

Service Dominant Architecture: Conceptualizing the Foundation for Execution of Digital Strategies based on S-D logic

Peter Weiß
Pforzheim University
peter.weiss@hs-pforzheim.de

Andreas Zolnowski
University of Hamburg
andreas.zolnowski@uni-hamburg.de

Markus Warg
Wedel University of Applied Sciences
mwa@fh-wedel.de

Thomas Schuster
Pforzheim University
thomas.schuster@hs-pforzheim.de

Abstract

Service Innovations are an opportune strategy for companies to compete in the digital age and to transform their business models taking a service perspective on their value creation. Digital business models require unique value propositions that incorporate digital technologies. Companies are required to build new digital capabilities to design and implement digital strategies. The paper takes a visionary perspective and motivates to view value creation through a service lens to respond to current challenges of digital transformation. We apply Service Dominant Architecture (SDA) to translate requirements of business initiatives into sustainable new IT infrastructure capabilities.

1. Introduction

New digital technologies are emerging rapidly. Thus, for companies' investments in their existing IT infrastructure and related capabilities remain a moving target. Incumbent companies face a major disadvantage as they have to build on their existing enterprise information systems (IS) and IT infrastructure (systems of record) [19]. Digitization requires companies to be more responsive to emerging customer needs or even invite customers to engage and interact with the company's resources in order to cocreate value. In addition, to compete in the digital age, enterprises must anticipate required future strategic moves. Service-Dominant Logic (SDL) is an inspiring source and offers guidance to develop compelling digital strategies making use of service innovations. SDL can be used to analyze and to anticipate future strategic moves (of the company itself but as well of the company's competitors) to drive digital transformation. This requires leadership with respect to deciding on how the organization and its IT infrastructure has to adapt to support company's future digital strategies. Once

strategic choices have been made, targeted business initiatives can be launched using acquired new enterprise capabilities. Capabilities are operationalized and implemented by the enterprise architecture (EA). EA builds, therefore, the foundation for execution [39, 33] and "strategic agility" [39]. Strategic agility grounds on the ability to mobilize and integrate required resources [3]. Decisions on strategic initiatives necessitate adequate investments in the foundation for execution [33] by introducing new IT infrastructure capabilities. Subsequently, we investigate which new enterprise capabilities [39] have to be embraced by companies such as open service innovation, value cocreation, resource integration, etc. [1, 25-27, 17, 30, 40, 38, 23] to ensure strategic agility, to develop digital strategies and to design and create unique value propositions [30] through creating lasting experiences by interacting with their customers. The paper is structured as follows. First two sections motivate our research and describe research contribution, approach, and objectives. In section three, we then review current challenges and look into requirements associated with digital transformation and strategic agility. In sections four and five, we introduce and conceptualize SDA as IT artifact enabling customer engagement systems (e.g., through implementing new IT infrastructure capabilities to create unique value propositions through service innovations). Next, section six provides insights concerning technologies to implement SDA service systems and related IT infrastructure as a prototype to conduct real life experiments. Finally, we summarize our research results and draw some conclusions.

2. Research Contribution and Approach

Firstly, our research documents and reflects on the foundations of and how we have created the IT artifact, namely Service Dominant Architecture (SDA) [38]. SDA enables new IT infrastructure capabilities (such as systems of engagement, interacting service systems, resource mobilization [3, 19]) on top of an existing

enterprise IT infrastructure and applications. Secondly, SDA as IT artifact conceptualizes new enterprise capabilities. Those capabilities translate then into respective new IT infrastructure choices based on IT capabilities with the aim to achieve strategic agility. Hence, our research contribution is an IT artifact, which introduces new IT infrastructure capabilities to implement digital strategies and digital business models. SDA can serve as an architectural vision and high-level figure as it is an abstraction and reference architecture encapsulating IT infrastructure capabilities. Our research approach embraces a design-science research as motivated by [12] and responds to real life business needs of an insurance company in Germany. The insurance company provides us with required real life context through selecting and defining respective use cases to elaborate on implementable solutions. Furthermore, we respond to the research challenges of service systems engineering as motivated by [3, p. 75]. In the remainder, we follow an incremental and iterative development approach to incorporate the feedbacks and results collected from our evaluation activities. Next sections elaborate on what the distinctive capabilities are which have to be considered for the conceptual design of SDA and will elaborate on an architectural blueprint [4:107, 22:35-59]. We encapsulate identified capabilities in purposed service systems of the IT artifact.

Figure 1. Mastering digital transformation and digital strategies: required steps (adapted and further developed based on [38])

3. Digital Transformation

Digitization and digital transformation affect business in many companies. Companies are confronted with fast changing markets and customer behavior because digital technologies affect life events of consumers and producers (see change drivers on the left column in Figure 1). Senior management has to make important decisions concerning infrastructure investments to introduce new strategic and operative capabilities for the company required to sustain in the digital age. Companies need to incorporate digital technologies to build new IT infrastructure capabilities [39, 22, 38] to achieve required strategic agility and to create unique value propositions [30].

