

Business Process Automation

Geschäftsprozesse nachhaltig automatisieren
mit BPMN und DMN

» Hier geht's
direkt
zum Buch

DIE LESEPROBE

Kapitel 4

Die Prozessgesteuerte Architektur

Aus der Architektur-Perspektive lassen sich die Bestandteile des Prozess-gesteuerten Ansatzes am besten erforschen.

Fragen, die dieses Kapitel beantwortet:

- Welche Prinzipien liegen der Prozessgesteuerten Architektur zugrunde?
- Welche Bestandteile umfasst die Prozessgesteuerte Architektur?

4.1 Übersicht

Die Prozessgesteuerte Architektur bringt klare Zuständigkeiten, Flexibilität, Transparenz, Qualität, Nachhaltigkeit und Effizienz. Somit kann sie in jede IT-Landschaft integriert werden, und vorhandene Funktionalitäten können wiederverwendet werden. Das heißt, ein Ziel der Prozessgesteuerten Architektur ist es, die Wiederverwendbarkeit der Funktionalität – der Fachprozesse oder des Fachprozesses – zu erreichen und nicht etwa die Wiederverwendbarkeit von Schnittstellen.

Der Fokus liegt auf der konsequenten Umsetzung der Geschäftsstrategie des Unternehmens und damit auf den fachlichen Prozessen, die zur Erreichung genau dieser Geschäftsstrategie beitragen. Alle systemspezifischen Details werden dazu in eine separate Schicht ausgelagert: in die *Servicevertrag-Implementierungsschicht*.

Wichtig ist: Der *Fachprozess bleibt in seiner Struktur möglichst unverändert*, also so, wie die Fachexperten ihn geschaffen haben, und auch die Kontrolle bleibt in der Verantwortung der Fachexperten (vgl. (Stiehl, 2013), S. 68–87). Dies hat folgende Vorteile: Die Fachexperten übernehmen Verantwortung für ihre fachlichen Prozesse, und damit erhöht sich auch die Akzeptanz der zukünftigen IT-Lösung.

Welche zentralen Architekturprinzipien und Entwurfsmuster liegen der Prozessgesteuerten Architektur zugrunde?

- **Separation of Concerns:** Dadurch werden Verantwortlichkeiten in einer Applikation getrennt. Einerseits werden die Verantwortlichkeiten der Fachlogik durch die PDA-Schicht und die externen systemspezifischen Details durch die Servicevertrag-Implementierungsschicht getrennt. Andererseits werden Persistenz, Ser-

vices, Entscheidungen, Benutzeroberflächen, Prozesse, Integration, Eventing etc. voneinander getrennt. Dies ermöglicht maximale parallele Entwicklung, sobald der Fachprozess mit dem Fachexperten erarbeitet wurde.

- Loose Kopplung:** Bei einer hohen *losen Kopplung* adressiert dieses Prinzip die Interaktionsfähigkeit von Systemen, ohne dass diese spezifische Details über die anderen Systeme kennen müssen. Die lose Kopplung erhöht zudem die Robustheit des Gesamtsystems und ist die Basis für Skalierbarkeit. Gleichzeitig erhöht sie die Flexibilität (leichte Austauschbarkeit einzelner Komponenten) und die Fehlertoleranz. Das Prinzip der losen Kopplung ist aufgrund seiner vielen Vorteile also ein wesentlicher Bestandteil der Prozessgesteuerten Architektur. Dies wird über die Servicevertrag-Implementierungsschicht zu den externen Systemen (Backend-Schicht) umgesetzt. Bei einer Änderung der Systemlandschaft muss lediglich in dieser Schicht die Verbindung zum konkreten System ausgetauscht werden. Die *Prozessgesteuerte Anwendung* bleibt davon unberührt.
- Adapter-Entwurfsmuster:** Dieses Muster hilft uns, wenn uns die fachlichen Anforderungen aus der *Prozessgesteuerten Anwendungsschicht* vorliegen und wir diese Anforderungen auf die systemspezifische Schnittstelle eines externen Systems abbilden müssen (vgl. auch (Starke, 2020), S. 131–132). Dadurch lässt sich in hohem Maße eine *lose Kopplung* zu Systemen erzielen.

Um das Ziel zu erreichen, besteht die *Prozessgesteuerte Architektur* aus drei Schichten (siehe Abbildung 4.1):

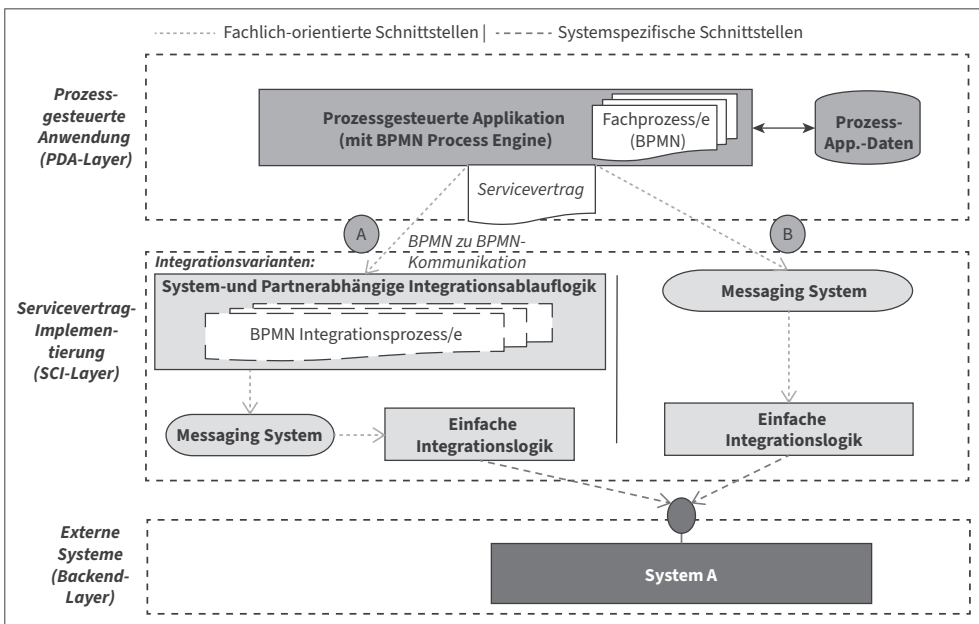


Abbildung 4.1: Die Prozessgesteuerte Architektur (in Anlehnung an (Stiehl, et al., 2019), S. 43)

1. aus der *Prozessgesteuerten Anwendung*, engl. *Process-driven Application (PDA)* (vgl. Abschnitt 4.2, »Die Schicht »Prozessgesteuerte Anwendung««).
Ein wichtiges Konzept ist der *Servicevertrag*, engl. *Service Contract (SC)* (vgl. Abschnitt 4.3, »Der Servicevertrag«). Er ist Bestandteil der Prozessgesteuerten Anwendung.
2. aus der *Servicevertrag-Implementierungsschicht*, engl. *Service Contract Implementation Layer (SCIL)*; vgl. Abschnitt 4.4, »Die Servicevertrag-Implementierungsschicht«)
3. aus der *Backend-Schicht*, engl. *Backend Layer*. Diese Ebene ist in den meisten Unternehmen vorhanden und wird durch die eigenen Anwendungssysteme oder *Microservices* repräsentiert. In ihnen liegen die Informationen und Funktionalitäten vor, auf denen die prozessgesteuerte Anwendung operiert. Der Backend Layer ist keine besondere Schicht im Rahmen dieser Architektur, sodass auf ihn nicht weiter eingegangen werde.

Entwickler und Architekten sollten verstehen, welche Deployment-Einheiten physisch entstehen. Dies beschrieb ich in Abschnitt 4.5, »Die physische Verteilungssicht«.

Zur Umsetzung der Prozessgesteuerten Architektur sind folgende Aspekte erforderlich:

- **Fundierte Anwendungskennnisse der Business Process Model and Notation (BPMN) und der Best Practices**, um die Prozessablauflogik zu beschreiben (vgl. Kapitel 2, »Einführung in die Prozessmodellierung mit BPMN 2.0.2 und DMN 1.3«)
- **Ein BPMN-basiertes Workflow-Management-System**, das sich in die Prozessgesteuerte Anwendung einbetten lässt. Dieses System stellt die Grundlage für die Ausführung der BPMN-basierten fachlichen Prozesse dar. Dabei ist es zentral, dass das Workflow-Management-System idealerweise sämtliche Elemente der BPMN-Palette zwecks der Modellierung von fachlichen Prozessen und der Integrationsprozesse unterstützt (vgl. Kapitel 1, »Einführung«).
- **Ein Decision-Management-System¹** zur Verwaltung und Implementierung von Entscheidungen. Ich empfehle, die *Decision Model and Notation (DMN)* als unabhängigen Standard zur Beschreibung von Entscheidungen zu verwenden (vgl. Kapitel 1 und Kapitel 2).
- **Eine Integrationssoftware bzw. ein Integrationsframework**, das allgemeine *Integration patterns* unterstützt, wie im Buch von (Hohpe & Woolf, 2004) beschrieben. Darüber lassen sich die externen Systeme im Unternehmen zentral anbinden.
- **Ein Messaging-System** zur Realisierung der losen Kopplung zu Anwendungssystemen und zur asynchronen (also zeitlich versetzten) Kommunikation.

