

Vorlesung
***Methodische Grundlagen des
Software-Engineering***
im Sommersemester 2014

Prof. Dr. Jan Jürjens

TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

Teil 1.2: Einführung in BPMN 2.0

v. 27.04.2014

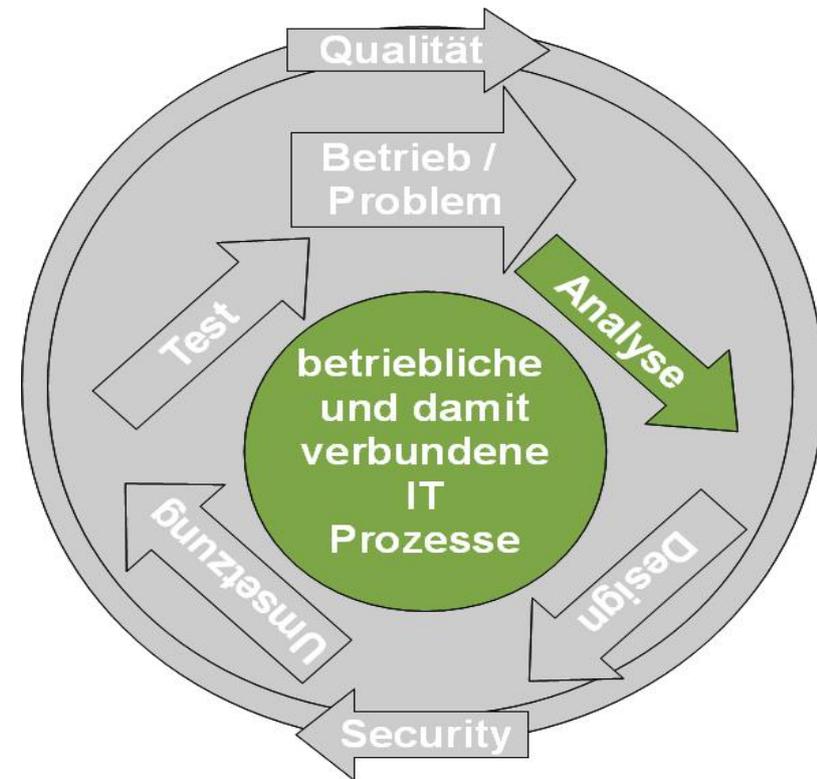
1.2 Einführung in BPMN 2.0 (Business Process Model and Notation)

[inkl. Beiträge der IBM Software Group und
Prof. Dr. Frank Leyman, Univ. Stuttgart]

Literatur:

J. Freund, B. Rücker: Praxishandbuch BPMN 2.0. Carl Hanser Verlag, 2012, 3. Auflage. Unibibliothek E-Book: <http://www.ub.tu-dortmund.de/katalog/titel/1361293>
• Kapitel 1-2

- **Geschäftsprozessmodellierung**
 - Grundlagen Geschäftsprozesse
 - Ereignisgesteuerte Prozessketten (EPKs)
 - **Einführung in die BPMN 2.0**
 - Workflow-Management-Systeme
 - Workflow-Automatisierung
- Process Mining
- Modellbasierte Entwicklung sicherer Software

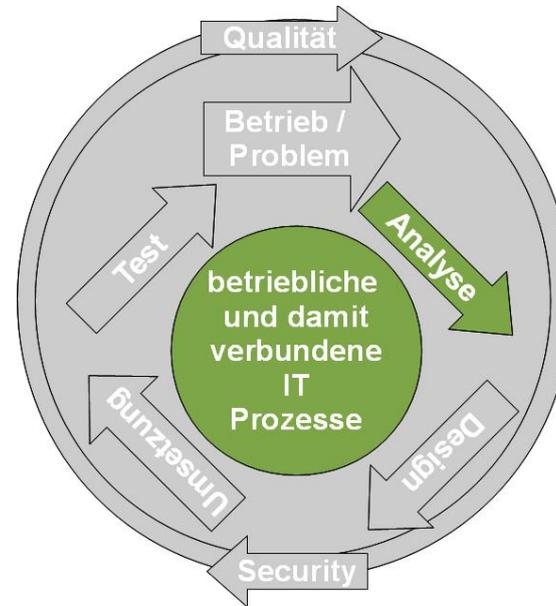


- **BPMN am Beispiel**
- **BPMN-Hintergrundwissen**
- **BPMN im Detail**
 - Grundlegende Konzepte
 - Weiterführende Konzepte

Motivation: Kommunikation über Prozesse

Christian Customer

Wenn ein Kunde anruft wird zuerst immer der Kunde identifiziert und dann der Standort erfragt. Anhand der Daten können wir feststellen was für eine Betreuung gebucht wurde. Wenn es sich um einen Standardkunden handelt, analysieren die Call Agents mithilfe unseres Hilfsleitfadens die Situation und versuchen den Fehler zu beheben. Wenn dies fehlschlägt, oder das Expertensystem die Weiterleitung angibt, wird der Fall aufgenommen und das Ticket zu Second Level weitergeleitet. Das Telefonat wird beendet. Wenn die Second Level den Fehler bearbeitet hat, wird entweder der Kunde vom Call Agent oder vom Second Level direkt angerufen. Wenn es ein Premium Kunde ist, wird der Kunde sofort von einem Second Level Mitarbeiter kontaktiert.



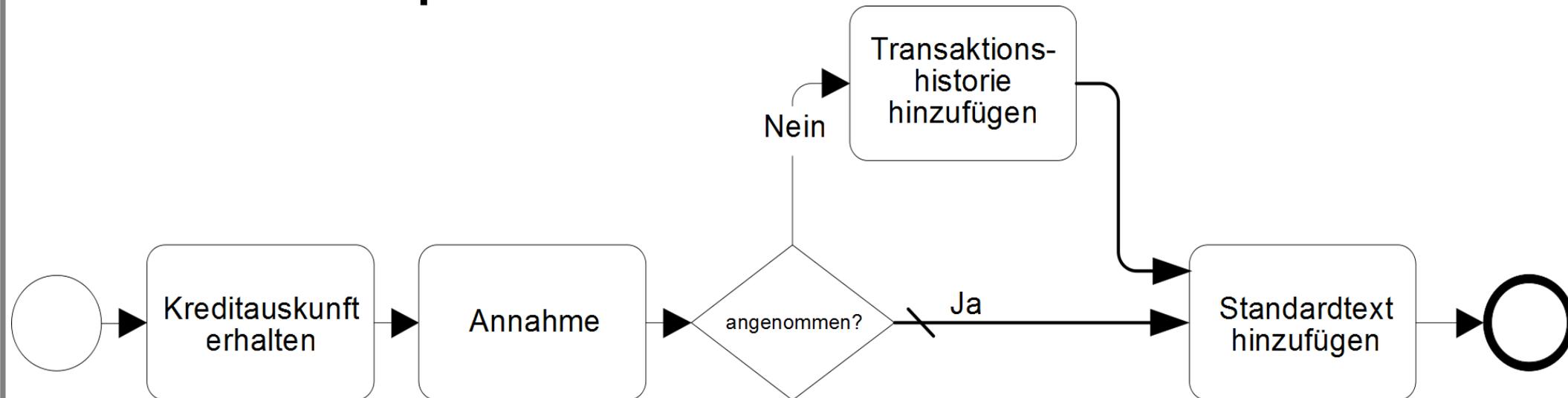
Paul Programmer

```
CallSupport();  
Kunde k = identifyCustomer();  
k.identifyDepartment();  
if(k.hasBasicSupport()) {  
    ....  
} elseif {h.hasPremiumSupport() }  
    ...  
}
```

????

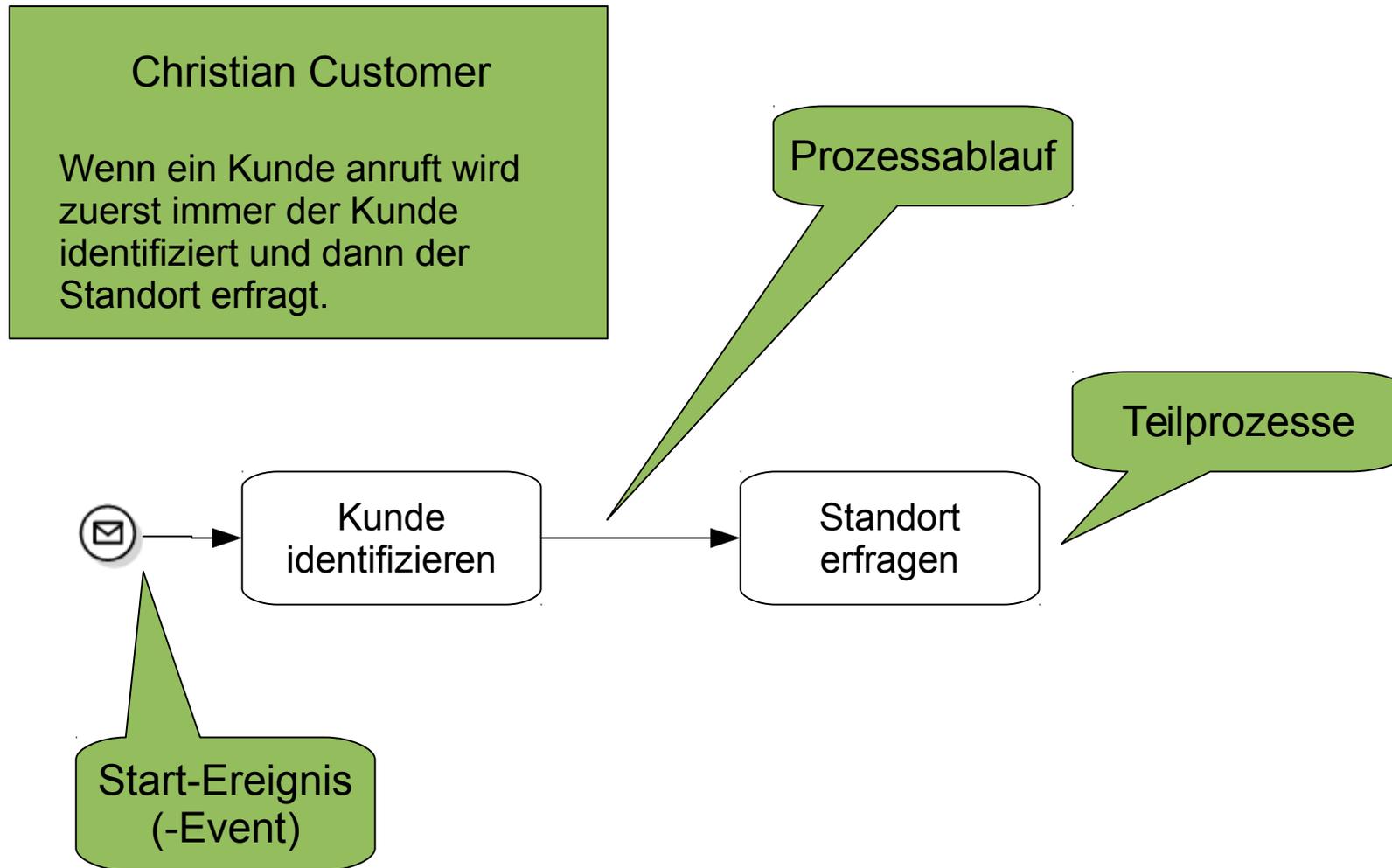
Was ist Business Process Model and Notation (BPMN) ?

- **Flussdiagramm-basierte Notation für Definition von Geschäftsprozessen.**

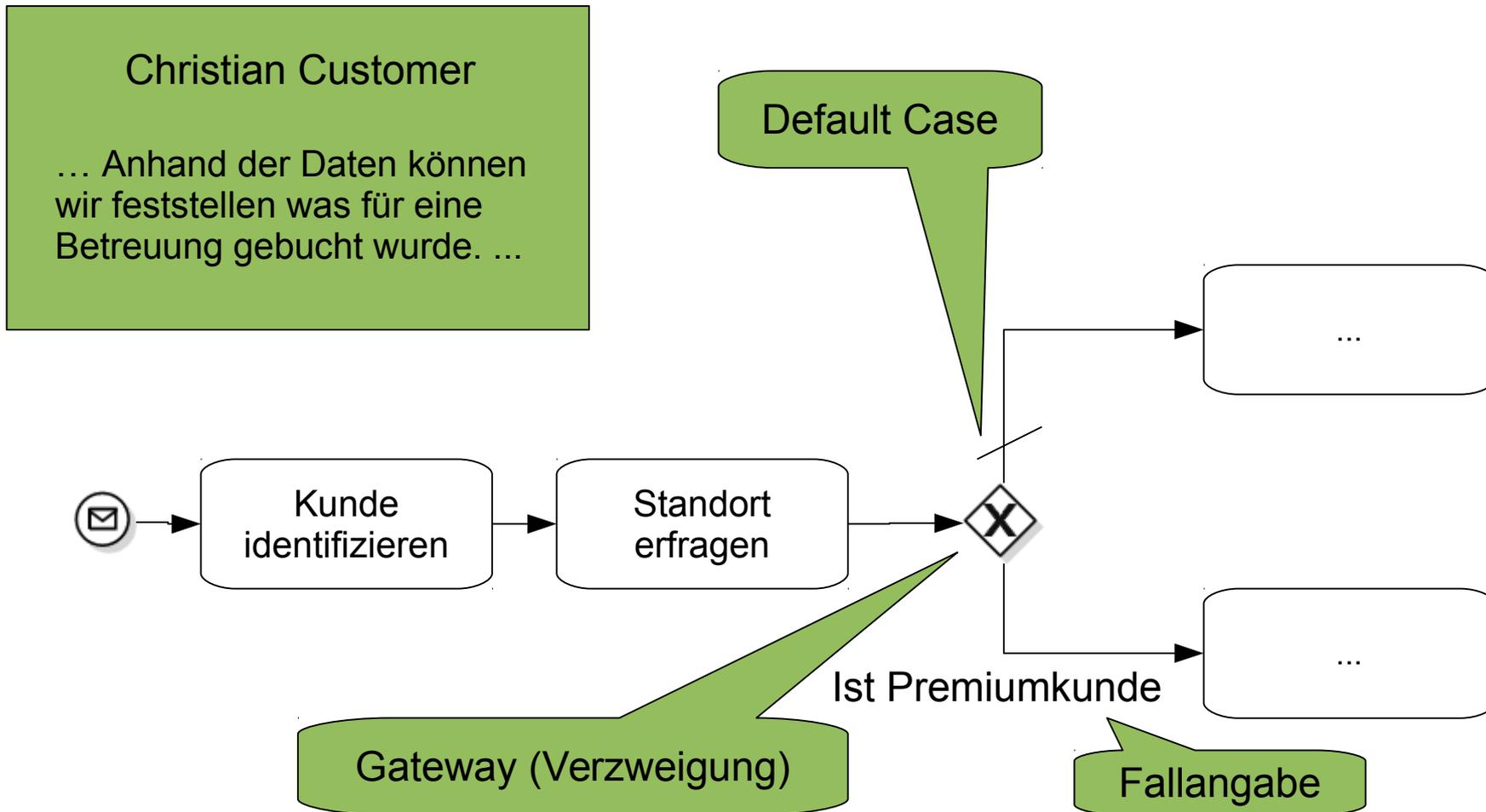


- **Vereinbarung verschiedener Notationsanbieter.** → Mit einziger Notation dem Anwender Arbeit und Verständnis erleichtern.

BPMN am Beispiel: Startereignis, Prozessablauf, Teilprozesse



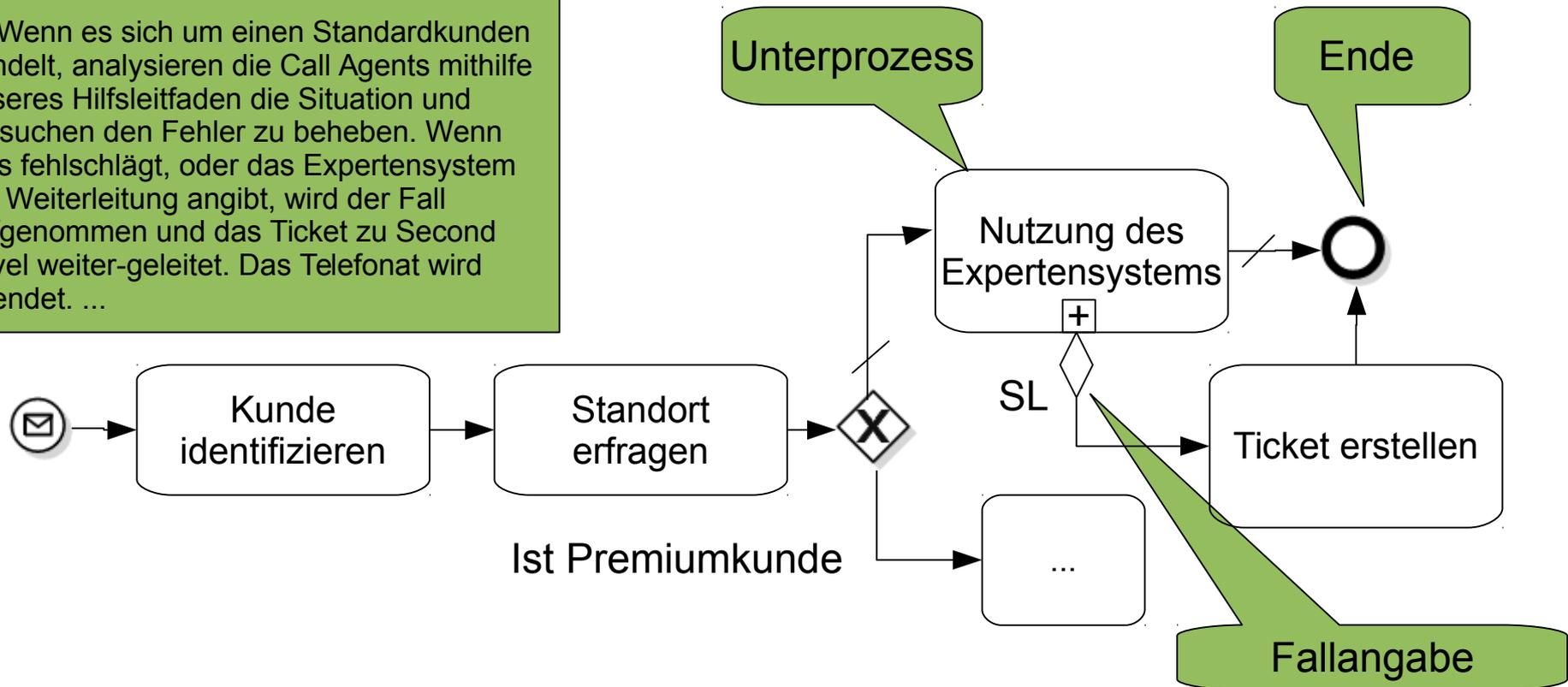
BPMN am Beispiel: Gateway (Verzweigung)



BPMN am Beispiel: Unterprozess

Christian Customer

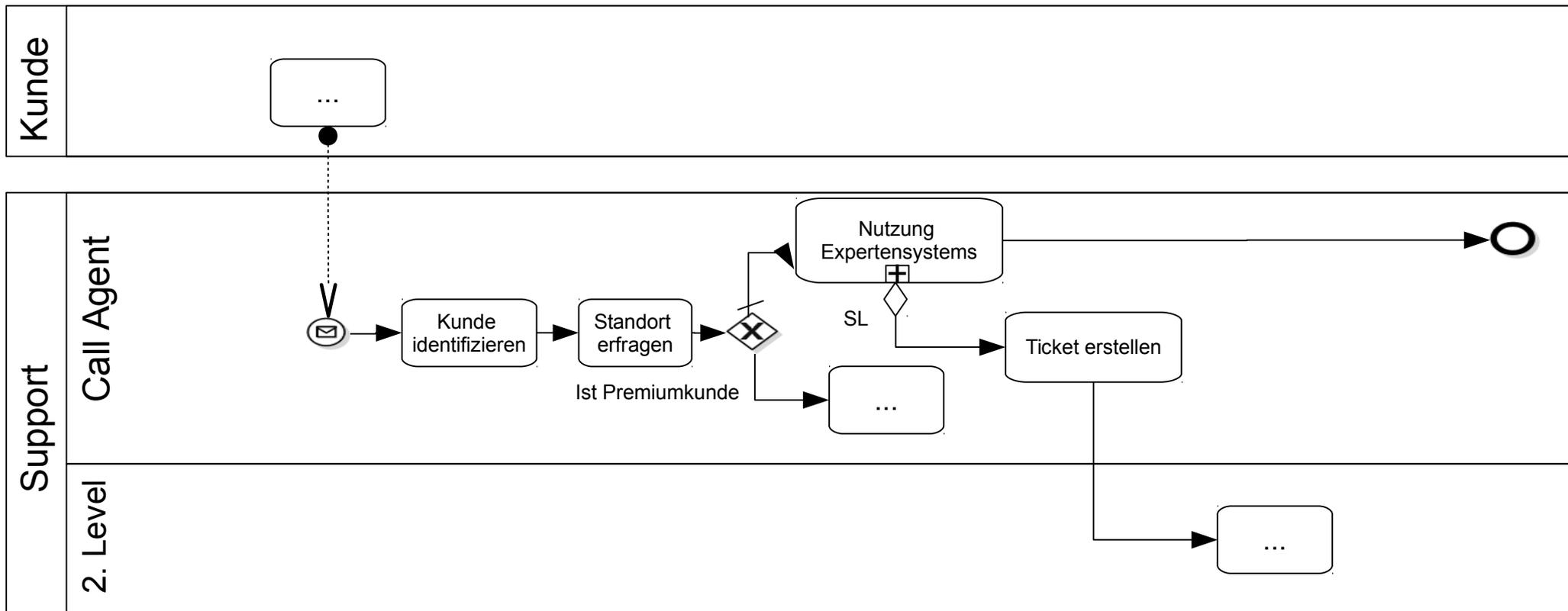
... Wenn es sich um einen Standardkunden handelt, analysieren die Call Agents mithilfe unseres Hilfsleitfadens die Situation und versuchen den Fehler zu beheben. Wenn dies fehlschlägt, oder das Expertensystem die Weiterleitung angibt, wird der Fall aufgenommen und das Ticket zu Second Level weiter-geleitet. Das Telefonat wird beendet. ...



BPMN am Beispiel: Swimlanes

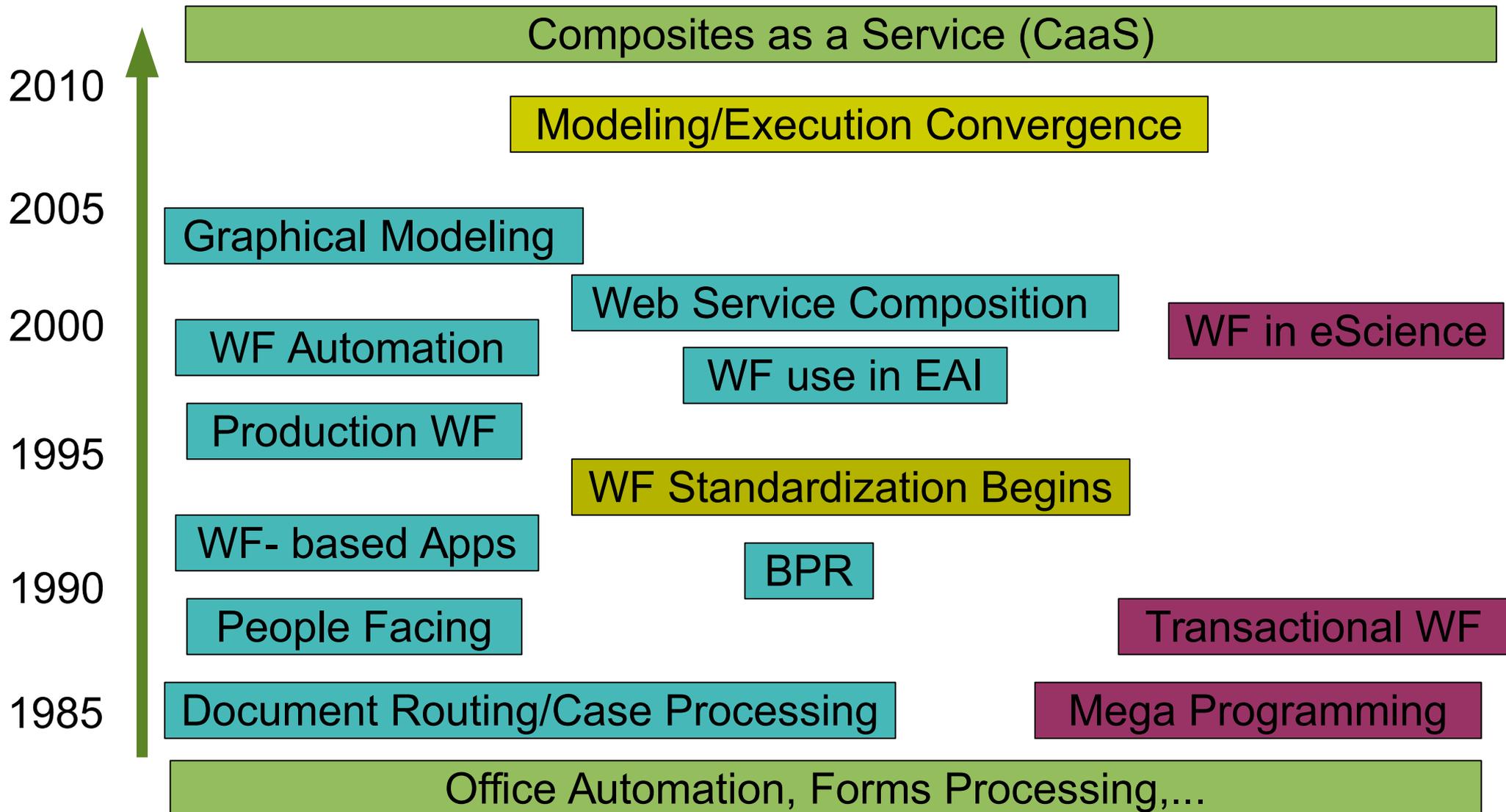
Christian Customer

... Wenn es sich um einen Standardkunden handelt, analysieren die Call Agents mithilfe unseres Hilfsleitfadens die Situation und versuchen den Fehler zu beheben. Wenn dies fehlschlägt, oder das Expertensystem die Weiterleitung angibt, wird der Fall aufgenommen und das Ticket zu Second Level weitergeleitet. Das Telefonat wird beendet. ...



- **BPMN am Beispiel**
- **BPMN-Hintergrundwissen**
- **BPMN im Detail**
 - Grundlegende Konzepte
 - Weiterführende Konzepte

Kurze Geschichte der Workflow-Technologie



Ziel: **GP-Modellierungsnotation** mit folgenden Eigenschaften:

- Kombination aus **grafischen Elementen** und zusätzl. **Informationen** (Attributen).
- Ausführbaren Prozess (z.B. BPEL) aus BPMN-Modell erstellen.
- Auch für **allgemeinere Geschäftszwecke** verwenden können.
- Akzeptabel und benutzbar für Wirtschaft.
- **Agnostische Methodik:**
 - Hinweise zu **Zweck** und **Detailgrad** der Modellierung.
 - Gesamte BPMN-Notation komplex, aber i.A. nur Teil benötigt.

Verschiedene Ebenen der Prozessmodellierung

BPMN

GP beschreiben,
wie Unternehmen
Ziele verfolgt.

