UNIVERSITY OF LJUBLJANA
SCHOOL OF ECONOMICS AND BUSINESS

MASTER'S THESIS

MICROFOUNDATIONS OF VALUE COCREATION: USING A MULTILEVEL DESIGN APPROACH TO BUILDING SYSTEMS OF ENGAGEMENT BY UTILIZING SERVICE DOMINANT ARCHITECTURE

Pforzheim, September 2021 NORMAN BÄCHTOLD

AUTHORSHIP STATEMENT

I, the undersigned Norman Bächtold, a student at the University of Ljubljana, School of Economics and Business, (hereafter: SEB LU), author of this written final work of studies with the title "Microfoundations of Value Cocreation: Using a Multilevel Design Approach to Building Systems of Engagement by Utilizing Service Dominant Architecture", prepared under supervision of Prof. Mojca Indihar Štemberger and co-supervision of Prof. Peter Weiß

D E C L A R E

1. this written final work of studies to be based on the results of my own research;

2. the printed form of this written final work of studies to be identical to its electronic form;

3. the text of this written final work of studies to be language-edited and technically in adherence with the SEB LU's Technical Guidelines for Written Works, which means that I cited and / or quoted works and opinions of other authors in this written final work of studies in accordance with the SEB LU's Technical Guidelines for Written Works;

4. to be aware of the fact that plagiarism (in written or graphical form) is a criminal offence and can be prosecuted in accordance with the Criminal Code of the Republic of Slovenia;

5. to be aware of the consequences a proven plagiarism charge based on the this written final work could have for my status at the SEB LU in accordance with the relevant SEB LU Rules;

6. to have obtained all the necessary permits to use the data and works of other authors which are (in written or graphical form) referred to in this written final work of studies and to have clearly marked them;

7. to have acted in accordance with ethical principles during the preparation of this written final work of studies and to have, where necessary, obtained permission of the Ethics Committee;

8. my consent to use the electronic form of this written final work of studies for the detection of content similarity with other written works, using similarity detection software that is connected with the SEB LU Study Information System;

9. to transfer to the University of Ljubljana free of charge, non-exclusively, geographically and time-wise unlimited the right of saving this written final work of studies in the electronic form, the right of its reproduction, as well as the right of making this written final work of studies available to the public on the World Wide Web via the Repository of the University of Ljubljana;

10. my consent to publication of my personal data that are included in this written final work of studies and in this declaration, when this written final work of studies is published.

Pforzheim, October 15th, 2021 Author's signature: 

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

A2A	Actor-to-Actor
ADR	Action Design Research
App.	Appendix
CEO	Chief Executive Officer
CRM	Customer Resource Management
CTO	Chief Technology Officer
DSR	Design Science Research
DDD	Domain-Driven Design
DSRM	DSRM Process Model
DT	Digital Transformation
EA	Enterprise Architecture
EAM	Enterprise Architecture Management
ERP	Enterprise Resource Planning
Fig.	Figure
GDL	Goods-Dominant Logic
IS	Information Systems
IT	Information Technology
MVP	Minimal Viable Product
SDA	Service-Dominant Architecture
SDL	Service-Dominant Logic
SoE	Systems of Engagement
SOI	System of Interaction
SOOP	System of Operant Resources
SOP	System of Participation
SoR	Systems of Record
UML	Unified Modeling Language

INTRODUCTION

With digital technology accelerating our lives more and more, companies are in a race to keep up with the new dynamics of their market environments. Gone are the days when traditional marketing tools could predict market demand and behavior and guide appropriate strategic initiatives. Instead, developing a new kind of organizational agility that keeps pace with ever-changing market conditions (Weill, Subramani, & Broadbent, 2002) becomes essential for companies' survival.

A thoroughly planned and executed digital transformation (DT) becomes a key objective in these disruptive times, especially for established firms (Vial, 2019). Herein, the challenge is not updating existing information technology (IT) systems. Many companies fail in their DT because they do not know how to address the needs of their customers and build the capacity to have customers connect and interact with corporate resources. An additional challenge is to align their IT infrastructure to provide the capabilities needed for such interaction and adapt to the market environment's ongoing changes (Weiß, Zolnowski, Warg, & Schuster, 2018).

Recently, information systems (IS) research has begun to project the concepts of service-oriented marketing, and more precisely, service-dominant logic (SDL) by Vargo & Lusch (2004), onto the challenges that a DT poses to firms. SDL is a fairly young school of thought within marketing, but its impact on the discipline is immense. As a counter-proposal to the traditional, goods-dominant logic (GDL) of marketing, the SDL sets out to offer a more complete view of social and economic exchange functioning. It leaves behind the dyadic perspective of the supplier providing value to a customer through the handover of a good. Instead, SDL argues that only lasting value-in-use is beneficial to an actor and that multiple actors mutually create this value in a bounded setting. This concept of value cocreation within a service ecosystem is at the center of the SDL (Vargo & Lusch, 2016).

Seeing that value cannot be created by a single offeror of a product or service but is the result of a group of actors jointly contributing to its creation has inspired IS research to propose strategic approaches to DT. To know the mechanics behind value cocreation allows for a better understanding of the dynamics mentioned above of modern markets. In an attempt to develop strategic approaches to DT, service science (Spohrer & Maglio, 2008) and, more precisely, the science of service innovation (Lusch & Nambisan, 2015) set out to find ways to transfer the mechanisms of value cocreation on enterprise architectures (EA). The argument is that if a firm can support the natural process of value cocreation, it can host a service ecosystem that continuously adapts its value-creating processes to the changing conditions of the surrounding market environment. The result is ongoing innovation within the service ecosystem (Koskela-Huotari, Edvardsson, Jonas, Sörhammar, & Witell, 2016).

To increase organizational agility and responsiveness, firms must evolve their EA and develop new organizational capabilities specific to service-based business models. Especially the introduction of new IT infrastructure capabilities is of interest since they are significant contributors to strategic agility (Weill et al., 2002, p. 61). The service-dominant architecture (SDA) recognizes this relationship and presents a framework that assists practitioners in building an SDL-inspired IT infrastructure. More specifically, it conceptualizes a new systems layer, the systems of engagement (SoE), on top of an organization's existing IT-landscape, the systems of records (SoR), as described by Moore (2011). SoE provide "all required subsystems and elements to interact with customers and to collaborate with other actors" (Warg & Deetjen, 2021, p. 10). Hence, the SDA offers guidance in establishing a service-oriented EA and acts thus as the foundation for execution of digital strategies (Weiß et al., 2018).

But building those service systems remains a complex task. In fact, solely designing service systems from an abstract or too technical perspective leads to poor performance, as this will miss out important considerations about the configuration of actors and resources and their interplay (Edvardsson, Skalen, & Tronvoll, 2012, p. 87; Pinho, Beirão, Patrício, & Raymond, 2014). Therefore, a holistic and actor-centered service system design approach is needed (Rittweger, Kronibus, & Weiß, 2020, p. 87). This approach has recently been motivated by the research on microfoundations for value cocreation (Storbacka, Brodie, Böhmann, Maglio, & Nenonen, 2016). It informs conceptualizations of an improved version of the SDA (Weiß, Kronibus, Riedel, & Rittweger, 2021) and a multilevel approach to service systems engineering (Grotherr et al., 2020; Grotherr, Semmann, & Böhmann, 2018). The chance is there to combine the ideas of these two streams of research to develop the SDA into a "design environment to design and operate service systems" (Rittweger et al., 2020, p. 119).

This is where the present work comes in. Storbacka et al. (2016) have significantly contributed to a better understanding of the functioning of value cocreation. The multilevel design process (Grotherr et al., 2018) uses this understanding to drive a holistic approach to the design of service-based EAs. What the two papers lack, however, is a connection to practical use cases. As a result, they fail to make their ideas applicable in real-world environments as they have to resort to abstract concepts difficult for practitioners to transfer. For another, they lack a target image that could guide users in their attempt to derive a digital strategy from their theoretical bodies. It is on these points that the SDA comes into play. Compared to related work, it provides a bridge into natural business environments because it is developed further using case studies (Warg & Deetjen, 2021; Weiß et al., 2021). Moreover, its conceptualization provides the target image of a service platform that can support value cocreation and service innovation in a service ecosystem (Weiß et al., 2018).

The evolution of SDA into a holistic design environment for the design and operation of service systems promises to be a relevant contribution to practice. It can help companies successfully enter their DT as it promises to address EA adaptation holistically. Research on

SDA thus contributes to ensuring the future viability of companies under the light of increasingly dynamic markets and societies.

Consequently, the purpose of this research is to provide practitioners and especially managers with a deep understanding of service-oriented DT. Specifically, its purpose is to give knowledge on designing, implementing, and continuously evolving service systems to foster service innovation. This purpose can be achieved by combining the SDA with a multilevel service systems design framework to create a new service systems design framework. Therefore, the research question of this thesis is "How can a multilevel service system design approach be applied to the design of effective systems of engagement using the Service-Dominant Architecture?"

Drawing from design science research (DSR) by Hevner, March, & Park (2004), the presented thesis first approached its research problem by establishing a theoretical knowledge base through a critical literature review. It then continued to examine the current development state of the SDA by conducting a case study (Yin, 2018) involving two research-related companies. The insights generated by these two strands of work informed the subsequent development of a new SDA artifact that combines the multilevel perspective of value cocreation with a practical use-case of SDA research, SimplyGo (Weiß et al., 2021).

The structure of this research paper is according to the chosen methodology. Chapter 1 presents the critical literature review of works related to the topics. Divided into three sections, the review examines the outer boundaries of this work first. It then goes over to establish a thorough understanding of service innovation before it discusses the topics of EA, the SDA, and the multilevel design process for designing service systems. Chapter 2 introduces the research project's methodology in more detail. It discusses the research framework, the DSR approach, and the design of the case study. Chapter 3 then presents the case study in detail. It first introduces the two involved companies and presents then the case study results. In the end, it summarizes the key takeaways that also serve as the practical contribution to the subsequent artifact development. This research step is then presented and discussed in chapter 4, describing the development approach and the project results. The second half of the chapter offers a general discussion of the findings of the thesis. Finally, chapter 5 then concludes the research project.

1 CRITICAL LITERATURE REVIEW

The following critical literature review aims to provide an overview of the scientific knowledge that is relevant and accessible to the topic of service innovation (Saunders, Lewis, & Thornhill, 2016, p. 74). It aims to collect, condense, and discuss this knowledge. A series of questions were developed to emphasize different aspects of the available body of knowledge and guide the review (Saunders et al., 2016, p. 74). First, the question is, what is the broader context of this research? Second, how can service innovation lead to successful DT? Third, how can SDA provide a conceptual foundation for implementing digital service

strategies? These guiding questions inform the structure of the critical literature review. The literature used has been obtained in a twofold approach, using a keyword search on the discussed topics and reviewing the cited works in main contributions to the discussed issues. The predominant use of high-quality journals and conferences in the critical literature review ensures quality and representativeness.

The critical literature review is structured in the following way: It first presents the broader context of the work formed by DT and SDL. Then, answering the second question will explore innovation in IS research and the meaning of service innovation. Finally, the discussion of the third question leads to examining EA, service systems design and engineering, and the SDA.

1.1 The Broader Context of the Research

We all encounter the term "digital transformation" almost every day these days. It is on everyone's lips and keeps politicians (BMWi, 2021), scientists (Fraunhofer IPT, 2021), and businesses (Red Hat, 2021b) equally engaged. But a closer look reveals that the use of the term and its interpretation sometimes diverge significantly. Similar applies to the approaches to dealing with the issue in practice. In the meantime, however, it seems to have been proven that a clearly defined corporate strategy is required to master DT successfully (Vial, 2019). As a marketing theory, SDL has entered IS research in recent years. It explains how social and economic exchange works and thus provides a solid knowledge base for developing service-based digital strategies (Lusch & Nambisan, 2015). The following subsections will dive into DT and SDL to help delineate the outer boundaries of this thesis and place it in context.

1.1.1 Digital Transformation

The extent of DT can be observed in all societies and industries, where new digital developments often bring about far-reaching changes in terms of human and organizational behavior and thus competitive dynamics (Agarwal, Johnson, & Lucas, 2011). To remain relevant and therefore competitive, organizations must rethink the way they operate. IS research has recently focused on how new digital strategies that merge business and IT strategy can help transform organizations into innovative, future-ready entities (Bharadwaj, El Sawy, Pavlou, & Venkatraman, 2013; Matt, Hess, & Benlian, 2015). However, although the available literature from both academia and practitioners on DT is vast, there seemed to be a lack of shared understanding of what the term indeed describes until not long ago. This changed when Vial (2019) published his seminal work on DT, in which he conducted an extensive literature review to gather the current IS research knowledge on the topic. Based on the results, he derived a conceptual definition of the term, built an inductive framework of DT, and proposed a future research agenda. Given the influence of Vial's work in younger academic literature, the paper provides the starting point for exploring DT. Where necessary,

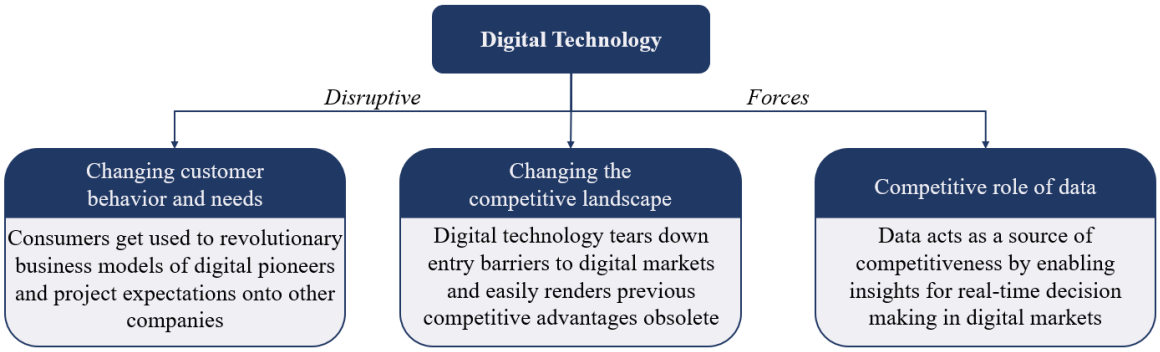
other vital contributions to the understanding of DT are drawn upon to paint a condensed but informed picture of the phenomenon.

The following section highlights the role of digital technology within digital communication. The phenomenon itself is then examined in-depth, while the third subsection assesses the relevance of DT to this work.

1.1.1.1 Digital Technologies as Disruptive Forces

Digital technologies are an essential property to DT. They act as both the cause of disruption and as critical ingredients for digitally infused strategies to address this disruption. (Vial, 2019). IS researchers usually group the most important digital technologies in this context under the SMACIT acronym, introduced by Sebastian et al. (2017; Vial, 2019, p. 122). SMACIT stands for Social, Mobile, Analytics, Cloud, and Internet of Things and represents a stack of easily and quickly accessible technologies that support digital business strategies within DT (Sebastian et al., 2017, pp. 197-198). Other technologies such as blockchain, robotics, or virtual reality typically require more effort to be usable in organizational contexts. Especially in the early stages of a DT, they are therefore not the first choice. However, in more mature phases, they can bring additional benefits (Sebastian et al., 2017, p. 197). As is evident from various sources, combinations of digital technologies, e.g., as platforms, are essential in the context of DT (Vial 2019, p. 122; Bonnet & Westerman, 2021; Westerman et al., 2014; Bharadwaj et al., 2013), too.

Fig. 1: Disruption Caused by Digital Technologies



Adapted from Vial (2019, pp. 122-124).

In his study, Vial identifies three types of disruptive forces digital technologies are responsible for (see Fig. 1): they affect customer behavior and expectations, competitive landscapes, and increase the importance of data (Vial, 2019, p. 122). Consequently, the disruptions triggered by digital technologies primarily affect the organizations' environment and can change it permanently. For example, digital pioneers can benefit from these disruptions. They can quickly enter and reshape existing markets through the innovative use of available digital technologies. Especially for established companies, this poses a challenge, as such

events even accelerate disruptive effects. They are quickly left behind without reacting appropriately to digitally induced disruptions (Sebastian et al., 2017, p. 197).

1.1.1.2 Understanding Digital Transformation

Companies need to arm themselves against the threats posed by these digital disruptions. This is done through a carefully planned and executed DT.

Table 1: Essential Properties of DT

Property of DT	**Description**
Target entity	DT is typically related to individual organizations but can also be applied to more comprehensive entities, such as entire economies that underlie digital disruption.
Scope	Refers to the extent to which the target entity is changed. For example, while Vial specifies DT's scope, Gong & Ribiere speak of broad/extensive change.
Means	Describes the tools used to achieve the significant changes in the target entity. The term refers to digital technologies. Vial names different technologies that fall under SMACIT.
Expected outcome	Improvement of an entity is the expected result of DT. Gong & Ribiere add that its value propositions are redefined, too.

Adapted from Vial (2019, p. 121); Gong & Ribiere (2021).

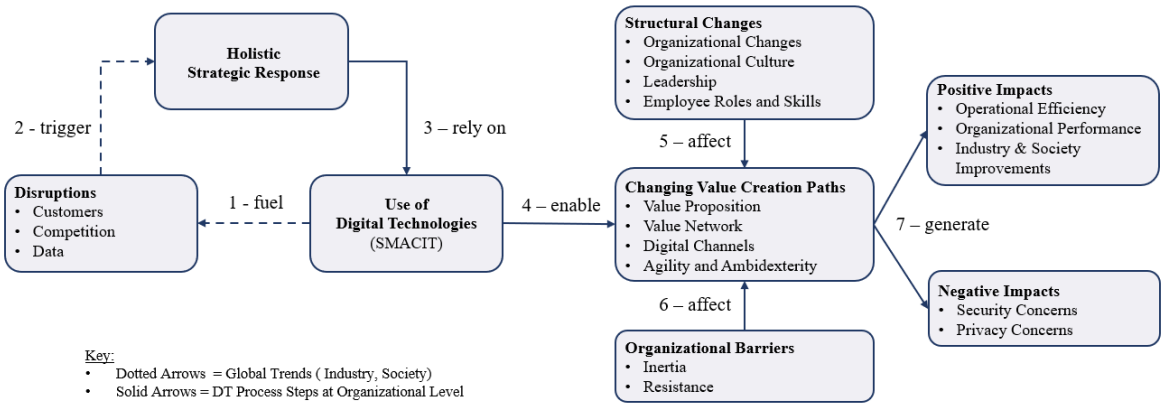
Vial defines a DT as "a process that aims to improve an entity by triggering significant changes to its properties through combinations of information, computing, communication, and connectivity technologies" (2019, p. 121). Gong & Ribiere (2021) have recently given a similar definition of DT. They define it as "a fundamental change process, enabled by the innovative use of digital technologies accompanied by the strategic leverage of key resources and capabilities, aiming to radically improve an entity and redefine its value proposition for its stakeholders" (Gong & Ribiere, 2021, p. 12). The two definitions highlight four essential properties of DT summarized in Tab. 1.

Despite some similarities and minor differences, one addition, in particular, stands out in Gong& Ribier's definition. According to them, change comes about through innovative approaches to using digital technologies in organizational contexts (Gong & Ribiere, 2021, p. 12). Their emphasis on innovation clarifies that digital technologies have been critical before and have always contributed to corporate success. For example, Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) systems were initially a significant factor of competitiveness. These technologies serve a functional purpose, increasing productivity and efficiency (Sebastian et al., 2017, p. 201). With DT, however, the focus of new deployments of digital technologies is changing. They become a central component in

reshaping value propositions to enterprise stakeholders. Hence, they contribute to increased competitiveness and differentiation (Bharadwaj et al., 2013, p. 472).

For companies, the question is how to take advantage of the new opportunities opened by digital technologies. As a central element of his work, Vial has created a framework in which he describes the path from the effects of digital disruptions at the beginning to achieving greater competitiveness as its outcome. Its message is that the use of digital technologies must be targeted and holistic. It requires strategic planning and implementation of transformation projects that take organizations into account in their entirety (Vial, 2019, p. 122). Fig. 2 shows that digital technologies are used within DT primarily to renew a company's value creation paths. The aim is to reconfigure the business model to respond appropriately to digital disruptions. However, this comes with the need for structural changes of the organization and the consideration of organizational barriers (Vial, 2019, pp. 127-129).

Fig. 2: A Framework of DT



Adapted from Vial (2019, p. 122).

Over the past 25 years, three different approaches have emerged as to the perfect DT strategy. Henderson & Venkatraman (1993) presented their idea of the Strategic Alignment Model early on. Bharadwaj et al. (Bharadwaj et al., 2013) criticized this approach. They gave their vision of a digital business strategy, which Matt et al. (Matt et al., 2015) complemented with a digital transformation strategy. Although all three approaches follow different ideas in detail, essential commonalities can be identified. They represent the central requirements for a successful DT.

Organizations must develop new organizational capabilities throughout their DT to become reactive to dynamic market conditions. It is dynamic capabilities (Teece & Pisano, 1994) that Henderson & Venkatraman highlight (Henderson & Venkatraman, 1993, pp. 473-474). Dynamic capabilities enable timely responsiveness and "rapid and flexible product innovation, along with the management capability to effectively coordinate and redeploy internal and external competences" (Teece & Pisano, 1994, p. 537). Weill et al. address a similar skill in their work on IT infrastructures (Weill et al., 2002, p. 61): the ability to implement a

set of business initiatives quickly and purposefully in response to external events is what they call strategic agility.

The different contributions to digital strategies[1] differentiate themselves in notions, but they agree on how organizations should develop new dynamic capabilities. For example, Bharadwaj et al. argue for resharpening the view on digital technologies. In their opinion, digital technologies are essential for carrying new business models that firms establish through a DT. Thus, enterprise IT should follow the business. However, the possibilities that new technologies offer also inform new business model ideas. Therefore, following the resource-based view of the firm (Wernerfelt, 1984), they argue for broadening the term of IT also to include digital resources such as institutions and practices (Bharadwaj et al., 2013, pp. 472-474). The same goes with Henderson & Venkatraman, who present different strategic stages in which either business influences IT strategy or vice versa (Henderson & Venkatraman, 1993, p. 477).

Companies should approach business and IT similarly in their DT. They can ensure this by managing appropriate projects and pausing in regular iterations to create space for interim evaluations. In other words, a DT must alternate strategy formation and implementation phases with stages in which the effects are observed and evaluated (Chanias, Myers, & Hess, 2019). This leads to a recurring, iterative cycle that is different for each organization due to individual circumstances (Hess, Benlian, Matt, & Wiesböck, 2016, p. 17). Following this, DT should be seen as a "moving target" that does not stop once it starts rolling (Chanias et al., 2019, p. 28).

1.1.2 Service-Dominant Logic

The "old" marketing, as universities still teach it today, tries to serve customer needs and be profitable at the same time. It uses classic tools such as market penetration, the "4 P's", and others to do this (Vargo & Lusch, 2004, p. 1). However, since the early 2000s, a change in marketing literature can be observed. Scholars are turning away from time-honored concepts, throwing them overboard, or reconceptualizing them in a new context. Various authors such as Normann (2001) or Grönroos & Voima (2013) have strongly influenced the changing understanding of marketing. However, Lusch & Vargo have undoubtedly made the most significant contribution to this development with their SDL. First presented in 2004 (Vargo & Lusch), the SDL has since been reinforced by many marketing scholars and continuously been updated by the original authors (Vargo & Lusch, 2008, 2016).

The SDL offers a new perspective on how social and economic exchange works. It redefines concepts such as the role of goods and customers, the importance of resources, and the purpose and mechanisms behind the interaction of market actors. According to the SDL, value cocreation is at the heart of all market activity (Koskela-Huotari et al., 2016). The shift from

[1] In the remainder, digital strategy is used as a collective term for a strategy with a focus on DT.

a goods-dominated logic (GDL) to an orientation that sees service as the fundamental purpose of exchange is the essence of SDL (Vargo & Lusch, 2004, pp. 5-6). To communicate their worldview, Lusch & Vargo introduced a change in language that differs from previously accepted goods-oriented marketing scholars and accompanied it with a set of foundational premises and axioms (2014).

The following section introduces the core concepts of SDL. For this purpose, it discusses the five axioms (see Tab. 2) and the theoretical ideas behind them. Subsequently, another section discusses other author's contributions to the service orientation of marketing.

1.1.2.1 The five axioms of SDL

Table 2: Axioms of SDL

Axiom	Description
Axiom 1	Service is the fundamental basis of exchange.
Axiom 2	Value is cocreated by multiple actors, always including the beneficiary.
Axiom 3	All social and economic actors are resource integrators.
Axiom 4	Value is always uniquely and phenomenologically determined by the beneficiary.
Axiom 5	Value cocreation is coordinated through actor-generated institutions and institutional arrangements.

Adapted from Vargo & Lusch (2016, p. 18).

Axiom 1

In its latest version, SDL's axiom 1 states that "service is the fundamental basis of exchange" (Vargo & Lusch, 2008, p. 6). Traditionally, marketing has considered goods as the most critical asset of a company to sell for profit. Hence, the focus is always on profit maximization, achieved through the highest possible standardization of goods production. In this view, service(s) hardly plays any role beyond being an add-on to products (Vargo & Lusch, 2004, p. 5). But with this GDL arise several problems, as identified by Vargo, Koskela-Huotari, & Vink (2020, p. 7): First, it views tangible outputs – preferably in the form of physical goods – as superior compared to other matters of exchange. Second, GDL misses to recognize the role of other parties such as customers or partners in the process of value creation and places the producing firm into the center of attention instead. As indicated in Fig. 1 earlier, customers have an increasingly stronger position towards firms in determining what they should offer thanks to digital technology. Third, GDL's focus is too narrow on determining value in monetary terms.

This orientation has changed under the service-centered orientation of marketing. Here, the purpose of exchange is value creation enabled by the exchange of service between different

actors. Now, value refers to the "comparative appreciation of reciprocal skills or services that are exchanged to obtain utility." Service is understood as "the application of specialized competencies (knowledge and skills) through deeds, processes, and performances for the benefit of another entity or the entity itself" (Vargo & Lusch, 2004, pp. 2, 7). From this understanding, intangible resources such as knowledge, skills, processes, etc., form the basis of any exchange, as they are the foundation of service delivery (Vargo & Lusch, 2008, p. 6). Thus, SDL promotes a shift from a goods-centric perspective to this new service-centric perspective.

The consequence of this shift is also that the role of physical goods is changing. If service is at the center of exchange, then goods become means of transportation that have this service inbuilt in them (Vargo & Lusch, 2004, p. 9). For example, the components of a suitcase contain knowledge about the production of this suitcase and its features. A consumer who buys this suitcase indirectly receives a service from the seller who transmits it through the sale of the good.

This reorientation affects the understanding of goods in general. The SDL emphasizes that there are different types of goods. Knowledge, skills around operational processes, and the like become the central element of service. These so-called operant resources are translated into service delivery - either intangible or tangible in the form of products. On the other hand, operand resources are those resources on which the operand resources act to create value. As Vargo & Lusch state, "operant resources are resources that produce effects," and they can take the form of individual knowledge and skills (2004, p. 2) or be bundled together to form higher-order resources in the state of routines, processes, or operations (2004, p. 5). Based on Zimmermann's statement, "Resources are not, they become" (1951, p. 15). Lusch & Vargo further argue that resources do not exist perse, but through the appraisal and action of an actor, potential resources are filled with what they call resourceness. Resourceness "reflects the quality and realization of potential resources"; thus, without resourceness, possible resources are not available for use (Lusch & Vargo, 2014, pp. 121-123).

Axiom 2

Axiom 2 of the SDL focuses on the creation of value for individual actors. It states that "value is cocreated by multiple actors, always including the beneficiary" (Vargo & Lusch, 2016, p. 9). Usually, the view on economic exchange is dyadic: The supplier of a product sells it to his customer, who experiences the value inherent in the product. With the concept of value cocreation, SDL puts aside both this dyadic view of interaction and the view of how value is created for a consumer.

With the focus of marketing on operant resources, the development of core competencies becomes central for firms. These are composed of various combinations of individual operant resources. In some cases, if they are highly qualified (Teece & Pisano, 1994, p. 537), they can lead to dynamic capabilities and thus competitive advantage. For this, companies

must progressively improve their operant resources, which they can only do in exchange with customers and partners. Through collaboration with a network surrounding them, firms collect feedback that helps them to improve their operant resources and value propositions (Vargo & Lusch, 2004, pp. 5-6).

A consequence of service orientation is a single firm does not create that value. Instead, value results from a firm's collaboration within a network learning from it and continuously responding to its individual and dynamic needs and changes. Thus, it implies that value is defined and cocreated with a firm's network of customers and strategic partners (Vargo & Lusch, 2004, pp. 5-6).

Therefore, the view on value cocreation teaches that actors, e.g., firms offering a product or service, cannot deliver value. Instead, they can only invite others, like customers and strategic partners, in the mutual creation of value (Vargo & Lusch, 2016, p. 10). At the same time, however, the cocreation of such value experience remains "inherently beneficiary oriented" (Vargo & Lusch, 2016, p. 10).

Axioms 3 & 4

Axiom 2 already implied what SDL calls an Actor-to-Actor (A2A) network perspective, clarifying that multiple actors cocreate value. Axiom 3 and Axiom 4 further embrace this perspective. Axiom 3 states that "all social and economic actors are resource integrators" (Vargo & Lusch, 2008, p. 9), and axiom 4 says that "value is always uniquely and phenomenologically determined by the beneficiary" (Vargo & Lusch, 2008, p. 9).

Vargo & Lusch characterize all those contributing to the cocreation of value as being similarly involved in the process of resource integration and service exchange. Consequently, they argue for leaving traditional designations such as "customer," "producer," "employee," etc., behind and instead refer to a more generic language of "actors" (Vargo & Lusch, 2011, p. 182). This view recognizes that all actors within a network basically provide service – may it be individual employees or an organization. According to SDL, all actors apply specialized competencies for their own or one other's benefit, and they receive similar service from them (Vargo et al., 2020, p. 9). It allows identifying individual actors based on distinct identities with which they associate themselves instead of referring to them in predefined terms. (Vargo & Lusch, 2016, p. 7).

The network perspective of SDL further allows to zoom out from the dyadic standpoint in which exchange happens only between two actors. Vargo & Lusch note that value-creating networks exist at three different levels of aggregation - macro, meso, and micro - and that conditions at a higher level provide the context for value cocreation at the level below. Inverted, this means that the exchange of resources and the creation of value occur in the context of a more extensive system. It follows that value is a context-dependent phenomenon (Vargo & Lusch, 2016, pp. 17-18). The example of purchasing a branded product illustrates

this, as it does not provide additional value to the customer compared to an otherwise identical unbranded product purely at the dyadic micro level. Instead, the product creates added value for the buyer through the context of the overlying social networks because it enjoys a more excellent reputation in society (Vargo & Lusch, 2016, p. 18). Recognizing this allows us to understand the concept of value cocreation: Resources necessary for value cocreation do not come only from the offeror but from various other actors, including the beneficiary. In addition, as different actors are involved in resource integration and towards the cocreation of value, the dynamic nature of the network perspective becomes apparent. It requires going beyond a sole network perspective and towards a systems perspective (Vargo & Lusch, 2016, p. 7).

However, understanding this phenomenon only based on knowledge about the setting in which actors cocreate value is insufficient. What is also needed is an understanding of the underlying mechanisms that facilitate and coordinate the resource integration, service exchange, and thus value cocreation within network contexts.

Axiom 5

The last axiom of SDL, axiom 5, is dedicated to the very mechanisms that explain how value cocreation occurs in actor networks. It states that value cocreation is coordinated through actor-generated institutions and institutional arrangements (Vargo & Lusch, 2016, p. 18).

Vargo & Lusch (2016) recognize the role of the two sociological concepts of institutions and institutional arrangements in facilitating and coordinating resource integration, service exchange, and thus value creation in networks. "Institutions," as used in SDL, refer to “regulative, normative, and cultural-cognitive elements” that provide societies with “stability and meaning” (Scott, 2013, p. 56). They do this by imposing constraints on human behavior through "legal, moral, and cultural boundaries" on the one hand and by reinforcing directed human action on the other. Over time, institutions change: they are repeatedly built up (known as institutionalization) and dismantled (deinstitutionalization). These effects contribute to the dynamics of social systems (Scott, 2013, p. 58). Institutional arrangements then are referred to as sets of interrelated institutions – they combine the elemental effects of individual institutions and give social systems their structure (Vargo et al., 2020, p. 12). More concrete, institutional arrangements facilitate and coordinate the activity in value cocreating network, that is, resource integration and service exchange, and influence how individual actors determine value, e.g., the service beneficiaries (Vargo & Lusch, 2016).

Together with the view on networks and their systemic nature presented above, institutions and institutional arrangements then allow to define networks “as resource-integrating, service exchanging actors that constrain and coordinate themselves through institutions and institutional arrangements” (Vargo & Lusch, 2016, p. 6). Therefore, “networks tend to be self-governed, self-adjusting service ecosystems engaged in value cocreation at various levels of aggregation” (Vargo & Lusch, 2016, p. 6).

This view on how interaction within networks of individual players occurs allows us to perceive markets as changing and dynamic systems. It is made up of and modified via institutionalization and deinstitutionalization, following the zooming-out approach of Vargo & Lusch (2011). Consequently, an understanding of institutions allows us to understand market dynamics and economic growth (Vargo & Lusch, 2016, p. 19).

1.1.2.2 Reframing Business: Literature That Influences the SDL

Founded by Vargo and Lusch (2004), SDL is undoubtedly the most respected and complete means to approaching a service-dominated marketing framework to date (Vargo & Lusch, 2016, p. 5). This becomes obvious when looking at the frequency of citations of Vargo and Lusch's works on Google Scholar, ResearchGate, and similar portals. But SDL is not the first, nor the only, attempt to create a service-oriented picture of our world. As early as 1993, Richard Normann and Rafael Ramirez called for a strategic shift away from a product focus to a service focus. Even then, they argued that the approach of placing value-creating activities along a value chain, which is still valid today, is outdated. Because the world was becoming more and more dynamic, for example, due to global competition, changing markets, and new technological achievements, new opportunities to generate value were opening up all the time. The old approach to strategy, they say, can no longer cope with these new dynamics (Normann & Ramirez, 1993, p. 65). Instead, Normann and Ramirez called for successful companies to focus not on themselves or their industry but on the value-generating system around them - made up of diverse actors - that worked together to co-produce value. According to the authors, the essential work of business strategies was to reconceptualize the roles and relationships within these value-creating groups of actors to enable new forms of value creation (Normann & Ramirez, 1993, pp. 65-66). In (2007), Michel, Vargo & Lusch acknowledge the contribution specifically of Richard Normann in conceptualizing the service orientation of marketing. As they note, the SDL and Normann's approaches agree on many points. One crucial point Normann raises that further informs SDL is his idea of resource density in service offerings. He argues that for the customer to have an optimal service experience, the optimal combination of resources must be accessible based on location and time.

To achieve this optimal combination, actors must mobilize the various resources that make up their service offering. Resource density describes the extent to which this can be made possible (Normann, 2001, p. 27). Resource liquefaction is an important concept here, stating that information (i.e., an operant resource) can be easily separated from the physical world and transported thanks to IT (Normann, 2001, pp. 31, 33). Furthermore, Normann speaks of unbundleability, which describes as much as the separation of "activities hitherto well defined and held together in time and place and by actor" (2001, p. 33). Liquefecation and unbundleability form Normann's concept of dematerialization mechanisms. In a subsequent

step, they allow the rebundling of information and activities into new densities that will create new, previously impossible value configurations and propositions (Michel et al., 2007, p. 154). Michel et al. consequently see Normann's work as enriching SDL (2007, p. 155).

1.1.3 Insights

In the sections ahead, DT and SDL have been considered in more detail to delineate the outer boundaries of this work.

DT entails change, especially for established companies with grown structures. Digital technologies and young, agile companies bring a new dynamic to existing markets. Digital strategies help companies to develop new value propositions for their customers using modern technologies. An essential requirement for realizing such new value propositions is an evolution of both business and IT capabilities. Dynamic capabilities are of particular interest. They allow companies to identify opportunities and potentials in turbulent markets and respond to them as "prime movers" (Normann, 2001, p. 51).

Achieving this strategic agility is perhaps DT's central goal. Ideally, the path along strategy execution alternates phases of project execution and evaluation of strategy progress. Another aspect that has remained unaddressed so far is the importance of leadership in DT. Purposeful and centralized leadership is a critical success factor for DT (Soto Setzke, Riasanow, Böhm, & Krcmar, 2021, p. 10). A strategic concept that aims to lead through DT should also be measured by how well it enables leaders to initiate change by introducing adequate language (i.e., rhetoric) and create concrete action (Eccles & Nohria, 1992).

This is where SDL comes into play. As marketing scholars repeatedly have argued, a product-centric focus of marketing cannot keep up with the dynamics in modern social and economic environments (e.g., Normann & Ramirez, 1993; Vargo & Lusch, 2004). SDL offers a new language and more precise knowledge about the market's functioning without rendering the GDL obsolete (Vargo & Lusch, 2016, p. 6). Therefore, it can be the basis for designing digital strategies to produce the organizational capabilities needed in turbulent times.

The only issue with SDL is its high-level orientation. Vargo & Lusch see the SDL on a path to becoming a generalizable theory of the market (2016, p. 6). This is probably true, as it contains a lot of by now accepted, generalizable thought (Brodie, Saren, & Pels, 2011, p. 76). But this grand-theoretical orientation makes the SDL alone impractical for strategizing on DT. Therefore, scholars of various disciplines who were inspired by the school of thought of the SDL began to develop appliable mid-range theories. Prominent examples are service science (Maglio & Spohrer, 2008) and service innovation (Lusch & Nambisan, 2015) that deal with strategizing on how to tackle DT (Brust, Breidbach, Antons, & Salge, 2017, p. 2). Especially service innovation, which the literature review will deal with in the subsequent chapter, is an important approach towards DT. However, understanding the ideas behind service innovation required a discussion of the SDL first.

1.2 Successful Digital Transformation with Service Innovation

After laying out the theoretical foundation for this project, it is time to focus on another central concept: service innovation. The second question to be answered within this critical literature review is "how can service innovation lead to successful DT?". This question already implies that service innovation holds an answer to the previously highlighted challenges of DT.

This sub-chapter is structured as follows: First, it provides an overview of the historical development of innovation research to capture the idea behind the term. Second, the chapter draws upon structuration theory to understand why emergence is important for service innovation. Third, with the help of a multi-level perspective, the process of value cocreation within service ecosystems is explained to establish a thorough understanding of the mechanisms that lead to service innovation. Fourth, the concluding part summarizes the most significant insights of this sub-chapter and builds a bridge to discussing the SDA.

1.2.1 The Role of Service Innovation in Digital Transformation

Innovation as a concept has been subject to business research for over a century, with various scientific disciplines from quality management to marketing to organizational behavior contributing to its understanding. Some see innovation as the results of research and development activities that offerors sell to their customers (Hauser, Tellis, & Griffin, 2006). A more network-centric approach emphasizes groups of various interacting and resource integrating actors. These actors' heterogeneity influences the outcome of innovation activities (Corsaro, Cantù, & Tunisini, 2012).

In recent years, researchers have also approached innovation from a service perspective. By adopting an SDL perspective, service science established the scientific, management, and engineering discipline necessary for SDL-inspired service and service innovation (Böhmann, Leimeister, & Möslein, 2014; Maglio & Spohrer, 2008; Maglio, Vargo, Caswell, & Spohrer, 2009). The central theme of service science is the service system (Maglio et al., 2009). Böhmann et al. define it as "a socio-technical system that enables value cocreation guided by a value proposition" (2014, p. 74). In contrast to the service ecosystem described in 1.1.2.1, the term refers to a technically supported system of actors in which value cocreation occurs. The service ecosystem superordinates the service system. Service ecosystems have five characteristics that are summarized in Tab. 3 below.

These characteristics have some further implications for the functioning of the service ecosystem. A shared institutional logic provides the fabric and mechanisms that coordinate the various actors' value-creating activities inside a service ecosystem (Warg & Deetjen, 2021, pp. 4-5). It is made up of both general institutions and individual actors' natures[2]. If actors

[2] The term "institutional logic" stems from Lusch & Nambisan (2015). Vargo & Lusch (2016, p. 18) speak of

inside a service ecosystem show adaptive behavior during their resource integration, the institutional logic of the service system changes, too. Additional dynamic is then induced by the market dynamics surrounding the service ecosystem. Hence, service and service ecosystems are intrinsically systemic and dynamic concepts" (Warg & Deetjen, 2021, p. 4).

Table 3: Characteristics of Service Ecosystems

No.	Characteristic
1	A service ecosystem is a relatively self-contained system with blurry borders
2	Social and economic actors are self-adjusting; they show adaptive behavior
3	Actors are involved in resource integration within this system
4	Shared institutional logic connects actors
5	The process of service exchange results in value cocreation for at least some actors

Adapted from Vargo & Lusch (2019, pp. 16-17); Frow & Payne (2019, pp. 88-89).

Consequentially, "each instance of resource integration, service provision, and value cocreation changes the nature of the system to some degree and thus the context for the next iteration and determination of value cocreation" (Vargo & Lusch, 2011, p. 185). In addition, the systemic functioning of the service ecosystem is emergent – it "does not necessarily require top-down government and control mechanisms like in the case of planned networked organizations" (Warg & Deetjen, 2021, p. 4). Therefore, the systemic functioning of a service ecosystem causes each action of an individual actor within it to change the way value cocreation occurs the next time and how actors will experience value provision.

Koskela-Huotari et al. (2016, p. 2964) define service innovation as the process of renewing and improving ways for actors within a service ecosystem to cocreate value through service exchange. With the discussion above in mind, service innovation becomes immanent within service ecosystems when the systemic functioning of the system improves how value is cocreated. Value cocreation is thus the result of the collective action and interaction between actors in the service ecosystem, always including the service beneficiary, and it is governed by shared institutional logics (Lusch & Nambisan, 2015, pp. 155, 161, 165)

Based on this rationale, a strategy that leverages service innovation within a service ecosystem can answer the issues mentioned above for organizations facing digital disruptions. It can do so if it provides two things: First, if it incorporates the knowledge how structures within ecosystems arise and how value cocreation within service ecosystems functions effectively on different levels (Warg & Deetjen, 2021, p. 5). Second, if it establishes the organizational ability to identify changing customer needs, opportunity spaces, or voids and appropriate solution capabilities to realize new, reengineered value propositions through various activities. This includes removing constraints between actors by reconfiguring institu-

institutional arrangements instead, but the two terms can be seen as synonymous.

tional arrangements for improved interaction within the service ecosystem and the unbundling and rebundling of multiple resources to create new resource densities (Michel et al., 2007, p. 154).

1.2.2 Social Theories Unravel Value Cocreation in Service Ecosystems

A digital strategy that aims to leverage service innovation should base on how structures within service ecosystems evolve and how value cocreation occurs. Furthermore, an organization should gain the ability to control these effects (Warg & Deetjen, 2021, p. 5). In this regard, this subchapter first discusses the relation between structuration theory (Giddens, 1984) and innovation. Then it presents a multilevel approach of how value cocreation occurs within service ecosystems (Storbacka et al., 2016).

1.2.2.1 Innovation Relates to the Duality of Structure and Emergence

The sections above identified the seemingly random creation of new properties of a service ecosystem through the action of individual actors as an essential effect for service innovation to occur. Giddens' structuration theory (1984) is concerned with the link between the structure of social systems and the behavior of individuals within these systems. Thus, structuration theory helps understand how new properties of social systems can emerge. Emergence is a phenomenon scientists have studied since the 1970s (Szabo, Teo, & Chengleput, 2014, p. 207). It describes how actors within a complex system, through their collective behavior, cause the creation of new properties of this system that "cannot be reduced only to the behavior of their individual components" (Szabo et al., 2014, p. 207). Emergence has been observed, for example, within flocks of birds or ant colonies (Szabo et al., 2014, p. 207).

Giddens begins defining his theory using the concept of social practice (1984, p. 2), which he describes as "an ongoing series of practical activities" (Giddens, 1993, p. 81). Most people, throughout their day, participate in various social systems, like "work, home, and polity" (Whittington, 2010, p. 110). They also have the agency to temporarily prefer to participate within one social system rather than commit to another one, e.g., "to work late or just go home" (Whittington, 2010, p. 110). Whether a person decides to participate in a social system or not, both are forms of social practice, and they both have direct effects on the social systems the person belongs to (Whittington, 2010, pp. 110-111).

Agency does not only depend on an individual's choice. It is further influenced "by control over resources," and "it is exercised through the following, or rejection, of rules" (Whittington, 2010, p. 111), e.g., in the form of institutions (Giddens, 1984). Such "rules and resources are the structural properties of social systems, where structures are relatively enduring and general principles of system ordering" (Whittington, 2010, p. 111). For Giddens (1984), it is clear that the extent of agency depends on the number of resources one holds and the plurality of rules one can negotiate. Thus, structure is essential to agency "as

it furnishes both the resources that make action possible and the rules that guide it" (Whittington, 2010, pp. 111-112).

The actions of each actor influence the structure of the social system. At the same time, this network's structure influences the individual's actions. Giddens (1984, p. 25) calls this observation the duality of structure: "The structural properties of a system are both the medium and the outcome of practices they recursively organize." The duality of structure explains that within social systems, a structural continuity is present at the same time as the opportunity "for deliberate innovation and change" (Whittington, 2010, pp. 112-113). However, innovation and change do not result from each individual's actions. Only the collective effects of multiple actors engaging with the same social system cause the creation of new properties that are bigger than the sum of these individual actions. As described by Whittington, innovation in social systems is the result of emergence in the form of emergent properties.