¹ Einige kennen diese Art von Systemen vermutlich noch unter dem Begriff *Geschäftsregel-Management-Systeme* (engl. *Business Rule Management Systems*). Heute spricht man von *Decision Management Systems*. Es geht um fachliche Entscheidungen, die ausgeführt werden. Geschäftsregeln sind Bestandteil einer Entscheidung.

4.2 Die Schicht »Prozessgesteuerte Anwendung«

Die Schicht *Prozessgesteuerte Anwendung (PDA)* beinhaltet die *Fachprozesse* der Anwendung, also ihre fachliche Prozessablauflogik zu einer bestimmten *Domäne*. Eine Domäne ist ein fachliches Anwendungsgebiet (vgl. auch (Evans, 2004), S. 512). Anders ausgedrückt, handelt es sich bei den »Fachprozessen« in der Prozessgesteuerten Anwendung um system- und partnerunabhängige fachliche Ablauflogik.

Dabei kann eine Prozessgesteuerte Anwendung einen oder auch mehrere fachliche Prozesse umfassen. Die fachlichen Prozesse werden mithilfe der Fachexperten in BPMN beschrieben. Genau so, wie diese Prozesse grafisch und in syntaktisch korrekter sowie methodisch angewendeter BPMN beschrieben vorliegen, werden sie von dem BPMN-basierten Workflow-Management-System ausgeführt. Mehr zur richtigen Anwendung von BPMN finden Sie in Abschnitt 2.1, »Prozesse mit BPMN beschreiben«.

Zum Beispiel können zur Domäne *Human Resources* folgende fachliche Prozesse gehören: *Urlaub beantragen, Dienstreise beantragen, neue Mitarbeiter einstellen* etc.

Zudem hat jede prozessgesteuerte Anwendung genau eine Datenbank, in der die zur Anwendung gehörenden fachlichen Daten gespeichert werden.

Prozessgesteuerte Anwendung (Process-driven Application, PDA):

Eine PDA zeichnet sich dadurch aus, dass sie einen oder mehrere fachliche Prozesse unterstützt. Dabei handelt es sich um Ende-zu-Ende ausgerichtete fachliche Prozesse, die für das Unternehmen Wettbewerbsvorteile erzielen. Die im Unternehmen bereits vorhandenen Daten und Funktionalitäten in Anwendungssystemen werden wiederverwendet (vgl. (Stiehl, 2013), S. 24).

Eine PDA weist folgende Eigenschaften auf (vgl. (Stiehl, 2013), S. 24–29):

- Bei einer PDA liegt der Fokus der fachlichen Prozesse auf der Differenzierung von Mitbewerbern und auf der Erzielung von Wettbewerbsvorteilen. Es ist nicht das Ziel, Standardprozesse zu implementieren.
- Existierende Funktionalitäten werden wiederverwendet.
- Es werden keine Änderungen an den anzubindenden Anwendungssystemen vorgenommen.
- Eine PDA ist stets lose an die Anwendungssysteme gekoppelt, an die sie angebunden wird. Eine PDA arbeitet nur auf den Daten, die sie fachlichen benötigt. Das Datenmodell wird auf Basis der fachlichen Anforderungen ermittelt. Es sind keine systemspezifischen Daten enthalten.
- Im Rahmen einer PDA werden auch Benutzeraufgaben koordiniert. Zur Interaktion von Benutzern mit der PDA werden rollenspezifische Benutzeroberflächen genutzt (z. B. Weboberflächen).

Die fachlichen Prozesse der PDA werden mithilfe der BPMN in einem Detaillierungsgrad beschrieben, der ausreicht, um sie von einem BPMN-basierten Workflow-Management-System ausführen lassen zu können. Die zur PDA gehörende Funktionalität wird intern von der Anwendung implementiert. Bei der Implementierung ist die *komponentenbasierte Implementierung* von wesentlicher Bedeutung (vgl. dazu Kapitel 8, »Die Prozessgesteuerte Architektur implementieren«). Natürlich lassen sich auch Skripte (*Script Task*) im Rahmen der Prozessausführung durch das Workflow-Management-System verwenden (vgl. dazu Abschnitt 2.1.5, »Die Aufgabentypen«). Zudem lassen sich auch die zur Prozessgesteuerten Anwendung gehörenden Regeln in Geschäftsregel-Management-Systemen zur internen Implementierung zählen, sofern diese nicht global (zentral) bereitgestellt werden.

Checkliste: Was gehört zur PDA- oder zur Servicevertrag-Implementierungs-Schicht?

Die Frage, in welcher Schicht eine Anforderung einzuordnen ist, ist abhängig davon, **wer** die Anforderung oder Ablauflogik anfragt:

1. **Fall: Die Fachexperten:** Wenn die Anforderung oder Ablauflogik von den Fachexperten kommt, dann ist diese Anforderung oder Ablauflogik Teil des Fachprozesses und damit des PDA-Layers.

Beispiel: Im Rahmen des Beschaffungsprozesses wollen die Fachexperten die Bestellanfragen zu einem Produkt zunächst bis zu einem bestimmten Zeitpunkt sammeln, um erst dann die Bestellung konsolidiert an den Lieferanten abzuschicken. Damit wollen die Fachexperten einen Rabatt für die große Bestellmenge zu den jeweiligen Produkten erhalten.

2. **Fall: Ein System oder ein externer Partner:** Wenn die Anforderung oder Ablauflogik von einem System oder einem externen Partner verlangt wird, dann ist diese Anforderung oder Ablauflogik Bestandteil der Servicevertrag-Implementierungsschicht (SCI-Layer).

Beispiel: Im Rahmen des Beschaffungsprozesses erwartet der Lieferant, dass Ihr Unternehmen (als sein Kunde) zum Ende des Tages eine Sammelbestellung zu den erforderlichen Produkten schickt. Das heißt, er wünscht keine Einzelbestellungen, die aus verschiedenen Abteilungen Ihres Unternehmens über den Tag verteilt eintreffen.

Wir beleuchten die Erarbeitung der fachlichen Prozesse sowie des Servicevertrags auf Basis der fachlichen Prozesse in Kapitel 6, »PDA-Layer: Die Fachprozess-Sicht erarbeiten«, und anschließend die Implementierung in Kapitel 8, »Die Prozessgesteuerte Architektur implementieren«.



4.3 Der Servicevertrag

Externe Systeme oder globale Services haben niemals direkten Kontakt zur Prozessgesteuerten Anwendung (PDA; vgl. (Stiehl, 2021), S. 13).

Auf Basis der fachlichen Anforderungen des Prozesses werden die Anforderungen an die externen Systeme und globalen Services definiert. Diese Anforderungen an die Außenwelt werden im Servicevertrag festgehalten (vgl. (Stiehl, 2013), S. 105). Wichtig ist, dass die erforderlichen Daten im *Servicevertrag* auf dem fachlichen Datenmodell der PDA basieren (vgl. (Stiehl, 2013), S. 105–106).

Dies bedeutet, dass keine systemspezifischen Daten Einzug in die Beschreibung der fachlichen Prozesse der PDA finden (vgl. (Stiehl, 2013), S. 105–106). Um die Form, in der der Servicevertrag beschrieben wird, geht es in Kapitel 6, »PDA-Layer: Die Fachprozess-Sicht erarbeiten«.

Bei dem Servicevertrag handelt es sich um ein konzeptionelles Konstrukt. Der Servicevertrag der PDA ist von der IT über die Servicevertrag-Implementierungsschicht von den externen Systemen (Backend-Schicht) bereitzustellen (vgl. Abschnitt 4.4, »Die Servicevertrag-Implementierungsschicht«). Zu beachten ist allerdings, dass bei den Services des Servicevertrags nicht primär die Wiederverwendbarkeit im Fokus steht (vgl. (Stiehl, 2013), S. 106).

Im folgenden Kasten fasse ich die zentralen Merkmale des Servicevertrags einer prozessgesteuerten Anwendung zusammen (vgl. (Stiehl, 2013), S. 359):

Der Servicevertrag (Service Contract, SC)

- Der Servicevertrag wird auf Basis der fachlichen Anforderungen der *Prozessgesteuerten Anwendung (PDA)* ermittelt.
- Er beinhaltet die relevanten Felder aus Perspektive der prozessgesteuerten Anwendung (PDA). Dabei basieren die Definitionen der Felder auf dem fachlichen Datenmodell der PDA.
- Die Schnittstellen des Servicevertrags haben nicht den Anspruch auf Wiederverwendbarkeit, sondern liefern fachliche Anforderungen. Dabei repräsentieren die Schnittstellen die bidirektionalen fachlichen Anforderungen der PDA – d. h. *was die PDA braucht* und *was die PDA zur Verfügung stellt*.

Zwischen den einzelnen Schichten der Prozessgesteuerten Architektur gelten im Kontext des Servicevertrags folgende Beziehungen (vgl. (Stiehl, 2013), S. 106, und siehe Abbildung 4.2):

1. *Eine prozessgesteuerte Anwendung besitzt maximal einen Servicevertrag*. Sie kann aber auch keinen Servicevertrag besitzen, wenn die Prozessgesteuerte Anwendung keine vorhandene Funktionalität verwendet (1:1).