Process Maps

(Prozesslandkarte)
einfaches Diagramm
der Aktivitäten

Process Descriptions

(Prozessbeschreibung)
erweitert, aber nicht genug
für kompletten Ablauf

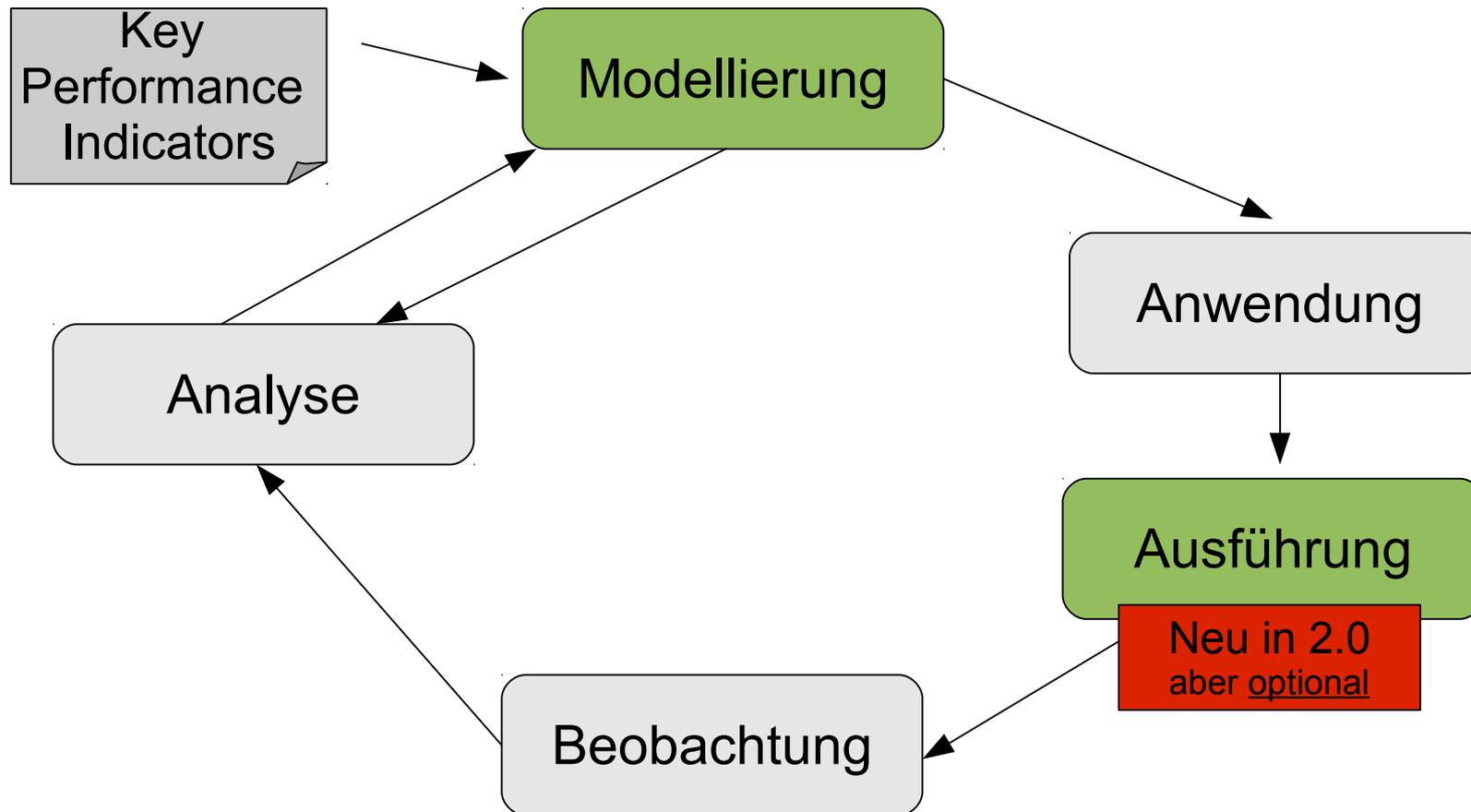
Process Models

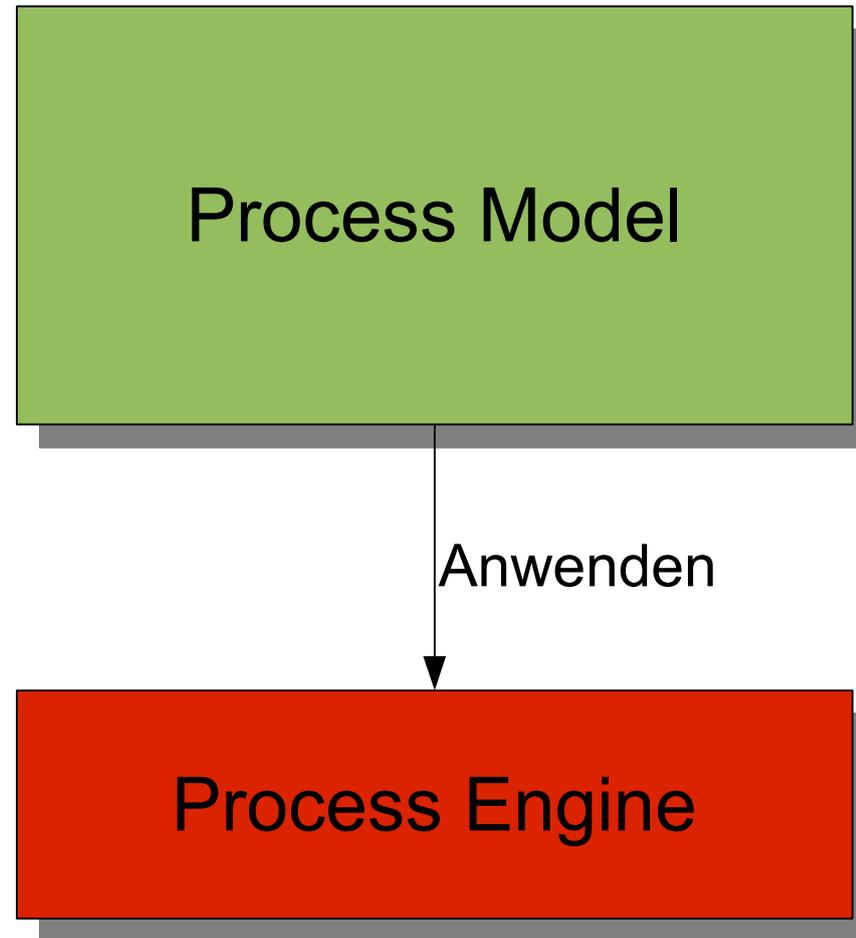
(Prozessmodell)
erweitert: analysieren,
simulieren und ausführen

einfach

voll-
ständig

BPMN Lebenszyklus: Wo BPMN ansetzt



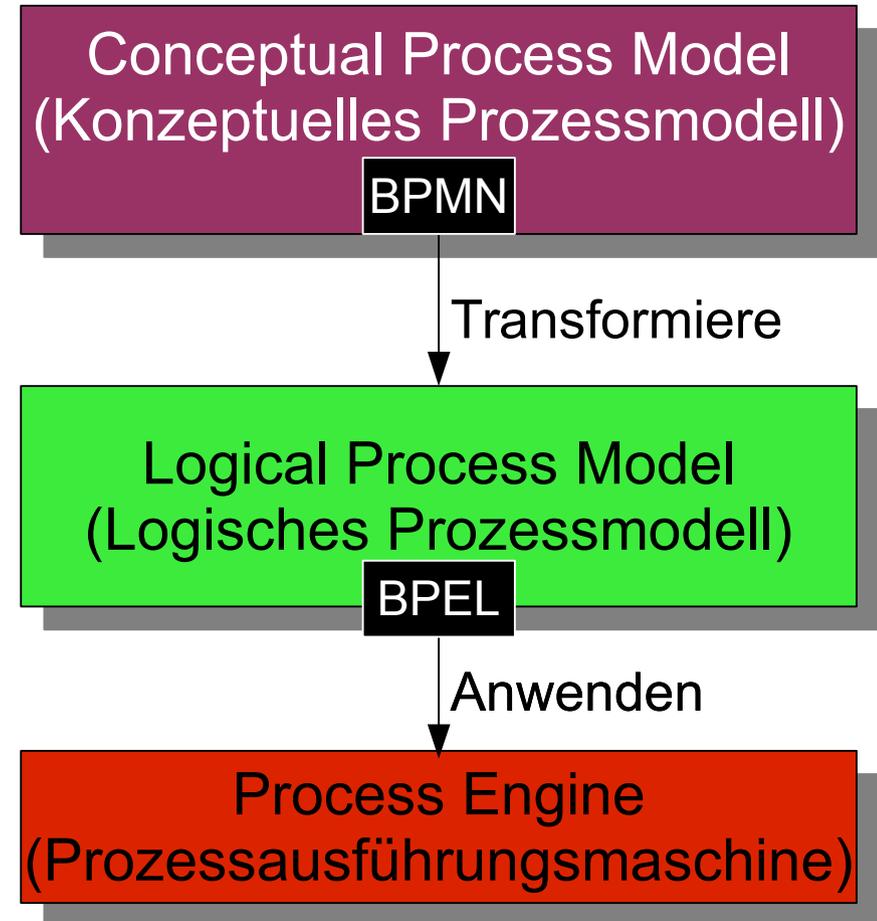


Ausführung von BPMN-Prozessen mittels BPEL-Repräsentation

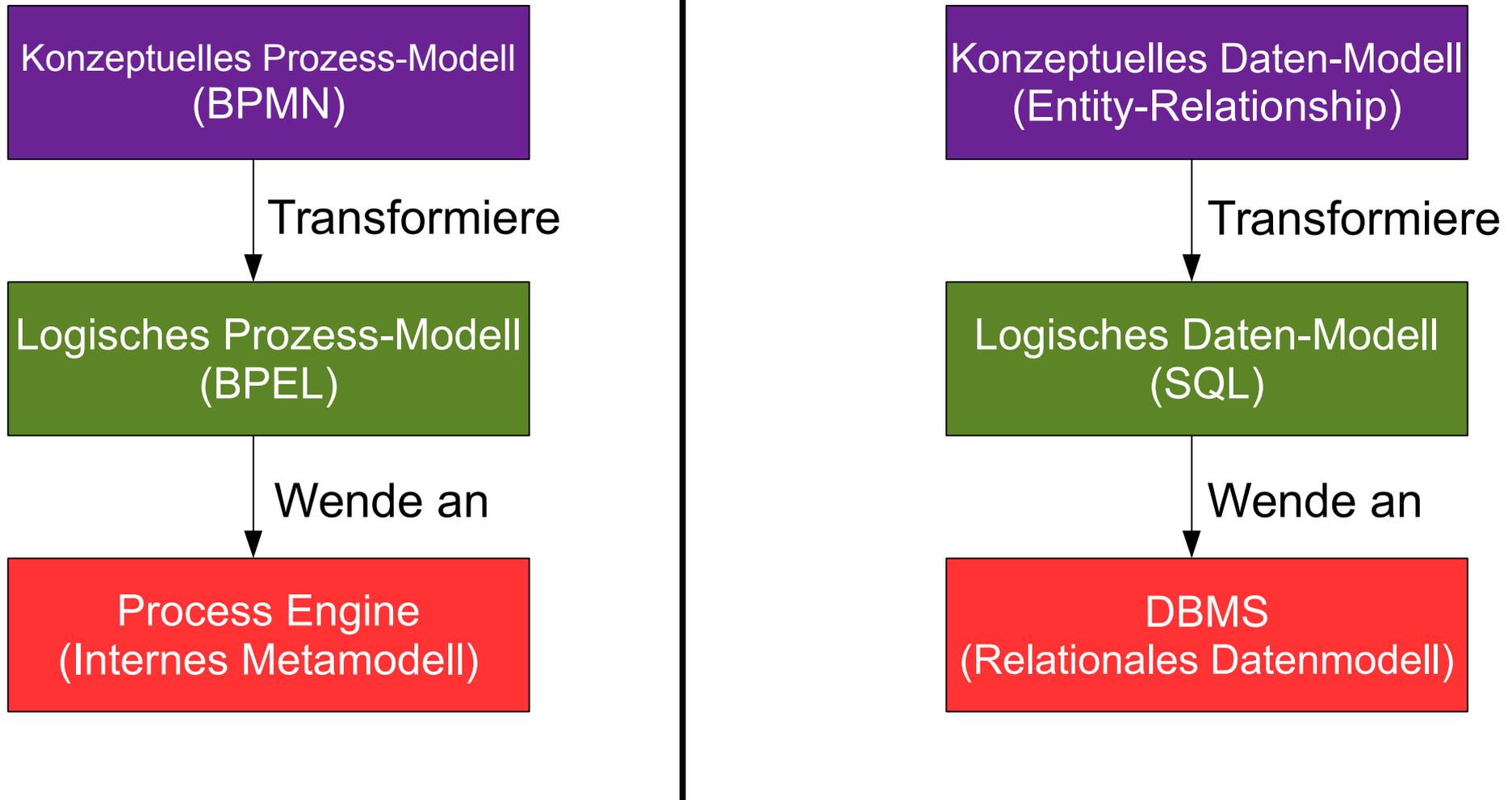
- Verständlich
- Graphisch

- XML-basiert
- Ausführbar
- Ausführungsumgebungen

GP direkt in **BPEL-Engine** ladbar,
ohne Veränderung durch
**menschliche Interpretation/
Übersetzungen.**



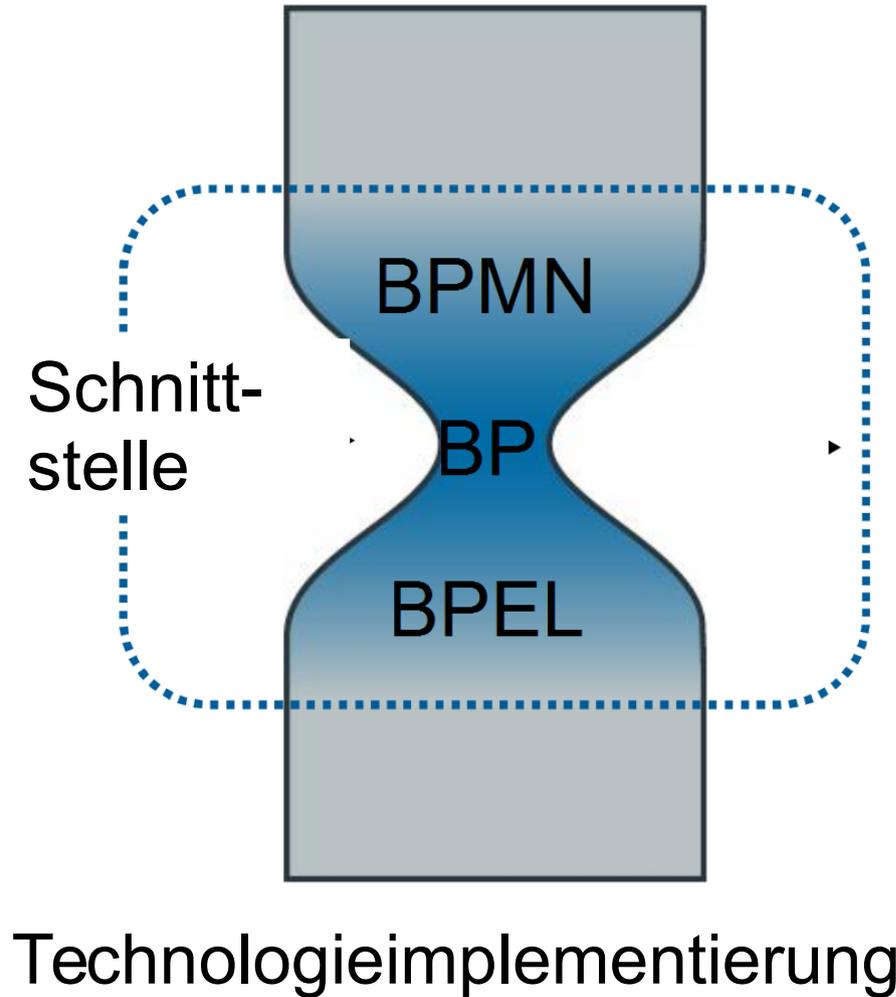
Vergleich: Prozess- vs. Datenmodellierung



Anwender:

Strategieberater
Wirtschaftsanalyst
Prozess-Designer
System-Architekt
Software-Engineer

Geschäftsumgebung:



Zweck:

↑
Modellierung

Ausführung
↓

Was ist die Business Process Execution Language (BPEL) ?

- **XML-basierte Ausführungs-Sprache.**
- Spezifikation von **Geschäftsprozessen** und **Interaktionsprotokollen** basierend auf Web Services. (Notation: s. übernächster Vorlesungsabschnitt.)
- Basiert auf verschiedenen XML Standards („layer“):
 - XML Schema 1.0
 - XPath 1.0
 - WSDL 1.1
- Einbindung und Koordination der Aktivitäten als **Web Services** (über WSDL).
- Ausführung eines **BPEL-Prozesses** durch Process-Engine.

BPEL wird von meisten Middleware-Anbietern unterstützt:

- Kombination zweier **Modellparadigmen**.
- **Vermeidung** der **Zersplitterung** des Workflow-Marktes.
- Der Runtime Standard.
- Gut definierte **Syntax** und **operationale Semantik**.
- Gewisse **Portabilität** über Anbietergrenzen.

Nachteil: keine „schöne“ Sprache.

Von BPEL zu BPMN: Historische Entwicklung

Business Process Management Institute (BPMI; jetzt Teil der Object Management Group (OMG)): BPML (XML-basierte Prozessausführungs-Notation).

BPML durch BPEL ersetzt.

2001: „Notation Working Group“ gegründet:
35 Firmen, Organisationen und Einzelpersonen.

2004: BPMN 1.0 (Business Process Modeling Notation)
Spezifikation veröffentlicht.

2006: BPMN 1.0 ist OMG-Standard.

2011: BPMN 2.0 (Business Process Model and Notation) verabschiedet.
(vgl. <http://www.omg.org/spec/BPMN/2.0>)

2013: Version 2.0.1 ist ISO-Standard: ISO/IEC 19510

Dez. 2013: Version 2.0.2

2.0.2: Nur kleinere Änderungen gegen-
über 2.0, daher im Folgenden 2.0.

- **BPEL** konzentrierte sich auf die **Syntaxsprache** und **operationale Semantiken**.
- Ziel: schnelle Markteinführung → **visuelle Repräsentation** ignoriert.
- → Schwer zu benutzen.
- BPMN 1.x füllte diese Lücke.
- **Nicht-IT Experten** konnten über Geschäfts-Prozesse kommunizieren.
- Großer Fortschritt in der BPM Technologie.

BPEL ein Ausgangspunkt für **BPMN**:

- Ziel: vollständige **bijektive Transformation**.

Erzeugung von BPEL **automatisch**

- z.B. mit Hilfe eines graph. Modellierungswerkzeuges von BPMN nach BPEL.
- **Problematik**: Trotz unterschiedlicher Ausdrucksmächtigkeit der Sprachen hohe Überdeckung.
 - Z.B. Datenaspekt bei BPMN vernachlässigt

- **Operationale Semantik.**
- „Natives“ **XML**-Austausch-Format.
- Meta-Modell.
- **Event- & Exception-Handler** (BPEL-Gegenstücke).
- Choreographie-Erweiterungen.

...und:

Muster-basiertes Abbilden zu BPEL (Teilmenge von BPMN):

- BPMN 2.0 enthält signifikante Teilmenge **isomorph** zu BPEL. →
Visuelle Darstellung von BPEL.
(Transformation: s. übernächster Vorlesungsabschnitt.)

- **Komplexität** ist groß.
 - Lauffähigkeit zu sichern, ist immer komplex.
- Kann nur Teilmenge von BPMN 2.0 zu BPEL einfach abbilden.
 - Kann BPMN-Prozesse modellieren, die keine **kanonische Repräsentation** in BPEL haben.
- Einige Funktionen haben noch keine **Laufzeitumgebung**.
 - Z.B.: dazugehörige Domain außerhalb des Anwendungsbereiches von BPEL (siehe BPEL4Chor).

BPMN 2.0 vs BPEL: Mögliche Sichtweisen

Standpunkt 1 (**BPEL-zentrisch**):

- **Teilmenge** von BPMN 2.0 ist **isomorph** zu BPEL.
 - Kanonisch zu BPEL transformiert und in **BPEL-Engines** ausführbar.
- Passende Teilmenge: visuelle Schicht oben auf BPEL.

Standpunkt 2 (**BPMN-zentrisch**):

- BPMN 2.0: Prozesssprache mit **wohldefinierter operationalen Semantik** durch BPEL.
- **BPMN 2.0 Engine** erstellbar, ohne separate **BPEL Engine** zu benutzen.

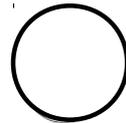
- **BPMN am Beispiel**
- **BPMN-Hintergrundwissen**
- **BPMN im Detail**
 - Grundlegende Konzepte
 - Weiterführende Konzepte

- **Flussobjekte** (Flow Objects): die Knoten (Aktivität (activity), Verzweigung (gateway) und Ereignis (event))
- **Konnektoren**: die verbindenden Kanten

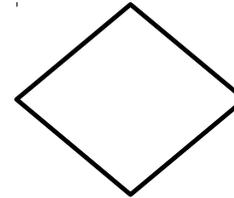
Aktivitäten



Ereignisse



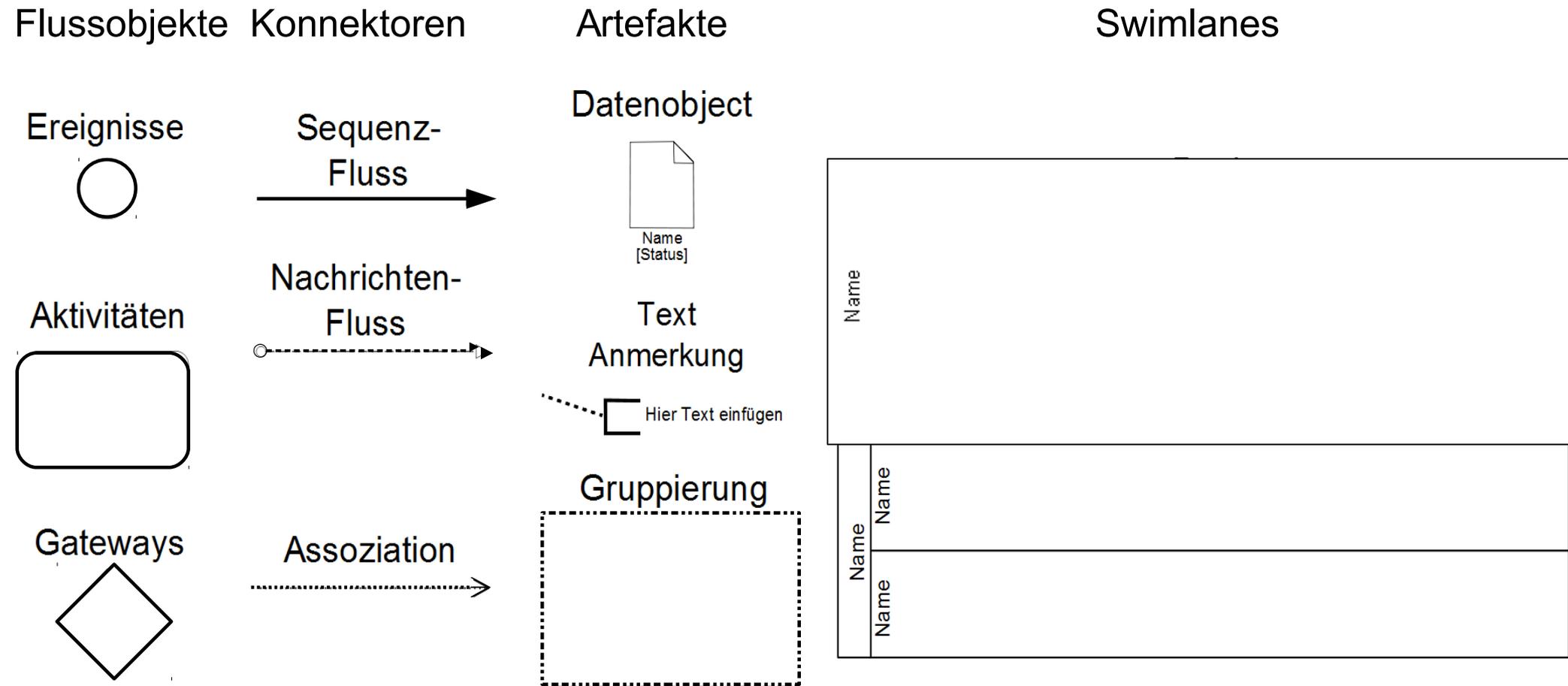
Gateways

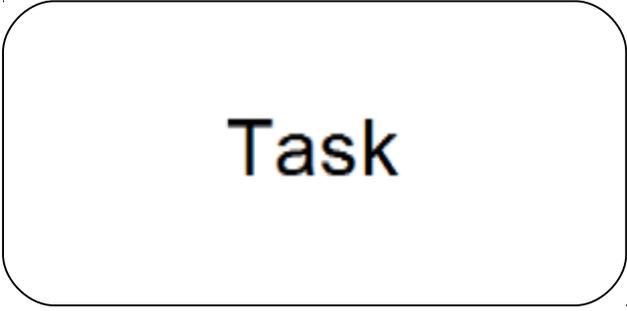


Verbindungen



- **Pools** und **Swimlanes**: die Bereiche, mit Aktoren und Systemen
- **Artefakte**: u.a. Datenobjekte, Gruppen und Annotationen



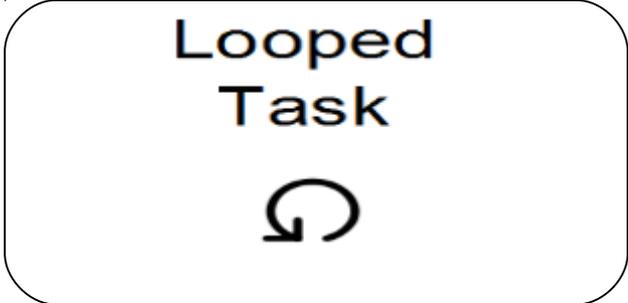


Task



Unterprozess

- **Aktivität:** In Geschäftsprozess verrichteter Arbeit.
 - Atomar oder nicht-atomar (zusammengesetzt)
 - Arten von Aktivitäten: **Unterprozess** und Aufgabe (**Task**).
- Aktivitäten werden als **abgerundete Rechtecke** dargestellt.
- Einfach oder als intern definierte mehrfache Wiederholung ausführbar.



Looped
Task

Aufgabe: atomare Aktivität.

Benutzt, wenn Prozess nicht in höherem Detailgrad dargestellt.

Spezielle Arten von Aufgaben für **Senden**, **Empfangen**, oder **anwenderorientierte Arbeiten**.

Kann Aufgabe **Markierungen** und **Symbole** zuweisen. Wichtig dabei:

- **Redundanz:** Dürfen nur auch ohne Markierung ersichtliche Informationen enthalten. (Markierungen werden bei Transformationen nicht mitgenommen, es droht Informationsverlust.)
- **Konfusionsfreiheit:** Markierung darf nicht zur Verwechslung der Elemente führen.

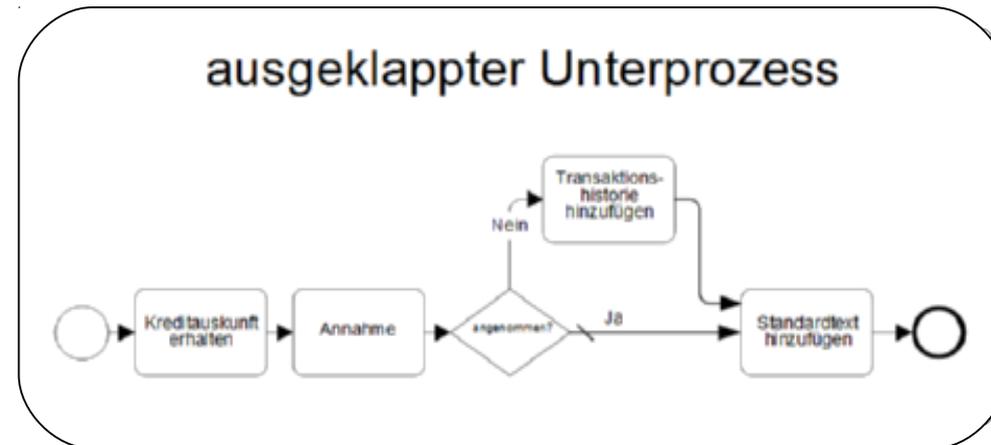
Abrechnung
senden

Behandlungs-
aufforderung



Bestellung
zusammenstellen

- **Unterprozess:** Zusammengefasste Aktivitätsmenge innerhalb Prozess.
- **Zusammengeklappter** Unterprozess: Markiert durch „+“ in der unteren Mitte des Symbols. Keine Details sichtbar.
- **Aufgeklappter** Unterprozess: Details innerhalb Unterprozessgrenzen sichtbar.
- Ermöglicht **hierarchische Prozessentwicklung**.



Verständnis-Frage: Aufgabe vs. Unterprozess

Was ist der Unterschied zwischen einer **Aufgabe** und einem **Unterprozess** ?

Verständnis-Frage: Aufgabe vs. Unterprozess

Was ist der Unterschied zwischen einer **Aufgabe** und einem **Unterprozess** ?

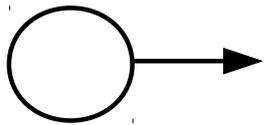
Unterprozess zum **untergeordneten detaillierten Prozess** aufschlüsselbar, Aufgabe nicht.

