STUDIENSKRIPT



## Smart Services I

DLBINGSS01

Übergeordnete Lernziele

##### Einleitung 9



Der Kurs **Smart Services I** vermittelt Ihnen Konzepte und Methoden zur Entwicklung von Smart Services. Hierzu wird zunächst eine Einführung des Begriffs im Kontext der Digitalisie- rung und der Industrie 4.0 vorgenommen. Darauf aufbauend wird gezeigt, inwiefern innova- tive Services am Beispiel digitaler Intermediäre auf bestehende Geschäftsmodelle oder sogar Märkte disruptiv einwirken können. Anschließend werden Ihnen ausgewählte Methoden und Techniken vermittelt, mit denen Digitalisierungspotenziale erkannt und modelliert werden können. Zudem werden Ihnen ausgewählte Architekturen und Plattformen zur Integration von Services vorgestellt. Abschließend werden relevante Technologien zur Implementierung von Smart Services vermittelt und es wird kurz dargestellt, wie die Qualität von Services verein- bart werden kann.



# Lektion 1

## Digitalisierung, Industrie 4.0 und Smart Services

#### LERNZIELE

Nach der Bearbeitung dieser Lektion werden Sie …

… mit den einführenden Grundlagen und Zusammenhängen der Digitalisierung und cyber- physischen Produktionssystemen vertraut sein.

… über die Grundlagen von Smart Services in der Industrie 4.0 informiert sein.

… Beispiele für Smart Services kennen.

DL-D-DLBINGSS01-L01

1. Digitalisierung, Industrie 4.0 und Smart Services

### Einführung

Herr Müller beabsichtigt, sich bei einem namhaften deutschen Autobauer ein Auto zu konﬁgurieren, welches auf ihn persönlich zugeschnitten werden soll. Ist das schon heute möglich oder gar selbstverständlich? Und wenn ja, dann in welchem Ausmaß bzw. Umfang und unter welchen Voraussetzungen bzw. Bedingungen?

Auf den ersten Blick hört sich das nicht besonders anspruchsvoll, sondern eher reali- sierbar an. Die genaue Betrachtung von Einzelfällen lässt – nach heutigem Stand – jedoch noch manche Fragen und vor allem Kundenwünsche offen und unerfüllt. Dies betrifft im konkreten Beispiel Fragestellungen der folgenden Art:

* Der Kunde ist 2,06 m groß. Er kann infolge der Sitzhöhe in den meisten Fahrzeugen nicht aufrecht und damit auch nicht bequem sitzen. Beim Konﬁgurieren seines neuen Fahrzeuges wählt er zunächst das Modell, den Motor, die Leistung und Karos- seriefarbe. Nun gibt er zusätzlich an, dass der Sitz 23 cm tiefer sein soll als bei der Standardausführung. Kann der Autobauer diesem Wunsch nachkommen?
* Über den Produktkonﬁgurator erhält der Autobauer diese Kundenanfrage. Da eine kundenspeziﬁsche Entwicklung, der sogenannte Engineer-to-Order-Prozess, aufwen- dig und für den Kunden zu teuer wäre, wird die Anfrage über den Serienprozess abgewickelt. Autobauer und Sitzhersteller führen gemeinsam eine Engineering- Simulation des Sitzes durch. Im positiven Fall wäre eine Gesamtkonstruktion erfor- derlich. Sind nach der Sitzänderung Gurtstraffer und Airbag passend bzw. (noch) in der richtigen Position? Sind die Werkzeuge für diese Produktion geeignet?
* Im Anschluss daran wird der Preis simuliert. Was kostet diese Änderung? Welcher Preis muss vom Kunden verlangt werden? Lässt sich diese Ausstattung auch an andere Kunden verkaufen? Welcher Produktionstermin ist realisierbar?

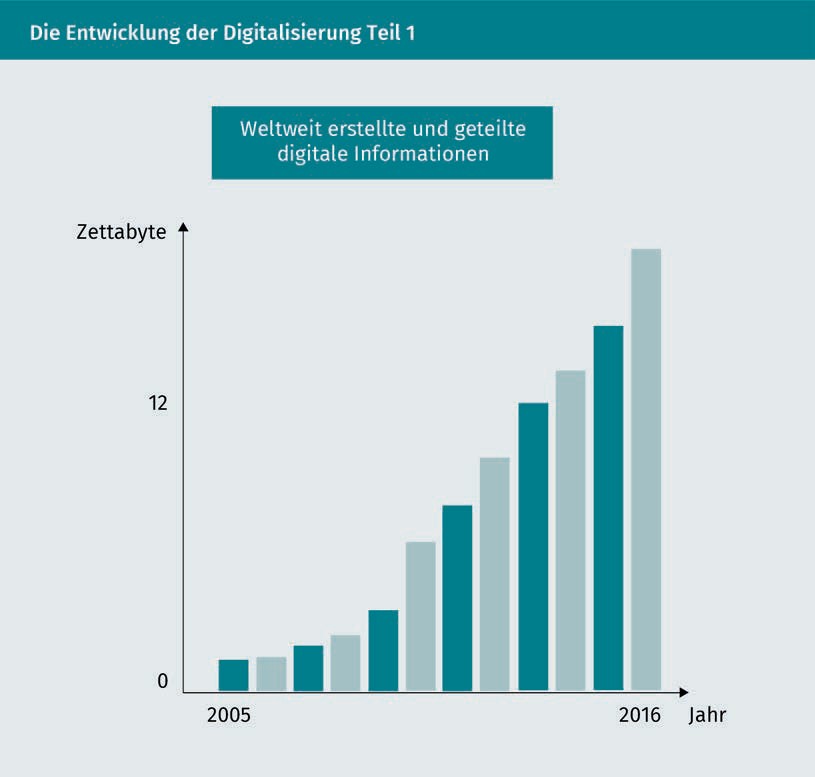
Sind all diese Fragen geklärt, können die Auftragsdaten des individuellen Sitzes in die Produktionsplanung ﬂießen. Die speziﬁschen Konstruktionsänderungen werden beim Lieferanten zur Maschinensteuerung geleitet. Dieses Beispiel verdeutlicht einen wesentlichen Aspekt der Industrie 4.0 als deutsche Initiative. Der globale Wettbewerb erfordert das Konzept des personalisierten Produktes (vgl. Kaufmann 2015, S. 1f.).

Weiter zeigt sich an diesem Beispiel die Vielschichtigkeit von Produkten und Dienstleis- tungen der heutigen Zeit. Daher werden im Folgenden die Grundlagen zum Thema Digi- talisierung geschaffen sowie die sich ergebenden die Chancen mit besonderem Schwerpunkt der cyber-physischen Produktionssystemen gegeben. Anschließend wird bezugnehmend auf die Industrie 4.0 der Begriff Smart Services eingeordnet, welche auf Daten basieren und den Kunden in den Mittelpunkt stellen. Abschließend werden ver- anschaulichende Beispiele aus verschiedenen Branchen zu Smart Services vorgestellt.

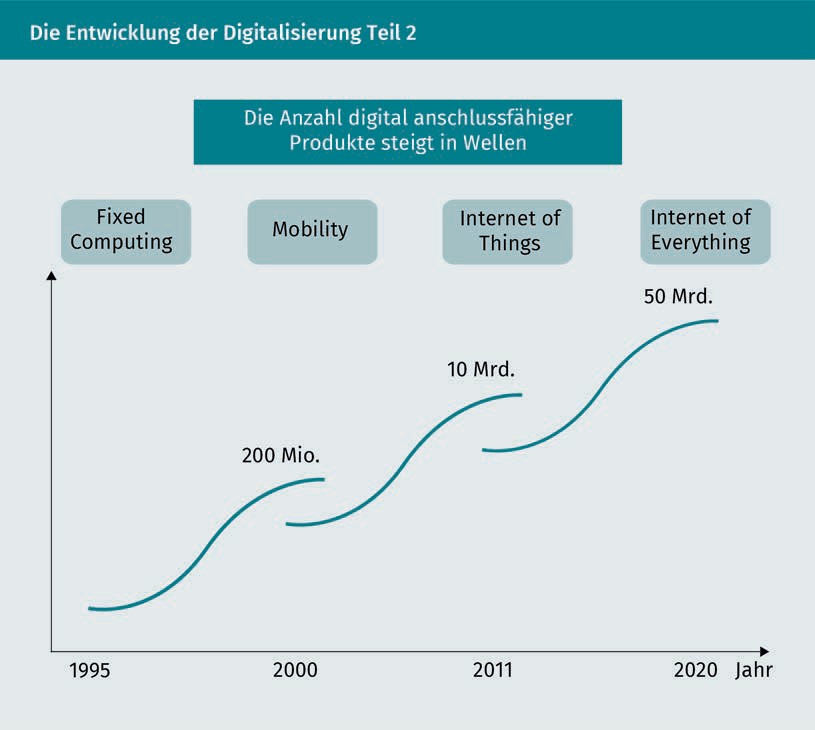
Digitalisierung, Industrie 4.0 und Smart Services

### Digitalisierung und cyber-physische Produktionssysteme

Die Digitalisierung stellt eines der zentralen Themen unserer Zeit dar und bedeutet einschneidende Veränderungen für alle wirtschaftlichen und gesellschaftlichen Berei- che. Digitales Leben und Arbeiten sind mit der Zeit Alltag geworden (vgl. Boes 2014, S. 11). Durch die Digitalisierung wird unser Leben im Hinblick darauf verändert, „wie wir Dinge tun“ (Borgmeier/Grohmann/Gross 2017, S. XV). Im Rahmen dieser Entwicklung werden zunehmend bestehende Produkte, Services, Verbunde von Produkt und Service, Geschäftsprozesse, Geschäftsmodelle, ganze Industrien und deren Interaktionen, wel- che auch untereinander einem Wandel unterliegen und von diesem betroffen sind, grundlegend verändert oder ersetzt. Man bezeichnet dies als „digitale Transformation“ (Borgmeier/Grohmann/Gross 2017, S. XV; vgl. Boes 2014, S. 11ff.). Heute sind es insbeson- dere Leistungsbündel (Verbund von Sach- und Dienstleistungen) und E-Dienstleistun- gen, in welchen im Kontext von Smart Services bzw. Industrie 4.0 ein hohes Wachstums- potenzial steckt. Kundenwünsche und Kundenzufriedenheit sind entscheidende Erfolgsfaktoren. Die sogenannte Integration des externen Faktors (bzw. des Kunden) in die Leistungsprozesse gewinnt zunehmend an Bedeutung. Dies gilt insbesondere für E- Commerce und E-Dienstleistungen im Kontext von Smart Services bzw. Service 4.0 (vgl. Wehrlin 2018, S. Vf.). Die folgenden Darstellungen veranschaulichen die Entwicklung der Digitalisierung.

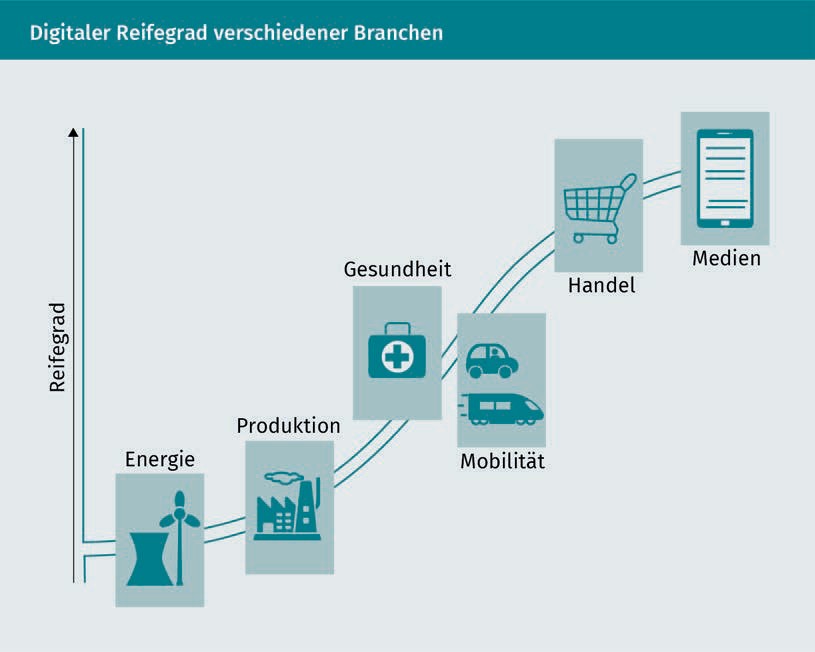


Digitalisierung, Industrie 4.0 und Smart Services



Die weltweit erstellte und geteilte digitale Information steigt an (vgl. Evans 2003, S. 3). Die Anzahl digital anschlussfähiger Produkte steigt in Wellen (vgl. Fullan/Donnelly 2013, S. 9).

Die Entwicklung führt unter anderem dazu, dass sich wertbasierte Partnerschaften und Koproduktionen wie auch Know-how in Big Data bzw. Datenanalytik als signiﬁkante Wettbewerbsvorteile herausstellen. Dadurch erfolgt zugleich eine Substitution und Ver- drängung der zuvor geltenden klassischen Entwicklungsbedingungen und -vorteile eines Produkts bzw. einer Dienstleistung (vgl. Borgmeier/Grohmann/Gross 2017, S. XV; Boes 2014, S. 11ff.). Durch die folgende Darstellung wird der digitale Reifegrad verschie- dener Branchen aufgezeigt.



Die zunehmende Digitalisierung begann zuerst in den leicht zu digitalisierenden Bran- chen Medien und Handel. Sie setzt sich fort über die Branchen Mobilität, Gesundheit, Produktion und schließlich auch Energie (vgl. acatech 2014, S. 17). Die Digitalisierung schreitet immer weiter voran und verändert neben den Branchen auch das Verhalten der User. Alleine in Deutschland nutzen über 62 Millionen Menschen das Internet. Immer mehr Menschen sind immer längere Zeit online. Social Media (bzw. online- basierte Plattformen) ermöglicht die Kommunikation und Vernetzung zwischen den Nutzern untereinander und mit den Dienstleistern. Von besonderer Relevanz ist der Datenaustausch zwischen Unternehmen bzw. Dienstleistern und den Nutzern bzw. Kun- den (vgl. Wehrlin 2018, S. 48f.).

Digitalisierung Bei fortgeschrittener Digitalisierung sind die Unternehmen darauf angewiesen, proﬁtable Geschäfts- modelle und Smart Services zu entwi-

ckeln.

In der weiteren Entwicklung im Zeitalter zur fortgeschrittenen **Digitalisierung**, Industrie

4.0 und IoT (Internet der Dinge) sind die Unternehmen darauf angewiesen, proﬁtable Geschäftsmodelle sowie Smart Services (bzw. intelligente, digitale respektive datenba- sierte Services/Dienstleistungen) zu entwickeln und diese erfolgreich umzusetzen. Hier- bei gilt es, den B2B-Bereich (= Business-to-Business) wie auch den B2C-Bereich (= Busi- ness-to-Customer) entsprechend zu erfassen und zu berücksichtigen (vgl. Borgmeier/ Grohmann/Gross 2017, S. XV; Kagermann 2014, S. 67ff.).

Im Rahmen der digitalen Transformation und deren Technologien (Internet of Things = IoT, Industrie 4.0) erfahren die Unternehmen, wie ihre Produkte und Dienstleistungen beim Kunden eingesetzt werden. Dies bedeutet: „[A]us B2B wird B2B2C“ (Gassmann/ Sutter 2016).

Digitalisierung, Industrie 4.0 und Smart Services

Es erfolgt eine Verschiebung der Machtverhältnisse zwischen OEM („Original Equipment Manufacturer“ bzw. „Erstausrüster/Originalausrüstungshersteller“) und Zulieferer. Zudem kommt es zu einer disruptiven Veränderung der Verkaufsmodelle. Signiﬁkante Wettbewerbsvorteile werden erlangt durch Service Engineering, Business Model Innova- tion sowie Know-how in Datenanalytik und datenbasierten Services, sogenannte Smart Services (vgl. Grohmann et al. 2017, S. 3ff.). Auf dieser Grundlage wird eine wertbasierte Kooperation zwischen Servicegeber und Servicenehmer als eine Kooperation der bei- den Wertschöpfungsprozesse möglich. Zudem kann auch eine Gewinnpartizipation sowie eine Teilung der Risiken der Wertschöpfungsergebnisse ermöglicht werden (vgl. Grohmann et al. 2017, S. 3ff.).

###### Cyber-physische Produktionssysteme

Industrie 4.0 beinhaltet neben einer vernetzten Produktion auch eine Vernetzung mit dem Kunden und weiteren Akteuren. Im Mittelpunkt der Industrie 4.0 steht die intelli- gente Fabrik, die sogenannte Smart Factory. Menschen, Maschinen und Produkte kom- munizieren mit neuer Technologie. Lieferanten, Kunden und auch Firmen anderer Bran- chen vernetzen sich. Der Informationsﬂuss zwischen den Akteuren basiert auf gemeinsamen Standards z. B. über eine Referenzarchitektur in Form von zur Verfügung gestellten Software-Applikationen und/oder Software-Services (vgl. Wehrlin 2018, S. 52). Häuﬁg wird Industrie 4.0 in erster Linie mit dem Einsatz neuartiger Technologie in Ver- bindung gebracht. Bei genauerer Betrachtung ist jedoch festzustellen, dass zahlreiche der entsprechenden Technologien bereits schon weit vor der Zeit bestanden, seit der explizit von Industrie 4.0 gesprochen wird.

Dies gilt unter anderem z. B. für Minicomputer, Radio Frequency Identiﬁcation (RFID) usw. In diesem Kontext verweist beispielsweise Armin Roth darauf, dass das wirklich Neue an Industrie 4.0 vor allem „in der Zusammenführung dieser Technologien im industriellen Umfeld zu einer einheitlichen gemeinsam agierenden Lösung“ (Roth 2016, S. 37) besteht (vgl. Kagermann 2014, S. 67ff.). Roth vertritt die Auffassung, dass ein sol- cher Verbund von standardisierter Kommunikationstechnik und Steuerungstechnik im Ergebnis zu der Verwirklichung der Idee hinter Industrie 4.0 steht. Er lässt sich beschreiben durch die fünf nachfolgend aufgeführten zentralen Paradigmen der industriellen Fertigung (PIF) (vgl. Roth 2016, S. 37):

* PIF 1 – vertikale und horizontale Integration,
* PIF 2 – dezentrale Intelligenz,
* PIF 3 – dezentrale Steuerung,
* PIF 4 – durchgängiges digitales Engineering,
* PIF 5 – cyber-physische Produktionssysteme.

Die genannten Paradigmen folgen einander in der Reihenfolge 1–5. Die direkt aufeinan- der folgenden Paradigmen überschneiden sich. Ein Paradigma bildet insofern auch zum Teil eine Schnittmenge zum nachfolgenden Paradigma.

Industrie 4.0 Hinter der Idee der Industrie 4.0 steht ein Verbund von standardisierter Kommunikations-

und Steuerungstech- nik. Dieser lässt sich durch fünf Paradig- men der industrie- llen Fertigung (PIF) beschreiben.

Bei den cyber-physischen Produktionssystemen handelt es sich um das fünfte neue Paradigma der industriellen Fertigung. Durch ein cyber-physisches Produktionssystem (CPS) wird die Gesamtheit einer Produktionsanlage im Zusammenhang mit dem Ansatz der Industrie 4.0 beschrieben (vgl. Roth 2016, S. 35ff.). Ein cyber-physisches Produkti- onssystem ist ein Konstrukt aus Produktionssystemen, welche Daten an das Steuer- ungssystem weiterleiten. Diese Datenweiterleitung erfolgt über Sensoren und Aktoren. Die gesammelten Daten werden im Steuerungssystem der Auswertung unterzogen. Danach werden die Daten vom Steuerungssystem wieder zurückgeleitet an die Produk- tion. Einem cyber-physischen Produktionssystem gehören zudem auch intelligente Pro- duktionsmittel an, die über Informationen ihres individuellen Produktionsprozesses verfügen. Die Daten und Dienste werden weltweit nutzbar über das Internet der Dinge bzw. Internet of Things und Cloud-Dienste (vgl. Roth 2016, S. 35ff.).

Produktionsverbund Sämtliche Objekte eines Produktions- verbundes erhalten eine neue Identität über das Internet. Die gegenseitige Ver- bindung produkti- onsrelevanter Sys- teme erfolgt über Schnittstellen.

Smart Services Sie sind eine Ergän- zung zu Smart Pro- ducts der Industrie

4.0.

Entsprechend dem Ansatz der Industrie 4.0 bekommen sämtliche Objekte eines Pro- duktionsverbundes eine neue Identität (IPv6-Adresse) über das Internet. Es entsteht eine gegenseitige Verbindung der produktionsrelevanten Systeme. Diese gegenseitige Verbindung erfolgt über Schnittstellen. Auf diese Weise wird es möglich, diese zu integ- rieren, zu optimieren und zu testen (vgl. Roth 2016, S. 35ff.).

Die reale und virtuelle Welt verschmelzen zunehmend miteinander. Hinzu tritt die Nut- zung von echtzeitfähigen Daten. Diese Entwicklungen eröffnen den Unternehmen neue Chancen und Möglichkeiten für die Unterstützung der Produktionsabläufe sowie der Abläufe in der Logistik. Als grundlegende Voraussetzung eines cyber-physischen Pro- duktionssystems wird eine vertikale Integration von allen Systemen in eine Art einheit- licher Systemlandschaft erachtet. Eine weitere Voraussetzung stellt die horizontale Integration für das komplette Wertschöpfungsnetzwerk dar. Die dezentral verteilten cyber-physischen Systeme können mithilfe eines Steuerungssystems relevante Infor- mationen abrufen, so z. B. Informationen hinsichtlich des aktuellen Produktionsprozes- ses, über den Rüstzustand sowie hinsichtlich der Auslastung (vgl. Roth 2016, S. 35ff.).

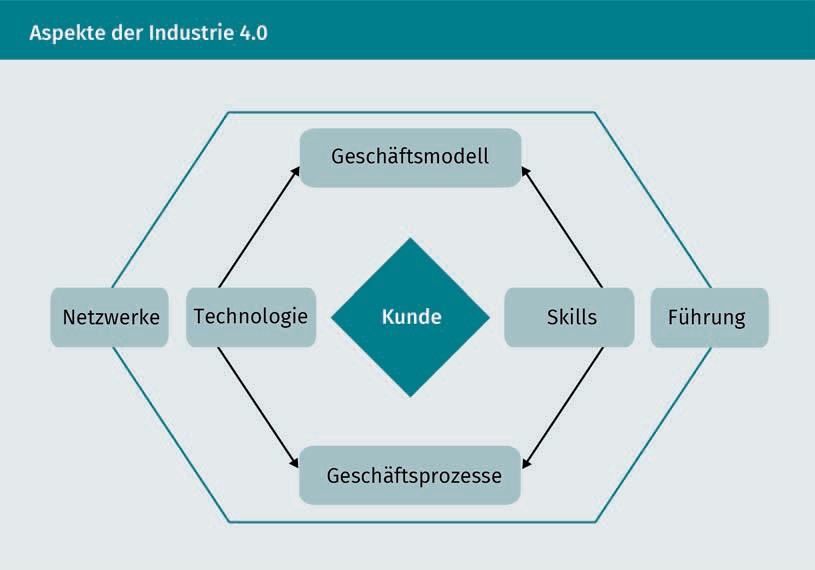
Das System ist auf der Grundlage dieser Informationen in der Lage, eigenständig Ent- scheidungen zu fällen. Zudem wird dem System eine situationsabhängige Reaktion ermöglicht. Insgesamt zeigt sich ein cyber-physisches System in Form eines kooperier- enden Gesamtsystems, welches im Zusammenhang mit der Industrie 4.0 steht (vgl. Roth 2016, S. 35ff.). Cyber-physische Produktionssysteme sind eine wichtige Basis, insbe- sondere in der verarbeitenden Industrie, für Smart Services.

### Smart Services in der Industrie 4.0

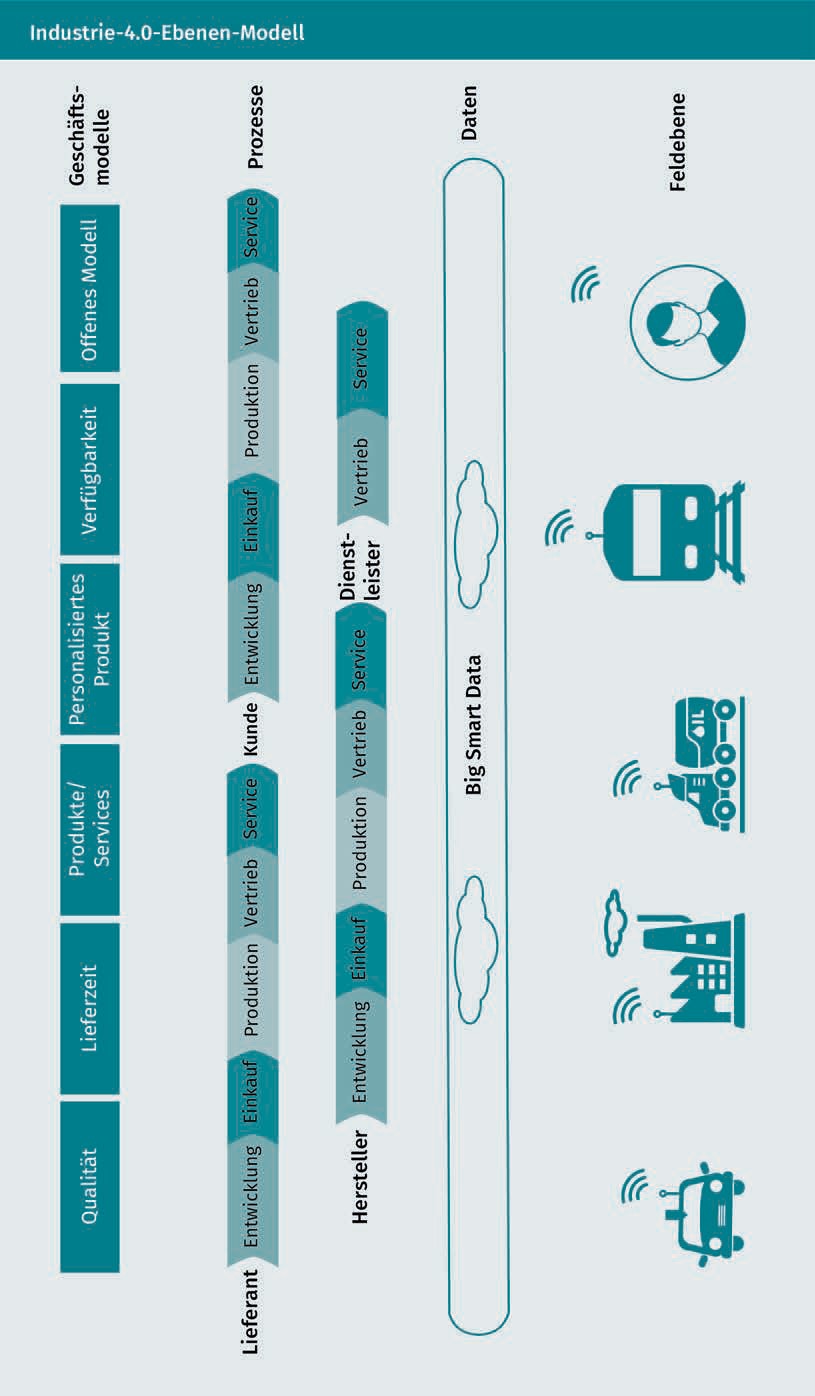
Smart Services gelten als eine Ergänzung zu den Smart Products der Industrie 4.0. Es wird davon ausgegangen, dass die neuen Formen von Dienstleistungen „intelligent“ sind und dass sie dies auch sein sollten. Es kommen unter anderem neue Technologien zum Einsatz, so z. B. Sensoren, Prozessoren, Künstliche Intelligenz (KI) und Cloud-Anbin- dung (vgl. Förderland 2018; BMWi 2017, S. 2ff.).

Digitalisierung, Industrie 4.0 und Smart Services

Durch den Einsatz neuer Technologien wird es möglich, Leistungen anzubieten, die für Nutzer maßgeschneidert sind. Die Leistungen erreichen die Nutzer zeitnah. Als Resultat entstehen intelligente Dienstleistungen sowie digital veredelte Produkte (vgl. Förder- land 2018; BMWi 2017, S. 4ff.). Die folgende Abbildung weist die Aspekte der Industrie 4.0 aus.

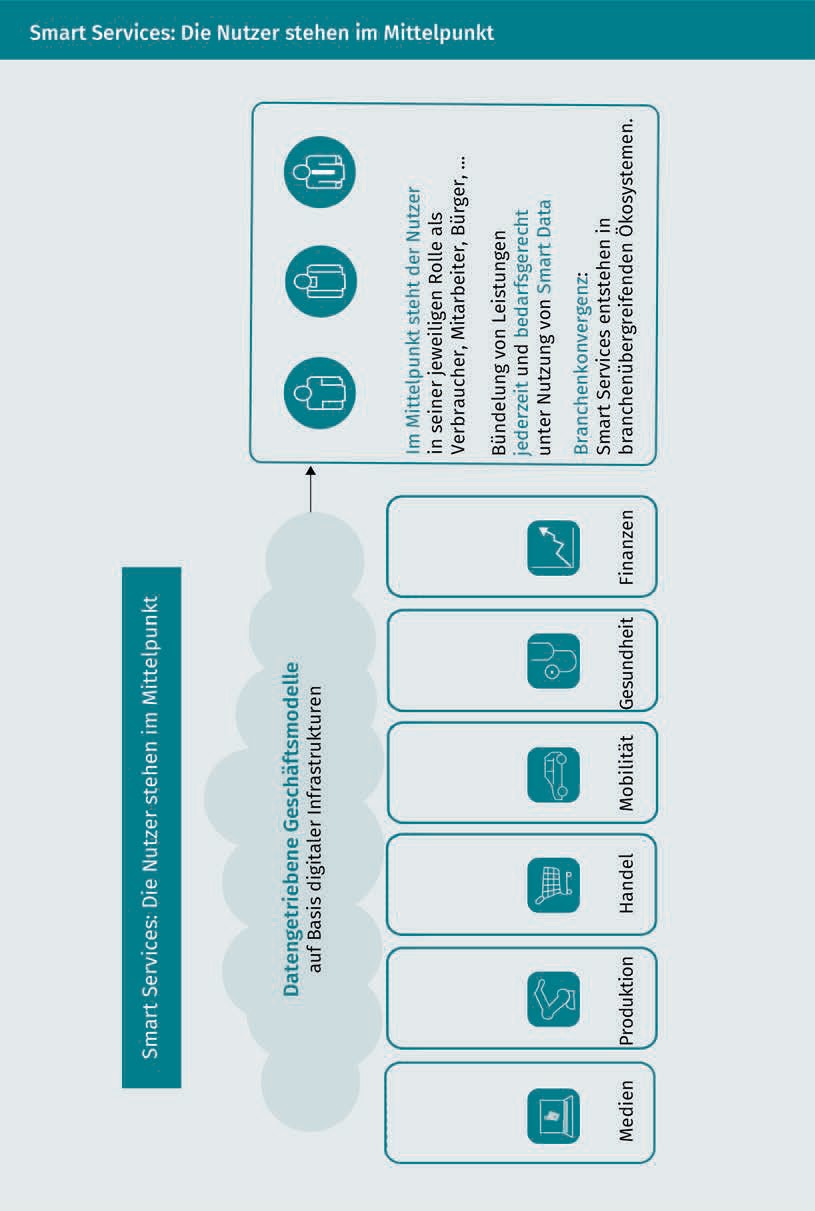


Aus den Aspekten der Industrie 4.0 wird bereits deutlich, dass der Nutzer bzw. der Kunde im Mittelpunkt der Digitalisierung und der Serviceleistungen steht (vgl. Borg- meier/Grohmann/Gross 2017, S. 110f.). Dabei unterliegen Leistungen schnellen Release- Zyklen. Dies ermöglicht eine agile Anpassung an die Herausforderungen und eine ent- sprechende Skalierung. Durch die erfolgende Disruption werden die bekannten Strukturen, Produkte, Angebote und Wertschöpfungsketten nicht mehr den aktuellen und künftigen Anforderungen gerecht. Durch den einschneidenden Umbruch werden große Zweige der Industrie verändert. Vor diesem Hintergrund wird deutlich, dass es erforderlich ist, dass sich die Unternehmen auf die vierte industrielle Revolution vorbe- reiten. Infolge von Smart Services ergibt sich eine Branchenkonvergenz. Die Ursache ist in der Entstehung von branchenübergreifenden Ökosystemen zu sehen. Die User sol- cher datengetriebenen Geschäftsmodelle stammen gegebenenfalls aus unterschiedli- chen Bereichen. In dieser Entwicklung bedarf es anstelle von Fabriken vermehrt Smart- Services-Rechenzentren. Es erfolgt ein Austausch und eine entsprechende Auswertung (Smart Data) von gewaltigen Datenmengen (Big Data) über die Cloud bzw. das Internet (vgl. Förderland 2018; BMWi 2017, S. 45; Kaufmann 2015, S. 7f.; Kagermann 2014, S. 67ff.). Die folgende Darstellung zeigt das Ebenen-Modell der Industrie 4.0 auf.



Digitalisierung, Industrie 4.0 und Smart Services

Die digitalen Geschäftsmodelle führen zu Umdenkungsprozessen. So richtet sich bei- spielsweise der Fokus anstatt auf „Ownership“ bzw. Besitztum zunehmend auf Ange- bote „as a Service“ (vgl. Förderland 2018; BMWi 2017, S. 45; Kagermann 2014, S. 67ff.). Diese Angebote stehen im Sinne der Nutzung eines Mietservice, z. B. Mietsoftware usw. Die nachfolgende Abbildung verdeutlicht den Beziehungskontext von Smart Services.



In der Wirtschaft erfolgt ein gravierender Paradigmenwechsel. Es steht nicht mehr der einzelne Anbieter mit seinen klassischen Produkten und Dienstleistungen im Mittel- punkt, sondern vielmehr der Nutzer mit seinen Prozessen (vgl. Kagermann 2014, S. 67ff.).

Digitalisierung, Industrie 4.0 und Smart Services

Der Nutzer erwartet als Kunde, Mieter, Patient, Mitarbeiter oder Reisender künftig, jederzeit sowie an jedem Ort die individuell auf ihn zugeschnittene Kombination von Produkten und Dienstleistungen zu erhalten (vgl. Kagermann 2014, S. 67ff.).

### Beispiele für Smart Services

Die kundenzentrierten Smart Services bzw. intelligente Dienstleistungen erfassen bei- nahe alle Bereiche. Durch Smart Services werden die Wertschöpfungsketten aufgebro- chen und zugleich revolutioniert, so z. B. in den Bereichen Produktion, Handel, Gesund- heitswesen, Mobilität, Medienkonsum oder Finanzen (vgl. Förderland 2018; BMWi 2017, S. 2ff., 20ff.; Kagermann 2014, S. 67ff.). Nachfolgend sind einige Beispiele dafür aufge- führt.

###### Automobil: Mobilität/Mobil ordern/Sharing/Dienstleistungsangebote nutzen

Infolge von Smart Services wird es unter anderem möglich, dass sich für manche Perso- nen der Kauf eines Autos nicht mehr lohnt bzw. es günstiger ist, über das Smartphone – in Abhängigkeit vom individuellen, situativen Bedarf und des Aufenthaltspunktes – ein Fahrzeug zu ordern. Dies wäre die Möglichkeit, welche sich dem Kunden über ein Carsharing-Angebot eröffnet. Eine weitere Form besteht in der Nutzung von Fahrdienst- leistern wie beispielsweise Uber (vgl. o. V. 2018a; BMWi 2017, S. 4ff.).

###### IT: Software as a Service (SaaS)

Kunden, die eine spezielle Software benötigen, können diese mieten, anstatt sie käuf- lich zu erwerben. Dies ist im Kontext von Software as a Service (SaaS) möglich. Die Soft- ware kann dabei für einen Bedarfszeitraum gemietet und genutzt werden (vgl. o. V. 2018a).

###### Handel: Bestand verwalten und ordern

So sind z. B. Kühlschränke informiert über den Inhalt. Sie verwalten die Bestände und kennen die Gewohnheiten, den üblichen Verbrauch und die Vorlieben der jeweiligen Eigentümer. Sie sind in der Lage, Nachschub zu ordern. Auf diese Weise entsteht Dis- ruption im Handel (vgl. o. V. 2018a).

###### Finanzdienstleistungen

Durch Fintech-Dienstleistungen wird eine schnellere und auch einfachere Bezahlung möglich (vgl. o. V. 2018a). Hierunter fallen weiter:

* Bezahldienste – Schnelle und einfache Bezahlung und damit digitale/elektronische Bezahldienstleistung,
* Robo-Advisor,
* Bezahldienste,
* Mobile Payment,
* Mikrokreditvergabe,
* Crowdlending,
* online/direct banking
* Crypto- Blockchain Anwendungen und
* digitalisierte Kreditvergabe oder Kapitalanlage.

Infolge von Smart Services wird es unter anderem möglich, dass sich für manche Perso- nen der Kauf eines Autos nicht mehr lohnt bzw. es günstiger ist, über das Smartphone – in Abhängigkeit vom individuellen, situativen Bedarf und des Aufenthaltspunktes – ein Fahrzeug zu ordern. Dies wäre die Möglichkeit, welche sich dem Kunden über ein Carsharing-Angebot eröffnet. Eine weitere Form besteht in der Nutzung von Fahrdienst- leistern wie beispielsweise Uber (vgl. o. V. 2018a; BMWi 2017, S. 4ff.).

Kunden, die eine spezielle Software benötigen, können diese mieten, anstatt sie käuf- lich zu erwerben. Dies ist im Kontext von Software as a Service (SaaS) möglich. Die Soft- ware kann dabei für einen Bedarfszeitraum gemietet und genutzt werden (vgl. o. V. 2018a).

###### Medien: Bedarfsgerecht anbieten***/***„On Demand“-Business

Der Kunde kann mit Smart Services jederzeit und bedarfsgerecht Dienstleistungen angeboten bekommen. Durch das entsprechende „On Demand“-Business wird es mög- lich, sowohl Ressourcen als auch Zeit einzusparen, so z. B. bei Streaming-Anbietern wie Netﬂix und Spotify. Diese sind dazu in der Lage, den Kunden über das Netz schnell und auch kostengünstig passende Medieninhalte anzubieten. Mediale Smart Services wer- den zunehmend auch in anderen Bereichen erfolgreich eingesetzt (vgl. o. V. 2018a).

###### Medizin

Die Anwendung von Smart Services bzw. intelligenten Dienstleistungen erfolgt zuneh- mend auch im Bereich der Medizin. Eine stärkere Vernetzung von Ärzten, Patienten und beispielsweise OP-Sälen kann zu einer höheren Efﬁzienz in der Versorgung führen. Zudem sind bessere Behandlungserfolge möglich (vgl. o. V. 2018a; BMWi 2017, S. 14ff.).

###### Maschinenbau: Digitale Plattformen/vollautomatische Marktplätze/ Produktion optimieren/aktive Maschinenauslastung

Smarte Dienstleistungen bringen Innovationen hervor. Dies gilt sowohl im privaten Sek- tor als auch in der Wirtschaft oder Verwaltung. Vor allem in der Industrie sind die mit smarten Dienstleistungen in Verbindung stehenden Innovationen relevant. Von diesen

Digitalisierung, Industrie 4.0 und Smart Services

hängt unter anderem die (internationale) Wettbewerbsfähigkeit ab. Gegenwärtig und zukünftig führen digitale Plattformen zur Entstehung und Entwicklung von vollautoma- tisierten Marktplätzen. Durch KI lässt sich die Produktion fortlaufend optimieren. Zudem ist es möglich, dass sich Maschinen um Aufträge bewerben, wobei sie dies aktiv und eigenständig übernehmen (vgl. o. V. 2018a; BMWi 2017, S. 20ff.).

Durch Smart Services im Maschinenbau entstehen vollkommen neue Herausforderun- gen. Diese haben häuﬁg auch neue Geschäftsmodelle und deren Weiterentwicklung zur Folge (vgl. BMWi 2017, S. 20ff., S. 45).

Zusammenfassung

Die Digitalisierung ist ein bedeutendes Thema von großer Tragweite. Sie führt u. a. zu einschneidenden Veränderungen für alle wirtschaftlichen und gesellschaftlichen Bereiche. Digitalisierung führt dazu, dass sich wertbasierte Partnerschaften und Koproduktionen wie auch Know-how in Big Data bzw. Datenanalytik als signiﬁkante Wettbewerbsvorteile herausstellen. Dadurch erfolgt zugleich eine Substitution und Verdrängung der zuvor geltenden klassischen Entwicklungsbedingungen und -vor- teile eines Produktes bzw. einer Dienstleistung.

Cyber-physische Produktionssysteme sind eine wichtige Basis, insbesondere in der verarbeitenden Industrie, für Smart Services. Diese sind ein Konstrukt aus Produkti- onssystemen, welche Daten an das Steuerungssystem weiterleiten. Diese Datenwei- terleitung erfolgt über Sensoren und Aktoren. Die gesammelten Daten werden im Steuerungssystem der Auswertung unterzogen. Danach werden die Daten vom Steu- erungssystem wieder zurückgeleitet an die Produktion. Einem cyber-physischen Produktionssystem gehören zudem auch intelligente Produktionsmittel an, die über Informationen ihres individuellen Produktionsprozesses verfügen. Die Daten und Dienste werden weltweit nutzbar über das Internet der Dinge bzw. Internet of Things (IoT) und Cloud-Dienste. Entsprechend dem Ansatz der Industrie 4.0 bekom- men sämtliche Objekte eines Produktionsverbundes eine neue Identität über das Internet. Es entsteht eine gegenseitige Verbindung der produktionsrelevanten Sys- teme. Diese gegenseitige Verbindung erfolgt über Schnittstellen. Für eine solche Entwicklung sind anstelle von Fabriken vermehrt Smart-Services-Rechenzentren erforderlich. Es erfolgt ein Austausch und eine entsprechende Auswertung (Smart Data) von gewaltigen Datenmengen (Big Data) über die Cloud bzw. das Internet.

Durch den Einsatz neuer Technologien vor allem Sensoren, Prozessoren, Künstliche Intelligenz (KI) und Cloud-Anbindung, können kundenindividuelle Leistungen bzw. intelligente Dienstleistungen oder Smart Services angeboten werden. In Folge die- ser verändern sich Wertschöpfungsketten mitunter grundlegend. Die hierdurch ent- stehenden digitalen Geschäftsmodelle führen zu Umdenkungsprozessen. So kann sich beispielsweise der Fokus anstatt auf „Ownership“ bzw. Besitz zunehmend auf Angebote „as a Service“ richten. Smarte Dienstleistungen bringen Innovationen her-

vor. Von diesen hängt unter anderem die (internationale) Wettbewerbsfähigkeit ab. Gegenwärtig und zukünftig führen digitale Plattformen zur Entstehung und Entwick- lung von vollautomatisierten Marktplätzen.

# Lektion 2

## Digitalisierung und

## Disruption



#### LERNZIELE

Nach der Bearbeitung dieser Lektion werden Sie …

… digitale Geschäftsmodell kennen.

… den Zusammenhang von Value Added Services und Smart Services verstehen.

… mit den Anknüpfungspunkten für neue Geschäftsmodelle vertraut sein.

… die Grundprinzipien des Intermediär-Geschäftsmodells verstehen.

… Beispiele disruptiver Geschäftsmodelle kennen.

DL-D-DLBINGSS01-L02

1. Digitalisierung und Disruption

### Einführung

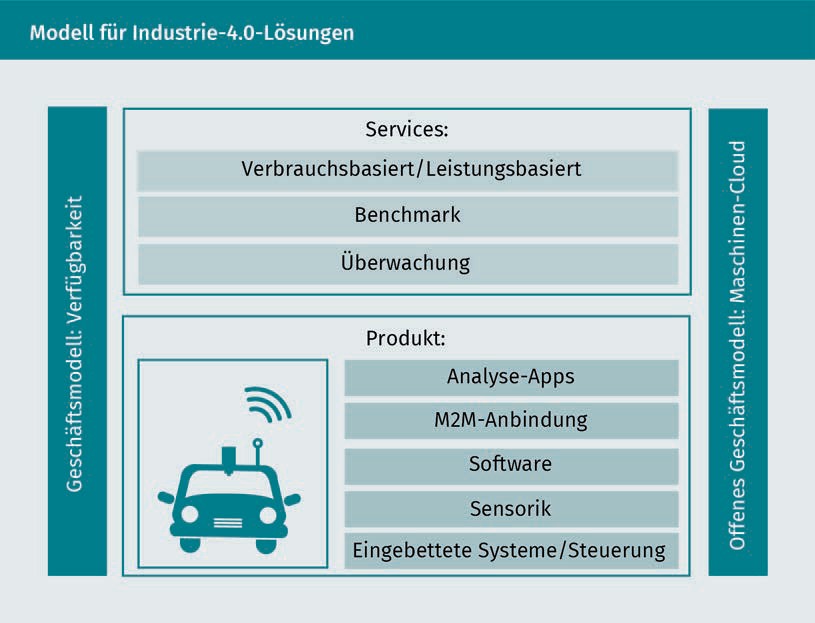
Smart Data entsteht durch die Verknüpfung einer großen Menge an Daten (Big Data). Auf der Basis von Smart Data kann Wissen generiert werden, das die Grundlage für Geschäftsmodelle bildet. Big Data wird zu Smart Data veredelt sowie in neuen, indivi- duell kombinierbaren Smart Products und Smart Services monetarisiert. Damit erschließen sich neue Geschäftsmodelle, die eine innovative Nutzung des hohen Potenzials der Datenmenge erlauben. Durch Smart Data wird es möglich, Kundenwün- sche zu antizipieren und zu befriedigen, noch bevor der Nutzer selbst von ihnen Kennt- nis hat. Es vollzieht sich ein Wechsel von produktorientierten hin zu nutzerzentrierten Geschäftsmodellen. Dafür bedarf es eines Umdenkens. Dieses Umdenken ist insbeson- dere in sämtlichen traditionellen Unternehmen erforderlich. Für Unternehmen, die ihre (internationale) Wettbewerbsfähigkeit erhalten oder verbessern möchten, ist es uner- lässlich, die Chancen zu nutzen, welche ihnen das Internet of Things (IoT) bietet. Die Unternehmen sind demnach gefordert, ihre Geschäftsmodelle proaktiv zu überarbeiten. Gegebenenfalls ist ein eigenes disruptives Denken und Handeln notwendig. Durch Intermediäre werden Plattformen zur Verfügung gestellt, die Anbieter wie auch Nach- frager schnell und bequem zusammenbringen. Gleichzeitig ist es den Intermediären bzw. Agenten möglich, mit der Zeit die Kontrolle über die Kundenschnittstelle zu bekommen.