3.1. Digital Business Models

Digital business models challenge physical ones on the basis of three components, namely content, experience and platform. Content is increasingly generated by users for users [22]. Customer experience is part of value cocreation and influences customers' perceived value as it acts as a filter to how customers' calculate and capture value from a company's offering (value proposition) [4]. Digital business models are

often based on digital ecosystems. Network effects are core concepts used by platform-based business models to achieve unique value propositions. To achieve this, platforms make use of service innovation. Platforms facilitate resource integration activities and hence, create spaces for cocreation by overcoming hitherto existing limitations concerning economies of scale and scope. Knowledge and information can be shared among the actors and create in this way strong network effects.

Thus, institutional arrangements and capabilities to manage complex networks of producers and consumers are important elements of digital business models. Platforms motivate a service perspective [4:107, 22:35-59]. Based on [30], companies can follow three major strategies in the digital transformation: (1) operational excellence, (2) customer engagement, and (3) new product and service.

Operational excellence is often strongly intertwined with the systems of record and is no longer a significant source of competitive advantage. Hence, strategies related to customer engagement and extending the total offering from products to solutions (combination of products and services) are promising elements to be incorporated to elaborate digital strategies. Taking a service perspective on value creation allows companies to strive for a higher level of customer orientation and customer loyalty. Hence, it is vital for companies to elaborate on their ability to reconfigure and reshape their value propositions by mobilizing and integrating resources adhering to S-D logic principles and related mechanisms. This allows companies to launch and pursue new business initiatives to compete in the digital age with digital business strategies and models. To achieve this, companies have to decide in which digital technologies to invest and how to establish required new capabilities in their organization [39, 22, 17].

Subsequently, we propose to view value creation through a service lens. Hence, we will have a look at S-D logic which provides useful concepts which can be used to develop digital strategies.

3.2. Service-Dominant Logic

From S-D logic perspective, service innovation is embedded in an actor-to-actor network, which “[...] underscores the importance of common organizational structures and sets of principles to facilitate resource integration and service exchange among those actors” [17]. As proposed by [17], service innovation can be conceptualized through a tripartite framework consisting of three major concepts, namely service ecosystem, service platform and value cocreation. We suggest adding service architecture as an additional

concept because it enables piloting of platforms and related complex service systems. Service systems are value cocreation configurations of people, technology and value propositions [25-27].

By adding a service systems view as argued by [25, 26] and the concept of “service ecosystems” as motivated by [1, 17, 35, 6], our perspective results in a broader view of service innovation in the context of digital transformation and service systems.

We see S-D logic and its foundational premises and concepts as an inspiring source to provide clear guidance concerning ingredients and dimensions of digital strategies. Furthermore, the conceptualization of service innovations [17] into service ecosystem, service platform, and value cocreation is a promising avenue of research to master digital transformation.

Table 1. Conceptualization of service innovation

Conceptualization	Definition
Service ecosystem (S-D logic: actor-to-actor net- work)	- self-contained, self-adjusting system of mostly loosely coupled social and economic (resource-integrating) actors - connected by shared institutional logics and mutual value creation through service exchange.
Service architecture (service systems: structure and mechanism)	- a structure for planning, designing and building solutions / piloting of complex service systems - enables customer centric solutions by configuring, mobilizing and integrating operant resources
Service platform (S-D logic: resource liquefac- tion; resource density)	- modular structure that consists of tangible and intangible components (resources) - facilitates the interaction of actors and resources (or resource bundles)
Value cocreation (S-D logic: resource integration, interaction)	- processes and activities that underlie resource integration - incorporate different actor roles in the service ecosystem.

Looking at the conceptualization of service innovation [17] and the systemic perspective motivated by service science [25-26] can span and frame our solution space. Table 1 shows major principles and capabilities derived from scientific literature

addressing service logic [8-11], S-D logic [33-34, 17] [1, 35] and platform perspective [22:35-59, 4:105-109].

3.3. Service Platform and Architecture

Many disruptions in industries are understandable through analyzing business logic and related “new” elements concerning value creation through a service lens. The concept of “platforms” is associated with respective new business logics and new capabilities (see Table 2) to disrupt existing markets and to transform existing market structures.

Table 2. Platform architecture: design artefacts and principles

Platform concept	Principle/ Description
Service platform	<ul style="list-style-type: none"> - S-D logic principles (resource integration, liquefaction, mobilization, density) - modular architecture - rules and protocols of exchange
Exchange	<ul style="list-style-type: none"> - exchange of information - exchange of goods and services - exchange of currency
Modularity	<ul style="list-style-type: none"> - decomposition/ partitioning into subsystems - standard interfaces (mashup of APIs) - set of “core” components (architecture elements, design artefacts) - value units (items of exchange, resources)
Core interaction	<ul style="list-style-type: none"> - participants (producer, consumer) - value unit (e.g. services) - filters (to exchange and deliver value units) - participants + value unit + filter
Resource orchestration	<ul style="list-style-type: none"> - demand-side model - open service innovation - economies of scale and scope - resource mobilization - key functions: pull, facilitate, match - balancing key functions
Scaling and evolution	<ul style="list-style-type: none"> - scale by layering new interactions on top of core interactions - add desirable new features and functionality - end-to-end principle

In the following, we focus on platform design and architecture as it is of major importance for our later

conceptual design and requirements for concrete implementation. As we have highlighted, platform is a pivotal concept for implementing digital strategies and business models [4:105] (see as well in section 3.1). Platforms are supposed to open up previously closed value creation activities through offering opportunities to interact and co-create. Platforms move value creation from “place to space” and support sharing of information and content between the various actors (such as suppliers, customers, and producers) to innovate and integrate resources along the customer’s processes [4:57].

