ACCELERATING THE DEPLOYMENT OF STRATEGIC SUPPLY CHAIN SERVICE PLATFORMS WITH PROGRAMMABLE ADD-ONS

Lappeenranta–Lahti University of Technology LUT

Master's Programme in Industrial Engineering and Management, Master's thesis

2022

Patrik Parkkinen

Examiners: Professor Timo Kärri

Postdoctoral Researcher Lasse Metso

ABSTRACT

Lappeenranta–Lahti University of Technology LUT

LUT School of Engineering Science

Industrial Engineering and Management

Patrik Parkkinen

Accelerating the deployment of strategic supply chain service platforms with programmable add-ons

Master’s thesis

2022

98 pages, 32 figures, 7 tables and 1 appendix

Examiners: Professor Timo Kärri and Postdoctoral Researcher Lasse Metso

Keywords: Digital service platform, Platform deployment, Platform scalability, Add-on

This thesis examines the deployment speed and scaling capability challenges encountered by manufacturing companies in the deployment of digital service platforms, and how these challenges can be solved using programmable add-ons. Programmable add-on is a small piece of software that is connected to the digital platform to enhance the platform functionality. This research area is currently important, as the manufacturing industry is becoming increasingly service-oriented and therefore organizations are using digital platforms to facilitate the service-oriented operations with their customers and partners.

The research was conducted as an empirical case study in a forest manufacturer’s supply chain organization. The empirical research data was collected with workshops and by analyzing the source data of the organization’s information systems. Based on the collected empirical research data, four different programmable add-ons were developed to solve challenges platform deployment teams were facing in the target organization.

Manufacturers are currently facing challenges in the deployment and scalability of digital service platforms. Issues related to the digital platform deployment speed were found to be related to the data integration problems of existing information systems, while scaling capability issues were estimated to be due to the performance and bottlenecks in existing business processes. Programmable add-ons can be developed flexibly using variety of methods, deployed on cloud platforms and integrated into digital platforms using APIs, which enables the creation of a highly scalable and flexible platform ecosystem. Based on the research, the programmable add-ons are useful in solving the deployment speed and the scaling capability challenges of digital service platforms.

TIIVISTELMÄ

Lappeenrannan–Lahden teknillinen yliopisto LUT

LUT Teknis-luonnontieteellinen

Tuotantotalous

Patrik Parkkinen

Strategisten toimitusketjun palvelualustojen käyttöönoton nopeuttaminen ohjelmoitavilla lisäosilla

Tuotantotalouden diplomityö

2022

98 sivua, 32 kuvaa, 7 taulukkoa ja 1 liite

Tarkastajat: Professori Timo Kärri ja tutkijatohtori Lasse Metso

Avainsanat: digitaalinen palvelualusta, alustan käyttöönotto, alustan skaalautuvuus, lisäosa

Tässä opinnäytetyössä tarkastellaan käyttöönottonopeuden ja skaalautuvuuden haasteita valmistavan teollisuuden digitaalisten palvelualustojen käyttöönotoissa, ja sitä, kuinka nämä haasteet voidaan ratkaista käyttämällä ohjelmoitavia lisäosia. Ohjelmoitava lisäosa on pieni ohjelmisto, joka yhdistetään digitaaliseen alustaan lisäämään alustan toiminnallisuutta. Tämä tutkimusalue on tällä hetkellä tärkeä, sillä valmistava teollisuus on siirtymässä yhä palvelukeskeisemmäksi ja siksi organisaatiot ovat alkaneet käyttämään digitaalisia alustoja helpottaakseen palvelulähtöistä toimintaa asiakkaidensa ja kumppaneidensa kanssa.

Tutkimus toteutettiin empiirisenä tapaustutkimuksena erään metsäteollisuusyhtiön toimitusketjuorganisaatiossa. Työn empiirinen aineisto kerättiin työpajoilla ja analysoimalla organisaation tietojärjestelmien dataa. Kerätyn empiirisen tutkimusaineiston pohjalta kehitettiin neljä erilaista ohjelmoitavaa lisäosaa ratkaisemaan alustojen käyttöönottotiimien kohtaamia haasteita kohdeorganisaatiossa.

Valmistavan teollisuuden yritykset kohtaavat tällä hetkellä haasteita digitaalisten palvelualustojen käyttöönotossa ja skaalautuvuudessa. Digitaalisten alustojen käyttöönottonopeuteen liittyvien ongelmien havaittiin liittyvän nykyisten tietojärjestelmien datan integrointiongelmiin, kun taas skaalautuvuuteen liittyvien ongelmien arvioitiin johtuvan olemassa olevien prosessien suorituskyvystä ja pullonkauloista. Ohjelmoitavia lisäosia voidaan kehittää joustavasti erilaisilla menetelmillä, ottaa käyttöön pilvialustoilla ja integroida digitaalisiin alustoihin API:en avulla, mikä mahdollistaa erittäin skaalautuvan ja joustavan alustaekosysteemin luonnin. Tutkimuksen perusteella ohjelmoitavat lisäosat ovat hyödyllisiä digitaalisten palvelualustojen käyttöönottonopeuden ja skaalautuvuuden haasteiden ratkaisemisessa.

ACKNOWLEDGEMENTS

First and foremost, I am extremely grateful to my supervisors, Professor Timo Kärri and Postdoctoral researcher Lasse Metso for their assistance and patience during the thesis. Their knowledge and feedback guided me in the right direction towards a final result of which I am proud. I would also express my sincere gratitude to Patrik Johansson and Anu Metsäranta from Stora Enso, who gave me the opportunity to complete my master’s thesis in their team, gave me moral support and encouragement in tough times and generously provided their professional expertise for me. Without all of you, it would have been impossible for me to complete this thesis.

Many thanks to all the professors and teachers of LUT University who have shared their knowledge with me along my academic journey and given me the opportunity to develop myself with their support. I would also like to thank my current and former colleagues in Stora Enso with who I have had the possibility to grow professionally and share my thoughts along the journey. Special thanks to the teams and individuals who I had opportunity to work with during the thesis.

Finally, I would like to thank my family: my parents, my siblings and my dog for the continuous support you have given me. Without your tremendous support, motivation and encouragement, this would have not been possible.

In Luumäki, 30.10.2022

Patrik Parkkinen

ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
BL	Bill of Lading
CRM	Customer Relationship Management
CSC	Customer Service Center
DevOps	Software Development and IT Operations
DMS	Document Management System
DXP	Digital eXperience Platform
E2E	End-to-End
ERP	Enterprise Resource Planning
ETL	Extract, Transform, Load
FaaS	Function as a Service
FHPE22	Forshaga PE Coating Machine 22
FHPE23	Forshaga PE Coating Machine 23
GOC	Global Ocean Center
HR	Human Resources
IaaS	Infrastructure as a Service
IMBM1	Imatra Board Machine 1
IMPE2	Imatra PE Coating Machine 2
IMPE3	Imatra PE Coating Machine 3
IMPE5	Imatra PE Coating Machine 5
IMPE6	Imatra PE Coating Machine 6

IoT	Internet of Things
LCNC	Low-Code/No-Code
ML	Machine Learning
PaaS	Platform as a Service
PM	Packaging Materials
PMO	Project Management Office
PSS	Product-Service System
REST	Representational State Transfer
RTA	Requested Time of Arrival
SaaS	Software as a Service
SBU	Strategic Business Unit
SC	Supply Chain
SOA	Service Oriented Architecture
SWB	Sea Waybill
UI	User Interface

Table of contents

Abstract

Acknowledgements

Abbreviations

1 Introduction ... 12
1.1 Background ... 12
1.2 Research questions and objectives ... 13
1.3 Research methods and data ... 15
1.4 Structure of the thesis ... 16
2 Digital service platforms ... 18
2.1 Servitization trend and motivation for digital business platforms ... 18
2.2 Digital business and service platforms ... 24
2.3 Platform modularization and add-ons ... 27
2.4 Digital platform deployment challenges and solutions ... 28
3 Programmable add-ons in digital platform ecosystems ... 31
3.1 Modular software development architectures ... 31
3.2 Platform ecosystem, architecture and programmable add-ons ... 33
3.3 Add-on deployment and platform integration architectures ... 39
4 Research methodology ... 41
5 Digital platform deployment and scalability challenges ... 44
5.1 Introduction to case company ... 44
5.2 Empirical background of study ... 46
5.3 Deployment and integration challenges in Platform A ... 47
5.4 Deployment and integration challenges in Platform B ... 49
5.4.1 Route planning data ... 50
5.4.2 Ocean shipping document handling ... 50
5.4.3 Machine chain follow-up ... 52
6 Supporting platform deployment with programmable add-ons ... 54
6.1 Programmable add-on development process ... 54

6.2 Developed add-ons ... 56
6.2.1 Phone number formatter ... 59
6.2.2 Shipping process data enhancer ... 61
6.2.3 Intelligent document processing ... 68
6.2.4 Process follow-up control tower ... 71
7 Results and evaluation ... 74
7.1 Key results of thesis ... 74
7.2 Discussion and evaluation of results ... 79
7.3 Future research areas and development directions ... 82
8 Conclusions ... 84
References ... 87

Appendices

Appendix 1. Training dataset size for each carrier, and machine learning model confidence results for intelligent document processing add-on

Figures

Figure 1. The relationship between platform deployment speed and scaling capability in the scope of this thesis.
Figure 2. The structure and content of this thesis and the output of each chapter.
Figure 3. Product-service continuity (Adapted from: Oliva & Kallenberg, 2003)
Figure 4. Main drivers and technical enablers towards platform enabled businesses. (Framework adapted from: Tiwana, 2014)
Figure 5. Digitalization effects throughout value system and value ecosystem. (Partly adapted from: Rabetino & Kohtamäki, 2018)
Figure 6. Different types of platforms (Adapted from: Gartner, 2016)
Figure 7. Platform architecture and add-on microarchitectures.
Figure 8. The functional elements of add-on microarchitecture can be flexibly allocated to platform or add-on. (Adapted from: Tiwana, 2014)
Figure 9. Different options for programmable add-on development. (Adapted from: Wasserman, 2020)
Figure 10. The structure of the empirical study.
Figure 11. Stora Enso Packaging Materials production units and annual production capacities (Source: Stora Enso, 2022a)
Figure 12. E.164 standard for international phone numbers. (Adapted from: Gódor et al., 2008)
Figure 13. A generated sample from the CRM contacts database using the formatting of the phone numbers.
Figure 14. Shipping document workflow in the area 1 deployment of the Platform B.
Figure 15. The machine outcoming product specifications and raw material specifications based on one input dimension (reel width).
Figure 16. Programmable add-on development pipeline and governance layer.
Figure 17. The value-complexity matrix used to prioritize the add-on development portfolio.
Figure 18. The deployed digital service platforms and programmable add-ons linked to the platforms.
Figure 19. Results for the phone number formatter add-on.
Figure 20. Construction of product transportation graph based on the transport event log data.

Figure 21. The rules used for calculating transportation graph sources and destinations.
Figure 22. Extracting all possible order routes from the transportation graph.
Figure 23. Sample of the route analysis table results.
Figure 24. Case 1: Shipping port changing from the vessel departure port.
Figure 25. Case 2: Change of the departure port based on the ship booking of the order.
Figure 26. Case 3: Transfer from feeder vessel to ocean vessel leaves a location trace to the ERP system.
Figure 27. Algorithm logic for inserting the booked transport leg into the planned leg.
Figure 28. Data enhancement results for the shipping process data enhancer add-on.
Figure 29. Unsolved cases after the implementation of the shipping process data enhancer add-on.
Figure 30. Architecture for Azure cloud-based document library.
Figure 31. Document workflow in the future system architecture for the ocean shipping document processing.
Figure 32. The determination of suitable barrier coating machines in each product attribute.

Tables

Table 1. Core elements of a platform ecosystem (Source: Tiwana, 2014)
Table 2. Strategic business units (SBUs) and product lines of Stora Enso Packaging Materials division (Stora Enso, 2022b).
Table 3. Descriptions of platforms in the scope of this thesis.
Table 4. Developed add-ons, functions, and the development objectives of the programmable add-ons.
Table 5. Development areas, measurement areas and measures of programmable add-on performance.
Table 6. Data quality improvements per add-on.
Table 7. Process performance improvements per add-on.

1 Introduction

This chapter explains the reasoning for the research and what it aims to achieve. The chapter contains the background, research questions and objectives, limitations, brief overview of the methodology and the structure of this thesis. The output of this chapter is the framework in which the results of the study will be presented.

1.1 Background

The modern highly dynamic market environment causes companies to struggle in competing in the markets and adapting to the fast development of the markets (Chan et al., 2018). In response to the increasing competition, many companies are starting to use digital platforms to leverage their business strategies (Cenamor et al., 2017). Digital platforms are technologies that allow firms to standardize, edit, and distribute data, and deliver services efficiently on an unprecedented scale (Yoo et al., 2010). As digital platforms develop and modern technologies emerge, they are transforming the way companies build competitive advantage and deliver value to their customers (Porter & Heppelmann, 2014). In addition, the growth in innovation cost, market dynamism and deepening specialization drive the trend of servitization which pressures companies to adapt digital service platforms technologies (Tiwana, 2014). Thus, digital platforms play a significant role in many companies' value propositions by enabling them to leverage emerging technologies such as big data, artificial intelligence and machine learning, and to offer more advanced service offerings to the customers (Cenamor et al., 2017).

The academic literature has extensively studied the benefits of implementing digital platforms for improving the performance and competitiveness of companies especially in servitization literature. For example, using digital technologies enable companies to improve service quality and reduce operational costs (Kindström & Kowalkowski, 2014). Digital platform enabled approaches may overcome the reduction of returns associated with extending the service offering of a firm by enabling value creation and capturing through monitoring, controlling, optimizing and automatic function in digital platforms and ecosystems (Kohtamäki et al., 2019). Digital platforms allow organizations to configure and deploy service implementations which were previously not possible due to the level of customization and scale (Silva & Soares, 2021). Finally, digital platforms connect value ecosystem actors in a more efficient and value-adding manner thus enabling companies to be more competitive in the markets (Tian et al., 2021).

However, recent academic literature has not sufficiently considered the issues in digital platform deployment and scalability, and solutions for these issues. Organizations are struggling with the platform integrations within their existing information systems (Schmidt et al., 2019; Dillion et al., 2010). In addition, they do not currently have the technical capability to integrate the digital platforms within the value ecosystem with the partners' platforms (Brechtel & Altmann, 2021). On the other hand, the existing business processes must, at least in some degree, be transformed into new digital business models in digital platforms (Strutynska et al., 2019). This means that the platform functionality must be extended with domain-specific applications which will facilitate the digital platform operations in the organization. Therefore, researching the issues organizations are facing while deploying digital platforms, and methods for solving these problems and extending the digital platform functionality is currently important to be examined.

1.2 Research questions and objectives

This thesis has two main objectives. First, this thesis has an objective to solve issues in digital platform deployment and scaling capability in a board and paper manufacturer's supply chain organization and thus accelerate the deployment and enhance the scalability of these digital platforms. The research was done for Stora Enso Packaging Materials (PM) division, where a deployment of two service platforms were ongoing during the thesis. Second, by extending on the current academic literature of digital platforms in manufacturing domain and the enhancement of platform functionality with programmable add-ons, this thesis also intends to shed light on the challenges manufacturers currently face when deploying digital platforms and the viable solutions to solve these challenges.

The solutions for the digital platform integration issues in the target organization were developed by using a programmable add-on approach. Programmable add-ons can be defined as "software subsystems or services that connect to the platform to add functionality to it" (Tiwana, 2014). Think of a web browser (e.g., Chrome) where you can install add-ins to enhance its functionality or a smart phone platform (e.g., Android) where you can install applications to add custom functionality into the platform. In a similar fashion, the purpose of the programmable add-ons is to solve the platform integration related problems and enable organizations to add custom domain-specific functionality into the business platforms related to their specific business area.

It is important to differentiate between deployment speed and scaling capability in the scope of this thesis. Deployment speed means how fast the next deployment area of the digital platform can be deployed. Scaling capability means how capable the platform is in scaling to the future deployment areas. By this logic, the hypothesis of this thesis is that with programmable add-ons the issues in digital platform implementation can be solved and thus deployment time of digital platform can be decreased and/or the scaling capability of the digital platforms can be increased. Figure 1 below illustrates this relationship.

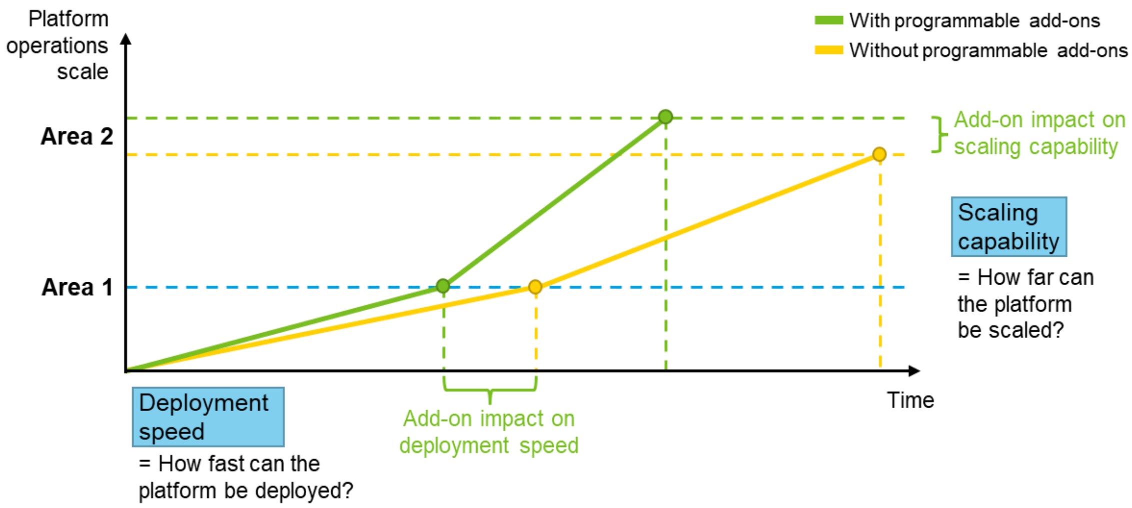


Figure 1. The relationship between platform deployment speed and scaling capability in the scope of this thesis.

It is evident from the academic literature that digital platforms have gained an increasing amount of attention in the past years, and that the deployment issues and strategies to overcome these have not been thoroughly researched. Therefore, it is important to also research what issues organizations face when deploying and scaling digital platforms and how these issues can be solved. To ensure a thorough and comprehensive review of the topic, this thesis will be structured with the following research questions:

RQ1 What are the functional problems related to digital service platform deployment speed and scaling capability?

RQ2 How can the programmable add-ons be developed and integrated into digital platforms?

RQ3 How effective are the programmable add-ons in solving the functional problems related to digital service platform deployment speed and scaling capability?

The scope of this thesis is limited to the development and deployment of programmable add-ons which includes two of the deployable digital service platforms and the developed add-ons related to these platforms in the target organization. Extensive empirical analysis to the integration problems of the platforms in the target organization is beyond the scope of this thesis due to the limited research time. The problems regarding platform deployment and scalability problems were discussed in workshop meetings with the platform deployment teams and taken as input to this study.

Second, this study does not consider the platform and add-on governance including the maintenance and ownership of the add-ons because the governance of programmable add-ons will not add value in the scope of this research. Programmable add-on governance will instead be suggested as a future research topic for the scholars interested in this subject area. Third, the scope this thesis is only limited to the functional problems in the platform deployment because the purpose of the programmable add-ons is to add functionality into the platform. The non-functional problems (e.g., performance, security, reliability) of digital platforms are left outside of the scope of this thesis.

1.3 Research methods and data

This thesis is conducted as a case study. Case study is a qualitative research method, where a particular case or cases are studied intensively for the purpose of understanding a larger class of units (Gerring, 2004). Characteristics of a case study include collecting material using several methods such as interviews, observations, discussions and studying documents (Hirsjärvi et al., 2009). The case examined in this thesis was applying the programmable add-on methodology in a manufacturing organization’s ongoing digital service platform implementation. The research data of this thesis was collected with workshops with the deployment teams of the digital service platforms and with examining the current situation of the challenges before and after implementing the programmable add-ons to the platforms.

The theoretical background of this thesis examines the current academic literature and trends in servitization, the role of digital platforms in it, and the key features of digital platforms. Also, the theoretical background defines the concept of programmable add-ons, the usage of add-ons in digital platform domain and development and integration principles related to them. The theory part of the thesis provides a scientific framework for the empirical research, as well as tools to support the empirical analyses.

The research material and data for the empirical study were collected in two parts. The material for defining the deployment problems were gathered using a workshop research method. Workshops are a research method where a group of people focus on the study of a domain-specific issue (Ørngreen & Levinsen, 2017). The workshops were facilitated for the platform deployment teams, where they could present the problems which they currently faced in the deployment. For the analysis of the problems and the performance of the add-ons, the current state analysis of the organization was conducted using the data from the information systems using numerical analysis. The performance of add-ons was measured by analyzing the data after the add-on implementation.

1.4 Structure of the thesis

The rest of this thesis is structured as follows. Second and third chapters contain the theoretical background for this thesis. Chapter 2 establishes the theoretical foundation for the research area of this thesis. It discusses the current trend of servitization in the manufacturing companies, the role of digital business and service platforms in this trend, the key features of digital platforms and common problems present in digital platform deployment. Chapter 3 contains the theoretical background for the programmable add-ons. It discusses the modularization from the software development point of view, application of programmable add-ons in digital platform domain, add-on development and finally add-on integration architectures present in the academic literature.

The fourth chapter contains the research methodology and the framework in which the empirical study will be conducted. Fifth and sixth chapters contain the empirical part of the study. First, chapter 5 establishes the setting for the empirical study part of this thesis. It contains the introduction to the case company, empirical background of the study and the deployment and integration problems currently present in the deployment and which will be answered in the sixth chapter. Chapter 6 contains the programmable add-on development process, solution descriptions and key results for each developed add-on. Chapter 7 contains results of this thesis. It discusses the key results gathered during the study, the evaluation of the results in respect to the research questions, the theoretical implications and contributions of the results and follow-up measures and recommendations to the target organization of the study and to the academic domain area. Finally, chapter 8 compiles the key findings and results of the study into a summary and conclusions. The structure of this thesis is summarized in figure 2.

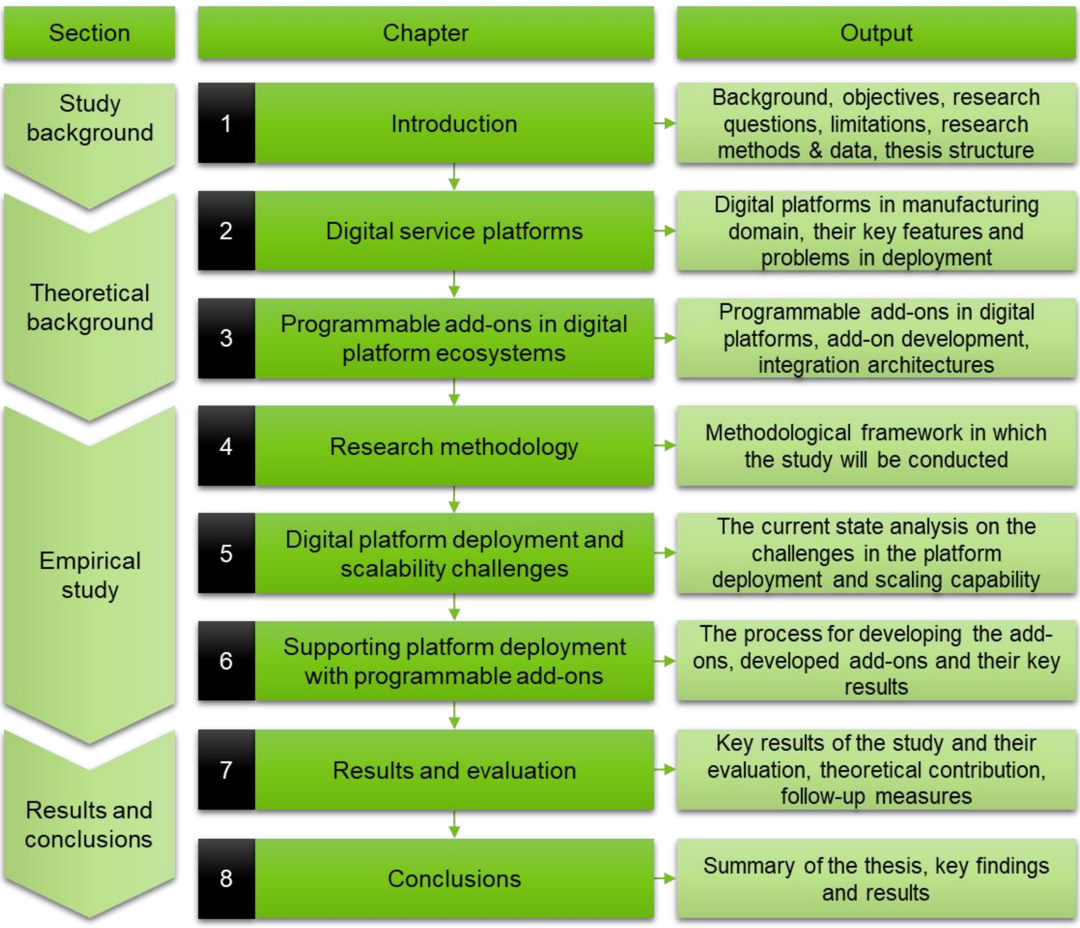


Figure 2. The structure and content of this thesis and the output of each chapter.

