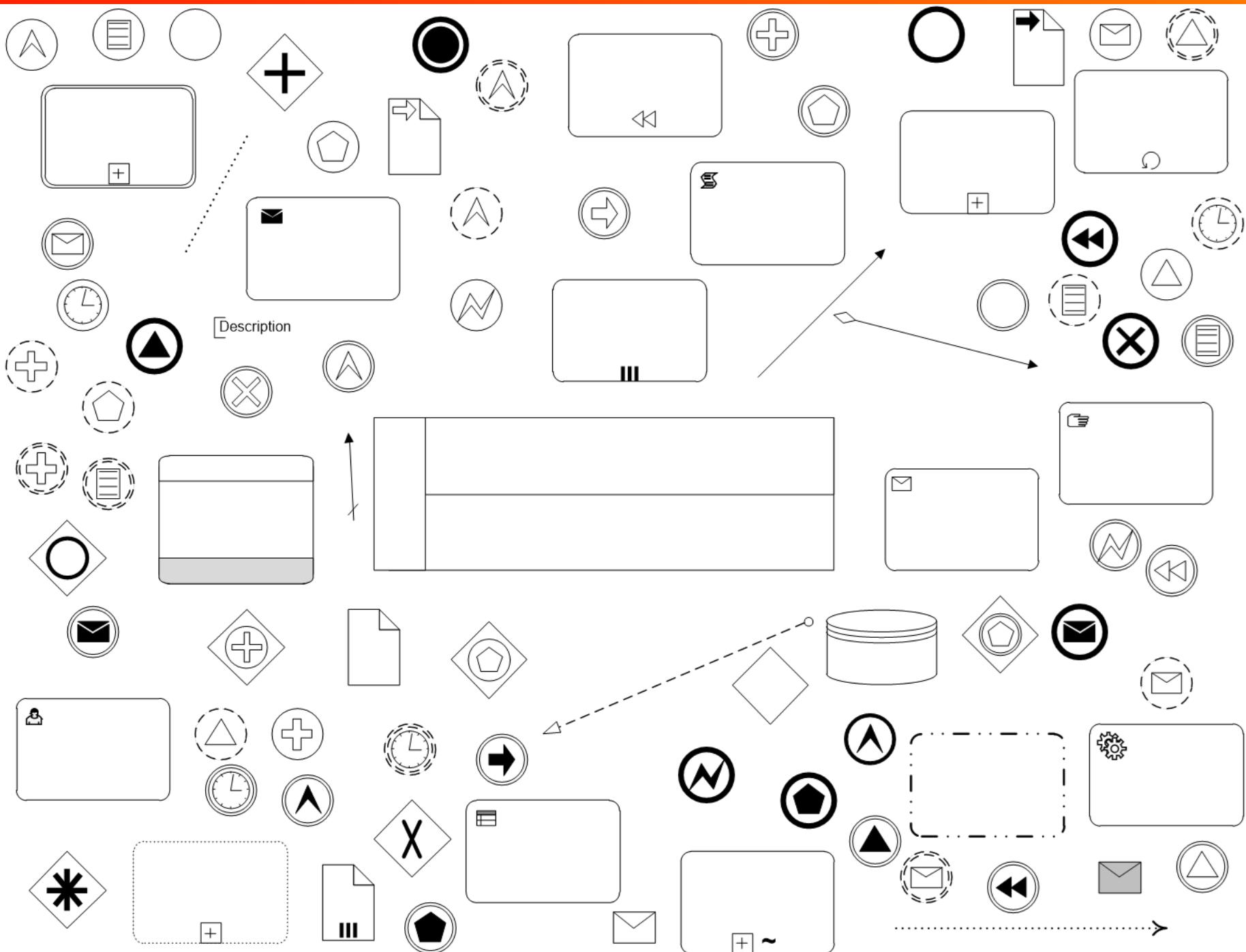


CAMUNDA

BPMN 2.0 Training

Version 2020-02



Agenda

Getting started with BPMN

Excursion: BPMN in the context of BPM

BPMN in detail

Orchestration and collaboration

Advanced BPMN constructs

... and a lot of exercises

Introduction of BPMN

Example Process Documentation

Example Business IT-Alignment

Process execution

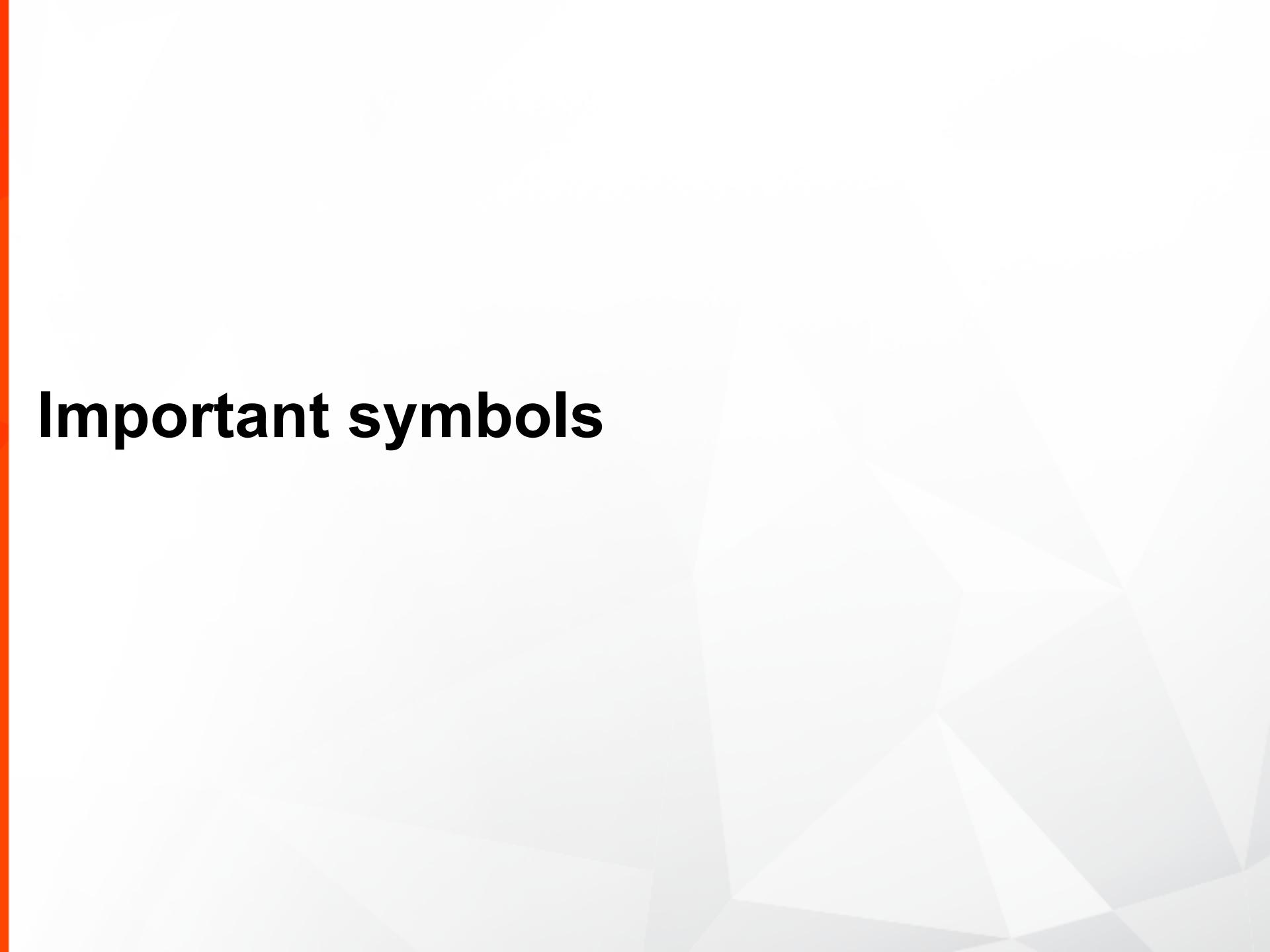
Modeling conventions

Business rules with DMN

BPMN conversation and choreography



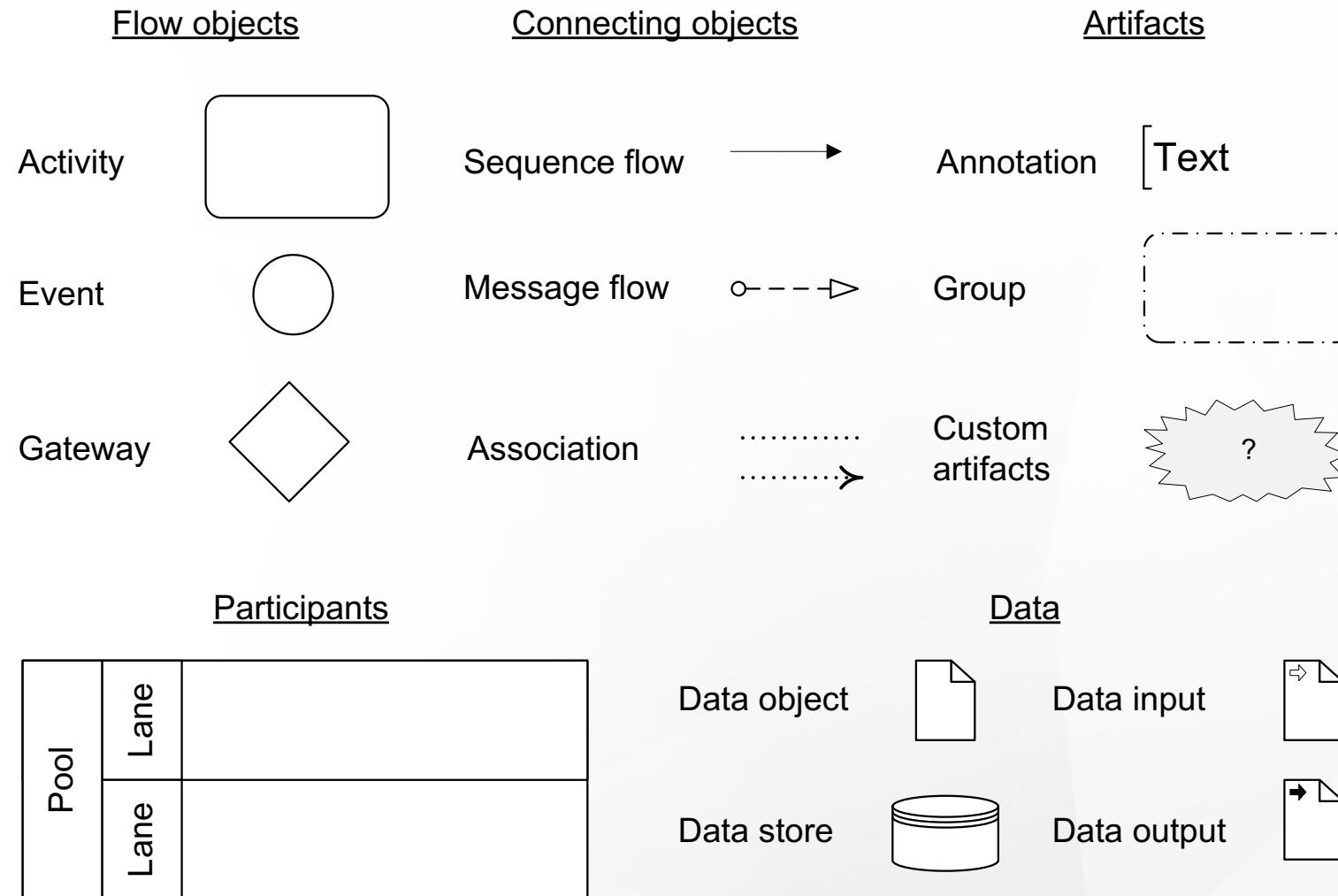
Important symbols



A first BPMN model...



Basic elements



What BPMN should be used for

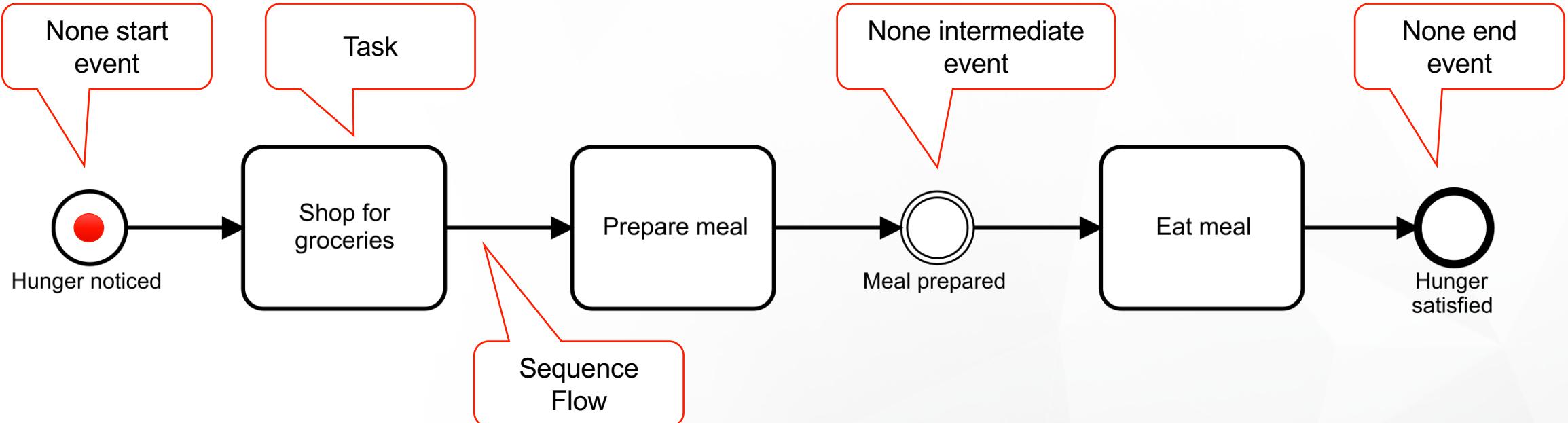
BPMN at its best:

- Processes

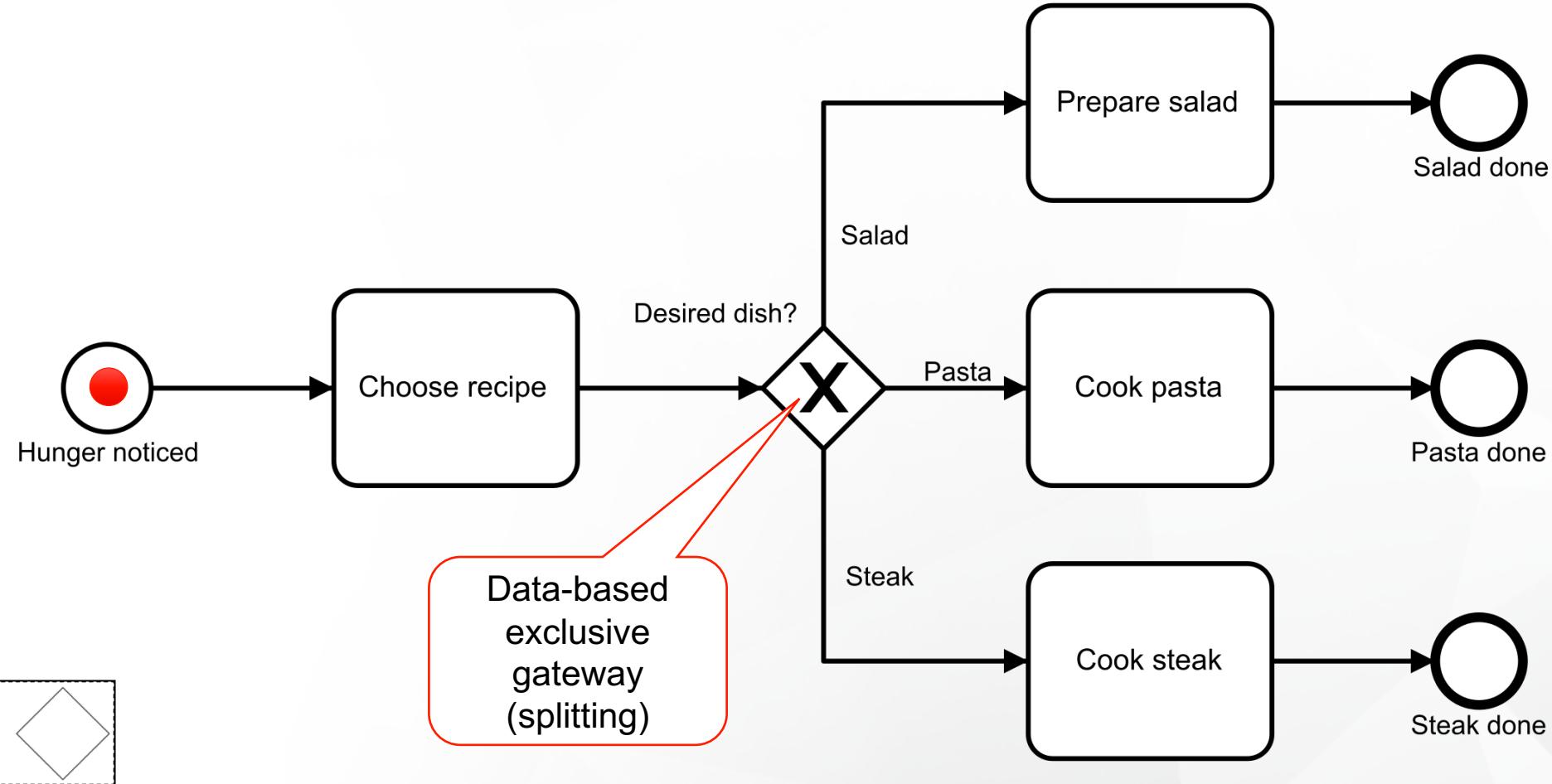
BPMN is not suitable for:

- Process landscapes
- Organizational structures
- Data
- Strategies
- Business rules
- IT landscapes

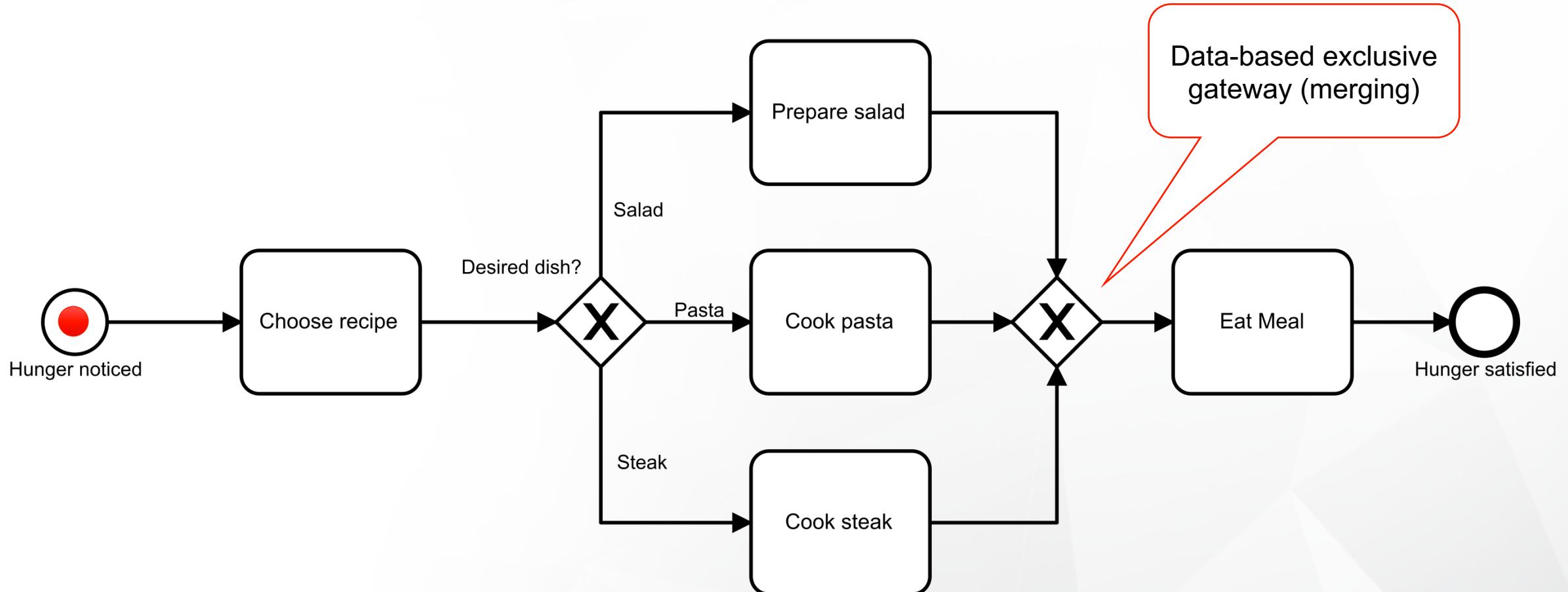
Tasks and events



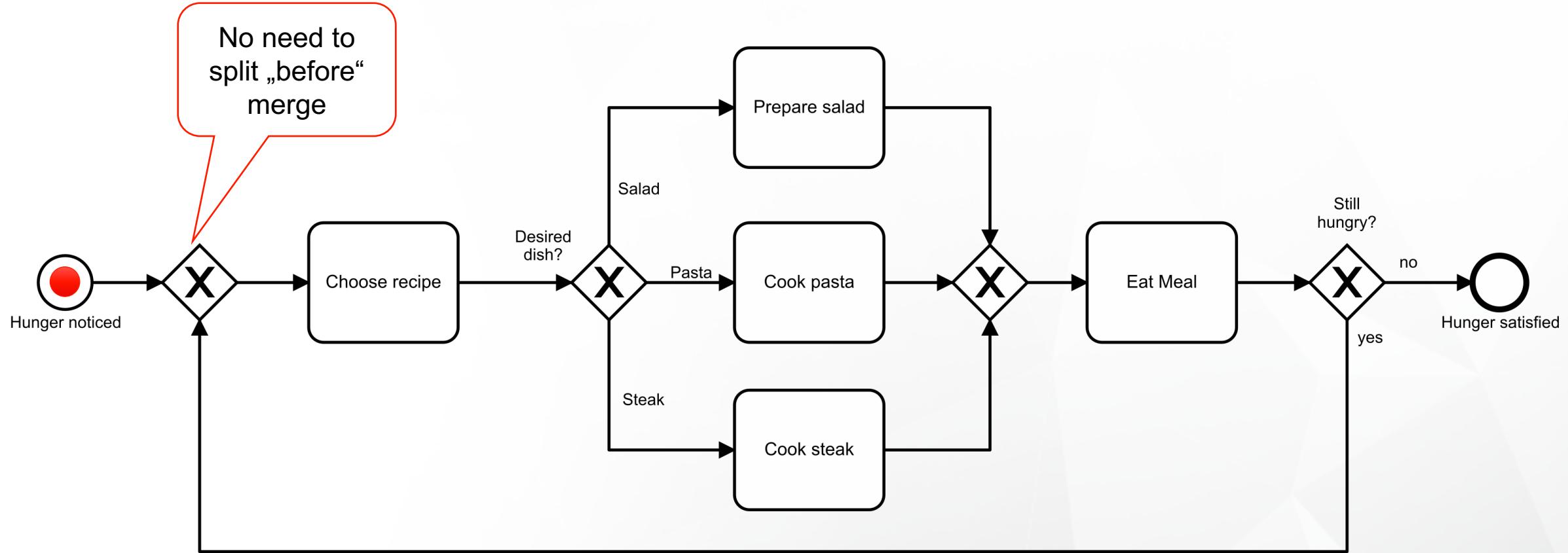
The XOR-gateway



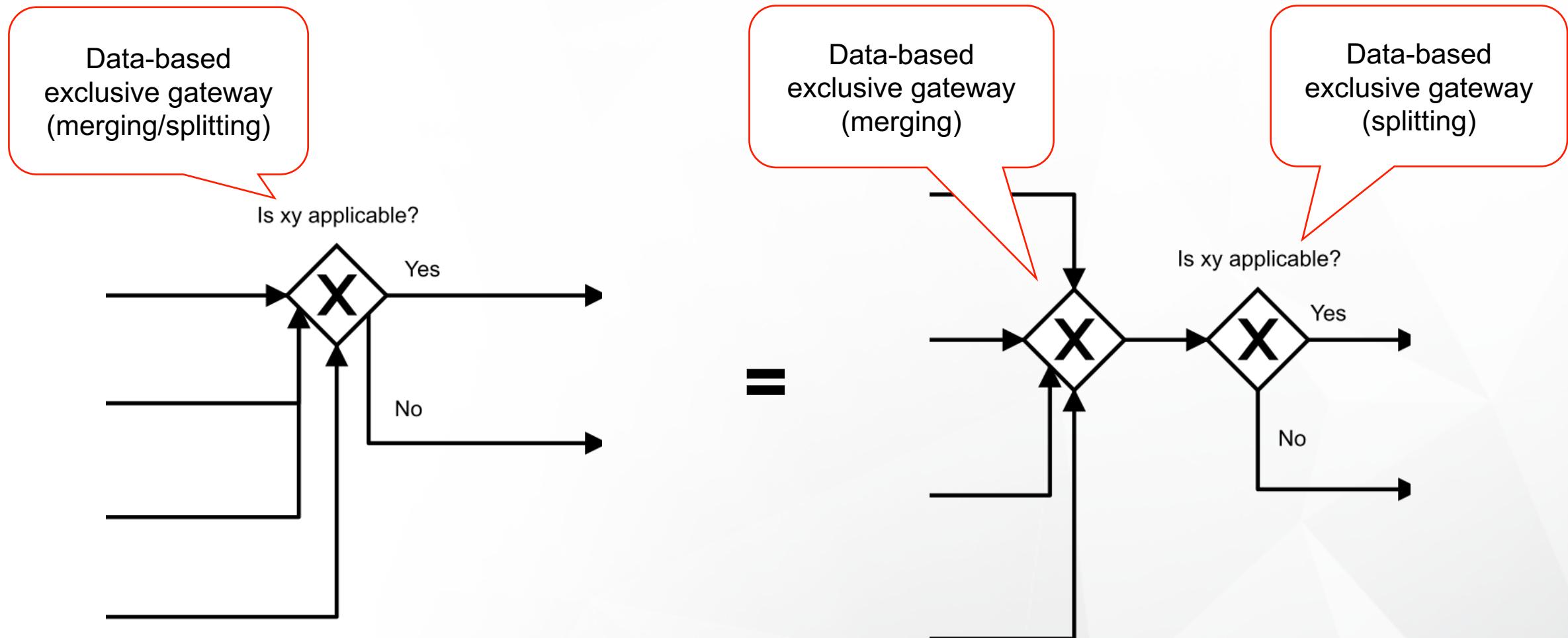
XOR-gateways can also merge



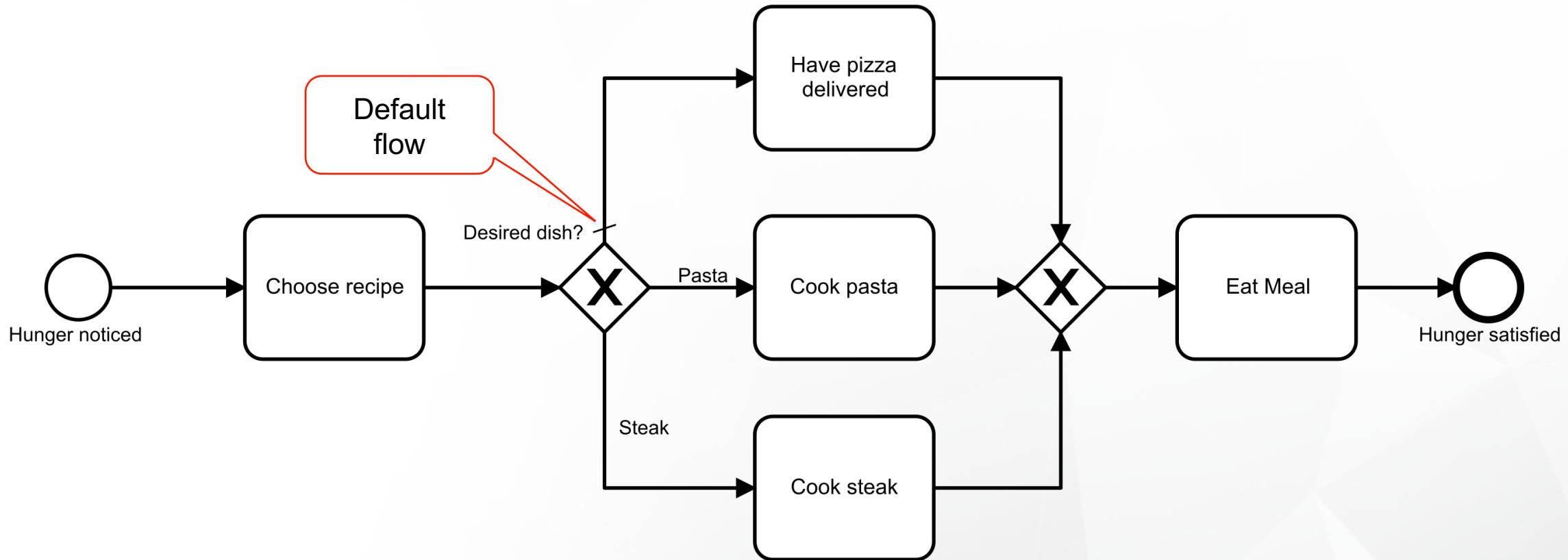
Don't be afraid of loops – they are welcome!



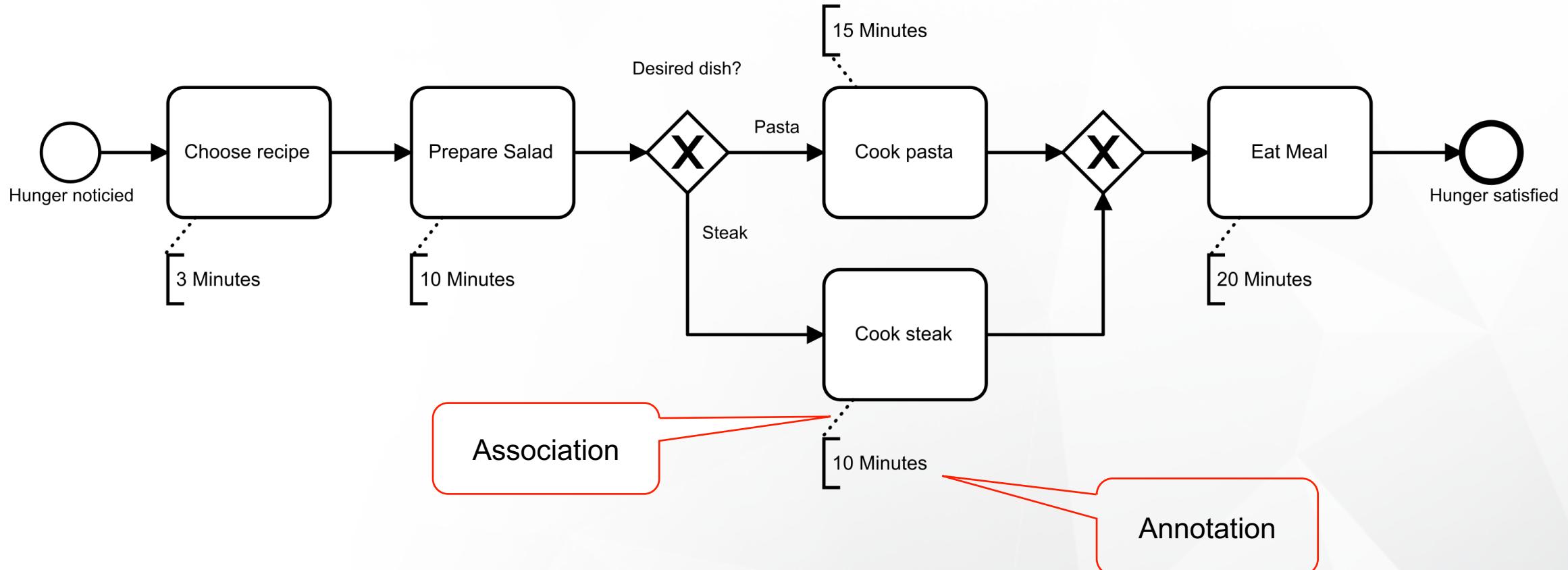
Combined merge/split



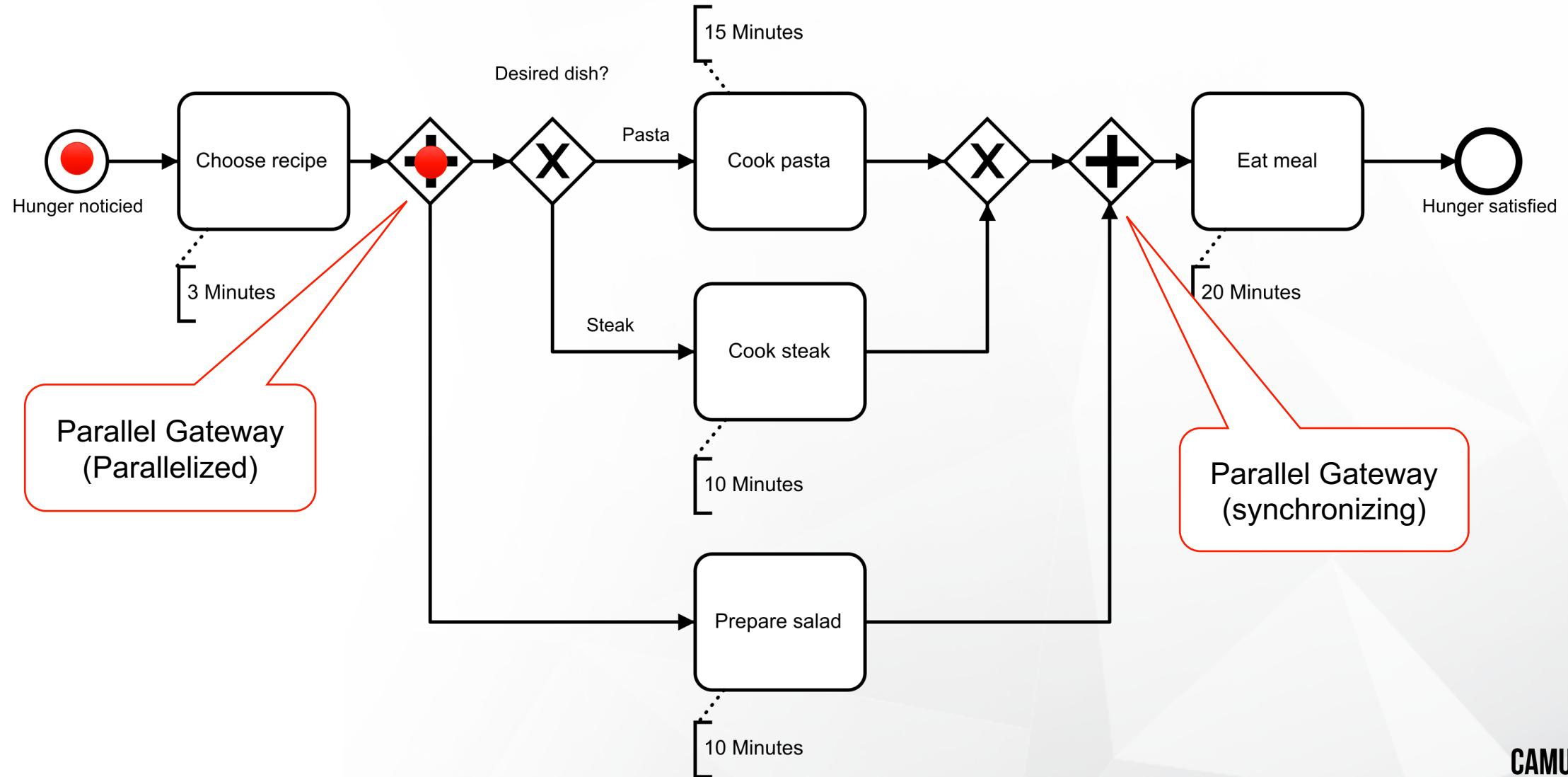
What happens if no condition applies?



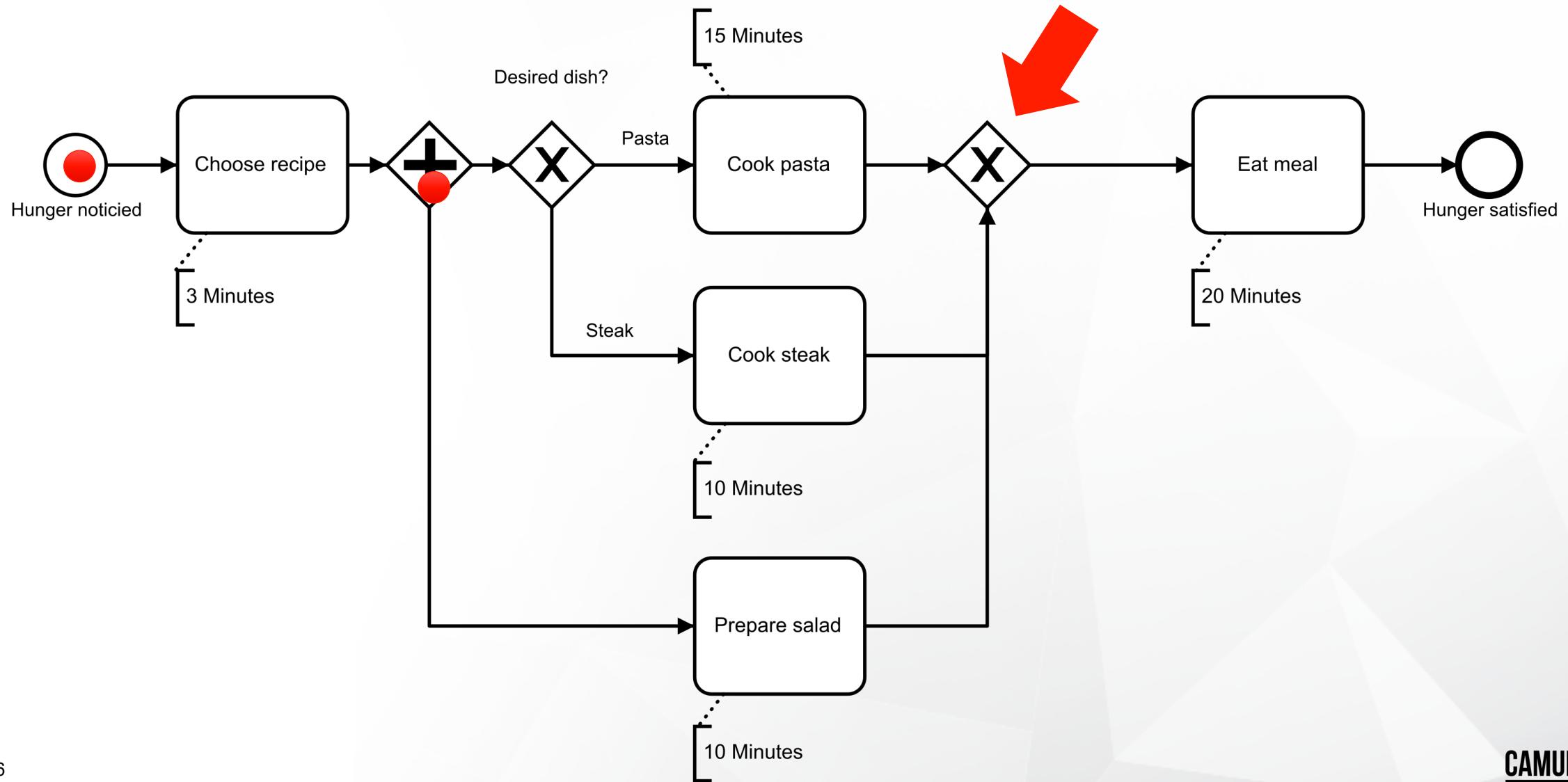
Preparing salad and main course



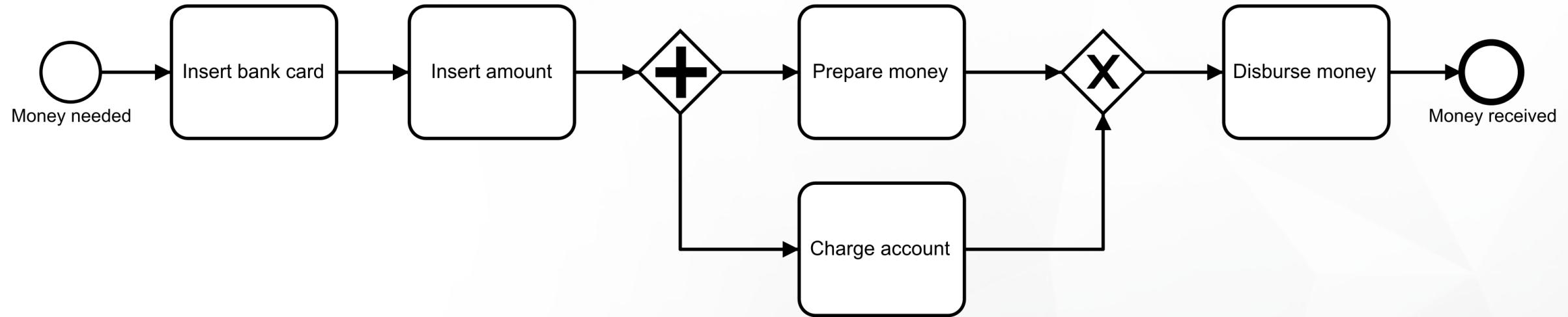
Preparing salad and main course at the same time



What happens in this process?

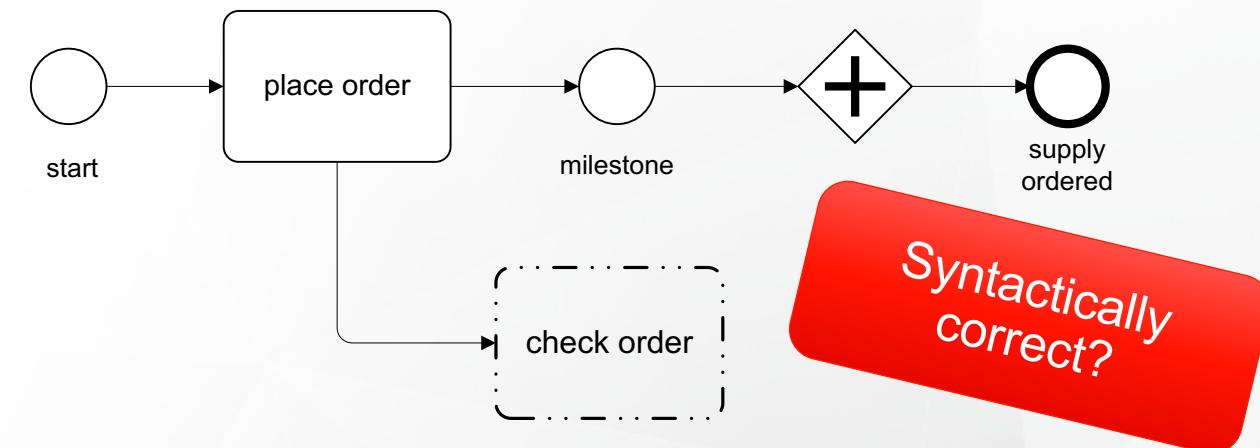


Example: ATM



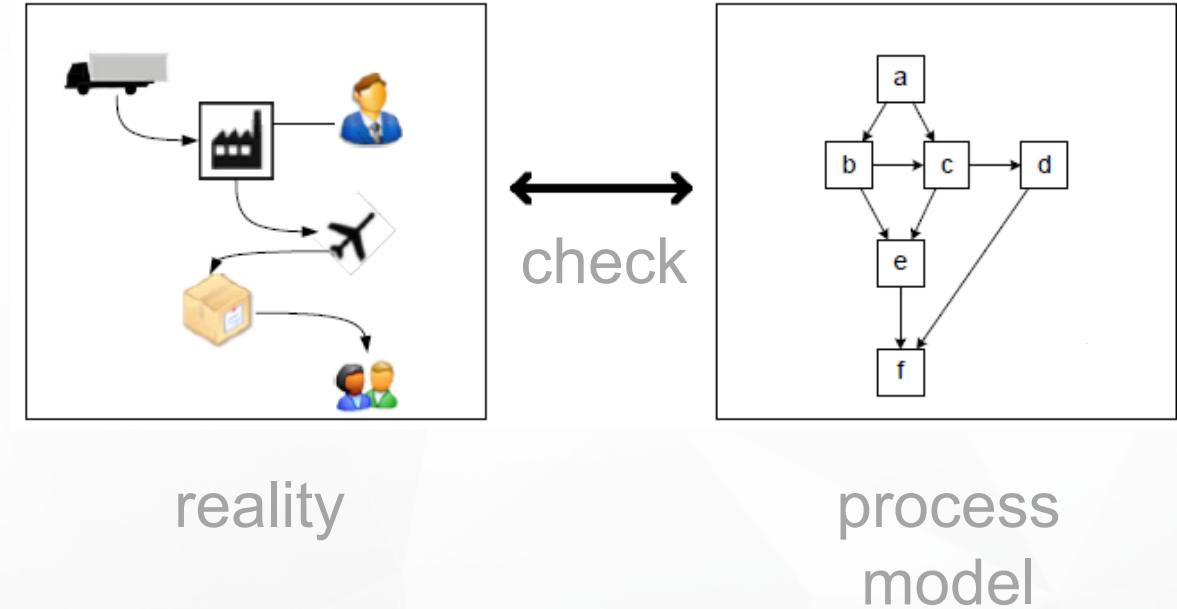
Syntactical correctness of process models

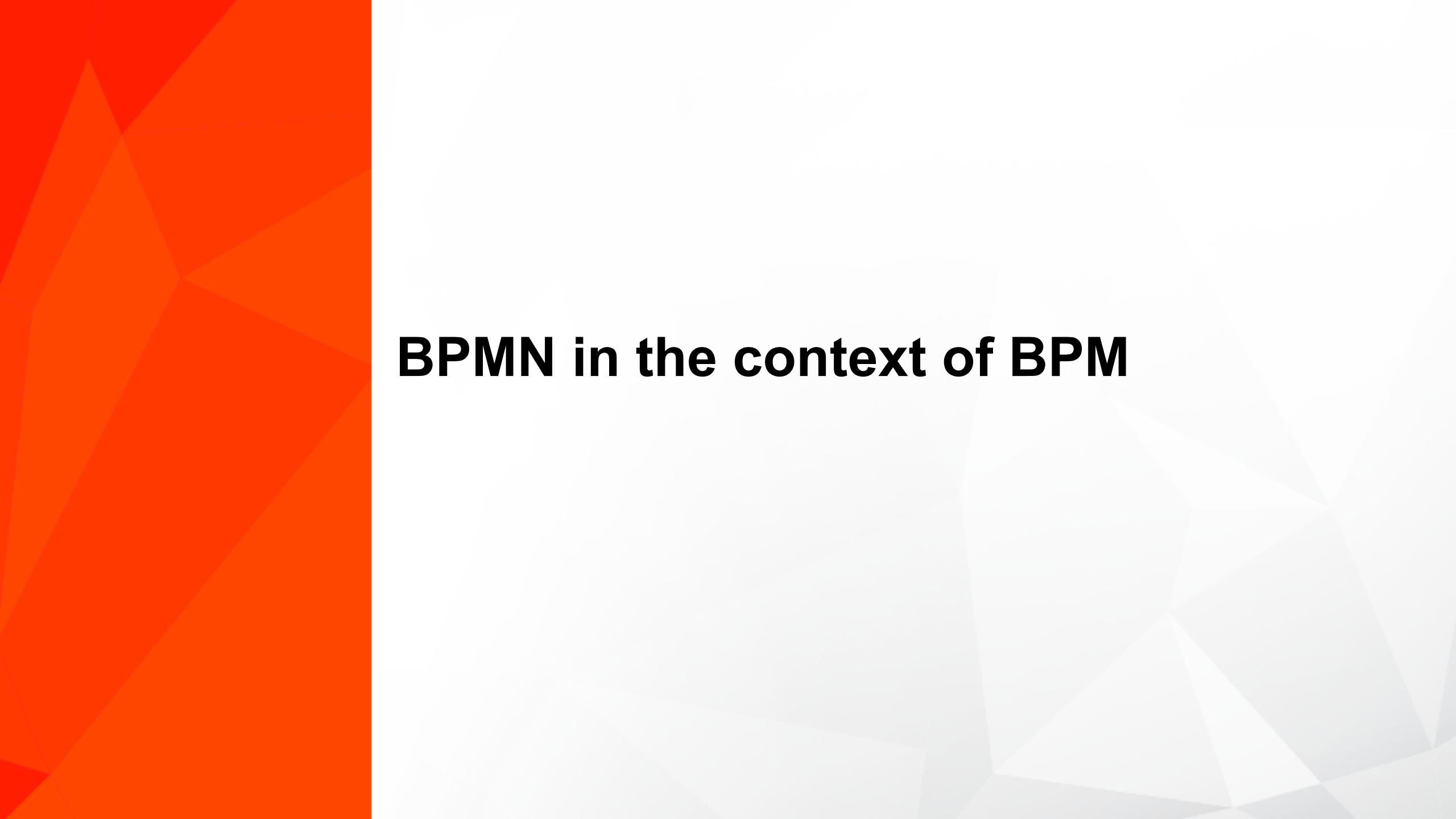
- Core Question: Is BPMN applied correctly?
- Typical misuse:
 - “new“ symbols are used incorrectly
 - Existing symbols are used incorrectly
 - Symbols are combined incorrectly



Semantic correctness of process models

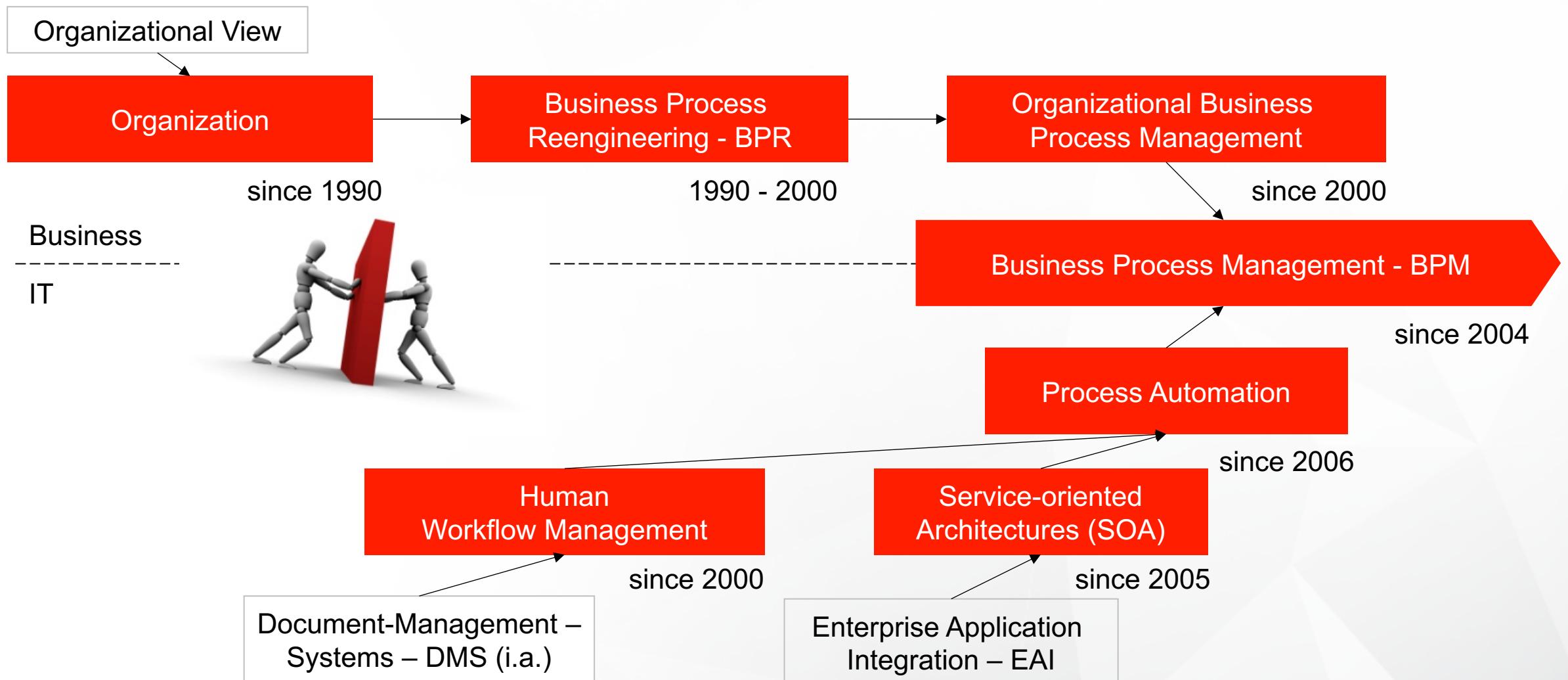
- Check for consistency of models related to a reality
- Knowledge about the reality must be available (humans!)
- Knowledge discovery often done in discussions/workshops with process participants
- Tool support only partially possible



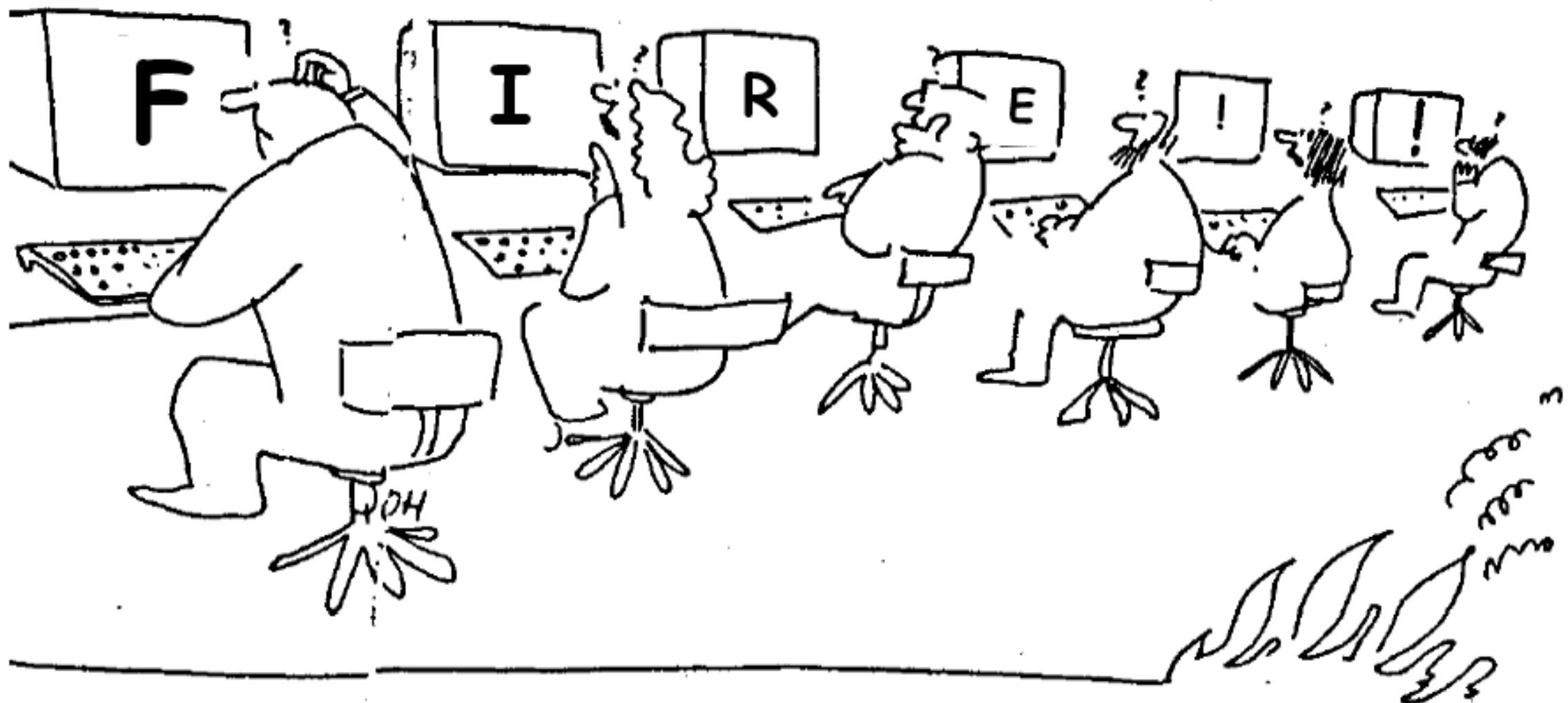


BPMN in the context of BPM

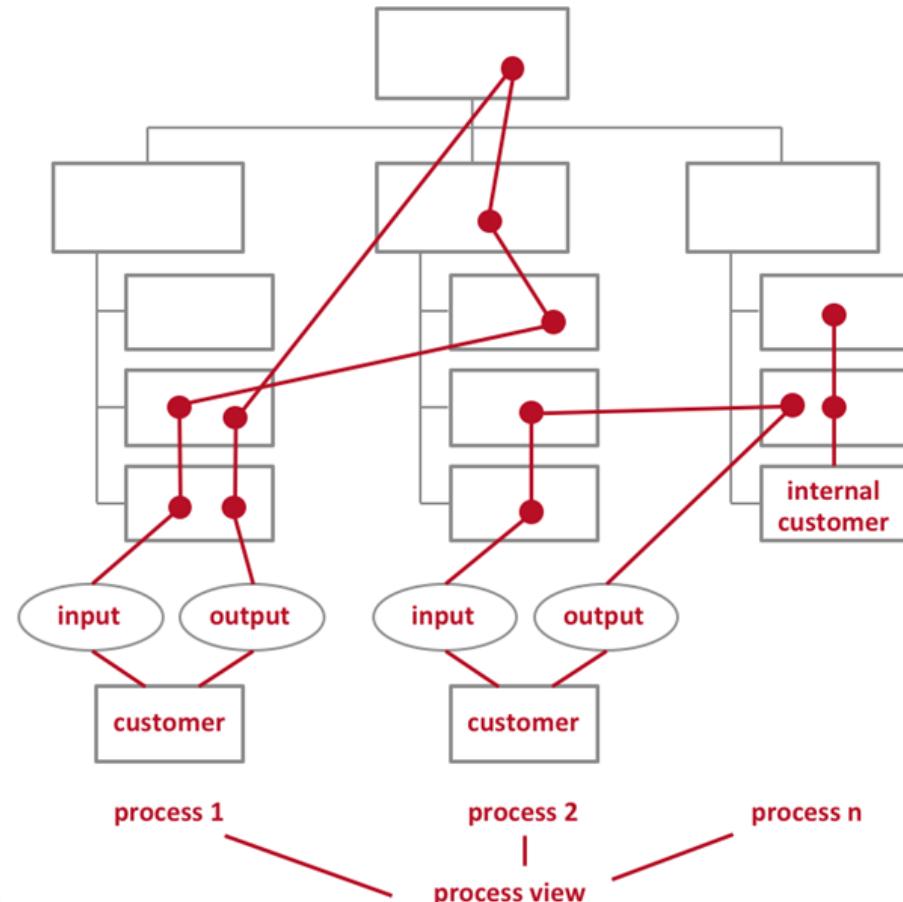
The two faces of BPM



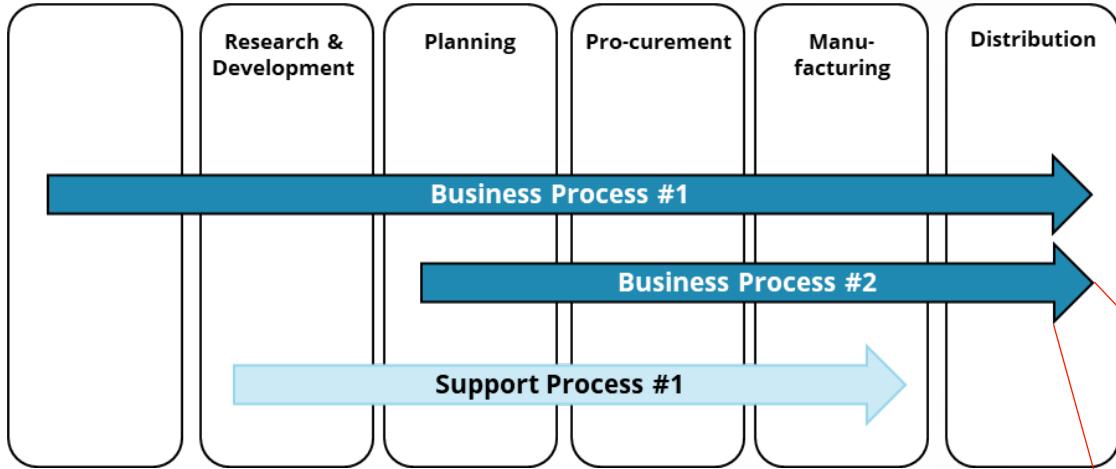
Do you know your processes?



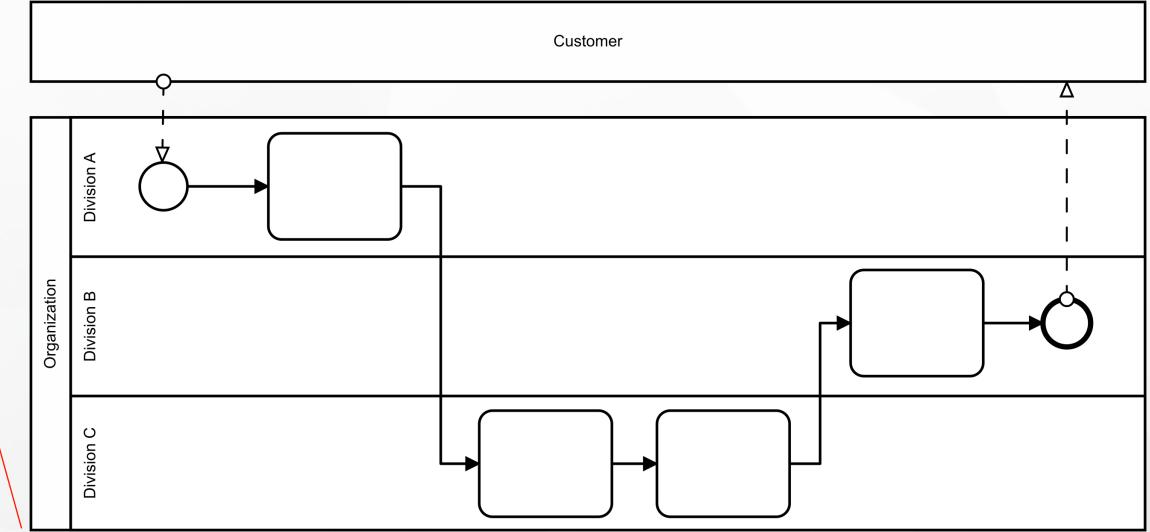
The problem of functional organizations



Process-centered organizations



Clear focus on **Business Processes**



Business Process Maturity Levels



#	Name	Description
1	Initial	Wherein business processes are performed in inconsistent sometimes adhoc ways with results that are difficult to predict.
2	Managed	Wherein management stabilizes the work within local work units to ensure that it can be performed in a repeatable way that satisfies the workgroup's primary commitments. However, work units performing similar tasks may use different procedures.
3	Standardized	Wherein common, standardizes processes are synthesized from best practices identified in the work groups and tailoring guidelines are provided for supporting different business needs. Standard processes provide an economy of scale and a foundation for learning from common measures and experience.
4	Predictable	Wherein the capabilities enabled by standard processes are exploited and provided back into the work units.
5	innovative	Wherein both proactive and opportunistic improvement actions seek innovations.

Business processes are different

Maturity Level AS-IS	Maturity Level TO-BE	Opportunities	Real life
2/3	2/3	AS-IS discovery & documentation; TO-BE transformation	Support processes
1	2/3	TO-BE creation	Core processes
1	1	-	Management processes

Example: Damage adjustment ≠ Damage adjustment



For car insurance

- Parameters
- Occurs often, damages are relatively low
 - Narrow possible bandwidth
 - Regulation must be *efficient*

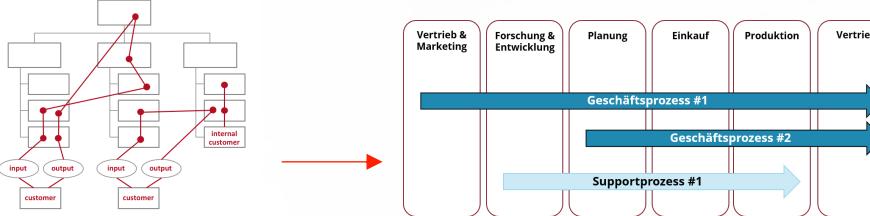
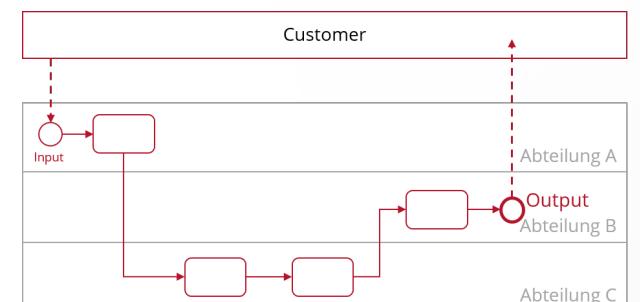
- Consequences
- Standardized work instructions
 - Dynamic only for exceptions (suspected fraud)

- Conclusion
- Automation possible and profitable

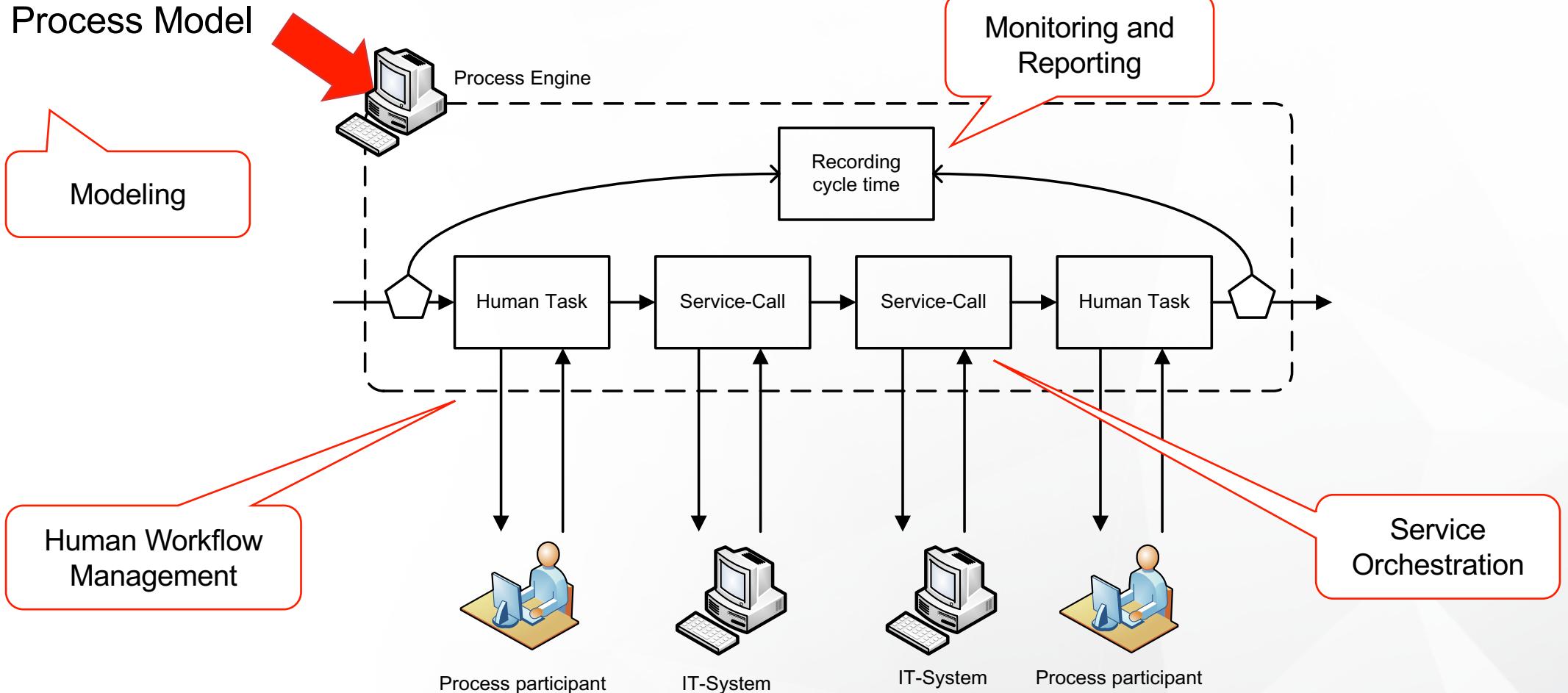
For personal insurance

- Mostly seldom, damages high
- Wide range of damages (pensions etc.)
- Regulation must be *effective*
- Less standardized flows, high dynamics
- Based on know-how of employees
- Automation can just partially support the workflow

Roles in BPM

Scope		Role
Entire organisation		<p>Project Manager Implements BPM as a methodology</p> <p>CPO Oversees the overall process architecture</p>
Business processes		<p>Process expert Is a process user, who actively involved in the process.</p> <p>Process owner Manages the process performance and KPIs</p> <p>Process modeler Creates BPMN process models</p>
External view		<p>Process consultant Works on a project and its conception and on the performance of processes</p> <p>Process auditor Audits processes</p>

The process engine



Human task management

Created + < > ↻ Add Comment +

Filter Tasks

User Task One	ProcessMigrationDemo	Demo Demo
Created 21 hours ago	50	
Review Invoice		
Invoice Receipt	Demo Demo	
Due in a day, created a day ago	50	
Invoice Am...	10.99	
Invoice Nu...	PSACE-5342	
Review Invoice		
Invoice Receipt	Demo Demo	
Due in a day, created a day ago	50	
Invoice Am...	10.99	
Invoice Nu...	PSACE-5342	

Review Invoice

Invoice Receipt

Set follow-up date in a day

Add groups Demo Demo

Form History Diagram Description

Please clarify:
This invoice needs clarification and could not be approved due to missing data.

