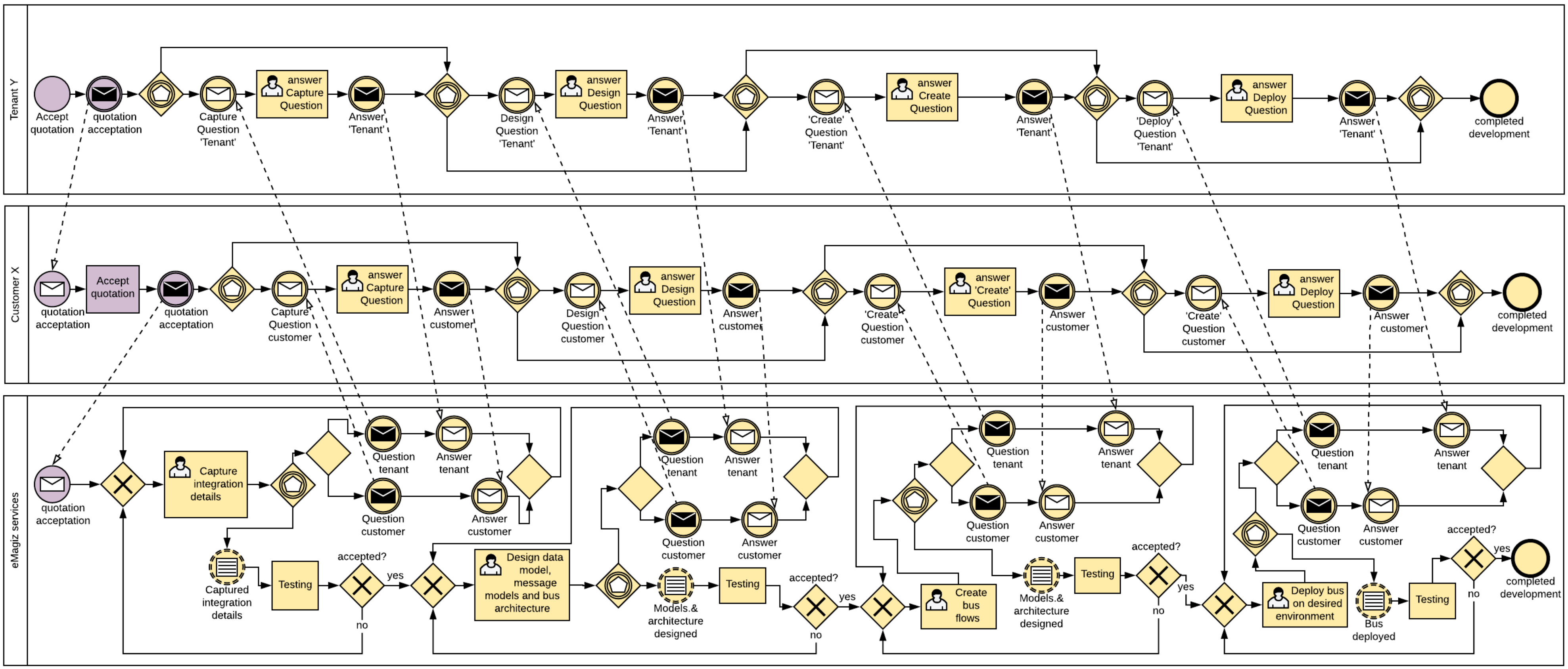

Figure 22 BPMN Current onboarding process - sales and development

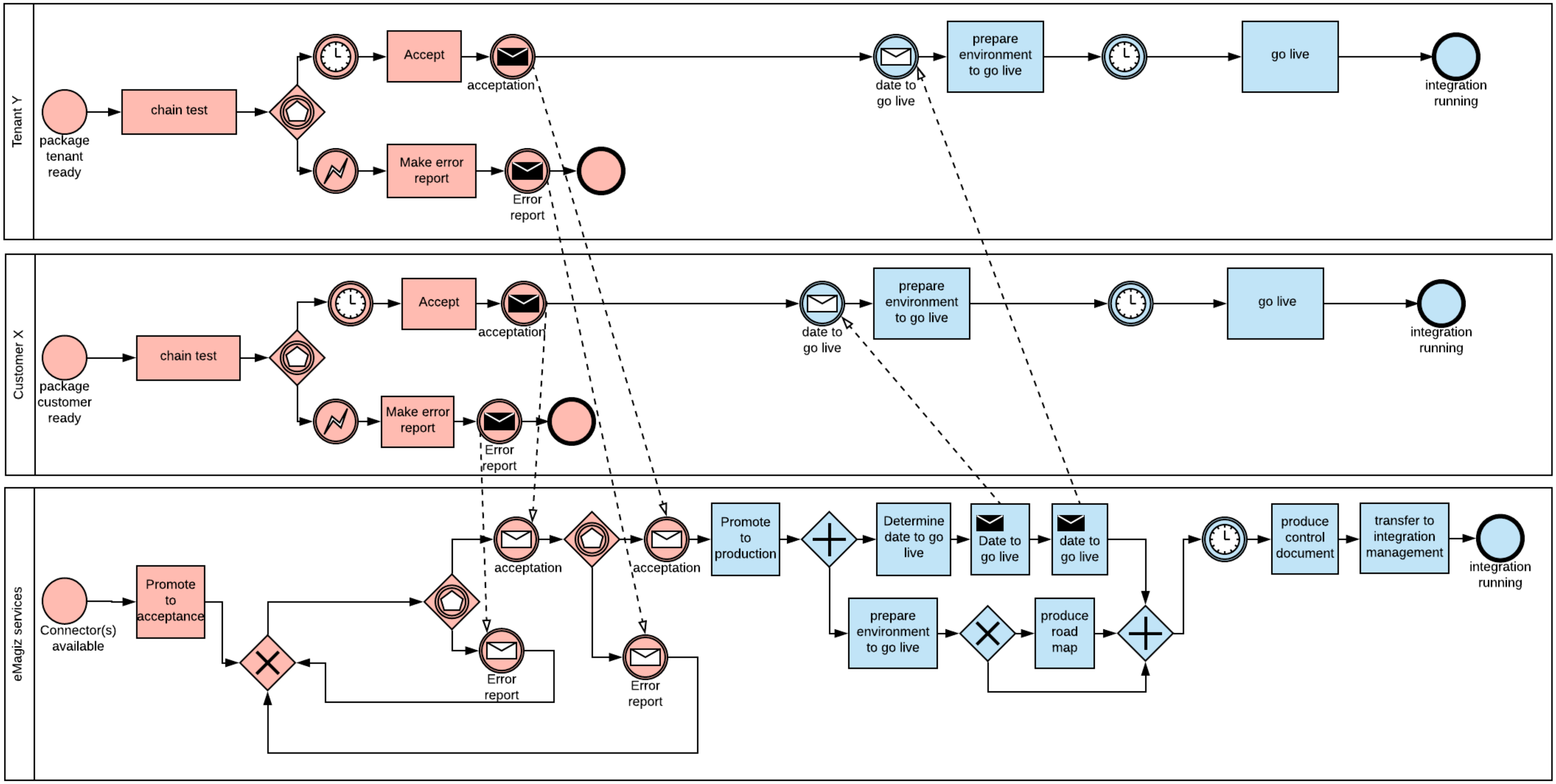

Figure 23 BPMN Current onboarding process - acceptance and going live

Appendix 4: PERT current situation

Appendix 4A: Extracting the PERT process from the BPMN models

In this appendix it is shown how each action, that is analysed in the PERT analysis of the current situation, is extracted from the BPMN models of the current situation (Appendix 3) and what the time span per action is.

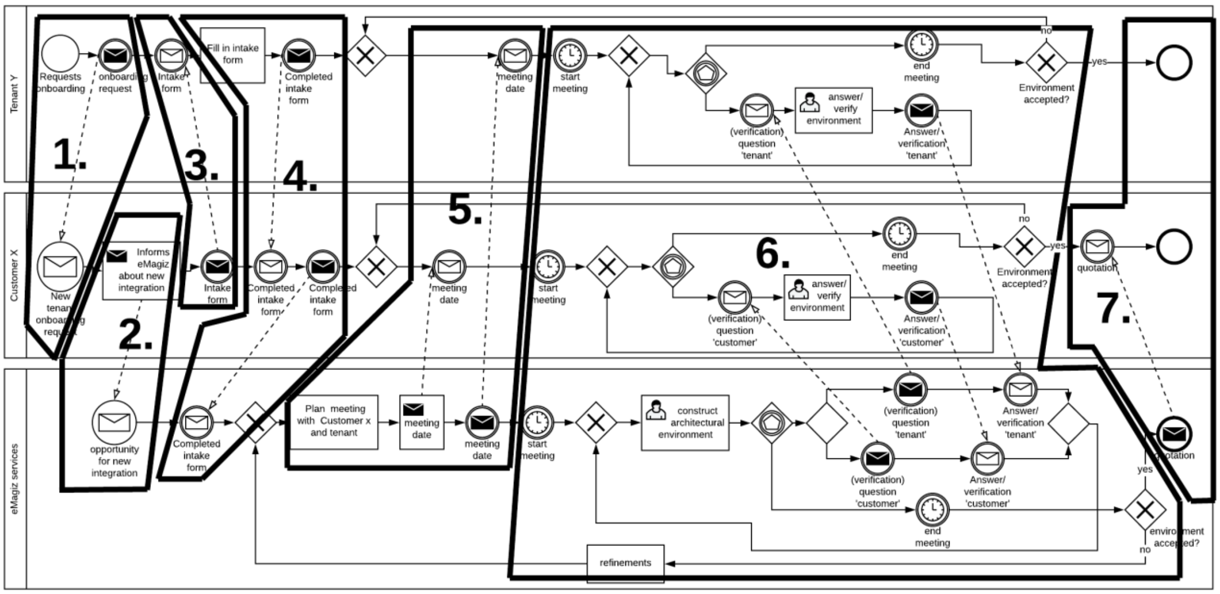

Figure 24 PERT steps current situation - 1

Action 1: Requesting a system onboarding by the tenant. This is the sending of a message from the tenant to Customer.
Time span: this is the starting point

Action 2: Informing eMagiz services about the request. Sending a message from the customer to eMagiz services.
Time span: from receiving an onboarding request up to and including sending the message.

Action 3: Sending the intake form to the tenant.
Time span: from receiving an onboarding request up to and including sending the intake form.

Action 4: Completing the intake form and sending it to eMagiz services. It is assumed that the tenant completes the intake form and sends it to eMagiz right afterwards. Therefore, this is seen as one action.
Time span: from receiving the intake form and up to sending the completed intake form.

Action 5: The planning of a meeting with the tenant and the customer.
Time span: from receiving the completed intake form up to and including setting the meeting date.

Action 6: This includes a lot of steps and loops in the BPMN model. Therefore, in practice the number of steps differ per meeting. Nevertheless, this are steps that The executing of a meeting with the tenant and the customer. are executed during the meeting. While the meeting has a maximum length of one day and is completed after the steps are completed.
Time span: from the moment the meeting date is set up to and including the meeting.

Action 7: The sending of a quotation to the customer.
Time span: from the end of the meeting up to and including sending the quotation.