2 Digital service platforms

This chapter establishes a theoretical foundation for the digital platform research area of this thesis. This chapter explains the current trend of servitization in manufacturing and industrial companies, the role of digital service platforms in this underlying paradigm shift and the characteristics, benefits and limitations of the supply chain service platforms. The output of this chapter is the foundation in which the programmable add-on methodology is utilized in the case research.

2.1 Servitization trend and motivation for digital business platforms

As the technology and industrial business environment continues to develop, manufacturing companies are becoming increasingly interested in the industry 4.0 technologies to ensure both long-term competitiveness in the markets and to enable adaptation to the dynamically changing environmental conditions (Hovarth & Szabo, 2019). The drivers for this transformation are fundamental in the entire industrial value creation domain. On the other hand, the operating environment is changing rapidly and dynamically due to additional competition (Hovarth & Szabo, 2019), shorter development period (Bauer et al., 2015), individualization (Aheleroff et al., 2021) and socio-economic changes (Götz & Jankowska, 2017), which forces industrial companies to develop and redesign their business processes to meet the increasing complexity (Lasi et al., 2014). On the other hand, the benefits of emerging technologies, such as automation, digitalization, networking and miniaturization forces companies to quickly adopt modern technologies to remain competitive in the markets (Lasi et al., 2014). These factors play a key role in the introduction of digital technologies in value chains, which in turn acts as a defining factor in transforming and improving the value chain and connectivity among manufacturing companies (Sivula et al., 2022).

According to Ferreira et al. (2016) and Martin et al. (2018), the current development in technological capabilities with the continuously more demanding and unique customer requirements pressures companies to adapt to the changed operating environment by transforming from the traditional product-based business models towards product-service based business models. This underlying transition is called servitization in the academic literature (Lee et al., 2014). By the definition of Baines et al. (2009) servitization is the innovation of organizational capabilities and processes, from product sales to integrated product services. Martinez et al. (2010) define servitization as the strategic innovation of an organization's capabilities and processes to shift from selling products to selling an integrated product and service offering that delivers value in use. These integrated product and service offerings are called Product-Service Systems (PSS), which are combination of the manufactured products with value-added services which the customers need (Lee et al., 2014).

The concept of product supporting services was originally proposed by Vandermerwe & Rada (1988) who argued value chains to automatically be partially service-oriented as the companies always sell their expertise in addition to their products. The transition towards service dominant business models is traditionally understood as a continuum, which was presented by Oliva & Kallenberg (2003) in their study (figure 3). The continuum represents the strategic positioning of a company moving to the right as the relative importance of the services increase and the products decrease.

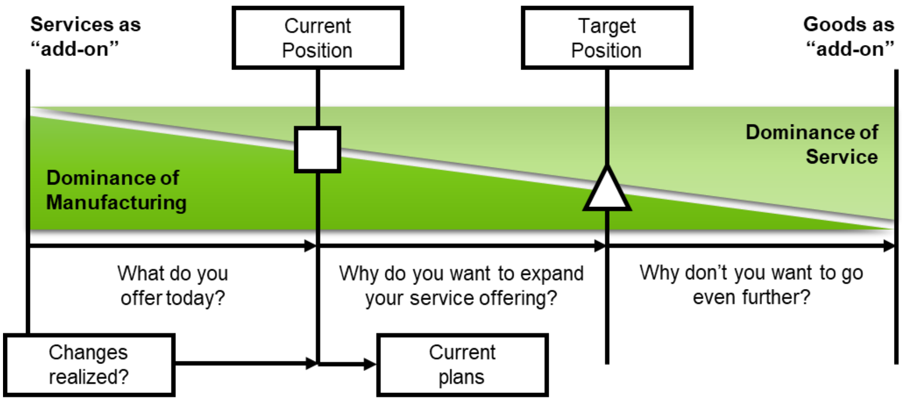


Figure 3. Product-service continuity (Adapted from: Oliva & Kallenberg, 2003)

However, more recent servitization literature has highlighted two key challenges organizations face in the manufacturing servitization trend. First, more recent academic literature has started to increasingly question the unidirectional transition towards service-dominant manufacturing business models (Tian et al., 2021). A literature study by Brax & Visintin (2017) concluded, that academic studies show a gradually increasing pattern towards more complex service-based portfolio of offerings rather than a simple “shift in a goods-dominant/service-dominant continuum”. Also, both Cenamor et al. (2017) and Baines & Lightfoot (2014) highlighted, that customers want manufacturers to provide a full spectrum of products and services adapted to their specific needs, which may differ from one customer to another. The resultant offering thus refers to individualized product-service combinations that play a key role in customers’ core operations (Cenamor et al., 2017). This fact is also supported by Kamalaldin et al. (2020) who concluded servitization to affect how companies maintain relationships and co-create value with customers and suppliers, and by Martin et al. (2019) who argued servitization to also transform how companies create, deliver, and capture customer value. This results in unreasonably high complexity of the product-service portfolio which may leave manufacturing firms facing an ambiguous business focus and mode complex interactions in the business environment (Cenamor et al., 2017).

Second, recent academic literature has highlighted that servitization does not automatically increase the performance or profits of a company although the servitization trend does reflect the customer requirements and individualization requirements (Cenamor et al., 2017). Benedettini et al. (2016) concluded, that servitization may increase the likelihood of bankruptcy due to the barriers of the transition into the PSS based business models. The investment in service business which increases the service offerings and costs but does not generate the correspondingly expected higher returns is called service paradox (Gerbauer et al., 2005). According to Smith et al. (2010), research on servitization paradoxes highlights the importance of managing concurrently the interrelated elements of PSS to overcome this paradox. Settanni et al. (2014) discussed that manufacturers need more efficient mechanisms to deliver adapted product and services offerings with a clear strategy and competitive prices and that way gain the benefits of servitization. In other words, PSS-dominant manufacturing firms should develop implementation mechanisms that assure competitive levels of both customization and organizational efficiency (Silvestro & Lustrato, 2015).

As a result of the aforementioned challenges in the complexity of PSSs and service paradoxes, servitization literature has been starting to research digitalization enabled platform approaches to overcome the challenges in servitization transition. (Kohtamäki et al., 2019; Eloranta & Turunen, 2016; Pekkarinen & Ulkuniemi, 2008). Increasingly researched area for PSSs in manufacturing organizations are digital platforms (De Reuver et al., 2018). Digital platforms can be defined as technological infrastructures that allow firms to develop, configure, and deliver advanced services efficiently (Cenamor et al., 2017). Gawer and Cusumano (2014) defined platforms as products, services, or technologies that act as a foundation upon which external innovators, organized as an innovative business ecosystem, can develop their complementary products, technologies, or services. Gartner (2016) defined digital platforms as a symbiotic collection of technology capabilities and components, which contain interoperable set of services that can be brought together to create applications, apps and workflows. Tiwana (2014) defined digital platform as an extensible codebase of a software-based system that provides core functionality shared by apps that interoperate with it, and the interfaces through which they interoperate.

The research area of digital platforms is interesting from the manufacturers' point of view, since digitalization has influenced servitization by creating new opportunities for services, platforms, intelligent products, and novel business models (Kohtamäki et al., 2019) encouraging more manufacturers to transform towards servitization. In addition, digitalization is increasingly viewed as an enabler and driver of the business model, value creation, and value capture of a company (Porter & Heppelmann, 2014). Digital platforms leveraging the value of digital technologies is particularly beneficial in the context of advanced service implementations that facilitate both customization and efficiency of delivering services to customers (Silvestro & Lustrato, 2015). For overcoming the service paradox, Kindström & Kowalkowski (2014) mentioned that digital technologies enable firms to improve service quality and reduce operational costs. Coreynen et al. (2017) and Vendrell-Herrero et al. (2016) also added that digital servitization may help manufacturing firms add services to their offerings. Finally, Sjödin et al. (2016) claim that digitalization capabilities are a key facilitator for advanced service offerings. The central theme in this area is digital servitization which can be defined as the transition toward smart solutions (product-service-software systems) enabling value creation and capturing through monitoring, controlling, optimizing, and automatic functions (Kohtamäki et al., 2020).

As the technology development and organizational culture has progressed, the product-service business models have gradually become a fundamental business constituent for the manufacturing sector companies (Kamal et al., 2020). The recent development of advanced and innovative technologies, for example artificial intelligence and machine learning, has allowed companies to develop efficient customer-centric service deliveries and improved service-oriented strategies by developing systems to process and utilize real-time business data (Jiang et al., 2021). This development has resulted digital service platforms to grow from intraorganizational information systems into digital ecosystems that are at the core how organizations interact in cooperation with each other and how services are provided to stakeholders of the company (Silva & Soares, 2021).

The motivation for manufacturing companies for mowing towards platform-enabled service-based business models is twofold. This dynamic is illustrated in the figure 4, where the market forces drive the transition towards digital platforms and the technological advances enable the adaptation. As discussed previously, the transition is driven by the market trends of competition, shortening development periods, individualization and socio-economic changes. However, the migration towards digital platforms is enabled and fueled by the confluence of digitization, and the ubiquity of cheap and fast Internet-based networks (Tiwana, 2014). Digitalization enables the activities and process to be organized on the digital platforms (Lasi et al., 2014) while the growing presence of cheap and fast wireless data networks cause loosely coupled networks rival efficiencies and allows scaling of systems without ownership (Tiwana, 2014).

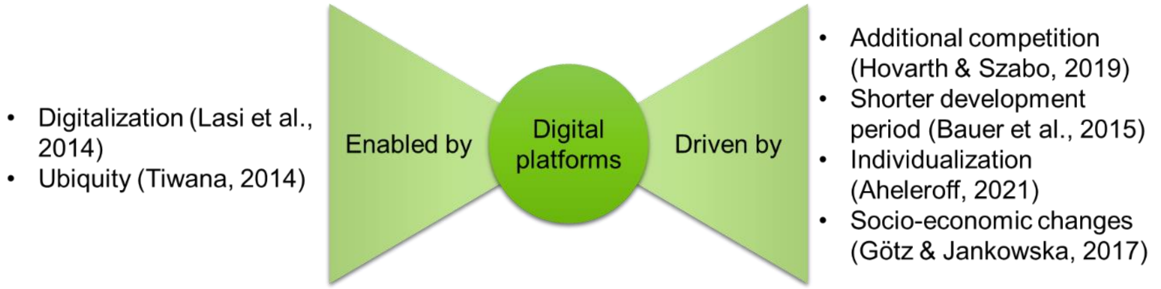


Figure 4. Main drivers and technical enablers towards platform enabled businesses. (Framework adapted from: Tiwana, 2014)

Manufacturers are adopting digital platforms to connect network actors and their resources in a more intelligent and value-adding manner (Tian et al., 2021). As discussed, this results in new opportunities for manufacturers to create new service offerings and sustain competitive advantage (Cenamor et al., 2017; Constantinides et al., 2018). The effects of the platform adoption of manufacturing companies can be observed from two perspectives. From a manufacturing engineering perspective, platforms as technological architectures enable economies of scope in supply and innovation (Tian et al., 2021). On the other hand, from an economics perspective digital business platforms enable economies of scope in demand as markets (Gawer & Cusumano, 2014). The connection between authors throughout the value system and value ecosystem is illustrated in the figure 5 which represents the value system flowing from raw material suppliers all the way to end-customers.

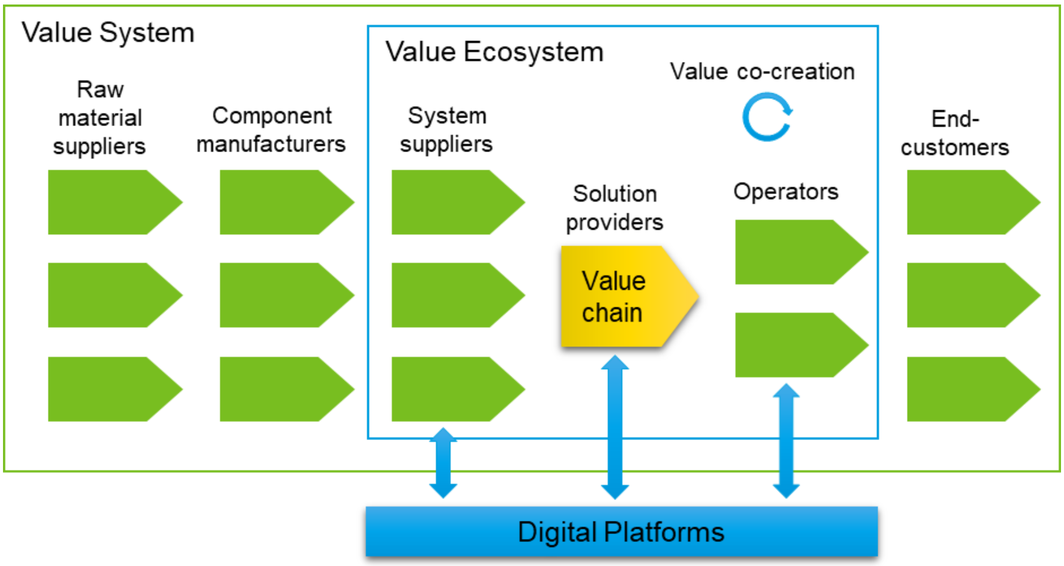


Figure 5. Digitalization effects throughout value system and value ecosystem. (Partly adapted from: Rabetino & Kohtamäki, 2018)

In the value system the role of digital platforms is to facilitate and enable connections between actors within a value ecosystem (Kohtamäki et al., 2019). Value ecosystem is a self-contained and self-adjusting alignment structure of mostly loosely coupled social and economic partners that need to interact for a mutual value proposition to materialize (Adner, 2017; Lusch & Nambisan, 2015). According to Kohtamäki et al. (2019), digital platforms work as an active agent in the value ecosystem capable of triggering or initiating service innovation process. With digital platforms, the value ecosystem actors participate in value co-creation process, which contains the processes and activities that facilitate the resource integration and incorporate different actor roles in the service ecosystem (Kohtamäki et al., 2019).

Digital platforms support the value co-creation by facilitating interactions among actors, adapting internal processes and helping the transparency of activities (Lusch & Nambisan, 2015). This way digital platforms play an enabling role ensuring that value co-creation process is efficient and effective inside the value ecosystem (Kohtamäki et al., 2019). The resulting ecosystem can thus generate synergy benefits where the co-created value is greater than the sum of individually created value (Yablonsky, 2018). This can be extended to manufacturing value chains since the digital platform can manage and orchestrate the entire product-service life cycle allowing manufacturers to develop, deploy and validate these highly customized offerings with an efficiency that other infrastructures would not allow for (Silva & Soares, 2021).

2.2 Digital business and service platforms

As discussed, digital platforms can be internal or extend operations beyond the boundaries of a single firm to the value ecosystems and to the entire value system. Cenamor et al. (2017) explained intraorganizational platforms to enable manufacturers to pursue both customization and operational efficiency and Eloranta & Turunen (2016) clarified that interorganizational platforms facilitate and orchestrate the complex relationships with customers and suppliers. The extension of digital platform into the stakeholders of an organization usually defines the characteristics required from the digital platform. As the digital platforms are used to facilitate the interaction between the value ecosystem actors, different stakeholders usually have different purpose-built platforms for the interaction. According to the definition of Gartner (2016), the digital business platforms can be divided into five distinct categories. The division is illustrated in figure 6. Gartner (2016) describes four different stakeholders of a company utilizing digital platforms which are external customers, internal employees, partners and Internet of Things. Based on this definition the five different platform types can be defined which are covered next.

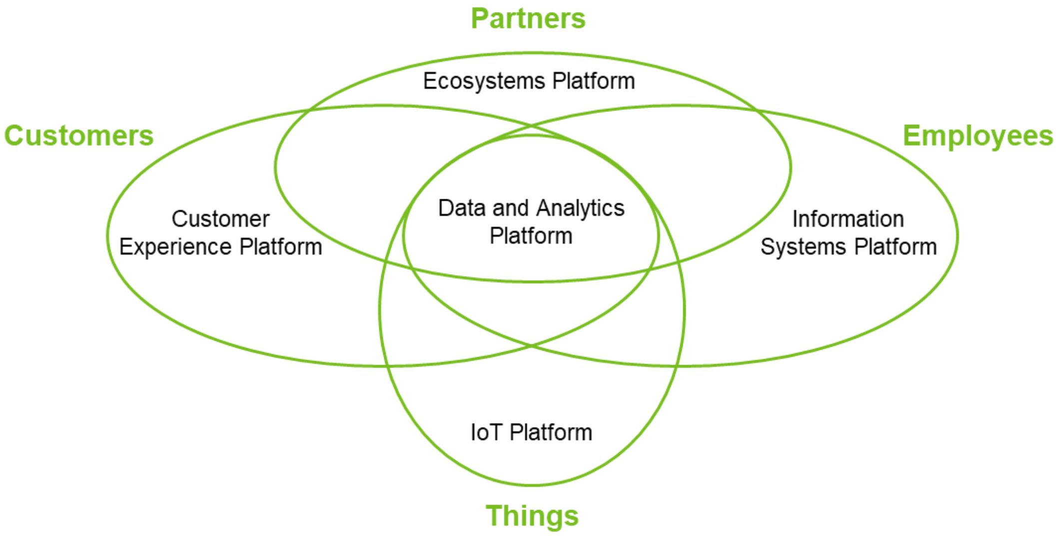


Figure 6. Different types of platforms (Adapted from: Gartner, 2016)

First platform type is data and analytics platforms, which are linked into all different platform types since data, analytics and algorithms are central to all digital business platforms (Gartner, 2016). A central theme of data and analytics platforms is the data fabric layer which facilitates data representation, storage, processing, access and exchange between other systems and digital platforms (Theodorou et al., 2021). The data fabric layer of data and analytics platforms can be realized using for example Data Lake technologies (Nargesian et al., 2019). Data and analytics platforms contain the information management and analytical capabilities, data management programs and analytical applications which accelerate data-driven decision making, automatic discovery and action (Gartner, 2016).

Second platform type is information systems platform, which has the function to support the back office and operations such as ERP and core systems (Gartner, 2016). Although organizations have already supported these systems, the shift towards value co-creation and value capture in platform ecosystems have incentivized enterprises to modernize and transform towards digital platform ecosystem strategy in internal information systems (Schreieck et al., 2021). According to Gartner (2016), information systems platform contains employee collaboration and workplace, back-office systems (finance, HR, purchasing etc.), core systems, supplier and customer portal and apps and business intelligence, endpoint computing and operational technology systems.

Third type of platform is a customer experience platform which contains the main customer-facing elements such as customer and citizen portals, multichannel commerce and customer apps (Gartner, 2016). These are also often called digital experience platforms (DXP) which are integrated set of technologies that engage users throughout their customer journey and provide seamless user experience across all user touch points (Shivakumar & Sethii, 2019). The main idea of DXP is that it manages the business processes and decreases production time which delivers maximum value to end customer with minimum cost of resources, and it can benefit in a digital business ecosystem with API based integrations to other complementing platforms and technologies (Shivakumar & Sethii, 2019). According to Gartner (2016), customer experience platforms contain multichannel interaction and commerce, social networks, customer analytics, customer portal and apps, and back-office systems.

Fourth type of platform is an Internet of Things (IoT) platform. IoT platform is a set of technology-enabled entities including physical smart objects and software services and systems that are connected and working together (Fahmideh & Zowghi, 2020). IoT platforms have the function to connect physical assets for monitoring, optimization, control and monetization to digital capabilities such as core information and operational technology systems (Gartner, 2016). The use of digital platforms is necessary for IoT since the data generated by these systems is large and the analysis of the data needs lots of processing power (Guth et al., 2018).

Final platform type is a ecosystems platform which has purpose of enabling partners to create value from the outside of the ecosystem in the digital world, with the ability to make assets like data, algorithms, transactions and business processes available through APIs to external business ecosystems (Gartner, 2016). Ecosystem platform consists usually three components including development platform, runtime platform and engagement platform (Patni, 2017). The purpose of ecosystems platform is that all other platforms will use the ecosystems platform to digitally interact and/or transact with the external world (Gartner, 2016). This way the digital platform ecosystem can be extended and integrated with organizations partners and other external stakeholders (Bondel et al., 2021).

2.3 Platform modularization and add-ons

From the perspective of this thesis, the key enabling characteristic of digital platforms is the modular structure of the digital platforms. Platform modularity allows digital platforms to drop or implement different combinations of products and services according to current business requirements allowing organizations to separately develop service solutions that can be then offered to the customers (Silva & Soares, 2021). Tiwana (2014) describes platform modularity as the decoupling of the platform and its add-on functionalities with the architecture in which the add-ons interact with the digital platforms. The modularity can be thought as Lego blocks (services) that can be easily rearranged to meet any need (Gartner, 2016). The openness and composite nature of a platform is ideally suited to the external-facing capabilities required by new digital business processes, moments and models (Gartner, 2016). This with the combination of physical goods, loosely coupled modules and standardized interfaces creates opportunities for new service offerings for manufacturing organizations (Silva & Soares, 2021).

The importance of the platform modularity in terms of servitization has previously been researched in academic literature. As already discussed, digital platforms allow organizations to configure and deploy service implementations which were previously not possible due to the level of customization and scale, and the flexibility of digital platforms has positioned them as the preferred infrastructure for developing a new paradigm of business models centered around customers, suppliers, and the developers' coalition (Silva & Soares, 2021). Therefore, to achieve the flexibility and configurability of the advanced service implementations required in the competitive markets, modularity is important especially in the domain of digital service platforms. Brax et al. (2017) discussed that the key features of modularity are customization and personalization of the service modules. Bask et al. (2010) discussed that the platform modularity allows leveraging service modules in combination of several configurations of product-service offerings. Silva & Soares (2021) mentioned that module decoupling allows organizations to optimize offerings by facilitating the creation of pre-defined sets of modules that make up a given service and make the customization process more agile.

However, the modular architecture of digital platforms has also disadvantages related to platforms and platform-based PSSs (Silva & Soares, 2021). Tiwana (2014) mentioned that modularity increases upfront of costs especially in digital platform design stages which may not lead to the operational efficiencies organizations expected. Ethiraj et al. (2008) discussed that the modularity might lead to an increased risk of imitation by the competitors which may decrease revenue of the add-on developers. According to Silva & Soares (2021), this may lead to an inefficient development of the digital platform ecosystem and even hinder the long-term feasibility of the digital platform.

2.4 Digital platform deployment challenges and solutions

Servitization and digital technologies have become a definitive factor for organizations to survive in competitive market environment. As already discussed, this has forced manufacturers to start implement new digital strategies and technologies into their operations to meet the increased customer demands and stay competitive in the market environment. However, digital platform deployment is a long and complicated process, and organizations are finding it difficult to implement digital platforms into their operations (Özcan et al., 2022). Academic literature has researched the challenges and critical success factors of digital platforms (e.g., Özcan et al., 2022) and more generally digital transformation (e.g., Shasi & Sinha, 2021; Hanelt et al., 2021; Brunetti et al., 2021). These challenges and the prominent solution for the platform implementation problems are described next in detail.