- Wir sind **hungrig** und möchten unseren **Hunger stillen**
- Dazu müssen wir uns zuerst entscheiden, ob wir **Pizza oder Pasta essen** möchten
- Je nach Wahl, müssen wir das entsprechende **Gericht zubereiten**
- zuletzt müssen wir die **Mahlzeit verzehren**

- Wir sind **hungrig** und möchten unseren **Hunger stillen**
- Dazu müssen wir uns zuerst entscheiden, ob wir **Pizza oder Pasta essen** möchten
- Je nach Wahl, müssen wir das entsprechende **Gericht zubereiten**
- zuletzt müssen wir die **Mahlzeit verzehren**

Erstes einfaches Beispiel

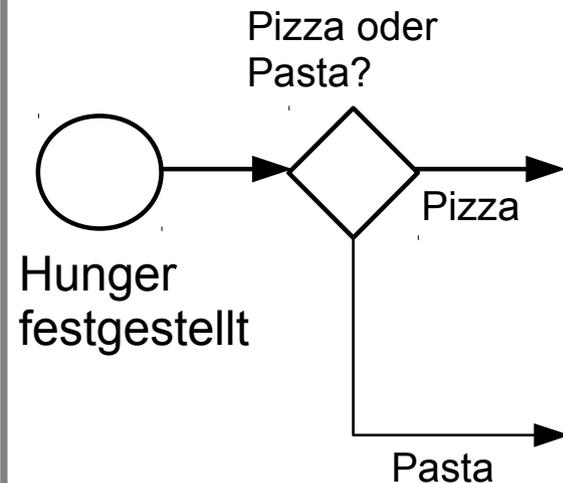
- Wir sind **hungrig** und möchten unseren **Hunger stillen**
- Dazu müssen wir uns zuerst entscheiden, ob wir **Pizza oder Pasta essen** möchten
- Je nach Wahl, müssen wir das entsprechende **Gericht zubereiten**
- zuletzt müssen wir die **Mahlzeit verzehren**



Hunger
festgestellt

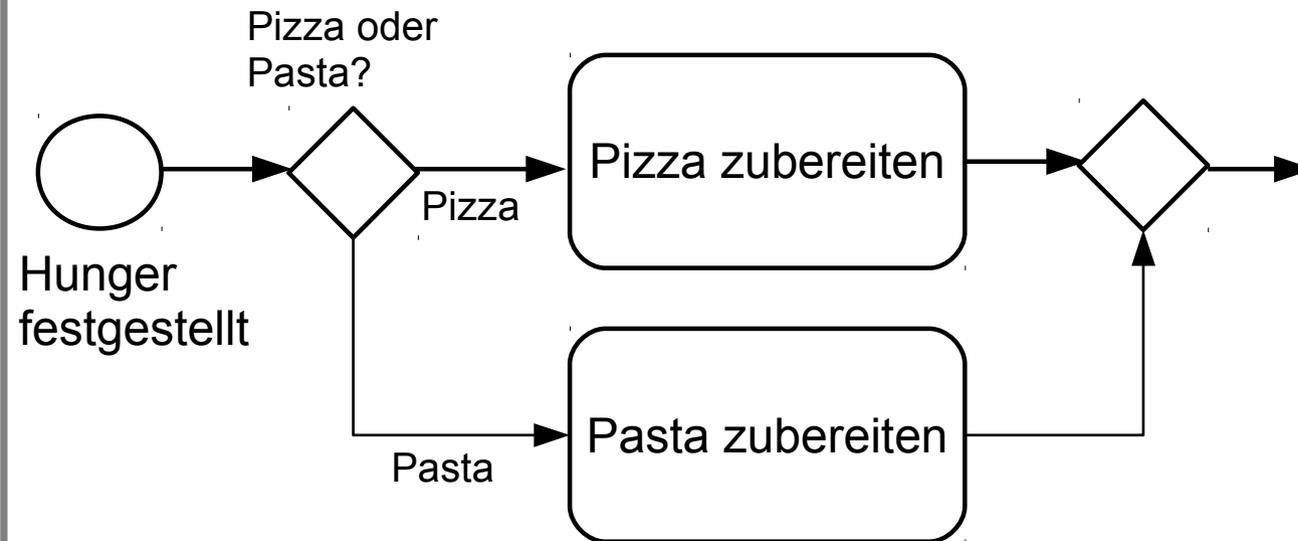
Erstes einfaches Beispiel

- Wir sind **hungrig** und möchten unseren **Hunger stillen**
- Dazu müssen wir uns zuerst entscheiden, ob wir **Pizza oder Pasta essen** möchten
- Je nach Wahl, müssen wir das entsprechende **Gericht zubereiten**
- zuletzt müssen wir die **Mahlzeit verzehren**



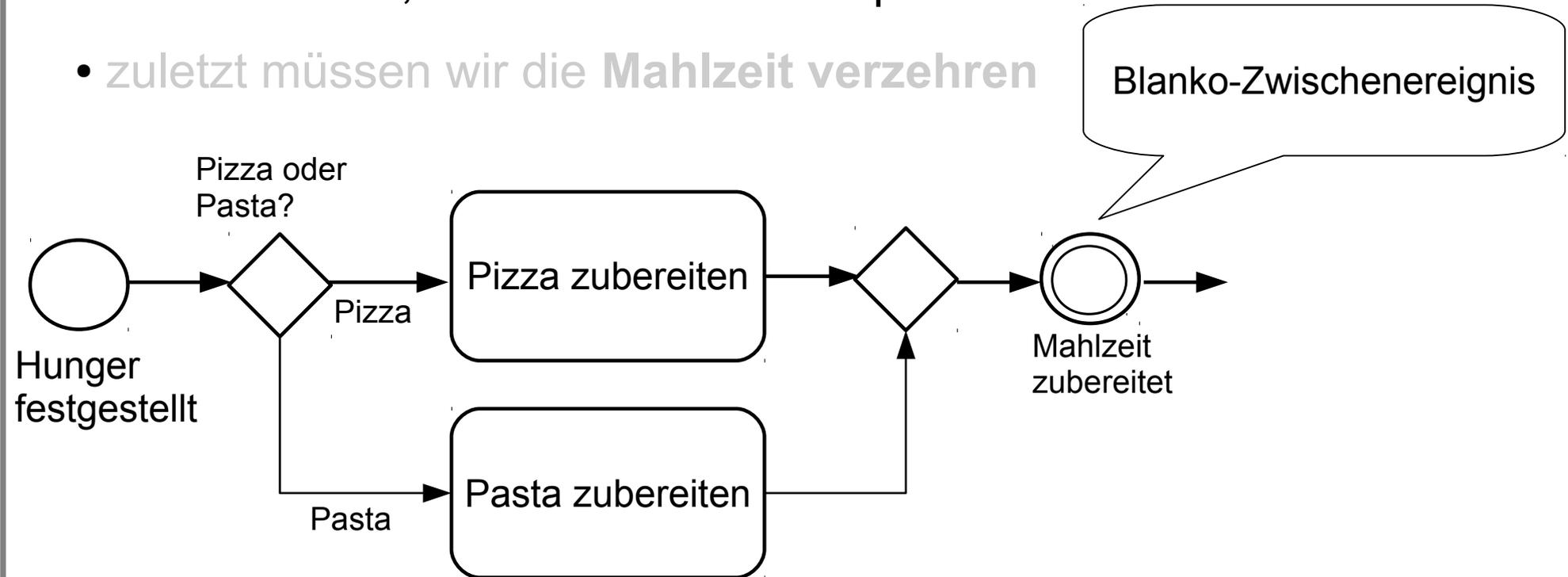
Erstes einfaches Beispiel

- Wir sind **hungrig** und möchten unseren **Hunger stillen**
- Dazu müssen wir uns zuerst entscheiden, ob wir **Pizza oder Pasta essen** möchten
- Je nach Wahl, müssen wir das entsprechende **Gericht zubereiten**
- zuletzt müssen wir die **Mahlzeit verzehren**



Erstes einfaches Beispiel

- Wir sind **hungrig** und möchten unseren **Hunger stillen**
- Dazu müssen wir uns zuerst entscheiden, ob wir **Pizza oder Pasta essen** möchten
- Je nach Wahl, müssen wir das entsprechende **Gericht zubereiten**
- zuletzt müssen wir die **Mahlzeit verzehren**

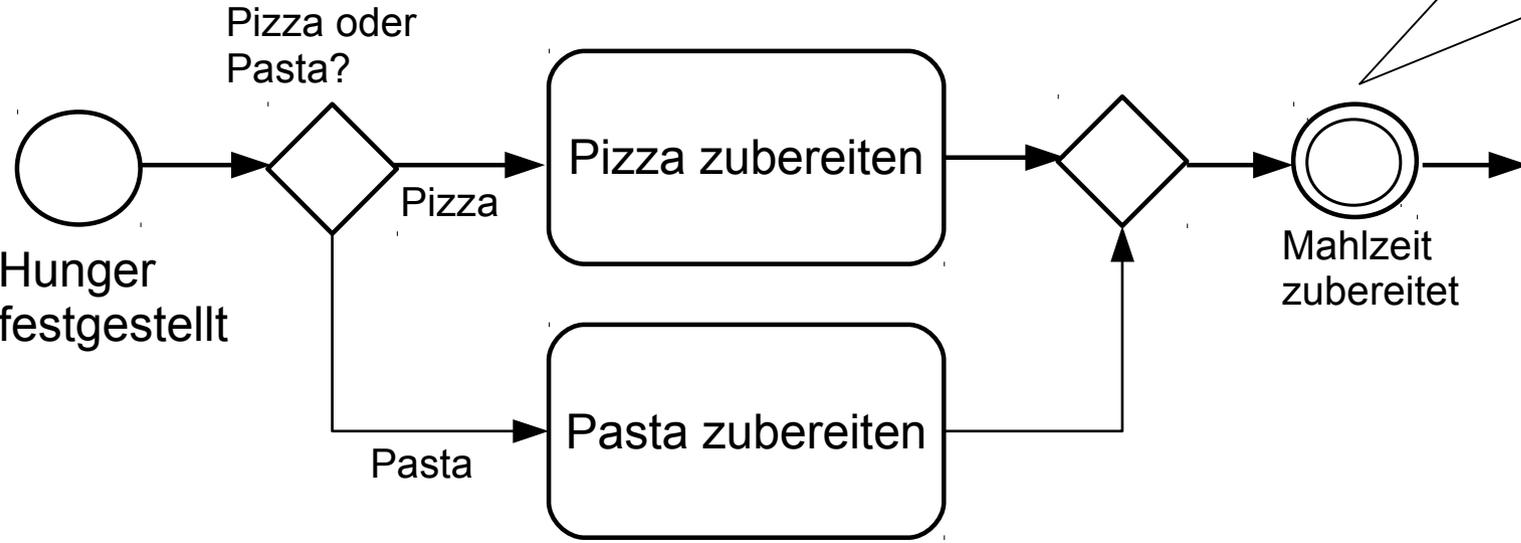


Erstes einfaches Beispiel

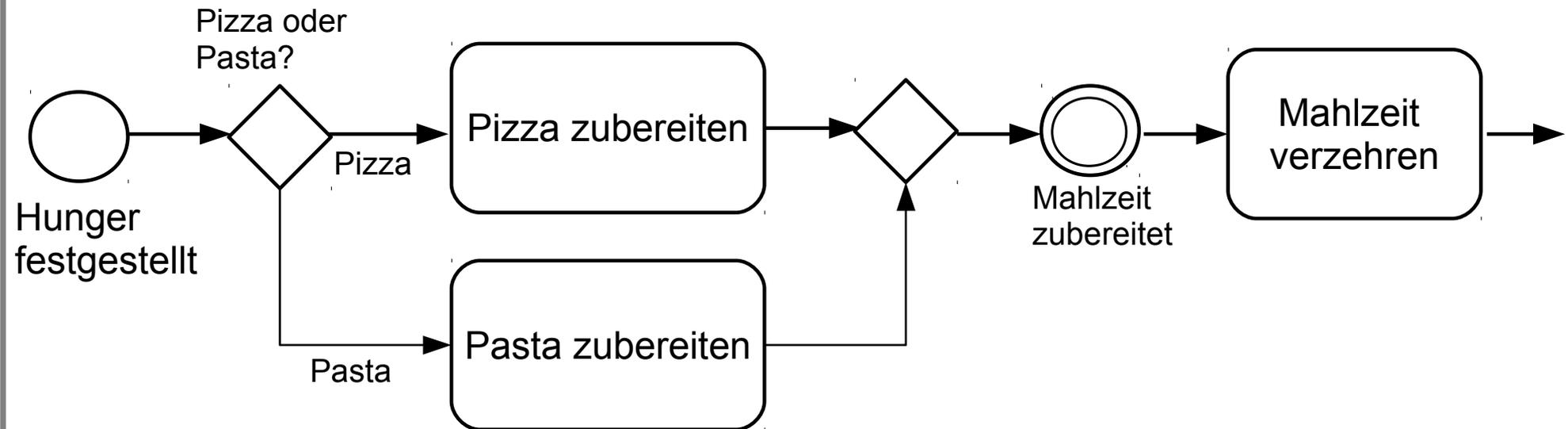
- Wir sind **hungrig** und möchten unseren **Hunger stillen**
- Dazu müssen wir uns zuerst entscheiden, ob wir **Pizza oder Pasta** **essen** möchten
- Je nach Wahl, müssen wir das entsprechend **zubereiten**
- **zuletzt** müssen wir die **Mahlzeit verzehren**

Kennzeichnet hier einen Status, der durch den Prozessfortschritt erreicht wird

Später: andere Arten von Zwischenereignissen

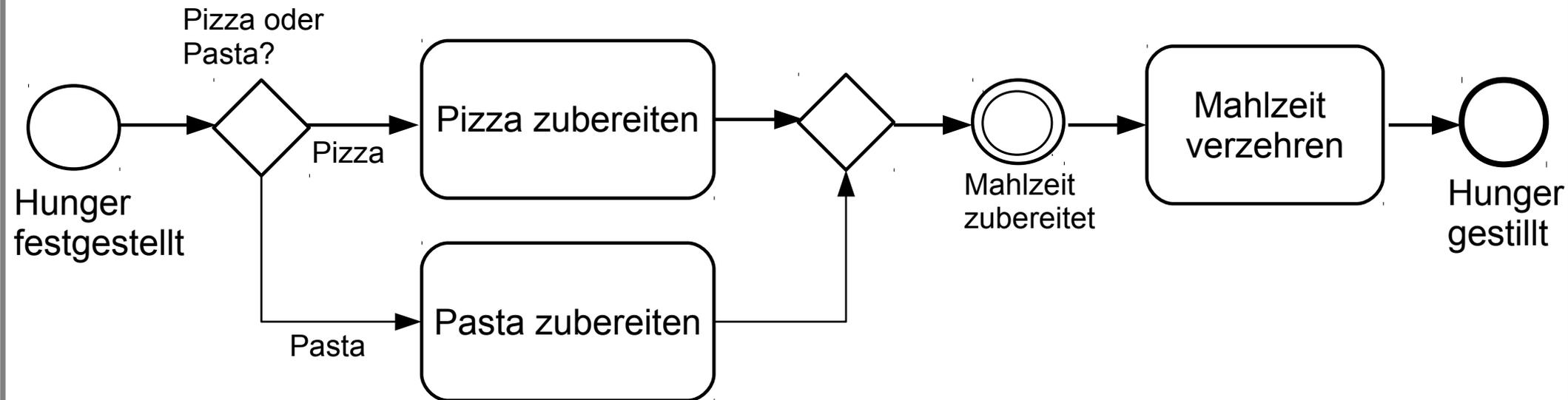


- Wir sind **hungrig** und möchten unseren **Hunger stillen**
- Dazu müssen wir uns zuerst entscheiden, ob wir **Pizza oder Pasta essen** möchten
- Je nach Wahl, müssen wir das entsprechende **Gericht zubereiten**
- zuletzt müssen wir die **Mahlzeit verzehren**



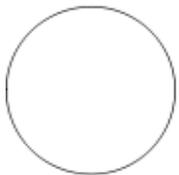
Erstes einfaches Beispiel

- Wir sind **hungrig** und möchten unseren **Hunger stillen**
- Dazu müssen wir uns zuerst entscheiden, ob wir **Pizza oder Pasta essen** möchten
- Je nach Wahl, müssen wir das entsprechende **Gericht zubereiten**
- zuletzt müssen wir die **Mahlzeit verzehren**

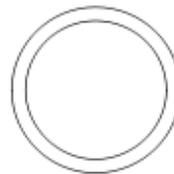


- **Ereignis:** Etwas, das während Geschäftsprozesses „passiert“ und Fluss des Prozesses beeinflusst.
- Hat **Auslöser** („Trigger“, s. nächste Folie) oder **Ergebnis**.
- Kann Fluss starten, unterbrechen oder enden lassen.
- Symbolisiert durch **Kreise**. Art des Randes bestimmt Ereignis-Typ.

Start



Zwischenereignis



Ende



Startereignisse bestimmen, wo Prozess beginnt.

Verschiedene **Auslöser**: Geben an, unter welchen Umständen Prozess startet:

- „**Nichts**“ („None“): Start von Unterprozess oder undefinierte Startumstände.
- **Nachricht**: Ereignis durch Nachricht ausgelöst.
- **Zeitmesser**: Ereignis durch Zeitüberschreitung ausgelöst.
- **Regel**: Durch Brechen oder Wirksam-Werden einer Regel ausgelöst (z.B. Tempolimit-Verstoß).
- **Mehrfach (OR)**: Jedes enthaltene Startereignis startet Prozess.
- **Mehrfach (AND)**: Alle enthaltenen Startereignisse benötigt, um Prozess zu starten.

Nichts



Nachricht



Zeitmesser



Regel



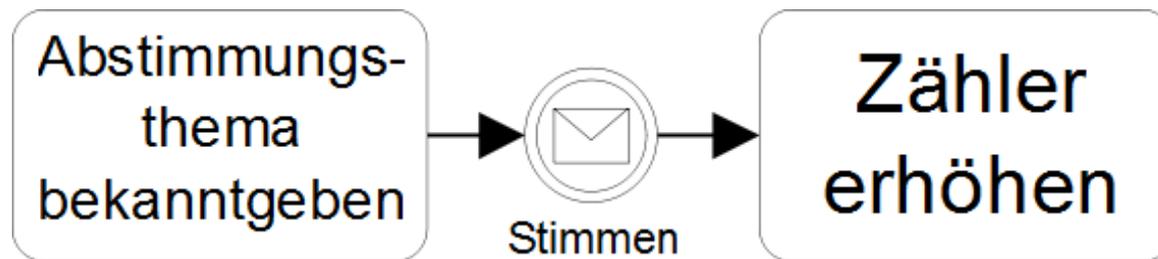
Mehrfach
(OR)



Mehrfach
(AND)



- **Im Prozessfluss** platzierte **Ereignisse** repräsentieren Dinge, die während der Prozessausführung passieren.
- Sie können
 - **Reaktion** auf ein Ereignis repräsentieren (z.B. Erhalten einer Nachricht).
 - **Auftreten** eines Ereignisses anzeigen (z.B. Versenden einer Nachricht).



- **Auslösende** Zwischenereignisse: Durch den Prozess getriggert.
→ Token löst Ereignis aus und wandert weiter, kein Abbruch!
- **Eintretende** Zwischenereignisse: angeheftet, am Rand zu unterbrechender Aktivität positioniert. → **Abbruch** der Aktivität!

Eingetretene Zwischenereignisse (Intermediate Events)

- **Zwischenereignisse** treten auf nach Start und vor Ende des Prozesses.
- Teil des normalen Flusses / an Rand der Aktivität anhängbar (später mehr).
- Darstellung: nicht ausgefülltes Symbol.
- Bei **eingetretenen Zwischenereignissen** im **Sequenzfluss** läuft Prozess erst weiter, wenn entsprechendes Ereignis (z.B. Erhalt einer Nachricht) eintritt.
- Verschiedene **Auslöser** spezifizieren Umstände des Ereignisses (wie bei Startereignissen).

Nichts

Nachricht



Zeitmesser



Fehler

Kompensation

Regel



Link



Mehrfach



Ausgelöste Zwischenereignisse (Intermediate Events)

- Teil des normalen Flusses.
- Darstellung: ausgefülltes Symbol.
- Bei **auslösenden Zwischenereignissen** im **Sequenzfluss** löst Prozess das Ereignis (z.B. Versenden einer Nachricht) aus und läuft sofort weiter.
- Verschiedene **Auslöser** (wie bei Endereignissen). Sowie:
 - **Kompensation**: Definiert Rück-abwickeln vorhergehender Tasks (z.B. Rücküberweisung von Geld).
 - **Link-Ereignis**: s. spätere Folie

Nichts



Nachricht



Zeitmesser

Fehler

Kompensation



Regel

Link



Mehrfach

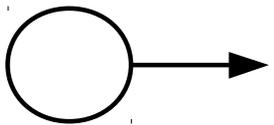


Ein Pizza Beispiel

- Wir möchten Pizza machen
- zuerst müssen wir die Pizza zubereiten (Teig ausrollen und belegen)
- dann die Pizza in den Ofen schieben
- anschließend 15 Minuten warten
- und zuletzt die Pizza aus dem Ofen nehmen

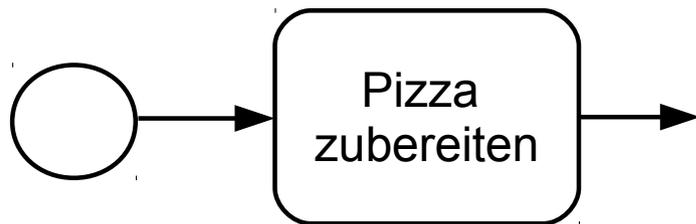
Ein Pizza Beispiel

- Wir möchten Pizza machen
- zuerst müssen wir die Pizza zubereiten (Teig ausrollen und belegen)
- dann die Pizza in den Ofen schieben
- anschließend 15 Minuten warten
- und zuletzt die Pizza aus dem Ofen nehmen



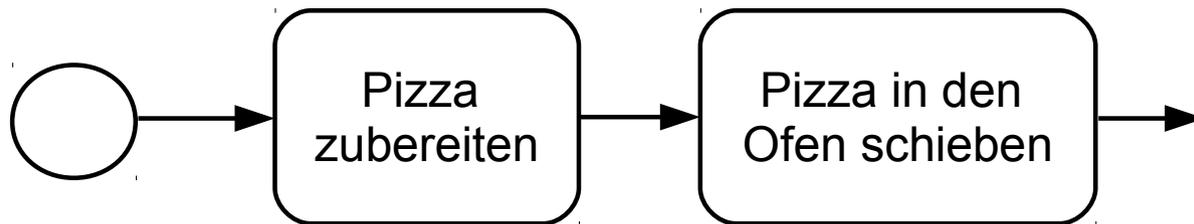
Ein Pizza Beispiel

- Wir möchten Pizza machen
- zuerst müssen wir die Pizza zubereiten (Teig ausrollen und belegen)
- dann die Pizza in den Ofen schieben
- anschließend 15 Minuten warten
- und zuletzt die Pizza aus dem Ofen nehmen



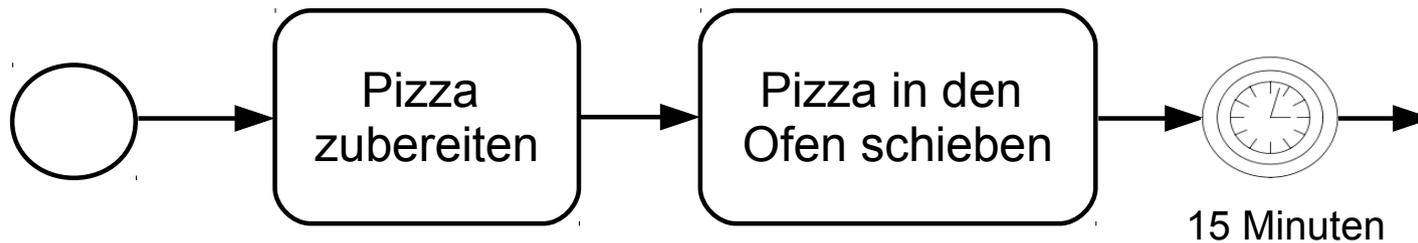
Ein Pizza Beispiel

- Wir möchten Pizza machen
- zuerst müssen wir die Pizza zubereiten (Teig ausrollen und belegen)
- dann die Pizza in den Ofen schieben
- **anschließend 15 Minuten warten**
- und zuletzt die Pizza aus dem Ofen nehmen



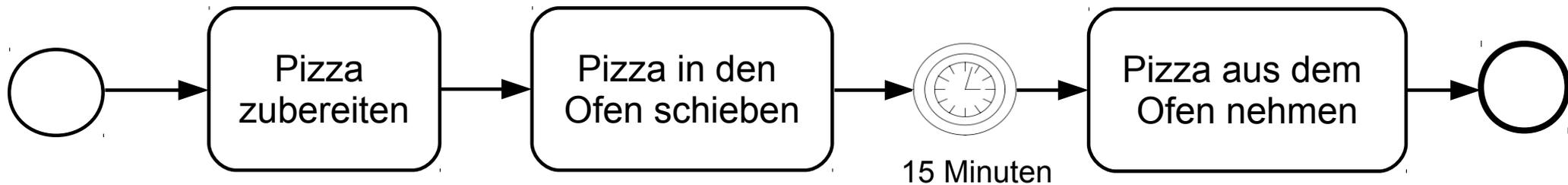
Ein Pizza Beispiel

- Wir möchten Pizza machen
- zuerst müssen wir die Pizza zubereiten (Teig ausrollen und belegen)
- dann die Pizza in den Ofen schieben
- anschließend 15 Minuten warten
- und zuletzt die Pizza aus dem Ofen nehmen



Ein Pizza Beispiel

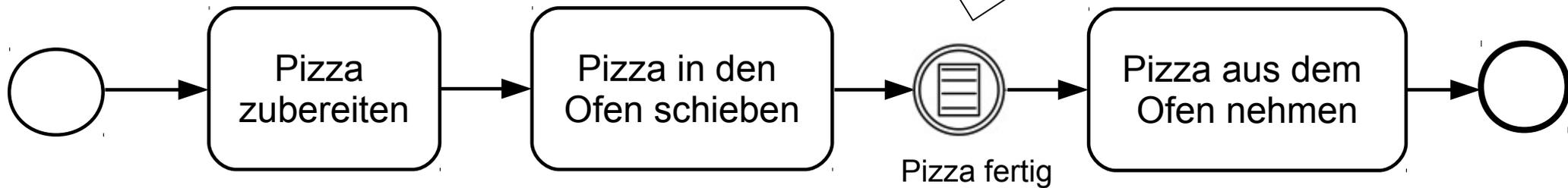
- Wir möchten Pizza machen
- zuerst müssen wir die Pizza zubereiten (Teig ausrollen und belegen)
- dann die Pizza in den Ofen schieben
- anschließend 15 Minuten warten
- und zuletzt die Pizza aus dem Ofen nehmen



Ein Pizza Beispiel

- Wir möchten Pizza machen
- zuerst müssen wir die Zutaten (einlegen)
- dann die Pizza zubereiten
- anschließend in den Ofen schieben
- und zuletzt die Pizza aus dem Ofen nehmen

Alternativ: Zwischenereignis mit Regel
Die Pizza wird erst aus dem Ofen geholt,
wenn sie fertig ist.



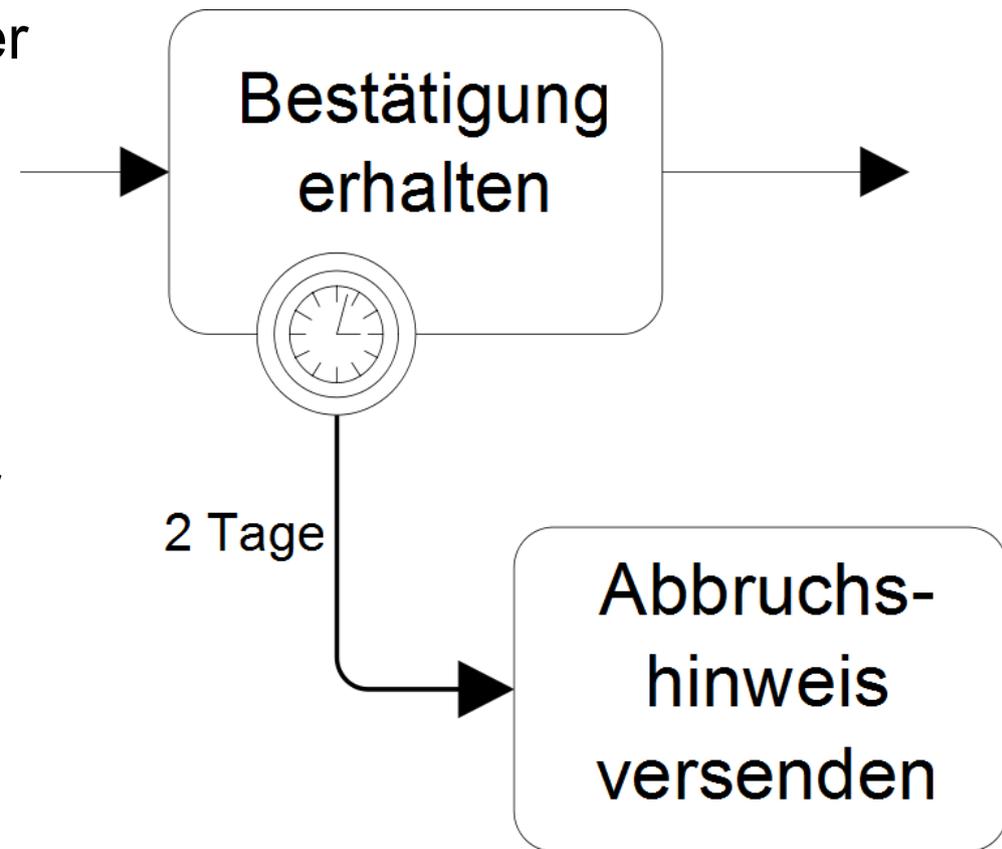
An **Aktivitäts-Rand** angehängte Ereignisse repräsentieren Auslöser (Trigger). → Können Aktivität **unterbrechen**.

An **Aufgaben** und **Unterprozesse** anhängbar.

Werden z.B. benutzt, um **Fehler** oder **Ausnahmen** zu behandeln.

Jegliche Arbeit innerhalb der Aktivität wird beendet und der **Fluss** setzt sich am Ereignis fort.