2. Ein Servicevertrag umfasst mindestens einen Service (1:n).
3. Ein Service umfasst mindestens eine Schnittstellenoperation (1:n).
4. Abhängig von der Systemlandschaft kann ein Servicevertrag einer PDA mehrere Implementierungen besitzen. Dies ist davon abhängig, ob ein und dieselbe PDA in unterschiedlichen Systemlandschaften eingesetzt werden soll.

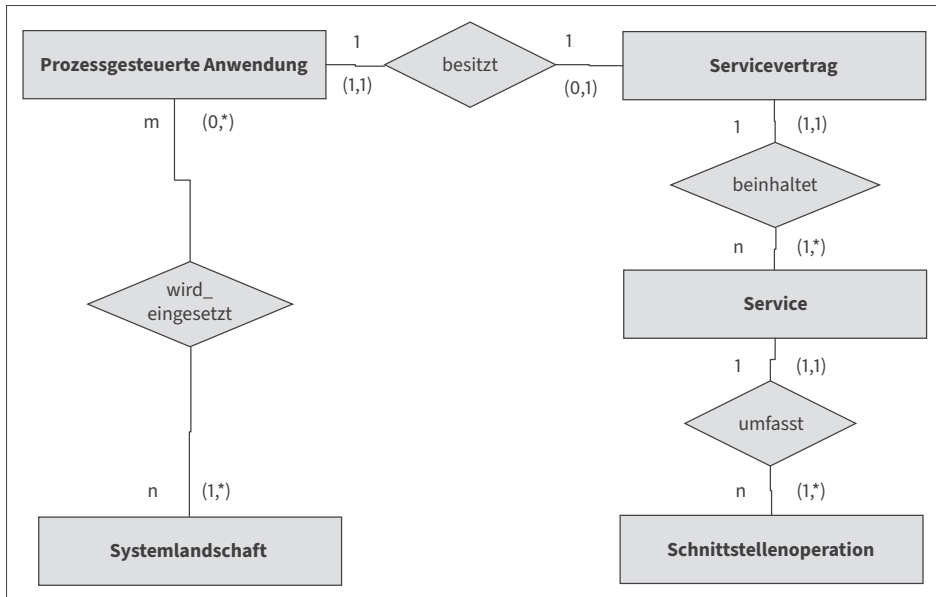


Abbildung 4.2: Beziehungen der Schichten/Konzepte der Prozessgesteuerten Architektur

4.4 Die Servicevertrag-Implementierungsschicht

Nachdem die Prozessgesteuerte Anwendung erarbeitet wurde, ist aus fachlicher Perspektive ein Bedarf an Informationen aus einem oder mehreren in der Systemlandschaft vorhandenen Anwendungssystemen entstanden. Des Weiteren kann die Anforderung auch von einem externen Partnersystem erbracht werden. Diese fachliche Schnittstellenanforderung wird im Servicevertrag der Prozessgesteuerten Anwendung festgehalten (vgl. Abschnitt 4.3, »Der Servicevertrag«).

Theoretisch können Sie die benötigten Anwendungssysteme aus einer Systemlandschaft auch direkt in die Prozessgesteuerte Anwendung einbinden. Viele Hersteller von Workflow-Management-Systemen werben mit ihrer umfangreichen Unterstützung von spezifischen System-Konnektoren und der Einfachheit der Anbindung. Leider fallen viele Menschen in Unternehmen auf diese Werbung herein und verwenden direkt diese Konnektoren aus ihren Fachprozessen heraus. Damit ist zwar die Erstimplementierung schnell umgesetzt, doch zukünftige Wartungen und Weiterentwicklungen der Anwen-

dung² werden erschwert, und das Unternehmen wird in seiner Agilität und Flexibilität deutlich gebremst oder sogar blockiert.

Denn stellen wir uns vor, 1000 Anwendungen nutzen einen ChatGPT-Konnektor, um bestimmte Erkenntnisse im Rahmen der Anwendung zu verarbeiten. Sie nutzen dieselben technischen Schnittstellen. Nun stellen wir uns vor, dass der Anbieter das Lizenzmodell anpasst und die Preise enorm erhöht. Also entscheidet sich das Unternehmen, diesen Service gegen einen anderen auszutauschen. Daraufhin beginnt die IT mit dem umfangreichen Migrationsprojekt des bisherigen *ChatGPT*-Service auf den anderen Service. Während dieses ohnehin schon umfangreichen Migrationsprojektes muss die IT natürlich noch ihren anderen Anforderungen gerecht werden. So muss sie

- die Anpassungen vorhandener fachlicher Prozesse an geänderte Rahmenbedingungen durchführen und
- sich um die Neuentwicklung fachlicher Prozesse zur Verbesserung der Wettbewerbsfähigkeit kümmern.

Die Konsequenz ist, dass die IT während des Migrationsprojekts nicht handlungsfähig ist. Denn die meisten Ressourcen stecken ja im umfangreichen Migrationsprojekt. Dazu gehört auch, dass die oben beschriebenen technischen Schnittstellen zum bisherigen *ChatGPT*-Service in den 1000 Anwendungen gegen die technische Schnittstelle zum neuen entsprechenden Service ausgetauscht werden müssen.

Man kann diese Geschichte noch weiterspinnen: Dies alles verschlingt enorme Ressourcen, nur um das Unternehmen weiterhin betriebsfähig zu halten. Auf neue Anforderungen kann das Unternehmen in einer solchen Phase ohnehin nicht mehr eingehen. Dementsprechend sollte nachvollziehbar sein, dass die direkte Anbindung von Systemen aus BPMN-basierten Fachprozessen kontraproduktiv ist: Sie erschweren die Wartung, behindern die Flexibilität bei der Anpassbarkeit der Prozesse, wenn fachliche Änderungen eintreten, und schränken zusätzlich die Handlungsfähigkeit der IT ein.

Zu allem Übel kommt hinzu, dass durch eine solche Vorgehensweise auch die komplexen Schnittstellen des Zielsystems in den Fachprozess eingebracht werden, was eine Menge Mapping-Aufgaben zur Folge hat. Das ist nicht die Aufgabe eines Fachprozesses, und die Lesbarkeit sowie Verständlichkeit des Fachprozesses werden für die Prozessbeteiligten erschwert.

Nachfolgend fasse ich die Merkmale der Servicevertrag-Implementierungsschicht in einem Kasten zusammen (vgl. (Stiehl, 2013), S. 109 und S. 111):

2 Anmerkung: In diesem Fall ist die Anwendung keine »Prozessgesteuerte Anwendung«, da sie nicht deren Prinzipien folgt.

Die Servicevertrag-Implementierungsschicht (engl. **Service Contract Implementation Layer, SCIL**)

Die Servicevertrag-Implementierungsschicht übernimmt die Implementierung des Servicevertrags in einer konkreten Systemlandschaft. Dies umfasst die Integrationsprozesse zu den jeweiligen Anwendungssystemen.

Die SCI-Schicht ist technologieabhängig; sie übernimmt insbesondere das Mapping und die Transformation des fachlichen Datenmodells von der prozessgesteuerten Anwendung auf die jeweiligen systemspezifischen Protokolle, Datentypen und -strukturen der Anwendungssysteme.

Die SCI-Schicht übernimmt die nachfolgenden Aufgaben gemäß *VETRO-Pattern* (vgl. auch (Juluri, 2018)):

- **Validate:** Die erhaltene Nachricht wird validiert, ob sie den vereinbarten »Vertrag« oder die vereinbarte Struktur erfüllt.
- **Enrich:** Gegebenenfalls werden Informationen in der Nachricht für den externen Service bzw. das externe System ergänzt.
- **Transform:** Die Nachricht wird vom Quellformat in das Datenformat des Ziel-Systems bzw. Ziels-Service umgewandelt.
- **Route:** Es erfolgt die Weiterleitung der transformierten Daten an den externen Service bzw. an das externe System.
- **Operate:** Dies umfasst die transaktional gesicherte physische Übertragung über gegebenenfalls systemspezifische Konnektoren in die jeweiligen Zielsysteme.

Die Servicevertrag-Implementierungsschicht umfasst sämtliche systemspezifischen Aspekte wie Mapping und Transformation vom fachlichen Datenmodell der Prozessgesteuerten Anwendung zum systemspezifischen Datenmodell des Anwendungssystems. Ein Anwendungssystem wird immer über die Servicevertrag-Implementierungsschicht an die Prozessgesteuerte Anwendung angebunden. Dies gilt für beide Richtungen – also auch, wenn ein bereits existierendes IT-System (z. B. ein Chatbot) einen fachlichen Prozess innerhalb der Prozessgesteuerten Anwendung anstößt oder fortsetzt.

Um sie zu implementieren, kennt die Prozessgesteuerte Architektur zwei *Integrationsvarianten*. Diese sollen nachfolgend kurz vorgestellt werden (siehe Abbildung 4.1):

1. System- und partnerabhängige Integrationsablauflogik (optional): Dieser Bestandteil der Integration ist *optional*. Er ist nur erforderlich, wenn mindestens zwei Systemaufrufe betroffen sind. Dabei spielt es keine Rolle, ob die beiden Aufrufe an dasselbe System oder an unterschiedliche Systeme geschickt werden.