First challenge organizations face when implementing and deploying digital platform is the lack of clear strategy, objective or vision of the digital transformation in the organization (Shasi & Sinha, 2021). According to Shasi & Sinha (2021), without clear implementation strategy, the digital platform implementation will fail. Because the platform implementation usually affects the entire organization, the implementation strategy must be clearly defined to the entire organization. Henriette et al. (2016) suggests the implementation of pilot groups or prototypes in the deployment process to solve the issues in objective. With pilot groups, the digital platform can be deployed on a smaller scale and with less risk, and they can bring necessary results and launch dynamics of change more easily (Henriette et al., 2016). Likewise, Shasi & Sinha (2021) propose implementing digital platforms with incremental steps. Adopting agile methodology and breaking the big goals into smaller tasks improves the vision of the implementation thus increasing the probability of success (Shasi & Sinha, 2021). Thus, it can be concluded that to overcome the barrier of lack of clear implementation strategy, careful planning of the digital platform implementation and is important.

Second challenge which organizations regularly face in digital platform implementations is the cultural and skills related challenges (Brunetti et al., 2020). According to Brunetti et al. (2020), the key organizational challenge of digital platform implementations is in developing and updating the necessary digital skills for platform implementations within companies. Organizational culture can be problem as the mindset and cultural change can be resisted and the organizations might not let go of the legacy systems and processes easily (Shasi & Sinha, 2021). Hanelt et al. (2021) emphasizes the role of leadership in overcoming barriers in digital platform implementation related to the organizational culture and skills. The output of their research is that managers of the organizations should be more aware of the strategy processes and contents related to digital platforms, they should accelerate the intellectual capacity in the organization to continuously conceive digital processes and products, and finally they should harmonize the skills of the organization with regards to the digital platform process changes. Shasi & Sinha (2021) propose integrating and collaborating with business functions across the organization and engaging all key stakeholders to the implementation as a solution. Enabling knowledge and information sharing among all business functions will help the organization to work toward a single goal and by engaging the stakeholders to the change the organization will be more committed to the digital platform implementation (Shasi & Sinha, 2021).

Third problem organizations face in digital platform deployments are data and infrastructure challenges (Brunetti et al., 2020; Shasi & Sinha, 2021). The infrastructure challenges of the digital platforms are well acknowledged in the academic literature. Özcan et al. (2022) define the IT architecture to be one of the critical failure factors in digital platform implementations. According to them, digital platforms require standardization of interfaces, which enables platform providers to offer complementary services and standardized processes, which improves the transaction process of users. Hansrod et al. (2022) also define infrastructure as one of the key pitfalls of the digital platform implementation. They argue that the digital platforms must be deployed on a cloud platform because an on-premises solution will be much slower, less scalable and have higher operational and maintenance costs. Brunetti et al. (2020) discuss infrastructures and technologies challenges from information, interaction and AI point of view. Based on their research, continuously adapting to new digital technologies with innovative infrastructures and services is currently a challenge to organizations.

However, the data integration and interoperability related challenges and solutions of digital platforms have been left to a lesser emphasis in academic literature, although scholars are aware of the existence of these problems. Based on the recent academic literature, companies are more concerned with the security of the data (Shasi & Sinha, 2021) and data ownership (Müller, 2019) in digital platforms than with the integration to the existing business systems of the organization. Schmidt et al. (2019) mention data accuracy and exchange, and the interconnection between existing systems as main technological challenges industrial companies perceive when implementing digital platforms. Dillon et al. (2010) discusses that the key with cloud-based systems integration with the organizations existing systems is interoperability, which has the primary goal of realize seamless data streams across clouds and between cloud and local enterprise applications. They mention that propriety cloud APIs make it difficult to integrate cloud services with an organization's own existing legacy systems.

3 Programmable add-ons in digital platform ecosystems

This chapters gives a theoretical background to the programmable add-ons. This chapter explains the modularity and service-orientation from the software development point of view, application of add-ons in the digital platform and ecosystem domain, add-on development and brief introduction to system architecture design principles. The output of this chapter is the methodological framework which will be utilized on the empirical part of this thesis.

3.1 Modular software development architectures

In recent years, the software engineering domain area has undergone major technological developments such as mobile computing, cloud computing, DevOps and elastic computing. (Mazlami et al., 2017). The evolution of these digital technologies has been driven by the increased requirements of the software systems (Blinowski et al., 2022). For example, the accelerating progress of network speed, reliability and security has increased the demand to move the software and services from on-premises solutions to third-party managed solutions which are accessible through the network e.g., cloud computing solutions (Alshuqayran et al., 2016). The transition to cloud-based computing has in turn provided new opportunities to deploy scalable applications in an efficient way, allowing enterprise applications to dynamically adjust their computing resources on demand (Villamizar et al., 2015). This development has further created the need to create new software development methods and software architectural styles that fit the cloud computing architecture (Alshuqayran et al., 2016).

Traditionally software systems are built using monolithic approach (Lewis & Fowler, 2014). A monolith is a software application whose modules cannot be executed independently (Dragoni et al., 2017) as the application usually uses a single large codebase that offers several services using different interfaces (Villamizar et al., 2015). Currently many organizational capabilities and functions are underpinned by large monolithic software systems, including enterprise resource planning (ERP) and customer relationship management (CRM) systems which limits the application upgradeability and maintainability

(Baškarada et al., 2020). From the system's point of view, as the application's size and complexity grow, problems start to arise (Blinowski et al., 2022). According to Blinowski et al. (2022), a growing monolithic software system comes with three main problems. First, changes in one module may lead to unexpected behavior in other modules and a cascade of errors. Second, size of the monolith results in longer start-up time, which in turn slows down the development and becomes an obstacle to continuous deployment. Third, it is increasingly harder for the development team to keep changes that related to a particular module to only affect this very module, and in effect, to retain a modular structure of the application. From organizational perspective, the monolithic software systems may slow organizational ability to quickly respond to changes in the environment, e.g., changes in customer and competitor behaviors which hinders organizational agility and competitiveness (Baškarada et al., 2020).

However, the rise of the cloud services such as Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) solutions has resulted in increased questioning of the monolithic software development approach. One of the main reasons companies must move applications to IaaS or PaaS solutions is to gain efficiency in their operations trying to be able to scale enterprise applications on-demand and supporting peak periods (Villamizar et al., 2015). Scaling monolithic applications is a challenge because if popular services or functions need to be scaled because they are highly demanded, the full set of services also will be scaled at the same time (Villamizar et al., 2015).

The limitations and problems of monolithic software systems in cloud environments have been overcome with service-oriented architectures. Currently a commonly used service-oriented software engineering architecture is microservices architecture (Hannousse & Yahiouche, 2021). A microservices architecture is a cloud-native architecture that aims to realize software systems as a package of small services (Balalaie et al., 2016). Microservices are small autonomous service components deployed independently, with a single and clearly defined purpose and accessed through a REST (representational state transfer) interface (Lewis & Fowler, 2014). They decompose applications vertically into subset of business-driven independent services where each service can independently be developed, deployed and tested by different development teams (Auer et al., 2021). Each such microservice runs its own processes and communicates with lightweight mechanisms.

Microservices have a variety of different advantages compared to monolithic architecture. Because software processes and microservices build applications and application components are explicitly linked and not invoked according to workflow, SOA-like discovery and service repositories are not relevant (Lewis & Fowler, 2014). Microservice discovery and binding is instead defined to support scalability under load and resilience, which is one of the key properties of software in the cloud (Lewis & Fowler, 2014). From software development point of view, microservices can be developed in different programming languages, can scale independently from other services, and can be deployed on the hardware that best suits their needs (Auer et al., 2021). Also, microservices are easier to maintain and more fault-tolerant since the failure of one service will not disrupt the entire system (Auer et al., 2021). Lastly, microservices can be shared among applications, and cloud providers may offer serverless pay-as-you-use models for microservice hosting (Lewis & Fowler, 2014).

Another type of cloud-native software development architecture is a serverless computing architecture. Serverless computing is an event-driven application design and deployment paradigm in which computing resources are provided as scalable cloud services (McGuire, 2019). In serverless computing, functions (modules) of the application will be executed on demand without requiring the application to be running all the time (Varghese & Buyya, 2017). Like PaaS, developers can write arbitrary code and are not limited to using a prepackaged application. The version of serverless that explicitly uses functions as the deployment unit is also called Function-as-a-Service (FaaS). (Baldini et al., 2017) With the adoption of event-based platforms for enabling serverless computing, more applications will make use of FaaS. Examples of platforms that currently support serverless architecture includes AWS Lambda, IBM OpenWhisk and Google Cloud Functions. (Varghese & Buyya, 2017)

3.2 Platform ecosystem, architecture and programmable add-ons

Following the definitions of microservices and serverless computing architecture defined in the previous chapter, the concept of programmable add-ons is defined. In the scope of this thesis, the programmable add-ons can be defined as microservices or serverless functions, which are applied in the digital business platform domain to enhance the platform functionality. Together the digital platform and the add-ons included in it compose platform ecosystem (Tiwana, 2014). The core elements of a platform ecosystem are defined in the table 1 below.

Table 1. Core elements of a platform ecosystem (Source: Tiwana, 2014)

Element	Definition	Example
Platform	The extensible codebase of a software-based system that provides core functionality shared by apps that interoperate with it, and the interfaces through which they interoperate	iOS, Android Dropbox, Twitter AWS Firefox, Chrome
App	An add-on software subsystem or service that connects to the platform to add functionality to it. Also referred to as a module, extension, plug-in, or add-on	Apps Extensions
Ecosystem	The collection of the platform and the apps specific to it	
Interfaces	Specifications that describe how the platform and apps interact and exchange information	APIs Protocols
Architecture	A conceptual blueprint that describes how the ecosystem is partitioned into a stable platform and a complementary set of apps that are encouraged to vary, and the design rules binding on both	

As defined in the chapter 2, digital platforms are software-based products or services that serve as a foundation on which the interaction between actors is conducted. The technical definition of digital platforms also emphasizes the modularity of platform which means that the platform functionality can be extended with complementary product- or service-modules built by the platform provider or a third-party developer (Tiwana, 2014). Digital platforms incorporate various of these modules to extend the original functionality of the digital platform. These complementary product- and service-modules have multiple names in the academic literature, but by the definition of Tiwana & Konsynski (2010), these complementary modules are called "add-on software subsystems". Another definition by Ghazawneh & Hendfridsson (2013) defines these modules as "executable pieces of software that are offered as applications, services or systems to end-users". The definition of Tiwana (2014) refers to these modules as applications or apps which connects to the platform to extend its functionality. In the scope of this thesis, these complementary product- and service-modules are called "programmable add-ons" or "add-ons".

Extending the definition of Tiwana (2014), add-ons access and build on the functionality of the platform through a set of interfaces that allow them to communicate, interact, and interoperate with the platform. The collection of the platform and apps that interoperate with it represents the platform's ecosystem. Finally, the ecosystem architecture is a blueprint that describes how the ecosystem is partitioned into a stable platform and a complementary set of add-ons. The concepts of digital platform ecosystem, platform architecture and add-on microarchitecture are conceptualized in the figure 7 below.

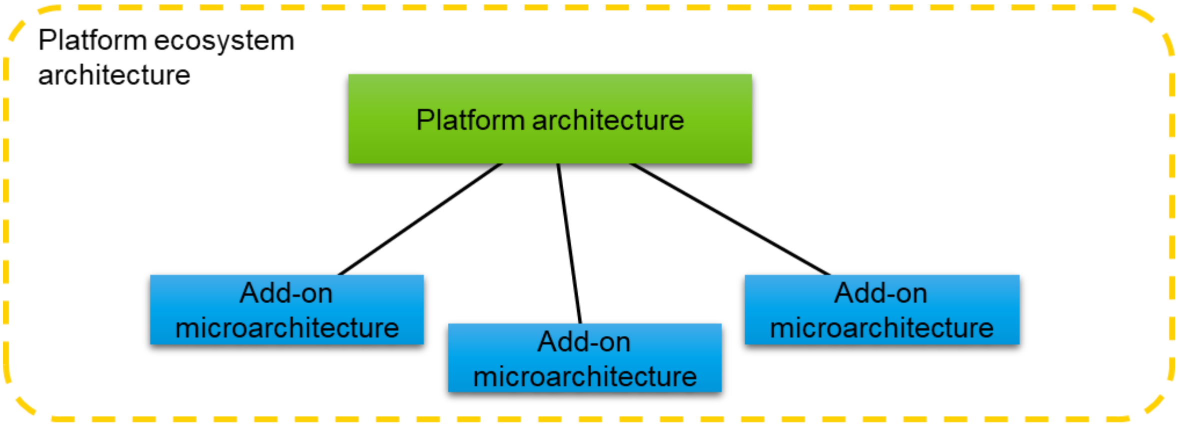


Figure 7. Platform architecture and add-on microarchitectures.

The platform and its complementary add-ons form a platform ecosystem architecture (Bresnahan & Greenstein, 2014). The definition by Tiwana (2014) divides ecosystem architecture into the digital platform architecture and the individual microarchitectures of the add-ons. From this perspective, the platform architecture is the foundational functionality of the digital platform ecosystem, which is made available to the add-ons through a set of interfaces. Although the platform has a specific architecture that all add-ons see and can access, the architecture of individual add-ons within the same platform can be different from one add-on to another. Therefore, an add-on's microarchitecture will describe for an individual add-on, how it interacts, communicates, and interoperates with the digital platform. (Tiwana, 2014) As a result of the loose coupling functionality of the digital platform ecosystem, the add-on's microarchitecture does not need to follow the architectural pattern of the digital platform (Hein et al., 2018). This results in flexibility, openness and spontaneous of the development of programmable add-ons (Orton & Weick, 1990). The add-on's microarchitecture is free to be decided by the add-on developer, as long as the communication between the platform and the add-on is agreed and unified (Tiwana, 2014).

According to the definition by Tiwana (2014), the add-on's internal microarchitecture is the description how the functional components of add-ons are divided between the digital platform and the add-on. In this definition, the components are divided into four distinct parts. The first part of the add-on's microarchitecture is the presentation logic, which is concerned with handling user interaction and updating the view of the add-on as presented to the user (Couloris et al., 2005). Second part is the application logic or business logic which is the detailed add-on -specific processing associated with the add-on that makes it uniquely valuable (Couloris et al., 2005). The third part is the data access logic which is the processing required to access and retrieve data through for example data base queries (Tiwana, 2014). The final part is the data storage which defines in where the data processed by the add-on is stored and retrieved (Tiwana, 2014).

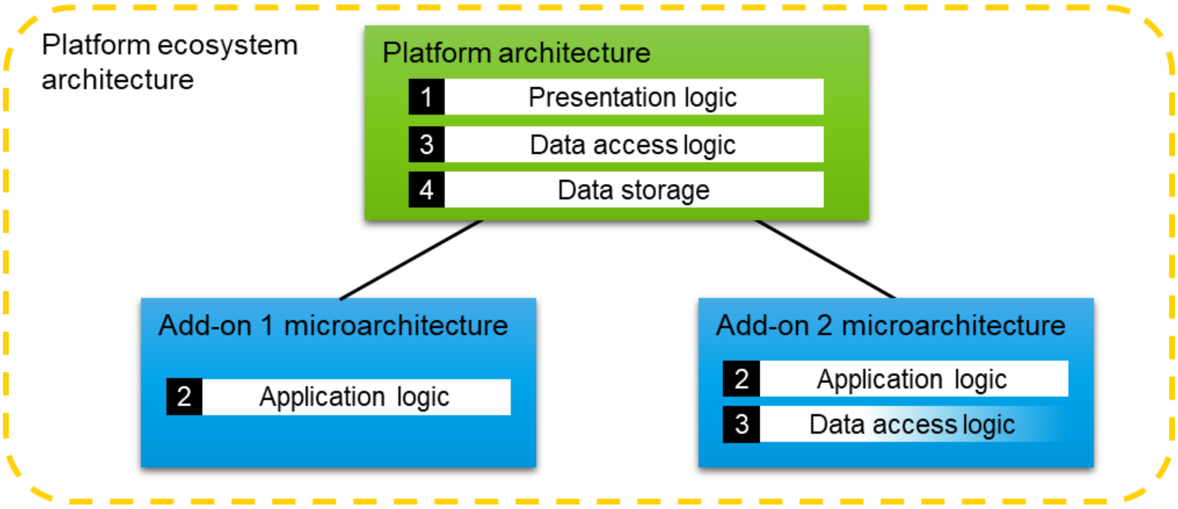


Figure 8. The functional elements of add-on microarchitecture can be flexibly allocated to platform or add-on. (Adapted from: Tiwana, 2014)

The unique aspect of the platform architecture and the add-on microarchitecture is that for each individual add-on, the functional elements of the add-on can be flexibly allocated to the platform architecture and add-on microarchitecture (Tiwana, 2014). Figure 8 above illustrates this characteristic. In the figure, add-on 1 is only responsible of the application logic and add-on 2 is responsible of the application logic and partly of the data access logic as the data access responsibility is shared with the digital platform. The flexible allocation of functional elements means that the add-on developer has freedom to choose, which parts of the platform functional elements the add-on utilizes, and which parts are implemented inside the add-on (Tiwana, 2014). The modularity of the connections between the add-on and the digital platform allows the add-on to implement only minimal amount of logic while relying on the functional logic of the digital platform.

The concept of “programmable add-ons” used in this thesis extends directly from the definition of Tiwana (2014). Basic idea of programmable add-ons in the scope of this thesis is that the programmable add-on should only handle minimal number of functional elements of the add-on microarchitecture presented above. This way the programmable add-on extends from the already implemented functionality of the digital platform and only adds the business logic to the digital platform which is needed by the user. This approach has two key benefits. First, the development of these add-ons is fast since most of the add-on logic is handled by the platform. In addition, since setting up of the needed infrastructures, managing data integrity across different environments, and enhancing the robustness of the system is handled by the Platform as a Service (PaaS) provider (Sahay et al., 2020), the developers can focus on capturing the underlying business logic into the programmable add-on. The add-on development can be conducted using agile software development principles such as continuous integration and development, since the programmable add-ons are only loosely coupled with the digital platforms (Hein et al., 2018).

Second benefit of the programmable add-ons is that developing the programmable add-ons is easier compared to full application development since the developer needs to only capture the business logic into the application. The programmable add-ons offer potential for a citizen development approach. Citizen development is a software development approach that uses the end-users of the platform, who has no extensive programming background, contribute to the software development process (Sahay et al., 2020). The utilization of citizen developers has especially been on the rise due to low-code development platforms (Lebens et al., 2022). Low-code and no-code (LCNC) development platforms are software platforms that enable developers of different domain knowledge and technical expertise to develop full applications ready for production (Sahay et al., 2020). Low-code platforms allow citizen developers to build fully operational applications without the knowledge of front- or back-end development. (Lebens et al., 2022). Thus, citizen developers can focus on the business logic of the application rather than dealing with additional details of setting up of the needed infrastructures, managing data integrity across different environments, and enhancing the robustness of the system. (Sahay et al., 2020)

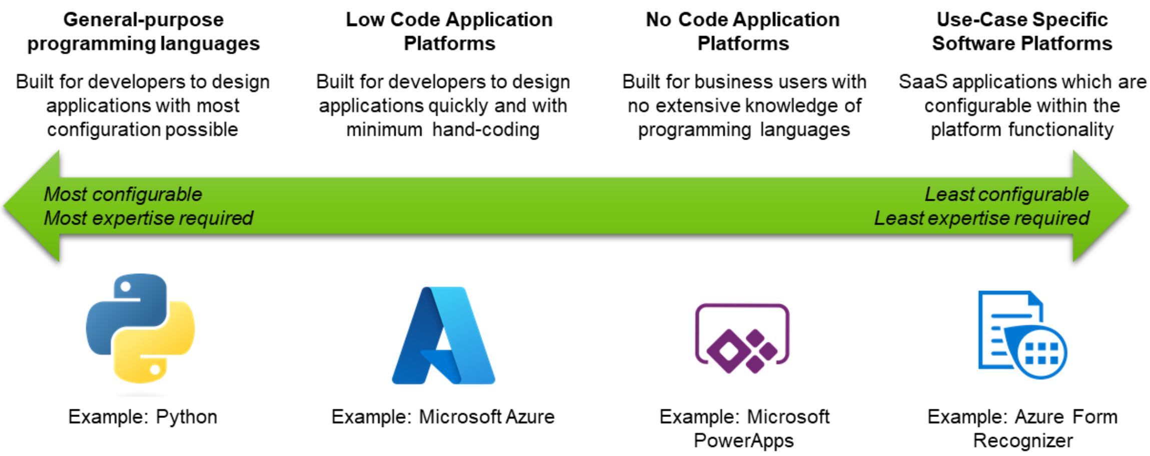


Figure 9. Different options for programmable add-on development. (Adapted from: Wasserman, 2020)

Figure 9 illustrates the spectrum of software development platforms which can be utilized for programmable add-on development. The least configurable option for programmable add-on development is the use case specific software platforms (Wasserman, 2020). These platforms are solutions, which have already a configured use case or a domain area for which the platform is designed for. One example of these types of tools is Azure Form Recognizer which is an AI service applying machine learning to extract text or key-value pairs from documents. The machine learning model training is done entirely via Form Recognizer studio and no coding expertise is needed. (Microsoft, 2022a)

LCNC platforms are in the middle of the spectrum and there exist many different variations of the LCNC platforms in the markets (Sahay et al., 2020). For example, Microsoft Power Automate offers low-code drag-and-drop tools for developing for example workflow automation solutions. Azure also offers some low-code applications, but these are more directed to developers compared to the MS Power Tools, which are directed to business users (Microsoft, 2022b). Citizen developers can add machine learning capabilities with Azure Machine Learning Studio, which offers AutoML and drag-and-drop tools for machine learning model building and training. Finally, the most configurable option for programmable add-on development the general-purpose programming languages, of which one example is Python.

3.3 Add-on deployment and platform integration architectures

The academic literature has researched different approaches of digital platforms and its add-ons implementation into an organization. Based on a paper by Liu et al. (2022), add-ons for digital platform integration can be divided into three categories including layered architecture, microservices and event-driven architecture. Each approach has benefits and drawbacks to the integration of the add-ons and to the characteristics and performance of the ecosystem.

Layered architecture divides the functional components of digital platforms into several layers which have related functionality and are stacked on top of each other (Chavan et al., 2015). The layered architecture is commonly used in platform ecosystems. Coito et al. (2020) discussed a middleware layer for an intelligent industrial automation which acts as a gateway between IoT devices and the digital platform. Terroso-Saenz et al. (2019) used layered architecture for an energy data management and analysis IoT platform. Layered architecture is best suitable for applications that comprise distinct classes of services that can be arranged hierarchically (Chavan et al., 2015). However, the layered architecture style limits flexibility and scalability of the layered systems when designing large-scale open distributed systems (Liu et al., 2022).

Second type of programmable add-on integration architecture is microservices architecture which uses the service components a small autonomous service components deployed independently, with a single and clearly defined purpose (Lewis & Fowler, 2014). Innerbichler et al. (2017) used microservices based architecture style to enable IoT-based real-time monitoring, optimization and automated negotiations in manufacturing supply chains in a collaborative platform. The benefits of the microservices are the scalability of the system but Liu et al. (2022) mentioned that the microservices architecture does not naturally lend itself to support high-performance data-intensive applications due to the distributed nature of microservices and the potential need to replicate databases across services.

Third type of programmable add-on integration architecture is the event-driven architectural style. Event-driven architectures are designed to process events as the central architectural concept (Dunkel et al., 2011). One event-driven architecture used in digital platforms is serverless architecture. Keshavarzian et al. (2019) proposed for a human activity recognition IoT platform a FaaS cloud model architecture, which solves scalability issues by running each function in a separate container. Verba et al. (2019) suggested a Fog-of-Things platform and approach to measuring and estimating the runtime parameters and migration benefits of applications deployed in edge devices that carry out computations. Benefits of event-driven architectures are that they are designed to handle continuous stream of events (Dunkel et al., 2011). However, according to Liu et al. (2022) event-driven architecture is challenging to test since the entire system has many interactions although each component of the platform can be tested individually.