Invoice Document invoice.pdf

Creditor Papa Steve's all you can eat

Amount 10.99

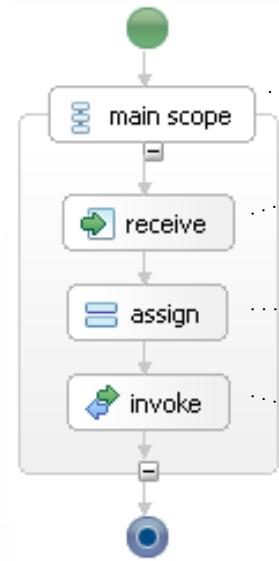
Invoice Category Travel Expenses

Invoice Number PSACE-5342

Could you clarify the invoice?

Save Complete

BPEL (Business Process Execution Language)



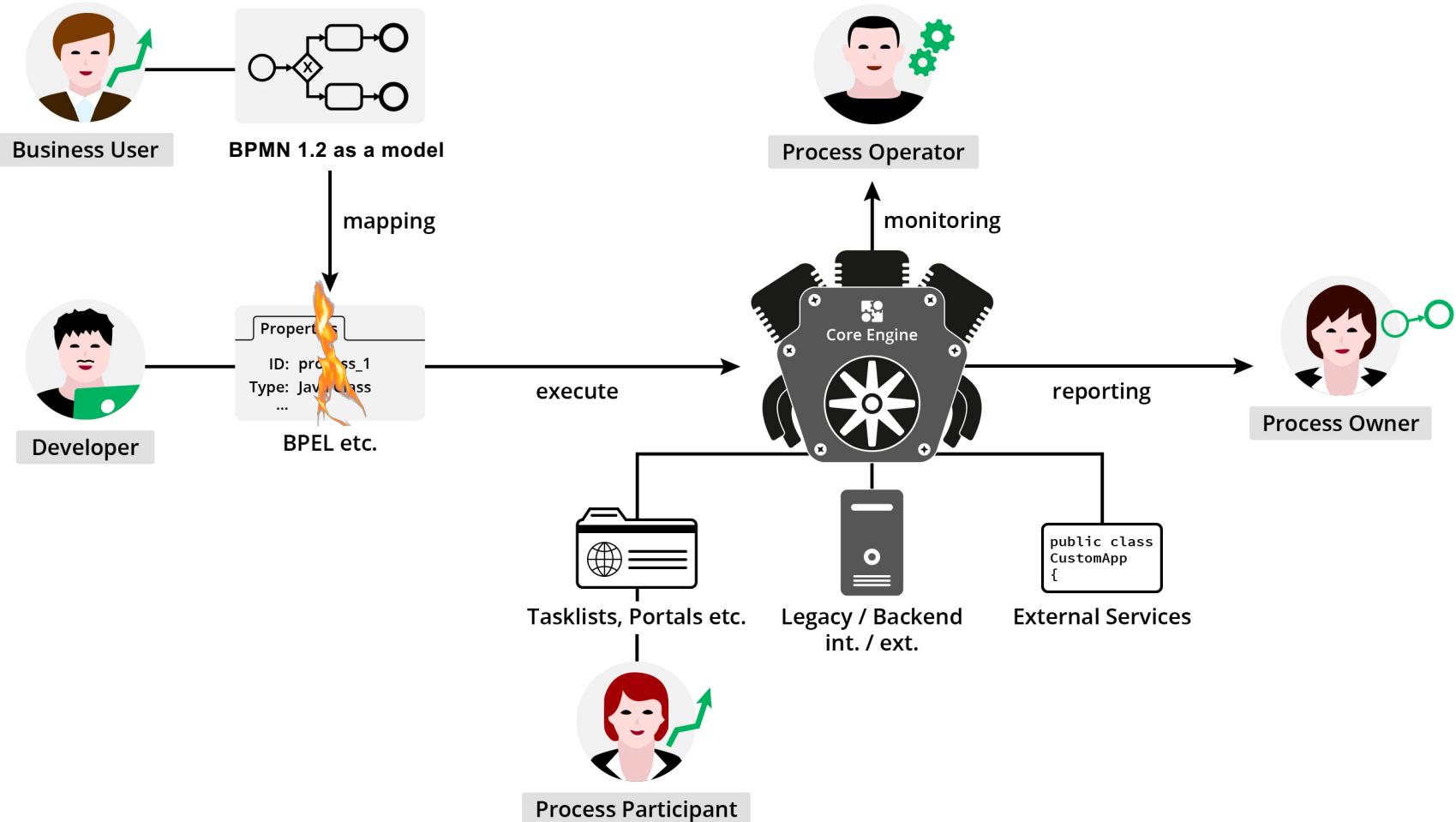
```
<?xml version="1.0" encoding="UTF-8"?>
<bpws:process name="SimpleBPEL"
  targetNamespace="http://www.camunda.com/example" ... >
...
<bpws:partnerLinks>
  <bpws:partnerLink name="client"
    myRole="SimpleBPELProvider"
    partnerLinkType="tns:SimpleBPEL"/>
  <bpws:partnerLink name="someWebservice"
    partnerRole="ServiceProvider"
    partnerLinkType="tns:SimpleService"/>
</bpws:partnerLinks>
<bpws:variables>
  <bpws:variable messageType="tns:RequestMessage" name="input"/>
  <bpws:variable messageType="tns:InvokeMessage" name="parameter"/>
</bpws:variables>
<bpws:sequence name="main scope">
  <bpws:receive name="receive"
    createInstance="yes"
    operation="initiate"
    variable="input"
    partnerLink="client"
    portType="tns:SimpleBPEL"/>
  <bpws:assign name="assign" validate="no">
    ...
  </bpws:assign>
  <bpws:invoke name="invoke"
    operation="methodName"
    inputVariable="parameter"
    partnerLink="someWebservice"
    portType="tns:ServiceInvoke"/>
</bpws:sequence>
</bpws:process>
```

Development of BPMN

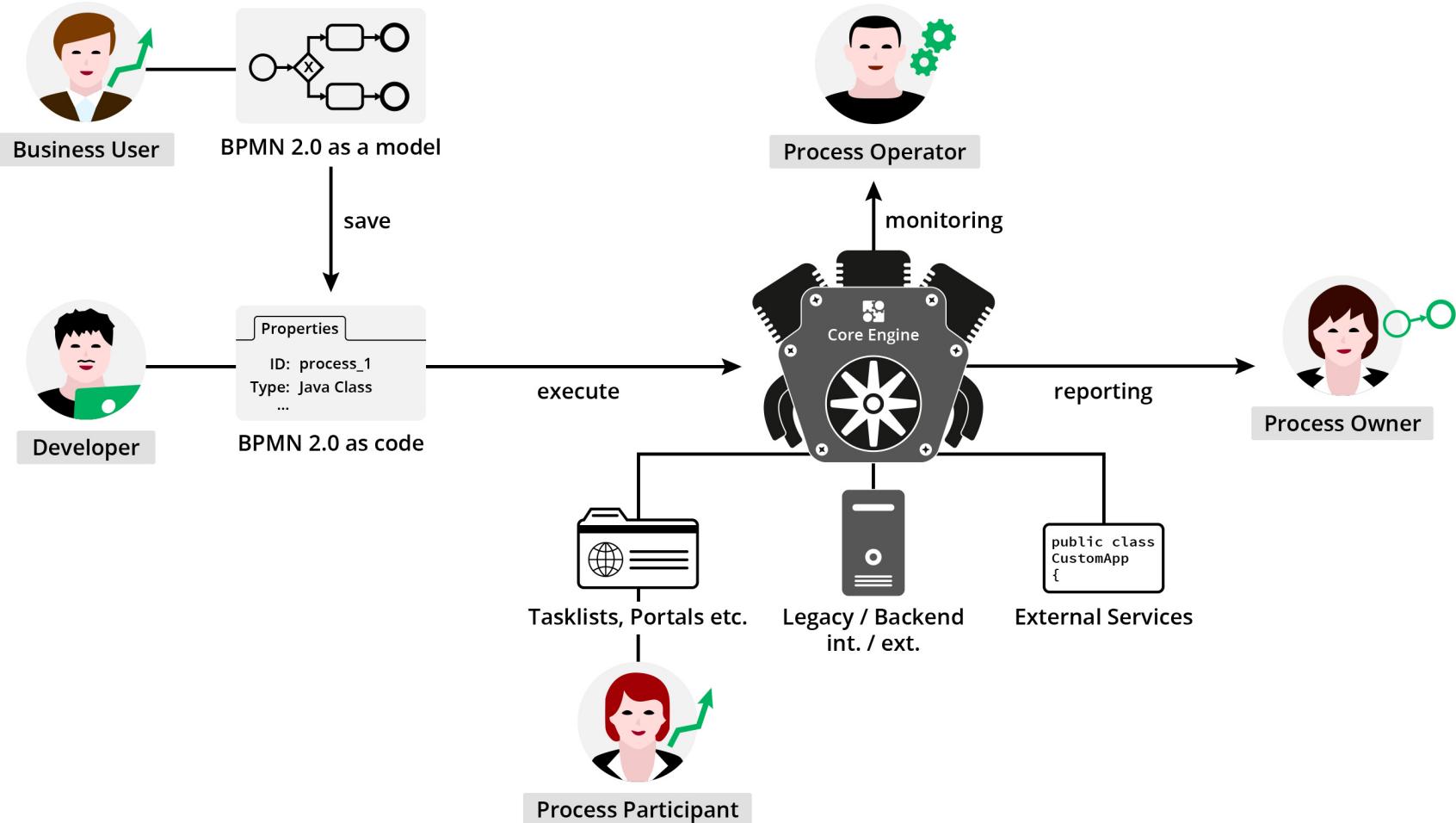
until 2002	Development Stephen A. White, IBM	
2002	First release Business Process Management Initiative (BPMI)	
2005	Development Object Management Group (OMG)	
2013	Further Standardization International Standardization Organization	

Current Version: BPMN 2.0.2 since January 2014

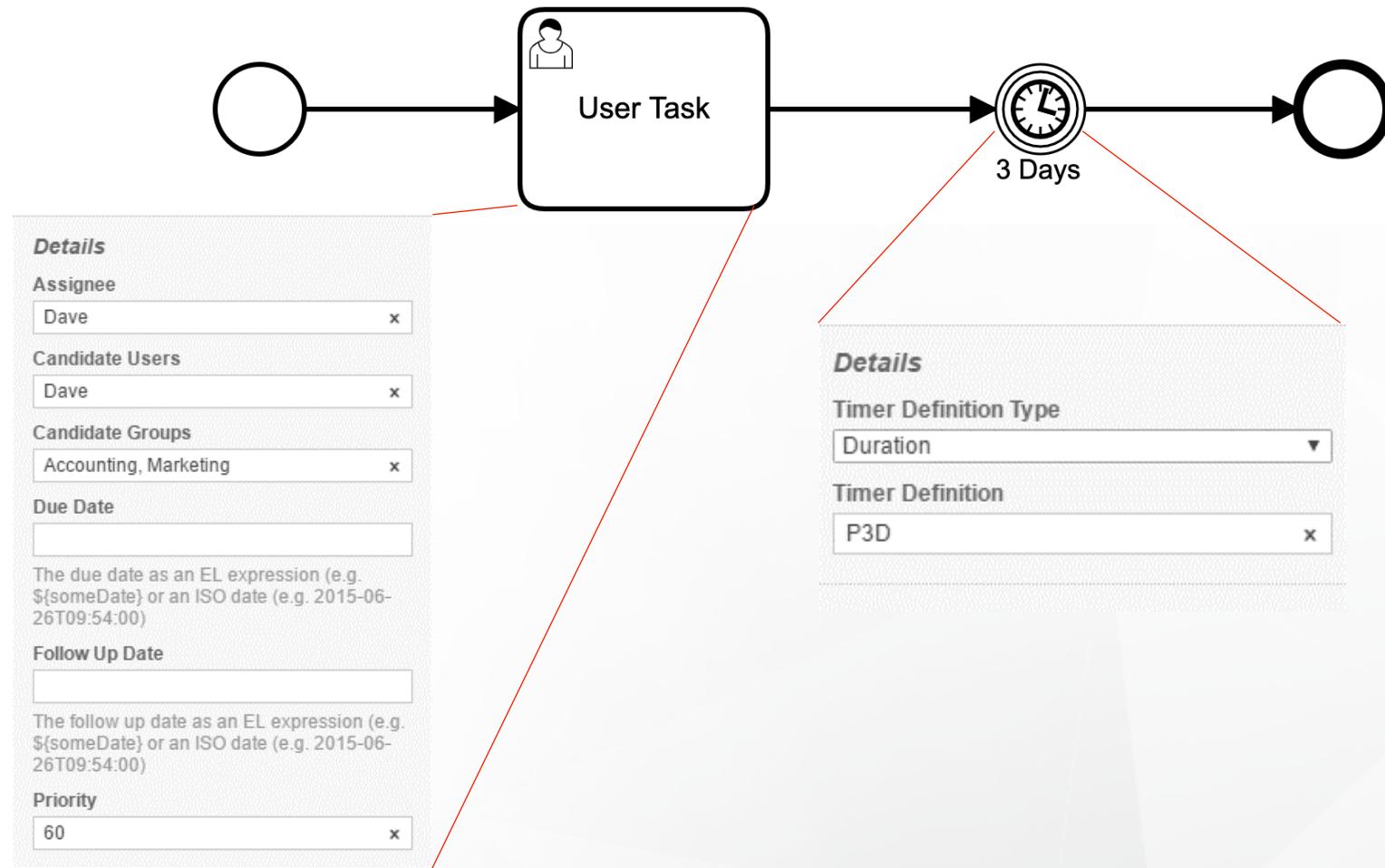
Proposed architecture with BPMN 1.2



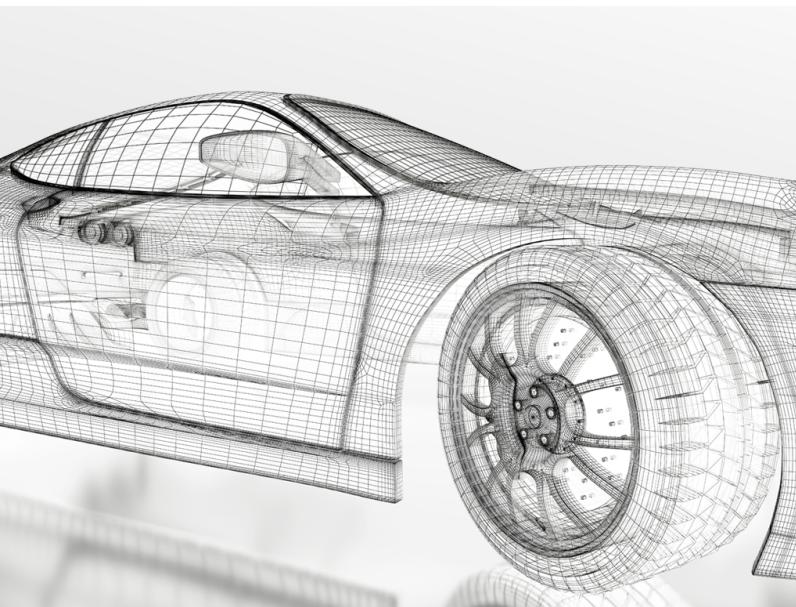
Proposed architecture with BPMN 2.0



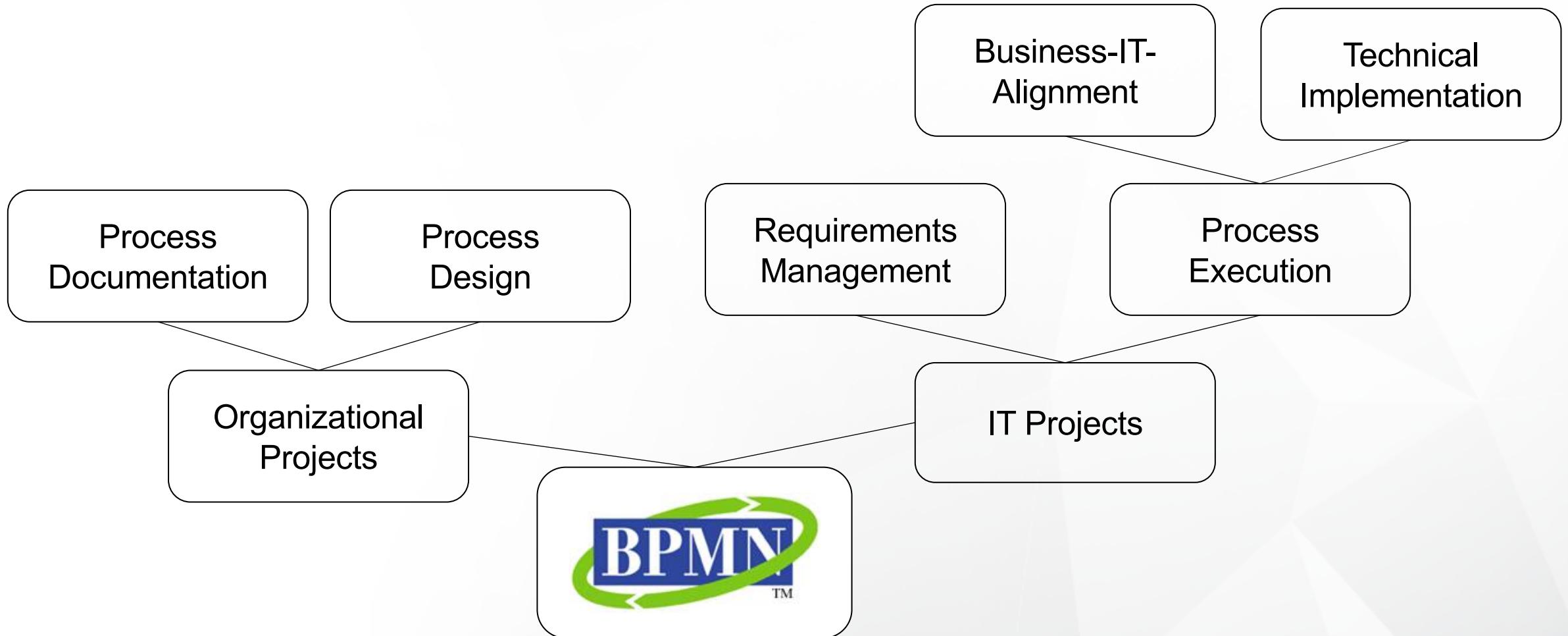
Symbols and attributes



Models and their target groups



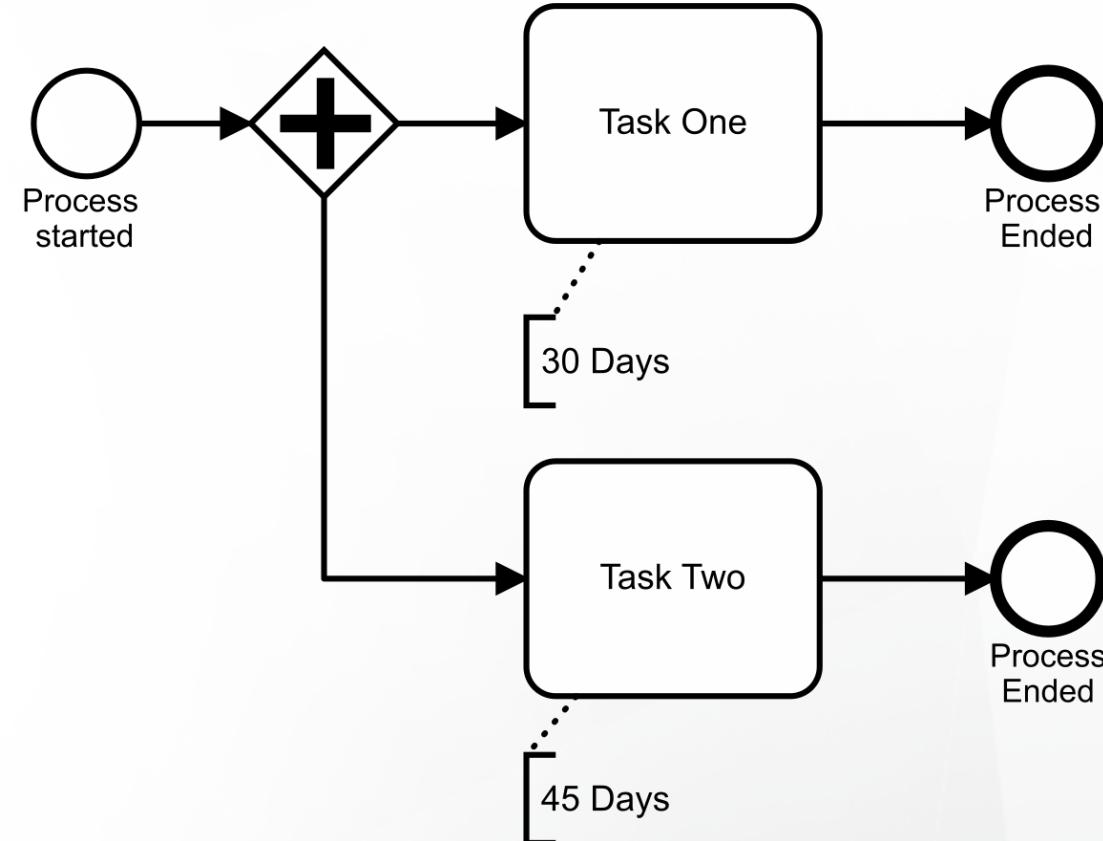
Different application scenarios



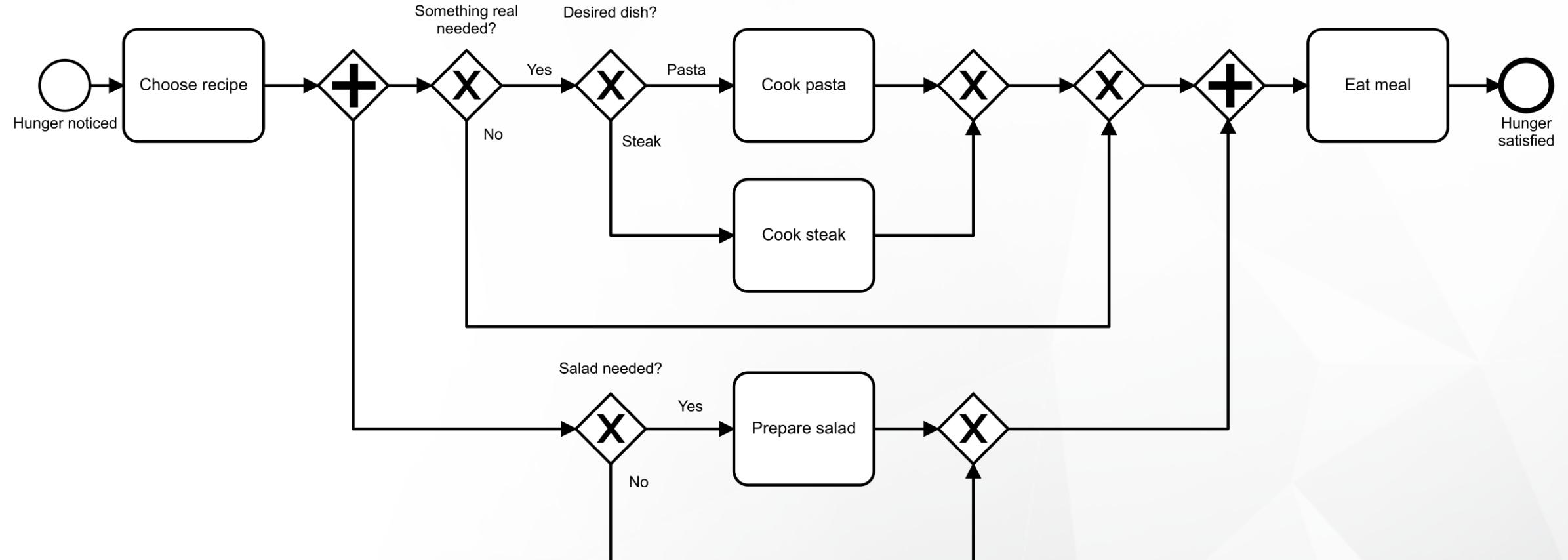


Let's continue with BPMN.....

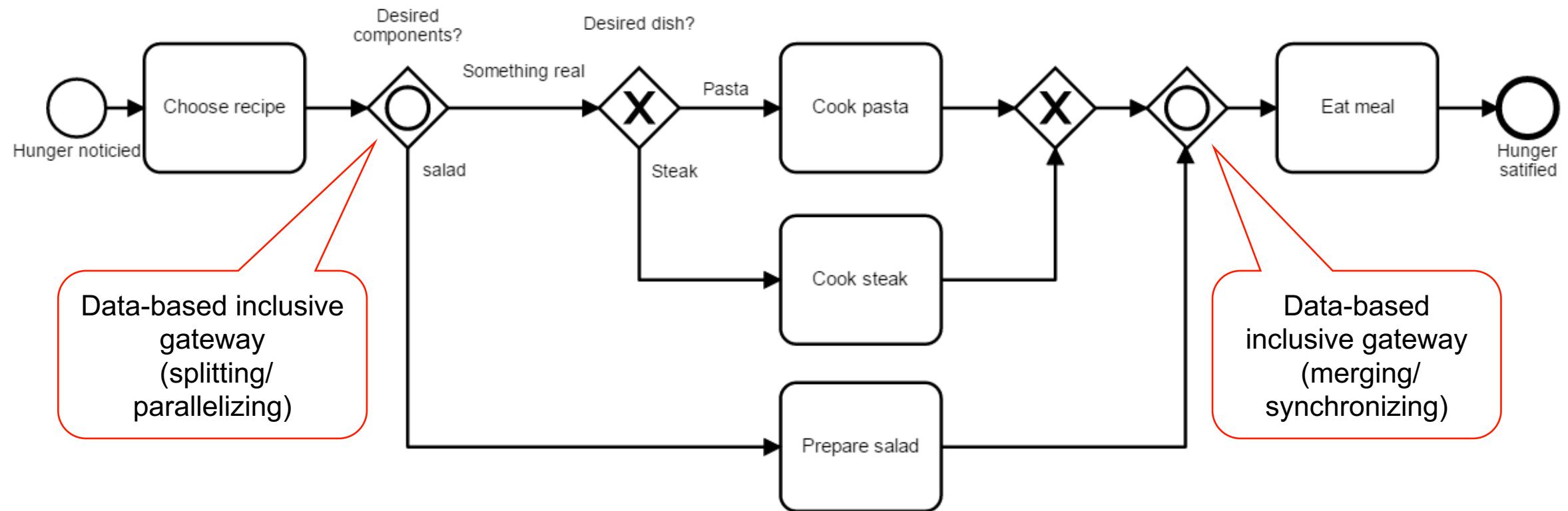
How long does the process instance take to finish?



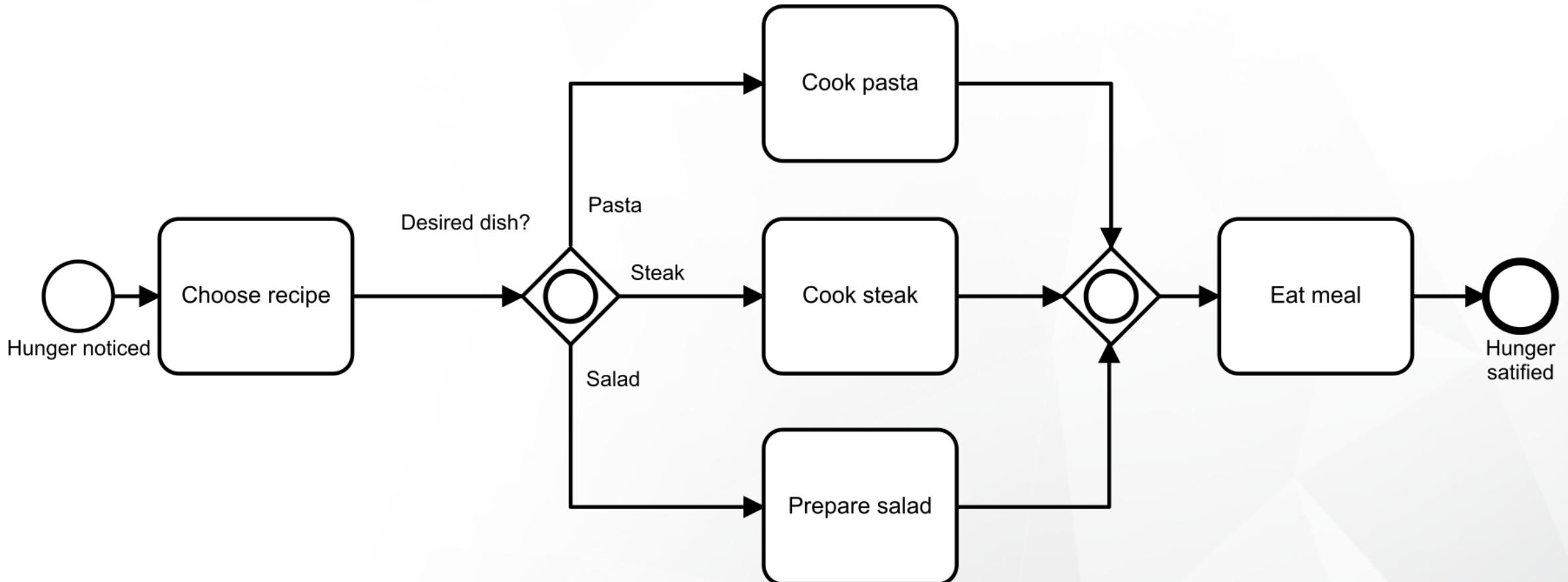
Various options in the combination of our meal



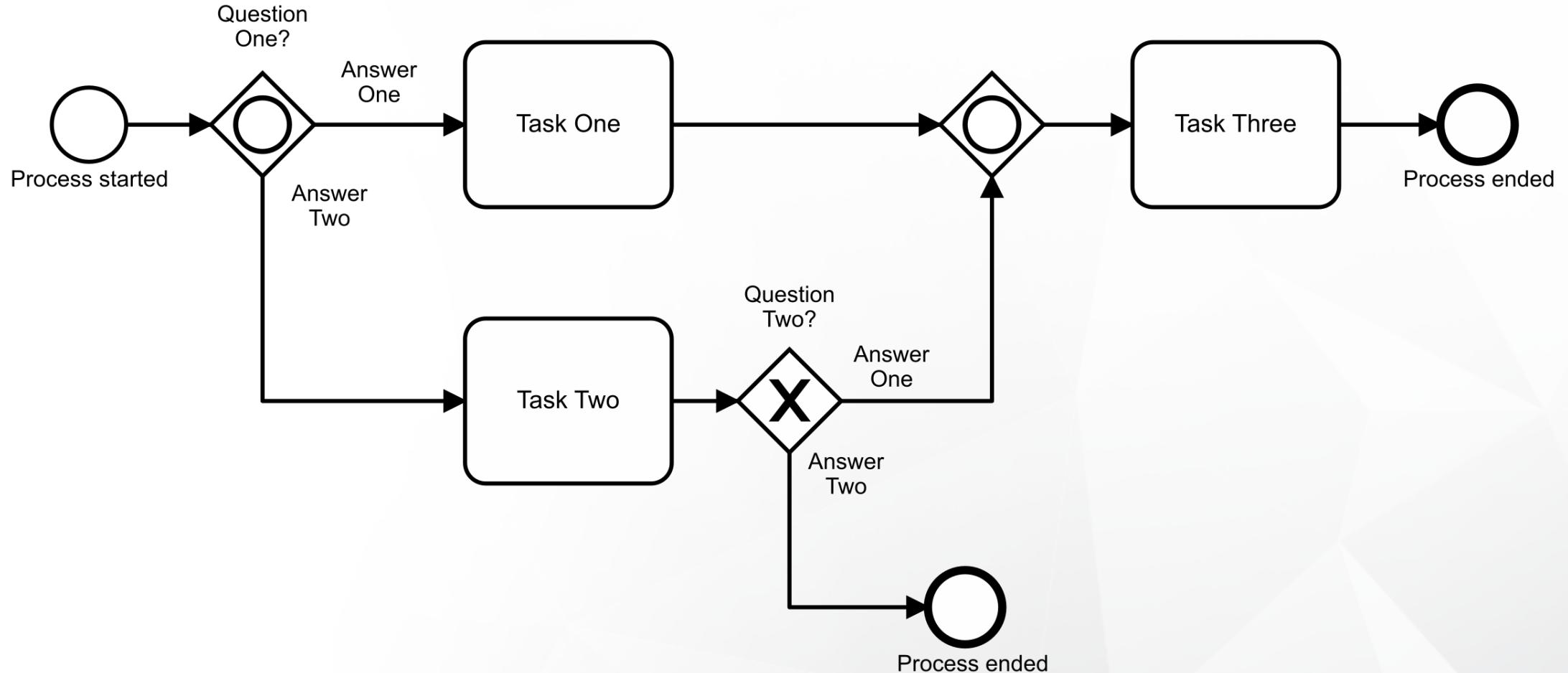
The OR-gateway enables a compact representation of complex path variants



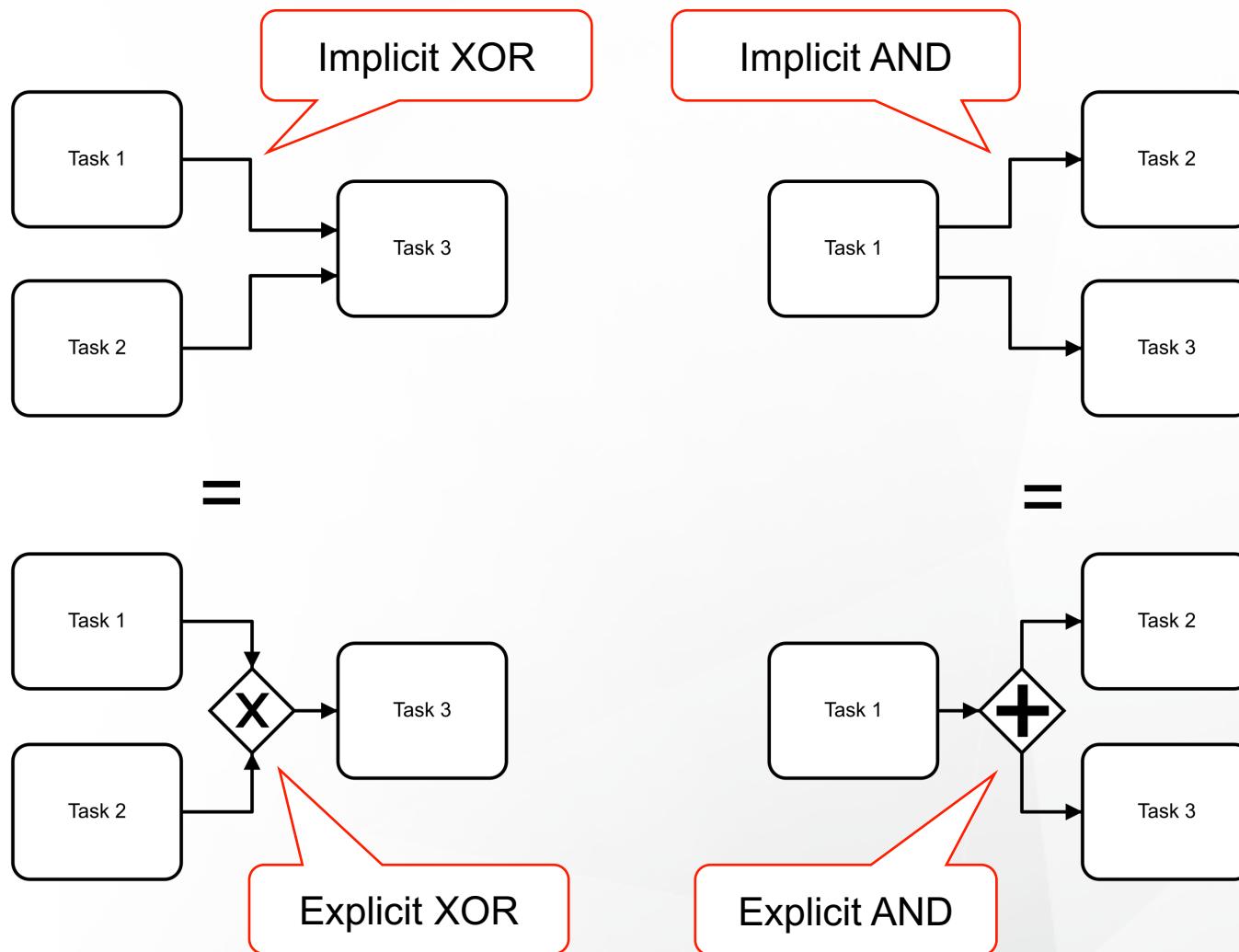
An incredibly (?) compact version



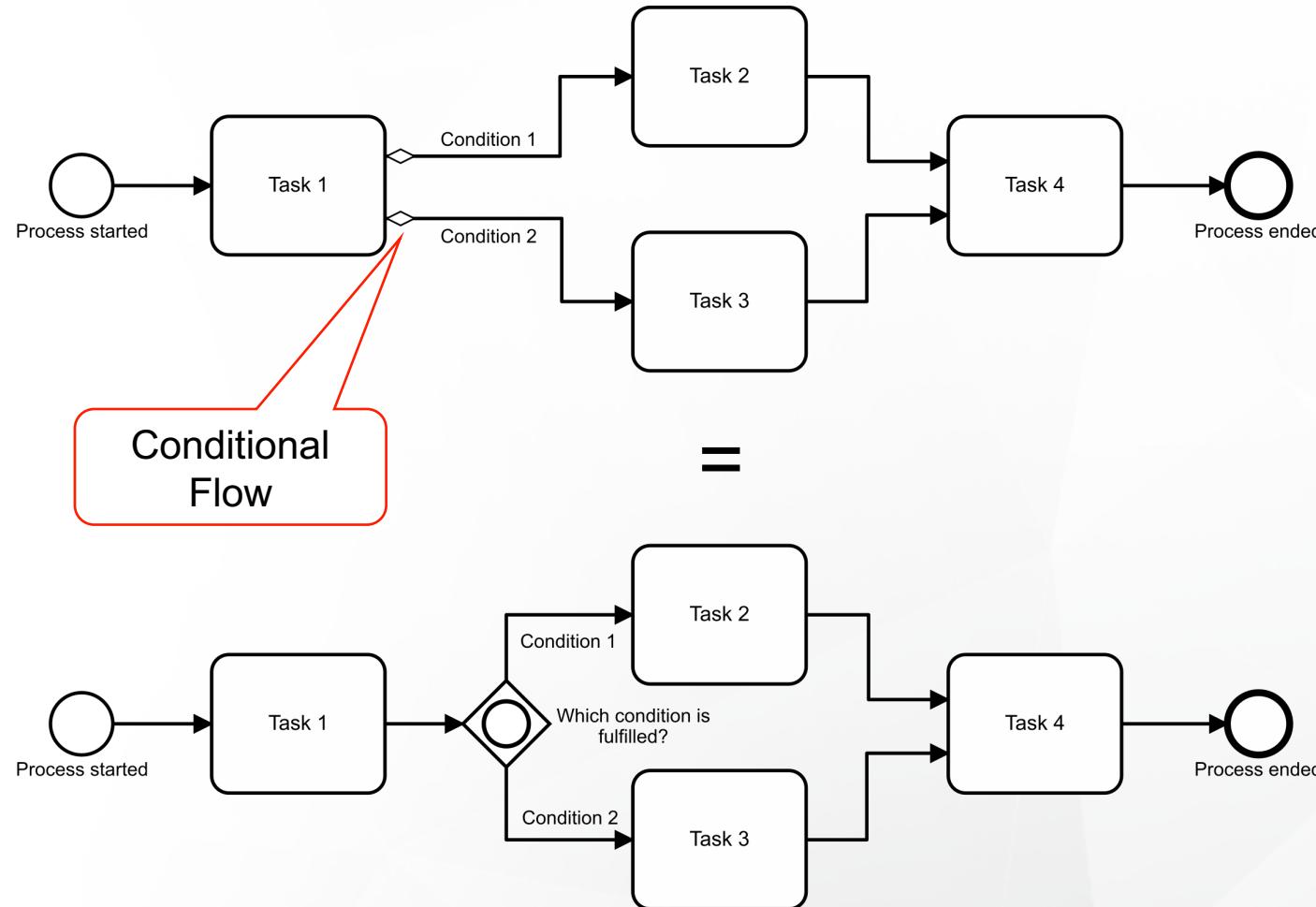
How long does the second OR-gateway need to wait?



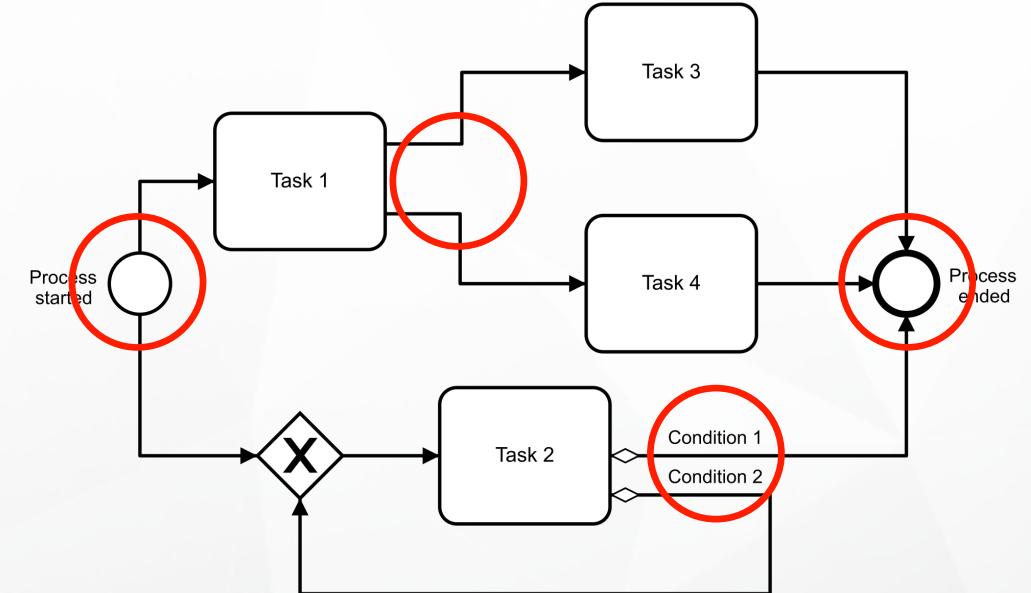
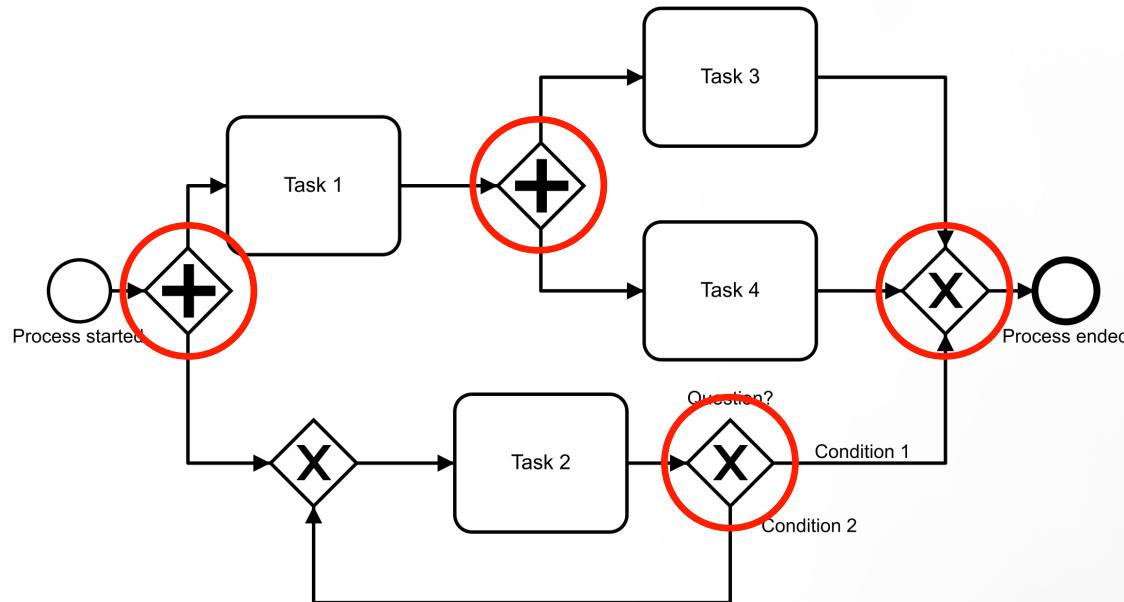
Implicit vs. explicit modeling



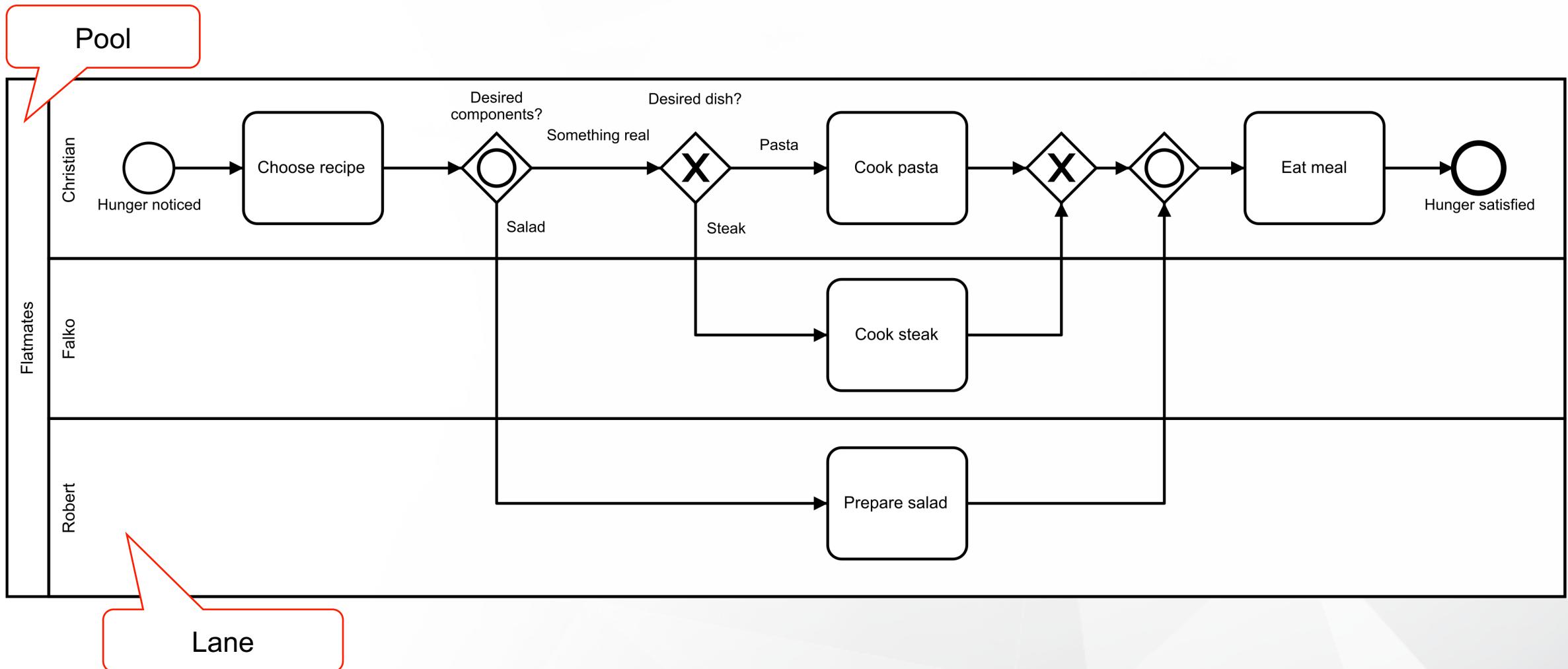
OR-Split with conditional flows



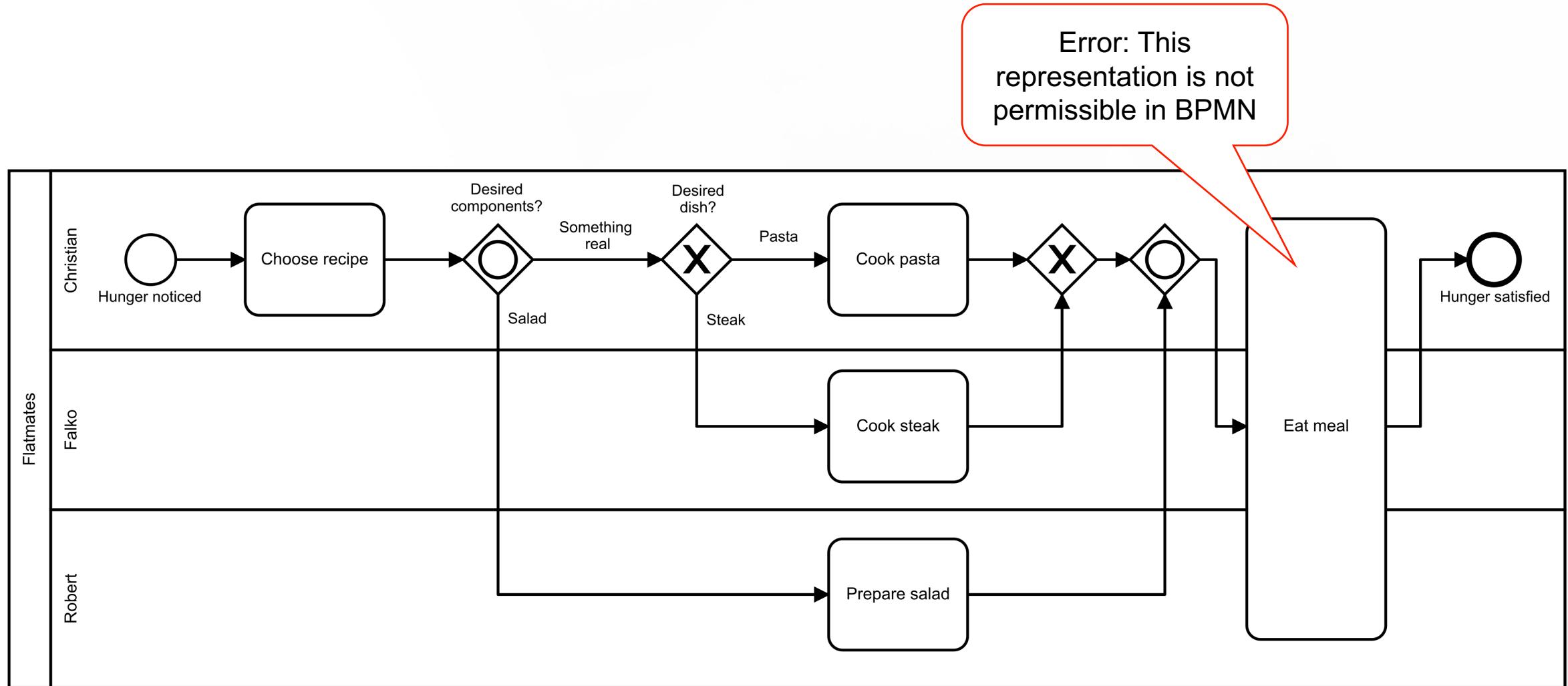
The same process?



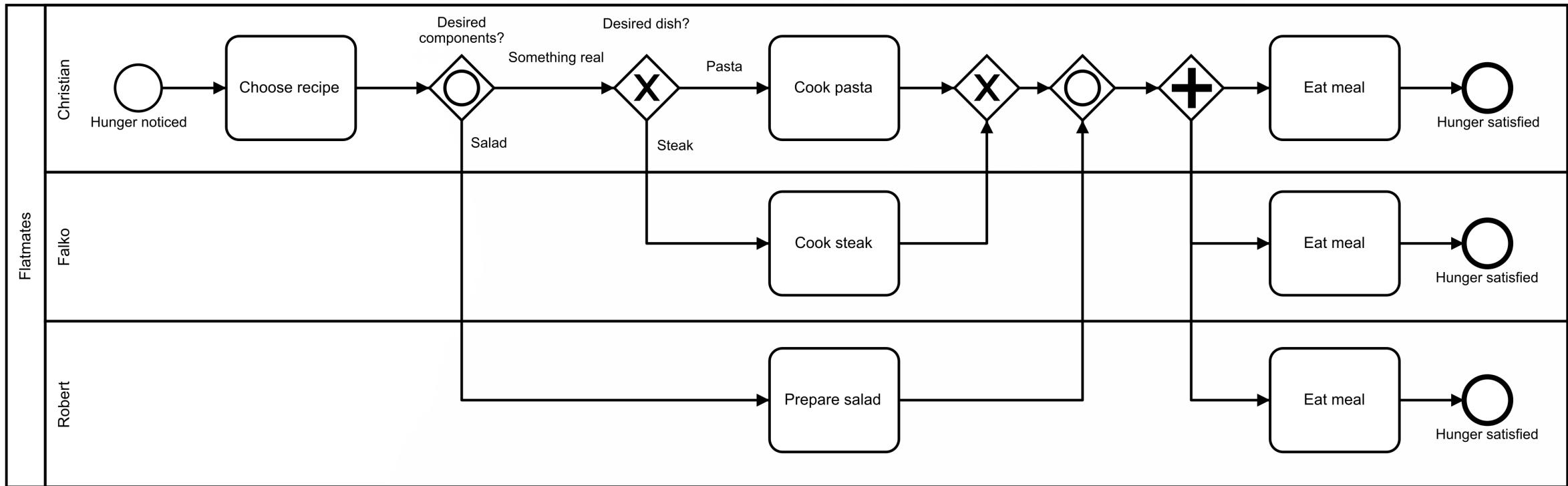
Responsibilities represented using lanes



Incorrect lane handling



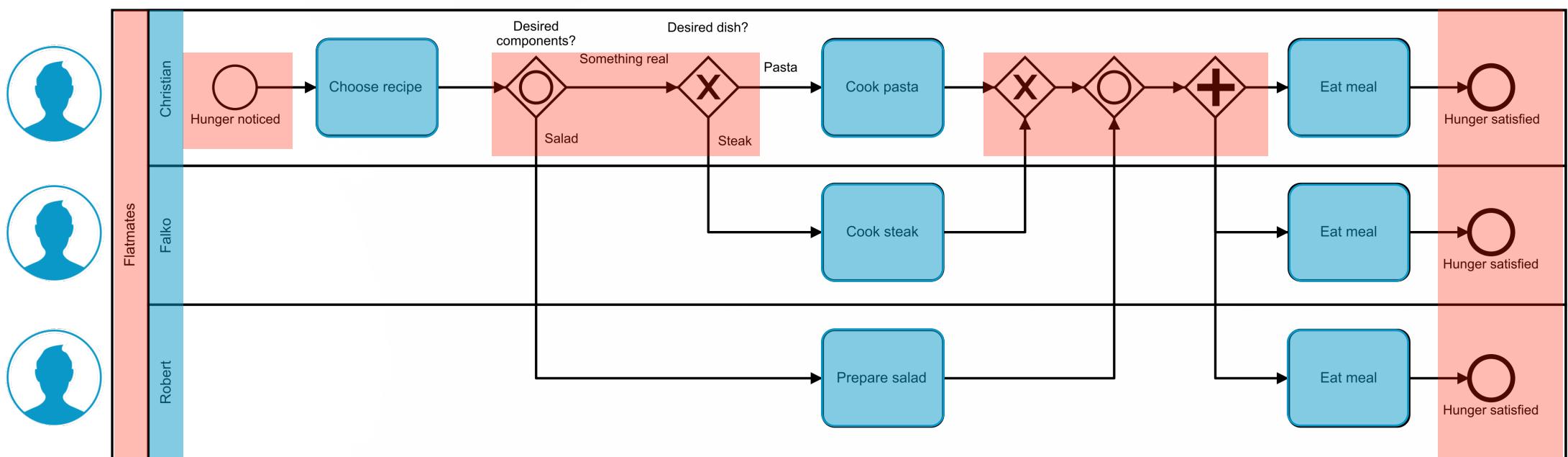
Correct lane handling



Formal vs. practical point of view

C

The pool controls
the process



The process
controls the
participants

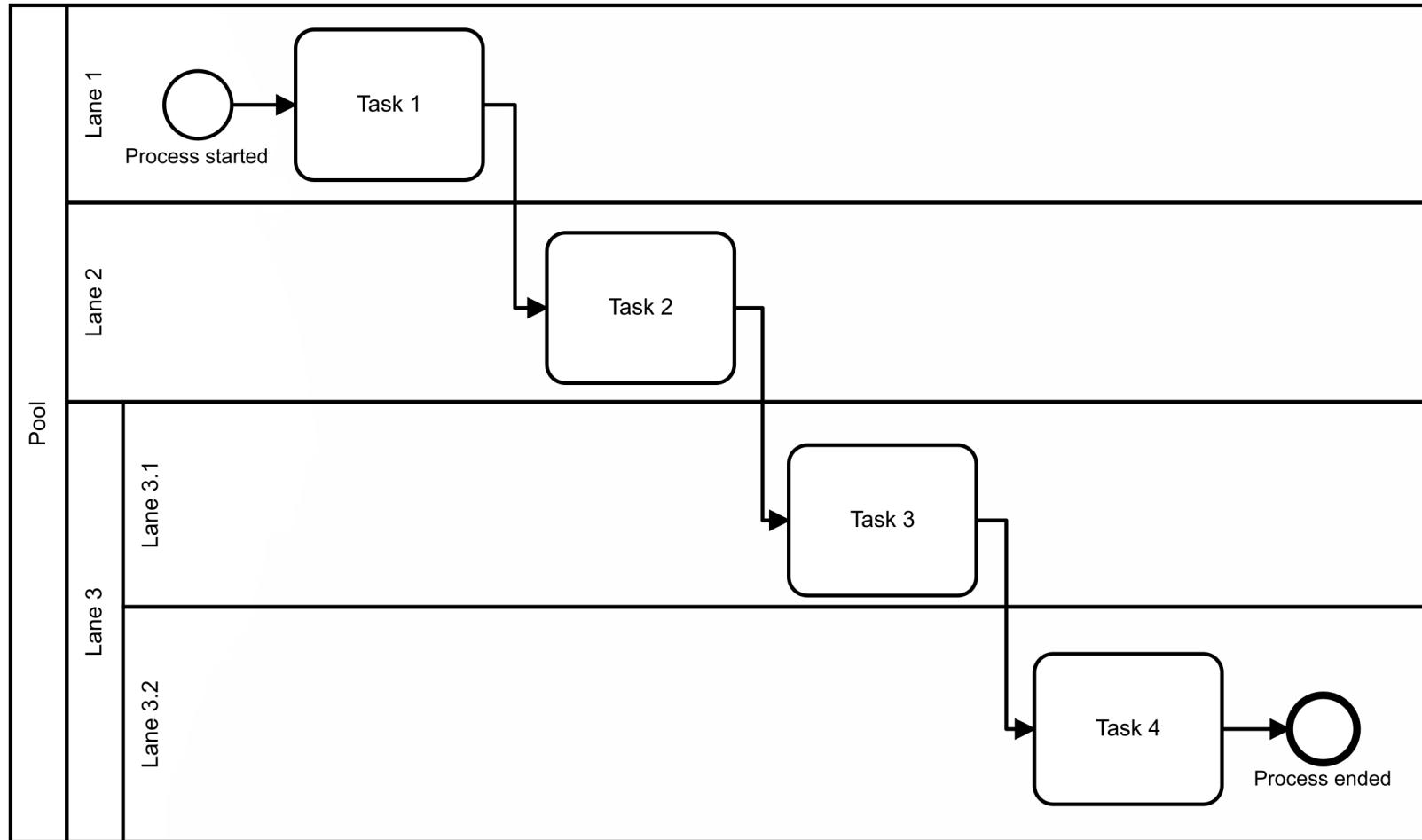


Controlled by the pool



Controlled by humans

Interlacing lanes into sub-lanes





Cawemo

Demo

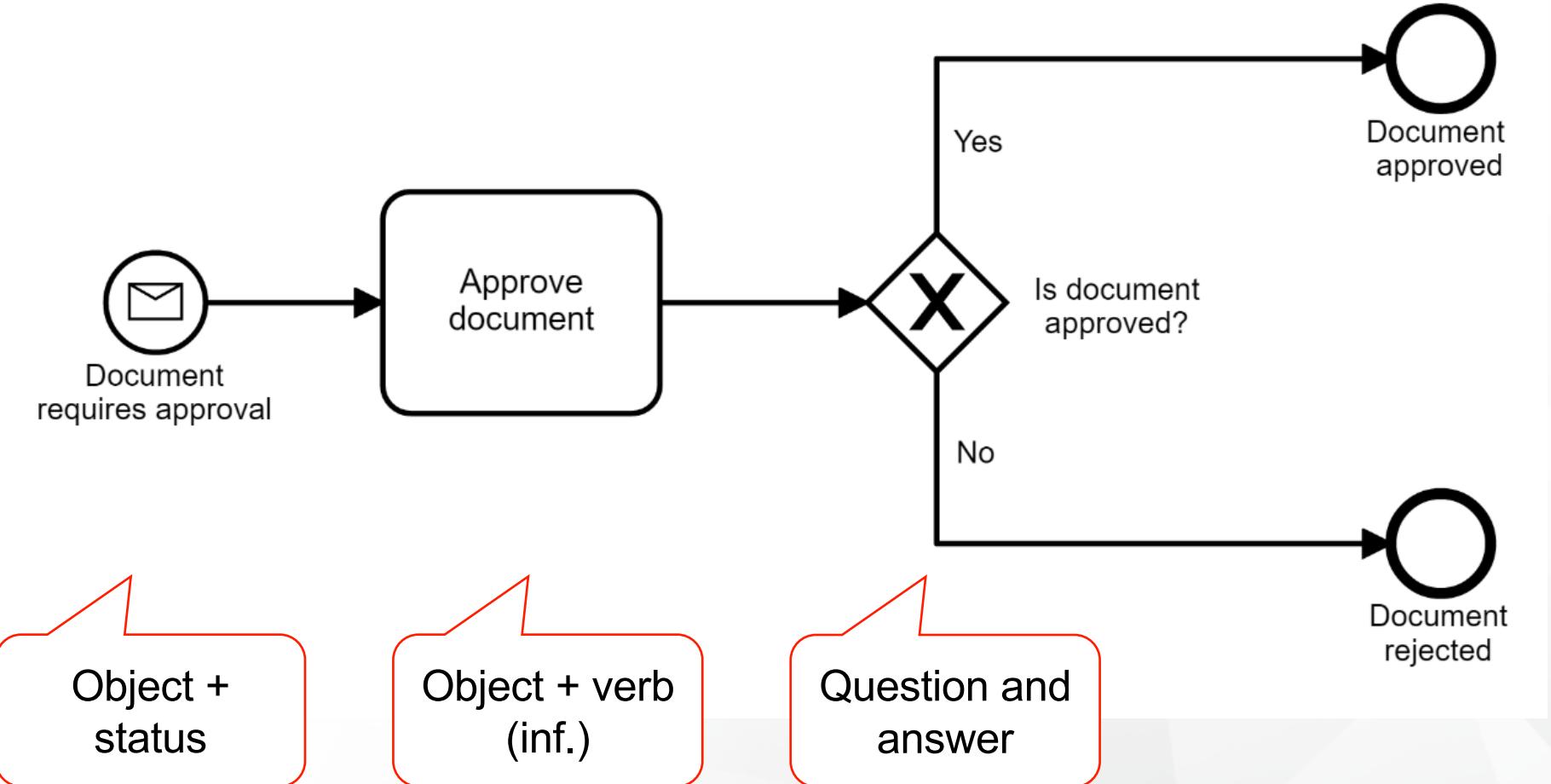


Dispatch of Goods

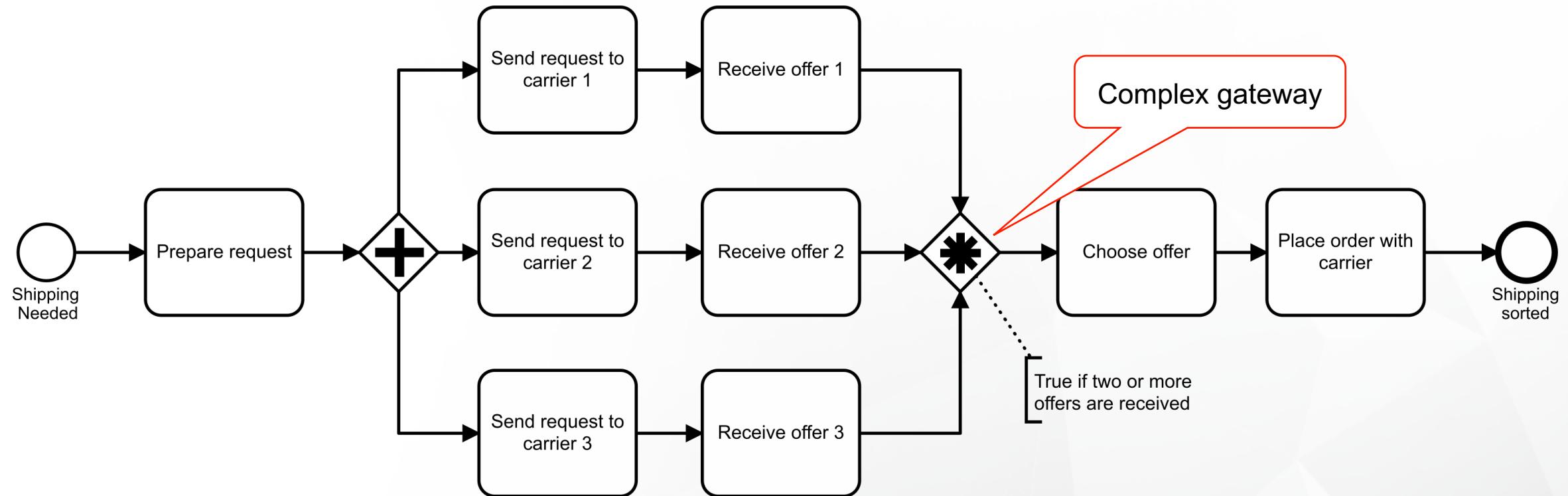
Exercise

<https://training.camunda.com/bpmn/>
user: bpmn
password: camundarocks

Some conventions: naming and layout



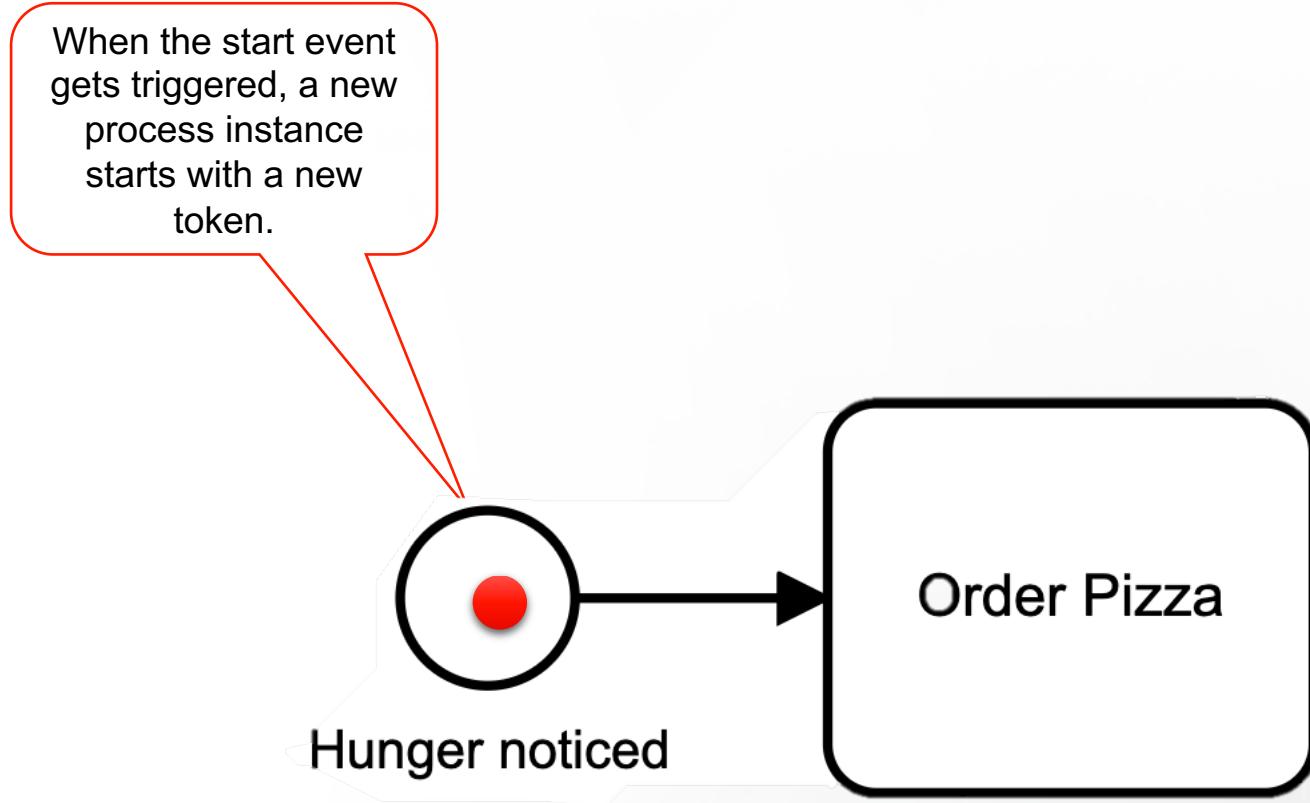
The complex gateway



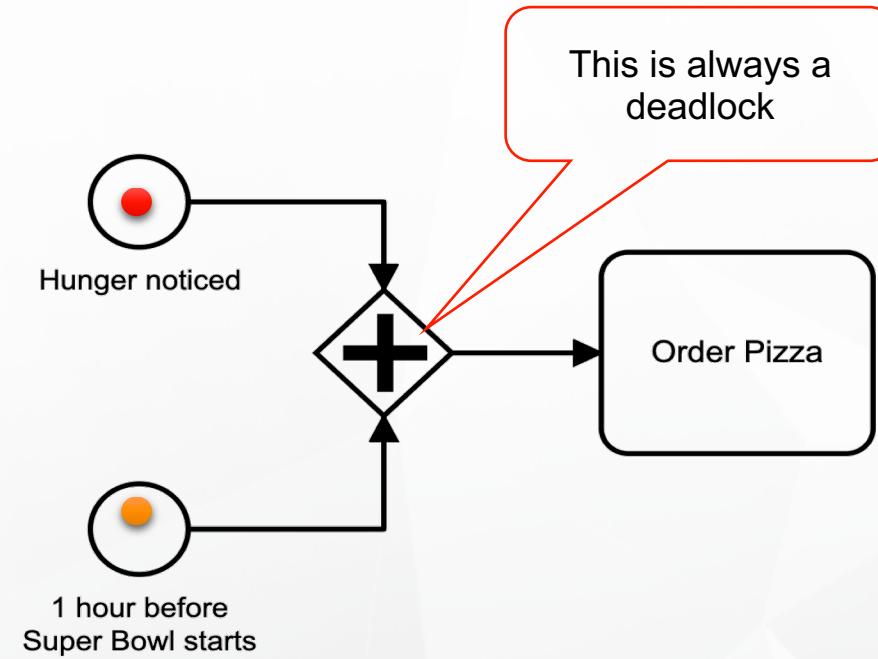
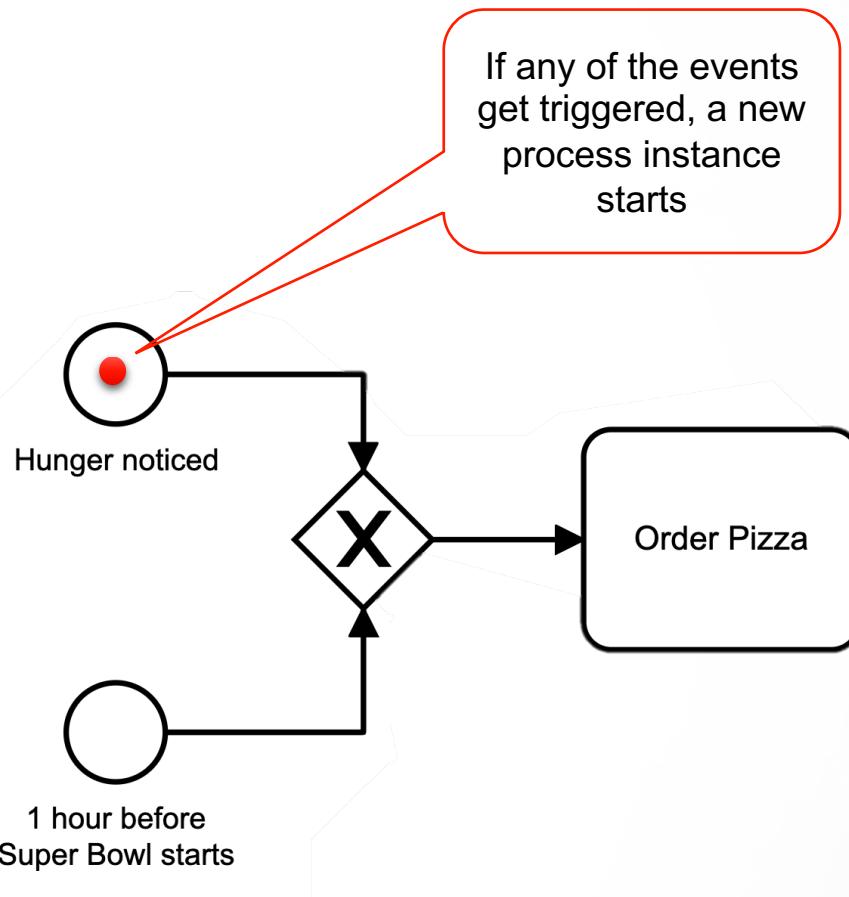