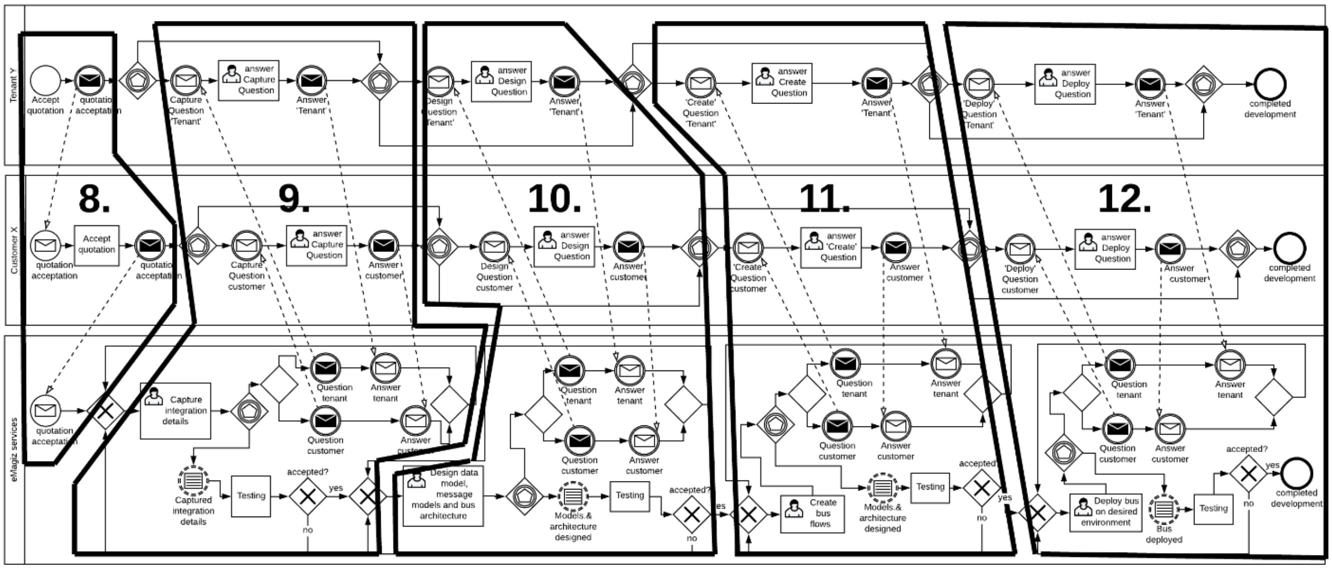

Figure 25 PERT steps current situation - 2

Action 8: accepting the quotation and informing eMagiz services about it.
Time span: from the moment the quotation is received up to and including the sending acceptation to eMagiz services.

Action 9: Completing the Capture phase. This is the first phase of the integration Lifecyle. In this phase a lot of information is needed from both customer and tenant.
Time span: from the moment the quotation is accepted up to and including the completion of the Capture phase.

Action 10: Completing the Design phase. This is the second phase of the integration Lifecyle. In this phase a lot of information is needed from both customer and tenant.
Time span: from the moment the Capture phase is completed up to and including the completion of the Design phase.

Action 11: Completing the Create phase. This is the third phase of the integration Lifecyle. In this phase a lot of information is needed from both customer and tenant.
Time span: from the moment the Design phase is completed up to and including the completion of the Create phase.

Action 12: Completing the Deploy phase. This is the fourth phase of the integration Lifecyle. In this phase a lot of information is needed from both customer and tenant.
Time span: from the moment the Create phase is completed up to and including the completion of the Deploy phase.

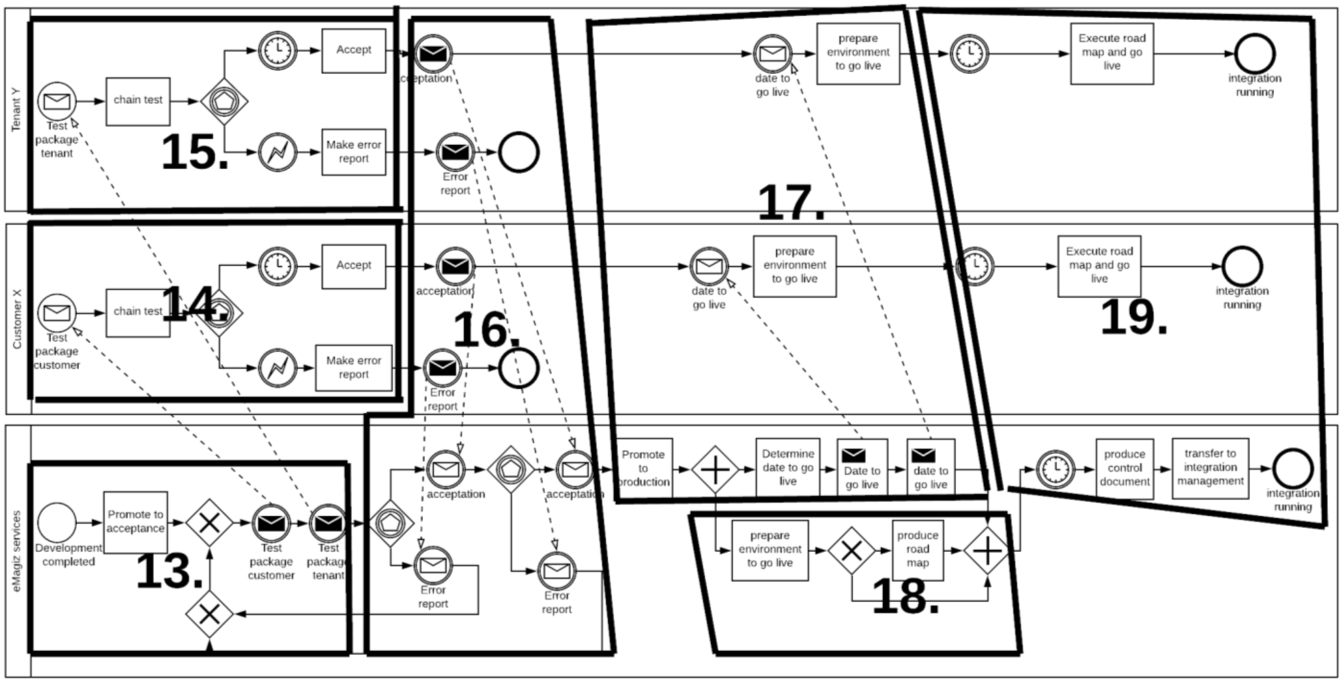

Figure 26 PERT steps current situation - 3

Action 13: giving a test package to both the tenant and the customer.
Time span: The moment from completing the four phases of the integration Lifecycle up to and including the sending of a test package to both the customer and the tenant.

Action 14: chain testing by the customer.
Time span: the moment from receiving the test package up to and including the execution of a chain test.

Action 15: chain testing by the tenant.
Time span: the moment from receiving the test package up to and including the execution of a chain test.

Action 16: Inform eMagiz services that the test package is accepted.
Time span: the moment from when the chain test is executed up to and including sending eMagiz services an acceptation message

Action 17: Plan a date to go live.
Time span: the moment from receiving an acceptation message up to and including the moment a date is set to go live.

Action 18: Prepare to go live.
Time span: the moment from receiving an acceptation message up to and including the moment the preparations are completed.

Action 19: Go live.
Time span: the moment from when both a date is set and the environment is prepared up to and including the integration is live.

Appendix 4B: Time estimates per action of the PERT process

The estimates made per action are estimated based on working days of 8 hours, where one week consists of 5 working days. A time estimates differ from 0,125 days (an hour) up to and including 25 days (5 weeks). This time range is determined based on information given by Cape consultants and CargoMatcher. In this appendix for each action a worst, best, and most likely time estimate is made of the throughput time for that action. Action 1 is the starting point therefore each time estimate is equal to zero.

In the process there are actions which look like one another. For these actions the estimated times are considered equal. First an ID is given for each type of action. Second, the characteristics of the type of action are given. Third, the three time estimates (worst, best, and most likely) are given and motivated.

ID: C1
Characteristics: The action that needs to be executed is the sending of a message by eMagiz services or CargoMatcher. The messages that are send are standard messages, such as an already constructed form or sending information that is known. In other words, it does not require any further actions to collect this information.

Best time estimate: 0,125 days (an hour)
argument: a system onboarding is a high priority for both eMagiz services and CargoMatcher. Both parties have employees which daily task is to handle questions and requests from their customers. Therefore, in the best case they will respond within an hour.

Most likely time estimate: 0,5 days
Argument: Answering questions and handling requests is a high priority. Therefore, it is likely that they will respond somewhere the same day (0,5 days).

Worst time estimate: 1 day
Argument: In the worst case it was not possible to respond the same day. Therefore, in the worst case it takes one working day to respond. This is assumed as the worst case, since multiple employees are responsible for this type of action.

ID: C2
Characteristics: This involves actions, during the development of the system integration, where information is requested by eMagiz services from either CargoMatcher or the tenant. The requested information is critical and necessary for completing the development step.

The estimated times for this type of action is based on one contact moment where information is needed. Nevertheless, this is not the amount of contact moments that is estimated for each action involved. Therefore, the estimated times for these actions are multiplied by the estimated amount of contact moments.

Best time estimate: 0,125 days (one hour)
Argument: The involved party that is asked a question receives the question and acts immediately to get the information and to send the information. Here the necessary information is accessible, and the involved employee knows where to get the information. The involved party has a high level of involvement.

Most likely: 1 day
Argument: the employee that receives the question could not act immediately, because collecting the information takes time. For example, the employee has to contact their system provider for information.

worst time estimate: 5 days
Argument: the employee that receives the question has little time or has to wait for information. Here the involvement level is low.

ID: C3
Characteristics: This involves an action where a physical appointment needs to be made by eMagiz services with both the parties.

Best time estimate: 0,5 days
Argument: within the same day as eMagiz services receives the necessary information an appointment is made. Next to that, both parties responded within the same day.

Most expected time estimate: 3 days
Argument: probably an agenda invitation is sent to both parties. Before there is found a suitable date and for all the three involved parties there will be sent multiple email messages which will take a few days before a suitable date is found.

Worst expected time estimate: 10 days
Argument: one of the involved parties does not respond for a long time, or it is almost impossible to find a suitable date.

ID: C4
Characteristics: This involves an action which consists of having a physical appointment.

Best time estimate: 5 days
Argument: If having the meeting is a high priority for the three parties. Then in the best case they all have time for a meeting one week later.

Most likely time estimate: 10 days
Argument: it is expected that it will take about 2 weeks to find time in the agendas to have the appointment.

Worst time estimate: 25 days
Argument: it is expected if it is not possible to find time in their agenda they have to plan it more as a month later since it is assumed that the agenda is more empty one month from now as one week from now.

These consultants gave for example the information that they aim to respond to an email message within an hour and that making an appointment with customers could take over a month (5 weeks).

ID: C5
Characteristics: this involves the action of conducting a chain test.

Best time estimate: 1 day
argument: the integration is a very high priority; therefore, the chain test is done immediately with all a lot of special test scenarios. Everything works as it should.
Most likely time estimate: 5 days
There are some small failures which have to be resolved, or it takes a few days to execute all the testing.
Worst time estimate: 10 days
There are some bigger issues which need to be resolved which needs to be resolved by eMagiz services thereafter a new chain test needs to be executed.

ID: C6
Characteristics: this involves actions that need to be executed by eMagiz services from without the eMagiz environment.

Best time estimate: 0.125 days
Argument: this action is executed right after the previous action, executing the action takes in the best case less then an hour.
Most likely estimate: 0.5 days
Argument: it is the most likely that this task is executed somewhere within the same day as the previous action.
Worst estimate: 5 days
Argument: There are other higher priorities or there are other issues but it will nevertheless be executed at a maximum of 1 week.

Action 1: Requesting a system onboarding by the tenant.
Best time estimate: 0
Argument: starting point
Most likely time estimate: 0
Argument: starting point
Worst time estimate: 0
Argument: starting point

Action 2: Informing eMagiz services about the request.

Best time estimate: 0,125 days
Argument: C1
Most likely time estimate: 0.5 days
Argument: C1
Worst time estimate: 1 day
Argument: C1

Action 3: Sending the intake form to the tenant.

Best time estimate: 0,125 days
Argument: C1
Most likely time estimate: 0.5 days

Argument: C1
Worst time estimate: 1 day
Argument: C1

Action 4: Completing the intake form and sending it to eMagiz services.

Best time estimate: 1 day
Argument: the intake form is received. Thereafter the questions are answered the same day. The next day eMagiz services has received the necessary information.
Most likely time estimate: 5 days
Argument: the information must be collected from multiple places. This takes time. It is estimated that a week later this information should be available.
Worst time estimate: 10 days
Argument: the information is not easily accessible therefore special actions to collect the information is necessary which is estimated to delay the action for a week.