4 Research methodology

This chapter describes the research methodology of this thesis in detail. This chapter contains theoretical background to the used research methods, the plan for conducting the empirical research, and the measures which will define the performance of the developed programmable add-ons during the research. The output of this chapter is the framework in which the empirical study will be conducted.

The empirical research in this thesis was conducted as a two-part case study. Case study is a scientific research method, where a particular case is studied intensively to understand a larger group of cases (Gerring, 2004). Case study research method usually involves collecting research material using several methods such as interviews, observations, discussions and studying documents (Hirsjärvi et al., 2009). The case examined in this thesis was applying the programmable add-on methodology in a manufacturing organization's digital service platform implementation. The empirical research combined both qualitative and analytical research methods to find the answers for the research questions of the thesis. The structure for conducting the empirical research is visualized on the figure 10 below.

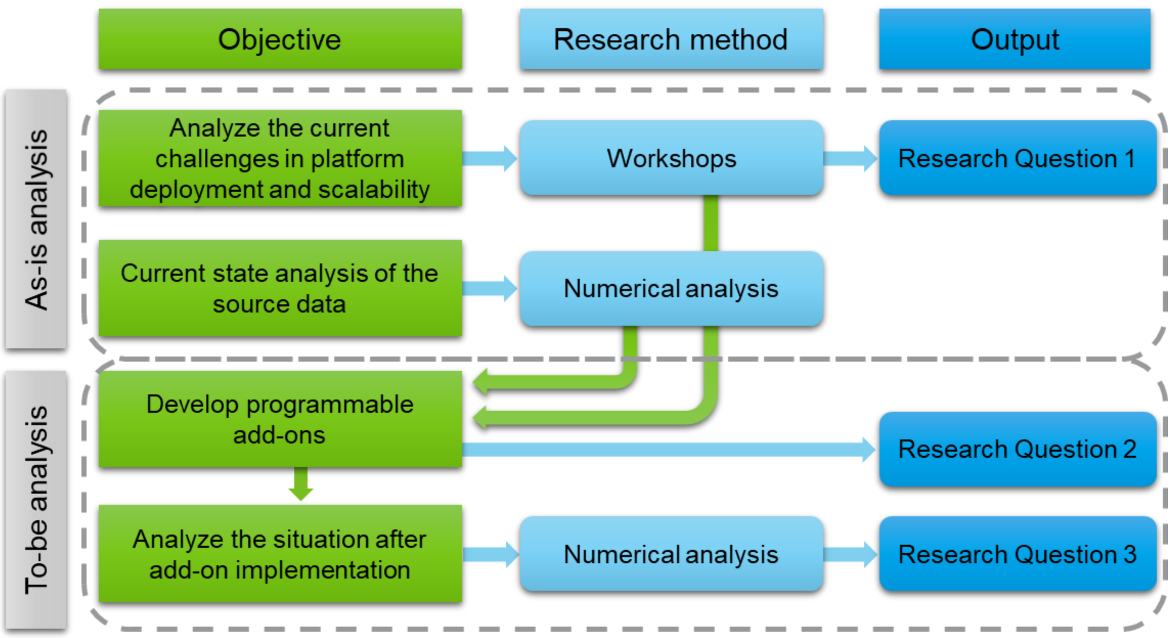


Figure 10. The structure of the empirical study.

The empirical study began with the as-is analysis of the digital platform deployments. The qualitative research data of the deployment and implementation teams' challenges was collected with workshops. Workshops are a research method where a group of people focus on the study of a domain-specific issue (Ørngreen & Levinsen, 2017). Workshops as a research method are becoming more commonly used, and it is often used in the information systems and design fields to evaluate artifacts or to co-create business innovations (Thoring et al., 2020). Workshops can be facilitated as online meetings which makes them especially useful research method for remote and online based teams (Shamsuddin et al., 2021). This was the case with the platform deployment teams in the scope of this thesis and thus the workshop method was used to collect the research data.

The workshops were facilitated for the platform deployment teams as online meetings, where they could present the problems which they currently faced in the deployment. Focus of these workshops were examining the current situation of the platform implementations and where the deployment teams were facing most problems related to the data integrations or business processes. Second discussion point of the workshops was to ideate the practical solutions for the problems. The theme in which the solution discussion was facilitated was ideating and discussing possible "black box" solutions to these problems. Therefore, the discussions based on the workshops worked as an input to the problems in digital platform deployment and for the designing and development of the programmable add-ons.

The second part of the current state analysis and the to-be analysis after the programmable add-on implementation were conducted as a numerical analysis. Numerical analysis is the study of algorithms that use numerical approximation to analyze the problem (Gautschi, 2012). In the scope of this thesis, numerical analysis is used to analyze the performance and efficiency of the programmable add-ons. The data correction rate is measured using the numerical approximation of the effect of the programmable add-on. The improvement to the business process is measured directly from the process's, throughput time. The performance of add-ons was measured by analyzing the source data before and after the add-on implementation.

The programmable add-ons were measured with two different metrics: data accuracy metrics and process impact metrics. The impact of the add-on to the data was measured with data accuracy (1). Data accuracy represents the number of correct cases in the sample divided by the number of sample cases. The formula for measuring the data accuracy is described in the formula 1 below.

$$accuracy\ (\%) = \frac{number\ of\ correct\ cases}{number\ of\ cases} \quad (1)$$

For machine learning based add-ons, the accuracy of the model was instead measured with a model confidence measure. The model confidence represents the probability in which the machine learning model will output the correct results. The machine learning model accuracy results were used to validate the performance of the add-ons, but since the goal of the only add-on using machine learning methodology has purpose to increase the process throughput, this result can't be compared to the as-is process and thus it is used only to validate the programmable add-on.

The process improvement metrics measure the programmable add-ons' impact to the business process. The metric for measuring the process improvement is the process throughput time. Process throughput time is measured from the beginning to the end of the measured process part. The duration of the process part formula (2) is described below.

$$duration = process\ end\ time - process\ start\ time \quad (2)$$

The workshop research method was chosen for the current state analysis because the platform deployment teams were from different countries and the meeting had to be facilitated through Microsoft Teams. From the perspective of the digital platform deployment and the challenges related to it, it was important to facilitate the discussion so that the group could collaboratively contribute and discuss the topics. However, the research data collected from a workshop method can be more time consuming and harder to analyze than for example a questionnaire, but on the other hand, workshops can provide more in-depth understanding of the current state of the challenges.

The numerical analysis of the programmable add-ons was used to measure the effectiveness of the programmable add-ons. The purpose of numerical analysis was to provide fact-based evidence which would prove the effectiveness of the programmable add-ons. Because this thesis is limited to a single company and two digital platform implementations, this method generally produces results that can't be generalized beyond the scope of this thesis, but on the other hand, the numerical analysis proves the effectiveness of the programmable add-ons in the target organization which can be used to draw the conclusions in this thesis.

5 Digital platform deployment and scalability challenges

This chapter describes the challenges in digital platform deployment in the scope of this thesis. This chapter contains the introduction to the case company, the background of the empirical study and the findings of the deployment and scalability problems in the deployable platforms. The output of this chapter is the challenges which the platform deployment teams were facing in the deployment projects.

5.1 Introduction to case company

Stora Enso is a leading provider of renewable products in packaging, biomaterials, wooden construction and paper, and one of the largest private forest owners in the world (Stora Enso, 2022a). Stora Enso employs around 22 thousand people around the world, and it consists of six different divisions including Packaging Materials, Packaging Solutions, Biomaterials, Wood Products, Forest and Paper. The case study of this thesis is conducted for Stora Enso Packaging Materials (PM) division, which is the largest division of Stora Enso with 3,89 bn EUR net sales in 2021 (Stora Enso, 2022a). The units and annual production capacities of PM division are listed in the figure 11 below.

Packaging Materials

Consumer board	Location	Grade	Capacity 1,000 t
Beihai	CHN	LPB, CKB, FSB, FBB	525
Fors	SWE	FBB	455
Imatra	FIN	SBS, FSB, FBB, LPB, CUK	1,195
Ingerois	FIN	FBB	295
Skoghall	SWE	LPB, CUK	925
Total			**3,395**

Containerboards	Location	Grade	Capacity 1,000 t
Heinola	FIN	SC fluting	300
Ostrołęka	POL	Testliner, PfR fluting, sack paper, wrapping paper, RCF-based liner and fluting	780
Oulu	FIN	Kraftliner, white-top kraftliner	450
Varkaus	FIN	Kraftliner, white-top kraftliner	405
Total			**1,935**

Barrier coating	Location	Grade	Capacity 1,000 t
Beihai	CHN	Barrier coating	80
Skoghall (Forshaga)	SWE	Barrier coating	120
Imatra	FIN	Barrier coating	455
Total			**655**

Figure 11. Stora Enso Packaging Materials production units and annual production capacities (Source: Stora Enso, 2022a)

Stora Enso PM division has 9 production units in 4 different countries including Finland, Sweden, Poland and China. PM division has 5 units in Finland (Imatra & Ingerois), Sweden (Fors & Skoghall) and China (Beihai) producing consumer board products with total annual production capacity of 3,4 million tons. 4 of the units in Finland (Heinola, Oulu & Varkaus) and Poland (Ostrołeka) are producing containerboard products with total annual production capacity of 1,9 million tons. Finally, the PM division has barrier coating capacity in China (Beihai), Sweden (Forshaga) and Finland (Imatra) for around 655 thousand tons of annual production capacity. (Stora Enso, 2022a) The strategic business units and the product lines of Stora Enso PM division are listed in the table 2 below.

Table 2. Strategic business units (SBUs) and product lines of Stora Enso Packaging Materials division (Stora Enso, 2022b).

SBU	Product Lines
Consumer Board	**FBB & SBS:** Cosmetics, chocolate, cigarette, pharmaceutical and food packaging, laminated boxes for wine and spirits and graphical products **Liquid Packaging Aseptic & CKB:** Liquid packaging board for aseptic packaging and coated unbleached kraft board **Liquid Packaging Fresh & FSB:** Liquid packaging board for fresh products, cupstock for paper cups and tray board for food packaging
Containerboard	**RCP CtnB, Sack & MG Kraft:** Corrugated board, industrial paper sacks, grocery/bread paper bags and e-commerce bags **Kraftliners:** Corrugated packaging, fresh food and agricultural packaging, e-commerce packaging, packaged food packaging, consumer products packaging and industrial packaging **SC Fluting:** Corrugated packaging, fruits and vegetables, other food packaging and various heavy-duty applications
China	**Consumer products:** Carton boxes, rigid boxes, corrugated boxes, paper bags and user manuals and moulded fiber

The case and the empirical setting of the study is in the Supply Chain (SC) organization of the Stora Enso PM division. Stora Enso PM Supply Chain organization provides design, planning and fulfilment services to the end-to-end (E2E) supply chain. The supply chain is responsible for designing and planning the production of the end products and intermediate products and designing the logistics and warehousing of the production. PM Supply Chain organization has currently 5 different teams in Service Design, Master Planning, Customer Fulfilment, Supply Chain Excellence and PM Logistics. (Stora Enso, 2022b)

5.2 Empirical background of study

The empirical setting of the study is linked to a larger development effort currently ongoing in the target organization. During the empirical study, Packaging Materials SC is transforming its operations into a new customer focused and data-driven operating model which improves the efficiency, scalability and customer service of the supply chain operations. This transformation is a strategic and holistic transformation to the entire end-to-end (E2E) supply chain process. From the perspective of the theoretical framework of this thesis, the operating model transformation follows the servitization of manufacturing value chains designed to offer differentiated and consistent services to customers. While the operating model transformation targets the ways of working, the new operating model also implements modern digital tools and service channels to improve the customer experience and service delivery.

To support the operating model transformation, Packaging Materials SC is deploying several technical enablers as a basis to support the expansion of the operating model transformation. These technical enablers are digital platforms and technologies that facilitate the new service-oriented operations and operating model in the SC organization. The technical enablers allow new differentiated service deliveries, the automatization of current digital workflows and value co-creation between the organization and the customers. From the technical enablers supporting operating model transformation, two supply chain service platforms are selected to the scope of this case study. For the purposes of this thesis, these platforms will be called “Platform A” and “Platform B”.

Platform A is a customer service and experience platform that manages the interaction between the supply chain organization and the customer. The platform is used for managing customer contacts and communication, and it offers possibility for different digital touchpoints and channels to best suit the customers’ needs. The goal of this platform is to increase the customer contact resolution performance and to increase the customer focus in the front-line customer excellence team. The low value-add contacts can be moved from assisted channels to self-service channels while all exceptions and special inquiries get routed to process expert and service support specialists in Platform B. The digital platform is deployed and scaled to the sales office of one geographical area at a time and thus the scaling area of the digital platform is geographical area.

Platform B is an internal execution platform that is used to facilitate the expert and specialist work in the supply chain service area. The platform is used to increase the capability for process monitoring, modeling, execution and automatization. The platform monitors the E2E internal execution of the supply chain, and the specialists and experts working on the platform handle and solve both customer service and internal execution exceptions occurring in the supply chain. Platform B is deployed and scaled through one internal process area at a time. The platforms in the scope of this empirical study are summarized in the table 3 below.

Table 3. Descriptions of platforms in the scope of this thesis.

	Platform A	Platform B
Purpose	Function as the interface and facilitate the communication between sales and supply chain coordination and customers	Facilitate expert and specialist work in the supply chain service area, monitor the internal execution of supply chain processes
Users	Sales coordinators, supply chain coordinators	Process support specialists
Service area	Customer	Internal execution
Scaling area	Geographical area	Internal process area

The deployment of these digital service platforms was in the early phases of deployment during the thesis. The scaling area 1 of the platform deployments was ongoing and area 2 was in the planning phase during the conduction of the empirical research. Due to time constraints, the scope of the thesis is limited to the area 1 deployment challenges the platform deployment teams faced during the implementation. The deployment teams were separate between the two platforms in the area 1 deployment which is why the challenges of the deployments were researched separately between the two platforms. The deployment and integration challenges in the scope of this thesis are described in the next chapters.

5.3 Deployment and integration challenges in Platform A

Platform A deployment and integration challenges were identified based on the workshop meeting with the platform deployment team. Based on the workshop, most of the challenges related to the Platform A deployment area 1 were in migrating the current processes and working practices into the new platform, which could be solved by the deployment team itself and did not need any more effort. However, the Platform A deployment team had a challenge migrating the customer contacts from the customer relationship management (CRM) system to the Platform A, which disabled a key function of the Platform A. Platform A has a functionality to make and receive phone calls through the platform which the target organization was planning to utilize in the customer communication. However, to enable this functionality, the phone numbers needed to be entered in E.164 format into the platform. E.164 is a standardized format for international telephone numbers that defines the structure for writing and storing phone numbers. This structure is visualized in the figure 12 below.

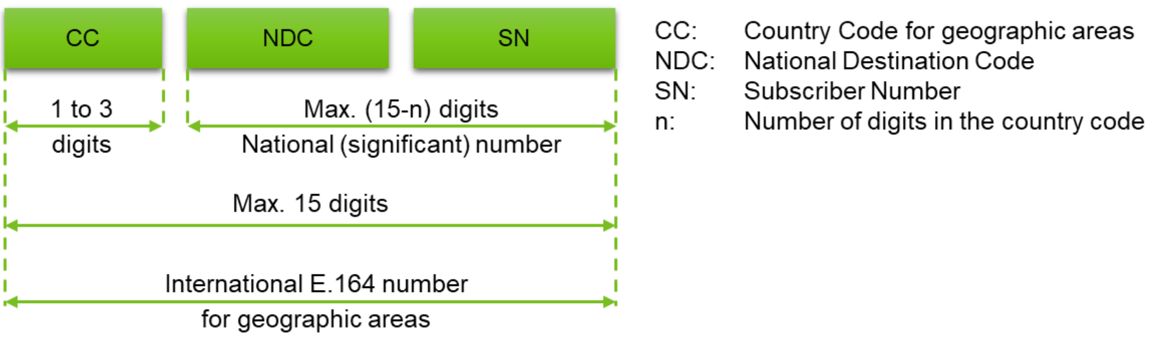


Figure 12. E.164 standard for international phone numbers. (Adapted from: Gódor et al., 2008)

In Platform A deployment area 1, the phone numbers were extracted from the CRM system as a table since the platform allowed importing multiple contacts from a table at the same time. The table extracted from the CRM contained around 42 thousand contacts each with office phone or mobile phone number or both. This resulted the total number of transferrable phone numbers be around 50 thousand. The challenge which the implementation team faced was that the phone numbers in the CRM database were not standardized or required to be in any format and thus they could not be directly imported into the Platform A. Because the CRM system did not enforce the phone number data to be in any specific format, only small portion of the phone numbers was in the E.164 format. In addition, some of the phone number fields contained free text information and multiple phone numbers with different endings, which made transforming the phone numbers challenging. A generated sample from the CRM contacts database is visualized on the figure 13 below. The information in the table is not the factual information extracted from the CRM database, but the figure illustrates the structure and formatting “Office Phone” field which had phone numbers in multiple different formats. The formatting of the office phone field is preserved in the illustration.

Full Name	First Name	Last Name	Primary Account	Office phone
Browne, Allen	Allen	Browne	Micro Design	+3856287860
Cypert, Timothy	Timothy	Cypert	Jacobs	+33 72 252 7012
Pierson, James	James	Pierson	Poore Simon	+40 4421 906110
Porter, Bridgett	Bridgett	Porter	Bombay Company	+50-24-422-9441
Camp, Tanya	Tanya	Camp	Garden Guru	81 63 153 7239
Yost, Rodney	Rodney	Yost	Smitty	+48 (0) 20 178 05 02
Collins, Reina	Reina	Collins	Hanover Shoe	+7
Parker, Arlen	Arlen	Parker	Opticomp	+7(495) 714-69-73 доб. 477
Smith, Karen	Karen	Smith	National Auto Parts	+1 (817) 479-1732
Patel, Terrance	Terrance	Patel	Waccamaw	+210208955836 ext. 9464
Bonanno, Bernice	Bernice	Bonanno	Mission You	+90 13 376 24 71 wew. 60
Roberts, Jordan	Jordan	Roberts	Star Interior Design	0033 (0) 218341685

Figure 13. A generated sample from the CRM contacts database using the formatting of the phone numbers.

The phone numbers could be formatted by hand by the platform deployment team, but as the database contained over 50 thousand phone numbers, this process would have taken an exceptionally long time for the deployment team and delayed the area 1 deployment. Thus, the deployment team searched for a programmatic approach for entering the phone numbers into the Platform A in the correct format.

5.4 Deployment and integration challenges in Platform B

Like with the Platform A, the challenges of the deployment team of Platform B were identified with a workshop meeting. Based on their input, three challenges related to platform deployment or scaling capability were picked to the scope of this thesis. These challenges are described in detail in the next chapters.

5.4.1 Route planning data

The first challenge the Platform B deployment team described was in the order transportation route planning data integration. The ERP system in the target organization has multiple predefined routes between each mill-customer pair. Usually, one of the predefined routes is the primary route which is the most cost-effective way of transporting the product from the mill to the customer. The rest of the predefined routes are alternative and more quicker routes if for example the requested time of arrival (RTA) of the customer can't be fulfilled with the primary route.

When a sales coordinator enters an order into the ERP system, the sales coordinator chooses a "planned route" from these predefined routes which is then used as basis for the logistic planning for this order. In Platform B deployment area 1 the planned route is also used as a reference route for mapping the document process into the real-life transport process. When a new transport document is imported to the Platform B, it is mapped to the planned route. This way the progress of the order in the supply chain can be monitored and the product arrival time to the customer estimated.

When the order is produced, a dispatcher reserves or books the necessary transports legs for the order from the mill to the customer. However, the booked transport route does not always correspond to the planned transport route which causes challenges in connecting the transport document process into the logistic process. This poses a challenge for the internal execution sensing and monitoring: How can the specialists in the Platform B define if the order is still following the intentional but not planned route or if the order is not on the intentional route (= exception in supply chain).

5.4.2 Ocean shipping document handling

The second challenge which the Platform B deployment team faced was the ocean shipping document handling. To enable the document process tracking in area 1 deployment, the information from the ocean shipping documents needs to be extracted and linked to the original document. This was established for the Platform B purposes on Microsoft SharePoint platform, where the document information could be tagged to the metadata fields of the document. Information extraction from the shipping documents in the area 1 deployment was performed by the global ocean center (GOC) uploading the shipping document to the SharePoint folder and then manually reading and populating the document metadata information on the SharePoint platform. With this approach the global ocean center (GOC) processed and extracted key information from the ocean transport documents by hand. This takes time as there are hundreds of documents which are needed to be processed daily. After the document was uploaded and tagged, an email was sent to the overseas sales desk to approve, modify or reject the document. Platform B was then able to connect to the SharePoint environment to read the document metadata information and track the document process related to the shipping process. The before process architecture is described in the figure 14 below.

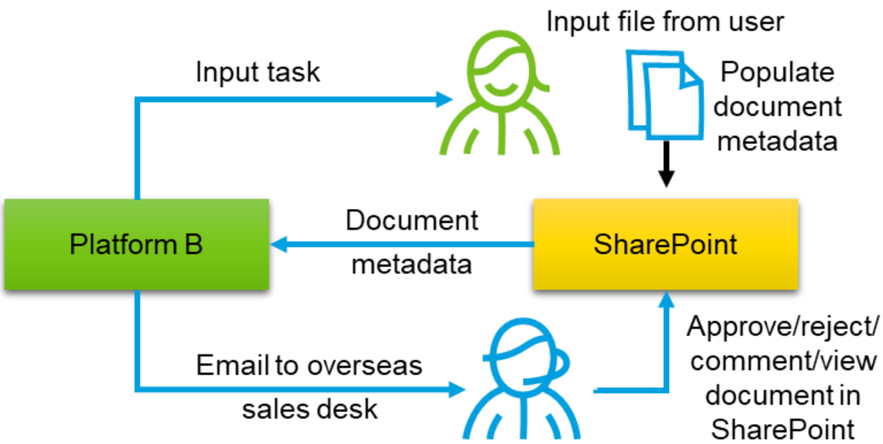


Figure 14. Shipping document workflow in the area 1 deployment of the Platform B.

The approach which the area 1 used to manage the ocean shipping documents had two flaws for the scaling capability of the Platform B. First, the document metadata population was manual and must be done by hand. The process support specialists will need to handle hundreds of ocean shipping documents daily in the scope of area 1 deployment which will require the workload of multiple people to keep up with the shipping process. In addition, the Platform B can’t reliably scale beyond the area 1 deployment before the ocean shipping document process is both faster and more efficient. The second problem which the approach will face is that the SharePoint DMS approach is not the most robust solution for a DMS since the platform uses manually populated metadata fields to store the information. The SharePoint environment is complicated to automate with third-party solutions which is why the cloud DMS architecture will be redesigned during the development of the intelligent document processing add-on.

5.4.3 Machine chain follow-up

The third challenge which the deployment team faced related to the machine chain of the produced orders. When an order is trimmed to a machine in the ERP system, a machine chain is created. Machine chain is the process chain in which the customer order goes through the machines from raw material to the finished product. Machine chain can have multiple machines depending on the customer specifications. It always has at least a board machine, but it can, depending on the product specifications, also have barrier coating machines (PE-machines), rewinding machines and sheeting machines. The machine chain is partly automatically determined by the customer order specifications and partly chosen by the master planning organization when planning the production of the machines.

The challenge which the target organization faced is that the barrier coating machines have two types of limitations which places rules to follow when choosing an order's machine chain. First type of restriction is the machine outcoming specifications which determines the specifications of products the barrier coating machine can produce. Second type of restriction is the raw material specifications which determines which type of products the barrier coating machine can convert.

The ERP system in the target organization is currently steering order handling based on the machine outcoming specifications, but not yet based on raw material from the board machine and based on PE-machine raw material specifications. This is intentional as the machine chain can't unambiguously be determined with only the PE-machine raw material specifications. In these situations, the master planning organization determines the used PE-machine. The problem with this is that the process is manual and the raw material specifications. In addition, the order steering instructions are currently listed on a process guideline document and on several machine-specific PDF-documents in more detail. The current process is illustrated in figure 15 based on one example product specification dimension.