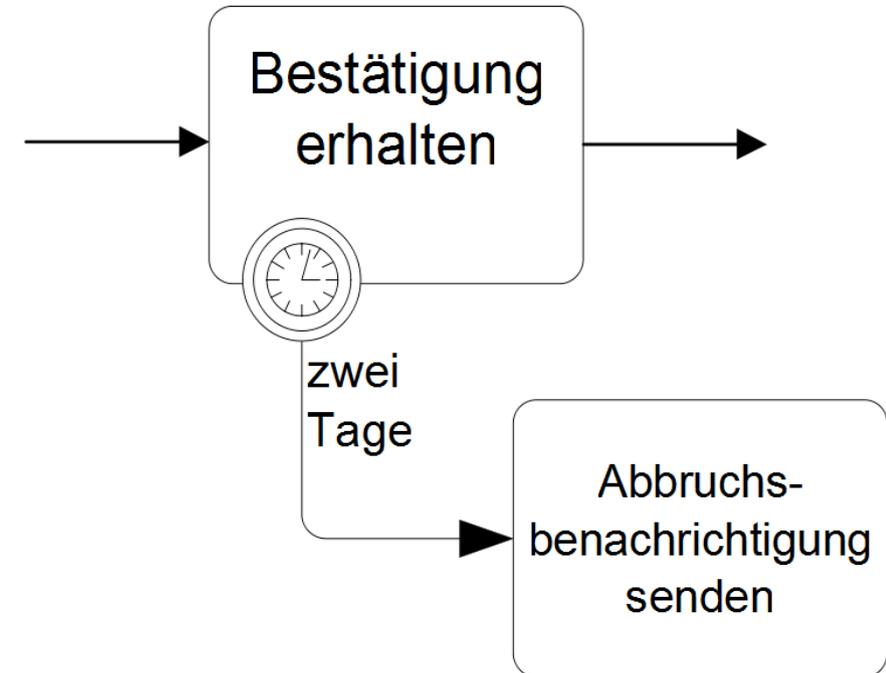
Zeitmesser (Timer), Fehler oder Nachrichten können **Auslöser** sein.



Zeitmesser: Prozessfluss vs. Aktivitätsrand

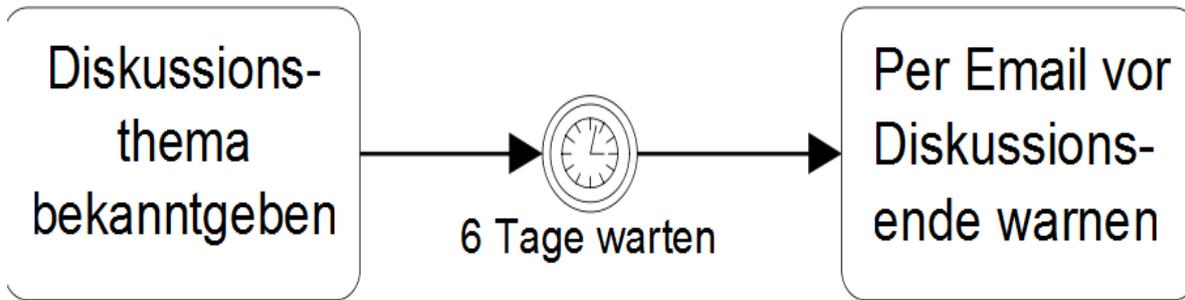


Zeitmesser um Verzögerungen
in einem Prozess zu realisieren

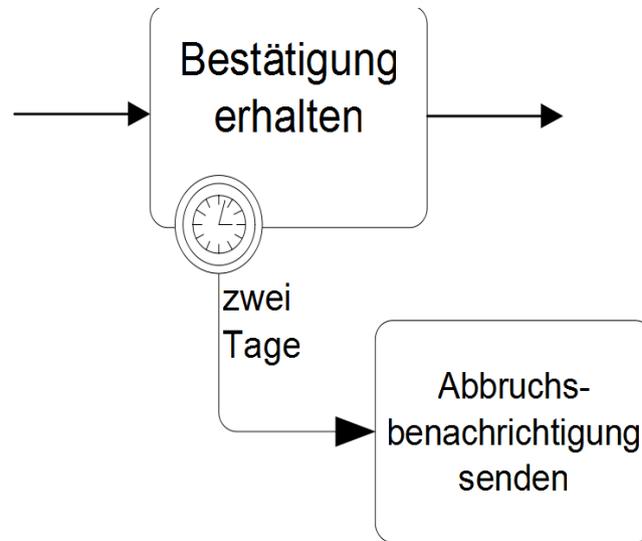


Zeitüberschreitungen um Fehler
zu behandeln

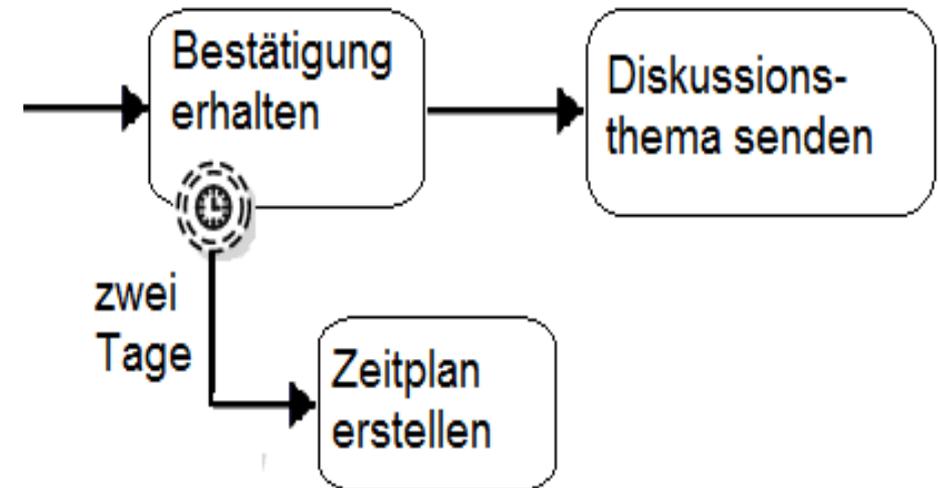
Zeitmesser: Prozessfluss vs. Aktivitätsrand



Zeitmesser um Verzögerungen in einem Prozess zu realisieren



Zeitüberschreitungen um Fehler zu behandeln



nicht-unterbrechendes angeheftetes Zwischenereignis

Angehängte Zwischenereignisse (Intermediate Events)

- An Rand der Aktivität anhängbar.
- Darstellung: nicht ausgefülltes Symbol.
- Reagiert bei **angehängten Zwischenereignissen** auf entsprechendes Ereignis (z.B. Erhalt einer Nachricht).
- Aktivität bei
 - Durchgezogenem Rand abbrechen.
 - Gestricheltem Rand nicht abbrechen.
- Verschiedene **Auslöser** spezifizieren Umstände des Ereignisses (wie bei Startereignissen) sowie:
 - Abbruch, Fehler und Kompensation

Nachricht



Zeitmesser



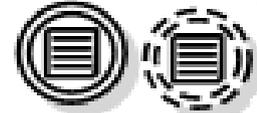
Fehler



Kompensation



Regel



Link

Mehrfach

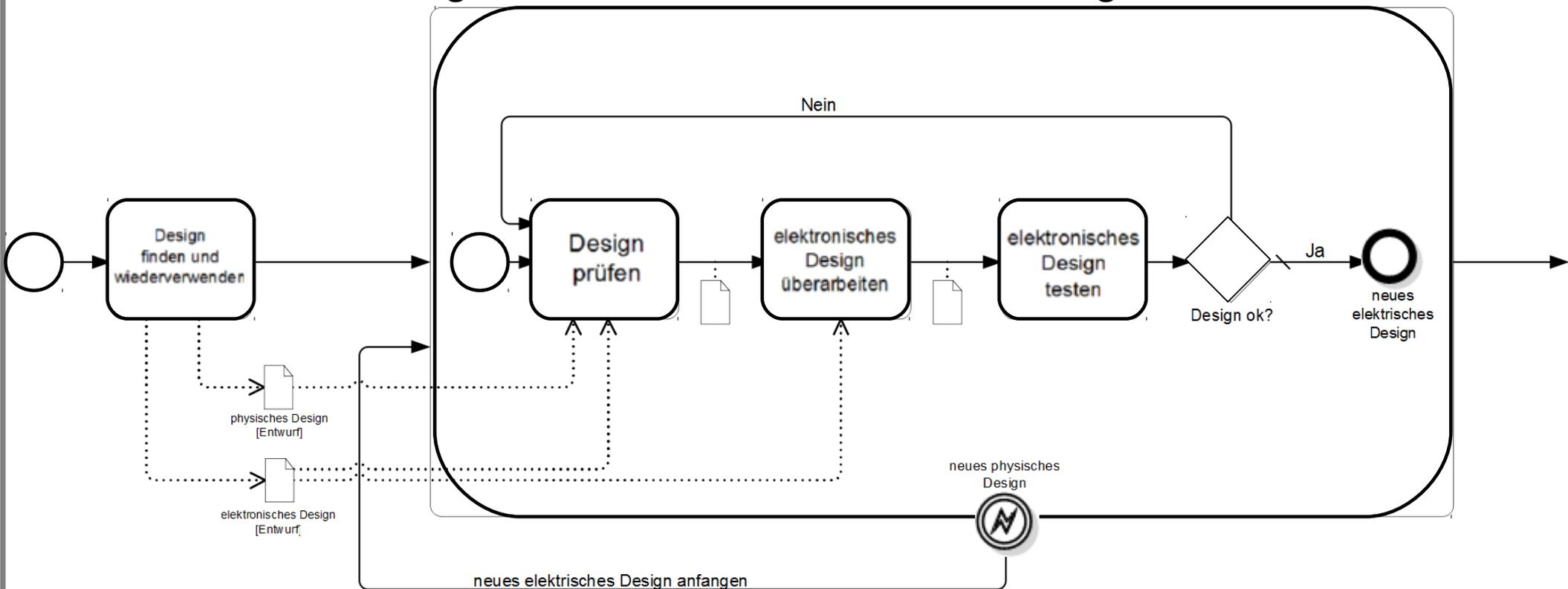


Abbruch



Beispiel für Einsatz eines Fehlerereignisses für Spezifikation der Ausnahmebehandlung (Exception Handling).

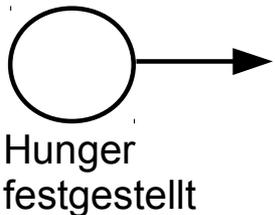
Anmerkung: Namen von **Fehlerereignissen** werden bei **Ausführung** berücksichtigt. → Nach Erreichen des Endereignisses „neues elektrisches Design“ wird mit **normalem Fluss** fortgesetzt.



- Wir haben Hunger und möchten Pizza bestellen
- dazu müssen wir uns zuerst eine Pizza aussuchen
- können wir uns jedoch nicht entscheiden und das Aussuchen dauert länger als 30 Minuten so kochen wir einfach Pasta
- wenn das nicht der Fall ist, bestellen wir die Pizza
- und warten bis wir die Pizza erhalten haben
- in beiden Fällen essen wir zuletzt die Mahlzeit

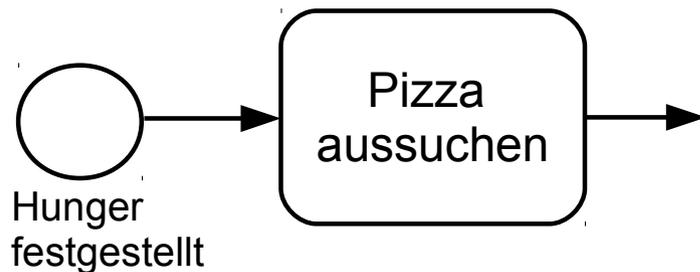
Ein weiteres Pizza Beispiel

- Wir haben Hunger und möchten Pizza bestellen
- dazu müssen wir uns zuerst eine Pizza aussuchen
- können wir uns jedoch nicht entscheiden und das Aussuchen dauert länger als 30 Minuten so kochen wir einfach Pasta
- wenn das nicht der Fall ist, bestellen wir die Pizza
- und warten bis wir die Pizza erhalten haben
- in beiden Fällen essen wir zuletzt die Mahlzeit



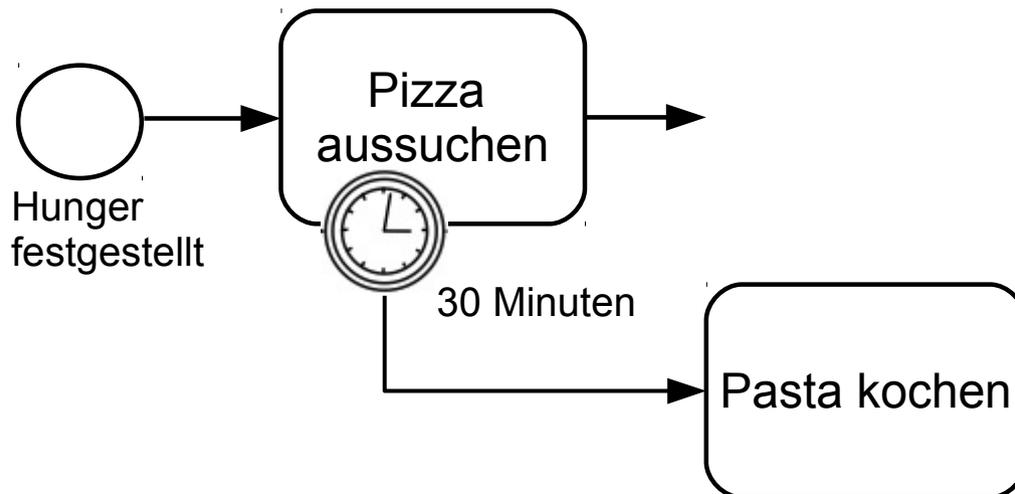
Ein weiteres Pizza Beispiel

- Wir haben Hunger und möchten Pizza bestellen
- dazu müssen wir uns zuerst eine Pizza aussuchen
- können wir uns jedoch nicht entscheiden und das Aussuchen dauert länger als 30 Minuten so kochen wir einfach Pasta
- wenn das nicht der Fall ist, bestellen wir die Pizza
- und warten bis wir die Pizza erhalten haben
- in beiden Fällen essen wir zuletzt die Mahlzeit



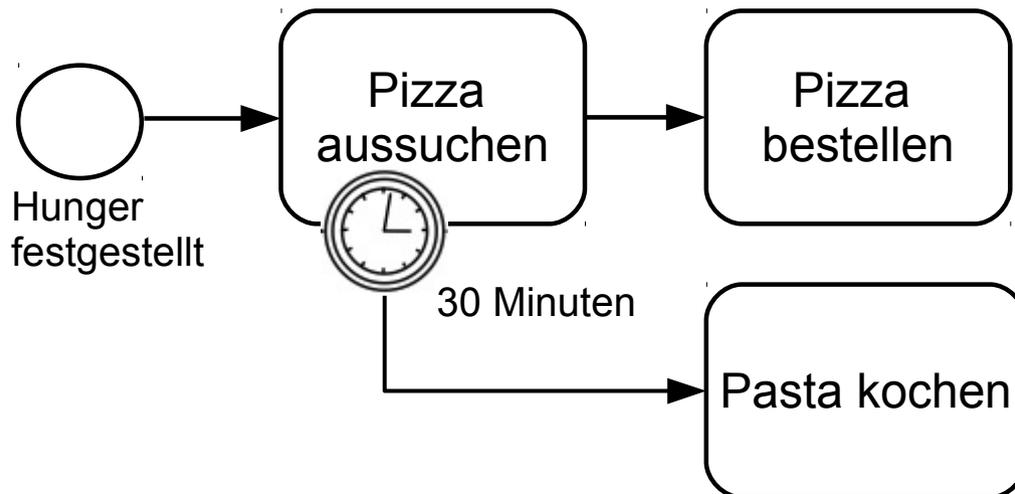
Ein weiteres Pizza Beispiel

- Wir haben Hunger und möchten Pizza bestellen
- dazu müssen wir uns zuerst eine Pizza aussuchen
- können wir uns jedoch nicht entscheiden und das Aussuchen dauert länger als 30 Minuten so kochen wir einfach Pasta
- wenn das nicht der Fall ist, bestellen wir die Pizza
- und warten bis wir die Pizza erhalten haben
- in beiden Fällen essen wir zuletzt die Mahlzeit



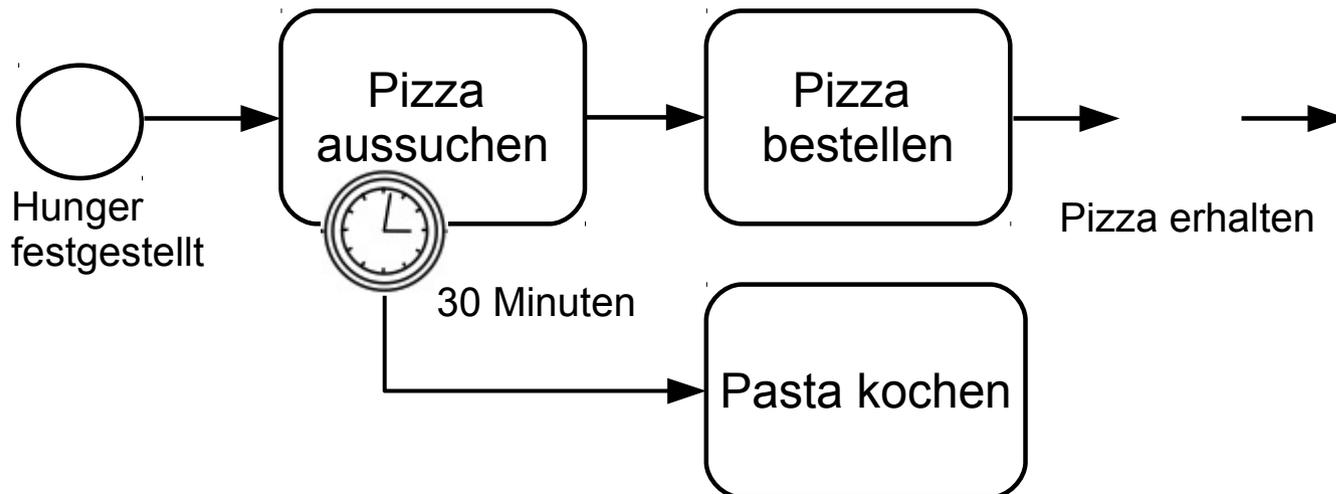
Ein weiteres Pizza Beispiel

- Wir haben Hunger und möchten Pizza bestellen
- dazu müssen wir uns zuerst eine Pizza aussuchen
- können wir uns jedoch nicht entscheiden und das Aussuchen dauert länger als 30 Minuten so kochen wir einfach Pasta
- wenn das nicht der Fall ist, bestellen wir die Pizza
- und warten bis wir die Pizza erhalten haben
- in beiden Fällen essen wir zuletzt die Mahlzeit



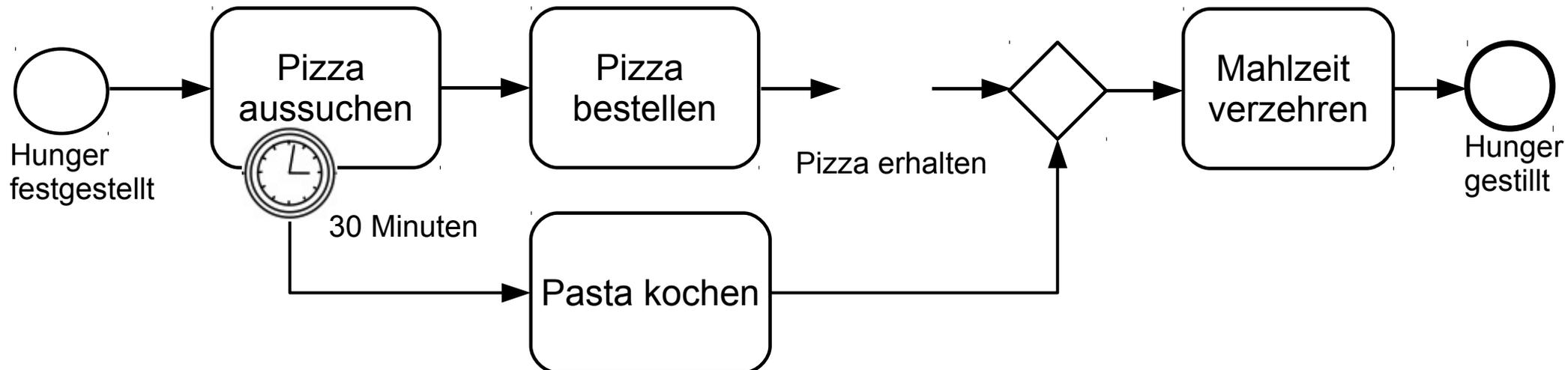
Ein weiteres Pizza Beispiel

- Wir haben Hunger und möchten Pizza bestellen
- dazu müssen wir uns zuerst eine Pizza aussuchen
- können wir uns jedoch nicht entscheiden und das Aussuchen dauert länger als 30 Minuten so kochen wir einfach Pasta
- wenn das nicht der Fall ist, bestellen wir die Pizza
- und warten bis wir die Pizza erhalten haben
- in beiden Fällen essen wir zuletzt die Mahlzeit



Ein weiteres Pizza Beispiel

- Wir haben Hunger und möchten Pizza bestellen
- dazu müssen wir uns zuerst eine Pizza aussuchen
- können wir uns jedoch nicht entscheiden und das Aussuchen dauert länger als 30 Minuten so kochen wir einfach Pasta
- wenn das nicht der Fall ist, bestellen wir die Pizza
- und warten bis wir die Pizza erhalten haben
- in beiden Fällen essen wir zuletzt die Mahlzeit

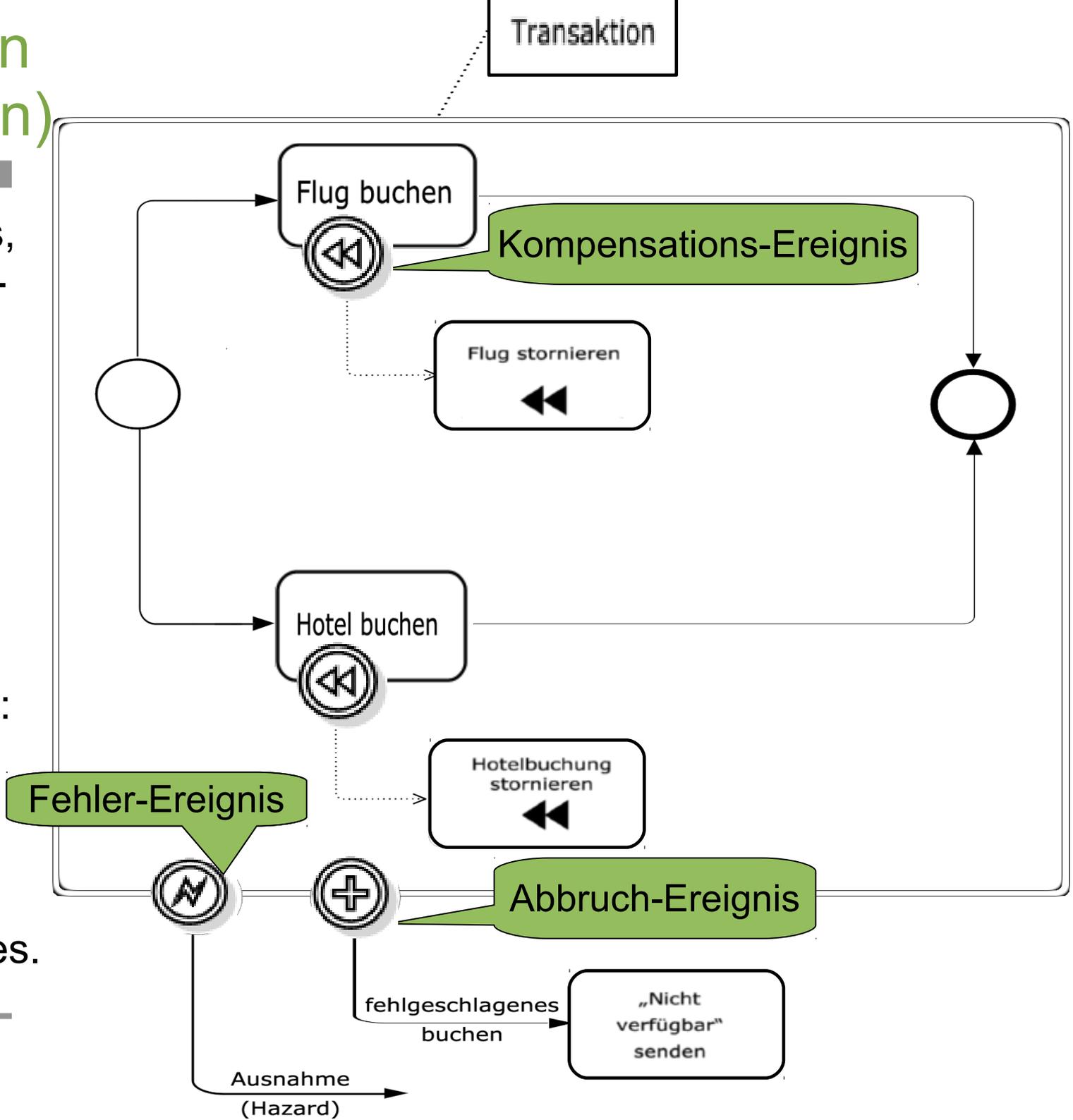


Kompensation (Compensation)

Für **Abbruch**: Fluss, der bei Abbruch-Ereignis startet.

Für **Fehler** (Ausnahme): Fluss, der bei Fehler-Ereignis startet (keine Kompensation).

Für **Kompensation**: Aktivitäten mit entsprechender Markierung außerhalb des normalen Flusses.



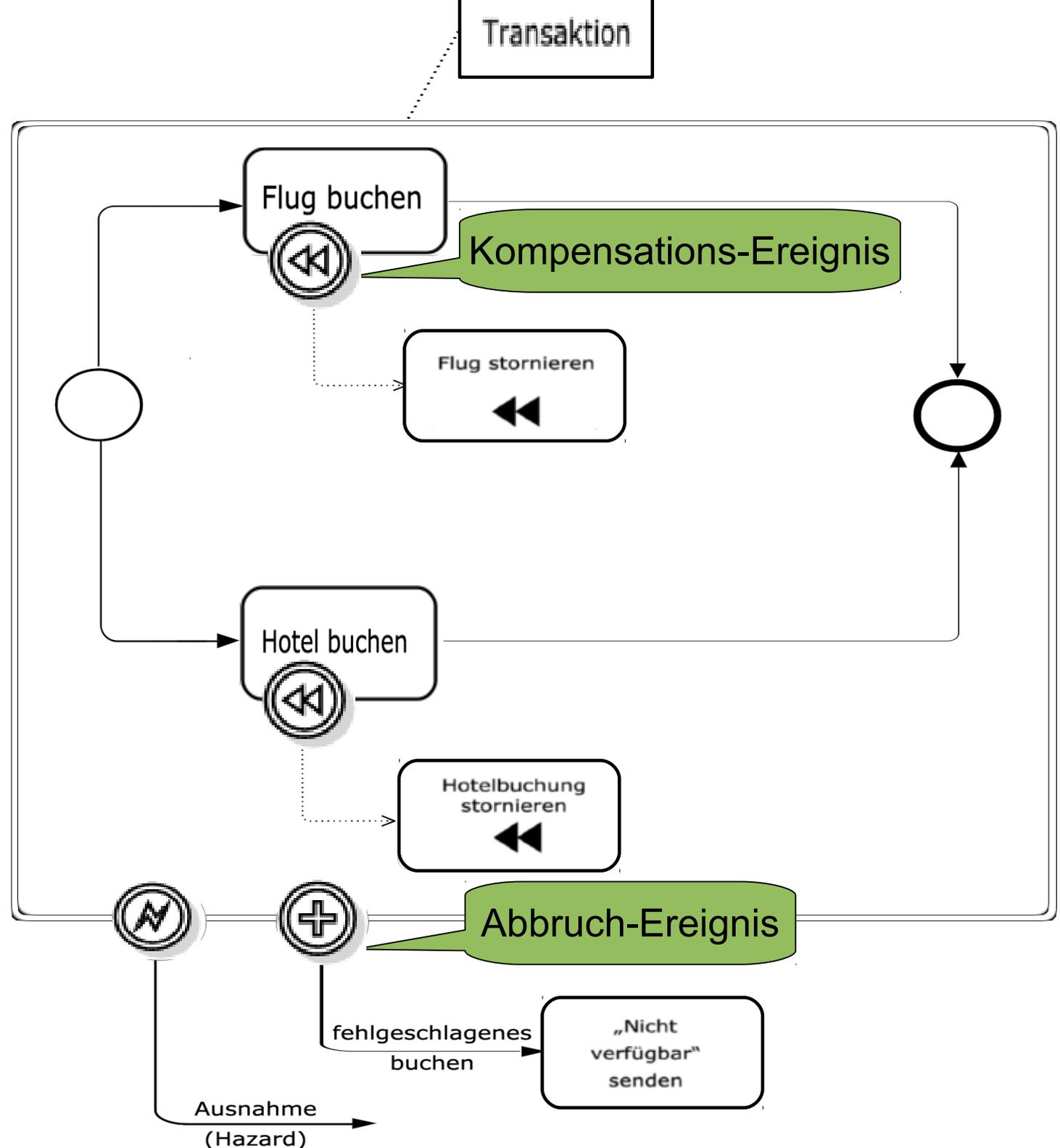
Transaktionen

Transaktionen:
Aktivitäten mit
doppeltem Rand.