Action 5: The planning of a meeting with the tenant and the customer.
Best time estimate: 0.5 days
Argument: C3
Most likely time estimate: 3 days
Argument: C3
Worst time estimate: 10 days
Argument: C3

Action 6: The executing of a meeting with the tenant and the customer.
Best time estimate: 5 days
Argument: C4
Most likely time estimate: 10 days
Argument: C4
Worst time estimate: 25 days
Argument: C4

Action 7: The sending of a quotation to the customer.
Best time estimate: 0.125 days
Argument: the quotation is made during the meeting and is send right after the meeting
Most likely time estimate: 1 days
Argument: after the meeting the consultant must construct the quotation and must discuss this with other employees before sending it.
Worst time estimate: 5 days
Argument: after the meeting the consultant must construct the quotation and must discuss this with other employees, but they are not accessible the same day or the day after.

Action 8: accepting the quotation and informing eMagiz services about it.
Best time estimate: 1 days
Argument: in the best case the company decides and inform eMagiz services within one day that they need the system integration and accept the contract conditions.
Most likely time estimate: 5 days

Argument: they have to discuss the quotation within the company and decide within a week that they need the integration
Worst time estimate: 25 days
Argument: at a maximum of 1 month it is expected to be the longest time a company will take to accept a quotation.

Action 9: Completing the Capture phase.
It is identified that the Capture phase has the most contact moments in comparison with the other three phases of the integration lifecycle. It is assumed that on average 5 contact moments are necessary.
Best time estimate: 0.625 days (5*C2)
Argument: C2
Most likely time estimate: 5 days (5*C2)
Argument: C2
Worst time estimate: 25 days (5*C2)
Argument: C2

Action 10: Completing the Design phase.
It is identified that the Design phase has more contact moments in comparison with the create and deploy phases of the integration lifecycle, but less as the Capture phase. It is assumed that on average 4 contact moments are necessary.
Best time estimate: 0.5 days (4*C2)
Argument: C2
Most likely time estimate: 4 days (4*C2)
Argument: C2
Worst time estimate: 20 days (4*C2)
Argument: C2

Action 11: Completing the Create phase.
It is identified that the Create phase has more contact moments in comparison with the deploy phase of the integration lifecycle, but less as the Capture and design phases. It is assumed that on average 3 contact moments are necessary.
Best time estimate: 0.375 days (3*C2)
Argument: C2
Most likely time estimate: 3 days (3*C2)
Argument: C2
Worst time estimate: 15 days (3*C2)
Argument: C2

Action 12: Completing the Deploy phase.
It is identified that the Deploy phase has the least amount of contact moments in comparison with the other three phases of the integration lifecycle, but nevertheless has contact moments. It is assumed that on average 2 contact moments are necessary
Best time estimate: 0.25 days (2*C2)
Argument: C2
Most likely time estimate: 2 days (2*C2)
Argument: C2

Worst time estimate: 10 days (2*C2)
Argument: C2

Action 13: giving a test package to both the tenant and the customer.
Best time estimate: 0.25 days
Argument: C6
Most likely time estimate: 0.5 days
Argument: C6
Worst time estimate: 5 days
Argument: C6

Action 14: chain testing by the customer.
Best time estimate: 1 days
Argument: C5
Most likely time estimate: 5 days
Argument: C5
Worst time estimate: 10 days
Argument: C5

Action 15: chain testing by the tenant.
Best time estimate: 1 days
Argument: C5
Most likely time estimate: 5 days
Argument: C5
Worst time estimate: 10 days
Argument: C5

Action 16: Inform eMagiz services that the test package is accepted.
Best time estimate: 0.125 days
Argument: both companies inform eMagiz services immediately after accepting the test package.
Most likely time estimate: 0.5 days
Argument: both companies will inform eMagiz services the same day as accepting the test package.
Worst time estimate: 5 days
Argument: They forget to inform eMagiz services immediately.

Action 17: Plan a date to go live.
Best time estimate: 0.5 days
Argument: C3
Most likely time estimate: 3 days
Argument: C3
Worst time estimate: 10 days
Argument: C3

Action 18: Prepare to go live.
Best time estimate: 0.25 days
Argument: C6
Most likely time estimate: 0.5 days

Argument: C6
Worst time estimate: 5 days
Argument: C6

Action 19: Go live.
Best time estimate: 5 days
Argument: C4
Most likely time estimate: 10 days
Argument: C4
Worst time estimate: 25 days
Argument: C4

Appendix 4C: PERT calculations current situation

Value	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19
best	0	0,125	0,125	1	0,5	5	0,125	1	0,625	0,5	0,375	0,25	0,25	1	1	0,125	0,5	0,25	5
most likely	0	0,5	0,5	5	3	10	1	5	5	4	3	2	0,5	5	5	0,5	3	0,5	10
worst	0	1	1	10	10	25	5	25	25	20	15	10	5	10	10	5	10	5	25
T	0,000	0,521	0,521	5,167	3,750	11,667	1,521	7,667	7,604	6,083	4,563	3,042	1,208	5,167	5,167	1,188	3,750	1,208	11,667
variance	0,000	0,021	0,021	2,250	2,507	11,111	0,660	16,000	16,504	10,563	5,941	2,641	0,627	2,250	2,250	0,660	2,507	0,627	11,111

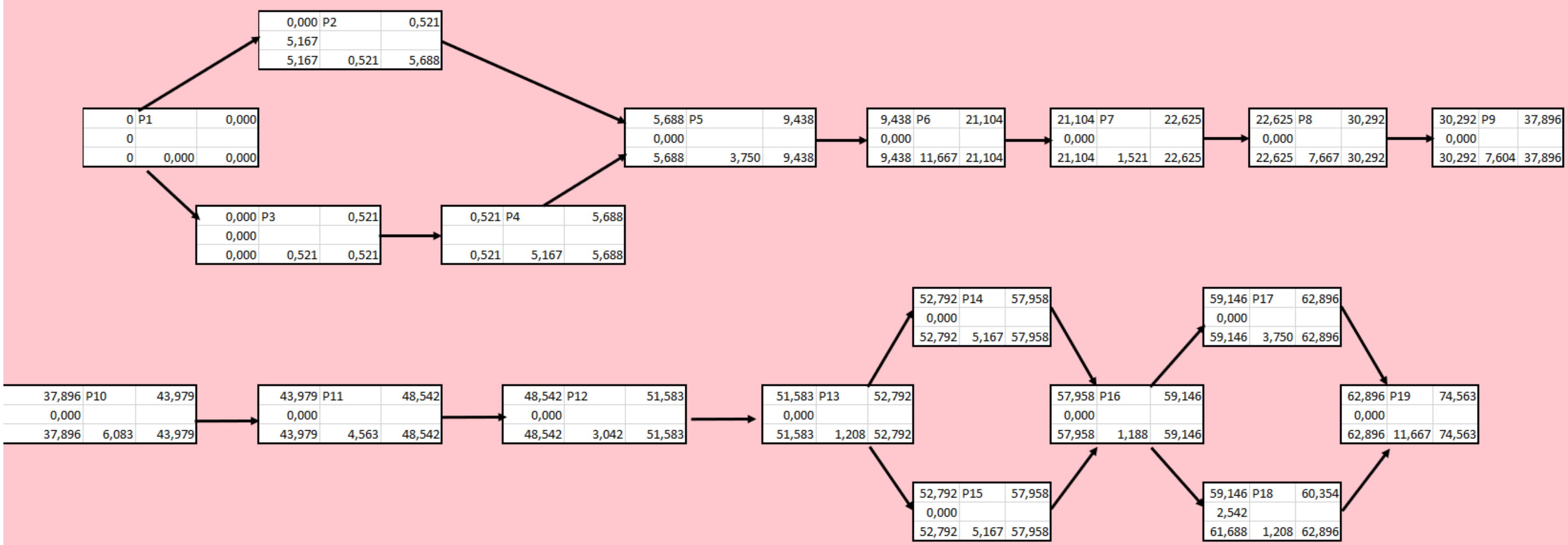

Figure 27 PERT Excel calculations current situation

Appendix 5: User stories and development tasks

Sprint 1					
As..		..I want to..		..So..	
As,	Customer	, I want to,	Select a standard system	, So,	I can make clear which standard system I use
- Add the possibility to filter on system types - Add the possibility to filter on software package providers - Make an overview of all the supported standard systems - Add the possibility to select one system (max) - Add the possibility to answer questions corresponding to the system					
As,	Developer	, I want to,	Ask questions to the customer at different steps of the intake wizard	, So,	I can get the necessary information from customers
- Add the possibility to add questions to each individual system - Add the possibility to ask standard question which are independent of the choice of system					
As,	Customer	, I want to,	have an overview of my onboarding request	, So,	I can check the correctness of the entered information
- Add an overview page for the overview of the onboarding request - Add a send and confirm button to confirm the onboarding request					
As,	Developer	, I want to,	receive the information for the configuration of connectors	, So,	I can fulfil an onboarding request
Sprint 2					
As..		..I want to..		..So..	
As,	Customer	, I want to,	Select my standard market process message	, So,	I can make clear for which type of market process message I want to have a connector
- Add the function to select a market segment - Add the function to select a process - Add the function to select process messages - Add extra information and instruction					
As,	Customer	, I want to,	receive status updates	, So,	I can keep track of my onboarding request
- Add the onboarding dashboard - Add statuses to the onboarding dashboard - Add progress bar to the onboarding dashboard					
As,	Customer	, I want to,	receive download package	, So,	I can install the onboarding package
- Add download page (page where request is fulfilled) - Add upload functionality - Add download functionality					
As,	Customer	, I want to,	have the possibility to let tenants enter information in the intake wizard	, So,	Tenants can answer questions themselves
- Set tenant pages (Set the pages and microflows that need to be visible for tenants)					

Sprint 3					
As..		..I want to..		..So..	
As,	Customer	, I want to,	manage my account and the tenant accounts	, So,	I can select and make changes for these accounts
- Add the possibility to register tenants - Add the possibility to change account details - Add the possibility to request to delete an account					
As,	Functional Admin	, I want to,	Edit the onboarding wizard	, So,	I can update and/or change the wizard
- Add the possibility to edit system providers - Add the possibility to edit systems - Add the possibility to edit processes - Add the possibility to edit process messages - Add the possibility to edit system questions - Add the possibility to edit standard questions - Add the possibility to edit market segments - Add the possibility to edit system updates - Add the possibility to edit all accounts					
As,	Customer Admin	, I want to,	Manage all customers	, So,	I can handle the customer support
- add the possibility to register customers - add the possibility to assign developers to customers					