In Integrationszenarien, in denen die Integrationsablauflogik sowohl mit BPMN als auch mit der Integrationssoftware realisiert wird, ist die BPMN-basierte Variante zu bevorzugen. Denn mittels BPMN schaffen Sie Transparenz, was Ihre Integrationsablauflogik betrifft. Wir verzichten nur aus Performancegründen auf diesen BPMN-basierten

Integrationsprozess, wenn das WfMS die Masse der Nachrichten nicht verarbeiten kann. In diesem Fall wird auch die Integrationskomponente verwendet.

Wenn zur Umsetzung einer fachlichen Schnittstellenoperation lediglich ein externer Systemaufruf notwendig ist, greifen Sie auf die *einfache Integrationslogik* zurück (siehe den nächsten Punkt).

2. **Einfache Integrationslogik:** Diese Integrationsvariante ist immer vorhanden und verpflichtend als Bestandteil der Implementierung der Servicevertrag-Implementierungsschicht. Sie übernimmt die Integrationsaufgaben gemäß VETRO-Pattern (siehe die Infobox »Die Servicevertrag-Implementierungsschicht« oben). Somit kann die einfache Integrationslogik als alleinige Integrationsvariante oder als eine weitere Variante zusammen mit der *system- und partnerabhängigen Integrationsablauflogik* bestehen.

Auf die Details dieser beiden Integrationsvarianten gehe ich näher in Abschnitt 6.6.2, »Vorhandene Funktionalität aus der Systemlandschaft anfordern«, und in Kapitel 7, »SCI-Layer: Die Integrationssicht beschreiben«, ein. Die Implementierung selbst wird in Kapitel 8, »Die Prozessgesteuerte Architektur implementieren«, beschrieben.

4.5 Die physische Verteilungssicht

In der physischen Verteilungssicht spreche ich von *Deployment-Einheiten*. Wie diese letztendlich physisch aufgesetzt werden, spielt keine Rolle: So können alle Komponenten physisch auf einem Server oder auf unterschiedlichen Servern deployt werden. Meistens werden sie beispielsweise in eigenen *Containern* (z. B. mit Docker) aufgesetzt.

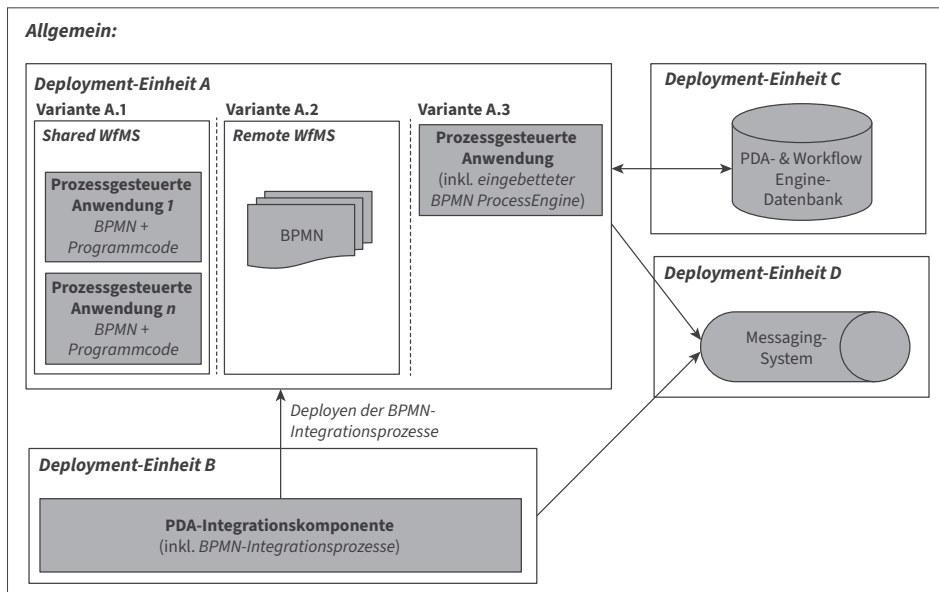


Abbildung 4.3: Die allgemeine Deployment-Sicht der Prozessgesteuerten Architektur

In den folgenden Abschnitten erläutere ich, wie sich die einzelnen Einheiten der Prozessgesteuerten Architektur physisch bereitstellen lassen (siehe dazu auch Abbildung 4.3).

4.5.1 Deployment-Einheit A »Prozessgesteuerte Anwendung«

Bei der Prozessgesteuerten Anwendung gibt es **verschiedene Varianten**, abhängig von der Art und Weise, wie das Workflow-Management-System im Verbund mit der Prozessgesteuerten Anwendung verwendet wird:

1. Variante A.1: Shared Workflow-Management-System

Bei dieser Variante haben Sie ein zentrales Workflow-Management-System für unterschiedliche prozessgesteuerte Anwendungen. Das Workflow-Management-System wird wie ein Applikationsserver eingesetzt. Dabei kann es sich um einen Cloud- oder einen On-Premises-Applikationsserver handeln. Die einzelnen Prozessgesteuerten Anwendungen mit jeweils BPMN und zusätzlichem Programmiercode (z. B. für Datenhaltung, eigene zur Prozessgesteuerten Anwendung gehörende fachliche Komponenten – jedoch keine programmierte Prozessablauflogik) werden in dem *Shared Workflow-Management-System* deployt. Es ist entscheidend, dass die Ablauflogik der Prozesse durch das zentrale Workflow-Management-System zur Verfügung gestellt wird. Darum muss sich der Programmierer nicht mehr kümmern. Hier lassen sich in den BPMN-Service-Tasks die fachlichen Komponenten injizieren (vgl. Kapitel 8, »Die Prozessgesteuerte Architektur implementieren«).

2. Variante A.2: Remote Workflow-Management-System

Bei der Variante mit einem *Remote Workflow-Management-System* lässt sich kein zusätzlicher Programmcode zum Aufruf der Geschäftslogik mit deployen (d. h. beispielsweise, innerhalb einer BPMN-Service-Task kann keine Anwendungslogik aufgerufen werden). Es kann lediglich der fachliche Prozess in BPMN deployt werden. Die zugehörige Implementierung der Fachlogik, z. B. zu einer BPMN-Service-Task, erfolgt außerhalb des Workflow-Management-Systems (WfMS). Dabei geht der PDA-Ansatz davon aus, dass jedes WfMS seinen eigenen Mechanismus bereitstellt, um die Prozessaufgabe außerhalb zu implementieren und die Interaktion mit dem Workflow-Management-System zu ermöglichen (wie beispielsweise bei *5Minds ProcessCube* oder der *Camunda-BPM-Plattform* mit den *External Task Clients* für die verschiedenen Programmiersprachen). Dabei wird in einer separaten physischen Komponente zur Prozessgesteuerten Anwendung für die Implementierung der fachlichen Aufgabenlogik die Geschäftslogik als separate Klasse implementiert. Diese Geschäftslogik-Klasse rufen Sie mit dem jeweiligen Mechanismus des Workflow-Management-Systems (WfMS) auf, um mit dem WfMS zu interagieren: zum Beispiel für das Abrufen der Prozessvariablen und für die Mitteilung, dass die Aufgaben erledigt sind (siehe Abbildung 4.4, *Fall A »neue Funktionalität«*). In dem Fall, dass Sie eine vorhandene Funktionalität wiederverwenden können (siehe Abbildung 4.4, *Fall B »Wiederverwendung existierender Funktionalität«*), würde ich hier die Informationen gemäß Servicevertrag an ein Messaging-

System senden und damit über die SCI-Schicht kommunizieren. Die genaue Implementierung erläutere ich in Kapitel 8, »Die Prozessgesteuerte Architektur implementieren«. Abbildung 4.4 veranschaulicht die einzelnen physischen Einheiten für den Fall eines Remote Workflow-Management-Systems.

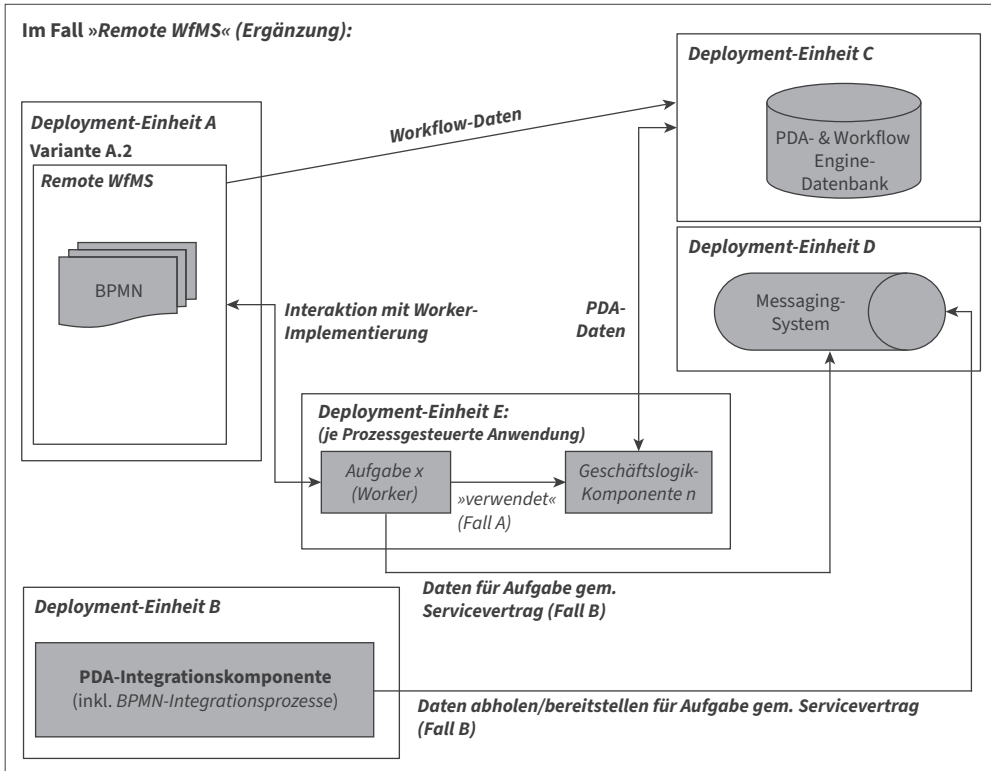


Abbildung 4.4: Die spezifische Deployment-Sicht der Prozessgesteuerten Architektur bei Verwendung eines Remote Workflow-Management-Systems