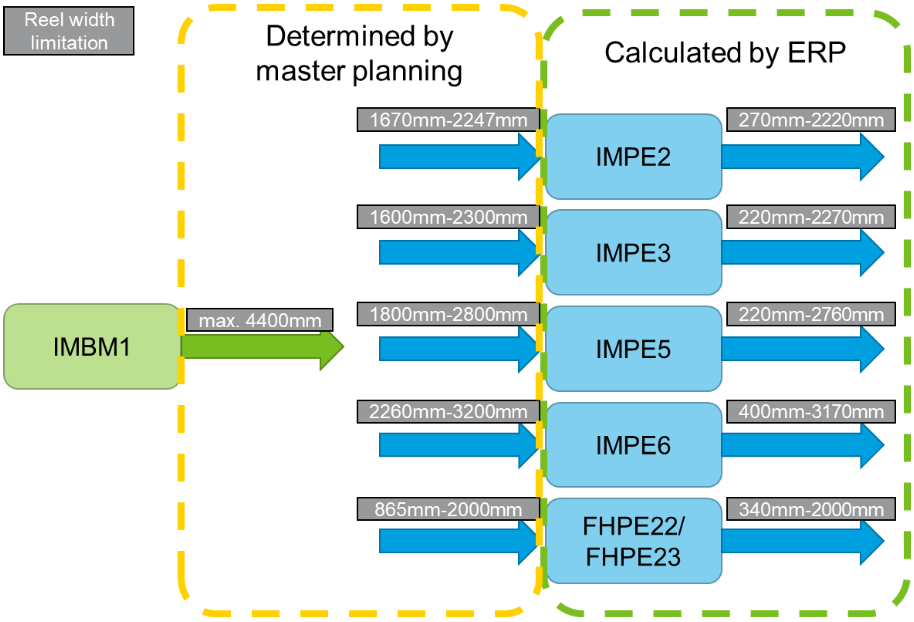


Figure 15. The machine outcoming product specifications and raw material specifications based on one input dimension (reel width).

The machine specifications have multiple product dimension criteria which all must be met for the order to be converted on a machine. In addition, the order can have a preferred barrier coating machine based on the order attributes, although based on the PE-machine specifications the order could be coated on other machines too. These preferred barrier coating machines are determined by a separate set of business rules, and it also includes the available capacities and the RTA date of the order. This preferred barrier coating machine calculation is however left outside the scope of this thesis and the focus is in determining the available barrier coating machines based on the machine specifications.

6 Supporting platform deployment with programmable add-ons

This chapter describes the programmable add-on development process, the developed add-ons during the thesis and key results for each individual add-on. The output of this chapter is the key results of the developed add-ons and the empirical findings of this thesis which are used to answer the research questions of this thesis.

6.1 Programmable add-on development process

Before the development of programmable add-ons was started in the target organization, a governance model for programmable add-on development was built. The purpose of the governance model was to productize the development of the programmable add-ons and establish a standardized model in which the programmable add-ons can be designed, developed, deployed and integrated into the digital service platforms. The basic principles of the programmable add-on governance model are briefly explained in this chapter before continuing to the detailed explanations of the developed programmable add-ons. The approach for developing the programmable add-ons was designed to be as lightweight and fit for purpose as possible in order to minimize the administrative burden and keep the focus on the development and integration of the programmable add-ons. The development of the add-ons was designed to be mostly value-driven meaning the add-on development backlog was formed by determining where most value could be achieved with the least amount of effort.

The developed governance model had two different layers. First layer was the governance layer which was concerned with the portfolio of the add-ons. The governance layer contains the add-on portfolio management principles, add-on documentation and decision-making forums. The second layer of the governance model is the development pipeline for a single add-on. It contains the development phases and the process tasks in each of the development phases. With the development pipeline, the progress of the programmable add-ons was tracked and planned. The governance model developed for the add-on development is visualized on the figure 16 below.

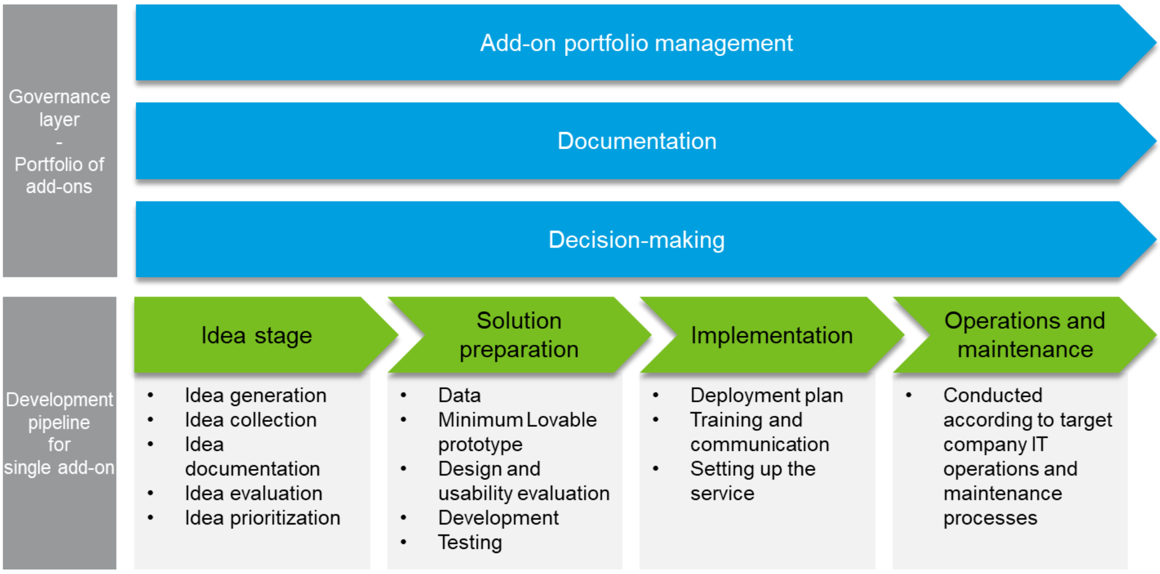


Figure 16. Programmable add-on development pipeline and governance layer.

The programmable add-on project portfolio was managed and prioritized using a value-complexity matrix which is a simple four field matrix comparing the add-ons in value obtained by development and the difficulty of development. The value-complexity matrix used to manage and prioritize the programmable add-on portfolio is illustrated in the figure 17 below. The programmable add-ons in the project portfolio are compared to each other in two dimensions. The first dimension is the value obtained by development which evaluates the add-on usefulness and impact on the digital platform deployment speed and scalability. The second dimension is the difficulty of development which focuses on the time needed for the programmable add-on development, the cost of the development and the technical challenge associated with it.

The key point in the prioritization is that the value and difficulty of add-ons are not directly measured, but the add-ons are compared to each other. With the value-complexity matrix, the add-ons are categorized into four distinct categories. The focus of add-on development is put on the add-ons which have high value obtained by development and are easy to develop. High difficulty and high value add-ons are selected carefully, and easy low value add-ons are rated by the added value of the programmable add-on. Low value high difficulty add-ons can be directly closed without solving. With these priorities and project constraints, the add-ons can be placed on the development roadmap and to the development backlog.

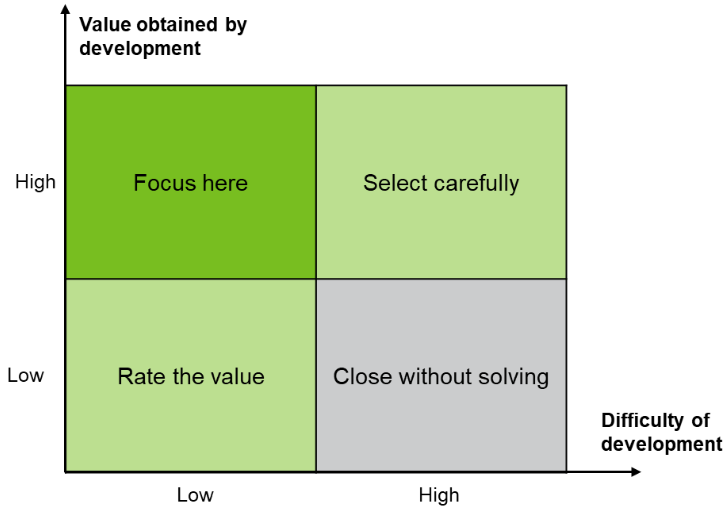


Figure 17. The value-complexity matrix used to prioritize the add-on development portfolio.

Decision-making forums contained the add-on development project meetings which tracked the development progresses of the programmable add-ons. At regular intervals, the new add-ons, which development will start, were chosen and the status of the add-ons in the development phase were checked. In the meetings, the evaluation of new ideas, review of the add-ons which were in development, and review of the add-ons which were moving to implementation phase were discussed in order to ensure that everything is ready, and the implementation will be successful. Also, the development roadmap or programmable add-ons was updated if there were changes to it since previous meetings. The role of meetings was to work as a gate keeper for the development of the programmable add-ons, meaning the meetings gave the permission to start the solution preparation for an individual add-on. The project management office (PMO) was designed to be lightweight, which was in the scope of this thesis a person who monitored and controlled the status of the add-on portfolio.

6.2 Developed add-ons

The challenges which the deployment teams faced and discussed during the workshops were used as an input in the programmable add-on development process. The challenges chosen to be solved with programmable add-ons were selected based on the suitability of the challenge from the add-on approach point of view. In addition, the selected challenges were prioritized according to the governance model described in the previous chapter before the development. For this thesis, four challenges were selected from the platform deployment teams' input, which were described in the chapters 5.3 and 5.4. Since each challenge is designed to be solved with a single add-on, four different programmable add-ons were developed during the thesis. The add-ons developed in the scope of this thesis are described in the table 4 below.

Table 4. Developed add-ons, functions, and the development objectives of the programmable add-ons.

#	Add-on	Related platform	Function of the programmable add-on	Development objective
1	Phone number formatter	Platform A	Enable the usage of platform functionality by transforming the CRM data into correct format	Deployment speed
2	Shipping process data enhancer	Platform B	Increase the platform data coverage and integrity by consolidating data from multiple sources	Deployment speed
3	Intelligent document processing	Platform A and B	Automate document processing and information pickup as a part of the document workflow	Scaling capability
4	Process follow-up control tower	Platform B	Allow the process support specialist to ensure business rule compliance of the barrier coating process	Scaling capability

The first developed programmable add-on was "Phone number formatter" add-on which was developed to Platform A. The purpose of this add-on was to enable platform functionality by transforming existing CRM contact data into correct format. This add-on enhanced the deployment speed of the platform. Second add-on was "Shipping process data enhancer" which increased the platform data integrity and coverage by consolidating data from multiple data sources. This add-on was developed to Platform B, and it increased had the objective to increase the deployment speed of the platform.

Third add-on developed during the thesis was "Intelligent document processing" add-on. This programmable add-on automated the document processing and information collection in the shipping document process area, and it had impact on both deployed platforms. The objective of the add-on 3 was to increase the scaling capability of the service platforms. Final developed programmable add-on was "Process follow-up control tower" add-on which was developed to the Platform B ecosystem. This add-on allowed a process support specialist to monitor the barrier coating process in the target organization and ensure that it follows the business rules related to it. This add-on also enhanced the scaling capability of the platform. The deployed digital service platforms and the developed programmable add-ons linked to the platforms are visualized in the figure 18 below.

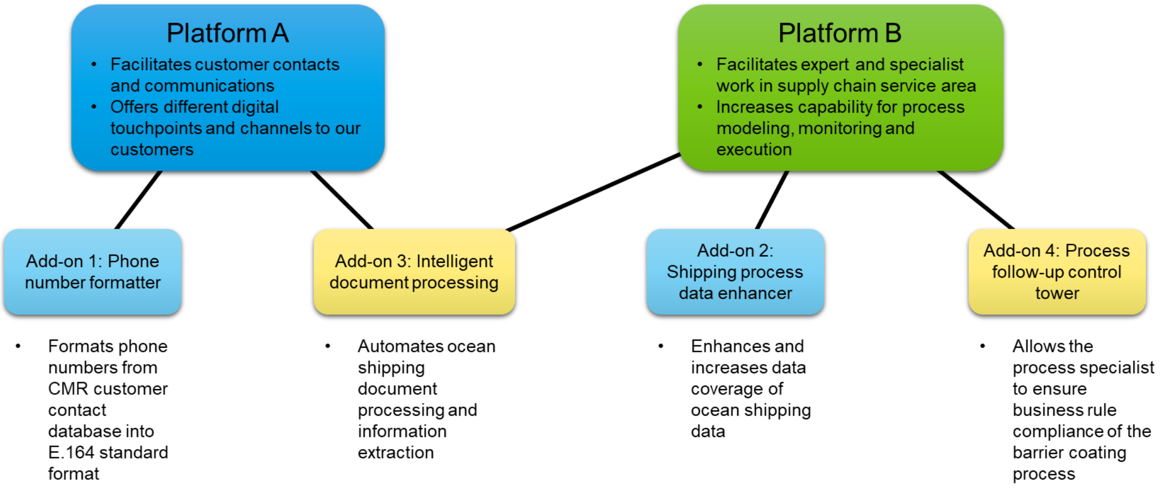


Figure 18. The deployed digital service platforms and programmable add-ons linked to the platforms.

For each of the developed programmable add-ons, a performance measure was selected. The purpose of the programmable add-on measurement was to quantify how effective the developed programmable add-ons were in the development objectives of deployment speed and scaling capability. The current state analysis before the add-on implementation is measured in part of the as-is analysis. After implementing the programmable add-on, the same analysis is performed again which allows the performance of the programmable add-on to be quantified and evaluated. The measurement areas and selected measures for the programmable add-ons are described in the table 5 below.

Table 5. Development areas, measurement areas and measures of programmable add-on performance.

#	Add-on	Development area	Measurement area	Measure (formula)
1	Phone number formatter	Deployment speed	Data quality	% of phone numbers in E.164 format (1)
2	Shipping process data enhancer	Deployment speed	Data quality	% of correct planned routes (1)
3	Intelligent document processing	Scaling capability	Process time	Time needed to process a single document (2)
4	Process follow-up control tower	Scaling capability	Process time	Time needed to determine suitable machines (2)

The add-ons which accelerate the deployment of the digital service platforms are measured with data quality of the platform data. The add-on 1 performance is measured with the percentage of phone numbers in correct format and the add-on 2 performance is measured with evaluating the number of planned routes which match with the actual route of the order. For the last two add-ons, which enhance the scaling capability of the platform, the performance of the programmable add-ons is measured with the process time improvement. Add-on 3 is measured with the time a single document is processed and add-on 4 with the time a suitable machine for the order is determined. The approaches and key results for the developed programmable add-ons are described in the next chapters.

6.2.1 Phone number formatter

Phone number formatter was a programmable add-on which was developed to solve the Platform A deployment issue described in the chapter 5.3. The goal of this programmable add-on was to solve the phone number formatting problem in the CRM database extract, which prevented importing the phone numbers directly into Platform A. The deployment team had already started to format the phone numbers by hand, but as the database contained over 50 000 phone numbers, the process was necessary to be solved with a programmatic approach.

To solve the problem, a programmable add-on was developed using a ready-made “libphonenumber” package. Libphonenumber is a library developed by Google for parsing, formatting, and validating international phone numbers. The library has many ports to different programming languages and in the development a Python port of the "libphonenumber" package was used to develop the programmable add-on.

The programmable add-on had multiple benefits to the Platform A deployment. The add-on accelerated Platform A deployment by speeding up the phone number formatting process. As the phone numbers could be formatted instantaneously, the Platform A deployment team could continue their work. Second, the contacts from the CRM database could be directly imported into the Platform A and with the programmable add-on no additional work was required to convert the numbers into a correct form. Finally, the developed programmable add-on can be utilized in future to ensure that the phone numbers in the CRM database are in a standardized format. The phone numbers can be entered in any format the user wants and the add-on can automatically format the phone number into the correct format before entering the value into the database.

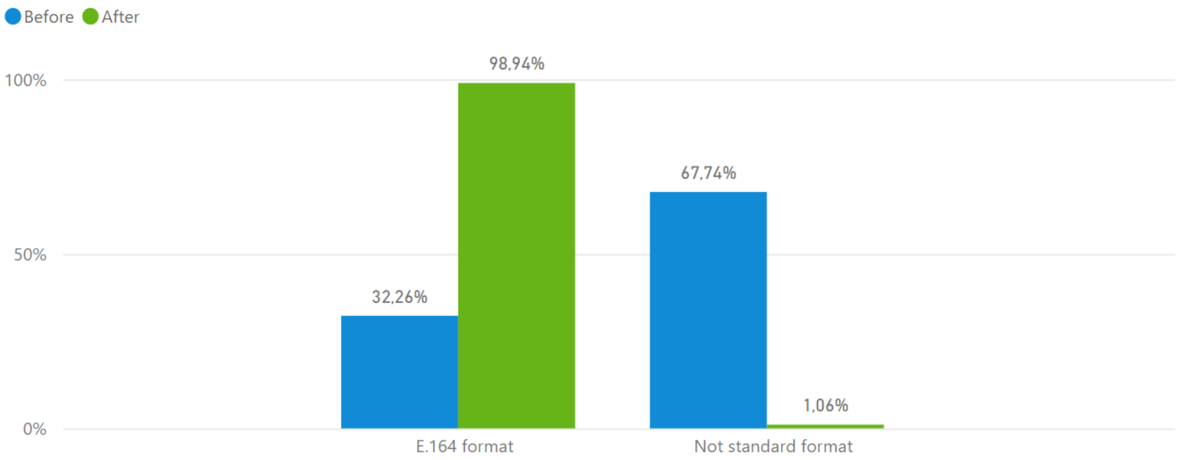


Figure 19. Results for the phone number formatter add-on.

The results for this programmable add-on are visualized on the figure 19 above. Before implementing the programmable add-on 32 % of the phone numbers in CRM database were in the correct format. The phone number formatter add-on achieved to format 99 % of the phone numbers into E.164 format. From the formatted phone numbers, the add-on detected that 1,4 % of the phone numbers are not valid by being for example too long or too short, but the algorithm could still format the phone numbers into a correct format. The deployment team could use this information to ensure the correctness of these phone numbers separately and either use them or filter them out. 1 % of the phone numbers were not able to be formatted into E.164 format. These cases were phone numbers which were for example free text information or contained only one number. These numbers could be deleted directly from the dataset and if the contact were still in use, the platform users can input the correct phone number manually afterwards.

6.2.2 Shipping process data enhancer

The second developed programmable add-on related to enhancing and increasing the data coverage the ocean shipping transportation data in Platform B. For this programmable add-on, the challenge the implementation team faced is described in the chapter 5.4.1. The goal of this add-on was to combine the real-life logistic process and the related transport document processes together. Furthermore, the add-on enabled automation of business processes related to the logistic workflow and automatic follow up of transport routes in next platform deployment areas. The initial approach for enhancing the logistic process data was to find the systematic errors in the data and correct these with an algorithmic approach. The solution for this problem was developed in two parts. First, an extensive analysis to the process data was made in order to detect the systematic errors in the dataset. Second, based on the systematic errors found in the data analysis, an algorithm was developed to correct the planned route information.

To find the predefined routes which always had systematic errors, a data analysis of the logistic process data was conducted. The data analysis contained transport data from the years 2021 and 2022 January to May. Overall, this dataset contained 430 thousand rows of planned route transport legs for orders and 1,5 million rows of actual transport events for orders. The comparison of the actual and planned routes was done with a graph-based approach. To transform the transactional process data into the actual transport route, a graph was constructed with NetworkX package in Python by transforming the transactions for each order of the dataset. The transaction in one row of the dataset represented a transport from source location into the destination location which can be transformed as an edge into the graph. The locations were then represented as nodes in the graph. Each of the orders could be transformed into a graph form and the actual routes from mills to the customer could be extracted. An illustrative figure from the transformation is presented in the figure 20 below.

Transport events

TRANSPORTID	MILLORDNO	LLOCATION	DLOCATION	
752212	K1ES-100077	FIIMAM	FIKTKH	A
753512	K1ES-100077	FIMUSS	FIKTKH	B
KTK-BIO10014	K1ES-100077	FIKTKH	ESBIOZ	C
6744803	K1ES-100077	ESBIOZ	ES4698	D
758448	K1ES-100077	FIIMAM	FILPPL	E

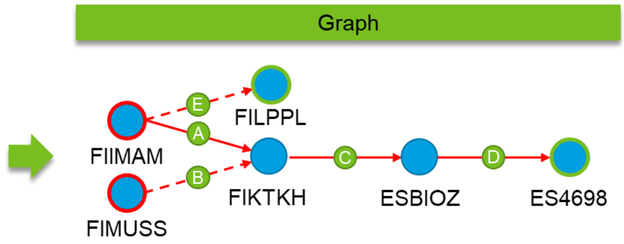
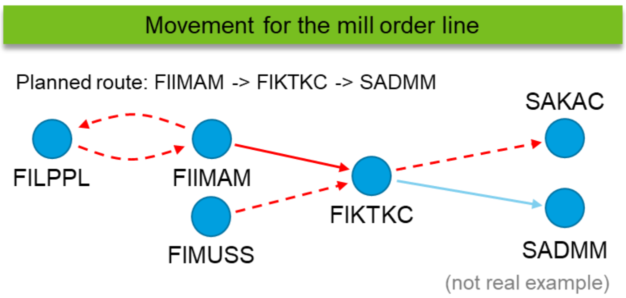


Figure 20. Construction of product transportation graph based on the transport event log data.

The planned route of the order was not needed for the graph construction, but it was used for analyzing and comparing the planned route to the actual transport routes of the order. Constructing a graph from the actual transport events allowed conducting more advanced analytics such as define the order of the transport legs, calculate the count of the previous transport legs and define easily, if a part of a transport is planned or not.

From the graphs, the source and destination locations of the orders could be extracted. The root nodes or starting locations are the nodes, which have an in-degree of 0 and the destination locations are the nodes which have an out-degree of 0. In addition, the planned route source and destination locations are always added as source and destination locations respectively since the starting location, for example a mill, can also have in-degree above 0. The method for finding the route end points is illustrated in the figure 21 below.



The only node with in-degree = 0 is “FIMUSS” and the nodes with out-degree = 0 are “SAKAC” and “SADMM”.
These are the first source and destination nodes

```
sources = ["FIMUSS"]
destinations = ["SAKAC", "SADMM"]
```

The planned route start and end locations are added as sources and destinations and prioritized.

```
sources = ["FIIMAM", "FIMUSS"]
destinations = ["SADMM", "SAKAC"]
```

Figure 21. The rules used for calculating transportation graph sources and destinations.

Based on the sources and destinations, the actual routes were calculated. The NetworkX package was used to calculate each route from the source location to the destination location. This way all possible actual routes could be extracted from the transactional data. In addition, the transactions which did not contribute to transporting the products from the mill to the customer could be disregarded in this analysis. An example of this is the FIIMAM-FILPPL leg in the figure 22 below. The extraction process for all actual routes of an order is also visualized in this figure.

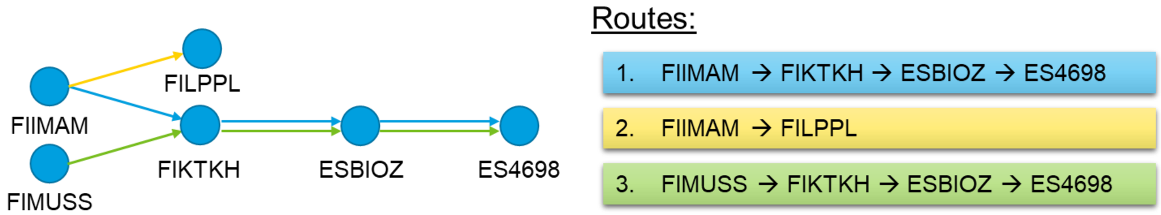


Figure 22. Extracting all order routes from the transportation graph.

When the actual routes were extracted from the transactional data, the actual routes could be compared with the planned route. The comparison was performed using a Python “difflib” library. With this library the routes could be compared and the differences between them could be extracted. An example of the result of this analysis is illustrated in the figure 23 below. This table was then analyzed on PowerBI. The analysis was performed to find which changes to the planned routes were systematic and which were random. These systematic errors could then be algorithmically corrected before loading the shipping transaction data to the Platform B.