Platforms transform nowadays mainly physical business models from hitherto supply economies of scale (pipelines) [22:34, 22:59] into demand economies of scale using mainly network effects. As a result, they support two-sided markets which offer a location for both consumers and producers to interact and co-create. Platforms mobilize, orchestrate and integrate resources and hence facilitate service innovations [17, 4:105, 38, 22:34, 22:59]. In this context, strategic agility [39] is key to allow companies to respond to fast changing markets, to overcome inertia, to react on new competitors in their markets and mobilize required resources to offer personalized solutions, and to support customers’ life events and related customer processes. Platforms are designed and build around various principles and effects. The main concept represents “core interactions” which constitute the “why” or purpose [2:54] of the overall platform design [22: 38].

Platforms are complex, multisided systems which facilitate and ease interactions of large networks of users. They create different types of “network effects” which nurture platform businesses [22:35]. Platforms introduce new capabilities, mainly complex interactions based on intensive exchange of information, goods or services and some form of currency. “Exchange of information” is one of the fundamental characteristics of platforms [22:36]. Platforms are “information factories” without control over their inventories of information [22:42]. Platform design includes decisions concerning rules for value unit creation and integration into the platform and “what differentiates a high-quality from a low-quality unit” [22:44]. Users of platforms are producers and consumers. Roles are disjoint as users may change roles [22:39].

4. Building Foundations for Execution

Previously, we have described what the ingredients of digital strategies and what their purpose are. Now, we introduce Service Dominant Architecture (SDA) to

overcome current challenges of service systems engineering [3]. SDA contains mechanism and structures to build service ecosystems on the basis of interacting service systems as an important catalyzer of future service innovations. SDA intends to become reference practice and an integral element of digitization and digital transformation strategies.

4.1. Use Cases

In the remainder, we follow an incremental and iterative development approach to incorporate the feedbacks and results collected from our evaluation activities. Evaluation [12] is considered as a continuous task and will be based on implementing selected use cases [41]. Finally, we look at the technologies to implement SDA with the aim to build and operate an experimental prototype. In this way, we will be able to launch experiments to achieve “proof of concept” and further feedback for next development iterations. The context of our research is the insurance business. We have analyzed the business needs of an insurance company in Germany based on a set of use cases (see Table 3). In this way, we have elicited related business needs. In addition, we have been able to elaborate on an effective problem representation [12]. Following a use case-based development approach ensures to keep focus on business needs as well as collecting user feedback to conduct evaluation activities continuously along the various steps of the solution design and implementation activities.

Selected use cases help to scrutinize and highlight the relevance of our research activity. Furthermore, use cases will serve as a base for related evaluation activities and the proof of concept of our real life experiments.

Table 3 shows an excerpt of the first bundle of the first generation of use cases which have been selected together with an insurance company. We conclude that next generation of use cases and value propositions will be more demanding as user and customer expectations are increasing rapidly with newly emerging digital technologies. As we have already highlighted, user experience is key for companies’ business models to sustain in the digital age.

Table 3. Initial use cases for design and piloting

No.	Use case
1	Life insurance
2	Car insurance
3	Household insurance
4	Emerging digital markets

Use case #1 takes focus on life insurance. Design of a compelling value proposition requires offering insurances with flexible fees dependent on actual customer behavior and provision of access to personal customer data (vital functions trackers, analytics apps). The second use case (#2) addresses car insurances which are offered on a flexible basis or as “as-a-service” offerings. Flexible fees are offered on the basis of technical car and behavioral driver data combined with external third party services. Next, use case #3 aims to make use of sensor data and to monitor apps of the insured facility. Insurers and technical service providers may collaborate to create new value propositions for customers. Last but not least, the fourth use case (#4) contains insurance offerings making intensive use of digital technologies to create new value propositions and offerings. This use case aims to create new customer experiences through unprecedented market offerings. This includes related new IT capabilities to launch strategic business initiatives to create new markets and to react instantly to competitive market offerings as a response to future market dynamics.

4.2. Business IT Alignment

Existing IT infrastructure capabilities have to be aligned with new business requirements to enable the future launch and support of digital business initiatives. Furthermore, investments in IT capabilities are made to compete through service innovations by embracing customer engagement systems [19] and new packaging of products and services to achieve unique value propositions [30].

Business and IT have to collaborate and work jointly towards customer engagement and digitized solutions. To achieve this, companies need to incorporate besides digital technologies customer experience in their business models.