Appendix 6: Screenshots of prototype

Appendix 6A: Home screen prototype

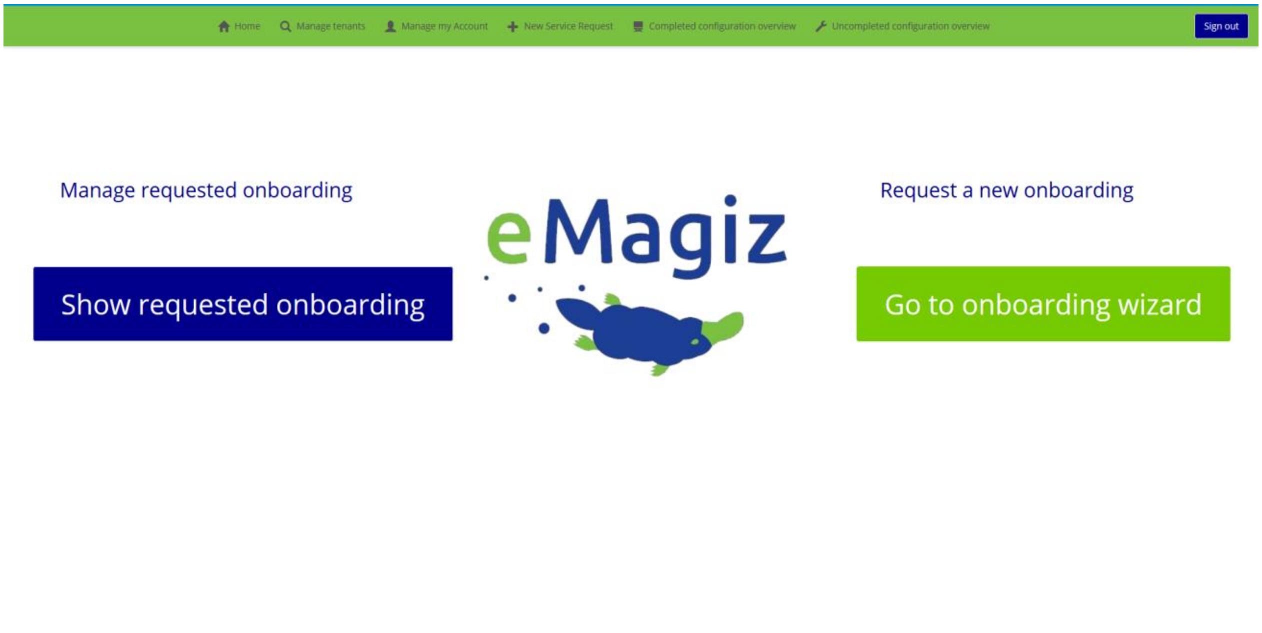

Figure 28 Prototype: home screen prototype

Appendix 6B: First screen tenant onboarding intake wizard

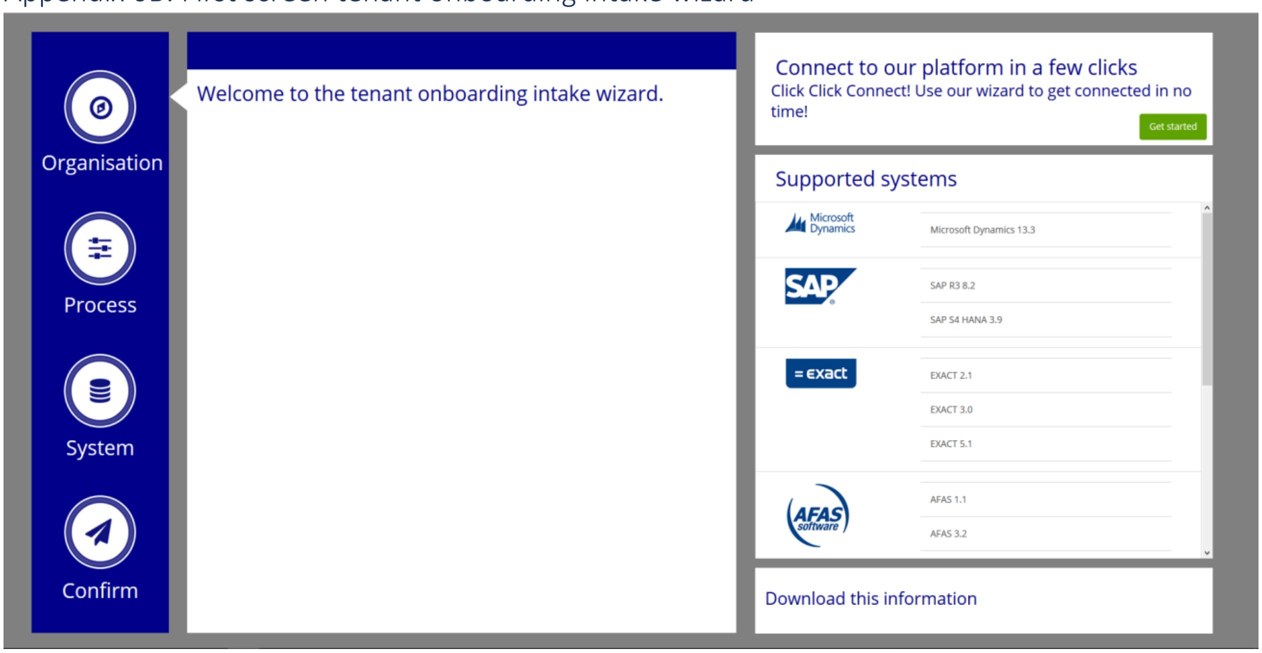

Figure 29 Prototype: first screen tenant onboarding intake wizard

Appendix 6C: Organisation screen 1 tenant onboarding intake wizard

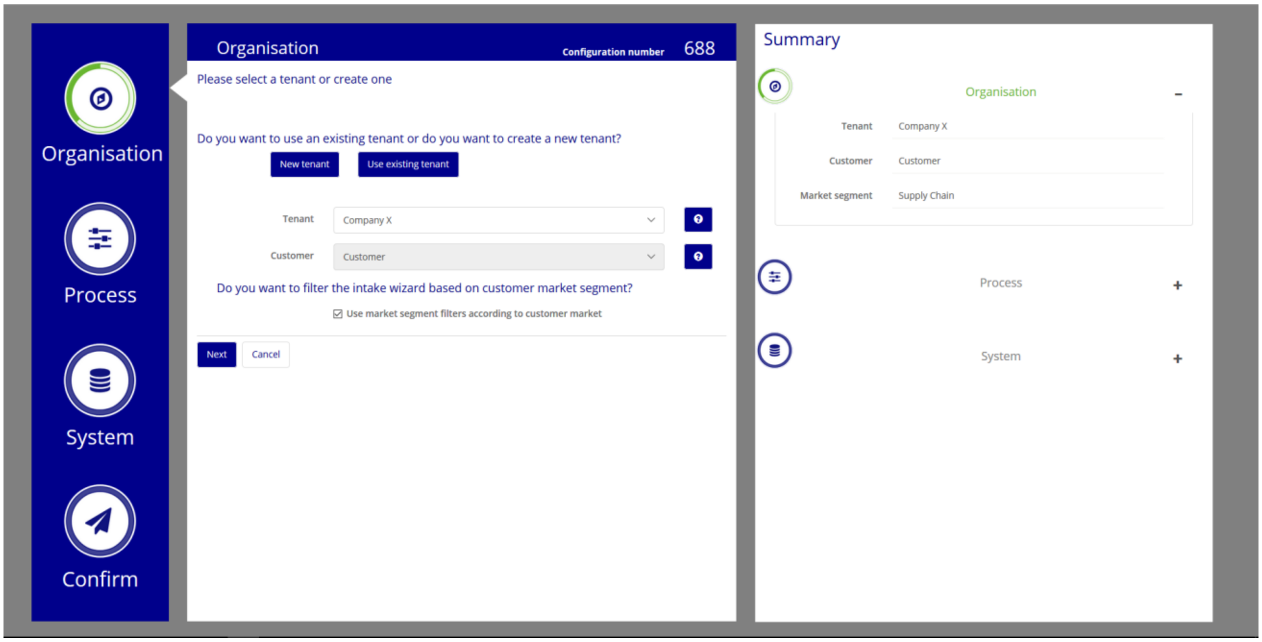

Figure 30 Prototype: organisation screen 1 onboarding intake wizard

Appendix 6D: Organisation screen 2 tenant onboarding intake wizard

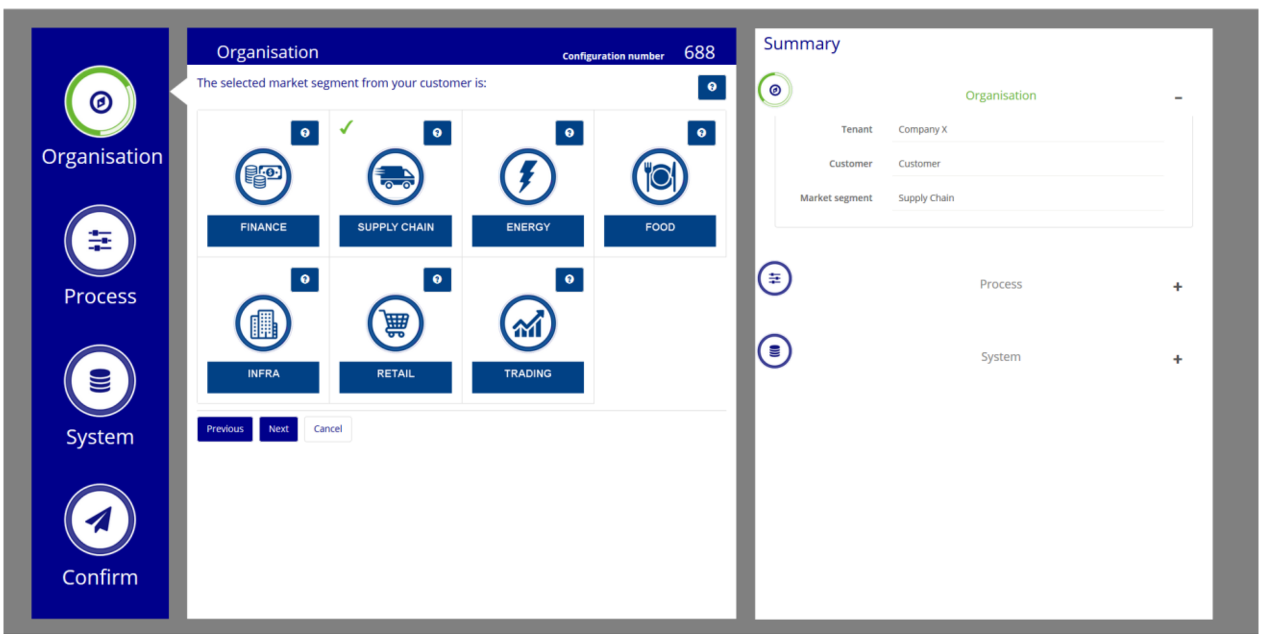

Figure 31 Prototype: organisation screen 2 tenant onboarding intake wizard

Appendix 6E: Process screen 1 tenant onboarding intake wizard

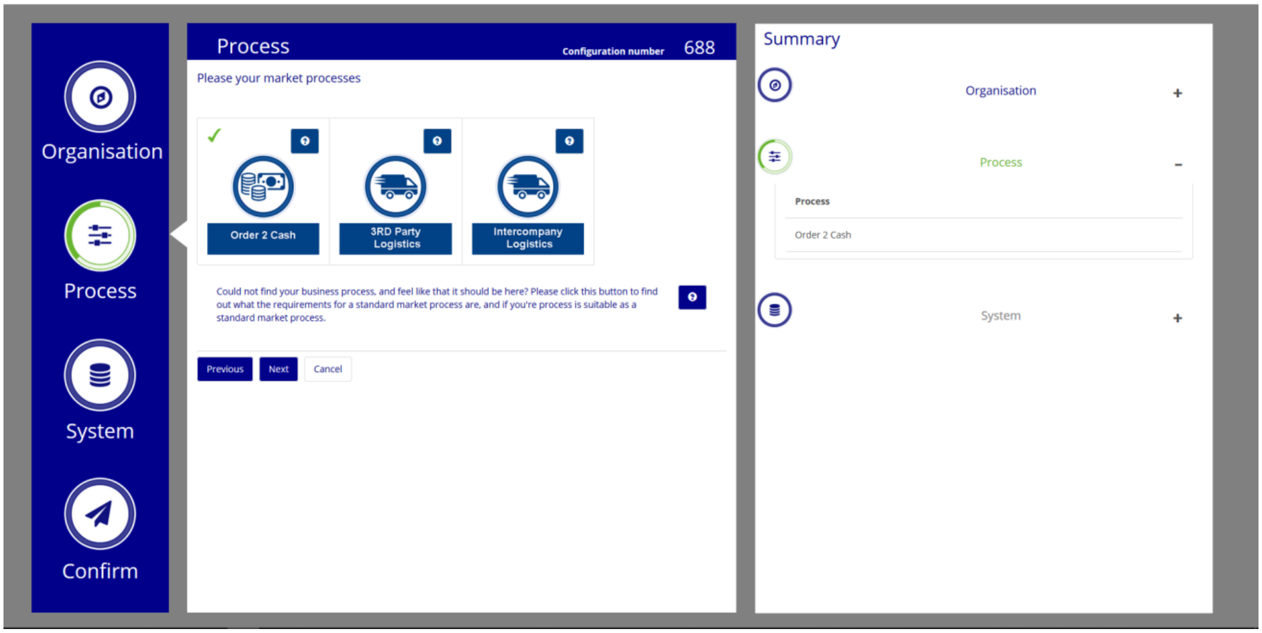

Figure 32 Prototype: process screen 1 tenant onboarding intake wizard

Appendix 6F: Process screen 2 tenant onboarding intake wizard

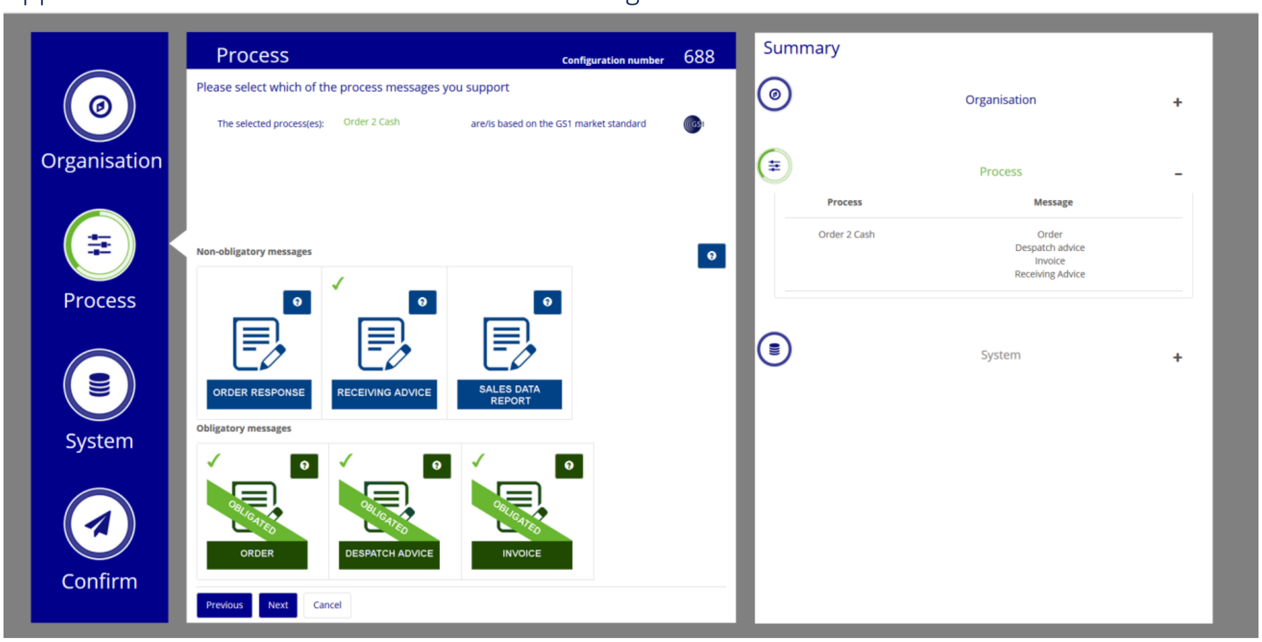