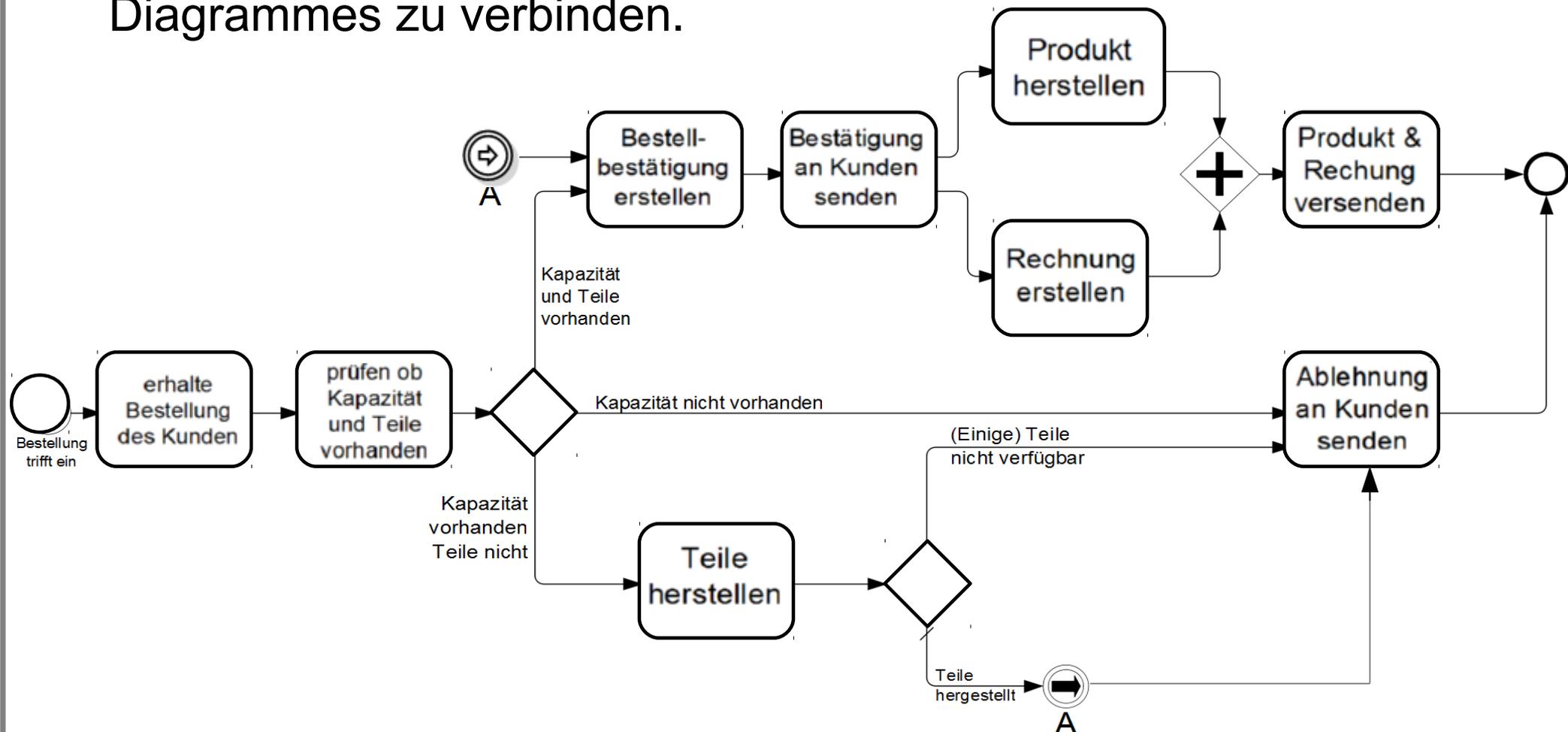
Von Transaktions-
protokoll unterstützt
(z.B.
WS-Transaction).

Bei **erfolgreichem
Abschluss:** weiter
mit normalem
ausgehenden
Sequenzfluss.

Bei **Abbruch:**
Kompensationen
für enthaltene
Aktivitäten
(Atomarität von
Transaktionen)



Link-(Verbindungs-)Ereignisse: graphische Vereinfachung, um Sequenzfluss-Konnektoren an verschiedenen Teilen des Diagrammes zu verbinden.



Beendende Ereignisse (End Events)

- **Beendende Ereignisse** zeigen Prozessende an.
- Verschiedene Arten von Ereignissen zeigen, warum Prozess beendet wird. Analog zu Zwischenereignissen. Insbesondere:
 - „**Nichts**“-Ereignisse: Um Ende eines Unterprozesses zu definieren, oder wenn Ende undefiniert ist.
 - **Terminierung**: Terminiert alle noch aktiven Teile des Prozesses (vgl. [FR12] Abs. 2.6.7).
 - **Abbruch**: Nur im Kontext von Transaktionen (vgl. [FR12] Abs. 2.6.13).
 - Beendendes Ereignis „Link“ aus BPMN 1.0 gibt es in BPMN 2.0 nicht mehr.

Nichts



Nachricht



Fehler



Kompensation



Terminierung



Abbruch



Mehrfach



Sequenzfluss



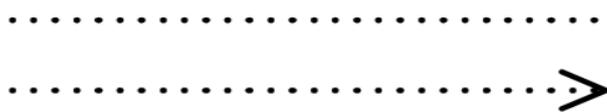
- **Sequenzfluss** (Sequence Flow): Reihenfolge der Aktivitäten in Prozess festlegen.

Nachrichtenfluss



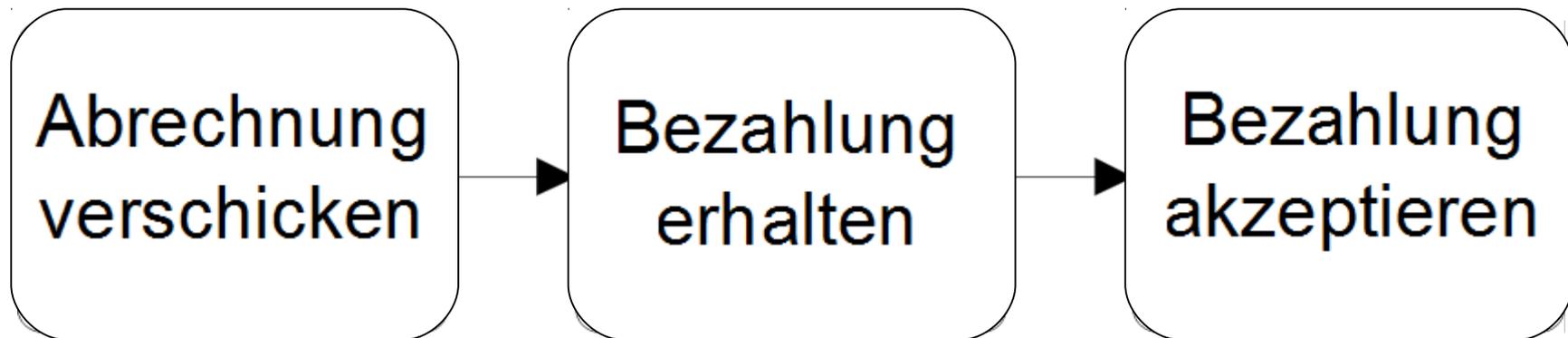
- **Nachrichtenfluss** (Message Flow): Richtung des Flusses zwischen zwei Entitäten, die senden und empfangen können.

Assoziation



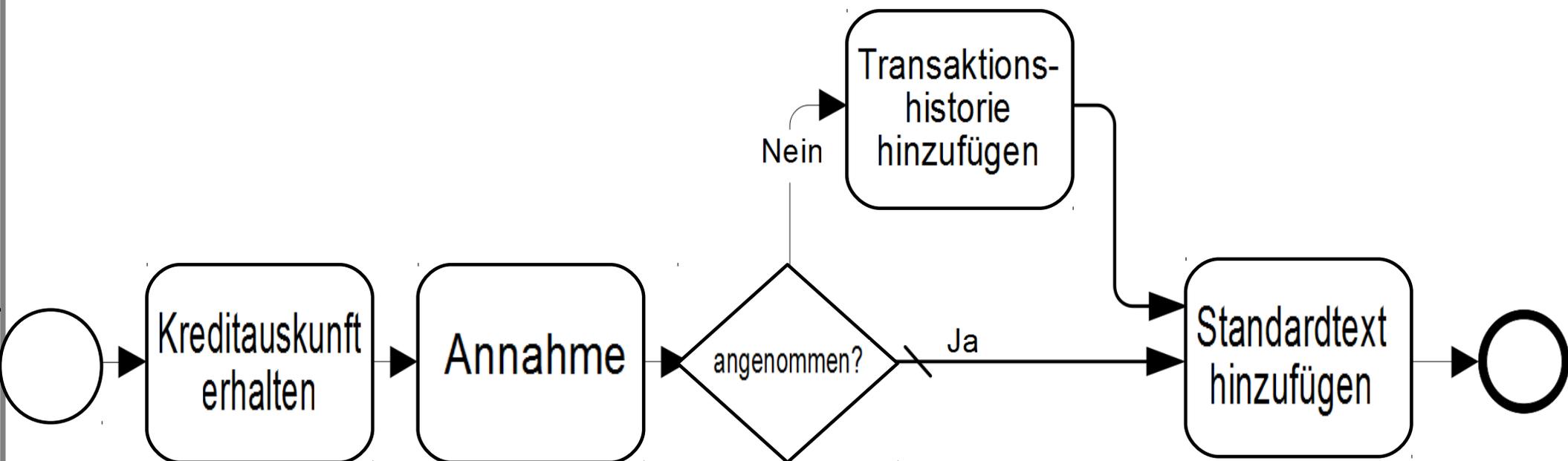
- **Assoziation** (Association): Zuordnung von Daten, Informationen und Artefakte an Flussobjekten.

- **Sequenzfluss:** Reihenfolge der Aktivitäten in Prozess festlegen.
- **Ausgangspunkt** und **Ziel** sind: Ereignisse (Events), Aktivitäten (Activities) oder Gateways.
- Sequenzfluss kann keine **Unterprozess-** oder **Poolgrenzen** überschreiten.



Normaler Sequenzfluss: Fluss, der ...

- ... bei **Startereignis** beginnt, durch Aktivitäten (ggf. alternative oder parallele Pfade) fließt und in **Endereignis** abschließt.
- ... nicht den **Ausnahme-** oder **Kompensationsfluss** beinhaltet.



Bedingter Sequenzfluss (Conditional Sequence Flow)

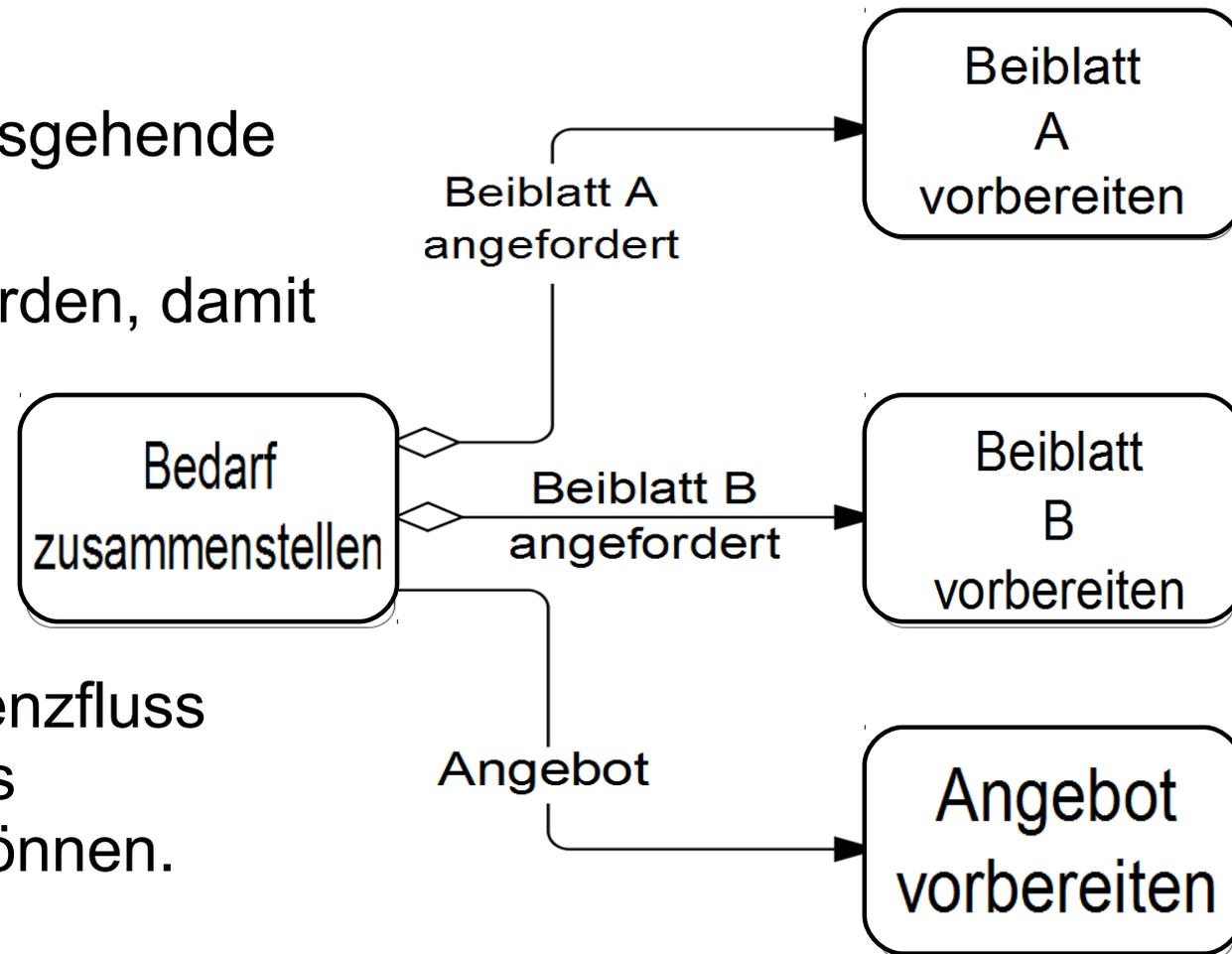
Sequenzfluss KANN definierte **Bedingung** haben, wenn er Aktivität verlässt.

- Aktivität muss **min. 2** ausgehende Sequenzflüsse haben.

Bedingung muss erfüllt werden, damit Fluss fortfahren kann.

- **Diamant-Symbol** zeigt, dass Sequenzfluss Bedingung hat.

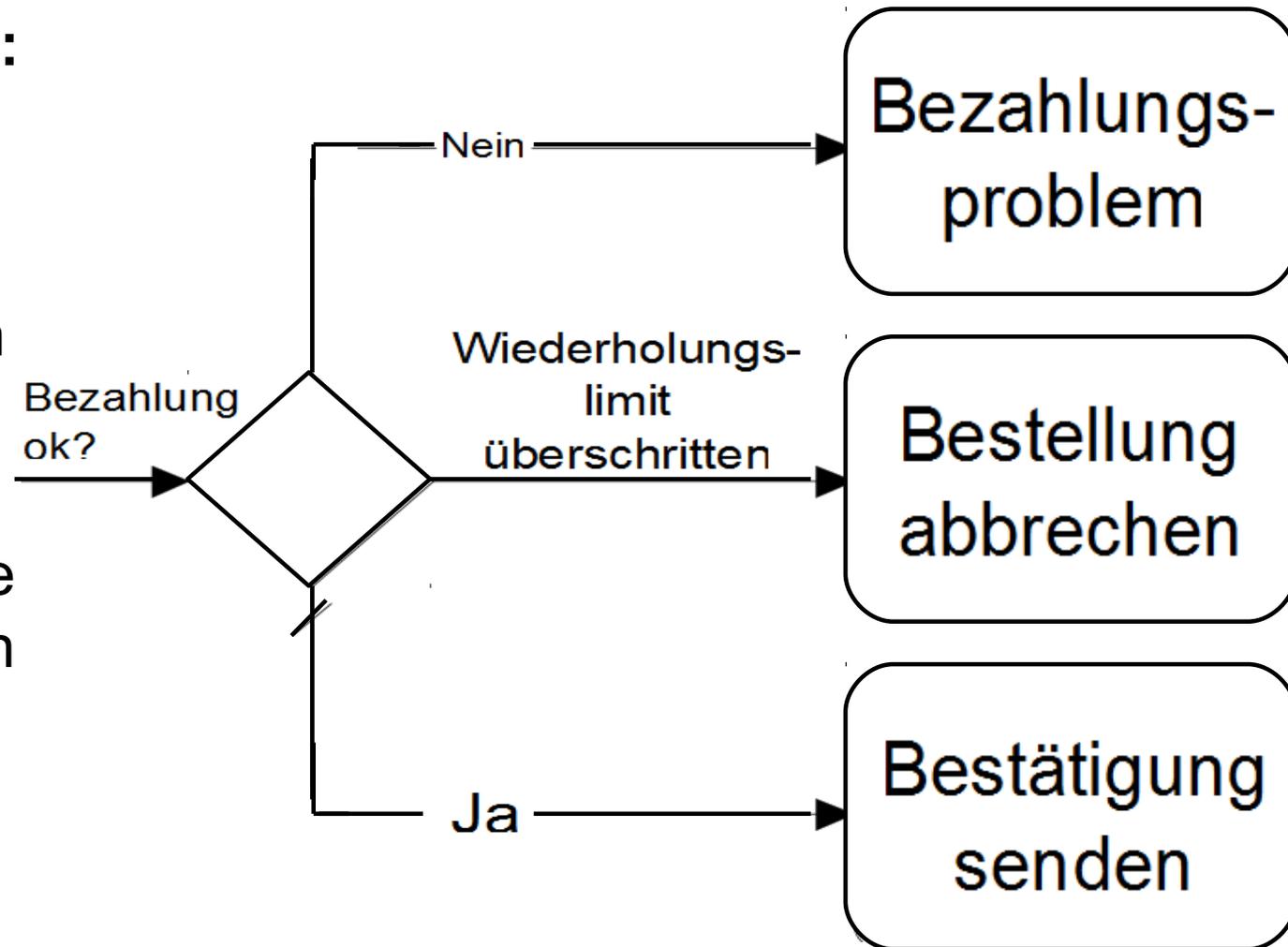
Min. 1 ausgehenden Sequenzfluss während Ausführung des Prozesses auswählen können.



Standard Sequenzfluss (Default Sequence Flow)

Bei **exklusivem oder inklusivem Gateway**:
ausgehenden
Sequenzfluss als
Standardpfad
angeben (Querbalken
durch den Strich).

Standardpfad wird
ausgewählt, wenn alle
anderen Bedingungen
nicht erfüllt.





Was sind die hauptsächlichsten **Einschränkungen** eines Sequenzflusses ?

Was sind die hauptsächlichsten **Einschränkungen** eines Sequenzflusses ?

- Sequenzfluss kann nur Aktivitäten, Ereignisse und Gateways verbinden. Er kann nicht die Grenzen eines Unterprozesses oder Pools überschreiten.

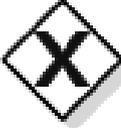
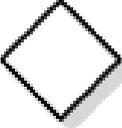
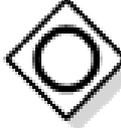
Wie vergleicht sich **BPMN** bezüglich Einschränkungen eines Sequenzflusses mit **EPKs** ?

Was sind die hauptsächlichsten **Einschränkungen** eines Sequenzflusses ?

- Sequenzfluss kann nur Aktivitäten, Ereignisse und Gateways verbinden. Er kann nicht die Grenzen eines Unterprozesses oder Pools überschreiten.

Wie vergleicht sich **BPMN** bezüglich Einschränkungen eines Sequenzflusses mit **EPKs** ?

- Bei EPKs müssen Aktivitäten und Ereignisse sich abwechseln, bei BPMN nicht.

Exklusiv	 oder 
Eventbasiert	
Inklusiv	
Komplex	
Parallel	

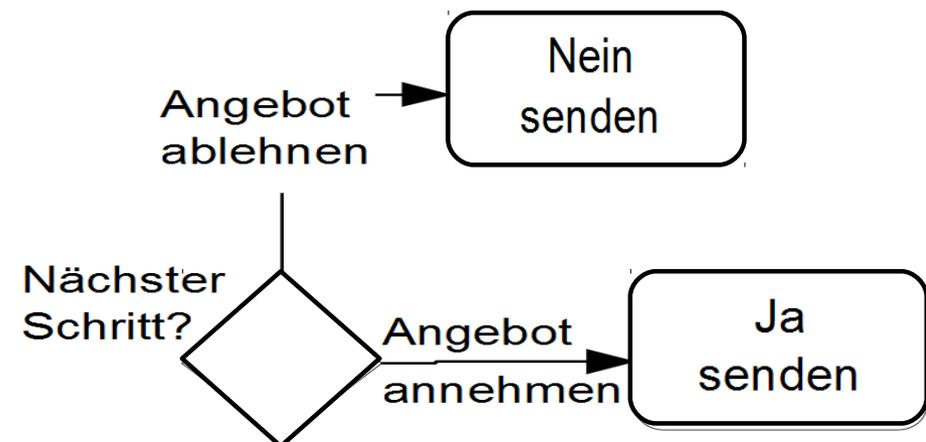
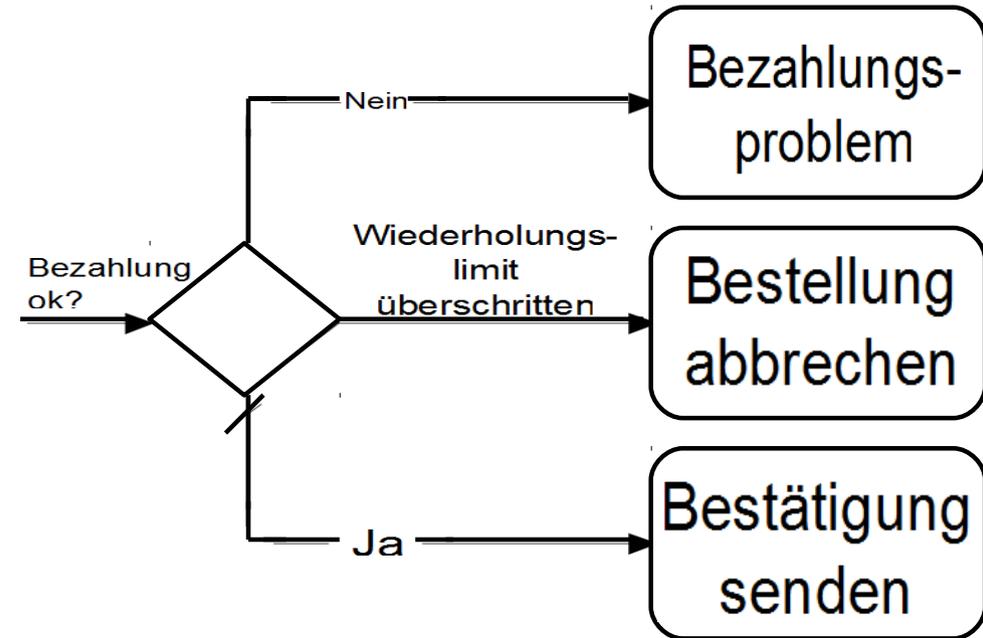
- **Gateway:** Steuert, wie sich Sequenzflüsse beim Vereinigen oder Aufspalten verhalten.
- Symbolisiert durch **Raute / Diamant**.
 - Symbole im Inneren repräsentieren unterschiedliches Verhalten.
 - Alle Gateways können Fluss spalten oder vereinigen.
- Gateway nur notwendig, wenn Fluss kontrolliert werden muss.

Exklusives Gateway

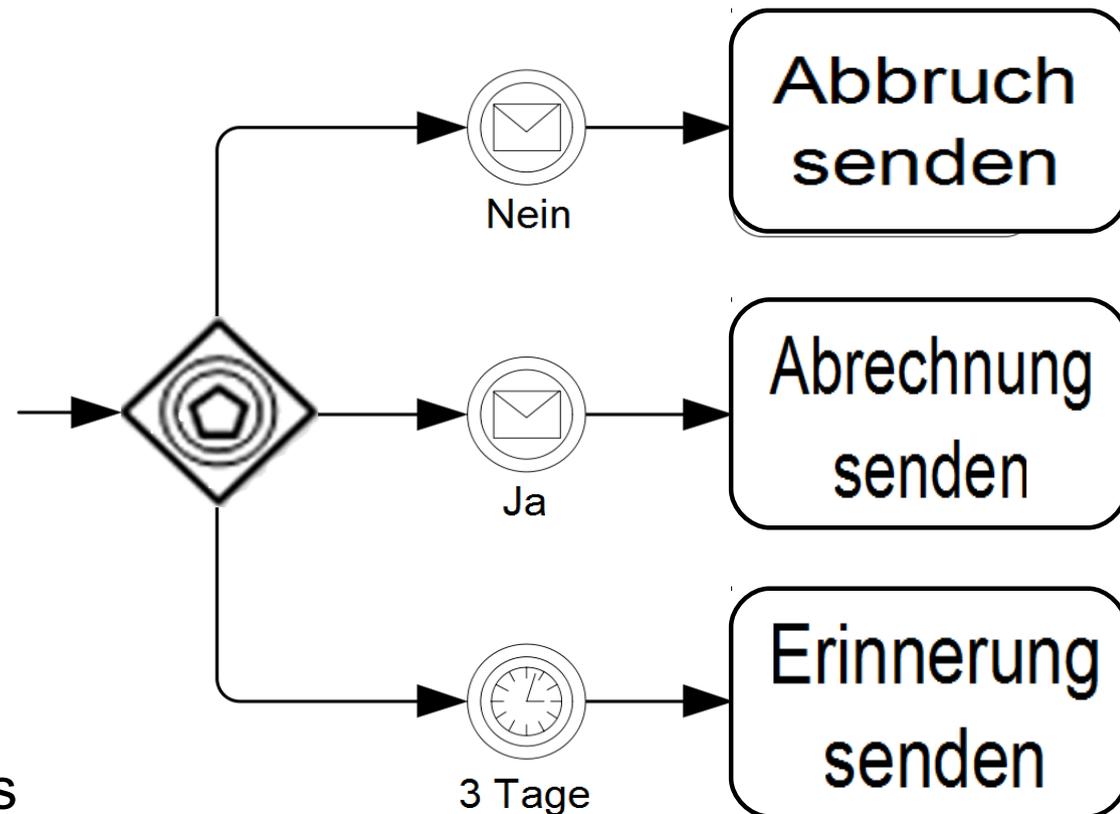
(Entscheidung): Alternative Pfade basierend auf definierten Bedingungen.

Optional symbolisiert durch „X“ im Diamant.

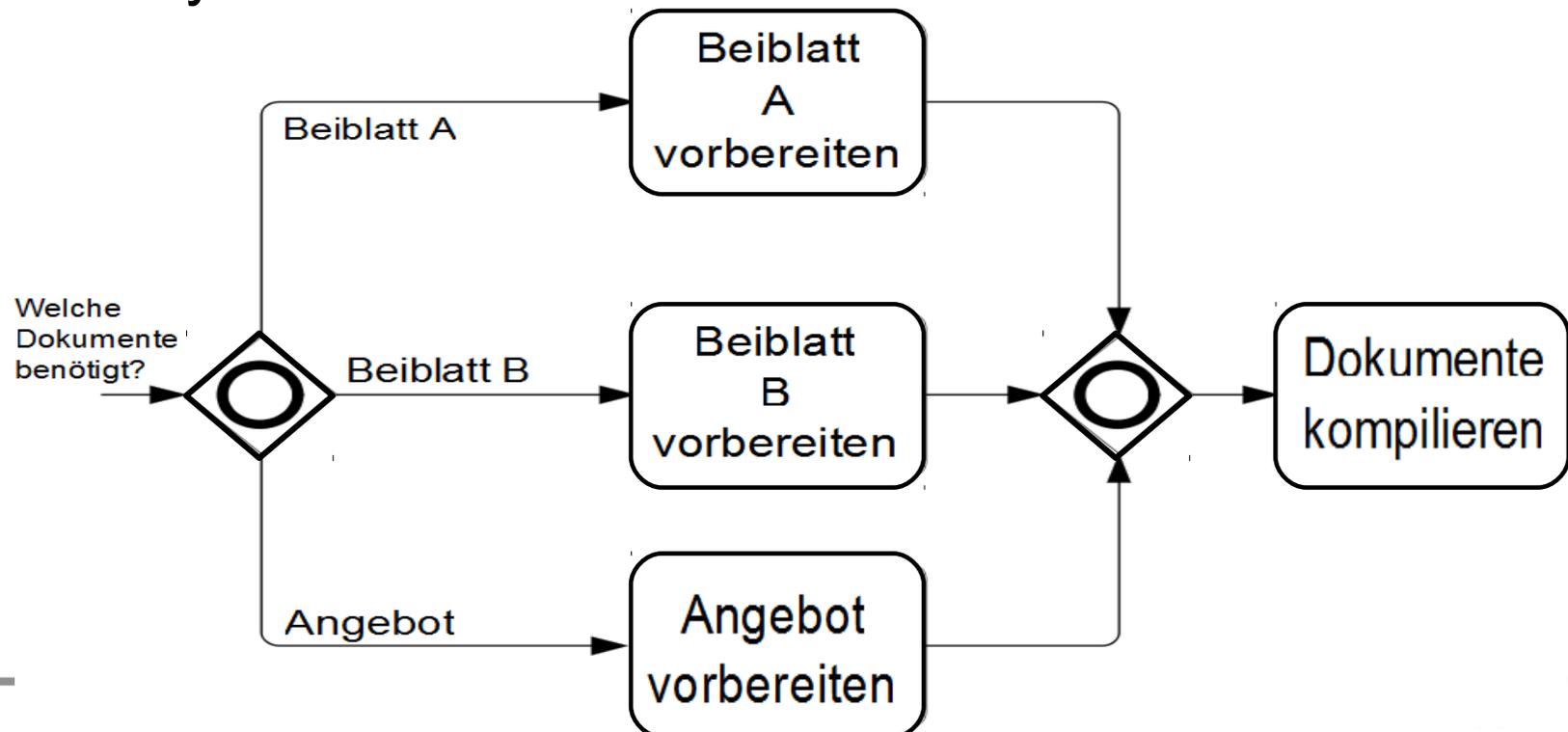
Meist-verbreiteter Typ von Gateways.