When dealing with the challenges of digital transformation, companies tend to be too much focused on technology and related challenges of implementation [13]. Thus, investments in IT infrastructure capabilities are often rather fragmented based on strategic initiatives of business units. They are often not made in adequate coherence with a shared vision and digital strategy. However, digital transformation requires joint efforts and a new level of interaction and intimacy between business and IT to master digital transformation.

Digital transformation encompasses to get more digital technology savvy by introducing digital technologies and digital capabilities. Often this requires considerable change in relation to structures, processes, people, and culture. Not only IT

infrastructures and systems are undergoing dramatic change, but also organizational structures, processes, systems, and people need to adapt to new requirements imposed by digital strategy and business models.

Digital transformation requires understanding the relationship between IT infrastructure capabilities of the enterprise and its “ability to implement its business initiatives” [39:62]. This relationship is addressed by an emerging discipline named Enterprise Architecture Management (EAM) [20, 21] (see Figure 2). The figure illustrates the various concepts required to align organizational architecture and IT infrastructure and systems (Figure 3).

[39] define strategic agility as “[...] set of business initiatives an enterprise can readily implement” [39: 61]. Enterprise capability encompasses coordinating respective set of elements such as customer base, brand, core competence, infrastructure, and employees, into an “integrated group of resources” [39:61].

4.3. Enterprise Architecture operationalizes IT capabilities

Enterprise architecture allows to link and translate high-level requirements (e.g., new capabilities derived from business initiatives and new business logics/ models) into IT infrastructure capabilities [39]. Enterprise architecture and standards are related to required key IT infrastructure capabilities to implement digital strategies, independent of their perspective,

namely demand- and supply-driven or rather driven internally [39]. In the next section, we introduce Service Dominant Architecture (SDA) [38] as architecture to master the previously motivated challenges of digital transformation through introducing required new capabilities (mainly related to platform business design and requirements of facilitating “systems of engagement” [19]).

In this way, SDA brings previously motivated IT infrastructure capabilities into action thereby building on existing IT resources and systems (“systems of record” [19]).

5. Service Dominant Architecture

Based on the literature on S-D logic and service systems, the Service-Dominant Architecture (SDA) was proposed in 2015 [38]. SDA builds on existing enterprise information systems and introduces new IT infrastructure capabilities based on S-D logic (see Table 1) on top of an existing IT infrastructure. In this way, SDA combines stability (high level of integration; systems of record) with flexibility and strategic agility (systems of engagement). The latter requires to implement platform-oriented IT capabilities (see Table 2). As a result, SDA enables a company to design platforms around respective core interactions [22:38-39] to filter and exchange information on value units to realize digital solutions and digital-enabled services (see layer “solutions” shown on top of the pyramid in

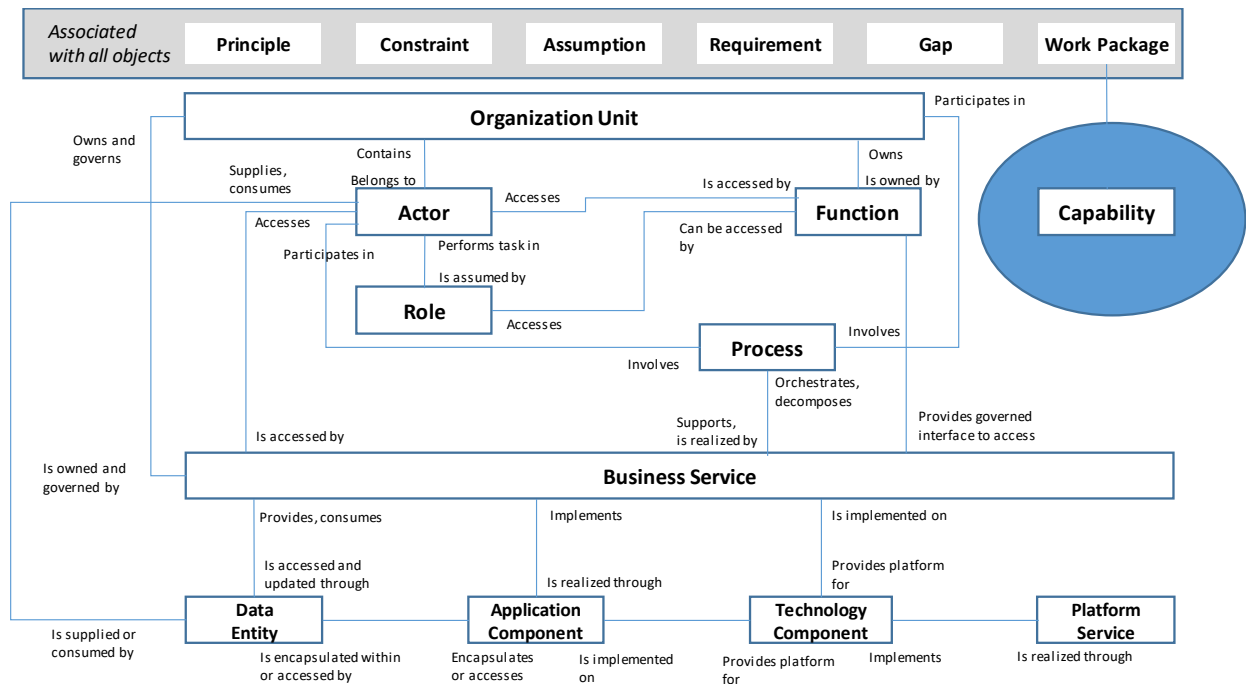


Figure 2. Linking Enterprise with IT Infrastructure Capabilities [20, 21]