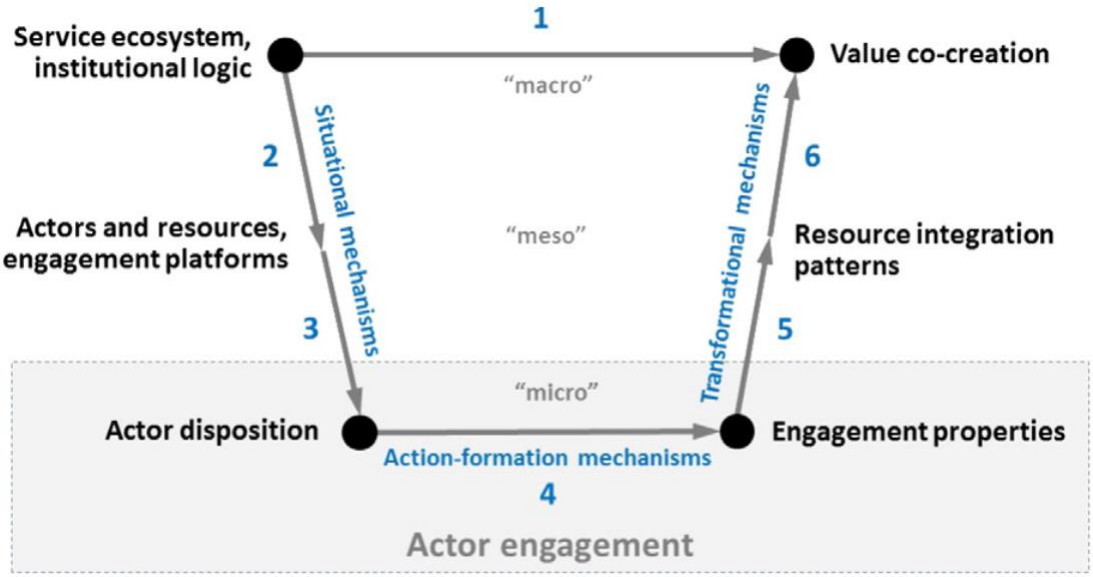
Applying structuration theory and emergence to service science provides insights into how structures are maintained and created within service ecosystems. Institutions are the rules within the system, and value cocreation corresponds to practice. Existing and new structures can be maintained and instantiated as a "virtual order" through practices guided by rules and resources (Warg & Deetjen, 2021, p. 6). As a result, innovation continuously emerges in a system that is continuous at its core.

1.2.2.2 Social Theory Explains the Process of Value Cocreation

As it has become apparent, the concept of value cocreation is central to SDL and service innovation. However, its nature remains vague because it is an abstract phenomenon that is hard to understand and observe in natural environments.

Storbacka et al. (2016) have developed a framework that better understands the mechanisms behind value cocreation. Building on the multilevel approach of the microfoundation movement (Felin, Foss, & Ployhart, 2015), the idea emerged to split the vague and difficult-to-observe link between service ecosystems and value cocreation into several levels of abstraction. They illustrated the resulting multi-layered nature of the phenomenon using Coleman's boat (Coleman, 1994, see Fig. 3). This view locates value cocreation within service ecosystems at the highest level of abstraction, the macro level. But a causal relationship between the two elements cannot be seen directly here. To understand how actors cocreate value, one needs to go down the ladder of abstraction. On the micro-level, actor engagement (AE) forms the microfoundation of value cocreation. Social mechanisms (Hedström & Swedberg, 1998) link the various levels and enable, control, and transfer AE into value cocreation. The multilevel perspective on value cocreation breaks down a complex phenomenon and makes it analytically and empirically explainable. The following section discusses the social mechanisms associated with AE (Storbacka et al., 2016, pp. 3008-3009).

Fig. 3: Coleman's Boat Applied to Value Cocreation.



Source: Storbacka et al. (2016, p. 3010).

Situational Mechanisms

As illustrated in Fig. 3, situational mechanisms are the first set of social mechanisms visible within Coleman's boat, reaching from the macro-level to the micro-level and influencing the individual actor's disposition to engage. Situational mechanisms are the forces that evolve from a specific situational setting that actors are exposed to, affecting those actors' behavior (Hedström & Swedberg, 1998, pp. 21, 23). In the context of value cocreation, situational mechanisms are the forces that originate from the service ecosystem's institutional setup, including the network's shared institutional logic and value propositions. (Storbacka et al., 2016, p. 3009). In this regard, "actor" should not be limited to individual humans. It can describe both humans and technological actors – individuals or groups – like entire organizations, AI, self-driving cars, etc. (Storbacka et al., 2016, p. 3010).

But those actors will typically only participate in the process of value cocreation if their own value expectation is seen as beneficial by them. To create value for themselves within the given setting, actors need the optimal combination of resources based on location and time at hand. Hence, resource density is central within situational mechanisms (Weiß et al., 2021, p. 4). This is where service platforms come into play. They are a new and much-discussed topic for which no uniform definition has yet been developed (Asadullah, Faik, & Kankanhalli, 2018; Hein et al., 2020; Tiwana, Konsynski, & Bush, 2010). Within service innovation, the concepts of service platforms (Lusch & Nambisan, 2015) and engagement platforms (Ramaswamy, 2009) are the predominant types of platforms to be found. However, service researchers' expressions of such platforms differ. The term is used for various product offerings, like the Apple iPad as a platform (Yoo, Henfridsson, & Lyytinen, 2010).

It also occurs in a more architectural orientation (Sebastian et al., 2017). Based on the literature reviewed, the search for a precise picture of what platforms are is secondary. Instead, the focus is on the mechanisms and capabilities that the concept offers to support value cocreation. The exact design of the platform then follows these requirements in real applications. It is, therefore, context-dependent.

Various works define service platforms differently (e.g., Lusch & Nambisan, 2015, p. 164; Sebastian et al., 2017, p. 203; Warg, Zolnowski, Frosch, & Weiß, 2019, p. 8). What these papers agree on is that service platforms are modular, mostly technology-based solutions. They consist of intangible and tangible components and instantiate capabilities necessary for delivering service offerings. The main task of service platforms is to integrate different actors and their resources, orchestrate them, and thus enable interaction in the form of service-by-service exchanges. Service platforms thus enable value cocreation.

One component of service platforms are engagement platforms (Weiß et al., 2021). They, too, are not defined uniformly (e.g., Breidbach, Brodie, & Hollebeek, 2014, p. 594; Ramaswamy, 2009; Ramaswamy & Gouillart, 2010; Storbacka et al., 2016, p. 3011). Engagement platforms aim to connect the actors and their resources to the service platform. They are to be understood as a kind of communication layer that provides virtual or physical touchpoints. Thus, it can be purely digital, purely physical, or hybrid. Furthermore, the engagement platform's purpose is to "provide structural support" for service-for-service exchange.

As a consequence, resource density is fostered by service platforms through (1) the separation and transportation of information independent from physical objects (Normann, 2001), (2) the provision of necessary connections for the service ecosystem's actors (Storbacka et al., 2016, pp. 3011-3012), and (3) the governance of service exchange through a set of rules and protocols (Brodie, Fehrer, Jaakkola, & Conduit, 2019, p. 184; Lusch & Nambisan, 2015, p. 166).

Action-Formation Mechanisms

Action-formation mechanisms are the second type of social mechanism in Coleman's boat. On the micro-micro level of Fig. 3, they explain how actor disposition results in concrete action of an individual actor that can then be observed in the form of engagement properties (Hedström & Swedberg, 1998, p. 1998; Storbacka et al., 2016, p. 3012).

Researchers have defined actor disposition on several occasions. In their essence, they all come down to the same idea. The term describes an individual actor's willingness to act upon connections, provided through an engagement platform, with their own resources in response to past events and towards a specific future goal (Brodie et al., 2019, p. 184; Storbacka et al., 2016, p. 3012). This definition clarifies that actors' active participation within the service ecosystem highly depends on the situational setups discussed before. For example, suppose an actor perceives the situational design or parts of it, such as the given

institutions or the configuration of the engagement platform, not as appropriate. In that case, AE is unlikely to happen (Storbacka et al., 2016). But if an actor decides to engage, this will most likely be through investing his own resources.

At this point, it is difficult but necessary to dissociate the action of an individual actor from the dynamic and interactive character of interaction within the service ecosystem because AE is on the personal level. But the actual engagement of the actor is an individual activity, although influenced by the situational conditions and contextualized through the action of all actors with a collaborative intention (Brodie et al., 2019, p. 184). The act of individual actors is then in the form of mentioned engagement properties empirically observable – other than the macro-level phenomenon of value cocreation (Storbacka et al., 2016, pp. 3008-3009). Characterizing engagement properties based on different aspects (see Tab. 4) makes it analyzable for further evaluation. Therefore, it is a key success factor for service ecosystem designers to ensure the willingness of individual actors to engage through resource integration, thereby ensuring AE (Grotherr et al., 2018, pp. 4-5). Only then macro-outcomes (here value cocreation) can occur (Coleman, 1994, p. 28).

Table 4: Aspects for Examining Engagement Properties

Aspects	**Description**
Relational	Refers to the relationships that an individual actor enters into with other actors in the ecosystem in terms of their scope, social centrality, and the relative power of the actor in those relationships, and whether the connections are direct or indirect.
Informational	Refers to whether an actor seeks to influence the ecosystem or is willing to be influenced by other actors / whether an actor needs support or offers support.
Temporal	Refers to the "duration, regularity, and frequency" of engagement and provides information about the impact of engagement on the actor. For example, irregular engagement indicates less impact than more frequent engagement.
Co-production vs. Value in use	Co-production engagement refers to the actor's involvement in joint design, development, production, or promotion activities within the network. Value-in-use relates to cases where the actor is involved in co-creating value using the resources of other actors.

Adapted from Storbacka et al. (2016, p. 3013).

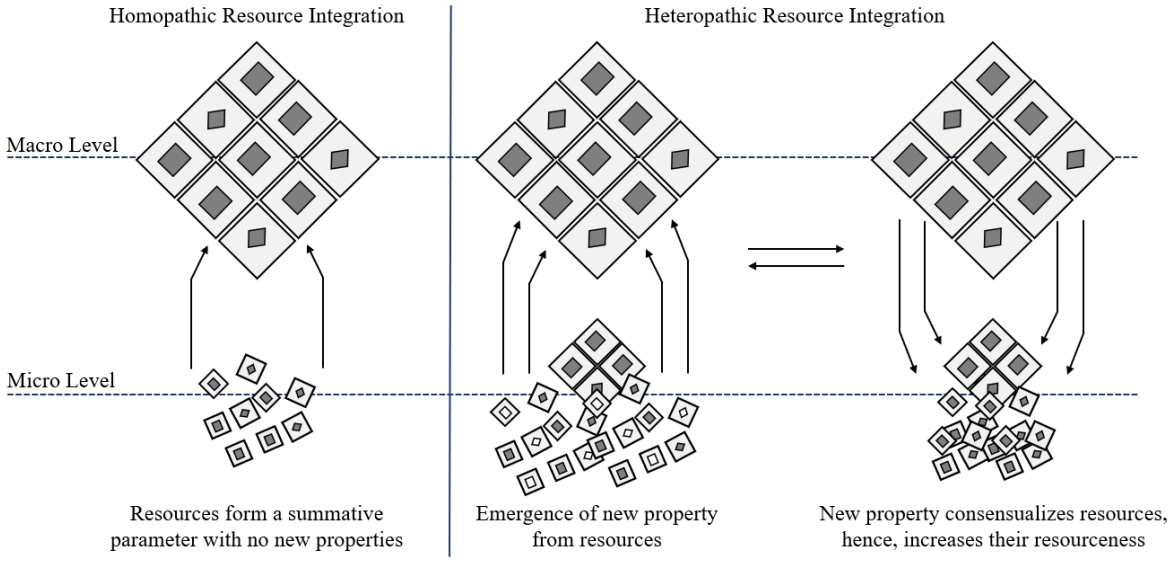
Transformational Mechanisms

As previously denoted, the discussed AE acts as the microfoundation for value cocreation. However, this is only true if the service system can transform AE into a mutual macro-level

outcome. Transformational mechanisms, the third group of the social mechanisms, hold the explanation for this issue (Hedström & Swedberg, 1998, pp. 22-23).

Resource integration patterns represent "transformational mechanisms at work" (Weiß et al., 2021, p. 3). Now, one must abandon the isolated view of individual actors called for in the previous paragraph. Instead, a consideration of all participating actors and their resources must now take place at the meso-level. First, individual actors within the service ecosystem integrate their own (usually operant) resources. Then, as shown in chapter 1, they apply these to the resources of others. Thus, what is central in the view of all acting actors is the integration of resources with the overarching goal of value cocreation (Storbacka et al., 2016, p. 3013).

Fig. 4: Homopathic vs. Heteropathic Resource Integration



Adapted from Chandler, & Strathoff (2016, p. 2984).

The discussion of the duality of structure has shown that emergence is key to innovation. It can occur through the collective action of multiple actors within a social system (Giddens, 1984). The SDL discussion identified resource integration as a single actor's most important type of action for successful value cocreation (Vargo & Lusch, 2004). Peters (2016, p. 3000) distinguishes two kinds of resource integration: She describes (1) homopathic resource integration as a summative, intra-ordinal process, and (2) heteropathic resource integration as an aggregative and trans-ordinal process. The distinction between the two helps to determine further the occurrence and effects of emergence within service ecosystems (see Fig. 4). The main differences between the two lie in their outcome. Whereas (1) only produces a combination of its basal resources, (2) causes the emergence of new properties (Peters, 2016, p. 3000) of the system that cannot be described by their basal resources, but which still rely on them (Coleman, 1994, p. 28). Examples of such new properties are "structures, qualities, capacities, textures, mechanisms" (Peters, 2016, p. 3000).

For value cocreation and service innovation to occur, these emergent properties are of particular importance. They change the service (eco)systems structure and thus influence how actors perceive the nature of integrated resources in terms of appraisal and action. In an optimal case, emergent properties increase resourceness and offer new and improved ways for actors to act upon resources for value cocreation. Homopathic resource integration, on the other hand, does not affect resource utilization. Its impact on value cocreation is therefore limited (Peters, 2016). Fig. 4 visualizes the difference between homopathic and heteropathic resource integration. As shown on the right, a new property emerges based on various resources – it is dependent on them but ultimately unique. Through consensualization (Meynhardt et al., 2016), this new property has its effect back on its basal resources, i.e., dictating the ways actors can use them to cocreate value.

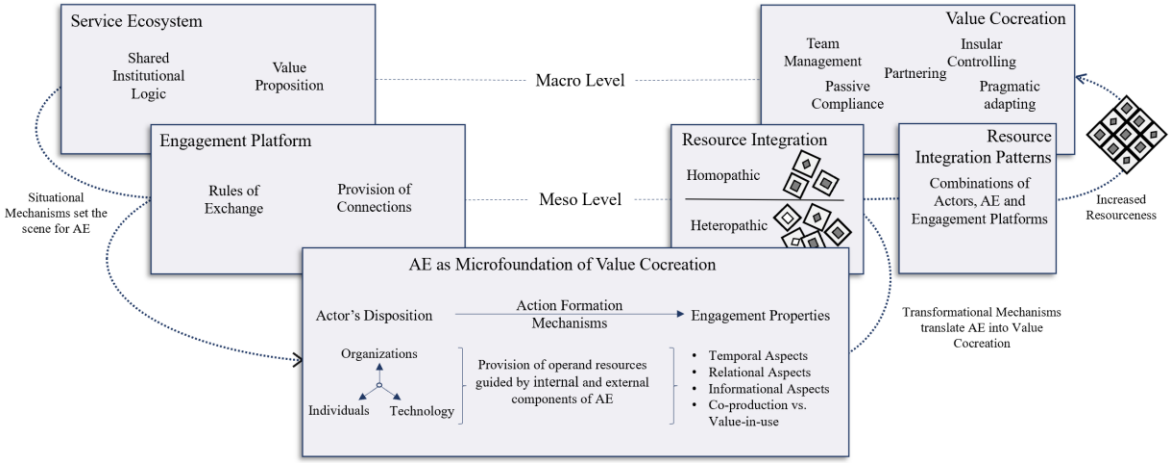
Peters continues to outline five different ways actors can cocreate value based on resource integration (2016, p. 3003): (1) a highly productive composition of actors in the network and a high number of interactions within that team characterizes the team management. (2) The passive compliance style is characterized by a few interactions among actors that the ecosystem's operator prescribes. (3) The insular controlling style is distinguished by low levels of interaction combined with high activity levels among the individual actors. (4) Partnering describes a moderate level of activity and interaction among individual actors. At the same time, they see the primary purpose of the ecosystem as interacting with the focal actor. (5) Many interactions with low levels of action by individuals characterized the pragmatic adaptation style. The actors' understanding of their role is characterized by their belief that they must adapt to changing circumstances.

Largely unnoticed until now has been the role of resource integration patterns. Such patterns generally result from a combination of actors, their corresponding AE, and an engagement platform that acts as an intermediary to manage and enable resource integration (see above). Resource integration patterns play an essential role in the transition from micro-level activities to macro-level outcomes. Take the example of visiting a restaurant: You go out to eat, and the experience of having dinner is different for every customer because of individual constellations. But the result is nevertheless expectable and usually satisfactory (homeopathic resource integration). The system in which the restaurant and you, the client, are placed may also be designed to allow new experiences to emerge during contact. These experiences could go beyond the usual experiences (heteropathic resource integration) and lead to an innovative process of value cocreation for all parties involved. This innovative process may have been unique. The question is, how can it be stabilized to be reproducible in the future?

Resource integration patterns address this very issue by stabilizing the configurations of a service system so that newly emerged patterns become reproducible (Storbacka et al., 2016, pp. 3013-3015). In other words, resource integration patterns "capture solutions for recurring design problems" on the meso-level (Grotherr et al., 2018, p. 5).

Fig. 5 illustrates the main steps of this process on the different levels of the multi-level approach to help understand the previous discussion about the process of value cocreation.

Fig. 5: Detailed Illustration of the Value Cocreation Process



Own work.

1.2.3 Recap

To compete in dynamic environments, companies must continuously innovate. They need the ability to recognize change and generate new perceptions of value for customers despite or with this change. Normann's term of the "prime mover" (2001) describe organizations that can respond to change faster and better than their competitors and thus have the potential to reframe the business within their market. On the other hand, Staber & Sydow (2002, pp. 408-409) speak of adaptive organizational capacity, which allows to "continuously develop and apply new knowledge" through dynamic capabilities to ensure sustainable competitiveness.

Service innovation can lead to such dynamic capabilities because it creates new pathways within a service ecosystem that actors can cocreate value (Koskela-Huotari et al., 2016, p. 2964). The presented multi-level approach for value cocreation (Storbacka et al., 2016) is essential for designing digital strategies and implementing respective actions for tackling DT. It communicates the ideas of SDL (Vargo & Lusch, 2004) in a more comprehensive form, which is the multi-level view. That being said, the second question guiding the critical literature review has been answered.

However, service innovation theory still lacks an important aspect that hinders it from applying to practical use. All the discussed theory so far (and this includes the content from 1.1) fails to go what Nunamaker, Briggs, Derrick, & Schwabe call the "last research mile" of IS research (2015). That is, the theory discussed so far has not undergone much "success-

ful transition to the workplace" (Nunamaker et al., 2015, p. 12), whereby it may have provided proof-of-concept, proof-of-value, and proof-of-use (Nunamaker et al., 2015, pp. 15-16). Going the last mile of research is necessary, though, because it "means using scientific knowledge and methods to address important unsolved classes of problems for real people with real stakes in the outcomes" (Nunamaker et al., 2015, p. 11). As Nunamaker et al. argue, the last mile of research is a necessity for scientific knowledge to become a routinely used tool in the field (2015, p. 11).

The transition of the discussed theory has only recently been made, with the major works discussed in the subsequent chapter.

1.3 Providing the Foundation of Execution for Digital Strategies

Technology plays a vital role in the meso-level of value cocreation. However, what remains undiscussed in the context of this critical literature review is how organizations can design technology to address the requirements for service innovation to take place in a real context. The third guiding question of this review, "How can the SDA provide a conceptual foundation for implementing digital service strategies?" addresses this remaining issue.

In brief, the following sections discuss why architecture is relevant in the context of DT first. Then, the review presents the SDA as an architectural blueprint for designing and implementing a service-driven digital strategy. Third, the latest developments in SDA research that transform it into both a framework and a design environment for operating service systems are introduced. Finally, in the end, it presents a recent approach to a multi-level design process for building service systems.

1.3.1 The Foundation of Execution for Digital Strategies

Many organizations fail to succeed in their DT because they do not approach it holistically. Too often, firms react to specific challenges with isolated digitization initiatives (Gebauer, Fleisch, Lamprecht, & Wortmann, 2020), leading to secluded, incompatible solutions and unfulfilled expectations (Kane, Palmer, Nguyen Phillips, Kiron, & Buckley, 2015). This problem can worsen with the increasing effort put into such initiatives (Gebauer et al., 2020, p. 315). Although it is necessary, organizations should not overemphasize technology's role in succeeding in a DT compared to other areas of development (Kane et al., 2015, p. 6).

As already discussed in 1.1, a successful DT needs to establish strategic agility. Practitioners must design a digital strategy that builds new business and IT capabilities (Weill et al., 2002). Strategic agility significantly correlates with the quality and extend of targeted IT infrastructure capabilities. Thus, they are the enabler of complementary business capabilities. (Weill et al., 2002, p. 61).

Enterprise Architecture Management (EAM) helps to establish strategic agility as it balances business aspects (such as business models, processes, and organizational structures) and technology aspects (such as the IT infrastructure) within an organization. It develops them in unison towards a common strategic goal (Ahlemann, Legner, & Schäfczuk, 2012, p. 8). EAM uses various frameworks and best-practice guides, for example, in engagement, motivation, design, communication, and processes (Proper, 2014, pp. 12-14). Respectively, an EA "focuses on those features of an enterprise that are necessary and sufficient to meet its essential requirements" (Proper, 2014, p. 10). It acts as the foundation of execution for digital strategies because it enables building new dynamic capabilities (Weiß et al., 2018, p. 1635).

Aiming for service innovation requires an architectural perspective, focusing on developing appropriate IT-infrastructures that support innovation. For example, they need to support resource integration or the sustainable emergence of new properties (see above). The following section presents the SDA as a blueprint for an EA helping to design and implement service-driven digital strategies (Warg et al., 2019, p. 7).

1.3.2 Enabling Service Innovation Using the SDA

In recent years, scientists have developed the SDA to help organizations build the foundation of execution for service-driven digital strategies (Weiß et al., 2018). The purpose of the SDA is to map the theoretical requirements, structures, and mechanisms necessary for effective service innovation onto EA (Warg, Weiß, Engel, & Zolnowski, 2016, p. 819). Its initial goal was to depict a target image of an IT architecture that provides the necessary IT infrastructure capabilities required for value cocreation within service ecosystems. To succeed, the SDA "complements the SDL from a practical and strategy execution perspective" by enabling respective service innovation activities (Weiß, Warg, & Zolnowski, 2019, pp. 8-9). Unlike all previously discussed literature, the SDA takes on a practical research approach: As being subject to a longitudinal single case study, it is under constant development, assessment, and re-evaluation, bringing the complex and abstract theory of SDL to a functional level (e.g., Warg & Deetjen, 2021, pp. 10-13).

Following the discussion in 1.3.1, the SDA links the business architecture with the IT architecture. It reflects the business needs for effective service innovation in the IT installations of an organization. Therefore, it provides a "technical environment that combines external resources from customers and partners, for example, user data or market data, with internal resources, for example, customer relationship management data, or services" (Weiß et al., 2019, p. 10).

The following subsections first assess the requirements that the IT infrastructure must meet to enable service innovation activities by drawing from two different lines of research. Then, it is how the SDA meets these requirements by introducing a new system layer and corresponding IT architecture capabilities to guide SDL-driven digital strategies.

1.3.2.1 Mapping Service Innovation Requirements onto IT Architecture

This section complements both chapters 1.1. and 1.2. It serves to apply the discussed theories to IT architecture requirements. The works around the SDA have drawn from said theory to conceptualize the requirements for enterprise IT (especially Weiß et al., 2018). However, the results of the teams around Ross (Ross, Beath, & Mocker, 2019; Ross et al., 2016; Sebastian et al., 2017) also deliver valuable insights into that matter. And although Ross et al. do not take a solution approach explicitly based on the SDL, their results are still relevant in terms of IT system requirements. They, too, recognize that digital business models are increasingly aiming at integration, interaction, and innovation (Ross et al., 2016, p. 3).

Both lines of research have in common that they argue for a "two-speed" architecture (Weiß et al., 2019, p. 11) to enable digital business models. Hence, practitioners need to consider two IT artifacts: The first of them is what Ross et al. refer to as an organization's "operational backbone" (2019, ch. 3). The SDA uses the term "Systems of Record" (Moore, 2011) to describe the same thing, which is "a coherent set of enterprise systems, data, and processes supporting a company's core operations" (Ross et al., 2019. ch. 3). The second artifact is named a "digital platform" (Ross et al., 2019, ch. 4) or, in the context of the SDA, "Systems of Engagement" (Moore, 2011). For the definition of this artifact, the discussion about service platforms in 1.2.2.2 accounts.

The operational backbone/SoR evolves from an organization's existing enterprise IT. Relevant technologies are ERP, CRM, and other basic systems that are de facto standards within any firm (Ross et al., 2019, ch. 3) nowadays. Therefore, this IT artifact is responsible for ensuring that transactions and processes are visible and seamless from end to end, that master data is reliable and accessible (i.e., single source of truth), and that repetitive business processes are automated. A system that supports these IT-enabled capabilities streamlines organizational core operations and thus increases organizational outcomes such as profitability or customer satisfaction (Ross et al., 2019, ch. 3). Organizations either have such an operational backbone already available or not. If the latter is the case, the transition from siloed and isolated systems to a coherent set of enterprise systems is part of their DT. This is because organizational excellence is also a prerequisite for effective digital platforms/SoE (Ross et al., 2019, ch. 3).

As the venue of innovation (Lusch & Nambisan, 2015, p. 157), digital platforms/SoE/service platforms[3] are responsible for enabling value cocreation processes. This artifact should connect various actors (human and non-human) to exchange service for service (Weiß et al., 2018, p. 1633). Therefore, consistent engagement routes, provided by, e.g., APIs, user roles, access rights (Ross et al., 2019, ch. 4) across multiple channels (Warg, Weiß, & Engel, 2015,

[3] As discussed in 1.2.2.2 the labelling of this IT artifact is second-order. Focus lies on the capabilities it provides.

p. 13) become a necessity. In addition, providing for the needed resource density is a significant task of this systems layer to support the cocreation of value.

Lusch & Nambisan (2015, p. 166) argue that a modular architecture helps achieve resource density. It allows the platform to be resolved into independent functional components that can be recombined to create new functionalities and value propositions. Similarly, Ross et al. (2019, ch. 4) argue that organizations need to build up a repository of "reusable digital components" that can be "useful in future digital offerings." Microservice architectures, consisting of multiple small-sized and specialized services, are capable of providing these reusable components. An artifact constructed from microservices can then quickly be reconfigured or extended through iterative, reoccurring design processes (Weiß et al., 2018, p. 1637). However, microservice architectures are not free of flaws and require a thorough examination of the specific context before they are applied (Richardson, 2021a). Moreover, modular architectures need mechanisms providing order and structure. Resource orchestration is required to provide systematic and optimal processes and workflows (Red Hat, 2021c).

Table 5: Purpose and Required IT Architecture Capabilities of SoR and SoE

	Operational Backbone / SoR	**Platform / SoE**
Purpose	Efficient and transparent operations; single source of truth	Facilitating value cocreating activities and thus innovation
Requirements on technology	Stable, shared, and standardized systems; a high degree of automation	Modular components and services that can be reused/rearranged/ extended; standardized cross-channel engagement routes; scalability
Requirements on data	Accurate and complete transfer of master data	Access to master data; quick availability; compatibility with various data formats (e.g., IoT, humans, etc.)
Requirements on organizational processes	Clear and transparent design and execution of core operations	Allow for iterative design and implementation of components; support of resource integration and orchestration; provision of exchange protocols;
Requirements on people	Process owners; data architects	Platform architects; component owners who constantly evolve the platform

Adapted from Ross et al. (2019).

Technology that is generally relevant to the platform artifact belongs to SMACIT. In the context of SDA, SoE are represented by IT-supported services with a focus on communication and collaboration, such as "wikis, collaborations tools, chat, crowd-sourcing, web conferencing, video streams, video conferencing, and similar services" (Moore, 2011, p. 4).

Both the operational backbone/SoR of an organization and the platform layer serve their distinct purposes. However, they are equally necessary for a DT as both contribute to an organization's strategic agility (Weill et al., 2002). Table 5 summarizes the two IT artifacts' most critical IT infrastructure capabilities to introduce to an organization. For a successful DT, practitioners must develop both artifacts coherently. Interoperability of the systems becomes a necessity. For example, the SoE requires access to historical data (e.g., user data) stored within and provided by the operational backbone. The other way around, the core systems must process transactions on the platform (Warg et al., 2016, p. 821).

1.3.2.2 SDA Operationalizes Required IT Architecture Capabilities

The SDA introduces the described new IT architecture capabilities with a focus on resource integration and value cocreation. It operationalizes the central elements of SDL and service innovation (Weiß et al., 2019, p. 9). As an architectural blueprint, it conceptualizes the SoE as a new systems layer "on top" of the existing SoR by introducing four unique components of a modular system, the SoE. The System of Interaction (SOI) is the first of these systems, responsible for enabling AE and actor interaction as resource integration and service-for-service exchange (Weiß et al., 2019, p. 9). The emphasis lies on providing consistent user experiences across multiple channels; hence customer focus is crucial. The SOI ensures customer-centricity by containing all necessary artifacts to provide seamless cross-channel interaction (Weiß et al., 2019, p. 9).

The second system is the System of Participation (SOP). It holds all functionalities necessary for the integration of strategically important partners into the service platform. The SOP allows connecting the SDA with these partners' external platforms or ecosystems and, therefore, integrating additional resources into the focal service ecosystem. In addition, the SOP provides the functionalities that are necessary for resource integration (Weiß et al., 2019, p. 9).

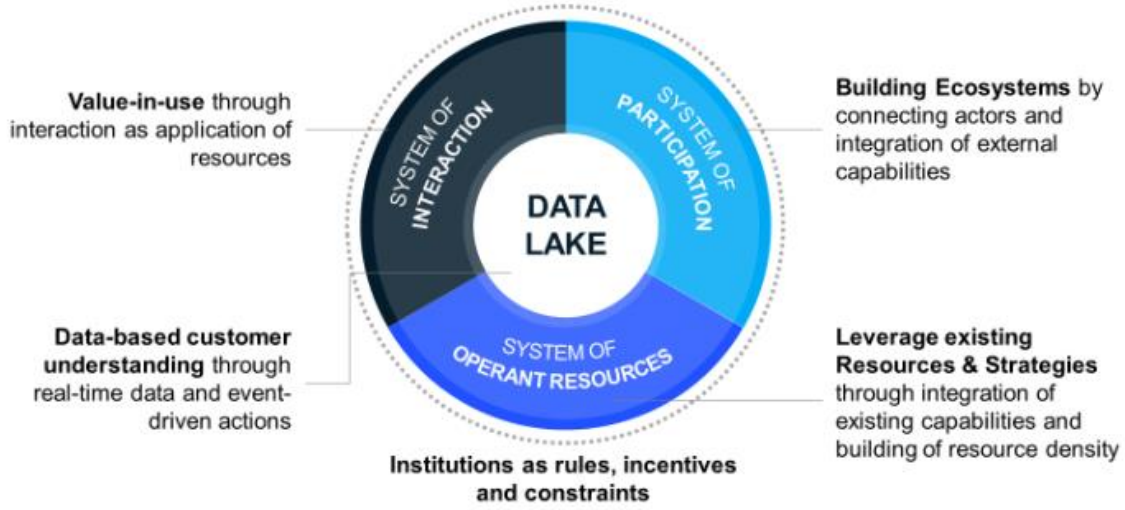
The third system is the System of Operant Resources (SOOR). This system connects the SoE with the organization's SoR. It accesses and transforms historical data and records, such as customer data, and transforms it into "dynamically applicable information resources" (Weiß et al., 2019, p. 10). Hence, the SOOR is the system that holds what has previously been described as reusable digital components. Therefore, it is also the system responsible for enabling resource density, and thus it "positively impacts the emergence and creation of innovative and relevant value propositions" (Warg & Deetjen, 2021, p. 11).

The fourth system is the System of Operational Data Stores (Warg & Deetjen, 2021, p. 11). Conceptualized as a data lake, it enables the remaining platform components to use data in real-time (Red Hat, 2021a). This can either involve data loaded from the SoR or data that has been recorded and stored during AE. The data lake enables "data-based understanding of the customer, their contexts, preferences and needs" and thus allows to enhance customer centricity and tailor-made value propositions (Warg & Deetjen, 2021, p. 11)

As discussed in 1.2.2.2, a shared institutional logic is an essential building block of service ecosystems. The SDA places the institutional logic around the SoE. It is conceptualized as the rules that coordinate the access and use of resources on the platform, like design patterns to "enable planned creation of solution designs for concrete challenges" and as incentives that promote AE (Warg & Deetjen, 2021, p. 11).

Fig. 6 visualizes and summarizes the SDA and the individual systems' purposes from a theoretical perspective. However, as mentioned earlier, the strength of SDA, compared to other research approaches in this field, lies within its applicability to real-life scenarios. The case study subject to this thesis (see 3.) will discuss such real-life applications of the SDA.

Fig. 6: SDA Introduces a New Modular Systems Layer to Enable Service Innovation



Source: Warg & Deetjen (2021, p. 11).

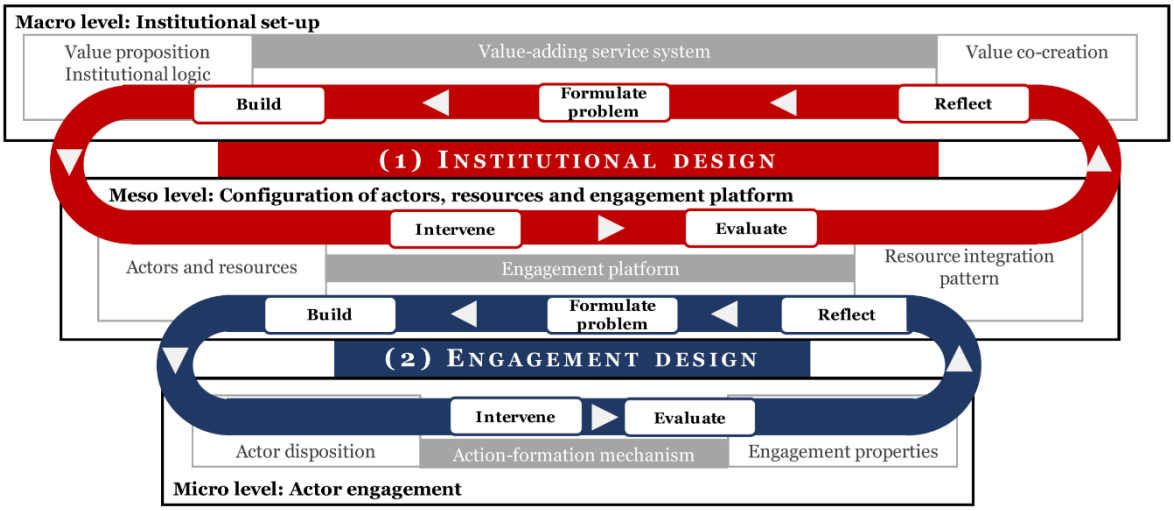
1.3.3 Evolution of the SDA: Towards a Service Systems Design Framework

The introduction to section 1.3.2 indicated that the original goal of SDA was to provide a vision for an IT architecture that enables service innovation. However, recent research draws on the multilevel perspective of Storbacka et al. (2016) and a service system design framework based on it (Grotherr et al., 2020; Grotherr et al., 2018) to broaden this scope. Now SDA carries the potential to serve as a "framework and design environment" for building and operating service systems (Weiß, 2020, p. 119; Weiß et al., 2021).

The multilevel design framework presented by Grotherr et al. (2018) aims to fill the gap between service science theory and practically realized service systems. It wants to establish "evidence-based design knowledge rooted in the design, implementation, and evaluation of real-world service systems" (Böhmann et al., 2014, p. 74). Grotherr et al. (2018, p. 3) propose an exploratory approach that implements and evaluates design decisions iteratively to evolve the service system setup. The multilevel design process combines the multilevel perspective on value cocreation (Storbacka et al., 2016) with an "iterative and validating design process" (Grotherr et al., 2018, p. 5) that draws from action design research (ADR) as described by Sein, Henfridsson, Purao, Rossi, & Lindgren (2011) and innovation piloting (Schwabe & Krcmar, 2000).

The design process consists of two intertwined design cycles oriented towards the multilevel perspective on value cocreation (see Fig. 7, Grotherr et al., 2018, p. 6). The first cycle, the "institutional design" cycle, aims to build an institutional setup with optimal configurations of actors and resources. Thus, its "goal is to enable the required level of resource mobilization and density" (Weiß et al., 2021) for the formation of effective resource integration patterns that lead to value cocreation (Grotherr et al., 2020, p. 286). The second cycle, the "engagement design" cycle, aims at building the sociotechnical components necessary for AE by actors "with various dispositions of resource mobilization" (Grotherr et al., 2018, p. 8). This includes engagement platforms and patterns that lead to action-formation mechanisms (Grotherr et al., 2018).

Fig. 7: Multilevel Design Framework for Service System Design



Source: Grotherr et al. (2018, p. 5).

Entering the Institutional Design

Grotherr et al. (2018, p. 6) argue that effectively configured resource integration patterns and thus value cocreation depend, on a basic level, on the service system's institutional setup. That is, "the frame in which service systems are designed and operated" (Grotherr et al.,

2018, p. 6). Thus, the institutional design must "initially create and, through recurring design iteration, […] refine the institutional set-up and related components" (Grotherr et al., 2018, p. 6). Therefore, the first design step consists of analyzing the current challenges and potentials of the service system, the strengths and weaknesses in the value creation process, and the institutional logic shared within the system (Grotherr et al., 2020, p. 286). Based on the analysis results, the formulation of new value propositions guides the following design activities (Grotherr et al., 2018, p. 7). Step two then involves the formation of design hypotheses that make the value proposition explicit. Practitioners should describe how actors and resources should be arranged and how their interaction and thus also the integration of resources should look concretely. In this regard, Grotherr et al. (2018, p. 7) also emphasize the importance of the engagement platform design. Appropriate platform configurations must accompany the intended interaction patterns. This includes, for example, the roles of the various actors and their respective touchpoints. In step three, intervene, the design hypotheses are instantiated in the actors' and resources' environment for testing and evaluation reasons. As already known, AE happens at the micro-level (see above), so this phase connects the institutional and the engagement cycles downstream. The goal is to observe the design hypotheses' effects by initiating resource integration activities which might eventually lead to value cocreation and new emergent properties (Grotherr et al., 2020, p. 287).

Transition to the Engagement Design

Consequently, the engagement design cycle intends to practically implement the design hypotheses in IT infrastructures on the meso-level to foster AE on the micro-level. Grotherr et al. (2020, p. 287) refer to such infrastructures as socio-technical components: The term addresses the idea that IT infrastructure alone will not get an actor to engage with the system. Therefore, it requires additional thought to design IT systems in a way that appeals to individual actors. Grotherr et al. (2020, p. 287) also speak of engagement patterns without further specifying the term. But since AE is dependent on situational mechanisms, this term most likely refers to the adequate design of appropriate customer touchpoints and journeys.

Step one, formulate problem, is concerned with translating the design hypotheses into concrete design variables for the socio-technical components. Then, barriers to participation must be identified that might impede the effectiveness of these variables. These could be, for example, reservations of individual groups of actors resulting from the institutional setup. Adjustments must then be made based on assessing these barriers (Grotherr et al., 2020, p. 288). The revised design variables are then transformed into concrete socio-technical components that enable AE in step two, the build phase, where actual IT architectures are built (Grotherr et al., 2018, p. 8).

With step three, the intervene phase, the design process enters the micro-level of value cocreation: System designers stimulate AE on the engagement platform to test the effects of previously made design decisions (Grotherr et al., 2018, p. 8). To transform the actor's disposition to engage with the system into observable engagement properties, practitioners can,

for example, use incentives or initial content on the platform to motivate and guide a user. In such a way, still existing engagement barriers can be lowered (Grotherr et al., 2018, p. 9). Both the design and composition of the platform and such motivating measures act as the action-formation mechanisms discussed earlier.

The individual actor's engagement can then be observed and analyzed in step four, evaluation. Grotherr et al. (2020, p. 288) suggest several variables available to gain insights into the performance of a service system. These are, for example, activity data such as comments, likes, or tags. In addition, since digital technology is involved, metrics from the operating systems can be retrieved, e.g., log files, activity patterns on the platform, etc. From these data, designers can make assumptions about the current quality of the design and setup of the engagement platform, which they reflect in the last step of the engagement cycle, the reflect phase (Grotherr et al., 2018, p. 9).

From here, the design process splits into two paths: The engagement design cycle can begin a new iteration, using insights from the previous iteration. It is the task of software designers and engineers to implement changes that eliminate identified weaknesses. Alongside this, insights gained from observing AE can be fed back into the institutional design cycle (Grotherr et al., 2018, p. 9).

Transition to the Institutional Design

To ensure long-term AE and thus the success of the service system, adjustments to the socio-technical components are not enough. Observing AE in the engagement design cycle helps identify weaknesses in the current design of the service platform. However, addressing these weaknesses only on a socio-technical level does not regard possible weak spots in the institutional setup of the service ecosystem. AE cannot be improved if the institutional approach is also misaligned, for example, if the value proposition for a group of customers does not meet their needs. Only fixing technical issues will not increase customer satisfaction if the system fails to address the right needs and desires.

Therefore, insights gained in the engagement design cycle must also be fed back to the institutional design cycle. Practitioners must analyze and reflect on these data (steps four and five of the institutional design cycle). Based on the results of these activities, the institutional design also enters the next iteration to increase the performance of the service (eco)system and enable better experiences for customers and partners (Grotherr et al., 2020, p. 8; Grotherr et al., 2018).

1.3.4 Prospect

DT, the SDL and service innovation are abstract scientific fields. They discuss topics relevant to practice but are not themselves in a position to make their knowledge practically available. For that, they would have to go the last research mile (Nunamaker et al., 2015).

However, the topics addressed bring valuable knowledge that more practice-oriented research can take advantage of. The SDA (Weiß et al., 2018) and the design process of Grotherr et al. (2020; 2018) take their core ideas and work to make them practical.

The SDA has initially been intended as a pure blueprint for SDL-inspired EAs. It still is today. It translates the ideas of SDL and service innovation into an IT architecture that enables service-driven business models. It does so by pointing to a two-speed architecture that provides both stability (SoR) and flexibility (SoE) and operationalizes through its design the central elements of service innovation, i.e., service ecosystems, service platforms, and value cocreation (Lusch & Nambisan, 2015). This answers the third guiding question that guided this third section of the critical literature review. However, with increasing research on service innovation and the emergence of the multilevel approach to value cocreation, SDA sees potential to evolve (Weiß et al., 2021).

Grotherr et al. (2020; 2018) take the approach of translating the multilevel perspective of Storbacka et al. (2016) into an applicable design process. They aim to enable business experts and software engineers to design socio-technical components and the natural flow of value cocreation. The design process harnesses the functioning of exchange as taught by SDL and service innovation. This is an important step toward building functional, service-oriented service systems.

However, the design process has weaknesses. One of them is that it lacks a link to practice. Grotherr et al. do demonstrate a use case (2018, p. 9). But this is not very meaningful in its presentation. Another weakness is the construction of the process using ADR. ADR is an approach to conducting practice-oriented research (Sein et al., 2011). The issue with this approach is that Grotherr et al. did not prepare the described steps so that practitioners outside the research community could easily apply them to their contexts. For that, the process remains too close at service research with still relying on abstract and incomprehensive vocabulary and concepts. This is also due to the paper's third and most profound weakness, which is that the process does not present a target picture toward which practitioners could work, and that is easy to understand.

Here, SDA research (Weiß et al., 2021) comes in. It seeks to link the multilevel perspective of Storbacka et al. (2016) and Grotherr et al. (2020; 2018) with SDA. This should result in a holistic and comprehensive guide to design, construction, and operation of service systems. Unlike other work discussed, the SDA also ensures its practicality during its ongoing development. It does this through practical implementations, which this thesis will discuss later on in chapter 3.

1.4 Findings

As markets and societies become more and more dynamic through the rapid development and use of new technologies, companies must develop dynamic capabilities to stay responsive. Often misused is DT organizations' journey of creating these dynamic capabilities. Practitioners must design and execute a digital strategy that holistically adapts business and IT to the new challenges of our age (Vial, 2019). If a firm succeeds in doing so, it has the potential to become a prime mover of its market and the chance to redefine the rules of business within it (Normann, 2001).