Millorderline	Planned route	Actual route	Added	Removed	Changed
F2AT-162548001	DELBCM, AT6858	DELBCM, DETRVL			['AT6858']->['DETRVL']
F2AT-162550001	SEFORM, DEKAES, AT6858	SEFORM, DEKAES		['AT6858']	
F2AT-162651001	DELBCM, ATVIEPN, AT1210	DELBCM, AT1210		['ATVIEPN']	
F2BE-161258001	SEFORM, BEZEEL, BE1490	SEFORM, SEGOT, BEZEEL,	['SEGOT']		
F2BE-161328001	DEBAIM, DEBAIG, BE1070	DEBAIM, DEBAIG, BE1502			['BE1070']->['BE1502']
F2BE-161356001	SEFORM, BEZEEL, BE9030	SEFORM, SEGOT, BEZEEL,	['SEGOT']		

Figure 23. Sample of the route analysis table results.

The data-analysis revealed that before implementing the programmable add-on, 67% of the mill orders followed exactly the planned route. This result was in line with the discussions with the Platform B deployment team which they had already estimated to be around 70%. For order tracking and process analysis purposes in Platform B this number was too low and had to be increased. The data-analysis revealed that most of the planned route deviations happened during the ocean shipping leg of the products which is usually the second leg of the transport chain after the first transport leg from the mill to the port. This was the systematic error of the transportation process which was algorithmically fixed with the programmable add-on.

There were three root problem cases what was found to cause the planned route deviations in the actual route in shipping process. First problem was that the container stuffing location of the order is not present on planned route if the stuffing location differs from the ship departure location. This is because the stuffing location is determined closer to the shipping date based on the available capacity on the port. This case is visualized on the figure 24 below.

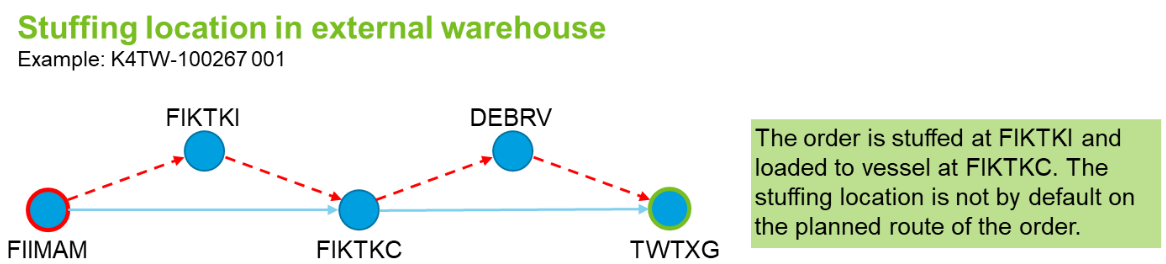


Figure 24. Case 1: Shipping port changing from the vessel departure port.

Second problem was that the actual departure location of the ship may also differ from the planned ship departure location. This can occur if the order is urgent and the next leaving ship to the destination port is departing from different port, or the loading capacity of the planned port is not high enough to load the goods on time. An example of this case is illustrated in the figure 25 below.

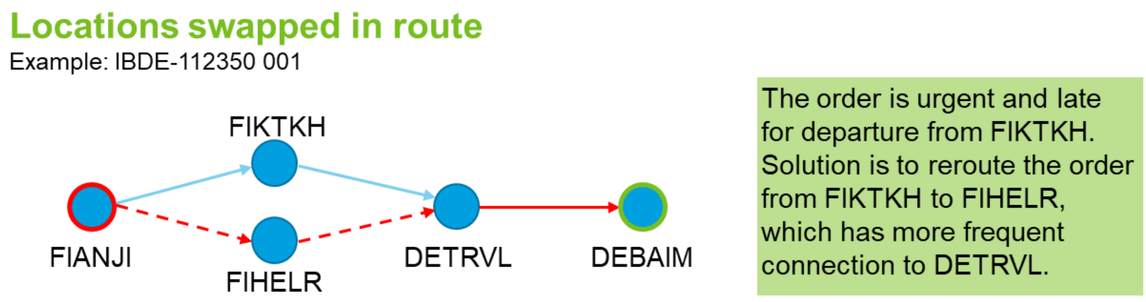


Figure 25. Case 2: Change of the departure port based on the ship booking of the order.

Third problem was that the ocean shipping leg may sometimes break into two different shipping transport legs. The goods are in these cases transported first with feeder vessels into ocean ports like Antwerp or Zeebrugge and there moved to an ocean vessel which transports the containers to the destination port. The transfer of the containers between the vessels leaves a location trace into the actual route in ERP system, which is not present in the planned route of the order and causes issues in tracking the order. This case is visualized in the figure 26 below.

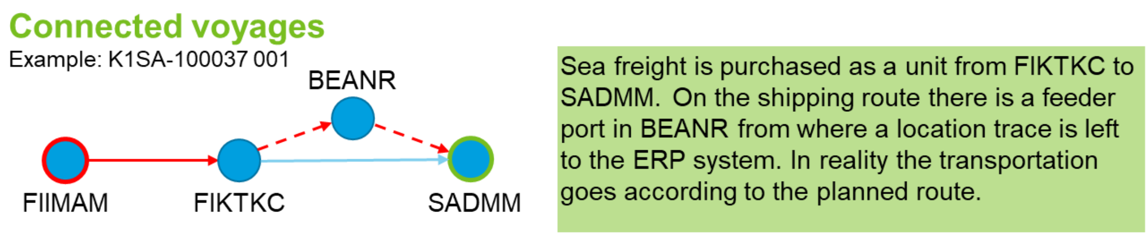


Figure 26. Case 3: Transfer from feeder vessel to ocean vessel leaves a location trace to the ERP system.

To fix the ocean transport leg problems related to the transportation data, a solution to combine two data sources was developed in the data extraction phase. The shipping transaction data was extracted to Platform B from the ERP system which is used in PM supply chain by all teams across the supply chain organization. However, one exception to this is the ship booking information of the ocean transports, which is currently handled by the dispatching organization in a different platform than the ERP system.

The booking data generated by the dispatching organization was missing from the planned transportation route of the order and thus the actual route had systematic deviations on the ocean transport part of the order route. From the system used by the dispatching team the ship booking information of the orders was extracted. The booking information extracted contains the actual booking leg of the transport. The ocean leg of the planned route was then replaced with the booked ocean leg using the booking leg starting and ending points as a reference points. This calculation layer determines the ocean leg from the planned route and replaces it with the booked ocean leg. The algorithm can figure the ocean leg if at least one reference point exists meaning that either the vessel departure or arrival location of the booking remains same compared to the planned order route. The figure 27 below illustrates the logic in which the programmable add-on is correcting the data.

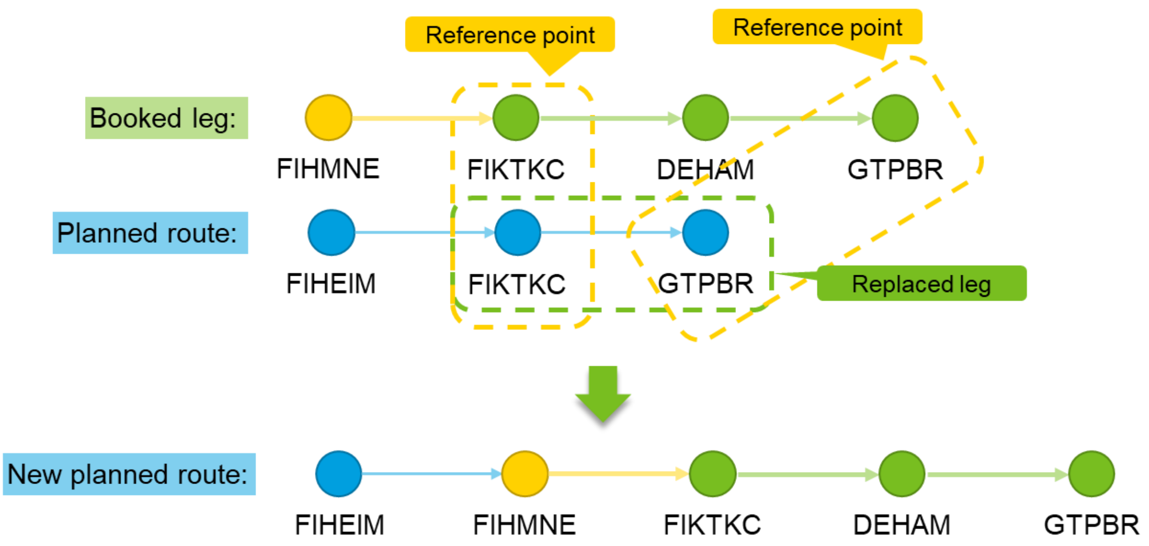


Figure 27. Algorithm logic for inserting the booked transport leg into the planned leg.

The route difference analyses performed in the first step was redone with the enhanced planned routes calculated by the programmable add-on. The analysis was made using the same source data which was analyzed in the first phase of the add-on development. Based on the analysis, the add-on was able to correct little over 33 % of the problematic cases resulting in 78 % coverage of the actual transport routes. The 33 % increase to the data coverage was a big step forward in the transportation process data as the shipping process could now reliably linked to the document processes. The routes which were already correct were only affected 0,18 % of cases. This means that very rarely the add-on modifies planned routes which are already correct in the planned part of the transportation route. The data enhancement results for this add-on are visualized on the figure 28 below.

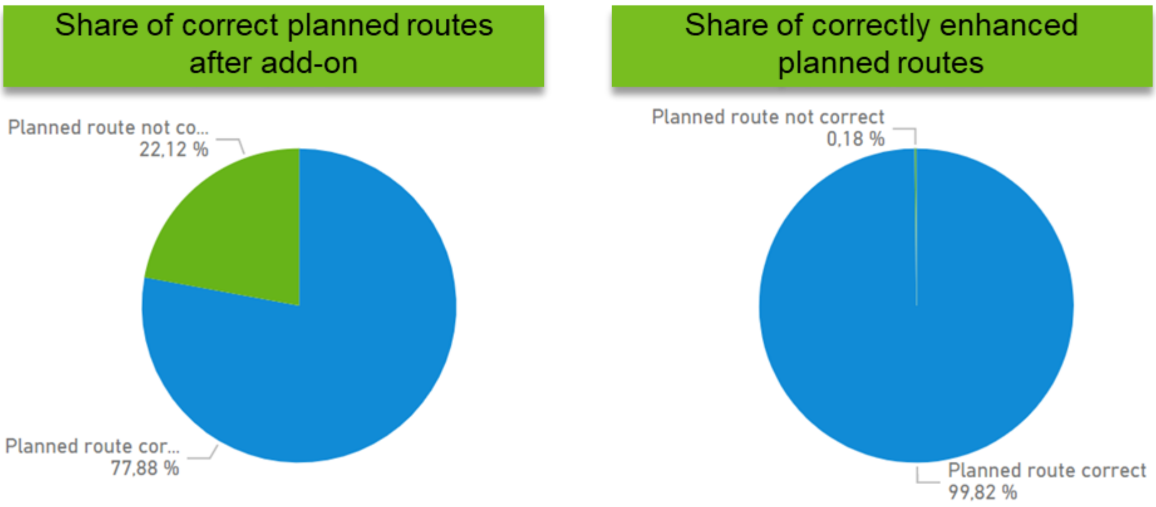


Figure 28. Data enhancement results for the shipping process data enhancer add-on.

Overall, the add-on increased the data coverage to 78 % of the entire order transportation process. Since the programmable add-on only fixed deviations in the ocean shipping part of the transportation chain, there were still some systematic deviations remaining on the other parts of the transport chain. The analysis performed after the add-on implementation could also identify where the deviations remained after implementing the add-on. These are described on the figure 29 below.

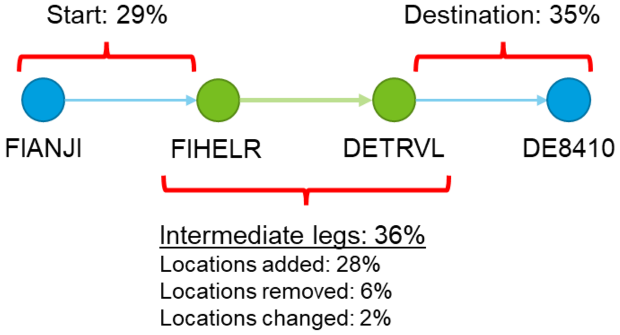


Figure 29. Unsolved cases after the implementation of the shipping process data enhancer add-on.

64 % of the deviations were related to either ending location of the actual route changing or the starting location of the actual route changing. Due to the ERP functionality of the target organization, not all planned transport legs are represented for all deliveries. The ERP system has rule-based restrictions which limits the transportation planning of the order in specific cases. These systematic errors caused by the functionality of the ERP system were described in detail to the target organization and corrective actions could be taken by the platform deployment team.

6.2.3 Intelligent document processing

Intelligent document processing was the third programmable add-on that was developed during the thesis. The goal of this add-on was to automate the manual business processes involved in the process area of the Platform B and thereby enhancing the scalability of the service platform B. Intelligent document processing is an add-on, which automates the document processing and key information extraction for the global ocean center (GOC) team in the target organization. The document processing is automated by introducing a machine learning based programmable add-on which extracts the information from the document automatically and stores them in a cloud-based database. The intelligent document processing add-on was the largest developed programmable add-on by size and had most impact to the business processes related to the platforms.

Based on the problem description in the chapter 5.4.2, the programmable add-on was developed and implemented in two phases. The first phase of the development was building the machine learning based document processing add-on which automatically processes the documents and reads the information from the ocean shipping documents. This was implemented by using an Azure Form Recognizer which is a cloud-based service that uses machine-learning models to extract key-value pairs, text, and tables from documents (Microsoft, 2022a). This first phase automatized the document key information extraction process. On the second phase the functionality of the machine learning based add-on was extended to a cloud DMS by adding already available Azure cloud functionality to the platform ecosystem. The developed architecture presented a more efficient way to archive shipping documents and its metadata in the Platform B operating environment.

The document processing and key information extraction parts were handled by Azure Form Recognizer. Form Recognizer uses machine learning (ML) to automatically pick up information from different document types for example invoices, receipts, business cards and identity documents (Microsoft, 2022a). The Form Recognizer model was trained by using Azure Form Recognizer Studio which allows the configuration of the model through a visual user interface. Azure Form Recognizer requires at least five documents per document type to train the model. The training process began by collecting the needed documents from the GOC team for model training and testing. The programmable add-on was developed for eight different ocean carriers which were included in the area 1 deployment of Platform B. Each carrier had two document types which was needed to be processed: bill of lading (BL) and sea waybill (SWB). From each of the carriers enough BLs and SWBs were collected to train and test the model, and to validate the performance of the ML model. In total, the dataset consisted of 92 documents from which 80 were used to train the model and the remaining 12 documents were used to test the performance of the programmable add-on.

The confidence level of the ML model by different carriers and by information fields is presented in appendix 1. The trained ML model achieved total of 84,1 % confidence level which means that the model correctly extracts on average 84,1 % of the information fields from the documents. From the extracted fields, “millorderlines” is the hardest information for the ML model to extract from the documents with 62,5 % confidence level. This is because millorderlines information is in a multi-page table format in the shipping documents, which are extremely hard for ML models to correctly detect. Without millorderlines the confidence level of the machine learning model is 96,8 %, which is a good result for a ML model. Some carriers’ documents were easier to process than others which is due to the delivered documents being in both text-based and image-based PDFs, and the image-based PDFs being harder for the ML model to extract the key information.

On the second phase of the add-on development the add-on was extended to also include the document image storage and document metadata storage in a cloud-based document library. Since the automation infrastructure is difficult to build on top of the SharePoint platform, a concept for a new cloud DMS was designed to allow the automation of document processing. The designed infrastructure and the API interface are visualized in the figure 30 below. The cloud document library can be accessed by the platforms via the API interface. The architecture contains three components. Azure Cosmos DB is responsible for storing the document metadata information the Form Recognizer ML model extracted automatically from each shipping document. Azure Blob Storage is used to store the document in PDF form for the user to inspect in the digital platforms. Finally, Azure Cognitive Search is responsible for searching and indexing the document cloud database.

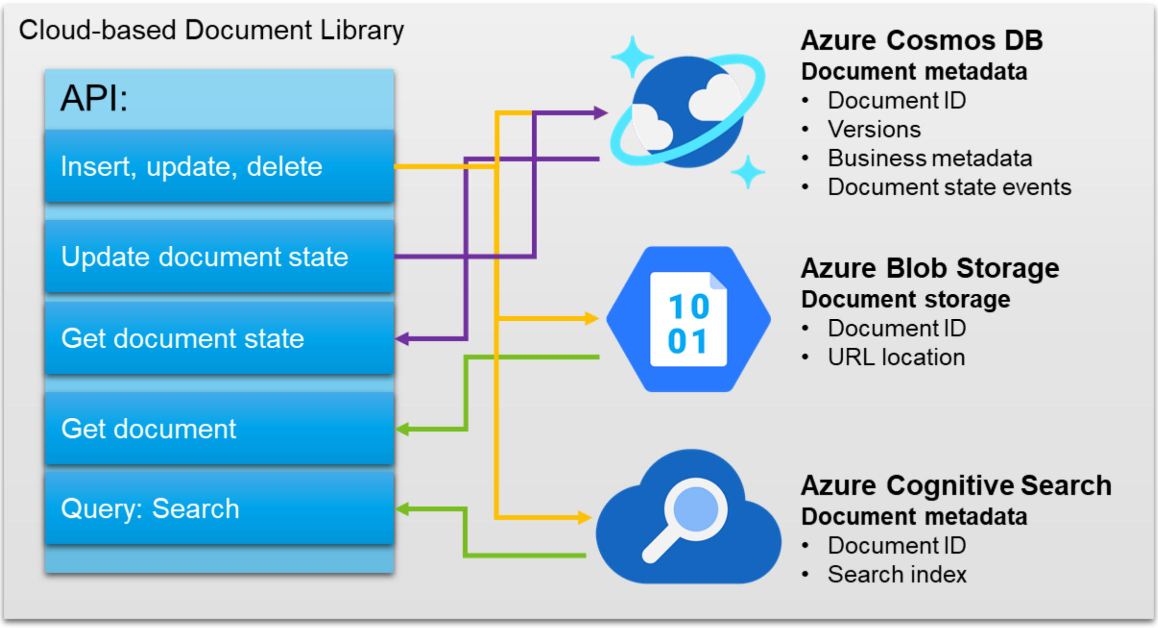


Figure 30. Architecture for Azure cloud-based document library.

The automation layer for the cloud-based document library was developed using Azure Logic App. The Azure Logic App monitors continuously the Azure Blob Storage where incoming ocean shipping documents are stored from the shared mailbox the GOC team uses. When a new document is detected by the Logic App, it sends the document to the Form Recognizer add-on which then processes the document and returns the document metadata to the Logic App. Then the Logic App saves the document metadata information to Cosmos DB. The digital service platforms can access the document metadata information directly from the Cosmos DB, and with this database, the business process for document approval and rejection can be performed in both platforms and accessed by all teams in the target organization. More detailed visualization of the cloud DMS workflow and architecture is described in the figure 31 below.

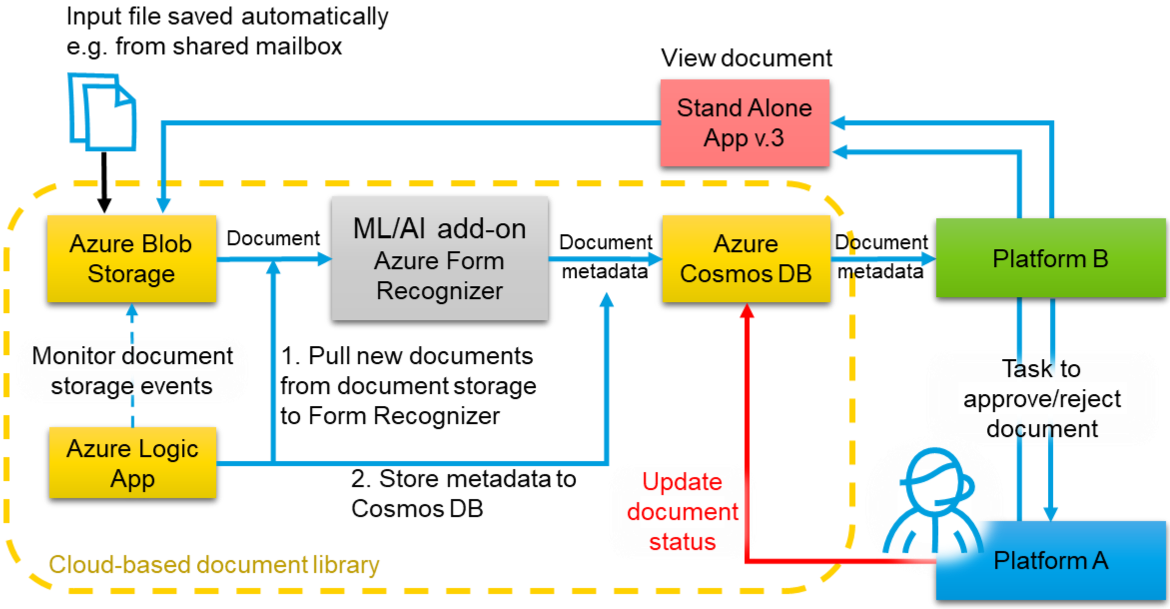


Figure 31. Document workflow in the future system architecture for the ocean shipping document processing.

The Intelligent Document Processing programmable add-on and cloud-based document library increased the scaling capability of both platforms in the scope of this thesis. The add-on itself eliminated a manual and repetitive task of extracting key information from the shipping documents and thus removed the user errors from this process part. The cloud DMS modernized ocean shipping document management to a more scalable and efficient system enabling automation of other process parts in the digital service platforms. The Intelligent Document Processing add-on processes a single shipping document on average in 10 seconds while the GOC team needed on average 2 minutes to process a single document based on their input. The efficiency and performance which this add-on adds to the digital platform is thus particularly important for the future scalability of the platforms.

6.2.4 Process follow-up control tower

The fourth developed programmable add-on was a process follow-up control tower add-on, which was developed based on the problem description described in the chapter 5.4.3. The goal of this add-on was to allow the process support specialist to ensure business rule compliance of the production process. The add-on was in first step developed for Imatra board machine 1 (IMBM1) and for six different barrier coating machines which use IMBM1 board as raw material. These barrier coating machines are:

- Imatra PE coating machine 2 (IMPE2)
- Imatra PE coating machine 3 (IMPE3)
- Imatra PE coating machine 5 (IMPE5)
- Imatra PE coating machine 6 (IMPE6)
- Forshaga PE coating machine 22 (FHPE22)
- Forshaga PE coating machine 23 (FHPE23)

The barrier coating machine for the order can be chosen freely based on the available capacity of the PE-machine within the limitations of the barrier coating machine specifications.

This programmable add-on was developed using Python and the solution uses two different scripts to calculate the available machine chain. The first script transforms the machine limitations in the written text PDF to a lookup table which is referenced by the second script. The calculation is done in two parts because the calculation of suitable barrier coating machines can be precalculated based on the instructions and then mapped to the order lines based on the order attributes. This method allows faster determination of the machines as the rules are not needed to be calculated each time the add-on is used. The lookup tables are calculated for each order attribute dimension which determines or limits the suitable PE-machine.

The second developed script calculates the suitable barrier coating machines based on the product attributes and the precalculated lookup tables. The lookup tables give a set of barrier coating machines for each product dimension. The barrier coating machine, which can be utilized for the barrier coating for the order, can be determined based on the intersection of these machine sets. Figure 32 illustrates this process for an example order. In the figure, four example order attributes are described. Based on the customer order, each attribute is needed to be checked, if the order can be converted on the particular machine. In the example, IMPE3 is the only PE-machine which meets the limitations in all product dimensions because it is capable to convert the order in respect to all order attributes.