Figure 3) for customers [38]. This is enabled mainly through the platform layer (systems of engagement). In Figure 3 major building blocks and elements of the high-level architecture are shown. SDA embraces a systems perspective [3, 25] configuring actors and resources guided by unique value propositions. SDA operationalizes concepts from service science [25] in an architectural blueprint for the implementation of a service platform and thus, addresses and builds prerequisites for the implementation of innovative service based solutions [4, 17, 30]. To generate customer-centric solutions, the SDA implements the capabilities to capture (integration, participation), exchange (interaction), and orchestrate relevant resources. SDA represents a conceptualization of relevant S-D logic principles and related capabilities which are operationalized by three distinct purposed subsystems (in the remainder referred to as “technical service systems” (see Figure 3, right column). By this, the SDA is aiming (1) to accelerate the capabilities in all customer-centric areas, (2) to achieve useful collaboration and cocreation, (3) to deepen the data-based customer understanding, and (4) to create networks of partners and other external service providers [28].

The architectural blueprint of the SDA enables the integration and orchestration of resources (such as processes, data, applications, functions) into agile, flexible and collaborative services in real-time. This facilitates the ability to use existing resources and thus, supports the implementation and development of service systems of different granularity (micro, meso, and macro). SDA orchestrates the various resources, processes, and internal and external components

needed for developing new solutions (Figure 3). To attain this goal, SDA introduces additional information system layers to an existing IT landscape on top of existing enterprise information and legacy systems (systems of record) [19]. Next steps foresee to further detail and concretize above mentioned technical service systems. This is perceived as continuous task and is part of our real life experiments and iterative development process. The above high-level concepts have to be broken down into implementable IT concepts. Thus, concepts and appropriateness of a platform-based solution design will be proposed and discussed in the next section.

6. Implementation and Evaluation

In this section we will present and discuss prominent design paradigms, which have shown their strengths and limits regarding concrete implementation and operation. New initiatives such as container-based operations (e.g., driven by docker, rkt or LXD) and principles for emphasizing modularization much stronger than SOA (like microservices) have emerged and propose new views and ways how to cope with above mentioned complexities and challenges. In the following, we focus on platform design and architecture as it is of major importance for our later conceptual design and requirements for concrete implementation. As we have highlighted, platform is a pivotal concept for implementing digital strategies and business models [4:105] (see as well in section 3.1).

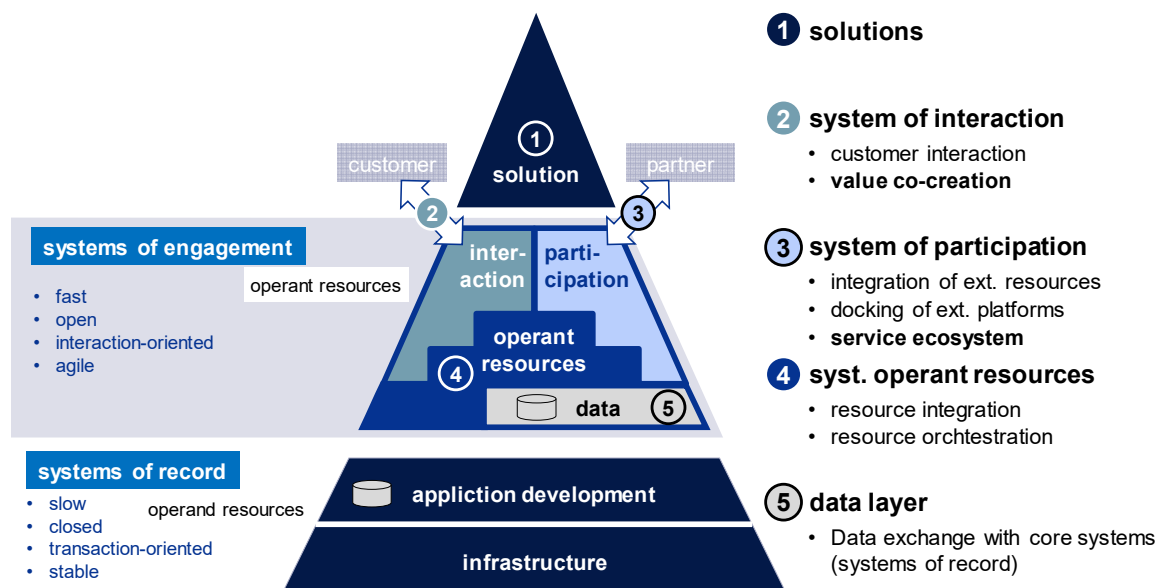


Figure 3. Service Dominant Architecture: Solutions based on resource integration and service systems [38]

6.1. Solution Design and SDA Prototype

Microservices are a promising approach to build real life solutions with a strong relation between business and information systems. Closely related to this are operations in successful companies nowadays driven by the so-called DevOps approach. Current market competition enforces faster and more convenient development of solutions, strictly oriented towards customer requirements and embracing collaboration of business and IT within organizations. For software architectures of distributed systems, the term microservice has gained a lot of attention recently [15, 24].