3. Variante A.3: Eingebettete Process Engine

Hierbei ist die Process Engine in die Prozessgesteuerte Anwendung »eingebettet«. Mit »eingebettet« meine ich, dass die Process Engine wie eine Bibliothek eingebunden wird. Das bedeutet, für jede prozessgesteuerte Anwendung haben Sie eine eigene Process Engine. Damit haben Sie (ähnlich wie mit einem Microservice) eine kleine eigene deploybare Einheit: Die Anwendung und die Process Engine stellen zusammen eine physische Einheit dar.

4.5.2 Deployment-Einheit B: PDA-Integrationskomponente

Bei dieser Einheit handelt es sich um eine eigene Komponente für die Integration. Hier kann alles eingesetzt werden, was die Integration der PDA mit der »Außenwelt« gemäß Servicevertrag ermöglicht.

Dabei werden mithilfe dieser Integrationskomponente die zwei Integrationsvarianten abgebildet (vgl. Abschnitt 4.4, »Die Servicevertrag-Implementierungsschicht«):

1. system- und partnerabhängige Integrationsablauflogik
2. einfache Integrationslogik

Im Fall der *system- und partnerabhängigen Integrationsablauflogik* gehört der BPMN-Integrationsprozess zu dieser Integrationskomponente. Damit aber der fachliche Prozess in BPMN innerhalb des Workflow-Management-Systems der prozessgesteuerten Anwendung mit dem BPMN-basierten Integrationsprozess kommunizieren kann, muss dieser im selben Workflow-Management-System wie der fachliche Prozess deployt werden. Dazu wird aus der Integrationskomponente heraus zum Deployment des BPMN-Integrationsprozesses an das Workflow-Management-System der prozessgesteuerten Anwendung ein Schnittstellenaufruf realisiert, um den BPMN-Integrationsprozess in das Workflow-Management-System der PDA zu deployen.

In diesem Fall nutzen wir die BPMN-zu-BPMN-Kommunikation eines BPMN-basierten Workflow-Management-Systems (WfMS). Wenn Prozesse per BPMN Message Events oder BPMN-Sende- und Empfangsaufgaben kommunizieren, ist es einfach, innerhalb des WfMS die Kommunikation zwischen Prozessen umzusetzen. Da beide Prozesse unter der Kontrolle ein und desselben Workflow-Management-Systems stehen, muss der Hersteller dieses Systems die transaktional gesicherte Abwicklung beider Operationen (senden, empfangen) sicherstellen. Tritt bei der Kommunikation ein Fehler auf, muss der Zustand *vor* dem Versand der Nachricht wiederhergestellt und eine Fehlernachricht an den Betrieb generiert werden. Der Hersteller muss also eine sogenannte *Transaktionsklammer* für die betroffenen Prozesse einhalten.

Was bedeutet das konkret? Aus dem einen Prozess muss der Sende-Schritt sauber abgearbeitet werden, und der andere Prozess muss die Information korrekt empfangen. Das sind zwei zusammengehörige Aktionen, die gemeinsam erfüllt sein müssen (siehe Abbildung 4.5):

- 1. Aktion: Die *Senden*-Information kommt aus Prozess A und
- 2. Aktion: Prozess B muss die Information am Starterereignis *empfangen* sowie den Prozess instanziiieren.

Wenn innerhalb der Aktion 1 bei Prozess A ein Fehler auftritt, darf Prozess B den Prozess nicht instanziiieren. Der Prozess A muss bei einem Fehler anhalten und einen Fehler melden. Wenn der Fehler behoben wird, muss die Aktion 1 neu gestartet werden. Mit anderen Worten: Entweder gelingt dieser Schritt ganz oder gar nicht.

Wenn man den Fach- und den Integrationsprozess physisch auf unterschiedlichen WfMS ablaufen lassen würde, müsste die korrekte Ausführung beider Aktionen überflüssigerweise durch den PDA-Entwickler sichergestellt werden. Das ist zusätzlicher Aufwand, den wir uns durch den Einsatz eines WfMS sparen können.

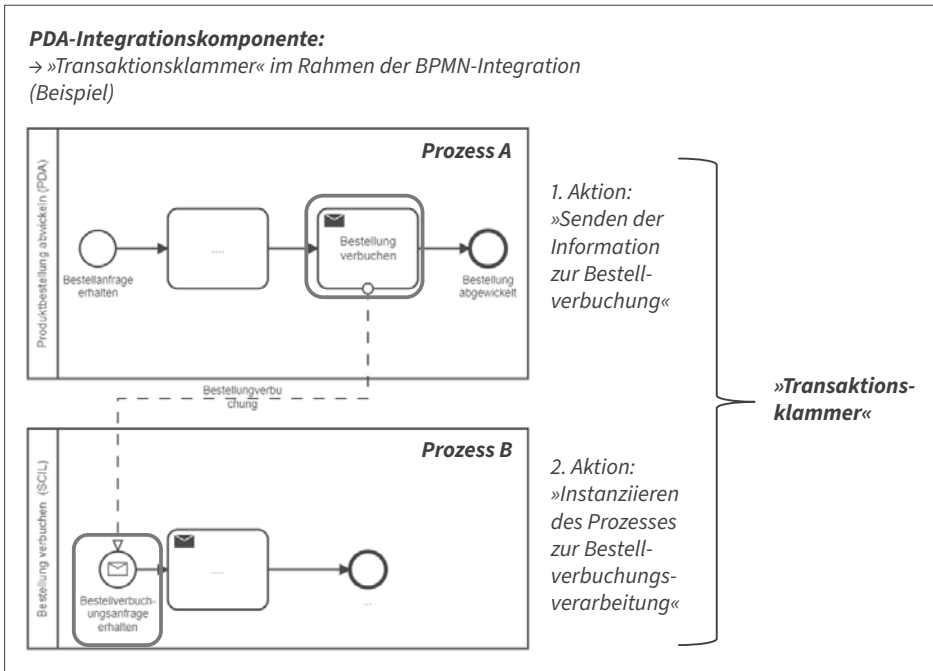


Abbildung 4.5: Bedeutung der »Transaktionsklammer« im Rahmen der BPMN-Integration (Beispiel)

Bei der system- und partnerabhängigen Integrationsablauflogik ist der Einsatz des BPMN-basierten Integrationsprozesses die bevorzugte Variante. Wir verzichten nur aus Performancegründen auf diesen BPMN-basierten Integrationsprozess – wenn also das WfMS die Masse der Nachrichten nicht mehr verarbeiten kann. In diesem Fall wird auch für diese Integrationsablauflogik die Integrationskomponente verwendet. Beim Einsatz des BPMN-basierten Integrationsprozesses liegt der Vorteil auf der Hand: Man erhält Transparenz über die Integrationsablauflogik, da diese mit BPMN grafisch ersichtlich wird. Für die »letzte Meile« der Integration ist stets die Integrationskomponente einzusetzen, da es sich bei ihr um Spezialsoftware handelt, die für derartige Integrationsaufgaben optimiert wurde.

Optional können auch die jeweiligen BPMN-Prozessdefinitionen für die Integration mit ihrem ergänzenden Implementierungscode von den zugehörigen einfachen Integrationslogiken der PDA-Integrationskomponente physisch durch separate Komponenten getrennt werden. Bei der Trennung in separate Komponenten meine ich nicht unbedingt die Aufteilung auf zwei physische Rechner. Stattdessen kann es wichtig sein, die umfangreiche Implementierung der Integrationsablauflogik innerhalb des BPMN-Integrations-

prozesses von der einfachen Integrationslogik innerhalb der Integrationskomponente physisch zu trennen. Die Integrationskomponente übernimmt weiterhin die Integrationsaufgaben nach dem VETRO-Pattern (vgl. Abschnitt 4.4, »Die Servicevertrag-Implementierungsschicht«). Ob die separaten Komponenten nun auf zwei unterschiedlichen Rechnern oder nur virtualisiert in zwei verschiedenen Containern deployt werden, ist unerheblich. Dabei handelt es sich bei dieser Trennung um einen *Tradeoff*: Einerseits lässt sich durch diese Trennung das Prinzip des *Separation of Concerns* konsistent umsetzen (vgl. Abschnitt 4.1, »Übersicht«), andererseits werden dadurch weitere kleinteilige Komponenten erzeugt.