Figure 33 Prototype: process screen 2 tenant onboarding intake wizard

Appendix 6G: System screen 1 tenant onboarding intake wizard

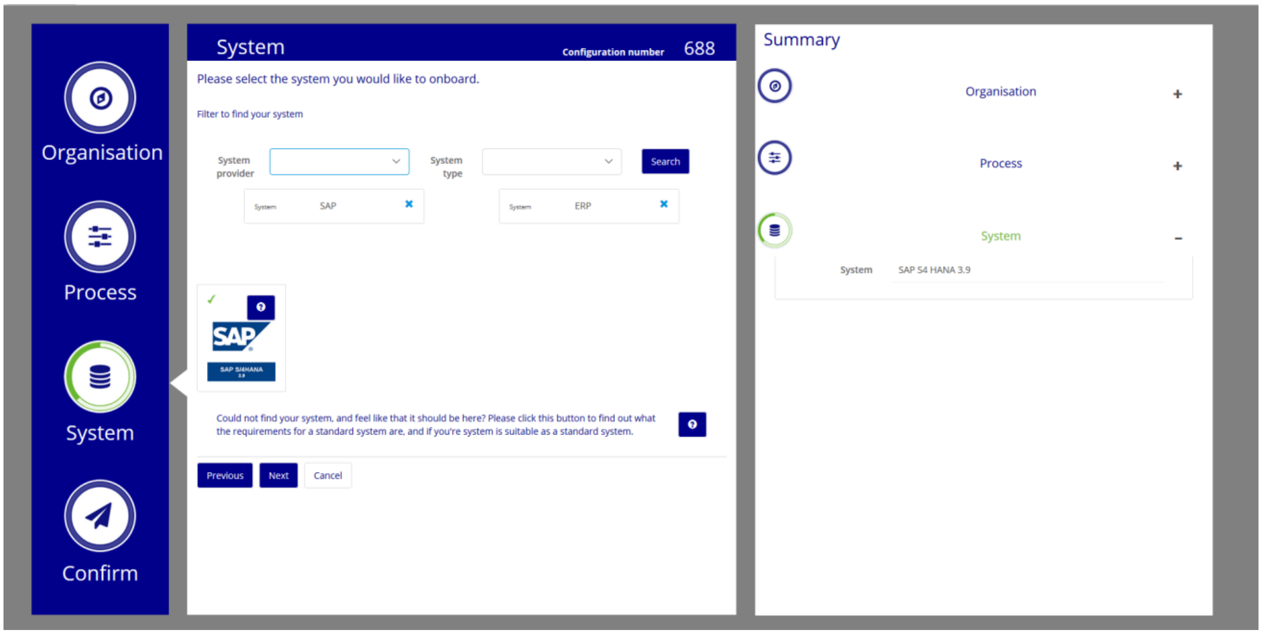

Figure 34 Prototype: system screen 1 tenant onboarding intake wizard

Appendix 6H: System screen 2 tenant onboarding intake wizard

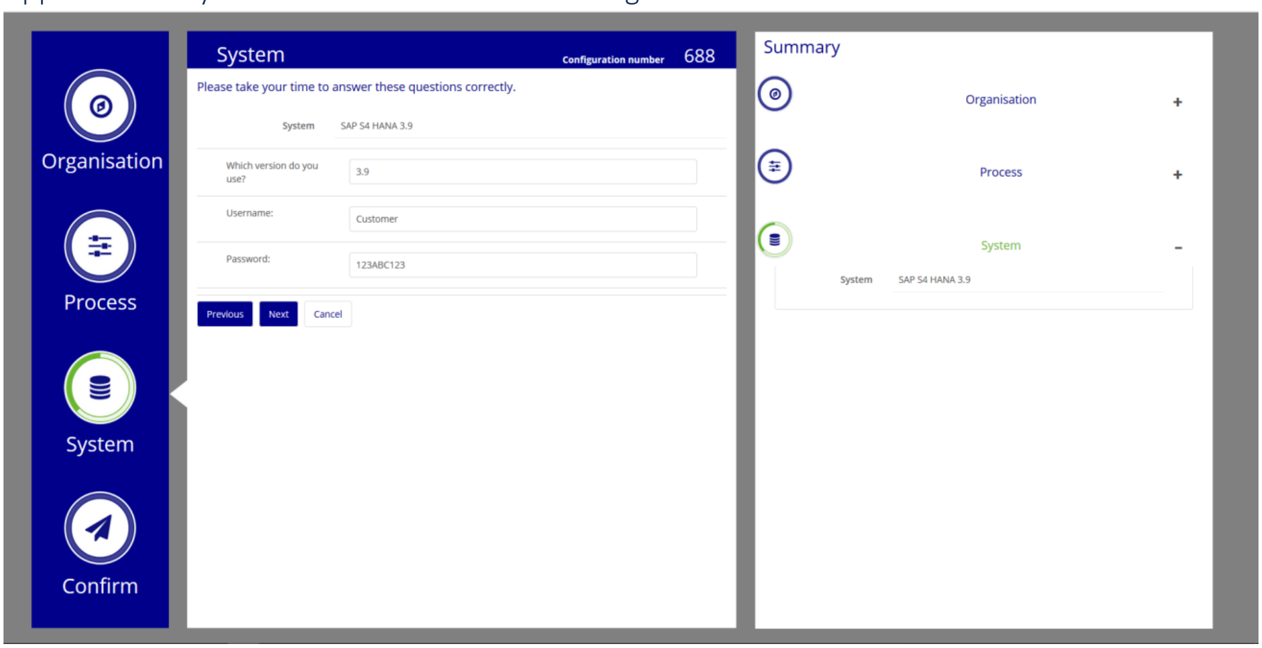

Figure 35 Prototype: system screen 2 tenant onboarding intake wizard

Appendix 6I: Confirm screen tenant onboarding intake wizard

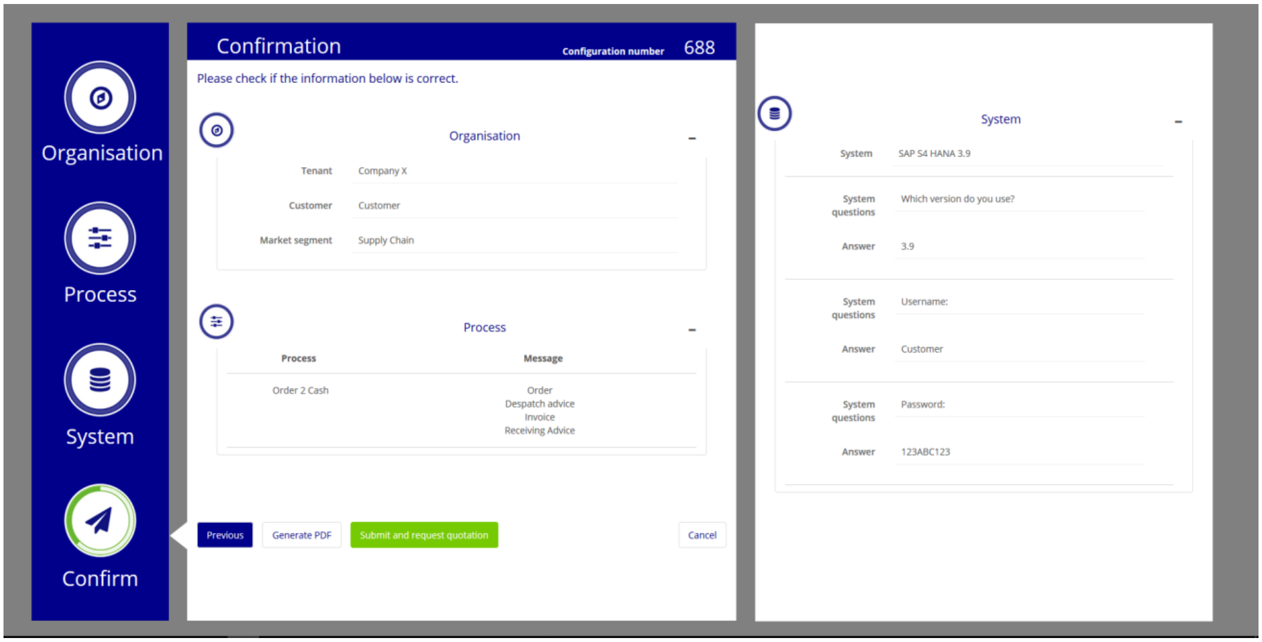

Figure 36 Prototype: confirm screen tenant onboarding intake wizard

Appendix 6J: Tenant onboarding intake wizard completion notification

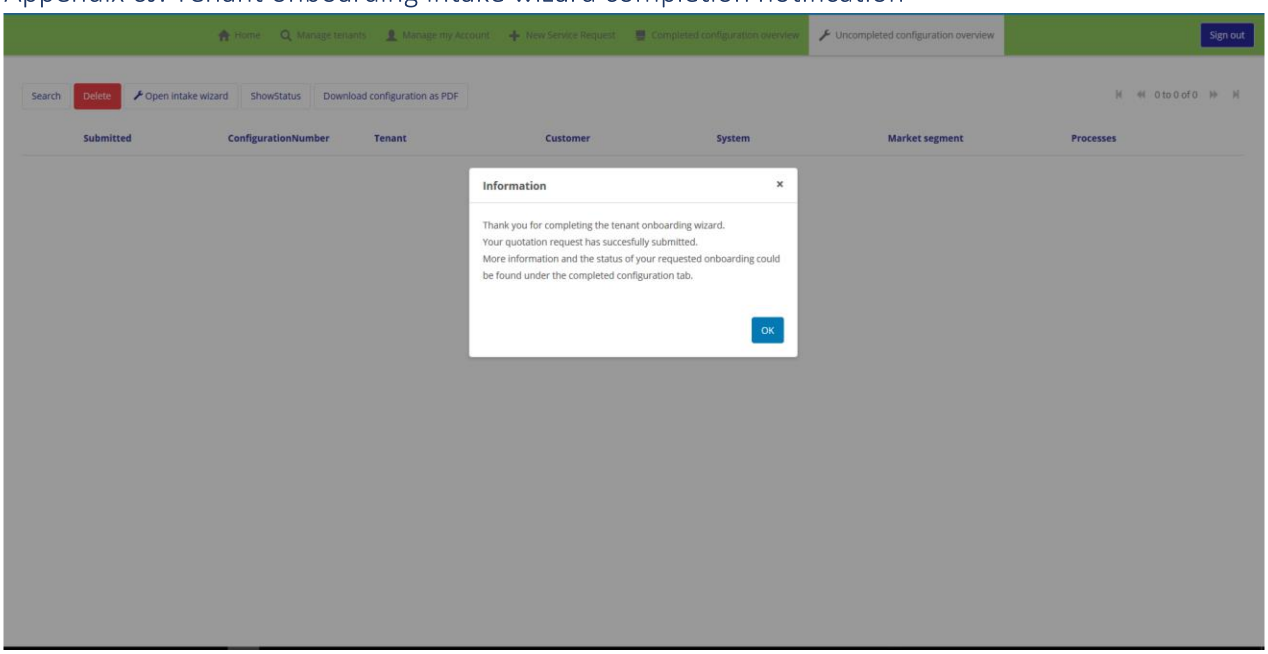

Figure 37 Prototype: tenant onboarding intake wizard completion notification

Appendix 6K: Status dashboard after completing tenant onboarding intake wizard

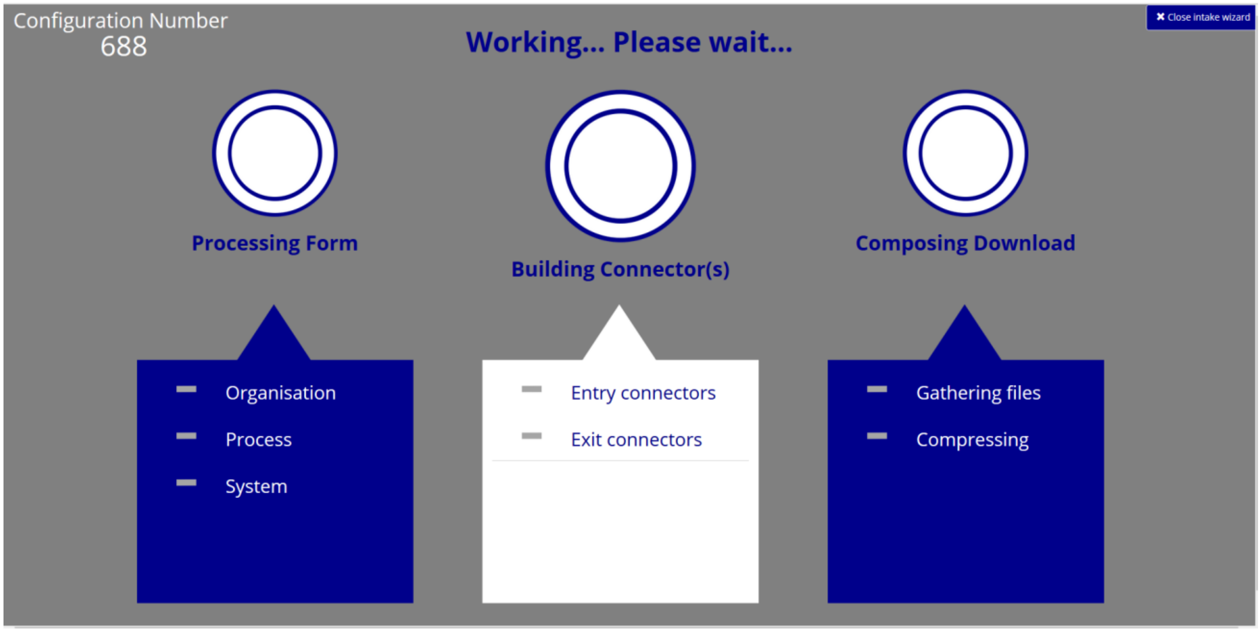

Figure 38 Prototype: status dashboard after completing tenant onboarding intake wizard