Microservices are an architectural style for software systems. In short, microservices allow developing a system through a set of small sized services. Each service is executed independently (process space), uses its data (database) and offers lightweight communication mechanisms to other services (often HTTP/HTTPS). According to this approach, services are built around business capabilities (see Figure 2). This fosters separation of concerns. In contrast, architectures of huge systems are classically organized in layers; teams often consist of specialists who possess skills for a specific layer (like UI, logic and data). This organizational team formation will resemble the system's architecture – known as Conway's Law. "Any organization that designs a system [...] will inevitably produce a design whose structure is a copy of the organization's communication structure" [5]. Once such a system has grown up to a certain point, additional modules and changes in the middleware layer require high efforts. The latter is a result of dependencies within the layer and to the layers above and below. Thus, if a team changes a single module (or services) often the whole middleware layer, it will build and deploy everything again. Due to dependencies in (method) calls, a change is not isolated to a single module. Since a microservice focusses on a single business capability and utilizes a broad implementation stack of all technical layers (including user-interface, storage, and external communication), the team organization has to be different. Teams are typically cross-functional (see Figure 4), including skills required for project management, business logic, database, and user-interface design. In contrast to the approach mentioned above, the results are loosely coupled services. Furthermore, the team can redeploy these services whenever needed. Of course, this cannot be done without any cost. The team usually uses fully automated deployment mechanisms to put changes online, recently in addition to container-based deployments. Also, each team will have to monitor

service health. Hence, the team is responsible for operation [15].

Along with the characteristics above, microservices foster the following principles [15]: (1) decentralized governance, (2) shared nothing, (3) decentralized data management, (4) smart endpoints and dumb pipes, (5) infrastructure automation, (6) evolutionary design. Smart endpoints and dumb pipes are a major difference to typical service-oriented (SOA) approaches [18]. In SOA often a component called Enterprise Service Bus (ESB) is employed to support communication. The ESB usually offers sophisticated communication services such as service orchestration and choreography, and even control of business rules. Microservices use dumb pipes for communication to create services which are as decoupled and as cohesive as possible (services in style of UNIX or according to pipes and filters architectural pattern [15]). One of the big challenges regarding the implementation of microservice architectures is the decomposition of the system into adequately tailored services. Therefore, several different strategies and approaches are available. One possibility to decompose an application is according to business capabilities (see Figure 2) or use cases (see Table 3).

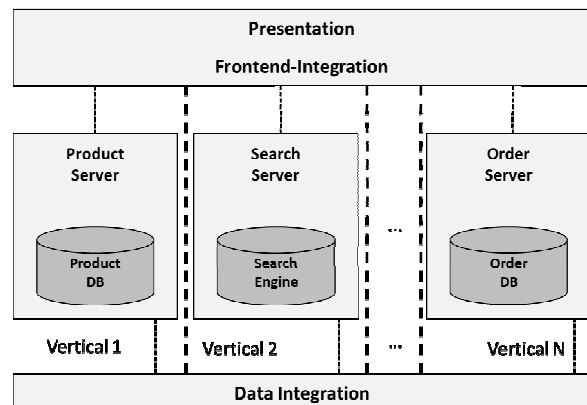


Figure 4. Decomposition according to Business Capabilities and Use Cases

In contrast to classical monolithic systems where the application is split horizontally (see Figure 4) this approach decomposes the application vertically into cohesive sub-systems also called verticals. Each vertical comprises all technology tiers (presentation, business logic, and data access). Once these services need to be scaled out, this is the place where container-based management comes into play. Each container operates as a fully isolated sandbox, with only minimal operating system components present in it. The system resources of the underlying system are shared between all containers of a single node. A team can define a

container for each microservice, which then allows redistributing this microservice multiple times on the same node or different nodes. Hence, the team can react swiftly to all business driven requirements. This means, if the business demand is to replicate a certain microservice multiple times, it can easily be redeployed until all service requirements are met. Another strong advantage of this approach is, that services which reside on the same node do not influence themselves directly since they run in an isolated environment (the container).

6.2. Evaluation

SDA provides the basis for real life experiments [12]. SDA and related subsystems are currently being implemented as a prototype and will be evaluated on the basis of data and processes yielding from selected use cases. SDA informs about both required investments and how to build required new IT infrastructure capabilities. SDA provides the management with a communication tool clarifying strategic directions and to achieve required agility to respond to changes in their environment. The aim is to build a foundation for execution [32, 33] which includes operational model, enterprise architecture / IS architecture and related IT artifacts [12] and decisions concerning targeted investments to achieve required IT infrastructure capabilities [39].