SDL and service innovation have generated the knowledge necessary to develop appropriate digital strategies for DT. SDL is on the verge of becoming a general theory of the market. Its ambition is to deliver a more appropriate picture of social and economic exchange (Vargo & Lusch, 2004, 2008, 2016). Such knowledge is invaluable for digital strategy design and execution because measures can be implemented more profoundly and targeted. Service innovation (Lusch & Nambisan, 2015) builds the bridge between the general SDL and IS strategy research. With their multi-level approach for value cocreation, Storbacka et al. contributed to a more comprehensive understanding of the mechanics and nature of the phenomenon. They provide a system view of value cocreation that can be implemented in socio-technical solutions within organizations.

However, for this to happen, it requires more than abstract theory. Practitioners cannot understand and implement abstract service theory readily into their business environments. What it takes is proof of the service theory through practical application. Only then a broad practical application can occur (Nunamaker et al., 2015, p. 11). The SDA (Weiß et al., 2018) aims to serve this need. It translates the service theory into a blueprint for an IT architecture to support a service-driven business model, thereby providing the foundation of execution for digital strategies. Unlike related research, it ensures practicality by testing and reinforcing its ideas in real-life environments (Warg & Deetjen, 2021). With recent efforts to combine the SDA with the design process of Grotherr et al. (2020; 2018), SDA researchers are on their way to transform their framework into a generally valuable guide for building and operating service systems. Hence, the SDA is on the verge of becoming a central part of organizations' digital strategies.

2 RESEARCH DESIGN

This chapter serves to describe the research design of this master thesis. For this purpose, it once again presents the initial situation presented in the introduction. Subsequently, it presents the methodological approach of the research.

2.1 Research Framework

This thesis intends to provide practitioners and especially managers with a deep understanding of service-oriented DT. Specifically, its purpose is to give knowledge on designing, implementing, and continuously evolving service systems to foster service innovation. This purpose can be achieved by combining the SDA with a multilevel service systems design framework to create a new service systems design framework. Therefore, the research question of this thesis is "How can a multilevel service system design approach be applied to the design of effective systems of engagement using the Service-Dominant Architecture?"

The work presented here has been conducted near the research network around the SDA. Thus, it is positioned within the ecosystem of SDA research, and its results will feed back into the ongoing development of scientific knowledge around the SDA.

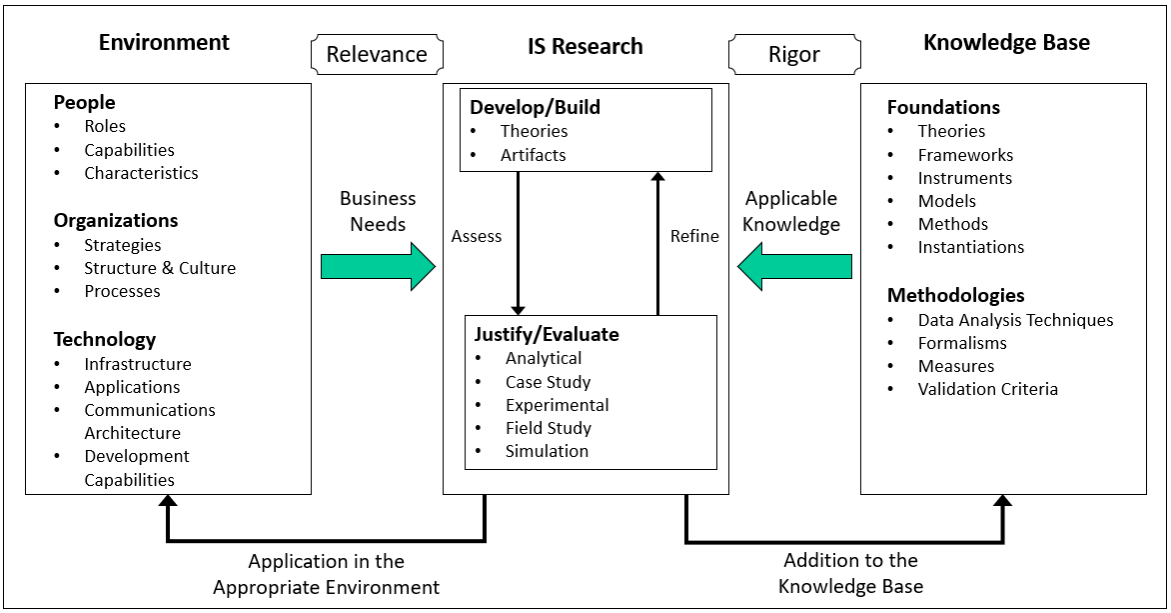
2.2 Design Science Research

The DSR approach based on Hevner, March, & Park (2004) and shown in Fig. 8 provided a general guideline for this research. DSR is the appropriate methodology when scientists need to solve real-life business needs (relevance) by developing theories and artifacts based on an existing theoretical knowledge base (rigor, Hevner et al., 2004, pp. 75-78). In the context of SDA and service innovation research, DSR is a valuable methodology because it compensates for missing empirical research in service science outside of the SDA realm. The relevance side can also serve to infuse new knowledge that can be drawn upon in developing respective artifacts in such a case. Thus, by expanding the knowledge base and presenting practical value in the research environment, DSR can contribute to research and practice (Hevner et al., 2004). Accordingly, the target audience of this work is twofold. The insights gained can assist SDA researchers in developing practical research findings. Business and IT professionals can use the research findings of this thesis to gain inspiration and guidance in their efforts to design a digital strategy for DT.

The previous critical literature review served the purpose of establishing the necessary rigor for the DSR project (Saunders et al., 2016, p. 70). It examined “the body of theory that has accumulated” (Saunders et al., 2016, p. 74) around the topics of DT, SDL, service innovation, the SDA, and service systems engineering. From the practical side, a case study (Yin, 2018) provided relevance to the research in business needs and practical knowledge to feed

the development process. Section 2.3 further describes the case study. Chapter 3 presents its results.

Fig. 8: The DSR Cycle

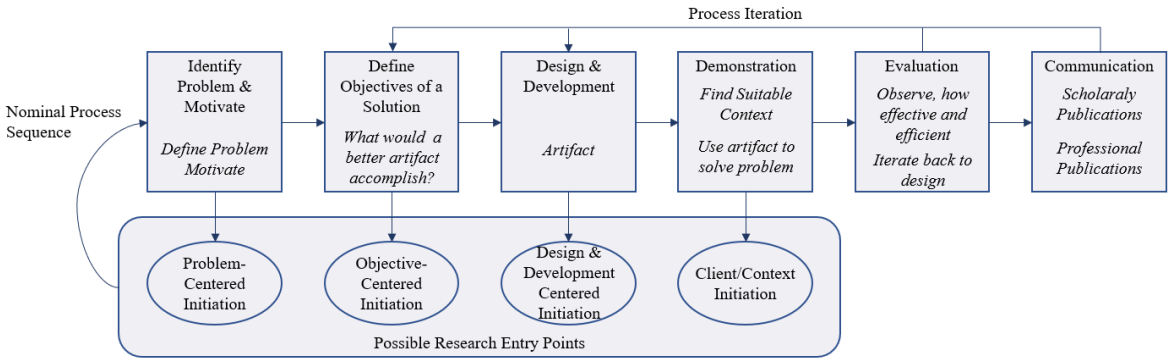


Source: Hevner et al. (2004, p. 80).

Central to the DSR project was developing a new IS artifact that responds to the research problem. In DSR, an artifact can be the instantiation of information systems, but it is not limited to that. Hevner et al. (2004, pp. 82-83) define it also to include the "constructs, models, and methods applied in the development and use of information systems." The artifact developed as part of this thesis falls under this latter definition. The starting point for the development was the multilevel design process of Grotherr et al. (2018). The findings from the case study and the critical literature review were then used to create a framework within the SDA to enable practitioners from business and IT to engage in interdisciplinary work for developing digital service systems.

The development phase was designed and conducted alongside the DSRM process model (DRSM) proposed by Peffers, Tuunanen, Rothenberg, & Chatterjee (2007) and shown in Fig. 9. The DSRM provides researchers with a general guideline for developing an artifact under DSR. The DSRM does not provide for a fixed sequence of activities but instead allows researchers to enter the development process at any stage, depending on the research context (Peffers et al., 2007, p. 56). As this thesis lies inside the SDA research ecosystem, it had access to a real-life instantiation and demonstration of the SDA. It entered the DSRM at activity 4 – Demonstration – to subsequently provide insights to be evaluated in activity 5. Based on the observations, the research iterated back to activity 2 to develop said artifact. At the end of this work, the insights are communicated through this research report and fed back into the SDA research community.

Fig. 9: DSRM



Adapted from Peffers et al. (2007, p. 54).

2.3 Case Study Design and Approach

The case study fulfilled a twofold purpose for the research. First, it served to explore the current performance of the SDA under real-life conditions to uncover development potential. Second, it intended to show how practitioners proceed in developing digital service systems to gain additional input for developing the framework. In general, the case study approach was considered the right one because it allowed examining an object under investigation in depth. Case studies consider that the findings of an object of study are composed of multiple variables of interest and can be based on numerous sources of evidence (Yin, 2018, pp. 14-15). Because the examined case is a real-life appliance, this was an important criterion for selecting the correct methodology. Therefore, Yin (2018) was chosen as the guiding literature to design and conduct the case study.

The case study was designed around five components (Yin, 2018, p. 26). A case study question gave the case study a general direction. In addition, four theoretical propositions ensured that the case study focused its attention on the right aspects in answering this question. These propositions were derived from the theoretical knowledge base around SDA and addressed different parts of the case study question. Tab. 6 presents the four propositions below.

Table 6: Theoretical Propositions of the Case Study

	Proposition
1	SDA helps practitioners understand the concepts of SDL.
2	SDA poses the foundation of execution for digital strategies.
3	SDA gives clear guidelines on how to design, build and operate service systems.
4	SDA helps with measuring the quality of service.

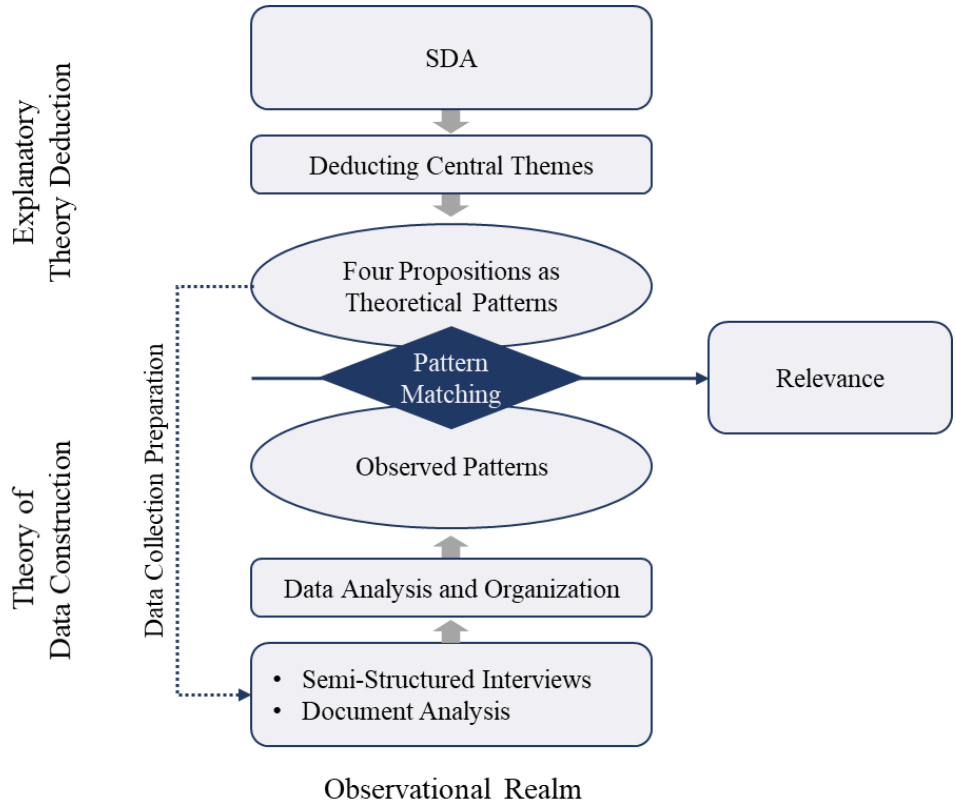
Own work.

In line with Yin (2018, p. 51), the case study was designed as an "embedded single-case study." The SDA, in the form of a technical artifact and architectural philosophy, formed the case. Within this case are two units of analysis related to the ongoing SDA research: Companies A and B. Chapter 3 presents them in more detail. Pattern matching (Sinkovics, 2018) guided the data collection and analysis within the case study has been chosen. It is a method of quantitative analysis that allows devising a theoretical pattern before collecting the actual data. This pattern then guides the analysis of the collected data, leading to a new practical pattern. Subsequently, pattern matching helps identify matches and deviations between theory and practice (Yin, 2018, p. 175).

The four theoretical propositions were chosen as the corresponding theoretical patterns to conduct the case study. They represented observations expected or hoped to be observable during data collection (Sinkovics, 2018, p. 472). Hence, the theoretical propositions also guided the data collection of analysis. Such an approach is in line with Yin (2018, p. 165), who states that data collection and resulting data must fit the analysis method.

Fig. 10 illustrates how the theoretical and practical patterns were created and matched to overview the research design. After the pattern matching procedure, the identified deviations between theory and practice helped formulate requirements for developing the targeted artifact of the DSR project.

Fig. 10: Case Study Design



Adapted from Sinkovics (2018, p. 471).

2.3.1 Data Collection Methods

The case study used two types of data collection methods: Semi-structured interviews and document analysis. These are presented subsequently in more detail.

2.3.1.1 Semi-Structured Interviews

Four semi-structured interviews (Saunders et al., 2016, p. 391) allowed in-depth conversations with practitioners from the case companies. As the research topic was complex and the interview partners, all C-level staff (see 2.3.2), promised to be knowledgeable, the chosen interview format seemed appropriate. Semi-structured interviews allow in-depth conversations (Saunders et al., 2016, p. 391). In addition, establishing personal contact with the interviewees was beneficial. C-level interviewees are usually not motivated to contribute to off-curriculum activities, but personal interaction can leverage their enthusiasm about the research (Saunders et al., 2016, pp. 393-394).

As for the preparation of the interviews, the knowledge of the critical literature review and the theoretical propositions helped to begin with creating a catalog of 12 level-two questions (Yin, 2018, p. 100). These questions were then divided thematically into IT-related and business-related questions to develop an interview guide for each interview that matched the position of the respective interviewee. Thoroughly preparing the interviews in advance addressed the threads to data quality regarding interviewer and interviewee bias (Saunders et al., 2016, p. 399).

Due to the COVID-19 pandemic, the interviews were conducted exclusively via video conferencing. The duration of each interview was around 30 minutes. The shortest interview took 29:15 minutes, the longest 44:41 minutes. Two one-hour coffee talks were held with the fourth interviewee. For the first three conversations, there was permission to record and transcribe. Notes were taken during the two coffee conversations with interviewee four.

2.3.1.2 Document Analysis

The purpose of collecting and evaluating documents publicly available from the case companies was to triangulate the interview results (Yin, 2018, pp. 113-117). Retrieved documents were presentations, press releases, scientific publications, and web pages.

2.3.2 Sample Description

The sample for the interviews contained four elements. All of them are C-level managers from two companies, A and B, working either in IT or business positions around the case. Tab. 7 summarizes the characteristics of each sample element.

Table 7: Sample Elements of the Case Study

Case Element	Position	Company	Interview type	Duration
1	Chairman of the Advisory Board	A	Semi-structured Interview	35:00 minutes
2	Chief Executive Officer (CEO)	A	Semi-structured Interview	44:41 minutes
3	Chief Technology Officer (CTO)	A	Semi-structured Interview	29:15 minutes
4	Managing director	B	Coffee Talk Interview	2 x 60 minutes

Own work.

2.3.3 Data Analysis Methods

Due to the methods described above, the data obtained from the case study was predominantly qualitative. Using thematic analysis (Saunders et al., 2016, p. 579) and the theoretical propositions as a guideline, practical patterns were sought and identified in these data to provide the basis for the subsequent pattern matching.

The resulting findings from practice were then matched to the theoretical propositions (Sinkovics, 2018) to find matches and deviations between theory and practice. The use of coding and corresponding software for data analysis did not seem necessary for reasons of practicality.

3 CASE STUDY DESCRIPTION

This chapter focuses on presenting the results of the case study. As mentioned, it has been conducted with two firms inside the ecosystem of SDA research to generate practical relevance for the DSR project. Both companies are subject to SDA research and serve to test new research results and feed the research with the latest practical knowledge about service science.

The next section presents the two case companies in more detail. After that, the following section presents the case study results. It does so in the way the pattern matching was conducted, meaning that the practical insights retrieved from the case study are subsumed under the corresponding theoretical proposition. Finally, a third section concludes the case study by theorizing on the results of the pattern matching.

3.1 Introduction of the Case Companies

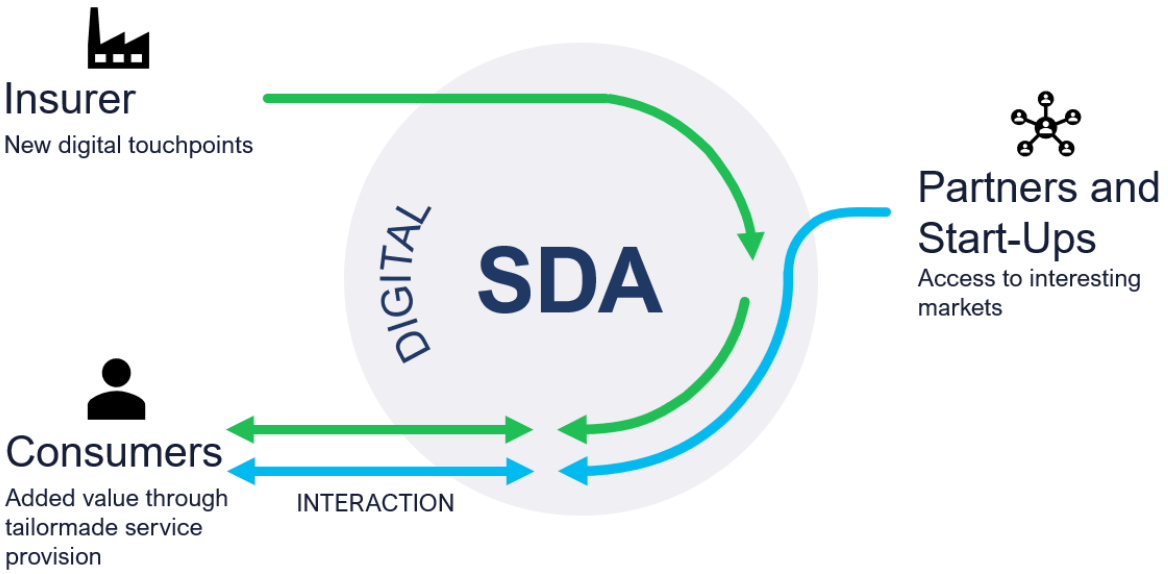
The case study conducted in this thesis dealt with two companies that were considered embedded units of analysis under SDA. The research design already referred to them as Company A and B. Both companies are presented in the following, focusing on their organization and business models.

3.1.1 Company A: SDA SE Open Industry Solutions

Company A is a German company based in Hamburg. It is a joint venture of the major German insurers SIGNAL IDUNA, Allianz X, Debeka, and the IT consulting company msg. It also cooperates with IBM Germany and McKinsey (SDA SE Open Industry Solutions, 2021b). The company was created in the course of SDA research and serves as a practical implementation of the theoretical artifact of the SDA (Warg et al., 2019).

Company A develops and operates an IT platform modeled on the SDA. The current business model provides insurance companies with a development environment and production line for their service ecosystem and platform. Customers can either join Company A's platform or create their own instance of it. Customers can thus draw on existing digital service capabilities or develop new ones by benefiting from existing knowledge and tools inside Company A. As a consequence, Company A is a service provider that supports its customers in their DT (see Fig. 11).

Fig. 11: Company A's Value Proposition



Adapted from SDA SE Open Industry Solutions (2021a).

Three interviews were conducted with top managers from Company A as part of the case study: The company founder and simultaneous chairman of the supervisory board, the CEO,

and the CTO. The interviews were thematically adapted to the interviewees' areas of responsibility.

3.1.2 Company B: ION Access & Health GmbH

Company B is an international joint venture based in Munich. Participating companies are inHealth, Nortal from Estonia, Daman from the United Arabic Emirates, and ottonova from Munich. Like Company A, ottonova is the subject of SDA research (Rittweger et al., 2020; Weiß et al., 2021).

Company B was necessary for the case study because it developed an app in response to the COVID-19 pandemic that connects medical providers, regulatory authorities, access authorities, and travelers through a platform. An app, "SimplyGo," is designed to make pandemic travel possible again by providing travelers with daily updates on regulations in the destination area. It also enables the transmission of health data such as vaccination status to official parties, ensuring the smoothest possible immigration to the destination country (ION Access & Health GmbH, 2021a). With its platform, Company B also joined the ecosystem of airport service provider SITA to quickly deploy its capabilities to the world's biggest travel hubs (ION Access & Health GmbH, 2021b). Unlike Company A, SimplyGo also connects directly with the end consumer and actively promotes AE. Therefore, the case study emphasized the activities related to actor-centric platform design and operation at Company B.

Two coffee talks were held with the managing director of Company B and simultaneous CIO of ottonova from Munich.

3.2 Presentation of the Case Study Results

The presentation of the case study results and the analysis of the data obtained bases on the theoretical pattern, which chapter 2.2 presented as four propositions. This approach corresponds to pattern matching (Sinkovics, 2018). It compares the empirical findings of the data analysis with the theoretical propositions to match theory and practice in the subsequent section

In the following, the four theoretical propositions serve as headings. The corresponding findings from practice are presented below each of them.

3.2.1 “SDA helps practitioners understand the concepts of SDL”

The first proposition claims that the SDA would actively contribute to practitioners' understanding of SDL concepts in their everyday operations. The interviews partly confirm this statement as the data makes it apparent that a distinction between the two firms and other,

less SDA-influenced environments is necessary. On the one hand, there are the two Companies A and B. They are young, dynamic, and, above all, were founded with a clear focus on service operation. Their ongoing involvement in SDA research means that they have always been close to the topic. On the other side are companies like the customers of Company A. These are primarily large, established insurance companies with a profit- and product orientation. Their managers have internalized a philosophy that follows GDL (Element 2, App. 2).

Company A has fully adopted the language, logic, and mechanisms that the SDL specifies and are necessary for service innovation. As Element 1 said, the organization applies the SDL "one to one: building up more and more resource liquefication and resource density and then enabling ever new combinatorial evolutions and also emergence" (App. 3). To achieve this, Company A currently works on a service catalog. Once ready for use, this catalog holds multiple so-called service stacks. As building blocks, each of these service stacks contains the institutions and technical functionalities needed to create digital service offerings. For example, this can be a chat or contract functionality (Warg & Deetjen, 2021, p. 13) together with the security standards and tools to ensure secure communication or data transfer (Element 1, App. 3).

From a strategic point of view, the intention behind such a service catalog is to create a repository of many different service functions that can be used to build individual service systems for individual customers. In addition, by slowly venturing into markets outside the insurance industry, Company A begins to add even more service stacks with functionalities from other business areas within its service catalog. In doing so, they hope to create a platform that benefits from a diverse set of digital capabilities that enables the creation of previously unreachable service functions. This can result in true emergence for Company 1: "What I mean by emergence is [...] that we have a generic, specialized catalog of building blocks like Lego bricks, where you can build not only the pony farm and the knight's castle but also the pirate ship. All of a sudden, you can build a pirate ship out of what you have without having to spend any money. you have to spend money."

Company B has also managed to internalize the idea of SDL. The main difference is that it does not strictly follow the SDL language. Nevertheless, they have taken up the ideas of service innovation and value cocreation as their business philosophy. This becomes apparent through how they conceptualize their platform as an intermediary of service-for-service exchange. For them, patterns of how actors engage with SimplyGo have become a central design question as they have realized that actor-centric value creation paths are essential.

For companies that do not come from the SDA research ecosystem, the situation is different. In the interview, Element 2 stated the following: "That [the SDL] is always so intrinsic to us, but companies that have been around for 20 or 30 years and, let's say, managers who have been trained completely differently and somehow graduated 30 or 40 years ago, they really are in a different world. You can't underestimate that" (App. 2).

Company A's customers have indeed recognized that a product orientation is no longer viable for their future. But for managers and IT experts who do not know the SDL, the service idea is challenging to grasp in its maturest form (Element 1, App. 3). Companies that come from a GDL have mental barriers to opening up to an ecosystem. However, openness is vital for exchanging operant resources and service. Companies are not used to sharing, for example, their unique selling point in the form of data or process knowledge with other partners to grow together (Element 2, App. 2). Company A, therefore, brings the topic closer to its customers in a logical sequence: after customers accept the importance of service, they address the issue of interaction next. With interaction, the topic of agile working comes up as an essential enabler of iterative action and learning. Understanding the customer through interaction is crucial to provide better service. Company A proceeds then with examining important actors around their customers. The fact that there are not only human actors in the service world is not self-evident to them. Finally, the narrative comes to platforms that connect and support different actors and their interaction with the customer. EA is also recognized as necessary in this narrative process, which ultimately leads to SDA as a solution. Company A sees the regular need of going through this story with their customers to help them understand the essential functions behind a service business model and its requirements. This persuasion process is lengthy and can take two years (Element 1, App. 3).

Practitioners without a service background find it difficult to understand what SDL and service innovation are all about. They find it difficult to accept the consequences of an SDL-based business model. As shown, Company A has a realistic, observable implementation of the theoretical framework. First, it enables the company to operate an innovative business model for itself and customers based on SDL ideas. On the other hand, this real-world instance provides Company A with the opportunity to break down mental and organizational barriers among its customers to facilitate their access to the world of services. In this respect, however, the SDA remains a tool for Company A to convince others of the potential of the SDL. As a stand-alone solution, the SDA is still too obscure for companies outside of research to adopt and implement on their own.

3.2.2 “SDA poses the foundation of execution for digital strategies”

The second proposition emerges directly from the SDA literature. Weiß et al. claim that the SDA is the basis for a strategic approach to service innovation as it builds the foundation of execution for digital strategies (2018, pp. 1635-1636). The case study results verify this statement for both Companies A and B. Moreover, company A's customers also consider the SDA an essential component of their DT.

Above all, the SDA provides Company A and its customers with a guide for designing and implementing their digital, service-oriented business models. It is helpful in two different but related tasks: First, the SDA acts as a “heuristic search logic” for practitioners to define

new strategic goals. Second, it subsequently helps design and build the appropriate IT architecture to reach these goals (Element 2, App. 2).

Company A uses the SDA to check for itself or with its customers where interesting inventory data and other operant resources exist (SOOR) that other companies do not have in that specific form (i.e., a unique selling point). They also search for currently existing customer relationships. With this information, they work on ideas to bring these operant resources (knowledge, processes, etc.) to the customer to generate new value experiences. Conceptualizing the SOI helps to think of new ways of interacting with the consumer. The concept of the SDA's SOP opens new perspectives for integrating partners to generate additional value. As a result of this search process, Company A can already paint a pretty concrete picture of a future business model containing required organizational capabilities (Element 2, App. 2).

An appropriate IT architecture must support these new business functions (Element 2, App. 2). In the next step, the SDA supports the design of this IT architecture. Here, Company A's modular service platform comes into play: It makes it possible to build ever new platform configurations with new, innovative service offerings from its existing service stacks. Minimal Viable Products (MVP) can thus be realized in the shortest possible time and gradually expanded. There is no need for prototypes that have to be subsequently adapted or completely reprogrammed. As a result, Company A can offer a test platform that costs five times less than traditional approaches to software prototyping. And these MVPs are future-proof because customers can always extend their service solutions with new service stacks (Element 1, App. 3). Another benefit of the service stack approach was highlighted by Element 1 through the following: "And the value here is clearly that […] we can do a very small use case for a small and medium-sized company, but we can do just as much as a digital transformation as we have shown at Signal Iduna. Of course, for a digital transformation, you then have to set up more on the platform. But you can kiss this basic SDA framework awake very quickly and in a very manageable way. And I think that was very important."

Company A actively promotes this approach. For example, in a case study (Warg et al., 2019), the company succeeded in using the SDA to redesign the care application process for patients in need of long-term care with their health insurer:

Those relying on long-term care must overcome high administrative hurdles to apply for financial support in Germany. This is due to a lack of process automation by health insurers. Consequentially, processes are slow and paper-based. Entries in the SoR are manually performed, and experts must certify the patient's need for care. As a result, a care application typically takes five to nine days to complete.

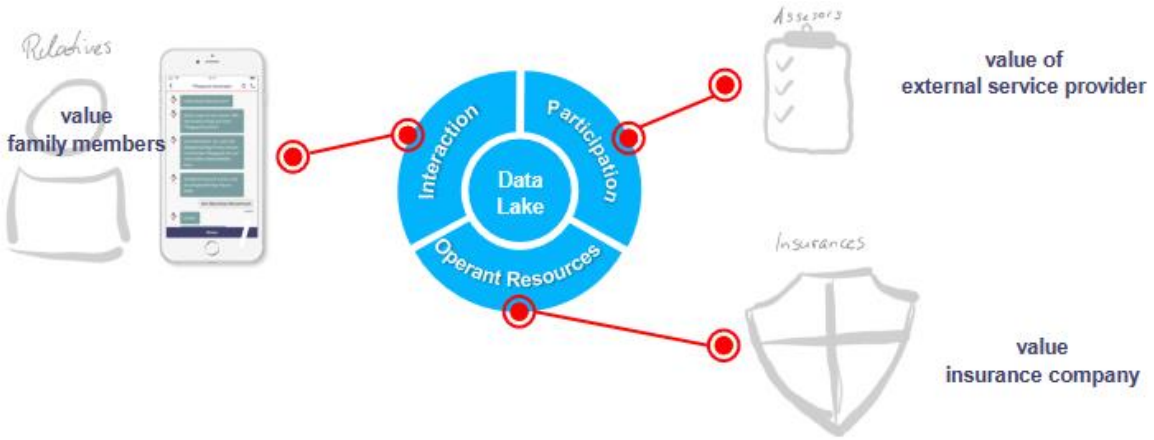
Company A has worked with a health insurer to build a platform that introduces an entirely new application process. At the heart of the process is the user app "edith.care," through which patients fill out their care applications quickly and easily. The platform behind edith.care integrates all players in the ecosystem: patients or relatives are connected in real-

time via the SOI (i.e., the app). The SOP connects external experts. The SOOR and the data lake handle the connection and communication with the insurer's SoR (see Fig. 12).

The resulting service system automates the entire process except for the application process. However, because this takes place in real-time in interaction with the platform and the other actors, Company A reduced the duration of the application process to five minutes.

The case study shows that the SDA fulfills its purpose of supporting a holistic strategic approach towards DT. The literature states that the SDA helps to envision newly necessary enterprise capabilities to support SDL-based business models. It also says that the SDA provides a blueprint of the IT architecture and related IT capabilities needed to support such business models. Based on the previously discussed evidence, this can be confirmed.

Fig. 12: Customer-Centric Redesign of the Application Process



Source: Warg et al. (2019, p. 10).

3.2.3 "SDA gives clear guidelines on how to design, build and operate service systems"

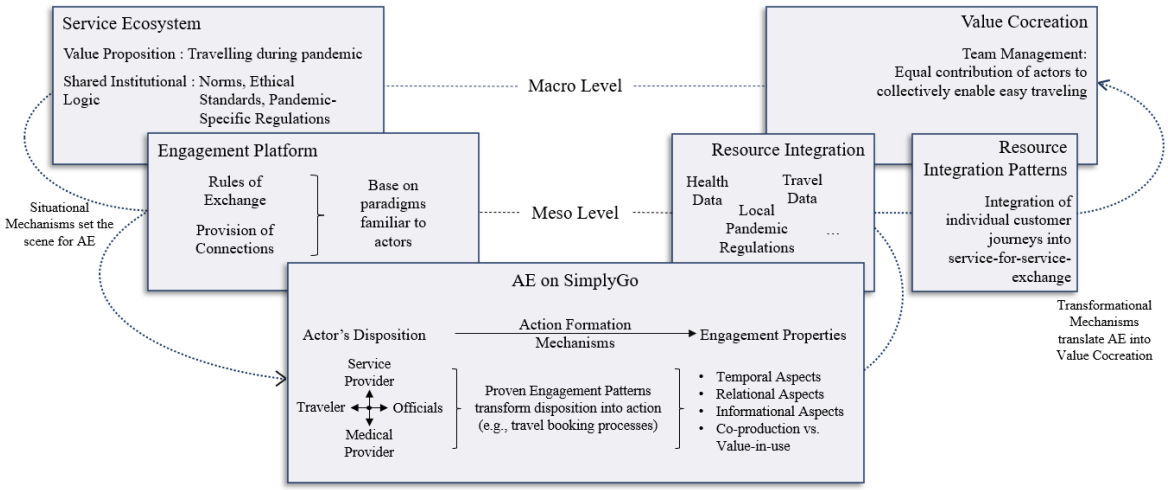
Proposition 3 addresses the topic of service systems engineering. Under proposition 2, it became clear that the SDA helps design new service-driven business models and appropriate IT architectures and thus acts as the foundation of execution for digital strategies. However, this insight does not provide sufficient information about how well the SDA supports practitioners in different steps of strategy execution, especially in building appropriate IT infrastructures. Therefore, the following section looks at the strengths of SDA in guiding service system design and engineering.

3.2.3.1 Company B

The data collection at Company B was beneficial for understanding how practitioners engineer service systems in compliance with the multilevel approach of value cocreation (Storbacka et al., 2016). The insights are not focused solely on technical works but show how well the practitioners design and run their business model holistically along with Coleman's bathtub, too (see Fig. 13).

At the macro-level of SimplyGo lies the ecosystem with its four types of stakeholders mentioned earlier. Company B offers each group of actors a different value proposition. Yet, at the same time, they are all united by a shared institutional logic composed of various institutions. These are, for example, pandemic travel restrictions, data protection, or user requirements of specific user groups such as children and the elderly. Combined, these institutions provide the unified institutional logic under which SimplyGo can operate (Element 4, App. 5).

Fig. 13: Value Cocreation at SimplyGo



Own work.

Company B further provides a platform - SimplyGo - at the meso-level that integrates the various actors and their resources into the ecosystem and enables the exchange of such. Because the platform's design contributes significantly to the actor's disposition at the micro-level, they put great emphasis on familiar engagement environments for the various actors. For example, the SimplyGo app borrows from well-known travel booking portals. It builds on proven processes that consumers are used to for booking and traveling. The same is true for medical providers. Testing procedures and the processes of data evaluation and transmission are always the same. SimplyGo joins them rather than developing something of its own that is incompatible. In terms of immigration authorities, SimplyGo uses SITA's network. SITA is one of the leading airport service providers handling passenger data across multiple airports internationally. They transmit passenger data to the authorities at the destination

upon departure in air travel. (ION Access & Health GmbH, 2021b). SimplyGo supplements this data with health information such as vaccination status or test status (Element 4, App. 5) to enable smooth immigration procedures.

Through this platform design, SimplyGo attempts to replicate natural, established processes. Concrete engagement patterns are technically mapped inside the platform to instantiate and foster action-formation mechanisms on the micro-level. AE then becomes visible, depending on the type of actor, in the form of group-specific engagement with the respective customer journey (Element 4, App. 5).

Back at the meso-level, SimplyGo's task is to transform resource integration activities into value cocreation. Resource integration patterns are of importance here. At SimplyGo, these patterns are visible as individual customer journeys for each type of actor. The platform acknowledges their unique characteristics and puts itself in between the various actors as a type of intermediary for their resource integration and interaction. It, therefore, stabilizes the behavior of different actors and provides the infrastructure that enables service-for-service exchange (Element 4, App. 5).

The result is value cocreation at the macro level. SimplyGo simplifies pandemic travel for each actor by integrating and orchestrating their resources. SimplyGo is an observable service-for-service exchange, as actors share their resources for mutual value experiences. Travelers can once again travel more comfortably and always stay informed. Medical providers can transmit health data quickly and easily. Immigration authorities can keep the extra administrative workload in check, and local service providers can offer their services (Element 4, App. 5, ION Access & Health GmbH, 2021a).

3.2.3.2 Company A

At Company A, the case study revealed technical insights into the design and development works regarding the firm's IT infrastructure. The following sections present Company A's service platform structure and the related engineering methods they use.

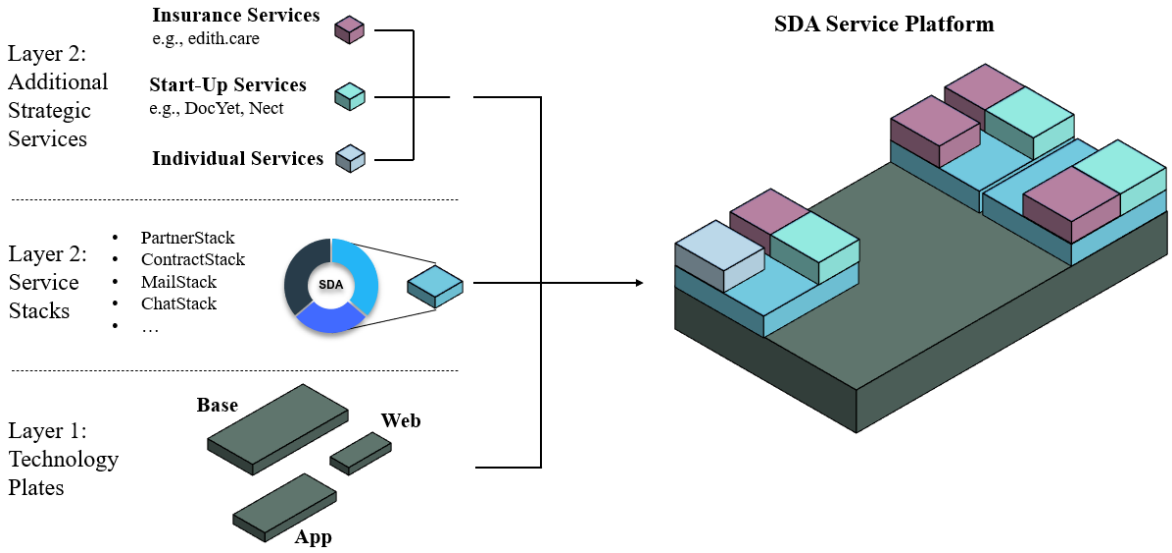
Structure of the Platform

As mentioned, Company A maintains a service platform that pursues two goals: On the one hand, they intend independent service modules to create new organizational capabilities for the company itself. But, on the other hand, this modular platform should also build service systems for Company A's customers using existing service stacks. For this purpose, the platform's architecture is divided into three levels (Element 3, App. 4): On the lowest level, figuratively speaking, there is a macro-architecture consisting of three plates. On top of these lie the generic service stacks as functional service components that Company A manages itself. Finally, at the top level are various additional service stacks integrated by, for example, external partners (see Fig. 14).

The macro architecture of the service platform consists of three independent plates: a base plate, an app plate, and a web plate. For example, the app plate contains institutions that define the framework and "rules of the game" to ensure the development of coherent apps compatible with other platform building blocks. The base plate includes best practices, guides, and a pipeline through which software developers can quickly deploy changes to the platform. It also provides unified developer tools to create an overall coherent system. Thus, the technology plates serve the purpose of ensuring that uniform programming paradigms, tools, libraries, etc., are available and used across the enterprise for developing socio-technical components (Element 3, App. 4).

Company A builds its generic service stacks on the middle layer, atop the technology plates, and according to their general specifications. Each stack serves a specific purpose, for example, contract management, mailing, chat, etc. In addition, they build every stack according to the SDA architecture. That is, each stack has an SOI, SOP, SOOR and can access a data lake (see Fig. 14). Finally, non-specialist stacks form the top layer of the platform's architecture. Some of these stacks are external, and some are individually programmed for a specific purpose. They bring additional capabilities to the platform (Element 3, App. 4).

Fig. 14: Architecture of Company A's Service Platform



Adapted from Warg & Deetjen (2021, p. 13)

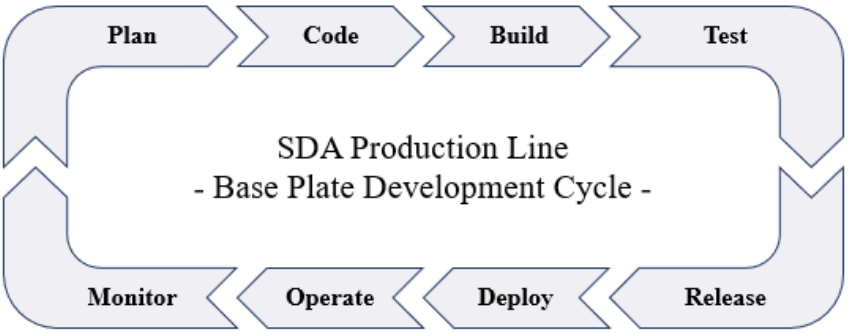
The entire platform is microservice-oriented. The individual service stacks are composed of microservices. But Company A also considers the separate service stacks as microservices since they hold specific functionalities: "Our definition of microservices is not a hard definition, but something that can be described as a "rule-of-thumb." For example, we say that there should be complete data replication across all services. And the individual services hold the data they need to answer other requests. They keep track of the request, can call other services, and can thus also use central replications of the ODSs to provide information" (Element 3, App. 4). The resulting modular structure of the platform enables Company A to

combine the various service stacks, again and again, to create new service solutions with improved resource density for better value propositions and customer experiences (Element 1, App. 3).

Development Processes

Company A is completely agile-oriented in its organization. Product owners and independent developer teams manage the various building blocks. As a consequence, work within the teams follows agile principles (see Fig. 15). Nevertheless, each team must not lose sight of the big picture, the complete service platform. To this end, the teams use Spotify Backstage as a central interface in their work. On the one hand, this makes the platform more understandable by providing visualization aids (company-wide consensus). On the other hand, it provides insights into the service capabilities on the platform (for example, in the form of a service catalog) that can be accessed (Element 3, App. 4).

Fig. 15: Development Process on the Base Plate



Adapted from IfSD Hamburg (2020, p. 22).

As already explained under Proposition 2, at the strategic level, the SDA serves as a search logic in the first step and then for the concretization and implementation of technical requirements. This two-level approach is reflected in Company A's process for developing new digital service solutions. From the very beginning, they emphasize interdisciplinary collaboration and customer-centricity. In this sense, Company A's work philosophy lends from software-engineering paradigms such as Design Thinking or Domain-Driven Design (DDD) across all phases of solution development (Element 2, App. 2). Especially the fundamental role of DDD was stressed by Element 2 when he said the following: "[…], so Domain-Driven Design is a basis for how we set ourselves up. We use it as a guide, which ultimately also helps us to find the right language at the technical level if we can orient ourselves in domains" (App. 2). In addition, Element 3 highlighted that technical experts communicate with potential customers from the beginning of a new project. This method ensures that the developed IT solutions are also a technical fit for the desired business solution. (Element 3, App. 4).

In the initial phases of software development, Company A focuses on defining a problem space and creating a vision that can fill it. The concept is then transformed into a technical prototype and tested to gather feedback in a natural or experimental environment. Customer journeys are a critical aspect of designing the prototype's engagement patterns. This step is followed by optimization and further development of the solution until it becomes a deployable service. Therefore, the development process is iterative and follows basic principles of agile software development, both in detail and in its entirety (Element 3, App. 4).

3.2.3.3 Insights

Although the companies studied have successfully developed service platforms and the SDA has given them important insights and a general direction, Proposition 3 cannot be confirmed. The SDA did provide the general approaches to how the companies should build their IT architecture. However, it has not given them a concrete target image that they could follow. The two companies' platforms differ substantially and result from year-long development (especially at Company A). In addition, the argument that already applied to Proposition 1 should also be listed here. Both companies are young and agile and were built and developed with a clear, SDL-inspired service mindset. For them, the conditions could not have been better to create an SDA-oriented IT landscape. For established companies, this step is not so easy. This can be concluded from the fact that the joint venture partners founded company A for the very purpose of developing a service platform that they could use to run their own service businesses. They do not have the necessary capabilities to do so. This conclusion also arises from the fact that Company A's customers need a lot of education in building and operating service platforms (Element 1, App. 3).

3.2.4 "SDA helps with measuring the quality of service"

Proposition 4 addresses another essential aspect of operating a service business model. As with any business, it is also for a service business important to regularly assess and adjust its setup (Grotherr et al., 2020). The case study data confirm this theoretical proposition. As the SDA points towards a (modular) IT architecture to carry the service business model, there are, by nature, possibilities to track and assess activities within the service ecosystem and related IT infrastructure. This is true for both Company A and B.

For example, in Company A, assessing the quality of service is done by working with live metrics of various systems and service stacks. For applications like edith.care, how actors interact with the system, and how frequently they do so is of interest. As part of the Design Thinking methodology, collecting and analyzing end-user and customer feedback is also essential. In this regard, Element 3 stressed the importance of actively asking for user feedback: "People only ever come to that when they're annoyed. So, you always have to ask for it proactively to pull that up with positive reviews and get a few more things out of it" (App.

4). Furthermore, Company A regularly invites key stakeholders such as customers to attend scrum meetings to increase transparency and keep development on track (Element 2, App. 2).