Events in BPMN

Start Event



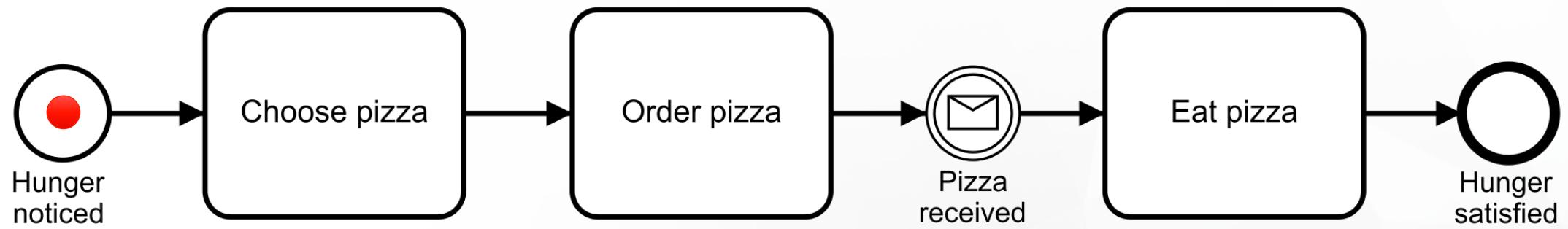
Multiple Start Events are possible!



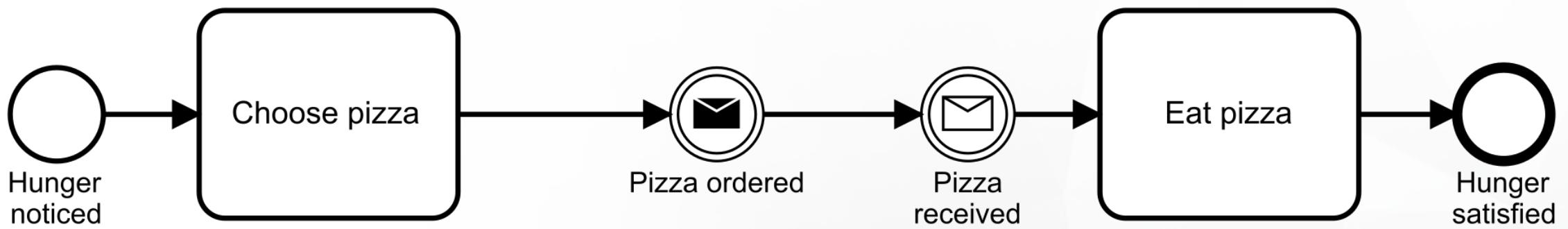
Most important events in BPMN

	Catching events		Throwing events	
	Start events	Intermediate events	End events	
	The process is started by the event. 	The process continues only, if the event occurs. 	The event is reacted to, the activity is canceled. 	The process triggers the event and continues immediately. 
None: Untyped events; none intermediate events can mark a change of status.				
Message: Receiving and sending messages.				
Timer: Cyclic timer event, points in time, or time spans.				
Conditional: Reacting to changed conditions and relation to business rules.				

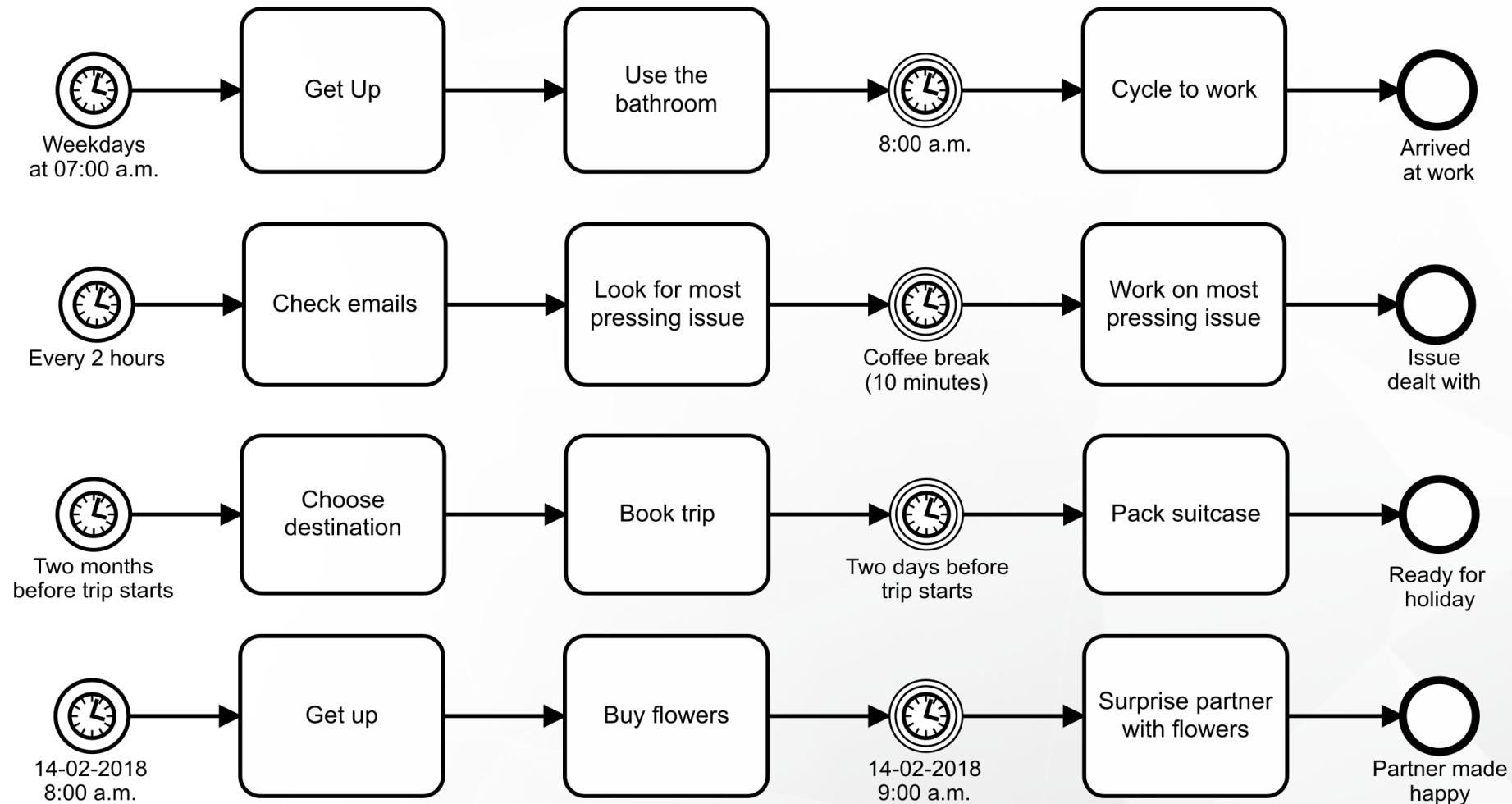
The process will wait for the message



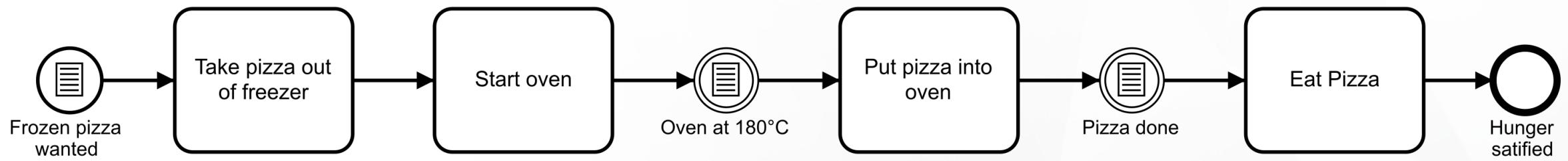
„Throwing“ message event



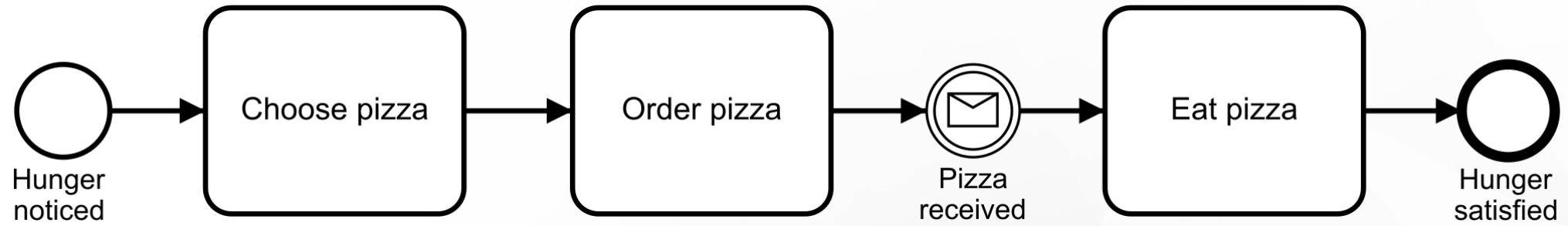
Examples for timer events



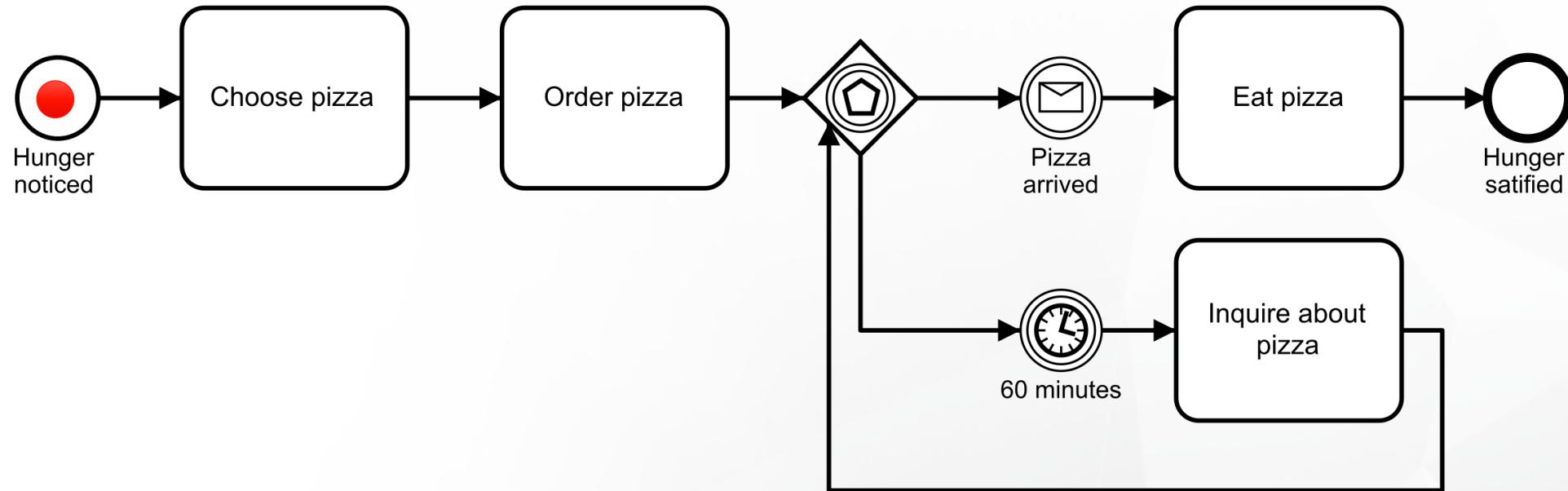
The conditional event



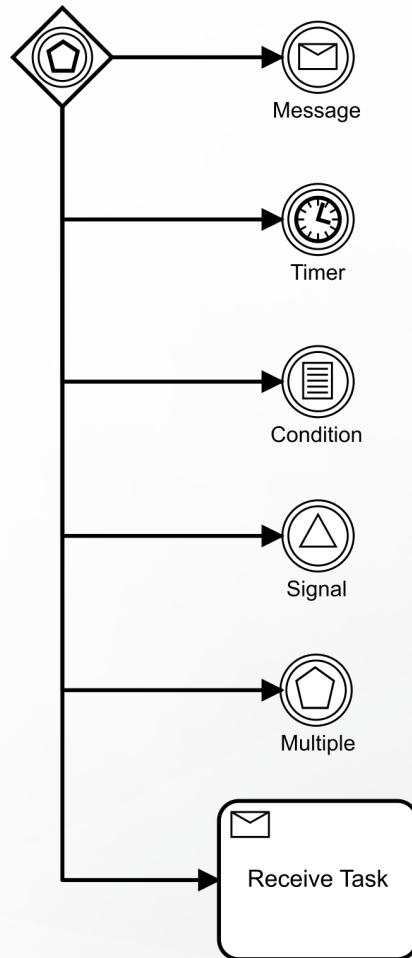
What happens if the pizza didn't arrive?



The first event that is triggered determines the sequence flow



Applying the event-based gateway



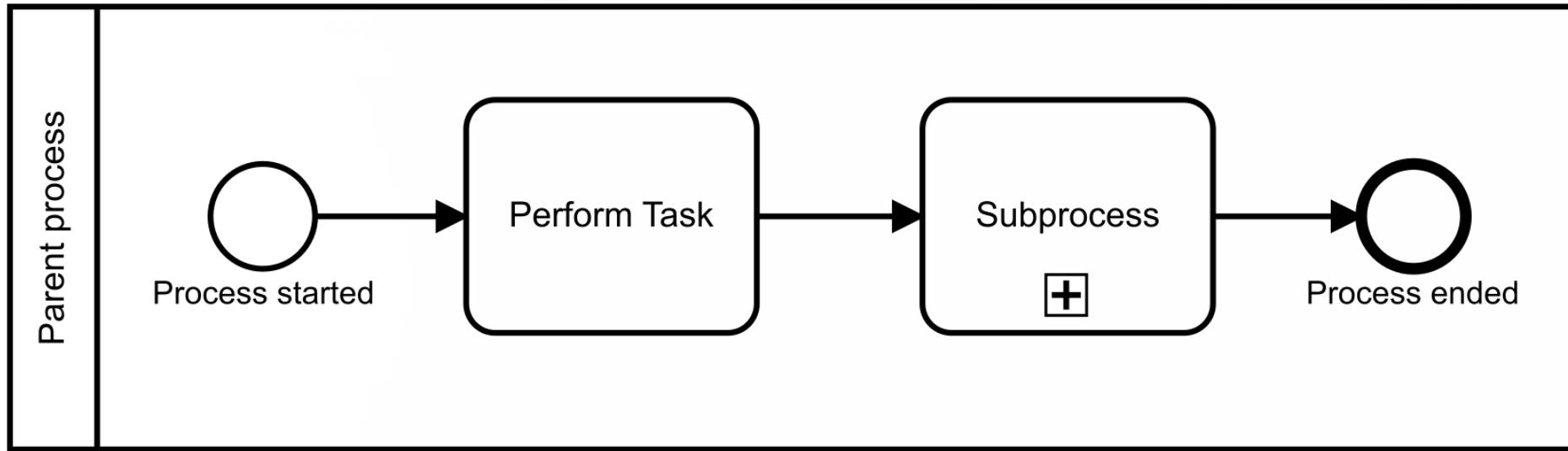


Recourse

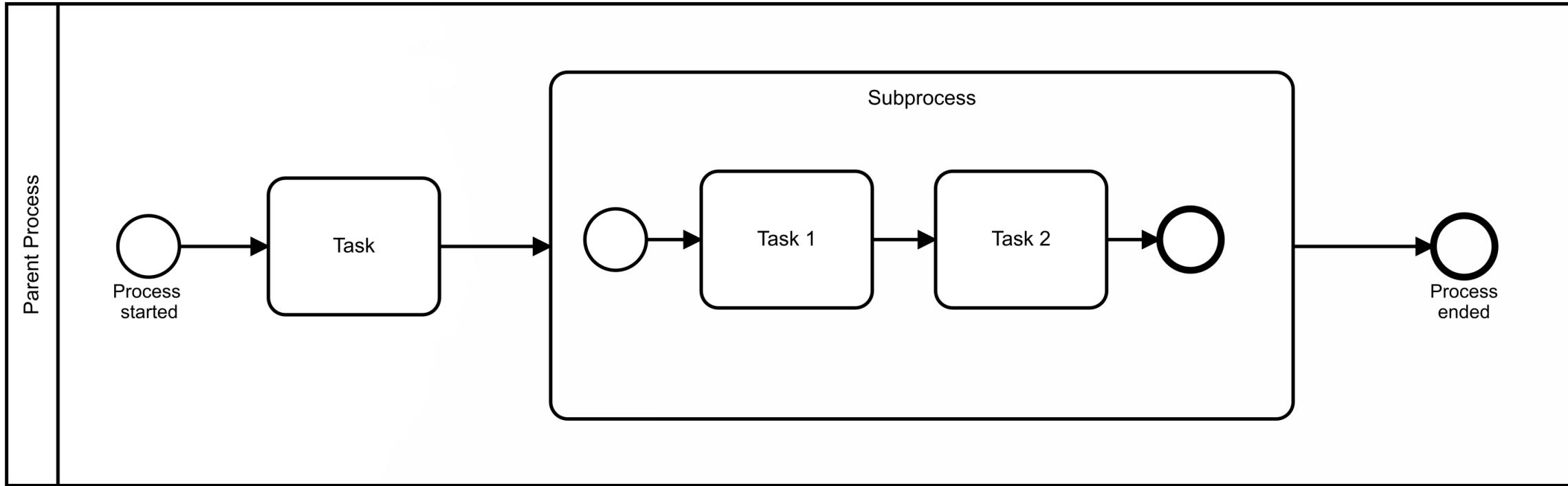
Exercise

<https://training.camunda.com/bpmn/>
user: bpmn
password: camundarocks

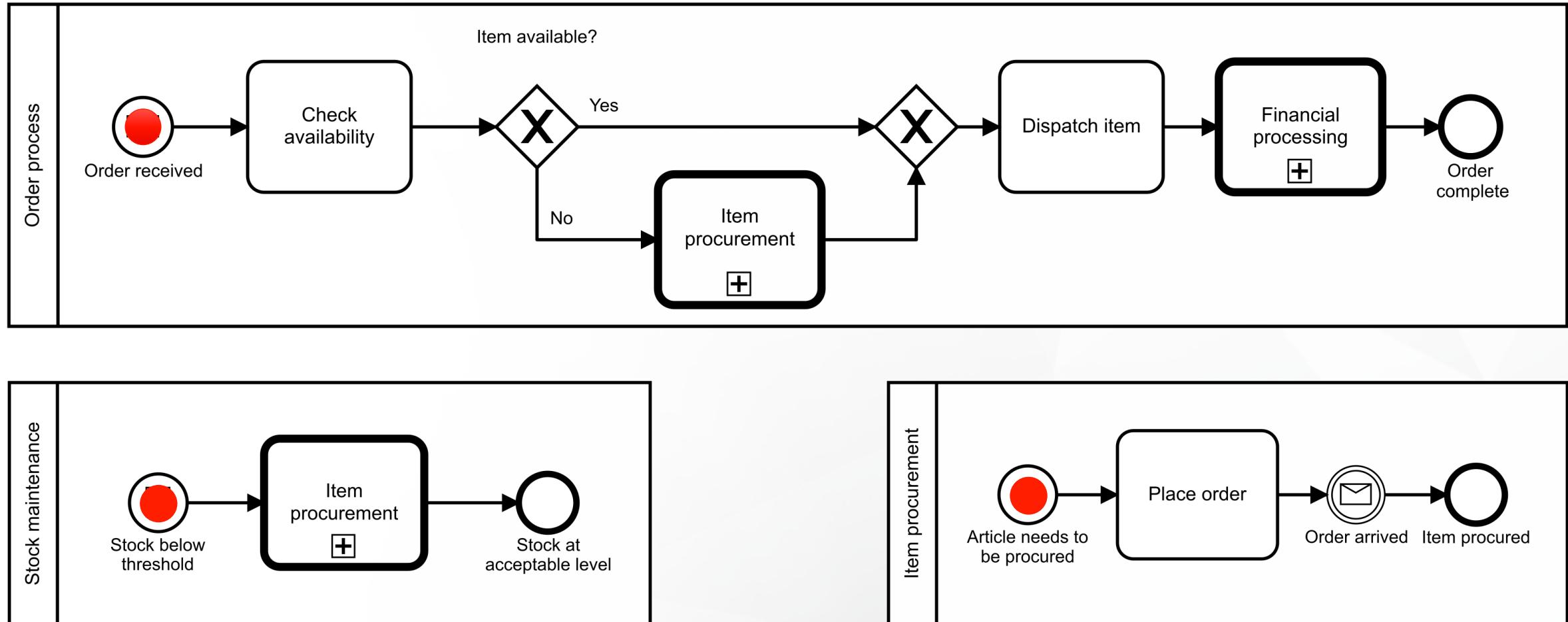
A task and a collapsed subprocess



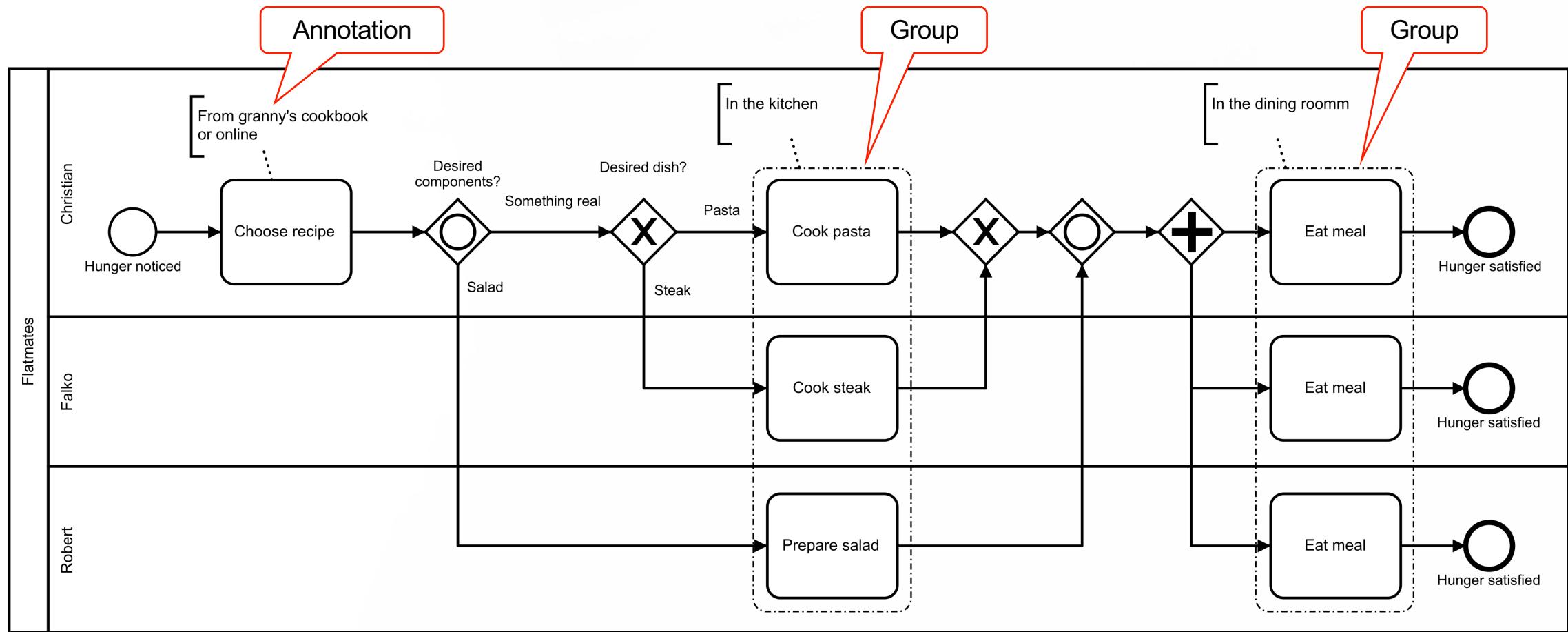
The expanded subprocess



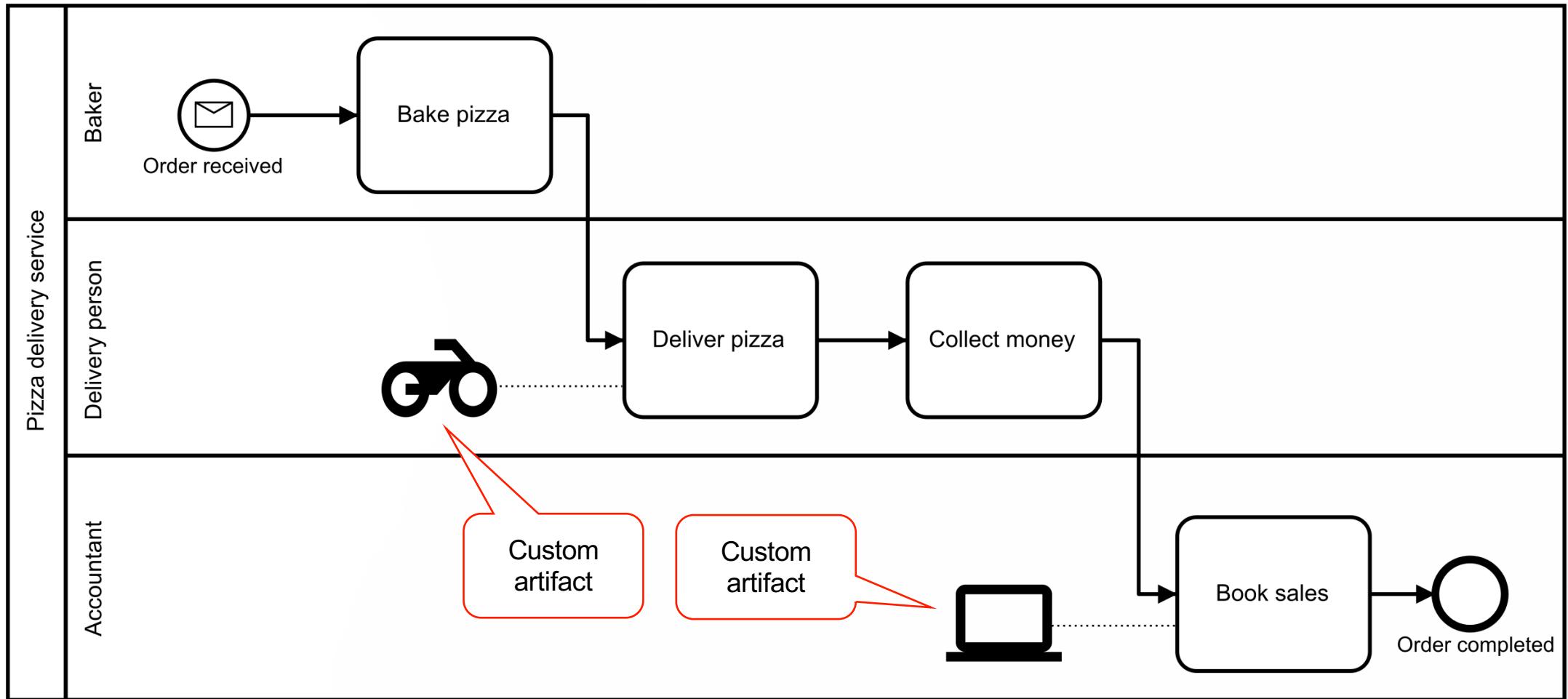
Reusing process models using call activities



Annotations and groups



Custom artifacts



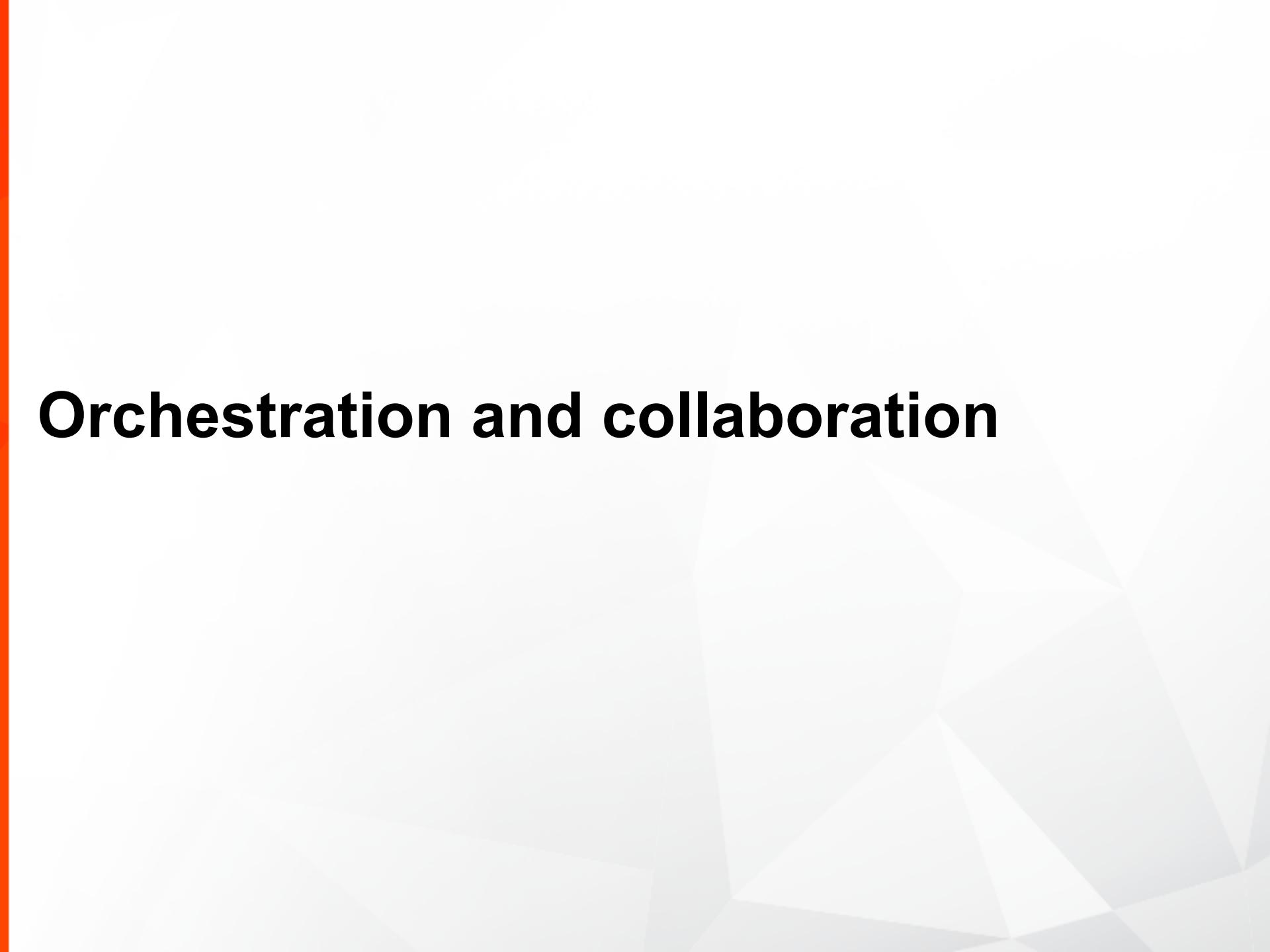
Search for Errors

Exercise

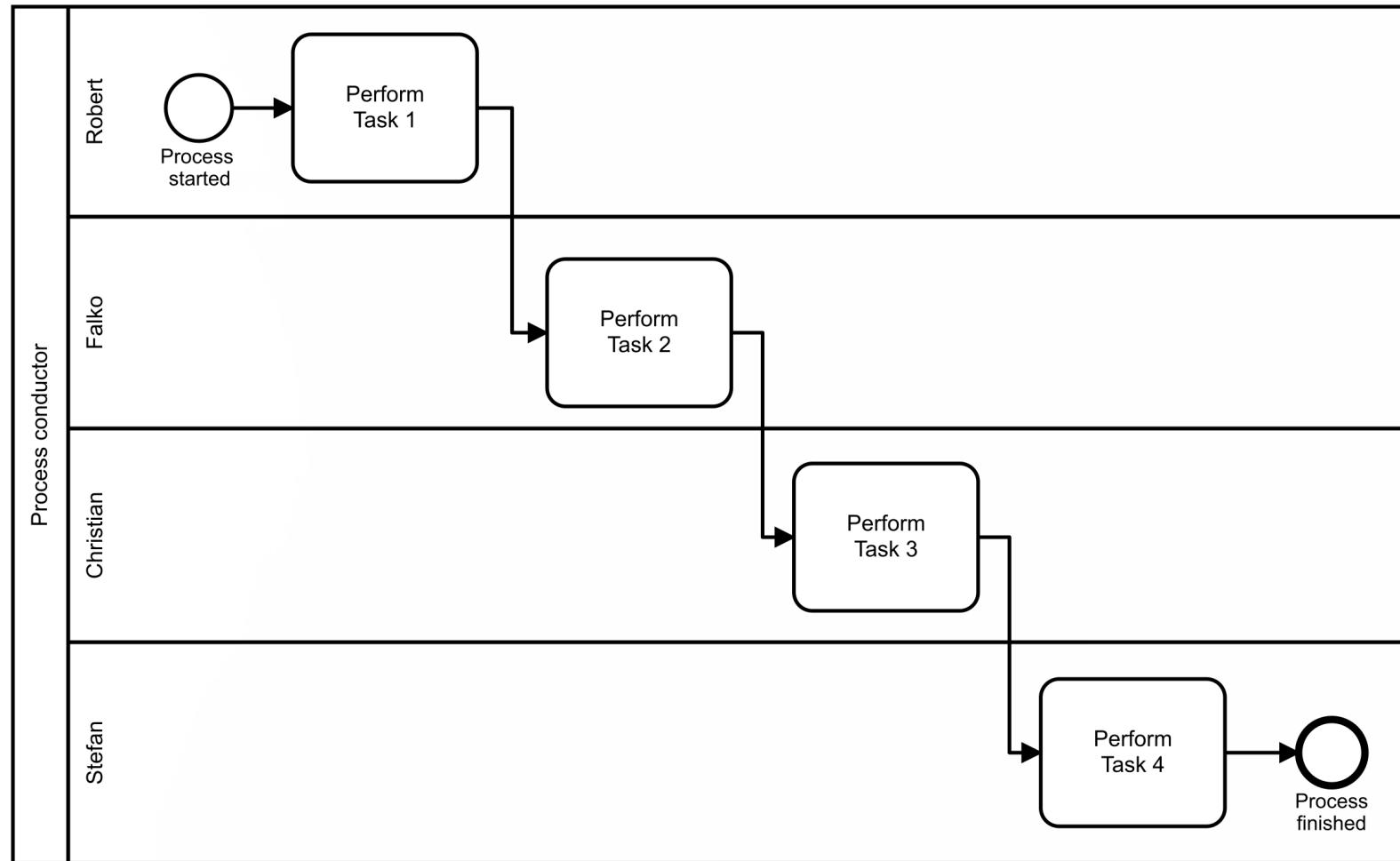
<https://training.camunda.com/bpmn/>
user: bpmn
password: camundarocks



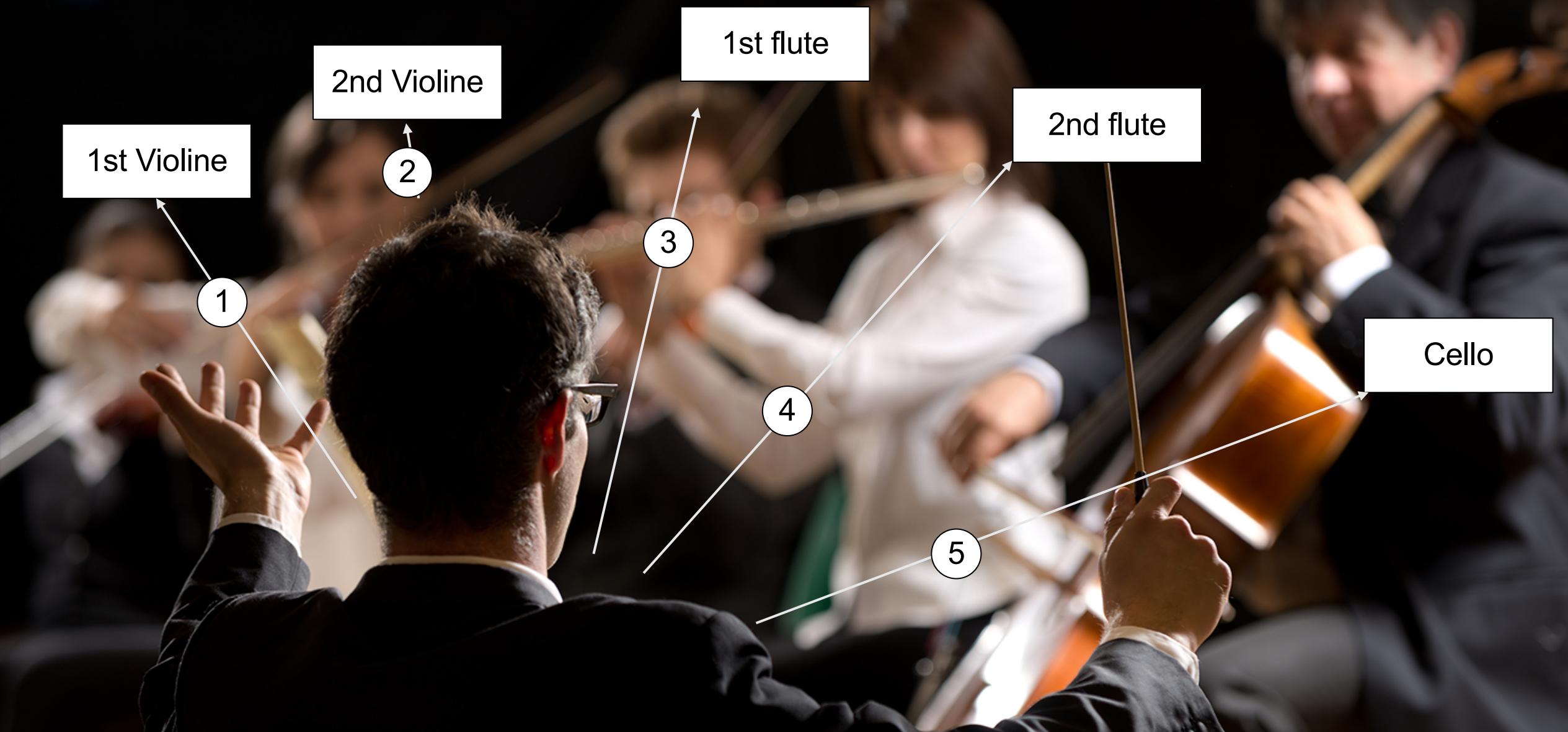
Orchestration and collaboration



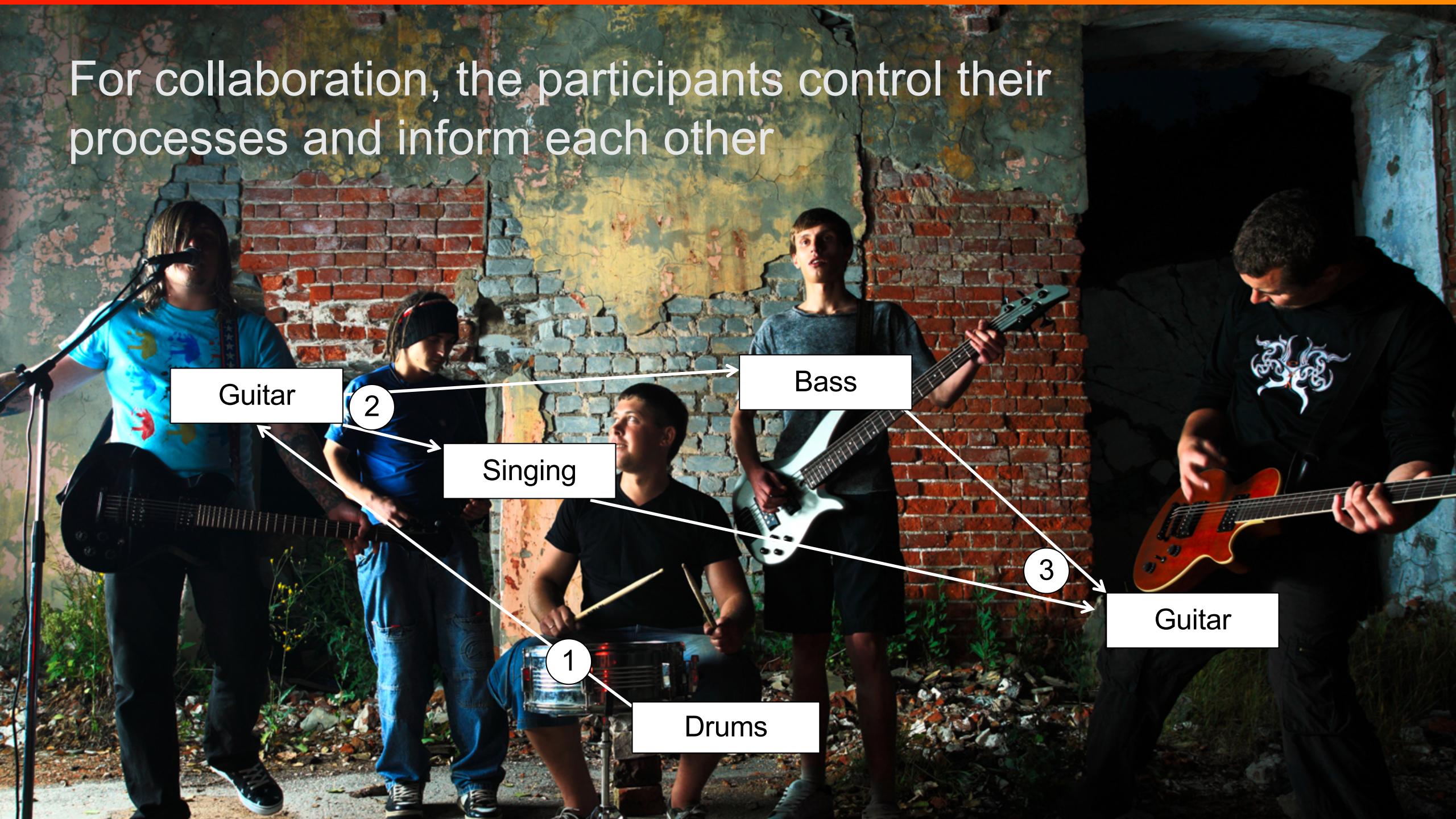
The process orchestration



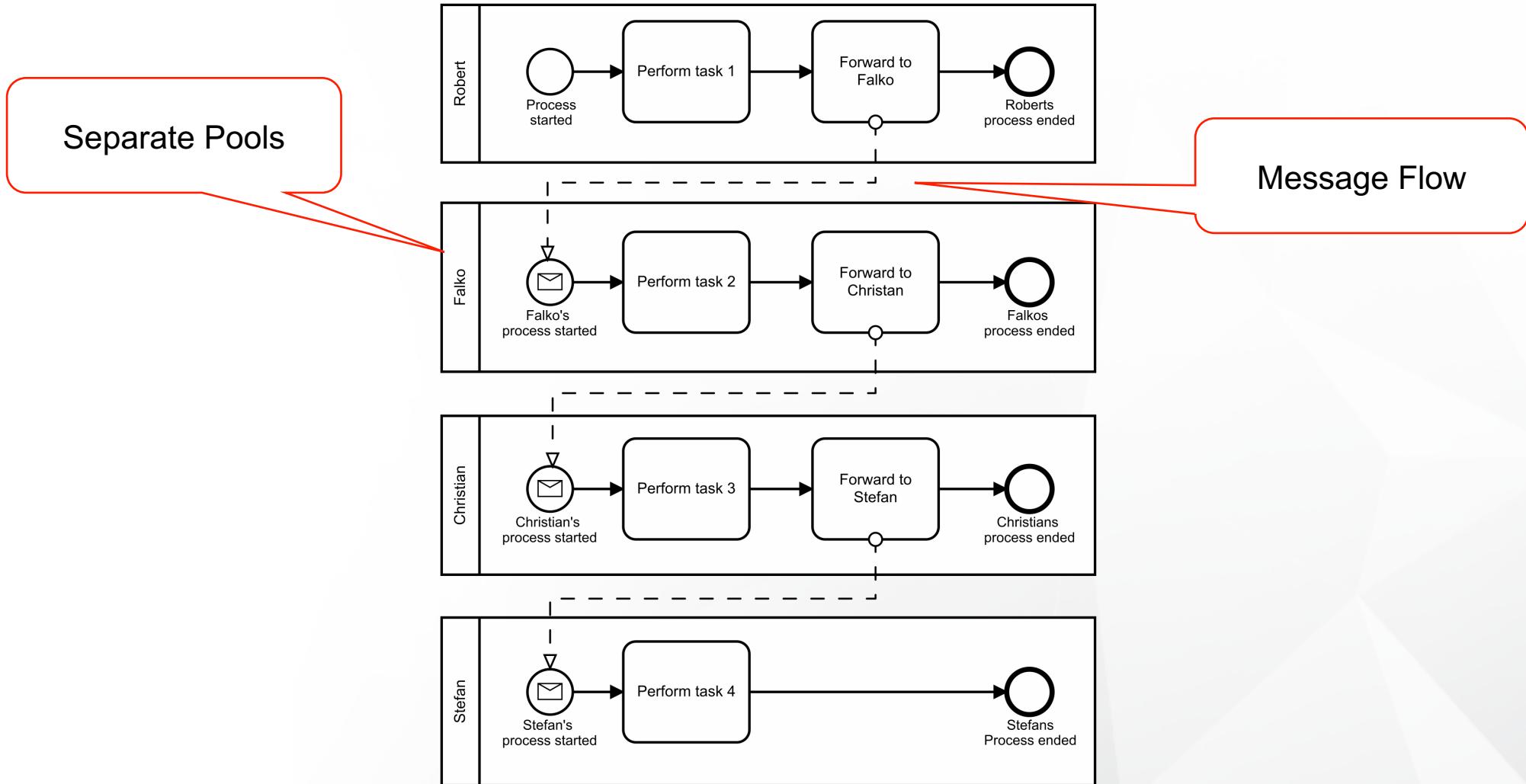
For the orchestration, the conductor controls the process



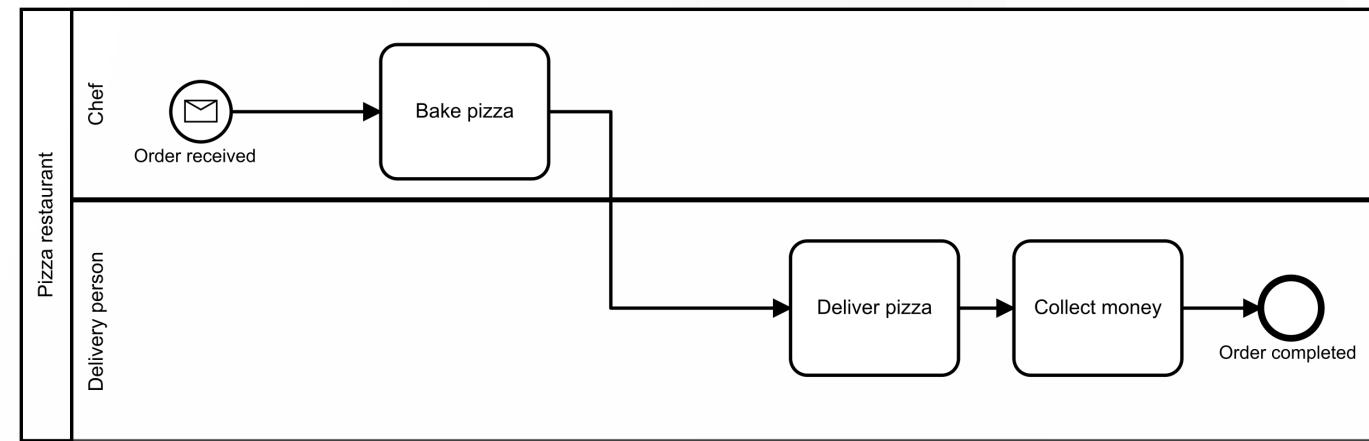
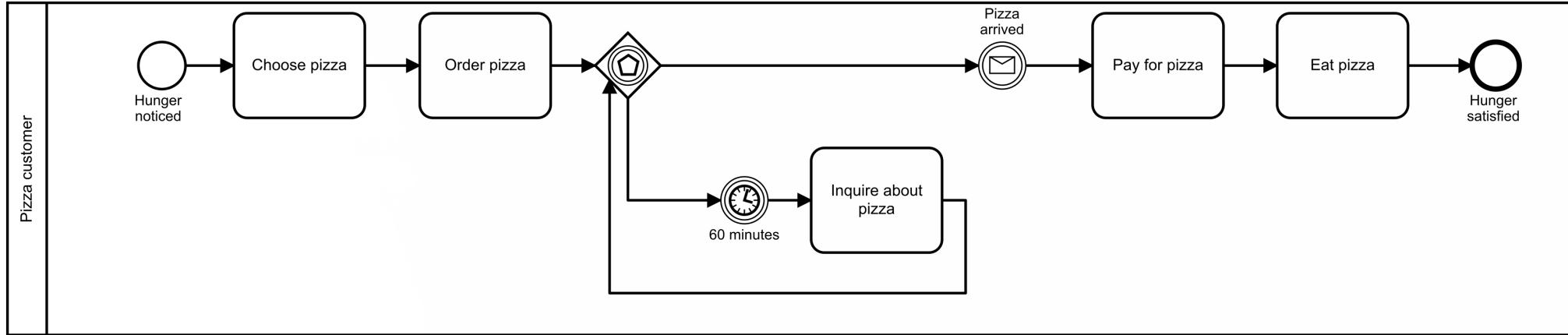
For collaboration, the participants control their processes and inform each other



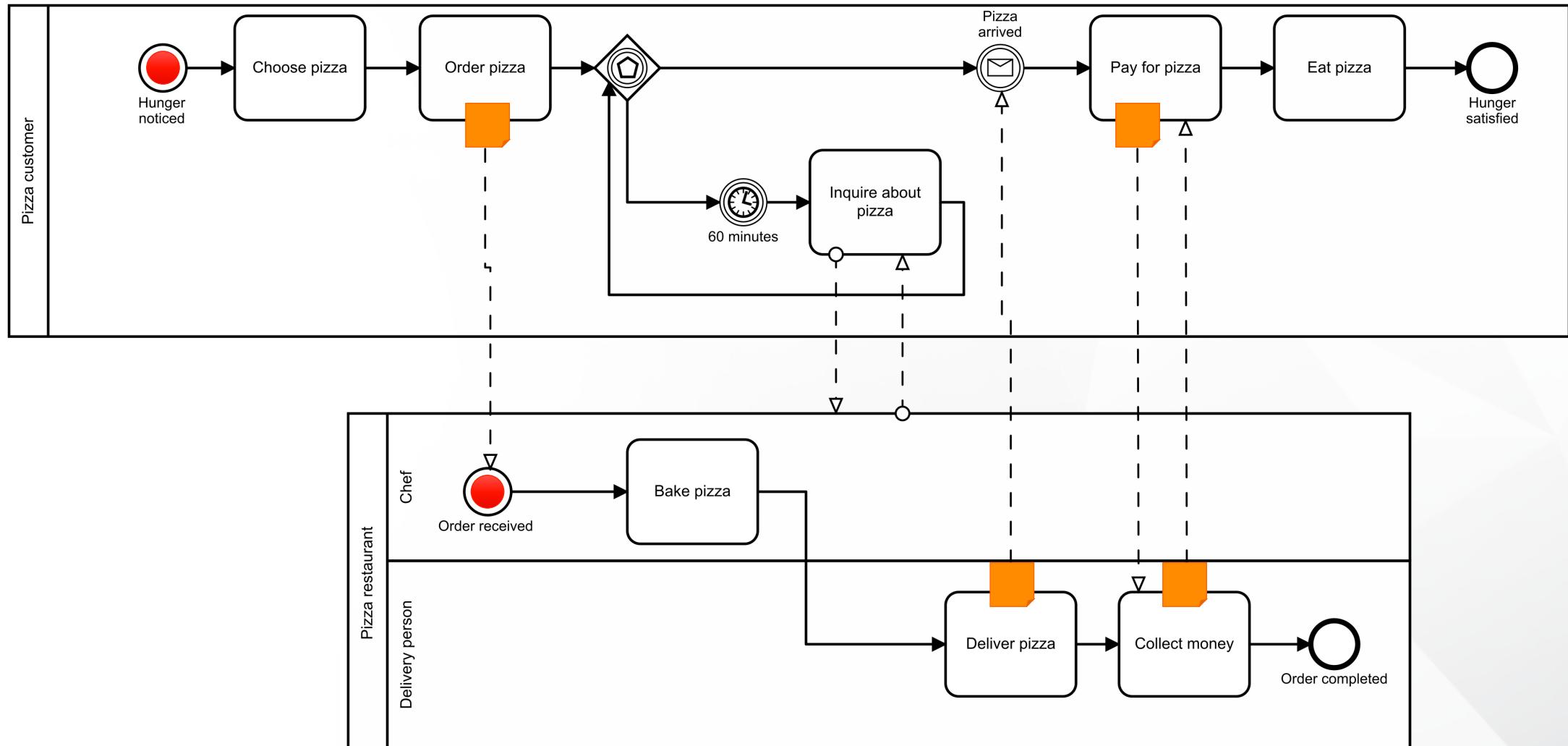
Each process participant operates in a separate pool



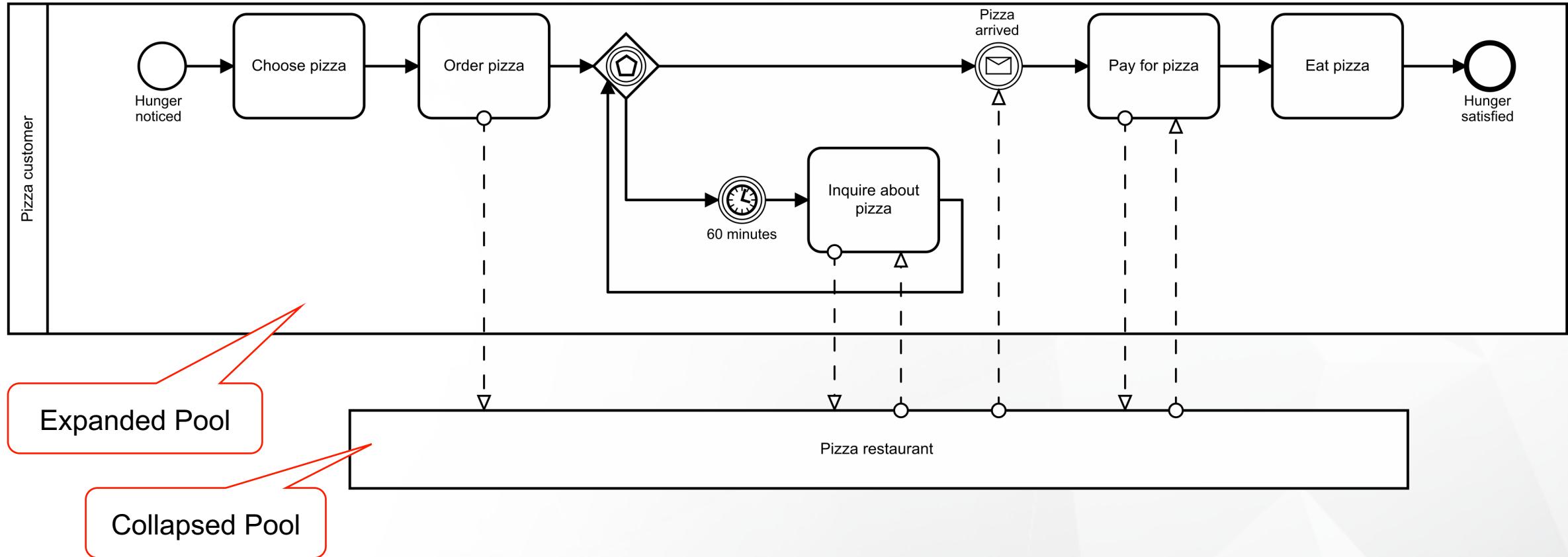
Two processes – How do they work together?