Evaluation is considered to be a crucial task and will be conducted continuously. Evaluation depends on implementing selected use cases and related requirements by means of IT solutions based on SDA experimental prototypes. In this way, we will be able to launch a set of various experiments to achieve required “proof of concept” and to receive further feedback and data for the next development iterations.

Currently, SDA is evaluating various solution designs and technologies and an SDA prototype incrementally. At this stage of development, activities focus primarily on implementing the SDA stable core. Later, the technical service systems (see Figure 3) will be continuously expanded through adding additional features and functionality. Various architectural paradigms are being evaluated. We will launch real life experiments to evaluate SDA in the context of available use cases, which stem from the digital transformation endeavor of an insurance company.

7. Summary and Conclusion

Digitalization and digital transformation require a dramatic change of the enterprise IT. It requires to focus on the customers’ needs and to develop

customer-centric solutions. Service science (service systems) and S-D logic can make a significant contribution to developing compelling digital strategies. We have proposed Service Dominant Architecture (SDA) as reference architectural framework to inform companies about how to master systematically the challenges related to digital transformation. SDA encapsulates required enterprise capabilities in its subsystems and core components. In this way, SDA reduces risks concerning necessary investments in digital technologies and IT infrastructure capabilities. Enterprise-wide IT architecture and standards are essential to achieving strategic agility through interaction with and integration of internal and external resources. Those resources are used in value cocreation activities by enabling interactions and resource integration activities among the actors on the platform. Next steps foresee to realize further iterations and to continue the evaluation of our conceptual design in a real world application scenario. Our approach sees use cases as an integral element to develop and implement experimental prototype artifacts along with an incremental and iterative development approach. Selected use cases stem from the insurance business, namely an insurance company, who will be our application domain for experimenting and evaluating our developed IT artifact and solution design.

8. References

- [1] M.A. Akaka, S.L. Vargo, “Technology as an operant resource in service (eco)systems”, *Information Systems and e-Business Management*, 12 (3), 2014, pp 367–384.
- [2] Arthur, W.B., *The Nature of Technology: What it is and how it evolves*. Free Press, New York, 2009.
- [3] T. Böhm, J.M. Leimeister, K. Möslin, “Service Systems Engineering - A Field for Future Information Systems Research”, *Business & Information Systems Engineering*, 56 (2), 2014, pp. 73-79.
- [4] Chesbrough, H., *Open Service Innovation*, Jossey-Bass, San Francisco, 2011.
- [5] M.E. Conway, “How Do Committees Invent?” vol. *Datamation magazine*, 1968.
- [6] B. Edvardsson, B. Tronvoll, “A new conceptualization of service innovation ground in S-D logic and service systems”, *International Journal of Service Quality and Service Sciences*, 5 (1), 2013, pp. 19-31.
- [7] V. Eloranta, T. Turunen, “Platforms in service-driven manufacturing: Leveraging complexity by connecting, sharing, and integrating”, In: *Industrial Marketing Management*, Volume 55, May 2016, Pages 178–186.
- [8] C. Grönroos, P. Voima, “Critical service logic: making sense of value creation and co-creation”, *Journal of the Academy of Marketing Science*, 41, 2013, pp. 133-150.