Appendix 6L: Completed onboarding process

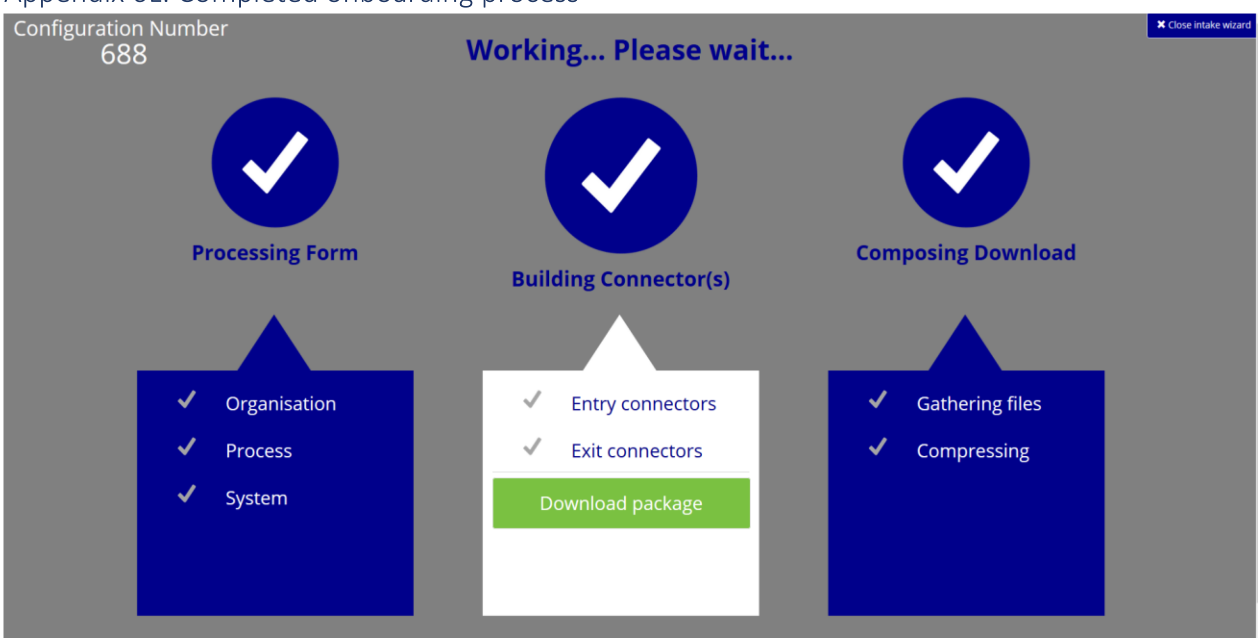

Figure 39 Prototype: completed onboarding process

Appendix 6M: Package download and installation screen

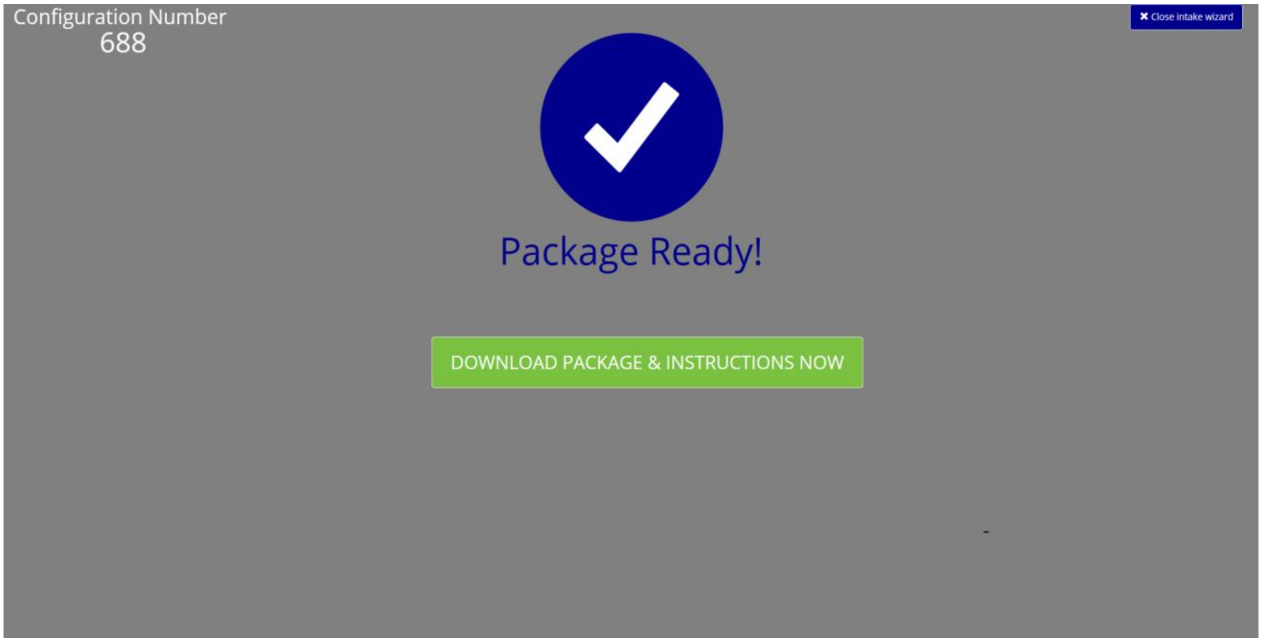

Figure 40 Prototype: package download and installation screen

Appendix 7: BPMN models desired situation

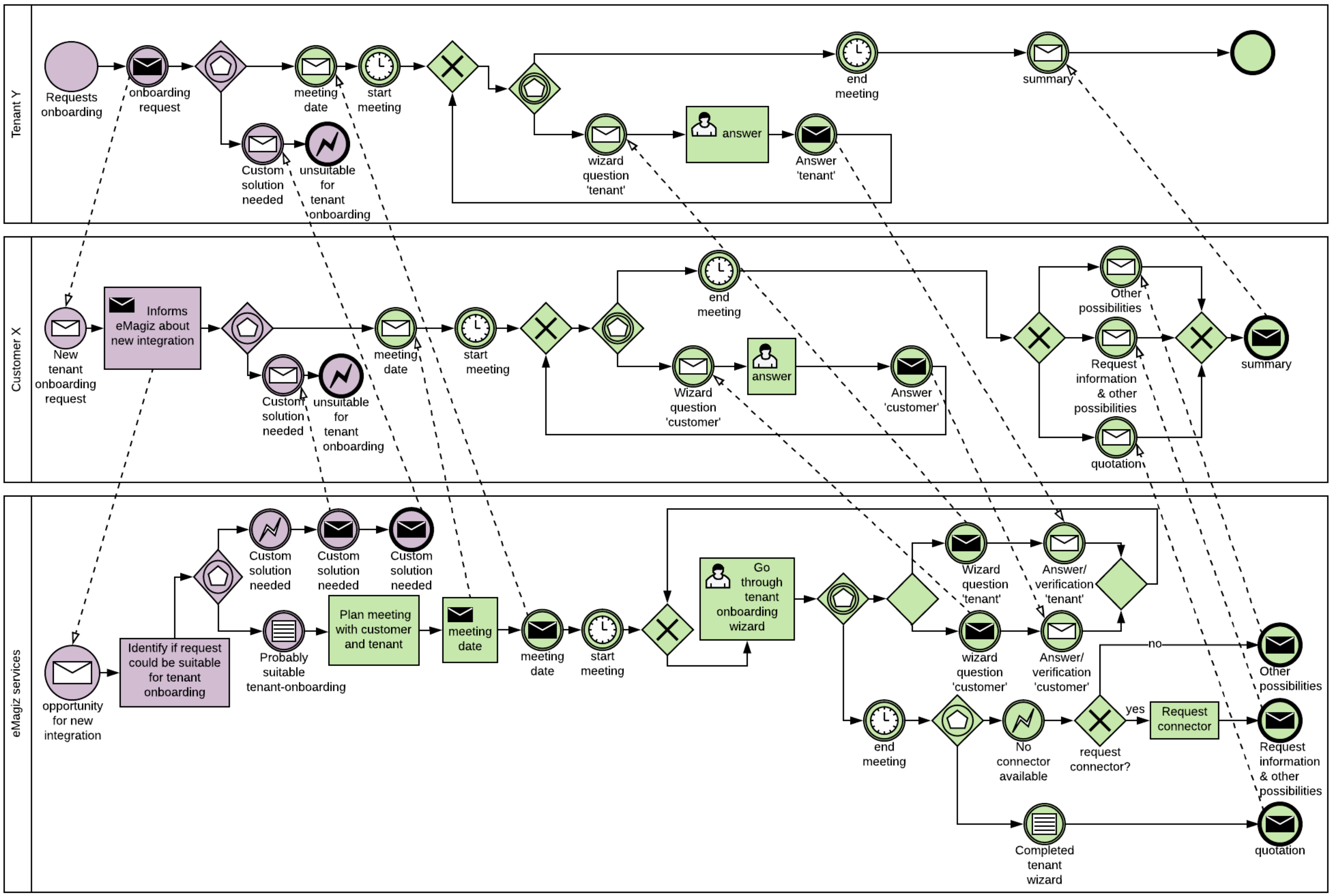

Figure 41 BPMN desired onboarding process - presales and intake – implementation phase 1