Meiner Meinung nach ist diese weitere Zerlegung nur dann sinnvoll, wenn der BPMN-Integrationsprozess umfangreichere Integrationslogik umfasst, beispielsweise im Rahmen von Kompensationen bei mehreren angeordneten Systemen. Das bedeutet. Wenn es möglich sein soll, dass eine Änderung durch die Prozessgesteuerte Anwendung in einem externen System auch rückgängig gemacht werden kann (z. B. eine Bestellung wurde in einem externen System verbucht und diese soll storniert werden können), dann muss für diesen Fall weitere Logik und eine spezifische Datenbank zur Vorhaltung der Information vorgesehen werden. Diese muss die Zuordnung der konkreten PDA-Prozessinstanzen zu den externen Systemen speichern, in denen eine Änderung vorgenommen worden ist (vgl. dazu (Stiehl, 2013), S. 118 und S. 125). Auf dieser Basis kann im Fall einer fachlichen Ausnahme oder eines Fehlers (z. B. Stornierung einer Bestellung) die Wiederherstellung der Zustände vor der Verbuchung in den jeweiligen Anwendungssystemen angestoßen werden.

4.5.3 Deployment-Einheit C »PDA- und Workflow-Engine-Datenbank«

Diese Deployment-Einheit umfasst den separat betriebenen Datenbankserver, der aufgrund dieser Trennung unabhängig skaliert werden kann. Der Datenbankserver umfasst die separaten Datenschemata für die prozessgesteuerte Anwendung und die *BPMN-Process-Engine* (also die *Workflow-Engine*). Dieser eine Datenbankserver kann auch die verschiedenen Datenschemata verschiedener prozessgesteuerter Anwendungen aufnehmen. Alternativ kann es erforderlich sein, dass die Daten der Prozessgesteuerten Anwendung und der BPMN-Process-Engine jeweils auf physisch getrennten Datenbanksystemen laufen.

4.5.4 Deployment-Einheit D »Messaging-System«

Hiermit ist das Messaging-System zur Realisierung der losen Kopplung und der asynchronen Kommunikation zwischen der Prozessgesteuerten Anwendung und der PDA-Integrationskomponente gemeint.

4.6 Zusammenfassung

Die zu Beginn des Kapitels formulierten Fragen können nun wie folgt beantwortet werden:

Welche Prinzipien liegen der Prozessgesteuerten Architektur zugrunde?

- Oberstes Ziel: Die Prozessgesteuerte Architektur bringt klare Zuständigkeiten, Flexibilität, Transparenz, Qualität, Nachhaltigkeit und Effizienz. Dabei ist ein Ziel die Wiederverwendbarkeit der Funktionalität, nicht die Wiederverwendbarkeit von (technischen) Systemschnittstellen. Damit steht die Ausrichtung auf die Erreichung der Geschäftsziele des Unternehmens im Fokus.
- Der Fachprozess bleibt in der Struktur, in der die Fachexperten ihn definiert haben. Dies sichert Wiedererkennung, Akzeptanz und eine gemeinsame Kommunikationsbasis im Hinblick auf die fachliche Prozessablauflogik zwischen Fachexperten und IT.
- Die fachlichen Prozesse (d. h. die fachliche Ablauflogik) werden mithilfe des BPMN-Standards so präzise beschrieben, dass sie von einem Workflow-Management-System ausgeführt werden können.
- Entscheidungen werden über ein Decision-Management-System verwaltet und ausgeführt.
- Zur Integration in die existierende Anwendungslandschaft empfiehlt sich eine Kombination aus Messaging und spezialisierter Integrationssoftware. Das Messaging-System entkoppelt die Prozessgesteuerte Anwendung von der Systemlandschaft und ermöglicht eine asynchrone Kommunikation. Die spezialisierte Integrationssoftware ermöglicht uneingeschränkte Kommunikation in heterogenen Systemumgebungen und befreit das Workflow-Management-System von derartigen überflüssigen Erweiterungen, z. B. von spezifischen Konnektoren zu den Anwendungssystemen.
- Die Prozessarchitektur basiert auf den folgenden zentralen Prinzipien und Entwurfsmustern: *Separation of Concerns*, lose Kopplung zu den Anwendungssystemen sowie Nutzung des Adapter-Entwurfsmusters.
- Fachprozesse kommunizieren nie direkt mit Anwendungssystemen. Diese Forderung gilt für beide Richtungen: sowohl vom Fachprozess zum Anwendungssystem als auch vom Anwendungssystem zum Fachprozess.

Welche Bestandteile umfasst die Prozessgesteuerte Architektur?

Die Prozessgesteuerte Architektur besteht aus den folgenden Bestandteilen und Konzepten (vgl. Abbildung 4.1):

1. **Prozessgesteuerte Anwendung** (vgl. Abschnitt 4.2): Hier wird der Fachprozess der Anwendung in BPMN beschrieben. Dabei liegt der Fokus auf der reinen Prozessablauflogik der Fachexperten. Demnach wird auf systemspezifische Informationen der Anwendungssysteme verzichtet.

2. **Servicevertrag** (vgl. Abschnitt 4.3): Fachliche Prozesse innerhalb der Prozessgesteuerten Anwendung haben niemals direkten Kontakt zu den Systemen und umgekehrt. Aus den fachlichen Anforderungen des Fachprozesses werden die zu erbringenden Anforderungen an die Anwendungssysteme formuliert.

Der Servicevertrag zu einer prozessgesteuerten Anwendung muss von der Systemlandschaft erfüllt werden, um die Geschäftsziele des Unternehmens zu erreichen.

3. **Servicevertrag-Implementierungsschicht** (vgl. Abschnitt 4.4): Diese Schicht implementiert den Servicevertrag der prozessgesteuerten Anwendung in einer konkreten Systemlandschaft des Unternehmens. Sie ist hochgradig technologieabhängig und umfasst die Integrationsprozesse zwischen Prozessgesteuerter Anwendung und der Backend-Schicht.
4. **Backend-Schicht**: Diese Schicht beinhaltet die Anwendungssysteme aus der Systemlandschaft des Unternehmens. Sie stellt keine besondere Schicht im Rahmen dieser Architektur dar, sodass ich nicht weiter auf sie eingegangen bin.
5. **Physische Verteilungsschicht**: Abschließend bin ich in Abschnitt 4.5 auf die physische Verteilungsschicht eingegangen und habe beschrieben, wie die einzelnen Artefakte der prozessgesteuerten Architektur physisch bereitgestellt werden.

Kapitel 5

Prozesse erheben

Die Prozesse sind zumeist in den Köpfen der verschiedenen Prozessteilnehmer zu finden. Nun gilt es, die einzelnen Informationen aus den Köpfen zu »externalisieren« und wie ein Puzzle zu einem Prozess zusammenzusetzen.

Frage(n), die dieses Kapitel beantwortet:

- Wie lassen sich fachliche Prozesse in einem Prozesserhebungsworkshop mit den Fachexperten erarbeiten?

Damit fachliche, operative Prozesse automatisiert werden können, müssen sie erst einmal transparent vorliegen. Dies gilt sowohl für IST-Prozesse, die zumeist nicht aktuell dokumentiert sind, und ganz besonders für zukünftige Prozesse (SOLL-Prozesse). Dabei ist es wichtig, zu verstehen, dass fachliche Prozesse verschiedene »Silos« überspannen (z. B. Abteilungen, Organisationen), sodass in die Erhebung Wissen aus den unterschiedlichsten Bereichen einfließen muss. Manchmal erlebe ich, dass die Prozesserhebung durch eine *schriftliche Befragung* einzelner Prozessbeteiligter erfolgt. Das Problem ist, dass Sie nicht wissen, an welcher Stelle im Fachprozess sich der jeweilige Beteiligte befindet. Aus den einzelnen Ergebnissen der schriftlichen Befragungen müssen Sie den fachlichen Prozess mühsam rekonstruieren. Der Auswertungsaufwand hängt auch von der Begabung der Befragten ab, die Aufgaben und Prozessinformationen schriftlich verständlich aufzubereiten, sowie von der Vollständigkeit und Genauigkeit der schriftlichen Aufarbeitung. Zudem spielen die persönliche Motivation und die persönliche Zeit der Prozessbeteiligten eine wichtige Rolle.

Eine weitere aktuelle Herangehensweise ist, die Prozessdetails aus vorhandenen IT-Systemen zu extrahieren. Ein Ansatz dafür ist das *Process Mining*, das Sie in Abschnitt 1.1.2, »Geschäftsprozessautomatisierung und die relevanten Technologien für den Prozessgesteuerten Ansatz«, kennengelernt haben. Damit bekommen Sie allerdings nur die *Ist-Situation* des Prozesses heraus und erhalten auch nur Fragmente des fachlichen Prozesses, da viele Aufgaben eines fachlichen Prozesses außerhalb von IT-Systemen ablaufen. Darüber hinaus erfordert es eine genaue Prüfung, um festzustellen, welche Aspekte fachlich bedeutsam sind und welche lediglich als Anpassung für die frühere Implementierung in diesem System dienten und somit als Workaround fungierten. Deshalb erweist sich dieser Ansatz als nicht unbedingt zielgerichtet.