SimplyGo relies on user feedback, too, to ensure service and engagement quality (Element 4, App. 5). To test SimplyGo's compatibility with real-life scenarios, Company B and SITA conducted a pilot project where they tried the platform with vaccinated and non-vaccinated passengers on their trip from Germany to the UAE (ION Access & Health GmbH, 2021b). And since SimplyGo consists of IT systems, also Company B uses live metrics to assess system health and performance (Element 4, App. 5).

3.3 Summary of the Case Study Results

The case study has provided insights into the current performance of SDA in real-life environments. During pattern matching, it confirmed two of the propositions made with insights from the companies examined. A third could be partially confirmed. One proposition the case study refuted. In these two latter cases, the patterns observable in practice did not match the theoretical patterns.

This finding is valuable for the further development of the SDA because it shows two things: First, the SDA lags behind the needs of practice in these two areas. They represent concrete needs of practice, which the SDA is currently not able to serve. Second, it was possible to identify aspects in the interviews in which practice is already further along than research. In this respect, the SDA can benefit from the insights gained through the case study.

3.3.1 Need for Further Research

The case study could only partially confirm proposition 1 and refutes proposition 3. Therefore, SDA must evolve in these areas.

By partially disproving proposition 1, it becomes apparent that there is still a gap between the SDL and the SDA. The SDA indeed manages to integrate service thinking into two businesses that are close to research. However, practitioners outside this narrow circle still have mental barriers that keep them stuck in their old thought patterns of a GDL. It takes communication, persuasion, and a lot of time to convince people outside the SDA research community of the added value of this approach, not to mention communicating the concepts behind it. This is indirectly confirmed by Element 2, who would like to see an SDA canvas that is simple, understandable, and easy to apply in practice (App. 2). Consequently, the SDA would benefit from being expanded in a direction that makes SDL concepts even more tangible.

In addition, the discussion around proposition 3 has shown that the SDA is not yet a widely applicable framework that practitioners can use to build and operate an SDL-influenced IT

architecture as part of their digital strategy. While the SDA provides a respective target picture, it does not describe the path to get there. The companies interviewed grew close to the SDA research, so evolving according to a service worldview was easy. Incumbent companies that already have difficulties understanding the SDL will also have problems here. They would benefit from developing the SDA in the current approach toward a design framework for designing and building SoE (Rittweger et al., 2020; Weiß et al., 2021).

3.3.2 Where Practice Enriches Research

On the other hand, the case study has also shown that the two companies under consideration can contribute to the further development of SDA in the two points mentioned above. They have successfully implemented service systems that provide the necessary capabilities to enable service innovation in each case. The insights can also be translated back into the SDA. This is true because, although both companies have introduced independent development processes at their core, they still rely on generally accepted approaches such as Design Thinking, DDD, customer journeys, or even the multilevel approach of value cocreation.

3.3.3 Conclusion

The case study was a valuable contribution to evaluating the current state of SDA and its further development. It examined both an actual instance of SDA and a platform solution influenced by SDA research. This helped uncover where SDA meets its theoretical claims in practice and what business needs still exist and thus need to be solved. SDA must be further developed and refined in these areas. The insights gained in the case study may be helpful in this regard. In general, however, the SDA is on its way to becoming an effective tool for practitioners to drive digital strategies if it successfully masters this next development step.

4 DISCUSSION OF ARTEFACT AND FINDINGS

The research question of this thesis was "How can a multilevel service system design approach be applied to the design of effective systems of engagement using the Service-Dominant Architecture?". The goal was to link the multilevel design process of Grotherr et al. (2018) with SDA to develop a framework that provides practitioners with the knowledge needed for designing effective service systems. Throughout the development phase of the DSR project, the presented research achieved this goal.

The following sections present and discuss the development and the resulting framework. Then they answer the research question and interpret what implications the solutions have for practice and research. Finally, a discussion of the existing limitations of the solution follows.

4.1 Artifact Development

The core of the DSR project was developing a new IS artifact to answer the research question and fulfill the purpose of this thesis. The insights from the critical literature review and the case study fed into the development phase, which is consistent with DSR (Hevner et al., 2004). The result of the development phase is a framework connecting SDA research with the multilevel process for service systems design of Grotherr et al. (2018) and general service innovation concepts (Storbacka et al., 2016). The framework extends the research body of SDA research.

4.1.1 Development Approach

Chapter 2.2 discussed the development approach based on the DRSM (Peffers et al., 2007). Accordingly, the development phase started at step 2 - defining the objectives of a solution - and only then proceeded to the actual development of a new IS artifact. The findings of the case study served to define the objectives the framework should fulfill. Propositions 1 and 3 could not be confirmed in practice and were thus the starting point for development. The framework had to meet the following two goals:

1. SDL theory should be brought down to a lower level of abstraction so that practitioners outside of service science and SDA research could understand it and see its added value.
2. The framework should be a broadly applicable guide for the iterative design, development, and operation of effective service systems. It should also be holistic, thus consider both business and IT as a central component of a digital strategy for DT. The business side should focus primarily on making emergence and resource integration patterns more comprehensible. On the IT side, it should highlight the essential role of modular architectures.

SDA research provided the ideal environment for achieving these objectives due to its proximity to real-life cases. A key issue preventing a broad understanding of SDL and service theory is that literature lacks reference to tangible use-cases. The case study demonstrated a link of SimplyGo to both SDA and the multilevel approach to value cocreation. Therefore, it was appropriate to use this example to link the new framework to reality and make the results more tangible.

The first step was to compare Company B's approach to designing SimplyGo with the process of Grotherr et al.'s work (2018). As discussed in chapter 3, Company B's understanding of running a service platform matches the multilevel approach to value cocreation (Storbacka et al., 2016). If SimplyGo were to be used as a practical example to combine Grotherr et al.' process with the SDA, this comparison was necessary to test whether SimplyGo also matches the procedures proposed by the multilevel design framework. Tables initially helped to reduce the abstraction of the ADR steps within the two design cycles to make them fit with practical development works. Over this course, working methods and tools for some of these steps were identified (e.g., see App. 16-18). Then, these refined, more practice-oriented process steps allowed comparison with how Company B designs and runs its SimplyGo platform.

The comparison revealed that practitioners at Company B rely on similar practices for designing service systems as Grotherr et al. (see previous chapters). This inspired a possible approach for building the framework: instead of extending the framework like initially planned, the SimplyGo service ecosystem, including the platform, could be visualized. A visual image of a real-life service system, including actors, resources, the service platform, mechanisms at work, etc., benefits that it can translate abstract theoretical concepts into an understandable format. Unlike the previously mentioned tables, such a visualization does not rely on research-specific terms like action-formation mechanisms, resource integration patterns, and the like. Instead, it makes these concepts observable.

DDD was an appropriate tool for this translation since it is a generally understood language in software design and engineering (Evans, 2003). DDD is a methodology that allows breaking down a distinct domain (like a business case) into sub-domains. These sub-domains can then be examined and serve as a blueprint for building appropriate modular IT architectures that support the domain (Richardson, 2021b). DDD is particularly valuable for modular IT architectures because it ensures the technical fit of an IT solution for the business problem at hand. Therefore, it increases the quality and effectiveness of software (Evans, 2003).

DDD allowed resolving SimplyGo into parts without losing sight of their interrelationships. Especially two DDD tools were necessary: context mapping is a method of strategic DDD that splits complex systems into individual sub-domains, known as bounded contexts. It can also represent these subdomains' interrelationships in a high-level view of the system (Vernon, 2016, Ch. 3). At a more detailed level, tactical DDD provides tools to further resolve the bounded contexts into aggregates that show and explain the business logic within

each of them. For this, it uses ubiquitous language inside each bounded context. This is crucial for building problem-oriented modular architectures (Vernon, 2016, Ch. 4).

DDD provided the ideal tool for developing the final framework. For one, it allowed resolving SimplyGo into several bounded contexts to show the main elements of its service ecosystem. In addition, using tactical DDD provided the necessary tools and language to design the various bounded contexts based on the findings from the case study in detail. Another important aspect is that DDD is a methodology that has been around for some years and is known to both software engineers and domain experts. And even if not, it is a methodology that is learnable with limited effort.

ContextMapper framework served as the tool to create the graphical representations of the framework. It is an open-source framework with which strategic and tactical DDD is possible via a standalone programming language. Among other things, ContextMapper supports the visualization of code in DDD format using PlantUML (University of Applied Sciences of Eastern Switzerland, 2021). Although the output of ContextMapper looks similar to diagrams from the Unified Modeling Language (UML), the produced context maps and tactical DDD maps do not necessarily need to adhere to UML rules. DDD is a standalone methodology and was treated as such during the development of the framework.

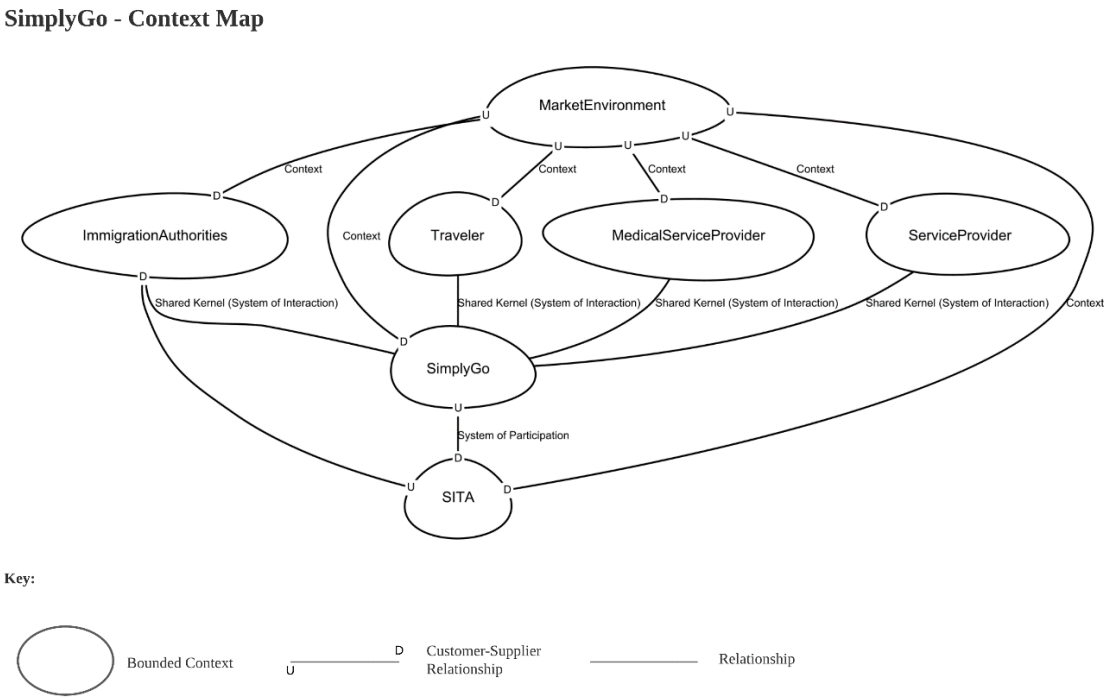
4.1.2 Framework Presentation

The framework that resulted from the DSR project consists of two components. First, a context map (Vernon, 2016, Ch. 3) shows the SimplyGo service system divided into seven independent bounded contexts. Second, these bounded contexts were then considered individually and split into their components using tactical DDD (Vernon, 2016, Ch. 4). All artifact components are available in the remaining text or App. 6-12. In addition, a GitHub repository under https://github.com/NormanBaechtold/ServiceSystem[4] holds the images in high-resolution, printable versions. The repository provided is an integral part of this research report.

The context map is the higher-level component of the framework (see Fig. 16). SimplyGo has been divided into seven bounded contexts: MarketEnvironment, ImmigrationAuthorities, ServiceProvider, MedicalServiceProvider, Traveler, SimplyGo, and SITA. The relationships of individual bounded contexts become visible via the strategic DDD notation (Vernon, 2016, Ch. 3). The MarketEnvironment stands above all other components. It creates the situational context to which the entire service ecosystem is subordinated. The four actors (immigration, service provider, traveler, medical provider) are not directly connected. Instead, SimplyGo serves as an intermediary of interaction between these groups.

[4] The GitHub repository contains the visiualizations shown in App. 6-15 plus the code that ContextMapper converted into the diagrams. In addition, it contains a high quality image that combines all the limited contexts in a detailed view.

Fig. 16: ContextMap of the SimplyGo Service System



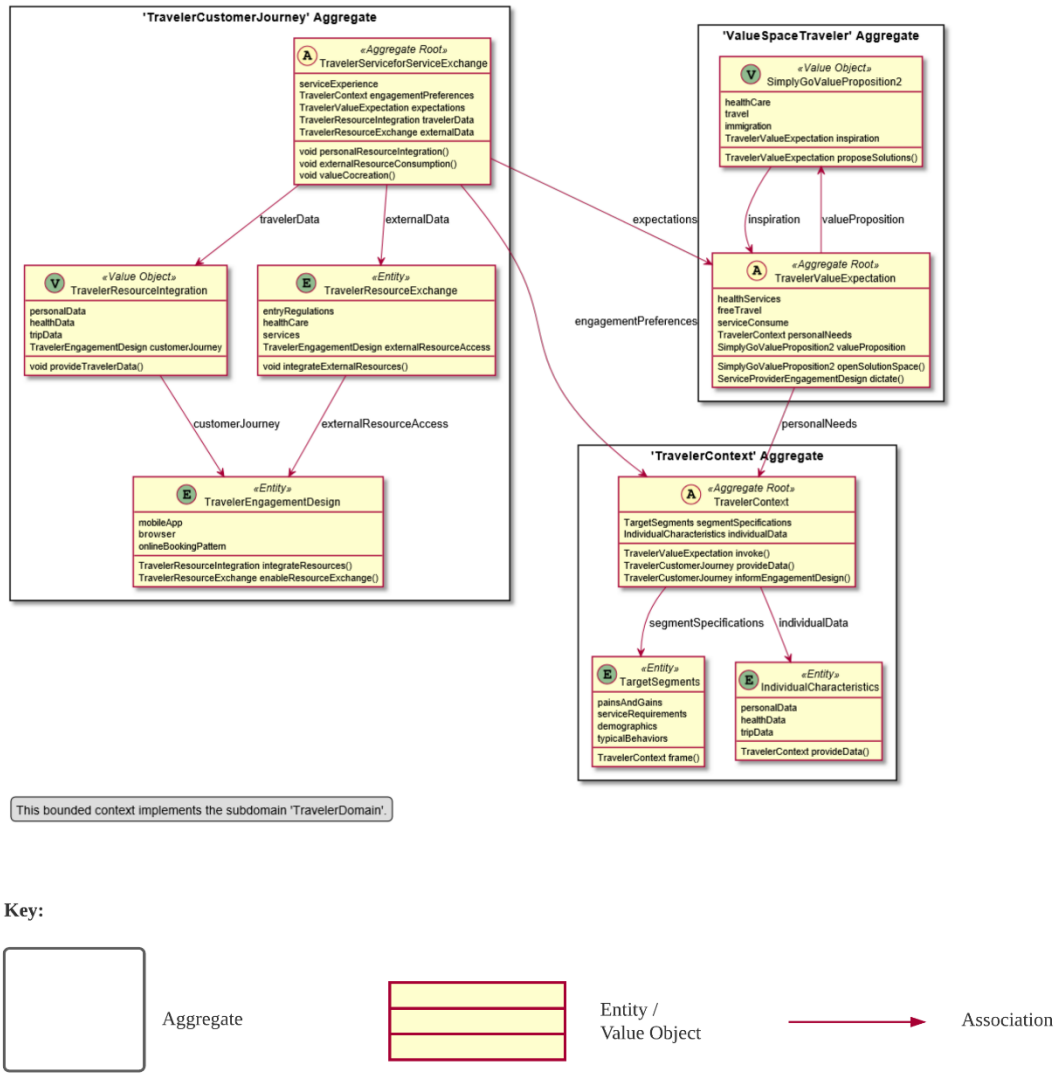
Own work.

Each of the actors has a shared kernel with SimplyGo. In strategic DDD, the shared kernel highlights that both bounded contexts access the same code, at least in parts (Vernon, 2016, Ch. 3). At SimplyGo, actors connect with the platform via interfaces that they can access and the engagement platform provides. Such a setup is in line with the concept of SDA's SOI. SimplyGo also has a partner involved in its activities. Through SITA, the platform distributes passenger data to immigration authorities. Here, communication is one-way only, with SimplyGo being the supplier and SITA being the customer of data distribution. The same accounts for the relationship between SITA and the immigration authorities. In terms of SDA concepts, this connection is most like the SOP because SITA's competencies increase the utility of passengers and immigration authorities. However, the latter is still linked to the platform directly and can, for example, use it to publicize information on current entry regulations (see Fig. 16).

In addition to this holistic view of SimplyGo, the framework provides a detailed picture of each bounded context. The bounded context of the traveler exemplifies this perspective of the framework. Fig. 17 shows three aggregates: The TravelerContext describes the individual properties, characteristics, and needs that make up the actor's unique context. The ValueSpaceTraveler contains aspects that form the basis of the actor's value experience within the service ecosystem. Both mentioned aggregates inform the TravelerCustomerJourney. It describes how service-for-service exchange and thus value cocreation occurs in the traveler's

realm. Within each aggregate are individual objects - entities and value objects - that make up their respective aggregate. UML class diagrams highlight their attributes and methods.

Fig. 17: Tactical DDD View of Traveler's Bounded Context



Own work.

For example, the individual characteristics of the traveler and the characteristics of the general customer group to which he belongs influence the TravelerContext (see Fig. 17). Personal data and segment-specific factors result in individual needs in the context of "travel" that the traveler needs to be solved and preferences for how the actor wants to integrate with SimplyGo. The TravelerContext influences both the value space of the traveler and the customer journey. The traveler's expectations towards SimplyGo and SimplyGo's value propositions inform each other in the value space. The TravelerCustomerJourney addresses the traveler’s expectations and preferences and guides through the processes of resource integration and exchange toward a personal experience of value-in-use. This customer journey

takes place on the communication layer of SimplyGo (see App. 11) and thus represents the shared kernel of the traveler with SimplyGo.

The presented bounded-context view can lay the foundation for the successful design and operation of service systems. It does this by accomplishing two things. First, it makes the process of value cocreation comprehensible. The framework gives a holistic view of how the different actors cocreate value, showing the components and the occurring social mechanisms residing on the multiple levels as described by Storbacka et al. (2016). Second, it points to the design parameters of SimplyGo's service systems to help practitioners identify similar design levers in their contexts:

Multilevel perspective on value cocreation

In combination with the SimplyGo bounded context (see App. 11), each bounded context view provides a holistic perspective on how value cocreation occurs for the respective actor. The multilevel view on value cocreation (Storbacka et al., 2016) becomes comprehensible because the framework shows the components of each bounded context and how they interact and thus influence each other.

For example, the MarketEnvironment (see App. 7) generally influences how the service ecosystem can enable value cocreation for the traveler. Aspects such as social or technological trends influence the preferences, beliefs, etc., of a customer segment to which an individual traveler can be assigned. Together with the customer's characteristics, this results in implications for the value space of the traveler and the customer journey. On the one hand, the general circumstances of the traveler lead to the derivation of expectations to SimplyGo's service offering. On the other hand, the traveler characteristics lead to preferences for interacting with SimplyGo via the SOI (e.g., design of the user interface, creation of activity patterns on the interface, etc.). The TravelerEngagementDesign regards both these factors and provides a customer-specific engagement design to improve the customer's experience on the platform. Such a view highlights how macro-level situational contexts influence the design of the engagement platform on the meso-level. The described mechanisms are situational mechanisms at work. How Company B designs the SOI of SimplyGo and thus the way how a traveler can interact with the platform determines the actor's disposition to engage on the micro-level.

The TravelerCustomerJourneyAggregate then provides first a view on the components and mechanisms involved in AE (see Fig. 17). Once SimplyGo's value proposition is convincing, the traveler will integrate resources such as personal data, health data, or trip data through the designed paths of the engagement design. The TravelerEngagementDesign, therefore, transforms an actor's disposition to engage into action in the form of data integration (i.e., action-formation mechanisms). Observable engagement properties become visible as to how the traveler uses the platform. Different metrics could be available, such as frequency of usage, how long it takes to check immigration requirements, etc.

A second view the TravelerCustomerJourney provides is the transition from the micro-level back to the meso-level in the process of value cocreation. The aggregate also represents how a traveler can access other actor's data through the traveler-specific interface on the platform (see Fig. 17). The TravelerEngagementDesign of the interface also provides the functionalities to access data such as COVID-19 test results, immigration regulations, and more. By providing personal data to other actors, the traveler helps them to provide customized information in turn. As described during the case study (chapter 3), SimplyGo acts as the intermediary for service-for-service exchange amongst the ecosystem's actors. It stabilizes the processes of this exchange and ensures the experience of value. Hence, what can be seen here in the framework are transformational mechanisms.

This last observation about the role of SimplyGo in providing the platform for service-for-service exchange holds another element. Like the restaurant in the example in 1.2.2.2, SimplyGo is the venue of interaction between different actors. Multiple aspects of this venue influence and govern this interaction: The described institutions, the value proposition, and, most importantly, the setup of the TravelerCustomerJourney on the SOI of SimplyGo. All these components together form the resource integration pattern of the traveler. The framework makes these patterns visible for the different actors without the need to specify its terms. Highlighting the interrelationships of the patterns' components shows service system designers the levers to detect, adapt, and improve them.

Guidance for design and operations

The framework uses tactical DDD to show the bounded contexts' ubiquitous languages, which consist of what will be called influential and functional elements in the following. The entities and value objects inside the aggregates in Fig. 17 represent these elements. Both types of elements make their respective bounded context functional together. As it identifies these influential and functional elements at SimplyGo, the framework also guides the identification, design, and re-design of such elements in other contexts.

Influential elements indicate that service system designers must consider their nature and the effects they cause in the design of functional elements. For example, at SimplyGo, the TravelerContext and thus both the IndividualCharacteristics and TargetSegments influence the TravelerEngagementDesign. Therefore, company B must identify these functional elements and their effects before designing the platform accordingly. Several tools are available to determine different customer segments and ways to address them with compelling value propositions and customer journeys. App. 16 lists some of them.

Under the influence of the influential elements is the design and thus the function of the functional elements. In Fig. 17, for example, these are the TravelerEngagementDesign or the TravelerCustomerJourney as a whole. As discussed before, these two elements are part of SimplyGo's SOI and thus the traveler's resource integration pattern. Their design directly

affects the value experience for the traveler and must therefore respond to the general institutional frame of the ecosystem and the implications of the influential elements.

From this point, the framework shows the link between the multilevel view of value cocreation (Storbacka et al., 2016), the multilevel design process (Grotherr et al., 2018), and the SDA (Weiß et al., 2018). The framework identifies the elements and mechanisms necessary for value cocreation, enabling the purposeful design of just that. From the perspective of the traveler, immigration authority, service provider, and medical service provider, the framework presents the influential and functional elements to consider when connecting the relevant stakeholders to SimplyGo's platform via the SOI. Service system designers who use the proposed framework to guide the design of their service system will be made aware of the elements to consider that are necessary for this process to run effectively according to the multilevel approach of value cocreation.

In addition, the framework integrates basic ideas of the multilevel design process. Due to its two-part approach, it considers both the business and the IT architecture (Grotherr et al., 2018). The presented views on the individual bounded contexts do the same: The value space of the respective actor follows the general situational context. These two then influence the design of the platform. Additionally, the choice of DDD as the modeling form highlights the modularity required for this development process.

Finally, the framework accomplishes something else: it enables emergence within the service system. Since the influential elements inside the traveler's bounded context are dynamic, they necessitate design-related adaptations of the TravelerCustomerJourney over time to keep resource integration and value cocreation stable. By optimizing the traveler's distinct resource integration patterns on a systemic level, new structures within the traveler's bounded context arise, changing the traveler's ways of interacting with SimplyGo and other actors and thus how value is cocreated. Thus, on a basic level, emergence in the framework can be described as the constant, iterative adaptation of the SOI to changing influential elements. Because these adaptations are the result of several influencing factors, stemming from the traveler's realm, the general environment of SimplyGo and finally SimplyGo's own properties, they cannot be reduced to their individual ingredients, although they are depended on these aforementioned factors. As a consequence, what can be observed when the traveler interacts with SimplyGo, is heteropathic resource integration at work. In an optimal world, this continuous adaption continuously optimizes how the traveler experiences value in the service system and thus is the central part of service innovation at SimplyGo.

4.2 Discussion of Findings

The development of the proposed framework answers the research question: Decomposing abstract service innovation and design theory and linking it to a real-life case based on SDA research allows understanding the composition and functioning of complex service systems.

This understanding enables practitioners to transfer design knowledge and apply it to the design of appropriate SoE in their contexts.

The framework presented in 4.1.2 is the result of mapping abstract service theory, like the SDL (Vargo & Lusch, 2004), service innovation (Koskela-Huotari et al., 2016), or the multilevel design process for designing service systems (Grotherr et al., 2018) onto a practical example from SDA research (Weiß et al., 2021). The format of the final framework is that of a holistic view of the service system of SimplyGo, providing information about the actors inside the platform's service ecosystem, the service platform itself, and how value cocreation occurs (Lusch & Nambisan, 2015). At the beginning of the research project, this format had not been envisioned. Instead, the initial idea was to refine the design process of Grotherr et al. (2018) with elements of the SDA, but without touching its processual format.

However, this approach did not seem to provide much value to the practical audience of this research report, not to speak of achieving the set research purpose and goal. This is because it did not allow to leave the theoretical realm but necessitated to continue using abstract logic and vocabulary of service science during the development. The final format achieves this. DDD as the basis of the framework allowed to resolve the SimplyGo case into its building blocks. This solution exposes their characteristics and interrelationships with other components in a language known by a broader audience.

To not follow Grotherr et al.'s processual approach was a purposely made decision. The philosophy of the proposed solution is to empower practitioners to holistically implement a service-driven business model. With the knowledge about DT in mind, this approach is more promising in the long term. As discussed in chapter 1, the IT architecture is responsible of implementing IT infrastructure capabilities required to run service-driven business models (Weill et al., 2002; Weiß et al., 2019). So, before practitioners design and build SoE to create new dynamic capabilities, they first need to design a service-driven business architecture. The framework contains the knowledge necessary to help experts accomplish both tasks. It describes both the business logic of SimplyGo by showing the arrangement of actors and resources and the system logic by showing the interrelationships and mechanisms of the system. The framework also shows specific configurations of parts of a service platform, such as the interface between the actor and the platform and how external influences affect their design. Thus, the framework first supports practitioners in designing a new service-oriented enterprise architecture. It then guides the subsequent design and implementation of the required IT infrastructure.

In order to demonstrate that the framework can fulfill this purpose, a reference is made again to SimplyGo in the following. Based on the case study and the publicly available documents on SimplyGo, individual functions of the platform can be identified (see Tab. 8). Taking the "Travel Planning" function, for example, the framework helps to create a first implementable draft of a corresponding IT solution. App. 13-15 hold three UML diagrams showing such a solution.

Table 8: Service Functions of SimplyGo

Use-Case / Service Function	**Description**
Travel Planning	Travel with peace of mind: Plan your trip and view the most important entry test requirements for your destination.
Corona Regulations	Get informed about the latest corona-related behavior rules and news which could impact your trip.
Find Corona Test	Find a lab and get tested: Quickly find medical providers in your surrounding area who offer required tests.
Booking Test	Book an appointment, conduct a test and receive the results directly via the app.
Share Test Results	View and share test results: Access your test results in your health record, share them with airlines and immigrations to receive permission to enter your destination.

Source: Weiß et al. (2021, p. 278).

The sequence diagram (Fowler, 2003, p. 53) and the activity diagram (Fowler, 2003, p. 117) in App. 13 & 14 show how the service works. A traveler files a new request to SimplyGo for the test requirements when entering the destination country. The platform forwards the request to the respective immigration authority, which returns the appropriate requirements for the individual traveler based on health status. The health data necessary for this decision is provided by the medical service provider, to which the authority files a separate request. For example, if the traveler is vaccinated, no test requirements are applicable. In another case, a negative PCR test could be required.

App. 15 then shows the class diagram (Fowler, 2003, p. 35) of the respective IT architecture that assists in the technical implementation of the service. In the class diagram, three aspects become apparent: First, the service consists mainly of requests and responses. Second, SimplyGo itself is designed as a proxy, i.e., a central intermediary responsible for forwarding messages to the respective recipients (i.e., resource orchestration) without the need to neither know them nor the content of the messages. This design is consistent with SimplyGo's actual setup as an intermediary of end-to-end encrypted messages (Element 4, App. 5).

Third, the class diagrams only contain attributes to show the requests' and returns' structure within the actors' context. For example, the traveler's request consists of a unique ID, a request type, and all the data of the requested trip (see class "Trip"), including the traveler's data. The immigration authority sends the traveler's data and a verification of its own identity to the appropriate medical service provider. Based on the identification of the traveler, the medical service provider returns the respective health status. The response of the authority contains then the appropriate test requirements. Only the SimplyGo class includes methods, as it is the proxy.

Assuming business and IT experts had created this design in a collaborative effort, it would be informative for the subsequent development of an initial prototype. For one, it communicates the design of the platform's core function. Second, it points towards the design of the interfaces for each actor, i.e., how they can integrate what type of resources. At the same time, it acknowledges that actors like the medical service provider or the immigration authorities have their own logic of processing data. SimplyGo does not want to interfere with these but exclusively aims at providing a convenient communication interface (Element 4, App. 5).

This short demonstration pictures how the framework helps to design a business architecture and subsequently appropriate IT solutions. It points towards the design elements inside the SimplyGo service ecosystem and the available design elements following the service function. Software engineers can also double-check using the framework whether they considered all relevant influential and functional elements in their implementation. The same logic can be applied when the solution should be expanded with the additional functions shown in Tab. 8. In addition, the framework assists in later iterations of refining the evolving IT architecture. Practitioners can quickly identify the exact module in which the EA needs to be adapted without overhauling it entirely.

The framework's capabilities to the service systems development process can be valuable for developing new dynamic capabilities. By facilitating the targeted adaptation or evolution of individual building blocks of the platform, as just argued, development teams in both institutional design and engagement design can respond quickly to changes around or in the service ecosystem. Staying in constant contact with stakeholders can reduce the time it takes to make adjustments to systems and open up new ways to deliver value to stakeholders more quickly. This creates a significant effect: the behavior of the actors in the service ecosystem contributes significantly to the further development of the service systems. The service offering is not developed top-down by the platform operator. Instead, adjustments take place bottom-up. The results are more customer centricity, speed, and ultimately innovation.

But the framework does not come without some limitations. For one, practitioners may find it challenging to transfer the knowledge about SimplyGo onto their applications. This is a necessary evil that lies within the solution. That is because, as said in objective one for the solution (see 4.1.1), the abstract theory needed to be left behind to foster a broader understanding of the implications and added value of the service theory in practical environments. This also justifies the second limitation: the framework is limited in using scientific language for clarity. Letting go of abstract terms was another necessity to make the concepts for building SoE accessible to a non-scientific audience. A third limitation is that the framework does not represent the setup of SimplyGo to an accuracy of 100% since it bases on the case study's insights. These came from informal conversations and publicly accessible documents. But such a level of accuracy is not needed. Again, the framework's important work is to highlight critical components of a functioning service system and how they relate to each other. A fourth limitation is that the framework solely focuses on the design of SoE. Ross et al. (2019,

Ch. 3) argue that a highly functional operational backbone (i.e., SoR) is equally essential for running a service business, as the SoE rely on its capabilities. But for the sake of focus on building SoE, the SoR have been neglected here.

Perhaps the most severe limitation to the framework could be that it leaves the processual format of Grotherr et al.'s framework (2018) behind. But the case study has revealed that it is more critical to both business experts and software engineers to have a clear vision in front of them that helps them coordinate their work. Both Company A and B do have this vision, in more or fewer parts provided by the SDA. The ways how they take their journey to reach this vision differ but are both successful. Therefore, it can be argued that by providing a holistic view of service system functioning, the framework offers more value to practitioners than a processual format could. It leaves room for individual solutions, which is good since every organizational context is different and requires context-specific solutions to context-specific problems. The framework assists in identifying them both without forcing its users into a predefined scheme that would probably be incompatible with their demands at some point of time during design and development.

Having discussed the framework's value and potential limitations, the remaining question is whether the framework successfully achieved the set objectives before the development (4.1.1). The discussion two paragraphs above has already addressed the first objective two paragraphs above. Using SimplyGo as a showcase allowed to leave the scientific realm of service science and enter a space that allowed to make the essential concepts comprehensive to a non-expert audience.

Different aspects contributed to achieving the second objective: Regarding the business aspect, leaving the processual view of Grotherr et al. (2018) fulfilled the framework's requirement for a holistic solution (see discussion in 4.1.2). Also, emergence and resource integration patterns have become more evident. On the IT side, DDD highlights the role of modularity in the design of service systems. Whether practitioners decide to implement such a microservice architecture for their SoE is not the central question here. Instead, the framework raises the importance of designing these systems with enough decoupling of individual components to allow for targeted adaptations where they are needed, without touching the entire architecture's design.

5 CONCLUSION

This research aimed to develop a multilevel design framework for designing and operating effective SoE. Using AE as a microfoundation of value cocreation, the framework should overcome the abstractness and complexity of previous scientific works to provide practitioners with a comprehensive guide to build such systems using the SDA. By analyzing real-life cases related to SDA research and drawing from a multilevel design process for designing service systems (Grotherr et al., 2018), the research project led to developing a framework

that makes the structure and functions of an SDL-inspired service system visible. In doing so, the framework proposed in this research report can educate practitioners about the essential building blocks of a functional service system. It empowers them to embark on a journey to design and continuously improve service-driven business architectures and respective SoE that implement the dynamic capabilities necessary to run the envisioned service system.

The development of the SDA is an ongoing research project. It is updated in regular iterations by the latest findings from service science. In addition, researchers ensure the applicability of the SDA through a longitudinal case study that serves to test its latest versions and gain additional insights that feed back into theoretical development (Weiß et al., 2019, p. 3). SDA research aims to make the abstract theory of SDL and service innovation applicable in business environments. For this purpose, it started as an architectural blueprint that conceptualized the SoE as a new systems layer on top of the existing SoR. These SoE implement the functional components of service innovation into a firm's EA (Warg et al., 2015).

Around the same time SDA development took off, Storbacka et al. (2016) published their multilevel approach to value cocreation, in which they successfully highlighted its components and related mechanisms. Especially their description of the three social mechanisms and AE as the microfoundation of value cocreation is a significant contribution to understanding the nature of the phenomenon. Grotherr et al. (2018) then took up this multilevel approach to conceptualize a design process for developing socio-technical systems and operating service business models. What is unique about their method is the holistic approach to building service systems. Drawing from the multilevel perspective of Storbacka et al. (2016), their design process addresses both a firm's business architecture with its institutional design cycle and the IT architecture with its engagement design cycle. However, because the process continues to rely on theoretical concepts and does not include a target image, its practical usefulness remains limited.

Although lacking practical use, the two works mentioned above opened the opportunity to expand the body of knowledge of SDA. The SDA contains what is missing from the multilevel design process: the target image of an SDL-inspired IT infrastructure. Consequently, the idea arose to extend the SDA with the approaches of Storbacka et al. and Grotherr et al. and turn it into a "design environment to design and operate service systems" (Weiß, 2020, p. 119). This is where the presented research comes in. A DSR project should help inform the further development of SDA in the proposed direction. Thus, the first step was to conduct a critical literature review to survey the current knowledge in relevant service science. Then, the case study served to examine the current state of the SDA and its performance in practice. Following DSR theory, these two strands of work provided the rigor and relevance needed to develop the framework presented (Hevner, 2007).

The case study examined a practical implementation of SDA in two companies, A and B. The interviews conducted with decision-makers of the two companies allowed to verify the theoretical propositions previously established through practical insights. It came out that

the approaches mediated by the SDA are a basis for the implementation of digital strategies in the observed companies. Company A emerges directly from the SDA research and, therefore, fully implements the SDA. Company B has been inspired by the principles of SDA in building its own service platform. Both firms run successful service business models thanks to their SDL-inspired IT infrastructures. Additionally, SDA's technological approach also enabled the companies studied to develop quantitative and qualitative tools to examine their service quality.

In contrast, however, the case study also produced insights into areas where the SDA needs to develop. For one, the SDA was an essential tool for the companies observed in understanding and adopting SDL concepts. At the same time, however, the customers of Company A in particular still have great difficulty fully absorbing the concepts of service-driven business models. Here, SDA has only provided indirect support by providing Company A with a narrative, for example, in the form of use cases. Secondly, the case study revealed that SDA cannot yet guide practitioners outside its research realm in designing, building, and operating their own service systems.

These case study findings and the theoretical knowledge gained from the critical literature review formed the starting point for developing the presented framework. From the refuted theoretical propositions of the case study, criteria for developing the framework were derived. Thus, on the one hand, it should help reduce the abstraction of the SDA further and make it available to a broad audience. On the other hand, it should provide a holistic guide to the design and operation of a service system (as proposed by Grotherr et al. (2018)). The result of the development process is a framework that represents SimplyGo's service system in its structure and function. It identifies components and mechanisms within a working, real-world use case inspired by the SDA. The framework contains elements of both business architecture and IT architecture. For communication, it uses language modules of DDD, which achieves several effects. First, DDD is a language understood by both business and IT experts. Through it, the framework emphasizes the need for interdisciplinary work. Second, the DDD format indexes the need for a modular structure of the service system.

It must be mentioned that the proposed solution is not free from bias. The development work could not wholly exclude bias because it found itself in an environment familiar with general service theory, being Companies A and B. However, this was not detrimental in the current stage of SDA development. On the contrary. The case study has contributed significantly to the success of the research project.

This research project and its outcome are relevant for both a practical and a scientific audience. The case study highlights how two digital firms operate their service-oriented business models to decision-makers such as general managers, product managers, or IT management staff. It provides them with insights into different business models, prevailing corporate philosophies (e.g., DDD as a general orientation in everyday work in Company A), approaches

to software development and refinement, and challenges the two firms face in their operation. The framework that is the focus of this paper then conveys the operation of an SDL-inspired service system built according to SDA concepts. It manages to highlight the necessary components of a functional service system and the occurring mechanisms involved in value cocreation on the three levels of abstraction without relying on complex, incomprehensive SDL language. In addition, it contains both elements from the business architecture of a firm (e.g., actor-centric value propositions) and the IT architecture. As a result, the framework empowers practitioners to shed light on the complexity of service-oriented EAs and transfer newly gained knowledge onto their own applications. This transfer of knowledge from a real use case is a significant practical contribution to help readers design and subsequently execute their digital strategies to master DT.

Scientific audiences benefit from the findings of this work, too. First, the case study results allow SDA researchers to assess the current state of their ongoing single-case study that Company A is (Warg et al., 2019). Researchers get impressions of how practitioners perceive the SDA and how it helps them in their operation. The insights of examining Company B further propel the current research stream of combining the SDA with the multilevel view on value cocreation (Weiß, Kronibus, Riedel, & Rittweger, 2021). Second, the framework has been the first attempt at such a combination. Therefore, it will further inform SDA research in its current mission of developing the SDA in a design environment for designing and subsequently operating service systems.

Outside of SDA research, the thesis also enriches service science. As the literature review has argued earlier, the issue of theoretical works around the SDL and service innovation is their lack of practical reference. This insufficiency hinders their perception by a broader audience. Such a reference is made with the presented framework, making theoretical concepts of the SDL and related works visible. For example, different design aspects and components have been identified as what the theory calls social mechanisms. The engagement paths on an engagement platform's user interface guide an actor in AE and thus act as action-formation mechanisms. Resource integration patterns in the form of the design of customer journeys determine the occurrence and quality of value cocreation, etc. Service scientists must make such visualizations more frequently to foster the broader application of theory in practice.

Finally, it is appropriate to make recommendations for the continuation of the presented research results. According to DSRM (Peffers et al., 2007), the next step is practically implementing the framework. This corresponds with the last research mile of Nunamaker (2015). To analyze the practicality of its newest artifact, SDA research must venture into unknown use cases. So far, practical implementation has taken place in research-related companies. For SDA to get closer to its goal, it must also be implemented, observed, and analyzed in settings outside this realm. The question for future research is whether the framework provides the practical benefits argued here. For the last mile of research, the artifact must be used in new contexts, and its performance must be observed in the process.

REFERENCE LIST

1. Abdel Razek, Rahman, A., Elstermann, M., Hengels, A., & van Husen, C. (2020). Multidimensionales Service Prototyping. In D. Beverungen, J. H. Schumann, V. Stich, & G. Strina (Eds.), *Dienstleistungsinnovationen durch Digitalisierung: Band 1: Geschäftsmodelle – Methoden – Umsetzungsbeispiele* (pp. 159-225). Berlin, Heidelberg: Springer Berlin Heidelberg.
2. Agarwal, R., Johnson, S. L., & Lucas, H. C. (2011). Leadership in the Face of Technological Discontinuities: The Transformation of EarthColor. *Communications of the Association for Information Systems, 29*, 33. doi:https://doi.org/10.17705/1CAIS.02933
3. Ahlemann, F., Legner, C., & Schäfczuk, D. (2012). Introduction. In F. Ahlemann, E. Stettiner, M. Messerschmidt, & C. Legner (Eds.), *Strategic Enterprise Architecture Management: Challenges, Best Practices, and Future Developments* (pp. 1-33). Berlin, Heidelberg: Springer Berlin Heidelberg.
4. Asadullah, A., Faik, I., & Kankanhalli, A. (2018). *Digital Platforms: A Review and Future Directions.* Paper presented at the Twenty-Second Pacific Asia Conference on Information Systems, Yokohama, Japan.
5. Bharadwaj, A., El Sawy, O. A., Pavlou, P. A., & Venkatraman, N. (2013). Digital Business Strategy: Toward a Next Generation of Insights. *MIS Quarterly, 37*(2), 471-482.
6. BMWi. (2021). *Taking control of the digital transformation*. Retrieved on 12.03.2021 from https://www.bmwi.de/Redaktion/EN/Dossier/digitisation.html
7. Bonnet, D. & Westerman, G. (2021). The New Elements of Digital Transformation. *MIT Sloan Management Review, 62*(2), 82-89.
8. Breidbach, C. F., Brodie, R., & Hollebeek, L. (2014). Beyond virtuality: from engagement platforms to engagement ecosystems. *Managing service quality, 24*(6), 592-611. doi: https://doi.org/10.1108/MSQ-08-2013-0158
9. Brodie, R. J., Fehrer, J. A., Jaakkola, E., & Conduit, J. (2019). Actor Engagement in Networks: Defining the Conceptual Domain. *Journal of Service Research, 22*(2), 173-188. doi: https://doi.org/10.1177/1094670519827385
10. Brodie, R. J., Saren, M., & Pels, J. (2011). Theorizing about the service-dominant logic: The bridging role of middle range theory. *Marketing Theory, 11*(1), 75-91. doi: https://doi.org/10.1177/1470593110393714
11. Brust, L., Breidbach, C., Antons, D., & Salge, O. (2017). *Service-Dominant Logic and Information Systems Research: A Review and Analysis Using Topic Modeling*.
12. Böhmann, T., Leimeister, J. M., & Möslein, K. (2014). Service Systems Engineering. *Business & Information Systems Engineering, 6*(2), 73-79. doi: https://doi.org/10.1007/s12599-014-0314-8
13. Chanias, S., Myers, M. D., & Hess, T. (2019). Digital transformation strategy making in pre-digital organizations: The case of a financial services provider. *The Journal of*

Strategic Information Systems, 28(1), 17-33. doi:https://doi.org/10.1016/j.jsis.2018.11.003
14. Coleman, J. S. (1994). *Foundations of Social Theory*. Cambridge, MA, USA: Harvard University Press.
15. Corsaro, D., Cantù, C., & Tunisini, A. (2012). Actors' Heterogeneity in Innovation Networks. *Industrial Marketing Management, 41*(5), 780-789. doi:https://doi.org/10.1016/j.indmarman.2012.06.005
16. DIN. (2019-09). DIN SPEC 33453. In *Entwicklung digitaler Dienstleistungssysteme*. Berlin, GER: Beuth Verlag GmbH.
17. Eccles, R. G., & Nohria, N. (1992). *Beyond the Hype: Rediscovering the Essence of Management* (Vol. 2003 Reprint). Washington D.C, USA: Beard Books.
18. Edvardsson, B., Skalen, P., & Tronvoll, B. (2012). Service Systems as a Foundation for Service Integration and Value Co-Creation. In S. L. Vargo, R. F. Lusch, Malhotra, & N. K. (Eds.), *Toward a Better Understanding of the Role of Value in Markets and Marketing* (pp. 79-126). ProQuest Ebook Central: Emerald Publishing Limited.
19. Evans, E. (2003). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Boston, MA, USA: Addison Wesley.
20. Felin, T., Foss, N. J., & Ployhart, R. E. (2015). The Microfoundations Movement in Strategy and Organization Theory. *Academy of Management Annals, 9*(1), 575-632. doi: https://doi.org/10.1080/19416520.2015.1007651
21. Fowler, M. (2003). *UML Distilled - A Brief Guide to the Standard Modeling Language*. Boston, MA, USA: Addison Wesley.
22. Fraunhofer IPT. (2021). Digital Transformation. Retrieved on 12.03.2021 from https://www.ipt.fraunhofer.de/en/Competencies/Technologymanagement/digital-transformation.html
23. Frow, P., & Payne, A. (2019). Value Cocreation: An Ecosystem Perspective. In S. L. Vargo & R. F. Lusch (Eds.), *The SAGE Handbook of Service-Dominant Logic* (pp. 80-96). Thousand Oaks, CA, USA: SAGE Publications Inc.
24. Gebauer, H., Fleisch, E., Lamprecht, C., & Wortmann, F. (2020). Growth paths for overcoming the digitalization paradox. *Business Horizons, 63*(3), 313-323. doi:https://doi.org/10.1016/j.bushor.2020.01.005
25. Giddens, A. (1984). The Constitution of Society - Outline of the Theory of Structuration. In. Cambridge, UK.: Polity Press.
26. Giddens, A. (1993). *New Rules of Sociological Method* (2 ed.). Cambridge, UK: Polity Press.
27. Gong, C., & Ribiere, V. (2021). Developing a unified definition of digital transformation. *Technovation, 102*, 102217. doi:https://doi.org/10.1016/j.technovation.2020.102217
28. Grotherr, C., Li, M. M., Schymanietz, M., Fritzsche, A., Semmann, M., Peters, C., . . . Möslein, K. M. (2020). Gestaltungsdimensionen der Digitalisierung – Wie Dienstleistungssysteme den Wandel antreiben und welche Faktoren eine Rolle spielen. In D. Beverungen, J. H. Schumann, V. Stich, & G. Strina (Eds.),

Dienstleistungsinnovationen durch Digitalisierung: Band 1: Geschäftsmodelle – Methoden – Umsetzungsbeispiele (pp. 277-325). Berlin, Heidelberg: Springer Berlin Heidelberg.