### Digitale Geschäftsmodelle

Durch smarte, datengetriebene Services entwickeln sich in der Industrie wie auch im Mittelstand neue Geschäftsmodelle. Diese sind unter anderem interessant aus der Per- spektive der Kundenbindung und als zweites Standbein (vgl. Steinhaus 2018; Kager- mann 2014, S. 67ff.).

Durch Smart Data sowie die darauf basierenden Smart Products und Smart Services lassen sich die generierten Datenmengen innovativ nutzen (vgl. Kagermann 2014, S. 67ff.; Kaufmann 2015, S. 12ff.; Steimel 2016). Die folgende Graﬁk verdeutlicht ein Lösungsmodell für Industrie-4.0-Lösungen.

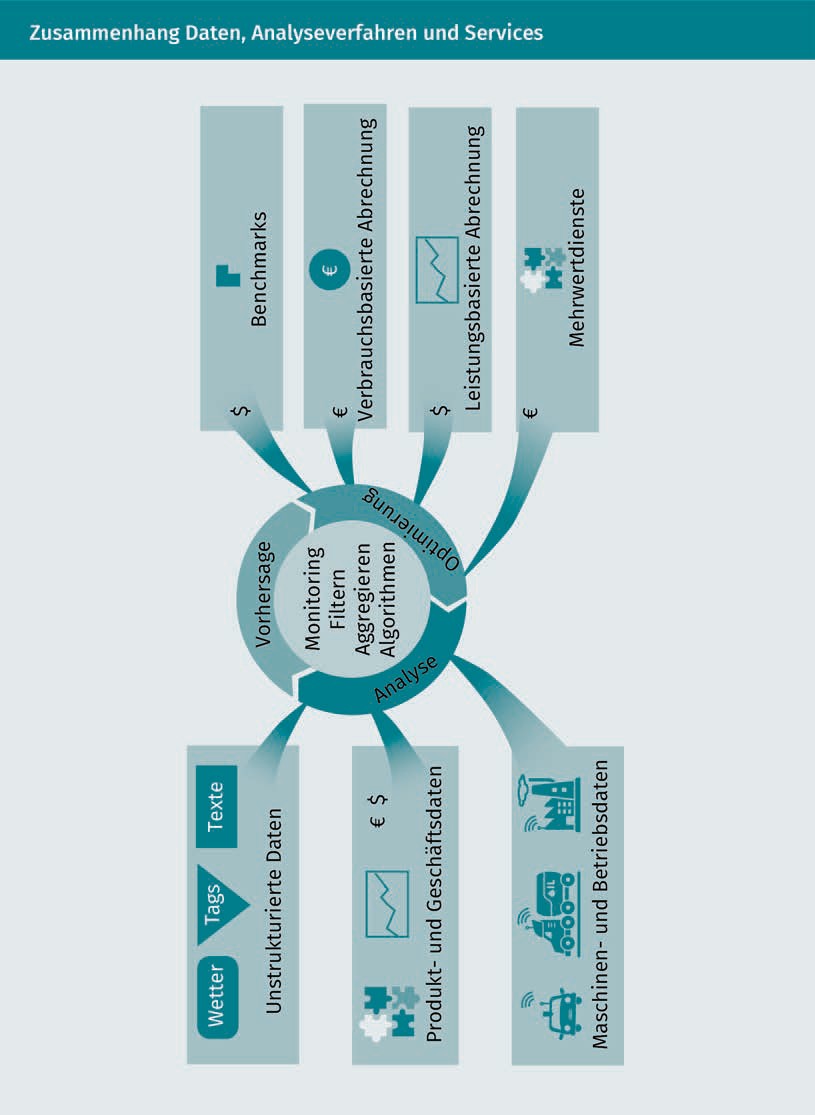
Digitalisierung und Disruption



Zahlreiche produzierende Unternehmen entwickeln sich vom Produkt- zum Lösungsan- bieter. Sie produzieren nicht nur Produkte, sondern bieten zusätzliche Services an. Bei- spiel: Der Hersteller einer Digitalkamera liefert auch die entsprechende Software zur Bildbearbeitung als Lösung mit. Dies bedeutet, dass das Servicegeschäft eine größere Bedeutung erhält. Dies beinhaltet neue Wege zum Kunden und eine Anbindung der Kunden. Die Bedürfnisse der Kunden können wahrgenommen und berücksichtigt wer- den. Dadurch sind neue Geschäftsmodelle geprägt (vgl. Kagermann 2014, S. 16; Kauf- mann 2015, S. 12ff.).

Werden die digitalen Daten genutzt und neu kombiniert, ergeben sich unter Umständen daraus (in Abhängigkeit von den Marktgegebenheiten) disruptive Geschäftsmodelle. Es ist jedem Unternehmen möglich, von Smart Data zu proﬁtieren (vgl. Kagermann 2014, S. 67ff.). Jedes Unternehmen kann auf dieser Grundlage neue Geschäftsmodelle etablie- ren. Dies gilt für Beratungsunternehmen zur Analyse und Nutzung von Smart Data, für produzierende Unternehmen wie auch für vorwiegend datengetriebene Dienstleister. Die Nutzung von Diensten und Softwaresystemen ist künftig auch für kleine und mittel- große Unternehmen (KMUs) möglich, auch wenn diese derzeit infolge der aktuellen Lizenz- und Geschäftsmodelle noch als nicht ﬁnanzierbar gelten (vgl. Steimel 2016; Kagermann 2014, S. 67ff.; Kaufmann 2015, S. 12ff.). Die folgende Darstellung zeigt die Zusammenhänge von Daten, Analyseverfahren und Services auf.

Disruptive Geschäftsmodelle Diese entstehen durch die Nutzung und neue Kombina- tion der digitalen Daten.



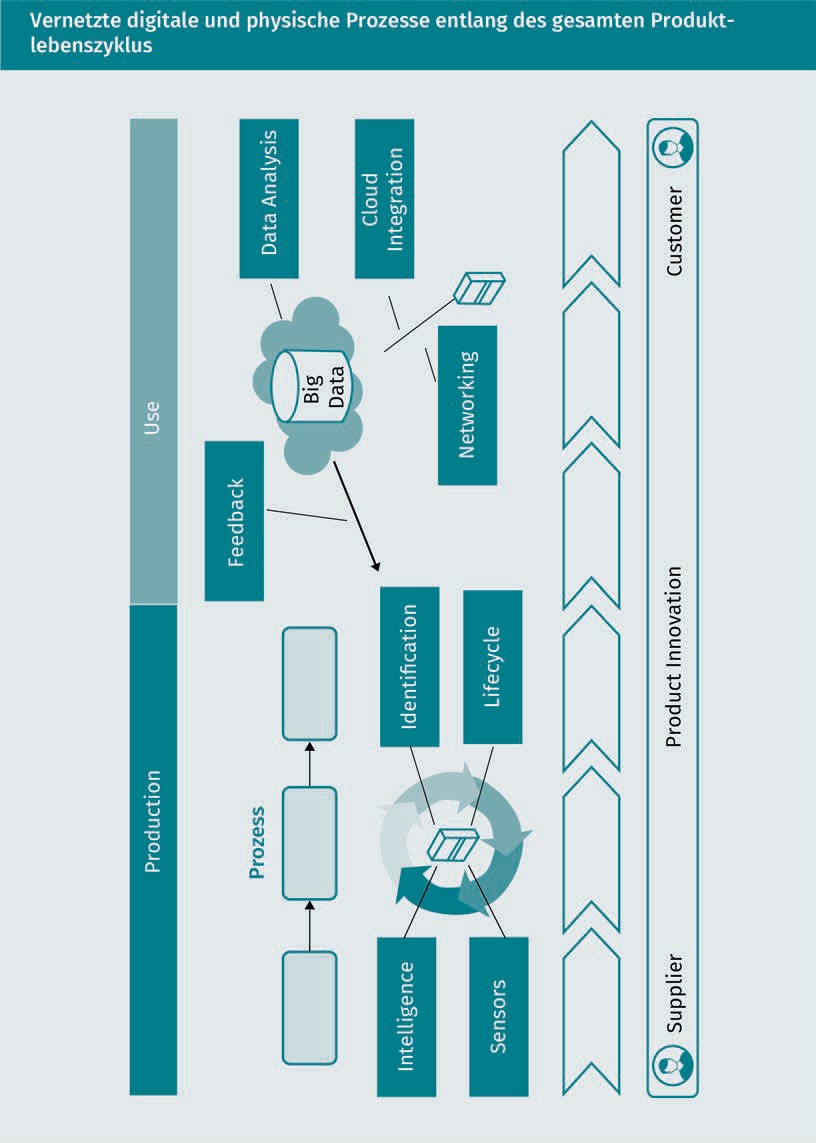
###### Geschäftsmodelle/Monetarisierung von Smart-Service-Modellen

Die Monetarisierung von „As a Service“-Modellen rückt zunehmend in den Mittelpunkt der Betrachtung. Mit dem Einsatz von Smart Data wird es möglich, Dienstleistungen günstiger anzubieten als zuvor. Die gewonnenen Daten können so verarbeitet und

Digitalisierung und Disruption

genutzt werden, dass sie dem Kunden einen Mehrwert bieten. Dem Kunden können auf ihn zugeschnittene oder zu ihm passende Angebote unterbreitet werden, was sowohl dem Kunden selbst als auch dem Unternehmen in Form von Umsatzsteigerungen nützt (vgl. Förderland 2018; BMWi 2017, S. 45; Kaufmann 2015, S. 12ff.).

Mittelständische Unternehmen können derartige datengetriebene Services als Produk- tionsbetriebe nutzen, um damit gegebenenfalls ein zweites Standbein aufzubauen, wel- ches zum Stammgeschäft bzw. ursprünglichen Produkt/Dienstleistung hinzukommt und auch eine Art Sicherungsfunktion ausübt. Durch Smart Services wird häuﬁg auch ein kontinuierlicher Kundenkontakt begünstigt. Dadurch erfolgt eine entsprechende Ver- besserung der Bindung an den Hersteller. Im Zusammenhang mit der engmaschigen Betreuung ist es auch denkbar und möglich, dass Kunden eine Art Funktion von „Mit- konstrukteuren“ bzw. „Co-Konstrukteuren“ tragen. Die Kunden verfügen aufgrund des täglichen Einsatzes der Produkte und/oder Dienstleistungen meist über ein wertvolles Erfahrungswissen (vgl. Steinhaus 2018). Dieses Wissen ist für die Unternehmen über Feedback nutzbar, z. B. die Auswertung von Kundenreklamationen oder Bewertungen mit anschließender Nutzung zur Produkt- und Dienstleistungsverbesserung. Die auf diese Art gewonnenen Informationen können wichtige Beiträge für Verbesserungen von Produkten, Dienstleistungen und Prozessen bzw. neuen Lösungen im Sinne des konti- nuierlichen Verbesserungsprozesses (KVP) bewirken (vgl. Wehrlin 2014, S. 146ff.). In der nachfolgenden Darstellung sind vernetzte digitale und physische Prozesse entlang des gesamten Produktlebenszyklus abgebildet.



Im Allgemeinen gelten datengetriebene Dienstleistungen als kostengünstig. Ist eine digitale Lösung entwickelt, verursacht sie für die zusätzlichen Kunden lediglich geringe Kosten. Die Entwicklungskosten sind in der Regel anfänglich bzw. im Einzelfall hoch. Sie müssen jedoch nicht für jeden Kunden neu erbracht werden (vgl. Steinhaus 2018).

Digitalisierung und Disruption

### Anknüpfungspunkte für neue Geschäftsmodelle

Strategien für Änderung und Innovation berücksichtigen unter anderem gegenwärtige und zukünftige Bedürfnisse, Bedarfe, Wünsche, Anforderungen – kurzum die Nachfrage der Kunden – nach/an entsprechenden Produkten und Dienstleistungen. Im Idealfall sind diese individuell und optimal auf den einzelnen Kunden zugeschnitten. Hierbei bilden Smart Products bzw. Smart Services und innovative Geschäftsmodelle eine wesentliche Grundlage.

###### Smart Products

Bei Smart Products bzw. intelligenten Produkten handelt es sich um internetfähige Pro- dukte, die mit Sensoren ausgestattet sind. Smart Products sind fähig zur Sammlung von Daten. Sie analysieren die Daten und können diese sowohl versenden als auch emp- fangen. Aus dieser Perspektive werden sie als „intelligent“ bezeichnet. Smart Products bzw. intelligente Produkte können künftig die folgenden Funktionen erfüllen (vgl. Stei- mel 2016):

* + - Smart Products können Aufgaben selbstständig lösen.
    - Sie sind in der Lage, mit anderen Dingen zu kommunizieren.
    - Intelligente Produkte passen sich eigenständig Änderungen an.
    - Sie wandeln sich, damit sie die Nutzerbedürfnisse weitgehend erfüllen.
    - Smart Products können sich selbstständig updaten.
    - Sie steuern ihre laufenden Kosten eigenständig und senken diese.
    - Intelligente Produkte optimieren sich selbst.
    - Sie erhöhen eigenständig ihre Produktivität.
    - Smart Products können Gefahren bzw. Ausfälle antizipieren.
    - Sie sind in der Lage, proaktiv zu agieren.

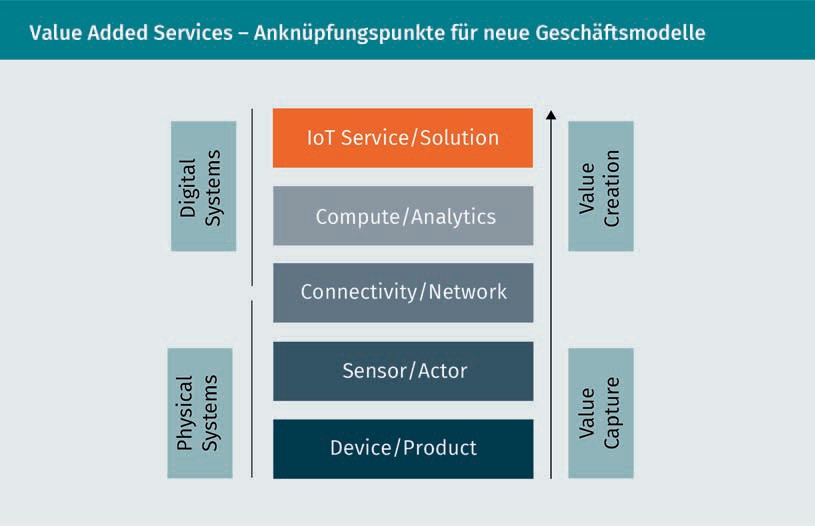
###### Von Value Added Services zu Smart Services

Ist es den Produkten möglich, Daten zu sammeln, bzw. existieren smarte Produkte, sind Unternehmen in der Lage, ihren Kunden Mehrwerte anzubieten, die über den eigentli- chen Produktnutzen hinausgehen. Diese zusätzlichen Serviceleistungen werden auch als Value Added Services bezeichnet. Daraus entstehen Smart Services (vgl. Steimel 2016; Kagermann 2014, S. 67ff.). Ein Unternehmen der Energiewirtschaft bietet beispiels- weise dem Kunden neben der Energie auch den zusätzlichen Service der täglichen Ver- brauchsübersicht in Form einer App an. Der Kunde sieht nun genau, wann und wo er wie viel Strom oder Gas zu welchem Preis verbraucht (hat).

Smart Products Diese sind intelli- gente Produkte, die internetfähig und mit Sensoren ausge- stattet sind sowie Daten analysieren, senden und empfan- gen können. Sie erfüllen vielfältige Funktionen.

###### Innovative Geschäftsmodelle/Service-Innovation – Weiterentwicklung von Geschäftsmodellen

Diese Entwicklung führt unter anderem dazu, dass sich Geschäftsmodelle nicht mehr lediglich auf eine reine Produktorientierung beschränken. Vielmehr sind die Geschäfts- modelle zusätzlich daten- und servicegetrieben. Als Konsequenz ergeben sich zahlrei- che Anknüpfungspunkte zur Gestaltung und Weiterentwicklung von zuvor noch unbe- kannten, also innovativen bzw. neuen Geschäftsmodellen (vgl. Steimel 2016). Die nachfolgende Abbildung verdeutlicht die Anknüpfungspunkte für neue Geschäftsmo- delle.

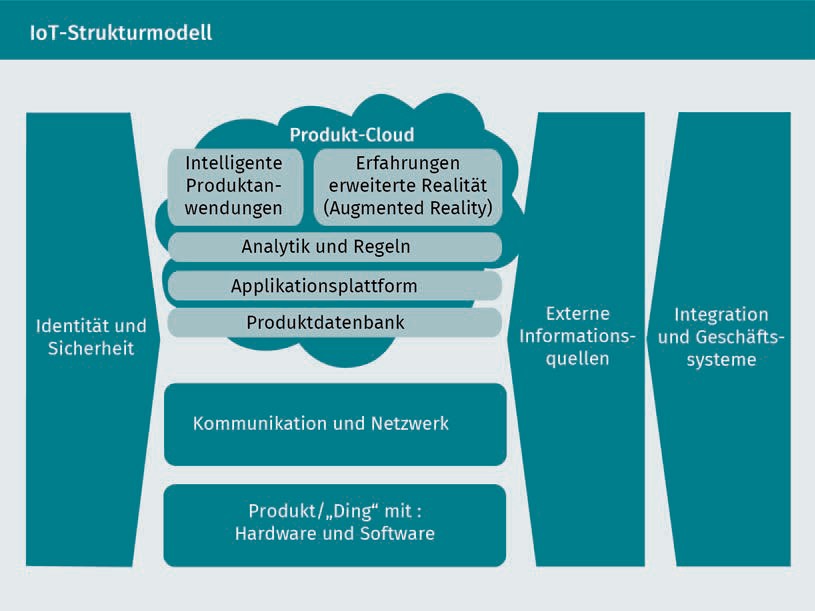


Smart Services ermöglichen es, den Nutzer in den Mittelpunkt zu stellen. Ein individuel- ler Service wird durch Smart Data möglich (vgl. Kagermann 2014, S. 67ff.). Es lassen sich ein personalisiertes Kundenerlebnis sowie ein individueller Service verwirklichen, wie im obigen Beispiel eines Energieanbieters. Dieser bietet nicht nur Strom oder Gas an, sondern auch die Steuerung von Geräten im Smart Home, die Optimierung der Energie- versorgung aus dem intelligenten Netz, die individuelle Verbrauchsoptimierung usw. In der Zukunft gewinnt Service-Innovation zunehmend an Relevanz (vgl. Steimel 2016).

„Every product is a service waiting to happen. […] Designing with data in mind. […] Achieving continual service change“ (Accenture zit. nach Steimel 2016).

Digitalisierung und Disruption

Für die Anbieter von Smart Services ist ein umfassendes Verständnis erforderlich hin- sichtlich des Nutzers, der Verhaltensweisen des Nutzers sowie der Ansprüche des Nutz- ers (vgl. Steimel 2016) Es zeigt sich ein Wechsel von produktorientierten hin zu nutzer- zentrierten Geschäftsmodellen (vgl. Kagermann 2014, S. 67ff.). Die Zusammenhänge werden in der folgenden Abbildung zum IoT-Strukturmodell verdeutlicht.



In enger Verbindung mit dem IoT-Strukturmodell steht das Digital Enabling. Dieses bezieht sich jedoch vorwiegend auf die technische Ausgestaltung von Plattformen. Hierzu bietet der Ansatz der Industrie 4.0 eine wesentliche Basis. Durch die technische Ausgestaltung der Plattform eines Unternehmens wird eine Differenzierung von den Wettbewerbern ermöglicht, so z. B. Markteintritts- und Wechselbarrieren. Der wirt- schaftliche Erfolg knüpft an fünf Wertschöpfungsebenen an: materielles Objekt, Senso- ren/Aktoren, Konnektivität, Datenanalytik und digitale Dienstleistung. Begrifﬂich wird ein „Datensee“ bzw. eine Ansammlung von Daten etabliert. Daten werden unterneh- mensintern und auch extern von Kunden, z. B. entsprechend ihren Gewohnheiten, Ver- haltensweisen, ihrem Konsum usw., erhoben und gesammelt, verbunden sowie einer Analyse unterzogen (vgl. Borgmeier/Grohmann/Gross 2017, S. 271f.).

Digital Enabling Beim Digital Enab- ling handelt es sich um die Ausgestal- tung und Befähigung von Plattformen.

### Digitale Intermediäre

###### Digitale Transformation – Grundprinzipien des Intermediär-Geschäftsmodells

Intermediäre Sie stellen Plattfor- men zur Verfügung, welche Anbieter und Nachfrager schnell und bequem zusam-

menbringen. Intermediäre können die Kontrolle über die Kundenschnitt- stelle bekommen.

Intermediäre stellen Plattformen zur Verfügung, welche Anbieter und Nachfrager zusammenbringen sowie Anbieter und Nachfrageer verwaltet und betreut. Intermediäre bzw. Agenten können die Kontrolle über die Kundenschnittstelle erlangen. Beispiel: Ein Hotelbuchungsportal verhindert, dass ein Gast direkt bei einem angeschlossenen Hotel bucht. Selbst wenn der Gast direkt auf die Homepage des Hotels geht, wird er, wenn er ein Zimmer buchen möchte, auf das Buchungsportal umgeleitet (vgl. Keuper 2015; Kagermann 2014, S. 67ff.).

Eine Analyse und Bewertung des Intermediär-Geschäftsmodells führt zu den dessen Grundprinzipien (vgl. Hebenstreit 2016):

* Verhalten fokussieren,
* Geschäftsmodell ﬂexibilisieren,
* Dynamik/Schnelligkeit,
* evolutionäre Struktur und Arbeitsweise.

Intermediär-Geschäftsmodell-Grundprinzip 1 – Verhalten fokussieren

Dem Unternehmen mit Intermediär-Geschäftsmodell wird durch sein Angebot der Zugang zu Leistungen und zum Markt vereinfacht. Für die Anbieter von Produkten und Dienstleistungen führt dies zu einem schnelleren Zugang und zusätzlichem Umsatz. Für die Kunden bedeutet dies einen Zugang zu passgenauen Dienstleistungen und/oder Produkten. Das Unternehmen mit Intermediär-Geschäftsmodell ermöglicht durch sein Angebot den zwei Zielgruppen das Realisieren von persönlichen wie auch beruﬂichen Zielen. Damit werden unmittelbar verhaltenspsychologisch wirksame Attribute relevant und mit einbezogen (vgl. Hebenstreit 2016). Beispiele sind pizza.de, mytaxi, Check24, immoscout, myhammer usw.

Intermediär-Geschäftsmodell-Grundprinzip 2 – Geschäftsmodell ﬂexibilisieren

Das Intermediär-Geschäftsmodell gilt stark virtualisiert. Der Intermediär lagert einen Großteil der Wertschöpfung an seine Partner aus. Dagegen werden Marketing und Ver- waltung im Unternehmen abgebildet. Die Basis des Geschäftsmodells bildet ein digita- les System aus Webangebot, Anfragestrecke (Weg und Etappen einer Anfrage) und eCRM (Electronic Customer Relationship Management bzw. eBeziehungsmanagement zum Kunden). Auf der Basis eines bestimmten fachlichen Schwerpunktes ist es dem Intermediär möglich, weitere fachliche Bereiche mit relativ geringem Aufwand hinzuzu- nehmen. Es ist vorstellbar, das Angebot national zu beginnen und mit der Zeit zu inter- nationalisieren. Aus dem Intermediär-Geschäftsmodell entwickelt sich eine umfangrei- che und detaillierte Kundendatenbank, die jederzeit die Erschließung weiterer strategischer Schritte in neue Geschäftsfelder ermöglicht (vgl. Hebenstreit 2016), wie es beispielsweise bei Netﬂix und Amazon der Fall war.

Digitalisierung und Disruption

Intermediär-Geschäftsmodell-Grundprinzip 3 – Dynamik/Schnelligkeit

Die digitale Technologie stellt die Basis der bedeutendsten Elemente des Intermediär- Geschäftsmodells dar. Auf dieser Grundlage ist es möglich, sie leicht zu skalieren und anzupassen. Es ist denkbar, neue Werte- und Nutzenversprechen direkt mit den Kunden zu testen. Auch die Auslagerung der Erbringung von Leistungen an Partner wirkt sich günstig auf das Intermediär-Geschäftsmodell aus. Das Risiko der Entwicklung, die zeitli- che Belastung sowie die operativen Prozesse werden vollständig an die Anbieter ausge- lagert. Für das Unternehmen wird es möglich, schnell und unkompliziert auf Entwick- lungen am Markt zu reagieren oder neue Angebote zu platzieren. Als gegebenenfalls problematisch wird lediglich der stark vorherrschende Fokus auf zwei speziﬁsche Ziel- gruppen erachtet (vgl. Hebenstreit 2016). Beispiele sind Check24, Facebook usw.

Intermediär-Geschäftsmodell-Grundprinzip 4 – evolutionäre Struktur und Arbeitsweise

Intermediäre sind mit digitalen Systemen tätig. Dadurch ist es ihnen möglich, sämtliche Vorteile des Lean-Start-ups und des Customer Developments zu nutzen. Intermediäre eignen sich, um analytisch auf der Grundlage von digitalen Daten passgenaue Angebote für Unternehmen und Endnutzer kontinuierlich weiterzuentwickeln (vgl. Hebenstreit 2016).

###### Intermediär-Geschäftsmodell

Intermediäre zeichnen sich dadurch aus, dass sie in Märkten mit hoher Intransparenz (wenn beispielsweise Kunden den Markt bzw. die Angebote nicht gut überblicken kön- nen) agieren. Bei diesem Geschäftsmodell werden Anbieter und Nachfrager auf eine neue bzw. innovative Art und Weise miteinander verbunden. Die Anbieterseite weist in der Regel kleinere Unternehmen und Selbstständige bzw. KMUs auf. Meist reicht deren Marketing-Power nicht über die Grenzen lokaler Märkte hinaus. Zudem ist es für den Konsumenten oft nicht möglich, die Qualität der Produkte/Dienstleistungen einzu- schätzen (vgl. Hebenstreit 2016).

Der Mittelstand bzw. kleine und mittelständische Unternehmen gelten als das „Rück- grat“ oder der „Motor“ der Wirtschaft. „Der Standort Deutschland proﬁtiert von der Innovationskraft der ‚Hidden Champions‘ und Marktführern für Nischenprodukte“ (Matouschek 2017).

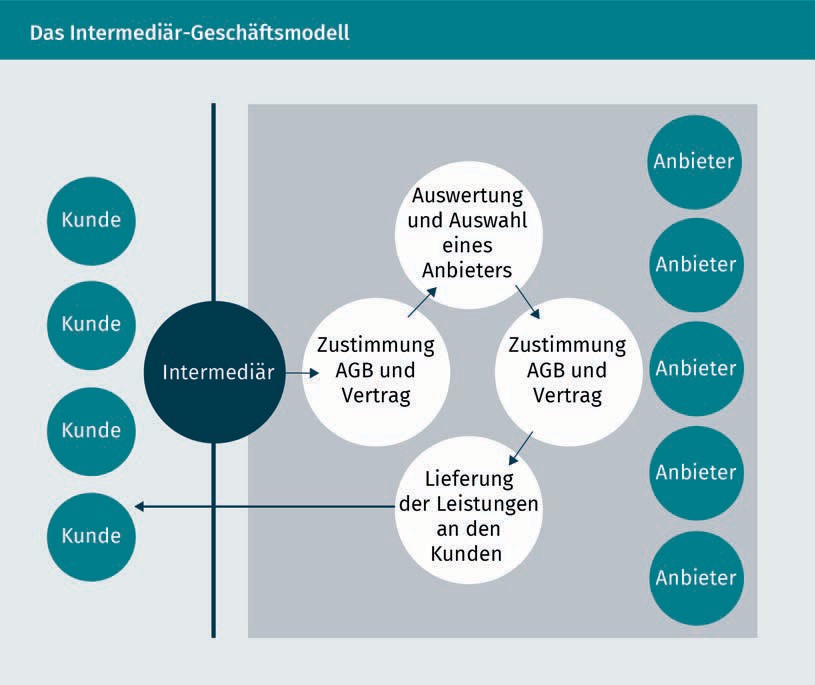
Wie Studienergebnisse der Beratungsgesellschaft PwC (Studie zu Versicherungen bei den KMUs, vgl. Matouschek 2017) aufzeigen, besteht eine hohe Nachfrage nach Digita- langeboten von Versicherungen. Jedoch wird diese hohe Nachfrage noch nicht gedeckt. Daraus resultieren neue Chancen und Herausforderungen. So ist es erforderlich, dass Versicherungen Multi-Channel-Strategien entwickeln. Diese Strategien sollten die Kun- denansprüche erfüllen. Onlineangebote sollten auch persönlicher Fachexpertise und Beratung gerecht werden. Die Geschäftsmodelle, die in einzelnen Märkten bereits exis- tieren, sollten möglichst weitgehend den Bedarf digital decken (vgl. Matouschek 2017).

Intermediäre Geschäftsmodelle bzw. Agentenmodelle werden zunehmend bedeutend, weil das empirische Wissen über den Kunden wie auch über dessen Bedürfnisse von grundlegendem Wert ist. Durch E-Commerce-Marktplätze, wie z. B. Alibaba, Amazon oder eBay, wurden Größenvorteile erzielt und eine gewaltige Marktmacht aufgebaut. Viele Anbieter, die auf deren Dienste regelrecht angewiesen sind, können kaum noch darauf verzichten. Es entwickelt sich ein gewisses Abhängigkeitsverhältnis. Der digitale Kundenzugang ist neben dem Handel auch in der Logistik relevant, z. B. für 4PL-Provi- der (Fourth-Party Logistics) ohne eigene Transportinfrastruktur. Diese können mittels Aggregation verschiedener Angebote den optimalen Frachtführer für den Kunden aus- wählen. Dafür können sie eine entsprechende Provision beziehen (vgl. Keuper 2015).

Intermediäre Sie schaffen Sicht- barkeit, Zugang und Bewertungen im

Markt.

Durch Intermediäre werden Sichtbarkeit, Zugang und Bewertungen im Markt geschaf- fen. Sie ermöglichen dem Nachfrager dadurch den Zugriff auf zuvor ungenutzte Res- sourcen. Zudem führen Intermediäre dem Anbieter neue Aufträge zu. Die Unternehmen erzielen den Umsatz durch eine Provision für die Aufträge, die sie vermitteln. Das Inter- mediär-Geschäftsmodell lässt sich exemplarisch am Unternehmen CoMatch verdeutli- chen. CoMatch vermittelt Top-Management-Berater an solche Kunden, die einen ent- sprechenden Beratungsbedarf aufweisen. Dadurch werden klassische Unternehmensberatungen ausgeschlossen (vgl. Hebenstreit 2016). Die nachfolgende Abbildung verdeutlicht die Zusammenhänge des Intermediär-Geschäftsmodells.



Digitalisierung und Disruption

Durch die Abbildung wird deutlich, dass der Intermediär zwischen dem Anbieter und dem Kunden steht. Er fungiert quasi als eine Art Vermittler, welcher Anbieter und Kunde zusammenbringt. Beispiel Hotelbuchungsportal: Der Kunde gelangt auf das Portal des Intermediärs und wird dort fündig. Er bucht ein Hotel. Sowohl der Kunde als auch der Anbieter gehen über den Intermediär einen Vertrag ein. Sie stimmen den AGBs und dem Vertrag zu. Der Intermediär nimmt eine Auswertung und Auswahl des Anbieters vor. Der Kunde hat keine Möglichkeit, direkt beim Anbieter zu buchen. Selbst auf der Homepage des anbietenden Hotels wird er, wenn er buchen möchte, wieder zum Buchungsportal geleitet. Hat der Kunde gebucht, liefert der Anbieter an den Kunden. Der Kunde zahlt an den Anbieter. Der Anbieter bezahlt an den Intermediär die vertrag- lich vereinbarte Vermittlungsprovision.

Beim Intermediär-Geschäftsmodell handelt es sich bei digitalen Gründungen um ein beliebtes und aktuell am häuﬁgsten verwendetes Geschäftsmodell. Als typische Bei- spiele lassen sich Lieferdienste, Hotelbuchungsplattformen oder Beratervermittlungen nennen. Das Intermediär-Geschäftsmodell ist mit dem Vorteil verbunden, dass es nicht erforderlich ist, eigene Umsetzungskapazitäten aufzubauen. Dies bedeutet, dass bei- spielsweise die Buchungen für ein Hotel erfolgen, ohne dass das Hotel selbst extra Marketing betreiben muss. Die Hotelbuchungsplattform führt dazu, dass die Gäste von sich aus auf das Hotel aufmerksam werden und Zimmer buchen. Das Hotel müsste ansonsten selbst die eigene Homepage bewerben usw. Entsprechend diesem Modell erfolgt die Nutzung der Kapazitäten von Partnern, welche im Normalfall über keinen Marktzugang in der Breite verfügen. Als Nachteil des Modells lässt sich nennen, dass es für einen Intermediär erforderlich ist, stets den Platz zwischen Kunde und Anbieter zu sichern, z. B. mit kostenintensivem Marketing, mit erstklassigen Nutzererlebnissen auf der Plattform oder mit reibungslosen Abläufen (vgl. Hebenstreit 2016).

Das Verhältnis zwischen Anbieter und Nachfrager wird durch Verträge geregelt. Durch Verträge wird auch sichergestellt, dass der Intermediär an sämtlichen wirtschaftlichen Interaktionen beteiligt ist, die unter den zwei Parteien erfolgen. Auf diese Weise wird

z. B. auch bei CoMatch der exklusive Zugang zu den Kunden und Anbietern gesichert. Bei CoMatch kommen in diesem Kontext insbesondere „Ablöse“-Summen zum Einsatz. Sobald der Kunde und der Anbieter ein direktes wirtschaftliches Verhältnis eingehen, wird diese Summe fällig. Im Intermediär-Geschäftsmodell erfolgt die Abwicklung des Umsatzes durch Provisionen. Beispielsweise werden durch die virtuelle Unternehmens- beratung CoMatch Berater im höheren Management vermittelt. Hierfür erfolgt eine Pro- visionsberechnung von bis zu 15 % des Tagessatzes. Üblicherweise können Hotelsuch- maschinen bis zu 50 % an Provision am jeweiligen Buchungspreis beziehen. Dabei besteht die Hauptaufgabe der Intermediäre in der Sicherstellung des exklusiven Markt- zugangs. Damit lassen sich die wesentlichen Aufgaben zusammenfassen: umfangrei- ches Online-Marketing, optimale, einfache Prozesse und hohe Kundenorientierung (vgl. Hebenstreit 2016).

Intermediäre bieten Vorteile, durch welche Kunden dauerhaft an die Plattform und nicht an den Leistungserbringer gebunden werden.

###### Märkte mit Intransparenz

Intermediär-Unternehmen bzw. Nutzer von Intermediär-Geschäftsmodellen sind in sol- chen Märkten aktiv, die eine hohe Intransparenz aufweisen. Die Kunden können Anbie- ter bzw. Unternehmer/Selbstständige nicht ohne Barrieren identiﬁzieren, deren Leis- tungsqualität lässt sich nicht oder nur eingeschränkt vergleichen, was auch für deren Beurteilung gilt. Sicherheiten, welche meist durch Unternehmen geschaffen werden, wie z. B. Größe, Umsatz, wichtige Referenzen, sind nicht sichtbar oder können nicht ein- deutig zugeordnet werden. Die Unternehmen mit dem Intermediär-Geschäftsmodell ermöglichen einen Zugang zu kleinen Marktteilnehmern. Sie stellen über interne Aus- wahlprozesse die Qualität und Leistungsfähigkeit der Anbieter sicher (vgl. Hebenstreit 2016).

###### Märkte, die ﬁnanzielles Potenzial aufweisen

Das Intermediär-Geschäftsmodell ermöglicht die Nutzbarmachung von wirtschaftlichen Ressourcen, welche zuvor nicht nutzbar waren oder deren Nutzung nur erschwert mög- lich gewesen ist. Durch das Intermediär-Geschäftsmodell lässt sich eine Senkung der Kosten für die Bereitstellungen bisher ungenutzter Ressourcen erwirken. Beispiel: Ein kleines, bislang weitgehend unbekanntes Landhotel arbeitet mit booking.com zusam- men. Dadurch lassen sich die Werbekosten erheblich senken. Die Reichweite und Sicht- barkeit des Unternehmens wird wesentlich verbessert. Das Hotel bekommt mehr Buch- ungen, ist besser ausgelastet und arbeitet wirtschaftlicher.

Häuﬁg verfügen die Anbieter solcher Ressourcen lediglich über eine sehr einge- schränkte Reichweite sowie eine eingeschränkte Sichtbarkeit. Das Intermediär- Geschäftsmodell ermöglicht dem kleinen Leistungsanbieter einen Zugang zu einem größeren Markt (vgl. Hebenstreit 2016).

###### Geschäftsmodell aus zwei Modellen

Die Grundlage des Intermediär-Geschäftsmodells basiert auf einem Konstrukt, in wel- chem zwei Geschäftsmodelle mit unterschiedlichen Kundengruppen verbunden sind. Dabei weisen die zwei Kundengruppen ein gegenseitiges Interesse an den Ressourcen der anderen auf. Eine der Gruppen beabsichtigt den Einkauf einer Leistung oder Lösung am Markt. Die zweite Gruppe möchte am Markt teilnehmen und sichtbar werden. Die beiden Zielgruppen stellen entsprechende Anforderungen hinsichtlich der Exzellenz sowie der Einfachheit von Prozessen des Intermediärs. Beim Intermediär-Geschäftsmo- dell muss eine der zwei Gruppen zuerst an den Intermediär gebunden werden. Dies stellt eine Herausforderung dar. Für die zweite Gruppe muss ein Erfordernis entstehen. Ein Intermediär wird meist versuchen, dass er zuerst die Gruppe mit dem höheren Bedürfnis bindet. In der Regel handelt es sich bei dieser Gruppe um diejenige, welche zuvor noch keinen Marktzugang hatte und auch nicht sichtbar war. Aus diesem Grund sind Intermediär-Unternehmen bestrebt, dass sie zuerst diese Gruppe binden, was bei- spielsweise über die Generierung von Lead-Listen möglich wird (vgl. Hebenstreit 2016).

Digitalisierung und Disruption

###### Sicherung des Marktzugangs und Kundenbindung

Zwar werden in manchen Geschäftsmodellen Aspekte der Neukundengewinnung für wichtiger gehalten als die Kundenbindung, jedoch verhält es sich im Intermediär- Geschäftsmodell so, dass diese Unternehmen wesentlich stärker auf die Kundenbin- dung fokussieren. Der Grund hierfür liegt mitunter darin, dass die Kundenbindung in der Regel günstiger ist als die Neukundenbindung. Durch die Einfachheit sowie die Exzellenz wird verhindert, dass Kunden der zwei Seiten dieses Geschäftsmodells auf Ersatzangebote ausweichen. Dies ist eine wesentliche Grundvoraussetzung für das ein- wandfreie praktische Funktionieren des Intermediär-Geschäftsmodells. Es ist erforder- lich, dass das Einkaufen von Leistungen über die Plattform einfacher ist und auch wert- schöpfender als vergleichsweise bei Anbietern von entsprechenden Ersatzprodukten. Vor dem Hintergrund der Kundengewinnung ist es notwendig, dass das Marketing darauf zielt, das Angebot des Intermediärs vor den Angeboten kleinerer Marktteilneh- mer zu listen. Beispielsweise bucht booking.com Adwords-Anzeigen auf die Suchen nach Hotelnamen, damit der Intermediär vor den Hotels gelistet ist. In diesem Kontext ist das Ziel zu sehen, dass User wieder auf die Plattform des Intermediärs gebracht werden und letztlich dort ihre Buchung vornehmen (vgl. Hebenstreit 2016).

Die digitalen Ökosysteme sind in zahlreichen Branchen relevant, z. B. in der Automobil- industrie (Connected Cars), im Anlagen- und Maschinenbau (Internet of Things/Indust- rie 4.0), im Finanzwesen und Bankensystem usw. Das Intermediär-Geschäftsmodell bzw. Agentenmodell greift auch für den Bereich Banking (vgl. Keuper 2015).

Beim Intermediär-Geschäftsmodell handelt es sich um das gegenwärtig am häuﬁgsten eingesetzte Modell. Unter Nutzung dieses Geschäftsmodells sind Unternehmen wie z. B. Priceline (booking.com, kayak.com) in der Lage, einen zentralen Zugang zum Markt auf- zubauen. Für den Zugang werden hohe Vergütungen bezogen. Priceline erzielt mit sei- nem Angebot bis zu 600.000 Euro Gewinn pro Mitarbeiter. Im Vergleich dazu erzielt bei- spielsweise Hilton 60.000 Euro Gewinn pro Mitarbeiter. In dem Falle, dass es dem Intermediär glückt, die Kunden und Anbieter frühzeitig zu binden, resultieren neue Wertschöpfungen daraus. Durch diese neuen Wertschöpfungen ist es möglich, dass sich komplette Märkte kurzfristig gravierend verändern. Z. B. werden durch CoMatch ein Zugang zu freien Top-Beratern bzw. passgenaue Projektbesetzungen angeboten. Diese werden gezielt für Projekte ausgewählt, womit sich die Honorare von Unternehmensbe- ratungen erübrigen. Neben passgenauen Projektbesetzungen wird den Unternehmen Beratung zu wesentlich günstigeren Konditionen angeboten. Jedoch bestehen extreme Herausforderungen. So ist es erforderlich, zwei Zielgruppen zu gewinnen und wert- schöpfend miteinander zu verbinden. Hierzu wird geraten, den direkten Kontakt zwi- schen den zwei Zielgruppen vertraglich zu unterbinden. Dies ist ein strategischer Aspekt, welcher im Intermediär-Geschäftsmodell stets zu berücksichtigen und zu über- prüfen ist (vgl. Hebenstreit 2016).

### Beispiele disruptiver Geschäftsmodelle

Im Kontext von Smart Services bzw. Data-driven Services werden zunehmend neuartige Geschäftsmodelle für die Industrie, auch für mittelständische Unternehmen, entwickelt.

###### Digitale Disruption und Geschäftsmodelle

Die Entwicklungen der digitalen Disruption lassen sich insbesondere am Beispiel der Geschäftsmodelle im Captive-Finance-Ökosystem verdeutlichen. Unter Captive Finance versteht man beispielsweise, wenn ein Automobilhersteller eine hauseigene Finanzie- rungsgesellschaft oder Bank betreibt, die den Kunden Sonderﬁnanzierungsangebote unterbreitet (Beispiel: Volkswagenbank). Die Captive-Finance-Industrie wurde bis zum Jahr 2016 von Disruptionen weitestgehend verschont. Zwischenzeitlich zeigen sich jedoch einschneidende Veränderungen. Heute üben auch in der Captive-Finance- Industrie Start-ups einen direkten oder indirekten Einﬂuss auf die bestehenden Geschäftsmodelle aus. Dabei lassen sich die entsprechenden Gründungen in drei Kate- gorien untergliedern (vgl. Mende 2016):

* + - Geschäftsmodell – neue direkte Konkurrenten,
    - Konkurrenten in angrenzenden Geschäftsmodellen,
    - intermediäre Plattformen.

Geschäftsmodell – neue direkte Konkurrenten

Im Jahr 2003 begann Tesla damit, die Automobilindustrie mit seiner stringenten Aus- richtung auf E-Mobilität aufzubrechen. Im Jahr 2015 gründete Tesla eine Financial-Ser- vices-Einheit in Deutschland. Damit verfügte das Unternehmen auch über die umfas- senden Potenziale der Absatzﬁnanzierung im Captive-Markt. Tesla kann z. B. einen End- to-End digitalisierten Antragsprozess abbilden. Dieser ist für den überwiegenden Teil der eingesessenen Captives jedoch noch ein entferntes Zukunftsszenario. Durch das Beispiel zeigt sich, dass nicht nur die Platzierung neuer Player im Automobilmarkt mög- lich ist. Vielmehr können dazu auch Captive-Einheiten aufgebaut werden. Diese können für sich typische Start-up-Vorteile beanspruchen, wie beispielsweise schlanke Struktu- ren oder innovative Systeme. Auch aus den eigenen Reihen kann neue direkte Konkur- renz entstehen. Derzeit stehen digitale Einheiten und Innovation Centers im Mittelpunkt der Entwicklungen bei OEMs und Captives. Es zeichnet sich die Entwicklung zu einem neuen „digitalen Direkt-Captive“ analog zum Direktbankenmodell ab (vgl. Mende 2016).

Konkurrenten in angrenzenden Geschäftsmodellen

Auch in angrenzenden Industrien können neue Player auftreten. Diese dringen mit ihren Produkten zunehmend in das traditionelle und neuere Captive-Geschäft ein. Die neuen Player bieten Crowdfunding und Peer-2-Peer-Lending an. Sie beinhalten Mög- lichkeiten, Kredite zu günstigeren Konditionen zu erhalten oder mit erleichterten Kredit- würdigkeitsregeln. Hierzu bestehen am Markt verschiedene erfolgreiche Beispiele wie Lendico, Lending Club oder SoFi. Solche Anbieter beeinﬂussen infolge ihres Plattform- charakters das Finanzierungsgeschäft. Zudem können sie ihren Kunden die Möglichkeit bieten, dass sie ihr Geld mit attraktiven Zinssätzen in transparenten Risikoklassen

Digitalisierung und Disruption

investieren. In diesem Geschäft suchen Player wie Daimler FS, Volkswagen FS und BMW Bank mit Tages- und Festgeldangeboten ihren Platz im Niedrigzinsumfeld. Captives ver- suchen in erster Linie im Bereich der Mobilitätsdienstleistungen zu expandieren, so

z. B. die Initiativen Car2Go und VW FS Rent-a-Car. Dieses Geschäft bietet infolge des gro- ßen Potenzials von Mobility und Internet-of-Things-Trends für innovative Player ent- sprechende Angriffsﬂächen. Die Player sind in der Lage, schnell skalierbare und kun- denzentrierte Geschäftsmodelle aufzubauen, so z. B. der evolutionäre Taxi- und Lieferspezialist Uber, das innovative Park- und Tankerlebnis EvoPark, MobileNow oder der umfassende Mobilitätsplaner von Moovel (vgl. Mende 2016).