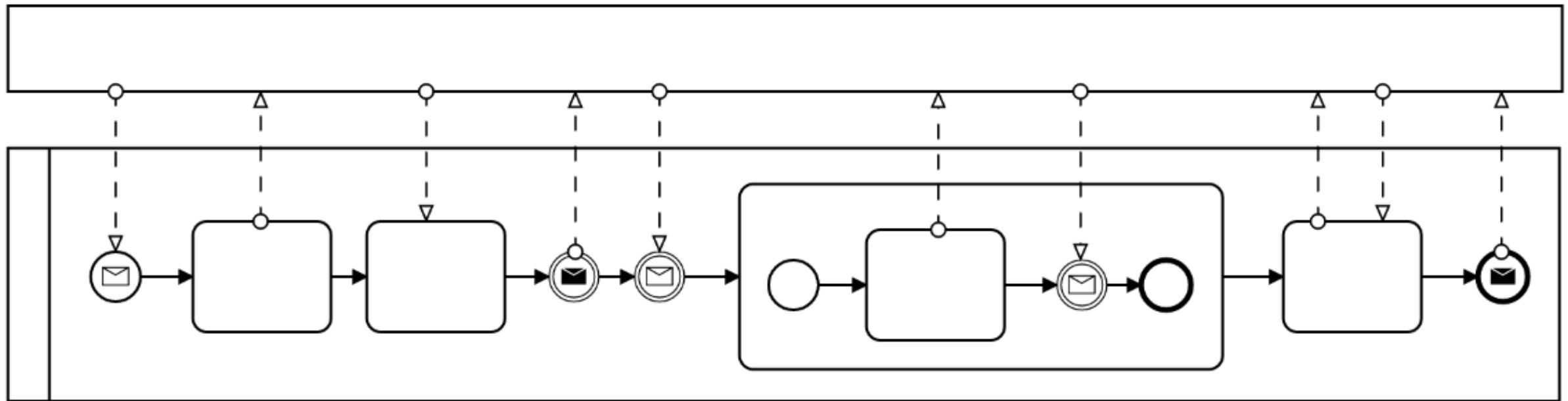
The collaboration diagram



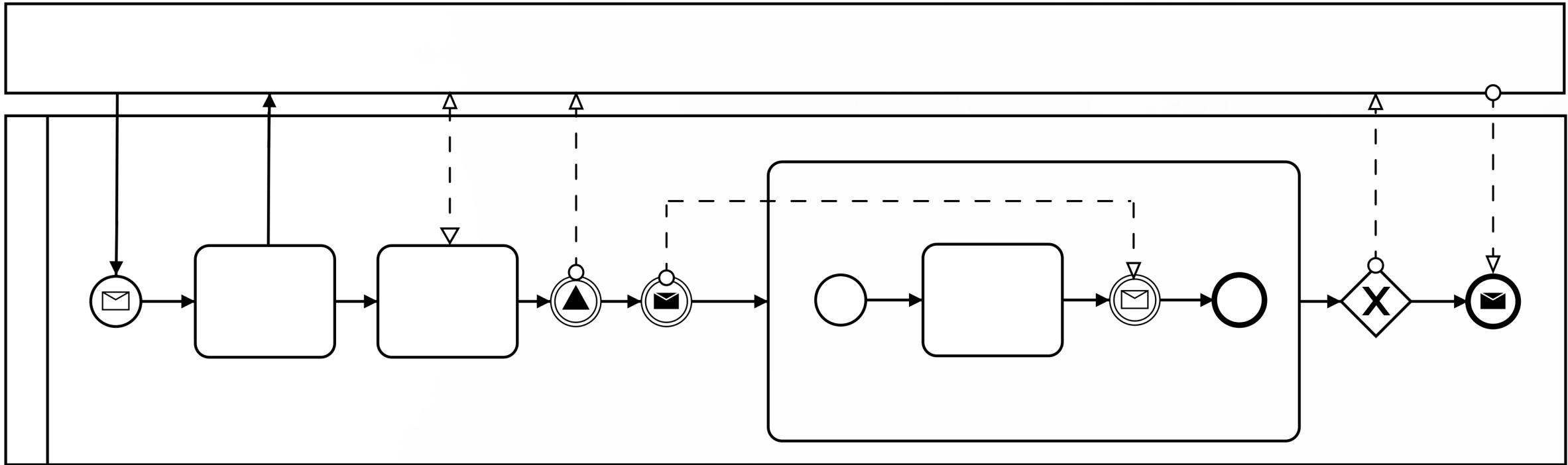
The customer's perspective



Acceptable constructs for working with pools and message flows

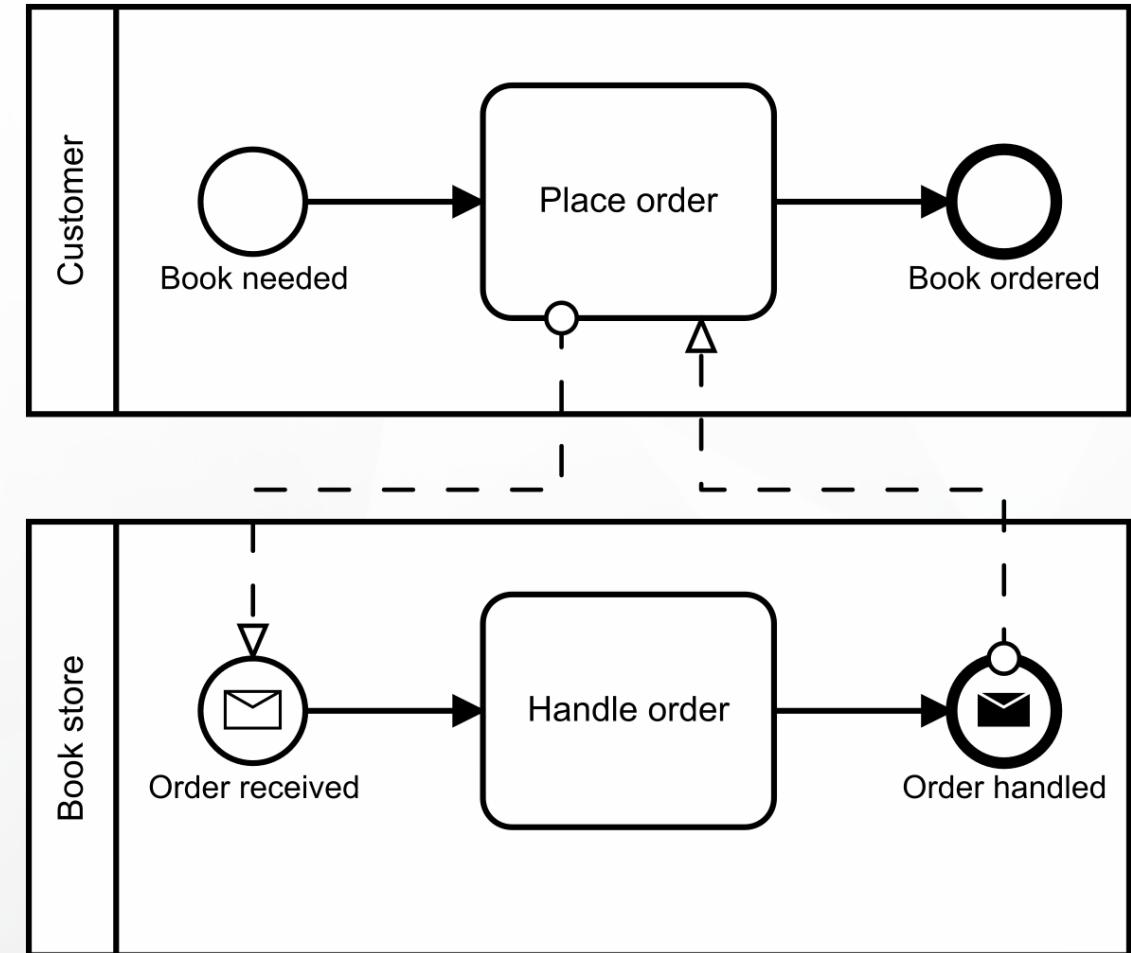


Forbidden constructs



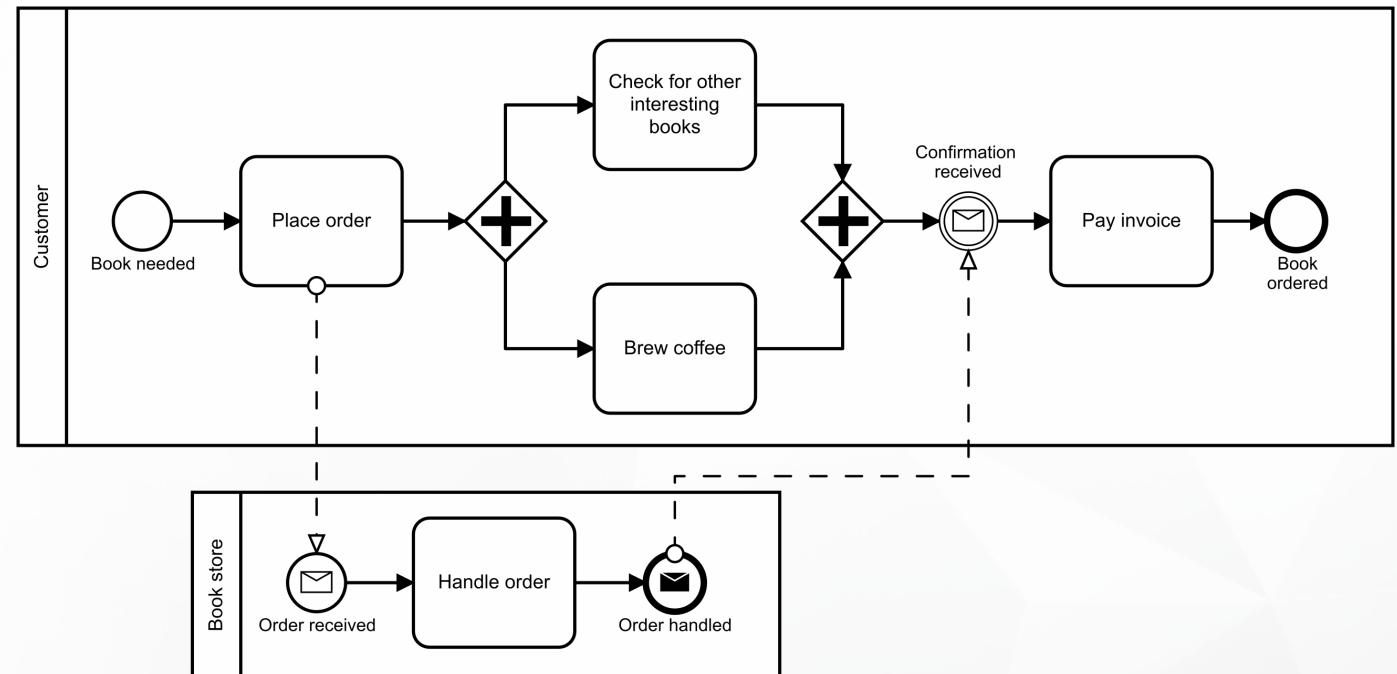
Synchronous communication

- Synchronous means sending a message and awaiting the answer in a “short” time
- The sender waits until the message arrives (blocking possible!)
- The sender cannot perform any actions between sending and receiving the messages



Asynchronous communication

- Asynchronous means
 - sending out a message
 - optionally waiting for a reply to the message
- Between sending and receiving the message further actions can be performed





Credit Scoring

Exercise

<https://training.camunda.com/bpmn/>
user: bpmn
password: camundarocks



Self Service Restaurant

Exercise

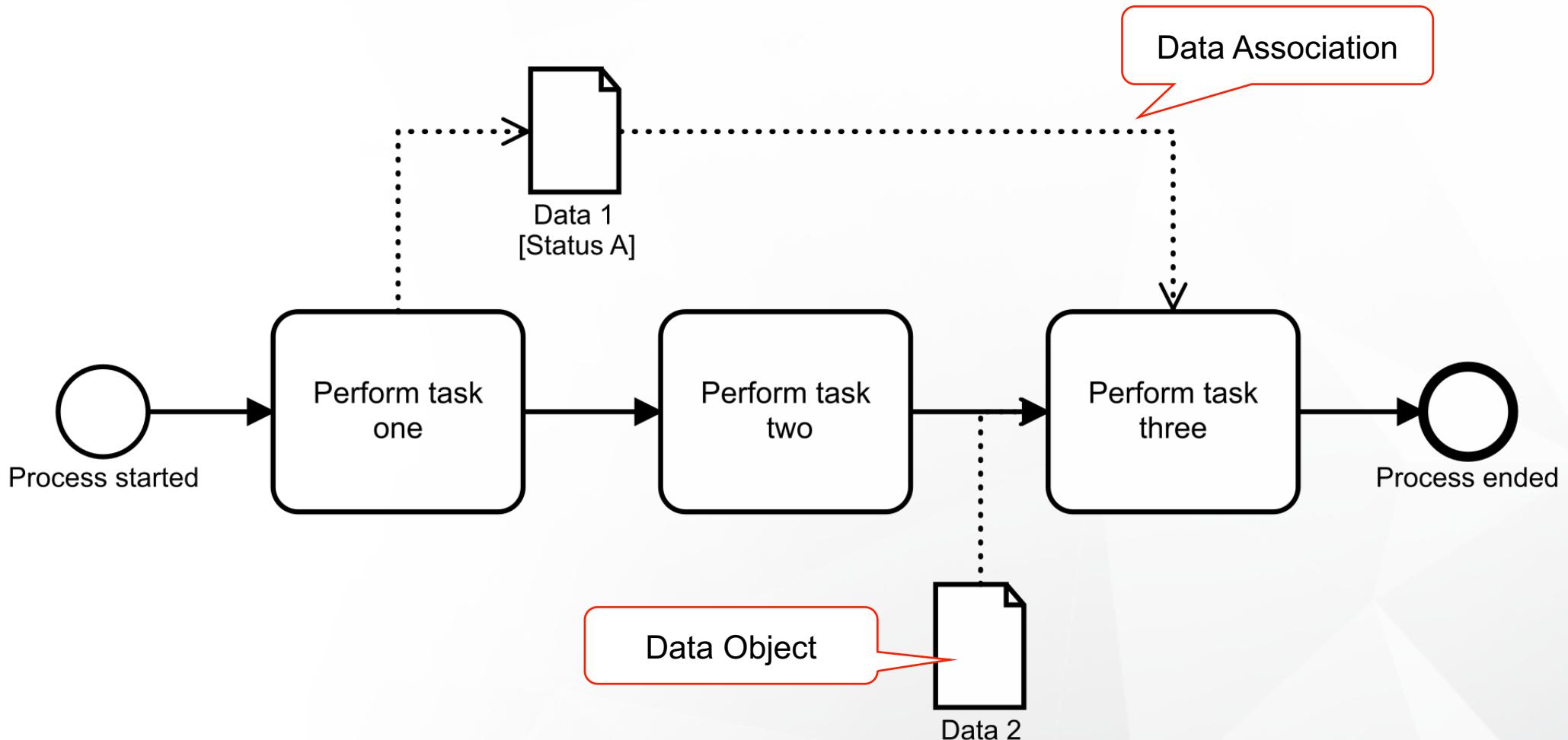
<https://training.camunda.com/bpmn/>
user: bpmn
password: camundarocks



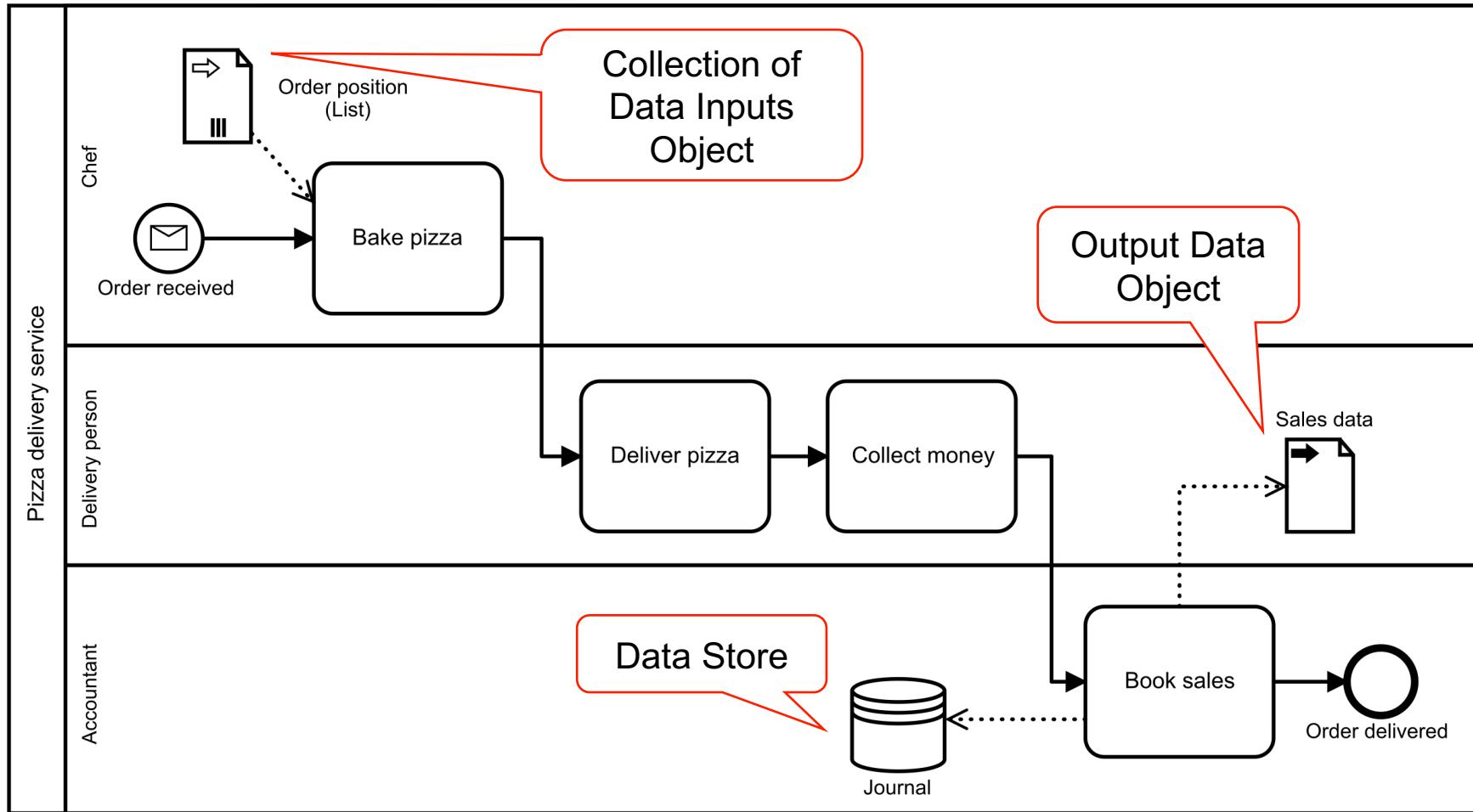
Modeling Data



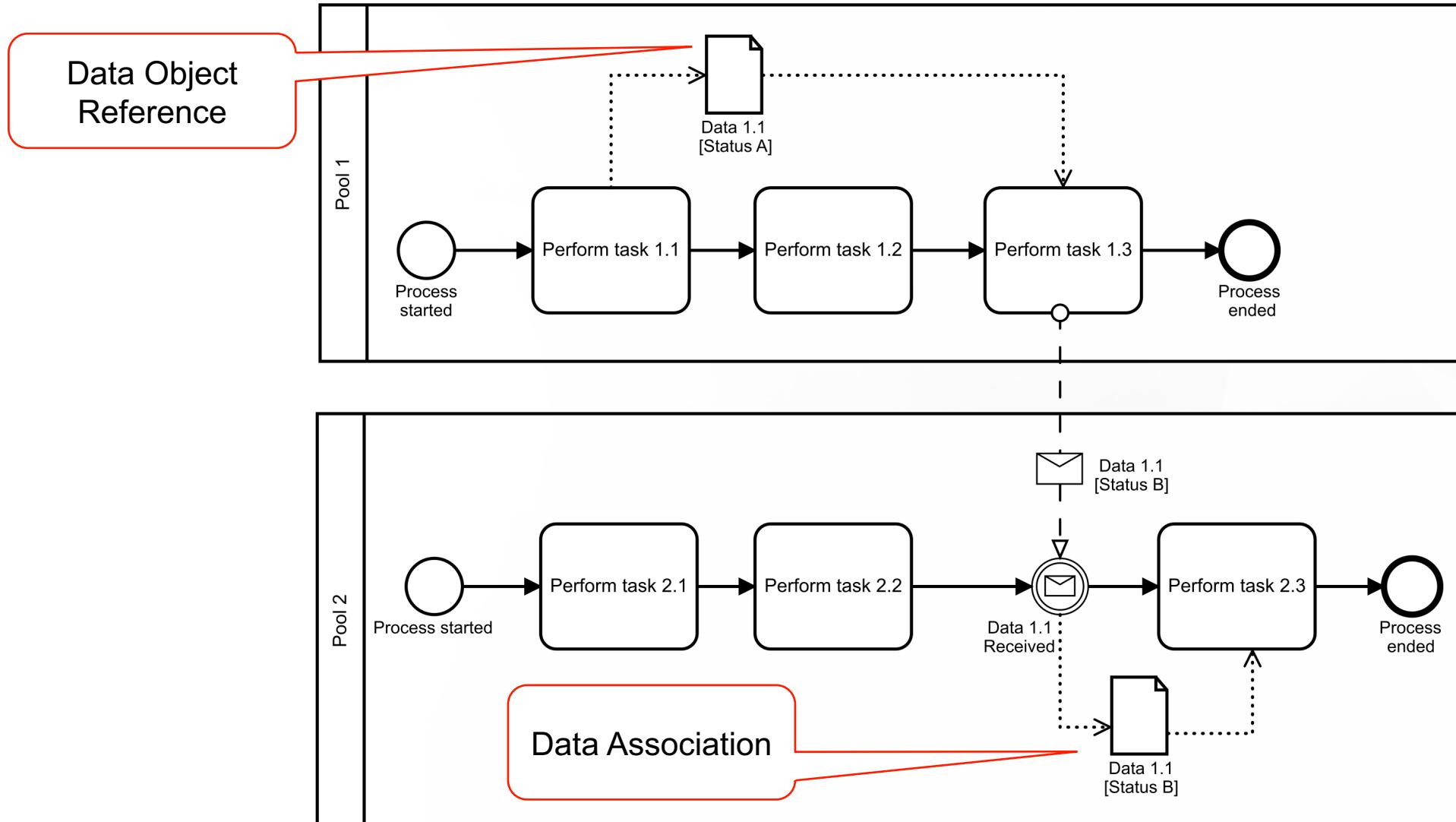
Using data objects



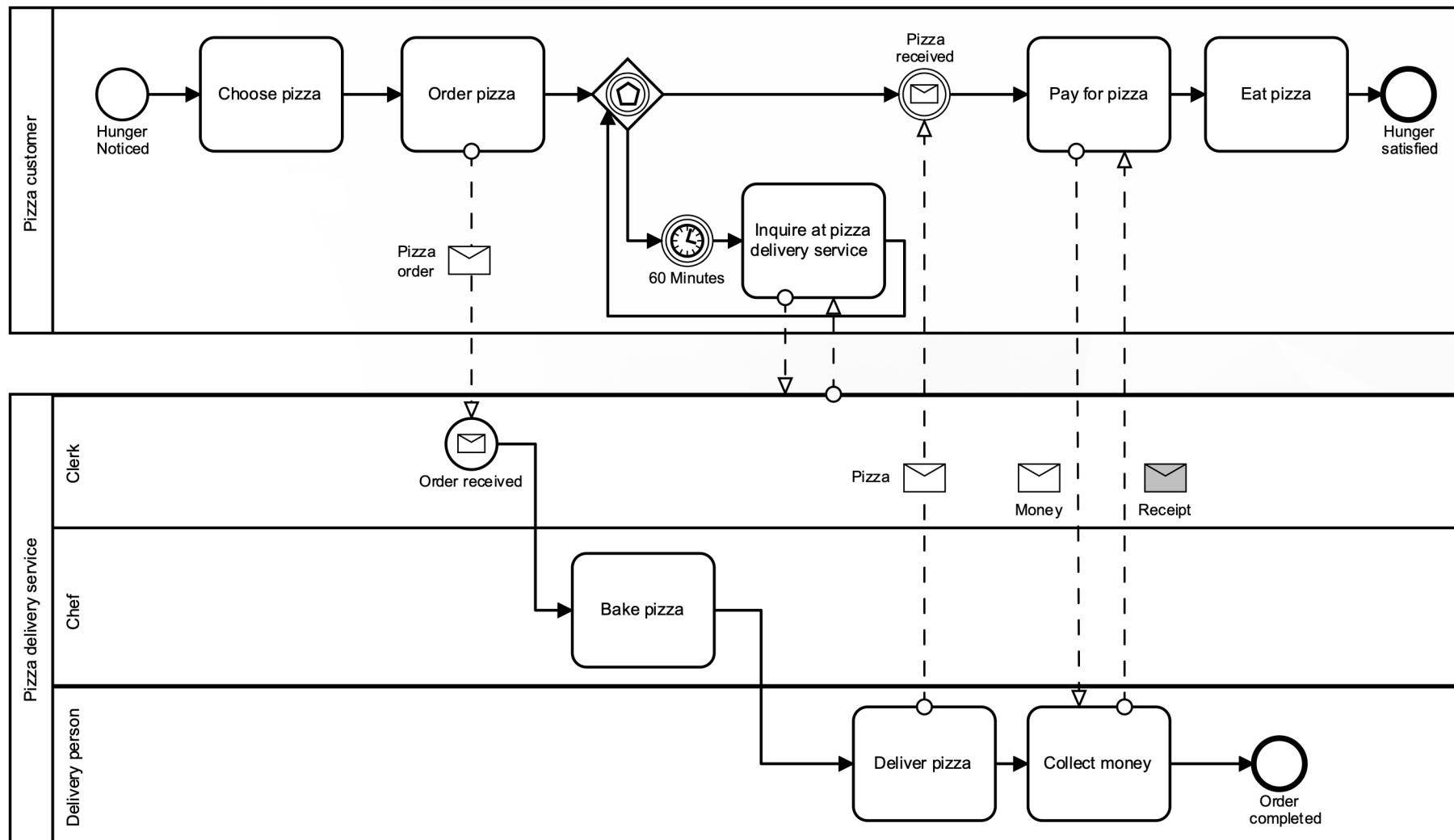
Input and output data object, collection, data store



Using data elements



Pizza process with messages





Attaching boundary events

Boundary events

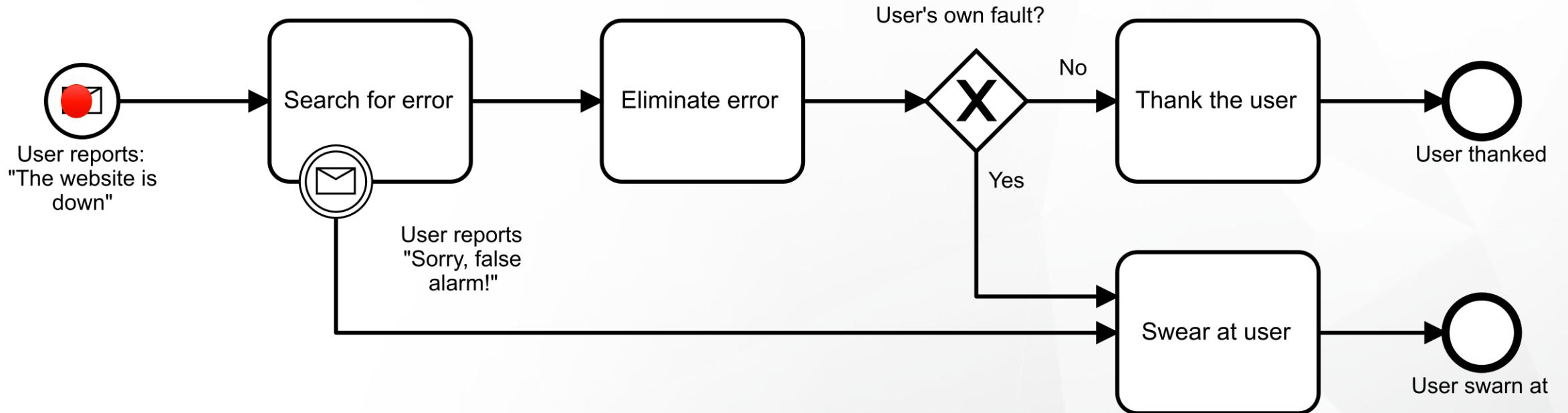
	Catching events				Intermediate events	Throwing events		
	Start events		Intermediate events			End events		
	The process is started by the engine.	The event subprocess is started, the parent process is canceled.	The event subprocess is started, the parent process is not canceled.	The process continues only, if the event occurs.		The event is reacted to, the activity is canceled.	The event is reacted to, the activity is not canceled.	
None: Untyped events; none intermediate events can mark a change of status.	(?)	(?)	(?)	(?)	(?)	(?)	(?)	
Message: Receiving and sending of messages.	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	
Timer: Cyclic timer event, points in time, or time spans.	(⌚)	(⌚)	(⌚)	(⌚)	(⌚)	(⌚)	(⌚)	
Conditional: Reacting to changed conditions and relation to business rules.	(☐)	(☐)	(☐)	(☐)	(☐)	(☐)	(☐)	
Link: Two associated link events represent a sequence flow.				(→)		(→)		
Signal: Signaling across different processes. A signal can be reacted to several times.	(△)	(△)	(△)	(△)	(△)	(△)	(△)	
Error: Triggering and treatment of defined errors		(₩)			(₩)	(₩)	(₩)	
Escalation: Reporting to the next higher level of responsibility.	(Ⓐ)	(Ⓐ)	(Ⓐ)		(Ⓐ)	(Ⓐ)	(Ⓐ)	
Terminate: Triggers the immediate termination of the process.							(●)	
Compensation: Handling or triggering of a compensation.		(◀)			(◀)		(◀)	
Cancel: Reaction to canceled transactions or triggering of cancellations.					(✖)		(✖)	
Multiple: Occurrence of one of several events; triggering of all events.	(pentagon)	(pentagon)	(pentagon)	(pentagon)	(pentagon)	(pentagon)	(pentagon)	
Parallel multiple: Occurrence of all events.	(⊕)	(⊕)	(⊕)	(⊕)	(⊕)	(⊕)	(⊕)	

NEW in BPMN 2.0

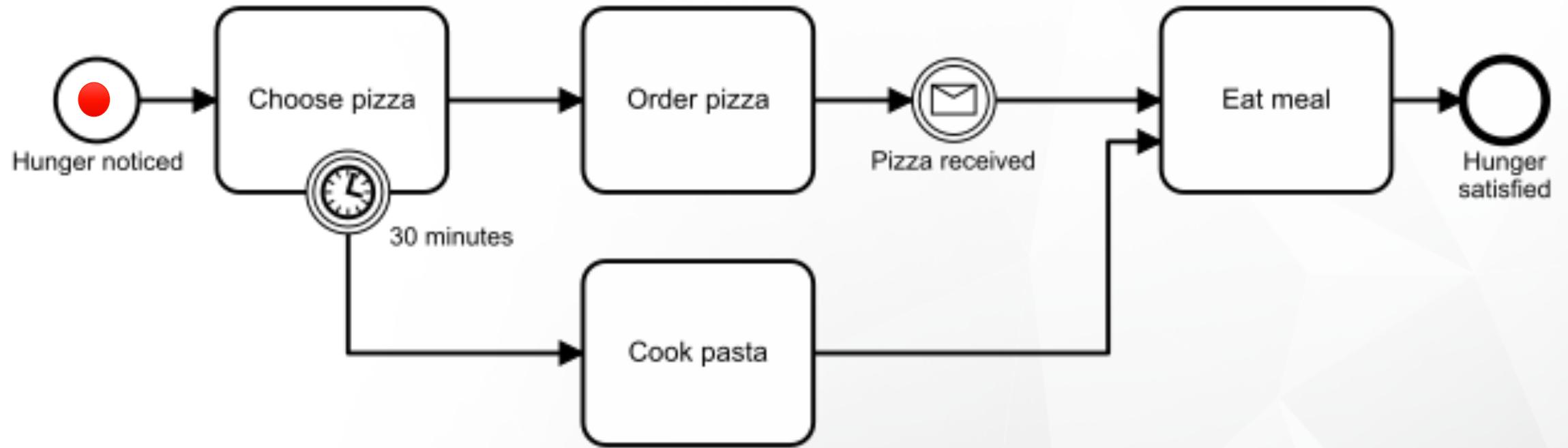
Symbols of BPMN (following the BPMN poster of the Berlin BPM offensive: www.bpmn.de/poster)
Taken from "Praxishandbuch BPMN" by Jakob Freund and Bernd Rücker, © 2010 Carl Hanser Verlag Munich

HANSER  camunda
the business process company

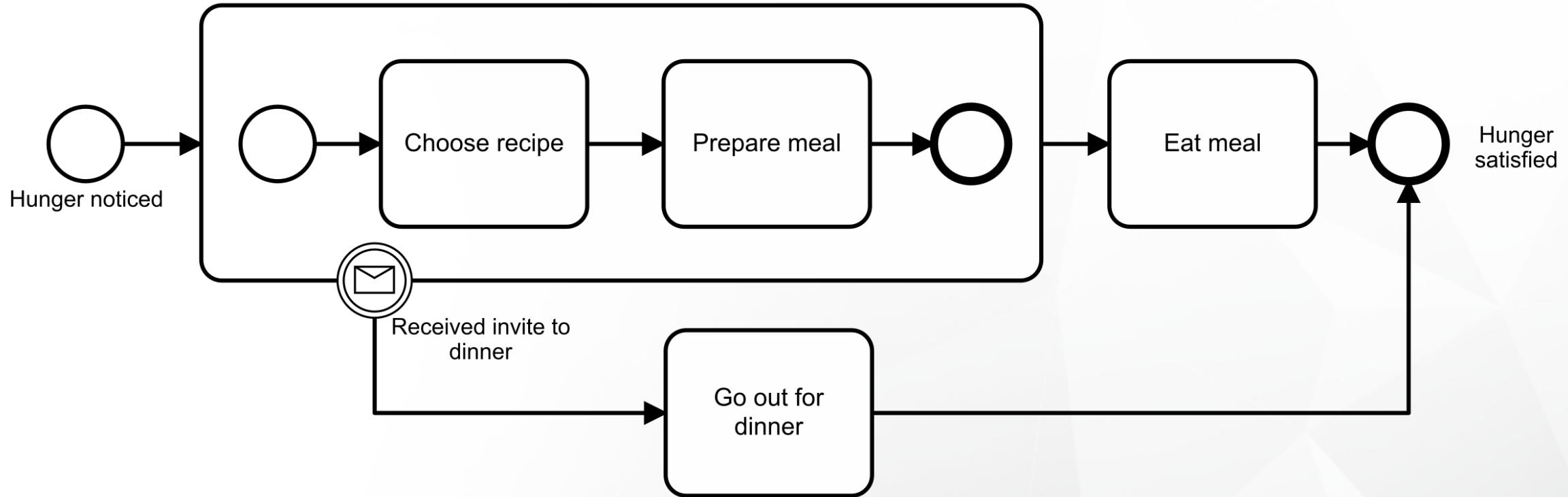
The attached message event results in the cancellation of the task „Search for error“



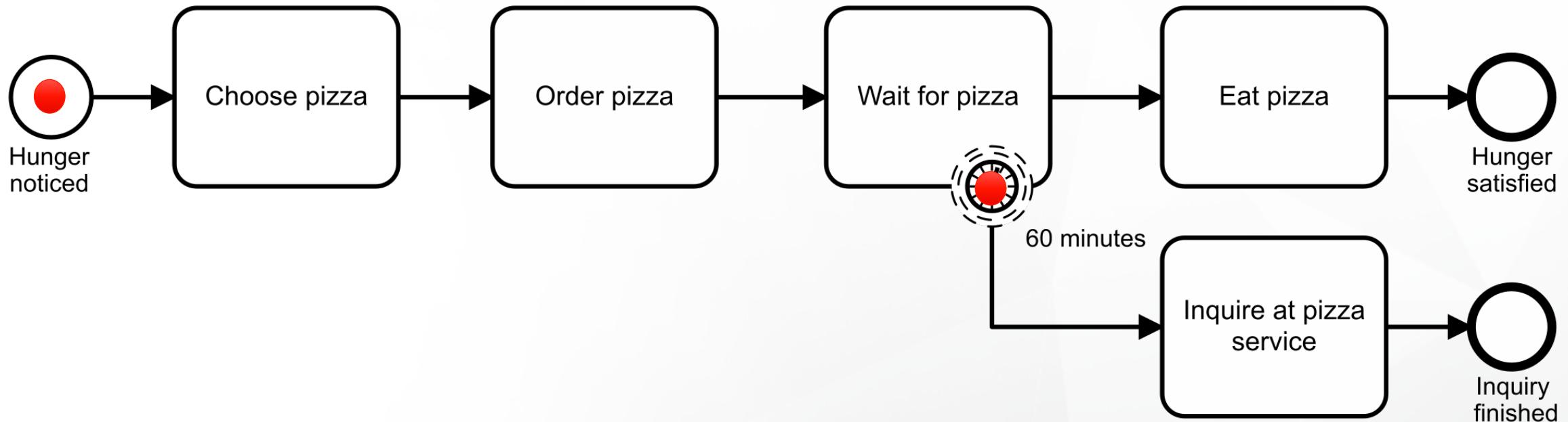
Interrupting boundary events



The catching event cancels the entire subprocess

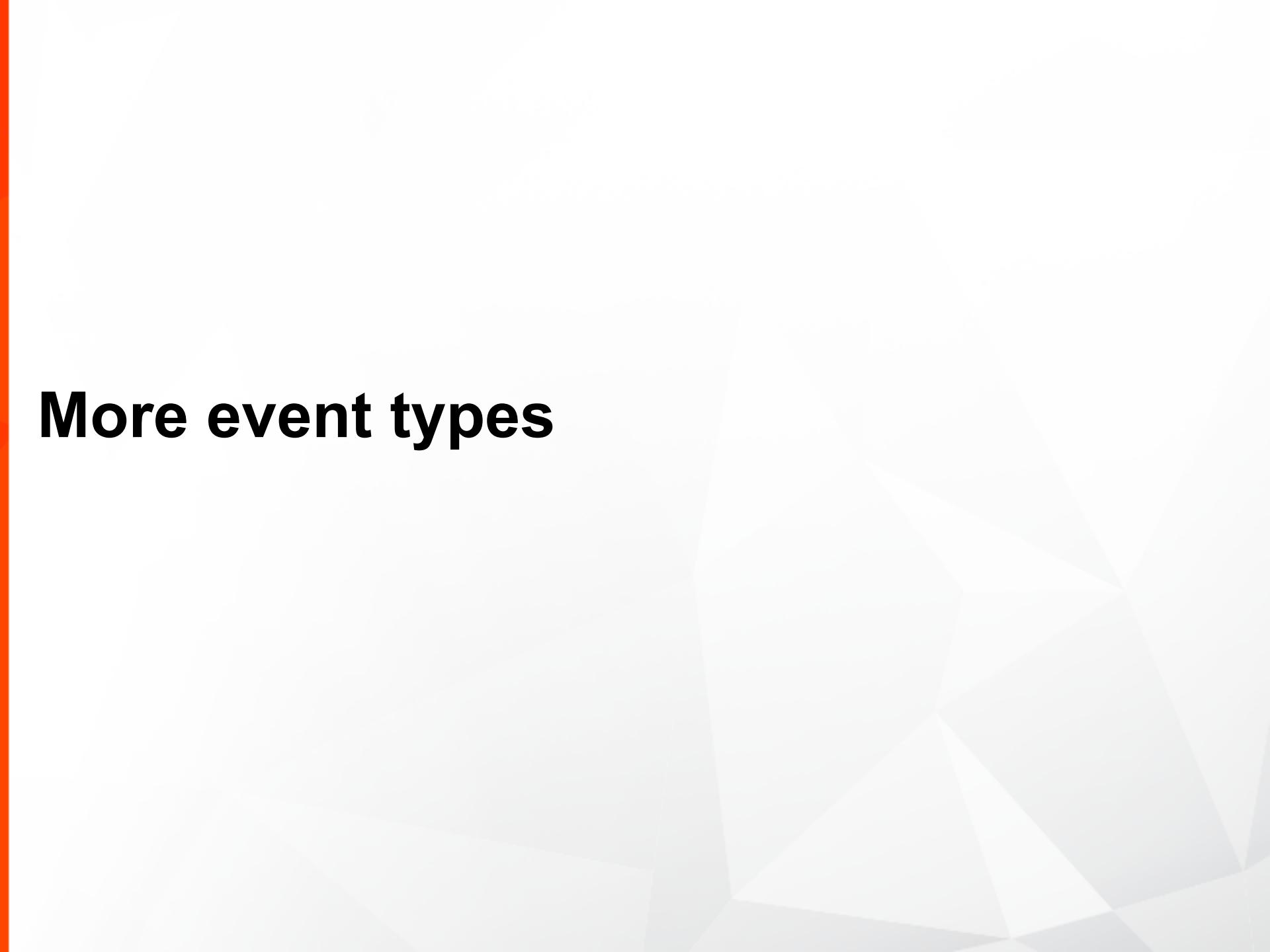


Non-interrupting boundary events





More event types



More event types

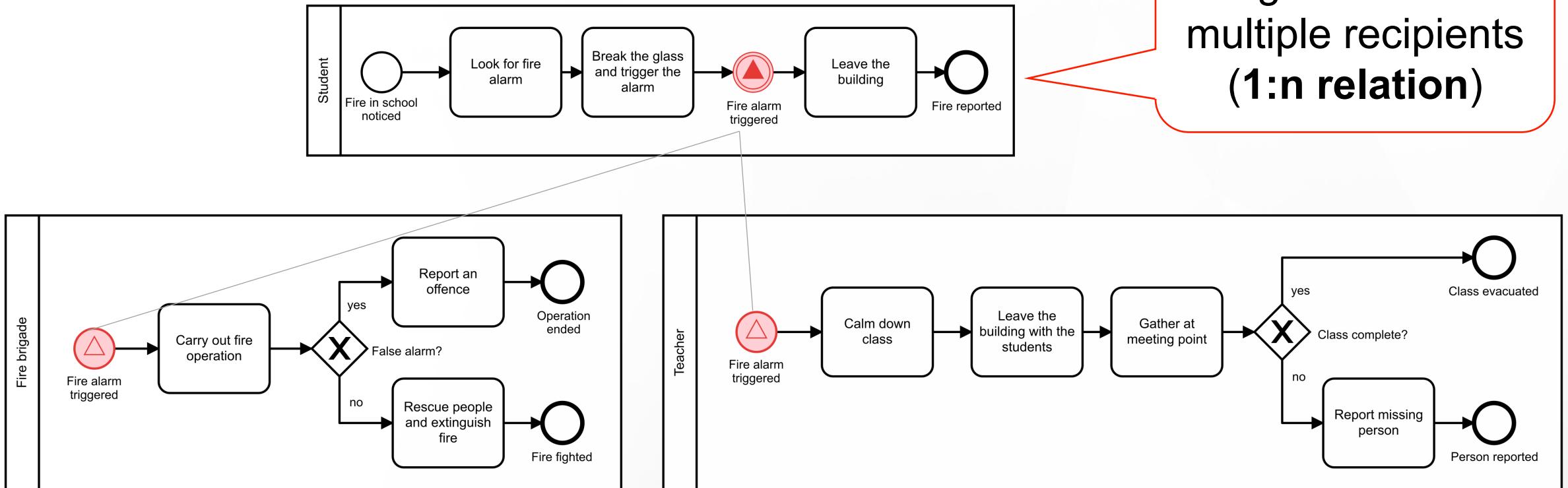
	Catching events								Throwing events							
	Start events				Intermediate events				End events							
	The process is started by the engine.	The event subprocess is started, the parent process canceled.	The event subprocess is started, the parent process is not canceled.	The process continues only, if the event occurs.	The event is reacted to, the activity is canceled.	The event is reacted to, the activity is not canceled.	The process triggers the event and continues immediately.	The process triggers the event at the end of the process path.								
None: Untyped events; none intermediate events can mark a change of status.	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)
Message: Receiving and sending of messages.	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)
Timer: Cyclic timer event, points in time, or time spans.	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)
Conditional: Reacting to changed conditions and relation to business rules.	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)
Link: Two associated link events represent a sequence flow.				(➡)				(➡)								
Signal: Signaling across different processes. A signal can be reacted to several times.	(△)	(△)	(△)	(△)	(△)	(△)	(△)	(△)	(△)	(△)	(△)	(△)	(△)	(△)	(△)	(△)
Error: Triggering and treatment of defined errors			(ℳ)		(ℳ)			(ℳ)								(ℳ)
Escalation: Reporting to the next higher level of responsibility.			(Ⓐ)		(Ⓐ)			(Ⓐ)		(Ⓐ)		(Ⓐ)		(Ⓐ)		(Ⓐ)
Terminate: Triggers the immediate termination of the process.																(●)
Compensation: Handling or triggering of a compensation.			(◀)					(◀)								(◀)
Cancel: Reaction to canceled transactions or triggering of cancellations.								(✗)								(✗)
Multiple: Occurrence of one of several events; triggering of all events.	(○)	(○)	(○)	(○)	(○)	(○)	(○)	(○)	(○)	(○)	(○)	(○)	(○)	(○)	(○)	(○)
Parallel multiple: Occurrence of all events.	(+)	(+)	(+)	(+)	(+)	(+)	(+)	(+)	(+)	(+)	(+)	(+)	(+)	(+)	(+)	(+)

NEW in BPMN 2.0

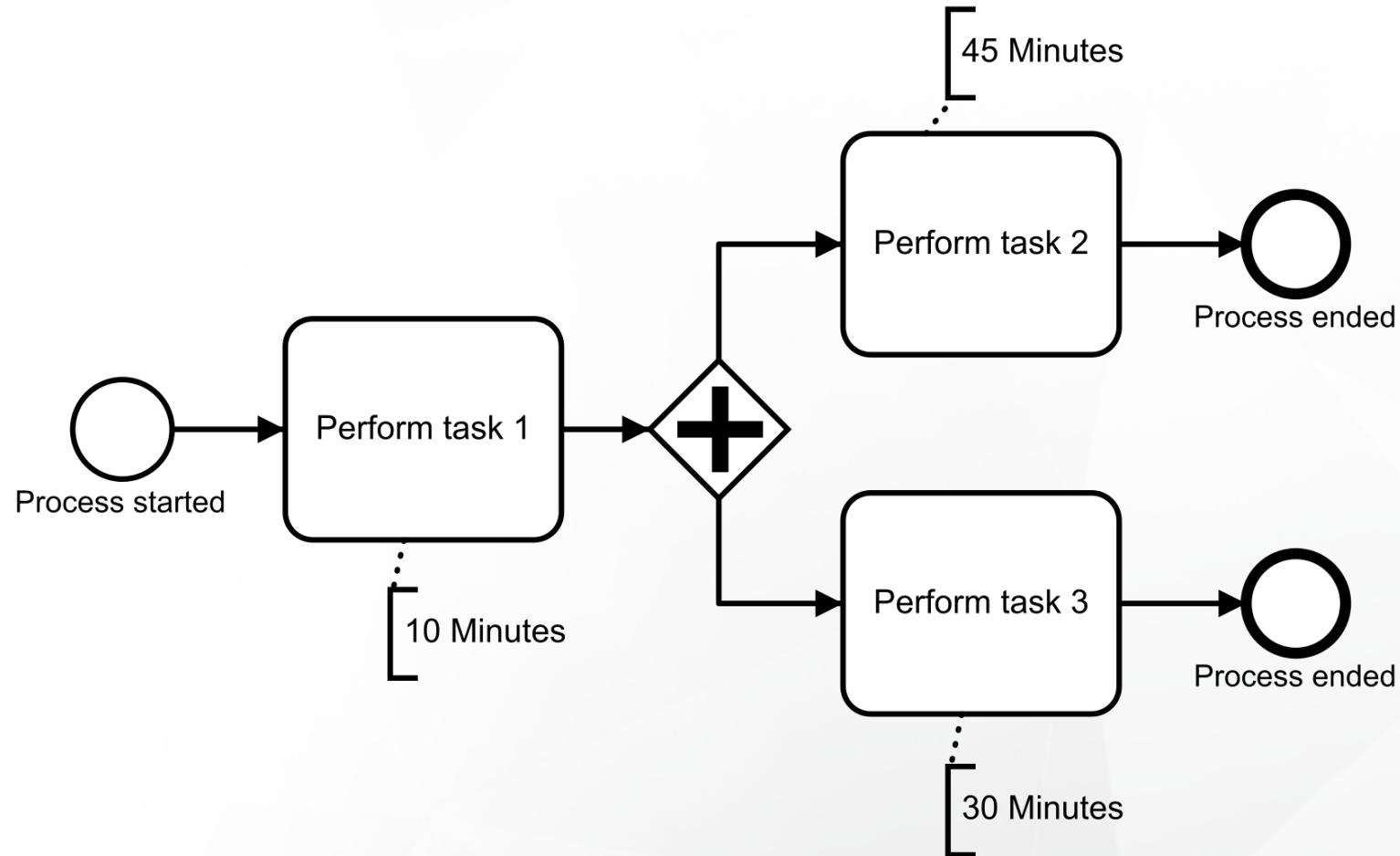
Symbols of BPMN (following the BPMN poster of the Berlin BPM offensive: www.bpmn.de/poster)
Taken from "Praxishandbuch BPMN" by Jakob Freund and Bernd Rücker; © 2010 Carl Hanser Verlag Munich

Signal events

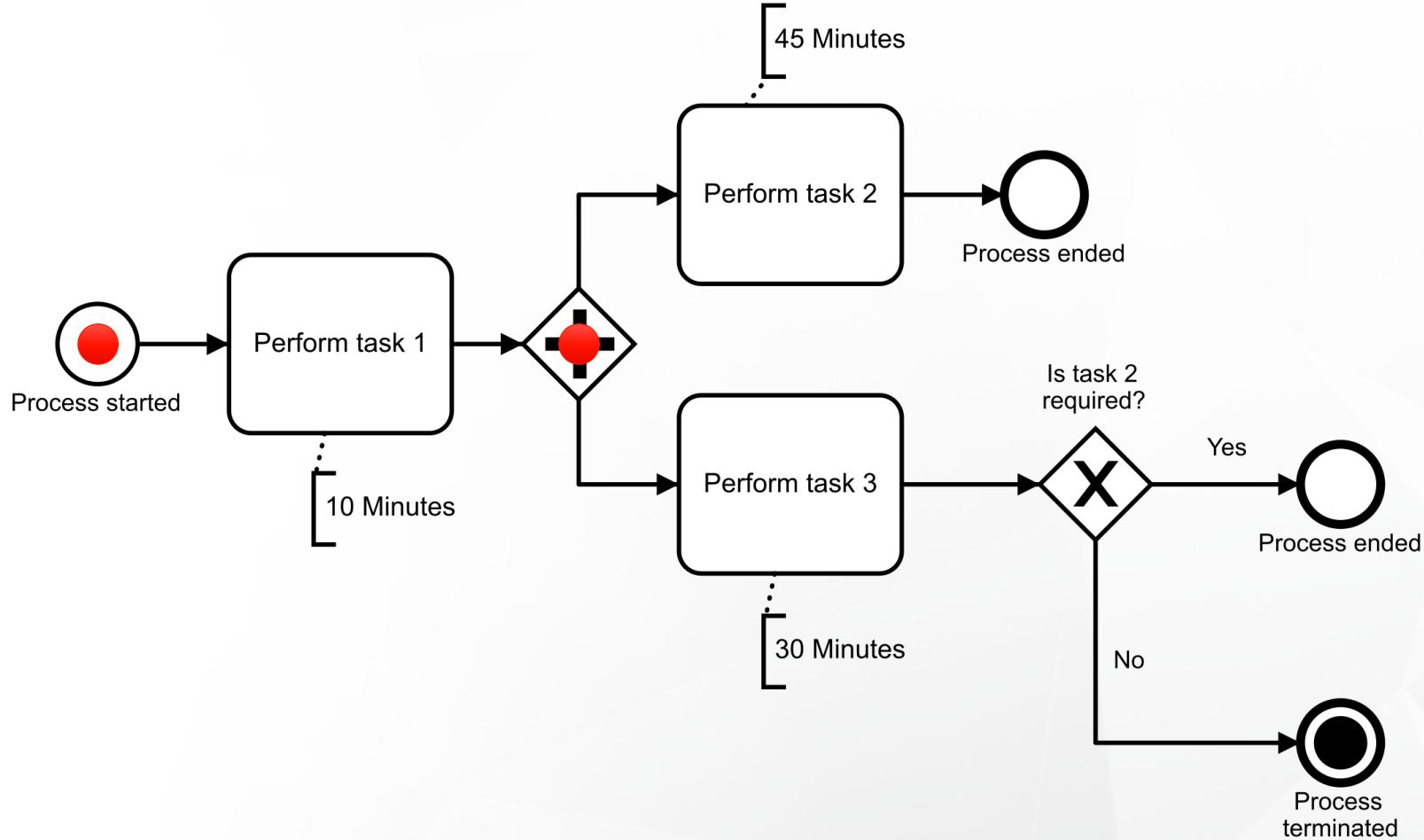
A signal can have multiple recipients
(1:n relation)



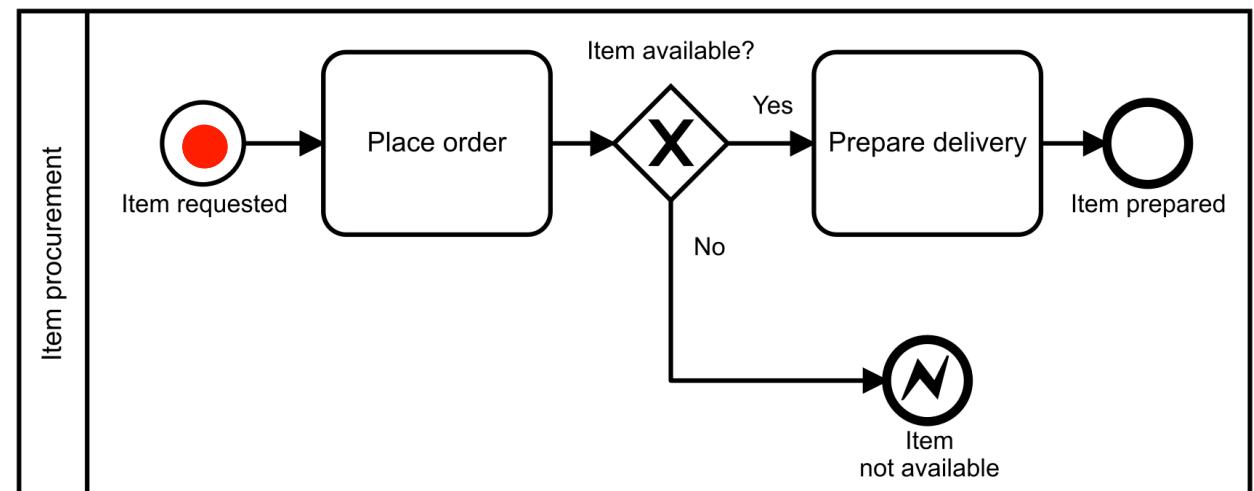
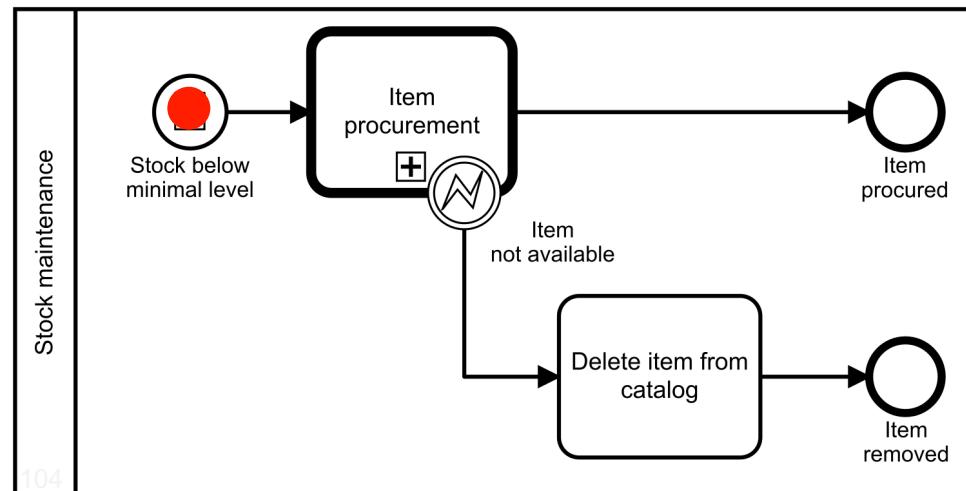
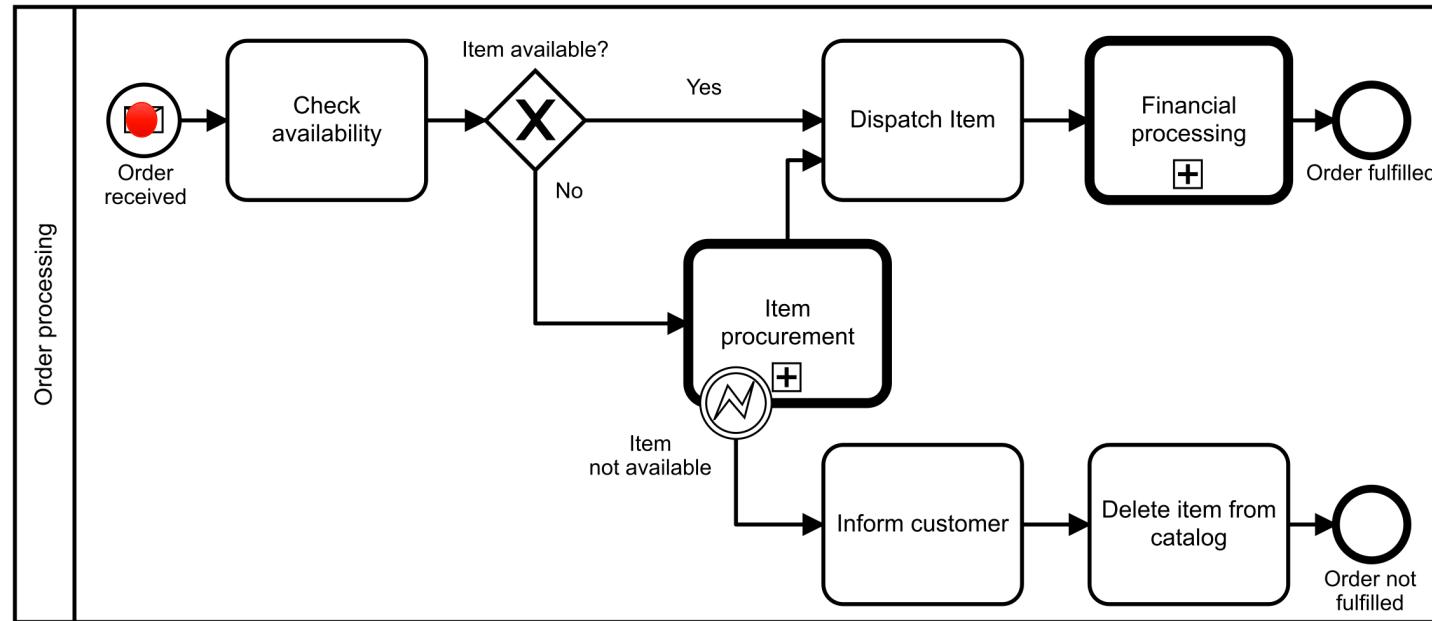
The process always ends after 55 minutes



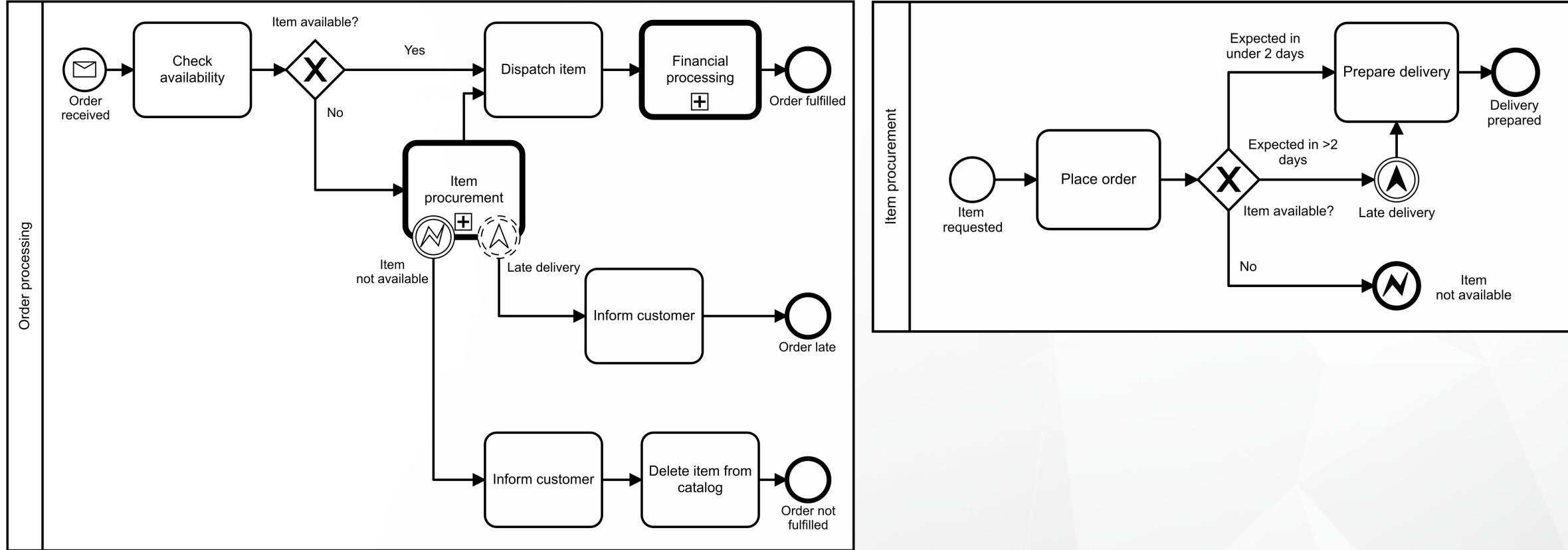
The process may be terminated after task 3



The subprocess reports an error to its parent



The difference between error and escalation



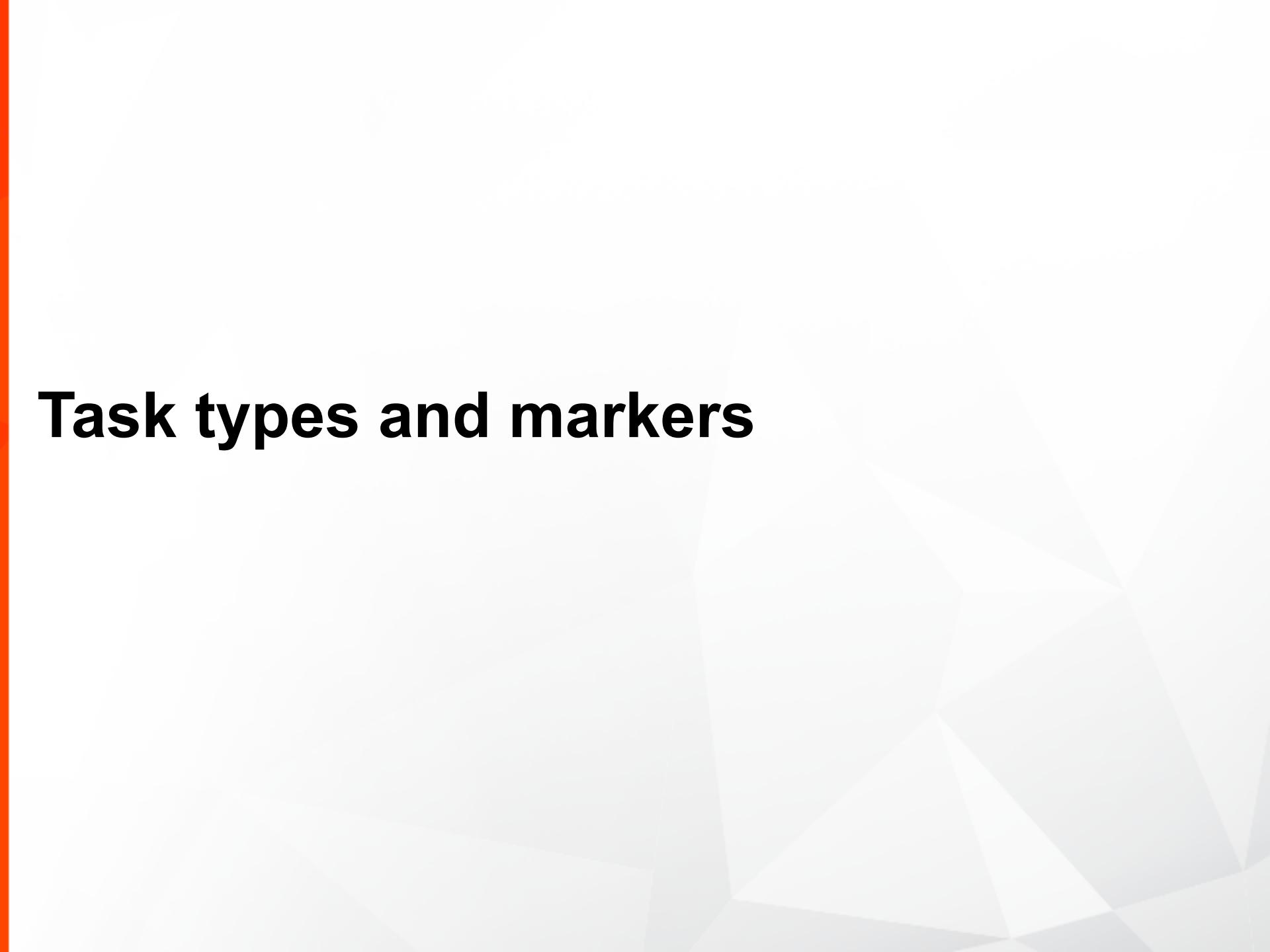
Handling problems

Exercise

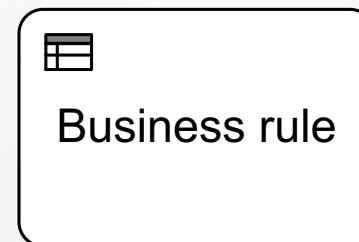
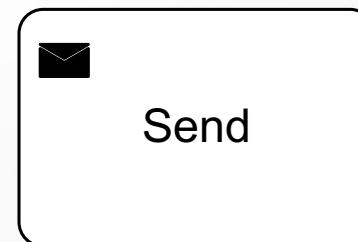
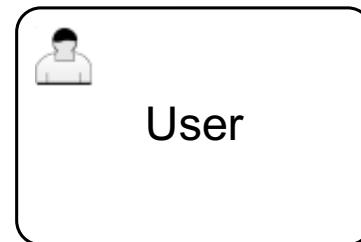
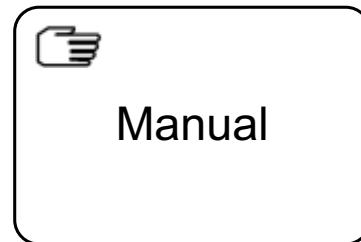
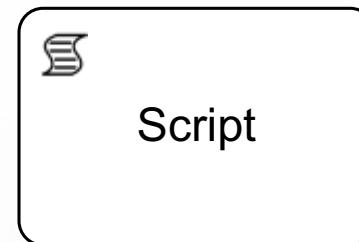
<https://training.camunda.com/bpmn/>
user: bpmn
password: camundarocks



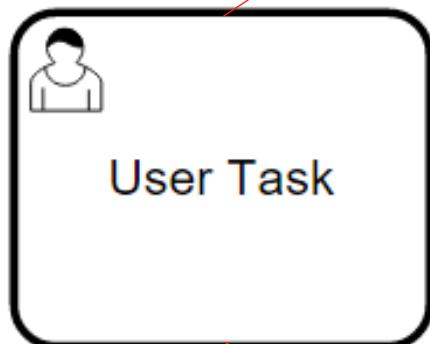
Task types and markers



Task types



User Task



Form History Diagram Description

Please clarify: This invoice needs clarification and could not be approved due to missing data.