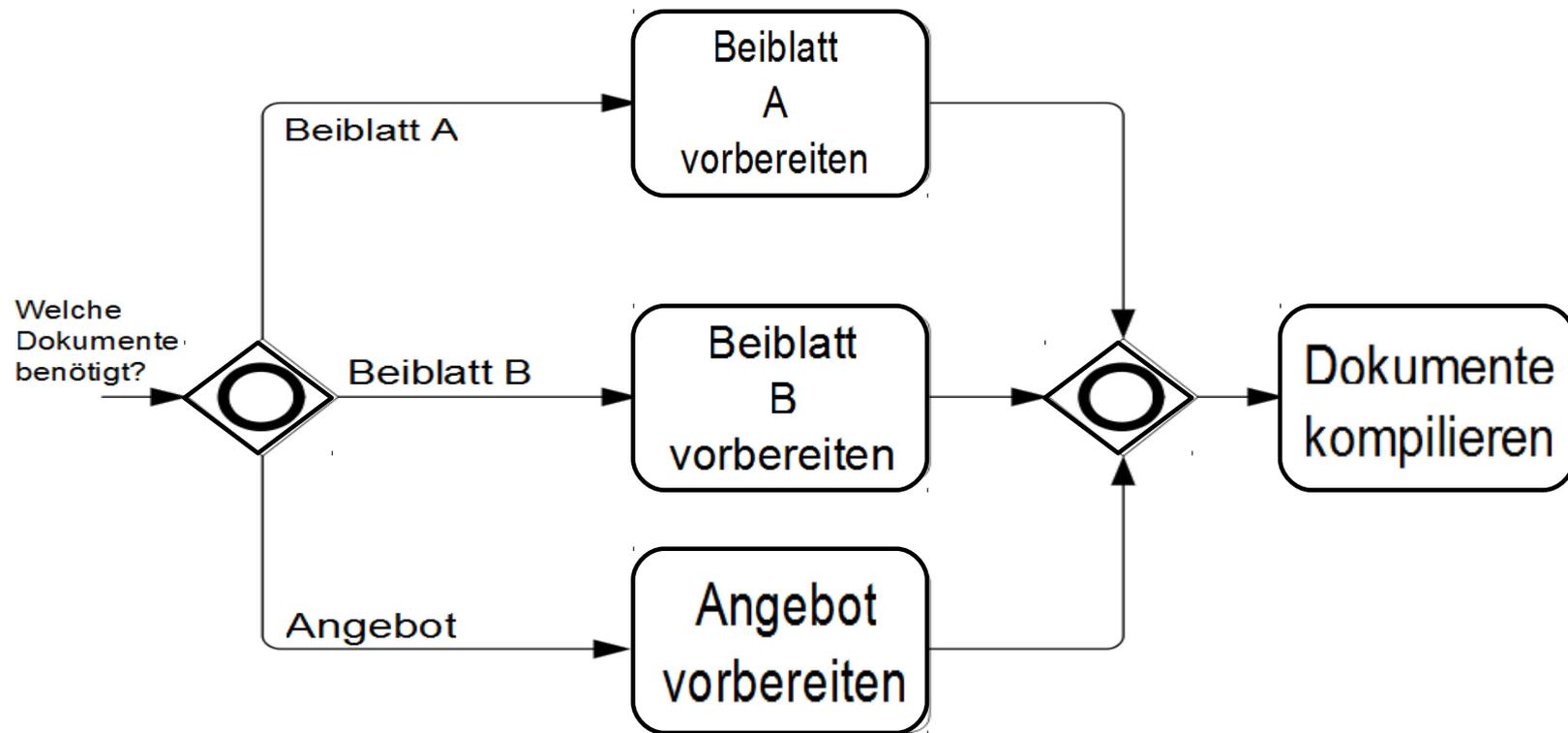
- **Ereignis-basiertes Gateway:** Verzweigungspunkt, wo Alternativen aufgrund von Ereignissen, die hier eintreffen, ausgewählt werden.
- Merkmal: Mehrfaches **parallele** Auftreten von **Zwischenereignissen**.
- **Ereignis**, das auf Gateway-Raute folgt, definiert eingeschlagenen Weg.
 - Erstes eintreffendes Ereignis „gewinnt“ Entscheidung.



- **Inklusives Gateway:** Entscheidungen mit mehr als einem (gleichzeitig) möglichen Ergebnis.
- Identifiziert durch Symbol „O“.
- Normalerweise durch dazugehöriges zusammenführendes inklusives Gateway beendet.

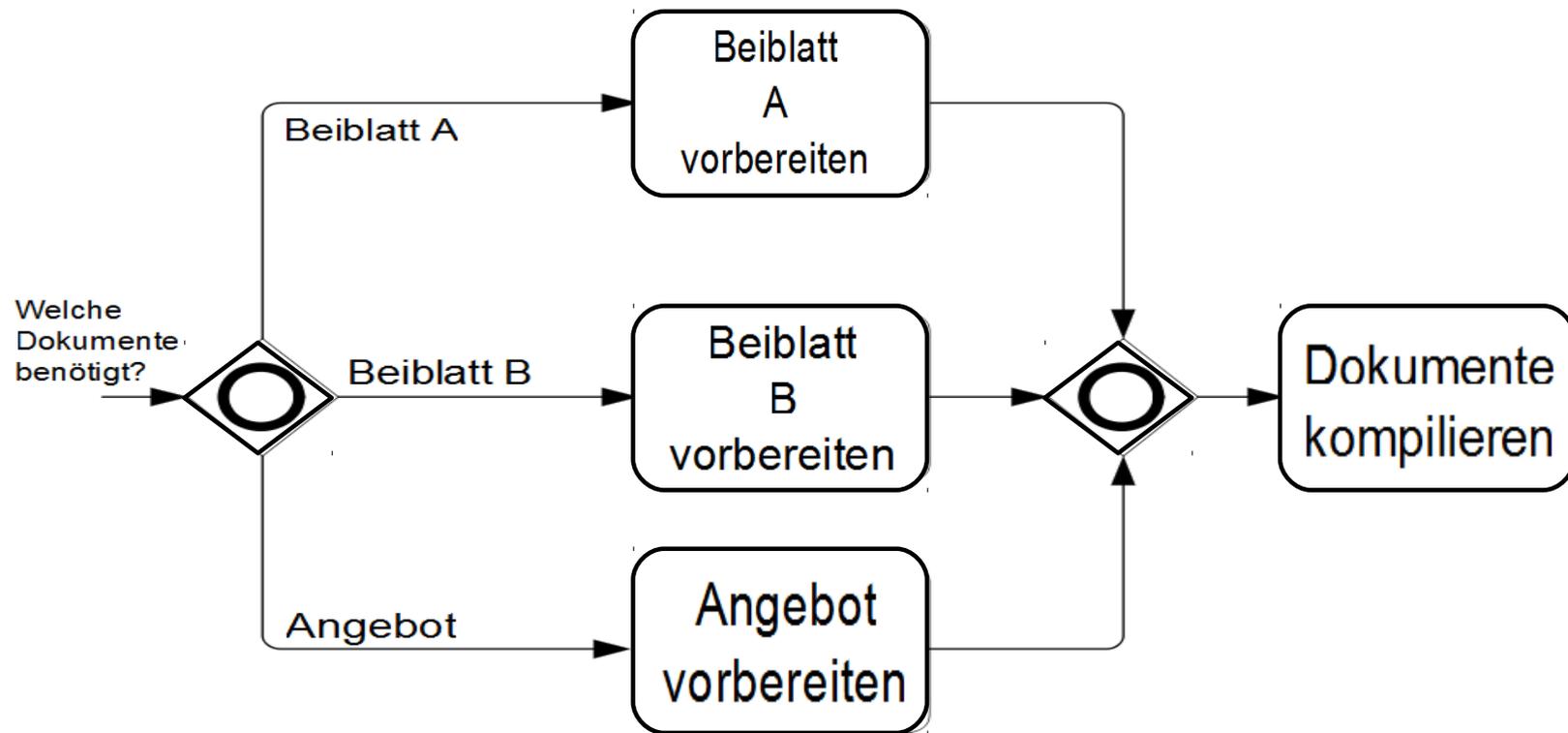


Wieviele **Beiblätter** müssen vorbereitet werden, bevor die Dokumente kompiliert werden ?

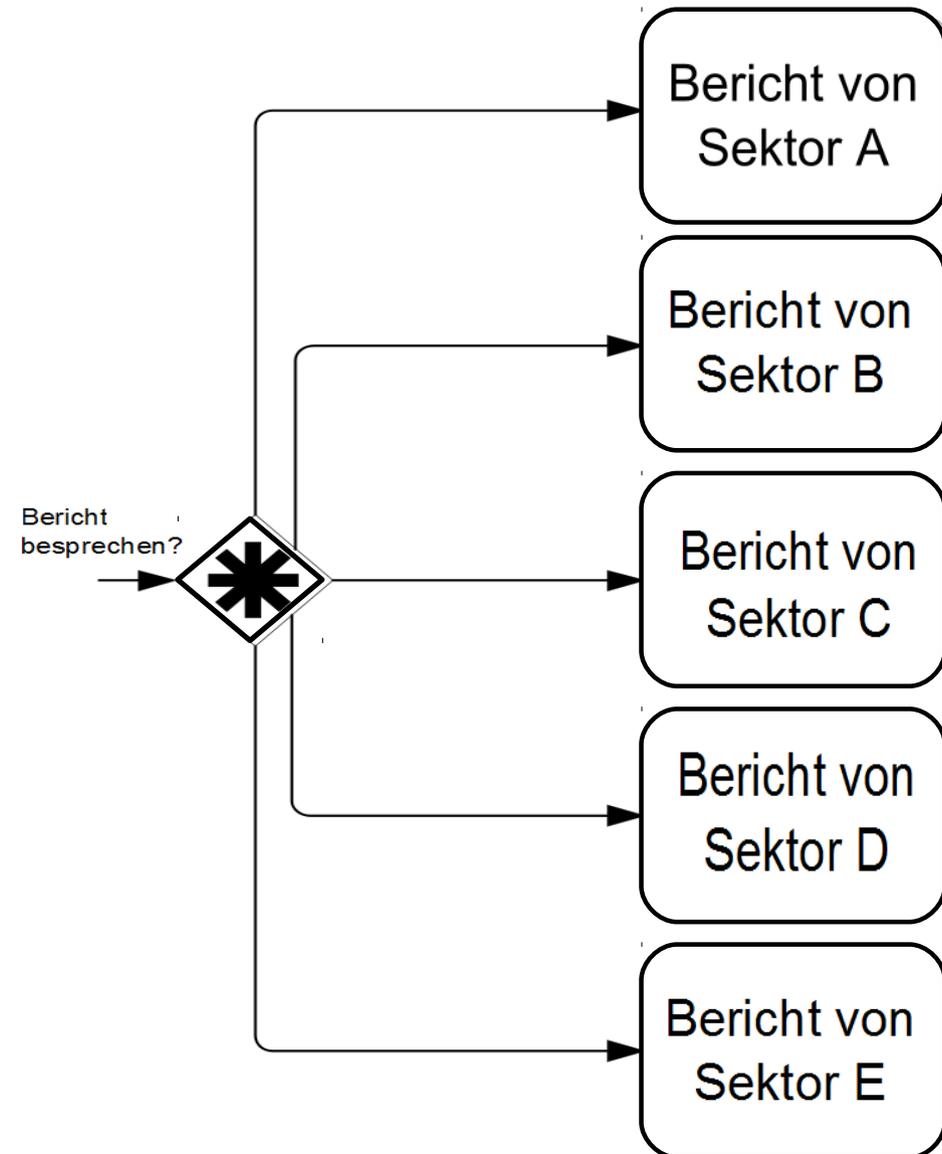


Wieviele **Beiblätter** müssen vorbereitet werden, bevor die Dokumente kompiliert werden ?

Beim inklusiven Gateway muss man nicht alle Flüsse durchlaufen. Es reicht also ggf. nur **ein Angebot** und kein Beiblatt vorzubereiten.



- **Komplexes Gateway:** Entscheidung, an der detailliertere Definitionen des Verhalten angegeben werden kann.
- Gekennzeichnet durch **Sternsymbol**.
- Für **aufspaltende** und **zusammenführende Gateways** definierbar.



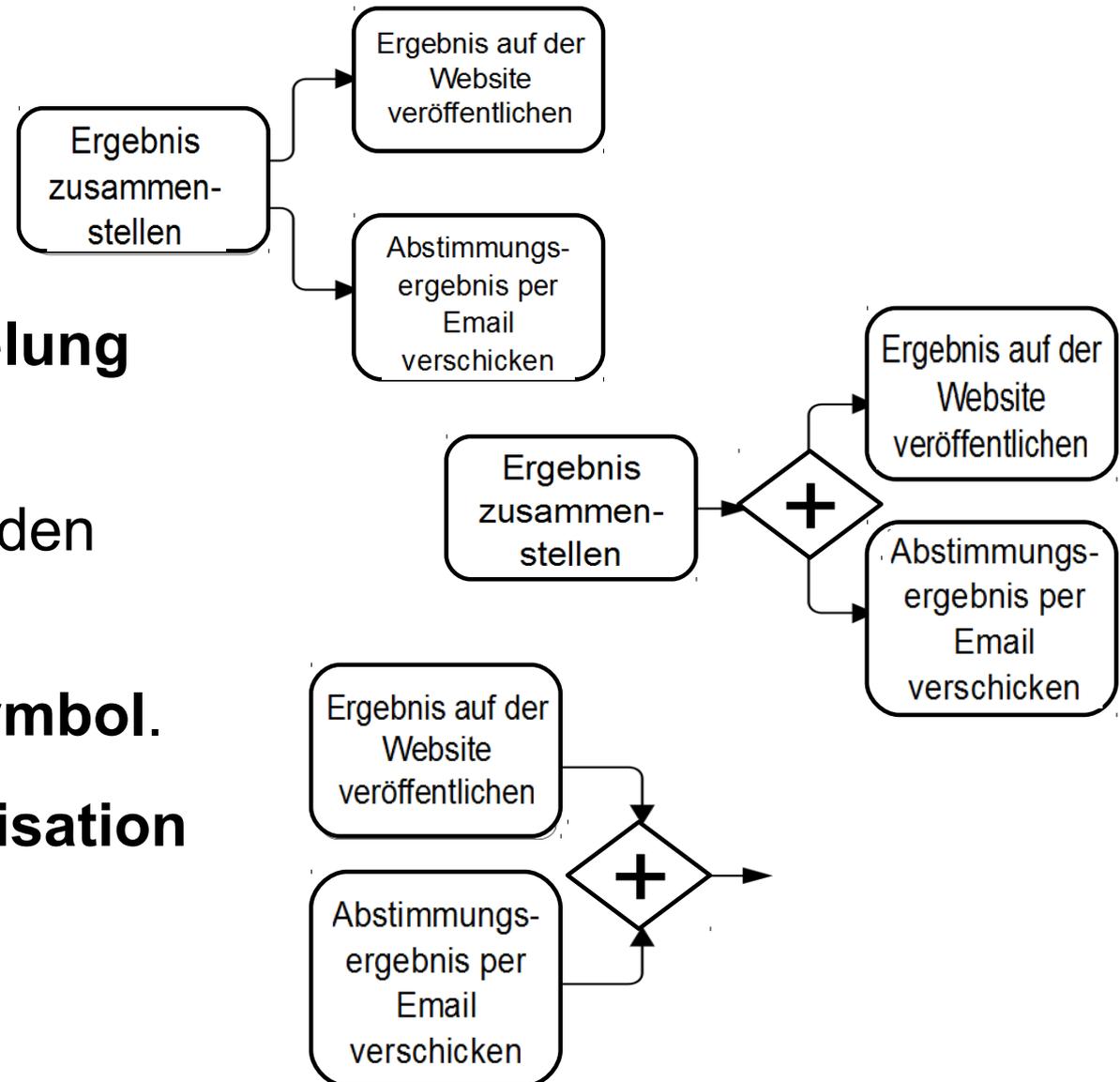
Paralleles Gateway:

Stelle, an der parallele Pfade definiert werden.

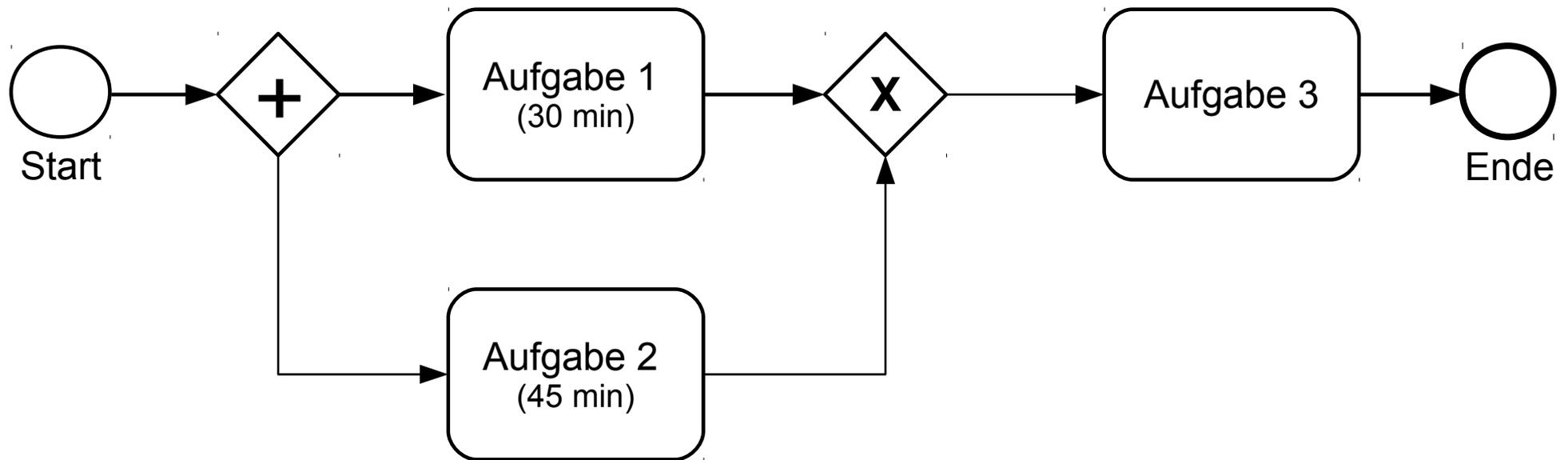
- Meistens nicht zur **Gabelung** des Pfades erforderlich.
- Aus methodischen Gründen verwendbar.

Kennzeichnet durch „+“-Symbol.

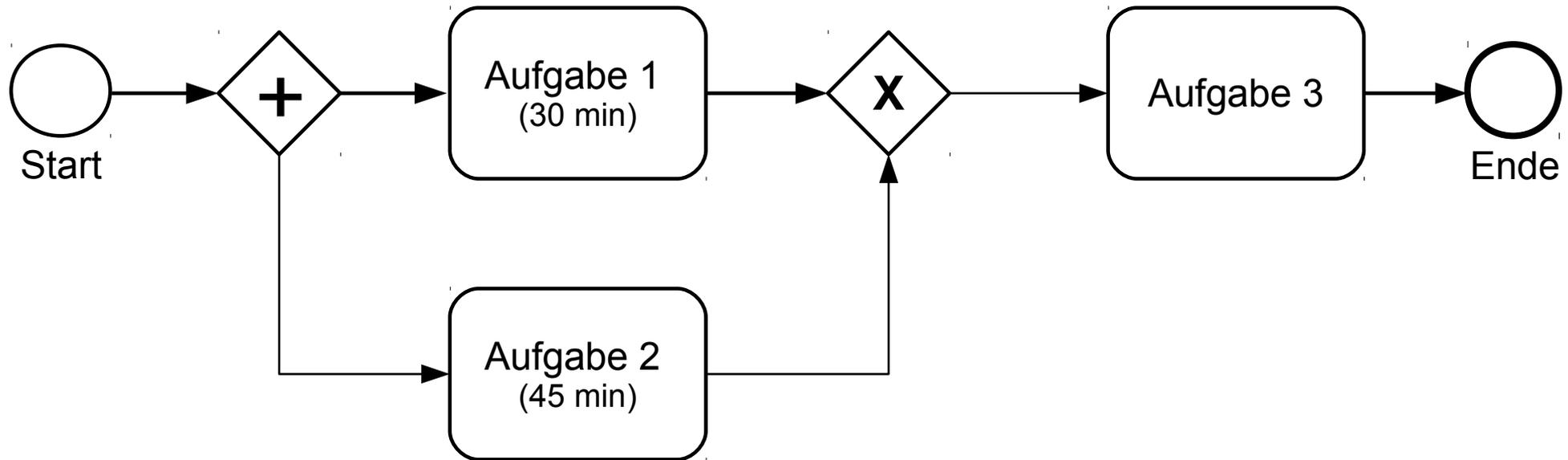
Verwendung zur **Synchronisation** von parallelen Pfaden.



Frage: Was passiert hier mit Aufgabe 3?

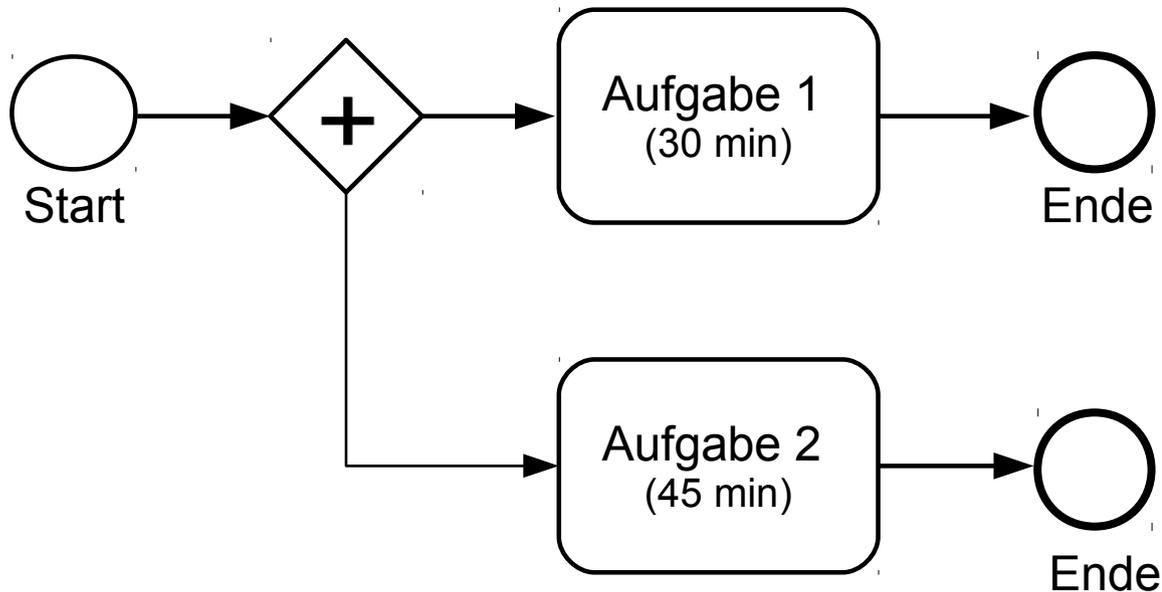


Frage: Was passiert hier mit Aufgabe 3?

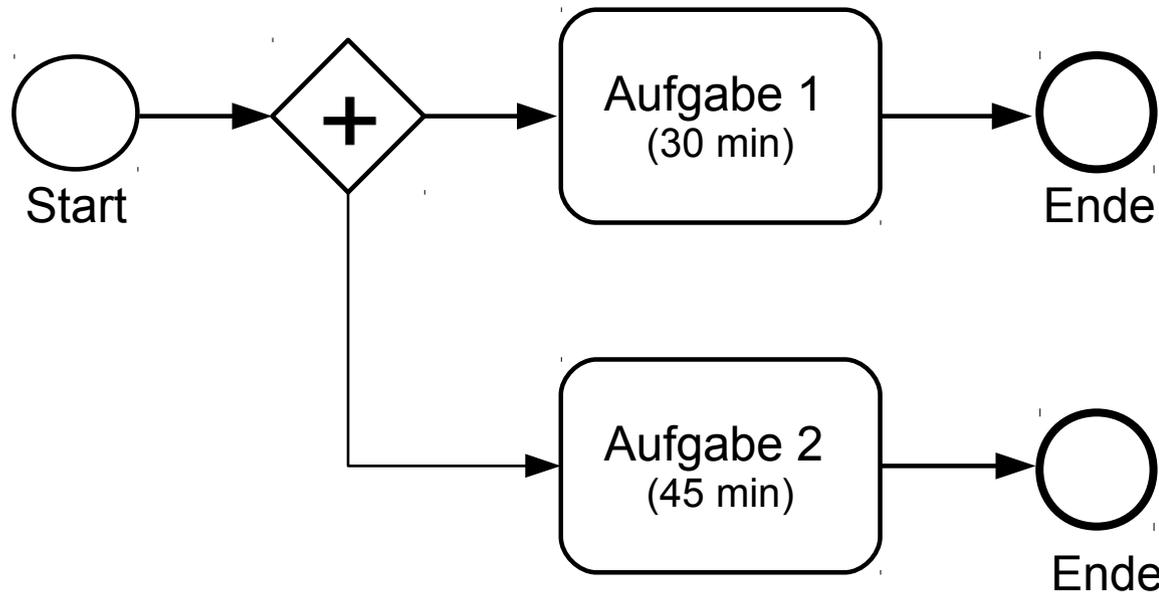


Antwort: Aufgabe 3 wird zwei mal ausgeführt, einmal nach 30 und einmal nach 45 Minuten. **Exklusive** Gateways sind **keine synchronisierenden** Gateways.

Frage: Wie lange „lebt“ die Prozessinstanz?

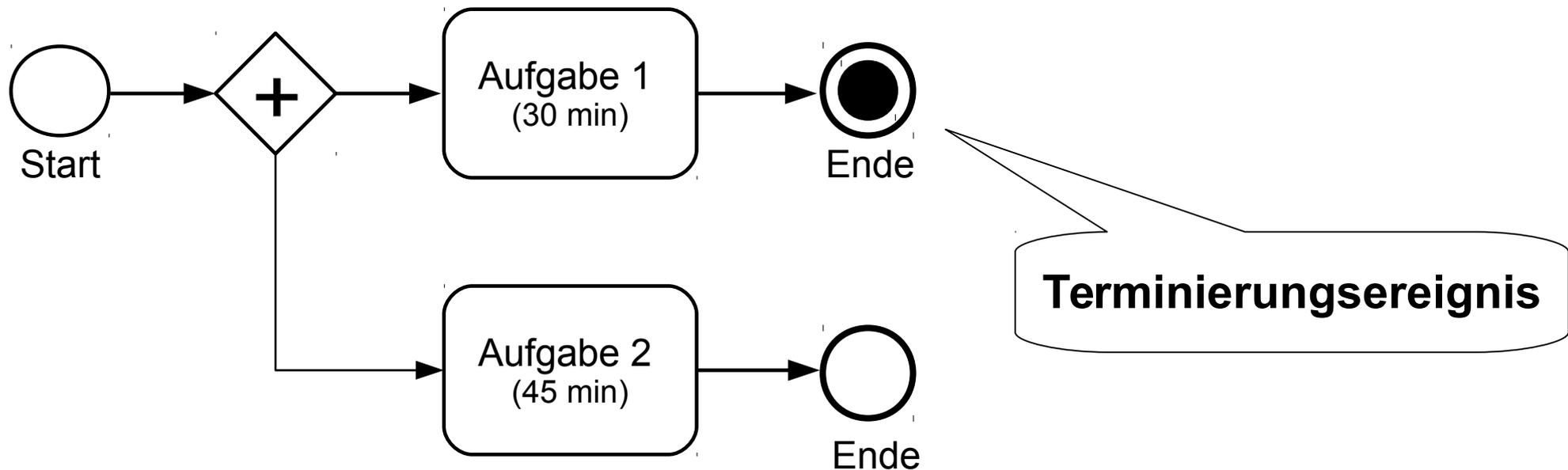


Frage: Wie lange „lebt“ die Prozessinstanz?