29. Grotherr, C., Semmann, M., & Böhmann, T. (2018). *Using Microfoundations of Value Co-Creation to Guide Service Systems Design – A Multilevel Design Framework*. Paper presented at the Thirty ninth International Conference on Information Systems,, San Francisco.
30. Grönroos, C., & Voima, P. (2013). Critical Service Logic: Making Sense of Value Creation and Co-Creation. *Journal of the Academy of Marketing Science, 41*, 133-150. doi: https://doi.org/10.1007/s11747-012-0308-3
31. Hauser, J., Tellis, G. J., & Griffin, A. (2006). Research on Innovation: A Review and Agenda for Marketing Science. *Marketing Science, 25*(6), 687-717.
32. Hedström, P., & Swedberg, R. (1998). Social Mechanisms: An Introductory Essay. In P. Hedström (Ed.), *Social Mechanisms - An Analytical Approach to Social Theory* (pp. 1-31). Cambridge: Cambridge University Press.
33. Hein, A., Schreieck, M., Riasanow, T., Setzke, D. S., Wiesche, M., Böhm, M., & Krcmar, H. (2020). Digital platform ecosystems. *Electronic Markets, 30*(1), 87-98. doi: https://doi.org/10.1007/s12525-019-00377-4
34. Henderson, J. C., & Venkatraman, H. (1993). Strategic alignment: Leveraging information technology for transforming organizations. *IBM Systems Journal, 32*(1), 472-484. doi: https://doi.org/10.1147/sj.382.0472
35. Hess, T., Benlian, A., Matt, C., & Wiesböck, F. (2016). Options for formulating a digital transformation strategy. *MIS Quarterly Executive, 15*(2), 123-139.
36. Hevner, A. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems, 19*(2), 87-92.
37. Hevner, A., March, S. T., & Park, J. (2004). Design Science in Information Systems Research. *Management Information Systems Quarterly, 28*, 75.
38. IfSD Hamburg. (2020). Plattformen und Ökosysteme als Chance zur Befähigung. Retrieved from https://www.ifsd.hamburg/.cm4all/uproc.php/0/Pra%CC%88sentation%20SDA-IfSD.pdf?_=17164cd3008&cdp=a&cm_odfile
39. ION Access & Health GmbH. (2021a). A secure and standardized platform for entry regulation. Retrieved on 14.04.2021 from https://ion-ah.com/work-with-us/simplygo-platform
40. ION Access & Health GmbH. (2021b). Travel app SimplyGo and SITA complete successful trial of COVID-19-focused health data platform [Press release]. Retrieved from https://uploads-ssl.webflow.com/5f5f45ba968dd38be82f0851/601af4d361da053fef740fc0_210203_SimplyGo%20EN.pdf
41. Kane, G. C., Palmer, D., Nguyen Phillips, A., Kiron, D., & Buckley, N. (2015). *Strategy, not Technology Drives Digital Transformation*. Retrieved from

42. Koskela-Huotari, K., Edvardsson, B., Jonas, J. M., Sörhammar, D., & Witell, L. (2016). Innovation in service ecosystems—Breaking, making, and maintaining institutionalized rules of resource integration. *Journal of Business Research, 69*(8), 2964-2971. doi:https://doi.org/10.1016/j.jbusres.2016.02.029
43. Lusch, R. F., & Nambisan, S. (2015). Service Innovation - A Service-Dominant Logic Perspective. *MIS Quarterly, 39*(1), 155-176. doi: https://doi.org/10.2307/26628345
44. Lusch, R. F., & Vargo, S. L. (2014). *Service-Dominant Logic*. Cambridge, UK: Cambridge University Press.
45. Lusch, R. F., & Vargo, S. L. (2019). An Overview of Service-Dominant Logic. In S. L. Vargo & R. F. Lusch (Eds.), *The SAGE Handbook of Service-Dominant Logic* (pp. 3-21). Thousand Oaks, CA, USA: SAGE Publications Inc.
46. Maglio, P. P., & Spohrer, J. (2008). Fundamentals of service science. *Journal of the Academy of Marketing Science, 36*(1), 18-20. doi:10.1007/s11747-007-0058-9
47. Maglio, P. P., Vargo, S. L., Caswell, N., & Spohrer, J. (2009). The service system is the basic abstraction of service science. *Information Systems and e-Business Management, 7*(4), 395-406. doi: https://doi.org/10.1007/s10257-008-0105-1
48. Matt, C., Hess, T., & Benlian, A. (2015). Digital Transformation Strategies. *Business & Information Systems Engineering, 57*(5), 339-343. doi: https://doi.org/10.1007/s12599-015-0401-5
49. Meynhardt, T., Chandler, J. D., & Strathoff, P. (2016). Systemic principles of value co-creation: Synergetics of value and service ecosystems. *Journal of Business Research, 69*(8), 2981-2989. doi:https://doi.org/10.1016/j.jbusres.2016.02.031
50. Michel, S., Vargo, S., & Lusch, R. (2007). Reconfiguration of the conceptual landscape: A tribute to the service logic of Richard Normann. *Journal of the Academy of Marketing Science, 36*, 152-155. doi: https://doi.org/10.1007/s11747-007-0067-8
51. Moore, G. (2011). Systems of Engagement and the Future of Enterprise IT. Retrieved from https://info.aiim.org/systems-of-engagement-and-the-future-of-enterprise-it
52. Normann, R. (2001). *Reframing Business: When the Map Changes the Landscape*. Chichester, England: John Wiley & Sons.
53. Normann, R., & Ramirez, R. (1993). From value chain to value constellation: Designing interactive strategy. *Harvard Business Review, 71*(4), 65-77.
54. Nunamaker, J. F., Briggs, R., Derrick, D., & Schwabe, G. (2015). The Last Research Mile: Achieving Both Rigor and Relevance in Information Systems Research. *Journal of Management Information Systems, 32*, 10-47. doi: https://doi.org/10.1080/07421222.2015.1094961
55. Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems, 24*(3), 45-77. doi:https://doi.org/10.2753/MIS0742-1222240302

56. Peters, L. D. (2016). Heteropathic versus homopathic resource integration and value co-creation in service ecosystems. *Journal of Business Research, 69*(8), 2999-3007. doi:https://doi.org/10.1016/j.jbusres.2016.02.033
57. Pinho, N., Beirão, G., Patrício, L., & Raymond, P. F. (2014). Understanding value co-creation in complex services with many actors. *Journal of Service Management, 25*(4), 470-493. doi: https://doi.org/10.1108/JOSM-02-2014-0055
58. Proper, H. (2014). *Enterprise Architecture: Informed Steering of Enterprises in Motion.* Paper presented at the International Conference on Enterprise Information Systems, Angers, France. doi: https://doi.org/10.1007/978-3-319-09492-2_2
59. Ramaswamy, V. (2009). Leading the transformation to co-creation of value. *Strategy & Leadership, 37*(2), 32-37. doi:https://doi.org/10.1108/10878570910941208
60. Ramaswamy, V., & Gouillart, F. (2010). Building the Co-Creative Enterprise. *Harvard business review, 88*(10), 100-109, 150.
61. Red Hat. (2021a). What is a data lake? *Storage.* Retrieved on 05.07.2021 from https://www.redhat.com/en/topics/data-storage/what-is-a-data-lake
62. Red Hat. (2021b). What is digital transformation? *Digital Transformation.* Retrieved on 12.03.2021 from https://www.redhat.com/en/topics/digital-transformation/what-is-digital-transformation
63. Red Hat. (2021c). What is orchestration? *Automation.* Retrieved on 05.07.2021 from https://www.redhat.com/en/topics/automation/what-is-orchestration
64. Richardson, C. (2021a). A pattern language for microservices. *A pattern language for microservices.* Retrieved on 25.08.2021 from https://microservices.io/patterns/decomposition/decompose-by-subdomain.html
65. Richardson, C. (2021b). Pattern: Decompose by subdomain. *A pattern language for microservices.* Retrieved on 11.08.2021 from https://microservices.io/patterns/decomposition/decompose-by-subdomain.html
66. Rittweger, R., Kronibus, A., & Weiß, P. (2020). *Value Co-creation as the Core of Service Innovation: Impacts of a Case Study of a Fully Digitized Health Insurance Company.* Paper presented at the Advances in the Human Side of Service Engineering, San Diego, CA, USA.
67. Ross, J. W., Beath, C. M., & Mocker, M. (2019). *Designed for Digital - How to Architect Your Business for Sustained Success* (P. Michelman Ed.). Cambridge, MA, USA: The MIT Press.
68. Ross, J. W., Sebastian, I., Beath, C., Mocker, M., Moloney, K., & Fonstad, N. O. (2016). *Designing and Executing Digital Strategies*. Paper presented at the Thirty Seventh International Conference on Information Systems, Dublin, Ireland.
69. Saunders, M., Lewis, P., & Thornhill, A. (2016). *Research Methods for Business Students* (7th ed.). Harlow, England: Pearson Education Limited.
70. Schwabe, G., & Krcmar, H. (2000). *Piloting Socio-Technical Innovation*. Paper presented at the 8th European Conference on Information Systems, Vienna, Austria.
71. Scott, W. R. (2013). *Institutions and Organizations: Ideas, Interests, and Identities* (4 ed.). Thousand Oaks, CA, USA: SAGE Publications.

72. SDA SE Open Industry Solutions. (2021a). Home. Retrieved from https://sda.se/
73. SDA SE Open Industry Solutions. (2021b). Über uns. Retrieved from https://sda.se/ueber-uns/
74. Sebastian, I. M., Moloney, K. G., Ross, J. W., Fonstad, N. O., Beath, C., & Mocker, M. (2017). How big old companies navigate digital transformation. *MIS Quarterly Executive, 16*(3), 197-213.
75. Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011). Action Design Research. *MIS Quarterly, 35*(1), 37-56. doi: https://doi.org/10.2307/23043488
76. Sinkovics, N. (2018). Pattern matching in qualitative analysis. In Catherine Cassell, Ann Cunliffe, & G. Grandy (Eds.), *The SAGE Handbook of Qualitative Business and Management Research Methods* (Vol. 2, pp. 468-485). Thousand Oaks, US: Sage Publications Ltd.
77. Soto Setzke, D., Riasanow, T., Böhm, M., & Krcmar, H. (2021). Pathways to Digital Service Innovation: The Role of Digital Transformation Strategies in Established Organizations. *Information Systems Frontiers*. doi: https://doi.org/10.1007/s10796-021-10112-0
78. Spohrer, J., & Maglio, P. P. (2008). The Emergence of Service Science: Toward Systematic Service Innovations to Accelerate Co-Creation of Value. *Production & Operations Management, 17*(3), 238-246. doi: https://doi.org/10.3401/poms.1080.0027
79. Storbacka, K., Brodie, R. J., Böhmann, T., Maglio, P. P., & Nenonen, S. (2016). Actor engagement as a microfoundation for value co-creation. *Journal of Business Research, 69*(8), 3008-3017. doi:https://doi.org/10.1016/j.jbusres.2016.02.034
80. Szabo, C., Teo, Y. M., & Chengleput, G. K. (2014). *Understanding complex systems: Using interaction as a measure of emergence*. Paper presented at the Winter Simulation Conference 2014, Savannah, GA, USA.
81. Teece, D., & Pisano, G. (1994). The Dynamic Capabilities of Firms: an Introduction. *Industrial and Corporate Change, 3*(3), 537-556. doi: https://doi.org/10.1093/icc/3.3.537-a
82. Tiwana, A., Konsynski, B., & Bush, A. (2010). Research Commentary —Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics. *Information Systems Research, 21*, 675-687. doi: https://doi.org/10.1287/isre.1100.0323
83. University of Applied Sciences of Eastern Switzerland. (2021). A Modeling Framework for Strategic Domain-driven Design. Retrieved from https://contextmapper.org/
84. Vargo, S. L., Koskela-Huotari, K., & Vink, J. (2020). Service-Dominant Logic: Foundations and Applications. In E. Bridges & K. Fowler (Eds.), *The Routledge Handbook of Service Research Insights and Ideas* (pp. 3-23). New York, USA: Routledge.

85. Vargo, S. L., & Lusch, R. F. (2004). Evolving to a New Dominant Logic for Marketing. *Journal of Marketing, 68*(1), 1-17.
86. Vargo, S. L., & Lusch, R. F. (2008). Service-dominant logic: continuing the evolution. *Journal of the Academy of Marketing Science, 36*(1), 1-10. doi:10.1007/s11747-007-0069-6
87. Vargo, S. L., & Lusch, R. F. (2011). It's all B2B…and beyond: Toward a systems perspective of the market. *Industrial Marketing Management, 40*(2), 181-187. doi:https://doi.org/10.1016/j.indmarman.2010.06.026
88. Vargo, S. L., & Lusch, R. F. (2016). Institutions and axioms: an extension and update of service-dominant logic. *Journal of the Academy of Marketing Science, 44*(1), 5-23. doi:10.1007/s11747-015-0456-3
89. Vernon, V. (2016). *Domain-Driven Design Distilled*. Boston, MA, USA: Addison-Wesley Professional.
90. Vial, G. (2019). Understanding digital transformation: A review and a research agenda. *The Journal of Strategic Information Systems, 28*(2), 118-144. doi:https://doi.org/10.1016/j.jsis.2019.01.003
91. Warg, M., & Deetjen, U. (2021). *Why co-creation in service ecosystems needs an architectural view*. Paper presented at the Co-VAL Conference: Value Co-creation, Innovation and Digital Transformation in Public Services, Universidad de Alcalá, Madrid.
92. Warg, M., Weiß, P., & Engel, R. (2015). *Whitepaper: Service Dominierte Architektur (SDA): die digitale Transformation erfolgreich meistern*.
93. Warg, M., Weiß, P., Engel, R., & Zolnowski, A. (2016). Service Dominant Architecture based on S-D logic for Mastering Digital Transformation: The Case of an Insurance Company. In T. Russo-Spena & C. Mele (Eds.), *Proceedings - What's Ahead in Service Research?* (pp. 807-826). Naples, Italy: RESER, University of Naples "Federico II".
94. Warg, M., Zolnowski, A., Frosch, M., & Weiß, P. (2019). *From Product Organization to Platform Organization - Observations of Organizational Development in the Insurance Industry*. Paper presented at the The 10 years Naples Forum on Service, Ischia, Italy.
95. Weill, P., Subramani, M., & Broadbent, M. (2002). Building IT Infrastructure for Strategic Agility. *MIT Sloan Management Review, 44*(1), 57-65.
96. Weiß, P. (2020). *Microfoundations for Building Systems of Engagement: Enable Actor Engagement Using Service Dominant Architecture.* Paper presented at the Advances in the Human Side of Service Engineering, Cham, Switzerland.
97. Weiß, P., Kronibus, A., Riedel, F., & Rittweger, R. (2021). *Digital Service Innovation and Actor Engagement: A Multilevel Design Perspective - Impacts from a Case Study of an Insurtech*. Paper presented at the 12th International Conference on Applied Human Factors and Economics, Manhattan, NY, USA.

98. Weiß, P., Warg, M., & Zolnowski, A. (2019). *Building Systems of Engagement to overcome the challenges of digital transformation*. Paper presented at the The 10 years Naples Forum on Service, Ischia, Italy.
99. Weiß, P., Zolnowski, A., Warg, M., & Schuster, T. (2018). *Service Dominant Architecture: Conceptualizing the Foundation for Execution of Digital Strategies based on S-D logic*. Paper presented at the 51st Hawaii International Conference on System Sciences, Hilton Waikoloa Village, Hawaii.
100. Wernerfelt, B. (1984). A Resource-Based View of the Firm. *Strategic Management Journal, 5*(2), 171-180.
101. Westerman, G., Bonnet, D. & McAfee, A. (2014). The Nine Elements Of Digital Transformation. *MIT Sloan Management Review, 55*(3), 1-6.
102. Whittington, R. (2010). Giddens, structuration theory and Strategy as Practice. In D. Golsorkhi (Ed.), *Cambridge Handbook of Strategy As Practice*. New York: Cambridge University Press.
103. Yin, R. K. (2018). *Case Study Research and Applications: Design and Methods* (6th ed.). Thousand Oaks, California: SAGE Publications, Inc.
104. 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, 21*(4), 724-735. doi: https://doi.org/10.1287/isre.1100.0322
105. Zimmermann, E. W. (1951). *World Resources and Industries*. New York, NY, USA: Harper and Brothers.

APPENDICES

Appendix 1: Povzetek (Summary in Slovene language)

Ker digitalna tehnologija vse bolj vpliva na naše življenje, se podjetja trudijo slediti novi dinamiki svojega tržnega okolja. Skladno s tem morajo organizacije razviti dinamične zmogljivosti, ki kratkoročno omogočajo izvajanje usmerjenih poslovnih pobud. Storitveno prevladujoča arhitektura (service-dominant architecture - SDA) je pristop k poslovni arhitekturi (enterprise architecture - EA), ki na podlagi obstoječih zapisov gradi plast sistemov sodelovanja. Na to novo arhitekturno plast vplivajo koncepti storitveno dominantne logike (service-dominant logic - SDL). Določa zmogljivosti informacijske infrastrukture, ki jih morajo podjetja uvesti kot nove pristope k na storitvah temelječi digitalni transformaciji (DT). Nedavni napredek pri razčlenitvi konceptov SDL, da bi bili uporabni za praktike, je sprožil zamisel o razvoju SDA. Zahvaljujoč večstopenjskemu pristopu za ustvarjanje vrednosti in večstopenjskemu načrtovalskemu procesu za oblikovanje sistemov storitev se je pojavila priložnost za razvoj SDA v okolje za oblikovanje, gradnjo in delovanje podjetij, ki temeljijo na storitvah.

V raziskavi smo pokazali, kako je mogoče pristop oblikovanja sistema storitev na več ravneh uporabiti pri oblikovanju učinkovitih sistemov sodelovanja z uporabo arhitekture, ki prevladuje v storitvah. Njen namen je ponuditi praktikom in zlasti managerjem globoko razumevanje storitveno usmerjene digitalne transformacije. Natančneje, cilj je pokazati izvajalcem, kako lahko oblikujejo, izvajajo in razvijajo storitvene sisteme za spodbujanje inovacij storitev.

Na podlagi raziskovalnega projekta oblikovalske znanosti naloga prispeva k stalnemu razvoju SDA v raziskavah in praksi. Zagotavlja kritičen pregled ustrezne literature. Pregled in študija primera, izvedeni z dvema podjetjema, zagotavljata strogost in pomembnost za kasnejši razvoj novega artefakta SDA.

Rezultat te študije primera je nov okvir za vodenje praktikov pri oblikovanju in kasnejši implementaciji poslovne arhitekture, ki jo navdihuje SDL. Združuje večstopenjski pogled na ustvarjanje vrednosti, večstopenjski proces oblikovanja za oblikovanje storitvenih sistemov in primer iz resničnega življenja iz raziskav SDA. S tem praktiki dobijo celovit model delujočega storitvenega sistema, ki prikazuje komponente poslovne in IT arhitekture, ki jih je treba upoštevati za omogočanje ustvarjanja vrednosti v ekosistemu storitev. Praktiki lahko s predlaganim okvirom spoznajo delovanje servisnega sistema, ki ga navdihuje SDL, in nato svoje znanje prenesejo v svoje aplikacije. Postavitev okvira spodbuja interdisciplinarno sodelovanje med poslovnimi in IT strokovnjaki, da se zagotovi skladen razvoj poslovne arhitekture v okviru digitalne transformacije.

Naloga je uspešno združila različna dela v bolj celovit okvir na več ravneh. Kot takšna prispeva k uporabi v praksi in oblikuje nadaljnji teoretski razvoj SDA. Vendar pa je potrebno več raziskav. Zlasti testiranje predlaganega okvira v kontekstih, ki niso povezani z raziskavami SDA, bo potrebno za tudi v prihodnje. Zato predstavljena študija postavlja

temelje za zorenje SDA v splošno uporaben okvir za vodenje digitalnih strategij v celotni digitalni transformaciji.

Appendix 2: Interview Transcript Element 2

Interview Transcript[5]

Interviewee: Element 2, CEO, SDA SE

Interviewer 1: Norman Bächtold, Student, SEB LU / Hochschule Pforzheim

Interviewer 2: Fabian Riedel, Student, Hochschule Pforzheim

Date: 02.06.2021

Audio Length: 00:44:41 hrs

Begin of Audio

NB:	Dann würde ich jetzt einfach mal einsteigen mit der Frage, was aus deiner Sicht wichtig ist für den Erfolg digitaler Service-Strategien. Also wir sehen da zwei Teilbereiche, zum einen die geschäftliche Seite – Business Strategy – und dann die IT-Seite mit der IT-Strategy. Und was da aus deiner Sicht die Erfolgsfaktoren für Unternehmen wie SDA SE sind, damit das Geschäftsmodell Service funktioniert.
E2:	Also wichtig ist, dass wir ein B2B-Dienstleister sind. Wir sind darauf angewiesen, dass unser Modell beim anderen, beim zweiten B verstanden wird. Wir sind darauf angewiesen, dass man genau diese Trennung übersetzen kann. Einmal ist es wichtig, wie du gesagt hast, diesen Business-Value festzumachen, weil unter dem Begriff Digitalisierung selbst – das wisst ihr auch – kann man glaube ich alles fassen und jetzt muss man sich überlegen, diesen Value, den ich aus der Digitalisierung ziehen möchte, auf etwas herunterzubrechen, damit ich ihn an ein konkretes Ziel hängen kann. Weil sonst kann ich keine Schritte machen. Und das ist immer der schwierigste Schritt: Man fängt mit Digitalisierung an, man fängt mit Service an, hat große Begriffe und muss sie jetzt übersetzen – Was ist denn jetzt das Ziel, das hinter dem Mittel Digitalisierung oder mit dem Mittel Digitalisierung, Digitaler Strategie oder sogar digitaler Transformation erreicht werden soll. Sind das eher operative Effizienzziele, die ich erreichen will? Ist mein Business Value also eher in der Saving-Ecke und ich sag ich will Prozesse effizienter machen? Hab ich Vertriebsziel dahinter und sage naja, ich sehe Digitalisierung eigentlich über den Weg Interaktion zu erhöhen in mehr Value aus beispielsweise neuen Kunden heranzuziehen. [inaudible]. Oder aber bestehende Kunden sag ich mal, meinen Teil, den ich aus deren Portemonnaie ziehe, zu erhöhen. Oder ist es so, ich möchte, keine Ahnung, dass überall Service und Zufriedenheit regieren. Und letztlich ist es eine Mengenlehre: Ich möchte ja nicht nur das eine, das greift übereinander an und das müssen wir trennen. Und diese Arbeit des Trennens ist für uns das wichtigste, um Kunden zu gewinnen. Und dann gehen wir nochmal auf die Middle-Lane und sagen, wie schlägt sich denn das, was wir als Business Value erreichen wollen, in die IT-Strategie eines Unternehmens runter. Also wie übersetze ich jetzt meine Ziele, in Business Value und wie unterstützt mich IT und ein Modell hinter der IT und dann sind wir nämlich bei der Architektur, diese Ziele zu erreichen. Und das ist das, wo wir unterwegs

[5] Passages referenced in the research report have been translated and italicized in the transcripts.

	sind. Und dann muss man sich überlegen, wie gestalte ich das ganze zukunftsfähig, sodass, wenn ich meine Annahmen nochmal überarbeiten muss, dass ich trotzdem damit umgehen kann? Also wie kann ich mich aufstellen? Und dann kommen diese ganzen Begriffe, die wir ja auch mitführen in der SDA und die wir als Company anderen Companies verkaufen. Wir sind ja immer in der Zwitterrolle – warum entwickle ich moduler. Das mache ich ja nicht als Selbstzweck, sondern das mache ich einmal technisch aus Security-Gründen, aus Produktionsgründen aber auch, um es neu zusammenstecken zu können, um neue Arten von Business Value generieren zu können. So, diese Story sortieren zu können und überhaupt erst ein Ziel zu finden und meine Ziele runterbrechen zu können und mich dann fragen zu können, mit welchen Mitteln – und die SDA ist ein Mittel und eine Architektur drumherum – kann ich diese Ziele erreichen. Wenn ich diesen Weg gegangen bin, dann mach ich Schritte in Richtung IT-Strategie und Business-Value-Weiterentwicklung. So als… Ich hoffe, das war nicht zu komplex.
NB:	Ne, das ist sehr gut. Das klingt auf jeden Fall auch ein bisschen so als ob sich eure Praxis mit der Theorie deckt. Ähm… Das heißt also, die SDA, als theoretisches Framework, hilft euch auch dabei, zum einen Mal im Unternehmen selber eine gewisse Sprache zu finden…
E2:	Ja.
NB:	… um diese Service-Strategie voranzutreiben. Aber die SDA als theoretisches Framework hilft euch auch, eure Kunden und Partner mit ins Boot zu holen. Habe ich das richtig verstanden?
E2:	Genau, wir verwenden die SDA auch untechnisch als Suchlogik. Also wenn wir Beispiele, wenn wir versuchen, mit Kunden oder für uns selbst neue Lösungen zu finden, dann haben wir die SDA immer technisch integriert, als Architektur integriert. Wir suchen sie aber quasi auch als, ja… ich möcht's nicht einer wissenschaftlichen Heuristik gleichsetzen, aber wir suchen sie schon heuristisch. Also wir gehen mit dem Unternehmen hin, aber auch für uns selber und sagen „Hey, wo sind denn interessante Bestandsdaten und Ressourcen, operante Ressourcen, die ihr nutzen könnt, also wo habt ihr eigentlich etwas, was andere noch nicht haben? So, letztlich ist das, ich mein' die ganzen Five Forces sind irgendwie darauf entstanden, dass man irgendwie mal kuckt… Ja, also sehr alte Strategie, aber ja nicht alles von denen ist falsch, also vieles schon… Wo habe ich eigentlich ne Position, die andere noch nicht haben? Und die sind halt in meinen Bestandsdaten und in meinen Kundenbeziehungen und in meinem Wissen drin und in meinen Daten. Und dann kucken wir, wo kannst du das eigentlich näher an den Kunden bringen. Dann sind wer im System of Interaction, also wie kann ich das eigentlich in Interaktion übersetzen? Weil ohne Interaktion entsteht kein Business. Also jedes Geschäft, es ist ja seit Urzeiten so: Ohne eine Interaktion gibt es kein Geschäft, weil letztlich jeder Vertrag ein zweiseitiger Vertrag ist. So, deswegen ist Interaktion dem Geschäft immanent. So und dann die Dritte, und das ist eigentlich die schwierigste Dimension der Suchlogik, okay, wenn wir das verstanden haben, wo gehen wir über das System of Participation, wo gibt es jetzt Partner und Anreicherungen von außen, die als intelligent orchestrierte Ressourcen mir Mehrwert schaffen. Entweder für die Interaktion oder für die Daten. Und daraus suchen wir dann das wieder, was ich eben beschrieben hab: So, in welche Richtung möchtest du denn damit gehen? Ist es eher ein digitales Dokumentenmanagement, da geh' ich auf operative Exzellenz, gehe ich auf Service-Mehrwerte und sag, ich kann einen neuen, innovativen Service anbieten? Und so suchen wir tatsächlich. Beispiele, die ihr auf der Homepage findet, wie dieses Beispiel edith.care.

	Exactly, we also use the SDA non-technically as search logic. So when we take examples, when we try to find new solutions with customers or ourselves, we always have the SDA technically integrated and integrated as architecture. But we also search for it as, yes... I don't want to equate it with a scientific heuristic, but we do search for it heuristically. So we go there with the company, but also for ourselves and say "Hey, where are interesting inventory data and resources, operant resources that you can use, so where do you actually have something that others don't have yet?". So, in the end, I mean, the whole Five Forces were somehow based on the fact that you somehow look at... Yes, so very old strategy, but not everything of them is wrong, so a lot of it is... Where do I actually have a position that others don't have yet? And they are in my inventory data and in my customer relationships and in my knowledge and in my data. And then we look at where you can actually bring that closer to the customer. Then we're in the system of interaction, so how can I actually translate that into interaction? Because without interaction, there is no business. So every business, it's been like that since time immemorial: Without an interaction there is no business, because ultimately every contract is a two-way contract. So, that's why interaction is intrinsic to business. So and then the third, and this is actually the most difficult dimension of the search logic, okay, once we understand that, where do we go beyond the System of Participation, where are there now partners and enrichments from the outside that add value to me as intelligently orchestrated resources. Either for the interaction or for the data. And from that, we then look for again what I just described: So, what direction do you want to go with this? Is it more digital document management, do I go for operational excellence, do I go for service added value and say I can offer a new, innovative service? And that's actually what we're looking for. Examples you can find on the homepage, like this example edith.care. Hat vielleicht auch der Markus Warg erwähnt. Das ist genau so entstanden: Wir haben gesagt „Okay, wo ist denn ne Herausforderungen, die ein Kunde hat und die noch nicht gelöst ist? Welche Interaktion findet statt? Welchen Partner kann ich intelligent anbinden und wo hab‘ ich Ressourcen, die die anderen auf ihrer Ebene nicht haben?“. Und das ist auch ne Form von Wettbewerbsfähigkeit. Um mal an ner Case Study zu bleiben, haben wir ein Beispiel: Aus dem Gesundheitssektor. Es entstehen im Moment viele Anwendungen im Bereich der digitalen Gesundheitsanwendungen. Die kennt ihr ja auch. Also wir haben klassischerweise eins, das geht auf koronare Krankheiten, Schlaganfall. Aber alle Krankheitsbilder kriegen gerade digitale Anwendungen, so. Diese digitalen Anwendungen entstehen – ich bleib jetzt mal in einem Schubladenbild, das stimmt auch nicht ganz, also die Zugespitztheit bitte bisschen entschuldigen – die entstehen irgendwo in Berlin. Das heißt in Berlin fängt eigentlich einer out-of-the-box an und sagt „Ich hab‘ ne gute Idee“. So, was können die oft viel besser? Die binden sich an den Nutzer. Die machen besseres Frontend und dadurch, dass sie’s selbst nicht haben, orchestrieren die schneller. Also das Layer System of Interaction bauen, die vielleicht sogar technisch besser muss ich ehrlich sagen, auf, weil die auch sehr spitz einen Kunden, hier einen Patienten, adressieren können. So und dann sind die meist sehr viel schneller live da, als wir das schaffen. Die holen sich grade ne Verbindung zu Apple Health und dann sind die auf der Uhr. Was die aber nicht haben, und das hat noch nicht mal Apple. Apple hat eine Historie. Wenn ich jetzt auf Daten lernen will, brauche ich eine Historie. Sind wir bei dem Beispiel

	Schlaganfall. Eine Idee, die wir gerade verfolgen: Da ist eine KI entstanden, die ist in ner Klinik entstanden.
NB:	Das ist dieses ai4Medicine, richtig?
E2:	Genau. Als gültige Case Study. So, der hat im klinischen Umfeld was gemacht. Das überträgt er jetzt. Jetzt hat er ne App, da hat er Apple Health drin. Dann komm ein User ran. Der braucht jetzt aber, um sein klinisches Modell zu validieren, braucht der historische Daten. Das kann eine Versicherung leisten, wenn man an deren Daten kommt. So, und da ist wieder die Suchlogik. Ich habe irgendwo jemanden, der etwas reinhievt, als Partner, über das System of Participation. In diesem Bereich, sag ich mal, ist die Zulieferung der Ressource letztlich eine KI-Fähigkeit, oder die Ressource ist die Kompetenz, aus Daten zu erkennen, was sind denn Prädispositionen für Schlaganfall und wie kann ich diesen begegnen? Kommt über's System of Participation rein. System of Interaction entweder was Selbstgebautes oder ich geh in die Kunden-Apps rein und von unten krieg ich die Bestandsdaten. Und dann kann ich auf einmal Bestandsdaten, also historische Daten, matchen mit Live-Daten von einer Apple-Watch und vielleicht noch neuen Abrechnungen, die reinkommen. Und in dem Moment schaff ich ein Service-Bündel, ein orchestriertes Service-Bündel, das überlegen ist jeglicher Kombination, in der nichts miteinander verbinde. So, und deswegen nehmen wir die SDA auch immer als Suchheuristik, um nach neuen Lösungen zu suchen und das hat sich doch tatsächlich bewährt.
FR:	Sie haben jetzt gesagt, die SDA hat für Sie mehrere Facetten auf einmal als die konkrete Implementierung in eine Service-Plattform und auf der anderen Seite diese, ähm, Heuristik, mit der Sie arbeiten. Was für Voraussetzungen müssen dann in Ihren Partnerunternehmen gegeben sein, dass eine Zusammenarbeit stattfinden kann? Also wenn das Unternehmen aus mehreren Bereichen besteht und die schon ein bestimmtes Geschäftsmodell haben -Vertrieb, oder Strukturen in der Entwicklung. Wie müssen da die Voraussetzungen sein, dass die SDA-Heuristik und die SDA-Plattform funktioniert in diesem Kontext?
E2:	Also eine basale Voraussetzung ist tatsächlich, und das ist die schwerste, das ist Offenheit auf verschiedenen Ebenen. Das ist Offenheit für… ähm… Lösungen von außen anzunehmen. Also viele Industrien sind es einfach nicht gewohnt, Lösungen von außen zu integrieren, obwohl da ja alle BWLer sitzen und man im ersten Semester quasi über das Ricardo-Theorem eigentlich lernen sollte, dass klassischerweise, wenn einer was besser kann und selbst wenn er keine absoluten Vorteile hat, es aus komparativen Vorteilen sich lohnt, Leute, die woanders was besser können, zu integrieren. Das fällt aber vielen Industrien schwer. Das ist für mich einmal diese Offenheit, das ist letztlich ein Philosophieumschwung. Ich bau es nicht mehr selber und nehm' nicht nur einen Standard von außen, sondern ich nehme tatsächlich Ressourcen und Kompetenzen von außen wahr. Zweiter Teil ist: Ich gebe etwas zurück. Also man muss auch sagen, wir reden ja immer mit großen Unternehmen und die großen Unternehmen sind eher konsumierend unterwegs. Also in die fliegt etwas rein. Wir sind in einer Welt unterwegs, wo auch kleinere Startups sagen „Okay, ich hab aber die Nische besser verstanden. Deswegen möchte ich von euch etwas zurück". Und dann kommen wir zum dritten: Man muss ein bisschen von dem Geld abstrahieren, ich möcht jetzt nicht auf den alten Satz raus „Daten sind das neue Öl". Ähm… Die wollen tatsächlich nicht Geld, weil die Startups können Geld bekommen, sondern die wollen teilweise auch Daten oder Kundenzugang zurück. Das ist auch natürlich neu für viele Unternehmen. Sie kämpfen einerseits mit uns um eine Intensivierung der Kundenschnittstelle, sollen

	aber vielleicht einen Teil der Kundenbeziehungen abgeben. Das ist neu. Dann, ein wichtiger Punkt ist Datenhoheit. So und dann rutschen wir in das ganze Feld der technischen Kompetenzen. Ne also, die müssen sich drauf einlassen, ihre Systeme zu öffnen. Der Weg in die Cloud muss zumindest möglich sein. Also wir stehen ja auch unterhalb der Cloud aber die Bedingungen müssen auch erfüllt sein. Wir haben, und das ist eigentlich der Schritt des letzten Jahres, viel daran gearbeitet das, ähm… was in der SDL als Institutionen bezeichnet wird, so den Rahmen einfacher zu machen, um über diese Stöckchen zu springen. Und das ist wo wir verstärkt dran arbeiten. Also das fängt an bei mehr Dokus, ist jetzt bei Service Katalogen, Developer Pools. Also das ist eigentlich das ganze Feld der Ermöglichung von Cocreation, was herausfordernd ist. Sie wollen es alle, aber es ist halt ein herausforderndes Feld von Companies, die eigentlich sich jahrelang und jahrzehntelang als interne Systeme, die von außen maximal konsumieren, in eine kokreative Zukunft zu entwickeln. Das fällt denen schwer. So und dafür versuchen wir auch Rahmen und Institutionen zu setzen damit Austausch mit anderen einfacher wird. Also wenn man mal ne Flugebene hochspringt, man versucht immer Beschränkungen abzubauen. Ich mein wir haben in der ganzen Wirtschaft sehen wir immer es gibt die Rückkehr zum Nationalismus in einigen Bereichen. Also… Boris Johnson hat damit nen Wahlkampf gewonnen, Trump wäre fast dageblieben, alle anderen treiben es auch. Also, jetzt sagen wer mal so: Daraus übertragen die Idee, ich kann doch selber besser und ich beschütz lieber meine Grenzen und konsumiere aber zu meinem eigenen Vorteil. Die ist dem Menschen ja immanent. Sonst würde sich das nicht immer wieder fangen. Die ist aber wirtschaftlich und kulturell widerlegt, aber der Sprung ist immer wieder da. So, jetzt muss ich Institutionen schaffen, die den Abbau von Beschränkungen möglich machen. So, und diese Metaebene zu erreichen, das versuchen wir auf ganz ganz vielen kleinen Ebenen. Sei es mit gemeinsamen Workshops, Hackathons, mit Dokumentationen, Institutionen. Nur dafür gibt es eigentlich Standards. Ähm… dafür sind von außen Validierungen erforderlich. Also es uns zum Beispiel unglaublich geholfen, dass eine McKinsey jetzt offiziell auf uns sitzt und dass jetzt zum Beispiel ein Gartner das Bild einer Composable Finance rumtreibt, was genau unserem Modell entspricht. Das hilft einfach und man merkt immer wieder, dass Menschen in Kokreation zu bringen, die teilweise diese Erfahrung noch nicht haben, eins der größten kulturellen Hindernisse ist. Die Technik ist es ja faktisch nie und auch das Thema Datenschutz, was immer angeführt wird, ist es auch nie. Es ist total lösbar. Auch das Thema Cloud ist lösbar. Es sind glaub ich eher kulturelle Hindernisse oder Erfahrungshindernisse, die man versucht, so zu überspringen, die uns einfach Geschwindigkeit kosten.
NB:	Mhm… Das ist super interessant, ja. Das hängt auch… Also, da seh‘ ich auch ganz eindeutig den Vorteil von der SDL. Also was die SDL macht, sie bietet euch ja praktisch ein Lexikon an neuer Sprache, mit der ihr dann ja auch Kunden zugehen könnt und mit der ihr dann Kunden auch praktisch eure Welt besser erklärbar machen könnt.
E2:	Ja.
NB:	Und das wird von der SDA dann ja eindeutig mitgetragen.
E2:	Ja genau. Genau, man hat dann… Menschen suchen überall nach Modellen. Das ist ja nicht nur in der Wissenschaft so. So die ganze Interaktion funktioniert ja auch über Unterstellungen und Modelle.
NB:	Ja.