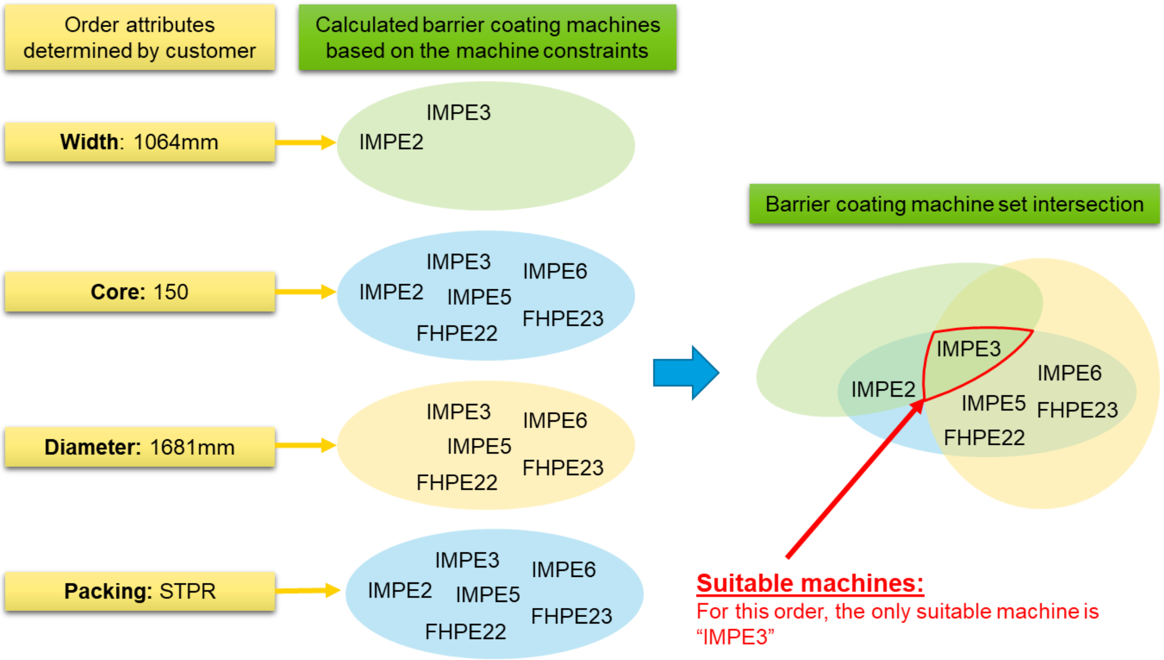


Figure 32. The determination of suitable barrier coating machines in each product attribute.

The programmable add-on successfully increased the scaling capability of the Platform B as it eliminated a manual and repetitive task performed by the process support specialists in the process. As the sales desk places the order and picks the barrier coating machine, the programmable add-on monitors the order flow and picks up the orders which do not follow the process guidelines. The process support specialist can then either change the PE-machine to one of these defined machines based on the available capacity or ignore the error.

The programmable add-on also enabled process automation as it has the capability to automatically pick the suitable barrier coating machine based on the available capacity on the machines. This way the sales coordinator does not have to enter the preferred barrier coating machine in the order entry phase, but the platform and the programmable add-on can automatically determine in which barrier coating machine the order should be converted. The process follow-up control tower add-on can determine the available machines instantly based on the order attributes, when before the process took on average 15 seconds with a custom-made application based on the input of the platform deployment teams.

7 Results and evaluation

This chapter summarizes the key results of the thesis by answering the research questions, evaluates the results achieved in the thesis and finally suggests follow-up measures and future research recommendations based on the academic literature and theoretical and empirical knowledge gained during the study. The output of this chapter is the theoretical and empirical knowledge which this thesis contributes to digital platform and programmable add-on research and to manufacturing organizations deploying digital platforms.

7.1 Key results of thesis

The outcomes of this thesis have provided insight into how the deployment and scalability challenges of digital platforms can be solved with programmable add-on approach. The results indicate that the programmable add-ons can increase the deployment speed and the scaling capability of digital service platforms. The programmable add-ons which were developed and integrated into the two supply chain service platforms increased the deployment speed of the digital platforms and created the capability to scale the service platform operations into future deployment areas. The challenges which the service platform deployment teams faced during the deployment were possible to solve without implementing programmable add-ons into the digital platform ecosystems. However, without programmable add-ons the platform deployment would have delayed the platform deployment significantly (add-on 1), limited the process coverage of the platform (add-on 2) or hindered the performance of the digital platforms (add-ons 3 and 4) thus reducing or even eliminating the benefit obtained implementing digital platforms.

The main benefits of programmable add-ons were that they brought solutions quickly to the deployment and scalability challenges the platform deployment teams faced during the implementation. With programmable add-ons the deployment teams could continue their work and the service platforms could be deployed faster and with more features. In addition, the programmable add-ons increased the throughput of the existing business processes in the target organization and thus made the processes compatible with the deployed digital platforms with minimal modifications to the process. By this logic it can be argued that the main benefit of programmable add-ons is not necessarily the capability to solve the deployment and scalability challenges of digital platforms, but to deliver results and value in a quick and efficient manner and to allow the business users to develop add-ons with either domain-specific applications or LCNC development tools depending on their needs and technical capabilities.

The research of the thesis was structured with three research questions. The conclusions for the research questions based on the results will be presented next.

RQ1 What are the functional problems related to digital service platform deployment speed and scaling capability?

Academic literature has predominantly examined the role of deployment strategy (Shasi & Sinha, 2021) and organizational culture (Brunetti et al., 2020) in the digital platform deployments and integrations. However, the functional challenges in digital service platform integrations are not examined as well as the forementioned topics. In the empirical study performed in this thesis, the deployment speed related issues can be concluded to derive from the data integration problems the platform deployment teams faced during the first platform deployment areas. The emerged issues had three different main root causes. First cause for the integration problems was the lack of structural integrity of the source data (add-on 1), which is also supported by the study of Schmidt et al. (2019). Second reason is the existence of information silos and multiple source data systems (add-on 2) and third the characteristics and communication of the legacy source systems, which affects the quality of the source data (add-on 2). These last two points are also supported by the study of Dillion et al. (2010) who discussed the integration of the cloud systems to local enterprise applications to be a challenge. These problems would cause the deployable digital platforms to have limited functionality or inferior performance which meant that solving these challenges was necessary before the platforms could be deployed into the production environment. As the effort of the platform deployment teams was directed to solving these data integration problems, the deployment speed of the platforms suffered severely in the target organization.

The scalability issues of digital platform were not discussed in the academic literature examined in the theoretical part of this thesis, but these problems were evident in the target organization. As a result of the empirical study, scaling capability problems can be evaluated to be more related to the existing business processes in the target organization which did not have the capability and throughput to operate in the implementable digital platform ecosystem. The causes of platform scalability issues were that the current process was highly manual (add-on 4), or the implementable process was entirely new or temporary from the platform's point of view (add-on 3). The deficiencies in the scaling capability of the platforms caused bottlenecks in the business processes which in turn prevented the platforms scaling to the current or future deployment areas. The problems in platform scaling capability thus had to be solved with adding new functionality to support more efficient way of working (add-on 4) or automate the current processes eliminating the need of human resources entirely in the process (add-on 3).

RQ2 How can the programmable add-ons be developed and integrated into digital platforms?

Based on the academic literature development methods for programmable add-ons are almost endless. According to Wasserman (2020) the spectrum of development tools ranges from use-case specific software platforms to general-purpose programming languages. This way the programmable add-ons can also be developed by citizen developers which is supported also by Lebens et al. (2022). The add-ons can be integrated into digital platforms in several ways but according to the academic research most common architectures are layered architecture (e.g., Chavan et al., 2015), microservices architecture (e.g., Innerbichler et al., 2017), and event-driven architecture (e.g, Keshavarzian et al., 2019).

The digital platforms in the scope of the empirical part of the thesis offer basic functionality for data extraction, transformation and loading (ETL). The platform ETL functionality could be utilized for basic transformation of the source data from the existing systems. However, capturing more advanced business logic with the standard ETL tools the platforms provided could not be achieved. The programmable add-ons developed on the scope of this thesis were hosted on the Azure cloud environment. Platform B has also an own cloud environment for hosting Python code, which was used on Platform B programmable add-ons. The development of these add-ons was done with Python or with Azure Form Recognizer, which could then directly be hosted in the Azure cloud platform and accessed via API interface.

RQ3 How effective are the programmable add-ons in solving the functional problems related to digital service platform deployment speed and scaling capability?

The programmable add-on approach was extremely effective and was very fit for purpose in solving the functional problems in digital platform deployment. In terms of the digital platform deployment speed, the programmable add-ons produced impressive results. The phone number formatter add-on (add-on 1) could format the phone numbers into the E.164 standard format. The approach which was used to format the numbers was able to identify which were not possible to exist according to the standard information. The deployment teams could then process these numbers, and either be fixed or deleted since there was not more than few hundred of invalid numbers after the formatting.

For the shipping process data enhancer programmable add-on (add-on 2) the add-on increased the coverage of the planned routes 33 % which resulted the correct share of routes increase from 66 % to 78 %. This programmable add-on enabled the platform deployment team to connect the ocean transportation process to the shipping document process in the platform B. The purpose of this add-on was to correct the systematic errors caused by the characteristics of the source systems, which this add-on performed well. In addition, the analysis after the implementation of the add-on revealed new development targets which could be added in future iterations of the programmable add-on. The add-on data quality improvements for add-ons 1 and 2 are summarized on the table 6 below. Since data quality was not the goal of add-ons 3 and 4, these were not measured for these add-ons.

Table 6. Data quality improvements per add-on.

#	Add-on	Data measure	Before add-on	After add-on
1	Phone number formatter	% of phone numbers in E.164 format	33 %	99 %
2	Shipping process data enhancer	% of correct planned routes	66 %	78 %
3	Intelligent document processing	*Not applicable*	*Not applicable*	
4	Process follow-up control tower	*Not applicable*	*Not applicable*	

For the scaling capability of the digital platforms, the programmable add-ons exceeded expectations. Intelligent document processing add-on had two main improvements for the processes in Platform B. First, it improved the process throughput time by decreasing the time used to process the ocean shipping documents. This allows the scaling of the Platform B to future deployment areas as the ocean document processing is not bottleneck for the process when the volume of ocean shipping documents is increased. Second, the add-on freed up the resources which were tied to the processing of the ocean shipping documents. This means significant cost savings to the target organization and improvement in the efficiency of the operations.

The process follow-up control tower add-on (add-on 4) had similar implications to the target organization. The time which was required for determining the suitable barrier coating machines was decreased significantly and with the programmable add-on the possible barrier coating machines are calculated automatically. The implementation of this programmable add-on released human resources from this process and thus increased the overall efficiency. The share of cases which followed the process description were also measured for the process follow-up control tower add-on for the target organization. The add-on process performance improvements for add-ons 3 and 4 are summarized on the table 7 below. Because the process performance enhancement was not the objective of add-ons 1 and 2, these were not measured for these add-ons.

Table 7. Process performance improvements per add-on.

#	Add-on	Process measure	Before add-on	After add-on
1	Phone number formatter	*Not applicable*	*Not applicable*	
2	Shipping process data enhancer	*Not applicable*	*Not applicable*	
3	Intelligent document processing	Time needed to process a single document	2 minutes	10 seconds
4	Process follow-up control tower	Time needed to determine suitable machines	15 seconds	<1 second

Overall, the programmable add-ons successfully accelerated the deployment and enhanced the scalability of the digital platforms. With implementing programmable add-ons, the digital service platforms were able to be deployed earlier and to further deployment areas.

The programmable add-on approach suited for both increasing the deployment speed of the platforms by solving the data integration issues, and to increasing the scaling capability of the platforms by enhancing and automatizing the business processes in the platforms.

7.2 Discussion and evaluation of results

The results achieved during the thesis are very promising from the digital platform implementation perspective. The programmable add-ons could solve the challenges in platform implementations in an efficient manner and with this accelerate the deployment and enhance the scaling capability of the digital service platforms. As the deployed digital platforms were important strategic element of the future operations in Stora Enso PM Supply Chain, the programmable add-ons supported the execution of the strategy. The technical capabilities in the target organization were on a necessary level to explicitly implement the programmable add-ons to the digital platforms and deploy to a cloud environment. In organizations where no such capabilities exist, the deployment and implementation of programmable add-ons may be more limited, or some prerequisite infrastructure is needed to be implemented in order to support the deployment of the programmable add-ons.

The implementation of programmable add-ons also has implications to the operations phase of the digital platforms. The programmable add-ons will need to be maintained and updated to remain operational in the digital platform ecosystem. The practices for the operation phase of the programmable add-ons were established in the governance model of the programmable add-on development but these were not extensively researched for the purposes of this thesis. The programmable add-ons should remain complementary to the platform operations and thus they should not become a critical part of the platform operations. Therefore, the integration of programmable add-ons should have a fallback functionality or secondary available service which can perform a similar functionality to the programmable add-on. Finally, the issues in information and data security should also be considered in programmable add-ons. Especially in functions where data is collected and aggregated from multiple inter-organizational systems, the boundary of the programmable add-ons, their data access and communication protocols should be well defined. The points regarding the operations phase of programmable add-ons were not addressed during the research and thus these topics should be taken into accord in future research.

This thesis contributes to the theoretical research by examining what challenges are related to the digital platform implementation in manufacturing organizations. Additionally, this study contributes to the research by demonstrating how deployment and scalability challenges in digital platform implementations can be solved with implementing programmable add-ons. In doing so, this study expands the existing academic literature of digital platforms and programmable add-ons. The results achieved during the study build on the existing evidence of the importance of digital platforms to organizations and the benefits of modular and loosely coupled software systems.

While the previous digital platform research has predominantly focused on the benefits of digital platforms especially in terms of the servitization and PSS-based business models in manufacturing organizations, these results demonstrate that organizations are currently facing challenges in deploying the digital platforms into their operations and scaling the platforms further deployment areas. Addressing and extending this research gap, the theoretical contribution of this thesis includes the challenges in data, integration of existing systems and integration of existing processes into digital platforms manufacturing organizations are facing. Furthermore, this thesis contributes to the digital platform research by practically demonstrating a method for developing and integrating programmable add-ons and with this approach solving the challenges of digital platform deployment and scalability from practical perspective. However, this thesis only displays one approach for solving these issues in one manufacturing organization deploying digital platforms and further research is needed to address these limitations of the research.

From managerial perspective this thesis shines light on the importance of digital platforms and programmable add-ons in servitization. Digital service platforms are an important piece of infrastructure in organizations searching competitive advantage through servitization and digitalization. The academic research emphasizes the role of digital service platforms in enabling more efficient service-offerings and thus gaining better competitive advantage. However, deploying the digital service platforms in a manufacturing organization is a complicated process which is why organizations deploying digital service platforms must overcome the challenges in digital platform deployments and implementations efficiently to be able to gain benefits from the digital platform ecosystem. The modular nature of digital platforms can be seen as a key enabler in efficiently deploying the platforms and solving the issues in platform deployment and operations. From the servitization point of view the modularity allows organizations to dynamically adjust the service offerings based on the customers' needs and requirements. From software development perspective the modularity allows deploying the programmable add-ons to cloud and dynamically develop and scale the software systems based on the platform requirements.

These two perspectives advocate the use and importance of programmable add-ons in implementing digital platform technology in manufacturing organizations. Because the infrastructural elements of the programmable add-ons are managed by the cloud platform and the digital platform is responsible for representing and storing the data, the needed development effort mostly includes capturing the business logic into the programmable add-on. For this reason, the speed-to-value of the programmable add-ons can become extremely high in digital platform implementations. Based on the empirical research, first iterations of an add-on could be delivered and integrated in days depending on the size of the add-on and the background work needed to capture the business logic.

The results of this thesis are however limited by few factors. The generalizability of the results is limited by the fact that the research was made as a case study for a single company. The conditions of the digital platform deployment such as information systems, the domain for digital platform implementation and the required infrastructure is highly individual and thus the results can't directly be extrapolated to a larger set of companies or organizations. To overcome this limitation, future research is needed to be conducted in different organizations and domain areas.

The reliability of the results is limited by the fact that the measurement for the results was conducted in a brief time period because the scope of the thesis was limited to researching the implementation challenges and solving them with programmable add-ons. However, the results of the study can be utilized in answering the research questions because the focus of the thesis was in researching the programmable add-on approach rather than quantifying the performance of the programmable add-ons. The validity of the measurement is only limited in the scope of the target organization since the numerical results were only collected for the purposes of validating the performance of the programmable add-ons in the target organization and thus the validity of the results does not affect the study or results in a general level.

Finally, the scope of the thesis limited the research period to the earlier deployment phases of the digital platforms and the issues in later stages of platform deployment could not be considered on the time frame of this thesis. Challenges and limitations in later deployment phases of digital platforms is thus left to be researched by scholars in digital service platform research area in the future.

7.3 Future research areas and development directions

This thesis suggests four future research directions in the digital service platform research area and to the programmable add-on development related to it. In the digital platform domain area, more research is needed of the deployment and scalability challenges and issues organizations face when deploying and scaling digital platforms. Current academic literature discusses the benefits of digital service platforms and the issues from strategic and organizational perspectives extensively, but the research of challenges of platform integrations and scalability in different phases of deployment is currently limited. Second, this thesis suggests more empirical research of the viable solutions to the integration issues in digital platform implementations. The contribution of this thesis is application of programmable add-ons in single target organization and thus a larger set of companies or organizations is needed to be researched in order to the results to be generalizable.

In the domain area of the add-ons, this thesis suggests researching the application of programmable add-ons in other digital platform ecosystems and manufacturing organizations. The needs and requirements for programmable add-ons are subjective and thus it would be valuable to gain insight into how other organizations would utilize the programmable add-ons in digital platform ecosystems. Finally, this thesis suggests the governance and maintenance of programmable add-ons as a future research topic for the scholars interested in this subject area. These points were not discussed and examined thoroughly during the thesis although they are important topics in applying and implementing the add-ons in practice.

In the target organization the future development direction for digital platforms and programmable add-ons is already set. The modularity of digital platforms and the ease of developing and integrating programmable add-ons into digital platforms has a high value-adding potential in many applications. In future, the programmable add-ons will be used in the integration of strategic supply chain partners in the target organization’s value ecosystem. The development efforts for integrating digital platforms with strategic supply chain partners has already been started in Stora Enso PM Supply Chain. The goal of this initiative is to enhance the real-time visibility in the PM supply chain from the mill all the way to the end customer.

The programmable add-ons play a key role in the supply chain integration as the integration of systems between organizations can be challenging. However, the potential benefits obtained by integrating the digital platforms between strategic partners is super high and the advantages of digital platforms can be gathered from the E2E supply chain with this approach. For example, in the service area of Platform A, the inter-organizational digital platform can offer multiple digital touchpoints for the customer in the entire delivery process. Likewise, in Platform B service area, the integration can offer real-time and detailed process monitoring and analysis in the E2E process of the value chain.

8 Conclusions

The main objective of this thesis was to accelerate the deployment and enhance the scaling capability of strategic supply chain service platforms deployed in Stora Enso PM division. The deployment speed and scaling capability of the digital platforms were increased by solving functional challenges in the platform deployment. This objective was approached by developing and integrating programmable add-ons to the digital service platforms, which solved functional problems, enabled tasks to be done or enhanced processes or user experience, and thus supported the platform deployment speed and scaling capability. Second objective of this thesis was to research what challenges manufacturing organizations face when deploying digital platforms in their organizations and how these challenges can be solved with implementing programmable add-ons in the digital platform ecosystems. This objective was approached by studying the academic literature of digital business platforms and programmable add-ons and conducting an empirical case study to a manufacturing organization.

Based on the academic literature, the usage and implementation of programmable add-ons is supported and driven by two main forces. The first driving force for digital platforms and programmable add-ons is the servitization trend in manufacturing companies. Manufacturers are becoming more service-oriented due to additional competition and increased customer demands in the markets, and the modular digital platforms are seen as the enabler for more efficient and dynamic service offerings. On the other hand, the modular software architectures are becoming increasingly popular due to the emergence of cloud-computing platforms, which require modular software development architectures for fast, scalable and agile software systems. Thus, it can be concluded that the modular approaches, such as programmable add-ons, are justified and valuable in implementing digital service platform technology in manufacturing organizations.

The empirical part of the thesis was conducted as a qualitative case study to a manufacturing organization. The case study was divided into two parts containing current state (as-is) analysis and future state (to-be) analysis. The current state analysis contained a data collection part by workshop meetings with the platform deployment teams and numerical analysis to the problem cases the deployment teams described. Based on the input from the current state analysis, four programmable add-ons were developed during the thesis. These add-ons addressed and solved the problems the digital platform teams were facing. Finally, the performance of the programmable add-ons was measured with analyzing the source data of the platforms in respect to a set of performance measures selected.

Based on the empirical research of the thesis, it can be concluded that manufacturers are facing challenges in digital platform deployment and scaling capability and these challenges can be solved in a quick and efficient manner with programmable add-ons. The results of the empirical study showed that the problems with deployment speed related to the data integration and quality problems of the source data in existing local enterprise applications which is also supported by the academic literature. The scaling capability issues were discovered to be more related to the throughput of existing business processes and the bottlenecks in them in the empirical study, but the scaling capability of digital platforms was not yet extensively researched in the academic literature.

Programmable add-ons can be developed and integrated into digital platforms using various different methods. The emergence of LCNC platforms allows easier and more business-user friendly development experience, and the programmable add-ons can be developed for example by citizen developers. Different integration architectures are being researched and demonstrated in practice by the academic literature and the best approach for integrating the programmable add-ons can be decided case-by-case. In the scope of the empirical research of this thesis, the programmable add-ons were deployed either to Azure cloud platform or to the cloud platform provided by the digital platforms. By using ready cloud-based solutions the programmable add-ons could be deployed and results achieved fast.

The programmable add-ons implemented in the scope of this thesis managed to solve both platform deployment speed and scaling capability issues very well. The programmable add-ons which accelerated the deployment of the digital platforms solved the issues which the deployment teams were facing, and the first deployment areas could be deployed earlier. On the other hand, the add-ons which enhanced the scaling capability of the digital platforms helped to streamline and automate the business processes in the target organization, and in that way allow the extension and scaling of the digital platform to next deployment areas. The main value of programmable add-ons in both of these cases was found to be the ability to deliver the value in quick and efficient manner to the digital platforms and to the teams deploying and integrating them to the organization.

Based on these conclusions, it is evident that the programmable add-on approach is capable of solving the issues in digital platform deployment and scalability. In addition, the programmable add-ons can deliver the value to the digital platforms quickly and efficiently meaning that for platform deployment teams and platform users the programmable add-ons can present a valuable approach to solve the integration and scalability issues and enhance the platform functionality. The outcome of this thesis has potential to be applied in other use cases and organizations too, but to better understand the generalizability of these results, future research should address the implementation of programmable add-ons in other digital platform ecosystems and in other manufacturing organizations. In addition, more research of digital platform scalability challenges would support the contribution of this and future scientific studies of digital service platforms.

The results of this thesis contribute to the research of digital platform implementations by shining a light on the challenges manufacturers face when deploying digital platforms and developing a solution for solving these issues. These results are important in understanding the realities of the digital platform implementations and the possibilities which are present in overcoming these realities. The digital platform technology manufacturers are currently implementing is, at least in most parts, still developing and as long as the technology and customer requirements continue to develop, there will be a continuous need for researching the implementation challenges and solutions in digital platform domain area.

References

Adner, R. (2017). Ecosystem as structure: An actionable construct for strategy. *Journal of Management*. Vol. 43, no. 1, pp. 39–58.

Aheleroff, S., Mostashiri, N., Xu, X. & Zhong, R. Y. (2021). Mass Personalisation as a Service in Industry 4.0: A Resilient Response Case Study. *Advanced Engineering Informatics*. Vol. 50, article no: 101438 [no pagination].

Alshuqayran, N., Ali, N. & Evans, R. (2016). A systematic mapping study in microservice architecture. In: *2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA)*, pp. 44–51.

Auer, F., Lenarduzzi, V., Felderer, M. & Taibi, D. (2021). From monolithic systems to Microservices: An assessment framework. *Information and Software Technology*. Vol. 137, article no: 106600 [no pagination].

Baines, T. S. & Lightfoot, H. W. (2014). Servitization of the manufacturing firm: Exploring the operations practices and technologies that deliver advanced services. *International Journal of Operations & Production Management*. Vol. 34, no. 1, pp. 2–35.

Baines, T. S., Lightfoot, H. W., Benedettini, O. & Kay, J. M. (2009). The servitization of manufacturing: a review of literature and reflection on future challenges. *Journal of Manufacturing Technology Management*. Vol. 20, no. 5, pp. 547–567.