Intermediäre Plattformen

Hinsichtlich der Intermediäre lässt sich feststellen, dass sich der Druck von neuen Play- ern im Markt durch Intermediäre bzw. intermediäre Plattformen zunehmend verstärkt. Als Beispiele lassen sich Vermittler und vor allem die stark expandierenden Vergleichs- portale nennen. Die Intermediäre bauen auf der Grundlage zunehmend intelligenter Suchmaschinen für den Endkunden eine Art Transparenz oder auch Scheintransparenz auf. Die Kunden können diesen Service bequem von zu Hause aus oder unterwegs nut- zen. Wie bei den angrenzenden Geschäftsmodellen zeigen sich im Intermediärmarkt sowohl breit aufgestellte Player, z. B. Check24, wie auch spezialisierte Portale. Diese sind mittlerweile in beinahe jedem Captive-Geschäftsfeld tätig. Check24 bietet den Kunden einen Kreditvergleich an. Im Jahr 2018 wird dieser sogar aktiv (selbst über TV- Spots) beworben mit Krediten ab 0,00 % Zinsen (auch wenn es sich bei solchen Null- Prozent-Zins-Angeboten verschiedener Anbieter vorwiegend um „Lockangebote“ han- delt).

###### Kundenkommunikation als Smart Service

Häuﬁg bestehen digitale Smart Services bereits. In diesen Fällen gilt es, sie richtig ein- zusetzen, so beispielsweise in der Kundenkommunikation, für den Support, zum Zweck der Fehlerbehebung usw. (vgl. Steinhaus 2018).

Exkurs

Crowdfunding

Unter Crowdfunding versteht sich eine bestimmte Finanzie- rungsart von Unter- nehmensgründun- gen oder von Projektrealisierun- gen.

Peer-2-Peer-Lending Bei Peer-2-Peer-Len- ding erfolgt die Ver- gabe von Krediten von Privatpersonen direkt an andere Pri- vatpersonen (ohne Vergabe durch Ban- ken).

Durch das Start-up Smoope wurde ein spezieller Messenger für den mobilen Kun- denservice konzipiert. Diese App ist von der Funktionsweise mit WhatsApp ver- gleichbar. Jedoch ermöglicht sie eine sichere Kommunikation mit dem jeweiligen Geschäftspartner. Der Kunde kann die App auf seinem Smartphone installieren und auf diese Weise mit Servicemitarbeitern chatten. Durch den Einsatz eines derarti- gen Messengers ist es möglich, langfristig alle Kundenkommunikationskanäle zu ersetzen. Dies bedeutet, dass dann auch Warteschleifen in Callcentern usw. entfal- len (vgl. Steinhaus 2018).

Der Maschinenbau zeigt eine Vielzahl an Möglichkeiten, wie sich digitale Services aus der automatischen Analyse von Nutzungsdaten ableiten lassen. Dabei entstehen für die Kunden Mehrwertdienste, so z. B. Remote-Wartungen bei technischen Problemen oder

die Optimierung des Betriebs auf der Basis von Sensordaten. In diesem Zusammen- hang lassen sich auch die Ersatzteilbestellung sowie Vorprodukte auf der Basis von ermittelten Daten nennen (vgl. Steinhaus 2018).

Beispiel disruptives Geschäftsmodell (Klöckner & Co)

Beim Stahlhändler Klöckner & Co werden das IoT und Industrie 4.0 überlegt genutzt. Die Kloeckner.i-Vision: „Unsere Vision ist eine digitalisierte Stahl- und Metallindustrie [sic!] in der alle Marktpartner schnell, einfach und efﬁzient mitei- nander interagieren“ (Kloeckner.i GmbH 2018). Das Unternehmen wird eher selbst zum disruptiven Zerstörer, als abzuwarten, bis es das Nachsehen hat, weil andere Unternehmen die Initiative übernehmen und dies tun. Es wurde frühzeitig erkannt, wie der digitale Fortschritt gegebenenfalls das Geschäftsmodell von Klöckner beeinträchtigen kann. Eine professionelle Analyse der Kundenwünsche erwies sich für die Entwicklung neuer bzw. innovativer Tools als hilfreich. Das Unternehmen verkauft Stahl online, was in dieser Branche zuvor kaum vorstellbar oder realisier- bar erschien. Das Unternehmen verfolgte die Vision, dass sich am Ende der Reform- kette eine gemeinsame Stahlhandelsplattform bildet, welche die Daten von Liefer- anten und Kunden enthält (vgl. Steimel 2016).

Zusammenfassung

Smart Data entsteht durch die Verknüpfung einer großen Menge an Daten (Big Data). Auf der Basis von Smart Data kann Wissen generiert werden, das die Grund- lage für Geschäftsmodelle bildet. Big Data wird zu Smart Data veredelt sowie in neuen, individuell kombinierbaren Smart Products und Smart Services monetari- siert. Werden die digitalen Daten genutzt und neu kombiniert, ergeben sich daraus disruptive Geschäftsmodelle. Smart Data ermöglichen es, Kundenwünsche zu anti- zipieren und zu befriedigen, noch bevor der Nutzer selbst von ihnen Kenntnis hat. Es vollzieht sich ein Wechsel von produktorientierten hin zu nutzerzentrierten Geschäftsmodellen. Dafür bedarf es eines Umdenkens. Dieses Umdenken ist insbe- sondere in sämtlichen traditionellen Unternehmen erforderlich. Für Unternehmen, die ihre (internationale) Wettbewerbsfähigkeit erhalten oder verbessern möchten, ist es unerlässlich, die Chancen zu nutzen, welche ihnen das Internet of Things (IoT) bietet. Die Unternehmen sind demnach gefordert, ihre Geschäftsmodelle proaktiv zu überarbeiten. Gegebenenfalls ist ein eigenes disruptives Denken und Handeln notwendig.

Durch Intermediäre werden Plattformen zur Verfügung gestellt, die Anbieter wie auch Nachfrager schnell und bequem zusammenbringen. Gleichzeitig ist es den Intermediären bzw. Agenten möglich, mit der Zeit die Kontrolle über die Kunden- schnittstelle zu bekommen.

Digitalisierung und Disruption

Durch Intermediäre werden Sichtbarkeit, Zugang und Bewertungen im Markt geschaffen. Sie ermöglichen dem Nachfrager dadurch den Zugriff auf zuvor unge- nutzte Ressourcen. Zudem führen Intermediäre dem Anbieter neue Aufträge zu. Die Unternehmen erzielen den Umsatz durch eine Provision für die Aufträge, die sie vermitteln. Beim Intermediär-Geschäftsmodell handelt es sich bei digitalen Gründ- ungen um ein beliebtes und aktuell am häuﬁgsten verwendetes Geschäftsmodell. Als typische Beispiele lassen sich Lieferdienste, Hotelbuchungsplattformen oder Beratervermittlungen nennen. Hinsichtlich der Intermediäre lässt sich feststellen, dass sich der Druck von neuen Playern im Markt durch Intermediäre/Plattformen zunehmend verstärkt. Als Beispiele können Vermittler und vor allem die stark expandierenden Vergleichsportale genannt werden. Die Intermediäre bauen auf der Grundlage zunehmend intelligenter Suchmaschinen für den Endkunden eine Art Transparenz oder auch Scheintransparenz auf. Die Kunden können diesen Service bequem von zu Hause aus oder von unterwegs nutzen.



# Lektion 3

## Potenziale für Smart Services erkennen

#### LERNZIELE

Nach der Bearbeitung dieser Lektion werden Sie …

… das Business Model Canvas kennen.

… Personas kennen und anwenden können.

… Customer Journeys verstehen und nachvollziehen können.

… Domain-driven Design kennen und anwenden können.

DL-D-DLBINGSS01-L03

1. Potenziale für Smart Services erkennen

### Einführung

Im Zusammenhang mit Smart Services ist es wichtig, Potenziale rechtzeitig zu erkennen und für die Gegenwart und Zukunft zu nutzen. Für das Erkennen von Potenzialen erwei- sen sich verschiedene Instrumente als hilfreich. Hierzu zählen unter anderem das Busi- ness Model Canvas, Personas, Customer Journeys sowie Domain-driven Design (DDD).

### Business Model Canvas

###### Business Model Canvas

Die Wertschöpfung eines Unternehmens setzt sich aus unterschiedlichen Komponen- ten zusammen. Diese Komponenten lassen sich mit dem Business Model Canvas analy- sieren, modellieren und dokumentieren.

Business Model Can-

vas Beim Business Model Canvas han- delt es sich um eine Methode zur über- sichtlichen Darstel- lung, Visualisierung und Entwicklung von Geschäftsmodellen.

Beim Business Model Canvas handelt es sich um eine Methode, mit der sich Geschäfts- modelle übersichtlich darstellen bzw. visualisieren und weiterentwickeln lassen. Es ist im Jahr 2004 durch Alexander Osterwalder im Zusammenhang mit seiner Dissertation an der Universität Lausanne konzipiert worden (vgl. Osterwalder 2004). Das Business Model Canvas solle eine „gemeinsame Sprache“ für die „Beschreibung, Visualisierung, Bewertung und Veränderung von Geschäftsmodellen“ vermitteln (vgl. Osterwalder 2004; Osterwalder/Pigneur 2011; Weitekamp 2016).

Durch das Business Model Canvas werden neun zentrale Faktoren des Erfolgs eines Geschäftsmodells skizziert (vgl. Osterwalder 2004; Osterwalder/Pigneur 2011; Weitekamp 2016):

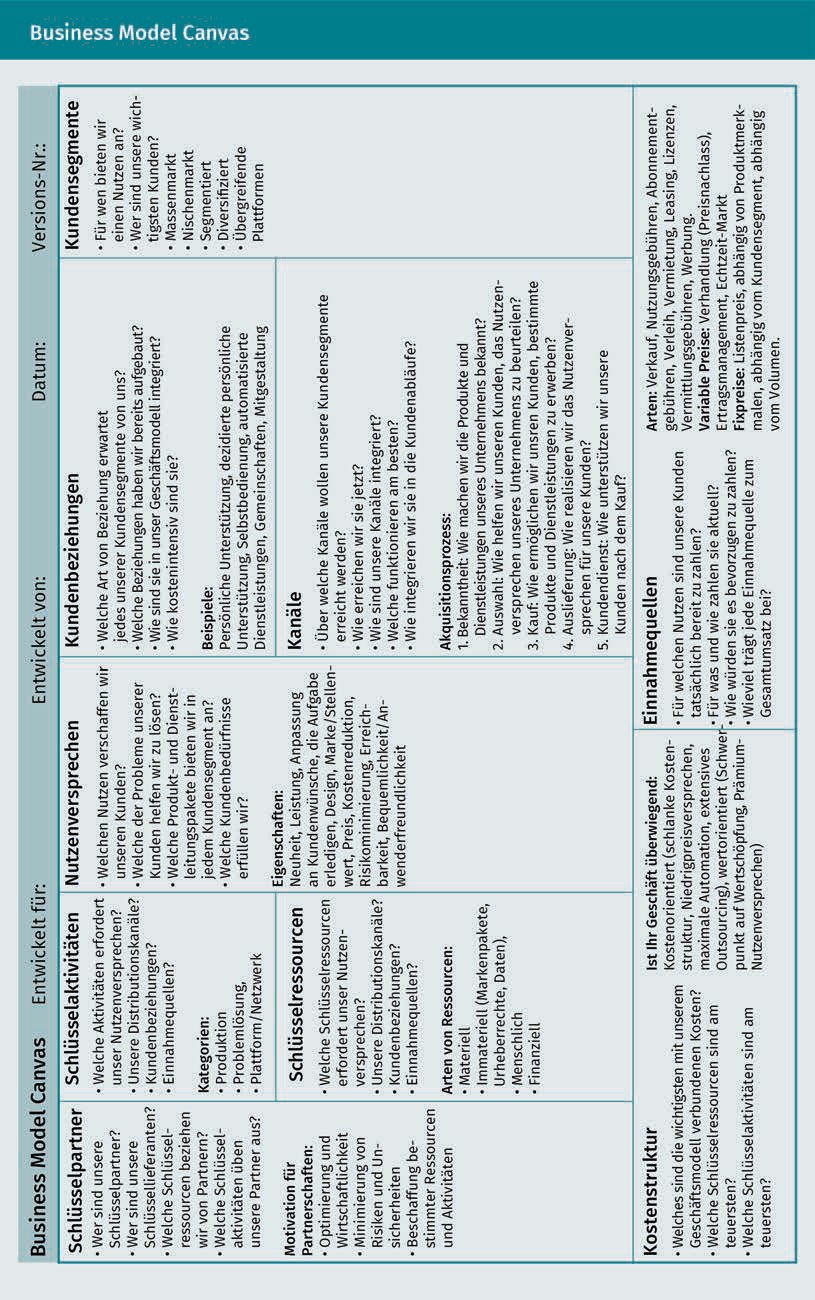
* Kundensegmente: Der Faktor Kundensegmente bezieht sich u. a auf Nutzer, Abon- nenten bzw. auf die zahlenden Kunden. Er beinhaltet somit sämtliche Personen bzw. Organisationen, für welche das betrachtete Start-up Werte zu kreieren beabsichtigt.
* Werteversprechen: Dieser Faktor betrachtet das Werteversprechen, denn für jedes Kundensegment besteht ein eigenes Werteversprechen. Das Werteversprechen bezieht sich auf die Bedürfnisse des jeweiligen Segments in einer abgestimmten Kombination, welche aus Produkt, Service und Dienstleistung besteht.
* Kanäle: Der Faktor Kanäle ist auf die einzelnen Kanäle und Touchpoints bezogen, über welche das betrachtete Start-up mit seinen Kunden kommuniziert bzw. seine versprochenen Werte übermittelt.
* Kundenbeziehungen: Unter dem Faktor Kundenbeziehungen wird beschrieben, wel- che Form des Umgangs das betrachtete Start-up mit seinen Kunden zu pﬂegen beabsichtigt. Ist eine Community geplant? Ist persönliche Betreuung und Beratung vorgesehen? Beabsichtigt das Start-up etwa die Zurverfügungstellung von automati- sierten Dienstleistungen?

Potenziale für Smart Services erkennen

* + Einnahmequellen: Der Faktor Einnahmequelle hinterfragt, mit welchen Preisstrate- gien das betrachtete Start-up Einnahmen erzielt und somit auch neue Werte – für das Unternehmen.
  + Schlüsselressourcen: Der Faktor Schlüsselressourcen bezieht sich darauf, welche Ressourcen und welche Infrastruktur das Start-up benötigt, um sein Produkt oder seinen Service anbieten zu können. Hierbei wird auch hinterfragt, welche Posten absolut unverzichtbar für das Geschäftsmodell sind.
  + Schlüsselaktivitäten: Der Faktor Schlüsselaktivitäten bezieht sich darauf, welche zentralen Aktivitäten erforderlich sind, um das Produkt oder den Service anbieten zu können.
  + Schlüsselpartner: Der Faktor Schlüsselparameter betrachtet, für welche Ressourcen das Start-up auf externe Zulieferer zurückgreifen muss und welche Schlüsselaktivi- täten auszulagern sind. Dabei sind diejenigen Unternehmen und Organisationen, die das Start-up unterstützen, auch dessen Schlüsselpartner.
  + Kostenstruktur: Der Faktor Kostenstruktur richtet den Fokus auf die übergeordnete Finanzplanung für das Start-up.

Die Schlüsselfaktoren eines Geschäftsmodells greifen beim Business Model Canvas ineinander und verdeutlichen die Beziehungszusammenhänge eines Geschäftsmodells. Für die Arbeit mit Business Model Canvas und insbesondere die Visualisierung lässt sich eine Plakatvorlage verwenden. Diese wird unter CC-Lizenz zur Verfügung gestellt. Das Plakat lässt sich ausdrucken und kann dann aufgehängt werden. Auf dem Plakat lassen sich die einzelnen Ideen und Ansätze auf Post-its notieren. Die Ideen und Ansätze können im Planungsverlauf weiterbewegt oder getauscht werden. Ergänzend kann mit Skizzen und verschiedenen Farben gearbeitet werden. Es lassen sich auch mehrere „Leinwände“ für alternative Konzepte erstellen (vgl. Weitekamp 2016; Osterwal- der 2004; Osterwalder/Pigneur 2011).

Zwar stellt das Business Model Canvas keinen Ersatz für den Businessplan dar, jedoch ist es eine hilfreiche Ergänzung. Mit dieser Methode verlieren sich Gründer nicht in lan- gen Ausführungen. Vielmehr erhalten diese einen Überblick über die relevanten Fakto- ren und erkennen, ob sie diese vollständig berücksichtigt haben. Sie können daraus auch erkennen, ob in ihrem Geschäftsmodell gegebenenfalls noch Schwachstellen bestehen. Selbst wenn noch kein eigener Businessplan existiert, ist es für den Gründer doch möglich, mit dem Business Model Canvas die eigenen Ideen zu sortieren, zu selektieren und auf dieser Basis eine Art Grundgerüst für das eigene Geschäftsmodell zu erarbeiten und weiterzuentwickeln. Diese Methode wenden außer Start-up-Gründern auch etablierte Unternehmer an für die Evaluation und Revision ihrer Geschäftsgrund- lagen. Für den Einsatz der Methode ist nicht immer ein wirtschaftlicher Hintergrund erforderlich. Auch im Non-Proﬁt-Sektor ﬁndet die Methode Anwendung, so z. B. an Uni- versitäten oder auch im Zusammenhang mit der Entwicklung von persönlichen Zielen und Projekten (vgl. Osterwalder 2004; Osterwalder/Pigneur 2011; Weitekamp 2016; Falt- ings 2018).



Potenziale für Smart Services erkennen

### Personas

Durch die digitale Transformation wird die Business- und Arbeitswelt rasch und weitrei- chend verändert. Dies betrifft auch die kundenbezogene Kommunikation. Jedoch erfol- gen die wesentlichen kommunikativen Erfolge jenseits von Big Data oder Algorithmen. Im digitalen Zeitalter führen nicht – wie häuﬁg unterstellt – Analytics und Mathematik zum Ziel, sondern vielmehr Menschenkenntnis sowie Einfühlungsvermögen. Die multi- sensorische Kommunikation bietet Chancen, welche weit über Worte, Bilder und Geschichten hinausgehen. Touchpoints sind die kommunikativen Berührungspunkte zwischen Anbieter und Kunde. Diese lassen sich analysieren und optimieren. Für die Erzielung von Erfolgen in der digitalen Gegenwart und Zukunft bedarf es entsprechen- der kommunikativer Wege (vgl. Schüller 2016, S. 11ff.).

Bei Personas handelt es sich um ﬁktive Stellvertreter einer Kundengruppe. Diese verei- nen deren charakteristische Eigenschaften, Erwartungshaltungen sowie Vorgehenswei- sen. In diesem Rahmen wird das anonyme Zielgruppengemenge ersetzt durch eine menschliche Gestalt. Es ist möglich, sich in diese menschliche Gestalt hineinzudenken bzw. hineinzuversetzen (vgl. Schüller 2017).

Empathie ist eine wesentliche Voraussetzung für die Kreativität. Personas können hierzu nützlich sein. Durch Personas wird es beispielsweise möglich, dass Mitarbeiter, die in indirektem Kundenkontakt stehen, den Menschen sehen und verstehen, der hin- ter der Bestellnummer bzw. einer Kundennummer oder einem Aktenzeichen steht. Im Kontext der Digitalisierung, mit der Verarbeitung von Algorithmen und Daten, können mithilfe von Personas Datenpakete mehr oder weniger lebendig werden (vgl. Schüller 2017).

So ist es z. B. hilfreich, wenn ein Mailing getextet wird, sich vorzustellen, dass die ent- sprechende Zielperson leibhaftig vor einem steht. Ist die betreffende Person nicht per- sönlich bekannt, kann es hilfreich sein, dem Empfänger einen Namen und ein Gesicht zu geben. Eine Persona baut auf sechs zentralen Elementen auf (vgl. Schüller 2017; Schüller 2016):

* + - Element 1 – Name und Foto: Das erste Element richtet den Fokus auf den Namen und das Foto. Die Ursprungsfragestellung besteht darin, wie ein typischer Vertreter aus der betrachteten Ziel- bzw. Kundengruppe aussieht und wie diese Person heißt. Für die Fotoauswahl kann auch eine Zeichnung verwendet werden. Durch Fotos von realen Menschen, welche beispielsweise von Bilderbanken bezogen werden, wird eine Persona meist zu speziﬁsch bestimmt.
    - Element 2 – Hintergrundinformationen: Das Element Hintergrundinformationen bezieht sich auf Alter, Geschlecht, Wohnort, Beruf, familiäre Verhältnisse, Einkom- menssituation, Hobbys sowie weitere Interessen.
    - Element 3 – Statements: Im Kontext des dritten Elements können beispielsweise wörtliche Aussagen, die für den entsprechenden Kundentyp als typisch gelten, zitiert werden. Es lassen sich geeignete Schlagworte auﬂisten, die dessen Werte, Stand- punkte, Ansichten sowie Einstellungen repräsentieren. Dem Kundentyp lassen sich typische Marken zuordnen.

Touchpoints

Dies sind kommuni- kative Berührungs- punkte zwischen Anbieter und Kunde.

Personas

Fiktive Stellvertreter einer Kundengruppe nennt man Personas. Sie vereinen charak- teristische Eigen- schaften, Erwar- tungshaltungen und Vorgehensweisen.

* + - * Element 4 – Erwartungen/Ziele: Dieses Element richtet sich auf die Frage, was die entsprechende Persona mit dem Kauf bzw. der Inanspruchnahme eines Produktes bzw. einer Dienstleistung zu erreichen beabsichtigt. Welche Probleme beabsichtigt sie gegebenenfalls zu lösen? Welchen Nutzen möchte sie dadurch erzielen? Welche Gefühle könnten damit verbunden sein? Welche Ängste sind gegebenenfalls damit für sie verbunden? Was könnte diese Person begeistern?
      * Element 5 – Kaufprozess: Das Element Kaufprozess stützt sich auf die Fragestellung, wie die entsprechende Persona einkauft, welche Customer Journey diese geht, auf welche Art und Weise und über welche Wege bzw. Kanäle sie sich informiert, wer auf sie Einﬂuss ausübt, welchen Stellenwert Ofﬂine und Online für diese Person haben und welche für sie die bedeutendsten Touchpoints sind.
      * Element 6 – ideale Lösung: Dieses Element richtet den Fokus auf die Frage, wie die ideale Produkt- oder Servicelösung aus der Perspektive einer solchen Persona aus- sehen kann oder wird.

### Customer Journeys

Customer Journey Als Customer Jour- ney bezeichnet man die Kundenreise, sozusagen die ein- zelnen Schritte des Kunden, in welchen er auf das Unterneh- men zugeht oder wie

er entlang der Touchpoints geht bzw. mit dem Unter- nehmen zusammen- trifft oder interagiert.

Damit Kunden zufriedengestellt werden, reicht ein gut durchdachtes Produkt oder eine Dienstleistung alleine nicht aus. Faktisch kann jeder Kontakt mit einem Kunden oder Interessenten über die weitere Beziehung zu einem Unternehmen entscheiden. Es kommt nicht nur auf den einzelnen Kontaktpunkt bzw. Touchpoint an, sondern in zunehmendem Maße auch auf die Kundenreise bzw. Customer Journey. Daraus ergeben sich drei grundlegende Fragestellungen bei der Betrachtung der Customer Journey:

* Welche Schritte hat der Kunde beispielsweise vor einem Anruf im Callcenter durch- laufen?
* Wie lange hat er auf der Internetseite gesurft, um entsprechende Informationen zu einem Produkt zu ﬁnden?
* Was ist der kritische Faktor (im Servicebereich) und wie sieht dieser aus?

In den zurückliegenden Jahren standen eher die Prozessefﬁzienz, Durchlaufzeiten und Kosten im Vordergrund. Heute richtet sich der Fokus auf die Kundenerlebnisse bzw. Customer Experience, hier vor allem auf ihre Analyse und damit ihrem Verständnis. Bedeutende Ansatzpunkte für die Kundenzufriedenheit können diesbezüglich aus den folgenden Fragestellungen gewonnen werden (vgl. Henn 2017):

* An welchen Punkten bzw. wo erlebt der Kunde Frust?
* Wo und in welcher Situation ﬁndet sich der Kunde nicht zurecht?
* Wo sind bzw. welche Aussagen und Erläuterungen sind missverständlich oder inkon- sistent?
* Wo ist der Zeitaufwand bzw. welcher Zeitaufwand ist für den Kunden unverhältnis- mäßig hoch?

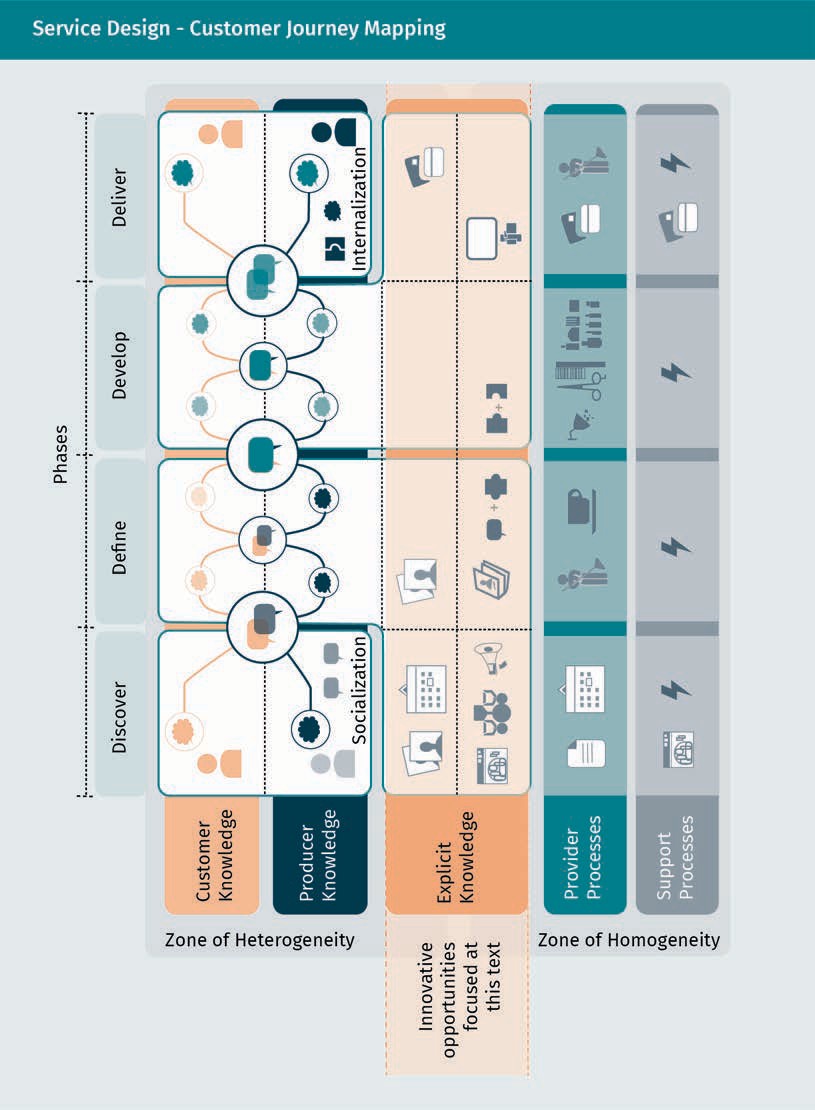
Potenziale für Smart Services erkennen

###### Customer-Journey-Mapping-Methode

Unter der Customer-Journey-Mapping-Methode versteht man das graﬁsche Abbilden der Kundenreise entlang sämtlicher Touchpoints mit allen Lust- und Frusterlebnissen. Diese Methode gilt sowohl als eine hilfreiche Ergänzung als auch Erweiterung von Methoden wie beispielsweise dem Service Blueprinting. Beim Service Blueprinting liegt der Fokus auf dem Prozessgedanken. Dagegen wurden die Kundenbedürfnisse zuvor nicht hinreichend berücksichtigt. Diese zwei Methoden beﬁnden sich noch in einer Phase, in der sie weiterentwickelt werden können, so z. B. durch die Berücksichtigung des Vorwissens des Kunden oder des Wissens des Anbieters sowie erforderlicher Sys- teme für eine Unterstützung der Prozessschritte. Für das Verständnis der Kundenreise bzw. Customer Journey und eine gute Customer Experience sind erforderlich (vgl. Henn 2017; Pninheiro/Manhães 2013):

* + Kenntnis über das Vorwissen des Kunden,
  + Wissen über den Kontext, in welchem die Kunden agieren,
  + Kenntnis über die Touchpoints, welche schon durchlaufen wurden,
  + Kenntnis des Wissens und der Systeme, welche zur Unterstützung am einzelnen Touchpoint erforderlich sind,
  + Kenntnis der Erwartungen und Bedürfnisse, welche persona-speziﬁsch zu erfüllen sind.

Die folgende Abbildung zum Service Design verdeutlicht das Customer Journey Mapping exemplarisch.



Es geht darum, das Vorwissen der Kunden sowie das Anbieterwissen in Serviceprozesse zu integrieren. Für den Fall, dass sich die Kunden mit einem neuen Produkt einer Ser- vice-Innovation nicht auskennen, gilt es, entlang der Customer Journey andere Wege einzuschlagen. Dadurch sollte es möglich sein, die entsprechenden Bedürfnisse zu erfüllen. Die Produkte und Services sind eng verbunden mit den Kundenerfahrungen. Dies gilt sowohl im Kontext der Produktnutzung, bei einem Servicevorfall wie auch bei

Potenziale für Smart Services erkennen

einer Anfrage. Aus der Perspektive des Kunden ergibt sich nur bei einer gleichzeitigen Optimierung von Produkten, Services und Customer-Service-Prozessen ein positives Gesamtergebnis. Hierfür können die folgenden kritischen Faktoren genannt werden: Einfachheit, dem Kunden Zeit zu ersparen, die Wirksamkeit der Prozesse sowie Erledi- gungs- oder Fallabschlussquote. Vor dem Hintergrund der Digitalisierung ist ein weite- rer Faktor relevant. Es handelt sich um die intelligente Vernetzung von Services und Prozessen. Heute zeichnen sich neue Serviceangebote, Plattformen und Systeme vor allem dadurch aus, „dass sie ein singuläres Problem; allenfalls ein kleineres Problem- bündel lösen“ (Henn 2017; vgl. auch Pninheiro/Manhães 2013).

### Domain-driven Design

Der Begriff Domain-driven Design (DDD) wurde durch Eric Evans geprägt (Evans 2003). Es handelt sich um den Begriff für eine von Anwendungsdomänen getriebene Herange- hensweise an das Design von komplexer, projektorientierter Software (vgl. Lobacher 2010, S. 5).

Domain-driven Design (DDD) gilt als weit mehr als eine Sammlung von Entwurfsmus- tern. Es handelt sich dabei vielmehr um eine Art Entwicklungsphilosophie. Es ist eine gewisse Zeit erforderlich, bis eine Loslösung von bisherigen Sicht- und Denkweisen erfolgt und sich diese neue Sichtweise entfaltet (vgl. Priebsch 2016).

Es besteht ein unterschiedliches Verständnis von domänenspeziﬁschen Konzepten als ein bedeutender Grund für die divergierende Vorstellung der zwei Gruppen (vgl. Loba- cher 2010, S. 10):

* + - Benutzer/User bzw. Kunde,
    - Anwendungsentwickler bzw. Dienstleister.

Die zwei grundlegenden Annahmen des Domain-driven Designs sind (vgl. Lobacher 2010, S. 6), …

* + - … dass der Schwerpunkt des Software-Designs auf Fachlichkeit und Fachlogik basiert.
    - … dass der Entwurf von fachlichen Zusammenhängen auf einem Fachmodell basie- ren sollte.

Es sollen implizite Zusammenhänge explizit gemacht werden.

Anstatt im ersten Schritt das MVC-Framework auszuwählen, ist es sinnvoller, die Kon- zentration auf die Fachlichkeit zu richten und weniger auf technische Aspekte (vgl. Priebsch 2016). Es wird empfohlen, nicht zu früh auf technische Details zu fokussieren. Es gilt als essenziell, mit dem Kunden in dessen Fachsprache zu sprechen. Der Kunde sollte keinesfalls mit technischen Begriffen verwirrt werden. Der Grundgedanke, dass

Domain-driven Design (DDD) DDD dient dem

Design von komple- xer, projektorientier- ter Software. Es han- delt sich um Entwurfsmuster und eine Art Entwick- lungsphilosophie.

MVC-Framework Das MVC-Modell (Modell – View – Controllers) ist ein Architekturmuster für die Unterteilung einer Software in die

Komponenten Datenmodell, Prä- sentation und Pro- grammsteuerung.

Domäne Eine Domäne ist ein abgegrenztes Prob- lemfeld, Fachgebiet, Geschäftsfeld oder ein Einsatzbereich. Plattformen führen zur Entstehung und Entwicklung von vollautomatisierten

Marktplätzen.

die Fachlichkeit in den Mittelpunkt der Softwareentwicklung gerückt werden soll (Domain-driven Design), gilt in erster Linie für die Entwicklung von komplexen Syste- men (vgl. Priebsch 2016).

###### Domäne und Modell (Domain Model)

Bei einer Domäne handelt es sich um ein abgegrenztes Problemfeld, Fachgebiet, Geschäftsfeld oder einen Einsatzbereich. Es gilt, ein einheitliches Verständnis zwischen den folgenden Gruppen zu schaffen:

* Domänenexperte und
* Applikationsentwickler.

Bei einem Modell handelt es sich um eine auf bestimmte Zwecke ausgerichtete verein- fachte Beschreibung der Wirklichkeit (vgl. Lobacher 2010, S. 12ff.).

Die Modellierung ist ein bedeutender Prozess, an welchem Domänenexperte (Kunde) und Dienstleister (PM bzw. Programmierer) beteiligt sind. Es handelt sich um einen ite- rativen (= sich mehrfach wiederholenden), agilen Prozess. Als Ergebnis entsteht ein Modell (vgl. Lobacher 2010, S. 15).

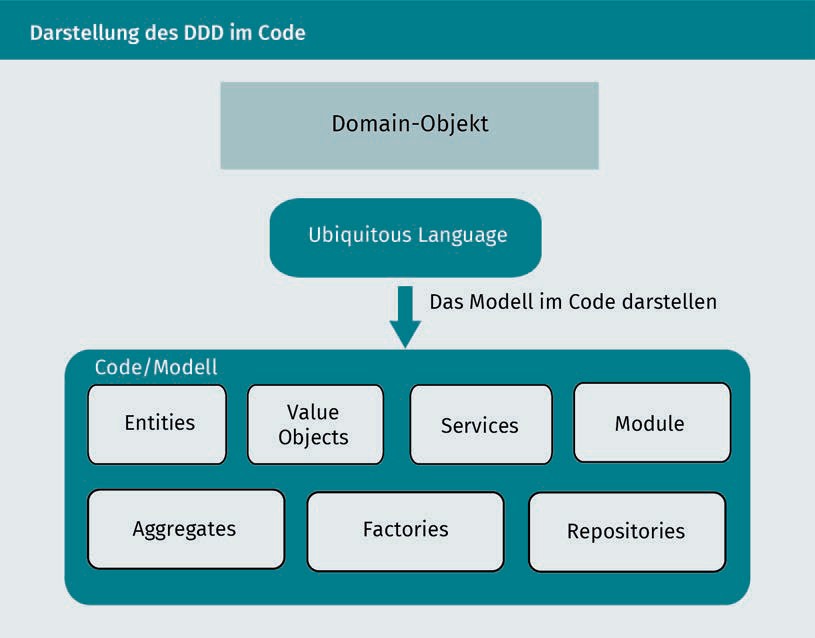
Kernelement des Modellierens

Das Kernelement des Modellierens ist eine Ubiquitous Language (UL) bzw. eine gemein- same oder allgegenwärtige Sprache. Es ist erforderlich, dass diese Sprache von allen gemeinsam gesprochen wird, d. h. insbesondere von sämtlichen Teammitgliedern, Füh- rungspersonen, Kunden usw. Sie bildet die Basis für sämtliche Aktivitäten und Erfolge im Projekt. Zugleich ist sie auch der Namensraum für alle Artefakte im Modell. Diese gemeinsame Sprache ist von hoher Bedeutung für die erfolgreiche Etablierung eines einheitlichen Domänenmodells. Das unter anderem mit der gemeinsamen Sprache gewonnene Modell bedarf der Implementierung. Zwischen Ubiquitous Language, Modell und Implementierung besteht ein interdependenter Drei-Faktoren-Beziehungs- kontext, d. h., eine Veränderung innerhalb eines Faktors bewirkt zugleich auch Verände- rungen in allen anderen Faktoren oder geht mit diesen einher. Die gemeinsame Spra- che ist grundsätzlich multilingual möglich, jedoch wird im Allgemeinen die englische Sprache bevorzugt. Ubiquitous Language wird auch als Consulting-Prozess aufgefasst (vgl. Lobacher 2010, S. 16f.).

Ubiquitous-Language-(UL-)Modellierung

Im Rahmen der Modellierungssprache entsteht in der Regel ein Diagramm (UML – Ubi- quitous Modelling Language). Dieses verpackt die Eigenschaften, Funktionalitäten sowie die Beziehungen der relevanten Bestandteile des Problemfeldes in Objekte und stellt deren Relationen dar. Das Diagramm beinhaltet meist Notizen bzw. Stichworte zu Post, Blog, Tags und Content (vgl. Lobacher 2010, S. 19).

Potenziale für Smart Services erkennen



Domain-Objekt

Evans trifft zwischen den Modellelementen eine wichtige Unterscheidung. Dabei han- delt es sich darum, ob ein Objekt eine Identität hat (vgl. Evans 2003). Es ist erforderlich, dass Objekte identiﬁzierbar sind. Dadurch wird es möglich, sie voneinander zu unter- scheiden. Sie können später auch wiedergefunden werden, damit sie für nachfolgende Operationen wiederverwendet werden können, z. B. Personen, Events, Konten usw. Es bestehen weitere Objekte, die lediglich die Repräsentation einer Eigenschaft darstellen,

z. B. Farben, Tags usw. Sie deﬁnieren sich durch alle Eigenschaften (vgl. Lobacher 2010, S. 22).

Entity

Zu Beginn richtet sich die Betrachtung auf das wichtigste Muster bzw. die Entität (Ent- ity), ein (persistentes) Objekt mit einer Identität. Dies können z. B. Personen bzw. Kun- den mit eindeutiger Identität sein (vgl. Priebsch 2016).

Zunächst wird für jede Entität ein eigenes ID-Objekt erstellt. Im folgenden Beispiel basiert die entsprechende PersonId auf einem Universally Unique Identiﬁer (UUID). (Es wird von der Existenz eines entsprechenden Objektes ausgegangen.) Die Klasse der Per- son und auch andere Codes sind von UUID abhängig. UUID stellt ein Implementie- rungsdetail von PersonId dar. Es ist möglich, in einer alternativen Implementierung von PersonId als Identiﬁkator beispielsweise einen Auto-Increment-Wert der Datenbank zu verwenden (Listing 1) (vgl. Priebsch 2016).

Universally Unique Identiﬁer (UUID) Bei dem Universally Unique Identiﬁer (UUID) handelt es

sich um einen Stan- dard für Identiﬁkato- ren der Softwareent- wicklung. Dessen Zweck besteht darin, Informationen in verteilten Systemen ohne eine zentrale Koordination ein- deutig zu kennzeich-

nen.



Die Person hat weiteren Attribute und Verhalten. Der Fokus des Domain-driven Designs richtet sich auf das Verhalten, anstatt auf die Daten. Vor diesem Hintergrund tragen Methoden mit Namen keine Getter oder Setter. Sie beschreiben das Verhalten des Objekts aus geschäftlicher Perspektive (vgl. Priebsch 2016).

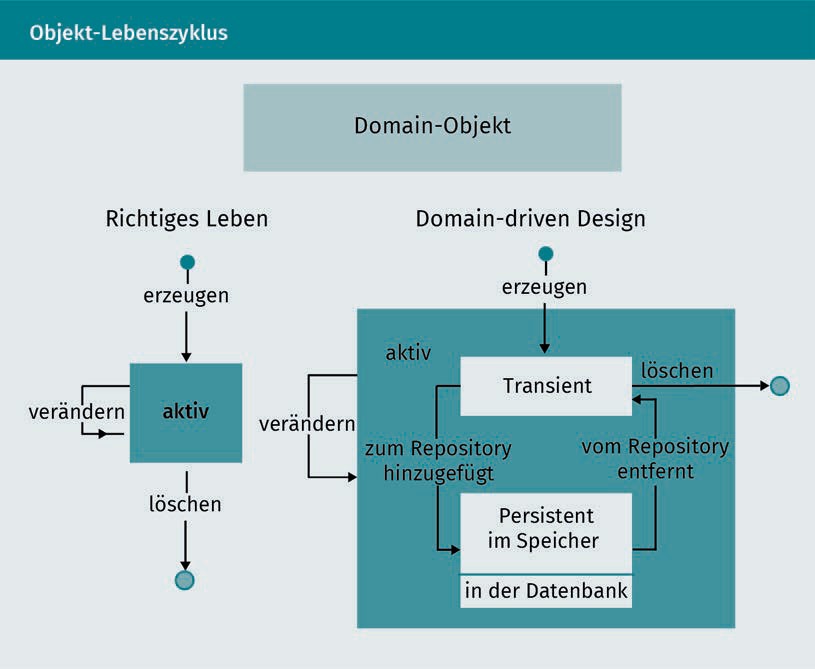
Im Kontext von Entitäten geht es nicht um die Erstellung eines allumfassenden Modells. Eine Person kann beispielsweise Angestellter, Kunde und/oder Geschäftspartner sein. In einem komplexen System kann die gleiche Person auch mehrere Rollen einnehmen. Im Domain-driven Design wird deutlich, dass diese unterschiedlichen Aspekte nicht in einer einzigen Klasse abgebildet werden sollen. Vielmehr sind mehrere Modelle zu schaffen. Auf diese Weise entsteht ein geringerer Grad an Komplexität. Durch die Per- sistenzinfrastruktur der Anwendung ist für Entitäten zu gewähren, dass im Speicher für jede ID jeweils nur eine Instanz existiert. Dadurch sollen zwei oder mehrere gegebe- nenfalls unterschiedliche Zustände von einer identischen Person im System und deren unerwünschte Folgewirkungen vermieden werden (vgl. Priebsch 2016).

Potenziale für Smart Services erkennen

Bei Value Objects handelt es sich um nicht identiﬁzierbare Objekte. Sie weisen keine eigene Identität auf, z. B. die Adresse eines Kunden/Lieferadresse. Dagegen sind Ent- ities identiﬁzierbare Objekte. Sie weisen eine eigene Identität auf, z. B. der Kunde selbst. Bei Services handelt es sich um nicht an das Objekt gebundene Funktionen oder das Handling von mehreren Objekten, z. B. eine Überweisung zwischen zwei Kon- ten (vgl. Lobacher 2010, S. 23f.).

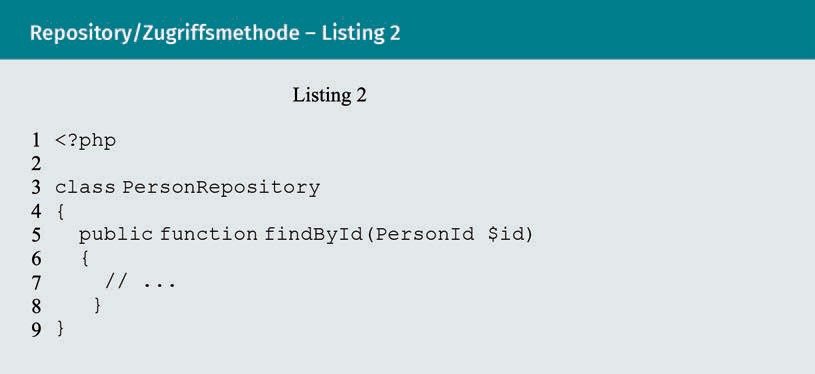
Objekt-Lebenszyklus

Die nachfolgende Abbildung stellt den Objekt-Lebenszyklus des richtigen Lebens dem des Domain-driven Designs gegenüber.

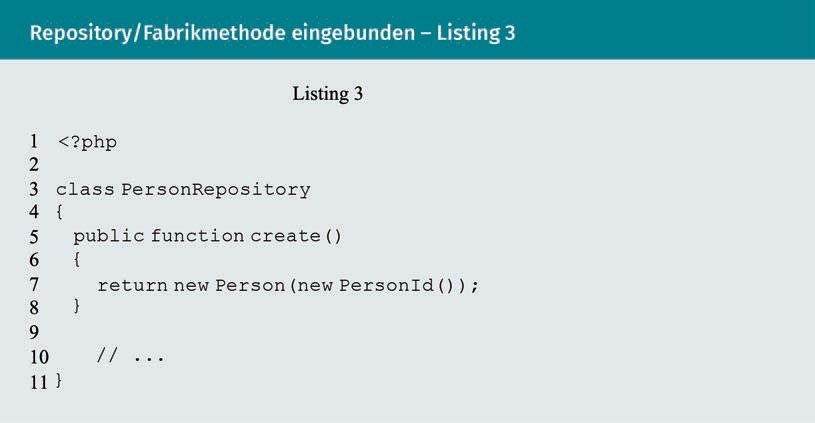


Repositories

Technische Details (der Persistenz) sollen nicht in die UL eindringen, wofür Reposito- ries erschaffen werden. Eine Repository ﬁndet Anwendung für das Laden einer Entität. Bei Repositories handelt es sich im DDD nicht um einen Bestandteil der Persistenzinf- rastruktur, sondern um einen Teil der Domäne. Ein Repository wird am besten als Inter- face deﬁniert. Es erleichtert beim Testen das Mocken. (Zur Vereinfachung des Listings wird nachfolgend kein Interface berücksichtigt.) Ein Repository sollte möglichst wenig Zugriffsmethoden haben. Nachfolgend wird das Beispiel einer Zugriffsmethode ver- deutlicht (Listing 2) (vgl. Priebsch 2016).



Die Fabrikmethode(n) zum Erzeugen einer Entität lässt (lassen) sich auch direkt in das Repository packen (Listing 3). In diesem Falle wird keine gesonderte add()-Methode für das Hinzufügen einer Entität zum Repository benötigt (vgl. Priebsch 2016).