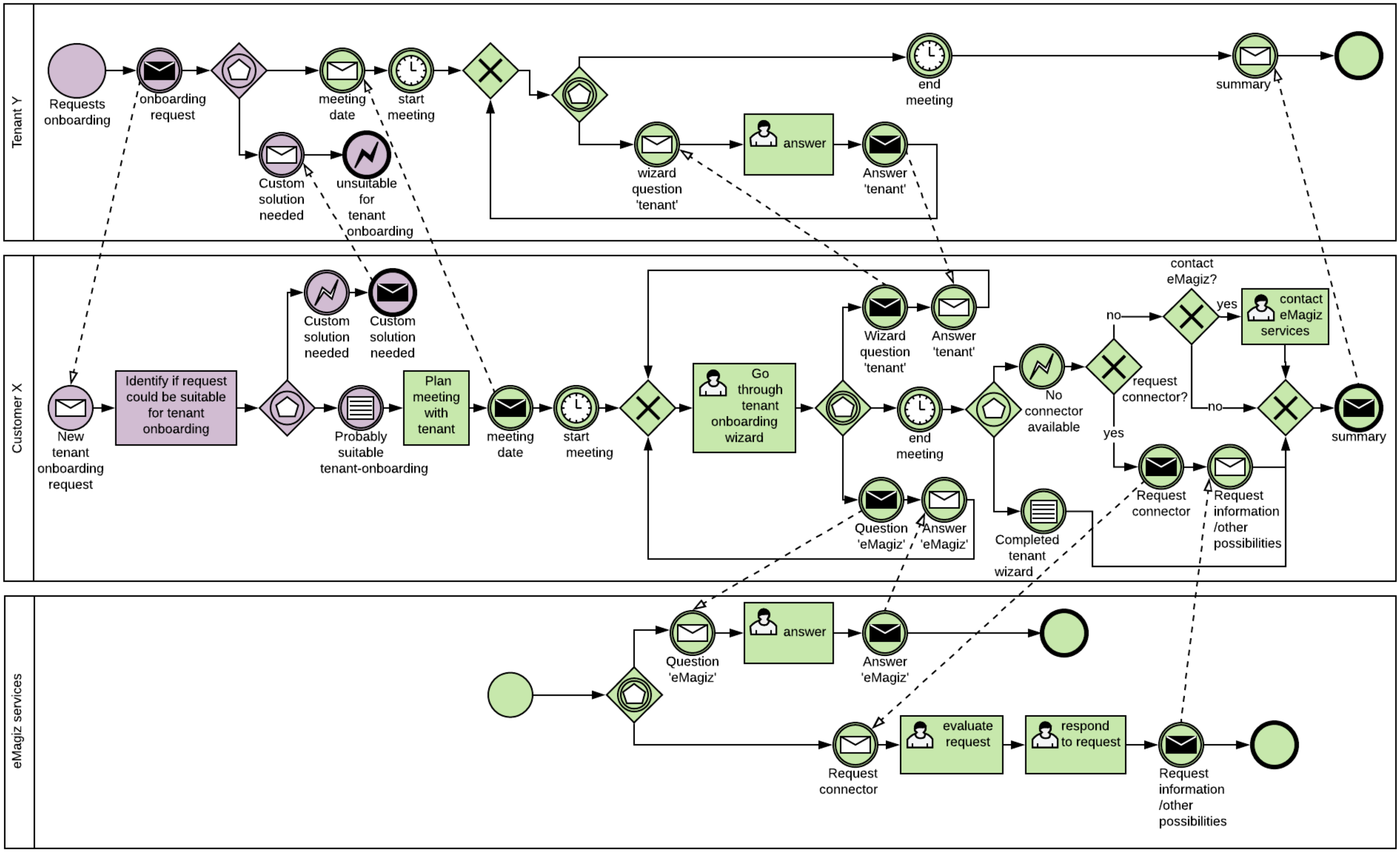

Figure 42 BPMN desired onboarding process - presales and intake – implementation phase 2

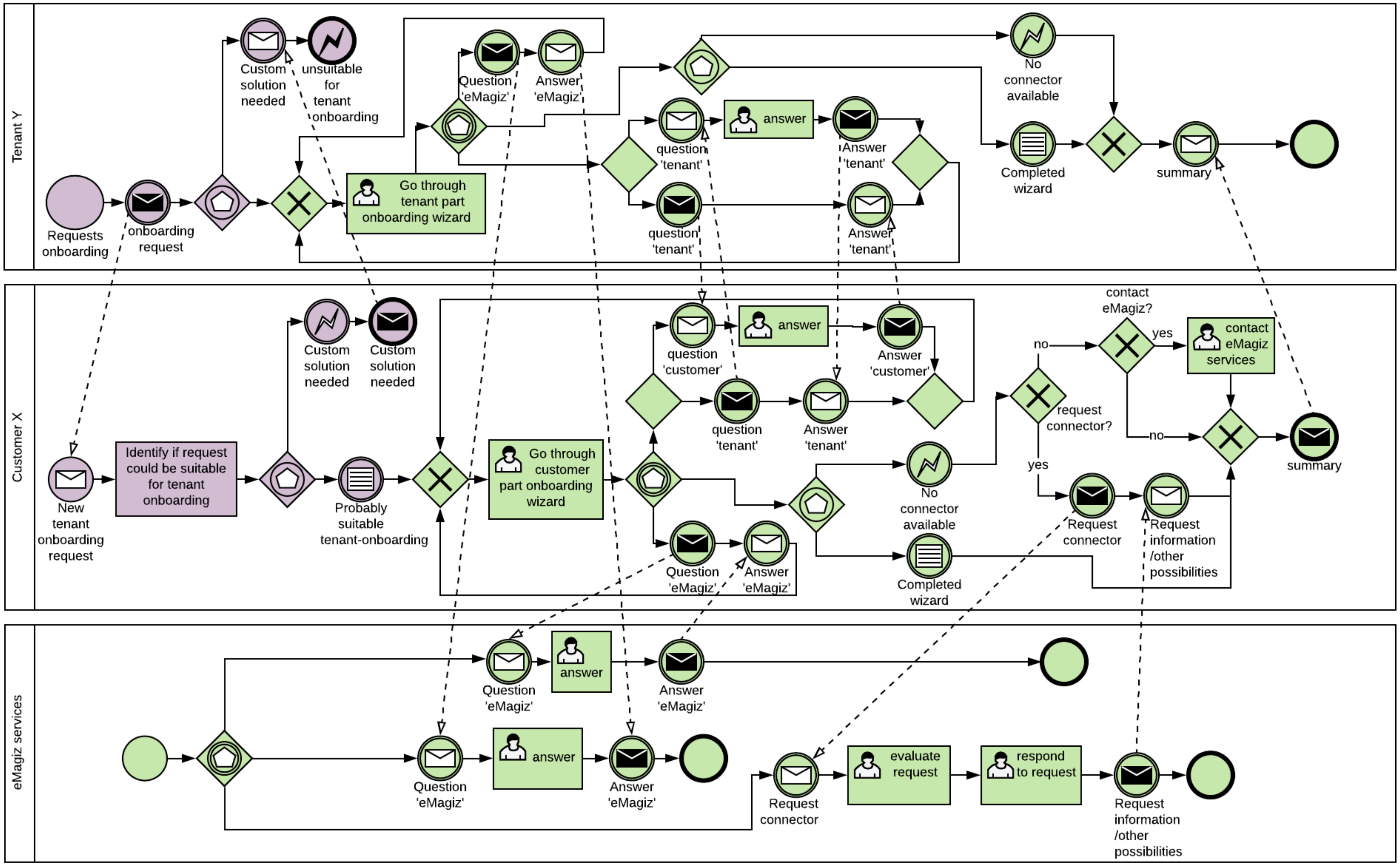

Figure 43 BPMN desired onboarding process - presales and intake – implementation phase 3

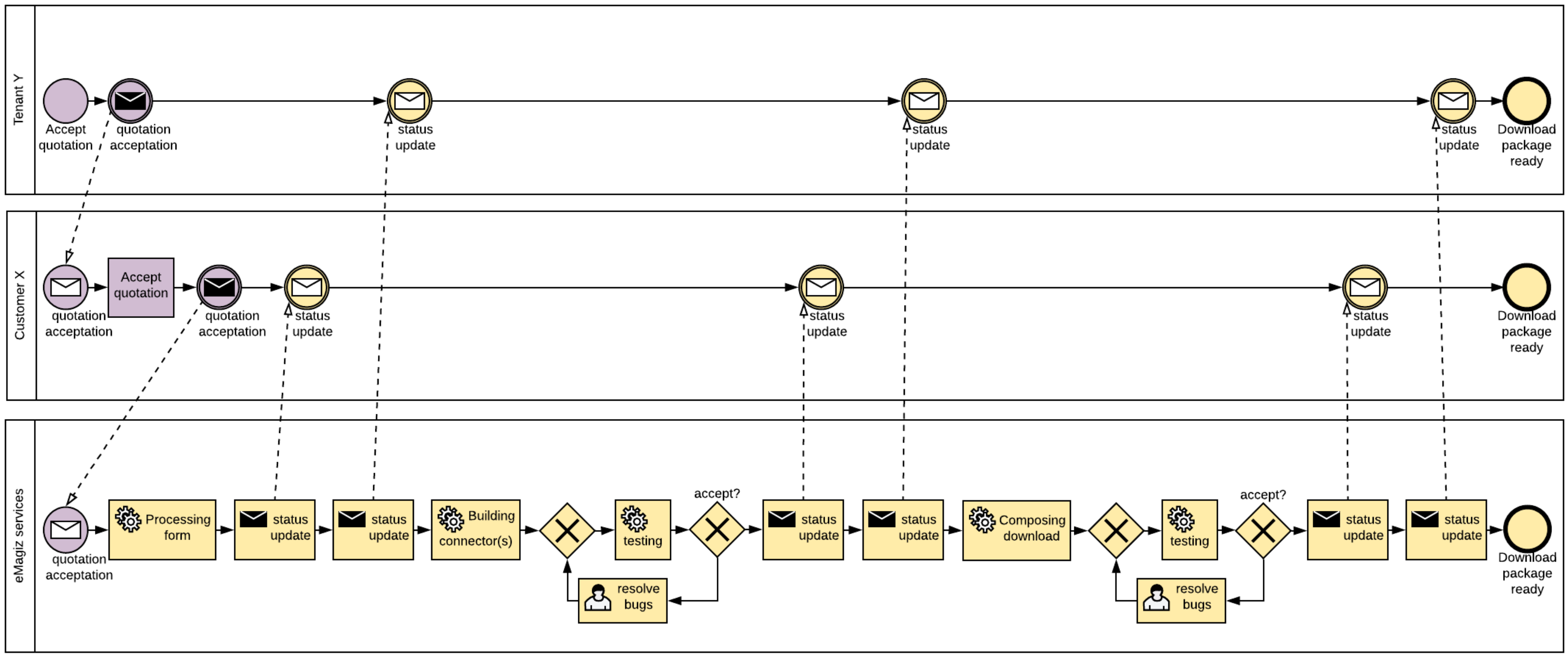

Figure 44 BPMN desired onboarding process - sales and configuration – implementation phase 1