Invoice Document

Creditor

Papa Steve's all you can eat

Amount

10.99

Invoice Category

Travel Expenses

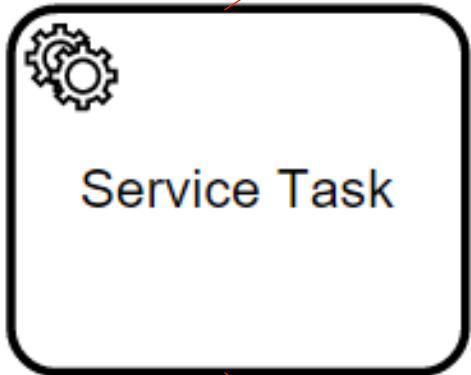
Invoice Number

PSACE-5342

Could you clarify the invoice?

Save **Complete**

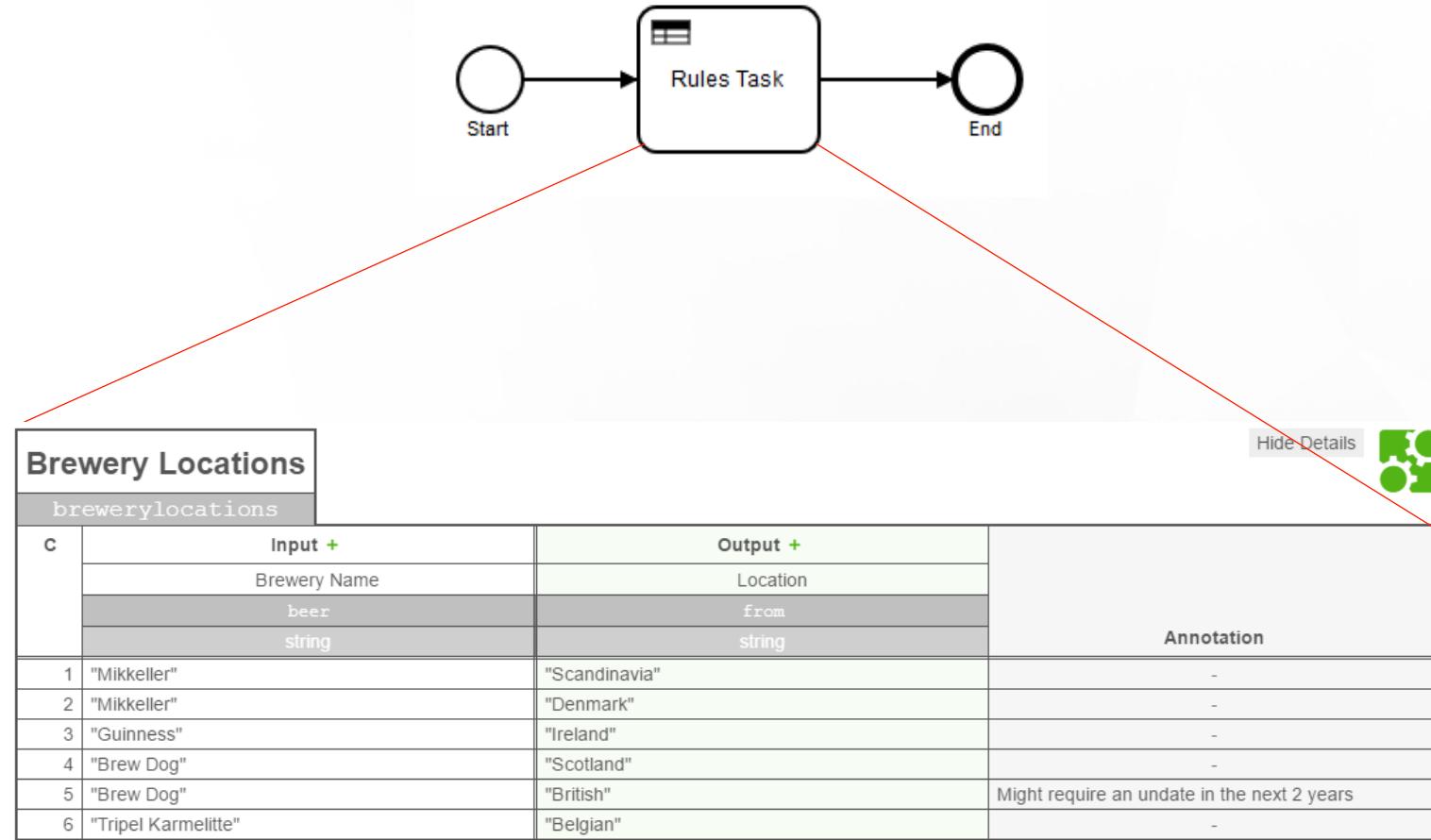
Service Task



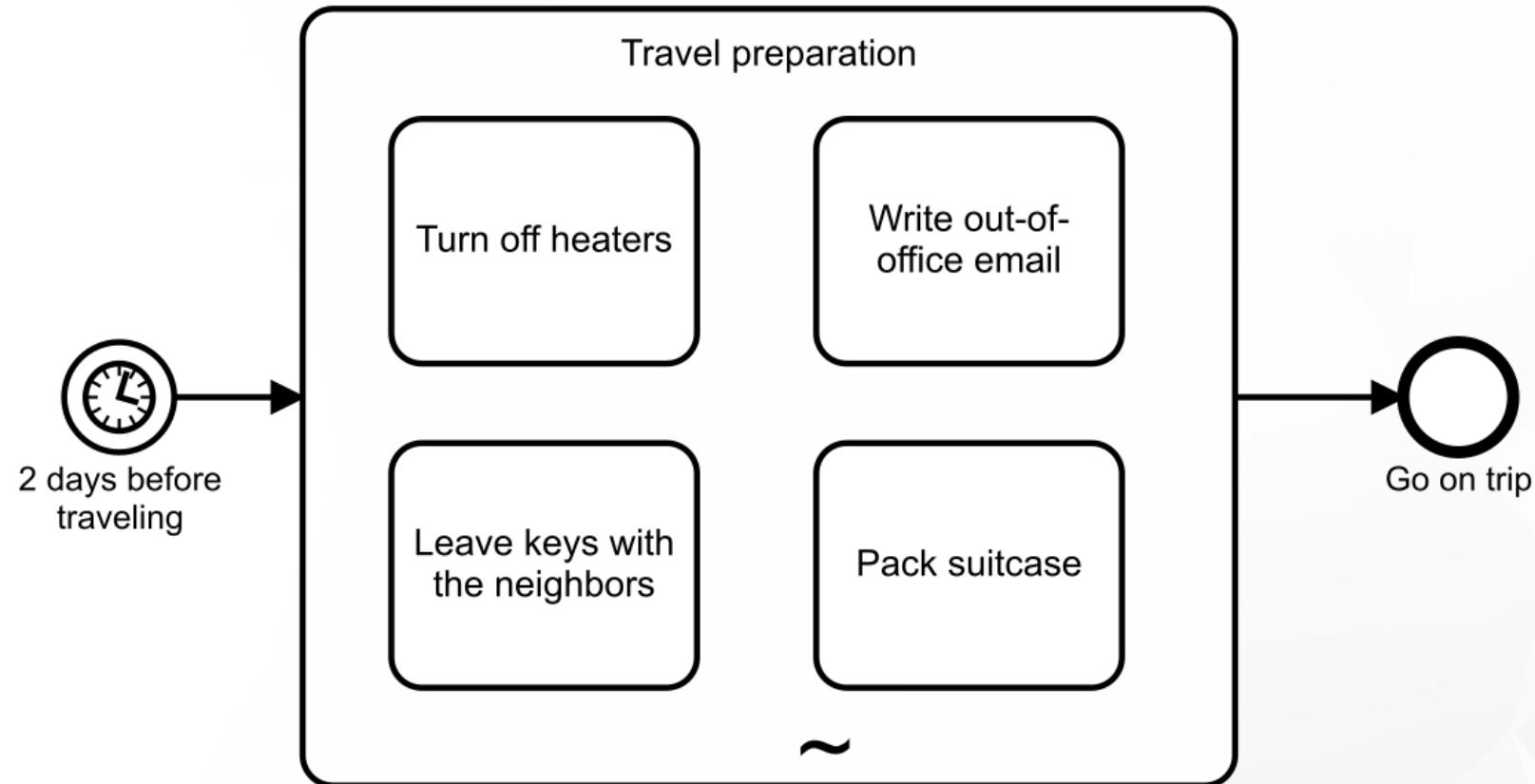
Service Task

```
public class CalculateMutualisabilityDelegate implements JavaDelegate {  
  
    @Override  
    public void execute(DelegateExecution execution) throws Exception {  
        boolean isItMutual = random();  
  
        if(isItMutual)  
        {  
            execution.setVariable("isMutual", true);  
        }else{  
            execution.setVariable("isMutual", false);  
        }  
        String missionOrder = "This is the Default 'mission order' i suspect "  
            + "there is more to it than this - but i don't actually "  
            + "know for sure.";  
        execution.setVariable("missionOrder", missionOrder);  
    }  
}
```

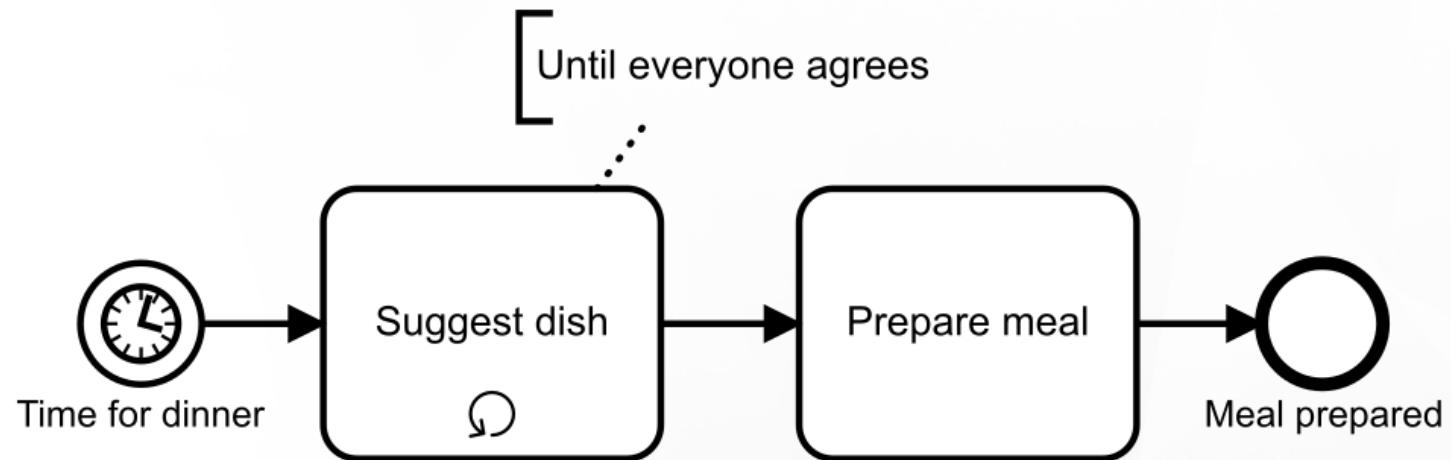
Business Rule Task



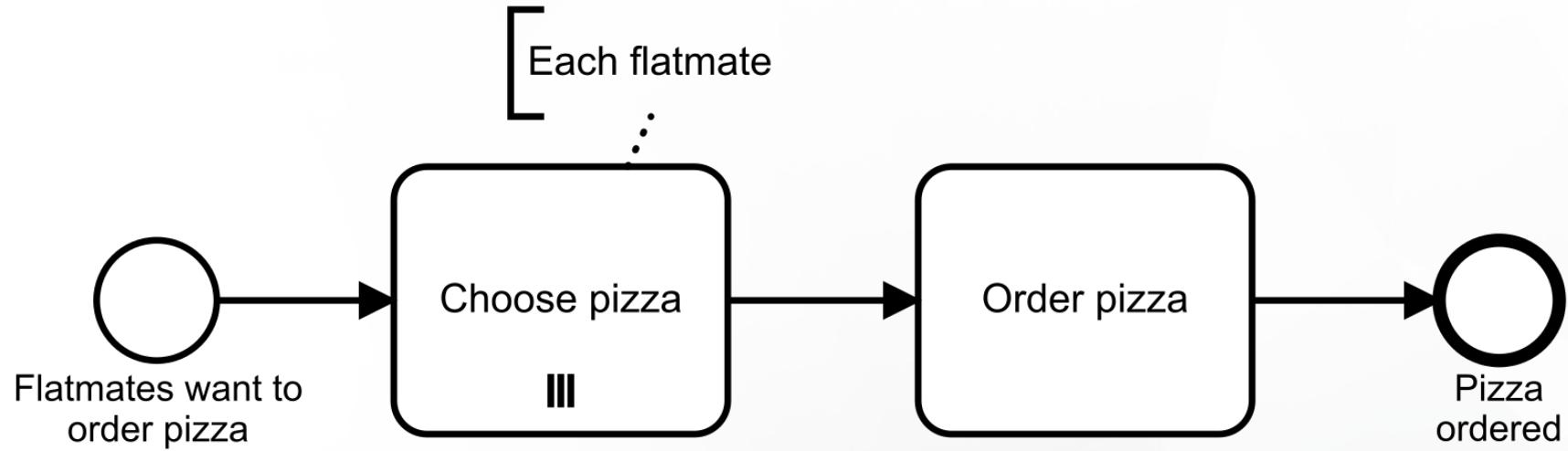
Adhoc subprocess



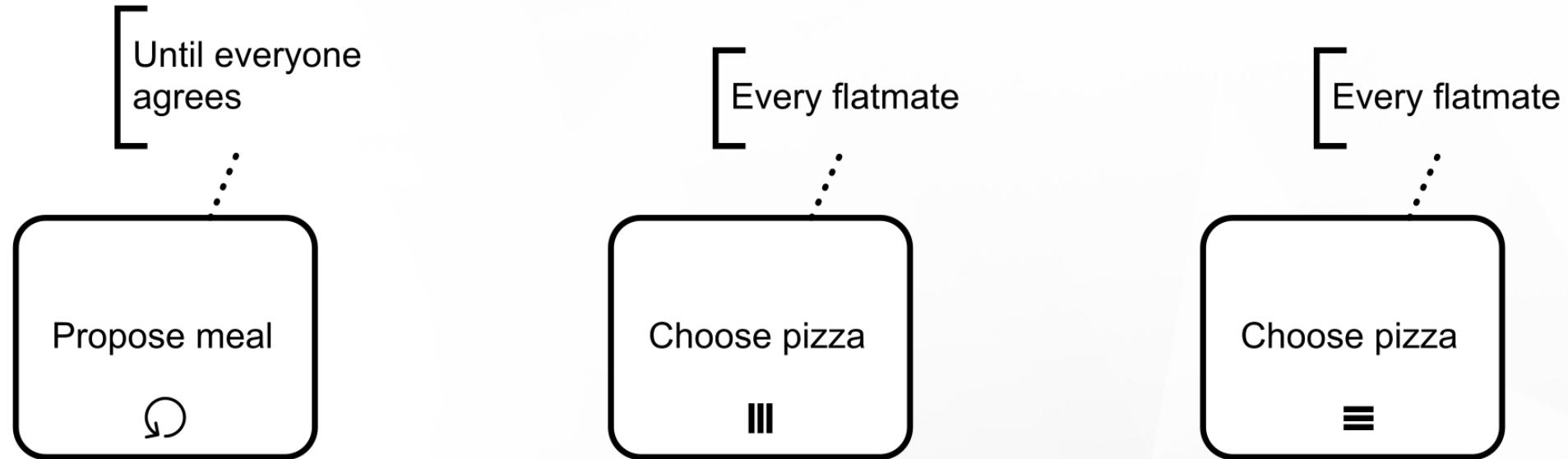
Loop marker



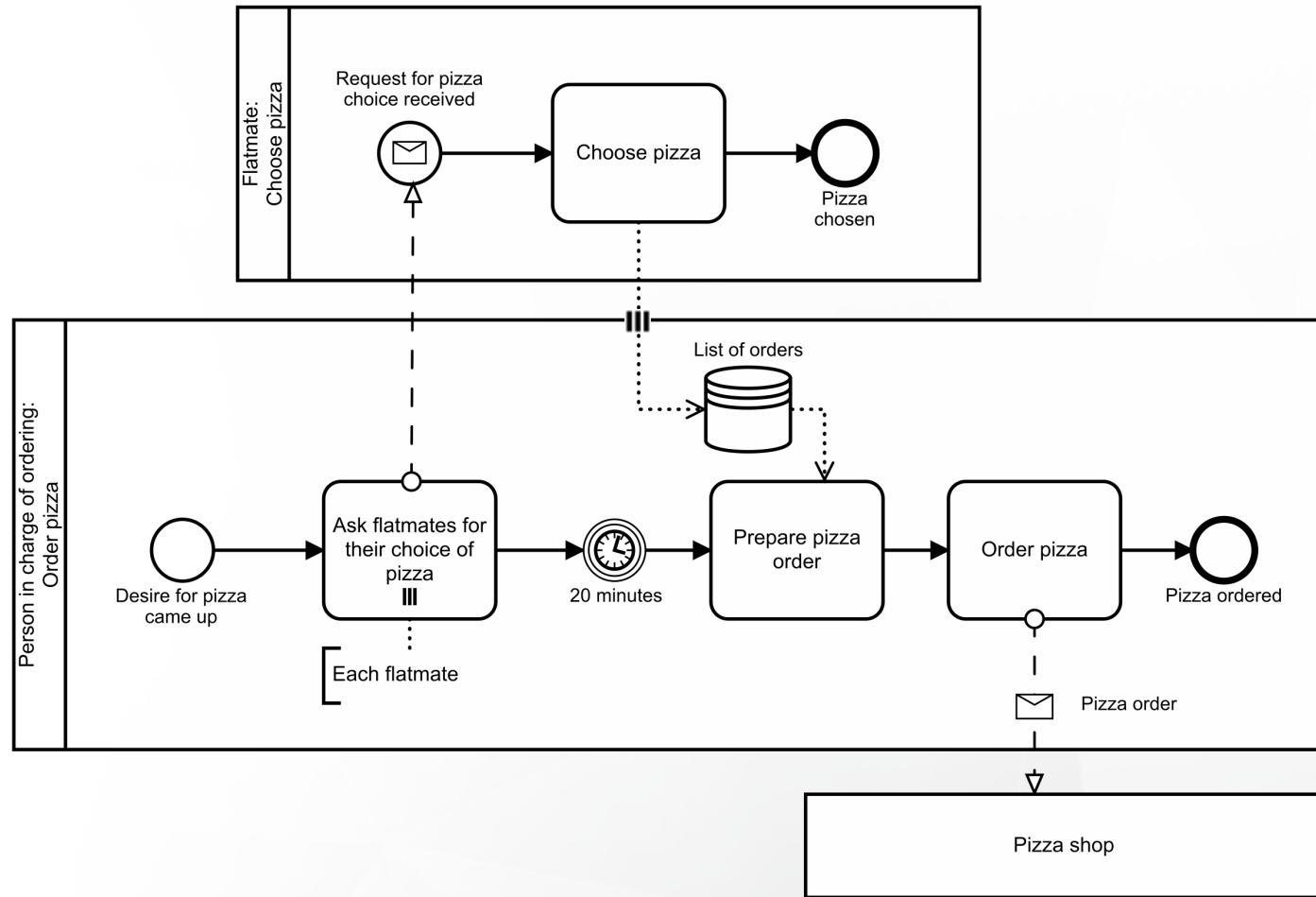
Multi instance marker



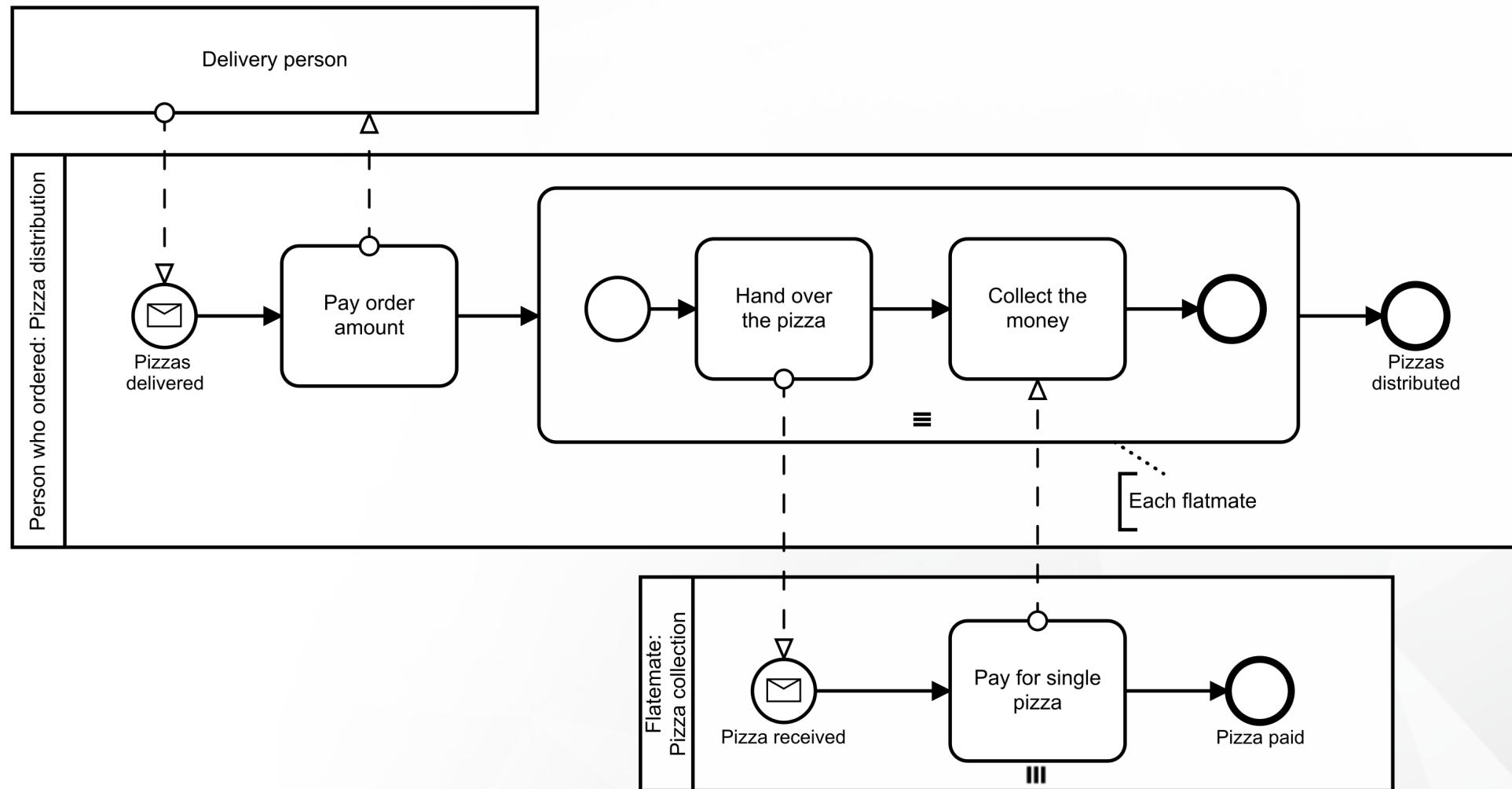
Multiple instance options



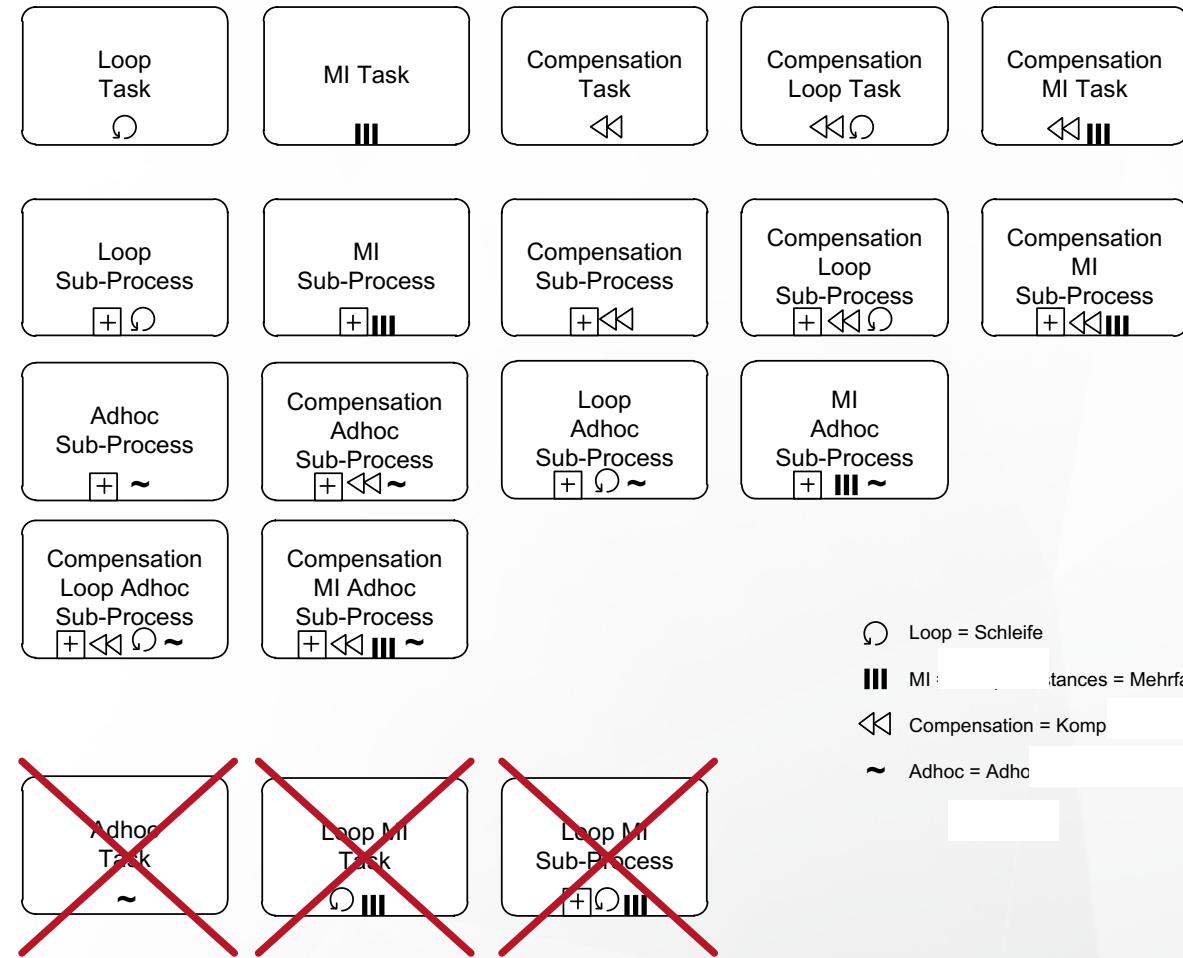
Multiple instance example: ordering pizza



Multiple instance example: pizza delivered

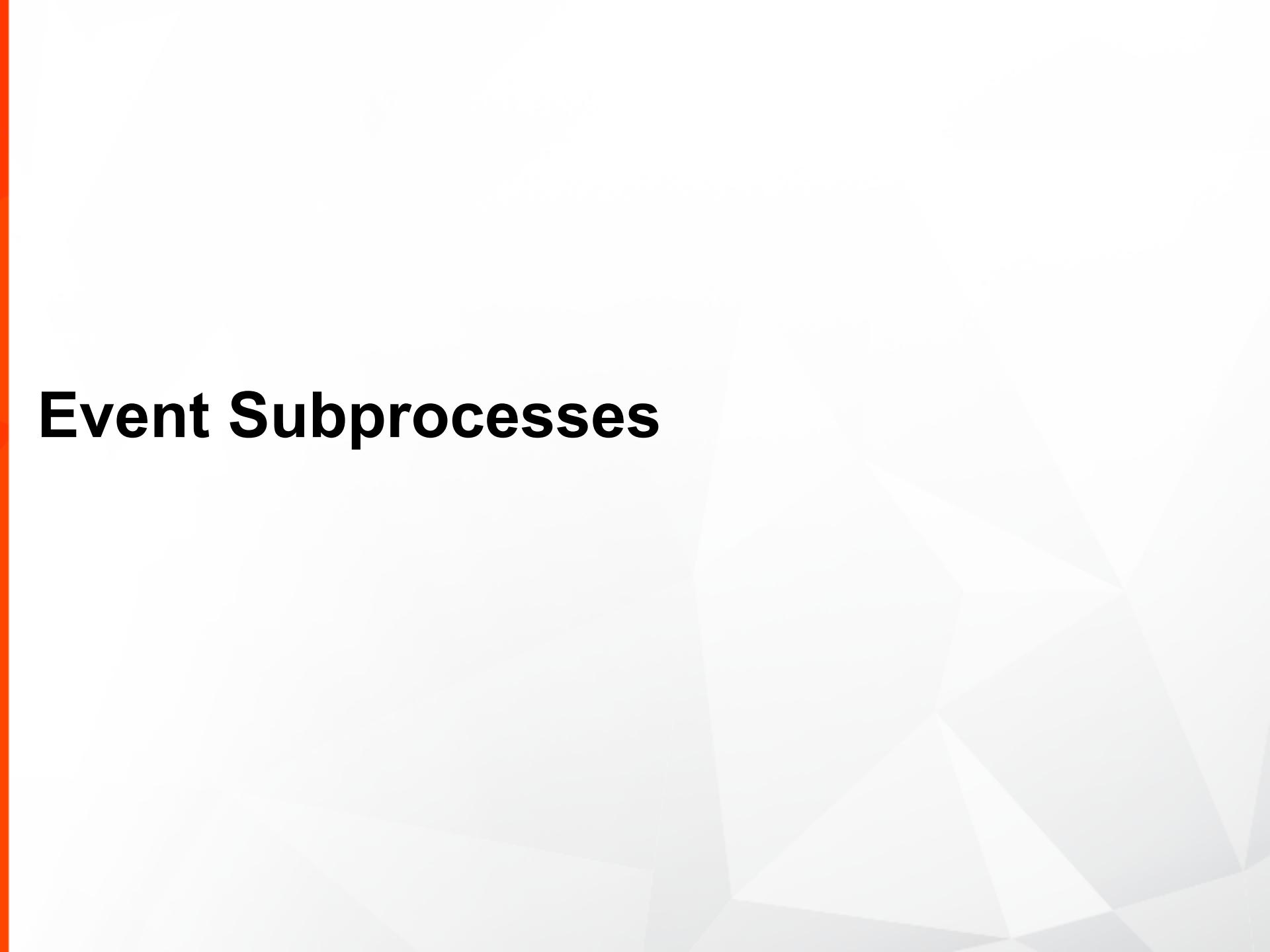


Allowed & forbidden markers





Event Subprocesses



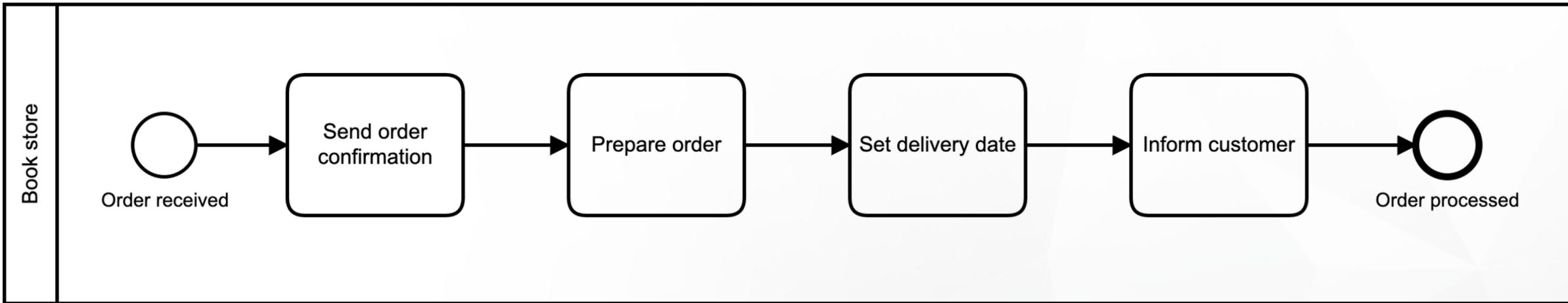
Event Subprocesses

	Catching events							Throwing events						
	Start events			Intermediate events				End events						
	The process is started by the engine.	The event subprocess is started, the parent process is canceled.	The event subprocess is started, the parent process is not canceled.	The process continues only, if the event occurs.	The event is reacted to, the activity is canceled.	The event is reacted to, the activity is not canceled.	The process triggers the event and continues immediately.	The process triggers the event at the end of the process path.						
None: Untyped events; none intermediate events can mark a change of status.	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)
Message: Receiving and sending of messages.	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)	(✉)
Timer: Cyclic timer event, points in time, or time spans.	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)	(🕒)
Conditional: Reacting to changed conditions and relation to business rules.	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)	(☰)
Link: Two associated link events represent a sequence flow.				(➡)				(➡)						
Signal: Signaling across different processes. A signal can be reacted to several times.	(△)	(△)	(△)	(△)	(△)	(△)	(△)	(△)	(△)	(△)	(△)	(△)	(△)	(△)
Error: Triggering and treatment of defined errors		(⚡)			(⚡)			(⚡)				(⚡)	(⚡)	(⚡)
Escalation: Reporting to the next higher level of responsibility.		(Ⓐ)	(Ⓐ)		(Ⓐ)		(Ⓐ)	(Ⓐ)	(Ⓐ)	(Ⓐ)	(Ⓐ)	(Ⓐ)	(Ⓐ)	(Ⓐ)
Terminate: Triggers the immediate termination of the process.													(●)	
Compensation: Handling or triggering of a compensation.		(◀)			(◀)		(◀)		(◀)		(◀)		(◀)	
Cancel: Reaction to canceled transactions or triggering of cancellations.					(✖)		(✖)		(✖)		(✖)		(✖)	
Multiple: Occurrence of one of several events; triggering of all events.	(pentagon)	(pentagon)	(pentagon)	(pentagon)	(pentagon)	(pentagon)	(pentagon)	(pentagon)	(pentagon)	(pentagon)	(pentagon)	(pentagon)	(pentagon)	(pentagon)
Parallel multiple: Occurrence of all events.	(plus)	(plus)	(plus)	(plus)	(plus)	(plus)	(plus)	(plus)	(plus)	(plus)	(plus)			

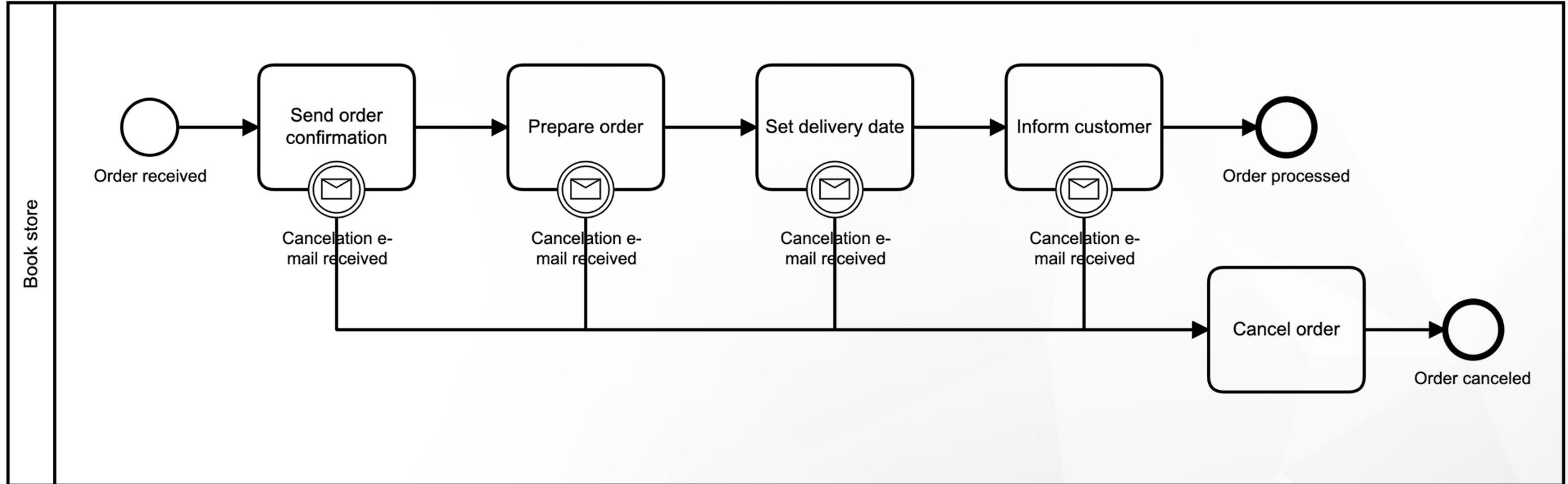
NEW in BPMN 2.0

Symbols of BPMN (following the BPMN poster of the Berlin BPM offensive: www.bpmn.de/poster)
Taken from "Praxishandbuch BPMN" by Jakob Freund and Bernd Rücker; © 2010 Carl Hanser Verlag Munich

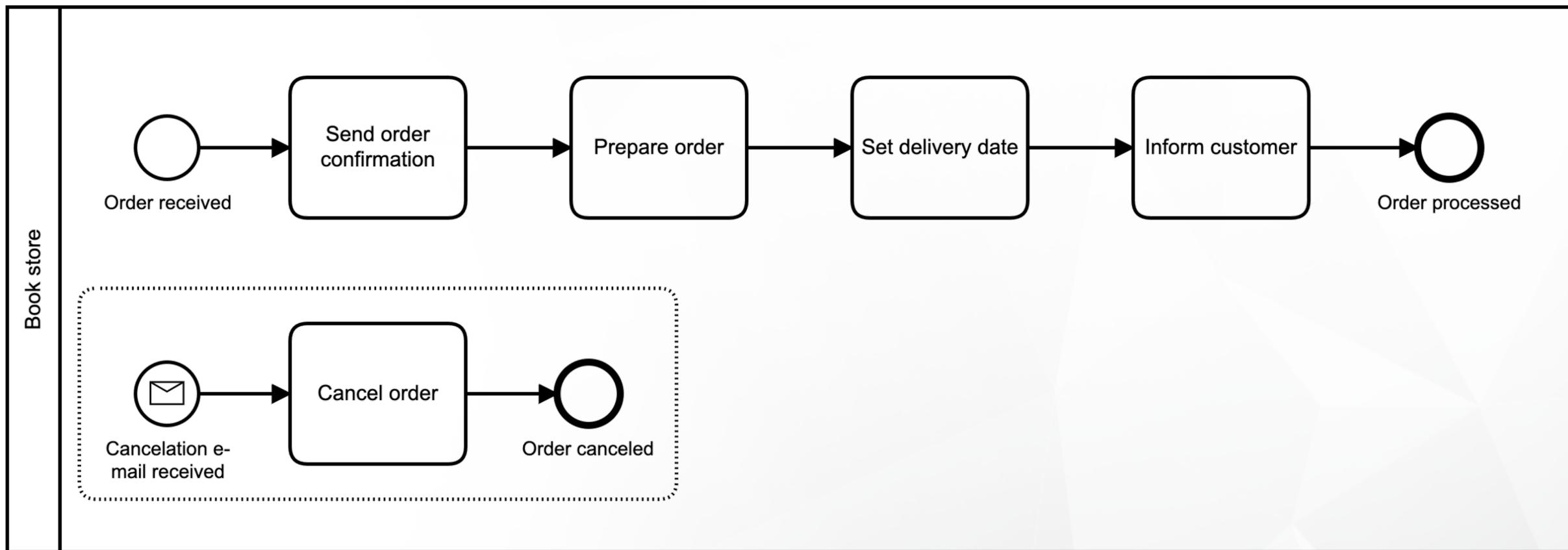
How can we handle a cancellation?



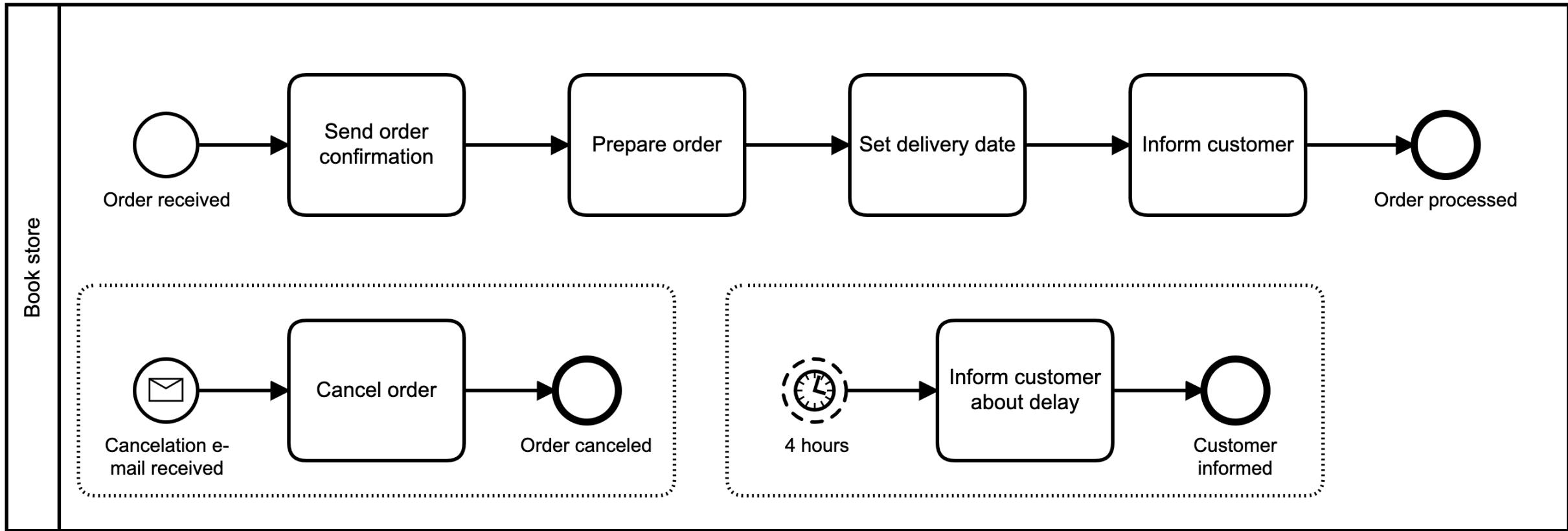
The way we already know: With boundary events!



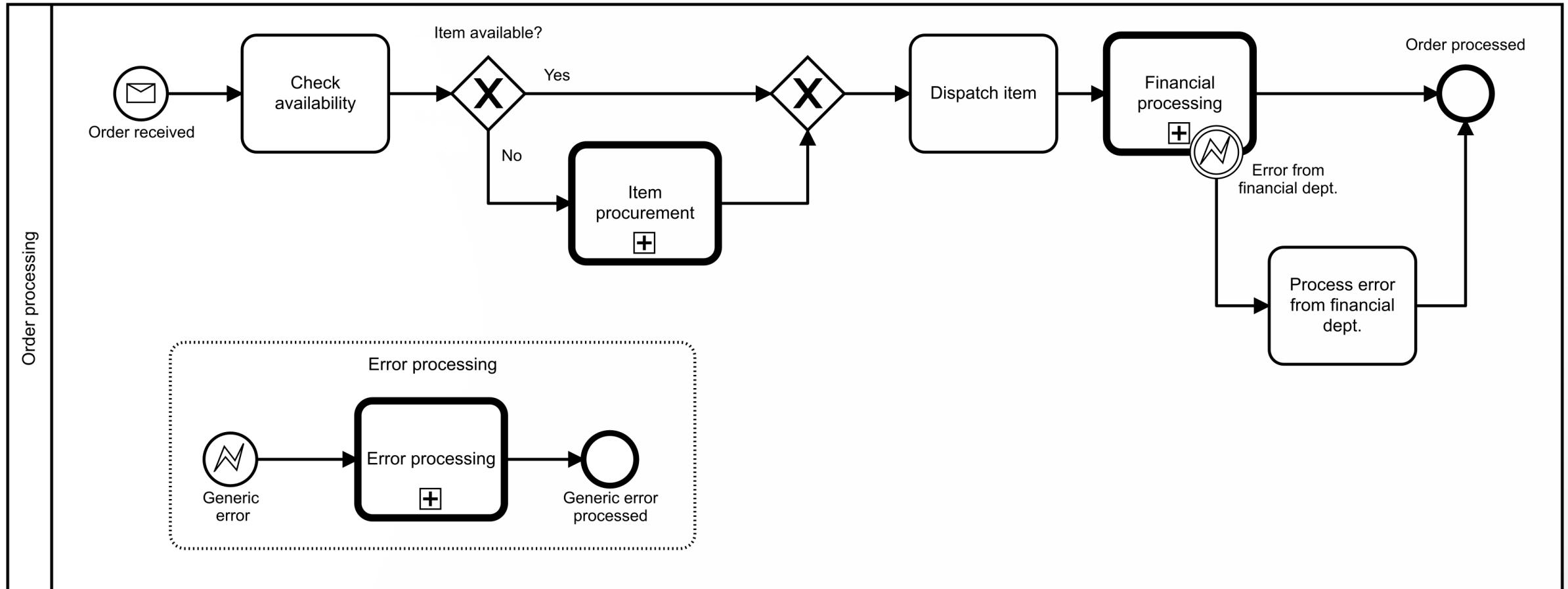
The new way: With an event subprocess!



Event subprocess: timeouts



Event subprocess: global error handling



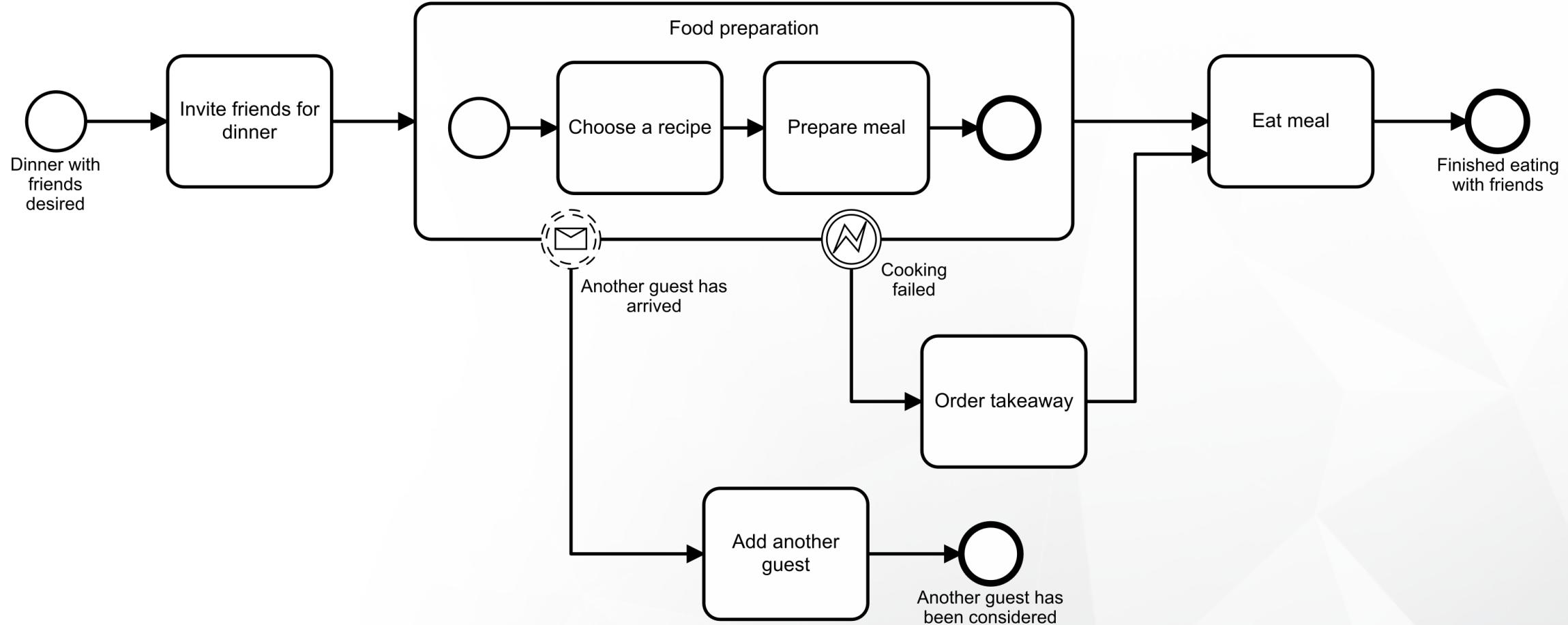


Calm Pizza Customer Down

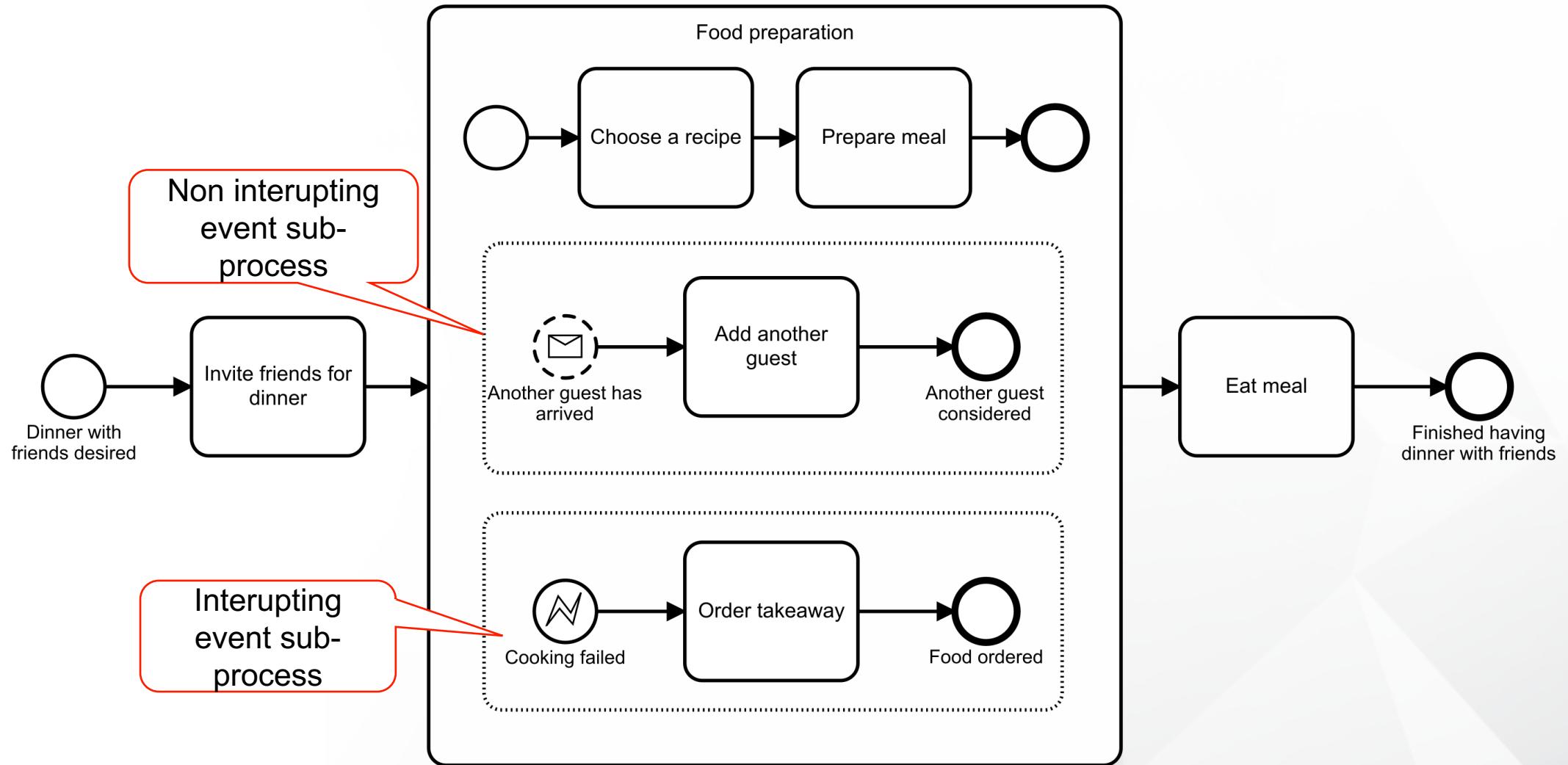
Exercise

<https://training.camunda.com/bpmn/>
user: bpmn
password: camundarocks

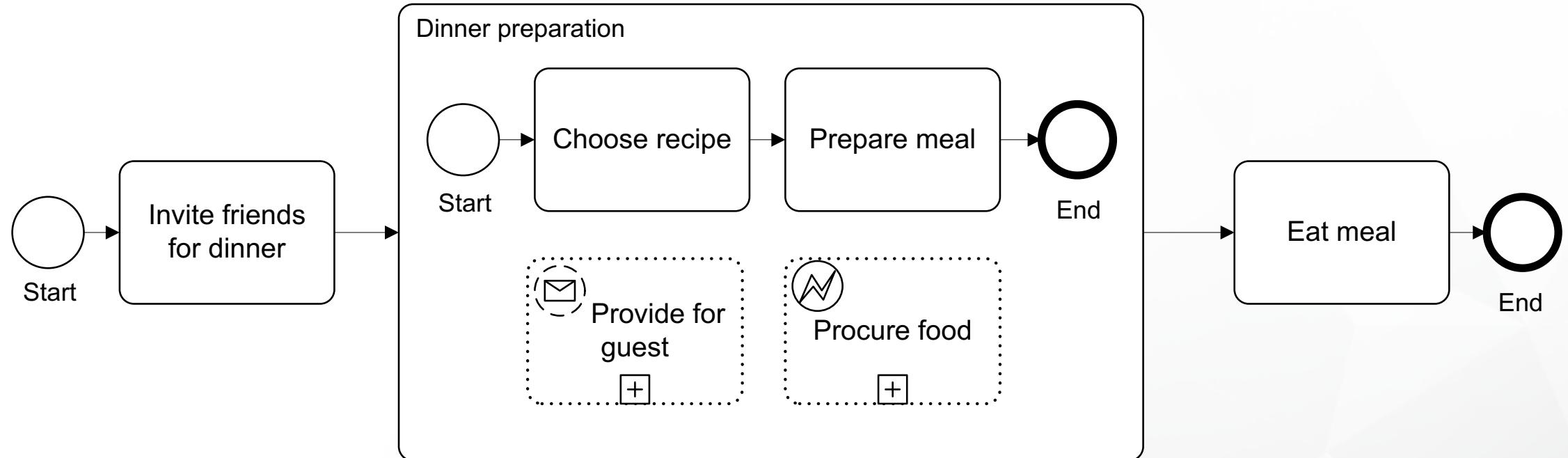
Attaching events to subprocesses



Local event subprocesses

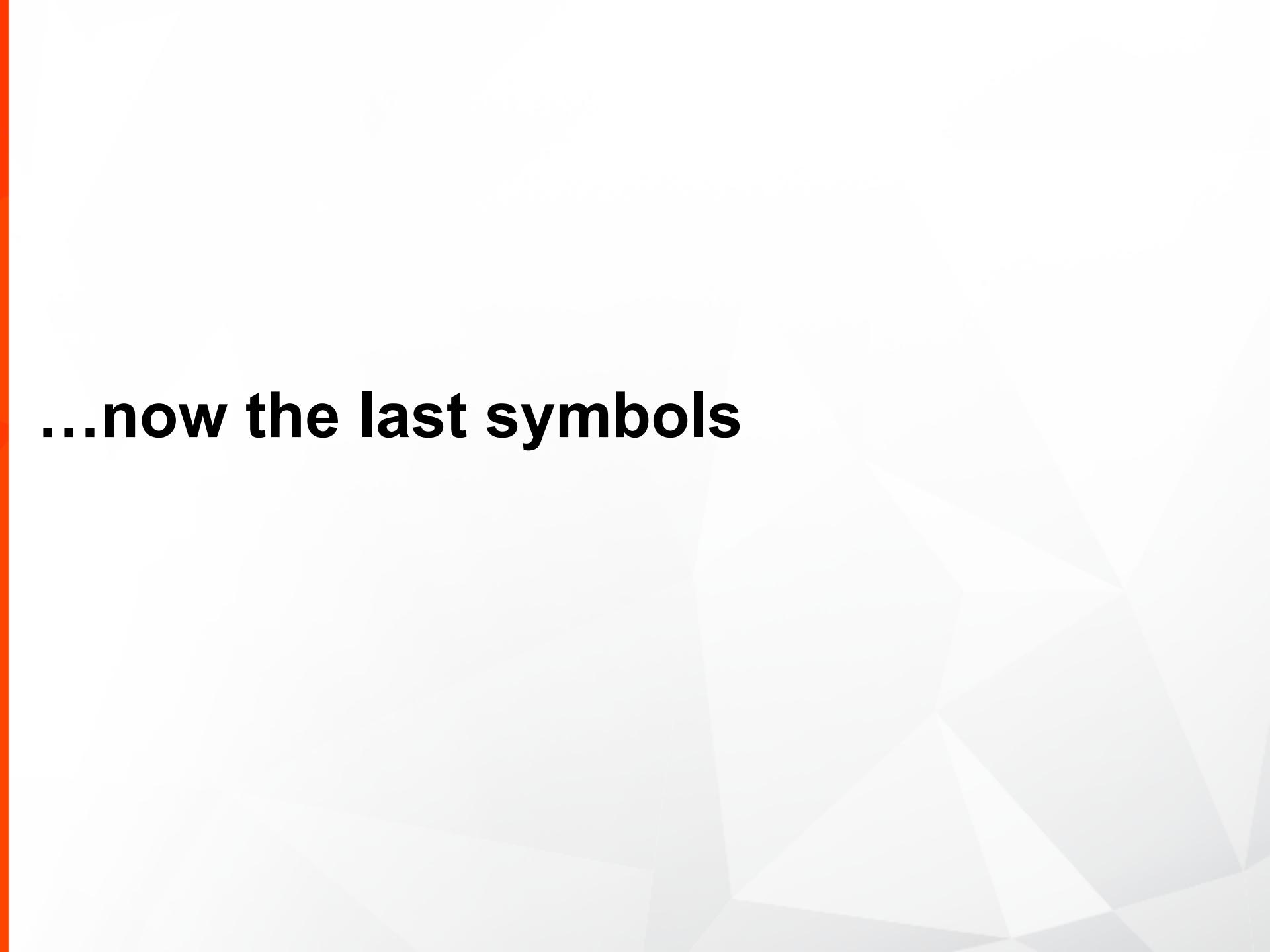


Collapsed event subprocesses

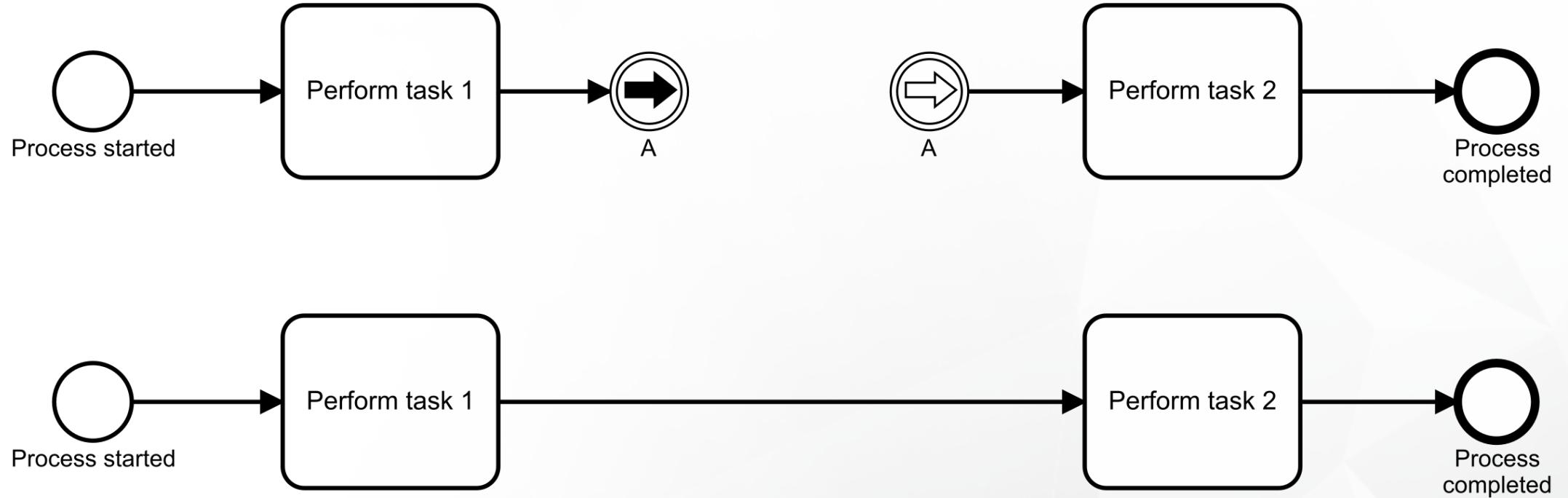




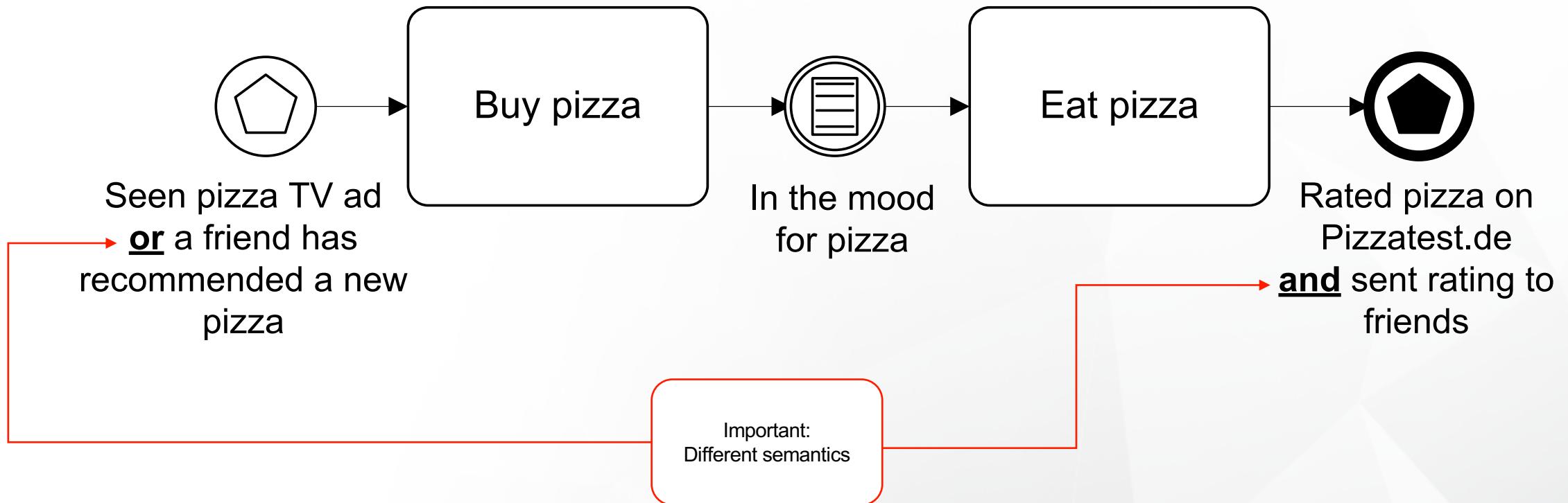
...now the last symbols



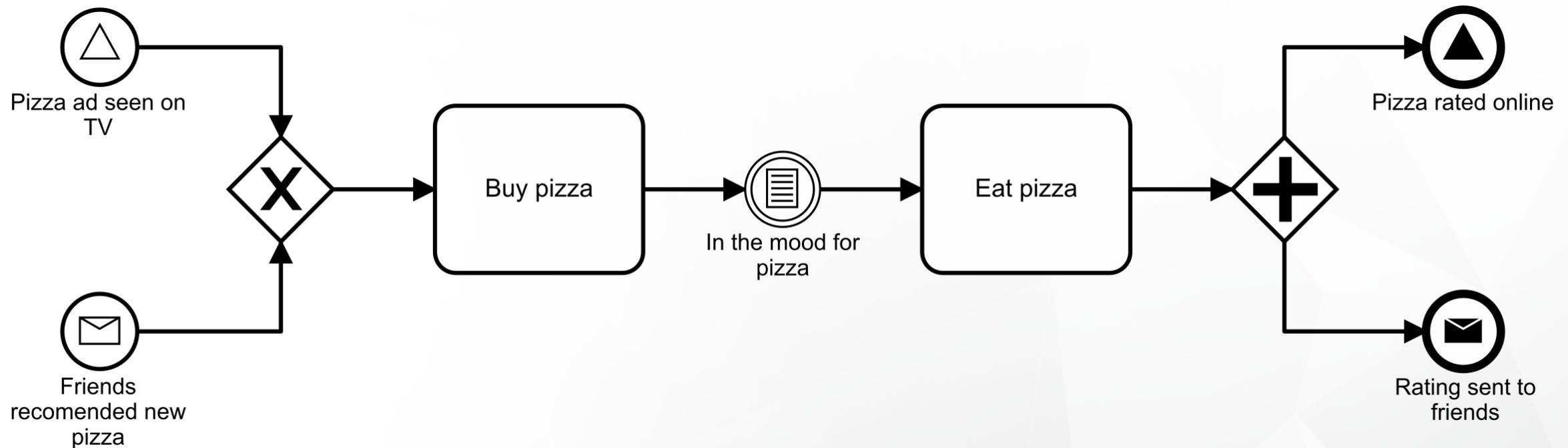
Associated link events can replace a sequence flow