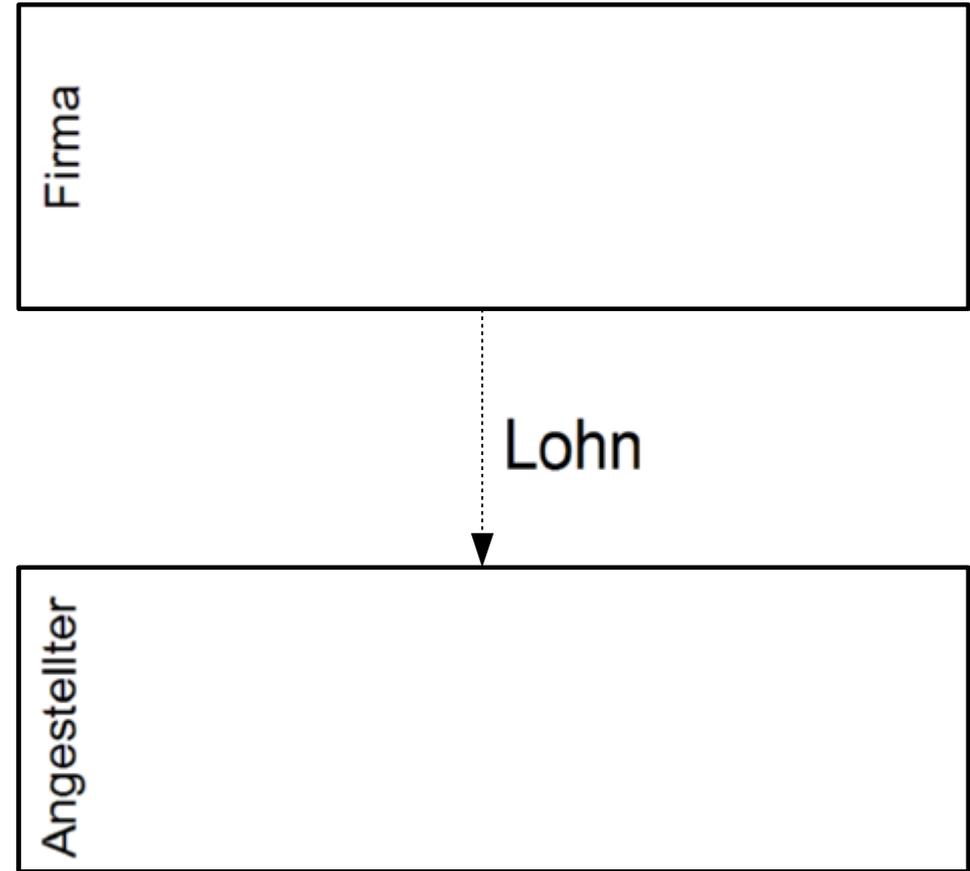
Antwort: 45 Minuten. Eine Prozessinstanz ist erst beendet, wenn alle Token „konsumiert“ wurden.

Frage: Wie lange „lebt“ die Prozessinstanz?

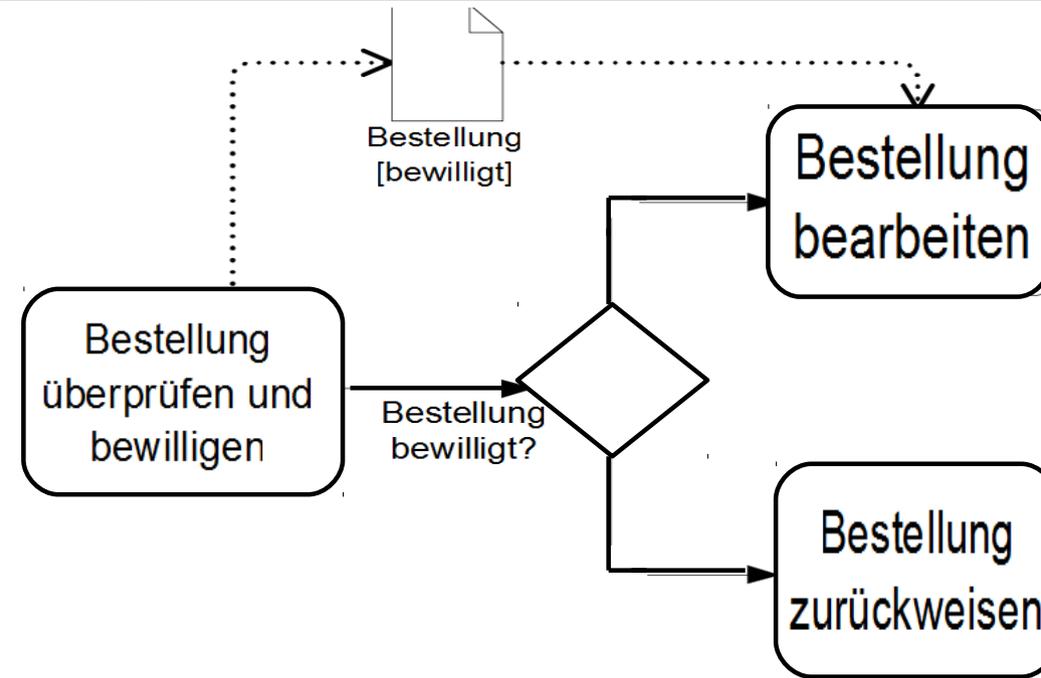


So lebt die Prozessinstanz allerdings nur 30 Minuten.

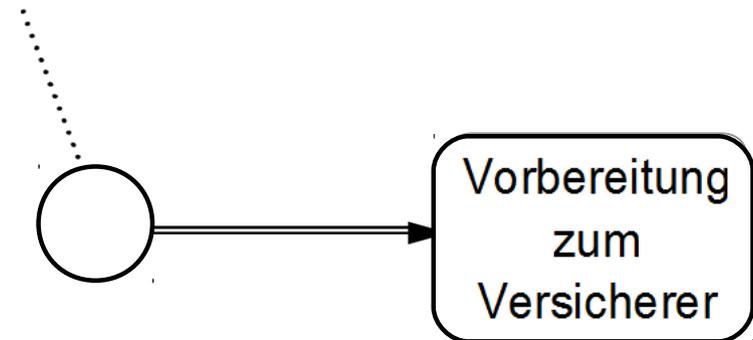
- **Nachrichtenfluss:** Zeigt Richtung der Nachrichten zwischen zwei Teilnehmern des Prozesses an.
 - Teilnehmer werden durch separate Pools repräsentiert.
- Nachrichtenfluss kann an **Rand** eines Pools oder an **Objekt** im Pool gebunden werden.
- Nachrichtenflüsse zwischen **zwei Objekten** im gleichen Pool nicht erlaubt.



- **Assoziation:** Wird benutzt, um zwei Objekte miteinander zu verbinden (z.B. Artefakte und Aktivitäten).
 - Assoziationen können angeben, wie Daten eine **Aktivität** verlassen oder wie sie eingehen.
- Textuelle Anmerkungen am **Objekt** zuweisbar.



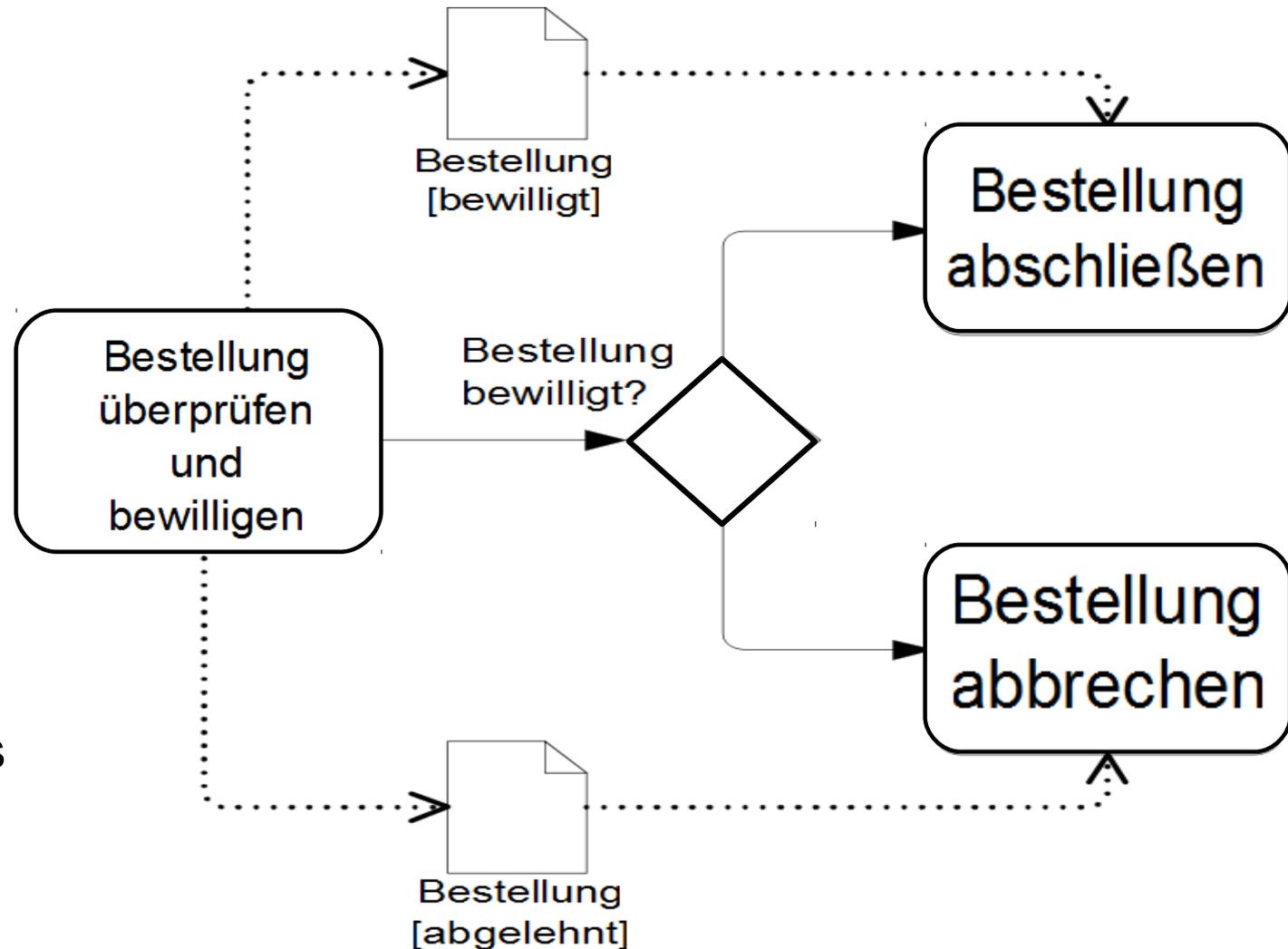
Aus der
Werbung



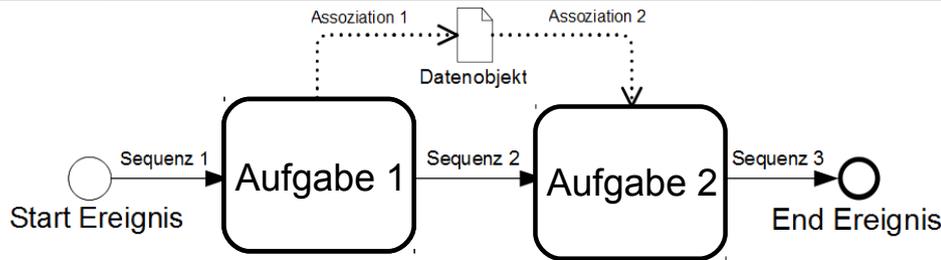
- **BPMN am Beispiel**
- **BPMN-Hintergrundwissen**
- **BPMN im Detail**
 - Grundlegende **Konzepte**
 - Weiterführende **Konzepte**

- **Artefakte:** zeigen Informationen, die über Flussdiagramm-Struktur des Prozesses hinausgehen.
- Drei Standard-Artefakte in BPMN:
 - **Datenobjekte** (Data Objects)
 - **Gruppen** (Groups)
 - **Anmerkungen** (Annotations)
- Anwender kann BPMN durch Hinzufügen neuer Artefakte erweitern.

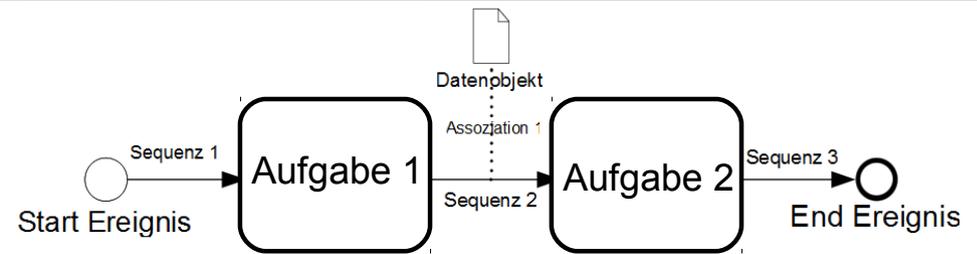
- **Datenobjekte:** zeigen Nutzung der Daten im Objekt des Prozesses.
- Können **In-** und **Output** einer Aktivität definieren.
- Können **Status** haben, der anzeigt, wie Dokument im Laufe des Prozesses verändert wird.



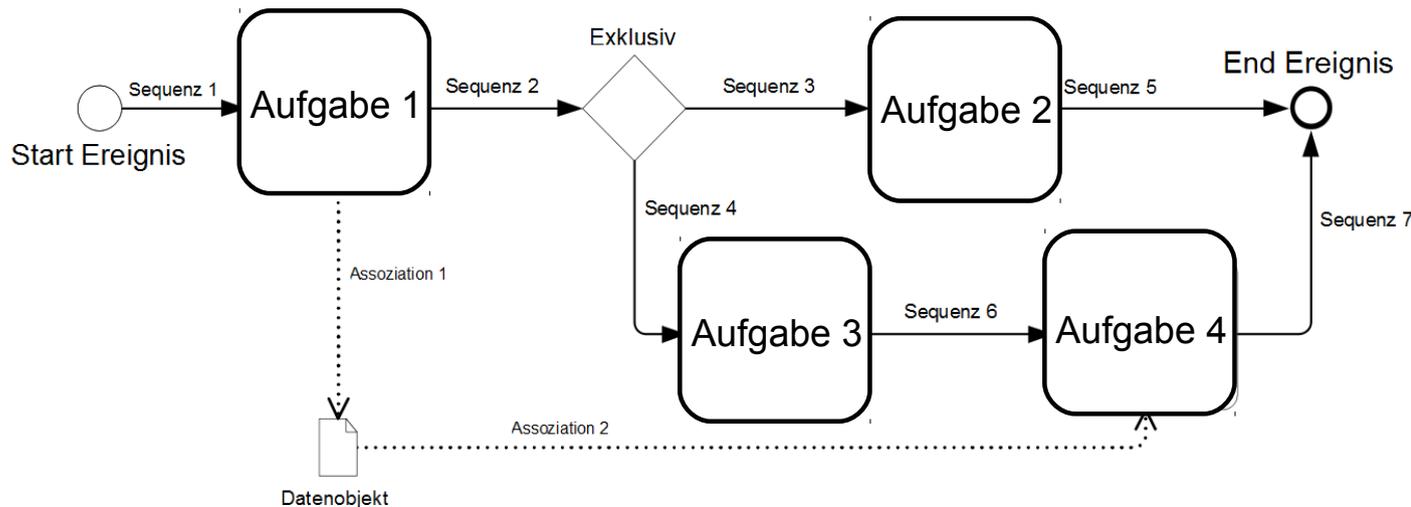
Datenobjekte: Datenfluss (Data Flow)



Sequenz- und Datenfluss
sind unabhängig



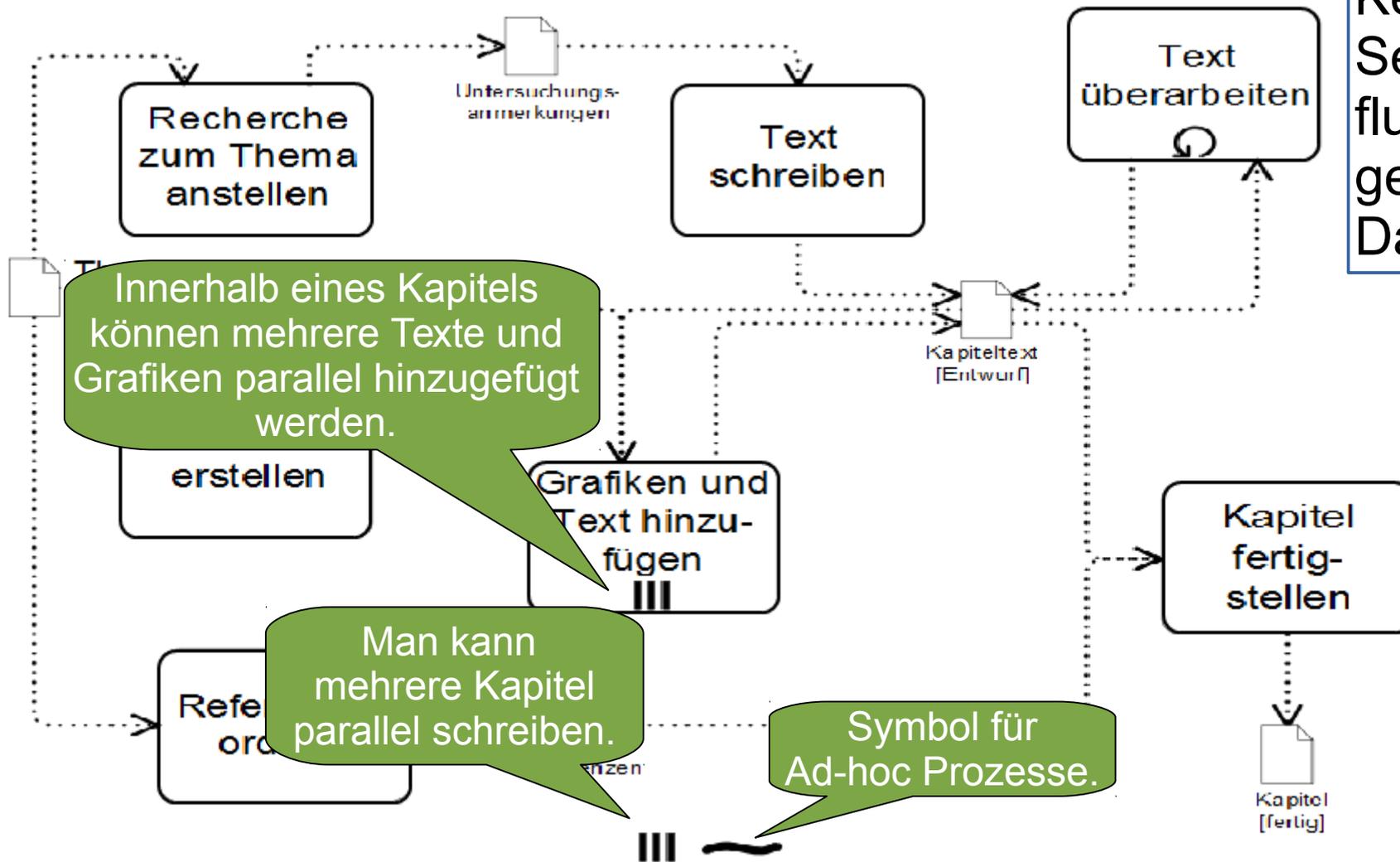
Sie können miteinander
verbunden werden



Anwendungsfall für
das Entkoppeln

Ad-hoc Prozesse: Nur Datenfluss spezifiziert

Ein Buchkapitel schreiben



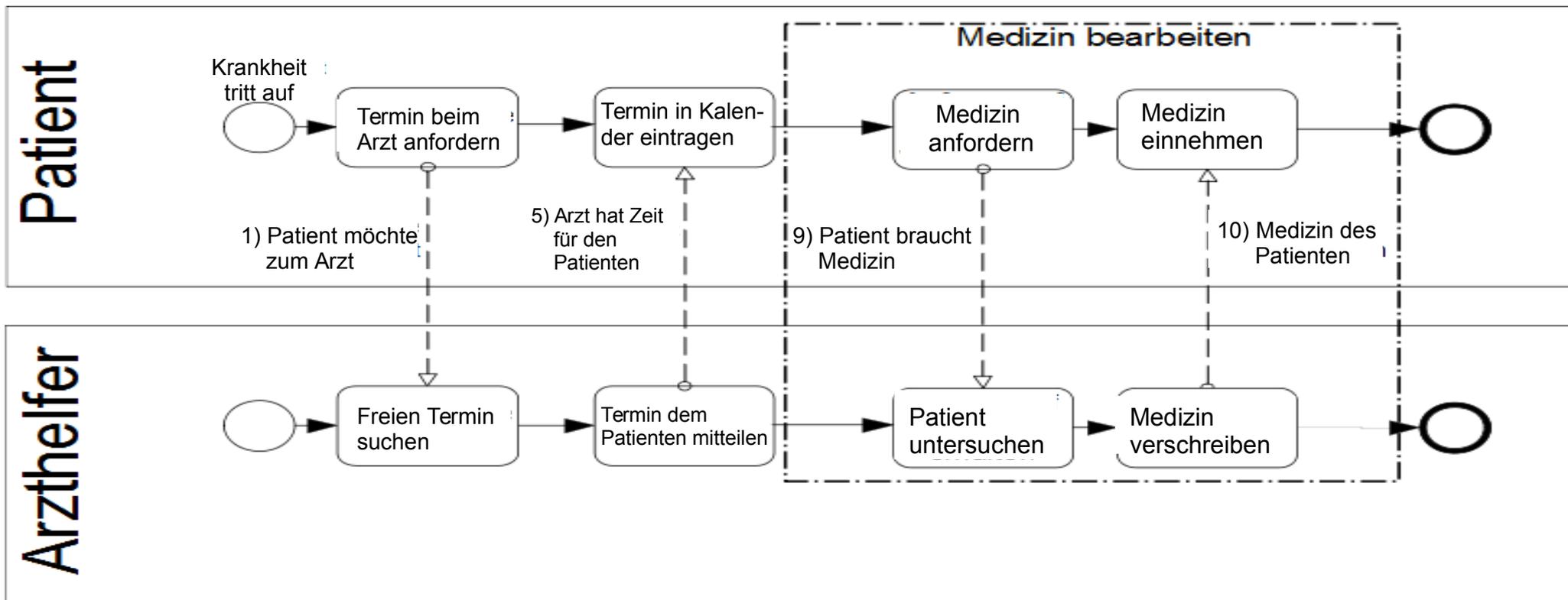
Kein Sequenzfluss vorgegeben, nur Datenfluss.

Innerhalb eines Kapitels können mehrere Texte und Grafiken parallel hinzugefügt werden.

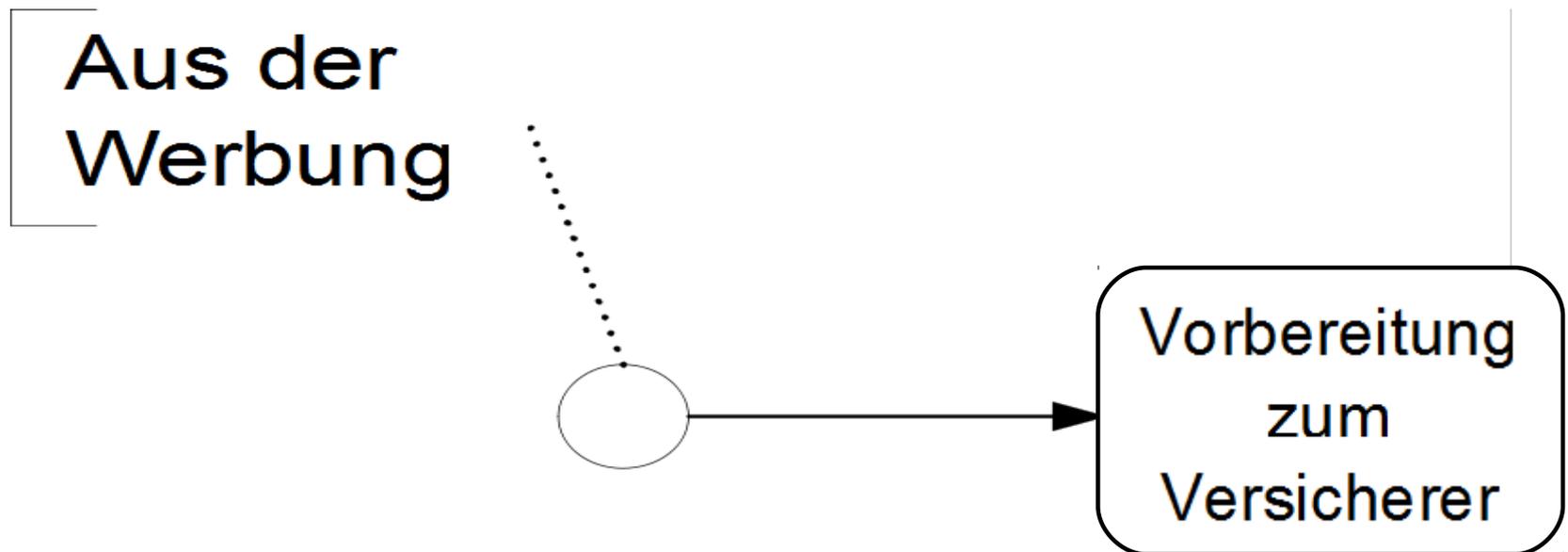
Man kann mehrere Kapitel parallel schreiben.

Symbol für Ad-hoc Prozesse.

- **Gruppen:** heben Sektionen eines Diagramms hervor, ohne zusätzliche Bedingungen wie bei Unterprozess hinzuzufügen.
- Kann Gruppen dazu benutzen, Elemente zu **kategorisieren**.
- Gruppen haben **keine Einschränkungen** wie Pools oder Bahnen.



- **Anmerkungen** können:
 - **weitere Informationen** über Prozess hinzufügen.
 - über **Assoziation** einem Objekt zugeordnet werden.

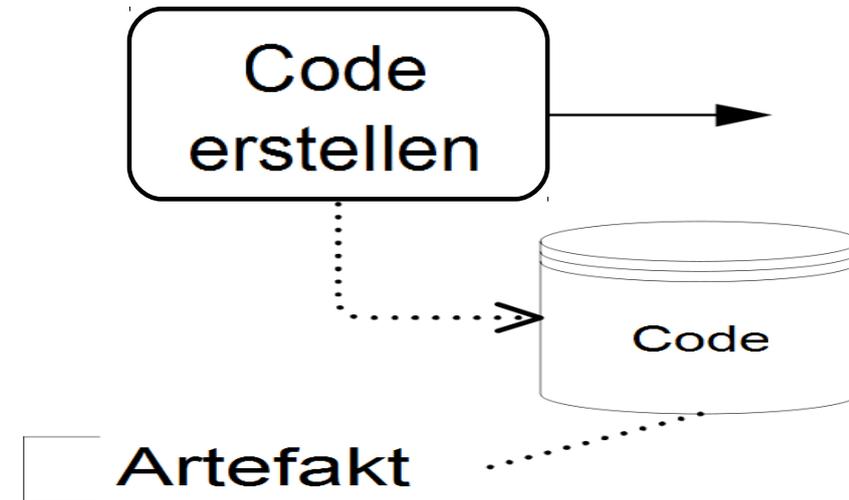


Kann BPMN um **neue Artefakte** erweitern. Beispiel: Artefakt für Code:

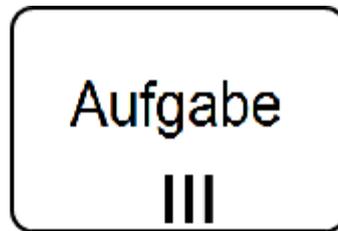
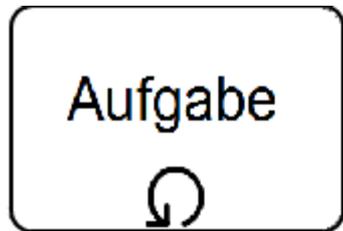
Spezielle Branchen oder Märkte können eigene **Artefaktmengen** haben.

Deren **Symbole** dürfen nicht in **Konflikt** mit vorhandenen Symbolen stehen.

- **Beispiel** für Konflikt: Artefakt „Uhr“ (im Sinne einer physischen Uhr) eingeführt mit Symbol, das Symbol für zeitgesteuertes Ereignis entspricht.