E2:	So und deswegen brauchen wir diesen Rahmen, deswegen brauchen wir Modelle, deswegen brauchen wir Bilder, damit die auf neue Wege einschwenken, weil wir bewegen uns in einem Bereich von Unsicherheit. Also das Thema Kokreation ist einfach in Unternehmen nicht so gesetzt. Dass Wirtschaftssystem, das ja… In der BWL lernt man halt Wettbewerb… und nicht Kokreation. Da war neulich n interessanter Artikel zum Wöhe, noch immer als Standardliteratur immer noch mal geupdated, geht halt total auf fundamentalen Wettbewerb und Gewinnmaximierung aus. Ist gar nicht so. Andererseits… ähm… wir bewegen uns ja sehr stark in einem Feld der Versicherungen. Den Versicherungen müsste es ja immanent sein… Man darf das ja nicht vergessen: Die großen Versicherungen sind immer noch von der Rechtsform her Versicherungsvereine auf Gegenseitigkeit und haben ja gar nicht das Ziel der Gewinnmaximierung. Das ist ja eigentlich sehr interessant. Die haben ja das Ziel der internen Kokreation und der Kooperation mit ihren Kunden ist ja deren Gründungsgeschichte. Deswegen müsste das denen eigentlich leichter fallen. Aber man hats ja über die Jahre verlernt. Ich glaube, eine Versicherung, sag ich mal, vor dem zweiten Weltkrieg, hätte uns kulturell – technisch natürlich nicht – hätte uns kulturell vielleicht einfacher verstanden als eine, das mein ich gar nicht böse, als eine in bestimmten Facetten durch-BWLisierte Versicherung heute. Und noch ein kultureller Punkt, aber bitte unterbrecht mich, wenn ich zu weit abschweife, aber es ist wieder auf einer Metaebene. Wir operieren ja sehr Stark mit dem Begriff Ökosysteme und Umwelt und das ist ja interessanterweise ne deutsche Erfindung und wenn man das mal sieht: Die große Erfindung an dem Begriff Umwelt war die Erkenntnis, dass wenn man in die Historie zurückgeht, dass man eigentlich erkannt hat, die Umwelt eines Hundes kann man nicht beurteilen, indem man sie mit den Kriterien des Menschen beobachtet. Sondern ein Hund nimmt seine Umwelt anders wahr als ein Mensch und die Umwelt des Hundes ist eine andere Umwelt als die des Menschen, auch wenn sie faktisch gleich ist. So, das ist ne interessante Erkenntnis. Also wir reden über ein Ökosystem, aber man kann kein Ökosystem malen, das für alle gültig ist, weil das Ökosystem von Versicherungsunternehmen 1 sieht aus Perspektive von Unternehmen 1 komplett anders aus als das Ökosystem von Unternehmen 2. Vielleicht nicht komplett. So und das ist das wo man sich auch immer mal wieder reinversetzen muss. Und auch diesen Ökosystemgedanken neben diesem ganzen Definitionswirrwar, was ist eigentlich ne Plattform und n Ökosystem, das ist nochmal kulturell. Und da bietet die SDL auch einfach Ansatzpunkte, indem sie, sag ich mal, diese Ressourcenintegration sehr offen gestaltet.
FR:	Ähm wenn Sie jetzt sagen, dass die Welt für jede Versicherung anders aussieht, dann bieten Sie ja mit der SDA dennoch diesen gemeinsam Faktor oder Nenner an, auf dem jetzt alle verschiedenen Versicherungen gebündelt werden – zum Teil für einen bestimmten Use-Case, wie zum Beispiel bei edith.care. Da findet dann eine Kooperation statt.
E2:	Genau, aber was wir definitiv auch machen ist a) wir belassen ja die Versicherungen oder Unternehmen ihren eigenen Instanzen. Also wir werden auf den allerersten, schnellen Weg immer falsch wahrgenommen und [die Versicherungen] sagen dann die SDA ist ja eine Plattform und an die muss ich mich jetzt anschließen. Das sind wir nicht. Sondern die SDA SE bietet ja Instrumente, damit die Versicherungen, sich eine Plattform und ihr Ökosystem aufbauen können. Das liegt ja in der Instanz des Unternehmens. Letztlich bleibt ja auch die Datenhoheit in ihrer Instanz. Also man muss sich ja nicht der SDA [SE]

	anschließen, sondern wir liefern Punkte. Das ist vielleicht ganz wichtig. Und der zweite Teil ist wirklich, ja wir haben eine gemeinsame Architektur. Die fußt auf einem gemeinsamen Verständnis. Also die SDA-Architektur ist mehr aus der SDL abgeleitet, als die Technik. Ähm… und sag ich mal, die Platte, wie man entwickelt, oder die Produktionsstraße für Services, einmal technisch verstanden als technische Services, aber jetzt auch mehr SDL-orientiert als Services verstanden, die dann entstehen. Die ist normiert oder folgt einem Denkmodell. Die Bausteine darauf, die sind ja bewusst so generisch konzipiert, dass man sie untereinander austauschen kann. Also die Versicherung A kann mit der SDA etwas anderes machen, als Versicherung B. Und sie kann auch andere Partner integrieren. Und das ist auch wichtig, weil jeder irgendwie immer noch nach Individualität strebt. Das ist so eine Mitigationsstrategie, um dieses „Ich geh jetzt von schwarz nach weiß“ oder umgekehrt von komplett geschlossenes System auf offenes System irgendwie einfacher zu machen. Das ist letztlich einfach auch ein Teil des Story-Telling, um Kooperation oder „Sich-Öffnen“ zu erleichtern.
FR:	Das bedeutet dann, dass für die SDA SE dieser Service-Katalog ganz zentral für die weitere Entwicklung des Unternehmens, weil dadurch erst die Attraktivität für die Partner und Kunden aufkommt?
E2:	Ja genau. Der Service Katalog, der ist ja sehr offen, und dass er die Institutionen mit sich trägt. Also es ist ja nicht nur ein reiner Katalog in dem man bestellen kann, sondern er vereint ja quasi auch verschiedene Normen unten drunter. Also in dem Service-Katalog sind auch Security-Normen und die Tools drunter abgebildet. Letztlich kann man sagen, der Service-Katalog, wenn man den sich genauer ankuckt und technisch ist es ja auch so, ist ja auch ne Plattform oder ein Front-End in dem man verschiedene Ressourcen wieder integriert, mit dem Ziel, eine, mit dem Ziel, eine Art von... ja… Rules einfach zu setzen, Institutionen oder Regeln zu setzen. Und dann kommen wer wieder, für mich ist das immer so ein Ying-und-Yang-Bild: Du musst Regeln setzen und auf der anderen Seite sollen die aber Freiheiten ermöglichen. Also das muss sich immer Hand in Hand geben. Und das versuchen wir gerade. Du schaffst durch einheitliche Regeln, noch was anderes freiheitlich machen zu können. Durch Bausteine, die untereinander kombinierbar sind, schaffst du die Freiheit, damit individuelle Lösungen zu bauen. *Yes exactly. The service catalog is very open, and it carries the institutions with it. So it's not just a pure catalog where you can order, but it also combines various standards underneath, so to speak. So the service catalog also includes security standards and the tools underneath. Ultimately, you can say that the service catalog, if you take a closer look at it and technically it is also like that, is also a platform or a front end in which you integrate different resources again, with the goal of setting a, with the goal of setting a type of... yes... rules simply to set, to set institutions or rules. And then you come back to who, for me it's always such a ying-and-yang picture: You have to set rules and on the other hand they should allow freedoms. So that always has to go hand in hand. And that's what we're trying to do right now. You create uniform rules that allow you to do other things freely. By using building blocks that can be combined with each other, you create the freedom to build individual solutions.*
NB:	Da hätte ich dann jetzt noch eine grundsätzliche Verständnisfrage. Du hattest ja gesagt, dass ein Versicherungsunternehmen, das jetzt mit der SDA SE zusammenarbeiten möchte, muss sich nicht unbedingt euch anschließen, sondern

	kann auch einfach sein eigenes Ökosystem und seine eigene Plattform aufbauen. Aber wie ist das Unternehmen dann noch mit euerm Unternehmen verbunden?
E2:	Genau. Also nicht ne völlig eigene Plattform, sie muss schon die SDA-Plattform nutzen, unsere Produktionsstraße, aber sie nutzt sie in einer eigenen Instanz.
NB:	Okay.
E2:	Also sie produziert nicht auf unserer Produktionsstraße – wenn man mal bei dem Bild bleibt – sondern sie baut sich eine eigene Produktionsstraße auf, auf der sie Services aufbaut, aber eben nach Regeln dieser Produktionsstraße, nach SDA-Regeln, sodass sie mit andern, die auch die SDA-Produktionsstraße nehmen – wir, aber auch andere Nutzer – dass austauschbare Objekte entstehen. Aber sie nutzt halt… Es gibt ja auch Plattformen, an die ich mich einfach anschließe, wo ich sage ich konsumiere nur per API. Bei uns kann man ja auch selbst entwickeln. Das ist vielleicht ein wichtiger Unterschied. Und dieses Instanzmodell ist halt wichtig für Datenhoheit und letztlich auch für die Wahrnehmung.
NB:	Dann würd‘ ich gern mal noch von der SDA SE komplett weggehen und noch ein anderes Thema anschneiden. Und zwar, die SDA als theoretisches Framework sieht sich ja ein Stück weit als IT-Strategie, die einem Unternehmen dabei helfen soll, sich digital zu transformieren und ein Service-Geschäftsmodell aufzubauen. Wenn wir uns jetzt einmal in die Lage von einem, sagen wir, mittelständischen Anlagenbauer versetzen. Wie hilft so einem Anlagenbauer die SDA als theoretisches Framework? Wie sollte sich so ein Unternehmen dem Thema digitale Transformation mit der SDA annähern?
E2:	Ich glaube, da ist dann wieder das heuristische Modell, was dahinter steckt, wichtiger um zu sagen „Okay, was ist eigentlich das, was mich ausmacht und wie kann ich das, was mich ausmacht in Services weiter darstellen und diese Services weiterentwickeln?“. Also ich würde immer danach suchen… Also mit einem Anlagenbauer… Die definieren sich ja historisch sehr stark aus ihren Maschinen heraus. Ich weiß aber die letzten zehn Jahre sind die ja auch in Services gegangen. Die verkaufen ja auch… So einen Voith… Die verkaufen ja auch ihre Services. Das Service-Verständnis, das ist ja dieses alte Rolls-Roye-Modell - Ich verkaufe nicht mehr den Flugzeugmotor sondern ich verkaufe jetzt die Minutennutzung – ist ja aber immer noch ein Stückchen entfernt von dem Serviceverständnis wie eine SDL es naheliegen würde. Demnach ist Service ja mehr. Da ist das Produkt auch drin, aber ich glaube die Orchestration nochmal aufzumachen und diese Suchlogik, wenn ich da schon bin… Also wenn ich sage „Okay, ich habe mal ein Produkt gehabt, nämlich eine Maschine. Die hab‘ ich verkauft.“ Jetzt komm ich an ein neues Modell und da sind die glaube ich alle schon und sagen: „Naja, hm… Dieser eigentliche Produktverkauf… Und dann hab‘ ich vorher zwar irgendwie Leasingmodelle gehabt aber das sind ja eigentlich nur Finanzmodelle gewesen. Jetzt muss ich eigentlich in das Modell. Ne der Kunde will aber eigentlich von mir gar nicht die Maschine, das haben wir mit dem ersten Schritt erkannt. Sondern der will ja die Leistung, die die Maschine bringt“. Also ob man da jetzt sagt der will den „Job tobe done“ erledigt haben. Der will ja nicht eine Maschine kaufen und Papier herstellen, sondern er verkauft seinem Kunden Papier, wenn ich jetzt bei Voith Paper bin oder sowas. Und jetzt die nächste Logik ist „Wie kann ich das jetzt tatsächlich weiterentwickeln? Wenn ich diesen Schritt schon gegangen bin, wie kann ich mit anderen den Service-Mehrwert meines Kunden, und dazu muss ich den Kunden verstehen, auf ein nächstes Level bringen?“. Und dann komme ich gleich wieder in dieses kooperative: Was kann ich beitragen und wo komme ich eigentlich in Win-Win-Situationen? Sei es in Logistik, sei es in

	Vermarktungspartnerschaften oder sei es in der gemeinsamen Produktweiterentwicklung. Ich glaube da kann die Service-Dominierte Logik helfen und davon ist die SDA wieder eine technische Umsetzung. Und dafür muss ich aber modular denken und dafür muss ich so aufbauen, dass ich diese einbinden kann. Ich glaube das sind die nächsten Schritte, die so anstehen. Wie dein Beispiel Maschinenhersteller, ich muss einen Service, Rolls-Royce-mäßig, ich wall ja vermieten und gehe nach Output, so funktioniert mein Modell auch und letztlich laufen so ja auch die Ausschreibungen. Wenn ich jetzt mal einen Maschinenhersteller als einen Windanlagenhersteller sehe, der verkauft ja auch keine Windmühle mehr sondern der [inaudible] und was da für eine Mühle steht interessiert mich gar nicht mehr. Und den Schritt sind schon viele gegangen. Okay, und was ist jetzt das nächste Modell als Anlagenbauer? Wie orchestriere ich eigentlich, also welcher Nutzenmehrwert steht dahinter? Da sag ich, ich möchte ja vielleicht nicht nur den Output – jetzt bei der Anlage – sondern ich möchte auch die Sicherheit, dass es immer funktioniert. Also gebe ich den Service dazu. Reicht der Service? Ne, dann möchte ich vielleicht auch noch die Sicherheit, dass im Service-Ausfallfall eine Kompensation bekomme. Dann hab‘ ich vielleicht schon sowas wie eine Industry-Insurance drauf. So, dann möchte ich vielleicht noch ein Monitoring drauf, und das Monitoring soll aber mit den anderen Maschinen zusammenpassen. Dann komme ich in ein Modell, wo ich sage, ist ja super, dass ich ein Monitoring für meine Maschinen habe als Service-Anbieter, aber wie wird so ein Monitoring denn interoperabel? Das nächste Service-Modell ist dann „Wer macht denn ein Monitoring auf dem Monitoring, weil ich möchte es nicht 24/7 machen?“. Und diese Idee, quasi im Service-Gedanken, im Nutzengedanken für den Kunden voranzugehen und das wieder runterzubrechen, das ist das, wo die SDA helfen kann. Und dann kann ich anfangen, das wieder in Business Value zu zerlegen und in Techniken und eine Modularität zu zerlegen… Ich hoffe, das war nachvollziehbar.
NB:	Ja doch, doch, sehr. Das heißt aber auch, wenn wir bei dem Beispiel bleiben, der Aufbau von einer Plattform, wie sie bei euch in Betrieb ist, gar nicht zwingend notwendig ist, sondern es geht dann auch ein Stück weit eher um die modulare Logik von verschiedenen Geschäftsbereichen, oder hab ich das falsch verstanden?
E2:	Ja auch. Die brauchen auch eine modulare Logik und die werden natürlich auch technisch eine Plattform brauchen, um sich zu öffnen. Ich glaube aber auch, dass viele Maschinenhersteller auf diesem Weg ja schon sind. Das sieht man ja gerade, dass die interoperabel werden. Und jetzt muss man schauen, wartet man auf Standards oder geht man selbst den ersten Schritt? Muss man aber auch ehrlicherweise sagen, wir als SDA SE sind da fachlich noch nicht in diesem Bereich. Wichtig ist dann aber auch noch das Modell, was ich eben gesagt habe, was man einbringen muss: Man muss in dieses Tauschmodell kommen, was bringen die denn mit? Die bringen ja auch wieder ein spezifisches Wissen mit. Also das sind auch Kundenbeziehungen, das sind Maschinen-, Produktions- und Prozesserfahrungen. Das wäre in deren System of Records die Historie. Jetzt müsste ich mir auch überlegen, und da wird’s interessant, wie kann ich mein Wissen mit Partnern besser machen? Das ist ja auch eine Idee, eine Dimension, die die SDA immer wieder nahelegt. In einem Ökosystem muss ich teilen und in einem Ökosystem lebe ich davon, dass es meinen Teilnehmern auch gut geht. So und dann komme ich wieder in die Logik, die auch den Versicherungen schwerfällt: Jetzt hab‘ ich ein Prozesswissen. Jetzt hat man mir aber gesagt Prozesswissen ist

	vielleicht mein USP, mein Wettbewerbsvorteil und der Porter hat mir mal erzählt den muss ich auf Biegen und Brechen verteidigen, weil das ist einer meiner fünf Kräfte und da ziehe ich am Besten Grenzen rum damit die andern die nicht erreichen können. Also muss man ja auch mal, vielleicht auf so einem Metalevel, diese Fundamentalität, am Anfang ich soll das abschotten, das ist mein USP, verteidigen, in den Markt bringen und jetzt kommt eine SDL und sagt „Nenene, orchestrieren und mit anderen teilen und du musst dir überlegen, wie du dein Prozesswissen nutzen kannst und wie dein Ökosystem davon profitiert. Das ist schon ein relativ fundamentaler Wandel. *But then the model that I just said is also important, what you have to bring in: You have to come into this exchange model, what do they bring with them? They also bring specific knowledge with them. So, these are also customer relationships, these are machine, production, and process experiences. That would be the history in their system of records. Now, I would also have to think about, and this is where it gets interesting, how can I improve my knowledge with partners? That's also an idea, a dimension that SDA always suggests. In an ecosystem, I have to share, and in an ecosystem, I live off the fact that my participants are also doing well. So, and then I get back into the logic that is also difficult for the insurance companies: Now I have process knowledge. But now I've been told that process knowledge is perhaps my USP, my competitive advantage, and Porter once told me that I have to defend it by hook or by crook, because it's one of my five strengths and it's best to draw boundaries around it so that the others can't reach it. So, you also have to defend, maybe on a metalevel, this fundamentality, in the beginning I should seal this off, this is my USP, bring it to the market and now an SDL comes and says "No, orchestrate and share with others and you have to think about how you can use your process knowledge and how your ecosystem benefits from it. That's already a relatively fundamental change.* Das ist uns immer so immanent, aber so Unternehmen, die da 20, 30 Jahre unterwegs sind, und sag ich mal Manager, die komplett anders geschult worden sind, und irgendwie vor 30, 40 Jahren einen Abschluss gemacht haben, die sind echt in einer anderen Welt. Das darf man nicht unterschätzen. *That's always so intrinsic to us, but companies that have been around for 20 or 30 years and, let's say, managers who have been trained completely differently and somehow graduated 30 or 40 years ago, they really are in a different world. You can't underestimate that.*
FR:	Hier würde ich noch gerne anknüpfen. Mit der Frage von Norman wurde ja darauf abgezielt, wie so ein Entwicklungsprozess in einem Unternehmen aussieht, also wie führt man die SDA ein und wie macht man einen neuen UseCase. Worauf ich jetzt abzielen möchte ist, mit welchen Methoden operationalisieren Sie dieses Vorgehen konkret? Sie haben ja angesprochen, oder du hast ja angesprochen, dass die SDA-Heuristik im Grund verwendet wird und das hast du dann ja auch erklärt, wie ihr da ungefähr vorgeht, wie ihr da ungefähr vorgeht, aber was für Methoden verwendet ihr. Zum Beispiel sowas wie den Marketing-Mix, Customer Journeys, Domain Driven Design, Value Mapping…. Dieses ganze Bündel an Methoden, die es in der BWL-Literatur gibt.
E2:	Genau, also das Domain Driven Design ist bei uns eine Grundlage, wie wir uns aufstellen. Daran orientieren wir uns und das hilft letztlich auch der Sprachfindung

	auf fachlicher Ebene erstmal, wenn man sich in Domänen orientieren kann. Ähm… Customer Journeys sind ja… Ohne Customer Journeys kommt man nicht zum Endkunden und letztlich versuchen wir, an den Customer Journeys nachher die Module abzuleiten. Customer Journeys kommen bei uns ehrlicherweise ein bisschen später. Wir sind so ein bisschen, wenn wir vorgehen, nehmen wir Elemente des Design Thinking. Aber tatsächlich nur, es gibt DAS Design Thinking, und das ist ja auch so ein Hype-Begriff. Was aber natürlich gut ist ist dieses Verfahren am Anfang sag ich mal, die Möglichkeitswege aufzumachen. Das ist etwas, womit man Offenheit fordert, mit verschiedensten Methoden. Da kann man sich an dem ganzen Modell bedienen, nochmal zu kucken, wie mach ich meinen Möglichkeitsraum nochmal transparent und verabschiede mich aus dem Vorhandenen und gehe ich nochmal diese Welle hoch im Design Thinking. Und jetzt kommt ja die Welle runter aus diesem Möglichkeitsraum, die auszusuchen. Dann bin ich wieder am Anfang. Dann muss ich mir erst mal ein Ziel suchen, wohin will ich eigentlich, welche Form von Business Value will ich erreichen? Und dann verknappe ich ja wieder auf Möglichkeiten, auf Use-Cases, die ich gehen kann. So, diese Verknappung versuchen wir über eine SDA-Heuristik wieder zu kriegen. Wir sagen, „Okay, was ist euer Ziel? Wie können wir das in dieser Systemlogik System of Interaction, Participation, Operant Resources, [inaudible)… Wie kann ich Modelle daran entlanghangeln, dass alle davon profitieren und wer kriegt eigentlich was und wie komm ich eigentlich in Austauschbeziehungen rein? Was bringt der Endkunde rein, wie komme ich in Interaktion, was bringt ein Partner rein, was kriegt ein Partner zurück, was kann ich eigentlich aus meinen Ressourcen mitbringen, was nimmst du als Suchlogik?“. Und daraufhin gehen wir wieder in die Customer Journey und sagen, idealtypischer Prozess, und sagen entlang der Customer Journey, welche Building Blocks brauche ich eigentlich? Und welche dieser Building Blocks sind eigentlich ehrlicherweise USPs oder wirklich super spezifisch und welche sind eigentlich, wenn ich sie dann runterbreche, gar keine USPs und so Standardblöcke, die ich auch mit anderen teilen kann? Und dann kommst du an einer Customer Journey, nochmal ein Validierungsmodell, da fällt mir nochmal was ein, da kann ich nochmal was von außen dazuholen. Deswegen hast du eigentlich auch immer so ein Customer Journey-Bild und unten drunter hast du wieder diesen Kreis der SDA und du kuckst „Ah, da fällt mir noch ein Partner ein, den ich nochmal einbinde“. In so einem Bild orientieren wir uns eigentlich durch für neue Use-Cases. *Exactly, so Domain Driven Design is a basis for how we set ourselves up. We use it as a guide, which ultimately also helps us to find the right language at the technical level, if we can orient ourselves in domains. Um... Customer journeys are... Without customer journeys, you can't get to the end customer, and ultimately, we try to derive the modules from the customer journeys. To be honest, customer journeys come a bit later for us. We are a bit, when we proceed, we take elements of design thinking. But really, there is only THE design thinking, and that is also such a hype term. But what is good, of course, is this procedure at the beginning, I would say, to open up the possibilities. This is something that demands openness, with a wide variety of methods. You can use the whole model to look again at how I can make my possibility space transparent again and say goodbye to the existing and go up this wave again in design thinking. And now the wave comes down from this possibility space to select. Then I'm back at the beginning. Then I first have to look for a goal, where do I actually want to go, what form of business value do I want*

	to achieve? And then I narrow it down again to possibilities, to use cases that I can pursue. So, we try to get this shortage again via an SDA heuristic. We say, "Okay, what is your goal? How can we do that in this system logic System of Interaction, Participation, Operant Resources, [inaudible)... How can I get models along that everyone benefits and who actually gets what and how do I actually get into exchange relationships? What does the end customer bring in, how do I get into interaction, what does a partner bring in, what does a partner get back, what can I actually bring from my resources, what do you take as search logic?". And then we go back to the customer journey and say, ideal process, and say along the customer journey, which building blocks do I actually need? And which of these Building Blocks are actually honestly USPs or really super specific and which are actually, if I then break them down, not USPs at all and so standard blocks that I can also share with others? And then you get to a Customer Journey, again a validation model, that's where I come up with something again, that's where I can add something again from the outside. That's why you always have a customer journey picture like this and underneath you have this circle of the SDA again and you look "Ah, I can think of another partner that I can integrate again". We actually use a picture like this to guide us through new use cases.
FR:	Und wie wirkt sich dann dieses Vorgehen auf eure Unternehmensstruktur aus? Das kann ja dann in dem Sinne keine klassische Struktur mehr sein, in der alles divisional eingeteilt ist und die Mitarbeiter ganz feste, strickte Rollen haben, nicht über den Tellerrand hinausschauen. Was ist da bei euch besonders wichtig?
E2:	Also wir haben tatsächlich in der Produktentwicklung, als sowohl im Tech, als auch in der Entwicklung sind wir agil aufgestellt, wirklich auch „by the book“ mit hohen Freiheitsgraden für die Product Owner. Wir haben konsequent das Thema Weiterentwicklung von Services mit reingebracht. Wir haben ein Pricing Model in dem wir sagen es gibt eigentlich keine Wartung. Weil Wartung ist irgendwie so Tod und Verwesung und Nichtweiterentwicklung des Services sondern die Weiterentwicklung ist immer mitgedacht Auftrag des Teams und ist auch im Preis mit abgebildet. Und die Teamzuschnitte diese agilen Teams sind dann tatsächlich wieder domain-driven. Wir sagen die Teams haben eine domain-driven Aufstellung, innerhalb ihrer Domäne ein eigenes Invest-Budget und die Weiterentwicklung ihrer Blöcke, die dann auf der Plattform als Services gemacht werden als Auftrag.
NB:	Dann hätte ich jetzt auch noch einmal eine Frage, und zwar das Thema Service Qualität. Ihr müsst ja immer euch selber wieder hinterfragen und schauen ob das, was ihr leisten wollt aktuell auch wirklich leistet. Wie evaluiert ihr eure eigene Leistung, wie schaut ihr auf die Service-Qualität, die ihr euren Kunden liefert? Aber wie stellt ihr auch sicher, dass der Kunde mit Hilfe der SDA das erreicht, was er erreichen möchte und was ihr ihm auch versprochen habt?
E2:	Also da sind wir bestimmt noch nicht komplett am Ende. Wir haben natürlich Sachen, wenn wir jetzt im App-Bereich was machen, haben wir Kriterien wie Monthly Active Users, nach dem auch Pricing-Updates funktionieren. Also da sind wir finanziell so gesteuert. Wir haben ganz kurzfristig, in die Entwicklung laden wir unsere Kunden mit ein. Das ist durchaus außergewöhnlich für die Branche, dass in diesen SCRUM-Prozess, in diese Reviews die Kunden eingeladen sind. Also alle zwei Wochen können die Kunden tatsächlich mit reinkucken, was ist eigentlich entstanden und können dann auch direkt Feedback geben, quasi in die Weiterentwicklung ihres Produkts. Was machen wir noch? Ja wir machen natürlich Kundenbefragungen, wir überarbeiten unsere Produkte, wir haben für jeden

	unserer Services technisch die Möglichkeit, die Nutzung zu beobachten. Das ist das tatsächliche Qualitätskriterium der Nutzung. Die haben natürlich noch andere Kriterien, die gehen so ein bisschen auf die Qualität des Codes ein. Also das Produkt ist ja letztlich eine Technik und die Codequalität, das ist in der Entstehung bis hin zu automatisierten Tools, die immer wieder kucken wie viel Codequality, gibt es neue Tools, das ist halt sehr wichtig um auch das Produkt wirklich am Laufen zu halten. Gerade weil wir Open Source bauen und ein wichtiger Aspekt von Qualität ist ähm… Nicht nur im Versicherungsbereich… Man sieht's jetzt quasi täglich bei den Erpressern als Beispiel, ist das Thema Security. Was bei uns tatsächlich einen sehr hohen Stellenwert hat ist Security. Wir bauen da ständig aus, auch mit automatisierten Tools, auf der Plattform, auch mit wirklich innovativen Ansätzen mit denen man kontinuierlich die Sicherheit dessen, was man mit der SDA gebaut hat, sicherstellen kann. Das ist wirklich auch gerade zum Qualitätsmaßstab geworden, sich da auch frameworktechnisch ganz anders aufzustellen. Und da sind immer kürzere Feebackzyklen vorgeplant. Wir operieren da nicht nach einer klassischen ISO 271. Also die Zertifizierung brauchen wir auch. Aber wir haben halt Frameworks, die es auch gibt in der technologischen Welt – beispielsweise [inaudible] – das sehr kurzfristig ausgerichtet ist, das aber auch Feedback von den Kunden holt und Feedback von anderen Kanälen. Was uns fehlt, ehrlicherweise, ist das Feedback vom Endkunden, weil unsere jetzigen Kunden, die Versicherungen, überraschenderweise immer noch relativ schlecht darin sind, sich dieses Feedback zu holen. Aber wir machen AB-Testing, Pentesting auf der technischen Ebene, ja wir machen Kundenbefragungen, wir haben selbst Hypothesen abgeleitet in den Produkten, auch Kundenbefragungen, qualitativ gemacht, wie man das im Design Thinking so macht, quantitativ, wir machen Crowdtesting mit den Anwendungen, aber es fehlt wirklich immer nochmal dieses kontinuierliche Feedback. Andere machen das anders. Im eCommerce hast du ständig „Wie war dein…", ständig kannst du Sternchen vergeben, das kannst du in der Versicherungswelt ja noch nicht. Wir haben auch wenig Einfluss darauf, war denn jetzt das Beratungsgespräch mit dem Berater gut und hat da jetzt vielleicht das Stückchen Technologie geholfen? Das ist immer noch sehr stichprobenartig.
NB:	Ja, okay super. Fabian hast du vielleicht noch was, sonst würde ich zum Abschluss kommen? Vielleicht noch in ein paar wenigen Sätzen aus deiner Sicht wo muss die SDA in Zukunft noch, hin was fehlt noch? Was würdest du dir zum Beispiel auch im Rahmen von unserer Forschungsarbeit als Ergebnis wünschen?
E2:	Das ist natürlich immer so ein wenig selbstinteressenhaft getrieben. Das Thema Institutionen und Durchsetzung ist wirklich wichtig. Wir sind damit im Servicekatalog und drum herum unterwegs aber, es ist wirklich imminent wichtig, welche Voraussetzungen über die Wissenschaft und die Logik hinaus, macht es kulturell und auf der Ebene der Regeln einfacher, dass ich sich so ein Ansatz durchsetzt? Ich mein in der Wissenschaftshistorie wäre das die Popper-Forschung, aber es gibt halt vielleicht…. Gar nicht bewusst, historische Modelle oder Heuristiken, die im Kopf dem entgegenstehen. Und da ist die service-dominierte Logik, wenn man sie mal verstanden hat, sehr überzeugend. Aber, du hast es ja selbst gesagt, sie gibt einem ein neues Instrumentarium und auch ein neues Vokabular an die Hand. Aber nicht jeder hat Zeit, das zu lernen. Und da braucht man, glaube ich, letztendlich immer Übersetzungshilfen und Stöckchenniedrigmacher. Und das auf verschiedensten Ebenen. Und das finde ich an der SDL das Gute, dass sie das von Anfang an gemacht hat, ähm… dieses

	Storytelling. Ich glaube, dass sich Modelle sowieso nur über Geschichten und über Beispiele durchsetzen und vielleicht müsste die service-dominierte Logik für sich, weil ich eben über Design Thinking gesprochen habe, nochmal eine eigene Methodik überlegen oder weiterentwickeln, wie man UseCases auf die SDL matched. Also dass man da ein Framework hat. Man darf nicht unterschätzen, wie stark so eine Hilfe sein kann. Ich mein dieses Business Canvas ist das beste Beispiel. Das ist letztlich ein Buch mit ein paar Sachen und auf einmal malen sich ganze Industrien weltweit ihr Business Canvasse auf. Vielleicht gibt es ein SDA Canvas. *And the service-dominated logic, once you understand it, is very convincing. But, as you said yourself, it gives you a new set of tools and a new vocabulary. But not everybody has time to learn that. And that's where, I think, ultimately you always need translation aids and sticklers. And that's on a wide variety of levels. And that's what I think is good about SDL, that they've done that from the beginning, um... this storytelling. I believe that models only become established through stories and examples anyway, and perhaps the service-dominated logic would have to consider or further develop its own methodology, because I just talked about design thinking, how to match use cases to the SDL. In other words, you have a framework. You should not underestimate how powerful such an aid can be. I mean this Business Canvas is the best example. It's a book with a few things and all of a sudden whole industries around the world are painting their business canvas. Maybe there is an SDA Canvas.*
NB:	Den gibt es ja.
E2:	Ich kenne den ja.
NB:	Ja genau, den hat ja eine Absolventin aus unserem Studiengang entwickelt.
E2:	Der ist aber noch nicht [inaudible] und der ist noch nicht so… der setzt voraus, dass man sich in der SDL auskennt. Den Business Canvas kann ich blind befüllen. Das ist ein Unterschied.
NB:	Das stimmt, ja. Da hast du recht.

Appendix 3: Interview Transcript Element 1

Interview Transcript

Interviewee: Element 1, Chairman of Supervisory Board, SDA SE

Interviewer 1: Norman Bächtold, Student, SEB LU, Hochschule Pforzheim

Interviewer 2: Fabian Riedel, Student, Hochschule Pforzheim

Interviewer 3: Peter Weiß, Professor, Hochschule Pforzheim

Date: 21.05.2021

Audio Length: 00:35:00 hrs

Begin of Audio

NB:	Es geht jetzt in den Interviews, die wir mit der SDA SE führen, ein bisschen darum, die Praxis zu überprüfen, wie die SDA in der realen Welt funktioniert und wie sie da abliefert. Wir wollen schauen, ob das, was da stattfindet auch mit der Theorie übereinstimmt, wo es Verbesserungspotenzial gibt und wo die SDA schon auf einem sehr guten Niveau liegt. Mit dir, lieber Markus, wollen wir jetzt, weil wir das große Glück haben, den Erschaffer der SDA vor uns zu haben, erst einmal damit einsteigen, in so ein paar Sätzen von dir zu hören, wie du denn die bisherige Entwicklung der SDA siehst, welche Rolle da auch Use-Cases wie zum Beispiel edith.care in der SDA SE gespielt haben und was auch dein Ausblick für die Zukunft ist. Das würde uns vielleicht für den Anfang mal interessieren, was da dein Blick auf die Dinge ist.
E1:	Also was vorab sehr sehr beglückend ist, ist dass jetzt über die ganzen Jahre sich eigentlich die Kernelemente der service-dominierten Logik, die wir ja übersetzt haben, und die Kernelemente wie funktionieren Service Systems, ja, für Entitäten, wie bauen die über Value Propositions, wie bauen die ihre Connection auf, wie passiert dieser Service Exchange? Dass sich das, was die Theorie da von sich gibt, ja, dass sich das durchgängig bewahrheitet hat und ich noch keine Konstellation gesehen habe bei einer digitalen Transformation, noch bei einer Lösung, wo ich gesagt habe „Mensch, da ist etwas unrund“. Weil normal ist ja die Theorie in der Position, dass sie danach erklärt, warum was schiefgegangen ist in der Wirtschaft. Hier muss man ganz klar sagen, für die Entwicklung von Plattformen und Ökosystemen ist das Erbgut Service Science und SDA goldwert.
NB:	Okay, das bedeutet also: So Themen wie Value Propositions, Cocreation, das sind so Stichworte die wir aus der SDL kennen und die haben sich deiner Meinung nach aber positiv in der Praxis festgesetzt und die werden auch in der alltäglichen Sprache verwendet – dank der SDA?
E1:	Absolut, also zu 100%. Ihr kennt ja den Aufbau der SDA mit der technischen Platte, dann kommen die generischen fachlichen Stacks, dann die individuellen fachlichen Stacks. Und ich sage mal, wenn die Value Proposition – ich glaube Arthur war es in 2009, der gesagt hat: „Wie entstehen Service Innovationen?“, da hat er ja diesen Begriff der „combinatorial evolution“ geprägt, dass man

	Fähigkeiten neu kombiniert und dann immer wieder Innovation prägt. Und ihr werdet auch kennen von Lusch & Nambisan und Vargo und diese service-innovative Deutung der Plattform und des Ökosystems und des Value-Cocreation-Prozesses, das ist da [im Betrieb der SDA SE] eins zu eins die Logik: Immer mehr Resource Liquefecation und Resource Density aufzubauen und dann immer neue combinatorial evolutions und auch Emergenz zu ermöglichen. Bei der Emergenz ist die SDA noch nicht. Sie merkt jetzt aber die Vorteile der Ressourcendichte – also dieser Schritt, der fehlt sicher noch. *And I will say, if the value proposition - I think Arthur was the one in 2009 who said, "How do service innovations emerge?", he coined this term of combinatorial evolution, that you recombine capabilities and then you keep shaping innovation. And you will also be familiar with Lusch & Nambisan and Vargo and this service-innovative interpretation of the platform and the ecosystem and the value cocreation process, which is the logic there [in SDA SE's operations] one to one: building up more and more resource liquefaction and resource density and then enabling ever new combinatorial evolutions and also emergence. The SDA is not yet at emergence. However, it is now noticing the advantages of resource density - so this step is certainly still missing.*
NB:	Also das ist auf jeden Fall schon mal sehr gut, vielen Dank. Ein weiteres Thema sind die Use Cases, die ihr in den letzten bei der Entwicklung der SDA und der Plattform innerhalb der SDA SE vorangetrieben habt. Als Beispiel wären da edith.care oder ai4medicine. Welche Rolle haben diese Use Cases gespielt?
E1:	Du, das ist ein weiteres Mega-Thema. Wenn du heute mit einem Stück Technologie um die Ecke kommst, ja, dann sagen alle Kunden, ob es ein Versicherungsvorstand ist oder ein Unternehmen wie Olympus oder alle anderen – die denken erst einmal wenn es um Plattformen und Ökosysteme geht – die kennen SAP, wo erst einmal nichts unter fünf Millionen Euro geht. Die denken, jetzt kommt jemand und der will dieses ganz große Ticket von dir ziehen und gehen daher gleich mal entsprechend in die Verteidigung dir gegenüber. Und der Mega-Move, den die Technologie jetzt macht, ist einfach über die Cloud-Technologien, dass es jetzt hinbekommst, für wirklich ganz kleines Geld, ja die Basisplatten der SDA, da reden wir von 100- bis 200.000 € jetzt erst mal, dass du die in so einer Plain-Vanille-Konfiguration hinstellst. Und dann kannst du und wir bieten da eben MVPs und dergleichen schon für 50.000 an, wenn man das über unsere Plattform laufen lässt, du kannst da wirklich MVPs machen in der richtigen Technologie, ohne dass du die [im Nachhinein] wegwirfst. Und viele Kunden haben mittlerweile den Schmerz, dass sie sagen „da kommt wieder einer, der macht wieder drei Prototypen und uns gefällt was und dann fangen wir aber danach wieder bei Null an, das in der Zieltechnologie umzusetzen“. Und der Value liegt hier eindeutig dabei, dass du sagen kannst „Hey, wir können von ganz klein ein Use-Case für einen kleinen Mittelständler, können wir genauso wie eine digitale Transformation wie wir es gezeigt haben bei der Signal Iduna“. Natürlich musst du für eine digitale Transformation dann mehr einrichten auf der Plattform. Aber du kannst sehr schnell und auch ganz überschaubar dieses Grundgerüst der SDA wachküssen. Und das war glaube ich sehr wichtig. *You, that's another mega issue. If you come around the corner today with a piece of technology, then all customers, whether it's an insurance executive or a company like Olympus or all the others - they first think when it comes to platforms*

	and ecosystems - they know SAP, where first of all nothing goes for less than five million euros. They think, now someone is coming and wants to pull this really big ticket from you, so they go on the defense against you right away. And the mega-move that the technology is making now is simply via the cloud technologies that you can now manage, for really very small money, yes the base plates of the SDA, we're talking about €100 to €200,000 now first of all, that you put them in such a plain vanilla configuration. And then you can and we offer MVPs and the like for as little as 50,000, if you run it through our platform, you can really make MVPs in the right technology without throwing them away [afterwards]. And many customers now have the pain of saying "here comes another one, he makes three prototypes again and we like something and then we start again from scratch afterwards to implement that in the target technology." And the value here is clearly that you can say "Hey, we can do a very small use case for a small medium-sized company, we can do just as much as a digital transformation as we have shown at Signal Iduna". Of course, for a digital transformation you then have to set up more on the platform. But you can kiss this basic SDA framework goodbye very quickly and in a very manageable way. And I think that was very important. Und als konkrete Antwort vielleicht: Deshalb sind halt einige Kunden, die sagen „wir wollen mehr mit Daten“ und da ist dann der ai4medicine-Use-Case. Und dann haben wir einen Case, das ist die Anbindung betrieblicher Beratung von betrieblicher Altersvorsorge, XEMPUS xBAV, das ist ein Case da siehst du wie das Zusammenspiel zwischen einer Lösung über die Plattform mit den Systems of Records optimiert werden kann. Also jeder Use Case, oder was ich zum Beispiel bei edith.care so liebe, ist wie man einen Prozess grundsätzlich neu macht durch die Möglichkeiten von Plattformen und nicht den alten Bullshit eins zu eins wiederherstellt. Also da hat jeder Use-Case seine eigene Story.
NB:	Ich verstehe. Und bezogen auf den Ausblick – wo siehst du die SDA in der Zukunft. Es gibt ja das Paper Service-Dominant Logic 2025. Sagen wir, wo siehst du die SDA in 2025?
E1:	Ja, also ein Thema an dem wir bei der SDA SE gerade dran sind ist das Thema des Service-Katalogs. Warum? Weil wenn wir es schaffen, SDA 2025 dahin zu bringen, dass wir Business Capabilities, wir nennen es Lego-Bausteine / fachliche Blöcke, dass diese Capabilities letztlich, Teece nennt es ja Capabilities, dass diese Capabilities tauschbar und handelbar werden. Das ist ja eigentlich das, was ich schon – wenn wir mal die IFSD-Webseite anschauen, ja also der Tausch von digitalen Lösungen ist der Handel dieses Jahrhunderts. Dass andere sich andocken auf Basis eines Service-Katalogs, auf den darin, ich sage jetzt mal verbrieften Institutionen als Regeln und Normen, wie wir Software herstellen, wie wir aber auch Gesundheitsdaten und andere Daten austauschen und wie wir vermehrt Capabilities austauschen. Dann wird eine Plattform entstehen, auf der auch das, was jetzt noch fehlt, diese Emergenz, da ist, dass aus ganz neuen Branchen Capabilities dazukommen und quasi wirklich die Business-Bricks, die da sind, eins und eins auf einmal drei Ergeben durch diesen Faktor, das dann einfach noch ein Wert generiert wird. Das wäre meine Hoffnung und mein Ausblick. Wir merken aber auch aus konkreten Anfragen, dass die genau so etwas wird. Die sagen, wir sind bereit, in Standards zu entwickeln wenn wir dann die Ergebnisse tauschen können und auch andere Ergebnisse von euch dafür bekommen.

	Yes, so one topic we are currently working on at SDA SE is the topic of the service catalog. Why? Because if we manage to get SDA 2025 to the point where we have business capabilities, we call them Lego building blocks / functional blocks, that these capabilities ultimately, Teece calls them capabilities, that these capabilities become exchangeable and tradable. That is actually what I already - if we take a look at the IFSD website, yes, so the exchange of digital solutions is the trade of this century. That others dock on the basis of a service catalog, on the institutions that are securitized in it, I'll say now, as rules and standards, how we produce software, but also how we exchange health data and other data, and how we increasingly exchange capabilities. Then a platform will emerge on which what is still missing, this emergence, is also there, that capabilities from completely new sectors will be added and, as it were, the business bricks that are there, one and one at a time, will result in three through this factor, which will then simply generate even more value. That would be my hope and my outlook. But we have also noticed from specific inquiries, that this is exactly the kind of thing they want. They say we are willing to develop standards if we can then exchange the results and also get other results from you in return.
FR:	Du hast jetzt schon mehrfach gesagt, die SDA sei noch nicht bei dieser Stufe, bei der die Emergenz verwirklicht wird. Wie das denn jetzt in der Praxis an einem ganz konkreten Beispiel aus? Also zum Beispiel bei edith.care, wo seid ihr da, oder sind das noch alles diese homopathic resource integration patterns oder aber wann kommt ihr in diesen heteropathic-Bereich rein?
E1:	Also die Beispiele, die wir machen, sind zum Gänsehaut kriegen schön, ja. Also wenn du siehst, warum dockt man betriebliche Altersvorsorge an eine Plattform wie die SDA an, und was sind die Vorteile, was Transparenz betrifft, was Prozess betrifft, wenn du die SDA-Struktur nimmst, dann ist das heute schon richtig gut. Was ich mit Emergenz meine ist, dass es uns noch nicht gelingt, dass wir noch nicht so viele Ressourcen aus der Lebensversicherung, aus der Sach-, aus der Mobilität aus Gesundheit bekommen, dass wir einen generischen, fachlichen Baustein-Katalog haben wie Lego-Steine, wo du dann halt nicht nur den Ponyhof und die Ritterburg sondern auch noch das Piratenschiff bauen kannst. Auf einmal kannst du dann aus dem, was du hast, auch noch ein Piratenschiff bauen, ohne dass du Geld in die Hand nehmen musst. *What I mean by emergence is that we have not yet succeeded in getting so many resources from life insurance, from property, from mobility, from health, that we have a generic, specialized catalog of building blocks like Lego bricks, where you can build not only the pony farm and the knight's castle, but also the pirate ship. All of a sudden, you can build a pirate ship out of what you have without having to spend any money.* Das ist das, was noch fehlt. Und das wird uns auch noch ein paar Jahre, das sage ich auch ehrlich, noch kosten, weil wir da auf noch sehr konkrete Use-Cases angewiesen sind und im Moment freuen sich alle wenn die ODS, also die Datenstrukturen, die generischen Stacks wie Partner, Zustimmung oder Verträge nutzen können.
NB:	Und woran liegt es, dass ihr da noch nicht so weit seid, wie ihr es gerne hättet?
E1:	Ja, weil die Gesellschaft halt, ich sag mal, die ist Ende 2016 gegründet worden, die hat Ende 2018 dann ein Jahr lang eine Due Diligence von der Allianz und der Debeka erfahren. Das legt so einen kleinen Laden relativ lahm. Dann bist du ein

	bisschen weg vom Kunden, das hat uns sicherlich Zeit gekostet. Jetzt kuckt uns ein anderer Großer gerade an, da geht es um Mobilität. Also du musst halt kucken, soll ich jetzt sagen „Lieber eine konkrete Lösung bauen“ oder in das riesige Ökosystem Mobilität mit dem Platzhirsch kommen. Da sagen wir natürlich „Hey, die Chance kommt nur einmal“ und deshalb geht es eben langsamer als das mir als Gründer lieb wäre voran.
NB:	Und welche Rolle spielt in dem Punkt, gerade auch in der Weiterentwicklung und dem Aufbau der Plattform, das Zusammenspiel mit euren Partnern in eurem Ökosystem. Ihr seid ja verbunden mit anderen Versicherungen, anderen Versicherungsdienstleistern aber natürlich auch mit den Versicherten. Welche Rolle spielen diese Partner bei der Entwicklung? Zum Beispiel auch die Anbindung dieser Partner an die Plattform.
E1:	Also, das Thema digitale Transformation und Kultur ist ja ein ganz wesentliches. Und diesen Schritt, du merkst heute noch, sie sind ein riesen Schritt gegangen. Also wir haben zum Beispiel die Leistungseinreichungsapp der Krankenversicherung ist am Markt etabliert, dass die Kunden einfach ihre Abrechnung darüber machen. Es gibt eine Initiative der SDA [SE], die sagt, das sollte man für alles, was eingereicht wird, machen. Also nicht nur für Rechnungen sondern alle Dokumente. Und dann sollte man es noch mit AI viel mehr aufbereiten und kundenfreundlicher machen, Statustracking und dergleichen. Da sagen alle Partner, die wir jetzt haben, „das finden wir super, das machen wir mit euch zusammen“ und da ist der große Vorteil, Norman. Das ist dann ein generisches Investment und dann teilen die die Kosten. Wenn das Ding zwei Millionen kostet, so eine Einreichungsapp, dann geht das durch vier und jeder bekommt diesen Rahmen und kann dann noch für die App seine eigene Corporate Identity und CX in sein Umfeld anpassen. Und das ist das, was super gut geht. Auf der anderen Seite merkst du bei jedem Thema, bei jedem, der dazukommt, bei jeder Kleinigkeit, wie ein Versicherer noch den anderen beäugt und die Kernangst hat, dass der andere mehr davon hat als er selbst.
NB:	Das heißt aber auch, dass innerhalb des Ökosystems der SDA SE dieses partnerschaftliche Denken, das von der SDL beworben wird, dass das noch ein Punkt ist, der weiter vorangetrieben werden muss. Dass man weggeht von diesem Konkurrenzdenken und eher zu so einem partnerschaftlichen Denken übergehen muss, dass das noch verbessert werden muss.
E1:	Ja, aus unserer Sicht hier als Service-Sciencer muss ich sagen, faszinierendes Beispiel, in einem ihrer älteren Paper haben Spohrer & Maglio mal den Punkt gebracht, dass ja die Service Systeme und Service System Entities miteinander verbunden sind, interconnected, weil Value Propositions, und er hat dann die Unterscheidung gemacht interne Value Proposition innerhalb von Organisationen und externe, die dann aus dem Ökosystem gezogen werden. Und innerhalb der SDA merke ich dann, dass die Value Proposition, die dieses Beispiel generisches Lösungsmuster, die teilen die Kosten. Das funktioniert. Nach außen, der Respekt von anderen, die mitspielen, da bin ich komplett bei dir, das muss noch viel besser werden.
NB:	Wie?
E1:	Indem du a) die Angst nimmst, dass es um Multimillionen geht und b) die Vorteile zeigst und da ist natürlich der große Hebel irgendwann, deshalb setzen wir bei dem Service-Katalog und der SDA auch auf Spotify Backstage, wenn du so Normen, die dich am Markt etablieren, wenn du sowas natürlich kannst wie beim Bezahlen, sowas wie PayPal, bei der Softwarentwicklung sowas wie Backstage für die