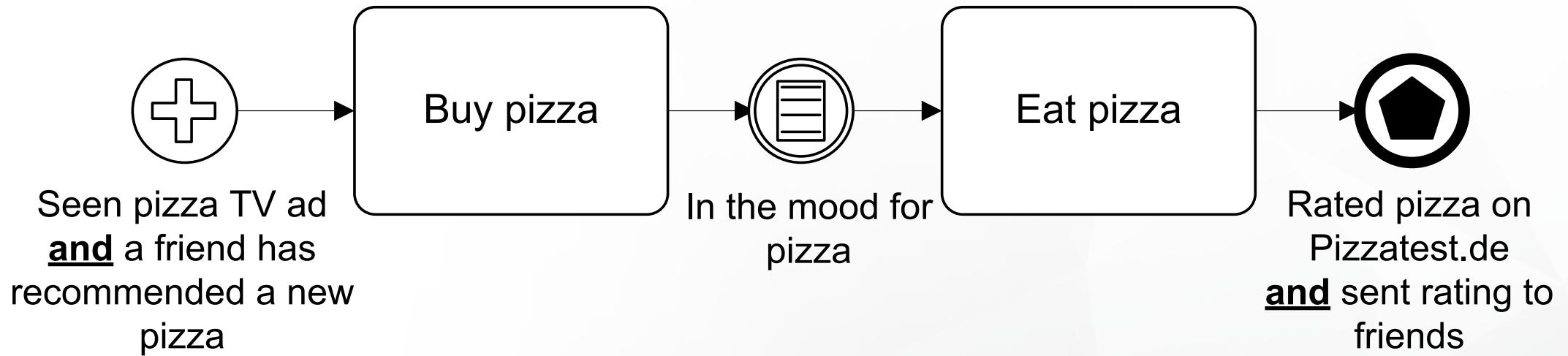
Event type: multiple



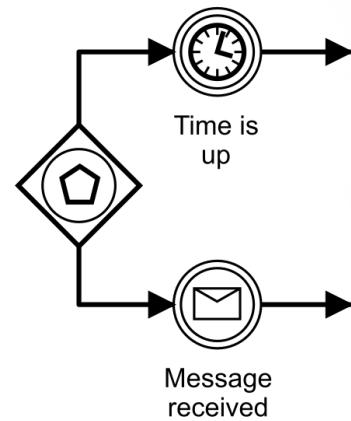
Alternative for multiple event



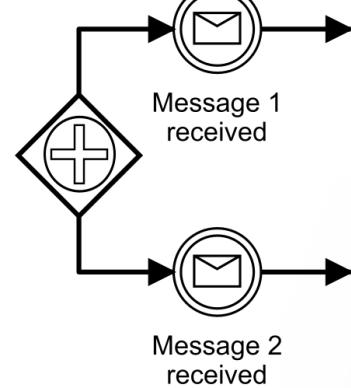
Event type: multiple parallel



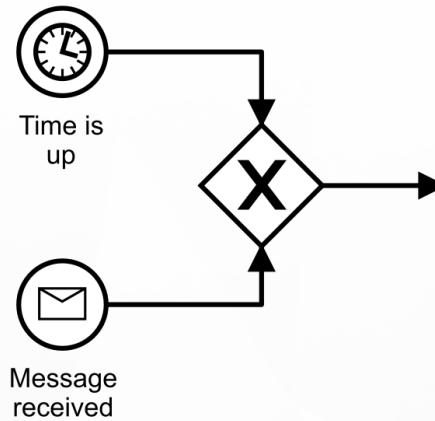
Instantiating gateways



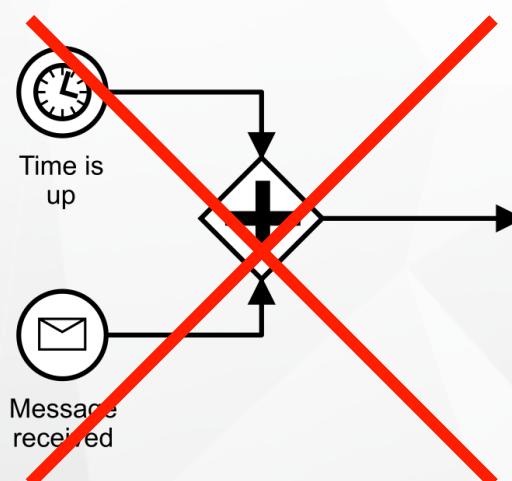
Time is up
OR
Message received



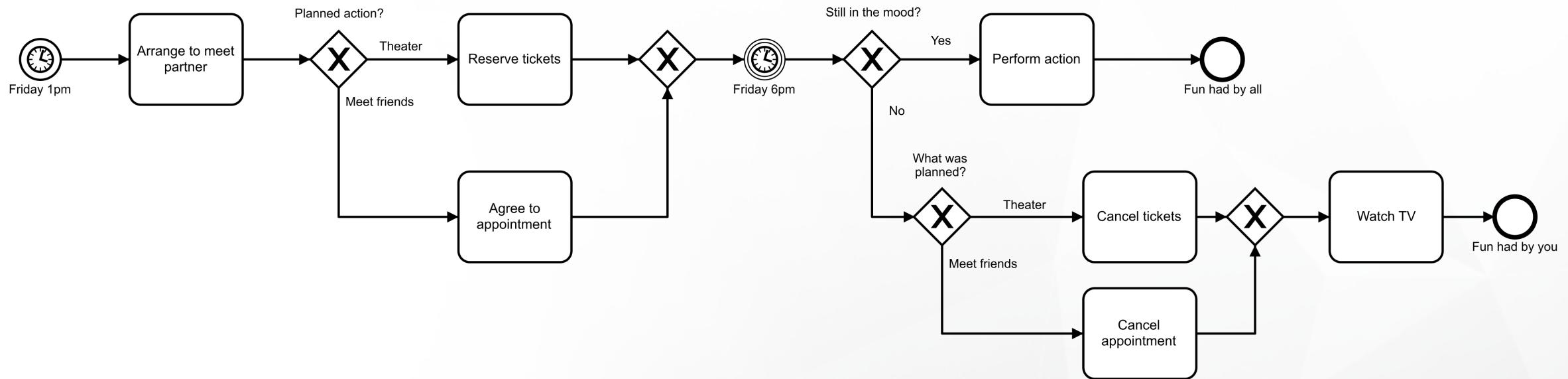
Time is up
AND
Message received



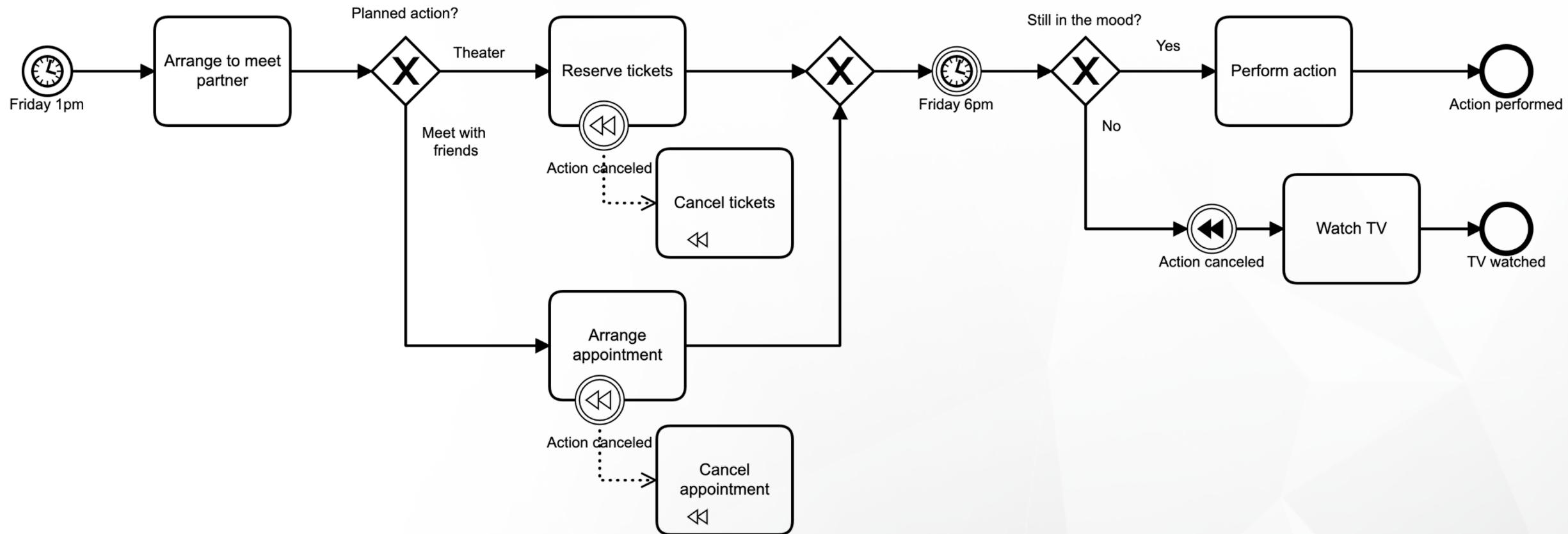
Message received



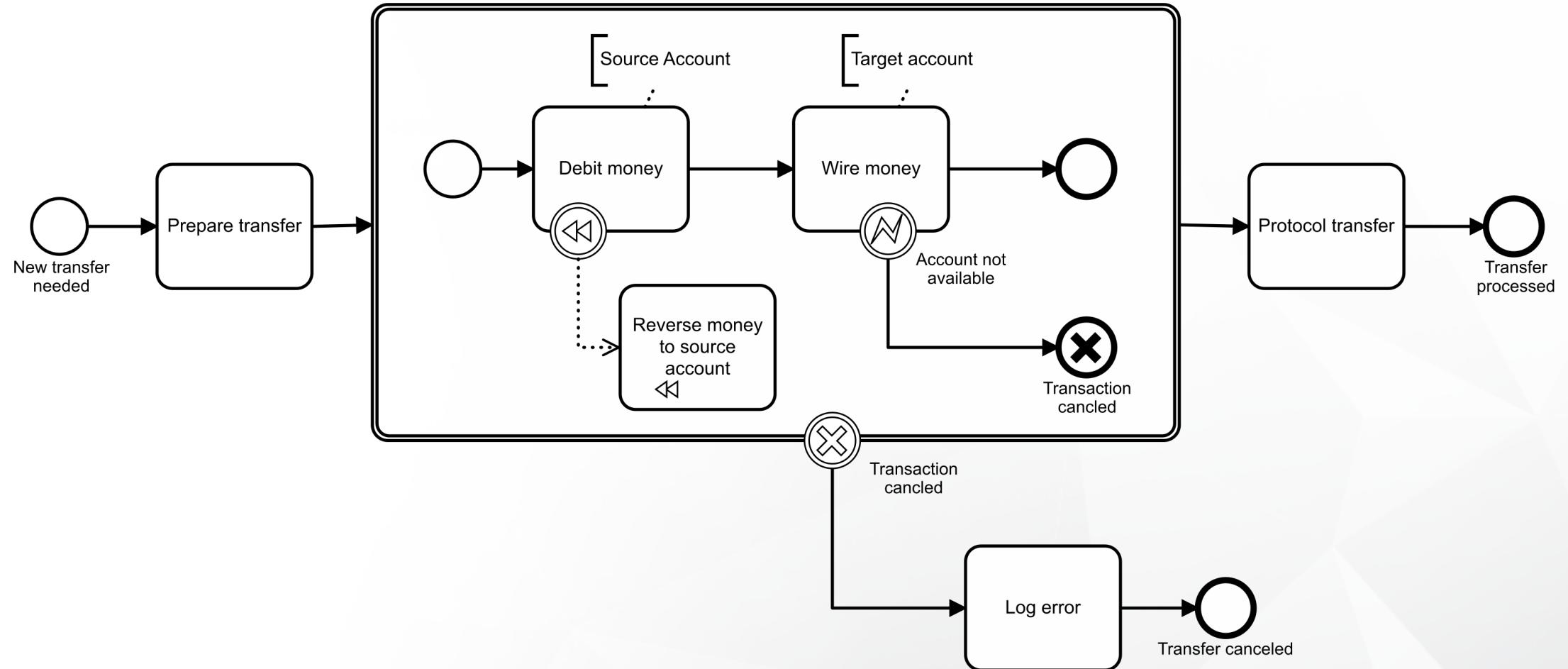
Compensation



Compensation event



Transactions





BPMN Roulette

Exercise

<https://training.camunda.com/bpmn/>
user: bpmn
password: camundarocks

Coupling of processes

Types of coupling

Direct coupling

- Message flows
- Signal events
- Call activities
- Decomposition (Sub-process)

Indirect coupling

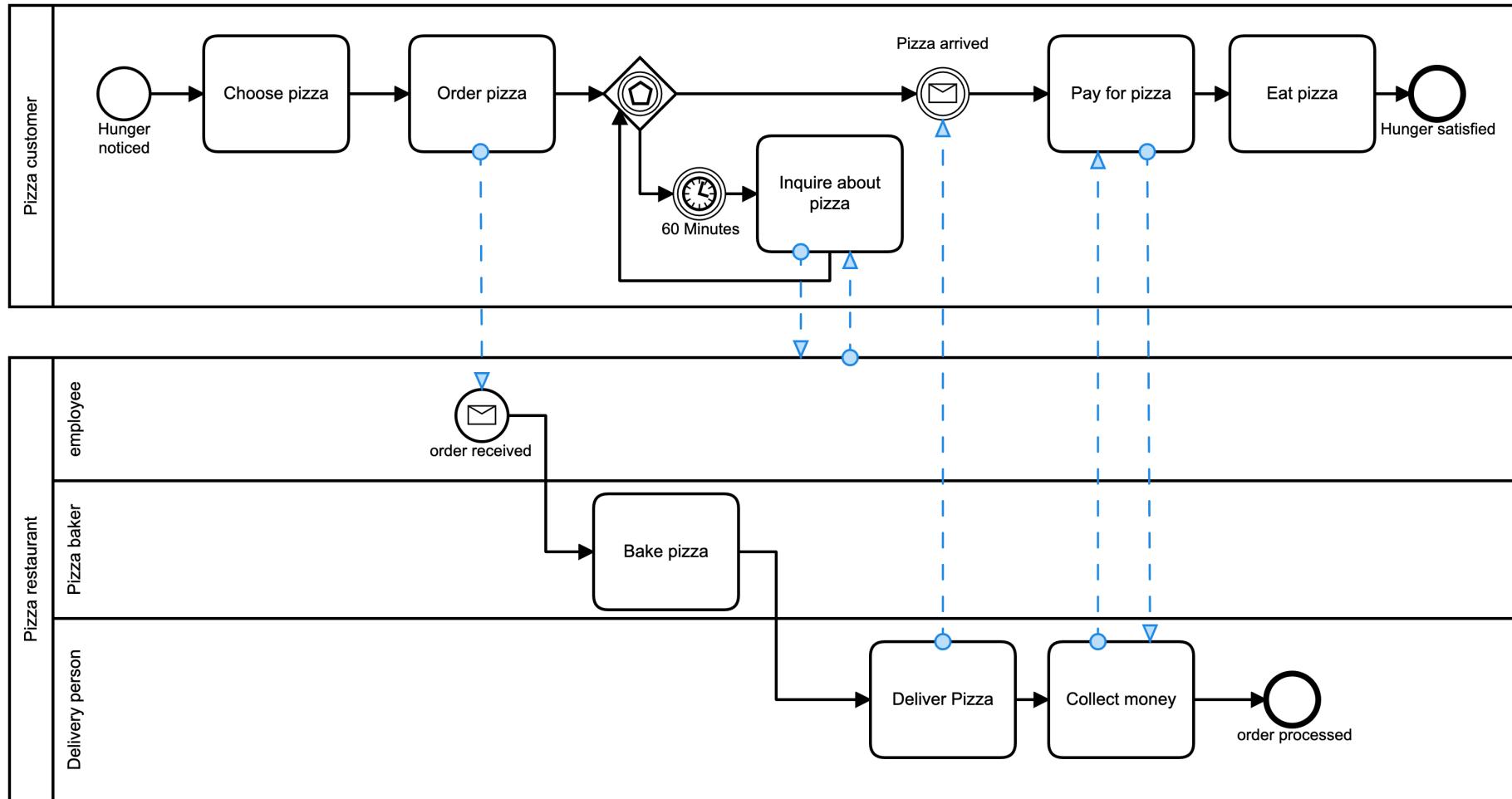
- System status
- Process data

Direct Coupling

Characteristics:

- The linked process is called / started directly
- Coupling is obvious in the model
- Synchronous calls are possible

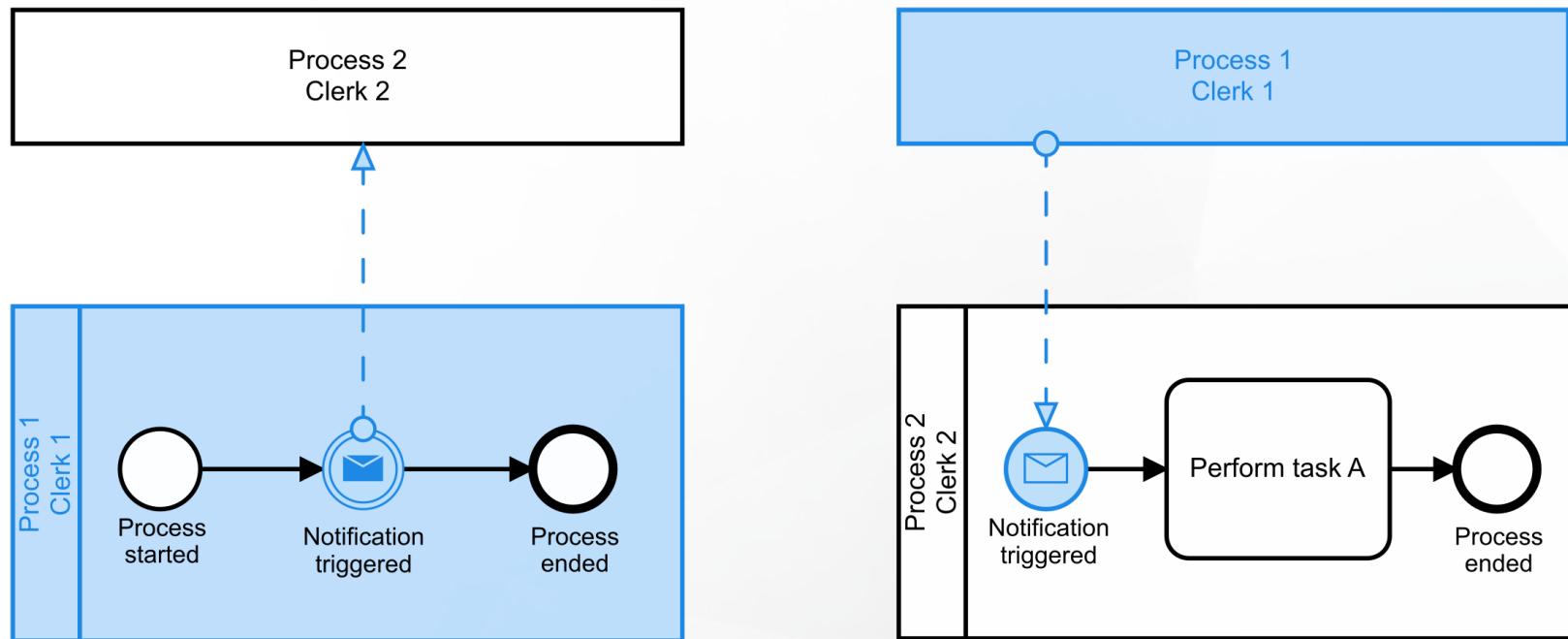
Direct coupling: message flows



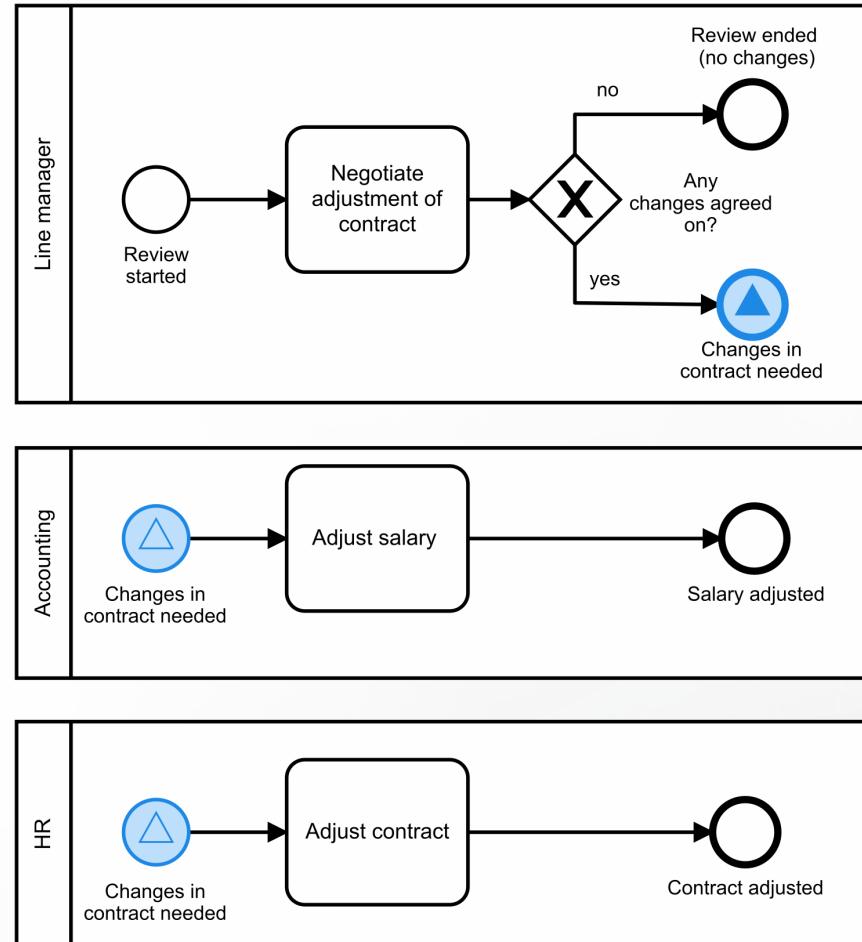
See also:
Orchestration and
collaboration

Coupling with message flows in practice

- Typically each model describes one process
- Collapsed pools are used to point out the coupling

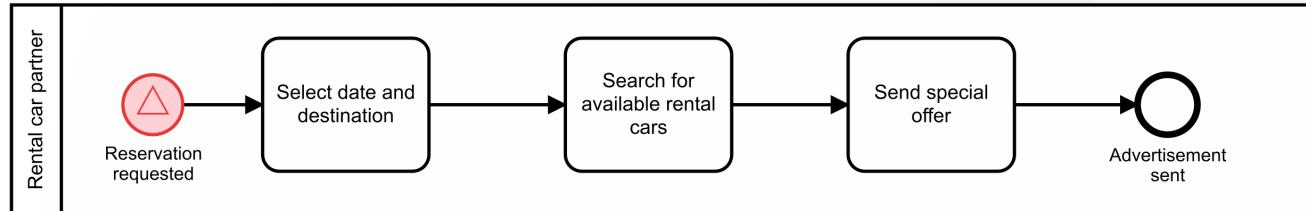
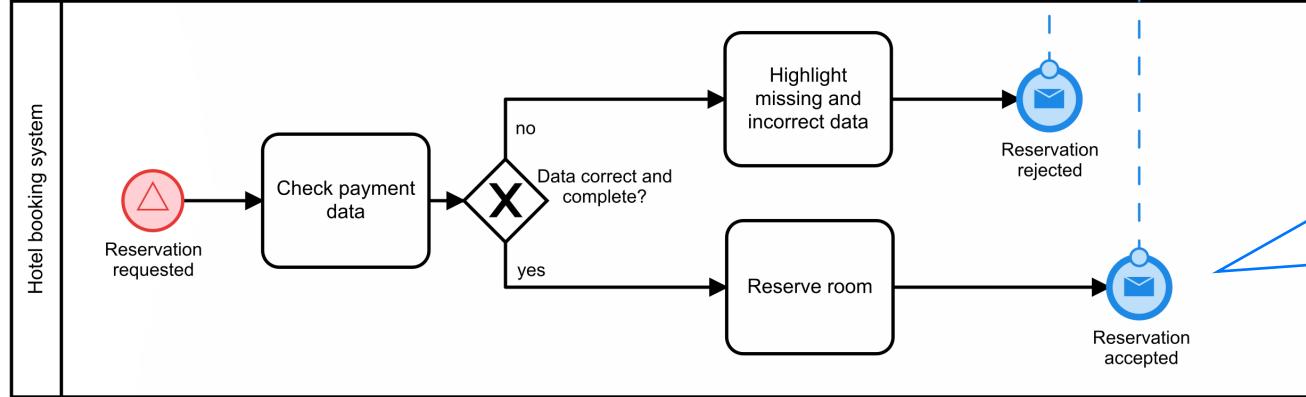
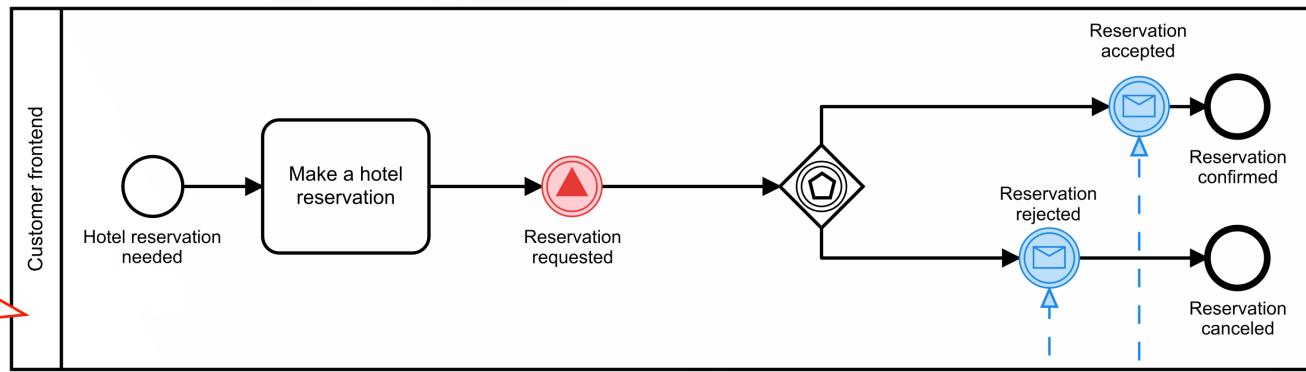


Direct coupling: signal event



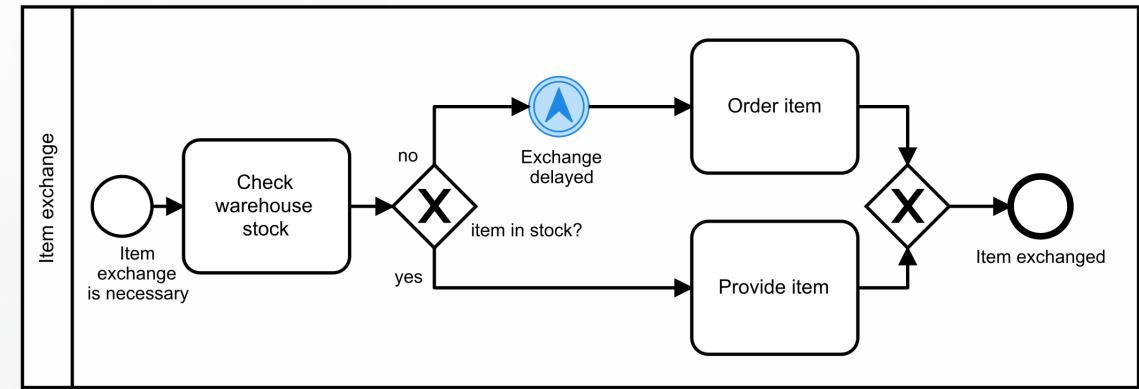
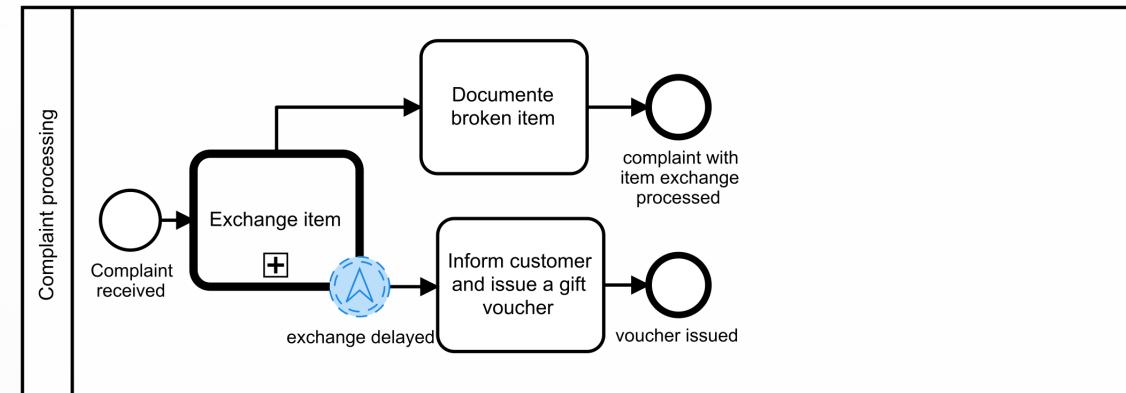
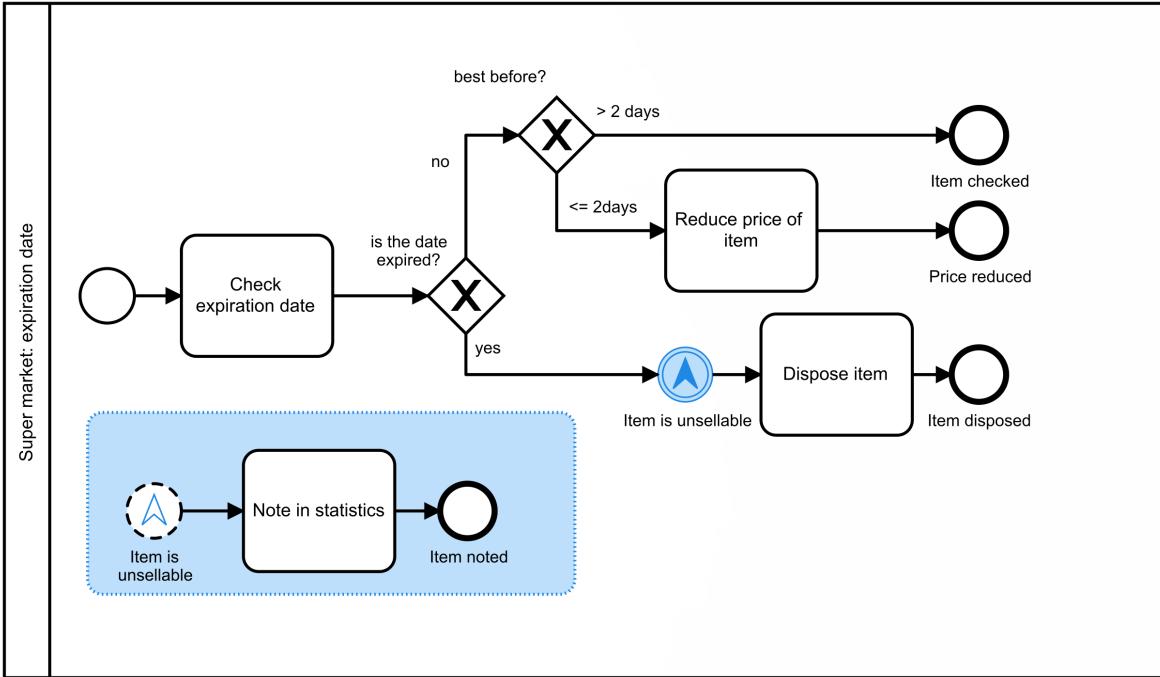
Difference: message vs. signal

A signal can have multiple recipients (1:n relation)



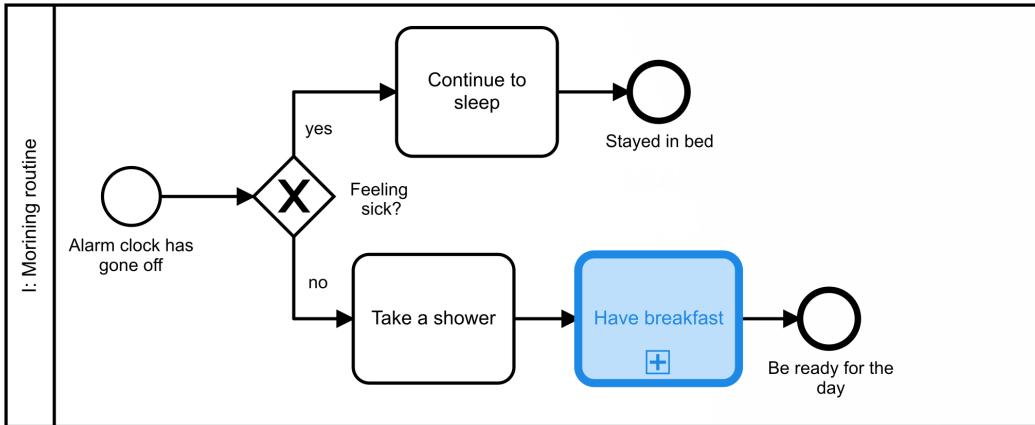
A message flow always has one sender and exactly one recipient (1:1 relation)

Difference: escalation

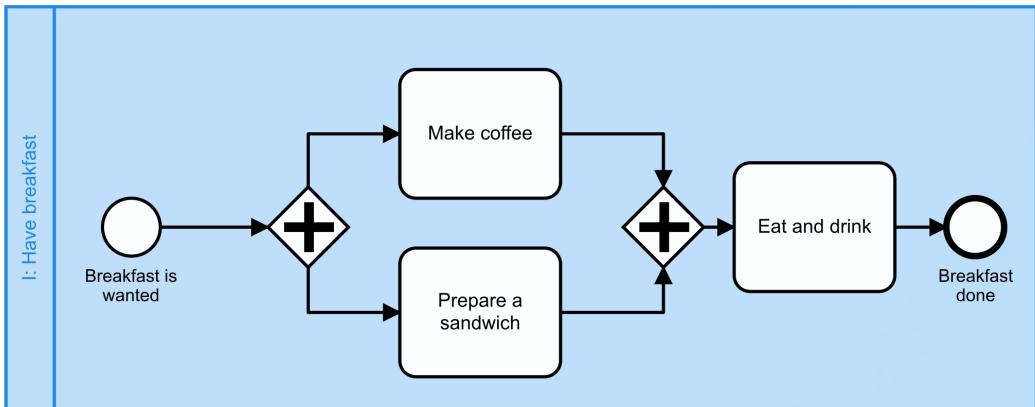


- Escalation and error events just allow a **vertical** communication (within one process hierarchy)
- They can only be received by attached boundary events or in an event sub process.

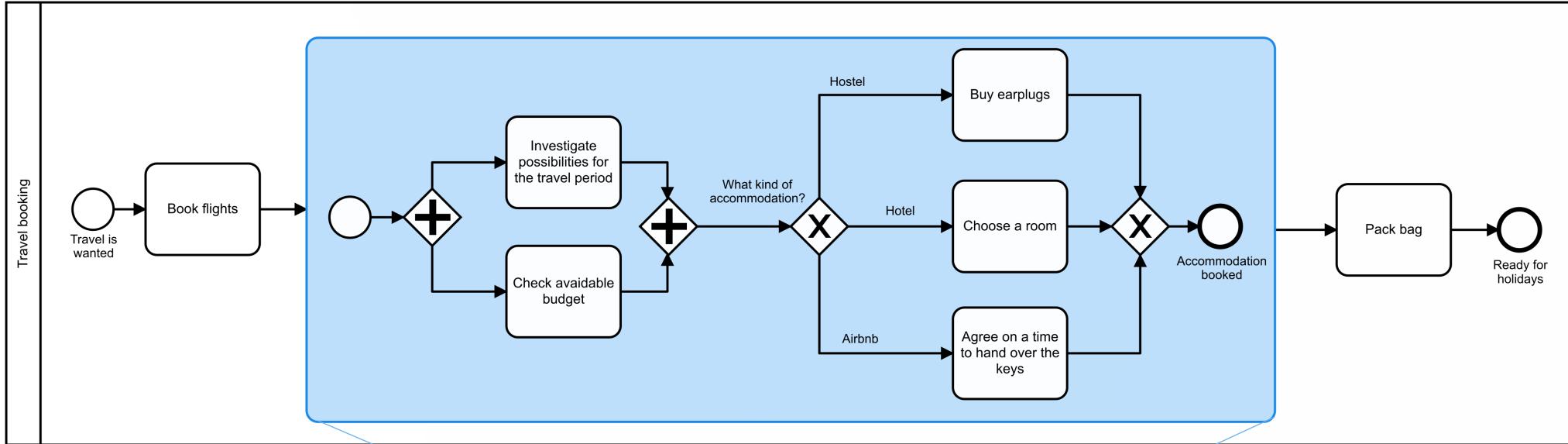
Direct coupling: call activity



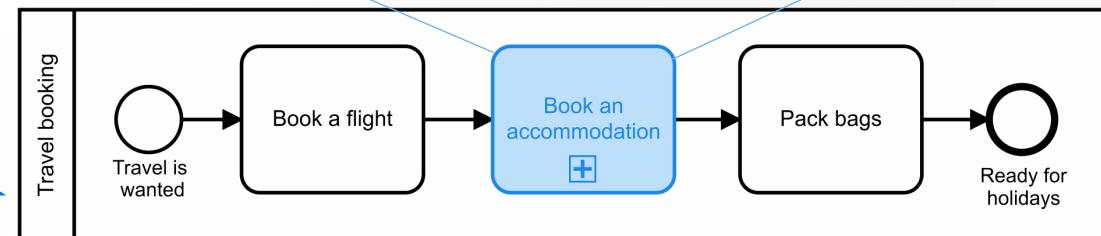
- The call activity is started directly by the main process



Direct coupling: decomposition (collapsed subprocess)



Disadvantage: By using the decomposition, the sub process is not visible



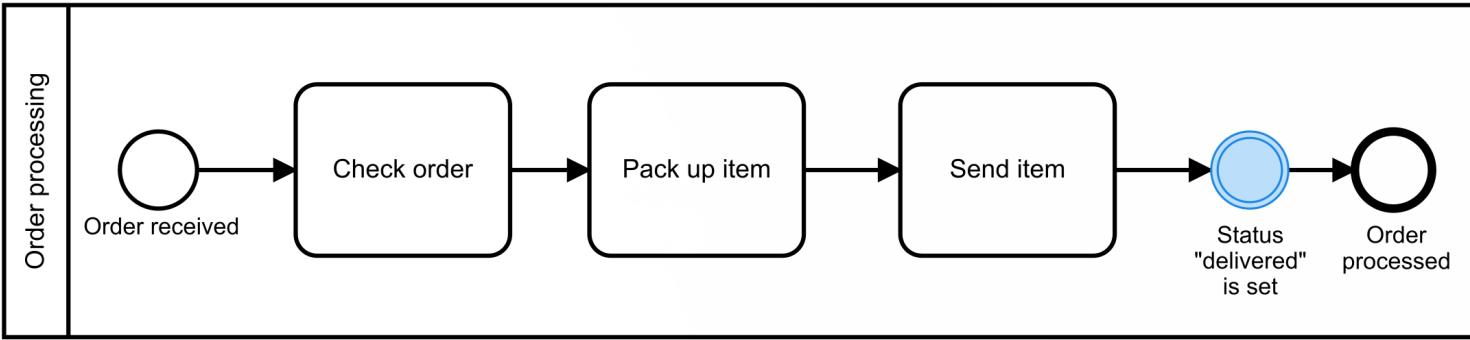
- The subprocess is started directly by the main process

Indirect coupling

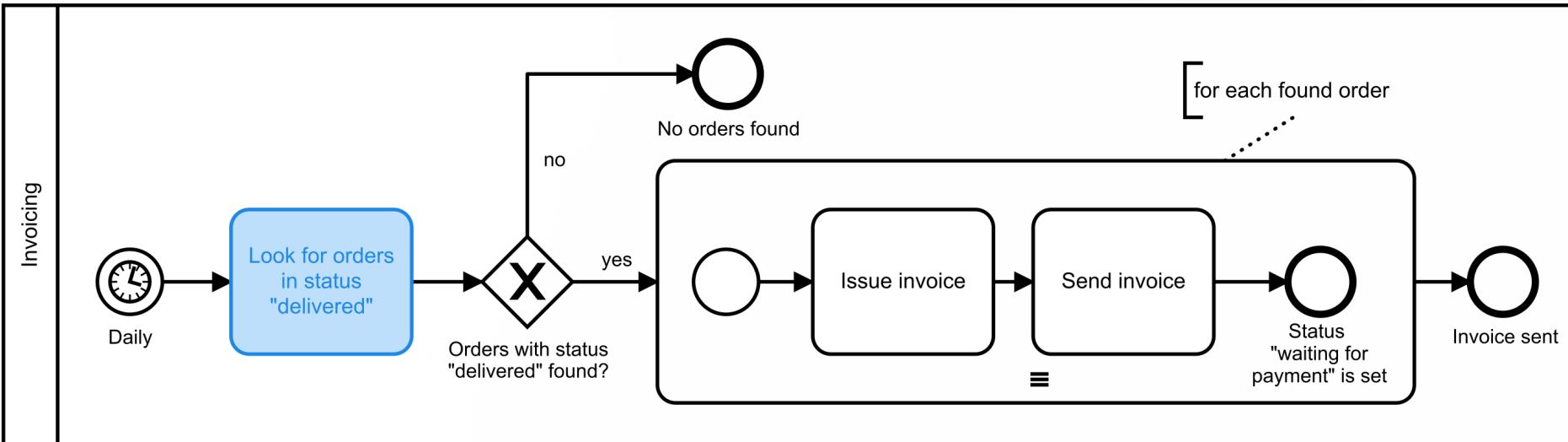
Characteristics:

- The linked process is not called / started directly
- Usually requires gathering of status data from the linked process
- Coupling is not obvious in the model
- Is necessary for coupling batch and non-batch processes

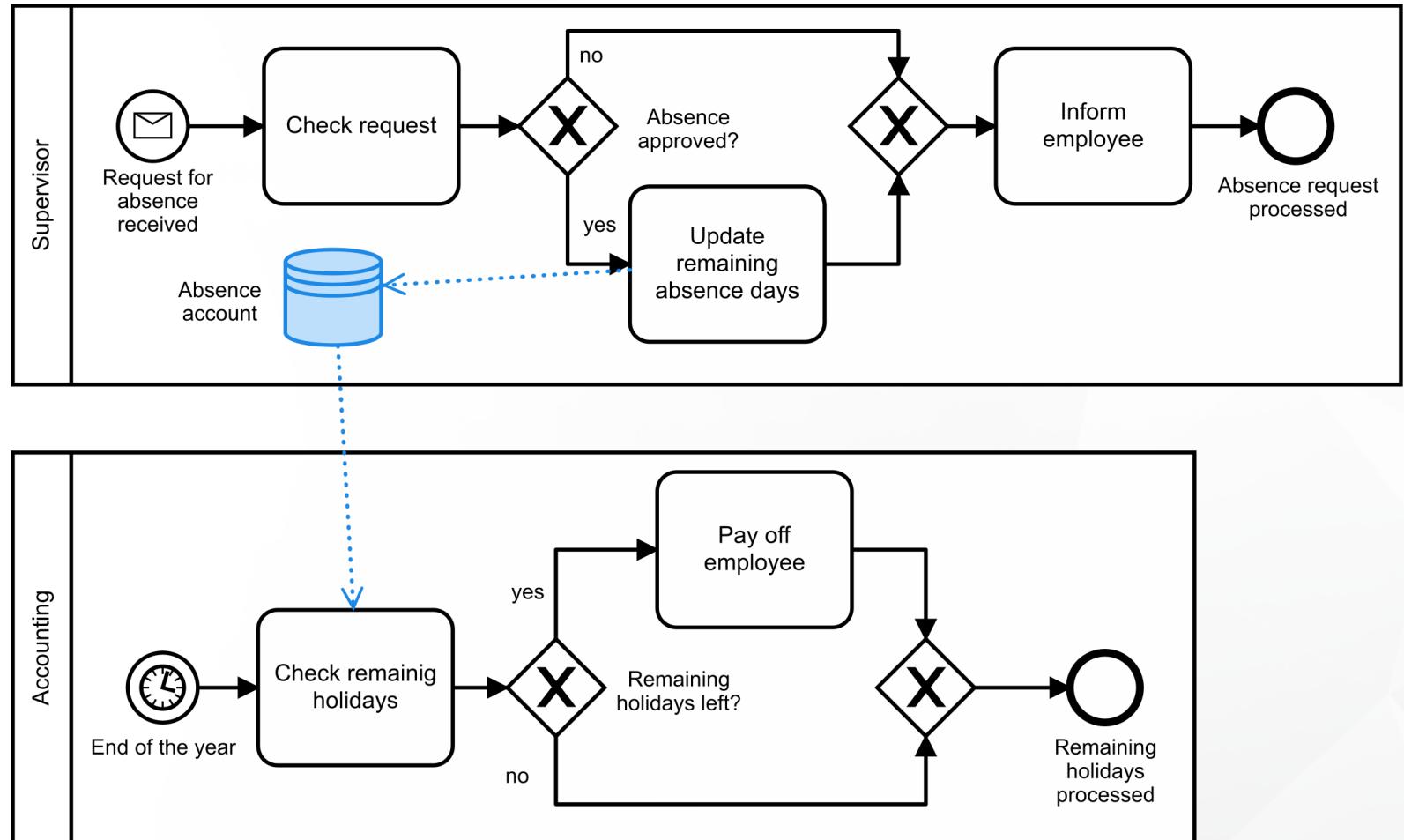
Indirect coupling: system status



Often the coupling is not obvious in the process



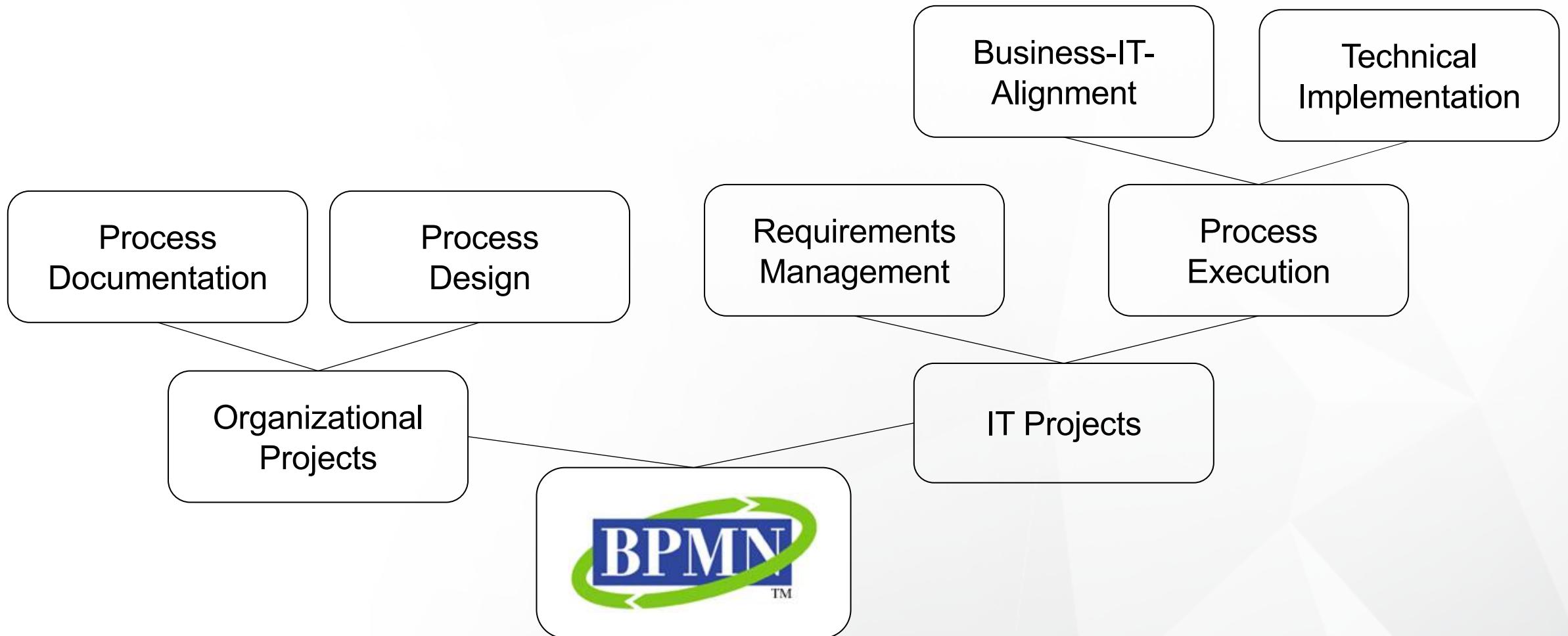
Indirect coupling: process data



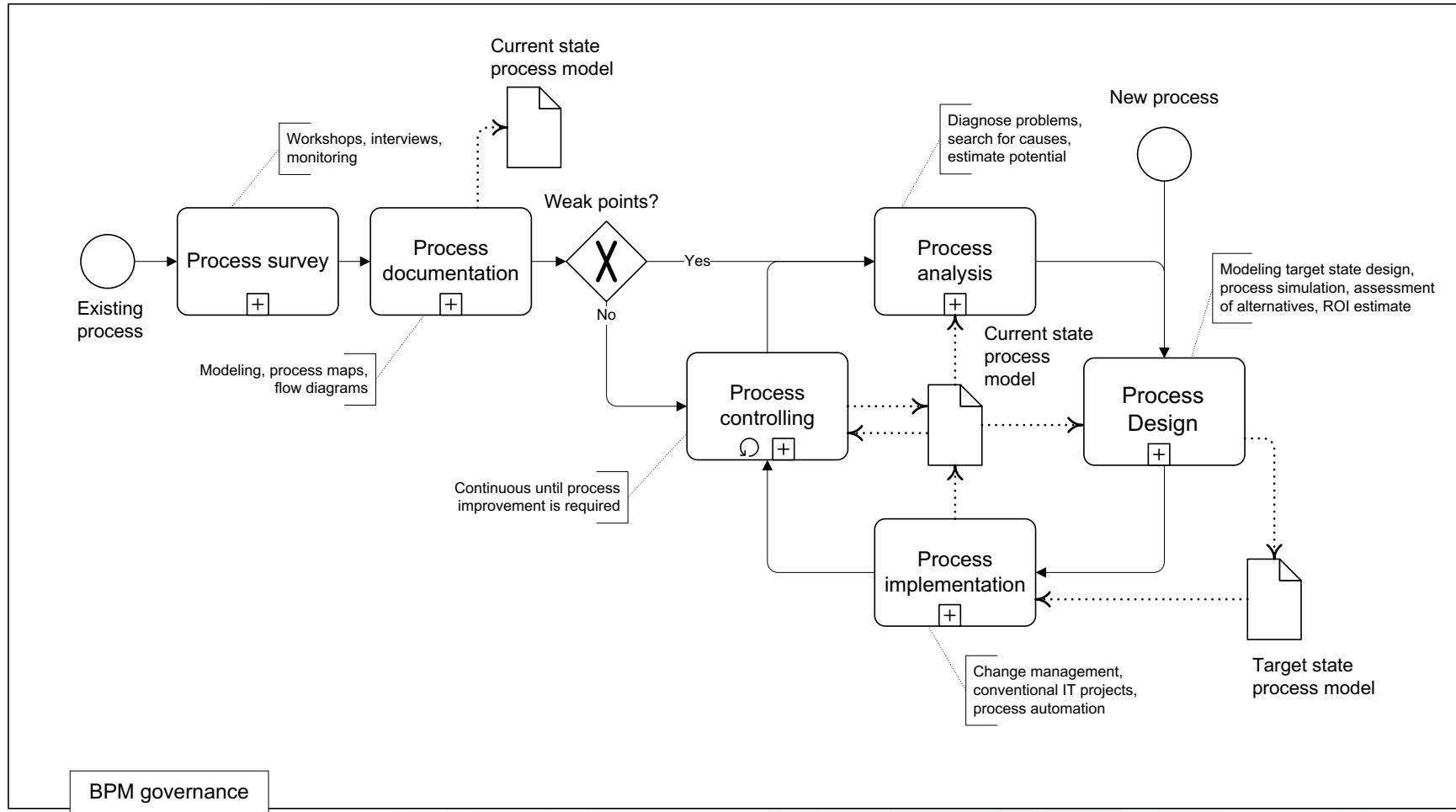


Introducing BPMN on a broad base

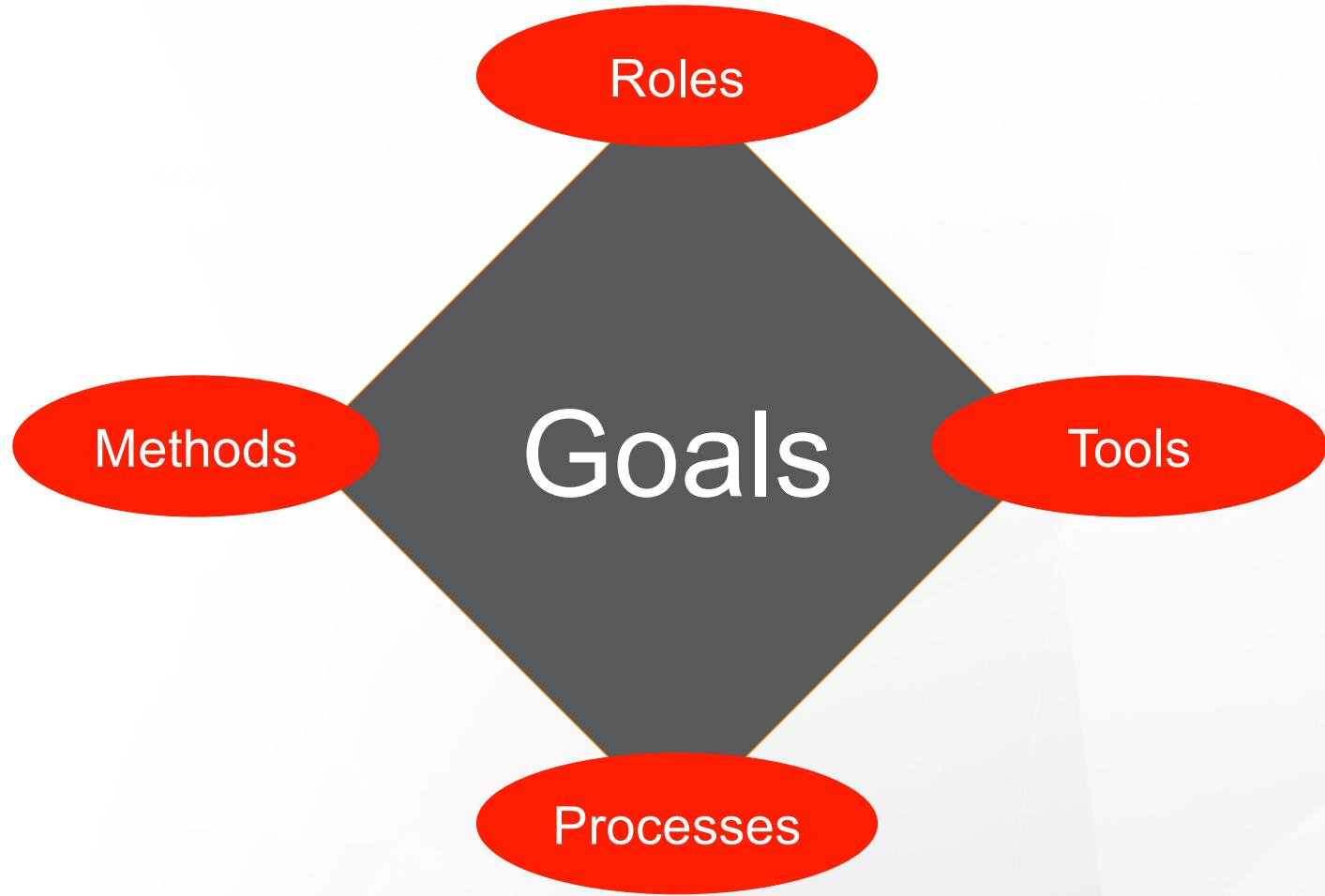
Different application scenarios



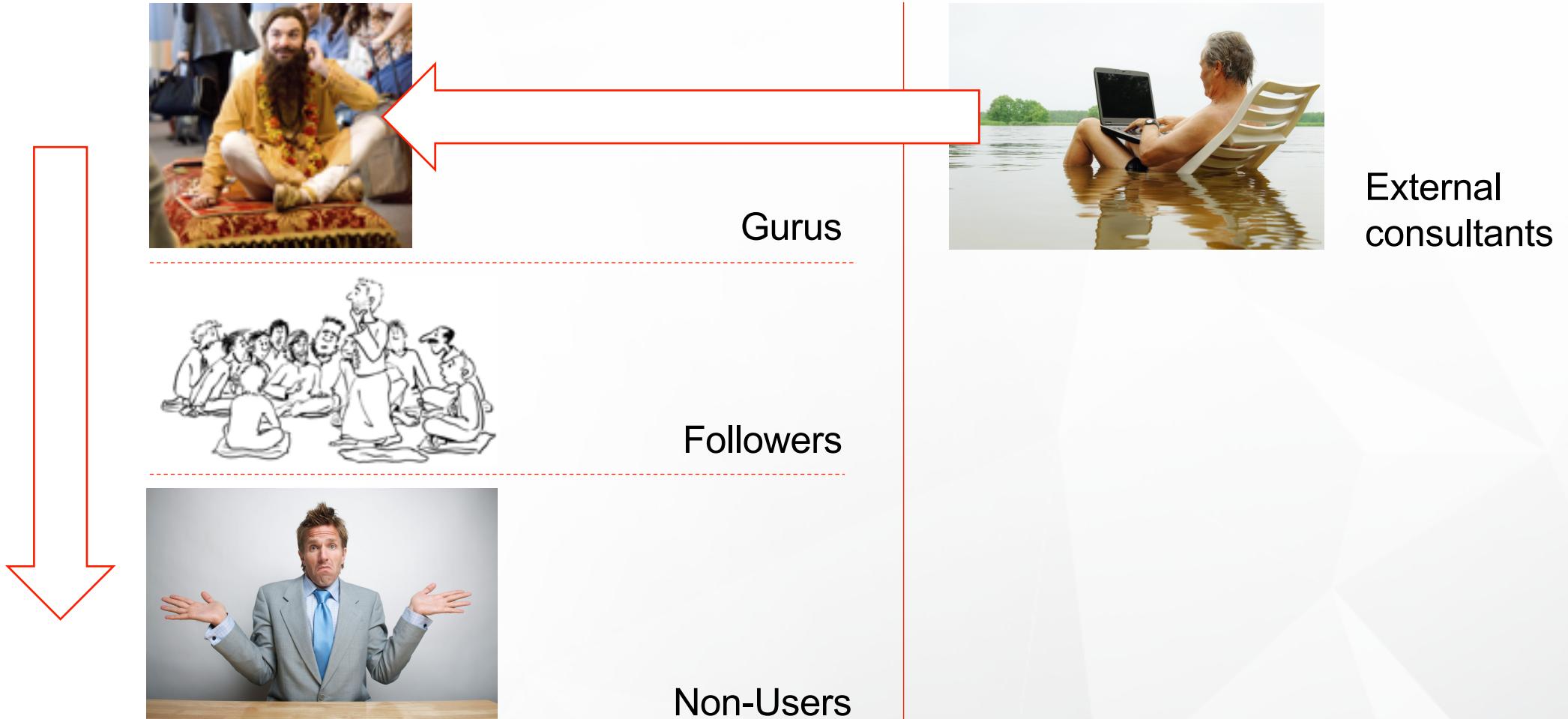
The camunda BPM lifecycle



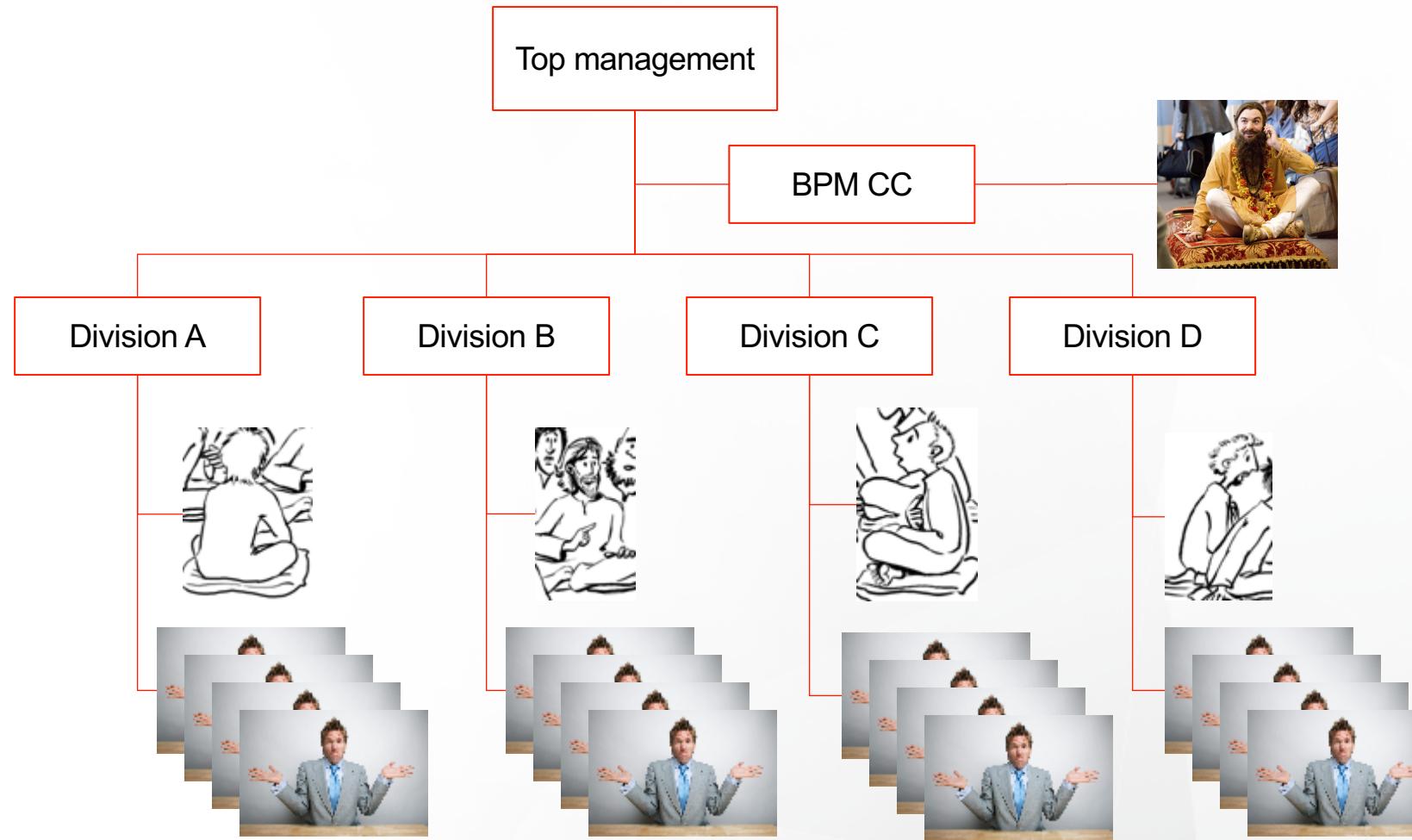
Clarifying goals is critical



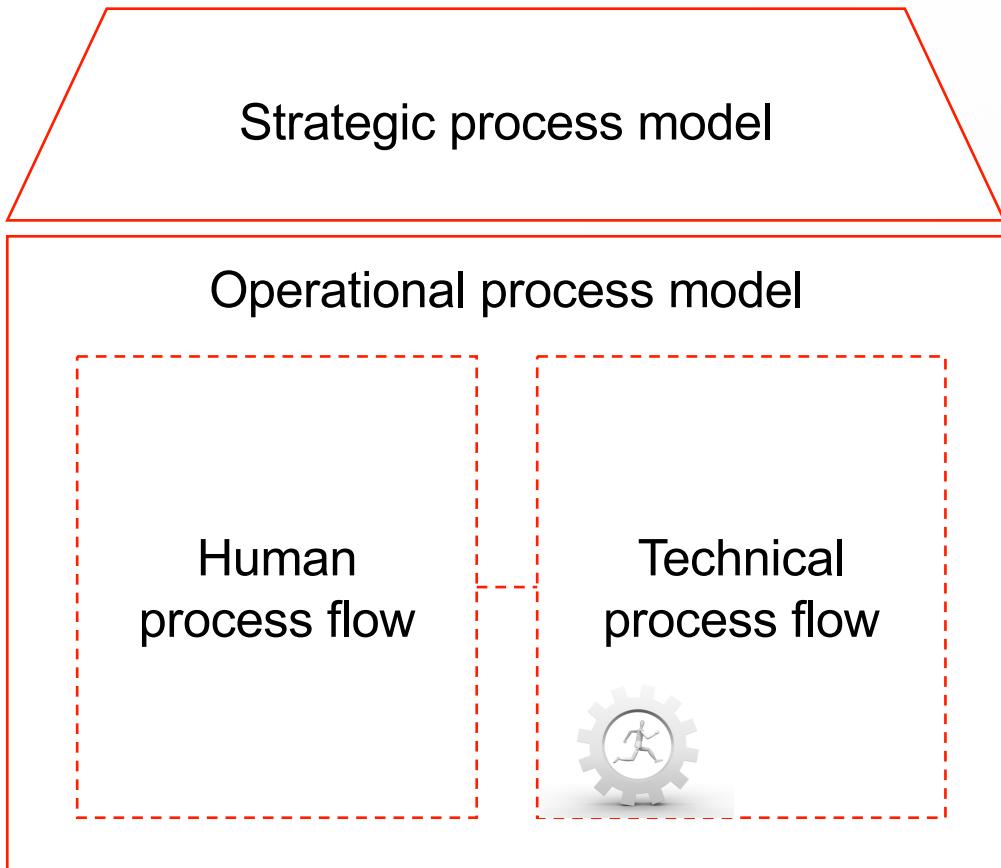
Knowledge allocation for BPMN



Assignment In larger organizations



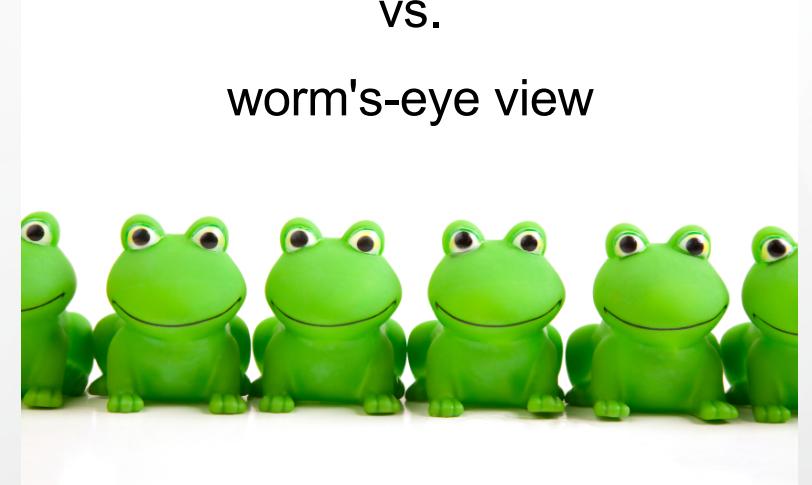
Camunda house



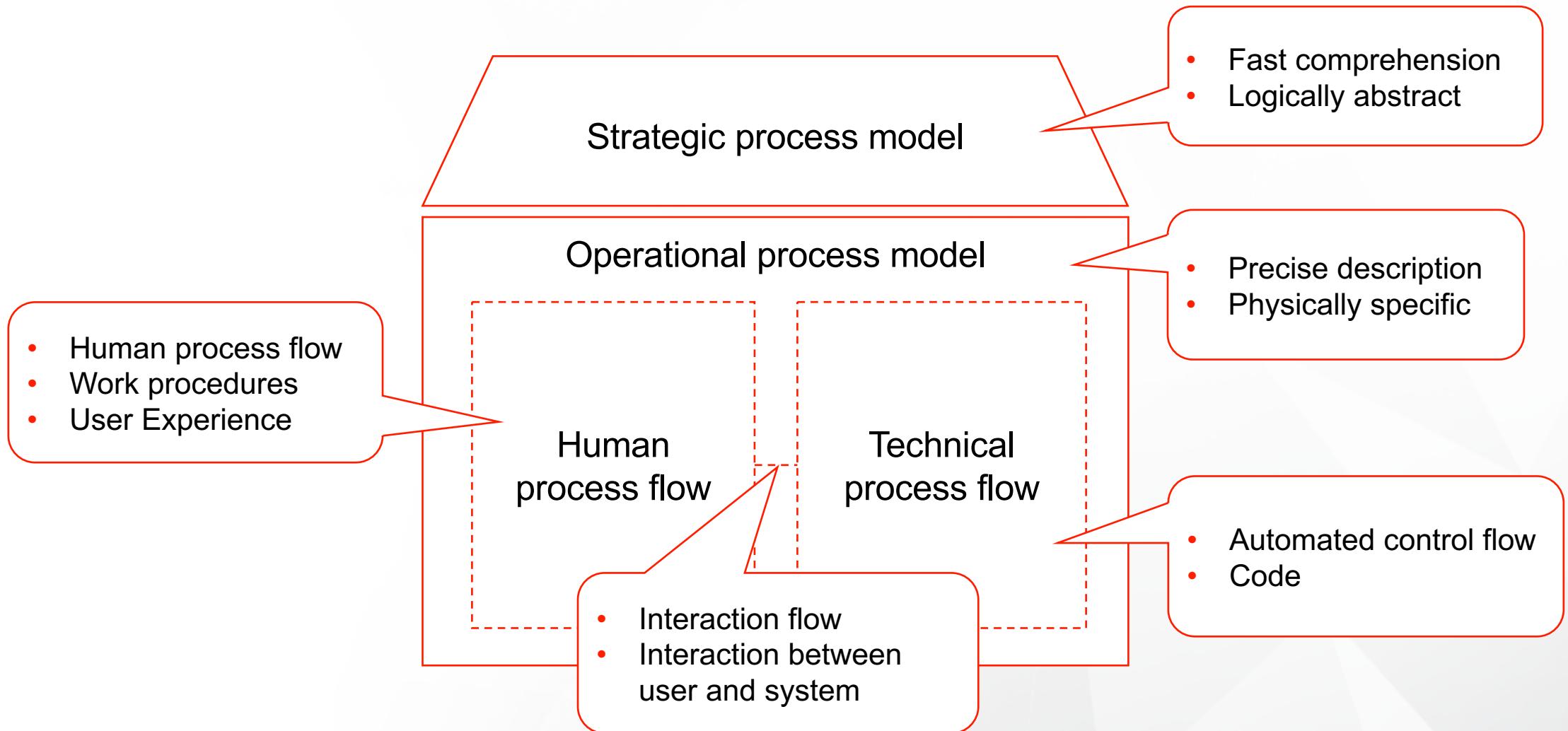
bird's eye view

vs.