- [9] C. Grönroos, A. Ravald, "Service as business logic: implications for value creation and marketing", In: *Journal of Service Management*, 22 (1), 2011, pp. 5-22.
- [10] C. Grönroos, "Service logic revisited: who creates value? And who co-creates?", *European Business Review*, 20 (4), 2008, pp. 298 - 314.
- [11] C. Grönroos, "Value co-creation in service logic: A critical analysis". *Marketing Theory*. 11 (3), 2011, pp. 279-301.
- [12] A.R. Hevner, T.M. Salvatore, J. Park, "Design Science in Information Systems Research", In: *MIS Quarterly* Vo. 28, No. 1, March, 2004, pp. 75-105.
- [13] G.C. Kane, D. Palmer, A.N. Philips, D. Kiron, N. Buckley, "Strategy, not Technology Drives Digital Transformation", *MIT Sloan Management Review*. Deloitte University Press. Summer, 2015, pp.3-24.
- [14] G.C. Kane, D. Palmer, A.N. Philips, D. Kiron, "Is Your Business Ready for a Digital Future?", In: *MIT Sloan Management Review*, 56 (4), 2015, pp. 37-44.
- [15] J. Lewis and M. Fowler: "Microservices" [martinfowler.com](http://martinfowler.com/articles/microservices.html). [Online]. Available: <http://martinfowler.com/articles/microservices.html>. [Accessed: 7-Apr-2017].
- [16] M. Lokkegaard, N.H. Mortensen, T.C. McAloone, "Towards a framework for modular service designs", *Research in Engineering Design*, 27 (3), 2016, pp 237-249.
- [17] F.R. Lusch, S. Nambisan, "Service Innovation: A Service-Dominant Logic Perspective", *MIS Quarterly*, 39 (1), 2015, pp. 155-175.
- [18] Z. Mahmood, "Service oriented architecture: tools and technologies" benefits, vol. 6, p. 7, 2007.
- [19] G. Moore, "Systems of Engagement and the Future of Enterprise IT: A Sea Change in Enterprise IT", *AIIM Whitepaper*, 2011, <http://www.aiim.org/futurehistory>; last visit 05 May 2016.
- [20] The Open Group, "Core Content Metamodel Concepts", <http://pubs.opengroup.org/architecture/togaf9-doc/m/chap34.html>; last visit 31.01.2017.
- [21] The Open Group, "TOGAF Content Model and Extensions" http://pubs.opengroup.org/architecture/togaf9-doc/m/Figures/34_contentfwk1.png; last visit 31.01.2017.
- [22] Parker, G.P.; Alstyne, Van, M.W.; Choudary, S.P., *Platform Revolution*. Norton & Company, New York London, 2016.
- [23] C.K. Prahalad, V. Ramaswamy, "Co-Opting Customer Competence", In: *Harvard Business Review*, January-February 2000, pp.79-87.
- [24] G. Steinacker, "Scaling with Microservices and Vertical Decomposition", dev.otto.de, last visit 15 April 2017.
- [25] J. Spohrer, P.P. Maglio, "Toward a Science of Service Systems", In: P.P. Maglio et al. (eds.), *Handbook of Service Science, Service Science: Research and Innovations in the Service Economy*, 2010, pp.157-194.
- [26] J. Spohrer, P.P. Maglio, "The Emergence of Service Science: Toward Systematic Service Innovations to Accelerate Co-Creation of Value", *Production and Operations Management*, 17 (3), May-June, 2008, pp. 238-246.
- [27] J. Spohrer, P.P. Maglio, J. Bailey, D. Gruhl, "Steps Towards a Science of Service Systems", In: *IEEE International Conference on Information Reuse and Integration*, Las Vegas, 13-15 August 2007.
- [28] C. Richardson, "Microservices: Decomposing Applications for Deployability and Scalability" *InfoQ*, [Online]. Available: <http://www.infoq.com/articles/microservices-intro>. [Accessed: 7-Apr-2017].
- [29] D.K. Rigby, "Digital-Physical Mashups". In: *Harvard Business Review*, September 2014.
- [30] J.W. Ross, I.M. Sebastian, C.M. Beath, "How to Develop a Great Digital Strategy", In: *MIT Sloan Management Review*, Vol. 58, No. 2, Winter 2017 Issue, pp. 6-10.
- [31] J.W. Ross, C.M. Beath, "Beyond the Business Case: New Approaches to IT Invest". In: *MIT Sloan Management Review*, Winter 2002, Vol. 43, No.2, 2002, pp.50-60.
- [32] J.W. Ross, "Enterprise Architecture: Driving Business Benefits from IT", In: *MIT Sloan Management, CISR Working Paper 359*, Massachusetts Institute of Technology, April 2006.
- [33] Ross, J.W., P. Weill, D.C. Robertson, *Enterprise Architecture as Strategy*, Harvard Business Review Press, Boston, Massachusetts, 2006.
- [33] S.L. Vargo, R.F. Lusch, "Evolving to a New Dominant Logic for Marketing", In: *Journal of Marketing*, Vol. 68, January 2004, pp. 1-17.
- [34] S.L. Vargo, R.F. Lusch, "Institutions and axioms: an extension and update of service-dominant logic", In: *Journal of the Academy of Marketing Science*. January, Vol. 44 No. 1, 2016, pp. 5-23.
- [35] S.L. Vargo, M.A. Akaka, "Value Cocreation and Service System (Re)Formation: A Service Ecosystems View", In: *Service Science*, Vol. 4 No.3, 2012, pp. 207-217.
- [36] S.L. Vargo, H. Wieland, M.A. Akaka, "Innovation through institutionalization: A service ecosystems perspective". In: *Industrial Marketing Management*, Vol. 44, January, 2015, pp. 63-72.
- [37] C.A. Voss, J. Hsuan, "Service Architecture and Modularity", *Journal of Decision Sciences*, 40 (3), 2009, pp.541-569.
- [38] M. Warg, P. Weiß, A. Zolnowski, R. Engel, "Service Dominant Architecture based on S-D logic for Mastering Digital Transformation: The Case of an Insurance Company", *RESER Conference Proceedings*, Naples, Italy, 2016.
- [39] P. Weill, M. Subramani, M. Broadbent, "Building IT Infrastructure for Strategic Agility", In: *MIT Sloan Management Review*, Fall 2002, Vol. 44, No. 1, pp. 56-66.
- [40] P. Weill, S.L. Woerner, "Thriving in an Increasingly Digital Ecosystem", *MIT Sloan Management Review*, 56 (4), 2015, pp.27-34.
- [41] Oestereich, B., S. Bremer, S., *Analysis and Design with UML 2.3*, Munich:Oldenbourg Verlag, 2009.