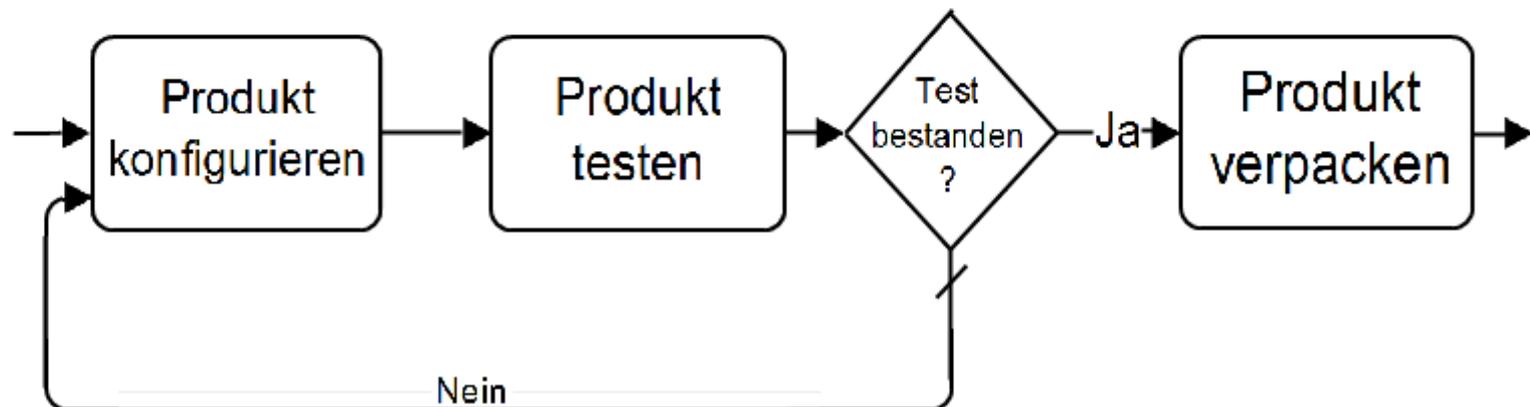
Schleifen (Looping)



Aktivitätsschleifen:

Do-While
While-Do

Multiple Instanzen



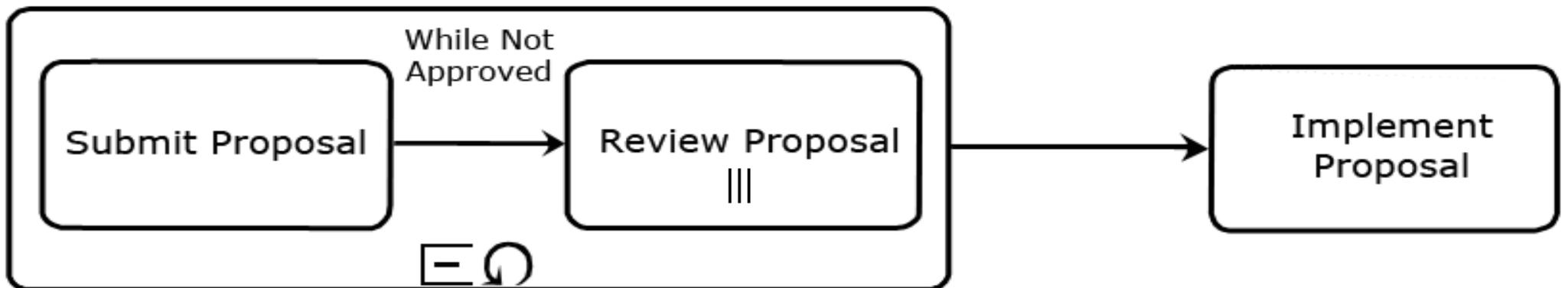
Sequenzfluss Looping

Schleifen: Beispiel für Do-While; Multiple Instanzen

Do-While:

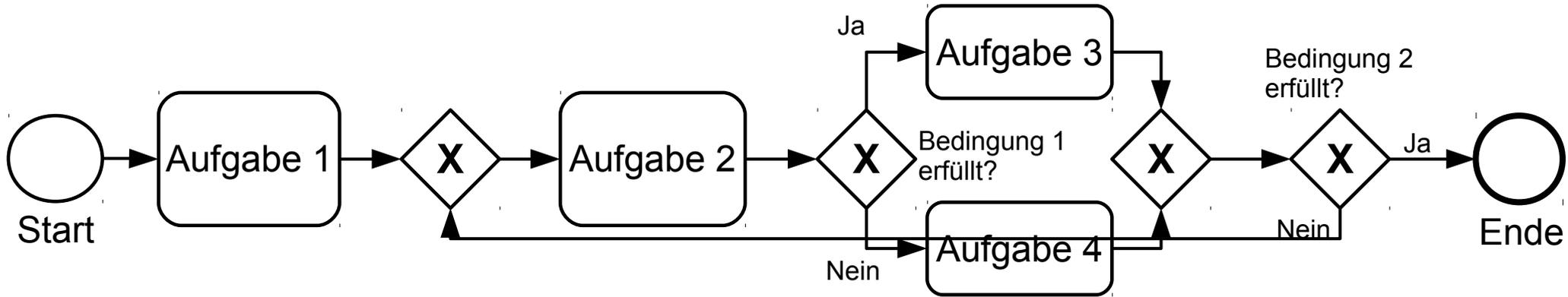


Multiple Instanzen: Mehrere Einreichungen erlaubt, die parallel bearbeitet werden:



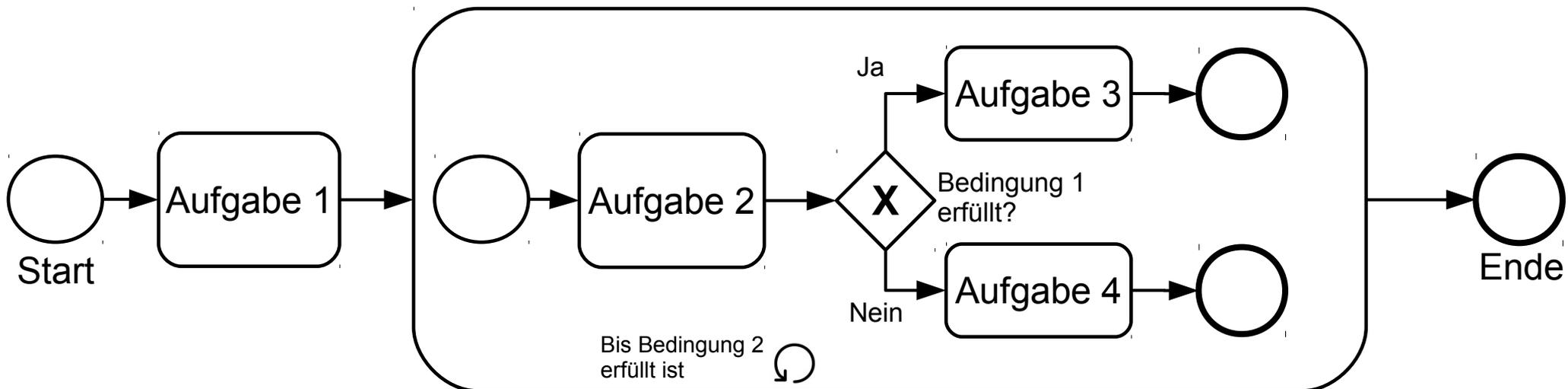
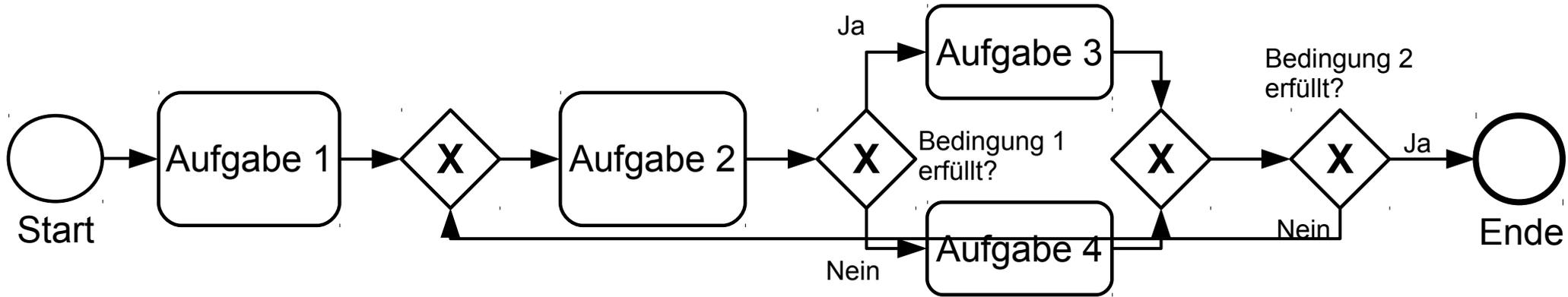
Ein Schleifen Beispiel

Wie kann man folgenden Prozess mit Hilfe einer **Schleife** darstellen?



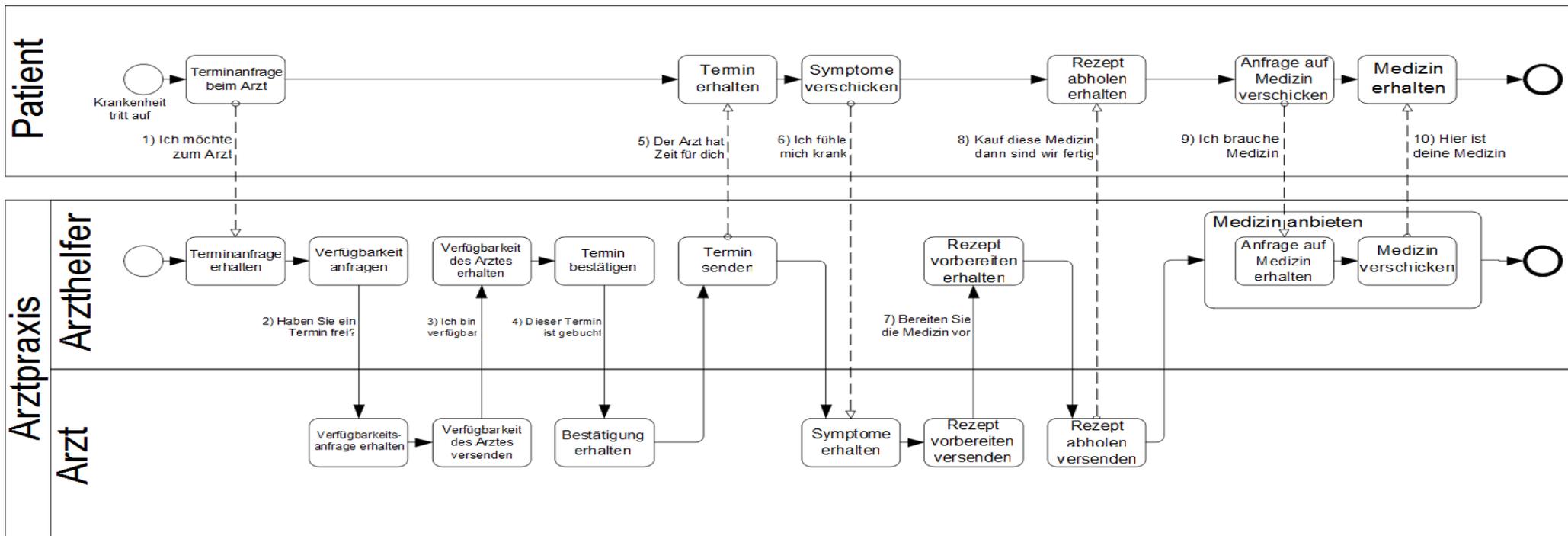
Ein Schleifen Beispiel

Wie kann man folgenden Prozess mit Hilfe einer **Schleife** darstellen?



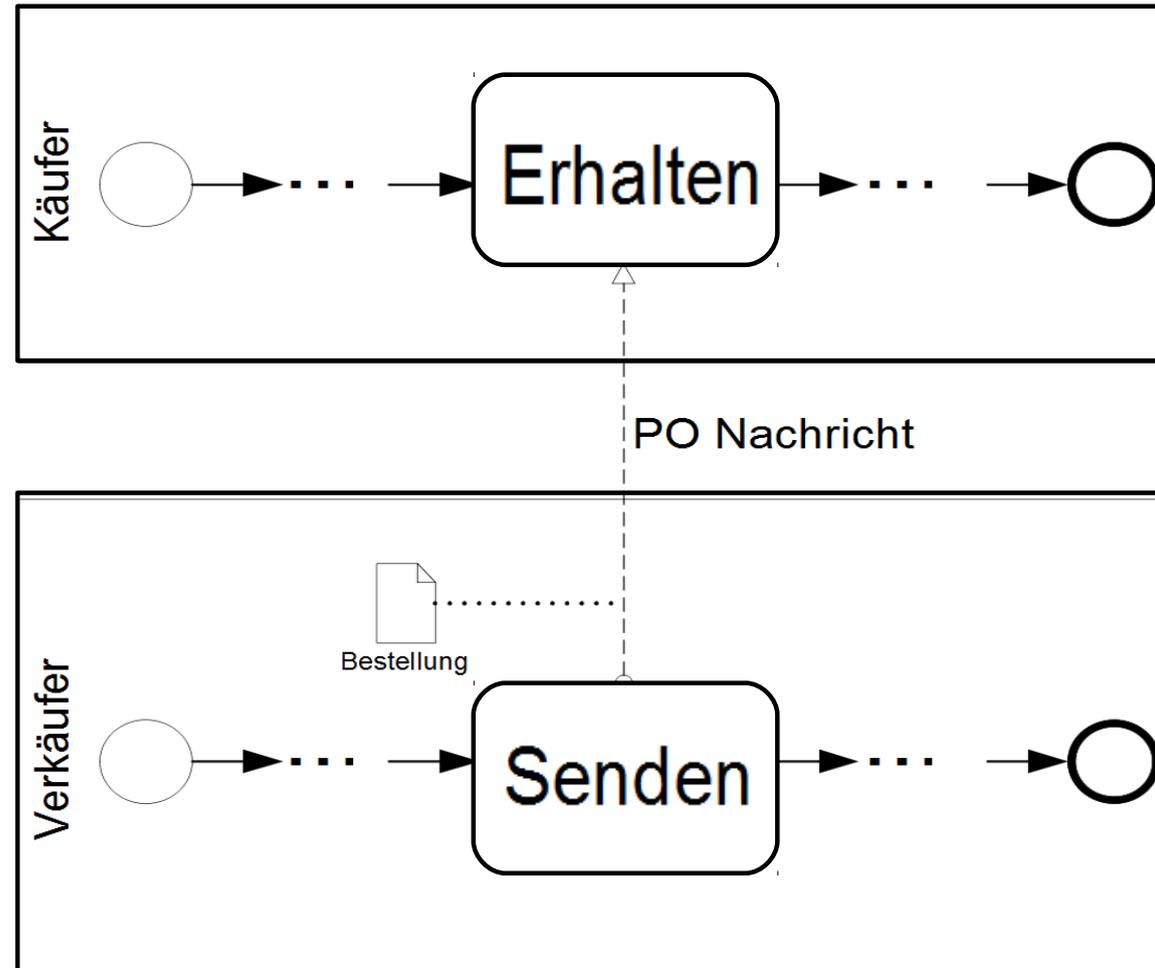
Zwei Notationen für Abgrenzung von Aktivitäten: **Pool** und **Bahn** (Swimlane).

- Pools repräsentieren Teilnehmer in einem interaktiven (B2B) Geschäftsprozessdiagramm.
- Bahnen repräsentieren Unterbereiche für Objekte innerhalb Pool.



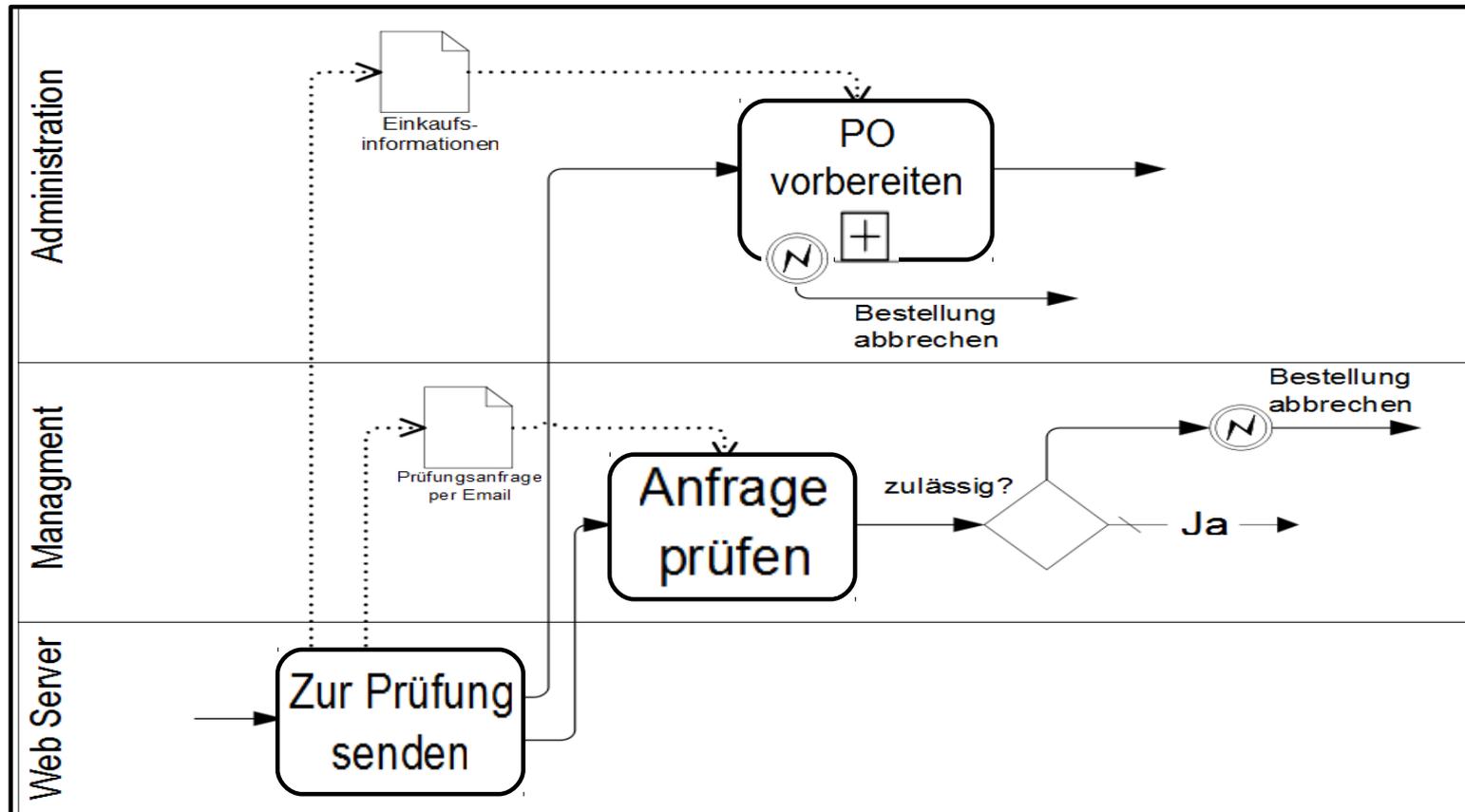
Pools repräsentieren Teilnehmer in einem interaktiven (B2B) Geschäftsprozessdiagramm.

- Teilnehmer ist **Rolle** in Geschäftsbetrieb (z.B. „Käufer“ oder eine Firma).
- Pool: **schwarzes Rechteck**, beinhaltet Prozess.
- **Interaktionen** zwischen Pools über Nachrichtenflüsse.
- **Sequenzfluss** kann Pool-Grenzen nicht überschreiten (d.h. Prozess ist in Pool eingeschlossen).

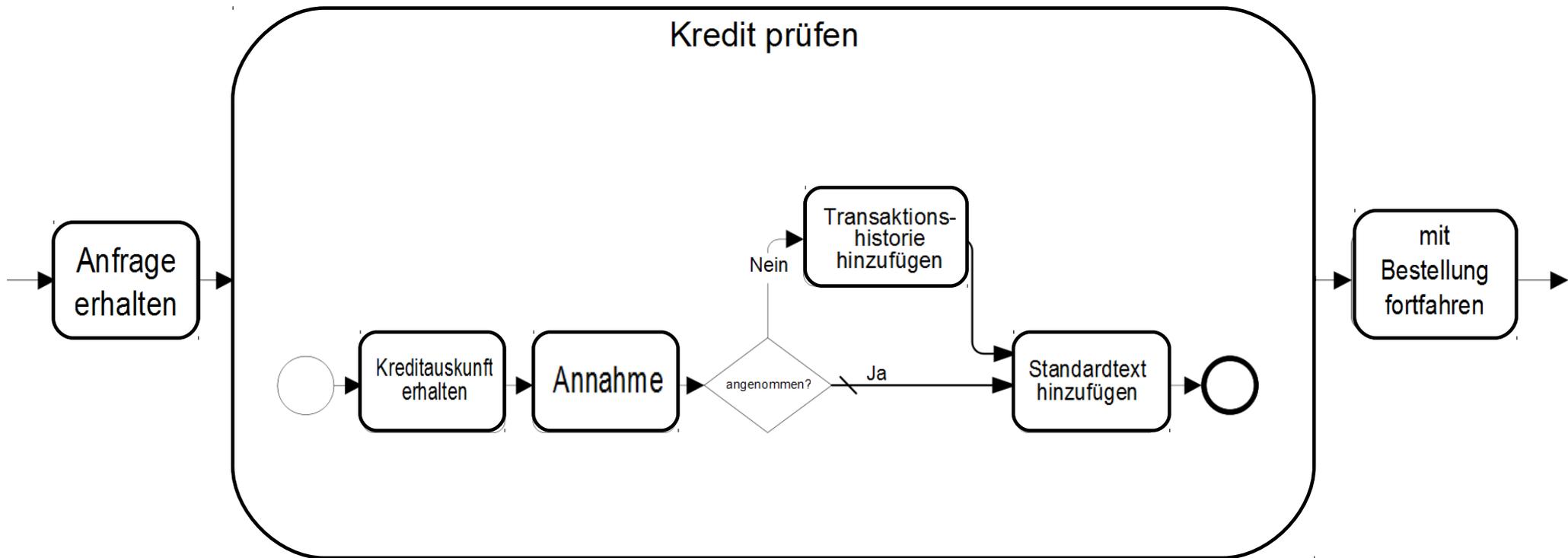


Bahnen repräsentieren Unterbereiche innerhalb eines Pools.

- **Rollen** aus Organisation (z.B. „Manager“, „Gesellschafter“), können jeden gewünschten Charakter übernehmen.
- **Sequenzflüsse** dürfen Bahn-Grenzen überschreiten.



- Kann Prozesse hierarchisch entwickeln. Verschiedene **Ebenen** durch Unterprozesse verwirklicht.
- **Sequenzflüsse** dürfen Unterprozessrand nicht überschreiten.
- **Nachrichtenflüsse** oder **Assoziationen** dürfen Unterprozessrand passieren.



Orchestrierung: Arbeitsfluss, interne Prozesse.

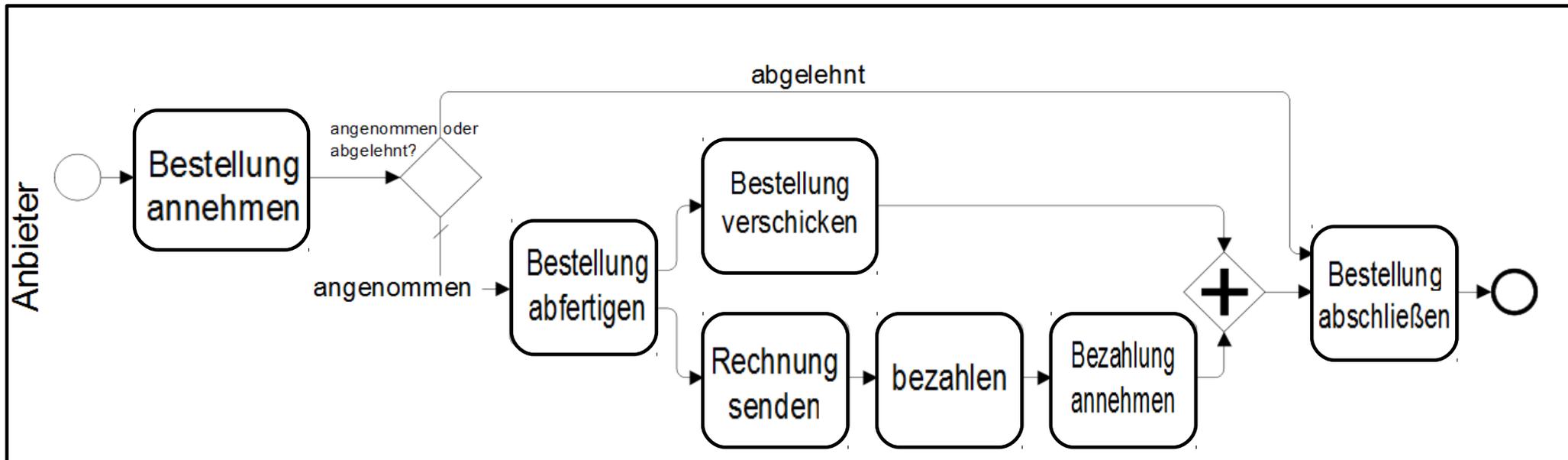
- In BPMN-Pool enthalten.

Choreografie: Kollaboration, globale Prozesse, B2B-Prozesse.

- Definiert durch Interaktion zwischen BPMN-Pools.

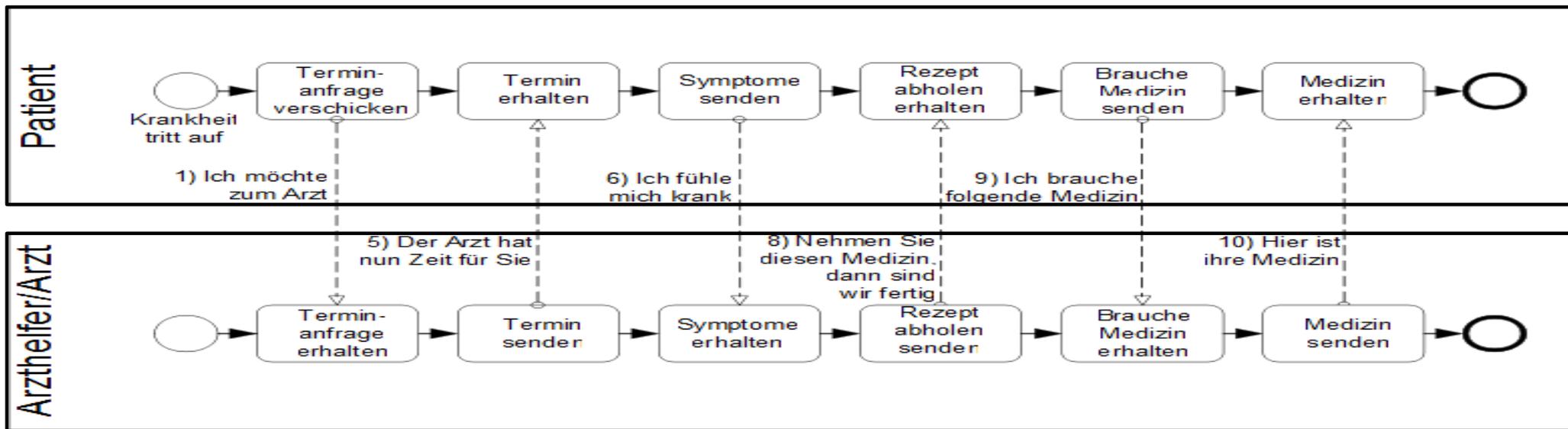
Orchestrierung beschreibt Organisations-interne Prozesse.

- Innerhalb eines Pools angesiedelt.



Choreographie-Prozess: Interaktion zwischen Geschäftsteilen
(verschiedene Pools).

Dargestellt als **Nachrichtenfluss zwischen Pools:**



... oder als **Sequenz von Aktivitäten**, die Interaktionen realisieren:





Wie kann man **Prozessebenen** einer **Prozeshierarchie** verbinden ?

Wie kann man **Prozessebenen** einer **Prozesshierarchie** verbinden ?

- Sequenzflüsse dürfen Rand eines Subprozesses oder einer Transaktion nicht überschreiten.
- Verknüpfung geschieht über **Assoziationen** und **Nachrichtenflüsse**.
- Alternativ: Verbindung auf höherer Abstraktionsebene modellierbar durch **Sequenz von Aktivitäten**, die Interaktionen realisieren.

Frage: Auswirkung Terminate-Endereignis

Frage: Auf welche Prozesse wirkt sich ein Terminate-Endereignis aus?
Werden auch darüberliegende Prozesse abgebrochen?

Antwort: Nein, höhere Prozessebenen oder parallele Instanzen desselben (Unter-)Prozesses sind nicht betroffen.

Quelle: BPMN-Spezifikation S. 473:

For a “terminate” End Event, the Sub-Process is abnormally terminated. In case of a multi-instance Sub-Process, only the affected instance is terminated—no other ongoing Sub-Process instances or higher-level Sub-Process or Process instances are affected.

Frage: Auf welche Prozesse wirkt sich ein Link-Ereignis aus? Werden auch darüberliegende Prozesse abgebrochen?

Antwort: Nein, höhere Prozessebenen oder parallele Instanzen desselben (Unter-)Prozesses sind nicht betroffen.

Quelle: BPMN-Spezifikation S. 267:

A Link Event is a mechanism for connecting two sections of a Process.

Link Events can be used to create looping situations or to avoid long Sequence Flow lines.

The use of Link Events is limited to a single Process level (i.e., they cannot link a parent Process with a Sub-Process).

Frage: Wie verhält sich ein XOR-Gateway beim Zusammenfügen von Sequenzflüssen, wenn es **gleichzeitig** Tokens von zwei Seiten bekommt.

Antwort: Dann werden auch zwei Tokens weitergeleitet.

Quelle: BPMN-Spezifikation S. 465:

Each token arriving at any incoming Sequence Flows activates the gateway and is routed to exactly one of the outgoing Sequence Flows.

=> XOR-Eigenschaft wird nur beim Aufspalten durchgesetzt, nicht beim Zusammenfügen !

Vorteile EPK:

- Einfache Notation
- Weit verbreitet
 - z.B. SAP/R3 Prozessmodelle [vgl. <http://www.drkarlpopp.de/KopplungGeschaeftsprozessAnwendungssystemSoftwareAris.html>]
 - Werkzeugunterstützung (ARIS, Adonis, ...)

Vorteile BPMN:

- Moderner Standard.
- Unterstützt durch Industrie-Konsortia wie OMG.
- Integration mit Workflow-Automation und SOA durch BPEL.
- Zunehmend unterstützt durch Werkzeuge.

Einführung in die BPMN 2.0: Zusammenfassung

In diesem Teil haben wir betrachtet:

- **BPMN am Beispiel**
- **BPMN-Hintergrundwissen**
- **BPMN im Detail**
 - Grundlegende Konzepte
 - Weiterführende Konzepte

Anhang (weitere Informationen und Beispiele zum Nacharbeiten)

Methodische Grundlagen
des Software-Engineering
SS 2014