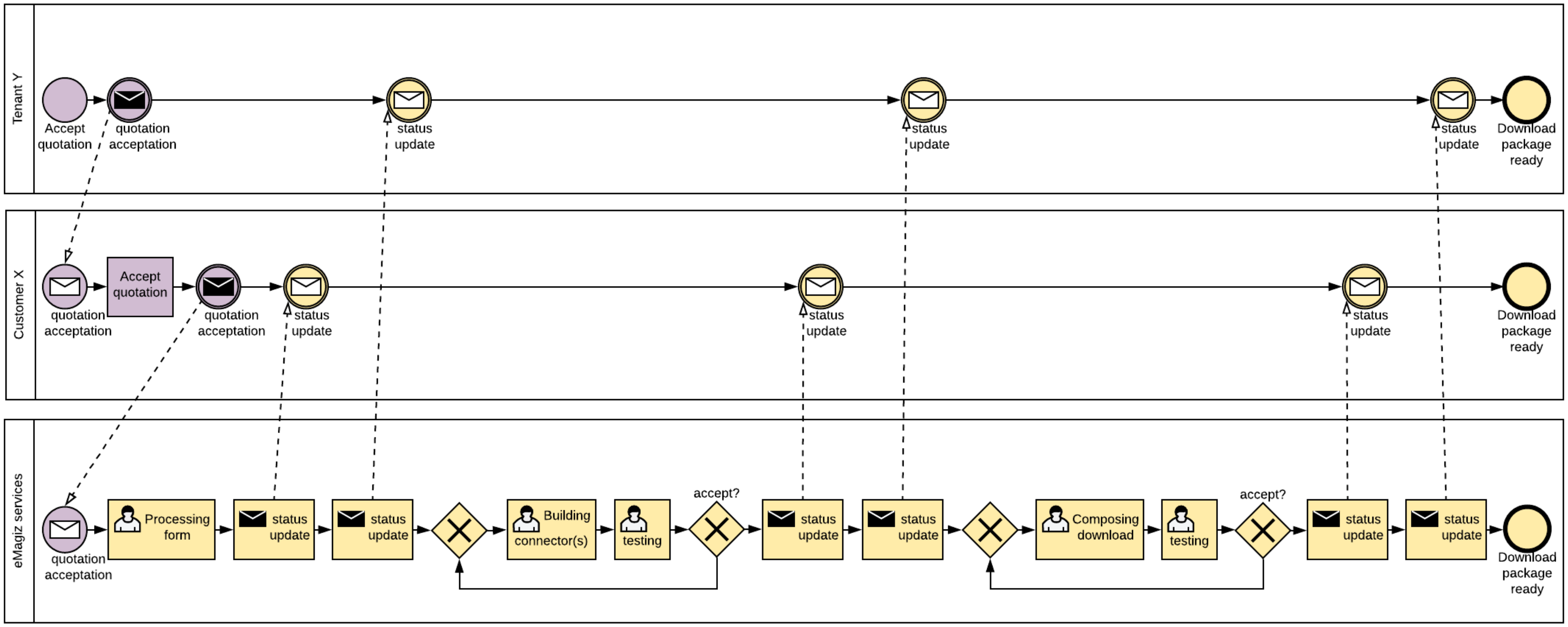

Figure 45 BPMN desired onboarding process - sales and configuration – implementation phase 2

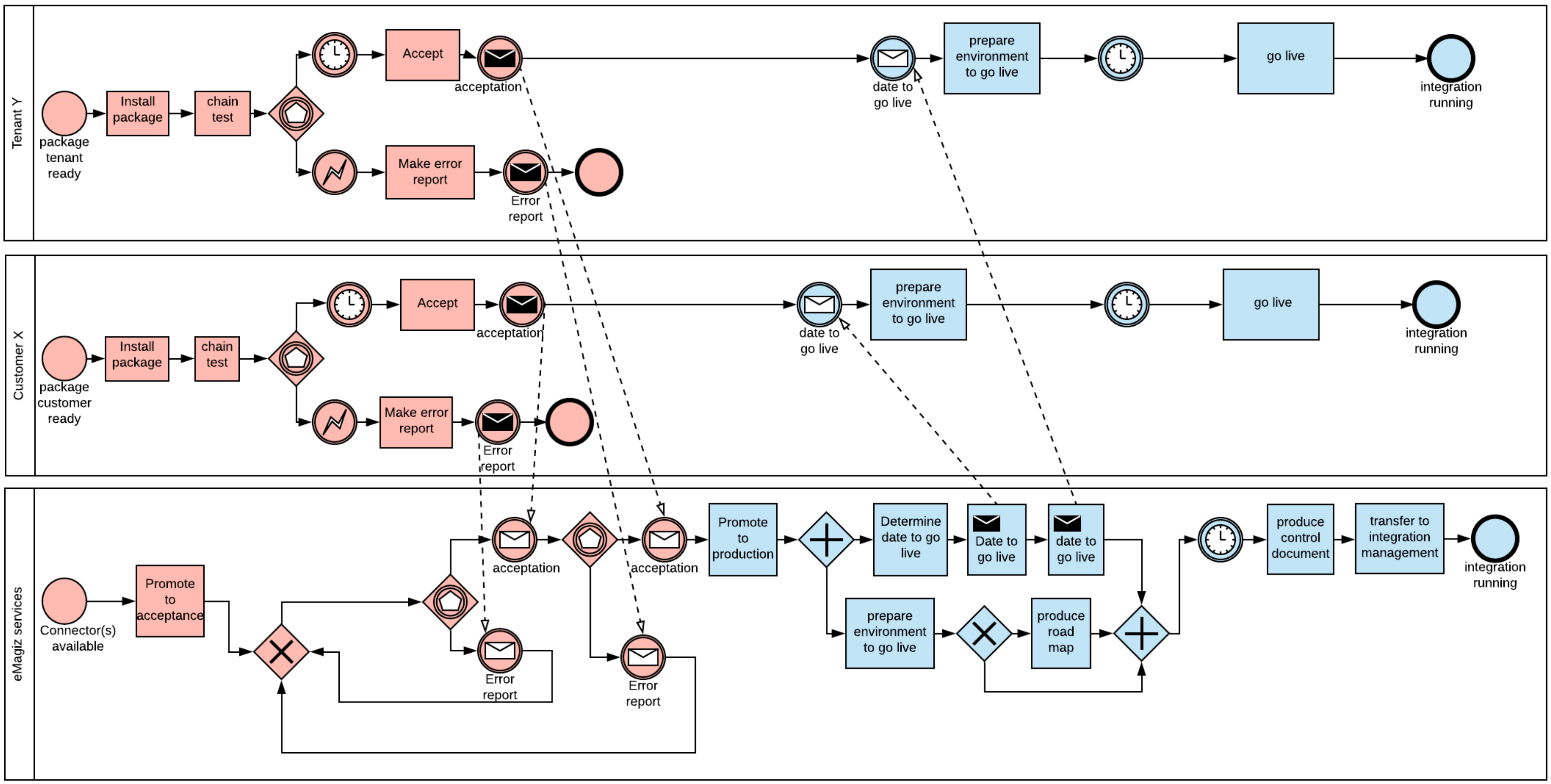

Figure 46 BPMN desired onboarding process - acceptance and going live – implementation phase

Appendix 8: Implementation of linguistic decision tree in the prototype

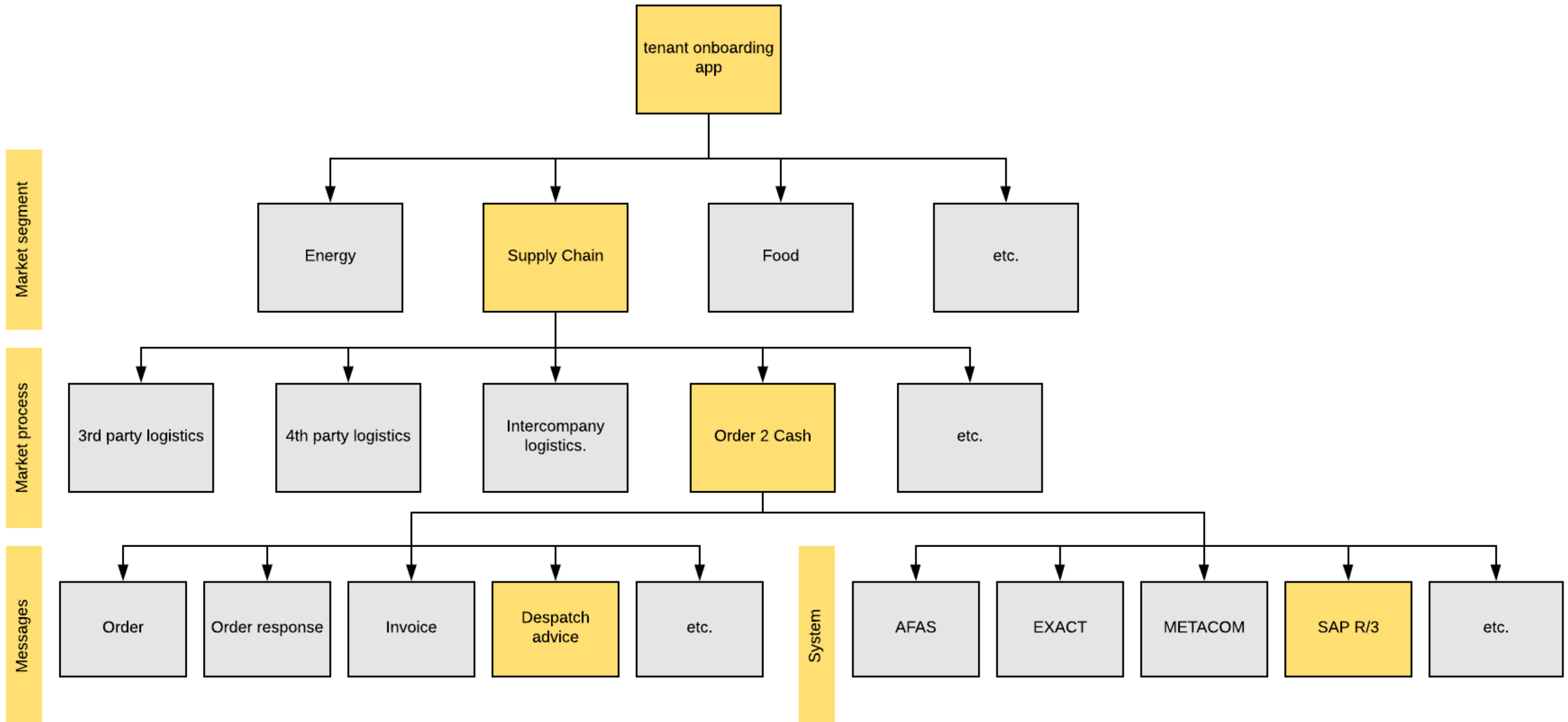

Figure 47 Implementation of a linguistic decision tree in the prototype