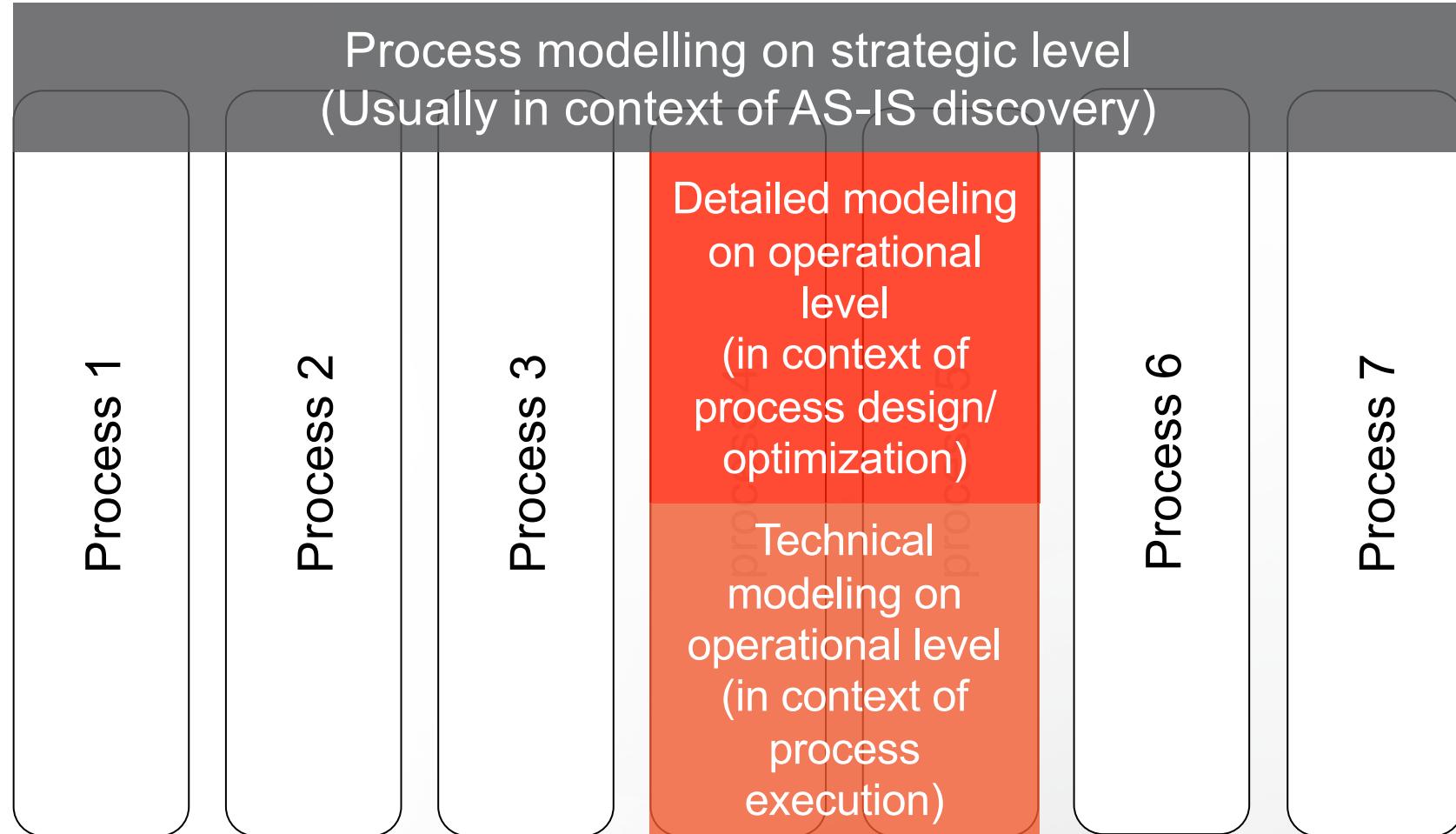
worm's-eye view



Camunda house



Strategic level on a broad scale, operational only if needed

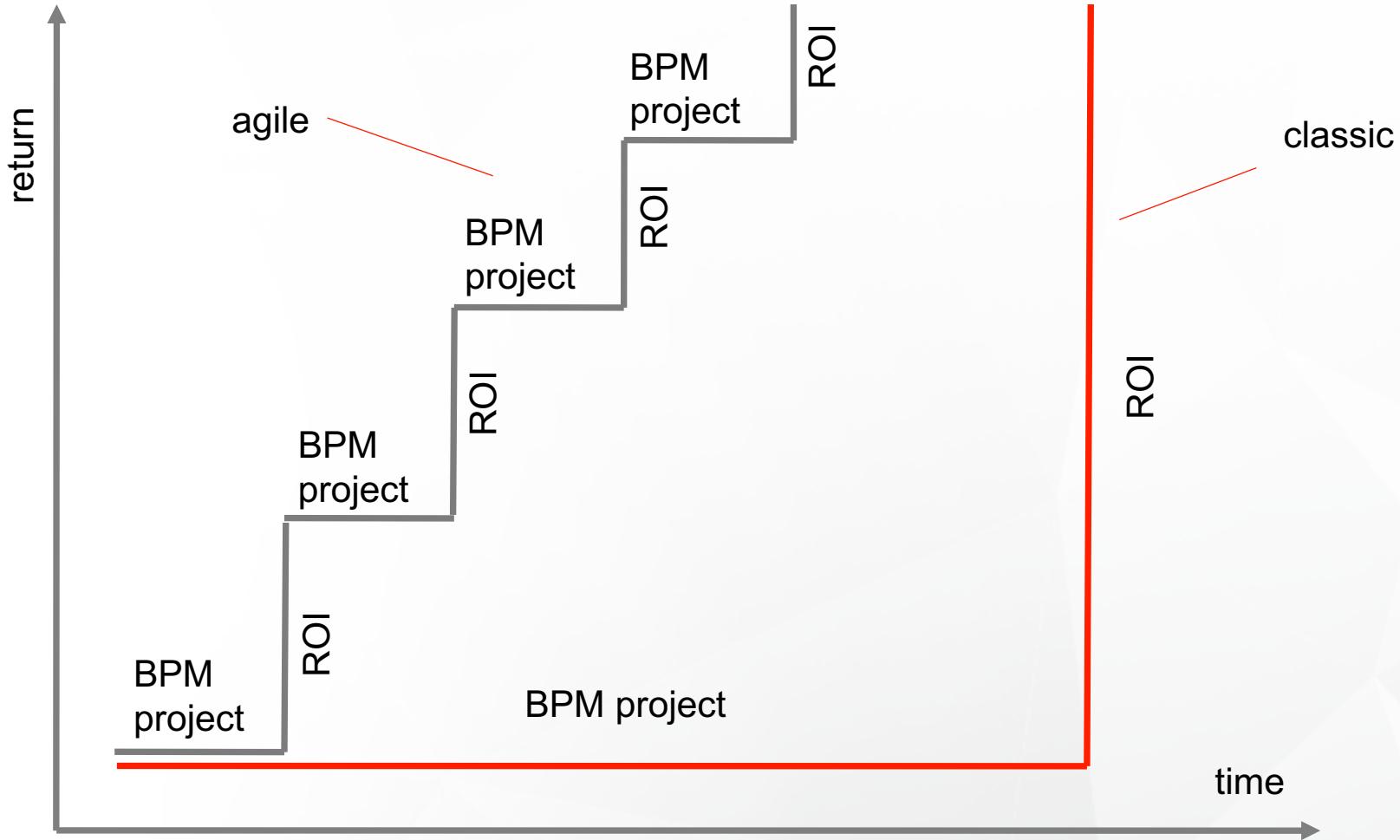


Top-Down-Bottom-Up: continuous improvement

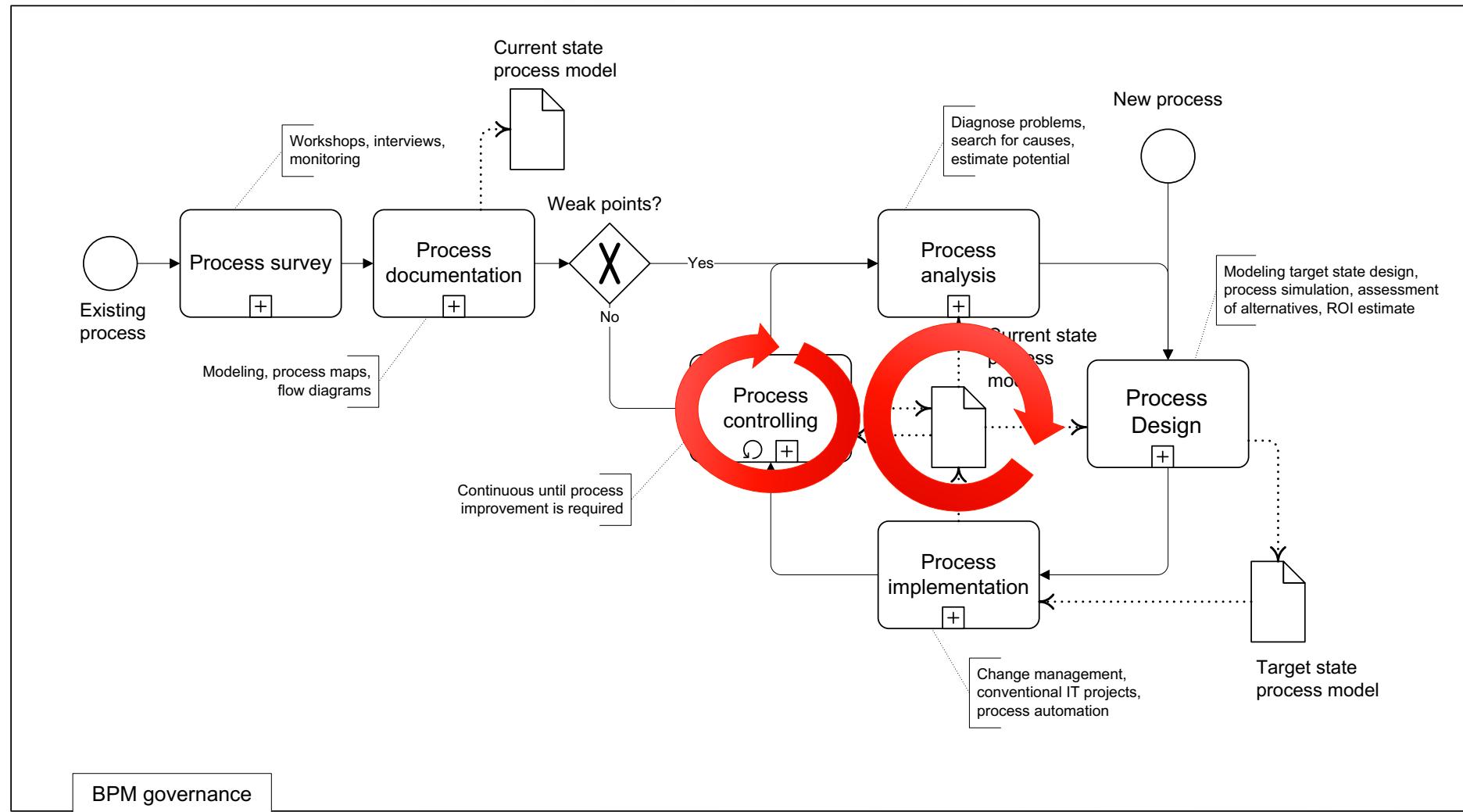


“BPMM Level 3: Wherein common, standardizes processes are synthesized from best practices identified in the work groups [...]”

BPM – agile or classic



CIP in the BPM lifecycle



Success factors for CIP

Modeler

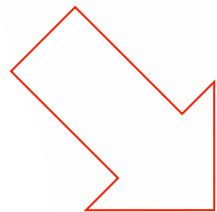
- Correct process models
- Comprehensible process models
- Omnipresent process models

Responsible executive

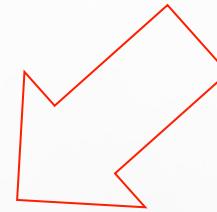
- Known and constant assignee for improvement
- Transparent consideration of suggestions
- Fast and consistent introduction and enforcement of suggestions

The alternative?

Annoyed
divisions and
domain experts



Dissatisfied
modelers



„Dead“
process documentation

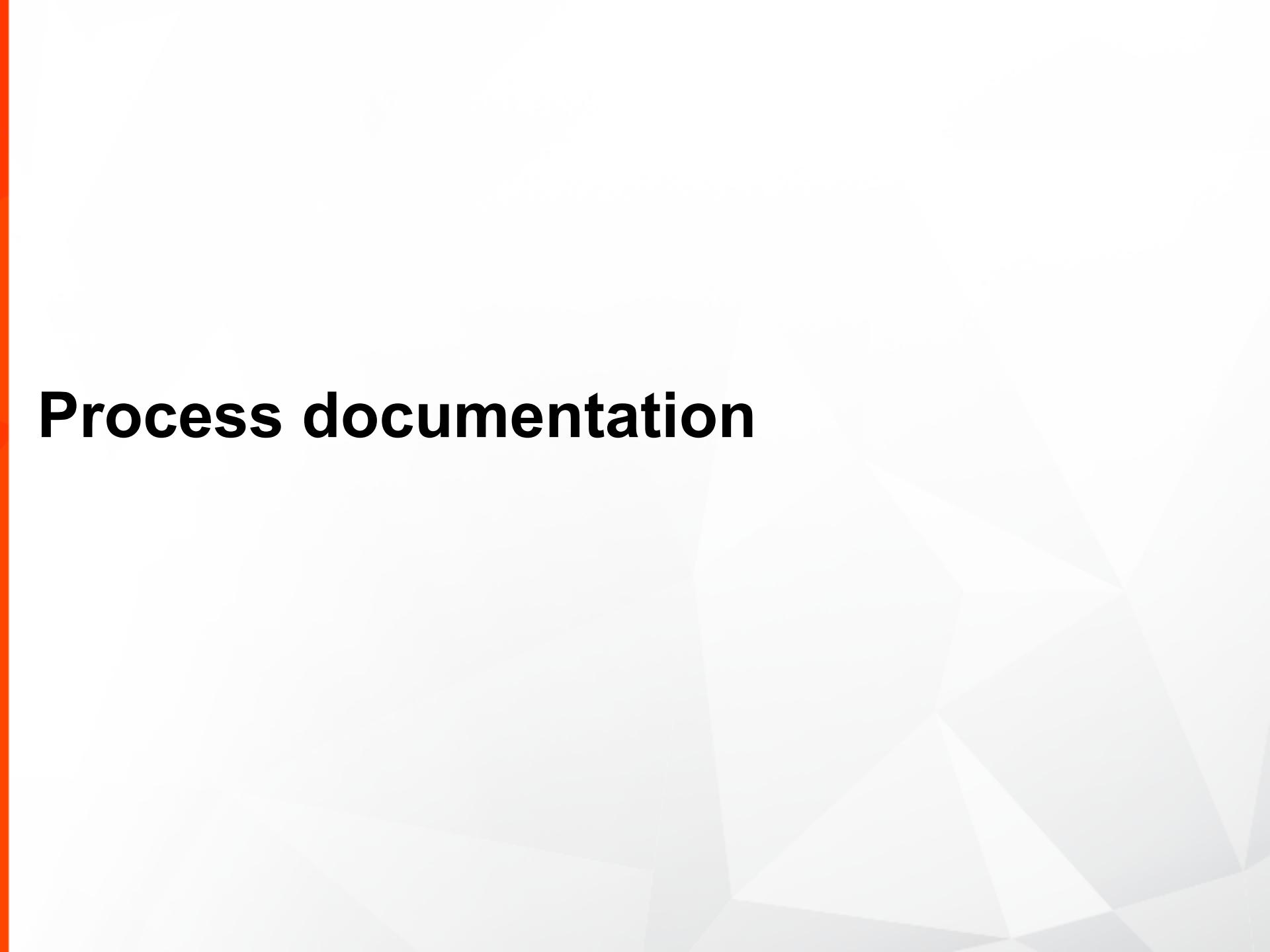


“Tupperware”

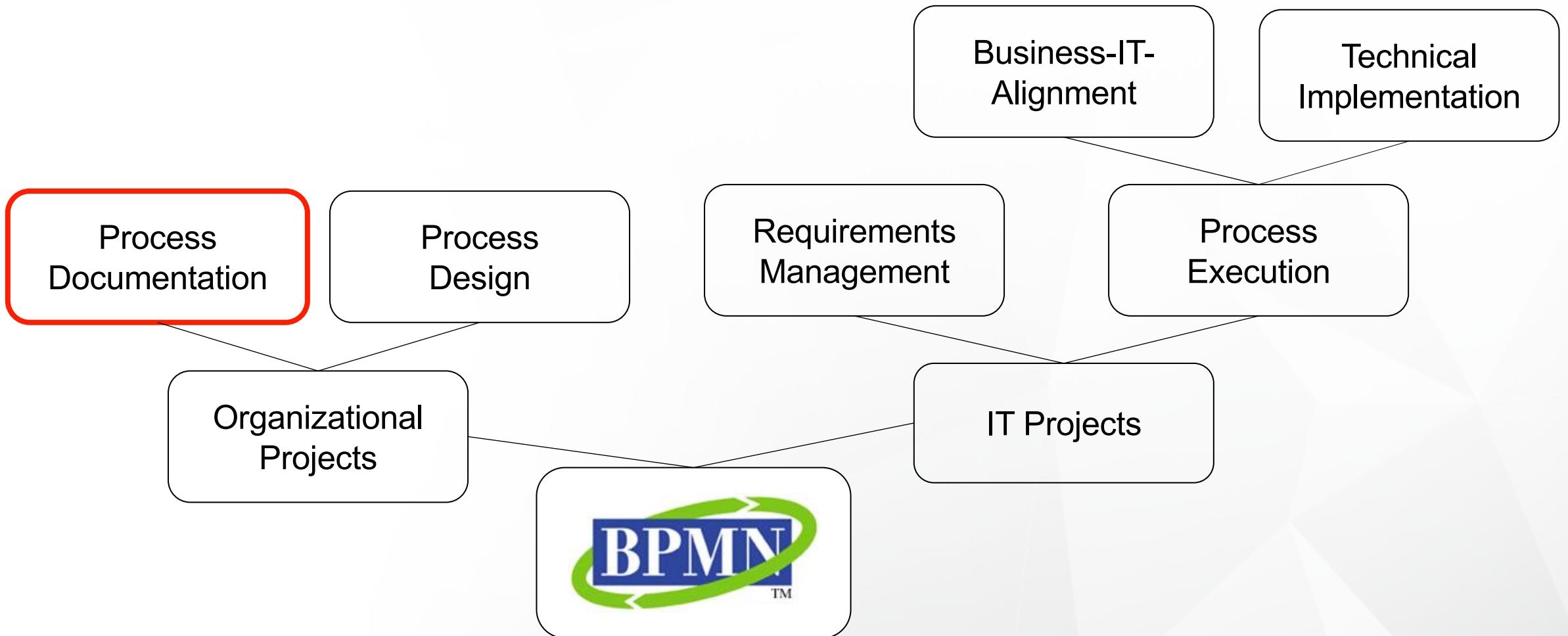
„Yes, there once was a project documenting the processes, but I guess it was canceled for some reason. The process models were only used for the ISO audits anyway.“



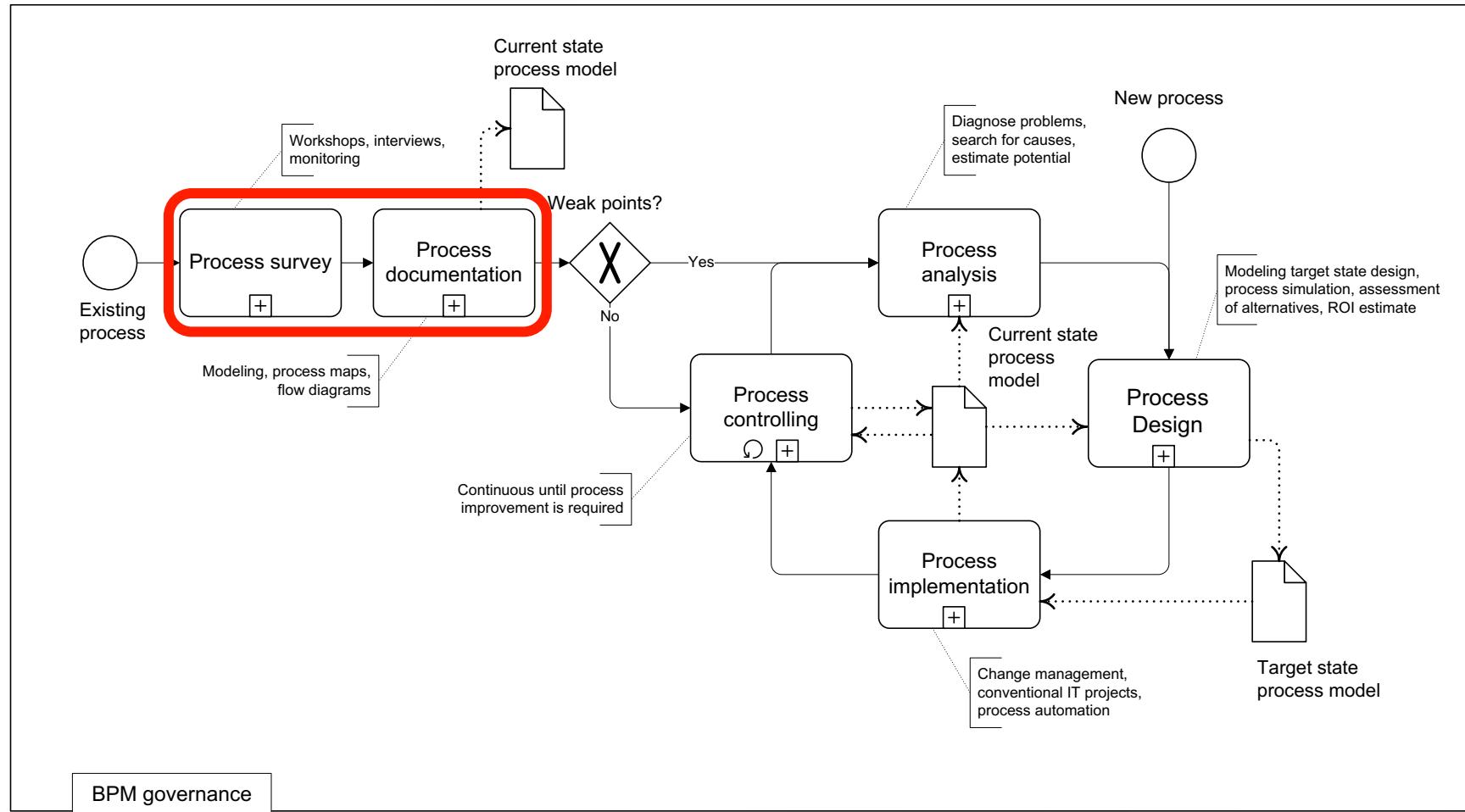
Process documentation



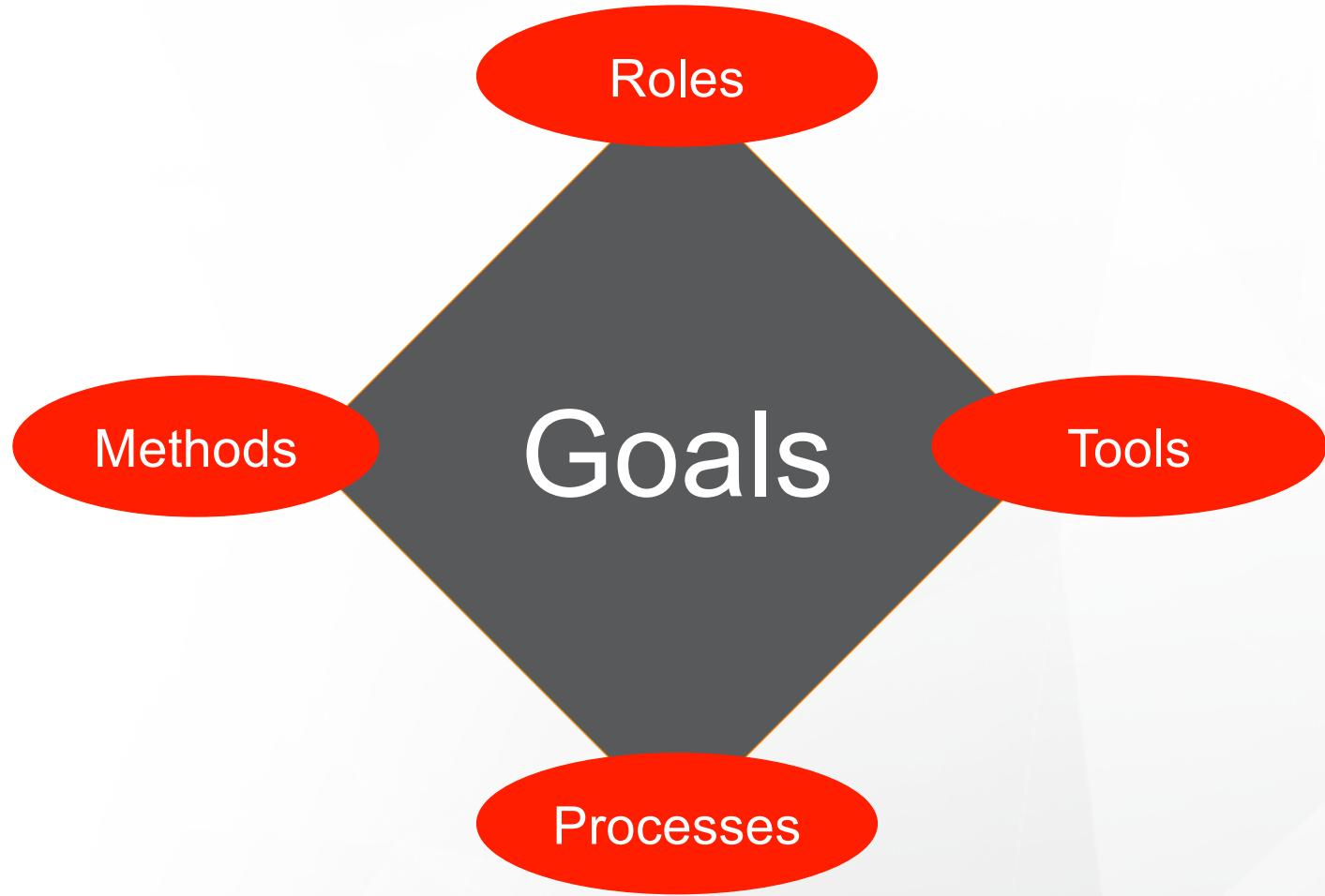
Different application scenarios



The Camunda BPM lifecycle

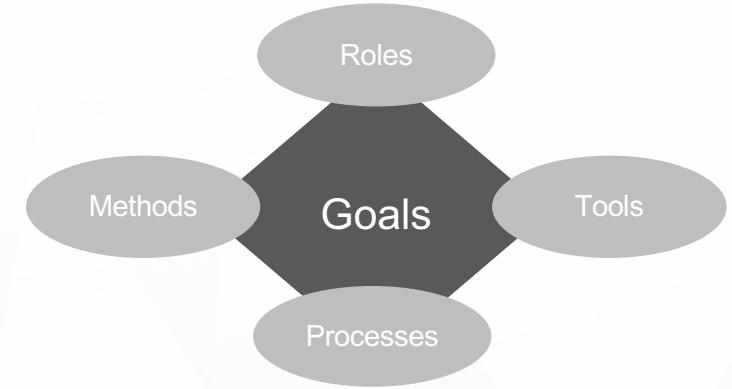


Clarifying goals is critical



Systematic clarification of all aspects: Goals

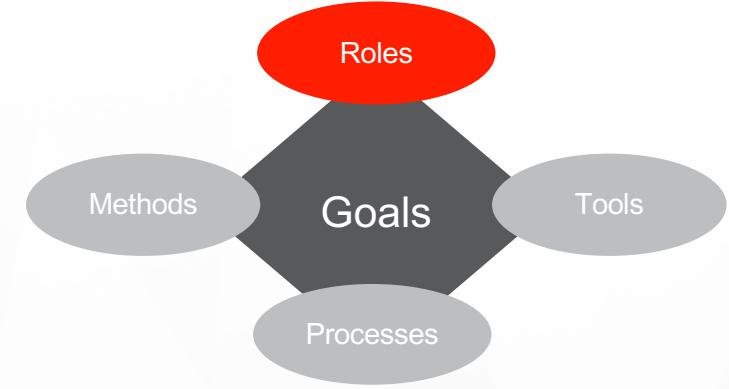
- Achieving visibility of the company processes
- Increase the comprehensibility of processes for employees
- Achieving an ISO certification



Systematic clarification of all aspects: Roles

Center of excellence (BPM CoE)

- Define the methodology, tools, and meta-processes
- Support the departments (decentralized teams)
- Establish QA

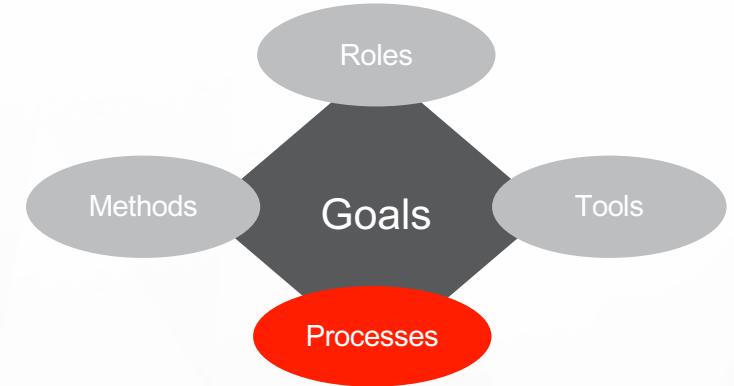


Departments:

- Dezentralized documentation (business analysis)

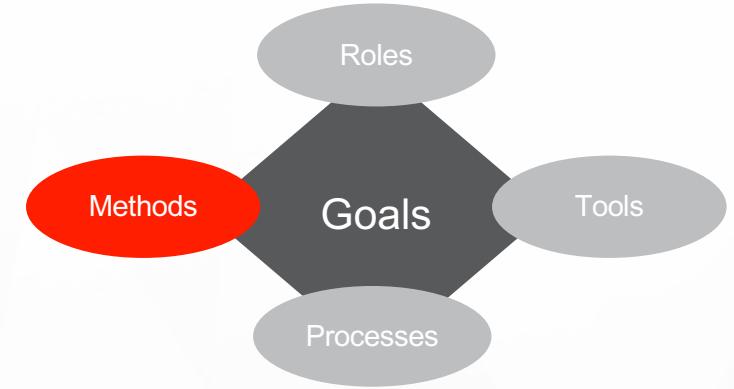
Systematic clarification of all aspects: Processes

- Process discovery and documentation
- Maintenance and housekeeping of the model repository
- Approval of process models (based on content / based on underlying methodology)



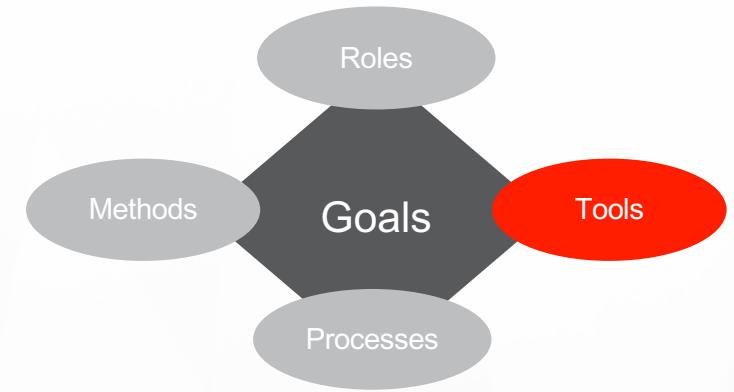
Systematic clarification of all aspects: Methods

- BPMN + Guidelines (conventions)
- Define the process architecture (e.g., process landscapes)
- Textual descriptions

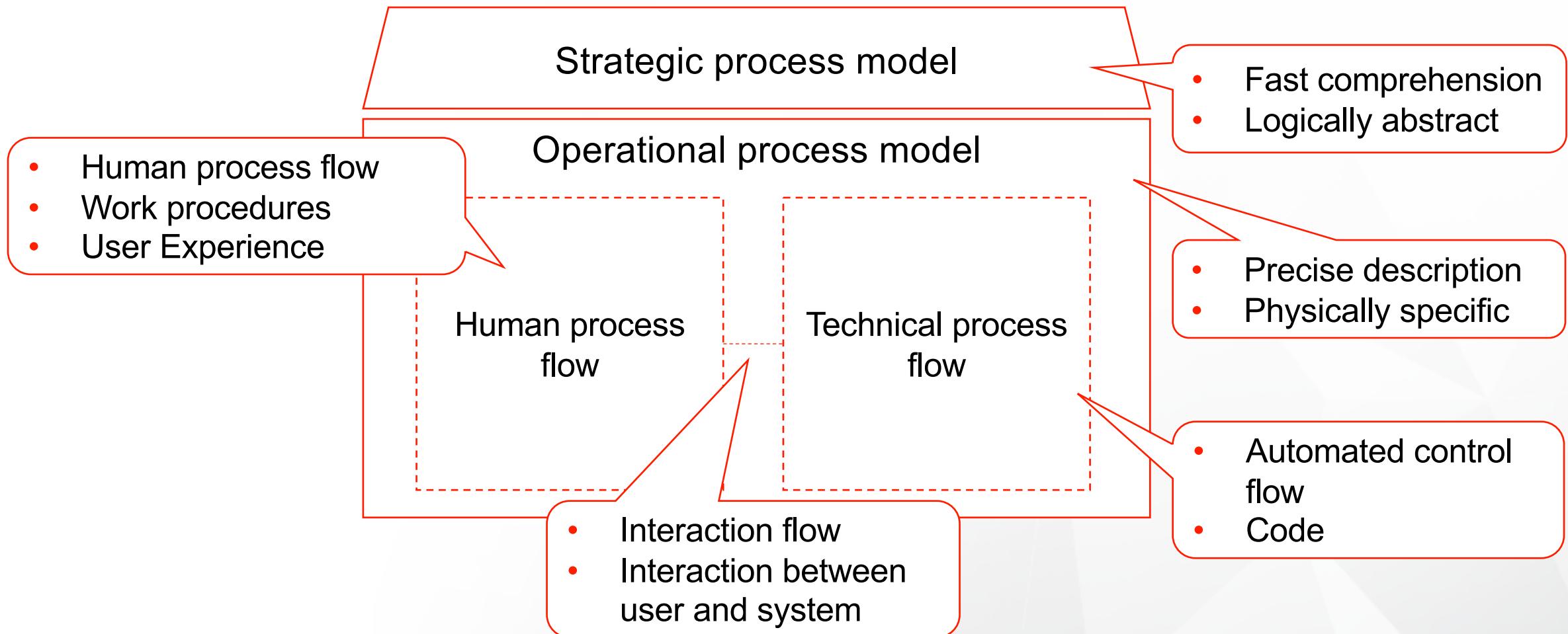


Systematic clarification of all aspects: Tools

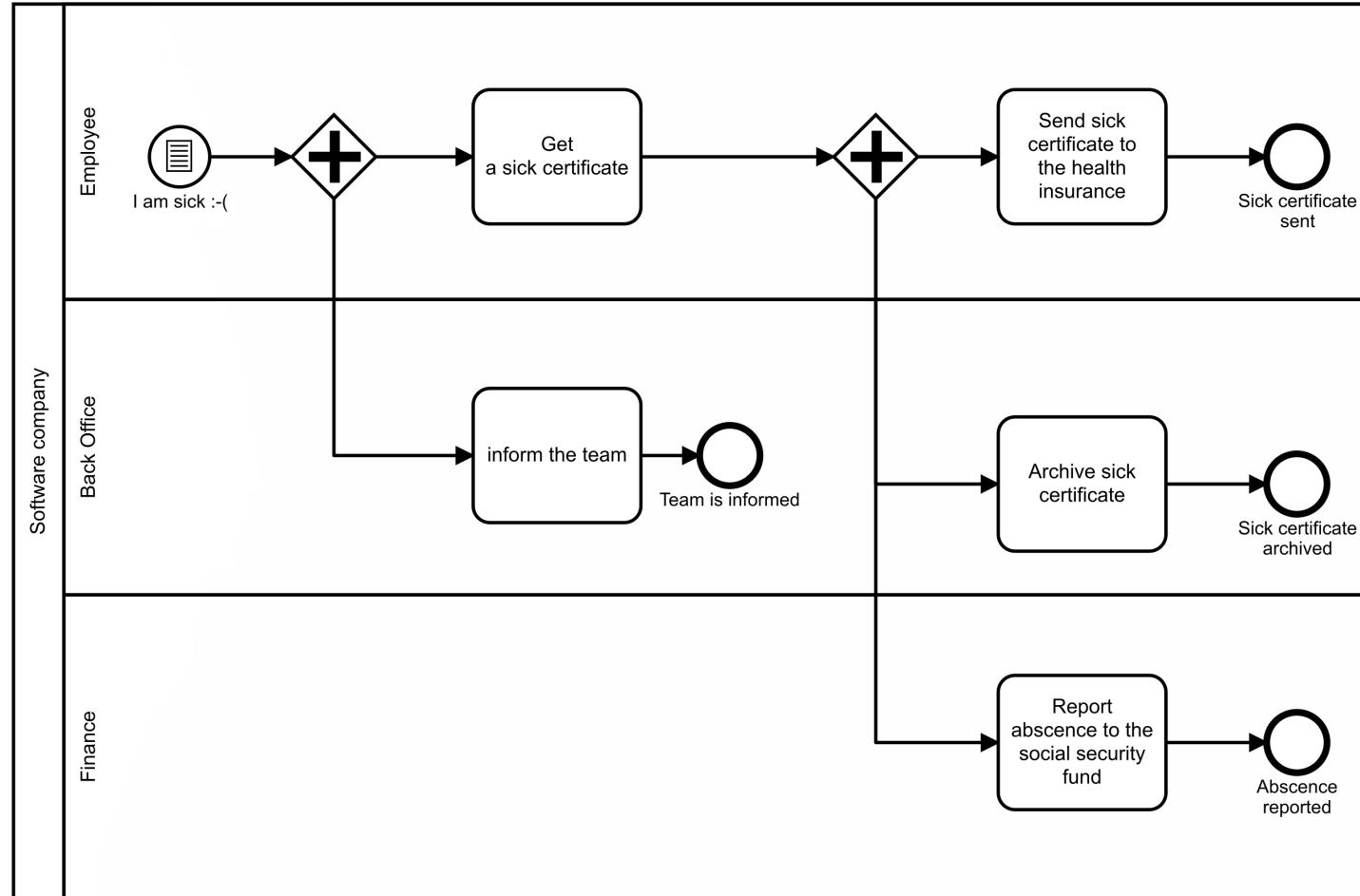
- Collaborative BPMN-modeling
- Support for administration of extensive documentation
- Support for customization (e.g., custom artifacts and attributes)
- Integration of the BPMN tool with other business applications (e.g. Wiki, Sharepoint)



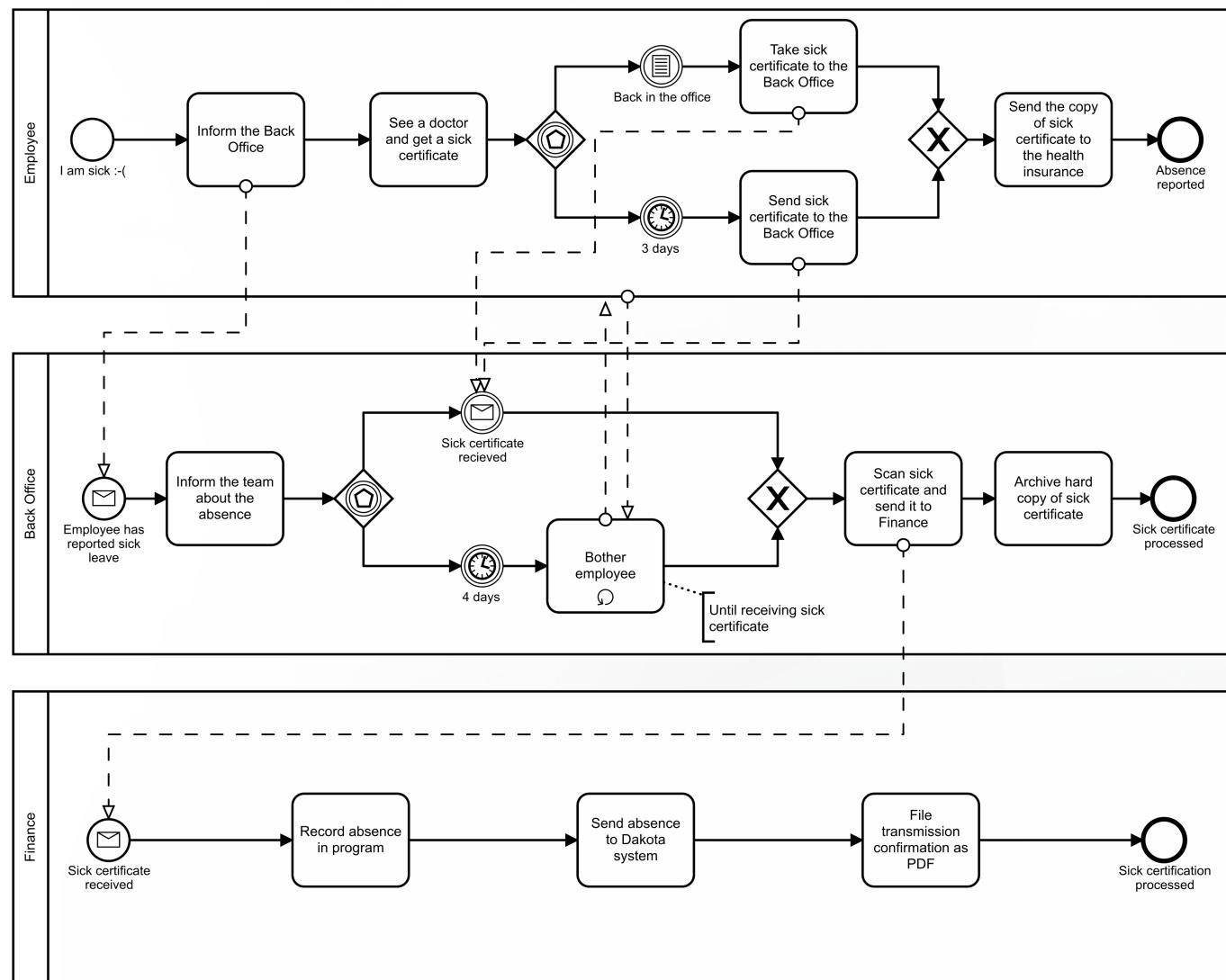
Relation to the Camunda House



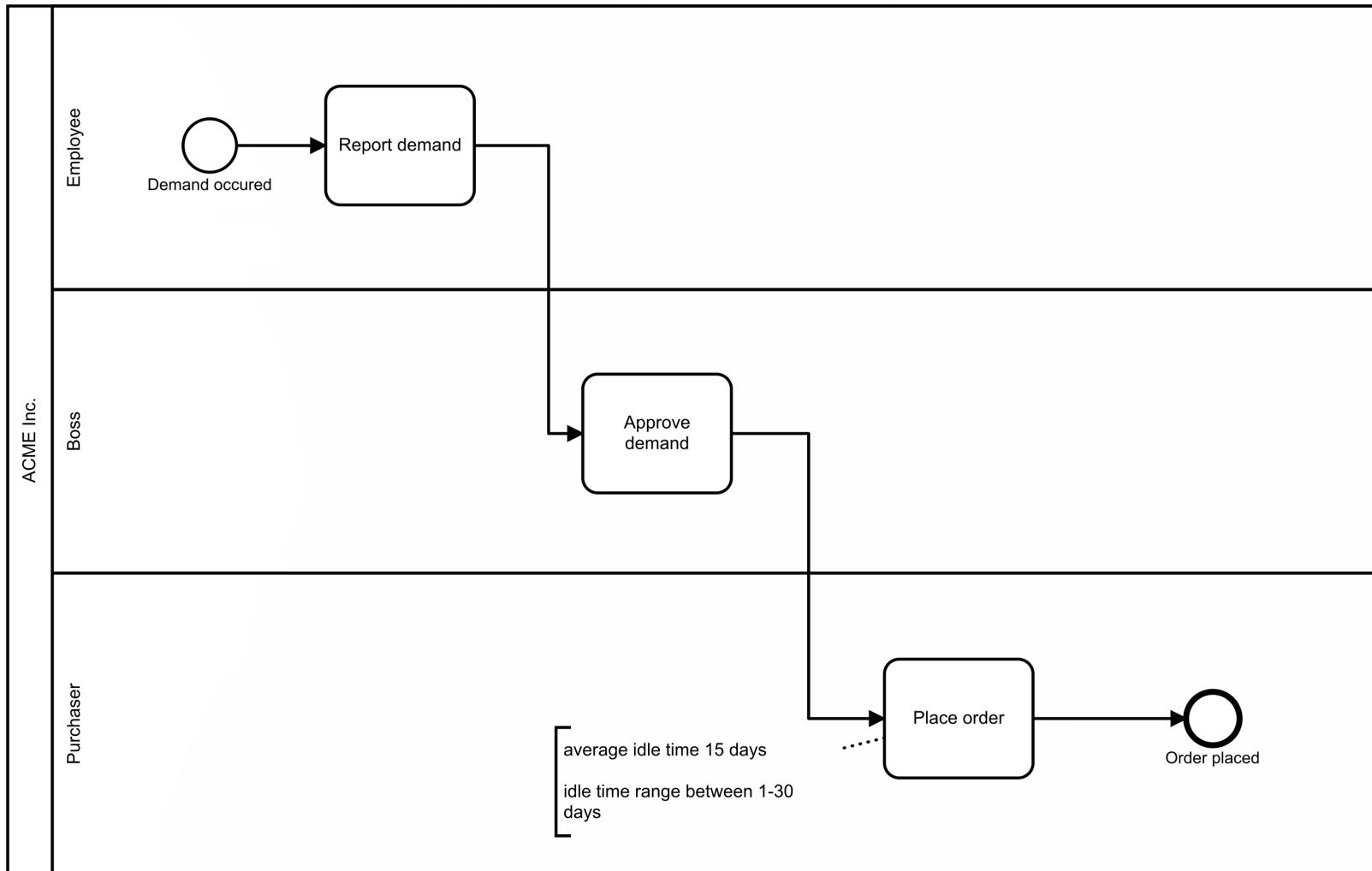
„Sick leave process“ at strategic level



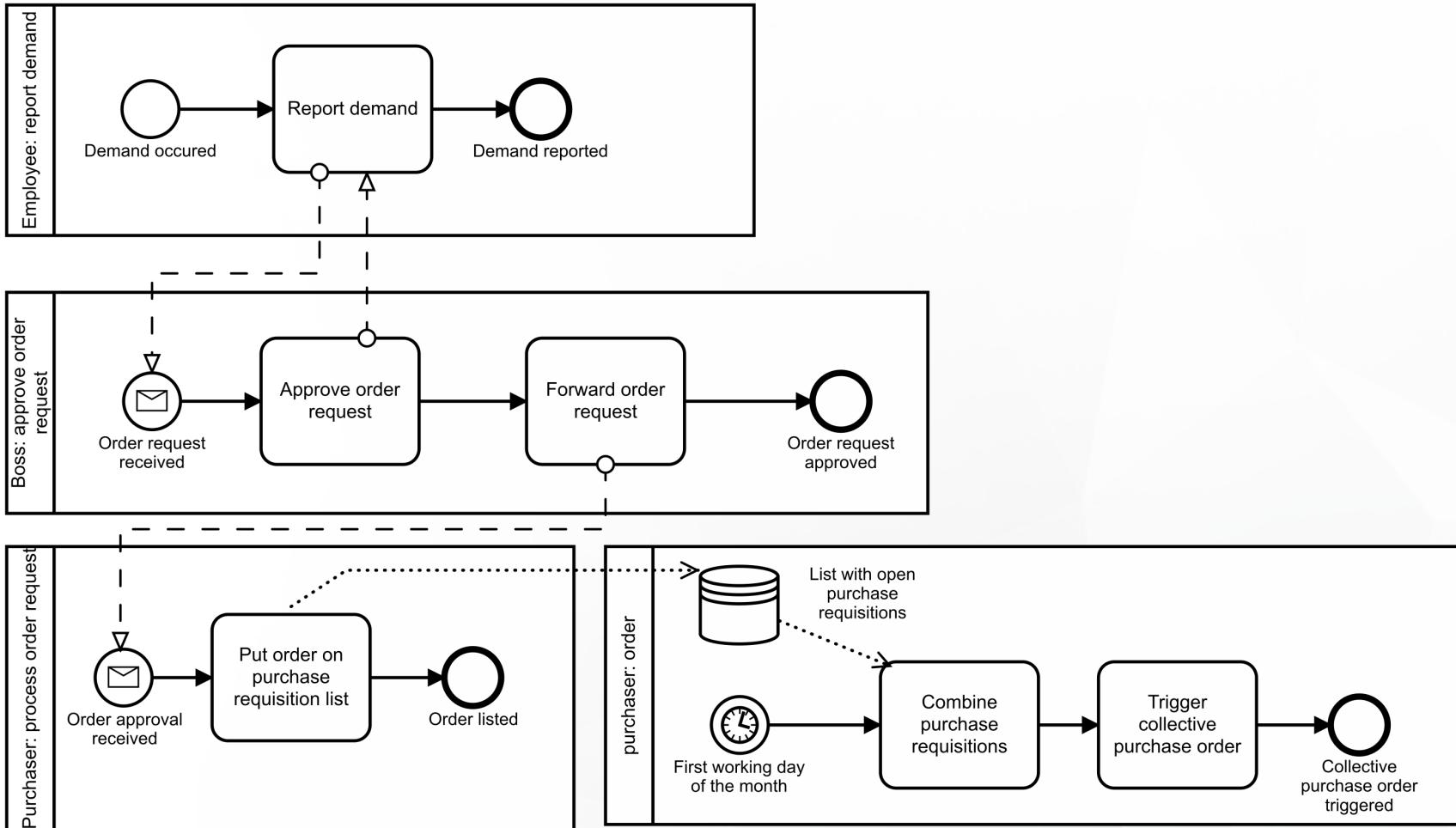
„Sick leave process“ at operational level



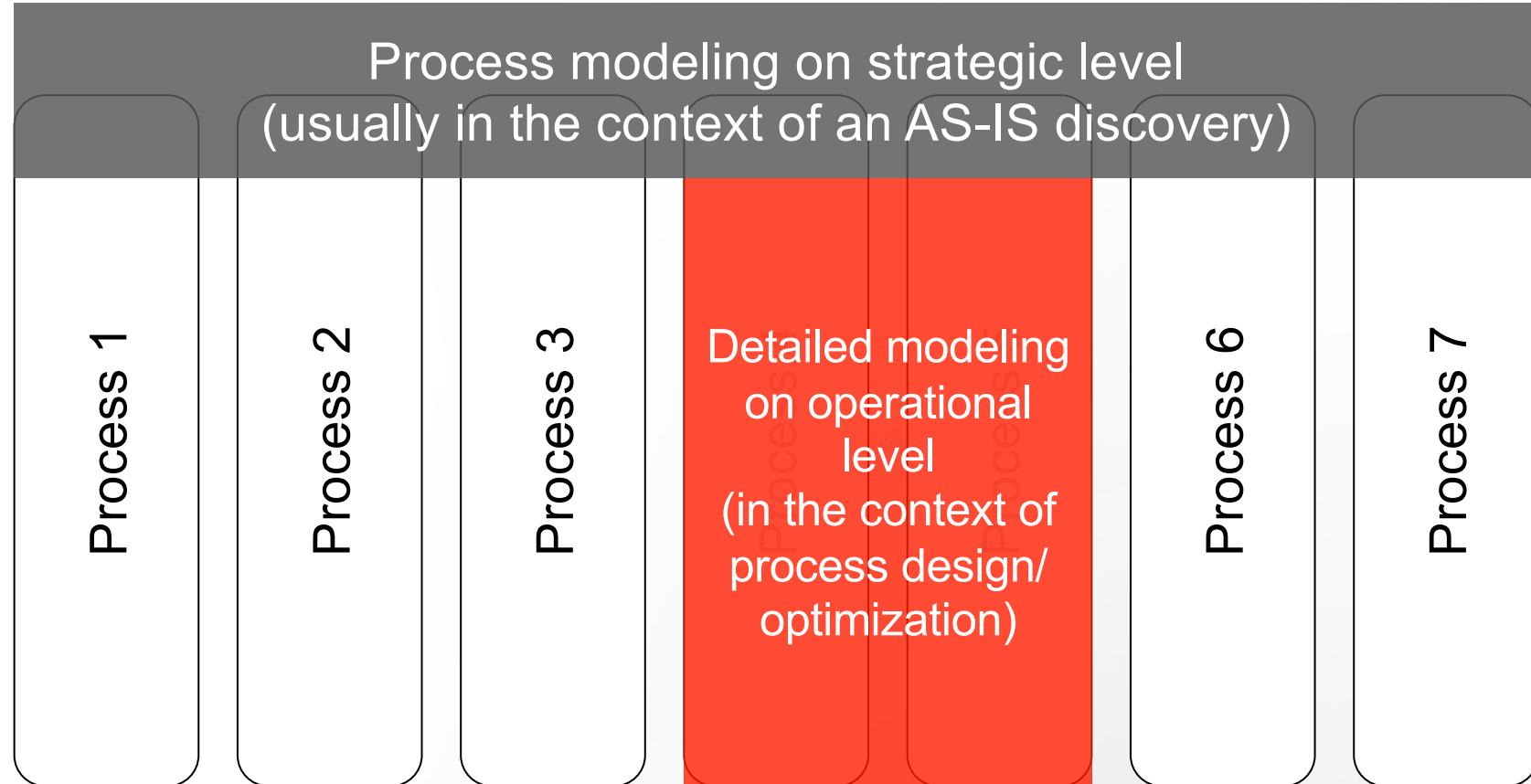
Procurement process at strategic level



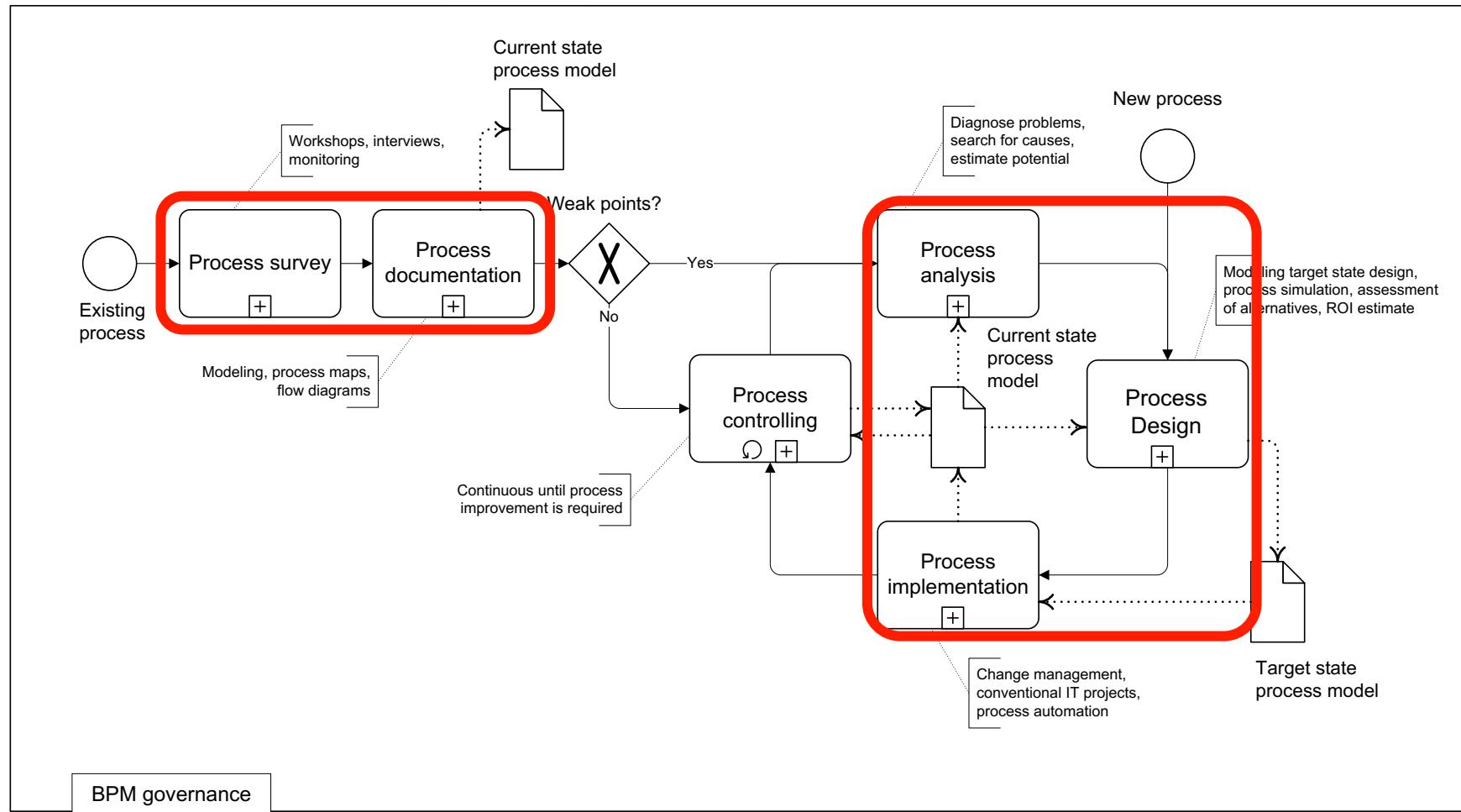
Procurement process at strategic level



Strategic level on a broad scale, operational only if needed



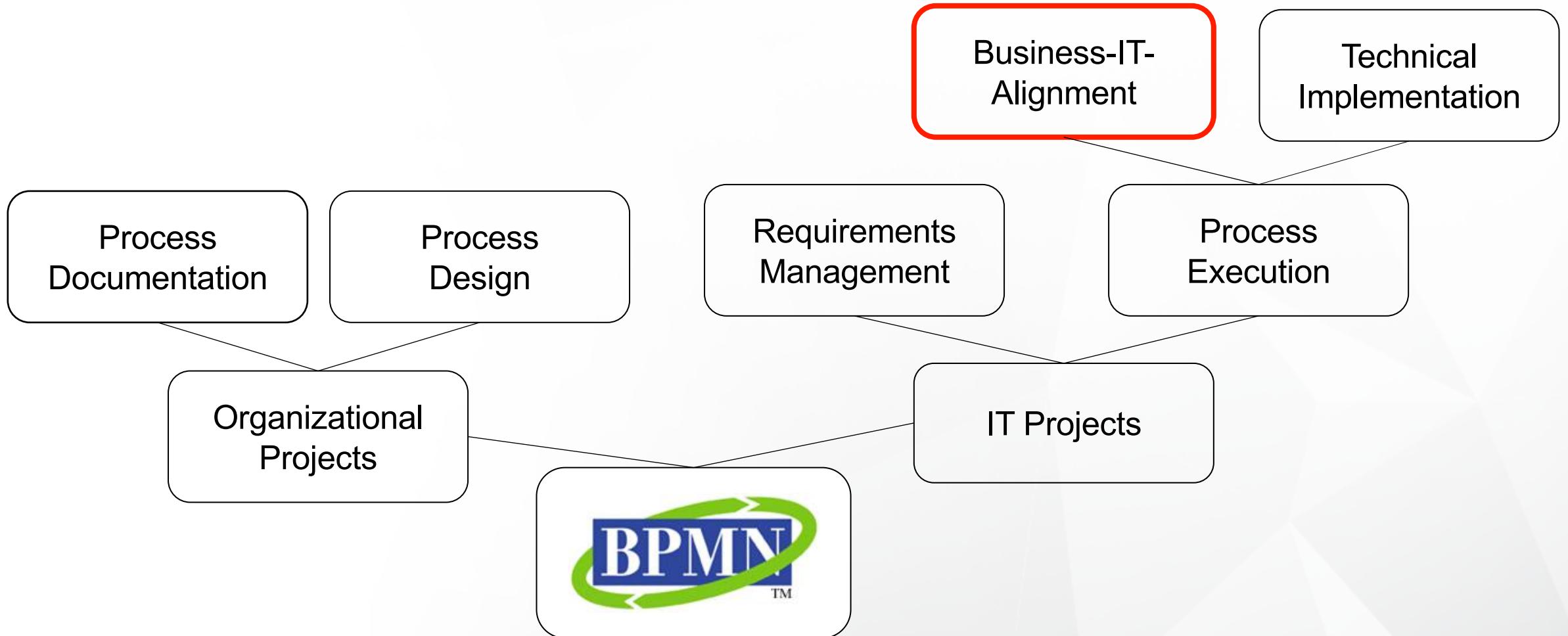
The Camunda BPM lifecycle



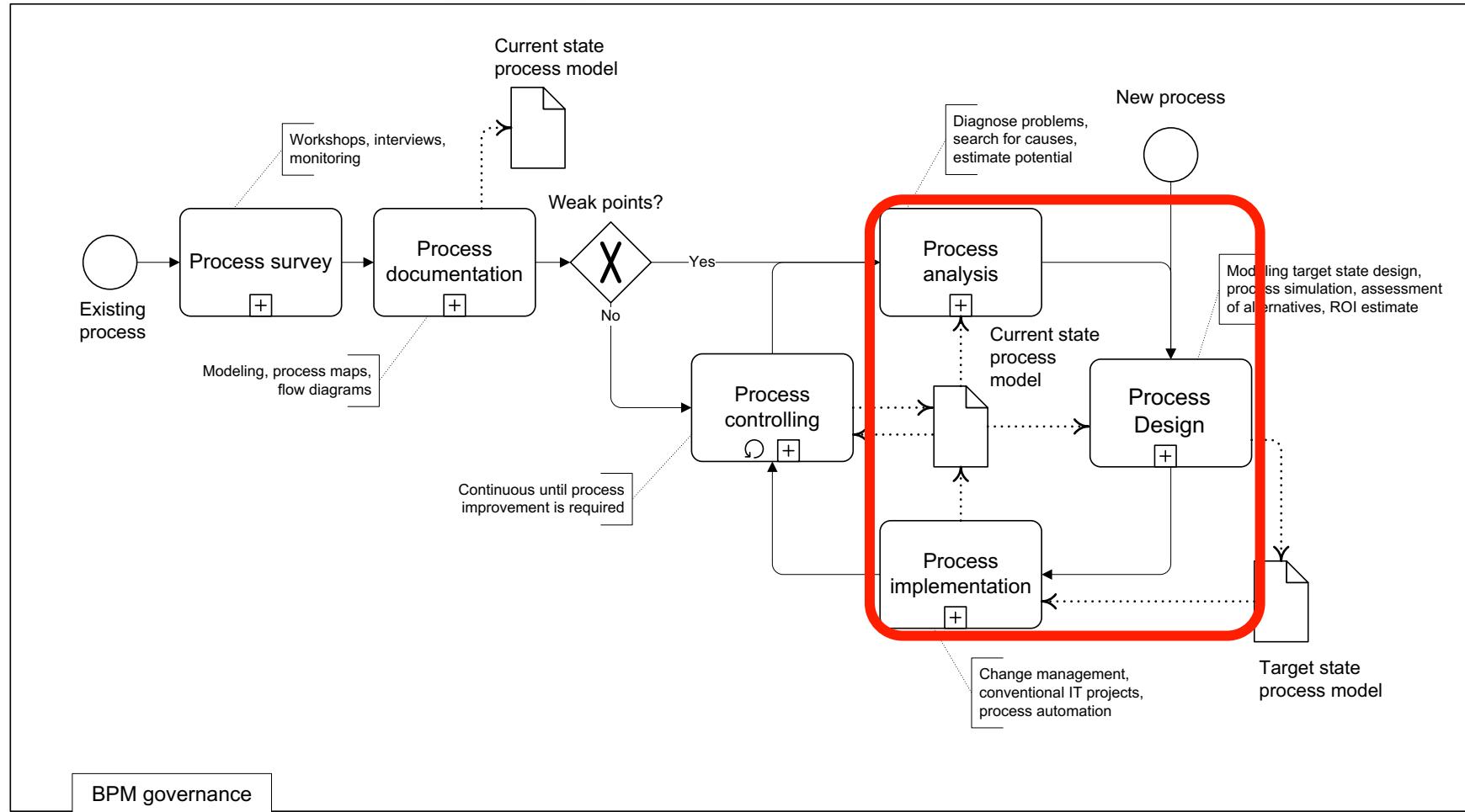


Business-IT-Alignment

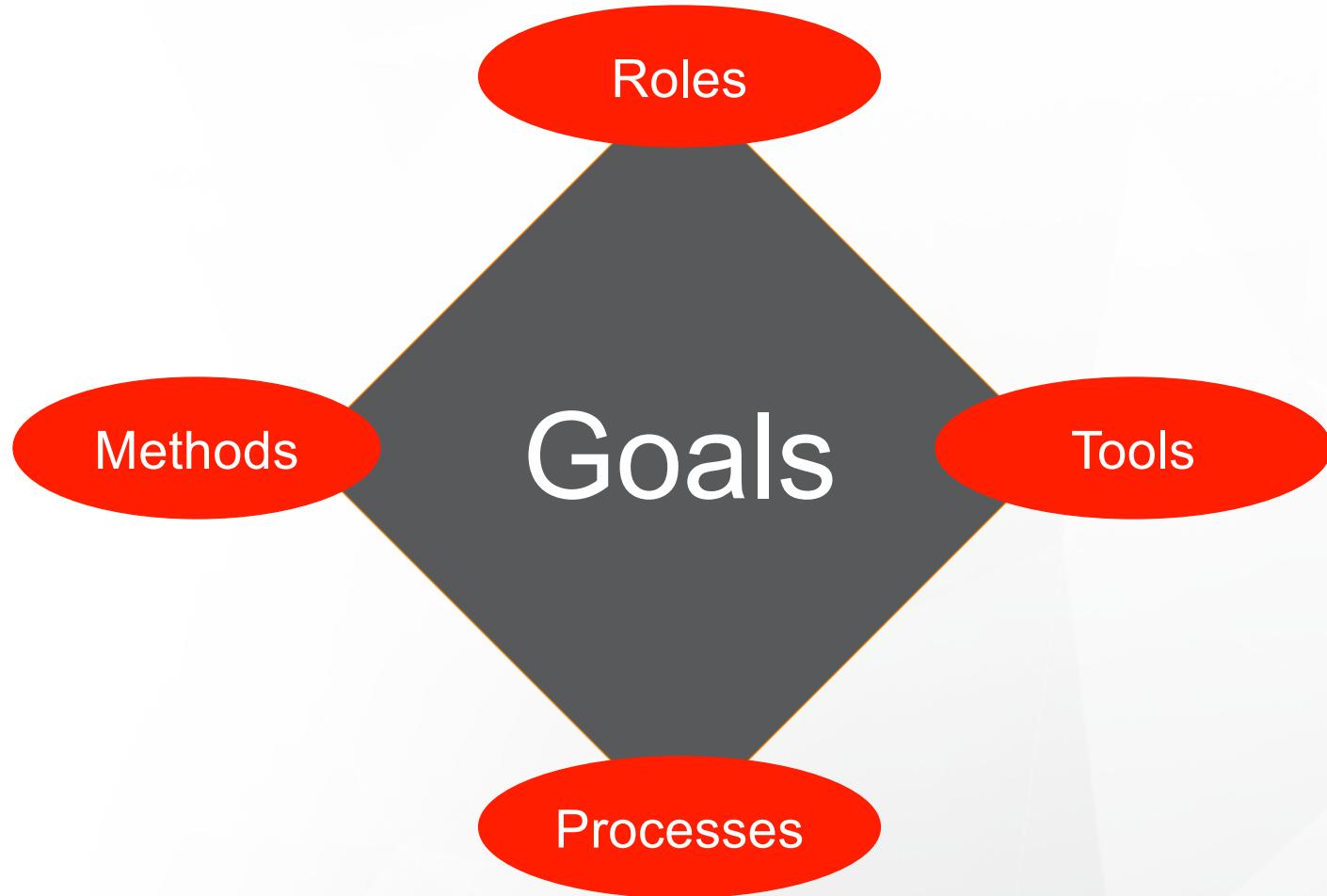
Different application scenarios



The Camunda BPM lifecycle

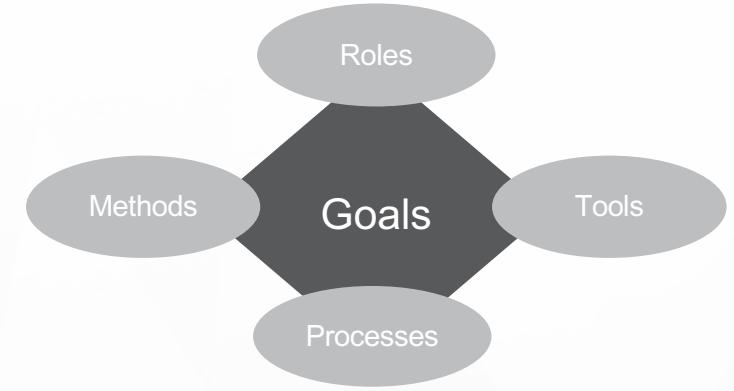


Systematic clarification of all aspects



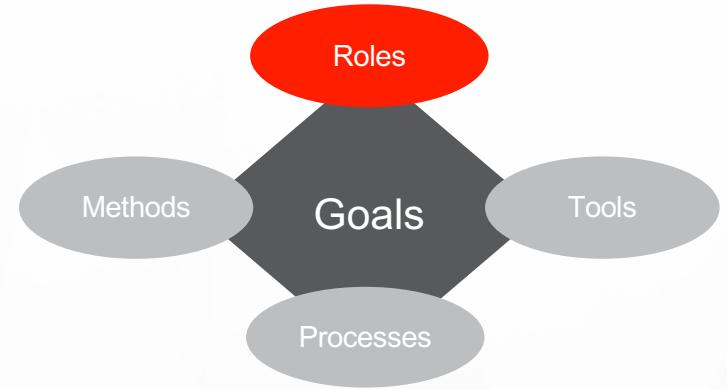
Systematic clarification of all aspects: Goals