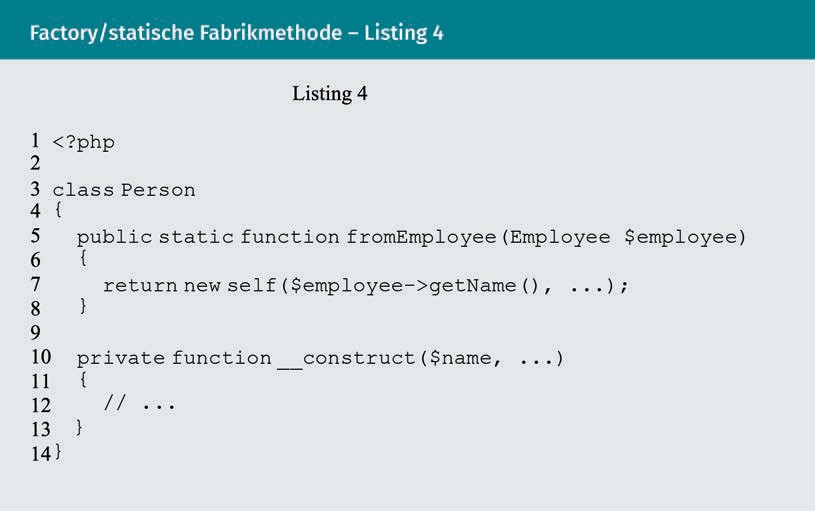
Eine zentrale Funktion des Repositories besteht im Identitätsmanagement von Entitä- ten. Ein Repository ist eine Art In-Memory-Cache für Objekte. Wenn ein Objekt abgeru- fen wird, das sich noch nicht im Speicher beﬁndet, wird es geladen. Beﬁnden sich dagegen angefragte Objekte bereits im Speicher, wird eine Referenz darauf zurückgege- ben. Damit werden mehrere Kopien einer gleichen bzw. identischen Person im Speicher vermieden. Eine weitere zentrale Funktion des Repositories besteht in der Speicherung von Änderungen, beispielsweise mittels einer Methode commit (vgl. Priebsch 2016).

Potenziale für Smart Services erkennen

Factory

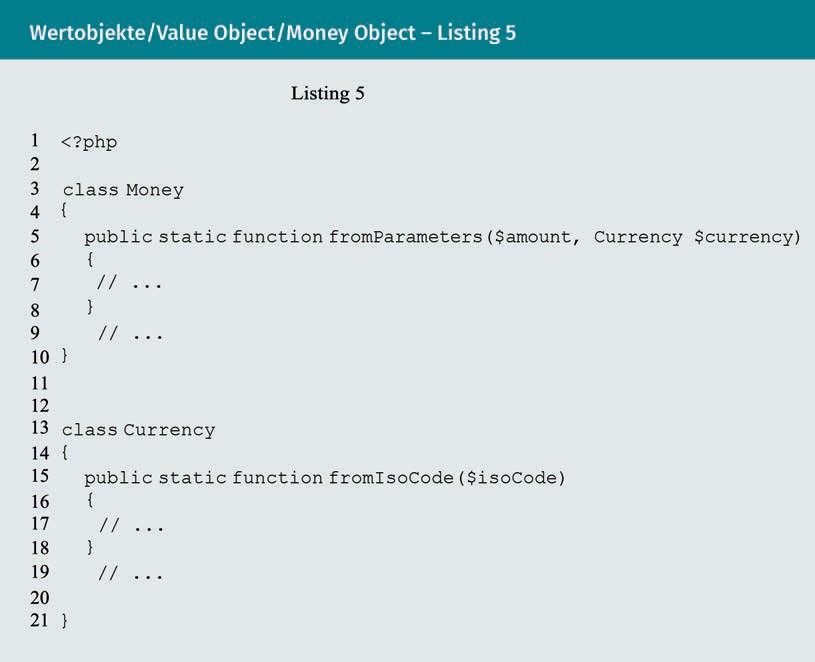
Im DDD zählen auch Factories bzw. Fabriken zur Domäne. Durch Factories werden Objekte erzeugt. Zur Entkopplung des aufrufenden Codes vom konkreten Klassennamen ist die Verwendung einer abstrakten Fabrik möglich. Wenn das erzeugende Objekt Abhängigkeiten aufweist bzw. die Erzeugung komplex ist, kann ein Fabrikobjekt bzw.

„Dependency Injection Container“ oder Builder-Entwurfsmuster Anwendung ﬁnden. Durch Fabrikmethoden lassen sich Objekte entkoppeln. Das folgende Beispiel (Lis- ting 4) zeigt eine statische Fabrikmethode auf (vgl. Priebsch 2016).



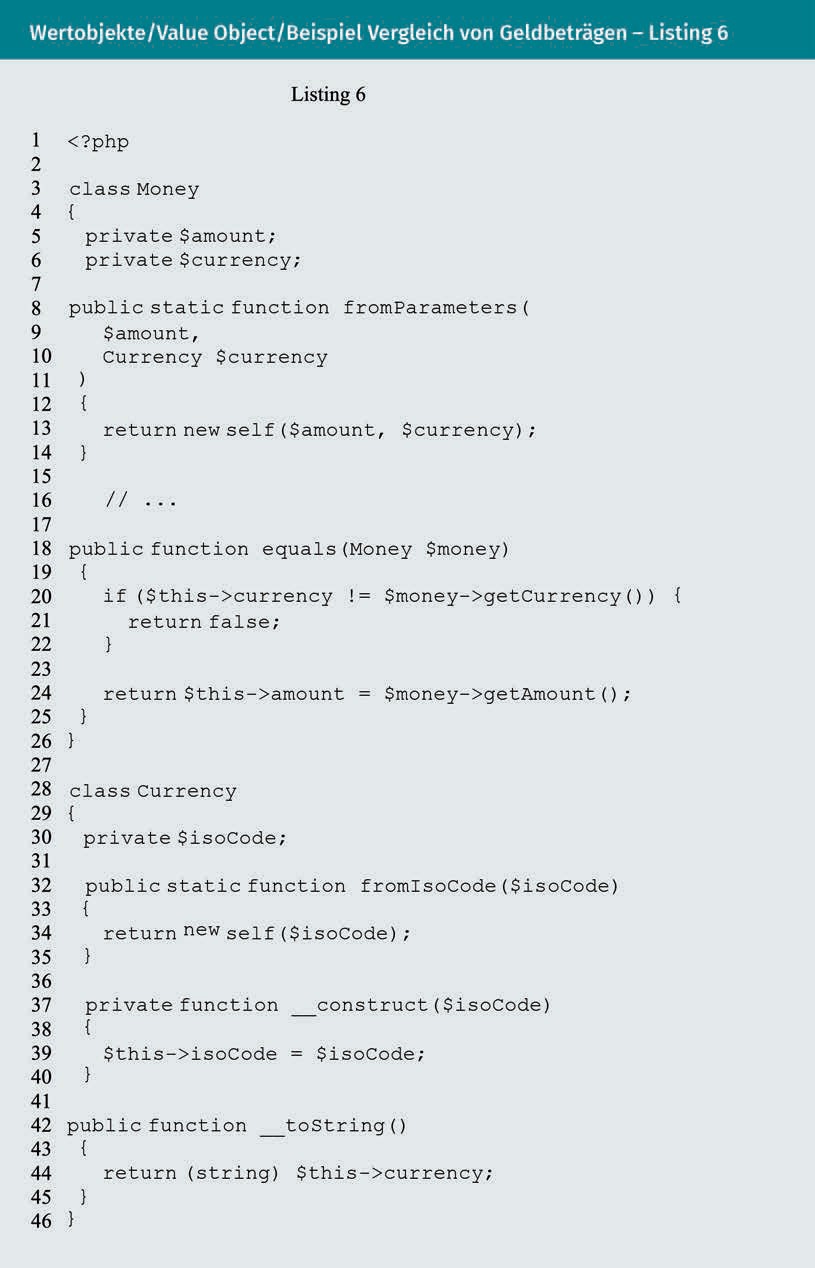
Wertobjekte/Value Object

Bedeutende Muster bzw. Konzepte des Domain-driven Designs sind Wertobjekte (Value Objects), so z. B. „Money-Objekt“. Als Integer- oder Fließkommazahl werden nicht Geld- mittel verwendet, sondern in Geschäftsanwendungen ﬁndet ein Money-Objekt Verwen- dung. Dies erleichtert die Umrechnung verschiedener Landeswährungen. Dabei kapselt ein Money-Objekt die Informationen beispielsweise zweier unterschiedlicher Landes- währungen in einem Objekt. Dies wird im Beispiel (Listing 5) im Einsatz von Fabrikme- thoden verdeutlicht (vgl. Priebsch 2016).



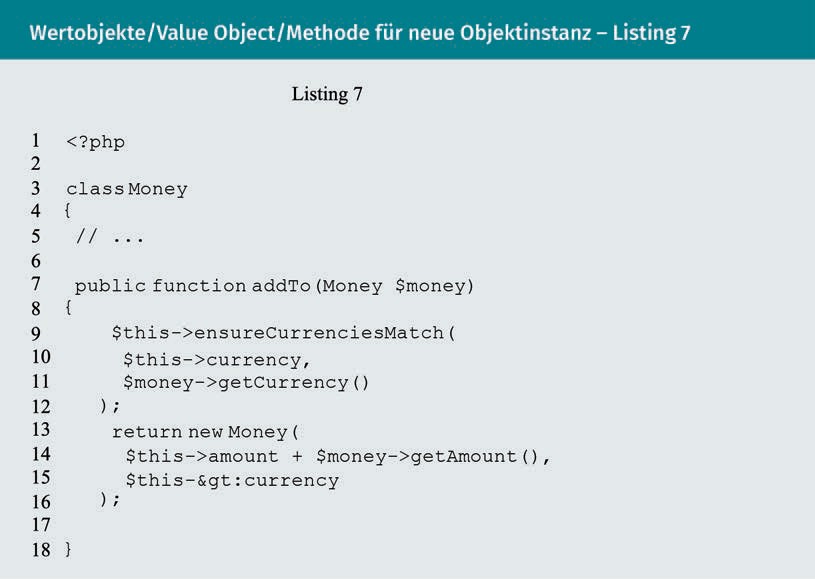
Anhand des folgenden Beispiels (Listing 6) lassen sich Geldbeträge vergleichen.

Potenziale für Smart Services erkennen



Wertobjekte werden im Falle mehrerer zusammengehöriger Informationen verwendet. Es lassen sich nicht nur Geldbeträge vergleichen, sondern auch weitere Einheiten wie Gewicht, Maße u. Ä. Wertobjekte werden auch für Einzelinformationen verwendet, bei- spielsweise zu Plausibilitätsprüfungen, Zeiterfassung usw. Dadurch lassen sich Code- Duplikation und Fehler vermeiden (vgl. Priebsch 2016).

Es ist erforderlich, dass Wertobjekte unveränderlich (immutable) sind. Deren Zustand darf sich nach der Erzeugung auf keinen Fall verändern. Geeignete Methoden geben in diesem Zusammenhang eine neue Objektinstanz zurück, was durch das Beispiel (Listing 7) verdeutlicht wird (vgl. Priebsch 2016).



Im Unterschied zu Entitäten verfügen Wertobjekte über keine Identität. Es lassen sich beliebig viele Instanzen erzeugen und Attribute vergleichen. Hierzu wird in der Regel eine Methode wie equals() oder isEqualTo() verwendet, die die Details des Vergleichens implementiert. Die Entscheidung, ob in einer bestimmten Situation ein Wertobjekt oder eine Entität verwendet wird, hängt nicht nur von technischen Faktoren ab. Vielmehr ist die Fachlichkeit bestimmend (vgl. Priebsch 2016).

Aggregate

Bei einem Aggregat handelt es sich um eine Objektstruktur, welche aus einer Entität und weiteren Objekten besteht. Bei einer Aggregate Root (= Wurzel) handelt es sich um eine Fassade, die für den Benutzer eine API für das gesamte Aggregat bietet. Es ist nicht erforderlich, dass Objekte innerhalb des Aggregats über eine global eindeutige ID ver- fügen. Es ist eine lokal eindeutige ID ausreichend. Objekte innerhalb eines Aggregats werden lediglich durch die Aggregate Root modiﬁziert. Zugleich stellt ein Aggregat die

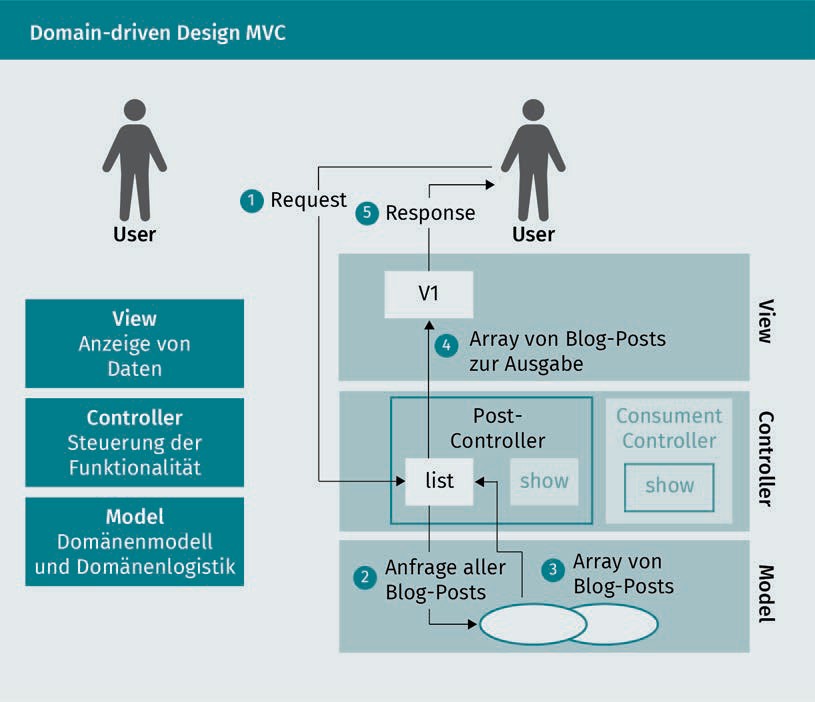
Potenziale für Smart Services erkennen

kleinste Einheit dar, die von der Persistenz geladen wird. Umfang und Struktur eines Aggregats sind abhängig von der Fachlichkeit. Die Aufgabe eines Aggregats besteht in der Sicherung der Konsistenz. Zudem sollte es zumindest einen geschäftlich relevanten Vorgang eigenverantwortlich durchführen können. Im Aggregat sind sämtliche Daten enthalten, die erforderlich sind, um für einen Nutzungsfall die Einhaltung sämtlicher relevanten Geschäftsregeln zu sichern. Durch das Beispiel (Listing 8) wird ein Aggregat- objekt für eine Person verdeutlicht. Annahme im Beispiel: Es ist Aufgabe des Aggrega- tobjekts, sicherzustellen, dass eine Person lediglich eine begrenzte Anzahl von Bank- konten eröffnet bzw. eröffnen kann (vgl. Priebsch 2016).



Ein Aggregat soll im Domain-driven Design die Einhaltung von Geschäftsregeln sicher- stellen. Damit soll auch die Konsistenz gewahrt bleiben. Eine etwaige (Daten-)Redun- danz ist in diesem Zusammenhang unproblematisch. Die Kommunikation zwischen den Beteiligten erfolgt in der Regel durch Messaging (vgl. Priebsch 2016).

Auch Dienste (Services) gelten als Entwurfsmuster, die im Domain-driven Design bedeutend sind. Durch einen Service wird Funktionalität gekapselt, die nicht unbedingt einer Entity bzw. einem Aggregat zuzuordnen ist. Unter Services fallen weniger techni- sche Services, beispielsweise E-Mail-Versand, PDF-Dokumenten-Erzeugung usw. Im Mit- telpunkt steht eher die Ermittlung eines Produktpreises, gegebenenfalls mit Berück- sichtigung von Kundengruppe, Rabatten usw., oder die Bonitätsbewertung mittels eines externen Dienstes. Was Service ist, kann nicht immer eindeutig bestimmt werden. Es kann auch nicht immer genau abgegrenzt werden, welche Funktionalität einer Entität oder einem Aggregat zuzuordnen ist (vgl. Priebsch 2016).



Mit Domain-driven Design sind die folgenden Vorteile verbunden (vgl. Lobacher 2010, S. 32):

* ein besseres Verständnis der Domäne,
* eine saubere Strukturierung der Codes,
* eleganter, schöner Code,

Potenziale für Smart Services erkennen

* Code für jeden verständlich,
* Handhabbarkeit hoher Komplexität,
* klare Trennung der Zuständigkeiten,
* einfache Erweiterbarkeit,
* schnellere Time-to-Market (TTM).

###### DDD in TYPO3 FLOW3/Extbase

* FLOW3: Die Architektur von FLOW3 basiert auf DDD. Es handelt sich um das erste PHP-Framework mit einer ganzheitlichen DDD-Umsetzung (vgl. Lobacher 2010, S. 34).
* TYPO3: Extbase & Fluid – seit TYPO3 4.3 zählt die Systemextension Extbase (Jochen Rau) zum Core, zusammen mit Fluid (Sebastian Kurfürst) und downport von FLOW3- DDD-Konzepten. Es besteht Zukunftssicherheit durch Aufwärtskompatibilität nach FLOW3 (vgl. Lobacher 2010, S. 35).

Domain-driven Design ist mehr als eine Sammlung von Entwurfsmustern. Die beschrie- benen Muster dienen als Voraussetzung für die Entwicklung einer erfolgreich domä- nengetriebenen Software (vgl. Priebsch 2016).

Zusammenfassung

Die Wertschöpfung eines Unternehmens setzt sich aus unterschiedlichen Kompo- nenten zusammen. Diese Komponenten lassen sich mit dem Business Model Can- vas analysieren, modellieren und dokumentieren. Das Business Model Canvas ist eine Methode zur übersichtlichen Darstellung, Visualisierung und Entwicklung von Geschäftsmodellen.

Personas sind ﬁktive Stellvertreter einer Kundengruppe. Sie vereinen charakteristi- sche Eigenschaften, Erwartungshaltungen und Vorgehensweisen.

Unter Customer Journey wird die Kundenreise, sozusagen die einzelnen Schritte des Kunden, verstanden, in welchen er auf das Unternehmen zugeht oder wie er ent- lang der Touchpoints geht bzw. mit dem Unternehmen zusammentrifft oder inter- agiert.

Domain-driven Design (DDD) dient dem Design von komplexer, projektorientierter Software. Es handelt sich um Entwurfsmuster und eine Art Entwicklungsphilosophie.



# Lektion 4

## Entwicklung und Speziﬁkation von Smart Services

#### LERNZIELE

Nach der Bearbeitung dieser Lektion werden Sie …

… mit der Modellierung des Gesamtkontextes vertraut sein.

… die Modellierung sachlicher Abläufe verstehen.

… die Modellierung technischer Schnittstellen nachvollziehen können.

… Werkzeuge für die API-Speziﬁkation kennen und anwenden können.

DL-D-DLBINGSS01-L04

1. Entwicklung und Speziﬁkation von Smart Services

### Einführung

Für die Entwicklung und Speziﬁkation von Smart Services stehen verschiedene Model- lierungsansätze zur Auswahl. Hierzu zählen die Modellierung des Systemkontextes, die Modellierung der fachlichen Abläufe, die Modellierung der technischen Schnittstellen sowie die Anwendung von geeigneten Werkzeugen für die API-Speziﬁkation.

### Modellierung des Systemkontexts

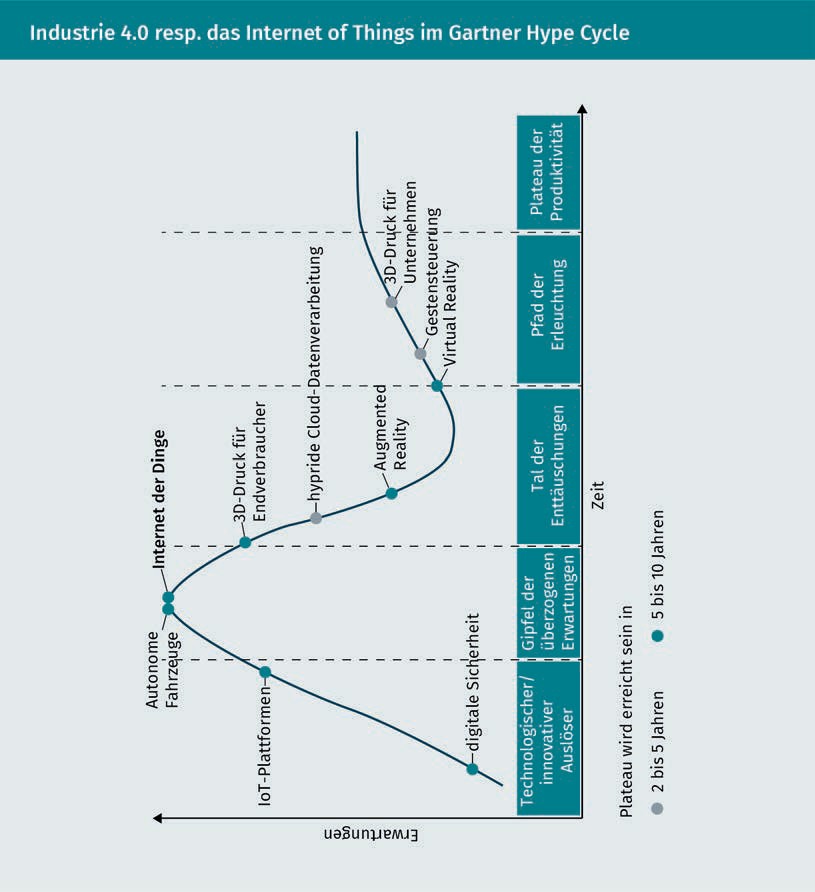
Gartner Hype Cycle Mit dem Gartner Hype Cycle wird ein idealtypischer Ver- lauf von Erwartun- gen an neue Techno- logien bzw. an ein neues Technologie- feld der Industrie 4.0 und des IoT in fünf Phasen dargestellt.

Durch den sogenannten Gartner Hype Cycle wird ein idealtypischer Verlauf von Erwar- tungen an eine neue Technologie oder ein neues Technologiefeld der Industrie 4.0 und des Internets der Dinge dargestellt. Im Cycle werden mehrere Phasen nacheinander durchlaufen (vgl. Gartner 2016):

* Phase 1: technologischer/innovativer Auslöser,
* Phase 2: Gipfel der überzogenen Erwartungen,
* Phase 3: Tal der Enttäuschungen,
* Phase 4: Pfad der Erleuchtung,
* Phase 5: Plateau der Produktivität.

Die nachfolgende Darstellung verdeutlicht den Hype Cycle nach Gartner für das Jahr 2015.

Entwicklung und Speziﬁkation von Smart Services

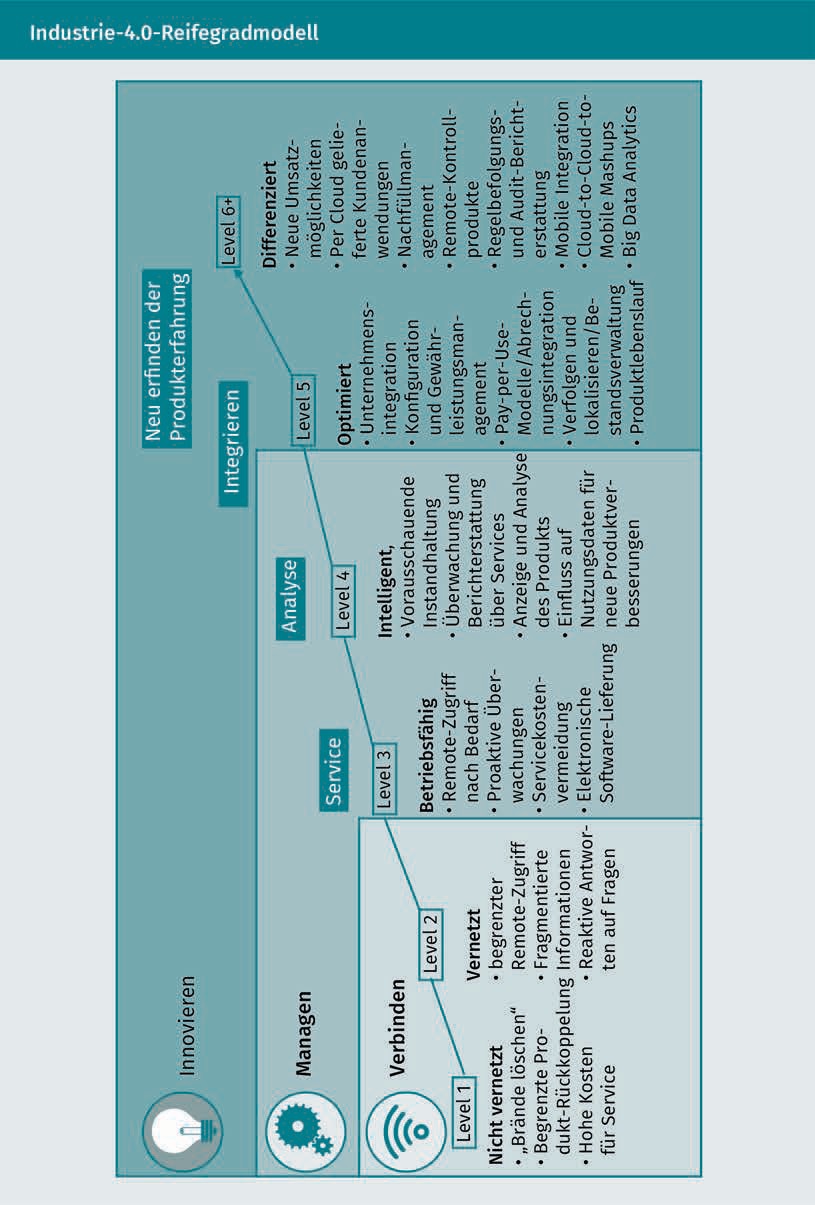


Im Gartner Hype Cycle erkennt man, dass das Internet der Dinge einen der wesentli- chen Bestandteile von Industrie 4.0 darstellt und sich in der zweiten Phase bzw. dem Gipfel der überzogenen Erwartungen zeigt. Demnach wird auch die Industrie 4.0 hin- sichtlich dieses Aspektes aktuell mit überzogenen Erwartungen in Verbindung gebracht. Dies bedeutet, dass es für die Unternehmen derzeit schwierig ist, die Erwartungen, wel- che die Kunden an Industrie-4.0-Lösungen richten, zu erfüllen. In zahlreichen Unter- nehmen besteht noch kein einheitliches Verständnis hinsichtlich der Industrie 4.0 sowie entsprechender Geschäftsmodelle (vgl. Echterhoff et al. 2016). Daher ist es für Unternehmen schwer, den eigenen Fortschritt hinsichtlich der Umsetzung von Industrie

4.0 zu bewerten sowie ihre Aktivitäten entsprechend zu fokussieren (vgl. Haggenmüller et al. 2016).

Das deskriptive Reifegradmodell ermöglicht es, Unternehmen bezüglich ihres Fort- schritts in der Umsetzung von Industrie 4.0 einzuordnen. Nachfolgend wird ein Reife- gradmodell, konkret das „Connected Product Maturity Model“ aufgezeigt (vgl. Lee 2014).

Entwicklung und Speziﬁkation von Smart Services



Im Reifegradmodell zeigt sich, dass insbesondere Level 5 und 6 der beschriebenen Eigenschaften einen besonders ausgeprägten Zustand von Industrie 4.0 repräsentieren. Demnach bedeutet Industrie 4.0 weit mehr als reines Automatisieren und Managen von Produktionsprozessen. Konkret zeichnet sich Industrie 4.0 durch die Innovation vollin- tegrierter und umfassend vernetzter Produkt-Produktions-Service-Systeme aus.

Der Automatisierungsgrad in der Produktion ist in vielen Fällen schon weit fortgeschrit- ten. Eine übergreifende Vernetzung von Produktionsanlagen sowie Montageabschnitten besteht jedoch noch nicht (vgl. Echterhoff et al. 2016; Haggenmüller et al. 2016).

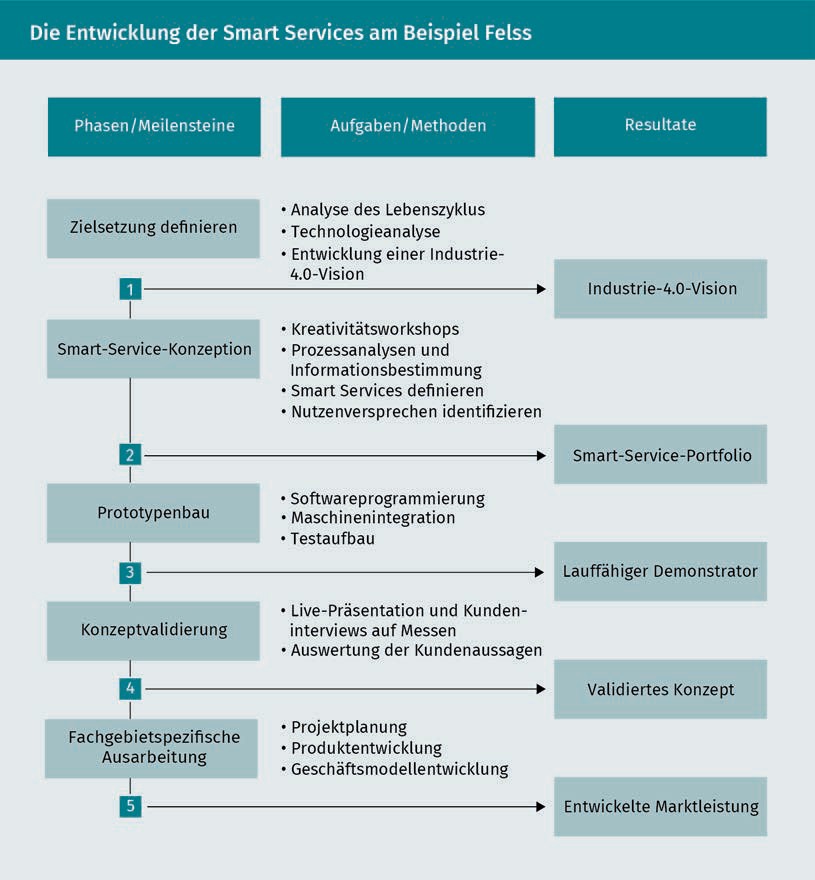
###### Entwicklungsphasen von Smart Services

Die Entwicklung von Smart Services kann in Entwicklungsphasen erfolgen. Der Entwick- lungsprozess vollzieht sich von der Vision bis zur entwickelten Marktleistung. Dies lässt sich am Beispiel des Unternehmens Felss verdeutlichen. So erfolgt beispielsweise bei Felss die Entwicklung von Smart Services in fünf Schritten (vgl. Haggenmüller et al. 2016):

* Entwicklungsphase 1: Deﬁnition der Zielsetzung,
* Entwicklungsphase 2: Smart-Service-Konzeption,
* Entwicklungsphase 3: Prototypenbau,
* Entwicklungsphase 4: Konzeptvalidierung,
* Entwicklungsphase 5: fachgebietsspeziﬁsche Ausarbeitung.

Durch die nachfolgende Abbildung wird das Vorgehen der Entwicklung von Smart Ser- vices anhand eines Phasen-Meilenstein-Modells aufgezeigt, das in einzelnen Phasen verläuft.

Entwicklung und Speziﬁkation von Smart Services



Phase 1: Deﬁnition der Zielsetzung

In der ersten Phase werden neue Lösungen gesucht, mit denen die komplexen Prozesse umfassend gestaltet, optimiert und gesteuert werden können. Dies soll direkt von den Anwendern erfolgen. Die Grundlage hierfür schafft eine Datensammlung. Die Daten sol- len zielgerichtet und intelligent genutzt werden für die Efﬁzienzsteigerung. Damit die Daten für Kunden individuell nutzbar gemacht werden können, werden intelligente Dienstleistungen (Smart Services) in das Leistungsangebot aufgenommen. Die Produkte sollen durch Digitalisierung besser in die Wertschöpfungskette integriert werden. Die Produkterfahrung soll durch die intelligenten Dienste innovieren (vgl. Haggenmüller et al. 2016).

Phase 2: Smart-Service-Konzeption

In der zweiten Phase entsteht die Konzeption der Smart Services. Ideen lassen sich bei- spielsweise für den Bereich „Entwicklung und Innovation“ in Kreativitätsworkshops gewinnen. Durch die Ergebnisse von Prozessanalysen wird deutlich, welche Informatio- nen für die jeweiligen Prozesse benötigt oder wofür diese genutzt werden. Durch ver- schiedene Tools können die entsprechenden Zielgruppen angesprochen werden, z. B. nutzen Entwickler das „Smart Design“. Dagegen nutzen Produktionsverantwortliche das

„Smart Production Management“. Dadurch entsteht eine zentrale und umfassende Datensammlung. Diese stellt die Basis für Auswertungen dar. Daten werden übergrei- fend analysiert und die gesamte Prozesskette wird optimiert. Das Ziel besteht darin, der

„smarten Fabrik“ näher zu kommen. Die nachfolgende Abbildung bietet einen Überblick über das Beispiel Smart Services der Firma Felss (vgl. Haggenmüller et al. 2016).



Phase 3: Prototypenbau

In der dritten Phase beginnt die Ausarbeitung und Detaillierung der Smart Services sowie die Programmierung der ersten Software- und Analyse-Prototypen. Es erfolgt die Vernetzung mit der Sensorik bzw. Testmaschinen, sodass (auch neue) Aufgaben maschinell selbstständig bearbeitet und gelöst werden. Beispielsweise können Mitar- beiter die Programmierung und Gestaltung der Smartphone-Apps in Eigenregie durch- führen (vgl. Haggenmüller et al. 2016).

Entwicklung und Speziﬁkation von Smart Services

Phase 4: Konzeptvalidierung

Die Konzeptvalidierung kann beispielsweise über Live-Präsentationen oder Kundenin- terviews auf Messen erfolgen. Die Kundenaussagen werden im Anschluss ausgewertet. Das Feedback wird für die Verbesserung von Smart Services genutzt.

Phase 5: fachgebietspeziﬁsche Ausarbeitung

In der fünften Phase können sich die Entwickler auf die konkrete Umsetzung und Wei- terentwicklung von zunächst drei bis fünf relevanten Smart Services konzentrieren, die von den Kunden in den Befragungen priorisiert wurden. Mehrere Aspekte der Smart- Service-Entwicklung können parallelisiert ablaufen. Im Kontext der Projektplanung bie- tet sich die Erstellung einer Roadmap an. Durch diese lassen sich die Folgeschritte detailliert beschreiben. Die Fragen richten sich darauf, bis wann ein benötigtes Soft- ware- und Hardware-Element zur Verfügung steht und welche technischen Bedingun- gen (z. B. an der Maschine) dafür erfüllt sein müssen. Die technische Entwicklung voll- zieht sich parallel. Beispielsweise werden erforderliche Sensoren und Schnittstellenpakete angepasst, damit sie industriell genutzt werden können. Im Folge- schritt soll eine vollvernetzte Maschine im industriellen Umfeld dauerhaft zum Einsatz gebracht werden, z. B. ein Feldtest mit Kundenfeedback. Die Erfahrungen des Feldtests können wiederum in die Entwicklung von Soft- und Hardware der Smart Services ein- ﬂießen. Im Anschluss folgt die Umsetzung in der Zusammenarbeit mit strategisch bedeutenden Kunden (vgl. Haggenmüller et al. 2016).

Auch die Entwicklung von neuen Geschäftsmodellen lässt sich auf diese Weise voran- treiben. Das Smart-Services-Paket ist mehr als eine Serviceverbesserung oder Prozess- optimierung. Eine Nutzung der produktbegleitenden Dienstleistungen ist auf der Grundlage von unterschiedlichen Geschäftsmodellen möglich. Es ist erforderlich, dass der spürbare Nutzen für den Kunden im Mittelpunkt der Entwicklung von Smart-Ser- vice-Geschäftsmodellen steht (vgl. Haggenmüller et al. 2016).

### Modellierung fachlicher Abläufe

Die Komplexität von Smart Services kann durch die Modellierung von Lebenszyklen transparenter werden. Entwickler von Smart Services sollten die mit dem Betrieb des Service verbundenen Effekte gut erkennen können. Hierzu bedarf es eines guten Ver- ständnisses der Beziehungen, die zwischen den einzelnen Produkt-, Software- und Dienstleistungskomponenten eines Smart Service bestehen. Das Ziel ist eine verbes- serte Handhabung der Komplexität. Hierzu kann die Modellierung von Lebenszyklen für Smart Services für verschiedene Stakeholder wesentlich beitragen (vgl. Wellsandt/ Anke/Thoben 2016, S. 233ff.).

In einem Ansatz von Wellsandt/Anke/Thoben (2016, S. 233ff.) erfolgt die Bewertung eines Modells der Prozesse und Strukturen zur Erbringung von Smart Services für ver- netzte Produkte. In diesem Rahmen werden Smart Services für vernetzte Produkte cha- rakterisiert. Dabei wird auf die Systembestandteile und die Bedeutung von Lebenszyk- len eingegangen und daraus werden Ziele der Modellierung abgeleitet. Ferner wird in diesem Ansatz ein Smart Service für die automatisierte Bereitstellung von Verbrauchs-

material am Beispiel von 3D-Druckern beschrieben sowie unter Nutzung der Lifecycle Modeling Language (LML) modelliert. Schließlich wird das Modell bezüglich der Erfül- lung der Ziele und Aussagekraft für verschiedene Stakeholder geprüft.

Hybride Leistungs-

bündel Smart Services erbringen CPS- gestützt Dienstleis- tungen für techni- sche Produkte.

Auf dieser Basis können Aussagen über die Eignung von Lebenszyklusmodellen für Smart Services im Konzeptionsprozess und Forschungsbedarfe abgeleitet werden (vgl. Wellsandt/Anke/Thoben 2016, S. 233ff.). Smart Services lassen sich als hybride Leis- tungsbündel verstehen, die CPS-gestützt (CPS steht für cyber-physisches System, engl.

„cyber-physical system“) Dienstleistungen für technische Produkte erbringen (vgl. Schä- fer/Jud/Mikusz 2015). Für die Betrachtung der Lebenszyklen sind die für Smart Services relevanten Systembestandteile wesentlich. Die nachfolgende Tabelle zeigt die Elemente eines allgemeinen CPS für die Ausprägung eines CPS zur Unterstützung von Smart Ser- vices auf (vgl. Wellsandt/Anke/Thoben 2016, S. 233ff.).

|  |  |
| --- | --- |
| Konkretisierung der CPS-Elemente für einen Smart Service – allgemeines CPS-Ele- ment | |
| Allgemeines CPS-Element | Ausprägung des CPS-Elements für einen Smart Service |
| physikalischer Prozess | Lokale, physikalische Funktion des Produkts |
| Sensoren und Aktoren zur Erfas- sung von bzw. zum Einwirken auf physikalische Vorgänge | Diverse Ausprägungen, abhängig vom konkre- ten Produkt, z. B. Sensoren für Füllstände, Temperaturen, Druck sowie Aktoren zur Schal- tung bzw. Steuerung |
| Eingebettete Systeme | Eingebettetes System zur Steuerung des physi- schen Produkts mit eingebetteter Software und Kommunikationsmodul |
| Digitale Netze | Diverse Technologien zur Verbindung von Embedded System und Betreiberplattform,  z. B. über Mobilfunk, WLAN oder Unterneh- mensnetze |
| Nutzung weltweit verfügbarer Daten und Dienste | (Cloudbasierte) Softwareplattformen des Betreibers sowie ggf. zusätzliche Internet- dienste, z. B. elektronische Marktplätze |
| Multimodale Mensch-Maschine- Schnittstellen | Diverse Ausprägungen der Benutzerinteraktion am Produkt selbst, mit mobilen Apps oder Web-Applikationen |

Entwicklung und Speziﬁkation von Smart Services

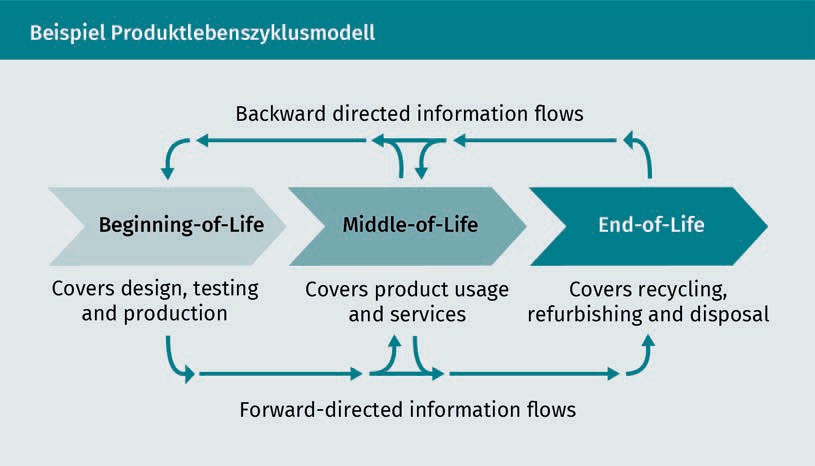
|  |  |
| --- | --- |
| Allgemeines CPS-Element | Ausprägung des CPS-Elements für einen Smart Service |
| Managementprozesse | Management des Service (Buchung, Konﬁgura- tion, Abrechnung) |
| Koordinationsprozesse | Erbringung des Service, Steuerung der Leis- tungserbringung |
| Logistikprozesse | Lieferung von physischen (z. B. Ersatzteile, Ver- brauchsmaterial) oder digitalen Bestandteilen (z. B. Softwareupdates) |

###### Lebenszyklen in Smart Services

Als Lebenszyklus werden verschiedene, aufeinanderfolgende Zeitabschnitte aufgefasst, die den Weg eines Produkts oder einer Dienstleistung kennzeichnen. Der Lebenszyklus lässt sich aus ﬂussbasierter Perspektive in Phasen untergliedern. Es ist eine Untertei- lung in die folgenden drei Phasen möglich (Kiritsis 2011):

* + - Beginning-of-Life (BOL),
    - Middle-of-Life (MOL),
    - End-of-Life (EOL).

Diese Phasen lassen sich in Prozesse unterteilen, so z. B. Material-, Energie- und Infor- mationsﬂüsse entlang des Lebenszyklus (vgl. Herrmann 2010). Nachfolgend wird ein Beispiel für ein Modell eines Produktlebenszyklus aufgezeigt.



In der Regel trägt der Anbieter eines Smart Service die Verantwortung für den gesam- ten Lebenszyklus des zugrunde liegenden Service-Systems. Es sind Sach- und Dienst- leistungen als einzelne Komponenten zu entwickeln, die integrativ zusammenwirken müssen. Dabei weisen die Bestandteile eines Smart Service unterschiedliche Lebens- zyklen auf. Es können beispielsweise unterschiedliche Versionen einer Komponente existieren, z. B. eine Softwareplattform (vgl. Wellsandt/Anke/Thoben 2016, S. 237ff.). Genauso können Änderungen im Lebenszyklus einzelner Komponenten neue Kunden- bedarfe, Gesetzesänderungen oder die Verfügbarkeit neuer Technologien ergeben (Wol- fenstetter et al. 2015). Aus diesem Grund sind die in der Abbildung dargestellten Lebenszyklen simultan zu betrachten.

Die folgenden Beispiele zeigen Auswirkungen, die sich infolge von Veränderungen im Lebenszyklus eines Leistungsbestandteils abzeichnen (vgl. Wellsandt/Anke/Thoben 2016, S. 233ff.):

* + - * Der Defekt eines Produkts führt zum Austausch des Produkts und es wird eine Logis- tikleistung erforderlich zur Anlieferung, Rücknahme und Anpassung der Konﬁgura- tion des Service (Anpassung auf neue Hardware).
      * Durch den Defekt eines Embedded Systems kommt es zum Austausch der ent- sprechenden Hardware, zum Update der Embedded Software oder zur Übertragung der alten Konﬁguration (die digitale Produktidentität soll erhalten bleiben).
      * Eine Änderung im Service-Management-Prozess (z. B. Einführung von Abo- oder Pre- paid-Modellen) erfordert die Anpassung der Betreiberplattform. Unter Umständen ist auch die Anpassung der Servicekonﬁguration (Abrechnungsmodell wählen) durch Kunden erforderlich und ein Update der Embedded Software.

Entwicklung und Speziﬁkation von Smart Services

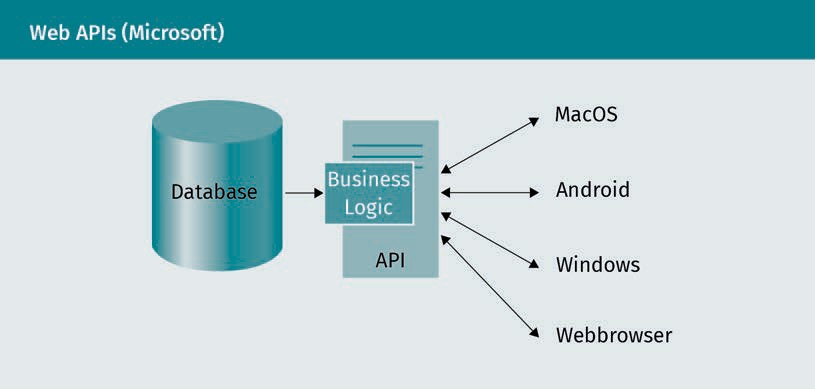
### Modellierung technischer Schnittstellen

Bei einer API handelt es sich um eine Schnittstelle. Diese wird durch ein Softwaresys- tem für die Einbindung anderer Programme bereitgestellt. Ursprünglich stammt der Begriff API aus dem englischsprachigen Raum. Es handelt sich um die Kurzform von

„Application Programming Interface“. Übersetzt bedeutet API „Schnittstelle zur Anwen- dungsprogrammierung“. Es ﬁndet sich aber auch häuﬁg die Bezeichnung „Programmier- schnittstelle“ (vgl. Behrendt 2018; Ding/Johnson 2006, S. 1–26).

„In computer programming, an application programming interface (API) is a set of sub- routine deﬁnitions, protocols, and tools for building application software. In general terms, it is a set of clearly deﬁned methods of communication between various soft- ware components. A good API makes it easier to develop a computer program by provi- ding all the building blocks, which are then put together by the programmer” (Labois- sonniere 2018).

Durch die Bereitstellung der API bzw. der Schnittstelle wird anderen Programmen ein Tool zur Verfügung gestellt, durch welches sich diese an das Softwaresystem anbinden können. Auf diese Weise können Softwareentwickler die Hardware, beispielsweise einen Monitor oder die auf der Festplatte beﬁndlichen Daten, beeinﬂussen. Dies ist möglich, ohne dass diese direkt angesprochen werden. Das Betriebssystem fungiert hierbei als Schnittstelle. Dieses kann über bereitgestellte Bibliotheken Anfragen der Programme entgegennehmen und an die Hardware weiterleiten (vgl. Behrendt 2018; Ding/Johnson 2006, S. 1–26).