	Entwicklertools und den Prozess, dann machst du Meter, denen sich dann eine Gothaer und andere lieber anschließen als wenn jetzt eine kleine SDA um die Ecke kommt und sagt wir entwickeln hier so und so, das solltet ihr auch tun. Da sagen die „Oh, warum sollen wir das machen. Wir haben zehn Mal mehr Programmierer als die SDA“. Und auch das, das muss ich hier gerade noch nachschlagen, finde ich für euch hier so spannend. Embedded, nested institutions. Das spielt die große Rolle - die Institutions passen noch nicht und deshalb ist es ein Job von uns, die Insititutions zu gestalten: Was müssen wir tun, an den Spielregeln, damit es für die anderen so sexy wird, damit es wieder eine Value Proposition wird.
NB:	Und wie ist deine Ansicht: Sollte man versuchen, auf bestehende Institutionen aufzubauen und die dann inkrementell zu verändern oder sollte man einen radikaleren Weg gehen und versuchen, einen großen Teil dieser Institutionen in einem kurzen Zeitraum zu verändern? Wie ist da deine Philosophie?
E1:	Also, das ist ein mega schweres Thema. Ich glaube nur, du solltest die, die in deine Strategie passen, ja, die dich forcieren, die solltest du natürlich übernehmen. Also wenn sich so ein Tool oder Ansatz jetzt durchsetzt, so wie Spotify Backstage, und der passt eins zu eins zur Entwicklung in unserem Ökosystem und unserer Plattform: Haken dran. Was natürlich toxisch ist und wo man höllisch aufpassen muss: Du kommst immer, der größte Bremser sind oft die die IT-Abteilungen in den Unternehmen, mit denen du gerne Geschäfte machen willst. Und du hast auf der strategischen Ebene die Value Proposition und den riesen Vorteil, dann kommt irgendwann die Prüfung durch die IT und dann sitzen da fünfzehn Abteilungs- und Bereichsleiter am Tisch die nur eine Angst haben, dass genau ihr Claim weggenommen wird oder neu aufgestellt wird. Das ist total menschlich, ihr kennt das. Change Management, die üblichen Zyklen, das ist total menschlich, dass die erst mal sagen „Woah hier, was wollen die denn?“. Und deshalb musst du aufpassen an der Ecke, um deine Frage zu beantworten, was du nicht machen darfst, ist die alten schlechten Institutionen, wie man historisch Software entwickelt hat, zum Beispiel, an dem Backstage-Beispiel, da musst du deiner Linie treu bleiben, weil sonst wirst du dein Modell zerschießen. Also das ist eine Mischung aus die, die passen übernehmen und bisschen anpassen gegebenenfalls und andere aber brachial ändern, sonst wirst du nicht die Transformation erreichen.
FR:	Wie könnte man denn das Unternehmen, wenn man sich jetzt einer Plattform anschließen möchte, dahingehend ausrichten, dass die Anknüpfung erfolgreich wird, also dass diese alten Barrieren abgebaut werden im Unternehmen?
E1:	Ja perfekt, da sind wir jetzt, das ist der Stand der für das nächste Jahr bildet. Indem du diese Institutionen in einem Service-Katalog so machst, dass du die größten gemeinsamen Vielfachen mit denen, die du haben willst, also zum Beispiel halt ai4medicine-case, dass du sagst „hey, wie kann man denn die Daten von denen mit denen, die ne Versicherung hat“, dann muss man ganz klar sagen ICD 10-Codes wahren das Gesundheitsbild mit denen die Versicherer sprechen, das passt am besten zu den Ergebnissen, die die Charité-Gruppe in Berlin hatte. Dann einigt man sich mit da auf die Regelwerke und bindet die so zusammen, dass sich der Widerstand, dass es am elegantesten zu lösen ist und dass das Zusammenspiel am besten funktioniert. Das sind die Regeln, die du bestimmen musst und die müssen so sein, dass die anderen leicht mitspielen können. Also auch da, ihr kennt’s: Die Rolle der Institutionen ist die, die Akteure zu verbinden und zu koordinieren.
NB:	Schön. Dann würde ich jetzt mal noch auf einen anderen Aspekt eingehen, eine bisschen andere Perspektive einnehmen sozusagen. Wir sprechen jetzt die ganze

	Zeit von der SDA im Ökosystem der SDA SE, wo du auch mit dabei bist. Die SDA SE ist ne Gesellschaft, die ja auch nach dem Prinzip der SDA aufgebaut wurde. Wenn wir davon jetzt mal ein bisschen weggehen und uns zum Beispiel den Fall eines mittelständischen Anlagenbauers, wie wir sie ja zu vielfachen bei uns in Deutschland haben, anschauen, der jetzt gerade die Not der Zeit erkennt und sagt wir müssen uns digital erneuern und der hört von der SDA und ihm gefällt dieser service-dominierte Ansatz. Wenn der jetzt sagt, dass er auch Service-Systeme und Service-Innovation betreiben möchte, worauf kommt es aus deiner Sicht an, dass ein Unternehmen mit der SDA eine erfolgreiche digitale Transformation erreichen kann – außerhalb von eurem Ökosystem?
E1:	Also bezüglich der Grundlogik hatte ich schon eine ganze Reihe von Gesprächen und da geht auch schon einiges. Deshalb bin ich auch so sicher, dass das passt. Jetzt also hier mal vertraulich, nur damit ihr das habt: Konkretes Beispiel eines riesigen Logistikers hier in Hamburg. Was machen die, als Logistiker, transportieren die Ware, die teilweise zerbrechlich und vergänglich ist. Von daher haben die hier, das ist ihr Kerngeschäft, von unten her die Entity und dann haben wir das durchgespielt mit dem CEO und haben gesagt „Hey, die haben ganz viele angebundene Fähigkeiten wie Sensoren und andere technologische Fähigkeiten, die ermöglichen die Fracht über die gesamte Strecke zu tracken, zu kucken, wie ist die Temperatur in dem Laster drin, wo ist das Ding gerade, welche Zeiten stimmen, welche Zeiten stimmen nicht, wie schnell fährt der den Laster“, ja. Oder Medizintechnik, da habe ich so viele Gespräche geführt. Deshalb kann ich sagen, diese Grundlogik, die ist es erstmal. Und es ist jetzt die verdammt strategische Entscheidung von den Aktionären und von der Führung der SDA zu sagen „Was ist denn die nächstbeste Branche wo wir reinwollen?“. Ich glaube, wenn du die SDL und die SDA bedienst, sollte es Interaktion geben, weshalb das Gesundheitsthema so vielversprechend auch noch über die nächsten Jahre ist. Das wird uns alle noch rocken und begeistern, weil wir noch keine geilen Value Propositions von unseren Krankenkassen erfahren. Und dann glauben wir, dass das nächste Thema Mobility ist. In den nächsten ein bis zwei Jahren wollen wir uns in Health noch richtig breit machen und dann in Mobility reinkommen und dann wollen wir eine nächste Branche kommen, zum Beispiel Logistik. Da haben wir auch Kooperationen, McKinsey sieht mich da jetzt als den, der Ahnung mit Plattformen und Ökosystemen hat und die binden mich da jetzt auch mit ein. Also wenn wir den Schritt Mobilität jetzt haben, dann haben wir Partner wie McKinsey und die machen uns alle Türen auf. Wir müssen nur sicherstellen, dass wir auch delivern können. Weil wir machen den Laden schneller kaputt als wir ihn aufbauen, wenn ich jetzt bei fünf ein Feuer anmache und sage „Wir sind die geilste Veranstaltung in der City“ und dann können wir nicht liefern. Das wäre toxisch. Deshalb: Health, Mobility und dann wollen wir im nächsten Jahr noch den Fuß irgendwo in das nächste Ökosystem bekommen.
FR:	In dem Zusammenhang änderst sich ja dann auch die Serviceerfahrung für den Kunden. Wie siehst du da jetzt die Auswirkungen?
E1:	Also jetzt mal ganz auf der Metaebene dieses Interviews: Über alle Branchen hinweg muss eben dieser Switch, wir kommen ja von der güterdominierten Logik, aber was die Unternehmen alle leicht verstehen ohne die SDL verstehen zu müssen ist, die müssen weg von der produkt-out-Denke, dass sie einfach so ein Produkt ran an die Rampe schieben und der Kunde soll glücklich damit werden, zu diesem wirklichen Gebrauch, zu der Positiverfahrung. Das haben alle Unternehmen verinnerlicht, Fabian. Und jetzt kannst du, jetzt bin ich der der dann noch bisschen

	so als der vertrottelte, zu komplexe Professor da noch irgendwie immer versucht da was rüberzubringen, weil keiner hat Bock auf unsere SDL, muss man ehrlich sein, also nur die diese verstanden haben, die sehen diese Schönheit des Themas. Wenn ich ansonsten komme mit Ressourcenintegration und Service Exchange, da schläft mir schon die Hälfte am Tisch weg, ja. *So now completely on the meta level of this interview: Across all industries, this switch, we come from the goods-dominated logic, but what the companies all understand easily without having to understand the SDL is that they have to get away from the product-out thinking, that they simply push a product onto the ramp and the customer should be happy with it, to this real use, to the positive experience. All companies have internalized that, Fabian. And now you can, now I'm the one who's still a bit of a dorky, overly complex professor who's still somehow trying to get something across, because no one is interested in our SDL, you have to be honest, so only those who have understood it see the beauty of the subject. Otherwise, when I come up with resource integration and service exchange, half of the people at the table fall asleep, yes.* Da muss ich ganz ehrlich sein. Aber was alle begreifen ist, es reicht nicht, ein Produkt an die Rampe zu stellen, du brauchst nen Serviceerlebnis und deshalb verstehen die dann auch den nächsten Schritt, den alle noch mitgehen, mit denen ich spreche: Das ist die Interaktion, dass die das wesentliche Thema ist, mit dem du das Service-Thema gestaltest. Und dann sind sie dabei mit einem Strang schon zu wissen, agiles Arbeiten brauchst du um den Kunden besser zu verstehen und dann kommen die irgendwann an den Punkt, dass es nicht nur die Menschen sind, die sich da besser verstehen müssen, sondern dass du auch diese Agilität in der technischen Interaktion brauchst. Und dann kommst du zu der Schiene, Fabian, Plattformen, wie baut man eine Architektur auf um das zu bekommen. Und wenn das verstanden ist, dann kommen die an sagen, wir wollen mit der Plattform Marktmacht. Andere kommen und sagen, sie wollen den Gebrauchsnutzen und das geile Kundenerlebnis. Dritte wollen den Non-Regret-Move. Die Vierten wollen führende Technologien. Und dann kommst du auf diese Ebene wo du irgendwann hingehen musst und sagst: „Kann ich alles verstehen, aber wenn du das alles willst, dann reicht nicht ein bisschen Fähigkeiten auf ne Middleware zu machen, dann brauchst du eine Architektur, dann brauchst du eine Logik wie die SDL oder was vergleichbares“. Also diese Kaskade an Schritten, die muss ich immer durchgehen und die dauert bei der Laura Kauter, der Vertriebsvorständin der SDA, das dauert eineinhalb bis zwei Jahre, um ehrlich zu sein. Dann kommen die nach drei Monaten zurück, sagen jetzt will es der nochmal verstehen, der nochmal verstehen, dann noch ne größere Runde. Das ist der Weg, den du da gehen musst. *But what everyone understands is that it's not enough to put a product on the ramp, you need a service experience, and that's why they also understand the next step that everyone I talk to is taking: that interaction is the essential topic with which you design the service topic. And then they already know that you need agile working in order to better understand the customer, and then at some point they come to the point that it's not just the people who need to understand each other better, but that you also need this agility in the technical interaction. And then you get to the track, Fabian, platforms, how do you build an architecture to get that. And when that's understood, then they come and say, we want market power with the*

	platform. Others come and say they want the utility and the awesome customer experience. Third want the non-regret move. The fourth want leading technologies. And then you get to this level where you have to go there at some point and say: "I can understand all that, but if you want all that, then it's not enough to do a bit of capabilities on a middleware, then you need an architecture, then you need a logic like SDL or something comparable". So, this cascade of steps, I always have to go through it, and it takes Laura Kauter, the head of sales at SDA, one and a half to two years, to be honest. Then they come back after three months, say now he wants to understand it again, he wants to understand it again, then another bigger round. That's the path you have to take.
NB:	Das hängt jetzt auch zusammen mit meiner letzten Frage. Du hattest ja bereits gesagt, dass ihr nächstes Jahr versuchen wollt, noch ein neues Ökosystem zu erschließen, zum Beispiel Logistik. Was ist aus deiner Sicht wichtig, wenn du mit der SDA in ein neues Ökosystem trittst. Was sind die Erfolgsfaktoren, um solch ein neues Ökosystem aufzubauen, welche Partner suchst du aus, die du anschließen möchtest. Was habt ihr da für Kriterien, wie konzeptioniert ihr die Interaktion mit diesen Partnern. Zum Beispiel, welche Rolle spielen Verträge, solche Dinge… Was siehst du da als elementar an?
E1:	Also leider sehe ich da immer einen bösen Zielkonflikt. Jetzt aus meiner wissenschaftlichen und konzeptionellen Überlegung wäre ich natürlich interessiert, einen weiteren Partner zu bekommen, der als Komplementor möglichst ergänzende Funktionen zu dem ganzen Spiel bringt und sehr weit weg ist. Ja, also es kann nicht weit weg genug sein, damit wir dann neue Fähigkeiten aufbauen. Wenn du aber in Sales-Gesprächen bist und jemandem erklärst, die fragen naturgemäß sofort nach und sagen „What's in for me? Was habt ihr denn schon?". Und dann kommen wir mit unseren Business Capabilities und technischen Platten und dann ist die Technik immer fein und da passen wir auch drauf auf, aber bei den Business Capabilities ist es dann leichter tatsächlich, einen Kunden der näher an diesem Ökosystem, wie wir es schon haben dran ist und mehr wiederverwenden kann, zu gewinnen, als einen, den ich jetzt aus theoretischer Sicht gerne hätte. Also das ist der grundsätzliche Zielkonflikt. Ich glaube aber ganz ehrlich gesagt, dass so Branchen, wie dargestellt, gute Zielbranchen sein könnten. Die zweite Branche, die unheimlich spannend ist, ich bin total begeistert von Olympus. Also eigentlich wäre das auch mal ein embedded Use Case. Die sind irgendwie 2012, 13 und 14, das wisst ihr ja alle, sind die wirtschaftlich untergegangen. Durch den Aufstieg der Digitalfotografie ist denen ja das Geschäftsmodell brachialst weggebrochen. Und die haben sich neu erfunden und machen, glaube ich, mittlerweile 70 bis 80 Prozent ihres Umsatzes mit Medizintechnik, das heißt wenn ihr euch röntgen lasst. Und die gehen auch sehr stark auf dieses Zusammenspiel: Was braucht der Kunde in der Interaktion, was bieten wir an Medizintechnik und wie müssen wir das befähigen, damit der Kunde eben ein super Service-Erlebnis hat, das könnte auch was sein.
FR:	Es gibt ja die Coleman'sche Badewanne und die wurde ja auch von Storbacka (2016) zum Multilevel-Framework adaptiert. Inwiefern ist jetzt sowas für euch in der Praxis relevant oder wo seht ihr Abweichungen zu diesen theoretischen Modellen?
E1:	Also du hast ja unten Micro-Ebene, wo du das Actor-Engagement forcieren musst und wenn du dann oben die Makroebene hast, da sollst du den Zusammenhang unpackagen und tieferlegen. Das ist ja dann oben rechts dieses Service-Ecosystems und Value Cocreation und doch irgendeine Form von Output und Ergebnissen, die

	da rauskommen. In der Mitte ist ja abstrahiert der Begriff Engagement Platform. Von daher ist natürlich in der Mitte dieser Badewanne, das ist ja auch „Why Architecture Matters“, was ich da auch sehr theoretisch rausposaune. Dann ist natürlich eine SDL als SDA operationalisiert in der Mitte als Engagement Platform genau das Bindeglied um die Microlevel und die Resource Integration Pattern festzulegen. Da sind wir wieder bei dem Service-Katalog, ja, um dann auch diese Makro-Phänomene zu erklären. Also ich glaube, auch wissenschaftlich ist das da genau das Mosaiksteinchen, das Storbacka, Böhmann und den anderen Autoren da gefehlt hat.
PW:	Was mir dazu noch einfällt ist auch das Paper von Nunamaker. Da gibt es ein Paper von 2015 „The last mile“. Er sagt es gibt proof of concept, proof of value, proof of use und ich glaube auch wenn man den Böhmann, den Grotherr und die alle mit ihren Frameworks anschaut, die hören genau da auf, wo es interessant wird. Und die Kritik ist, wenn man die SDA weiterentwickelt, kann man sagen, die haben ein Multilevel-Framework, aber die können alles auch deployen. Also es gibt ja auch Entwicklungen, auf die die gar keinen Bezug nehmen. Also Microservices ist ja ein Paradigma, auch dieses Domain Driven Design ist sehr interessant, Use-Case-Entwicklung im Kern und das lassen die alles außen vor. Also das heißt, wenn du jetzt wirklich IT-Unternehmen nimmst und die IT-Abteilung, dann sagen die: „Das ist alles so weit weg“. Und der Nunamaker sagt, auch da müsst ihr rein. Ihr müsst auch in dieses Proof of Use. Und da kann man sagen, dass die SDA den großen Vorteil hat, dass sie mit einem Multilevel-Framework kommt, Designwissen transportiert – also das ist ja auch das interessante, dass du wirklich reinzoomen kannst und sagen kannst, welches Design-Wissen muss ich vermitteln, also müssen meine Leute kennen, mit welchen Methoden und Verfahren müssen die arbeiten und wie können wir dann Lösungen auch zeigen. Und da glaube ich, kann die SDA wirklich noch einen Akzent setzen, dass wir sagen, wir sind eben nicht so hochfliegend sondern da gehen wir sogar richtig rein und setzen auf den Methoden auf, die die Leute in der Praxis brauchen. Weil wenn du dieses Wissen in einer geordneten Philosophie in der Praxis vermitteln kannst, Actor Engagement, sehr abstrakt, aber wo man einfach sagt, Leute, um was geht es und was sind eure Möglichkeiten da; ich glaube, da kann man sich richtig gut positionieren. Sowohl wissenschaftlich, wie praktisch.
E1:	Volle Unterstützung und den Nunamaker schaue ich mir noch an. Danke für den Tipp.

Appendix 4: Interview Transcript Element 3

Interview Transcript

Interviewee: Element 3, CTO, SDA SE

Interviewer 1: Norman Bächtold, Student, SEB LU, Hochschule Pforzheim

Interviewer 2: Fabian Riedel, Student, Hochschule Pforzheim

Interviewer 3: Peter Weiß, Professor, Hochschule Pforzheim

Date: 31.05.2021

Audio Length: 00:29:15 hrs

Begin of Audio

NB:	Wir würden gerne damit einsteigen, wie Sie die Entwicklung der SDA – vor allem von der technischen Seite her, innerhalb der SDA.SE in den letzten 5 Jahren in der das Unternehmen gewachsen ist – wie lief das bisher aus Ihrer Sicht, was lief gut, was lief schlecht? Welche Rollen haben auch die Use Cases in der Entwicklung als IT-Plattform gespielt? Und wo sehen Sie vielleicht auch die Plattform in der Zukunft? Welche Herausforderungen sehen Sie da noch?
KP	Das ist jetzt eine sehr global-galaktische Fragestellung.
NB	Ja, das ist so bewusst gewählt.
KP	Also das grundsätzliche Architekturmuster hat sich bewährt. Was ein Problem darstellt, ist die Integrationsthematik mit dem Data Lake. Es ist, wenn man es umsetzt, ein sehr gutes praktisches Muster, was auch die gewünschten Effekte bringt. Es hat psychologisch das Problem, dass es sich um eine Datenreplikation handelt. Für viele altgediente Entwickler gilt es, Datenreplikation ist zu vermeiden. Gegen dieses Thema müssen wir arbeiten. Ein weiteres Thema ist, inwieweit man die SDA für einfache, kleine Lösungen mit einer geringen Integrationstiefe aufsetzen kann, beziehungsweise wann ist eine höhere Integrationstiefe gerechtfertigt?
NB	Okay, das heißt das Thema Data Lake wird sie vor allem in der nächsten Zeit beschäftigen?
KP	Also fairerweise muss auch sagen, dass die SDA nur einen bestimmten Blickwinkel auf einen Unternehmens-Data Lake abbilden kann. Nämlich den auf die für den Endkunden zentrierte Kommunikation und für Sales Agents und nicht für einen gesamten Data Lake. Aber dieses Gesamtbild „ich habe diesen vollständigen Data Lake eines Unternehmens und ein Teil davon wird von der SDA gesehen“ ist sinnvoll und wird auch von den Häusern, die eben an einem Data Lake mit Hubber unterwegs sind, akzeptiert, verstanden und aufgesetzt. Für andere Häuser, die an solchen Ecken jetzt noch nicht sind, gibt es eben noch immer diese Denkweise „wieso, ich kann doch auch auf mein Quellsystem direkt zugreifen“.
NB	An der Stelle ist für uns auch relativ interessant die Software-Architektur der SDA generell, wir haben uns ein bisschen eingelesen in verschiedene öffentlich zugängliche Dokumente, die die SDA und auch der Herr Warg veröffentlich hat,

	und was uns da begegnet ist, ist die die Aufbauweise der SDA, wir haben da ja eine Basis Platte, eine App-Plate, und wo dann die einzelnen Stacks oben drauf aufgesetzt sind, ist diese Architektur so zu verstehen, dass das einzelne Mircoservices, die miteinander kommunizieren oder ist es eher eine monolithische Architektur , wo dann einzelne Mircoservices mit der Zeit ausgegliedert werden? Weil mit der Funktion der verschiedenen Platten und Stacks sind wir nämlich noch nicht so vertraut.
KP	Okay, also die SDA besteht nur aus Microservices. Das mit den Monolithen kommt erst dann, wenn man die SDA nutzen möchte, um aus einer gewählten Anwendung, also einer etablierten gewachsenen Anwendungslandschaft, zu modernisierende Teile herauszuschneiden und in die SDA zu überführen. Bei uns ist die SDA komplett Microservice-orientiert. Wir haben eine Meta-Architektur, wo wir verschiedene Service-Klassen definieren: Basis, Business und Action-Service, Participation-Service, die im Prinzip einfach Spielregeln darstellen, um innerhalb dieser Architektur ein komplettes Vertical als Microservice zu implementieren. Und dieses Vertical kann sich dann eben auf bestimmte Kernsysteme eines Hauses abstützen, eine eigene Business-Logik mit einbringen. Unsere Definition von Microservices ist ja keine harte Definition, sondern etwas was als „rule-of-thumb“ bezeichnet werden kann. So sagen wir zum Beispiel, dass es eine komplette Datenreplikation über alle Services geben soll. Und die einzelnen Services halten die Daten vor, die sie brauchen, um andere Requests zu beantworten. Sie verfolgen den Request, können andere Services aufrufen, und können damit auch zentrale Replikationen der ODSs nutzen, um eine Auskunft zu geben. Unsere Microservices sind auch ohne head. Ob ein Microservice mit Head kommt oder nicht, ist auch nicht klar definiert. Damit ich eben einen Microservice als vollständige Einheit betrachten kann, die von einem Team verantwortet und organisatorisch entwickelt und betreut werden kann, kommt der Head-Anteil mit zu, das heißt es kommt ein Front-End-Anteil mit hinzu. Das kann ich mircoservicetechnisch so weit denken, nur zur Deploymentzeit, auf einem mobilen Endgerät, wie einer App, werde ich dann nicht einzelne UI-Elemente einfach aktualisieren können. Ich muss die ganze App aktualisieren können. Das heißt an der Ecke bricht zum Lauzeitpunkt das Mircoservice-Veren, und ich muss Frontend-Bibliotheken anliefern und diese direkt in eine App integrieren App. Und genau das macht die App-Plate. Sie definiert einen Rahmen und die Spielregeln unter denen die Frontends entwickelt und geliefert werden, damit ich hinterher eine komplette App liefern kann. Damit haben wir jetzt die App-Plate motiviert und damit erklärt, worum es geht. Es ist einfach eine technische Restriktion. Wenn ich mir das im Web-Portal-Bereich ankucke, dann habe ich es da ein bisschen leichter. Ich kann über Web-Components, als Beispiel, auch komplette Frontends ausliefern. Das ist super, hat aber das Problem, das ich für das Frontend auch eine gewisse Framework-Bibliothek mitnehmen muss. Ich möchte nicht immer alles vom Scratch aus selbst programmieren. Das heißt jede Web-Komponente kommt mit einem eigenen Stack an Java-Script-Frameworks um die Ecke. Dann explodiert der Speicher. Der Browser frisst Speicher und das wird durch die Web-Components noch verstärkt. Dann habe ich beim Endbenutzer Laufzeit-Veren, Memory-Verbrauch, der nicht mehr akzeptabel ist. Also macht es auch da Sinn, mit einer App-Plate gemeinsame Bibliotheken rauszuziehen und dort zu optimieren. Dann haben wir als nächstes die Base-Plate. Die Base-Plate ist im Prinzip die Makro-Architektur, also diese Meta-Architektur von der ich gesprochen habe, plus

	die ganzen Spielregeln, wie ich da drinnen kommunizieren und arbeiten möchte. Dazu gehört der REST-Guide, also wie werden Änderungsoperationen über REST abgebildet. Die werden bei uns immer als Arbeitsaufträge abgebildet, sodass wir mit den Verfügbarkeiten vom Back-End an den Endbenutzer an den entsprechenden Auftragsmetaphern kommunizieren können. Dann haben wir synchrone Kommunikationsrichtlinien, wie die ablaufen, welche Token mitgeben werden. Weil wenn ich mich in einer verteilten Welt mit Microservices befinde, muss ich diese auch vernünftig debuggen können. Ich weiß nicht auf welcher Docker-Instanz in einem Kubernetes gerade meine Request ankommt, weil diese über drei/vier unterschiedliche Services geht, die das auch noch eskalieren. Also brauche ich entsprechende Tokens um einen Stack, die ich sauber filtern kann, um die ganzen Aufgaben abzubilden. Das alles wird auf der Makro-Architektur definiert. Dann haben wir daneben die SharedDev, die im Prinzip den gesamten Entwicklungszyklus abbildet. Das ist alles in der Basis-Platte enthalten, dafür habe ich Best-Practices, Guides, und eine komplette Tooling-Pipeline, mit der ich arbeiten kann. *Okay, so the SDA consists only of microservices. The monoliths only come into play when you want to use the SDA to cut out parts that need to be modernized from a selected application, i.e. an established, grown application landscape, and transfer them to the SDA. In our case, the SDA is completely microservice-oriented. We have a meta-architecture where we define different service classes: Basis, Business and Action-Service, Participation-Service, which basically simply represent rules of the game to implement a complete vertical as a microservice within this architecture. And this vertical can then be based on certain core systems of a company and incorporate its own business logic. Our definition of microservices is not a hard definition, but something that can be described as a "rule-of-thumb". For example, we say that there should be complete data replication across all services. And the individual services hold the data they need to answer other requests. They keep track of the request, can call other services, and can thus also use central replications of the ODSs to provide information. Our microservices are also headless. Our microservices are also without head. Whether a microservice comes with head or not is also not clearly defined. In order to be able to view a microservice as a complete unit for which a team can be responsible and which can be organizationally developed and supported, the head part is added, i.e. a front-end part is added. I can think of that in terms of coservice, only at deployment time, on a mobile device, like an app, I won't then be able to simply update individual UI elements. I have to be able to update the whole app. That means at the corner, at lukewarm time, the mircoservice vertex breaks, and I have to deliver frontend libraries and integrate them directly into an app App. And that's exactly what the app plate does. It defines a framework and the rules of the game under which the frontends are developed and delivered so that I can deliver a complete app afterwards. So now we've motivated the app plate and that explains what it's all about. It's simply a technical restriction.* *If I look at it in the web portal space, I have it a little bit easier there. I can also deliver complete frontends via web components, as an example. That's great, but it has the problem that I also have to take a certain framework library with me for the frontend. I don't always want to program everything from scratch myself. That means every web component comes with its own stack of java script frameworks around the corner. Then the memory explodes. The browser eats up memory and*

	that's compounded by the web components. Then I have runtime-versus-memory consumption at the end user, which is no longer acceptable. So it makes sense there too to pull out common libraries with an app plate and optimize there. *Then we have the base plate next. The base plate is basically the macro architecture, so this meta architecture I was talking about, plus all the rules of the game of how I want to communicate and work in there. That includes the REST guide, so how are change operations mapped via REST. We always map those as work orders, so we can communicate with availabilities from the back end to the end user on the appropriate work order metaphors. Then we have synchronous communication policies on how those go down, what tokens are passed along. Because if I'm in a distributed world with microservices, I need to be able to debug them reasonably. I don't know on which Docker instance in a Kubernetes my request just arrives, because it goes through three/four different services, which also escalate it. So I need appropriate tokens around a stack that I can filter cleanly to map all the tasks. All of that is defined on the macro architecture. Then next to that we have the SharedDev, which basically maps the entire development lifecycle. That's all contained in the base plate, for that I have best practices, guides, and a complete tooling pipeline to work with.*
NB	Okay und Prof. Warg hat in unseren Interview letzte Woche angesprochen – Spotify Backstage, die sie wohl verwenden um sich in Zukunft einen Servicekatalog zu erstellen, für ihn aus seiner Sicht, war der Service Katalog Ziel für die nächsten Jahre, deswegen würde ich gerne die nächsten zwei fragen auf den Service Katalog richten.
KP	Ich würde gerne noch eine Sache zur Architektur ergänzen.
NP	Ja gerne. Entschuldigung.
KP	Wir sind gerade bei den täglichen Doings mehr in diese Ecke reingekommen, dass wir uns in ein verteiltes Daten-Management reinentwickeln. Das heißt, wir hatten davor das Bild der drei Service-Systeme, die miteinander zusammen agieren. Und typischerweise gab irgendwo ein System, welches eine gewisse Daten-Entität ownt und dafür auch verantwortlich war. Jetzt kommen wir aber immer mehr dazu, dass wir erkennen, dass es diverse Datenentitäten gibt, die in diesen drei Service Systemen verteilt sind. Ein einfaches Beispiel: Wo ist der Konsens? Also, Konsens haben die Kunden an ein Versicherungsunternehmen gegeben. Der wird in einem Versicherungsunternehmen in einem Systems of Records verwaltet. Aber da der Gesamtservice von einem Versicherungsunternehmen und einem Partnerunternehmen angeboten wird, muss auch in dem Unternehmen, das über das System of Participation angebunden wird, ein Konsens abgelegt werden. Dann gibt es Konsense, die nur innerhalb der Plattform existieren. Damit habe ich drei Quellen aus denen Konsens kommen kann. Wenn ich dann für den Kunden eine zentrale Übersicht der Daten bereitstellen will, dann muss ich eben plötzlich diese drei Datenungssysteme orchestrieren und mit einander vermischen können. Das heißt diese verteilte Datenung, Daten-Administration wird immer wichtiger innerhalb der Architektur.
NB	Okay, das ist sehr gut. Dann …
PW	Okay, wenn ich da kurz einhaken dürfte – wird das dann selbst entwickelt? Oder gibt's da schon, weil klassischerweise bedeutet Microservice-Perspektive das isolierte Betrachten der Daten. Gibt es da irgendeine Philosophie, die sie da antreibt in der Zukunft?
KP	Das ist eine Kombination aus Basis-Services, UDS plus eines Konfigurationsservices, der uns nachher die Möglichkeit gibt, zu erkennen, aus

	welchen Quellen welcher Datensatz gekommen ist, wohin wir denn zurückspringen müssen beim Schreiben. Es bleibt innerhalb dieses Microservices.
PW	Also eigentlich eine Selbstentwicklung?
KP	Es ist ein Pattern, das sich da heraus entwickelt hat.
PW	Ja okay.
NB	Dann wollte ich nochmal wie gesagt zu Spotify-Backstage kommen. Wie genau nutzen sie das? Wie genau funktioniert die Team-Arbeit? Also ich habe schon rausgehört aus ihrer vorherigen Antwort, dass sie einzelne Teams verantworten für einzelne Microservices, die sie einpflegen. Das heißt Spotify-Backstage, dient ihnen sozusagen als Koordinationsplattform. Verstehe ich das richtig, oder wie genau, womit hilft ihnen Spotify Backstage, was ist der Nutzen für sie?
KP	Mehrere Nutzen. Also. Zum einen hilft es den Entwicklern schonmal, wenn man die Plattform anfassbar und verständlich macht. Ansonsten sehen sie nachher nur einzige Werkzeuge wie ein Git-Repository. Wenn sie eine Oberfläche sehen, und wenn sie sehen wie und ob die Verzahnungen einzelner Services miteinander funktionieren, hilft es ihnen die Plattform als solche zu verstehen. Dann habe ich den Plattform-View, was brauche ich für die Entwicklung? Und dann auch den Plattform-View, welche Fähigkeiten habe ich in der Plattform? Worauf kann ich zugreifen? Was kann ich für meine Lösung verwenden, die ich jetzt bauen will? Was kann ich jetzt verwenden an etablierten Services, auf die ich schon zugreifen kann? Das wird durch den Service-Katalog erfüllt. *Several benefits. On the one hand, it helps the developers if you make the platform tangible and understandable. Otherwise, they will only see single tools like a Git repository. If they see an interface, and if they see how and whether the interlocks of individual services work with each other, it helps them to understand the platform as such. Then I have the platform view, what do I need for development? And then also the platform view, what capabilities do I have in the platform? What can I access? What can I use for my solution that I want to build now? What can I use now in established services that I can already access? That is fulfilled by the service catalog.*
NB	Okay.
KP	Das hat auch Nutzen für die gesamte IT-Organisation. Früher gab es Governance-Abteilungen, die sich um die Architektur, die Standards und um ein standardisiertes und einheitliches Vorgehen gekümmert haben. Diese haben darüber entschieden, wie verschiedene Entwicklungsteams losmarschieren und entwickeln sollten. Das hat in der Praxis unterschiedlich gut funktioniert, weil es auch meistens am Menschen hängt. Es gibt immer eine Abhängigkeit zu anderen Abteilungen, die andere Ziele haben. Das ist dann der systematische, organisatorische Impact, den das da mitreinbringt. Dann gibt jetzt die andere Entwicklung, dass man sagt man hat agile Teams. Wenn ich an eine agile Organisation denke, dann habe ich jetzt ganz viele agile Teams, die mit einer eigenen Zielsetzung unterwegs sind. Innerhalb eines agilen Kontextes kann ein Team nur dann vernünftig arbeiten, wenn es die Sachen verantworten und beherrschen kann. Ein agiles Team, welches eine Abhängigkeit nach draußen hat, hat sofort einen Blocker im Backlog, wenn kein anderer reagiert. Das ruiniert den Sprint-Rundown. Also wird man schauen, dass man da rauskommt. Und was wir momentan an ganz vielen businessgetriebenen Streams sehen ist, dass die Autonomie der Teams sehr gestärkt wird, zulasten von unternehmensweitem Konsens. Was ist Architektur, was ist ein einheitliches Auftreten, was ist ein

	einheitliches Qualitätsversprechen? Das sind Sachen, die leiden, wenn ich jetzt diese institutionellen Regeln reinbringen will in die Teams. Da brauche ich dann wieder dort andere Unterstützung in den Teams, bevor diese wieder langsam zu werden, weil sie zu viel miteinander kommunizieren müssen. Das ist dann einer der Punkte, wofür dann auch Backstage bei Spotify gegründet wurde.
NB	Das macht Sinn, ja.
KP	Und das geht dann auch nicht mehr über einen Anweisungscharakter – „mach es genau so“ - weil dann würde ich ja von dem Freiheitsgedanken der Agilität abweichen. Sondern da gibt es dann diesen Golden-Path-Elemente. Also einfach nur heruntergebrochen aus dem Bereich des Scaffholdings, wie ich mir ein neues Projekt-Skelet aufsetzte, baue ich dann auch im Backend Source-Code-Repositories, Bildstrukturen, und so weiter, weil alles automatisch mit hochgefahren wird, werden dann in einem Scaffholding mehre Projekttypen angeboten. Dann sieht man genau die goldenen Projekte, die bitte bevorzugt werden sollen. Dafür bekommt das Team vollständigen Support. Für die anderen sind sie selbst verantwortlich, dürfen dafür aber keine Unterstützung erwarten. Also da geht darum, die Leute in eine bestimmte Richtung zu treiben.
NB	Also, das geht dann auch schon in die Richtung Koordination zwischen Business- und IT. Worauf wir auch zu sprechen kommen wollten ist grundsätzlich die Frage, wie Sie zusammen mit den Business-Strategien, also vor allem mit dem Stephan Hans als CEO bei der SDA.SE zusammenarbeiten. Wie übersetzten sie die Businessziele in die IT-Strategie, oder auch andersrum, wie informiert die IT-Strategie bei der SDA.SE die Business-Strategie? Wie arbeiteten Sie da zusammen, und wie kommunizieren sie dann auch die Geschäftsstrategien an die Entwickler?
KP	Wir sind ja eine kleine Unternehmung, also können wir noch sehr viel über die direkte Kommunikation abbilden. Nichtsdestotrotz sind dort auch Kommunikationsverluste. Wir haben einen regemäßigen Austausch zwischen der Business-Seite und der IT-Seite. Die IT-Seite ist auch gleichzeitig bei den Sales-Prozessen miteingebunden, um eben dem Kunden die Lösung zu erklären, wie sie technisch funktioniert, was wir wie abzustimmen haben. Wir haben dann auch bei den entsprechenden Product-Ownern Diskussionen mit den Kunden und den Stakeholdern. Da sind dann auch immer IT-Leute mit dabei und gestalten die Lösung mit. Also ich habe deswegen nicht nur einen technischen Blick auf die Plattform, sondern auch einen inhaltlich-fachlichen.
NB	Dieses Empowerment trägt dazu auch bei, dass sie ihre Teams, das hat der Professor Warg auch angesprochen, sie ihren Teams so einen gewissen Freiheitsgrad geben, ich weiß nicht mehr genau die Aufteilung, aber er meinte so was wie 30-70, das die Teams selbstständig Ziele stecken könne und dran arbeiten können, sie meinten ja auch gerade, dass mit den Golden-Path, aber daneben können sie auch andere Projekte umsetzten, ist das Teil ihrer Strategie, die Teams mehr einzubinden?
KP	Ja also der Hauptunkt dabei ist, wer es nachher ausbaden muss, der trägt die Verantwortung, und darf entscheiden.
NB	Wir verstehen die SDA ja als eine Plattform innerhalb ihres Unternehmens, die läuft, und dann kommen ja z.B. mit einem Use Case wie edith.care neue externe Akteure hinzu. Wie planen Sie, wie gehen Sie grundsätzlich dabei vor, ein Projekt zu planen und umzusetzen?
KP	Zuerst muss die Vision der Lösung nachher entwickelt werden. Was ist da das Bedürfniss des Endnutzers, wo sind seine Probleme, wie kann man ihm dort

	helfen? Da ist typischerweise so was wie Design Thinking ein Hilfsmittel, um erst einmal den Problem-Raum zu identifizieren. Wie kann man ein Problem lösen? Dann starten wir damit, die Lösung beim Kunden aufzubauen. Um zu sehen, ob die Idee überhaupt funktioniert und gut trägt. Da ist auch der größte Effort, das erstmal aufzubauen. Die Integration in andere Teilnehmer des Ökosystems kommt dann ein Stück später. Weil man kann dann dem Endkunden erst einmal etwas zeigen, das ihn zufrieden stellt und kann lernen, ob dieses Feature funktioniert. Da kann es zunächst einmal nur darum gehen, von einem System E-Mails zu verschicken und zu empfangen. Dann habe ich erst einmal meinen manuellen Prozess, aber ich bin ja auch in einer Ramp-up-Phase. Ich produziere eine Lösung, gucke wie sie angenommen wird und wie sie funktioniert, hole mir Endbenutzer-Feedback ein, was richtig ist und so weiter. Dann kann gezielt auch anderes beim Kunden argumentieren und sagen „Hey Leute, hier lohnt es sich, das Gemeinsam zu machen. Wir brauchen Automatisierung und Integration.“ Gerade dann, wenn ich mit größeren Häusern integriere, wird es auch aufwändiger. Es müssen mehrere Leute an Bord geholt werden. Meistens hat man mit es mit 5/6 Abteilungen zu tun, mit denen man reden muss - sowohl auf technischer als auch auf organisatorischer Seite. Und da hilft es dann auch immer, wenn man ein bisschen Gewicht hat, worüber man redet, was das bedeutet, und man ein bisschen mehr Erfahrung hat. Und da hilft auch das Wissen, ob die Lösung überhaupt angenommen wird. *First of all, the vision of the solution must be developed in the beginning. What is the end user's need, where are his problems, how can we help him there? Typically, something like design thinking is a tool to first identify the problem space. How can you solve a problem? Then we start by building the solution at the customer's site. To see if the idea works at all and carries well. That's also where the biggest effort is, to build that up first. Integration into other participants in the ecosystem then comes a bit later. Because you can then first show the end customer something that satisfies him and learn whether this feature works. Initially, it may be just a matter of sending and receiving e-mails from a system. Then I have my manual process first, but I'm also in a ramp-up phase. I produce a solution, see how it is accepted and how it works, get end-user feedback on what is right and so on. Then specifically, I can make a different argument to the customer and say "Hey guys, this is where it's worth doing this together. We need automation and integration." Especially when I integrate with larger houses, it also becomes more involved. You have to get multiple people on board. Most of the time, you're dealing with 5/6 departments that you have to talk to - both on the technical side and on the organizational side. And that's where it always helps if you have a little bit of weight about what you're talking about, what that means, and you have a little bit more experience. And it also helps to know whether the solution will be accepted at all.*
NB	Ja, okay! Jetzt ist die Zeit vorbei, wenn noch eine Frage möglich wäre, wäre das super!
KP	Okay
NP	Okay, das Thema Qualitätskontrolle. Welche Stellschrauben haben Sie, auf die Sie blicken? Gerade bei der Überwachung der Performance Ihrer Services, worauf schauen Sie da? Und wie verwenden Sie die Informationen, um Ihren Service zu verbessern?
KP	Wir haben die ganze Zeit Live-Metriken, die von den Services erhoben und berichtet werden. Wir haben Request -Ablaufe innerhalb der Plattform, und es werden auch entsprechende Grenzwerte daraufgesetzt, wann je ein Service

	funktionieren darf, und wo es eben Alerts geben soll, wo wir dann genauer hinschauen müssen, um ihn zu verbessern. Ein feinmaschiges Monitoring.
NB	Und menschliches Feedback spielt das auch eine Rolle? User Feedback?
KP	Ja, das war was ich gerade im Case davor meinte, dass wir das Feedback vom Nutzer holen und gucken, wie die Lösung ankommt, ob sie gut ist. Da gibt es da auch die Themen, dass ich Benutzer aktiv darauf anspreche, dass sie eine Bewertung geben. Wenn ich erwarte, dass sie das machen, machen das nur die Leute, die sich geärgert haben und nicht die Leute die positiv gestimmt sind. Denn die kommen nicht auf die Idee, Feedback zu geben. Dazu kommen die Leute immer nur, wenn sie sich geärgert haben. Also muss man das Ganze immer proaktiv einfordern, um das mit positiven Bewertungen nach oben zu ziehen, und noch ein paar Sachen herauszukriegen. *Yes, that's what I meant in the case before, that we get feedback from the user and see how the solution is received, whether it's good. There are also the issues that I actively ask users to give a rating. If I expect them to do that, only the people who are annoyed do that and not the people who are positive. Because they don't get the idea to give feedback. People only ever come to that when they're annoyed. So, you always have to ask for it proactively to pull that up with positive reviews and get a few more things out of it.*
NB	Okay, ja super! Herr Panier vielen Dank für die Zeit, die Sie sich genommen haben. Das war wirklich aufschlussreich. Dann, wie gesagt, wenn die Forschungsarbeiten vorbei sind, lassen wir Sie auch sehr gerne daran teilhaben, was dabei herausgekommen ist. Und dann wünschen wir Ihnen noch alles Gute für die Zeit, und bis dann!
KP	Okay, danke auch tschüss!

Appendix 5: Coffee Talk Notes Element 4

Coffee Talk Notes

Interviewee: Element 4, Managing Director, ION Access & Health GmbH

Interviewer 1: Peter Weiß, Professor, Hochschule Pforzheim

Interviewer 2: Norman Bächtold, Student, SEB LU, Hochschule Pforzheim

Interviewer 3: Fabian Riedel, Student, Hochschule Pforzheim

Date: 19.05.2021; 21.05.2021

Conversation Length: 2 x 1:00:00 hrs

Memo conversation No. 1 with Element 4
4 Players: 1) Travelers: High demand for information - must be ensured via digital systems 2) Governments: Want to protect population health and economy. Must develop rules and communicate them to travelers, event organizers, etc. 3) Medical providers: Have services locally. How can data be made available securely internationally and in different languages? Economic interest represented. Immunizations/vaccination records: electronic medical record not available everywhere. Need to be linked together. 4) Service providers: Want to offer restaurants, start events, etc. and need to know if people are protected → Platform must ensure that all stakeholders get their interests represented in compliance with data protection and local laws. Ethical component not to be neglected (do not enrich with data provision etc.). Must also be suitable for children and the elderly. → Result: Global platform with decentralized storage locations to take local legislation into account. Data records can also be moved (important for expats, for example). UAE as an important partner for using the platform because many expats live there, which means the country has many contacts in the world. This may allow SimplyGo to spread quickly. → What does the implementation of the solution look like? What will be done? Every actor has a customer journey that is designed differently. For example, a traveler: Every traveler is used to the process of booking a trip. SimplyGo wants to use these familiar patterns in the design of the user experience and thus pick up on familiar processes. For example, the customer should be able to specify that he wants to travel from DE to UAE on a certain date, and then the rules that apply to this period should be queried and delivered. These must also be updated at any time (similar procedure to booking portals). The same applies to medical providers: Test procedures are always, worldwide the same (unique ID of sample can be linked to person by lab/doctor).