Weder die schriftliche Befragung noch das Process Mining sind also effektiv und effizient, da Sie damit den relevanten fachlichen Prozess nicht zielführend erarbeiten.

Fachliche Prozesse müssen zusammen mit Vertretern der jeweiligen Prozessbeteiligten an einem Tisch erarbeitet werden, um die Akzeptanz des Ergebnisses sicherzustellen und ständige Fragen im Nachgang zu vermeiden. Zudem wird damit auch der Auswertungsaufwand der Prozesserhebungssitzung reduziert. Für diese kollaborative Gruppensituation empfiehlt es sich, die Prozesserarbeitung bzw. -erhebung in Form eines Workshops durchzuführen. Dieser *Prozesserhebungsworkshop* kann durchaus aus verschiedenen zielführenden Sitzungen bestehen. Dabei hängt der Erfolg eines Workshops – außer von der Auswahl der richtigen Teilnehmer (Prozessbeteiligten) – auch von den Erfahrungen, den Moderationsfähigkeiten und der Methodik des Moderators ab. Damit Sie einen Prozesserhebungsworkshop erfolgreich durchführen können, gebe ich Ihnen in diesem Kapitel eine grundlegende Anleitung mit praktischen Kniffen. Hier setze ich voraus, dass Sie bereits in grundlegenden Moderationstechniken geschult sind. Ansonsten finden Sie beispielsweise in (Lipp & Will, 2001) eine Einführung in die Moderation von Workshops.

Dieses Kapitel habe ich zur besseren Verständlichkeit inhaltlich wie folgt gegliedert:

1. In Abschnitt 5.1, »Ausgangssituation des Prozesserhebungswshops«, beschreibe ich Ihnen die Ausgangssituation, von der aus Sie den fachlichen Prozess im Prozesserhebungsworkshop erarbeiten.
2. In Abschnitt 5.2, »Prozesserhebungswshops«, beschreibe ich, wie Sie einen Prozesserhebungsworkshop mit den allgemeinen Phasen *Vorbereitung*, *Durchführung* und *Nachbereitung* durchführen.

5.1 Ausgangssituation des Prozesserhebungswshops

Solch ein Erhebungsworkshop kann in verschiedene, zielführende Workshop-Sitzungen unterteilt werden. Dabei werde ich einen Schwerpunkt auf *Remote-Workshops* legen, die in der heutigen Arbeitswelt immer mehr an Verbreitung gewinnen. Sie können meine Hinweise aber auch gut für Präsenzveranstaltungen verwenden.



Info: Workshop

Unter einem *Workshop* verstehe ich ein spezifisches Arbeitstreffen zu einem bestimmten Thema. Das Arbeitstreffen findet außerhalb des normalen Arbeitsgeschehens und nicht regelmäßig statt. Dieses spezifische Arbeitstreffen hat immer ein Ziel, und es entsteht ein Ergebnis. An dem Thema arbeitet eine Gruppe von Personen innerhalb eines bestimmten Zeitrahmens, die auch für das Ergebnis gemeinsam verantwortlich ist. Ein Moderator fördert den Arbeitsprozess der Gruppe mit Moderationstechniken (vgl. Lipp & Will, 2001, S. 14 und S. 16).

Um ein längeres, spezifisches Arbeitstreffen zu einem bestimmten Thema auch in klar abgetrennten Einheiten über mehrere Termine zielführend aufzuteilen, lässt sich dieses Arbeitstreffen in *Workshop-Sitzungen* untergliedern. Dabei ist wichtig, dass die Unterteilung so erfolgt, dass in jeder Sitzung ein bestimmtes Ziel mithilfe einer bestimmten Methode erreicht werden kann (vgl. (Schönfeld, 2017), S. 98, und (Unterauer, 2015), S. 164).

Die Ausgangssituation für den Prozesserhebungsworkshop ist der fachliche Prozess, der erarbeitet werden soll, nachdem er identifiziert worden ist. Das heißt, wir haben einen Fokus auf die Prozessautomatisierung oder die Arbeitsanleitung für die Prozessbeteiligten. Für andere Detailebenen, wie das Ableiten und die Identifikation von Geschäftsprozessen von der Geschäftsstrategie, ist methodisch ein anderer Fokus bei den zu ermittelnden Informationen nötig, den ich hier nicht darstelle. Das würde den Rahmen dieses Buches sprengen.

Zudem werden wir den fachlichen *SOLL-Prozess* erarbeiten, ohne auf aktuelle Systeme einzugehen, um flexibel für die Zukunft zu sein. Für manchen operativen Prozessbeteiligten ist es nicht einfach, von der aktuellen Situation zur zukünftigen Arbeitsweise zu kommen. Auch dafür werde ich Ihnen eine Brücke bauen. Dabei kann Ihnen die *Claim Analysis* helfen. Bitte bedenken Sie dabei, dass Sie mit unterschiedlichen Menschen arbeiten. Das bedeutet: Wenn dieses Werkzeug Ihnen immer noch nicht hilft, dann müssen Sie zumindest die IST-Situation grob erarbeiten und die Stärken und Schwächen mit den Teilnehmern ableiten, damit Sie die abgeleiteten Aspekte als Anforderungen in die SOLL-Situation übernehmen können.

5.2 Prozesserhebungsworkshops

In diesem Abschnitt leite ich Sie an, wie Sie einen Prozesserhebungsworkshop vorbereiten, durchführen und nachbereiten.

5.2.1 Einen Prozesserhebungsworkshop vorbereiten

Ich empfehle, den Prozesserhebungsworkshop wie folgt vorzubereiten:

1. **Jemanden für die Moderation auswählen:** Der Moderator oder die Moderatorin ist für den Erfolg des Workshops entscheidend. Der Moderator verfügt über Kenntnisse im Umgang mit Gruppen, beherrscht Moderationstechniken und verfügt über Wissen aus dem Geschäftsprozessmanagement. Ein Moderator sorgt für eine zielführende Gruppendiskussion mithilfe von Moderationstechniken. Er ist neutral dem Thema gegenüber und diskutiert nicht inhaltlich mit. Denn die Teilnehmer sind die Experten und müssen einen Konsens über ihr gemeinsames Ergebnis erreichen. Wenn der Moderator sich inhaltlich einbringt, ist die Gefahr sehr groß, dass das Ergebnis nicht von den Teil-

nehmern getragen wird. Interessierten Lesern empfehle ich zum Thema »Moderation« das Buch »Das große Workshop-Buch« (Lipp & Will, 2001).

2. **Auswählen, wer am Workshop teilnimmt (Wissensträger und Prozessbeteiligte):** Die Prozessbeteiligten des fachlichen Prozesses stellen die Teilnehmer des Workshops dar. Diese müssen bereit sein, gemeinsam ein Ergebnis erzielen zu wollen. Dabei muss jede Rolle im Prozess durch eine Person vertreten sein, um den fachlichen Prozess im Konsens erarbeiten zu können. Damit jeder sein Wissen aktiv beitragen kann und niemand sich »versteckt«, ist eine relativ kleine Gruppe vorzuziehen. Dafür empfehle ich nicht mehr als 10 Personen. Je kleiner die Gruppe ist, desto einfacher wird es, zu interagieren, und die Teilnehmenden trauen sich auch eher, frei zu reden. Hier sollte der *Prozessverantwortliche* dabei sein, der für die operative Ausgestaltung der Prozessziele verantwortlich ist. So können Sie sichergehen, dass das Ergebnis auch im Unternehmen akzeptiert wird. Zudem sollten auch Vertreter des *Entwicklungsteams* teilnehmen, um zu verstehen, was der fachliche Prozess für die Implementierung bedeutet.
3. Der **Moderator** verschafft sich einen **inhaltlichen Überblick** über den fachlichen Prozess, plant den Workshop mit den geeigneten Methoden und Tools und macht sich mit den persönlichen Erwartungen und Zielen der Teilnehmer vertraut (z. B. mithilfe einer Erwartungsumfrage).
4. Wenn der Workshop in Präsenz stattfindet, muss ein **geeigneter Raum reserviert** und sichergestellt werden, dass dieser die benötigte Größe und Ausstattung aufweist. Denn Sie brauchen Platz, um die Diskussion des fachlichen Prozesses gut sichtbar zu dokumentieren.
5. **Laden Sie die Teilnehmer spätestens vier Wochen vor dem Workshop** per E-Mail ein, und beschreiben Sie schriftlich kurz das Ziel, eine grobe Agenda und die eventuell vorzubereitenden Aufgaben der Teilnehmer. Natürlich geben Sie auch die Örtlichkeit und den Termin bekannt.
6. **Die Materialien vorbereiten:** Der Moderator bereitet die benötigten Materialien vor, wie eine Übersicht über die Regeln im Workshop, eine Beschreibung der Arbeitsmethoden sowie die benötigten Moderationsmaterialien.