APIs sind insbesondere im Kontext der Verwendung von Web-Diensten relevant. Diese erlauben den Entwicklern mit Unterstützung der zur Verfügung gestellten Schnittstel- len, bereitgestellten Content dynamisch in das eigene Programm zu integrieren. Dies bedeutet, dass APIs zum Austausch sowie zur Weiterverarbeitung von Daten und Inhal- ten zwischen verschiedenen Webseiten, Programmen und Content-Anbietern dienen. Zusätzlich wird durch APIs Dritten der Zugang zu bislang nicht zugänglichen Datenpools und Benutzerkreisen ermöglicht (vgl. Ding/Johnson 2006, S. 1–26; Behrendt 2018).

Aus vorwiegend technischer Perspektive lassen sich APIs als maschinelles Äquivalent zum User-Interface bezeichnen. APIs wurden jedoch für Menschen optimiert und sind somit auch „menschenlesbar“ (vgl. Behrendt 2018).

Eine API ist eine für Software angepasste Schnittstelle. Die API ist maschinenlesbar. Durch das Application Programming Interface wird ein klar abstrahierter und struktu- rierter Zugriff auf die Funktionen des Backends ermöglicht. Zudem lassen sich z. B. Daten in einer leicht weiterzubearbeitenden und reduzierten Form austauschen. Zu den Web-APIs zählen interne, externe, Plattform-, Authentiﬁzierungs- und Autorisierungs- APIs (vgl. Behrendt 2018; Ding/Johnson 2006, S. 1–26).

So besteht für Entwickler etwa über die YouTube-API die Möglichkeit, nach Videos mit gewünschten Parametern zu suchen, z. B. Name, Länge usw. Durch die API wird die Ant- wort in Form einer XML-Datei zurückgeschickt. Die Antwort dient dann beispielsweise zur Auswertung für die eigene Webseite (vgl. Behrendt 2018).

### Werkzeuge für die API-Speziﬁkation

###### Java SE/FX APIs

Zur API-Speziﬁkation ﬁnden sich beispielsweise auf der Webseite von Oracle aktuelle Ansätze. So z. B. „Java Platform, Standard Edition & Java Development Kit Version 9 API Speciﬁcation” in den folgenden Sektionen (vgl. Oracle 2018):

* + - Java SE: „The Java Platform, Standard Edition (Java SE) APIs deﬁne the core Java plat- form for general-purpose computing. These APIs are in modules whose names start with java”.
    - JDK: “The Java Development Kit (JDK) APIs are speciﬁc to the JDK and will not neces- sarily be available in all implementations of the Java SE Platform. These APIs are in modules whose names start with jdk”.
    - JavaFX: “The JavaFX APIs deﬁne a set of user-interface controls, graphics, media, and web packages for developing rich client applications. These APIs are in modules whose names start with javafx”.

Bei der Java-9-Speziﬁkation Public-Review-Version für Java SE 9 von 2017 sind im Ver- gleich zur Version 8 einige Neuerungen und Ergänzungen hinzugekommen, so z. B. „eine ausführliche Zusammenfassung sämtlicher Änderungen an API-Speziﬁkationen, Ent- würfe für die Java Language Speciﬁcation (JLS) und Java Virtual Machine Speciﬁcation (JVMS) sowie annotierte Versionen der Java-9-API-Speziﬁkation und der Java-Native- Interface-Speziﬁkation (JNI), die alle Unterschiede zu Java SE 8 aufzeigen“ (Schlosser 2017).

Entwicklung und Speziﬁkation von Smart Services

###### JSON API

Die Wiederverwertbarkeit von Codes über einzelne Projekte hinweg wird durch die Spe- ziﬁkation JSON API ermöglicht. JSON API führt zu verschiedenen Vorteilen (vgl. Otto-Wit- tes 2018).

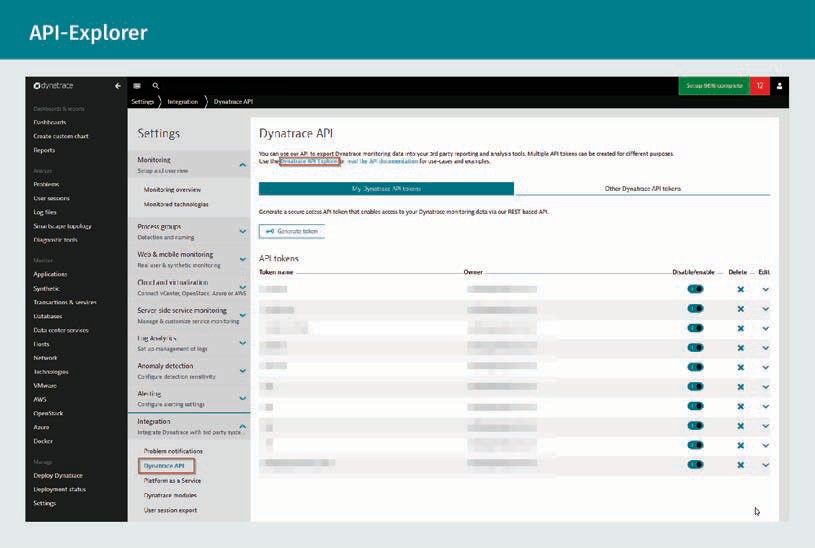
Durch zahlreiche Projekte werden JSON-basierte APIs zur Verfügung gestellt oder sie nutzen diese, meist nach dem RESTful-Prinzip. Eine Kombination aus JSON und REST hat sich stark verbreitet. Jedoch unterscheiden sich diese APIs zumindest geringfügig voneinander. Diesbezüglich kann die JSON-API-Speziﬁkation diese Unterschiede ver- mindern bzw. beseitigen. Die Teams, die an einer Implementierung von JSON arbeiten, gelangen hinsichtlich der exakten JSON-Struktur, Benennung von REST-Endpunkten etc. zu unterschiedlichen Ergebnissen. Die Konsequenz besteht darin, dass jedes Team ent- sprechend ein eigenes, individuelles API gestaltet und baut. Dies erweist sich jedoch hinsichtlich der Wiederverwertbarkeit von Codes über einzelne Projekte hinweg als äußerst ungünstig. Durch die Speziﬁkation JSON API wird dieses Problem gelöst (vgl. Otto-Wittes 2018).

###### API-Explorer/OAS

Durch die APM-Spezialisten von Dynatrace wurden ein API-Explorer und eine OAS-Spe- ziﬁkation (OpenAPI Speciﬁcation) für ihre Software entwickelt (vgl. Dynatrace 2018; Schlede 2018).

Die OpenAPI Speciﬁcation ist verbunden mit dem neuen API-Explorer für Nutzer der Dynatrace-Software. OAS ist ein Linux Foundation Collaborative Project, welches die Beschreibung einer Standardschnittstelle für REST APIs deﬁniert, die als unabhängig von Programmiersprachen gilt. Dynatrace stellt eine OAS-Speziﬁkation für alle Dynat- race-API-Endpunkte zur Verfügung. Dadurch kann das Unternehmen den API-Explorer bereitstellen (vgl. Dynatrace 2018; Schlede 2018).

Durch die Neuerungen kann automatisch für jeden Dynatrace-REST-Endpunkt eine OAS- Speziﬁkation angeboten werden. Die OAS-Speziﬁkationen eignen sich für die automati- sche Erstellung von API-Dokumentationen sowie für die Erstellung eines API-Explorers oder für die Überprüfung von neuen Endpunkten und veraltetem Softwarebereich (Deprecations) (vgl. Schlede 2018; Dynatrace 2018).



Für Nutzer ist es möglich, den API-Explorer direkt im Dashboard der Software aufzuru- fen. Sämtliche verfügbaren API-Endpunkte für die entsprechende Dynatrace-Umgebung werden aufgelistet. Ein Link zu der „Raw“-OAS-Speziﬁkation bietet die Möglichkeit, diese mit einem OAS-kompatiblen Tool direkt aufzurufen, so z. B. mit dem Swagger. Mittels des Explorers sind Entwickler in der Lage, API-Aufrufe zu entwickeln und auszuführen. Zudem steht auch die OAS-Speziﬁkation der Dynatrace API zur Verfügung, mit der jede gewünschte Sprachbindung schnell zur Verfügung gestellt werden kann. Durch die Soft- ware wird zudem die Erstellung ﬂexibler Kombinationen von verschiedenen API-Aufru- fen mittels automatisch angelegter cURL-Kommandos unterstützt. Diese lassen sich von den Entwicklern in bash-Scripts für die Automatisierung oder für eine Integration von Dynatrace-API-Aufrufen in Tools von Drittherstellern nutzen (vgl. Schlede 2018; Dynat- race 2018).

###### Microsoft Azure-Portal – API-Management-Instanz (APIM)

API-Management-Dienste

Entwickler können das Azure-Portal nutzen, um der API-Management-Instanz (APIM) beispielsweise manuell eine API hinzuzufügen. Ein häuﬁges Szenario zur Erstellung einer leeren API und des manuellen Deﬁnierens ist auch das Simulieren von API-Ant- worten. Zur Erstellung einer leeren API wird beispielsweise als Backend-API

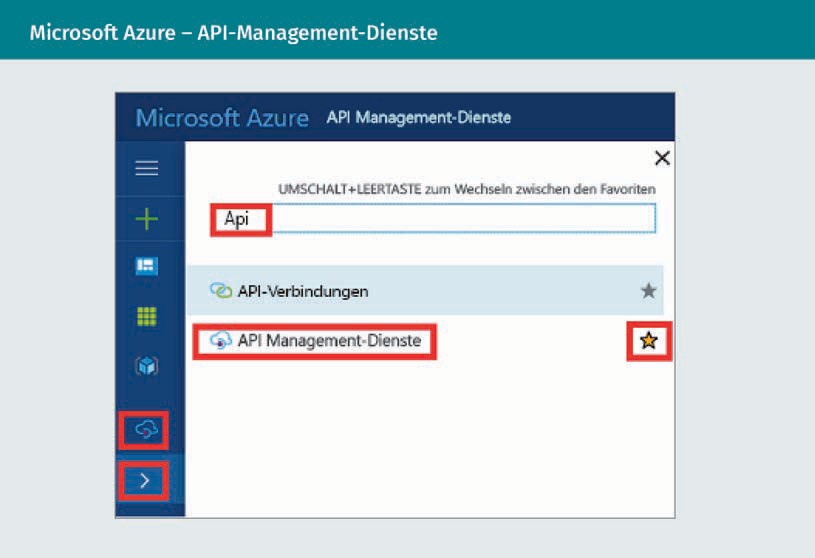
„httpbin.org“ (öffentlicher Testdienst) angegeben (vgl. Microsoft 2017).

Es wird vorausgesetzt, dass der Entwickler den Schnellstart „Erstellen einer Azure API Management-Instanz“ absolviert.

Entwicklung und Speziﬁkation von Smart Services

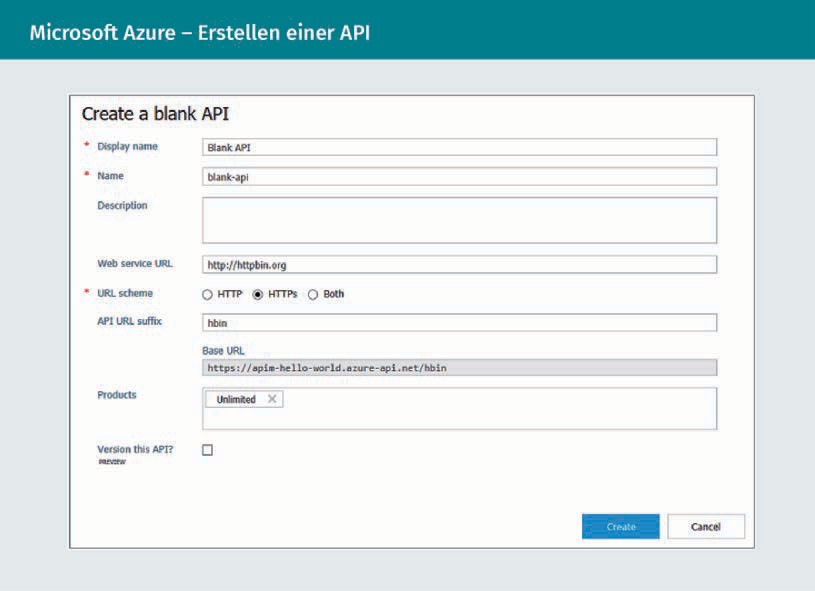
Um zur APIM-Instanz zu navigieren, führt der Entwickler etwa die folgenden Schritte aus (vgl. Microsoft 2017):

* „Melden Sie sich beim Azure-Portal.“
* „Wählen Sie auf dem Bildschirm oben links ‚Alle Dienste‘ aus.“
* „Geben Sie im Suchfeld den Suchbegriff ‚api‘ ein.“
* „Klicken Sie auf ‚API Management-Dienste‘.“
* „Wählen Sie Ihre APIM-Dienstinstanz aus.“



Um eine API zu erstellen, werden die folgenden Punkte ausgeführt (vgl. Microsoft 2017):

* „Wählen Sie unter ‚API Management‘ die Option ‚APIs‘ aus.“
* „Wählen Sie im linken Menü die Option ‚+ API hinzufügen‘ aus.“
* „Wählen Sie in der Liste ‚Leere API‘ aus.“
* „Geben Sie Einstellungen für die API ein.“



Es lassen sich zudem beispielsweise Vorgänge/parametrisierte Vorgänge hinzufügen und testen, weitere APIs anfügen, Backend-API mit OpenAPI-Speziﬁkation importieren, die standardmäßige XML-Darstellung einer SOAP-API importieren u. v. m. (vgl. Microsoft 2017).

Zusammenfassung

Mit dem Gartner Hype Cycle wird ein idealtypischer Verlauf von Erwartungen an neue Technologien bzw. an ein neues Technologiefeld der Industrie 4.0 und des IoT in fünf Phasen dargestellt. Die Komplexität von Smart Services kann durch die Modellierung von Lebenszyklen transparenter werden. Entwickler von Smart Ser- vices sollten die mit dem Betrieb des Service verbundenen Effekte gut erkennen können. Hierzu bedarf es eines guten Verständnisses der Beziehungen, die zwi- schen den einzelnen Produkt-, Software- und Dienstleistungskomponenten eines Smart Service bestehen. Das Ziel ist eine verbesserte Handhabung der Komplexität. Smart Services können als hybride Leistungsbündel aufgefasst werden, die CPS- gestützt Dienstleistungen für technische Produkte erbringen.

Bei einer API handelt es sich um eine Schnittstelle. Diese wird durch ein Software- system für die Einbindung anderer Programme bereitgestellt. Ursprünglich stammt der Begriff API aus dem englischsprachigen Raum. Es handelt sich um die Kurzform von „Application Programming Interface“ (API). Übersetzt bedeutet API „Schnittstelle zur Anwendungsprogrammierung“. Es ﬁndet sich aber auch häuﬁg die Bezeichnung

Entwicklung und Speziﬁkation von Smart Services

„Programmierschnittstelle“. Zur API-Speziﬁkation ﬁnden sich für Entwickler verschie- dene Werkzeuge und Hilfsinstrumente, so z. B. Java Platform, Standard Edition & Java Development Kit Version 9 API Speciﬁcation, JSON API, API-Explorer und eine OAS-Speziﬁkation (OpenAPI Speciﬁcation) oder Microsoft Azure-Portal – API- Management-Instanz (APIM)/API-Management-Dienste.



# Lektion 5

## Service-Architekturen

#### LERNZIELE

Nach der Bearbeitung dieser Lektion werden Sie …

… den Kontext Infrastructure/Platform/Software as a Service (SaaS) verstehen.

… Everything as a Service (EaaS) kennen.

… serviceorientierte Architekturen (SOA) kennen.

… Microservices verstehen und anwenden können.

DL-D-DLBINGSS01-L05

1. Service-Architekturen

### Einführung

Für die Gestaltung von Service-Architekturen bieten sich verschiedene Ansätze und Modelle an. Grundlegend sind der Kontext von Infrastructure/Platform und das Modell Software as a Service (SaaS). Weitere Ansätze sind Everything as a Service (EaaS), ser- viceorientierte Architekturen (SOA) und Microservices.

### Infrastructure/Platform/Software as a Service (SaaS)

Software as a Service

(SAAS)

Hierunter wird ein Vertriebsmodell ver- standen. Dabei wird die Software von einem Anbieter oder Service Provider geh-

ostet und für Anwender über ein Netzwerk zur Verfü-

gung gestellt.

Bei Software as a Service (SAAS) handelt es sich um ein Vertriebsmodell. SaaS lässt sich als Teil des Gesamtkonzepts Cloud-Computing auffassen. Dabei wird die Software von einem Anbieter oder Service Provider gehostet und für Anwender über ein Netz- werk (typischerweise das Internet) zur Verfügung gestellt (vgl. Rouse 2018). Im Zusam- menhang mit Software as a Service ist es möglich, dass der Kunde eine Softwarean- wendung online gegen eine Gebühr nutzen kann (vgl. o. V. 2018b). Üblicherweise werden diese monatlich oder jährlich abgerechnet. Es ist aber auch eine Abrechnung pro Anwendung grundsätzlich denkbar.

Beispiele für SaaS sind die Ersteigerung eines Gameboys bei eBay, die Bestellung eines Buches bei Amazon und die Steuererklärung am PC mithilfe einer Datev-Software.

Die genannten exemplarischen Anwendungen haben gemeinsam, dass der Kunde ein Computerprogramm nutzt, ohne dass es jedoch für ihn erforderlich ist, Software auf dem eigenen PC zu installieren. Der Anbieter, z. B. eBay oder Amazon, stellt das ent- sprechende Programm über den Internetbrowser zur Verfügung. Dieses Prinzip wird

„One-to-Many“ genannt. Die eine einheitliche Lösung kann gleichzeitig von vielen Nut- zern in Anspruch genommen werden. Die Daten, die jeder einzelne Nutzer in die Bild- schirmmaske eingibt, dürfen nicht mit den Daten anderer User vermischt werden (vgl. Lixenfeld 2008).

Derartige One-to-Many-Lösungen bestehen bereits längere Zeit, jedoch war der Umstand neu, diese für die Unternehmenssoftware zu verwenden, beispielsweise zum Kundenmanagement (CRM).

Weitere Services des Cloud-Computing umfassen Infrastructure as a Service (IaaS) und Platform as a Service (PaaS) (vgl. 1 & 1 Digitalguide 2018).

Mit SaaS können die Nutzer eine vergleichsweise aufwendige Hardware sparen. Ein internetfähiges Endgerät ist ausreichend, z. B. PC, Smartphone, Tablet, iPad usw. Für SaaS ist das jeweilige Betriebssystem nicht relevant. Die Software arbeitet ohnehin im Hintergrund beim SaaS-Dienstleister. Dagegen ist auf dem Gerät des Nutzers lediglich ein Webbrowser erforderlich, der die Steuerbefehle beim Bedienen weitergibt (vgl. Stel- zel-Morawietz 2010).

Service-Architekturen



SaaS hat sich zum vorherrschenden Bereitstellungsmodell für Software entwickelt. Dies hängt unter anderem damit zusammen, dass die zugrunde liegende Technologie für Internetdienstleistungen und die serviceorientierte Architektur heute weit ausgereift sind. Außerdem werden neue Entwicklungsansätze immer beliebter, so z. B. Ajax. Die Internetbandbreite ist heute in vielen Gebieten für SaaS ausreichend. Somit bekommen Anwender weltweit Zugriff auf die Software. Es besteht eine enge Verbindung von SaaS, dem Model Application Service Provider (ASP) und dem Software-on-Demand-Modell (vgl. Rouse 2018).

### Everything as a Service (EaaS)

Im Zusammenhang mit der digitalen Transformation wird zunehmend auch die „Cloudi- ﬁzierung der IT“ thematisiert. Hierunter versteht man den „Umbau der IT-Architektur hin zu Cloud-Lösungen“ (Schaudel 2017). Zunehmend wird „alles als Service zur Verfü- gung gestellt“ (ebd.).

Everything as a Service (EaaS) lässt sich deﬁnieren als „bedarfsorientierte Bereitstel- lung von IT-Ressourcen. Kosten für diese Ressourcen entstehen hauptsächlich durch deren Nutzung und es fallen i. d. R. keine Kosten für deren Kauf an“ (Fehling 2018).

Beispiele hierfür sind Amazon Web Services, Google App Engine, Open Telekom Cloud, Salesforce, Adobe Creative Cloud, Ofﬁce 365 u. v. m.

Everything as a Ser- vice (EaaS)

Dieser Begriff bezeichnet die bedarfsorientierte Bereitstellung von IT-Ressourcen.

Die häuﬁgsten Services lassen sich in den folgenden Gruppen zusammenfassen (vgl. Schaudel 2017):

* IaaS – Infrastructure as a Service (Public/Private/Hybrid),
* SaaS – Software as a Service,
* PaaS – Platform as a Service.

###### IaaS – Infrastructure as a Service

Infrastructure as a Service (IaaS) gilt als Kern des Cloud-Computing. Diese Lösung ist interessant für User bzw. Unternehmen, die kein eigenes Rechenzentrum betreiben bzw. im Unternehmen keinen eigenen Server stehen haben, jedoch Wert auf Datensi- cherheit legen. Unternehmen können diese Dienste in Anspruch nehmen, anstatt sich selbst teure Server zu kaufen und diese zu betreiben, es kann also günstiger sein, sich in einem Rechenzentrum einzumieten. Dabei entsteht die eigene Cloud. Durch den Anbieter wird die Infrastruktur zur Verfügung gestellt. Der Anbieter ist auch verantwort- lich für die Hardware und die ständige Wartung. Es lassen sich die drei folgenden For- men von IaaS unterscheiden (vgl. Schaudel 2017):

* Public IaaS: Virtualisierungstechnologien ermöglichen, dass sich mehrere Nutzer einen Server teilen.
* Private IaaS: Einsatz einer Virtualisierungstechnologie, jedoch erfolgt die Anpassung eines Servers speziell an die Bedürfnisse eines Unternehmens. Dieser wird in die Unternehmensumgebung eingebunden. Dies bedeutet, dass dieser Server aus- schließlich dem Unternehmen zur Verfügung steht. Es ist nicht möglich, dass andere Unternehmen diesen Server nutzen.
* Hybrid IaaS: „Hybrid“ bezieht sich auf eine „Bündelung“, d. h., es werden sowohl Public IaaS als auch Private IaaS genutzt.

Zusammenfassend lässt sich festhalten, dass IaaS mit den Vorteilen verbunden ist, dass die Infrastruktur skalierbar ist, lediglich das bezahlt werden muss, was tatsächlich benötigt und beansprucht wird, und keine hohen Anschaffungskosten anfallen. Es ist jedoch das Risiko einer etwaigen Abhängigkeit vom Dienstleister zu berücksichtigen (vgl. Schaudel 2017).

###### SaaS – Software as a Service

Hinsichtlich Software as a Service (SaaS) lassen sich zusammenfassend Vorteile nennen wie geringe Anschaffungskosten, Aktualität der Software oder dass keine Unterhal- tungs- oder Folgekosten anfallen. Die Risiken bestehen darin, dass im Falle einer Ver- tragskündigung die Software nicht weiter genutzt werden darf, selbst nicht in einer älteren Version (vgl. Schaudel 2017).

Service-Architekturen

###### PaaS – Platform as a Service

Das Modell Platform as a Service (PaaS) ist insbesondere für Systemarchitekten oder Programmierer interessant.

Da sich die Tools, die für die Entwicklung verwendet werden, ständig und vor allem schnell ändern, verursacht die Aktualität entsprechenden Aufwand. Vor diesem Hinter- grund bietet sich PaaS an, denn mit PaaS stehen einem Unternehmen stets aktuelle Datenbanken, Entwicklungswerkzeuge, Frameworks und Laufzeitumgebungen zur Verfü- gung. Dies bedeutet für das Unternehmen bzw. den Entwickler oder Programmierer, dass die Bereitstellung der gesamten Infrastruktur entfällt. Diese wird ausgelagert (vgl. Schaudel 2017).

Zusammenfassend lassen sich die folgenden Vorteile des Modells PaaS nennen: Es fal- len kürzere Entwicklungszeiten an, es besteht eine Standardisierung der Entwicklungs- prozesse und Zugriff auf zahlreiche APIs und Plug-ins. Jedoch sind auch Risiken bzw. Nachteile zu berücksichtigen, so z. B. mögliche Einschränkungen durch APIs und Pro- grammiermodelle (vgl. Schaudel 2017).

### Serviceorientierte Architekturen (SOA)

Serviceorientierte Architektur (SOA) gilt heute als serviceorientiertes Paradigma in der Softwareentwicklung. Dies gilt insbesondere im Zusammenhang mit der Entwicklung und Realisierung verteilter Geschäftsprozesse. Durch SOA wird die Erstellung von kom- plexen und verteilten fachlichen Anwendungen ermöglicht. SOA ist anwendbar, wenn ein Projekt auf zahlreichen Webservices-Kompetenzen aufbaut oder wenn Legacy- Anwendungen in den modernen Workﬂow einzubeziehen sind (vgl. Josuttis 2008, S. 15ff.).

SOA ist ein Paradigma bzw. ein Denkmuster oder Konzept, das zu einem Wertesystem für große Systeme führt, welche in unterschiedlicher Verantwortung liegen. Insofern handelt es sich bei SOA um keine konkrete Architektur. Es kann jedoch im Sinne einer Philosophie oder Vorgehensweise zu einer konkreten Architektur führen. SOA ist weder ein fest deﬁniertes Tool noch ein Framework. Es führt zu bestimmten Entscheidungen beim Entwurf einer Softwarearchitektur (vgl. Josuttis 2008, S. 15).

SOA soll so wie andere IT-Konzepte auch zur Verbesserung und Flexibilität beitragen. Es sollen (vgl. Josuttis 2008, S. 16f.) …

* + - … Lösungen gefunden werden,
    - Daten gespeichert werden,
    - Daten verwaltet werden,
    - Prozesse bzw. Geschäftsprozesse automatisiert werden, welche die Daten nutzen, sowie
    - qualitative Lösungen rechtzeitig gefunden werden.

Serviceorientierte Architektur (SOA) Diese gilt als service- orientiertes Para- digma in der Softwa- reentwicklung, insbesondere für Geschäftsprozesse. Sie soll wie andere

IT-Konzepte zur Ver- besserung und Flexi- bilität beitragen.

Hierzu ist Flexibilität erforderlich, eine klare Organisation, klare Rollen sowie klare Pro- zesse. SOA ist auch mit solchen nicht-technischen Aspekten verknüpft (vgl. Josuttis 2008, S. 17).

Als SOA-Treiber gelten (vgl. Josuttis 2008, S. 17ff.):

* verteilte Systeme,
* unterschiedliche Eigentümer,
* Heterogenität.

Es bestehen die folgenden SOA-Konzepte (vgl. Josuttis 2008, S. 21f.):

* Services,
* hohe Interoperabilität,
* lose Kopplung (Flexibilität, Skalierbarkeit, Fehlertoleranz).

Als SOA-Zutaten sind erforderlich (vgl. Josuttis 2008, S. 23ff.):

* Infrastruktur,
* Architektur,
* Prozesse,
* Governance.

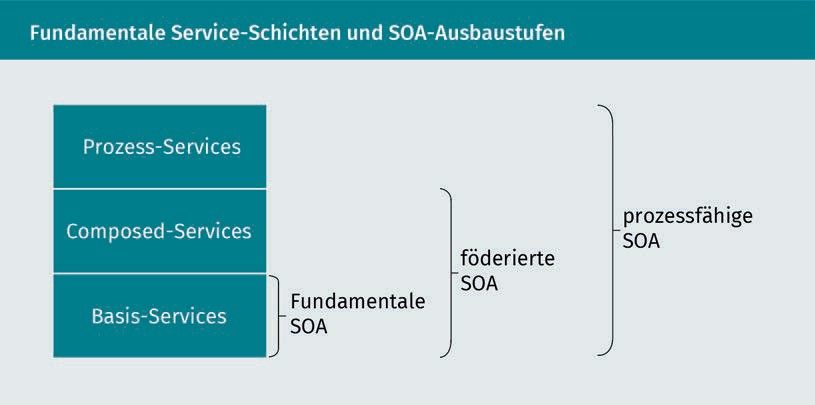
###### Klassiﬁzierung von Services

Services haben verschiedene Eigenschaften und lassen sich unterschiedlich deﬁnieren. Eine fundamentale Service-Klassiﬁzierung ist nach den folgenden drei Kriterien mög- lich (vgl. Josuttis 2008, S. 81):

* Basis-Services,
* Composed Services,
* Prozess-Services.

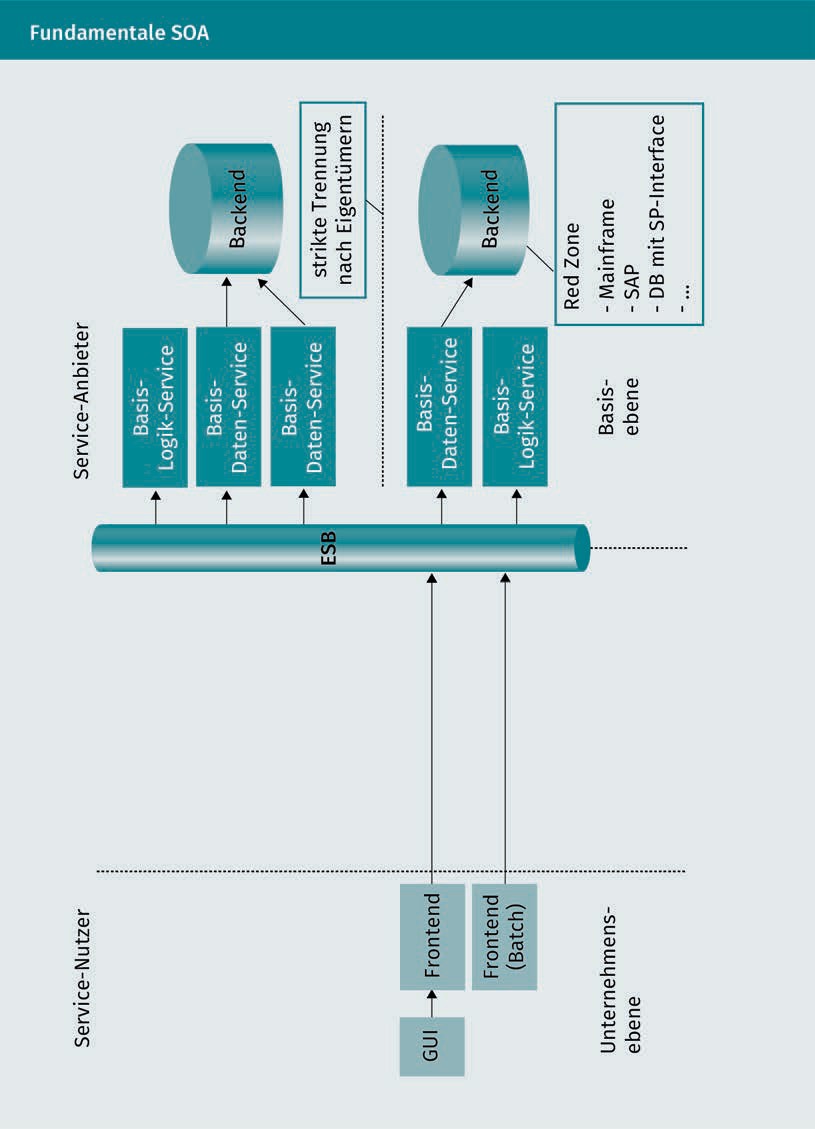
Auf dieser Basis können die Service-Schichten und SOA-Ausbaustufen deﬁniert werden.

Service-Architekturen



SOA-Ausbaustufe 1: fundamentale SOA

Eine Einführung von Business-Services führt zur SOA-Ausbaustufe 1, der sogenannten fundamentalen SOA. Dabei ergibt sich für die Nutzer die Möglichkeit, über einen ESB auf fachliche Schnittstellen von einzelnen Backends zu gehen, was die folgende Graﬁk verdeutlicht (vgl. Josuttis 2008, S. 85).

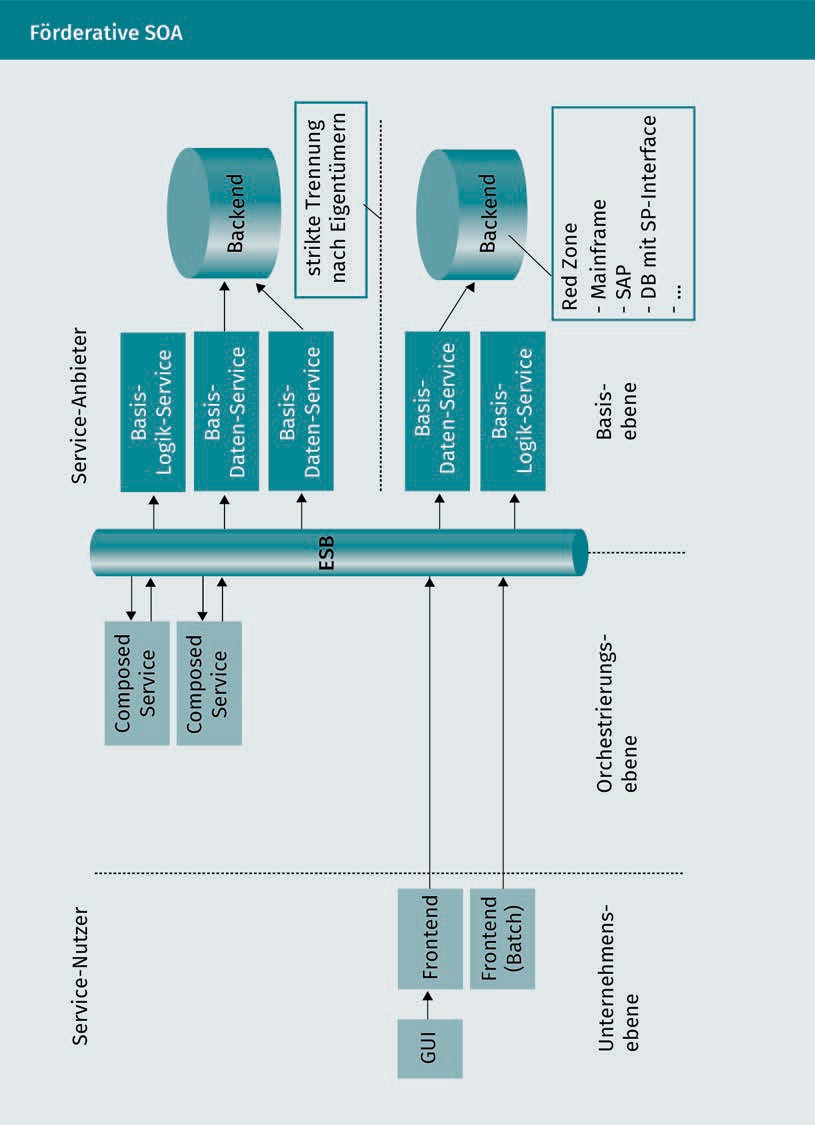


SOA-Ausbaustufe 2: föderierte SOA

Composed Services: SOA-Austauschstufe 2 zeichnet sich dadurch aus, dass Composed Services hinzukommen. Es handelt sich um die erste Kategorie Services, Basis-Services oder Composed Services, welche aus den Services kombiniert werden, die sogenannte Orchestrierung (vgl. Josuttis 2008, S. 87f.).

Service-Architekturen

Föderierte SOA: Über Basis- und Composed Services wird die zweite Ausbaustufe von SOA, die sogenannte föderierte SOA erreicht, auch Netzwerk-SOA genannt. Diese Aus- baustufe enthält eine zusätzliche Sicht für Composed Services, wie die folgende Abbil- dung verdeutlicht (vgl. Josuttis 2008, S. 89).

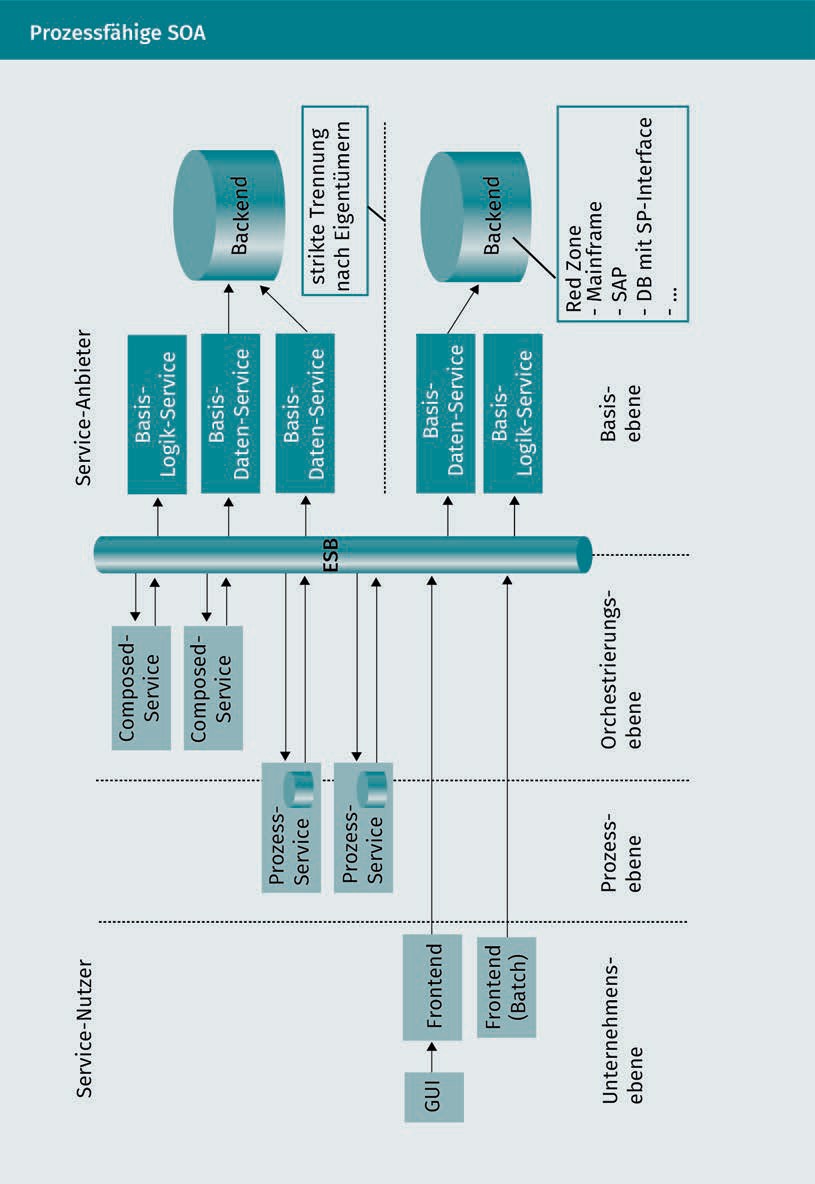


Ein Beispiel für Composed Services ist ein Service, der die Adresse konsistent für alle Backends ändert. Er ruft die jeweiligen Business-Services auf (vgl. Josuttis 2008, S. 89).

SOA-Ausbaustufe 3: prozessfähige SOA

Die Einführung von Prozess-Services führt zur Ausbaustufe 3, der sogenannten prozess- fähigen SOA. Es lassen sich Geschäftsprozesse verwalten, die über mehrere Frontends laufen, was die folgende Graﬁk verdeutlicht (vgl. Josuttis 2008, S. 92f.).

Service-Architekturen



Die Geschäftsprozesse können gegebenenfalls durch menschliche Interaktionen unter- brochen werden.

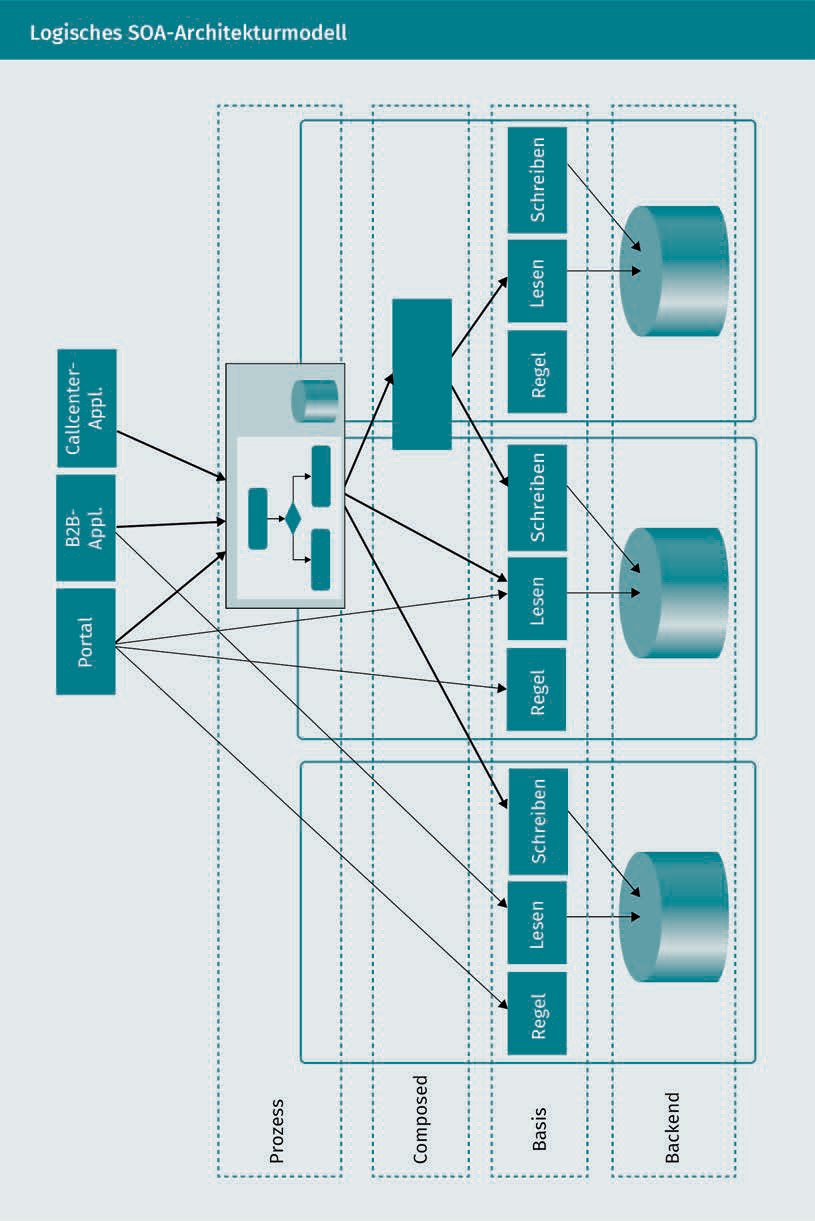
###### SOA-Architekturmodelle

Auf der Basis von SOA bestehen verschiedene Architekturmodelle.

Logische Architekturmodelle

Ein logisches Architekturmodell hebt insbesondere fachliche oder logische Aspekte einer SOA-Landschaft hervor. Dies wird in der folgenden Graﬁk dargestellt (vgl. Josuttis 2008, S. 135f.).

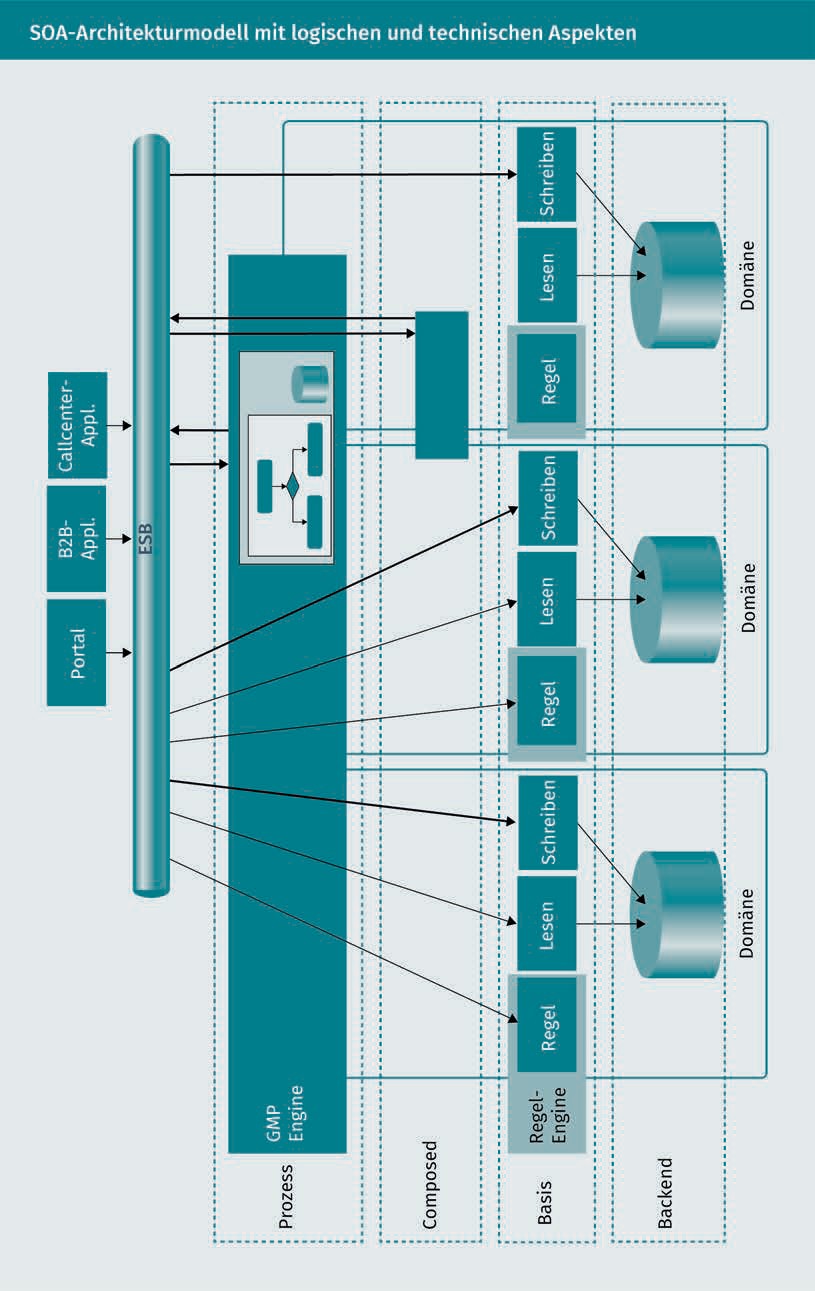
Service-Architekturen



SOA-Architekturmodell mit logischen und technischen Aspekten

Die jeweiligen Zusammenhänge innerhalb des SOA-Architekturmodells mit logischen und technischen Aspekten werden in der nachfolgenden Abbildung verdeutlicht (vgl. Josuttis 2008, S. 137f.).