SimplyGo does not want to intervene in existing processes (e.g. with barcodes), but to start with the existing and to retrieve and forward the number in connection with name and date of birth.

→ Thus, concrete patterns can be derived and mapped in a systematic flow (e.g. of the data used)

Example of SITA: Collects data from passengers and forwards it to immigration. Data stream can be expanded to include SimplyGo data. In this way, Immigration receives the passenger's health data together with the normal passenger data (suggestion by Prof. Weiß: PayPal).

→ The principle of the platform is encrypted end-to-end communication. This means SimplyGo can also serve as an intermediary and pass on data to companies such as SITA via the platform and its ecosystem as a provider.

→ Flexibility through local data storage. SimplyGo can keep a user's data in sync across multiple server locations, providing faster and easier service, e.g., when a customer travels from DE to UAE.

Memo conversation No. 2 with Element 4

AE at SimplyGo:

- People do not like change - getting used to online bookings. Similar procedures to follow and implement your own processes. Same for medical providers, immigration agencies, etc.
- Service innovation 1) in recognizing pain points of the customer in the engagement and 2) also bringing concrete solutions.
- Service design is about simplicity. Internal, cross-functional collaboration in teams is also important for this.

How does cooperation with customers work?

- With customers, close cooperation, asking many questions, monitoring processes, communication is very important, information must be in the right place at the right time. The latter is considered very important in data-based business models like insurance.
- Problem in healthcare: Stalled processes. These need to be broken up in small but steady steps. Particularly in collaboration with government agencies, there are also political difficulties, e.g., in billing for services in the insurance sector with strong lobbies that do not want transparency in the system. Patience is needed here (= institutionalization through micro-innovations). The GDPR is different: its precise requirements allow users to be more involved in the data exchange process (= resource integration).
- Ottonova sides with the customer in this trench warfare. With Health X at Ottonova, feedback is collected via HealthX. Out of 100 customers, 3-4 participate and provide feedback. Not yet implemented at SimplyGo. But a similar feedback solution is planned. Would then have to be institutionalized. Philosophy at Ottonova for interfaces with customers: Product owners take on a strong role internally. Here, internal coordination is important again, joint communication and interface-related work in the frontend play an important role.

Selection of technical solutions: As simple as possible. For example, blockchain is considered too complicated by Ottonova. They want to keep it simple.

Development Methods used:

- Agile / DevOps (for frontends used by SimplyGo) / standard systems in connection with official bodies due to BaFin license.
- For cross-functional teams: 30% independent responsibility and "only" 70% directly tied to company goals.

When is a service successful?

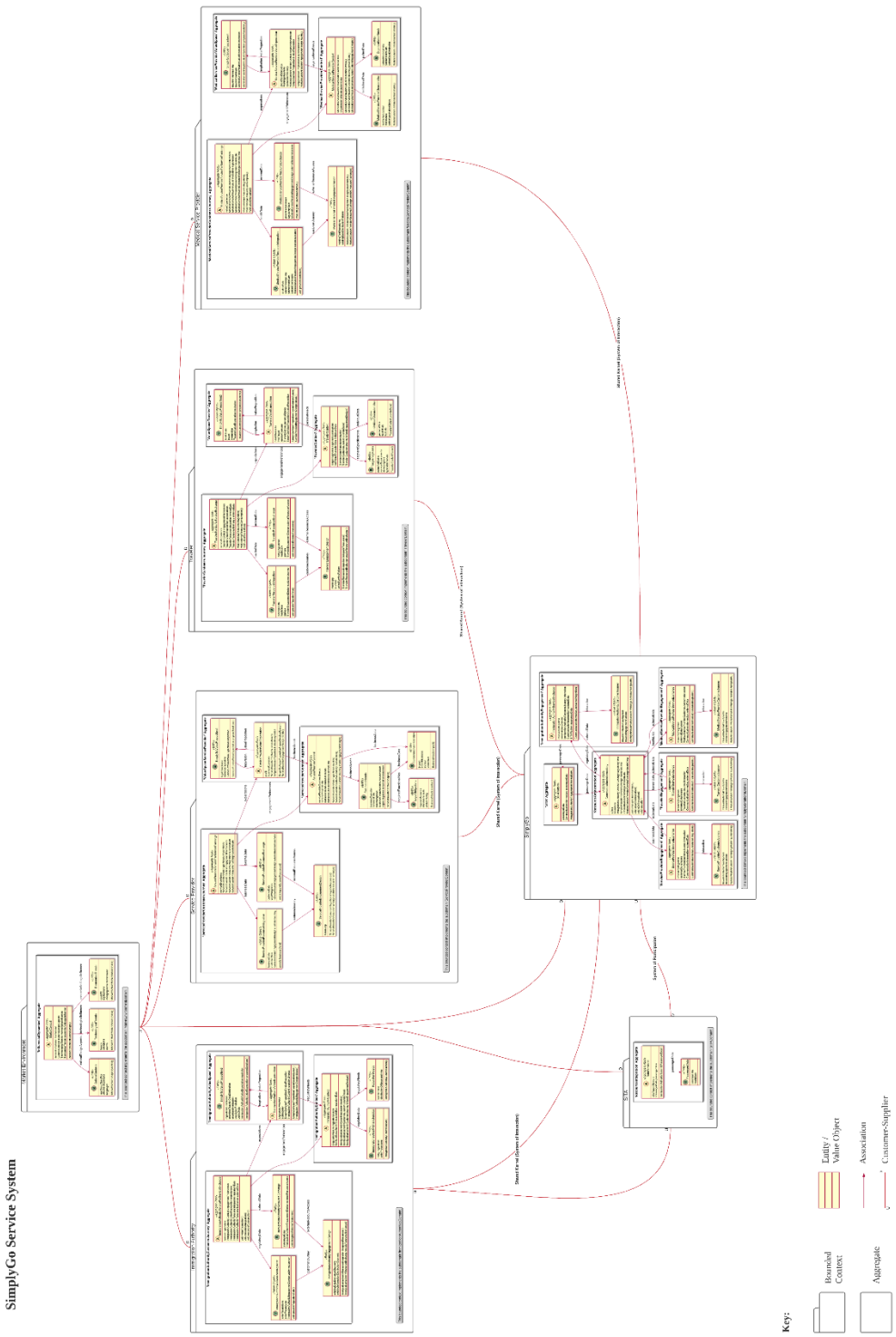
- For SimplyGo when contract with UAE is closed
- Look thereafter - very complex market environment, therefore difficult to predict.
- At Ottonova, a service is successful when customers are satisfied. Use of feedback measurements / customer calls / NPS scores Various satisfaction KPIs. Attempts are made to ensure that promises are kept.

How does roll-out of new services work?

- If it is complicated, there is an explanatory video or other interactive guidance.
- Information regarding data security etc.

Appendix 6: Complete View of The Service System Design Framework[6]

Fig. 18: Complete View on the SimplyGo Service System

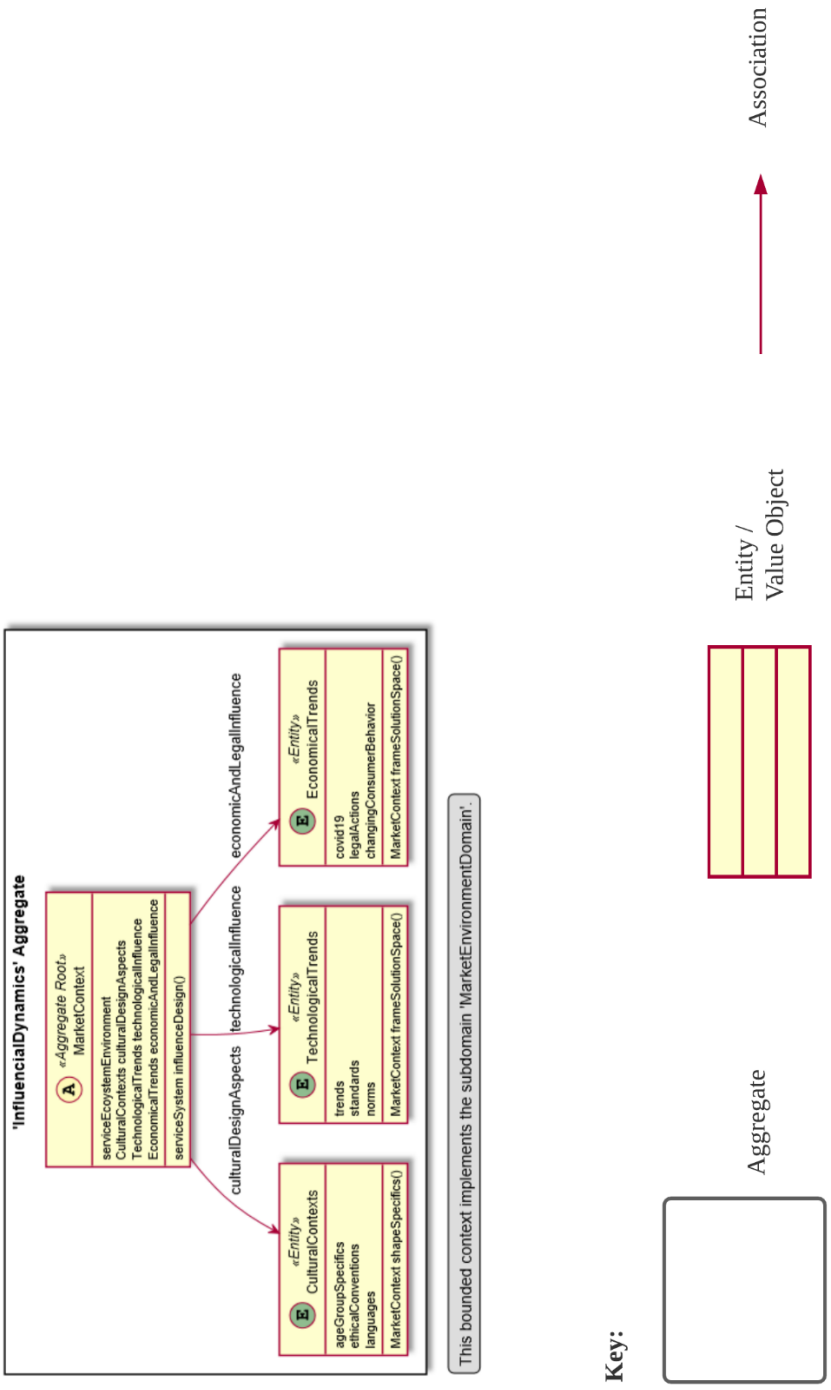


Own work.

[6] The size of the framework does not allow to be readable in this work. Please refer to the provided GitHub repository for a printable, high-resolution version (https://github.com/NormanBaechtold/ServiceSystem).

Appendix 7: Bounded Context “MarketEnvironment”

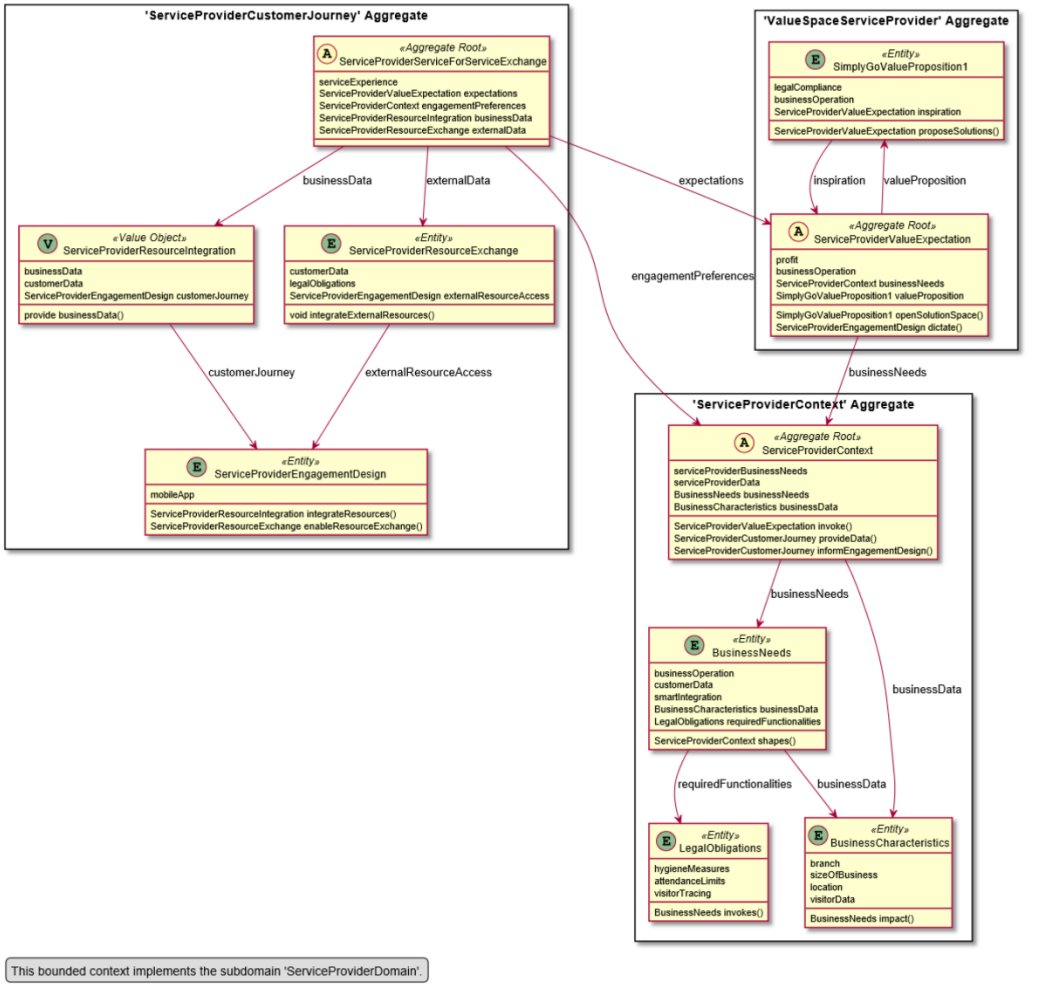
Fig. 19: Bounded Context of the Market Environment



Own work.

Appendix 8: Bounded Context “ServiceProvider”

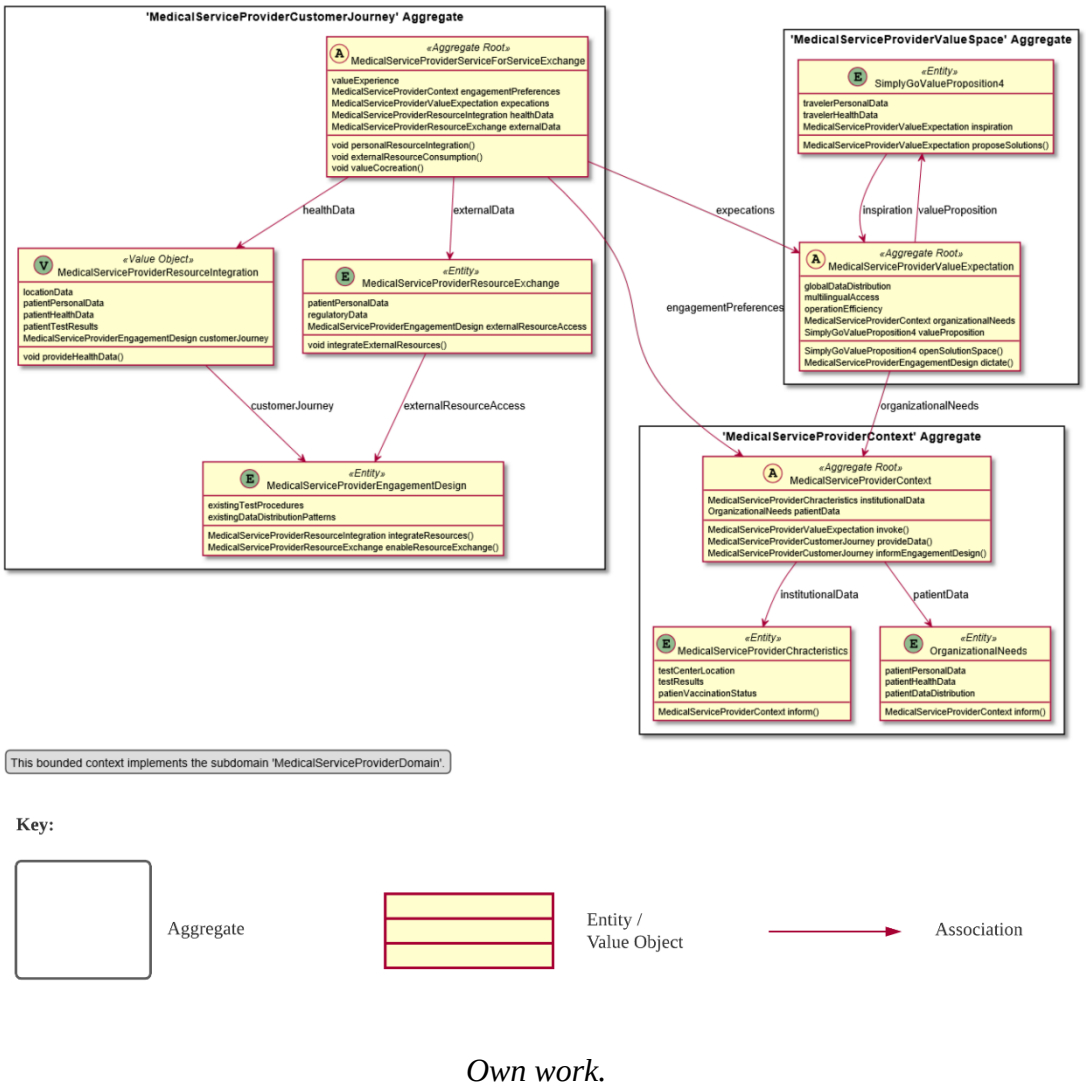
Fig. 20: Bounded Context of the Service Provider



Own work.

Appendix 9: Bounded Context “MedicalServiceprovider”

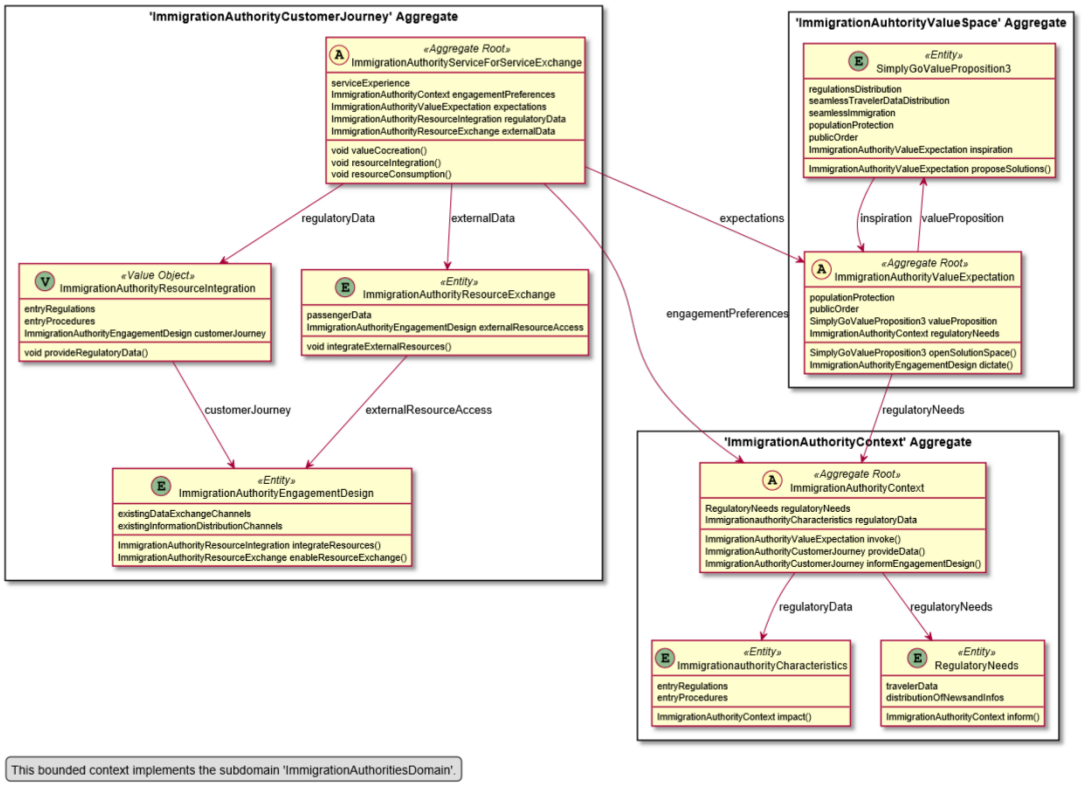
Fig. 21: Bounded Context of the Medical Service Provider



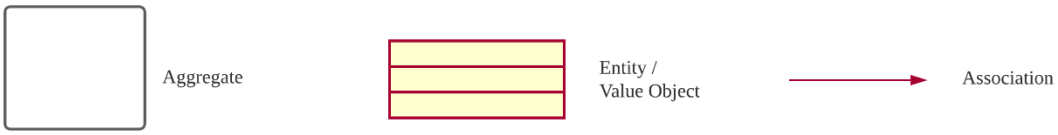
Own work.

Appendix 10: Bounded Context “ImmigrationAuthorities”

Fig. 22: Bounded Context of the Immigration Authorities



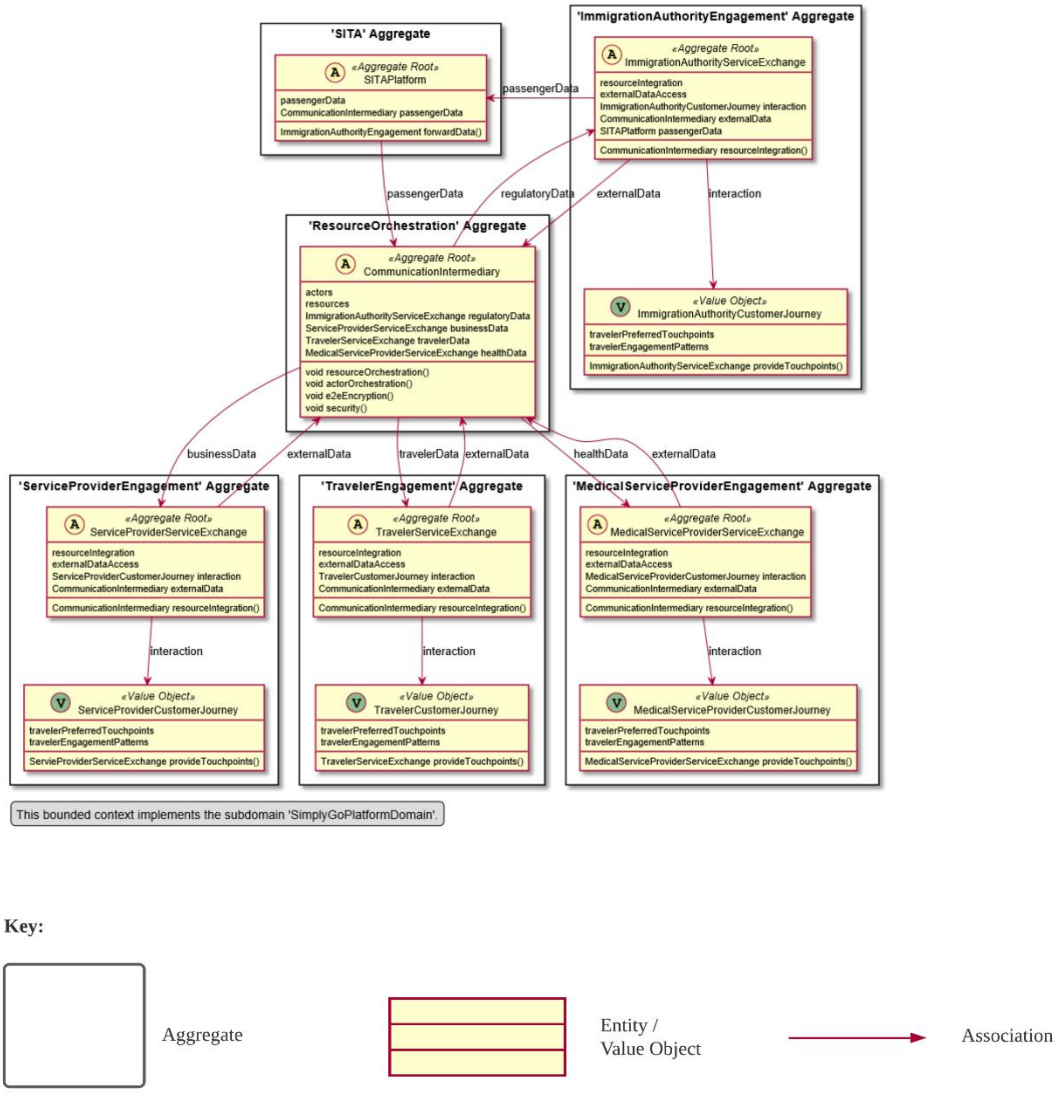
Key:



Own work.

Appendix 11: Bounded Context “SimplyGo”

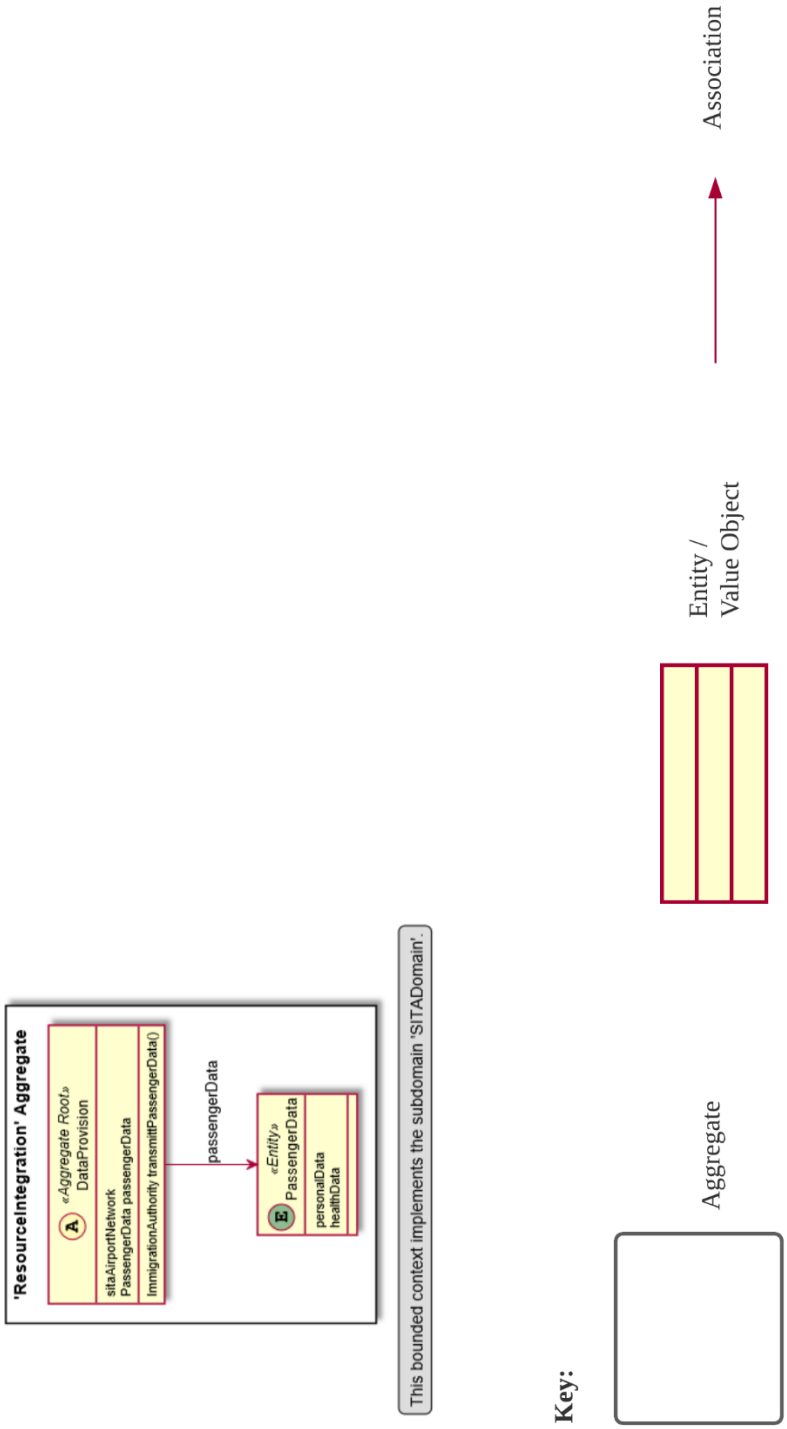
Fig. 23: Bounded Context of the SimplyGo Platform



Own work.

Appendix 12: Bounded Context “SITA”

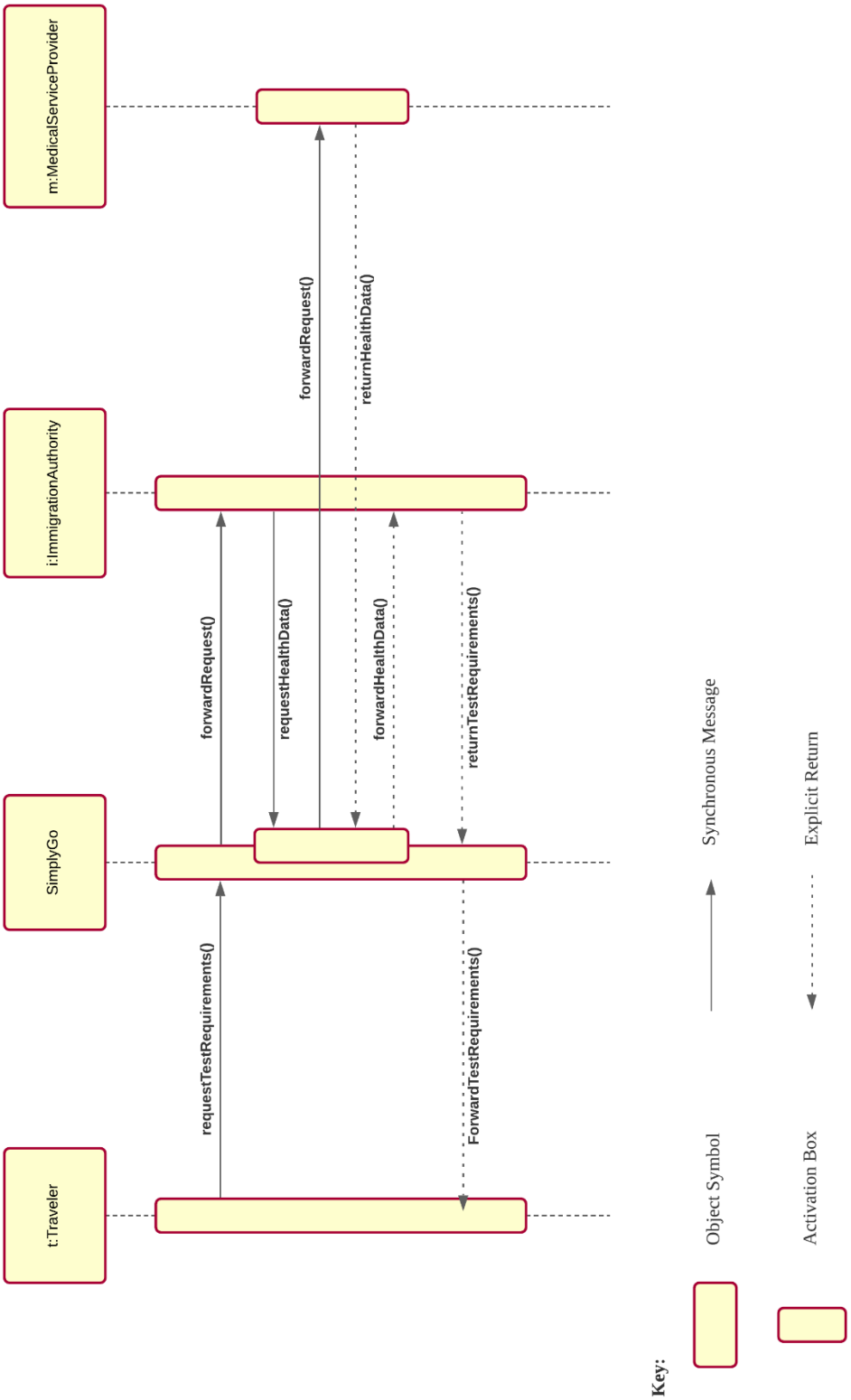
Fig. 24: Bounded Context of SITA



Own work.

Appendix 13: UML Sequence Diagram for Requesting Test Requirements

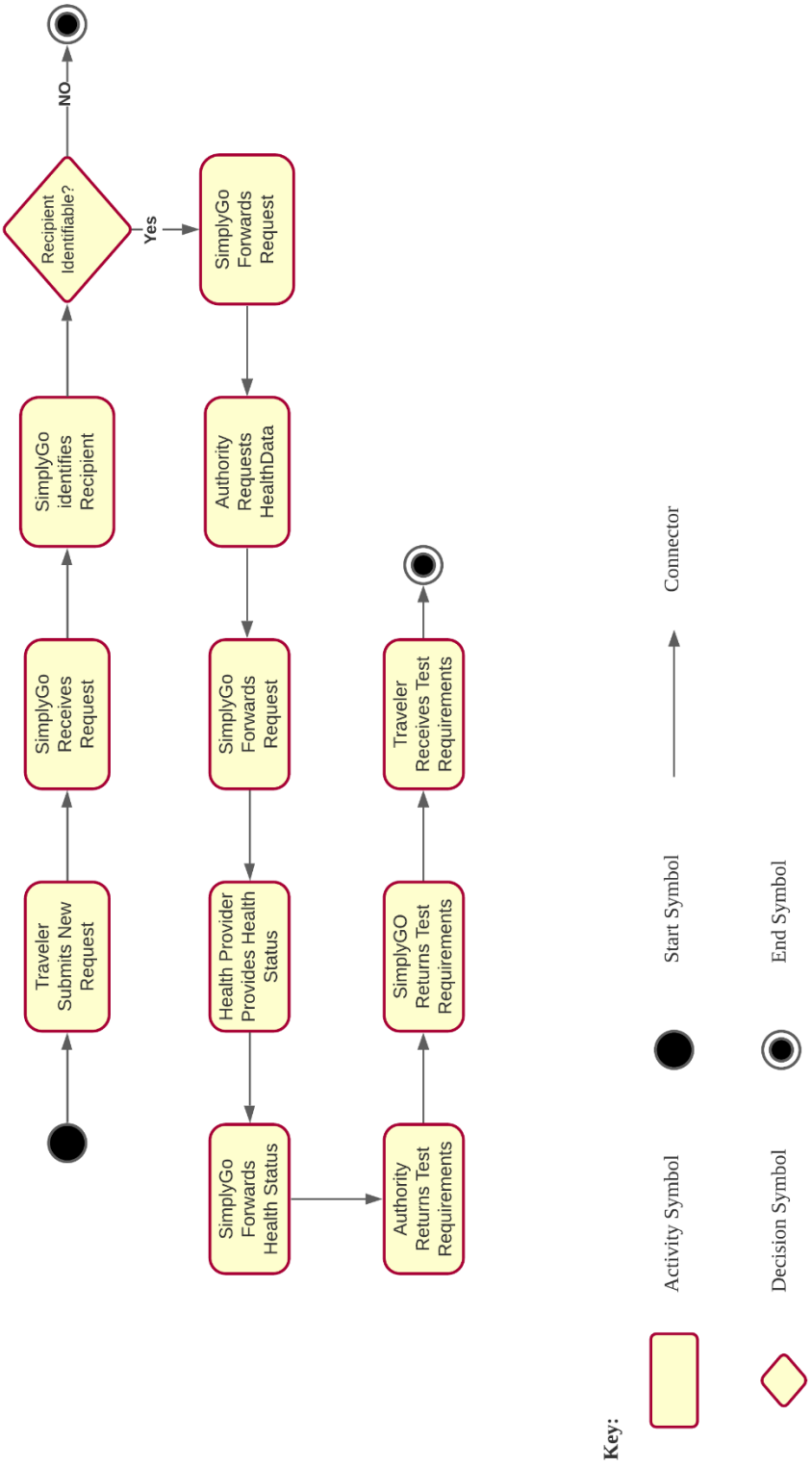
Fig. 25: UML Sequence Diagram for Requesting Test Requirements



Own work.

Appendix 14: UML Activity Diagram for Requesting Test Requirements

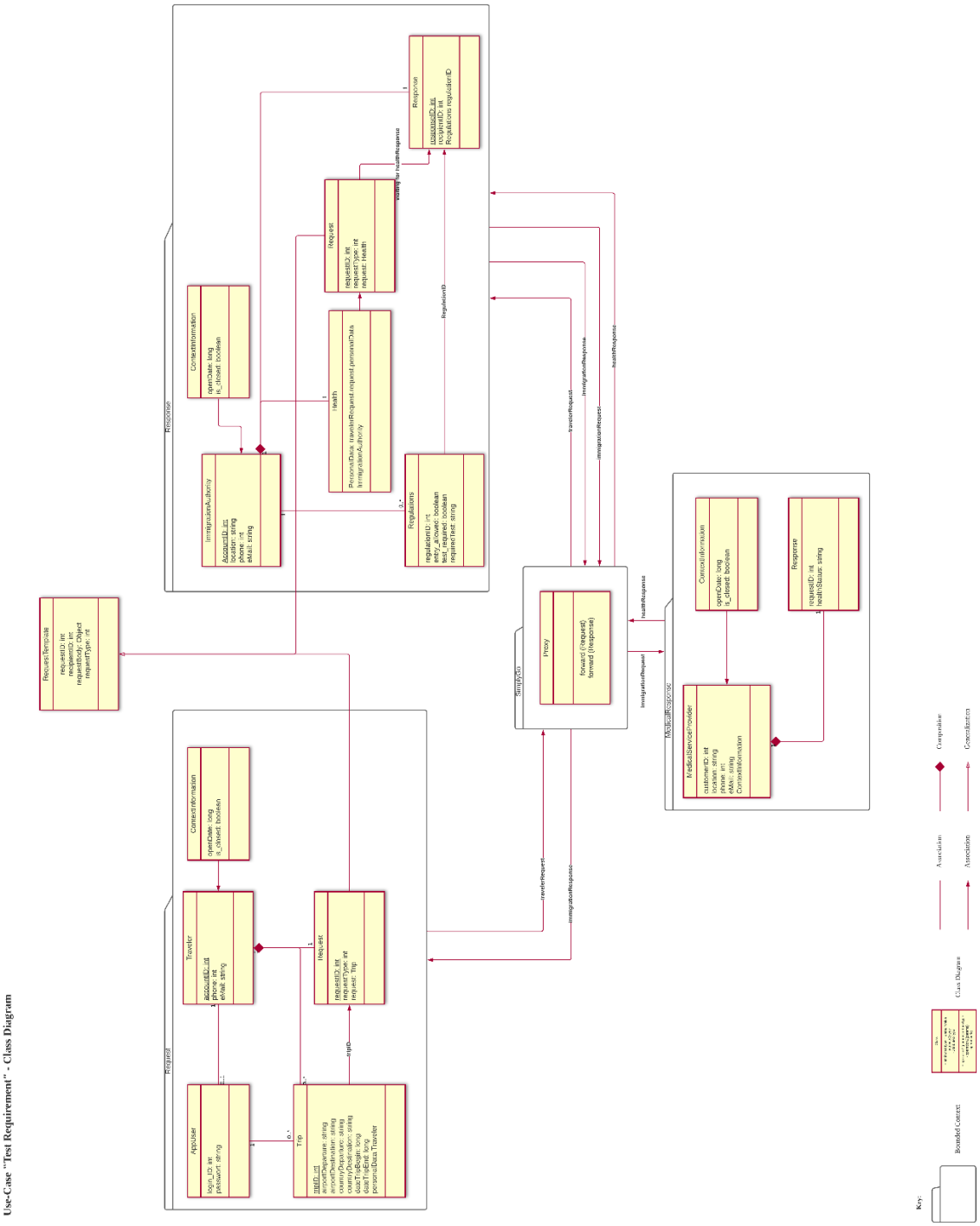
Fig. 26: UML Activity Diagram for Requesting Test Requirements



Own work.

Appendix 15: UML Class Diagram for Requesting Test Requirements[7]

Fig. 27: UML Class Diagram for Requesting Test Requirements



Own work.

[7] The size of the class diagram does not allow to be readable in this work. Please refer to the provided GitHub repository for a printable, high-resolution version (https://github.com/NormanBaechtold/ServiceSystem).

Appendix 16: Tools and Methods for Identifying New Value Propositions

Table 9: Tools and Methods for Identifying New Value Propositions

Methods for *Formulate Problem*-Phase		**Institutional Design Cycle**
Area of Analysis	**Method Name**	**Short Description**
Service System	Nightmare Competitor	Discover potential weak spots of the digital service to be produced and to unearth innovation and disruption possibilities for one's business model by sketching the worst possible rival for one's firm.
	Stakeholder Analysis	Determine and classify key stakeholders and their needs to evaluate the need for external stakeholders to be included in the development process.
	Stakeholder Map	Knowledge of how a value network impacts service delivery to the consumer is acquired by visually portraying the stakeholders who influence the development of the service.
	SWOT Analysis	Internal strengths and weaknesses are compared to external opportunities and dangers to determine the need for action and potential.
	Environmental Analysis	Collection of all framework conditions, influences, and factors in the context of digital services to identify factors that promote and inhibit development.
SOI (Customer-Centric)	Interview for Empathy with "5 Why's"	Survey of potential stakeholder groups for in-depth exploration of requirements for the service to be developed, including framework conditions. Helps to identify innovation potential and a person's beliefs, motives, and difficulties.
	MoSCoW Prioritization	A four-level priority scale is used to arrange and prioritize a vast number of customer-related needs, goals, or activities into important and less crucial categories to identify requirements that are critical for success.
	Personas	Fictional, yet real human profiles are created to enhance empathy for a certain user or stakeholder group inside the service ecosystem.
	Shadowing	Accompanying a representative of an actor group in their daily (work) life to understand work procedures, issues, and contexts.
	Pains & Gains	To identify and visualize the goals and problems of stakeholders, they are systematically observed with a focus on value generation for the target group.
SOOR (Internal Focus)	Brainstorming	Open and creative collection of as many ideas as possible in a very short time, incorporating different perspectives by jointly developing ideas in the team.
SOP (Partner-Centric)	Environmental Analysis	Collection of all framework conditions, influences, and factors in the context of digital services to identify factors that promote and inhibit development.
	Stakeholder Map	Knowledge of how a value network impacts service delivery to the consumer is acquired by visually portraying the stakeholders who influence the development of the service.

Formulation of new Value Proposition	"How might we" Questions	Exploration of several potential solutions to an issue from various angles. Detaches from established thought patterns and lays the groundwork for generating new solution ideas.
	Value Proposition Canvas	To discover possible variations and encourage market acceptability, customers' demands are compared to the value proposition of the digital service.

Adapted from DIN (2019-09).

Appendix 17: Tools and Methods for Building Design Hypotheses

Table 10: Tools and Methods for Building Design Hypotheses

Methods for *Build*-Phase	**Institutional Design Cycle**
Method Name	**Short Description**
Service Blueprint	Visualization of the service creation process, including all required process stages, events, choices, and contacts with customers.
Lego Serious Play	Encourage people to come up with new ideas, improvement of communication, and quicker problem solving through a facilitated combination of playing and modeling with LEGO bricks and the business world's interests.
Personas	Fictional yet real human profiles are created to enhance empathy for a certain user or stakeholder group inside the service ecosystem.
Hotspot View	Identifies opportunities for both the customer and the company. Determines the future service solution's intervention points.
System Map	The different contextual relationships of main factors and the flows of resources in the service ecosystem, such as materials, energy, information, and money, are represented holistically in this visual description of the service system.
Design Workshops	Smaller groups concentrate extensively on a topic for a set amount of time during these events. The emphasis is on the collaborative creation of a shared concept.

Adapted from Abdel Razek et al. (2020); DIN (2020; 2019-09).

Appendix 18: Tools and Methods for Visualizing a Planned Service System

Table 11: Tools and Methods for Creating First Visualizations of a New Service System

Methods for *Intervene*-Phase	**Institutional Design Cycle**
Method Name	**Short Description**
Job Mapping	Actors' process steps should be identified and detailed to lay the ground for subsequent development activities.
Design Fiction	Design fiction is a type of design that explores new ideas through prototyping and narrative to break out from the monotony of static settings and receive more realistic insights
Digital Mock-Up	Using software-based visualization of a service design and subsequent simulation of effect through modifications to the design, common knowledge of the digital service can be produced.

Adapted from Abdel Razek et al. (2020); DIN (2020; 2019-09).

Appendix 19: Working Tools and Methods for Prototyping of a New Service System

Table 12: Working Tools and Methods for Prototyping of a New Service System

Methods for *Formulate Problem*-Phase	**Engagement Design Cycle**
Method Name	**Short Description**
LiCoDi	Design alternatives can be generated by raising awareness of limitations in the "analog world" and revealing innovative possibilities through digitalization.
Low-Resolution Prototyping	The new service concept's physical and prototype representations are developed so that initial ideas can be tested fast and easily, and feedback may be gathered early on.
Video Prototyping	An audiovisual method for testing, refining, and conveying ideas in a closed environment.

Adapted from DIN (2019-09).