Balalaie, A., Heydarnoori, A. & Jamshidi, P. (2016). Microservices architecture enables devops: Migration to a cloud-native architecture. *IEEE Software*. Vol. 33, no. 3, pp. 42–52.

Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., Mitchell, N., Muthusamy, V., Rabbah, R., Slominski, A. & Suter, P. (2017). Serverless computing: Current trends and open problems. In: *Research advances in cloud computing*, Springer: Singapore, pp. 1–20.

Bask, A., Lipponen, M., Rajahonka, M. & Tinnilä, M. (2010). The concept of modularity: diffusion from manufacturing to service production. *Journal of Manufacturing Technology Management*. Vol. 23, no. 3, pp. 355–375.

Baškarada, S., Nguyen, V. & Koronios, A. (2020). Architecting microservices: Practical opportunities and challenges. *Journal of Computer Information Systems*. Vol. 60, no. 5, pp. 428–436.

Bauer, W., Hämmerle, M., Schlund, S. & Vocke, C. (2015). Transforming to a hyper-connected society and economy – towards an “Industry 4.0”. *Procedia Manufacturing*. Vol. 3, no. 1, pp. 417–424.

Benedettini, O., Swink, M. & Neely, A. (2017). Examining the influence of service additions on manufacturing firms' bankruptcy likelihood. *Industrial Marketing Management*. Vol. 60, no. 1, pp. 112–125.

Blinowski, G., Ojdowska, A. & Przybyłek, A. (2022). Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation. *IEEE Access*. Vol. 10, no. 1, pp. 20357–20374.

Bondel, G., Landgraf, A. & Matthes, F. (2021). API Management Patterns for Public, Partner, and Group Web API Initiatives with a Focus on Collaboration. *26th European Conference on Pattern Languages of Programs*. pp. 1–17.

Brax, S. A. & Visintin, F. (2017). Meta-model of servitization: The integrative profiling approach. *Industrial Marketing Management*. Vol. 60, no. 1, pp. 17–32.

Brax, S. A., Bask, A., Hsuan, J. & Voss, C. (2017). Service modularity and architecture–an overview and research agenda. *International Journal of Operations & Production Management*. Vol. 37, no. 6, pp. 686–702.

Brechtel, M. & Altmann, S. (2021). Digital platforms for inter-firm collaborations-identifying current challenges. In: *Proceedings of the ISPIM Conference Proceedings. International Society for Professional Innovation Management (ISPIM)*, pp. 1–9.

Bresnahan, T. & Greenstein, S. (2014). Mobile Computing: The Next Platform Rivalry. *The American Economic Review*. Vol. 104, no. 5, pp. 475–480.

Brunetti, F., Matt, D. T., Bonfanti, A., De Longhi, A., Pedrini, G. & Orzes, G. (2020). Digital transformation challenges: strategies emerging from a multi-stakeholder approach. *The TQM Journal*. Vol. 32, no. 4, pp. 697–724.

Cenamor, J., Sjödin, D. & Parida, V. (2017). Adopting a platform approach in servitization: Leveraging the value of digitalization. *International Journal of Production Economics*. Vol. 192, no. 1, pp. 54–65.

Chan, C. M. L., Teoh, S. Y., Yeow, A. & Pan, G. (2018). Agility in responding to disruptive digital innovation: Case study of an SME. *Information Systems Journal*. Vol. 29, no. 2, pp. 436–455.

Chavan, P. U., Murugan, M. & Chavan, P. P. (2015). A review on software architecture styles with layered robotic software architecture. In: *2015 International Conference on Computing Communication Control and Automation*, pp. 827–831.

Coito, T., Martins, M. S., Viegas, J. L., Firme, B., Figueiredo, J., Vieira, S. M. & Sousa, J. M. (2020). A middleware platform for intelligent automation: An industrial prototype implementation. *Computers in industry*. Vol. 123, article no: 103329 [no pagination].

Constantinides, P., Henfridsson, O. & Parker, G. G. (2018). Introduction—platforms and infrastructures in the digital age. *Information Systems Research*. Vol. 29, no. 2, pp. 381–400.

Coreynen, W., Matthyssens, P. & van Bockhaven, W. (2017). Boosting servitization through digitization: Pathways and dynamic resource configurations for manufacturers. *Industrial Marketing Management*. Vol. 60, no. 1, pp. 42–53.

Coulouris, G. F., Dollimore, J., & Kindberg, T. (2005). Distributed systems: concepts and design. *Pearson Education*.

De Reuver, M., Sørensen, C. & Basole, R. C. (2018). The digital platform: a research agenda. *Journal of information technology*. Vol. 33, no. 2, pp. 124–135.

Dillon, T., Wu, C. & Chang, E. (2010). Cloud Computing: Issues and Challenges. *2010 24th IEEE International Conference on Advanced Information Networking and Applications*. pp. 27–33.

Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R. & Safina, L. (2017). Microservices: Yesterday, Today, and Tomorrow. In: Mazzara, M., Meyer, B. (eds) *Present and Ulterior Software Engineering*. Springer: Cham.

Dunkel, J., Fernández, A., Ortiz, R. & Ossowski, S. (2011). Event-driven architecture for decision support in traffic management systems. *Expert Systems with Applications*. Vol. 38, no. 6, pp. 6530–6539.

Eloranta, V. & Turunen, T. (2016). Platforms in service-driven manufacturing Leveraging complexity by connecting, sharing, and integrating. *Industrial Marketing Management*. Vol. 55, no. 1, pp. 178–186.

Ethiraj, S. K., Levinthal, D. & Roy, R. R. (2008). The dual role of modularity: Innovation and imitation. *Management Science*. Vol. 54, no. 5, pp. 939–955.

Fahmideh, M. & Zowghi, D. (2020). An exploration of IoT platform development. *Information Systems*. Vol. 87, article no: 101409 [no pagination].

Ferreira, F. N. H., Cova, B., Spencer, R. & Proença, J. F. (2016). A dynamics-based approach to solutions typology: A case from the aerospace industry. *Industrial Marketing Management*. Vol. 58, no. 1, pp. 114–122.

Gartner (2016). Building a Digital Business Technology Platform. Gartner Research. Available at: https://www.gartner.com/en/doc/3340817-building-a-digital-business-technology-platform.

Gautschi, W. (2012) Numerical Analysis. Second Edition. Boston: Birkhäuser Boston.

Gawer, A. & Cusumano, M. A. (2014). Industry platforms and ecosystem innovation. *Journal of Product Innovation Management*. Vol. 31, no. 3, pp. 417–433.

Gerbauer, H., Fleisch, E. & Friedli, T. (2005). Overcoming the Service Paradox in Manufacturing Companies. *European Management Journal*. Vol. 23, no. 1, pp. 14–26.

Gerring, J. (2004). What Is a Case Study and What Is It Good for?. *American Political Science Review*. Vol. 98, no. 2, pp. 341–354.

Ghazawneh, A. & Hendfridsson, O. (2013). Balancing platform control and external contribution in third-party development: The boundary resources model. *Information Systems Journal*. Vol. 23, no. 2, pp. 173–192.

Gódor, B., Jakabfy, T. & Sallai, G. (2008). The Concept and Model of Unique Name Service for Convergent Networks. In: *Networks 2008 – The 13th International Telecommunications Network Strategy and Planning Symposium*, pp. 1–7.

Guth, J., Breitenbücher, U., Falkenthal, M., Fremantle, P., Kopp, O., Leymann, F. & Reinfurt, L. (2018). A detailed analysis of IoT platform architectures: concepts, similarities, and differences. In: *Internet of everything*. Springer: Singapore, pp. 81–101.

Götz, M. & Jankowska, B. (2017). Clusters and Industry 4.0–do they fit together?. *European Planning Studies*. Vol. 25, no. 9, pp. 1633–1653.

Hanelt, A., Bohnsack, R., Marz, D. & Antunes Marante, C. (2021). A systematic review of the literature on digital transformation: Insights and implications for strategy and organizational change. *Journal of Management Studies*. Vol. 58, no. 5, pp. 1159–1197.

Hannousse, A. & Yahiouche, S. (2021). Securing microservices and microservice architectures: A systematic mapping study. *Computer Science Review*. Vol. 41, article no: 100415 [no pagination].

Hansrod, A., Smith, S., Coppack, A. & Grant, E. (2022). Digital Platform Playbook. [e-book]. Available at: https://digital-platform.playbook.ee/

Hein, A., Böhm, M. & Krcmar, H. (2018). Tight and Loose Coupling in Evolving Platform Ecosystems: The Cases of Airbnb and Uber. In: Abramowicz, W., Paschke, A. (eds.) Business Information Systems. *BIS 2018. Lecture Notes in Business Information Processing*. Vol 320. Springer: Cham

Henriette, E., Feki, M. & Boughzala, I. (2016). Digital Transformation Challenges. *MCIS 2016 Proceedings*. Paphos, Cyprus, September 2016. Vol. 33.

Hirsjärvi, S., Remes, P. & Sajavaara, P. (2009). Tutki ja kirjoita. 15th ed. Helsinki: Tammi.

Horváth, D. & Szabó, R. Z. (2019). Driving forces and barriers of Industry 4.0: Do multinational and small and medium-sized companies have equal opportunities?. *Technological Forecasting & Social Change*. Vol. 146, no. 1, pp. 119–132.

Innerbichler, J., Gonul, S., Damjanovic-Behrendt, V., Mandler, B. & Strohmeier, F. (2017). NIMBLE collaborative platform: Microservice architectural approach to federated IoT. In: *2017 Global Internet of Things Summit (GIoTS)*, pp. 1–6.

Jiang Z., Feng G., Yi Z. & Guo X. (2021), Service-oriented manufacturing: A literature review and future research directions. *Frontiers of Engineering Management*. Vol. 9, no. 1, pp. 71–88.

Kamal M. M., Sivarajah U., Bigdeli A. Z., Missi F. & Koliousis Y. (2020). Servitization implementation in the manufacturing organisations: Classification of strategies, definitions, benefits and challenges. *International Journal of Information Management*. Vol. 55, no. 1, pp. 102–206.

Kamalaldin, A., Linde, L., Sjödin, D. & Parida, V. (2020). Transforming provider-customer relationships in digital servitization: A relational view on digitalization. *Industrial Marketing Management*. Vol. 89, no. 1, pp. 306–325.

Keshavarzian, A., Sharifian, S. & Seyedin, S. (2019). Modified deep residual network architecture deployed on serverless framework of IoT platform based on human activity recognition application. *Future Generation Computer Systems*. Vol. 101, no. 1, pp. 14–28.

Kindström, D. & Kowalkowski, C. (2014). Service innovation in product-centric firms: a multidimensional business model perspective. *Journal of Business & Industrial Marketing*. Vol. 29, no. 1, pp. 96–111.

Kohtamäki, M., Parida, V., Oghazi, O., Gerbauer, H. & Baines, T. (2019). Digital servitization business models in ecosystems: A theory of the firm. *Journal of Business Research*. Vol. 104, no. 1, pp. 380–392.

Kohtamäki, M., Parida, V., Patel, P. C. & Gebauer, H. (2020). The relationship between digitalization and servitization: The role of servitization in capturing the financial potential of digitalization. *Technological Forecasting and Social Change*. Vol. 151, article no: 119804 [no pagination].

Lasi, H., Fettke, P., Kemper, H-G., Feld, T. & Hoffman, M. (2014). Industry 4.0. *Business & Information Systems Engineering*. Vol. 6, no. 1, pp. 239–242.

Lebens, M., Finnegan, R. J., Sorsen, S. C. & Shah, J. (2022). Rise of the Citizen Developer. *Muma Business Review*. Vol. 5, no. 1, pp. 101–111.

Lee, J., Kao, H-A. & Yang, S. (2014). Service innovation and smart analytics for Industry 4.0 and big data environment. *6th CIRP Conference on Industrial Product Service Systems, IPSS 2014*. Vol. 16, no. 1, pp. pp. 3–8.

Lewis, J. & Fowler, M. (2014). Microservices. Available at: https://martinfowler.com/articles/microservices.html.

Liu, Z., Sampaio, P., Pishchulov, G., Mehandjiev, N., Cisneros-Cabrera, S., Schirrmann, A., Jiru, F. & Bnouhanna, N. (2022). The architectural design and implementation of a digital platform for Industry 4.0 SME collaboration. *Computers in Industry*. Vol. 138, article no: 103623 [no pagination].

Lusch, R. F. & Nambisan, S. (2015). Service innovation: A service-dominant logic perspective. *MIS Quarterly: Management Information Systems*. Vol. 39, no. 1, pp. 155–175.

Martin, S. L., Javalgi, R. G. & Ciravegna, L. (2018). Service advantage built on service capabilities: An empirical inquiry of international new ventures. *Journal of Business Research*. Vol. 88, no. 1, pp. 371–381.

Martin, P. C. G., Schroeder, A. & Bigdeli, A. Z. (2019). The value architecture of servitization: Expanding the research scope. *Journal of Business Research*. Vol. 104, no. 1, pp. 438–449.

Martinez, V., Bastl, M., Kingston, J. & Evans, S. (2010). Challenges in transforming manufacturing organisations into product-service providers. *Journal of Manufacturing Technology Management*. Vol. 21, no. 4, pp. 449–469.

Mazlami, G., Cito, J. & Leitner, P. (2017). Extraction of microservices from monolithic software architectures. In: *2017 IEEE International Conference on Web Services (ICWS)*, pp. 524–531.

McGuire, R. (2019). Serverless Compute – Logic Apps, Functions and Event Grid. [web]. Available at: https://devblogs.microsoft.com/premier-developer/serverless-compute-logic-apps-functions-and-event-grid/. [Referenced: 17.9.2022]

Microsoft (2022a). What is Azure Form Recognizer?. [web]. Available at: https://learn.microsoft.com/en-us/azure/applied-ai-services/form-recognizer/overview?view=form-recog-3.0.0. [Referenced: 24.9.2022].

Microsoft (2022b). Power Automate vs Logic Apps. [web]. Available at: https://learn.microsoft.com/en-us/microsoft-365/community/power-automate-vs-logic-apps. [Referenced: 24.9.2022].

Müller, J. M. (2019). Antecedents to digital platform usage in Industry 4.0 by established manufacturers. *Sustainability*. Vol. 11, no. 4, pp. 1121.

Nargesian, F., Zhu, E., Miller, R. J., Pu, K. Q. & Arocena, P. C. (2019). Data lake management: challenges and opportunities. *Proceedings of the VLDB Endowment*. Vol. 12, no. 12, pp. 1986–1989.

Oliva, R. & Kallenberg, R. (2003). Managing the Transition from Products to Services. *International Journal of Service Industry Management*. Vol. 14, no. 2, pp. 160–172.

Orton, J. D., & Weick, K. E. (1990). Loosely coupled systems: A reconceptualization. *Academy of management review*. Vol. 15, no. 2, pp. 203–223.

Patni, S. (2017). API Platform and Data Handler. In: *Pro RESTful APIs*. Berkeley, CA: Apress.

Pekkarinen, S. & Ulkuniemi, P. (2008). Modularity in developing business services by platform approach. *The International Journal of Logistics Management*. Vol. 19, no. 1, pp. 84–103

Porter, M.E. & Heppelmann, J.E. (2014). How smart, connected products are transforming competition. *Harvard Business Review*. Vol. 92 no. 11, pp. 64–88.

Rabetino, R. & Kohtamäki, M. (2018). To Servitize is to (re) position: Utilizing a Porterian view to understand Servitization and value systems. In: *Practices and tools for Servitization*. Palgrave Macmillan, Cham, pp. 325–341.

Sahay, A., Di Ruscio, D. & Pierantonio, A. (2020). Understanding the role of model transformation compositions in low-code development platforms. *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, pp. 1–5.

Schreieck, M., Wiesche, M. & Krcmar, H. (2021). Capabilities for value co-creation and value capture in emergent platform ecosystems: A longitudinal case study of SAP’s cloud platform. *Journal of Information Technology*. Vol. 36, no. 4, pp. 365–390.

Schmidt, M. C., Veile, J. W., Müller, J. M. & Voigt, K. I. (2019). Kick-Start for Connectivity: How to Implement Digital Platforms Successfully in Industry 4.0. *Technology Innovation Management Review*, Vol. 9, no. 10, pp. 5–15.

Settanni, E., Newnes, L. B., Thenent, N. E., Parry, G. & Goh, Y. M. (2014). A through-life costing methodology for use in product–service-systems. *International Journal of Production Economics*. Vol. 153, no. 1, pp. 161–177.

Shamsuddin, A., Sheikh, A. & Keers, R. N. (2021). Conducting Research Using Online Workshops During COVID-19: Lessons for and Beyond the Pandemic. *International Journal of Qualitative Methods*. Vol. 20, no. 1, pp. 1-7.

Shasi, C. & Sinha, M. (2021). Digital transformation: challenges faced by organizations and their potential solutions. *International Journal of Innovation Science*. Vol. 13, no. 1, pp. 17–33.

Shivakumar, S. K. & Sethii, S. (2019). Building Digital Experience Platforms A Guide to Developing Next-Generation Enterprise Applications. Berkeley, CA: Apress.

Silva, H. & Soares, A. L. (2021). Digital Platforms as Enablers of Smart Product-Service Systems. In: Camarinha-Matos, L. M., Boucher, X. & Afsarmanesh, H. (eds.) Smart and Sustainable Collaborative Networks 4.0. PRO-VE 2021. *IFIP Advances in Information and Communication Technology*. Springer: Cham. Vol. 629.

Silvestro, R. & Lustrato, P. (2015). Exploring the "mid office" concept as an enabler of mass customization in services. *International Journal of Operations & Production Management*. Vol. 35, no. 6, pp. 866–894.

Sivula, A., Shamsuzzoh, A., Ndzibah, E. & Timilsina, B. (2022). End-to-End Servitization Model in Industry 4.0. *Management and Production Engineering Review*. Vol. 13, no. 1, pp. 89–98.

Sjödin, D., Parida, V. & Kohtamäki, M. (2016). Capability configurations for advanced service offerings in manufacturing firms: Using fuzzy set qualitative comparative analysis. *Journal of Business Research*. Vol. 69, no. 11, pp. 5330–5335.

Smith, W. K., Binns, A. & Tushman, M. L. (2010). Complex Business Models: Managing Strategic Paradoxes Simultaneously. *Long Range Planning*. Vol. 43, no. 2-3, pp. 448–461.

Stora Enso (2022a). Annual Report 2021. Available at: https://www.storaenso.com/-/media/documents/download-center/documents/annual-reports/2021/storaenso_annual_report_2021.ashx.

Stora Enso (2022b). Packaging Materials. Stora Enso Intranet.

Strutynska, I., Kozbur, G., Dmytrotsa, L., Sorokivska, O. & Melnyk, L. (2019). Influence of Digital Technology on Roadmap Development for Digital Business Transformation. In: *2019 9th International Conference on Advanced Computer Information Technologies (ACIT)*, pp. 333–337.

Terroso-Saenz, F., González-Vidal, A., Ramallo-González, A. P. & Skarmeta, A. F. (2019). An open IoT platform for the management and analysis of energy data. *Future generation computer systems*, Vol. 92, no. 1, pp. 1066–1079.

Theodorou, V., Gerostathopoulos, I., Alshabani, I., Abelló, A. & Breitgand, D. (2021). MEDAL: An AI-Driven Data Fabric Concept for Elastic Cloud-to-Edge Intelligence. *International Conference on Advanced Information Networking and Applications*. Springer: Cham, pp. 561–571.

Thoring, K., Mueller, R. & Badke-Schaub, P. (2020). Workshops as a research method: Guidelines for designing and evaluating artifacts through workshops. In: *Proceedings of the 53rd Hawaii International Conference on System Sciences*.

Tian, J., Coreynen, W., Matthyssens, P. & Shen, L. (2021). Platform-based servitization and business model adaptation by established manufacturers. *Technovation*. Vol. 118, article no: 102222 [no pagination].

Tiwana, A. (2014). Platform Ecosystems: Aligning Architecture, Governance, and Strategy. San Francisco: Elsevier Science & Technology.

Tiwana, A. & Konsynski, B. (2010). Complementarities between organizational IT architecture and governance structure. *Information Systems Research*, Vol. 21, no. 2, pp. 288–304.

Vandermerwe, S. & Rada, J. (1989). Servitization of business: adding value by adding services. *European Management Journal*. Vol. 6, no. 4, pp. 314–324.

Varghese, B. & Buyya, R. (2017). Next generation cloud computing: New trends and research directions. *Future Generation Computer Systems*. Vol. 79, no. 1, pp. 849–861.

Vendrell-Herrero, F., Bustinza, O. F., Parry, G. & Georgantzis, N. (2017). Servitization, digitization and supply chain interdependency. *Industrial Marketing Management*. Vol. 60, no. 1, pp. 69–81.

Verba, N., Chao, K. M., Lewandowski, J., Shah, N., James, A. & Tian, F. (2019). Modeling industry 4.0 based fog computing environments for application analysis and deployment. *Future Generation Computer Systems*. Vol. 91, no. 1, pp. 48–60.

Villamizar, M., Garcés, O., Castro, H., Verano, M., Salamanca, L., Casallas, R. & Gil, S. (2015). Evaluating the monolithic and the microservice architecture pattern to deploy web applications in the cloud. In: *2015 10th Computing Colombian Conference (10CCC)*, pp. 583–590.

Wasserman, J. (2020). Investing in Developer Tools: Why Low Code Platforms are on the Rise. [web]. Available at: https://www.volitioncapital.com/news/investing-in-developer-tools-why-low-code-platforms-are-on-the-rise/. [Referenced: 2.10.2022].

Yablonsky, S. (2018). A multidimensional framework for digital platform innovation and management: from business to technological platforms. *Systems Research and Behavioral Science*. Vol. 35, no. 4, pp. 485–501.

Yoo, Y., Henfridsson, O. & Lyytinen, K. (2010). Research commentary—the new organizing logic of digital innovation: an agenda for information systems research. *Information systems research*. Vol. 21, no. 4, pp. 724–735.

Özcan, L., Koldewey, C., Duparc, E., van der Valk, H. & Otto, B. (2022). Why do Digital Platforms succeed or fail? - A Literature Review on Success and Failure Factors. *AMCIS 2022 Proceedings*. Minneapolis, USA. Vol. 15.

Ørngreen, R. & Levinsen, K. (2017). Workshops as a Research Methodology. *Electronic Journal of E-learning*. Vol. 15, no. 1, pp. 70–81.

Appendix 1. Training dataset size for each carrier, and machine learning model confidence results for intelligent document processing add-on

Carrier	Document count	BL number	Book request	Carrier name	Document type	Millorderlines	Millorderlines: Order line	Millorderlines: Millorderno	Voyage number	**Avg.**
Carrier 1	10	99.5 %	99.5 %	99.5 %	99.5 %	70.0 %	70.0 %	70.0 %	99.5 %	**88.4 %**
Carrier 2	15	99.5 %	99.5 %	99.5 %	99.5 %	79.2 %	82.3 %	76.0 %	99.5 %	**91.9 %**
Carrier 3	14	92.9 %	99.5 %	78.6 %	99.5 %	57.7 %	51.8 %	63.7 %	99.5 %	**80.4 %**
Carrier 4	12	91.7 %	91.7 %	99.5 %	91.7 %	57.6 %	65.3 %	50.0 %	91.7 %	**79.9 %**
Carrier 5	10	99.5 %	99.5 %	99.5 %	99.5 %	56.2 %	55.0 %	57.5 %	99.5 %	**83.3 %**
Carrier 6	10	99.5 %	90.0 %	99.5 %	60.0 %	56.2 %	57.5 %	55.0 %	99.5 %	**77.2 %**
Carrier 7	13	99.5 %	99.5 %	99.5 %	99.5 %	41.7 %	36.1 %	47.2 %	99.5 %	**77.8 %**
Carrier 8	8	99.5 %	99.5 %	99.5 %	99.5 %	81.2 %	91.7 %	79.2 %	99.5 %	**93.7 %**
Total/ Average	**92**	**97.7 %**	**97.3 %**	**96.9 %**	**93.6 %**	**62.5 %**	**63.7 %**	**62.3 %**	**98.5 %**	**84.1 %**