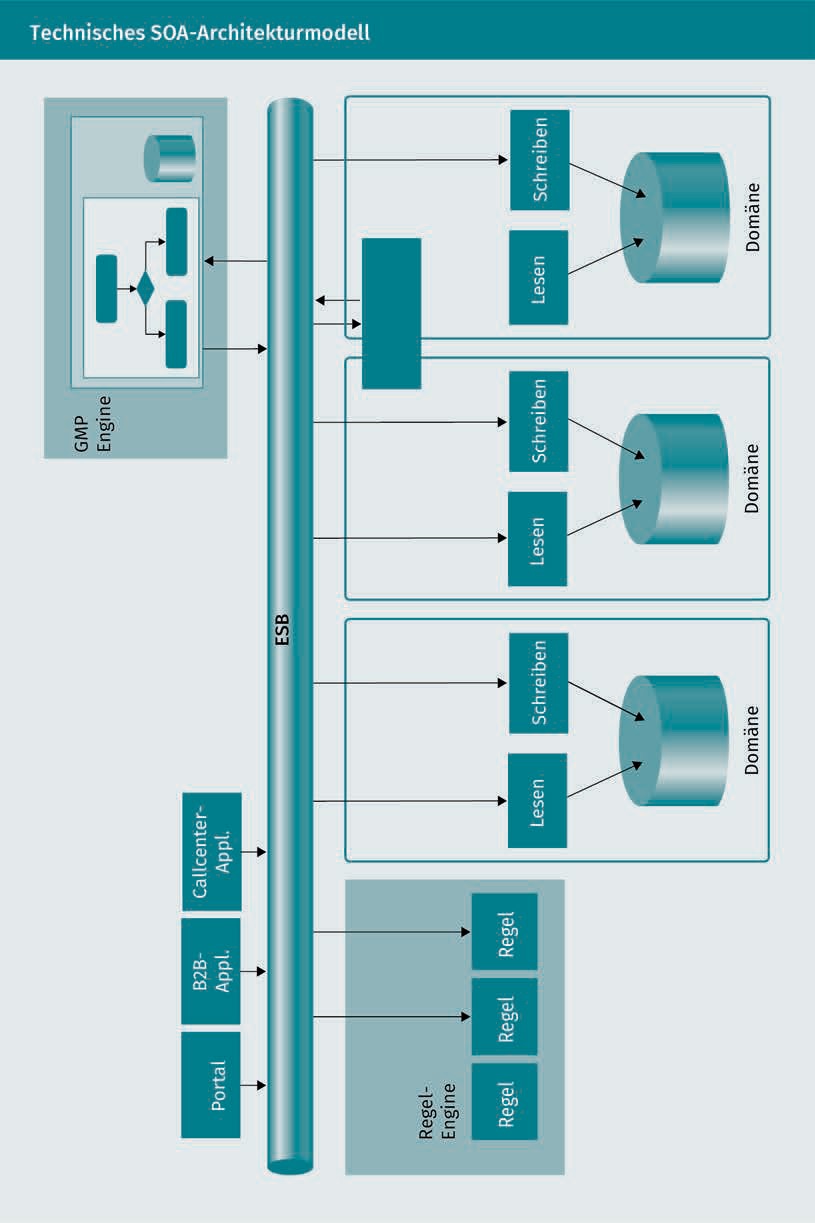
Service-Architekturen



Technisches Architekturmodell

Beim technischen Architekturmodell besteht eine technisch getriebene Sicht, wie in der folgenden Darstellung verdeutlicht wird (vgl. Josuttis 2008, S. 138f.).

Service-Architekturen



Es bestehen verschiedene Modelle für eine geeignete Strukturierung von SOA-basierten Systemlandschaften. Manche setzen mehr auf eine logische Sicht, andere sind vorwie- gend auf technische Aspekte ausgerichtet. Jedoch besteht darin auch ein gewisser Widerspruch, da es gilt, alle Perspektiven zu berücksichtigen. Ein erfolgreiches Manage- ment von SOA-basierten Systemlandschaften betrachtet technische Perspektiven als Implementierungsdetails. Sie sieht diese Landschaften aus organisatorischer Perspek- tive eher im Sinne einer logischen oder domänenorientierten Strukturierung. Bei Pro- zess-Services handelt es sich in der Regel um sogenannte Backend-Services. Diese las- sen keine direkte Kommunikation mit dem Anwender zu. Entsprechend werden für Frontends die Daten für Anwendungen in verschiedenen Backends gesammelt. Diese Daten werden in einen eigenen Workﬂow integriert und erst für die Durchführung bzw. Änderung Prozess-Service aufgerufen. Sie interagieren dann nicht mehr direkt mit dem Anwender (vgl. Josuttis 2008, S. 150). Es wird versucht zu vermeiden, dass Prozess-Ser- vices zu Fehlern führen. Hierzu bietet es sich in der Regel an, Frontenddaten und Einga- ben für typische Geschäftsvorfälle zu validieren (vgl. Josuttis 2008, S. 150).

### Microservices

Microservices Sie gelten als Ansatz zur Modellierung von Software. Dabei lau- fen Module als Pro- gramme bzw. eigene Prozesse ab. Es han- delt sich um ein Modularisierungs- konzept. Microservi- ces ermöglichen eine entkoppelte

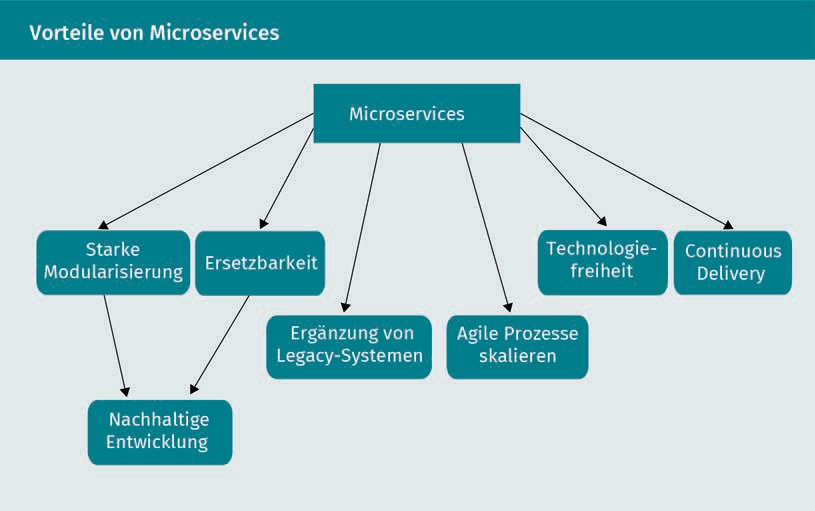
Arbeit.

Bei Microservices handelt es sich um einen Ansatz zur Modellierung von Software. Große Systeme werden in kleine Module unterteilt. Dies erlaubt eine einfachere Erstel- lung und Entwicklung von Software. Microservices nutzen als Module einzelne Pro- gramme. Diese Programme laufen als eigene Prozesse ab. Die Basis des Microservices- Ansatzes bildet die UNIX-Philosophie mit den folgenden Aspekten (vgl. Wolff 2015, S. 2):

* Dass ein Programm lediglich eine Aufgabe erledigen soll und dies gut.
* Dass Programme zusammenarbeiten sollen und können.
* Dass eine universelle Schnittstelle (Textströme) genutzt werden soll.

Der Begriff Microservices lässt sich nicht scharf abgegrenzt deﬁnieren. Er richtet sich jedoch nach den folgenden Kriterien: Es handelt sich um ein Modellierungskonzept, die Services können unabhängig voneinander „deployed“ werden, können in verschiede- nen Technologien implementiert sein, verfügen über einen eigenen Datenhaushalt bzw. sind in einer gemeinsamen Datenbank getrennt, bringen gegebenenfalls eigene Unter- stützungsdienste mit, sind eigenständige Prozesse und müssen über das Netzwerk kommunizieren bzw. nutzen Protokolle, die eine lose Kopplung unterstützen, z. B. REST oder Messaging-Lösungen. Microservices grenzen sich von Deployment-Monolithen ab (vgl. Wolff 2015, S. 2ff.). Die nachfolgende Abbildung führt die wesentlichen Vorteile von Microservices auf.

Service-Architekturen



Die aufgeführten Vorteile sprechen für eine Einführung von Microservices. Neben den Vorteilen sind jedoch auch die nachfolgend genannten Herausforderungen zu berück- sichtigen (vgl. Wolff 2015, S. 6):

* + versteckte Beziehungen,
  + schwieriges Refactoring,
  + hohe Relevanz der fachlichen Architektur,
  + komplexer Betrieb,
  + Komplexität von verteilten Systemen.

Zusammenfassung

Software as a Service (SaaS) versteht sich als ein Vertriebsmodell. Dabei wird die Software von einem Anbieter oder Service Provider gehostet und für Anwender über ein Netzwerk zur Verfügung gestellt.

Unter Everything as a Service (EaaS) versteht man die bedarfsorientierte Bereitstel- lung von IT-Ressourcen.

Serviceorientierte Architektur (SOA) gilt als serviceorientiertes Paradigma in der Softwareentwicklung, insbesondere für Geschäftsprozesse. Es soll wie andere IT- Konzepte zur Verbesserung und Flexibilität beitragen.

Microservices gelten als Ansatz zur Modellierung von Software. Dabei laufen Module als Programme bzw. eigene Prozesse ab. Sie sind ein Modularisierungskonzept. Mic- roservices ermöglichen eine entkoppelte Arbeit.



# Lektion 6

## Integrationsplattformen

#### LERNZIELE

Nach der Bearbeitung dieser Lektion werden Sie …

… die Eigenschaften und den Zweck von Integrationsplattformen kennen und verstehen.

… den Ansatz Enterprise Integration Patterns kennen und anwenden können.

… die Grundlagen der externen Integration mit Zapier, IFTTT & Co. kennen.

DL-D-DLBINGSS01-L06

1. Integrationsplattformen

### Einführung

Integrationsplattformen sind für den Erfolg und die Entwicklung von Smart Services von zentraler Bedeutung. Die Eigenschaften und der Zweck von Integrationsplattformen haben entsprechend weitreichend erfolgsrelevante Effekte. Als hilfreich für die prakti- sche Integration erweisen sich Enterprise Integration Patterns. Für die externe Integra- tion sind u. a. Zapier oder IFTTT zukunftsweisende Ansätze.

### Eigenschaften und Zweck von Integrationsplattformen

###### Enterprise Application Integration (EAI)/Integrationsplattform – Business Bus

Enterprise Applica- tion Integration (EAI) Bei der EAI steht die unternehmensweite Integration von Geschäftsfunktionen im Vordergrund.

Diese sind entlang der Wertschöpfungs- kette über diverse Applikationen ver- teilt auf verschie- dene Plattformen.

Sie sollen in einer

Daten- und Geschäftsprozessin- tegration verbunden

werden.

Bei Enterprise Application Integration (EAI) bzw. der Unternehmensanwendungsinteg- ration (UAI) handelt es sich um einen Ansatz der unternehmensweiten Integration von Geschäftsfunktionen. Die entsprechenden Geschäftsfunktionen sind entlang der Wert- schöpfungskette über diverse Applikationen verteilt auf verschiedene Plattformen. Die Intention besteht darin, dass sie im Sinne einer Daten- und Geschäftsprozessintegra- tion verbunden werden. Dadurch soll eine integrierte Geschäftsabwicklung über ein Netzwerk von unternehmensinternen Applikationen verschiedener Generationen und Architekturen erreicht werden. In diesem Kontext werden diverse, aufeinander aufbau- ende Methoden relevant. Dabei handelt es sich um die Datenintegration bzw. Enter- prise Service Bus (ESB), um die Anwendungsintegration bzw. Message Broker und die Prozessintegration bzw. das Prozessmanagement-Tool (vgl. Steinhaus 2015).

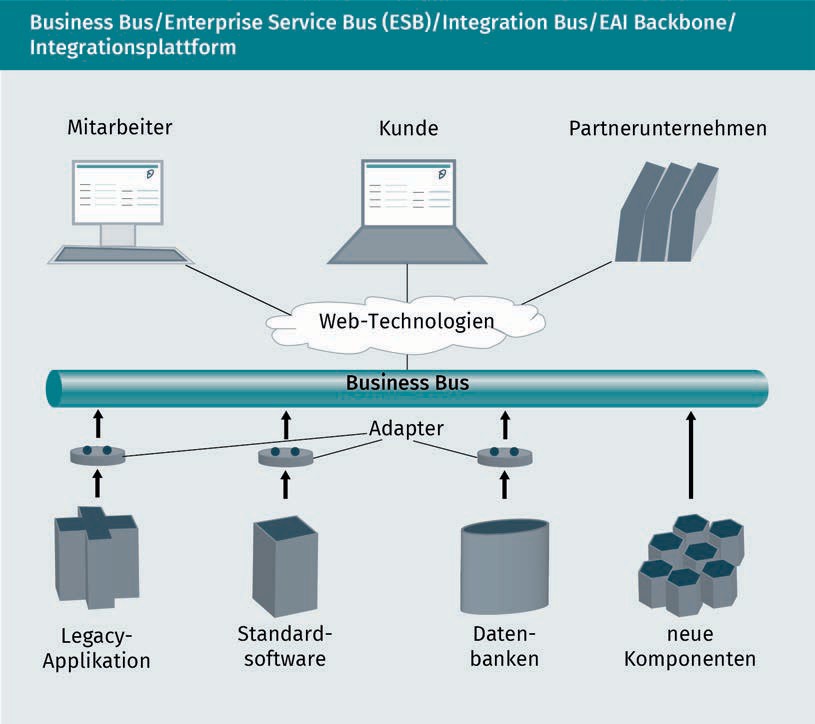
Enterprise Application Integration beinhaltet neben der Planung und den Methoden auch die Software für eine prozessorientierte Integration von heterogenen, autonomen Anwendungssystemen. Dabei lassen sich gegebenenfalls auch externe Anwendungssys- teme berücksichtigen. EAI gilt als eine prozessorientierte Integration verschiedener Anwendungssysteme in heterogene IT-Anwendungsarchitekturen, bei der keine Verän- derung der Implementationen einzelner Geschäftsfunktionen vorgenommen wird. Sämtliche funktionalen Schnittstellen können durch Adapter abstrahiert werden. Dazu erweisen sich Schnittstellenumsetzer als hilfreich (vgl. Hohpe 2002, S. 1–36; Hohpe/ Woolf 2003).

Unter Business Bus versteht man ein Konzept bzw. eine Implementierung sowie ein Kollaborationsprotokoll im Kontext der Applikationsintegration. Ferner handelt es sich auch um Komponenten entlang einer Wertschöpfungskette. Der „Business Bus“ bzw.

„Enterprise Service Bus (ESB)“ oder „EAI Backbone“ trägt auch die Bezeichnung „Integra- tionsplattform“.

Die folgende Graﬁk stellt den Beziehungskontext der Integrationsplattform bzw. Busi- ness Bus dar.

Integrationsplattformen



Auf dem Business Bus bzw. der Integrationsplattform ermöglichen dynamisch ausge- wertete Regeln sowie Prozessbeschreibungen die richtige Weiterleitung der Daten eines Geschäftsfalls. Es ist erforderlich, dass die Daten eines Geschäftsfalls nach korrekter Abfolge zu den einzelnen Funktionen gelangen und auch die Weiterleitung der Ergeb- nisse korrekt erfolgt.

Die Integration von unabhängigen Applikationen erfolgt mit Unterstützung durch die Enterprise Application Integration. Auf einem Business Bus können Ebenen unterschie- den werden. Einerseits besteht die Geschäftsebene. Diese bezieht sich beispielsweise auf Verträge sowie den gesetzlichen wie regulatorischen Rahmen. Zudem existiert die Prozessebene. Hierbei handelt es sich um die Geschäftsprozesse, die entlang einer Wertschöpfungskette erfolgen. Die Applikationsebene bezieht sich auf Geschäftsappli- kationen und Komponenten der Geschäftslösungen. Eine weitere Ebene ist die Softwa- reebene. Diese beinhaltet beispielsweise die Standardsoftware, die Standardkompo- nenten und die Basissoftware, so z. B eine Datenbank oder Middleware. Eine weitere Ebene ist die Systemebene mit Hardware, Betriebssystem und Netzwerk (vgl. Steinhaus 2015).

Die Integration versteht sich als lose gekoppelt bzw. loosely coupled. Sie beinhaltet eine strikte Trennung der Geschäftsprozesslogik von den Geschäftsfunktionen. Zu die- sem Zweck erfolgt eine Aufteilung der einzelnen Geschäftsprozesse in Makro- und Mik-

Integration Es handelt sich um eine Art loser Kopp- lung, die unter ande- rem eine strikte Trennung der Geschäftsprozesslo-

gik von den Geschäftsprozessen beinhaltet. Die Geschäftsprozesse werden in Makro- und Mikroabläufe aufgeteilt. Mikroab- läufe werden teil- weise in die Funktio-

nen integriert.

IT-Architektur Diese bezeichnet alle statischen und

dynamischen Aspekte der IT inner- halb einer Organisa- tion. Durch sie wird die Grundstruktur festgelegt und es werden Regeln deﬁ- niert, die für die Koordination aller Komponenten gel-

ten.

roabläufe. Mikroabläufe werden teilweise in die Funktionen integriert. Vor allem im Zusammenhang mit der Nutzung einer Standardsoftware kann keine Herauslösung erfolgen. EAI erlaubt die Möglichkeit der Abbildung der Geschäftsprozesslogik. Bei zahl- reichen Middleware-Produkten kann dies dagegen über die Integration einer Business Process Engine erfolgen. EAI gilt infolge der prozessorientierten Integration sowohl als eine technische Integrationsplattform wie auch als eine Art Komponente der Integra- tion unter der Organisationsarchitektur (Strukturen und Geschäftsprozesse) und der IT- Architektur des Unternehmens.

Zur praktischen Umsetzung solcher und ähnlicher Aufgabenstellungen wird häuﬁg der Ansatz der EAI herangezogen. Dieser Ansatz stellt eine Selektion von Patterns bzw. eine Art Muster, sogenannte Enterprise Integration Patterns, zur Verfügung (vgl. Hohpe/ Woolf 2003; Hohpe/Woolf 2017).

EAI wird in allen Bereichen der Prozessintegration angewendet, vor allem jedoch im E- Business und in Portalen. EAI gilt als erforderlich für Straight Through Processing.

In Großunternehmen wird durch EAI mit komplexen IT-Landschaften häuﬁg der Einsatz der klassischen Middleware-Produkte verdrängt. EAI ist ein bedeutender Bestandteil der IT-Architektur.

Unter IT-Architektur verstehen sich sämtliche statischen wie auch dynamischen Aspekte der IT innerhalb einer Organisation, so z. B. die Infrastruktur mit Hardware, Standorte, Netzwerke usw. Ein weiterer Bestandteil ist Software, beispielsweise Anwen- dungen. Zudem zählen auch Technologien, Schnittstellen, IT-unterstützte Funktionen und Prozesse dazu. Ein weiterer Bestandteil ist das Architekturmanagement: die Konﬁ- gurations- und Kapazitätsplanung, die Lastverteilung, die Datensicherung, die Verfüg- barkeit, die Ausfallsicherheit oder auch die Planung des Katastrophenfalls. Durch eine IT-Architektur wird die IT in einer Organisation auf zwei Ebenen beschrieben. Einerseits werden Grundstrukturen festgelegt, andererseits werden Regeln deﬁniert, die der Koor- dination des dynamischen Zusammenspiels sämtlicher Komponenten dienen.

###### Beispiel Integrationsplattformen im Bereich Smart Home

Integrationsplattformen, beispielsweise im Bereich Smart Home, lassen sich in ver- schiedene Arten bzw. Formen von Plattformen und Plattformenanbietern unterscheiden (vgl. Arnold 2015):

* reine B2C-Unternehmensplattformen, z. B. Gigaset, eQ-3 Homematic, SMA, devolo und Rademacher,
* weitere B2C-Integrationsplattformen, z. B. Nest, Samsung SmartThings und RWE,
* B2B-Plattform-Provider, z. B. Qivicon, iControls, HomeKit,
* Technologieallianzen, z. B. Thread und AllSeen.

Derzeit vollzieht sich auf dem Markt ein Wettlauf bzw. Wettkampf um die besten Inter- operabilitätskonzepte zur Gewinnung von Plattformpartnern. Beispielsweise besteht im Bereich Smart Home das Ziel, dem Endkunden das intelligente Zusammenspiel einer

Integrationsplattformen

Vielzahl von Endgeräten, Aktoren und Sensoren der unterschiedlichsten Hersteller im Smart Home zu ermöglichen. Dabei zeichnet sich zusammenfassend die folgende Ent- wicklung ab: „Reine B2C Smart Home Plattformen werden zu Auslaufmodellen im Smart Home“ (Arnold 2015).

### Enterprise Integration Patterns

Der Ansatz Enterprise Integration Patterns geht zurück auf das Werk von Gregor Hohpe und Bobby Woolf. Enterprise Integration Patterns beziehen sich auf das Entwerfen, Erstellen und Bereitstellen von Messaging-Lösungen. Sie stammen aus einer Beschrei- bung erfolgreicher Vorgehensweisen bzw. Mustern der Integration. Patterns sind akzep- tierte Lösungen für wiederkehrende Problemstellungen der Integration. Der Ausgangs- punkt des Ansatzes besteht in der Feststellung, dass eine Lücke im Zusammenhang asynchroner Messaging-Systeme besteht (vgl. Hohpe 2002, S. 1–36; Hohpe/Woolf 2003).

Das Standardwerk „Enterprise Integration Patterns“ der beiden bereits genannten Auto- ren beschreibt erfolgreiche Vorgehensweisen der Integration (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017).

###### Probleme der Integration

Das Schaffen von Integrationslösungen gilt als herausfordernde Aufgabe. Ob die Archi- tektur faktisch eine gute Wahl war, lässt sich meist erst nach einigen Monaten feststel- len oder dauert sogar mehrere Jahre. Es existieren keine Rezepte für Unternehmensin- tegrationslösungen. Zwar werden durch Integrationsberater Methoden und Best Practices angeboten, jedoch besteht keine Garantie für den Erfolg (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017).

In der Praxis erfolgen anspruchsvolle Applikationen nicht isoliert. Vielmehr sind Anwendungen oft gegenseitig verknüpft. So ist beispielsweise häuﬁg die Kundenverwal- tung mit der Logistik verbunden oder das Einkaufsmodul erhält Zugriff auf eine Aukti- onsplattform. Gegebenenfalls wird auch ein Smartphone mit dem Unternehmenskalen- der synchronisiert. Werden Anwendungen gegenseitig verbunden, sind sie in der Regel nützlicher. Die Voraussetzung hierfür besteht darin, dass sämtliche Integrationslösun- gen unter anderem die folgenden Herausforderungen bewältigen (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017; Synoa 2018):

* + - Herausforderung der Unzuverlässigkeit von Netzwerken,
    - Herausforderung der Trägheit der Netzwerke,
    - Herausforderung der Verschiedenheit der Applikationen,
    - Herausforderung der Unvermeidbarkeit von Veränderungen.

Herausforderung der Unzuverlässigkeit von Netzwerken

Es ist erforderlich, dass Integrationslösungen Daten von einem Rechner zu einem ande- ren Rechner über Netzwerke übertragen. Jedoch sehen sich – im Vergleich mit einem Prozess innerhalb eines Rechners – verteilte Systeme einer wesentlich umfangreiche- ren Liste von potenziellen Problemen gegenüber. Es ist möglich, dass sich die zu ver- bindenden Rechner auf unterschiedlichen Kontinenten beﬁnden. Die Daten sind ent- sprechend über Telefonleitungen, LANs, Router, Switche, öffentliche Netzwerke oder Satelliten zu übertragen. Dies bedeutet auch, dass jeder dieser Schritte das Risiko von Verzögerungen und/oder Unterbrechungen beinhaltet, was zu einer gewissen Unzuver- lässigkeit der Netzwerke führt (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017; Synoa 2018).

Herausforderung der Trägheit der Netzwerke

Werden Daten über ein Netzwerk versendet bzw. ausgetauscht, dauert dies wesentlich länger. Dagegen kann innerhalb eines Rechners eine Funktion schnell aufgerufen wer- den. Wird ein verteiltes System unter einer lokalen Applikation designt, ist mit negati- ven Wirkungen auf die Performance zu rechnen (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017; Synoa 2018).

Herausforderung der Verschiedenheit der Applikationen

Durch Integrationslösungen werden Daten zwischen Applikationen vermittelt. Diese sind in verschiedenen Programmiersprachen verfasst. Sie laufen auf unterschiedlichen Plattformen. Auch die Dateiformate sind unterschiedlich. Die Herausforderung besteht darin, dass die Integrationslösung die verschiedenen Technologien harmonisiert (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017; Synoa 2018).

Herausforderung der Unvermeidbarkeit von Veränderungen

Mit der Zeit verändern sich Applikationen. Es ist erforderlich, dass eine Integrationslö- sung diese Veränderungen bewältigen kann. Sind Systeme nicht optimal designt, besteht die Gefahr, dass eine Art Schneeballeffekt entsteht und gegebenenfalls eine Änderung eines Systems dazu führt, dass dadurch auch alle anderen Systeme betroffen sind bzw. verändert werden. Aus diesem Grund ist es notwendig, die Abhängigkeiten zwischen den Systemen zu minimieren. Dies wird in der Regel durch eine lose Kopplung der Systeme begünstigt (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017; Synoa 2018).

###### Asynchrone Messaging-Architekturen

Für die Unternehmensintegration haben sich asynchrone Messaging-Architekturen (Nachrichtenübermittlungsarchitekturen) als beste Strategie erwiesen. Diese beinhalten lose Lösungen, die berücksichtigen, dass Beschränkungen der entfernten Kommunika- tion, Latenz und Unzuverlässigkeit bestehen und zu überwinden sind. Aus diesem Grund basieren die meisten EAI-Gefolge und ESBs auf der asynchronen Nachrichten- übermittlung. Jedoch ist die asynchrone Nachrichtenübermittlung anfällig. Eine unab- hängige Designleitung kann Entwickler unterstützen, Fehler zu vermeiden. Auf diese Weise lassen sich stabile, auf die asynchrone Nachrichtenübermittlung gestützte Integ- rationsarchitekturen aufbauen (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017).

Integrationsplattformen

Der Ansatz Enterprise Integration Patterns beinhaltet verschiedene Muster der Unter- nehmensentwicklung, ohne jedoch einen Anspruch auf Vollständigkeit zu erheben. Ein besonderer Fokus wird auf die Lücke in einem synchronen Messaging gelegt. Synchro- nes Messaging bekommt in der Enterprise-Softwareentwicklung und insbesondere im Zusammenhang mit der Integration eine zunehmende Bedeutung. Die Integration ist von besonderer Relevanz, weil Anwendungen nicht isoliert oder unabhängig voneinan- der sind. Vor diesem Hintergrund sind Techniken erforderlich, mit denen verschiedene Anwendungen zusammenarbeiten können. Zusammen können sie zu einem größeren Nutzen führen als Einzelanwendungen. Die Herausforderung im Zusammenhang mit der Messaging-Technologie besteht darin, dass bislang keine effektive Übermittlung von Messaging realisiert wurde. Nachrichten sind von Natur aus asynchron. Es bestehen wesentliche Unterschiede in den Designkonzepten in einer asynchronen Welt (vgl. Hohpe 2002, S. 1–36; Hohpe/Woolf 2003).

###### Unterstützung durch Muster (Patterns)

Bei Mustern bzw. Patterns handelt es sich um eine Möglichkeit, die sich praktisch bewährt hat, um Kenntnisse von Experten zu gewinnen. Dies ist vor allem dann hilf- reich, wenn keine Lösung gefunden wird. Durch jedes Muster bzw. Pattern wird ein spe- ziﬁsches Problem angesprochen. Bei der Lösung handelt es sich nicht um den ersten möglichen Lösungsansatz, der erwogen werden könnte. Der Fokus richtet sich auf die Lösung, die sich im Laufe der Zeit in der Praxis bewährt hat. Eine solche Lösung bein- haltet die Erfahrungen, die berufserfahrene Entwickler und Architekten mit der Zeit gewinnen konnten. Sie konnten die Lösungen wiederholt verwenden und aus den Erfahrungen und Fehlern lernen. Auf diese Weise lassen sich immer bessere Lösungen ﬁnden und weiterentwickeln (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017).

Der von Hohpe und Woolf erarbeitete und in ihrem Werk „Enterprise Integration Pat- terns“ vorgestellte Musterkatalog beschreibt 65 Integrationsmuster, die aus zahlreichen Integrationsprojekten seit 2002 gesammelt wurden (vgl. Hohpe/Woolf 2003). Durch die Patterns wird eine technologieunabhängige Designleitung für Entwickler und Architek- ten zur Verfügung gestellt. Diese erlaubt die Beschreibung und Entwicklung von stabi- len Integrationslösungen (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017).

###### Zweck der Patterns

Die Unternehmensintegration ist kompliziert. Sie lässt sich nicht mit einfachen Rezep- ten oder Standardlösungen bewältigen. Es ist möglich, dass Patterns die Leitung durch das Dokumentieren gesammelter Erfahrung, Denkweisen und Lösungen zur Verfügung stellen, die ansonsten in der Regel nur Experten bzw. Architekten hervorbringen kön- nen. Bei den Patterns handelt es sich um akzeptierte Lösungen von wiederkehrenden Problemen innerhalb eines gegebenen Zusammenhangs. Die Patterns sind ausreichend abstrakt, um mit den meisten Integrationstechnologien zu harmonieren, jedoch spezi- ﬁsch genug, um gängige Muster zu entwerfen und diese den Entwicklern zur Verfügung

zu stellen. Patterns bieten zugleich eine Art Vokabular bzw. eine (symbolische) Fach- sprache für Entwickler, welche eine efﬁziente Beschreibung ihrer Lösung ermöglicht (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017).

Die heterogenen Systemlandschaften stellen Entwickler und Architekten zunehmend vor komplexe Herausforderungen, die es zu lösen gilt. Dies gilt insbesondere im Kontext der Integration von Anwendungen. Mit der Zeit können komplexe Systeme entstehen. Diese sind in der Regel sehr eng an die angebundenen Schnittstellen gekoppelt. Jedoch kann auch im Falle individueller Schnittstellen auf vorhandene Lösungen und Tools zurückgegriffen werden. Für die Einführung in Enterprise Integration Patterns ist die Berücksichtigung oft auftretender Problemstellungen der Integration von Enterprise- Applikationen in heterogene Umgebungen hilfreich. Hierzu lassen sich geeignete Lösungskonzepte herausarbeiten und nutzen (vgl. Innoq 2018).

###### Das Lösen der Herausforderungen (Lösungsansätze)

Im Laufe der Jahre haben sich hinsichtlich der Integration vier grundlegende Ansätze herauskristallisiert. Die folgenden Lösungsansätze erweisen sich zur Bewältigung der Herausforderungen als hilfreich (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017; Synoa 2018):

* Lösungsansatz Dateien,
* Lösungsansatz gemeinsame Datenbank,
* Lösungsansatz Remote Procedure Calls,
* Lösungsansatz Messaging.

Lösungsansatz Dateien

Bei diesem Lösungsansatz werden durch eine Anwendung Informationen in eine Datei geschrieben. Diese Datei wird durch eine andere Applikation gelesen. Hierzu ist es erforderlich, dass durch die Applikationen folgende Elemente miteinander vereinbart werden: Name der Datei, Speicherort der Datei, Dateiformat, Zeitplan für das Schreiben sowie für das Lesen der Datei und die Verantwortlichkeit, um die Datei gegebenenfalls zu löschen oder zu verschieben (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017; Synoa 2018).

Lösungsansatz gemeinsame Datenbank

Der Lösungsansatz einer gemeinsamen Datenbank basiert darauf, dass sich Applikatio- nen eine Datenbank nach einem abgestimmten Schema teilen. Physisch existieren die Daten nur ein Mal. Es ist nicht erforderlich, diese von einer Anwendung auf die andere zu übertragen (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017; Synoa 2018).

Lösungsansatz Remote Procedure Calls

Der Lösungsansatz Remote Procedure Calls geht davon aus, dass eine Applikation den Zugriff auf eine Funktion bietet, welche von einer anderen, externen Applikation aufge- rufen werden kann. In diesem Kontext erfolgt die Kommunikation synchron sowie in Echtzeit. Zu diesem Grundprinzip gehören u. a. sämtliche serviceorientierten Konzepte (SOA) und Microservices (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017; Synoa 2018).

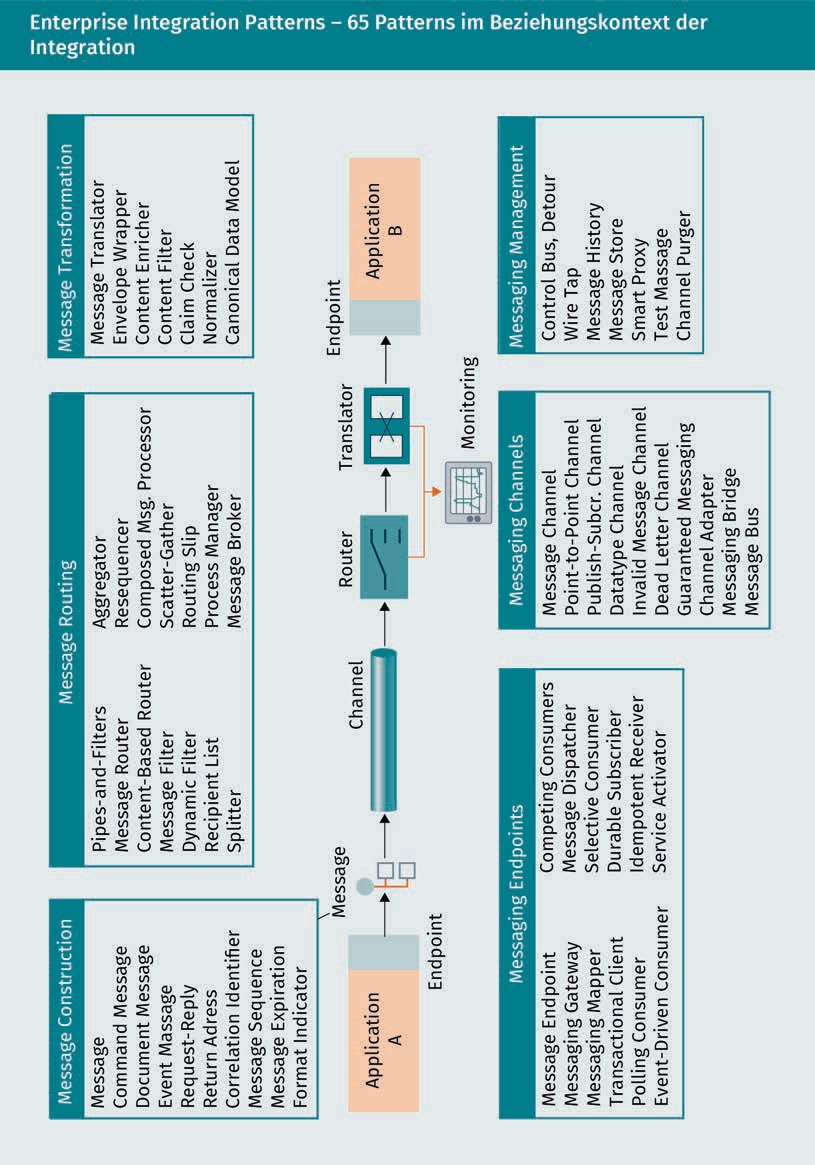
Integrationsplattformen

Lösungsansatz Messaging

Unter dem Lösungsansatz Messaging erzeugt eine Applikation eine Nachricht in einem gewöhnlichen Nachrichtenkanal. Durch eine andere Applikation können Nachrichten aus diesem Kanal gelesen werden, was auch zu einem späteren Zeitpunkt noch möglich ist. Die Voraussetzung hierfür besteht darin, dass sich die Applikationen auf den Nach- richtenkanal sowie das Nachrichtenformat einigen. Es erfolgt eine asynchrone Kommu- nikation (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017; Synoa 2018).

###### Die Patterns

Insgesamt wurden 65 Patterns (Nachrichtenübermittlungsmuster) herausgearbeitet und dokumentiert. Diese sind in der nachfolgenden Abbildung im Beziehungskontext der Integration aufgeführt (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017).



Die Patterns lassen sich in die folgenden Gruppen unterteilen (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017):

Integrationsplattformen

* Integration Styles beinhalten verschiedene Arten der Anwendung zur Integration. Es handelt sich um eine historische Sammlung von Integrationstechnologien, welche zur Verfügung gestellt werden. Sämtliche nachfolgenden Patterns folgen dem Mes- saging Style.
* Channel Patterns beschreiben, wie Nachrichten über einen Channel transportiert werden. Diese Patterns werden durch die meisten kommerziellen Open Source Mes- saging Systems durchgeführt.
* Routing Patterns thematisieren, wie Nachrichten von einem Absender zum richtigen Empfänger geroutet werden. Routing Patterns benötigen eine Nachricht von einem Kanal und veröffentlichen diese bzw. leiten diese in der Regel modiﬁkationsfrei zu einem anderen Kanal. Die Routing Patterns präsentieren Muster in Form von Spezia- lisierungen der Message Patterns.
* Transformation Patterns ändern den Inhalt einer Nachricht, um z. B. verschiedene Datenformate anzupassen, die durch das Senden und das Empfangssystem zur Anwendung kommen. Daten können hinzugefügt und weggenommen werden oder es lassen sich vorhandene umordnen. Das Grundmuster für diese Abteilung ist der Message Translator.
* Endpoint Patterns beschreiben, wie Nachrichtenübermittlungssystemkunden Nach- richten erzeugen oder verbrauchen.
* System Management Patterns beschreiben die Werkzeuge, die geeignet sind, ein komplexes Messaging-System am Laufen zu halten. Dies beinhaltet auch den Umgang mit Fehlerbedingungen, Leistungsengpässen und Änderungen in den Teil- systemen (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017).

###### Produktwerkzeug oder Nutzung von Enterprise Integration Patterns?

Die Patterns sind nicht an eine speziﬁsche Durchführung gebunden. Sie sind hilfreich, damit im Design bessere Lösungen hervorgebracht werden, sofern einige der folgenden Plattformen verwendet werden (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017):

* EAI- und SOA-Plattformen, wie IBM WebSphere MQ, TIBCO, Vitria, Orakeldienstbus, WebMethods (jetzt Software AG), Microsoft BizTalk oder Fiorano,
* offene Quell-ESBs wie Maulesel ESB, JBoss Sicherung, Öffnet ESB, WSo2, Frühlingsin- tegration oder Talend ESB,
* Message Brokers wie ActiveMQ, Apache Kafka oder RabbitMQ,
* Webdienst- oder RESTBASIERTE Integration, einschließlich Amazons Simple Queue Service (SQS) oder Google Cloud,
* JMS-basierte Message-Systeme,
* Technologien von Microsoft wie MSMQ oder Windows Communication Foundation (WCF).

Für die Anwendung gilt es, die Theorie in der Praxis erfolgreich umzusetzen. Hierzu ist beispielsweise die Java-basierte Auftragsverwaltung eines Dienstleistungsunterneh- mens hilfreich, die verschiedene Schnittstellen unterschiedlicher Geschäftspartner anbindet. Es lassen sich Implementierungen der Patterns mithilfe der Java-EE-7-APIs

wie JMS 2.0 und JAX-RS 2.0 den Ansätzen von Spring Integration und Apache Camel gegenüberstellen und die Vor- und Nachteile der konkreten Lösungen herausarbeiten (vgl. Innoq 2018).

### Externe Integration mit Zapier, IFTTT & Co.

###### Kombination von Integrationsansätzen (passend zum Use Case)

Jeder Integrationsansatz ist mit Vor- und Nachteilen verbunden. Vor diesem Hinter- grund werden Kombinationen von Integrationsansätzen praktiziert, die zum jeweiligen Use Case passen. In Integrationslösungen (z. B. für KMU) ﬁnden entsprechend Kombina- tionen von Integrationsansätzen Anwendung. Es werden sämtliche Ansätze eingesetzt (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017; Synoa 2018).

Die Lösungsansätze werden anhand des Beispiels des Legacy ERP Systems verdeutlicht:

* + - Bsp. Lösungsansatz-Dateien: Durch das Legacy ERP System wird zum Datenaus- tausch der Ex- und Import von CSV-Dateien für Artikelinformationen angeboten. Die entsprechenden Informationen werden im ESB ausgelesen, transformiert sowie in eine zentrale Datenbank geschrieben (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017; Synoa 2018).
    - Bsp. Lösungsansatz Gemeinsame Datenbank: Es ist möglich, dass ein anderer Ser- vice auf die Daten zugreift. Er konvertiert diese Daten für einen Artikelimport in einen Webshop (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017; Synoa 2018).
    - Bsp. Lösungsansatz Remote Procedure Calls: Zudem besteht innerhalb des ESB ein REST-Service. Dieser gibt zu einer Artikelnummer einen aktuellen Preis und einen Lagerbestand zurück (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017; Synoa 2018).
    - Bsp. Lösungsansatz Messaging: Ergänzend erfolgt eine Übertragung der aus den Artikeldaten und den Preisen erzeugten Preislisten im Excel-Format und per (S)FTP auf einen Server. Der Übertragungsauftrag wird über eine Nachricht realisiert. Der Prozess muss nicht darauf warten, bis die Übertragung abgeschlossen ist. Er kann darauf setzen, dass das Messaging-System die Übertragung so lange versucht, bis diese erfolgreich gewesen ist (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017; Synoa 2018).

Ein idealer Integrationsansatz existiert nicht. Es lassen sich lediglich zu Unternehmen passende Designs anwenden. Warum sollte man Aufwand und Performance für aus- schließlich synchrone Services investieren, manche Prozesse erfordern keine synchrone Kommunikation. Vor allem im Mittelstand sind oft auch pragmatische Lösungen sinn- voll. In einem KMU ist es nicht erforderlich, sich nach einer Konzernvorgabe auszurich- ten. Bei einer solchen zentralen Vorgabe besteht das Risiko, dass sie im Anwendungs- fall unzutreffend ist bzw. nicht passt oder zu höheren Kosten führt (vgl. Hohpe/Woolf 2003; Hohpe/Woolf 2017; Synoa 2018).

Integrationsplattformen

###### Externe Integration mit Zapier

Durch die Automatisierung lassen sich Arbeitsprozesse von Unternehmen wesentlich erleichtern. Sie vereinfacht z. B. im Marketing das tägliche Geschäft unter anderem durch automatisierte E-Mail-Kampagnen, Social-Media-Aktivitäten oder Lead Nurturing auf der eigenen Website (vgl. Viewneo 2017).

Was vor einigen Jahren lediglich mit einem hohen technischen Aufwand an Lösungen realisierbar gewesen ist, wird heute durch Plattformen wie Zapier und IFTTT für jeder- mann zugänglich, z. B. für das Smart Home. Auf diese Weise ist die gegenseitige Ver- knüpfung von Hunderten von Apps ohne Mühe möglich. Auch die Automatisierung im Bereich Digital Signage stellt eine oft aufgegriffene Thematik dar. Die Wartung von Con- tents ist mit hohem Aufwand verbunden. Sie bedarf Manpower. Auch ein manuelles Anpassen der Inhalte ist erforderlich. Diesbezüglich sind einfachere Wege gefragt. Es ergeben sich z. B. vielseitige Möglichkeiten, wenn Playlists dynamisch angepasst wer- den und Werbebotschaften dadurch die maximale Wirkung erhalten. Die Playlist passt sich der aktuellen Situation an: „Wenn es heute regnet, dann spiele die Playlist ‚Regen- schirm Angebote‘ ab“ (Viewneo 2017).

Auch wenn zuvor eine derartige Automatisierung nur für große, kostenaufwendige Pro- jekte möglich war, werden die Integration mit Zapier, z. B. mit Digital Signage Software, und alle Möglichkeiten für jedermann zugänglich (vgl. Viewneo 2017).

Bei Zapier handelt es sich um eine externe Plattform, die es dem Nutzer ermöglicht sein Lodgify-Konto mit über 1.000 externen Applikationen zu verknüpfen. Aufgrund die- ser Verknüpfung ist es möglich, einfach, schnell und automatisch Daten aus Lodgify- Buchungen mit anderen Plattformen zu teilen. Auf diese Weise lassen sich Arbeitsab- läufe optimieren (vgl. Gross/Lodgify 2017).

Zapier ist ein Dienst, der es dem Nutzer erlaubt, zwei (oder mehrere) ansonsten unver- bundene Dienstleistungen über seine APIs (oder Protokolle) zu nutzen bzw. zu verbin- den. Damit eröffnen sich beispielsweise die folgenden Möglichkeiten: Schaffen einer Trello-Aufgabe von einem Evernote-Zeichen, das Veröffentlichen eines RSS-Artikels über Facebook, neue E-Mails an einem Spreadsheet anhängen, E-Mail-Kapazitäten durch Nutzung von Dropbox usw. (vgl. Bozhanov 2018). Die auf diese Weise automati- sierten Workﬂows zwischen den verschiedenen Apps werden „Zaps“ genannt.