5.2.2 Einen Prozesserhebungsworkshop durchführen

Den Prozesserhebungsworkshop beginnen Sie mit der *Orientierungsphase*. Dabei begrüßen Sie die Teilnehmer, stellen sich (sofern erforderlich), das Ziel des Workshops und die Agenda vor. Anschließend initiieren Sie eine Vorstellungsrunde, in der jeder die Gelegenheit hat, sich und seine Erwartungshaltung zu präsentieren. So wissen alle, mit wem sie es zu tun haben, und die Kommunikation wird signifikant erleichtert. Danach erläutern Sie die Arbeitsregeln für die Sitzung und stellen sicher, dass diese jederzeit sichtbar sind und von den Teilnehmern eingesehen werden können (siehe den folgenden Kasten). Die Methodik erläutern Sie so weit, wie die Teilnehmer sie zur Durchführung des Workshops

zum gegenwärtigen Zeitpunkt benötigen. Ansonsten werden die Erläuterungen als zu theoretisch und langatmig wahrgenommen.

Info: Allgemeine Arbeitsregeln für Prozesserhebungsworkshops

Folgendes erscheint mir essenziell:

1. Der Beitrag eines jeden Teilnehmenden wird als *gleichwertig* und *wertschätzend* aufgenommen.
2. Die Teilnehmenden teilen ihr Wissen bereitwillig und dokumentieren es sichtbar für alle Beteiligten.
3. Für das Ergebnis des Workshops sind die Teilnehmenden *gemeinsam* verantwortlich.
4. Es wird der fachliche *SOLL-Prozess* (d. h. der zukünftige Prozess) definiert.
5. Der fachliche, operative Prozess wird *Ende-zu-Ende* definiert, d. h. vom Kunden (*Anfang*) bis zum Kunden (*Ende*).
6. Der rein fachliche Prozess *ohne systemspezifische Features* wird erarbeitet. Das heißt, es wird der Prozess in der Denkweise und Sprache der Prozessbeteiligten erarbeitet. Die Teilnehmer können sich dabei vorstellen, dass der Prozess als Arbeitsanleitung für ihre zukünftige Arbeitsweise dienen soll.

(In Anlehnung an: (Freund & Rücker, 2019), S. 98–100; (Gappmaier & Gappmaier, 2010), S. 34; (Stiehl, et al., 2019), S. 53–54; (Unterauer, 2015), S. 15)

Anschließend können Sie inhaltlich langsam einsteigen, um das Verständnis für den Rahmen des fachlichen SOLL-Prozesses zu entwickeln. Dazu lassen Sie den Prozessverantwortlichen die *Prozessziele* und die *Anforderungen sowie Erwartungen der Kunden* an das Produkt des Prozesses erläutern. Der Prozessverantwortliche muss vorab natürlich wissen, dass er diesen Input vorbereiten muss. Dann entsteht eine kurze Fragerunde, indem die anderen Teilnehmer dem Prozessverantwortlichen weitere Fragen stellen können. Das Ziel ist es, dass die anderen Teilnehmer die zukünftige Ausrichtung des fachlichen Prozesses verstehen.

Beispiel: Fallbeispiel »Seminarverwaltung« – Prozess »Seminaranmeldung abwickeln«: Erwartungen an den SOLL-Prozess

Hier finden Sie die fundamentalen Anforderungen und Erwartungen an den zukünftigen Prozess des Prozessverantwortlichen. Um diese als Prozessziele zu formulieren, muss das Trainingsunternehmen diese Aspekte noch präzisieren:

1. **Personalkosten senken durch Prozesseffizienz:** Insbesondere soll der Einsatz von Sachbearbeitern reduziert werden, und die Administrationsaufwände für Trainer und Kunden sollen durch Self-Services geringgehalten werden.



2. **Hohe Kundenzufriedenheit durch Prozessqualität erreichen:** Dies bedeutet: Die Erwartungen der Kunden an die Interaktion mit dem Trainingsunternehmen und die Erwartungen des Trainingsunternehmens müssen in Einklang gebracht werden. Dabei müssen sich Prozesse flexibel anpassen lassen. Die Erwartungen der Kunden werden in regelmäßigen Kundenbefragungen evaluiert und in der Folge werden die fachlichen Prozesse angepasst.
3. **Kunden zur Interaktion mit dem Trainingsunternehmen einen einfachen Zugang bereitstellen:** Was dies für die typischen Kunden (»Personas«) bedeutet, wird im Rahmen von Kundenbefragungen ermittelt.
4. **Individuelle Kundenanfragen innerhalb von 5 Tagen beantworten:** Gegebenenfalls müssen Zwischenstandsmeldungen an die Kunden versendet werden.

Damit die operativen Prozessbeteiligten eine bessere Vorstellung von den Ansprüchen des zukünftigen Prozesses erhalten, hilft es mir, eine *Anspruchsanalyse* (engl. *Claim Analysis*, eine Art *Stärken-/Schwächenanalyse*) auf Basis des Inputs des Prozessverantwortlichen und der bisherigen Erfahrungen der Prozessbeteiligten durchzuführen. So erhalten Sie als Moderator eine bessere Vorstellung von den inhaltlichen Punkten, die weiterhin relevant sind, und solchen, die zukünftig umgestaltet werden müssen. Lassen Sie dabei die Teilnehmer ihre Ansichten zu den einzelnen Kategorien »Pro«, »Kontra« und »Design-Potenziale« formulieren, und gehen Sie diese anschließend gemeinsam durch. Jeder inhaltliche Aspekt wird auf ein eigenes Post-it geschrieben. Dabei können Sie Tabelle 5.1 als Template verwenden. Diese Methode habe ich damals im Studium im Rahmen von *Interaction-Design* kennengelernt. Sie hat sich in meiner Praxis als sehr einfach und wirkungsvoll zugleich herausgestellt.

Pro	<i>Was läuft gut und soll auch zukünftig beibehalten werden?</i>
Kontra	<i>Was ist verbesserungswürdig?</i>
Design-Potenziale	<i>Welche Potenziale ergeben sich auch durch das Zusammenspiel bei der Umsetzung in Technologien (z. B. »Bestimmte Entscheidungsregeln auf Basis von künstlicher Intelligenz durchführen«, ...) – Hier dürfen Sie gerne Ihren Gedanken freien Lauf lassen und kreativ sein.</i>

Tabelle 5.1: Template für eine »Claim Analysis«



Beispiel: Fallbeispiel »Seminarverwaltung« – Prozess »Seminaranmeldung abwickeln«: Claim Analysis (bezogen auf die IST-Situation)

Pro (Was läuft gut und sollte beibehalten werden?)

- Automatische Bereitstellung der gepflegten Seminarinformationen in dem Onlineshop

- Personalbereitstellung (HR), Rechnungswesen und Customer-Relationship-Management-Prozesse: Dafür sollen weiterhin die Funktionalitäten der etablierten Systeme verwendet werden.

Kontra (Was ist verbesserungswürdig?)

- Die Interaktion zwischen dem Trainingsunternehmen und seinen Kunden im Rahmen der Seminaranmeldung erfordert einen viel zu hohen manuellen Aufwand durch Sachbearbeiter*innen.
- Die aktuelle Trainingssoftware ist nicht flexibel genug: Fachprozesse können derzeit nicht wirklich angepasst werden. Eine regelmäßige Anpassbarkeit der fachlichen Prozesse erfordert auch eine Flexibilität in der Anwendung.

Design-Potenziale

- Den Prozessgesteuerten Ansatz nach Volker Stiehl nutzen, um die fachlichen Prozesse flexibel, transparent, nachhaltig, effizient und qualitativ hochwertig in Form einer oder mehrerer Prozessgesteuerter Anwendungen zu implementieren.
- Die bereits vorhandene Funktionalität in den fachlichen Prozessen wiederverwenden (Adapter-Muster).
- Sachbearbeiter*innen und Trainer*innen bei regelbasierten und strukturierten Logiken durch Einsatz der Anwendung entlasten. Dazu muss die entsprechende Logik identifiziert und detailliert definiert werden.
- Die Akteure im Prozess (wie Kunden/Trainer*innen) automatisch je nach Bedarf im Rahmen des Prozessablaufs einbinden und über relevante Ereignisse informieren.

Im nächsten Schritt geht es darum, Prozessdetails zu erarbeiten. Dazu wenden Sie den *Top-down-Ansatz nach Bruce Silver* an, auf den ich in Kapitel 6 noch genauer eingehen werde:

1. Prozessumfang bestimmen

Verwenden Sie die Fragen zum Start und Inhalt des Prozesses sowie zu möglichen Szenarien für das Ende als Vorlage für die Diskussion. Sorgen Sie dafür, dass das Ergebnis sichtbar dokumentiert wird und jeder das Gleiche darunter versteht. Eventuell auftretende Begriffe werden ebenfalls mit ihrer Definition, relevanten Synonymen sowie ihrem Kontext für jeden sichtbar dokumentiert. Zudem sollten auch relevante Konzepte zum Verständnis des Prozessumfangs visualisiert werden. Dies hilft bei der weiteren Diskussion der Details und hilft jedem Teilnehmer zu verstehen, welche Bedeutung sein einzelnes Ergebnis im Prozess hat.

Achten Sie als Moderator darauf, dass die Teilnehmer die *Rahmenbedingungen zur SOLL-Situation* und die Ergebnisse der *Anspruchsanalyse* berücksichtigen. Sollten Sie merken, dass Sie dies vernachlässigen, fragen Sie nach, inwiefern damit die Rahmenbedingungen der SOLL-Situation abgedeckt werden.