###### Externe Integration mit IFTTT

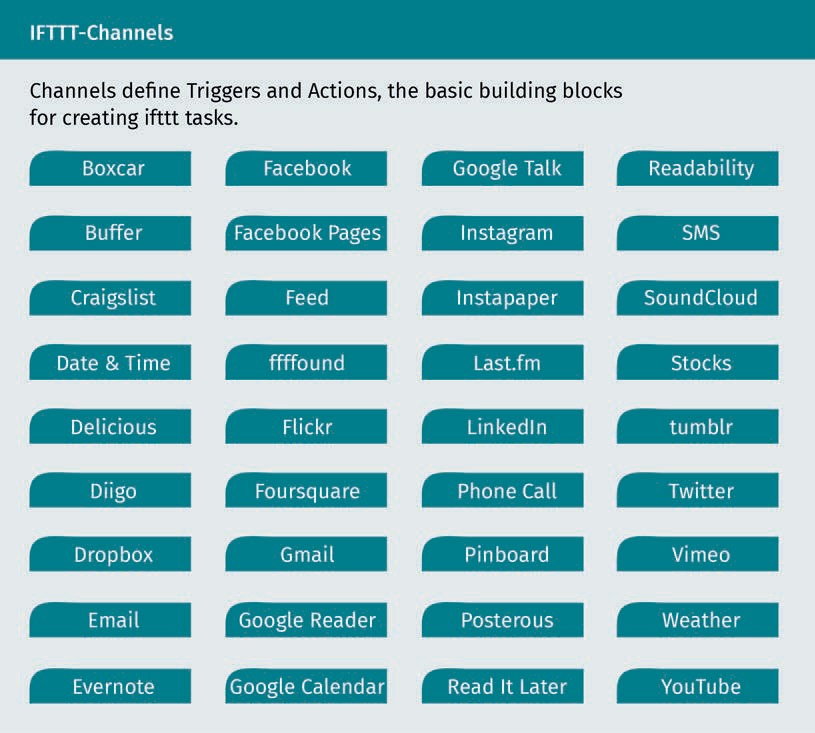
Auch wenn die Technik schon weit fortgeschritten ist, sind für das moderne Smart Home noch verschiedene Probleme zu lösen, so z. B. die Vernetzung von einzelnen Software- und Hardwarekomponenten. Wenn beispielsweise die Beleuchtung nicht zu den Sensoren passt oder die Sensoren nicht zur Heizung passen, besteht das Risiko, dass es zu einem Bruch im Smart Home kommt. Ein solcher Bruch kann zu negativen Folgen für den Komfort sowie für die intelligente Funktionsweise führen. Zur Behebung von Problemen dieser Art ist IFTTT nützlich (vgl. IFTTT 2018; Schiller 2016).

Im Jahr 2010 wurden IFTTT in San Francisco gegründet. IFTT heißt: „if this then that“ bzw. in der Übersetzung: „wenn dies, dann das“. IFTT wird ausgesprochen wie „Lift“, nur ohne

„L“ (vgl. IFTTT 2018).

IFTTT funktioniert nach dem Grundprinzip der Umsetzung einer Aktivität, welche ereig- nisgesteuert angetriggert ist. Für den Fall, dass ein ganz bestimmtes bzw. vorbestimm- tes oder deﬁniertes Ereignis eintritt, wird eine zuvor fesstgelegte Aufgabe durchgeführt. Die Aufgabe muss zuvor exakt deﬁniert worden sein. Als Ereigniserzeuger oder Verar- beiter einer Aktivität zählen bspw. diverse Webdienste, welche IFTTT in Form von sog. Channels anbieten, so z. B. Facebook, YouTube, Instagram, Evernote usw. (vgl. Allmich 2012).

Die nachstehende Graﬁk nennt Ereigniserzeuger und Verarbeiter einer Aktivität bzw. Webdienste und Channels, die IFTTT anbietet.



Von jedem der Webdienste werden in der Regel die unterschiedlichsten APIs (Funkti- onsaufrufe) angeboten, die die Entwickler nutzen können. Es ist möglich, über diese APIs Daten mit den jeweiligen Webdiensten auszutauschen. Durch IFTTT werden diese Services für die Ermittlung bestimmter Ereignisse bzw. zum Anstoß von Aktivitäten genutzt. Durch IFTTT können nur solche Ereignisse und Aktivitäten angeboten werden,

Integrationsplattformen

die von Herstellern der Webdienste bereitgestellt werden. IFTTT fragt die Ereignisse in kurzen Taktphasen ab. Zwar ist damit noch keine Echtzeitverarbeitung möglich, jedoch kann mit dem praktizierten Rhythmus schon gut operiert werden (vgl. Allmich 2012).

IFTTT wird vor dem Hintergrund stärker beanspruchter Funktionen der modernen Geräte und Applikationen immer bedeutender. IFTTT zeichnet sich durch eine einfache Funktionsweise aus. Auf der IFTTT-Webseite lassen sich sogenannte Rezepte ﬁnden. Dabei handelt es sich um eingestellte Funktionsabläufe. Beispiel: Auf welche Weise könnte ein Gerät A mit Gerät B oder ein Gerät C mit Software Y kommunizieren (vgl. IFTTT 2018; Schiller 2016)?

„IFTTT is the free way to get all your apps and devices talking to each other. Not every- thing on the internet plays nice, so we’re on a mission to build a more connected world” (IFTTT 2018).

Ideen hierfür werden von den Usern selbst geliefert. Denn IFTTT bedeutet: „Wenn das [passiert], dann [löse] das [aus]“ (vgl. Schiller 2016). In diesem Sinne kombinieren User verschiedene Komponenten miteinander.

Beispielsweise könnte sich ein User wünschen, dass jedes Mal, wenn die Wohnung betreten und dies anhand eines Motion Sensors ermittelt wird, ein Facebook-Post „Ich bin jetzt daheim“ verschickt werden soll.

Zusammenfassung

Bei Enterprise Application Integration (EAI) bzw. der Unternehmensanwendungsin- tegration (UAI) steht die unternehmensweite Integration von Geschäftsfunktionen im Vordergrund. Diese sind entlang der Wertschöpfungskette über diverse Applika- tionen verteilt auf unterschiedliche Plattformen und sollen in einer Daten- und Geschäftsprozessintegration verbunden werden.

Der Business Bus bzw. Enterprise Service Bus (ESB) oder EAI Backbone trägt auch die Bezeichnung Integrationsplattform.

Unter Integration wird eine Art lose Kopplung verstanden. Diese beinhaltet unter anderem eine strikte Trennung der Geschäftsprozesslogik von den Geschäftsprozes- sen. Die Geschäftsprozesse werden in Makro- und Mikroabläufe aufgeteilt. Mikroab- läufe werden teilweise in die Funktionen integriert.

Als IT-Architektur werden alle statischen und dynamischen Aspekte der IT innerhalb einer Organisation verstanden. Durch sie wird die Grundstruktur festgelegt und es werden Regeln deﬁniert, die für die Koordination aller Komponenten gelten.

IFTTT ist die bekannteste Plattform. Sie ist vor allem bei Privatanwendern populär. Diese kostenlose Plattform bietet eine Vielzahl an Integrationen an und gewährt den Nutzern Zugriff auf zahlreiche bekannte Cloud-Dienste für private und Busi- ness-Dienste. Zapier ist ﬂexibel wie IFTTT. Es richtet den Fokus auf große SaaS- Anbieter für Business-Kunden.

IFTTT bedeutet: „If this then that.“ Es funktioniert entsprechend dem Prinzip der Umsetzung einer Aktivität, die ereignisgesteuert angetriggert wird. Dies bedeutet, dass, wenn ein bestimmtes Ereignis eintritt, die Durchführung einer zuvor deﬁnier- ten Aufgabe erfolgt. Ereigniserzeuger und Verarbeiter einer Aktivität sind beispiels- weise verschiedene Webdienste, die IFTTT als Channels anbietet. Enterprise Integra- tion Patterns beziehen sich auf das Entwerfen, Erstellen und Bereitstellen von Messaging-Lösungen. Sie stammen aus einer Beschreibung erfolgreicher Vorge- hensweisen bzw. Mustern der Integration.

Patterns sind akzeptierte Lösungen für wiederkehrende Problemstellungen der Integration.

Zapier ist eine externe Plattform. Sie ermöglicht es dem Nutzer, sein Konto mit über

1.000 externen Applikationen zu verknüpfen und zwei oder auch mehrere Dienst- leistungen über seine APIs oder Protokolle zu nutzen bzw. zu verbinden. Zapier ist ein System, das die Integration zwischen einer breiten Auswahl an Applikationen und Cloud-basierten Services ermöglicht.



# Lektion 7

## Technologien für Smart Services

#### LERNZIELE

Nach der Bearbeitung dieser Lektion …

… kennen Sie die Formate JSON und XML für den Datenaustausch und können beide vergleichend beurteilen und anwenden.

… kennen Sie die wesentlichen Internetkommunikationsprotokolle.

… sind Sie über die Möglichkeiten der semantischen Beschreibungen informiert.

… kennen Sie den Ansatz des Complex Event Processing (CEP).

… sind Sie mit den Aspekten des Datenschutzes und der Sicherheit vertraut.

DL-D-DLBINGSS01-L07

1. Technologien für Smart Services

### Einführung

Für Smart Services ist der Einsatz von geeigneten Techniken erforderlich. Hierzu sind die relevanten Aspekte zu berücksichtigen, so insbesondere die bestehenden Formate für den Datenaustausch, Internetkommunikationsprotokolle, semantische Beschreibun- gen, Complex Event Processing (CEP) sowie die Gewährleistung von Datenschutz und Sicherheit.

### Formate für den Datenaustausch

In den vergangenen Jahren hat sich für Web-APIs und Konﬁgurationen eine Verdrän- gung von XML durch einfachere Formate, z. B. JSON oder YAML, vollzogen (vgl. Kesha- varzi/Bayer 2011). Beispiel: Twitter hat seine Unterstützung für XML in den eigenen Streaming-APIs eingestellt.

###### JSON

JSON (JavaScript Object Notation) Bei JSON handelt es sich um ein Textfor- mat für den Aus- tausch von Daten. JSON ist mit einer Skriptsprache ver- bunden und als Datenformat von

JavaScript abgeleitet. Es erfüllt jedoch auch ähnliche Aufga-

ben wie XML.

JSON (JavaScript Object Notation) ist ein Textformat für den Austausch von Daten (vgl. Karlstetter 2018). Es gilt als einfacher Standard zur strukturierten Kodierung von Daten in Form von menschenlesbarem Text. Hierdurch ergeben sich Vorteile der automatisier- ten Weiterverarbeitung und besseren Zugänglichkeit einer manuellen Inspektion oder Überarbeitung (vgl. Augsten 2018). JavaScript Object Notation ist ein Format für die Seri- alisierung von Objekten. Das Format wurde durch Douglas Crockford speziﬁziert. Zur Notation fand die Syntax von JavaScript Verwendung. Ein JSON-Dokument ist damit auch ein ausführbares JavaScript (vgl. Keshavarzi/Bayer 2011). Der Name JavaScript Object Notation deutet auf eine Verbindung von JSON und der Skriptsprache hin. Die Skriptsprache ist seit dem Jahr 1997 als ECMAScript standardisiert. JSON ist als Daten- format von JavaScript abgeleitet. Jedoch beschränkt sich die Anwendung nicht alleine auf die Skriptsprache. Durch JSON werden ähnliche Aufgaben erfüllt wie mit XML. Mit JSON werden Daten strukturiert und in menschen- wie maschinenlesbarer Form gespei- chert. Zu diesem Zweck ﬁnden Name-Wert-Paare und eine Formatierung mit geschweif- ten Klammern Anwendung. Das Format ermöglicht eine Serialisierung von programmin- ternen Datenstrukturen zur Datenkommunikation und Auslagerung auf Permanentspeicher. Als wesentlicher Vorteil von JSON gilt dessen Einfachheit der Implementation sowie der Anwendung. Eine weitere Alternative dazu ist YAML (vgl. Augsten 2018). JavaScript Object Notation ermöglicht die menschen- und maschinen- lesbare Speicherung und Übertragung von Daten. Als ähnliche Ansätze lassen sich z. B. YAML oder XML nennen (vgl. Augsten 2018).

Vorteile und Schwächen von JSON

Vorzüge der JavaScript Object Notation sind (vgl. Augsten 2018):

Technologien für Smart Services

* die Einfachheit der Implementation und Anwendung,
* eine einfache Einbindung in JavaScript mit „JSON.parse()“, ab ECMAScript-Version 5.

Vorteile von JSON sind (vgl. Keshavarzi/Bayer 2011; Müller 2007):

* Es verfügt über eine übersichtliche Notation.
* Es hat einen geringen Overhead.
* Es liefert gute Unterstützung durch eine Vielzahl von Programmiersprachen.
* Es bietet für einen reinen Datenaustausch beinahe sämtliche Vorteile von XML.
* Es weist eine kompakte, auf das Minimum reduzierte Syntax auf. Dadurch ist das übertragene Datenvolumen im Vergleich zu XML geringer.
* Es handelt sich um ein reines JavaScript. Damit entfallen lange Parsing-Vorgänge.
* Die serverseitige Verarbeitung der JSON-Objekte ist relativ einfach.

Nachteile/Schwächen von JSON (vgl. Müller 2007; Augsten 2018):

* Die Syntax wird häuﬁg als „gewöhnungsbedürftig“ bezeichnet.
* Es bestehen keine optimalen Möglichkeiten für die Integration von Metadaten oder Kommentaren.
* Es war lange Zeit nicht weit verbreitet (im Unterschied zu XML).
* Es hat lange Zeit keine breitere Akzeptanz erfahren.
* Es geht einher mit einer unscharfen Zahlendeﬁnition.
* Es gibt keine universellen Standards für die Konvertierung nicht unterstützter JavaScript-Datentypen.

###### YAML

YAML Ain’t Markup Language (YAML) wurde als ein Format für die Konﬁguration nach Ruby on Rails bekannt. Es handelt sich um eine Sprache für den Datenaustausch. Im Unterschied zu XML gilt diese Sprache als leicht zu lesen und auch einfach zu editieren. Die entsprechenden Vorteile ergaben sich vor allem infolge eines Verzichtes auf Klam- mern, Anführungszeichen oder Tags. Zudem werden durch YAML Datentypen und Type Casting unterstützt. YAML setzt zur Darstellung von hierarchischen Strukturen Einrü- ckungen ein. Auf diese Weise entsteht im Vergleich zu JSON eine kompaktere und über- sichtlichere Darstellung (vgl. Keshavarzi/Bayer 2011). Dies wird durch das Beispiel Lis- ting 2 in der folgenden Abbildung verdeutlicht.



###### XML

XML hat sich stark verbreitet. Entsprechend existieren Tausende von Markup-Sprachen, die auf XML basieren. Heute ﬁndet sich XML in beinahe allen Anwendungsbereichen. Eine Ausnahme stellt der Bereich des Webdesigns dar. Dort wird das Format oft gemie- den. IE-Namespace-Deklarationen in XML-Dokumenten werden oft ausgeﬁltert und nicht angezeigt (vgl. Keshavarzi/Bayer 2011).

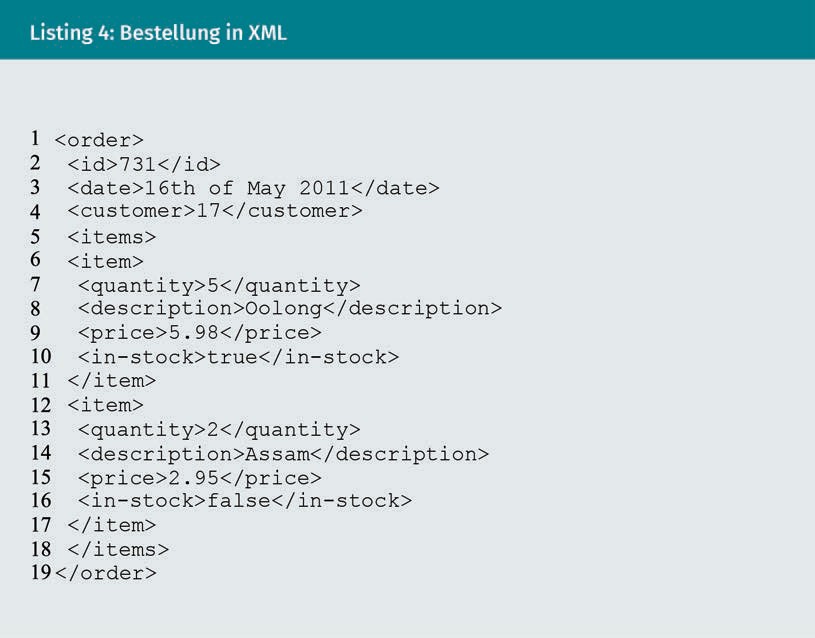
Durch XML wird kein fester Satz von Auszeichnungselementen wie beispielsweise bei HTML vorgeschrieben. XML ist ein Datenaustauschformat, mit dem es möglich ist, eine eigene Auszeichnungssprache mit beliebigen Tags zu deﬁnieren. Es ist nicht erforder- lich, Allzweck-Tags z. B. bei einem Absatz zu verwenden. Der Programmierer kann selbst entscheiden, welche Tags er verwendet und wie er diese formatiert. Auszeichnungsele- mente, wie etwa <GegliederteListe>, <RechnungsPosition> oder <Kapitel>, sind damit unproblematisch. Durch XML lassen sich Auszeichnungssprachen für spezielle Situatio- nen, z. B. den Datenaustausch zwischen zwei Unternehmen, oder für allgemein rele- vante, internetweite Anwendungsgebiete, festlegen (vgl. Kobligk 2004).

Hinter der Schaffung von XML stehen die folgenden Grundsätze (vgl. Kobligk 2004):

* anwendungsspeziﬁsche Auszeichnungen,
* eindeutige Strukturen,
* Trennung von Inhalt und Layout,
* Einfachheit,
* bestmögliche Fehlerüberprüfung.

Technologien für Smart Services

Die nachfolgende Abbildung zeigt das Beispiel Listing 4 im XML-Format auf.



###### Vergleich von JSON und XML

JSON und XML richten sich auf die gleichen Einsatzbereiche wie XML, unterscheiden sich jedoch in den folgenden Punkten (vgl. Augsten 2018):

* JSON verfolgt im Unterschied zu XML einen stark vereinfachten Ansatz mit Verzicht auf die Möglichkeit der Einbindung von Kommentaren in ein JSON-Dokument.
* Bei XML gehören eine strikte Typisierung, die Speziﬁkation einer formalen Doku- mentstruktur, benutzerspeziﬁsche Datentypen und Tags über Schemata zum Stan- dard. Auf diese Weise wird eine formale Validierung von XML-Dokumenten und - Datenströmen ermöglicht. Dieses Konzept ist jedoch bei JSON lediglich als Add-on verfügbar.
* Die Standardisierung des JSON-Schemas in Form eines Internet Drafts erfolgt sozu- sagen nebenbei.
* JSON formatiert Daten in ähnlicher Weise wie XML, jedoch im Vergleich dazu in äußerst kompakter Form.
* Der Größenunterschied eines formatierten Datensatzes ist von der Art der Kodierung abhängig. Diese erfolgt bei XML auf zwei verschiede Arten. Die Speicherung einfa- cher Datentypen einer Struktur ist in Form von XML-Tags oder in Form von Attributen möglich. Die Speicherung als Attribute ist in XML kompakter als in JSON.

Vor- und Nachteile von XML

Vorteile von XML sind (vgl. Müller 2007):

* + einfachere Lesbarkeit von teilweise komplexen, verschachtelt übertragenen Daten,
  + Etablierung als Standard-Austauschformat,
  + komfortable Übertragung von Kommentaren und Metadaten,
  + Vereinfachung des Austauschs mit Dritten.

Zu den Nachteilen von XML zählen (vgl. Müller 2007):

* + Die komplexe Syntax ist häuﬁg mit unnötigem „Ballast“ angereichert.
  + Sie ist zwar als vollwertige Markup-Sprache für XML unentbehrlich, für den reinen Datentransport jedoch nicht zwingend erforderlich.
  + Es ist erforderlich, die Struktur zuerst zu zerlegen, um die Daten in entsprechende JavaScript-Objekte umzuwandeln.

Durch JSON ist die Vorherrschaft von XML aufgelöst worden. Für spezielle Aufgaben, z. B. die Serialisierung von Objekten, ist eine schlankere Alternative möglich. JSON und YAML verdeutlichen Möglichkeiten einer verbesserten Lesbarkeit. Jedoch sind mit XML immer noch verschiedene Vorteile verbunden, so beispielsweise die Vielseitigkeit, Toolunter- stützung und Vielzahl der Standards, die auf XML aufbauen bzw. damit kombiniert wer- den können (vgl. Keshavarzi/Bayer 2011).

JSON erlaubt im Vergleich zu XML einen leichteren und einfacheren sowie strukturierte- ren Datenaustausch. JSON ist für den reinen Datenaustausch zukunftsweisend, es ver- drängt zunehmend XML. Es wird davon ausgegangen, dass in der Zukunft zunehmend mehr Webentwickler und Service Provider usw. JSON nutzen (vgl. Müller 2007).

### Internetkommunikationsprotokolle

Im Kontext der Entwicklung der Informations- und Kommunikationstechnologie sind vielzählige Systeme, Netze sowie Dienste entstanden. Für deren einwandfreie Funkti- onsweise sind Kommunikationsprotokolle eine wesentliche Voraussetzung, was insbe- sondere für Internetkommunikationsprotokolle gilt (vgl. Kaderali 2005, S. iii, 308).

Beim Internet handelt es sich um ein dezentrales, paketvermittelndes Datenkommuni- kationsnetz, das weltweit die Verbindung unzähliger Rechner (Hosts) und Übermitt- lungsknoten (Router, Nodes) ermöglicht. Dabei hat sich die Entwicklung und Weiterent- wicklung des Internets wesentlich im Kontext der Entstehung und des Ausbaus des Kommunikationsinternetprotokolls TCP/IP, dem Transmission Control Protocol/Internet Protocol, vollzogen (vgl. Kaderali 2005, S. 308).

Der ursprüngliche Gedanke von TCP/IP ist im Jahr 1974 von Vinton Cerf und Robert E. Kahn veröffentlicht worden (vgl. Cerf/Kahn 1974). 1980 erfolgte dann die Umstellung der Hostrechner des ARPANET (ca. 300) auf TCP/IP. 1983 konnte die Umstellung auf TCP/IP vollendet werden. Dies war sozusagen die Geburtsstunde des Internets. In den 1980er-

Technologien für Smart Services

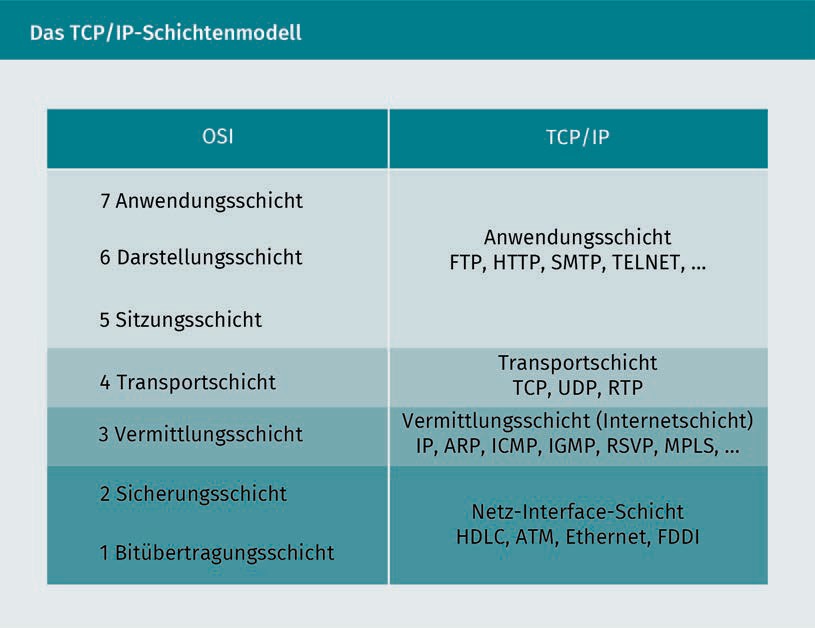
Jahren war TCP/IP frei verfügbar. Nach dem Militär wurde das Internet zunächst von wissenschaftlichen Einrichtungen und Hochschulen für den Informationsaustausch genutzt. Ab den 1990er-Jahren ist das Internet vermehrt auch für kommerzielle und pri- vate Anwendungen interessant geworden (vgl. Kaderali 2005, S. 308).

Das Übertragungsprinzip bzw. Prinzip der TCP/IP-Übertragung besteht in einer klaren Aufgabenverteilung zwischen den einzelnen Protokollen von TCP/IP. Die Aufgabe des Internetprotokolls/Internet Protocol (IP) liegt darin, die Datenpakete von einem Sen- der über das Kommunikationsnetz zu einem Empfänger weiterzuleiten. Geht ein Paket verloren, wird der Verlust den Kommunikationspartnern nicht angezeigt. Die entspre- chende Aufgabe obliegt dem oberhalb von IP operierenden TCP. Dieses fordert im Falle von Fehlern eine neue Übertragung von Datenpaketen an (vgl. Kaderali 2005, S. 308f.).

###### TCP/IP-Modell

Die TCP/IP-Protokollfamilie umfasst außer TCP und IP auch zahlreiche zusätzliche Pro- tokolle. Diese Protokolle können über diverse Schichten des OSI-Modells verteilt wer- den. TCP/IP-Protokolle können dem OSI-Modell schichtweise angeordnet werden. Im TCP/IP-Modell besteht im Unterschied zum OSI-Modell jedoch das Schichtenmodell nicht aus sieben Schritten. Vielmehr weist es lediglich vier Schichten aus. Jedoch las- sen sich diese Schichten eindeutig den OSI-Schichten zuordnen (vgl. Kaderali 2005, S. 310). In der folgenden Abbildung wird das TCP/IP-Schichtenmodell verdeutlicht.

Internetprotokoll/ Internet Protocol (IP) Dieses Protokoll hat die Funktion, die Datenpakete von einem Sender über das Kommunikati- onsnetz zu einem Empfänger weiterzu- leiten.



Der Vorteil des Einsatzes eines derartigen Schichtenmodells besteht darin, dass es bei- spielsweise für einen Anwendungsentwickler nicht erforderlich ist, zu wissen, welche Netztechnologie konkret bei einem potenziellen User bzw. Kunden gerade Verwendung ﬁndet. Mit dem Schichtenmodell wird die verwendete Technologie unterhalb der IP- Schicht verborgen. Es ist möglich, die Anwendung mit Verzicht auf das Wissen hinsicht- lich der Technologie, die verwendet wird, zu entwickeln (vgl. Kaderali 2005, S. 310).

Durch die Netz-Interface-Schicht werden die Datenübertragungsdienste zur Verfügung gestellt. Dies kann beispielsweise über einen STM-1 SDH-Kanal erfolgen oder über eine ATM-AAL5-Schnittstelle. Möglich ist dies auch über einen Frame Relay PVC oder eine Ethernetkarte. Hierzu zählen auch Kabel, Hardware (Netzwerkkarten), Software (Schnitt- stellentreiber) usw. Der Zugriff zur Netz-Interface-Schicht erfolgt über Address Resolu- tion Protocol (ARP). Das Address Resolution Protocol ist für die Zuordnung von Adres- sen der Vermittlungsschicht (IP-Adressen) zu Adressen der Netz-Interface-Schicht (z. B. MAC-Adressen) zuständig (vgl. Kaderali 2005, S. 310f.).

Inhalte der Internetschicht sind unter anderem das Internetprotokoll/Internet Protocol (IP). Dieses wird durch das Internet Control Message Protocol (ICMP) ergänzt, ferner das Internet Group Message Protocol (IGMP). Die zentrale Herausforderung liegt in der abschnittsweisen, verbindungslosen Übertragung von Datagrammen nach dem Best- Effort-Verfahren. Zudem besteht die Aufgabe des Routings bzw. der Wegewahl. Für die Erfüllung dieser Aufgaben wurden einige Protokolle, z. B. RIP2 (Routing Information Pro- tocol Version 2), BGP4 (Border Gateway Protocol Version 4) und OSPF (Open Shortest Path First), entwickelt (vgl. Kaderali 2005, S. 311).

Die Protokolle wurden mit der Zeit verbessert. Mit neueren Protokollen wird zusätzlich versucht, IP-Datagrammströme zu identiﬁzieren. Es wird auch versucht, diese verschie- denen Anwendungen, z. B. WWW oder Voice-over-IP, zuzuordnen. Der Zweck besteht unter anderem darin, dass im Anschluss eine priorisierte Behandlung der IP-Pakete mit Dienstgütern (Quality of Service – QoS) durchgeführt werden kann. Hierdurch wird eine Verkehrssteuerung ermöglicht. Diese ist an die Algorithmen der ATM-Technologie ange- lehnt. Von besonderer Bedeutung ist in diesem Kontext das Multiprotocol Label Swit- ching (MPLS). Das Multiprotocol Label Switching ermöglicht mit der sogenannten Label- Switching-Technologie eine schnelle Wegewahl und eine entsprechende Adressumwandlung, analog zur ATM-Technologie (vgl. Kaderali 2005, S. 311).

Die Transportschicht ist ein Verbindungsglied zwischen den Anwendungen Transport- schicht sowie Kommunikationsnetz. Dieses ist meist in das Betriebssystem eines Rech- ners (Host) bzw. Netzknotens (Node) integriert. Für diese Schicht sind zwei Protokolle bedeutend. Dabei handelt es sich zum einen um das Transmission Control Protocol (TCP). Dieses erlaubt eine gesicherte, Fehler korrigierende Verbindung (Flusskontrolle). Das zweite Protokoll ist das User Datagram Protocol (UDP) mit ungesicherter Verbin- dung und Fehlererkennung. Es verfügt aber über keine Fehlerkorrektur und Flusskon- trolle. Durch die zwei Protokolle wird die Ende-zu-Ende-Konnektivität zwischen zwei Endsystemen hergestellt. Sie ist mittels ihrer IP-Adressen gekennzeichnet. Diese ermög- lichen eine Segmentierung des Nachrichtenverkehrs (vgl. Kaderali 2005, S. 311).

Technologien für Smart Services

Im Kontext von UDP und multimedialen Anwendungen, z. B. Voiceover-IP, ﬁndet meist zusätzlich das Realtime Transport Protocol (RTP) Anwendung. Mit dessen Einsatz wird die Minimierung von Durchlaufzeiten und Jitter der IP-Pakete im Netz angestrebt (vgl. Kaderali 2005, S. 311).

Im Rahmen der Anwendungsschicht erfolgt die Bereitstellung der Applikationsproto- kolle. Der User kann deren Ergebnisse auf dem Monitor sehen. Dazu stehen die folgen- den Protokolle zur Verfügung. Hyper Text Transfer Protocol (HTTP) ermöglicht die Dar- stellung von World Wide Web-(WWW-)Inhalten. Weitere Protokolle sind Simple Mail Transfer Protocol (SMTP), Post Ofﬁce Protocol Version 3 (POP3) für E-Mail, File Transfer Protocol (FTP) für File Transfer, Telnet Protocol für Terminal-Anwendungen und Simple Network Management Protocol (SNMP) als Netzmanagementstandard (vgl. Kaderali 2005, S. 310f.).

### Semantische Beschreibungen

###### Semantisches Datenmodell

Unter einem semantischen Datenmodell versteht man im Kontext der Datenmodellie- rung eine Art abstrakte und formale Beschreibung sowie die Darstellung von einem Ausschnitt eines in einem bestimmten Kontext wahrgenommenen Umfeldes, beispiels- weise im Projekt. Für den Zweck der Formulierung von semantischen Datenmodellen kann auf die unterschiedlichsten Modellierungssprachen zurückgegriffen werden, so

z. B. auf das sogenannte Entity-Relationship-Modell (vgl. Simsion 2007, S. 49; Chaomei/Il-Yeol/Weizhong 2005, S. 189ff.).

Mit einem semantischen Datenmodell können verschiedene Ziele bzw. Zwecke erfüllt werden (vgl. Simsion 2007, S. 49; Chaomei/Il-Yeol/Weizhong 2005, S. 189ff.):

* + - Es lässt sich beispielsweise im Kontext der Konzeptionsphasen von Projekten in der Softwareentwicklung einsetzen. So kann es als Basis der Entwicklung und Nutzung einer Datenbank dienen.
    - Es kann auch als Bereichsdatenmodell oder als Unternehmensdatenmodell einge- setzt werden. So lassen sich für einzelne betriebliche Bereiche bzw. auch für ein gesamtes Unternehmen Modelle entwickeln, die in einzelnen Projekten in der Funk- tion von Entwurfsmustern oder als Referenzmodelle dienen. Nutzbar sind in diesem Kontext beispielsweise festgelegte Bezeichnungen und textuelle Beschreibungen. Nutzbar sind auch Bezeichnungen, unter welchen Komponenten von Benutzerme- dien bezeichnet werden, so z. B. Bildschirmeinstellungen, Listendarstellungen in Kurzform, Langform usw.
    - Ein semantisches Datenmodell ist auch nutzbar für die eindeutige wie auch über- sichtliche Darstellung von beliebigen Informationszusammenhängen, so z. B. Rollen, Rechte oder Beteiligte in Organisationsstudien oder Metamodelle, die in Methoden- handbüchern aufgeführt werden.

Semantisches Datenmodell

Ein semantisches Datenmodell bezieht sich auf eine Art abstrakte und for- male Beschreibung sowie Darstellung eines Ausschnitts des in einem Kon- text, beispielsweise im Projekt, wahrge- nommenen Umfel- des. Hierzu kommen verschiedene Model- lierungsprachen zum Einsatz.

Bestandteile eines semantischen Modells

Die Bestandteile eines semantischen Modells hängen von der gewählten Modellier- ungssprache ab.

Aus relevanten Informationsbegriffen werden beispielsweise Einheiten bzw. Entitäten bestimmt, sogenannte Entitätstypen. Über diese sind Informationen zu verarbeiten bzw. zu speichern.

Beispiel

Eine Onlinebank fasst Name, Geburtsdatum, Kontoeröffnungsdatum, Überweisungs- betrag, Saldo, Zinssatz und Postleitzahl sinnvoll wie folgt zusammen: PERSON, KONTO, ÜBERWEISUNG, ORT. Es gilt, die Beziehungen zu ermitteln, die zwischen den bestimmten Entitäten bestehen. Z. B. Annahme: Ein jedes KONTO gehört einer Per- son oder jede PERSON kann mehrere Konten haben. Die ermittelten Sachverhalte sind textuell zu beschreiben. Sie lassen sich graﬁsch beispielsweise mittels eines ER-Diagrammes darstellen.

Die erstellten semantischen Datenmodelle können sich hinsichtlich der verwendeten Form, der gewählten Inhalte oder der verwendeten Begriffe unterscheiden. Entspre- chend können sie durch die hier aufgeführten Kriterien bestimmt sein:

* Kriterium eingesetzte Modellierungsmethodik: Modelle, die mit UML erstellt wurden, werden anders aussehen als Modelle, die mit ERM erstellt wurden.
* Kriterium Modellerstellungszweck: Für ein Neudesign der Datenbank sind detaillier- tere Beschreibungen erforderlich als beispielsweise bei einem Wartungsprojekt, bei dem die Daten bereits vorhanden sind.
* Kriterium eingesetzte Modellierungswerkzeuge: Zum Einsatz kommen verschiedene Modellierungswerkzeuge wie z. B. die Form der Graﬁk von Beziehungen mit Rauten oder Linien usw., die gewählte Bezeichnung von Begriffen bzw. Beziehung, Relation usw. Diese Werkzeuge können jeweils unterschiedlich ausfallen.
* Kriterium projekt- bzw. unternehmensspeziﬁsche Regeln: Zu den Regeln zählt der Detaillierungsgrad der Modellierung, beispielsweise bei Beziehungen, ferner auch die Vorgabe, so z. B. dass ein Datenmodell zuerst im DV-Konzept erstellt werden soll.

Modellierungssprachen

Für die Beschreibung von semantischen Datenmodellen gilt das Entity-Relationship- Modell (ER-Modell) als das bekannteste Modell. Das Modell wurde immer wieder erwei- tert. Die betriebliche Praxis zeigt oft auch die Verwendung von vereinfachten Modellen, so z. B. die Martin-Notation (vgl. Simsion 2007, S. 49; Chaomei/Il-Yeol/Weizhong 2005, S. 189ff.). Des Weiteren ﬁndet das Modell auch Verwendung für die objektorientierte Modellierung und für Uniﬁed Modelling Language (UML).

Technologien für Smart Services

### Complex Event Processing

Bei Complex Event Processing (CEP) handelt es sich wörtlich übersetzt um die kom- plexe Verarbeitung von Ereignissen. CEP stellt ein Themengebiet dar, welches der Infor- matik zuzuordnen ist. Es handelt sich um die Thematik der Erkennung, der Analyse, der Gruppierung sowie der Verarbeitung von Ereignissen, die voneinander abhängig sind. CEP gilt als eine Art Oberbegriff für Methoden, Techniken und Werkzeuge. Diese dienen der zeitnahen Verarbeitung von Ereignissen, noch während sich diese ereignen.

Für ereignisgesteuerte Informationssysteme ist eine systematische wie auch automati- sche Verarbeitung von Ereignissen durch Complex Event Processing erforderlich.

Mit CEP wird aus Ereignissen ein weit größeres und wertvolleres Wissen abgeleitet. Die- ses Wissen besteht in Form von komplexen Ereignissen. Es sind Situationen, die ledig- lich als Kombination von mehreren Ereignissen erkannt werden können (vgl. Eckert/Bry 2009).

Für die Echtzeitverarbeitung der diversen Datenströme und um die Ereignisse extrahie- ren und analysieren zu können, ist es erforderlich, dass die Systeme hohe Lasten zu tragen in der Lage sind.

In diesem Kontext lassen sich die folgenden exemplarischen Einsatzgebiete nennen: die Netzwerküberwachung, die öffentliche Sicherheit, der Katastrophenschutz sowie das Energiemanagement. Die Anzahl der Ereignisse in Computersystemen wurde in den zurückliegenden Jahren wesentlich gesteigert durch (vgl. Eckert/Bry 2009):

* + - Service Oriented Architecture (SOA),
    - Event-Driven Architecture (EDA),
    - kostengünstige Sensortechnik,
    - die rechtlich, vertraglich oder betrieblich bedingte Überwachung von Informations- systemen.

Damit ist auch die Anforderung verbunden, dass solche Ereignisse automatisch, syste- matisch wie auch zeitnah zu verwalten bzw. zu verarbeiten sind.

Es bestehen die folgenden bedeutenden Anwendungsgebiete von CEP (vgl. Eckert/Bry 2009):

* + - Business Activity Monitoring,
    - Sensornetzwerke,
    - Marktdaten.

Mit Business Activity Monitoring werden Geschäftsprozesse überwacht. Zudem dient es auch der Beobachtung von unternehmenskritischen Ressourcen. Das Ziel besteht im frühzeitigen Erkennen von Problemen und Chancen. Zu diesem Zweck werden Ereig- nisse in Key-Performance-Indikatoren zusammengefasst, z. B. die durchschnittliche Dauer eines Prozesses (vgl. Eckert/Bry 2009).

Complex Event Pro- cessing (CEP)

Der Sammelbegriff CEP umfasst Metho- den, Techniken bzw. Werkzeuge, die es ermöglichen, Ereig- nisse zu verarbeiten, während sie erfol- gen.

Durch Sensornetzwerke werden Messdaten aus der Außenwelt (z. B. an Supervisory Control and Data Acquisition Systems) übermittelt, die beispielsweise für die Überwa- chung industrieller Anlagen eingesetzt werden. Dabei wird versucht, Messfehler oder andere Fehler möglichst zu vermeiden oder zumindest zu minimieren. Hierzu werden Daten von mehreren Sensoren miteinander kombiniert. Zudem ist es häuﬁg erforder- lich, aus einfachen numerischen Messungen (z. B. Temperatur, Rauch usw.) eine kom- plexere Situation (z. B. einen Brand) zu erschließen (vgl. Eckert/Bry 2009).

Marktdaten lassen sich auch im Sinne von Ereignissen betrachten, so z. B. Preise für Rohstoffe, Aktien usw. Es ist notwendig, die Marktdaten möglichst zeitnah sowie konti- nuierlich der Analyse zu unterwerfen. Nur auf diese Weise können Trends frühzeitig bzw. noch rechtzeitig erkannt werden. Dies ist zugleich die Grundlage, um richtig und rechtzeitig automatisch zu agieren, zu reagieren, zu intervenieren usw. Hierfür ﬁndet auch der Begriff Algorithmic Trading Anwendung. Die Anwendungen enthalten eine Ver- teilung von zu erkennenden Situationen sowie von zugehörigen Informationen über mehrere Ereignisse. Sie sind mittels CEP aus mehreren Ereignissen sowie deren Zusam- menhängen zu erschließen (vgl. Eckert/Bry 2009).

###### Complex Event Processing – Arten

Ursprünglich wurde der Begriff Complex Event Processing (CEP) durch zahlreiche vonei- nander unabhängige Forschungsrichtungen geprägt. Berücksichtigung ﬁndet dabei auch die ereignisorientierte Simulation über aktive Datenbanken sowie das Netzwerk- management (vgl. Luckham 2002).

CEP erlangt im Sinne eines eigenständigen Fachgebiets zunehmend Bedeutung. Dies gilt insbesondere in der Industrie. So wurde beispielsweise im Jahr 2008 die Event Pro- cessing Technical Society gegründet.

CEP bezieht sich speziell auf die Behandlung von Ereignissen, die erst durch das Zusammenwirken von mehreren Ereignissen eintreten. Typische Merkmale solcher Ereignisse sind: eine gegenseitige Abhängigkeit mit mehrfachem Eintreten infolge von physikalischen Eigenschaften, nacheinander ablösend oder nebeneinander gleichzeiti- ges Eintreten; wiederholtes Eintreten in gleicher Verkettung, Unabhängigkeit oder Zusammenhängen sowie gegebenenfalls Eintreten in nebenläuﬁgen Prozessen.

Gegenstand von CEP

Durch ereignisgesteuerte Computerapplikationen lässt sich der Programmablauf mittels einer sequenziellen Folge von Events steuern. So werden beispielsweise infolge von einzelnen Ereignissen, z. B. einem Mausklick, dem Eingang einer E-Mail, dem Ende eines Ladevorgangs usw., weitere abhängige Ereignisse ausgelöst. Dies ist z. B. der Fall, wenn eine Datei auf der Festplatte gespeichert wird, wenn eine Hinweismeldung auﬂeuchtet oder ein Programmfenster geöffnet wird. CEP beschäftigt sich mit Ereignissen, die mehrfach redundant, mehrläuﬁg nebeneinander, gegebenenfalls auch unzuverlässig verkettet erfolgen. Die Logik des Auftretens ist unscharf deﬁniert, sogenannte Fuzzylo-

Technologien für Smart Services

gik. Durch CEP können bereinigte Ereignisse als Status gewonnen werden, die gegebe- nenfalls die Beseitigung von unerwünschten bzw. überﬂüssigen Quellereignissen ermöglichen.

Begriff Event

Im Rahmen des Complex Event Processing ﬁndet der Begriff Event Anwendung, der ver- schiedene Aspekte beinhaltet. Darunter fällt insbesondere alles, was sich ereignet oder nicht, ein Objekt, welches das Ereignis unter anderem für den Zweck der maschinenba- sierten Verarbeitung repräsentiert, kodiert bzw. speichert. Typischerweise ist ein Event ein Ereignis, das innerhalb eines bestimmten Systems bzw. einer Domäne erfolgt. Das Event kann als bereits erfolgt oder als etwas innerhalb dieser Anwendungsdomäne Geschehenes gesehen werden. Event bezieht sich auf die Beschreibung einer Program- mierentität, die ein solches Ereignis eines IT-Systems repräsentiert.

CEP-Konzepte

In CEP können High-Level-Konzepte beschrieben sein, welche die Eventverarbeitung sowie die Erkennung von Ereignismustern übernehmen. Im Rahmen von CEP wird ein virtuelles Event deﬁniert im Sinne eines Ereignisses, das physikalisch bzw. in der realen Welt nicht eintritt. Das virtuelle Event wird wie jedes andere Ereignis in CEP gesehen.

Beinahe alles, was in der realen Welt existiert oder innerhalb eines Rechners abgebil- det wird, ist ein Ereignis zur Nutzung von CEP. CEP zielt darauf ab, die Beziehungen unter den diversen Ereignisebenen und unter den Design Patterns für die Erstellung der Ereignisse herzustellen. Deren Semantik soll ihr Speichermedium und ihren Über- tragungsmechanismus nicht beeinträchtigen. Eine solche Deﬁnition gilt in Teilbereichen als kompatibel zur Ereignisdeﬁnition der Wahrscheinlichkeitstheorie. Bei CEP bestehen zwei mögliche Fälle (vgl. Eckert/Bry 2009):

Komplexe Ereignisse werden a priori als bekannte Muster über Ströme von Ereignissen speziﬁziert. Hierbei stellen spezielle Ereignisanfragesprachen eine komfortable Mög- lichkeit dar, um komplexe Ereignisse zu speziﬁzieren sowie diese efﬁzient zu erkennen. Oder aber zuvor unbekannte Muster sollen in den Strömen als komplexe Ereignisse erkannt werden. Hierbei ﬁnden maschinelles Lernen sowie Data Mining auf Ereignisse Anwendung. Die nachfolgenden Ausführungen richten sich auf Ereignisanfragen.

CEP ist in enger Verbindung zu anderen Themen zu betrachten, beispielsweise der Visu- alisierung von ereignisbezogenen Daten für menschliche Benutzer, Message-based Middleware für die Weiterleitung von Nachrichten, Regelsystemen für die Speziﬁkation reaktiven Verhaltens (ECA-Regeln, reaktive Logikprogrammierung usw.) sowie Business Process Management (BPM) (vgl. Eckert/Bry 2009).

###### Ereignisanfragesprachen

Es ﬁnden überwiegend drei Spracharten für den Ausdruck von Ereignisanfragen Anwen- dung, die auf den folgenden Grundideen basieren (vgl. Eckert/Bry 2009; Paton 1998):

* Kompositionsoperatoren,
* Datenstromanfragesprachen.

Kompositionsoperatoren

Active Rules in Database Systems: Die Kompositionsoperatoren stammen aus aktiven Datenbanksystemen (vgl. Paton 1998). Im Unterschied dazu gelten neue Systeme, z. B. Amit (vgl. Adi/Etzion 2004, S. 177ff.), als unabhängig von einer Datenbank. Dabei werden Anfragen gegen einzelne Ereignisse mit verschiedenen Operatoren zu komplexen Ereig- nisanfragen gebündelt. Als typischer Operator gilt die Konjunktion von Ereignissen. Dazu zählen sämtliche Ereignisse, gegebenenfalls auch zu verschiedenen Zeitpunkten. Zudem zählt die Sequenz dazu bzw. nacheinander folgende Ereignisse, ferner die Nega- tion in einer Sequenz bzw. der Fall, dass ein Ereignis nicht in der Zeit zwischen zwei anderen Ereignissen erfolgt. Sehr komplexe Anfragen können auch mittels einer Schachtelung dargestellt werden. Durch einige Sprachen sind Einschränkungen mög- lich. So kann beispielsweise deﬁniert sein, welche Ereignisse für die Zusammensetzung eines komplexen Ereignisses zu betrachten sind. Über Selektion lässt sich z. B. aus- schließlich das erste oder auch letzte Ereignis eines bestimmten Typs auswählen. Mit- tels Konsumieren von Ereignissen kann deren Wiederverwendung in neuen komplexen Ereignissen verhindert werden, so z. B. in dem Falle, dass diese schon für ein komple- xes Ereignis ihren Einsatz gefunden haben.

Bei den Kompositionsoperatoren handelt es sich um eine intuitive Möglichkeit, kom- plexe Ereignisse zu speziﬁzieren. Diese lassen sich mit dieser Methode unterstützen. Sie ist geeignet für temporale Zusammenhänge und Negation. Der Ansatz beinhaltet das Selektieren wie auch Konsumieren von Ereignissen. Jedoch kann das intuitive Verständ- nis von Operatoren auch vereinzelt zu Problemen führen, so beispielsweise im Falle von mehreren Varianten der Interpretation einer Sequenz. Zudem besteht in der Praxis das Risiko, dass Daten in Ereignissen vernachlässigt werden. Dies gilt insbesondere hin- sichtlich Komposition und Aggregation. Einige CEP-Produkte basieren auf dem Kompo- sitionsoperator, so z. B. IBM Active Middleware Technology (Amit) sowie ruleCore (vgl. Eckert/Bry 2009; Adi/Etzion 2004, S. 177ff.).

Datenstromanfragesprachen

Zu den Datenstromanfragesprachen zählt z. B. CQL (vgl. Arasu/Babu/Widom 2006, S. 121ff.). Datenstromanfragesprachen basieren auf der Datenbankanfragesprache SQL. Ihnen liegt ein Basisprinzip zugrunde: Für Datenströme, welche Ereignisse als Tupel beinhalten, erfolgt eine Umwandlung in Relationen. Auf deren Grundlage lässt sich eine reguläre SQL-Anfrage auswerten. Das Ergebnis, beispielsweise in Form einer Relation, kann in einen Datenstrom umgewandelt werden. Der Vorgang kann konzeptionell an jedem Zeitpunkt einer vorgegebenen diskreten Zeitachse erfolgen (vgl. Jain et al. 2008, S. 1379ff.).

Generell ist eine Umwandlung der Datenströme in Relationen über diverse Fensterope- rationen möglich. Beispiele: „sämtliche Ereignisse der vergangenen 30 Minuten“ oder

„die letzten 20 Ereignisse“ usw.

Zur Rückumwandlung einer Ergebnisrelation in einen Datenstrom existieren die folgen- den Möglichkeiten (vgl. Eckert/Bry 2009):

Technologien für Smart Services

* Lediglich Tupel, welche im Vergleich zum vorangegangenen Ergebnis dazukommen, werden zum neuen Ereignis.
* Lediglich Tupel, welche entfallen, werden zum neuen Ereignis.
* Jedes Tupel des aktuellen Ergebnisses wird zum neuen Ereignis.

Die Datenstromanfragesprachen gelten für die Aggregation von Ereignisdaten als besonders geeignet, so z. B. für Marktdaten. Zudem wird durch sie eine weitgehend optimale Integration mit Datenbanken ermöglicht. Im Unterschied dazu sind Negation oder auch temporale Zusammenhänge in vielen Fällen nicht einfach auszudrücken. Eine Umwandlung von Strömen in Relationen und umgekehrt sowie das Erfordernis einer diskreten Zeitachse gelten als unnatürlich. Dennoch gelten SQL-basierte Daten- stromanfragesprachen als erfolgreiche und bewährte Ansätze. SQL-basierte Datenstro- manfragesprachen werden durch einige efﬁziente wie skalierbare Industrieprodukte unterstützt, so z. B. durch Oracle CEP, Coral8, StreamBase, Aleri und das Open-Source- Projekt Esper. Es ist zu berücksichtigen, dass erhebliche Unterschiede unter den Sprachvarianten bestehen können. Es existieren bedeutende, teilweise über die Grund- idee hinausgehende Erweiterungen (vgl. Eckert/Bry 2009).

Complex Event Processing gilt als industrieller Wachstumsmarkt und als bedeutender Forschungsbereich. Es besteht noch Bedarf an Erfahrungen zu Ereignisanfragesprachen in konkreten Projekten. Die Ursache ist unter anderem darin zu sehen, dass im Algorith- mic Trading, dem größten CEP-Markt, eine hohe Geheimhaltung besteht. Zudem existie- ren nicht ausreichend Benchmarks zum Vergleich und zur Voraussage der CEP-System- Performance. CEP ist für Ereignisanfragesprachen relevant. Ferner ist der CEP-Einsatz auch in Referenzarchitekturen und Design Patterns von hoher Relevanz (vgl. Eckert/Bry 2009).

Es besteht noch ein gewisser Forschungsbedarf im Complex Event Processing. Dieser richtet sich vorwiegend auf die formalen Grundlagen. Dies gilt insbesondere bezüglich der Ausdrucksstärke sowie der Optimierung. Zur Anfrageoptimierung gelten die Ausnut- zung von gemeinsamen Teilanfragen (Multi-Query-Optimierung) und die verteilte und parallele Auswertung als bedeutend. Zu den künftigen Forschungsthemen zählen zudem der Umgang mit ungenauen Ereignissen, so beispielsweise probabilistische Methoden sowie die Erkennung unbekannter komplexer Ereignisse, z. B. Data Mining auf Ereignisströmen (vgl. Eckert/Bry 2009).

###### Blockchain/Smart Contracts

Blockchain gilt als die zukunftsweisende Datenbanktechnologie. Mit Blockchain können die Grenzen der Finanzindustrie überwunden werden. Es lässt sich beispielsweise Rezeptbetrug verhindern, E-Autos können aufgeladen werden usw. Durch Blockchain werden neue Geschäftsmodelle ermöglicht. Die Technologie verändert ganze Branchen (vgl. Kaltofen 2016).

Bei Blockchain handelt es sich um eine neuartige Technik zum Speichern von Daten. Zudem ermöglicht diese ein sicheres Management von Informationen jeglicher Art. Blockchain hat den Ursprung in der Internetwährung Bitcoin. Mit Bitcoin lassen sich

Blockchain

Bei der Blockchain handelt es sich um eine neuartige Tech- nik zum Speichern von Daten.

Geldwerte ohne zentrale Bankinstanz überweisen. Transaktionen gelten als zentrale Grundeinheiten der Blockchain. Tauschen zwei Parteien Informationen gegenseitig aus, können die Daten im Anschluss veriﬁziert und validiert werden. Gleichzeitig lässt sich überprüfen, ob eine Partei über die entsprechenden Rechte verfügt, um die Transaktio- nen durchzuführen. Im Anschluss wird das Mining – übersetzt: „Schürfen“ – durchge- führt. Dabei werden nach einer bestimmten Zeit die Transaktionen zu Blöcken gebün- delt. Darüber wird ein Hash-Wert gebildet. Die Blöcke werden im Anschluss an die Kette angehängt und über ein Peer-to-Peer-Netzwerk verteilt (vgl. Kaltofen 2016).

Blockchain-Technologie wird als manipulationssicher erachtet. Dies ist darin begründet, dass die Blöcke mit einer Hash-Funktion verschlüsselt werden. Kopien der Datei wer- den im Internet verbreitet. Für eine Manipulation der Inhalte der Blockchain wäre es erforderlich, zumindest 51 % der Kopien zu ändern, was mit (zu) hohen Kosten verbun- den ist (vgl. Kaltofen 2016). Blockchain eignet sich für die Einrichtung von intelligenten, automatisierten Vertragsbeziehungen (vgl. Neidel 2016).

Bei Smart Contracts („intelligente Verträge“) handelt es sich um die technische Ausprä- gung von juristischen Verträgen, die deren Inhalte maschinenlesbar machen. Sie machen Bedingungen und Zustände von Vertragsinhalten bei dritten Quellen abfragbar und können diese eigenständig veriﬁzieren und daraus eigenständig Handlungen ablei- ten (vgl. Neidel 2016).

Smart Contracts gelten als die bekannteste Applikation von Blockchains. Sie stellen die neue Form von Verträgen dar. Den Kern bilden webbasierte Computerprotokolle, die Verträge abbilden. Sie unterstützen technisch die Abwicklung eines Vertrages. Die ent- sprechenden Computeralgorithmen können festlegen, welche Bedingungen zu welcher Entscheidung führen. Die Algorithmen sind in der Lage, beispielsweise Verträge in Echt- zeit zu überwachen und die Rechte der Vertragspartner automatisch durchzusetzen. Juristen wären für die Fassung und Überwachung der Verträge nicht mehr erforderlich (vgl. Kaltofen 2016).

Beispiel

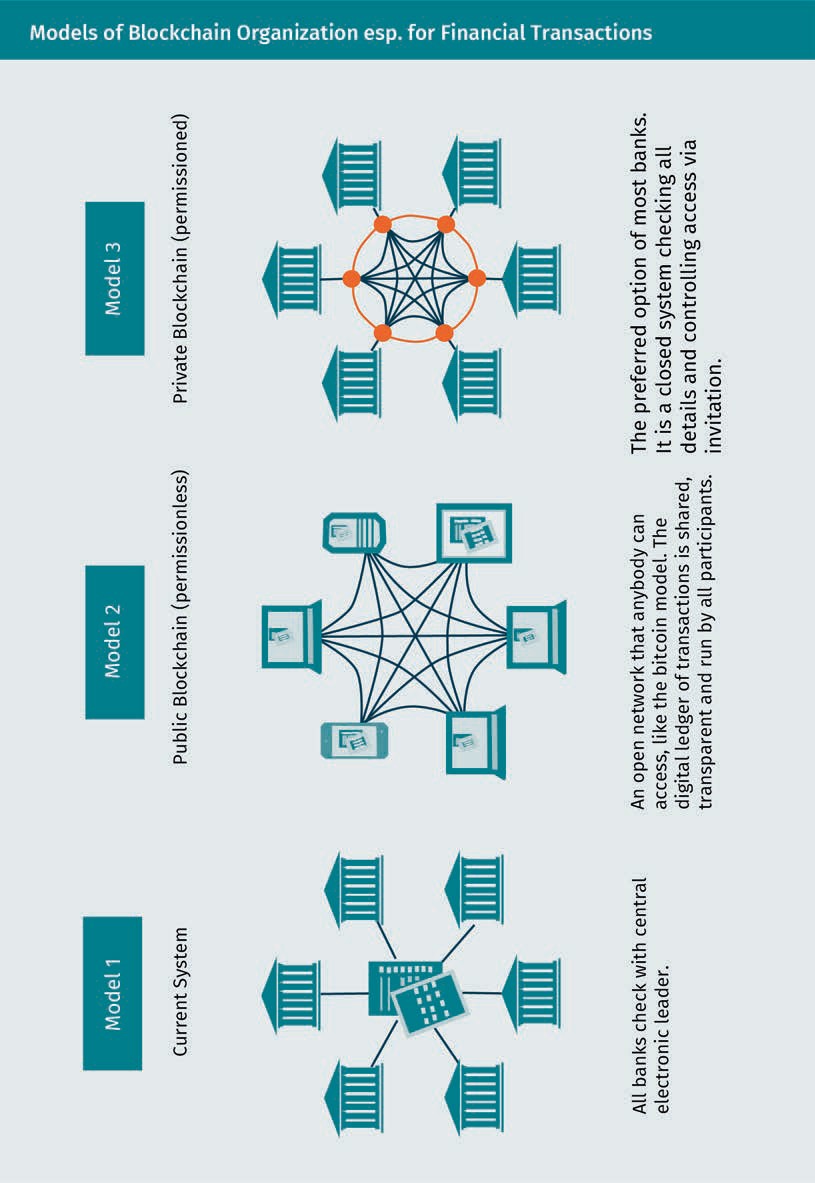
Ein elektronischer Vertrag mittels automatisierter Abwicklung eines Kredit- oder Leasingvertrages für ein KFZ – Smart Contract – ist physikalisch im Auto gespeichert. Es überprüft selbstständig monatlich, ob die vereinbarte Zahlungsrate des Fahrers eingegangen ist. Wenn die Zahlung nicht geleistet wurde, veranlasst der Vertrag automatisch Mahnungen. Er kann als äußerstes Mittel sogar eine Wegfahrsperre aktivieren und eine weitere Nutzung des Fahrzeugs physisch stoppen. Für diese ein- zelnen Prozessschritte ist keine Steuerung von außen notwendig. Das Auto erweitert quasi seine Entität in den Cyberspace hinein. Es eliminiert bisherige Kontroll- und Steuerinstanzen. Dies gilt auch für manuelle Instanzen. Dabei wird die Integrität, Historisierung sowie die Validierung sämtlicher Transaktionen in der Blockchain gespeichert und gesteuert (vgl. Neidel 2016).

Technologien für Smart Services

Eine Plattform für die technische Umsetzung ist Microsoft Azure Cloud. Durch Block- chain wird die Trauung per QR-Code über das Internet ermöglicht. Mittels einer Bitcoin- Variante soll es möglich werden, beispielsweise die Bezahlung an Ladestationen von Elektroautos zu realisieren. Durch die Blockchain-Technologie soll ein einheitliches wie auch kostengünstiges Bezahlsystem entwickelt werden, das unabhängig von Kassierern arbeitet. So könnte das E-Auto aufgeladen und der dafür benötigte Strom digital bezahlt werden (vgl. Kaltofen 2016).

Durch den kombinierten Einsatz der dezentralen Trust-Center-Einheit Blockchain sowie der damit verbundenen Vernetzung mit digitalen Aktoren und Schnittstellen zu Dritt- systemen wird es möglich, der Entität des Vertrages umfangreiche Überwachungs- und Handlungsmöglichkeiten zuzuweisen. Sämtliche nachgelagerten Prozessschritte und Wertschöpfungsstufen lassen sich digitalisieren und automatisieren. Sie können poten- ziell entfallen bzw. lassen sich für neue, wertbringende Services nutzen (vgl. Neidel 2016).

Die Kosten für Intermediäre entfallen gleichzeitig. Dies führt dazu, dass die Finanzin- dustrie versucht, die neuen Möglichkeiten für eine sichere, efﬁziente, ubiquitäre und globale Zahlungsarchitektur mittels angepasster Blockchain zu nutzen (vgl. Neidel 2016). Mögliche Modelle werden durch die folgende Abbildung veranschaulicht.



Infolge von Blockchains werden auch Geschäftsmodelle gefährdet. Starke Veränderun- gen ergeben sich beispielsweise in der Justiz. Mit dem Einsatz von Smart Contracts kön- nen Verträge ohne Einsatz von Notaren oder Anwälten geschlossen werden. Auch ist es möglich, dass sich teilweise die Tätigkeit von Gerichtsvollziehern erübrigt. Es lässt sich

Technologien für Smart Services

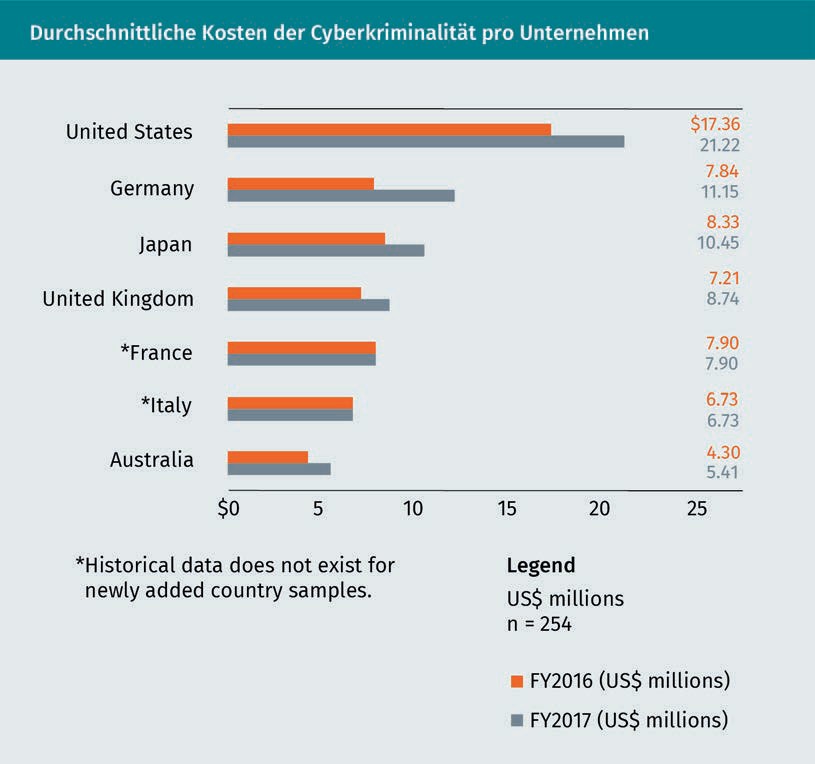
beispielsweise der Zugang zu unbezahlten Waren oder Services automatisch sperren. Ebenfalls ergeben sich im Immobilienbereich Veränderungen. So ist es denkbar, den Kauf und Verkauf von Immobilien mittels des Einsatzes von Smart Contracts abzuwi- ckeln, wodurch kein Makler mehr erforderlich wird. Dessen Provision kann dadurch ein- gespart werden. Im Bereich der Banken führt Blockchain zu einer schlankeren Gestal- tung der Infrastruktur (vgl. Kaltofen 2016).

### Sicherheit

###### Aspekte der Sicherheit und des Datenschutzes

Eine zentrale Voraussetzung für das Funktionieren von Smart Services liegt im Daten- austausch. Mit dem Austausch, der Verarbeitung und Speicherung von Daten sind jedoch die größten Risiken der gesamten Digitalisierung verbunden. Es ist bekannt, dass jede Komponente über Sicherheitslücken verfügt. Dadurch wird es Cyberkriminel- len möglich, Informationen der User abzufangen und für eigene Zwecke zu verwenden. Sensible Daten reichen von persönlichen Daten, aktuellen Standortdaten über Finanz- daten bis zu Krankheitsverläufen. Da viele Daten privat und sensibel sind, sollten sie besonders geschützt werden. Dies lässt sich jedoch nicht immer praktisch bewerkstelli- gen. Diesbezüglich sind Industrie und Politik gefordert, Überlegungen und Regelungen hinsichtlich des Datenschutzes zu treffen sowie verlässliche Standards zu deﬁnieren. Das Ziel des Datenschutzes besteht in einer Erhöhung der IT-Security (vgl. o. V. 2018a; BMWi 2017, S. 43f.).

Infolge der teuren und verheerenden Folgen von Cyberkriminalität für die Unterneh- men ist eine strategische Planung sowie Überwachung von Investitionen in IT-Sicher- heit zunehmend bedeutend (vgl. Accenture 2017). Die nachfolgende Abbildung weist die Kosten der Cyberkriminalität aus.



Gezielte Investitionen in innovative Technologien können einen signiﬁkanten Unter- schied bewirken. Es gilt, Cyberkriminalität zu zerschlagen (vgl. Accenture 2017). Damit die Unternehmen richtig auf die immer rafﬁnierteren Angriffe reagieren können, ist es erforderlich, dass sie eine dynamische sowie auch ﬂexible Sicherheitsstrategie anstre- ben. Anstatt den Fokus lediglich auf den Umfang der Sicherheitslösung zu richten, gilt es vielmehr, die Widerstandsfähigkeit zu stärken. Diese sollte von innen heraus im Zentrum stehen (vgl. Accenture 2017). In der Praxis zeigt sich, dass die IT-Landschaft von Unternehmen oft sehr stark durch ältere Legacy-Systeme geprägt ist. Es ist notwendig, dass bei derartigen Systemen ein spezieller Fokus auf deren proaktiven und professio- nellen Schutz gelegt wird. Denn ihre technologische Basis ist oft nicht mehr durch die aktuellsten oder neuesten Security-Produkte unterstützt. Insofern können sie ein bevorzugtes Einfallstor für Angreifer sein (vgl. Accenture 2017).

Cyberkriminalität stellt ein ernst zu nehmendes Problem und eine Herausforderung für IT-Entwickler dar. So wurde z. B. durch den Computerwurm Tuxnet am 23.12.2015 in der Ukraine das Stromnetz zum totalen Black-out gebracht. In der EU wurden durch WanaCry Win Computer gesperrt und bis zu vier Milliarden Euro Schaden verursacht. Weitere bekannte Beispiele sind der Hackerangriff auf die Bundesregierung durch eine russische Hackergruppe, die auch über eine gewisse Zeit im deutschen Verwaltungsnetz

Technologien für Smart Services

unterwegs war. Ein weiterer gefährlicher Trend liegt in der sogenannten Botnet-Attacke. Das Risiko wächst zunehmend, dass etwa Schadsoftware über PCs, Tablets, Smartpho- nes, Smart-TV-Geräte, smarte Kühlschränke usw. verbreitet wird. Selbst ein USB-Spei- cherstick kann im Neuzustand bereits eine Schadsoftware beinhalten, die das System zum Erliegen bringen kann oder Daten ausspioniert, z. B. durch sogenannte Trojaner usw.

Mit fortschreitender Digitalisierung und Vernetzung von Wirtschaft und Gesellschaft wird die Datensicherheit immer wichtiger. Dies gilt auch für verlässliche Standards. Schließlich hat es dazu geführt, dass beispielsweise das BMWi beabsichtigt, mittelstän- dische Unternehmen bzw. KMUs mittels geeigneter Informationen in die Lage zu verset- zen, dass sich diese besser vor den Gefahren, die im Netz lauern, schützen können. Es soll ihnen ermöglichen, die Chancen, die der digitale Wandel bringt, entsprechend zu nutzen (vgl. BMWi 2018).

Durch die Initiative „IT-Sicherheit in der Wirtschaft“ werden durch das BMWi kontinuier- lich konkrete Maßnahmen entwickelt, die zur nachhaltigen Verbesserung des Bewussts- eins für IT-Sicherheit, insbesondere in KMU führen sollen. Hierbei liegen die Schwer- punkte unter anderem in der (vgl. BMWi 2018) …

* … Optimierung des Ordnungsrahmens,
* Stärkung der nationalen und europäischen IT-Sicherheitswirtschaft,
* Kennzeichnung von IT-Sicherheitseigenschaften mittels Gütesiegeln sowie Zertiﬁka- ten.

###### Datenschutz in der Cloud

Da Cloud-Dienste im Zentrum der digitalen Transformation liegen, ist ein verlässlicher Datenschutz in der Cloud erforderlich. Auf der Plattform Trusted Cloud werden Cloud- Services und Dienstleistungen nur dann gelistet, wenn sie den Kriterien des Trusted Cloud Labels entsprechen. Sie müssen vertrauenswürdig sein und die Anforderungen an Transparenz, Sicherheit, Qualität sowie Rechtskonformität erfüllen. Trusted Cloud gilt vor allem auch für KMU als hilfreich (vgl. BMWi 2018).

###### Einheitliche Standards für die Industrie 4.0

Durch eine enge Vernetzung von Technologien und Wertschöpfung wird ein intensiver Austausch von Daten über Schnittstellen möglich. Für die Industrie 4.0 sind einheitliche Normen und Standards erforderlich (vgl. BMWi 2018).

###### Standardisierung und Interoperationalität

In den Informations- und Telekommunikationstechnologien (IKT) gelten Standardisie- rung und Interoperationalität als von technischer wie auch von wirtschaftlicher Bedeu- tung. Die Standardisierung und Interoperationalität trägt wesentlich zur Sicherheit bei.

Wem es gelingt, Standards zu entwickeln und diese durchzusetzen, kann sich im Wett- bewerb Vorsprünge sichern. Es wird zunehmend auf offene Standards und Interopera- bilität gesetzt. Dadurch soll die Funktionsfähigkeit von komplexen IKT-Systemen wie auch ein ungehinderter Zugang zu den IKT-Märkten gewährleistet werden. Gleichzeitig lässt sich dadurch ein fairerer Wettbewerb schaffen (vgl. BMWi 2018).

###### Sicherheit und Datenschutz

Aus datenschutzrechtlicher Perspektive drängt sich die Frage auf, ob bei Webservices eine Auftragsdatenverarbeitung vorliegt, beispielsweise in dem Fall, dass ein Portalbe- treiber in seinem Portal einen Webservice eines externen Anbieters implementiert hat, vor allem aber, wenn darüber ein Austausch von personenbezogenen Daten erfolgt. Dies gilt es, im Einzelfall zu entscheiden. Es ist abhängig von der erbrachten Leistung. Häuﬁg kann jedoch eine Funktionsübertragung vorliegen. Dies ist insbesondere dann der Fall, wenn neben der konkreten Verarbeitung von personenbezogenen Daten ein gesamter Aufgabenbereich übertragen wird. Es ist zu berücksichtigen, dass der Dritte dann nicht mehr lediglich Auftragnehmer, sondern selbst die verantwortliche Stelle ist. In einem solchen Fall richtet sich die Aufmerksamkeit nicht nur darauf, dass lediglich eine Verarbeitung der Daten in Auftrag gegeben ist, sondern auch darauf, ob die erb- rachten Leistungen und Prozesse gegebenenfalls alleine in der Verantwortung des Web- service-Anbieters liegen (vgl. Czernik 2016).

Wenn eine Funktionsübertragung vorliegt, ist zu überprüfen, ob eine Übermittlung von Daten gemäß § 28 BDSG als Verarbeitung für „eigene Geschäftszwecke“ erfolgen darf oder ob gegebenenfalls eine Einwilligung einzuholen ist (vgl. Czernik 2016). Die Vor- schriften für den Datenschutz wurden durch die Europäische Datenschutz-Grundverord- nung (DSGVO), welche seit dem 25.05.2018 gilt, wesentlich erweitert.

Im Zusammenhang mit der Absicherung des Systems und insbesondere bei Webservi- ces sind die folgenden Punkte zu beachten (vgl. Czernik 2016):

* die Absicherung der Übertragungswege bei Webservices (Verschlüsselung),
* ein Rollen- und Berechtigungskonzept,
* das Durchführen von Penetrationstests,
* die Protokollierung,
* das regelmäßige Einspielen sicherheitsrelevanter Updates,
* die regelmäßige Überprüfung der Verfügbarkeit und Funktionalität der Schnittstelle.

Zusammenfassung

Bei JSON handelt es sich um ein Textformat für den Austausch von Daten. Es ist ein einfacher Standard für die strukturierte Kodierung von Daten in Form von men- schenlesbarem Text. JSON ist mit einer Skriptsprache verbunden und als Datenfor- mat von JavaScript abgeleitet. Es erfüllt jedoch auch ähnliche Aufgaben wie XML.

Technologien für Smart Services

YAML ist ein Format für die Konﬁguration nach Ruby on Rails. Dabei handelt es sich um eine Sprache für den Datenaustausch, die leicht lesbar und einfach zu editieren ist. XML ist ein Datenaustauschformat. Eine Vielzahl von Markup-Sprachen basiert auf XML. Durch XML wird kein fester Satz von Auszeichnungselementen vorgeschrie- ben, sondern es lässt sich eine eigene Auszeichnungssprache mit beliebigen Tags deﬁnieren. Durch XML lassen sich Auszeichnungssprachen für spezielle Situationen, beispielsweise den Datenaustausch zwischen zwei Unternehmen, oder für allge- mein relevante, internetweite Anwendungsgebiete festlegen. Das Internetprotokoll (IP) hat die Funktion, die Datenpakete von einem Sender über das Kommunikati- onsnetz zu einem Empfänger weiterzuleiten.

Das semantische Datenmodell bezieht sich auf eine Art abstrakte und formale Beschreibung sowie die Darstellung von einem Ausschnitt eines in einem Kontext, wie beispielsweise im Projekt, wahrgenommenen Umfeldes. Hierzu kommen ver- schiedene Modellierungssprachen zum Einsatz. Zur semantischen Beschreibung sind verschiedene Webservice-Technologien hilfreich, so z. B. unterstützende Proto- kolle wie http, SOAP usw. Es ﬁndet XML als einheitliches Datenformat Anwendung.

Complex Event Processing (CEP) ist ein Sammelbegriff für die Methoden, Techniken bzw. Werkzeuge, die es ermöglichen, Ereignisse zu verarbeiten, während sie erfol- gen. Die teuren Folgen verschiedener Formen von Cyberkriminalität werden für die Unternehmen zunehmend zum Problem. Die IT-Sicherheit wird immer wichtiger. Der Fokus richtet sich zunehmend auf einen proaktiven und professionellen Schutz. Zur Datensicherheit bedarf es unter anderem verlässlicher Standards. Im Zusammen- hang mit der Absicherung des Systems und insbesondere bei Webservices sind die folgenden Punkte zu beachten: die Absicherung der Übertragungswege bei Webser- vices (Verschlüsselung), ein Rollen- und Berechtigungskonzept, das Durchführen von Penetrationstests, die Protokollierung, das regelmäßige Einspielen sicherheits- relevanter Updates sowie die regelmäßige Überprüfung der Verfügbarkeit und Funk- tionalität der Schnittstelle.



# Lektion 8

## Qualität und Betrieb von Smart Services

#### LERNZIELE

Nach der Bearbeitung dieser Lektion werden Sie …

… die Qualitätseigenschaften und Reife von APIs kennen.

… das Wesen von Service Level Agreements verstehen.

… die Grundlagen von Service Level Management kennen und anwenden können.

DL-D-DLBINGSS01-L08

1. Qualität und Betrieb von Smart Services

### Einführung

Der Betrieb von Smart Services ist wesentlich durch den Aspekt der Qualität geprägt. Daraus leitet sich unter anderem ab, dass Qualitätsaspekte, deren Eigenschaften und die Reife von Programmierschnittstellen bzw. APIs thematisiert werden müssen. Zudem stellen auch Service Level Agreements bzw. Vereinbarungen zwischen Auftraggeber und IT-Dienstleister relevante Grundlagen dar, welche sich im praktischen Service Level Management fortsetzen.

### Qualitätseigenschaften und Reife von APIs

Im Kontext der Entwicklung zum digitalen Unternehmen spielt der Themenkomplex Programmierschnittstellen bzw. APIs eine zentrale Rolle. Heute werden moderne Soft- waresysteme bereits von Beginn an durch APIs unterstützt bzw. damit ausgestattet (vgl. Krafft 2016).

Unter einer Schnittstelle versteht man eine Verbindung zwischen zwei Systemen, die eine Kommunikation bzw. eine Übertragung ermöglicht, z. B. die Steckdose als Schnitt- stelle für die Übertragung von elektrischem Strom (vgl. Czernik 2016).

API-Schnittstellen Diese ermöglichen eine einfachere Ver- netzung mit anderen Systemen. Die Ver- netzung durch API- Technik ist insbe- sondere für die Automatisierung der Kommunikation not- wendig. Mittels einer API wird es möglich, eine Software in ein anderes System ein-

zubinden.

Die API-Schnittstellen erlauben eine einfache Vernetzung mit anderen Systemen. Vor allem ist diese Vernetzung von Systemen durch API-Technik für die Automatisierung der Kommunikation zwischen Systemen zwingend erforderlich (vgl. Krafft 2016). Unter API versteht man ein Akronym für „Application Programming Interface“ (englisch für: Schnittstelle zur Anwendungsprogrammierung) (vgl. Czernik 2016).

Schnittstellen sind im IT-Bereich sehr bedeutend. Durch Schnittstellen wird die Kom- munikation sowohl unter Software- als auch Hardwarekomponenten ermöglicht. Bei- spiel: Ein USB-Anschluss am Rechner stellt eine Hardwareverbindung zum USB-Stick bereit. Auch ein Betriebssystem enthält Schnittstellen. So ermöglichen die Schnittstel- len des Betriebssystems Zugriffe auf die Festplatte oder auch auf die Graﬁkkarte über ein Anwendungsprogramm (vgl. Czernik 2016).

APIs nehmen eine zentrale Rolle bei der Systemintegration ein, da es durch eine API möglich ist, eine Software in ein anderes System einzubinden. Weiter kann durch die Implementierung einer API ein Programm die Funktionalitäten einer anderen Software aufrufen und nutzen. Dadurch lassen sich unterschiedliche Softwaremodule/ Programmteile (Services) unabhängig voneinander erstellen, die mittels einer Schnitt- stelle miteinander kommunizieren. Eine derartige modulare Programmierung hat den Vorteil, dass Software einfacher zu warten ist. Mittels zusätzlicher Module lässt sich die Software individuell erweitern (vgl. Czernik 2016).

Qualität und Betrieb von Smart Services

###### Webservice/Web-API

Es wird zwischen einer gewöhnlichen API und einer Web-API (Webservice) unterschie- den. Auch ein Webservice beinhaltet eine API, jedoch wird durch diesen zusätzlich eine Webdienstleistung angeboten. Beispielsweise ist es mittels eines Webservices einem Webseitenanbieter möglich, unterschiedliche Dienste anzubieten. So kann beispiels- weise ein externer Anbieter über ein Netzwerk (Intranet/Internet) den Datenaustausch in seinem Portal integrieren und Dienste anbieten (vgl. Czernik 2016).

Es besteht keine einheitliche Deﬁnition für Webservices. Ein Webservice zeichnet sich durch die folgenden Eigenschaften aus (vgl. Czernik 2016):

* Er bietet einen Dienst über ein Netzwerk an.
* Er bietet einen automatisierten Datenaustausch sowie die Nutzung von Funktionali- täten an.
* Ein Datenaustausch sowie die Funktionalität erfolgen unabhängig von der Program- miersprache bzw. Hardware. Er kann dadurch in unterschiedlichen Systemen integ- riert sein.
* Er gilt als eine Computer-zu-Computer-Kommunikation.

Webservices, die über eine API Grundfunktionalitäten aus dem eigenen Produkt ande- ren Entwicklern zur Verfügung stellen möchten, stehen der Herausforderung der gefor- derten API-Qualität gegenüber. Die Qualität ist ein wesentliches Merkmal einer funkti- onierenden und nützlichen API. Grundlegend bringt eine API von guter Qualität Potenzial zur Unterstützung der Verbreitung eines Produkts. Flickr, Twitter oder Amazon erreichen eine große Marktdurchdringung auch durch eine hohe Anzahl an unter- schiedlichen Clients und ungeplanten Anwendungen. Durch die API sind wesentliche Umsätze möglich. Jedoch gilt es, in diesem Zusammenhang zu berücksichtigen, dass die Änderung einmal veröffentlichter APIs schwierig ist. Das Problem besteht darin, dass Applikationen Dritter dann gegebenenfalls nicht mehr richtig funktionieren oder sogar destruktiv sind. Es ist erforderlich, dass Interfaces sowie deren Dokumentation allen zugänglich sind. Sie sind entsprechend zu konzipieren. Bei einem interessanten Produkt ist es ohnehin nicht möglich, den Zugang zur Dokumentation zu kontrollieren (vgl. Law- renz 2009).

### Service Level Agreements (SLAs)

Für den langfristigen Erfolg von Dienstleistungsunternehmen sind die Kundenorientie- rung und ein professionelles Service Level Management (SLM) mit seinen entsprechen- den Service Level Agreements (SLAs) wesentliche Faktoren. Zur Professionalisierung von Schnittstellen zwischen Servicenehmer und IT-Dienstleister sind geeignete und prakti- kable Wege und Umsetzungsmöglichkeiten gefragt und auch erforderlich. Hierzu gilt es, SLAs zu entwerfen und zu überwachen (vgl. Scholderer 2016, S. 1ff., 4ff., 10ff., 14ff.).

Web-API (Webser- vice)

Auch ein Webservice beinhaltet eine API, jedoch wird durch diesen zusätzlich eine Webdienstleis- tung angeboten. Ein solcher Service steht der Anforderung der API-Qualität gegen- über.

Es ist erforderlich, dass die Informationstechnologie (IT) die von ihr zu erfüllende Funk- tionalität mit einer vorhersagbaren und garantierten Qualität erbringt. Nur mit einer Bereitstellung der Funktionalität im Kontext sogenannter IT-Services kann sichergestellt werden, dass der Servicerechner die gewünschte Qualität erhält und leistet bzw. dazu in die Lage versetzt wird oder werden kann, diese wie vorgesehen bzw. beabsichtigt zu nutzen. Für die qualitätsgesicherte Steuerung eines IT-Betriebs und die Bereitstellung von IT-Services bzw. IT-Service-Management (ITSM) bedarf es geeigneter Methoden und Modelle, wie z. B. (vgl. Scholderer 2016, S. 1):

* + - COBIT (Control Objectives for Information and Related Technology),
    - ISO 20000 bzw. ISO/IEC 20000,
    - ITIL (Information Technology Infrastructure Library).

Es bestehen verschiedene Deﬁnitionen des Begriffs des IT-Services (vgl. Scholderer 2016, S. 4ff.): Ein IT-Service …

* + - ... ist eine von mehreren Funktionen, die von IT-Systemen zur Unterstützung von Geschäftsfeldern zur Verfügung gestellt werden. Funktionen bestehen aus Software, Hardware und Kommunikationseinrichtungen.
    - … ist ein Zugang zu einer Anwendung (z. B. Reservierungssystem) oder über eine Reihe von Hard- und Softwareplattformen.
    - … ist eine Gruppe von Komponenten, die zur Unterstützung eines oder mehrerer Geschäftsprozesse vorgesehen ist. Der IT-Service umfasst Konﬁgurationselemente wie z. B. Router, Arbeitsplatz usw. Er ist aus der Perspektive von Servicenehmern und Nutzern eine Art geschlossene, einheitliche und kohärente Einheit.
    - … ist eine Dienstleistung für Kunden von Dienstleistern. Ein IT-Service basiert auf Einsatz von Informationstechnologie. Er unterstützt die Geschäftsprozesse eines Ser- viceunternehmens. Ein IT-Service besteht aus einer Kombination von Menschen, Prozessen und Technologie. Es ist erforderlich, dass der IT-Service in einem SLA deﬁ- niert wird.

Aus der Perspektive von Servicenehmern gehören zu Webservices neben der Leistungs- erbringung (Menschen, Prozesse und Technologie) (vgl. Scholderer 2016, S. 5ff.):

* + - Kommunikationsservices,
    - Systemservices,
    - Anwendungsservices,
    - Prozesse.

IT-Services zeichnen sich dadurch aus, dass sie für den Abnehmer in einer verständli- chen Form beschrieben sind sowie direkt via Nutzung der Ressourcen von vernetzten IT-Systemen realisiert werden (vgl. Scholderer 2001).

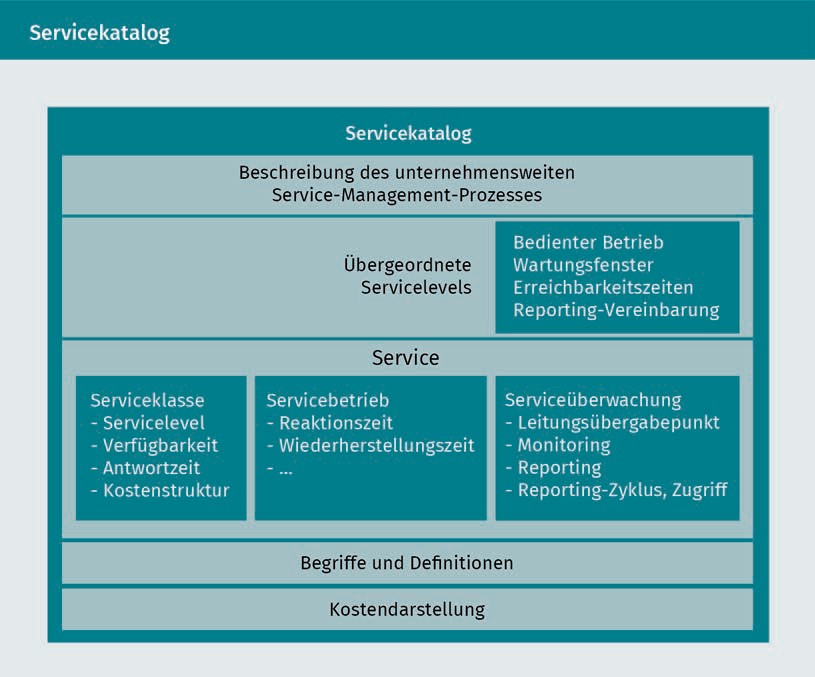
Qualität und Betrieb von Smart Services

###### Operational Level Agreements (OLAs)

Bei einem Operational Level Agreement (OLA) handelt es sich um eine Vereinbarung, die intern zwischen den Abteilungen eines Unternehmens getroffen wird. Bei großen Rechenzentren ist der interne Betrieb in der Regel in einzelne IT-Abteilungen aufgeglie- dert, so z. B. Servicebetrieb, Applikations-Hosting usw. Wenn gegenüber Endkunden beispielsweise Serviceleistungen angeboten werden, die sehr komplex sind, so setzen sich die entsprechenden IT-Services in der Regel aus verschiedenen IT-Abteilungen zusammen und es gilt, auch ITIL Operational Level Agreements zu deﬁnieren (vgl. Bon 2008; Scholderer 2016, S. 6).

Dies lässt sich durch ein Zusammenspiel von OLA und SLA über ein gemeinsames Tem- plate erreichen. Die Struktur eines IT-Servicekatalogs muss demnach im OLA enthalten sein. Die für das SLM verantwortliche Abteilung schließt ein OLA mit dem betreffenden Dienstleister ab (vgl. Scholderer 2016, S. 6f.).

Durch einen Servicekatalog werden IT-Services beschrieben, die ein Dienstleister sei- nem Servicenehmer anbietet (vgl. Scholderer 2016, S. 7). Die nachfolgende Abbildung verdeutlicht die Zusammenhänge.



###### Service Level (SL)

Ein Service Level (SL) quantiﬁziert einen IT-Service (Kütz 2010). Es drückt das Ergebnis aus, welches eine Eigenschaft der Dienstleistung charakterisiert. SL beinhaltet Ausdrü- cke mit einfachen, eindeutig interpretierbaren Ergebnissen, z. B. Kennzahlen. Die nach- folgende Abbildung zeigt ein Beispiel für Top-Ergebniskennzahlen (vgl. Scholderer 2016, S. 10).



Bei einem Service Level Agreement handelt es sich um einen Servicevertrag, der die Beziehung zwischen Servicenehmer und Dienstleister beschreibt und regelt (vgl. Schol- derer 2016, S. 11).

Die Erstellung und Überwachung hinsichtlich der Einhaltung des vereinbarten IT-Ser- vices werden verankert im ITIL-Prozess Service Level Management (vgl. Central Compu- ter and and Telecommunications Agency [CCTA] 2011; Scholderer 2016, S. 11).

### Service Level Management (SLM)

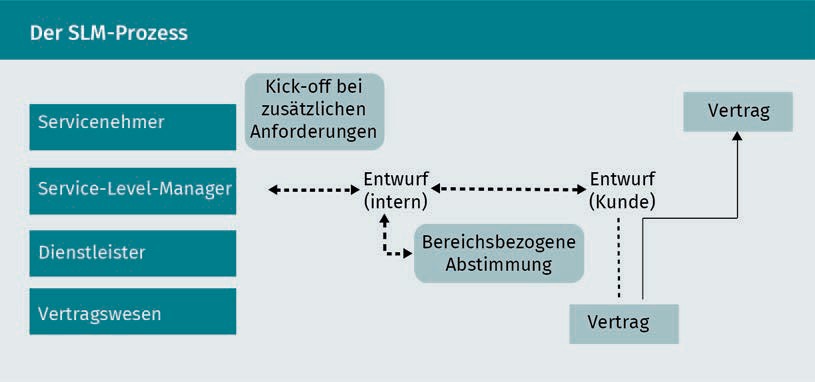
Unter Service Level Management (SLM) versteht man die Steuerung sowie die Sicher- stellung einer ausreichenden Servicequalität in Übereinstimmung mit Geschäftspriori- täten sowie akzeptablen Kosten. Der Zweck besteht in der Verbesserung der IT-Services für sämtliche IT-Benutzer (vgl. Scholderer 2016, S. 14).

Qualität und Betrieb von Smart Services

SLM ist ein kontinuierlicher Prozess zur Umsetzung des Servicegedankens und der Opti- mierung der Servicequalität (vgl. Scholderer 2016, S. 14). Der kontinuierliche Ablauf besteht in den folgenden vier Prozessen (vgl. Scholderer 2016, S. 14):

* Feststellung und Anpassung von Servicezielen sowie von Service-Level-Kennzahlen,
* Management der deﬁnierten Ziele bezüglich Leistung, Kosten und Qualität,
* Leistungsnachweis für erbrachte Serviceleistungen,
* Verbesserung wie auch Optimierung der Leistungsfähigkeit.

Im SLM-Prozess bestehen zahlreiche Rollen, mittels derer der IT-Service an Serviceneh- mer auszuliefern ist. Die wichtigsten Rollen im SLM-Prozess sind hierbei der Service- nehmer, der Service-Level-Manager und der Dienstleister. Häuﬁg sind Abstimmungen der erforderlichen bzw. beteiligten IT-Abteilungen notwendig. Dabei gilt es, Daten zu klären. Durch das SLM wird der Abstimmungsaufwand verringert. Dies wird möglich durch eine Ergänzung des Servicekatalogs mittels entsprechender Parameter (vgl. Scholderer 2016, S. 14f.). In der folgenden Darstellung werden die Zusammenhänge ver- deutlicht.



In der Abbildung zeigt sich der Ablauf des SLM-Prozesses. Dem Servicenehmer bietet sich die Möglichkeit der Auswahl aus einem Servicekatalog oder aus einem vorkonfekti- onierten Serviceangebot, aus welchem er seinen benötigten Service auswählt. Er kann aber auch individuell Service Level Requirements (SLR) formulieren. Dabei entsteht entweder direkt ein IT-Rahmenvertrag, der die Einordnung einzelner SLAs erlaubt, oder es erfolgt die Einberufung von Treffen, damit die Anforderungen seitens des Service- nehmers genau abgestimmt werden können (vgl. Scholderer 2016, S. 15).

Im Rahmen des SLM entsteht in der Aktivität „SLA-Eingang“ ein erster SLA-Entwurf für den Servicenehmer.

Zusammenfassung

Durch API-Schnittstellen wird eine einfachere Vernetzung mit anderen Systemen ermöglicht. Die Vernetzung durch API-Technik ist insbesondere für die Automatisie- rung der Kommunikation notwendig. Mittels einer API wird es möglich, eine Soft- ware in ein anderes System einzubinden. Auch ein Webservice beinhaltet eine API. Der Webservice bietet jedoch zusätzlich eine Webdienstleistung an. An einen Web- service wird auch die Anforderung der API-Qualität gestellt. Diese Anforderung gilt es zu erfüllen.

Ein Operational Level Agreement (OLA) ist eine Vereinbarung, die intern zwischen den Abteilungen eines Unternehmens getroffen wird. Mit einem Service Level (SL) wird ein IT-Service quantiﬁziert. Ein Service Level Agreement (SLA) ist ein Service- vertrag, der die Beziehung zwischen Servicenehmer und Dienstleister beschreibt und regelt. Als Service Level Management (SLM) bezeichnet man die Steuerung und Sicherstellung einer ausreichenden Servicequalität in Übereinstimmung mit Geschäftsprioritäten sowie akzeptablen Kosten zum Zwecke der Verbesserung der IT-Services für alle IT-Benutzer.