- Generating the business and IT requirements
- Process execution with a process engine
- Achieving Business-IT-alignment



Systematic clarification of all aspects: Roles

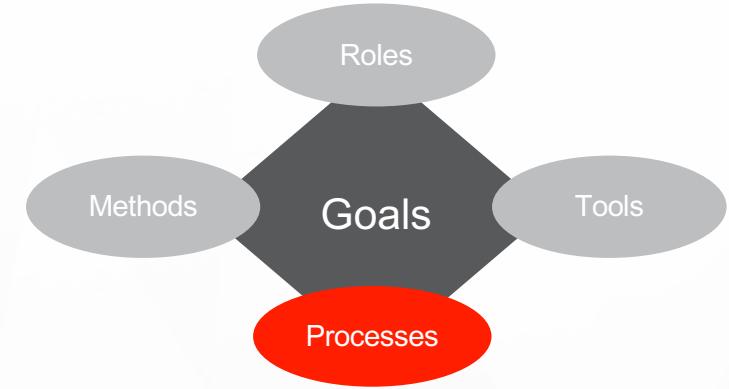
- Centre of excellence (BPM CoE)
 - Define the methodology, tools, and meta-processes
 - Support of the departments (decentralized teams)
 - Establish QA
- Departments:
 - Dezentralized documentation (business analysis)
- IT:
 - Requirements engineering by IT experts
 - Technincal implementation of the requirements by IT experts



Roles: possibly setting up a BPM Center of Excellence, Subject Matter Experts, Process Modelers, Process Engineers, Technical Consultants, Quality Assurance People

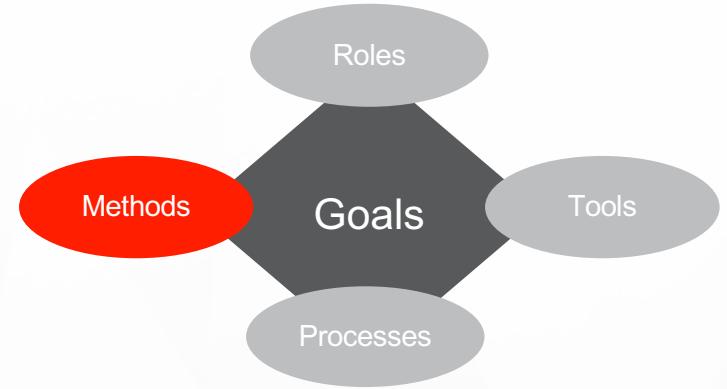
Systematic clarification of all aspects: Processes

- Process discovery and documentation
- Maintenance and housekeeping of the model repository
- Approval of process models (based on content / based on underlying methodology)
- Software development, distribution, tests



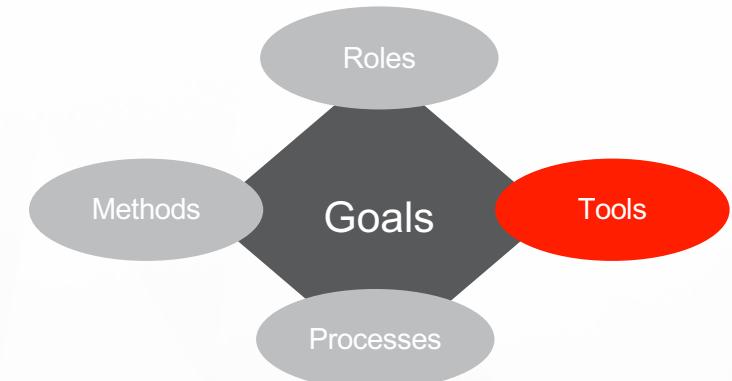
Systematic clarification of all aspects: Methods

- BPMN
- Define the process architecture (e.g., process landscapes)
- Textual descriptions
- Guidelines (conventions) for BPMN models and source code
- Requirements engineering, software and system architecture
- UI/UX- Design

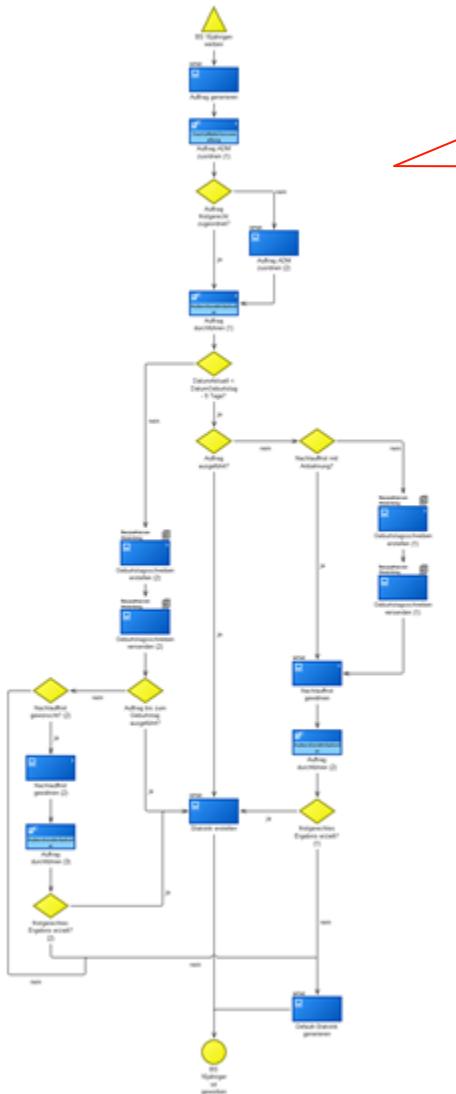


Systematic clarification of all aspects: Tools

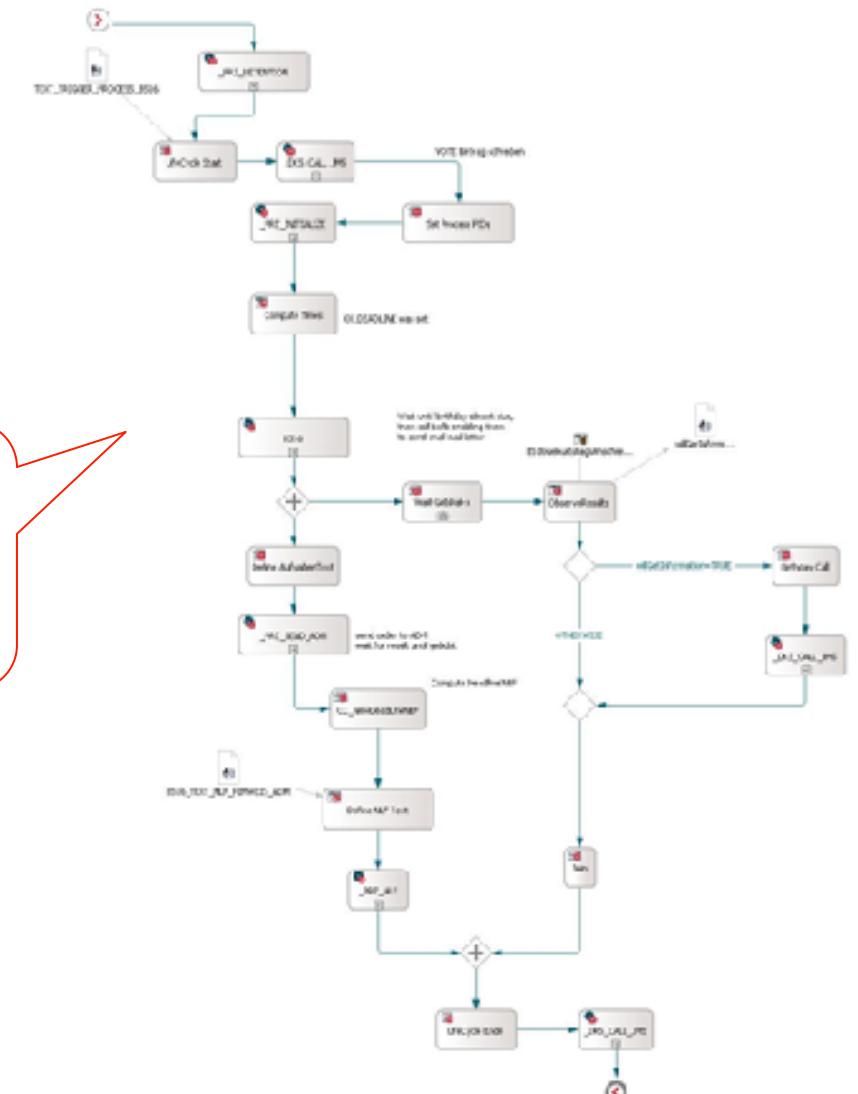
- Collaborative BPMN- modeling
- Support for administration of extensive documentation
- Support for customization (e.g, custom artifacts and attributes)
- Integration of the BPMN tool with other business applications (e.g. Wiki, Sharepoint)
- Process engine for the execution of technical process models



A customer's AS-IS situation

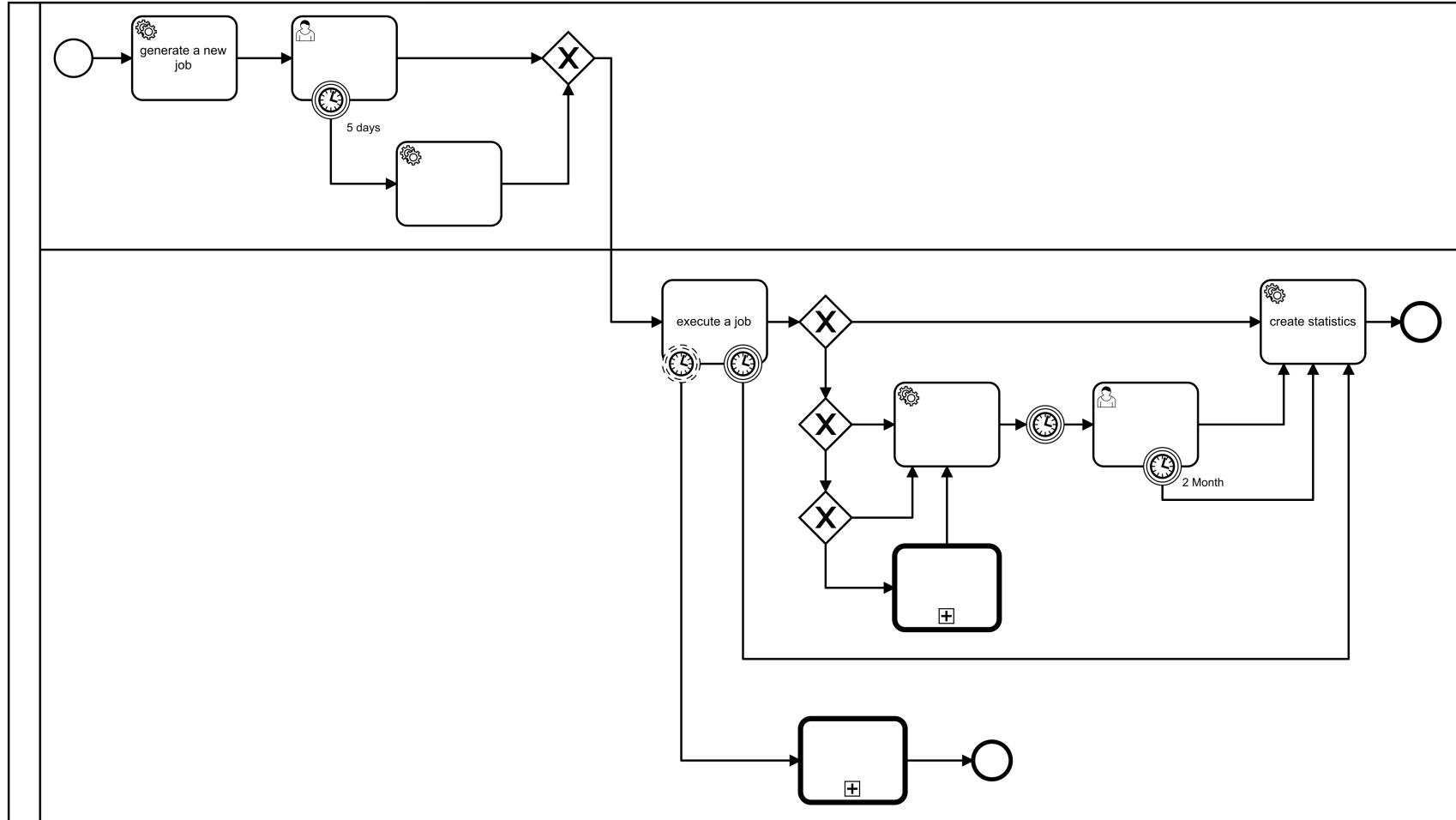


Analytical model (ADONIS)

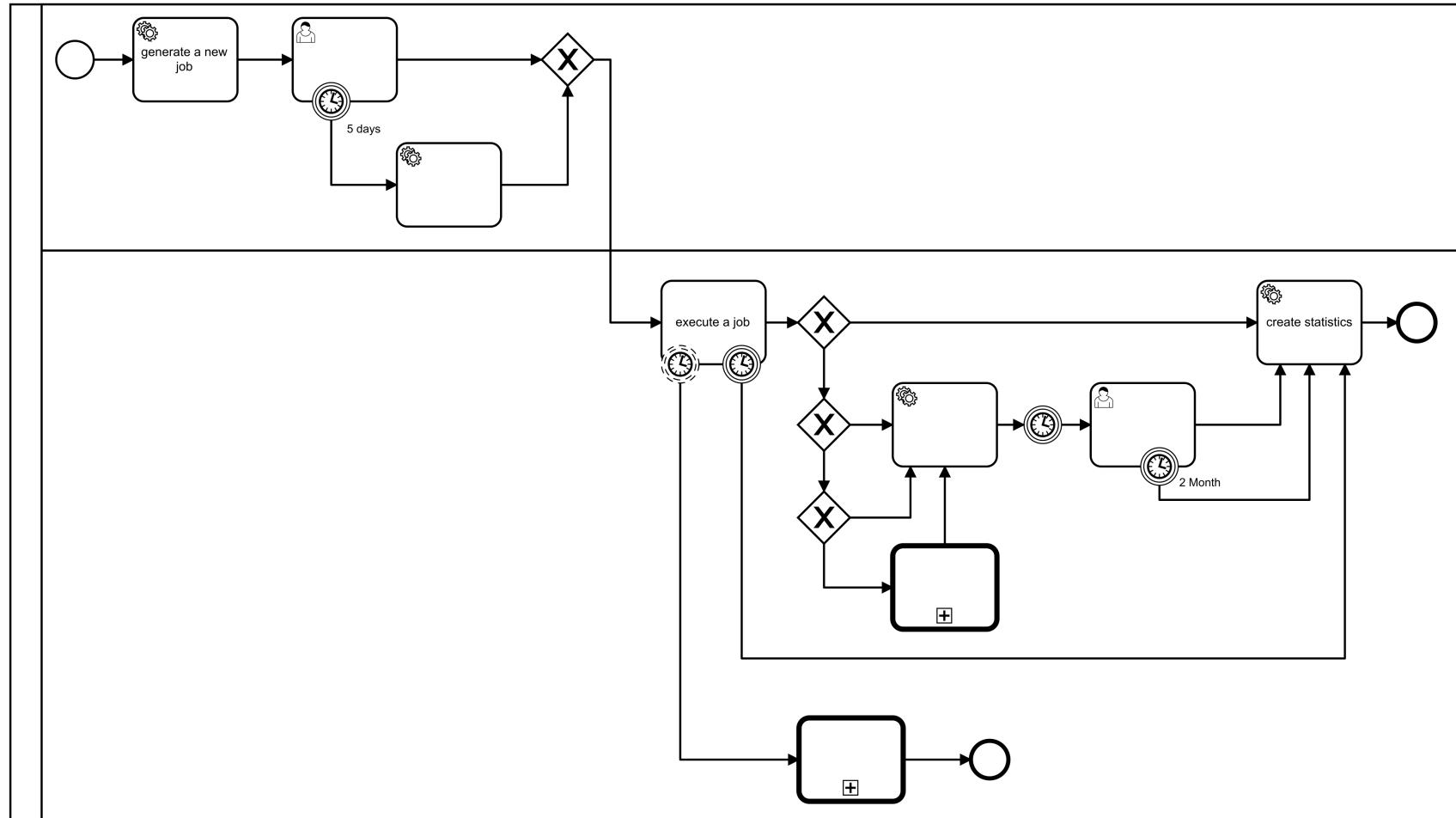


Executable model (Sungard IPP)

Analytical model in BPMN 2.0 (BPMN-modeling Tool X)



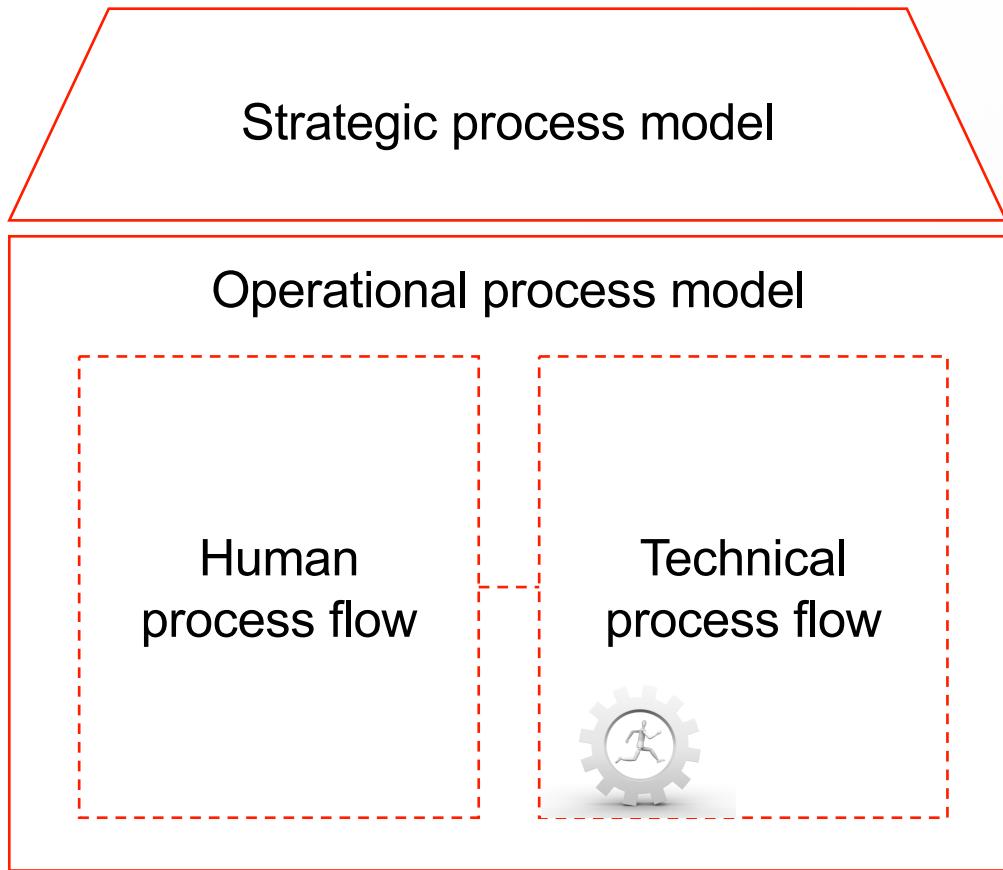
Executable model in BPMN 2.0 (Camunda Modeler)



The real world is slightly more complex ...

- Complex processes => large and comprehensive process models
- Individual requirements of stakeholders about how a model should look like and what it should contain:
 - analytical coarse grained
 - analytical fine grained
 - executable fine grained
- Individual expectations about what parts the model should contain:
 - Human view: „What parts are done by human process participants?“
 - Technology-driven view: „What part is performed by the process engine?“
 - Or both and: „How do they interact?“

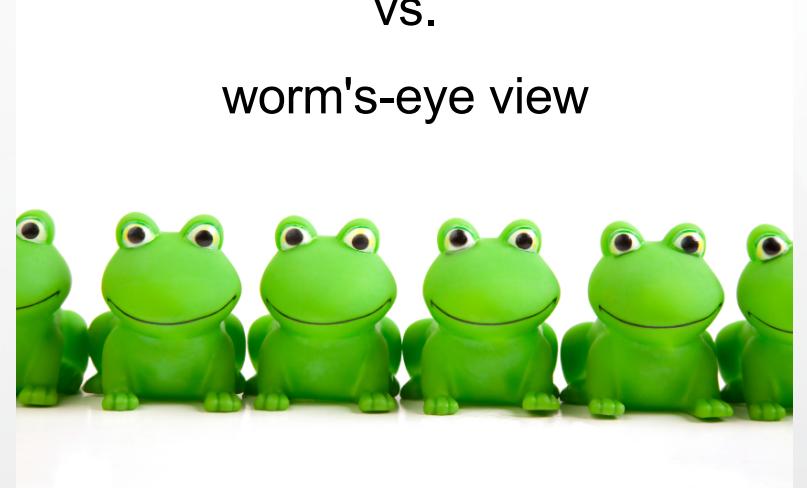
Camunda House



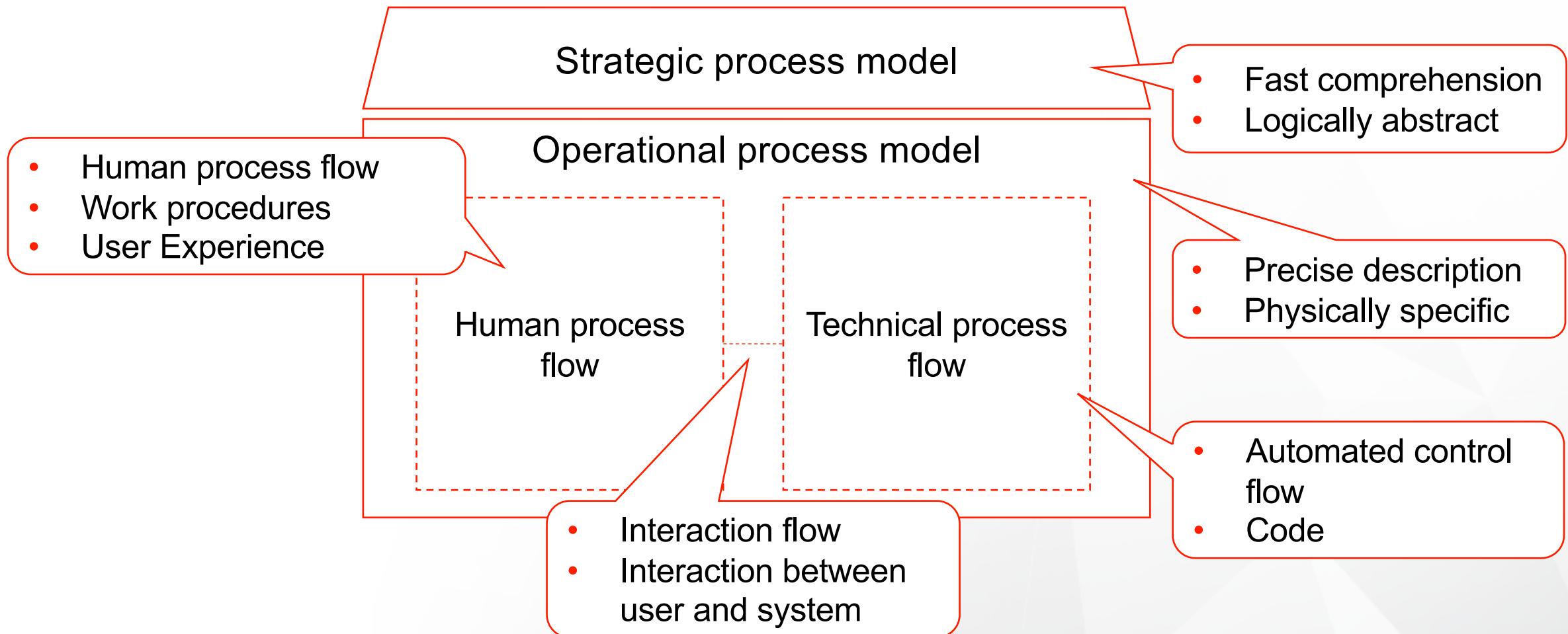
bird's eye view

vs.

worm's-eye view



Relation to the Camunda House

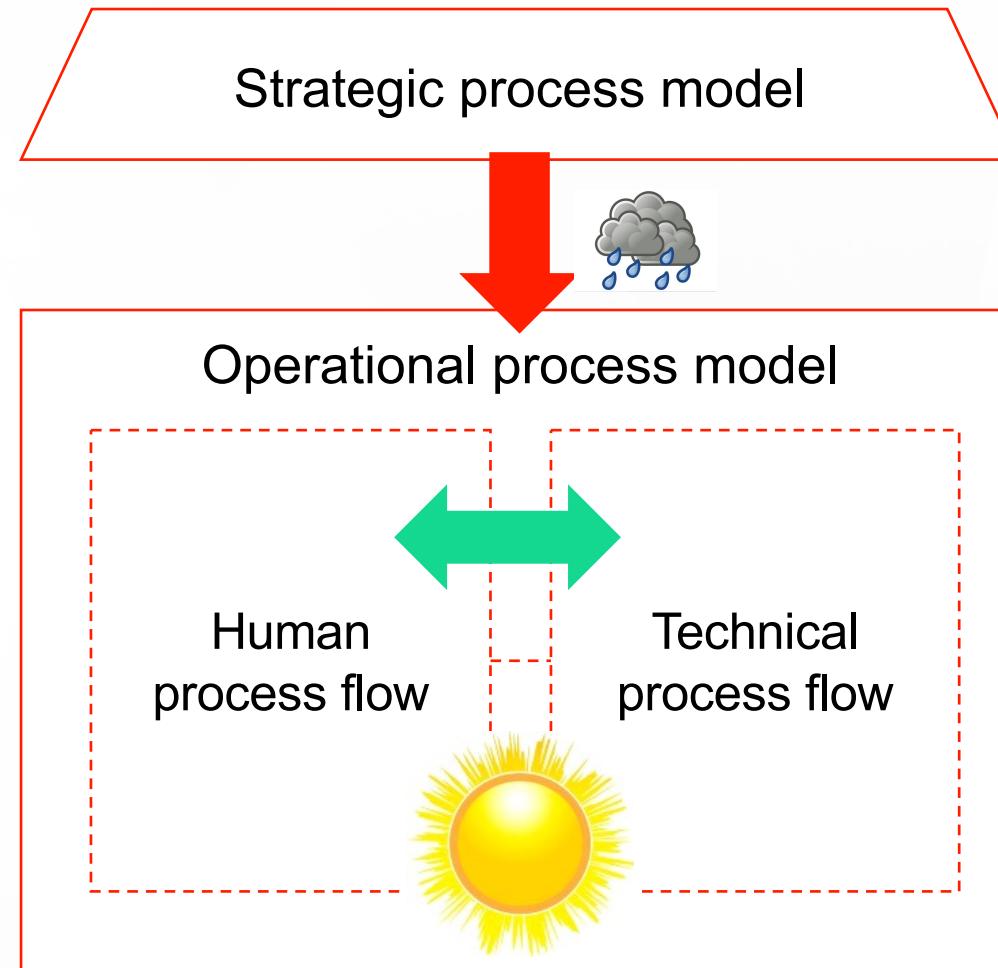


Example "Incident Management"

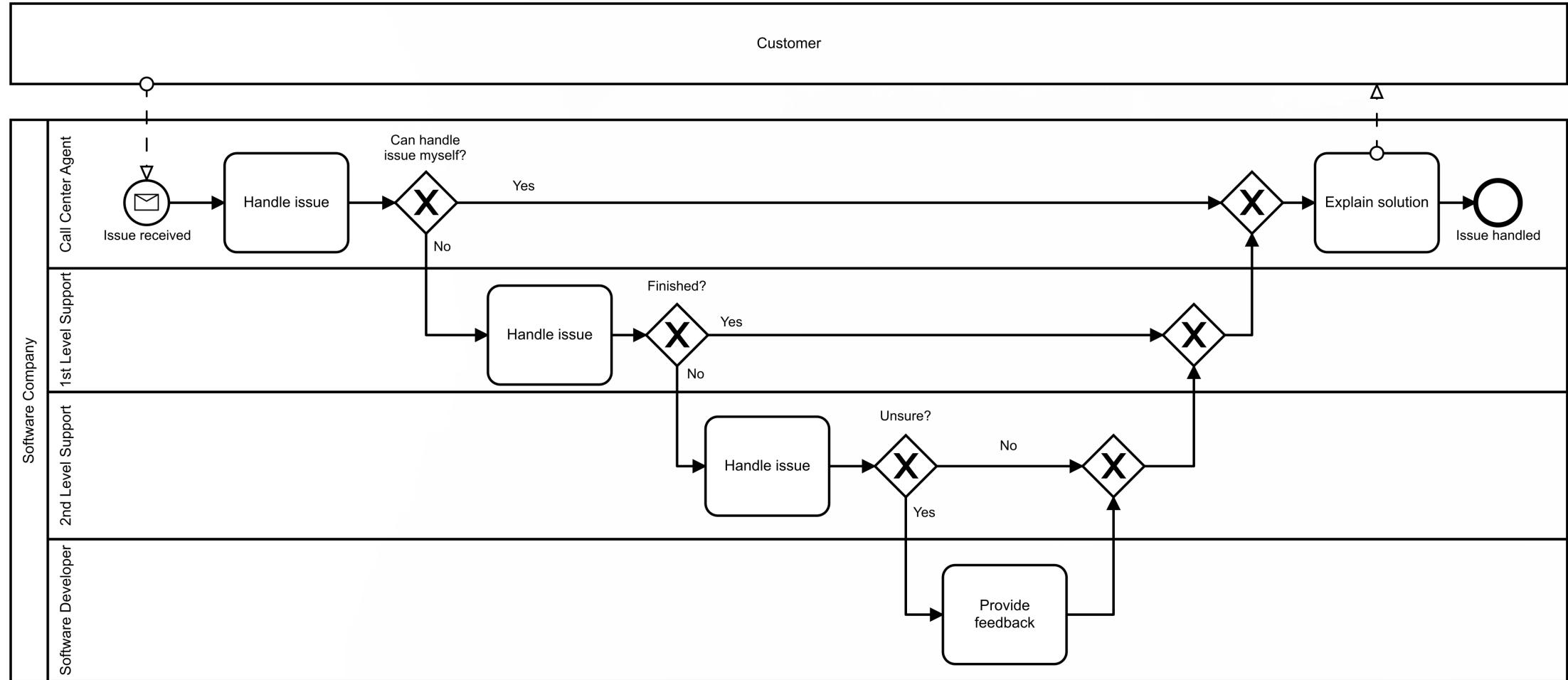
- Support process of a software vendor
- Part of the „BPMN 2.0 by Example“ document of the OMG (www.BPMN.org)
- Completely modeled using the Camunda House
- Implemented as a showcase on the Camunda Platform

<https://github.com/camunda-consulting/code/tree/master/one-time-examples/incident-management>

Core idea #1: Move „problems“ to an upper level



Incident management at strategic level

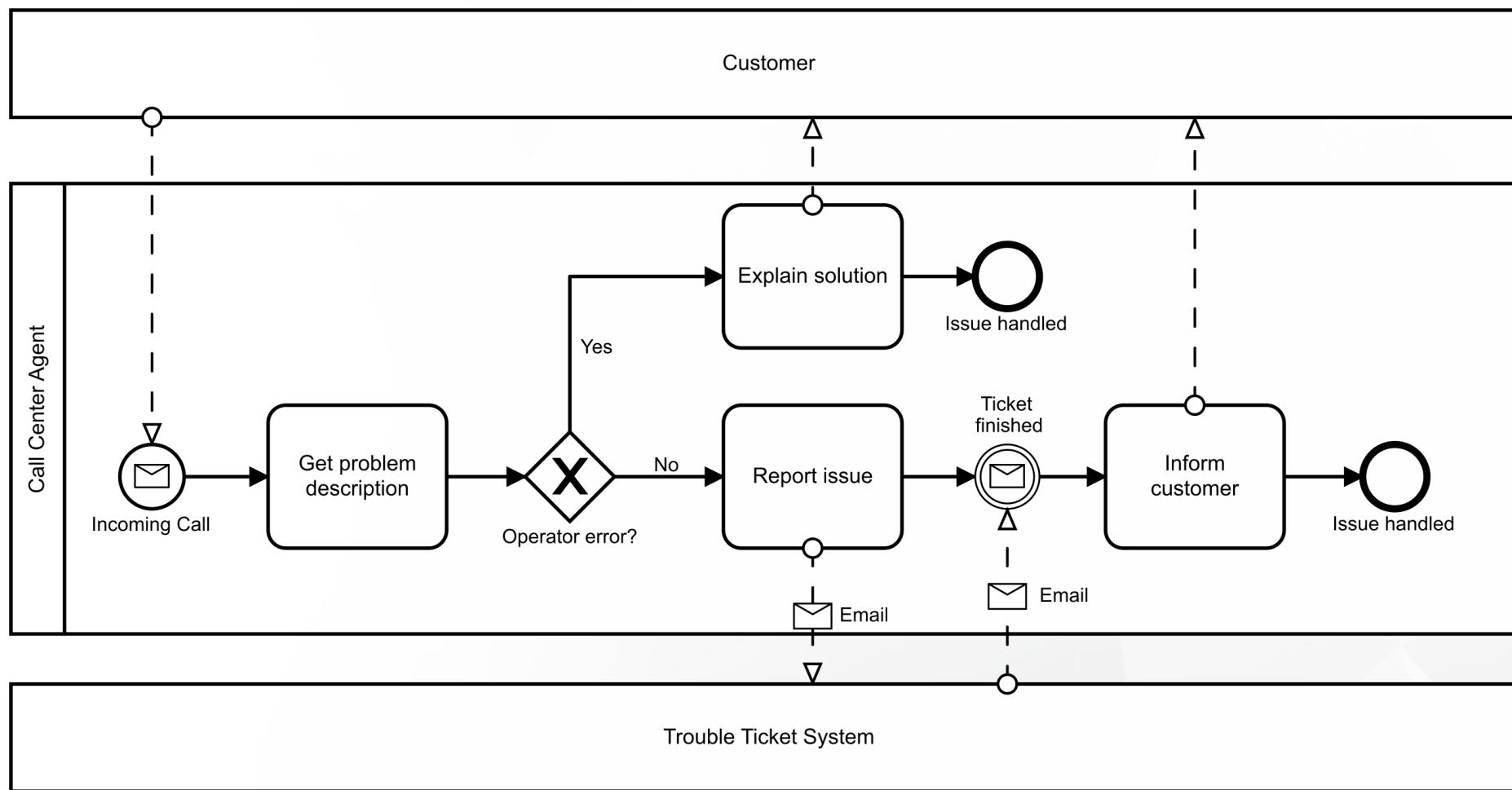


Core idea #2: operational level provides different views

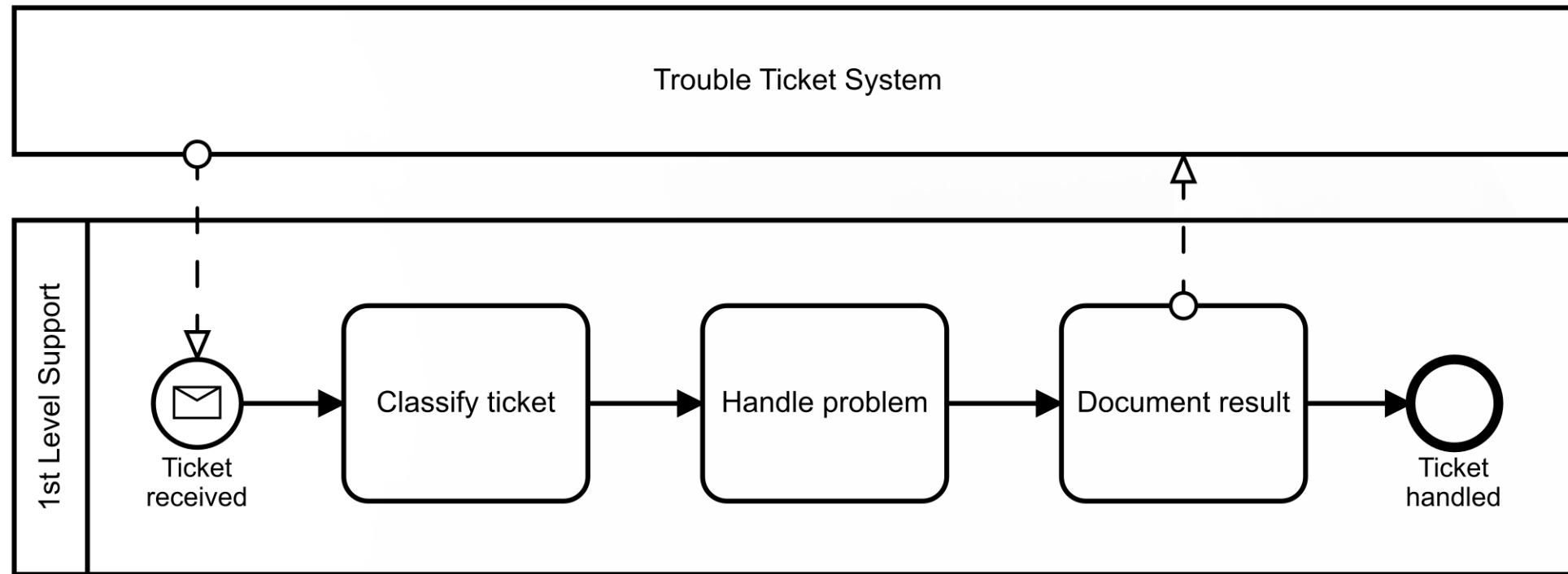
Role	Process Participant	Process Analyst	Process Engineer
Objective	How should I work?	How is the work done?	What is the engine doing exactly?
View	Own orchestration	Whole collaboration	Orchestration of the process engine

Operational process model

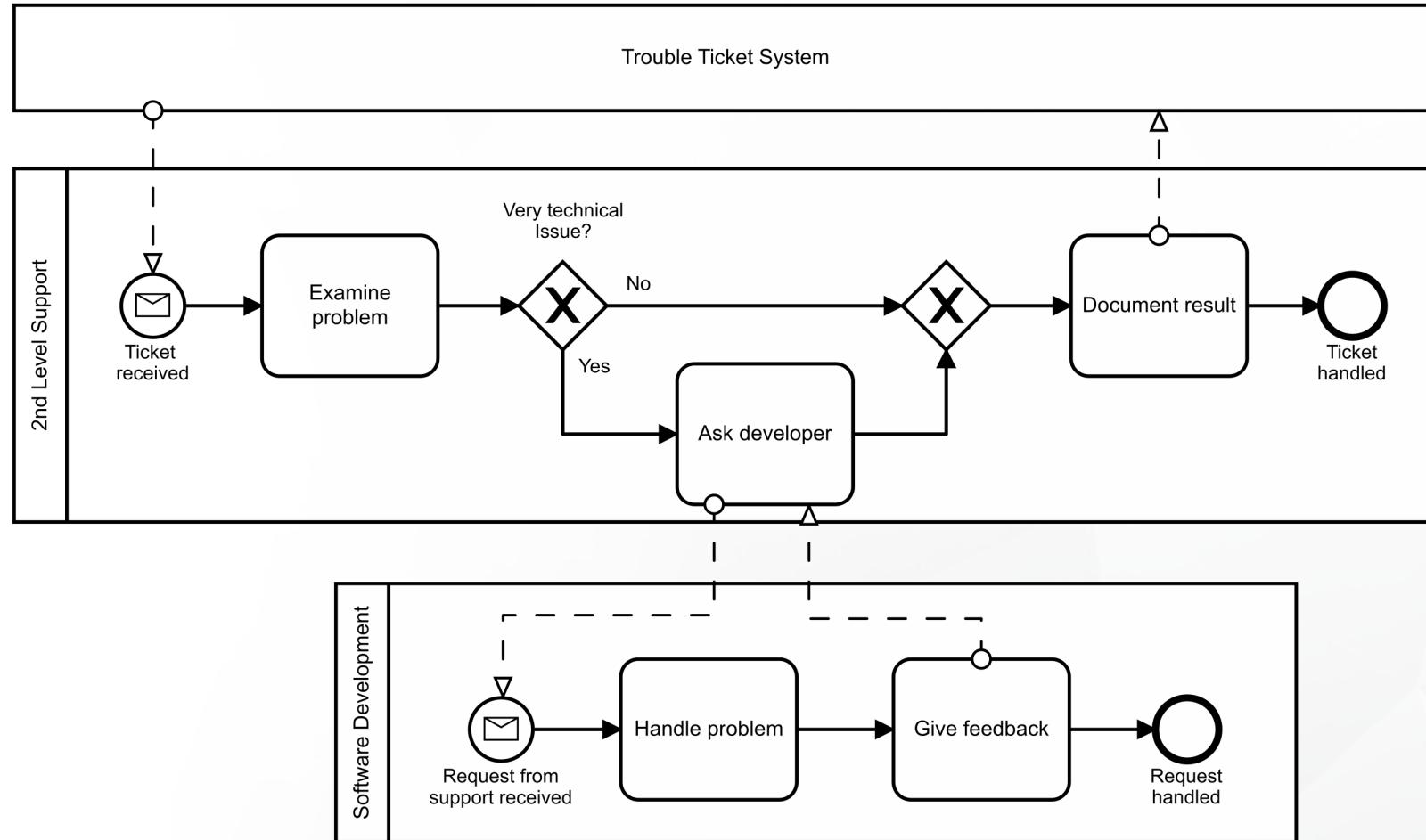
The call center agent's process



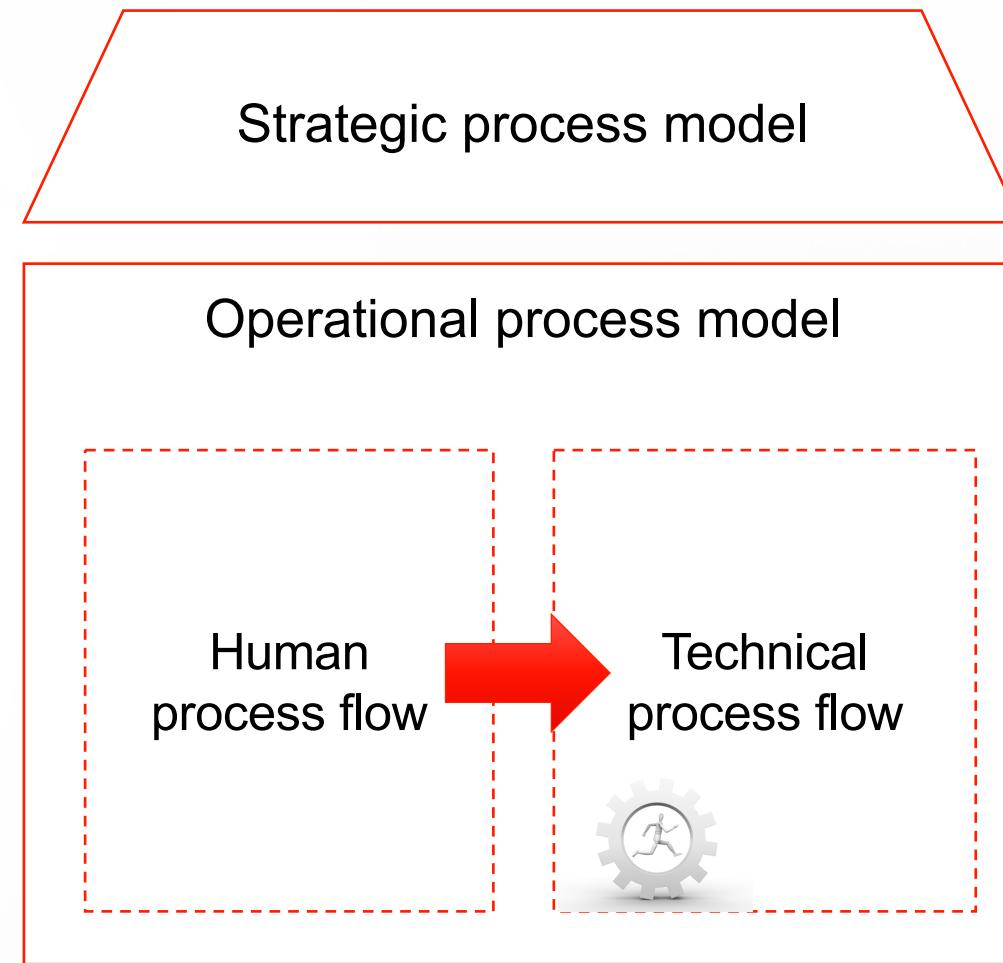
The 1st level support agent process



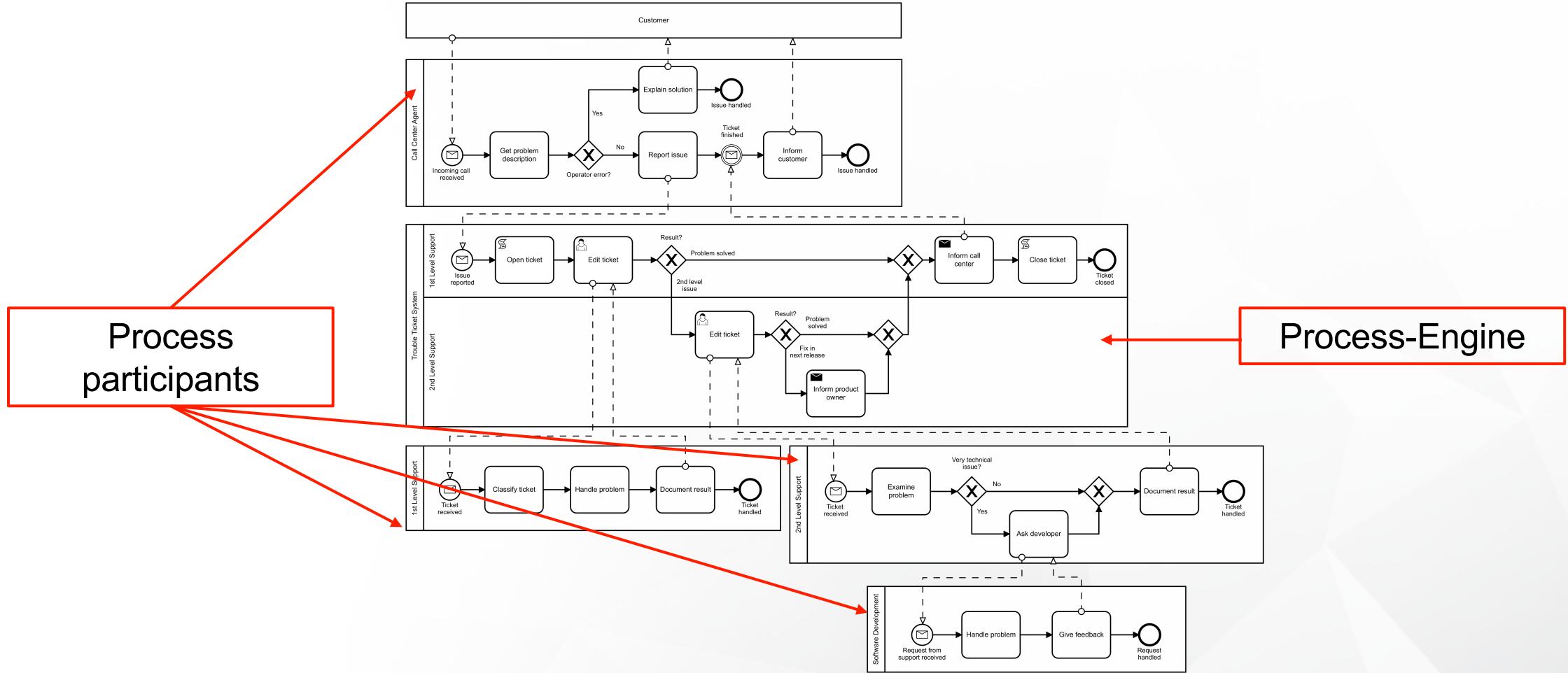
2nd level support agent and software developer



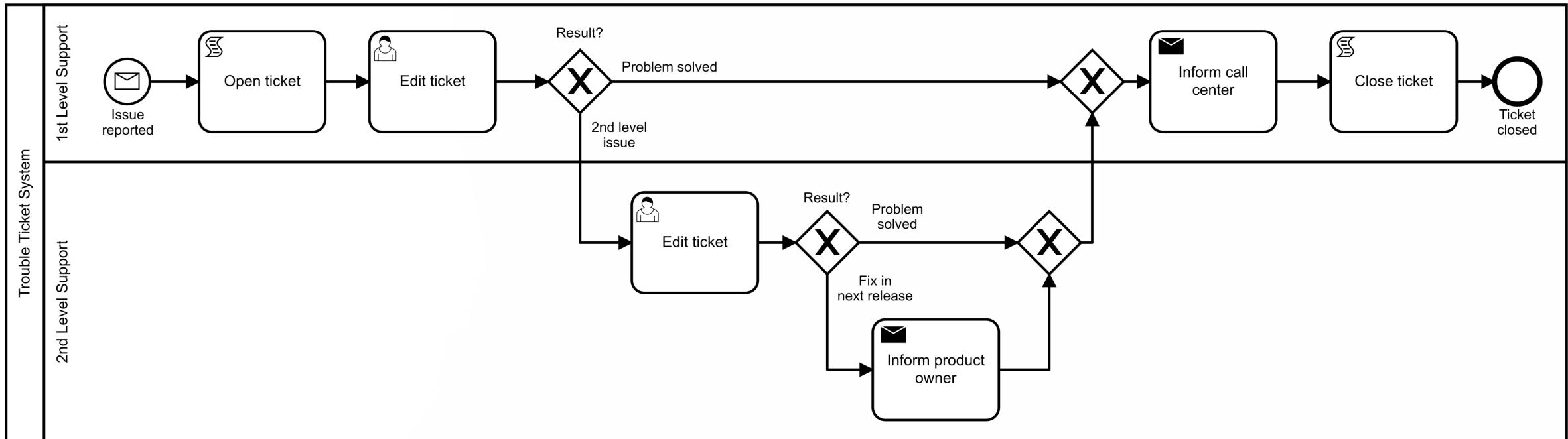
How to create the technical flow



Deriving the technical process model



The executable process model





Exercise

Invoice receipt

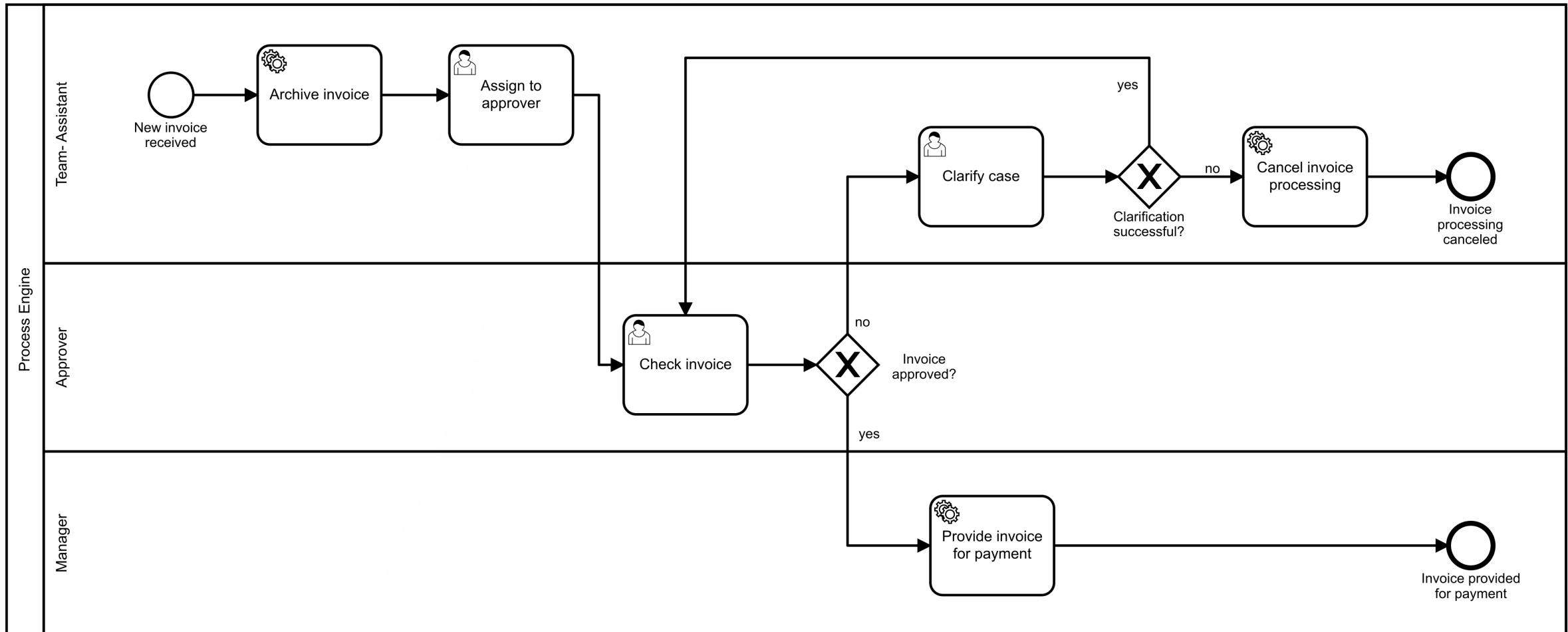
<https://training.camunda.com/bpmn/>
user: bpmn
password: camundarocks



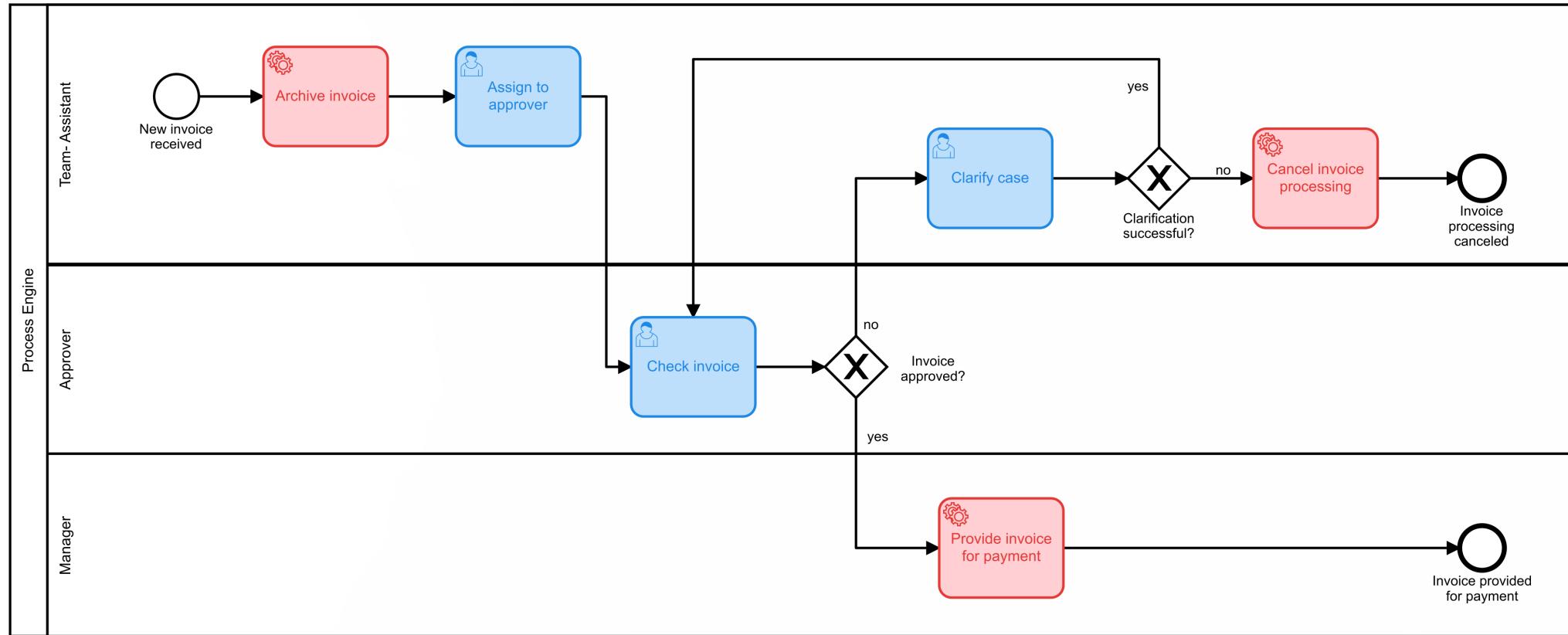
Process automation



Example invoice receipt



Process automation



Human Task Management



System Integration





Live Demo

Automation with Camunda Plattform

Screenshots Tasklist

Camunda Tasklist

Create a filter + < Created + >

My Tasks
Review Invoice
Created 2 days ago
Invoice Amount: 10.99
Invoice Number: PSACE-5342

My Group Tasks
Accounting
John's Tasks
Mary's Tasks
Peter's Tasks

All Tasks (6)

Approve Invoice

Invoice Receipt
Set follow-up date
Form History Diagram Description

Do you approve this invoice?

Invoice Document: invoice.pdf
Creditor: Great Pizza for Everyone Inc.
Amount: 30
Invoice Category: Travel Expenses
Invoice Number: GPFE-23232323
 Do you approve?
Save Complete

Assign Reviewer

Review Invoice
Created 2 days ago
Invoice Amount: 10.99
Invoice Number: PSACE-5342

Prepare Bank Transfer

Invoice Receipt
Due in 5 days, Created 2 days ago
Invoice Amount: 900
Invoice Number: BOS-43934

Approve Invoice

Invoice Receipt
Due in 5 days, Created 2 days ago
Invoice Amount: 30
Invoice Number: GPFE-23232323

Camunda Tasklist

Create a filter + < Created + >

My Tasks
Review Invoice
Created 2 days ago
Invoice Amount: 10.99
Invoice Number: PSACE-5342

My Group Tasks
Accounting
John's Tasks
Mary's Tasks
Peter's Tasks

All Tasks (6)

Approve Invoice

Invoice Receipt
Set follow-up date
Form History Diagram Description

Diagram View

Approve Invoice process diagram:

```
graph LR; Start(( )) --> Assign[Assign Approver Group]; Assign --> Review{Review Invoice}; Review -- no --> NoReview[Review successful?]; NoReview -- yes --> Approve[Approve Invoice]; Approve -- no --> NoApprove[Invoice approved?]; NoApprove -- yes --> YesApprove[ ]; YesApprove --> Bank[Prepare Bank Transfer]; Bank --> Archive[Archive Invoice]; Archive --> End((( )))
```

The diagram illustrates a workflow for invoice approval. It starts with an 'Invoice received' event, followed by assigning an approver group. The next step is 'Review Invoice'. If the review is successful (yes), the invoice is approved. If not (no), it goes back to the review step. After approval, the invoice is prepared for bank transfer, and finally archived.

Screenshot Cockpit

Camunda Cockpit Processes Decisions Cases Human Tasks More ▾ Demo Demo Home ▾

Dashboard » Processes » Rechnungseingang : History | Runtime

Definition Version: 2 ▾

Definition ID: rechnungseingang:2:55872f5e-c117-...

Definition Key: rechnungseingang

Definition Name: Rechnungseingang

Tenant ID: null

Deployment ID: 558581ac-c117-11e6-9079-0242ac1...

Instances Running:

- current version: 0
- all versions: 0

Related:

- Reports
- Migration

show Heatmap

Rechnungseingang

Team Assistent

Rechnung eingegangen → Freigebenden auswählen → Rechnung freigeben

Freihandler

Rechnung freigeben → Klären erfolgreich? (Decision Diamond)

Klären erfolgreich? ja → Rechnung klären → Rechnung nicht erfolgreich verarbeitet (End Event)

Klären erfolgreich? nein → Rechnung freigeben (Loopback)

Buchhaltung

Rechnung freigeben → Friegabe erfolgt? (Decision Diamond)

Friegabe erfolgt? ja → Rechnung freigegeben (End Event)

Friegabe erfolgt? nein → Bezahlung vorbereiten → Rechnung archivieren → Rechnung erfolgreich verarbeitet (End Event)

Buchhaltungs-extern

Process Instances Job Log KPI

Cycle Time

Dauer Rechnungsfreigabe

(Calculated from 'Rechnung eingegangen' to 'Rechnung freigegeben')

Ratio

RG zurückgewiesen: 3.25

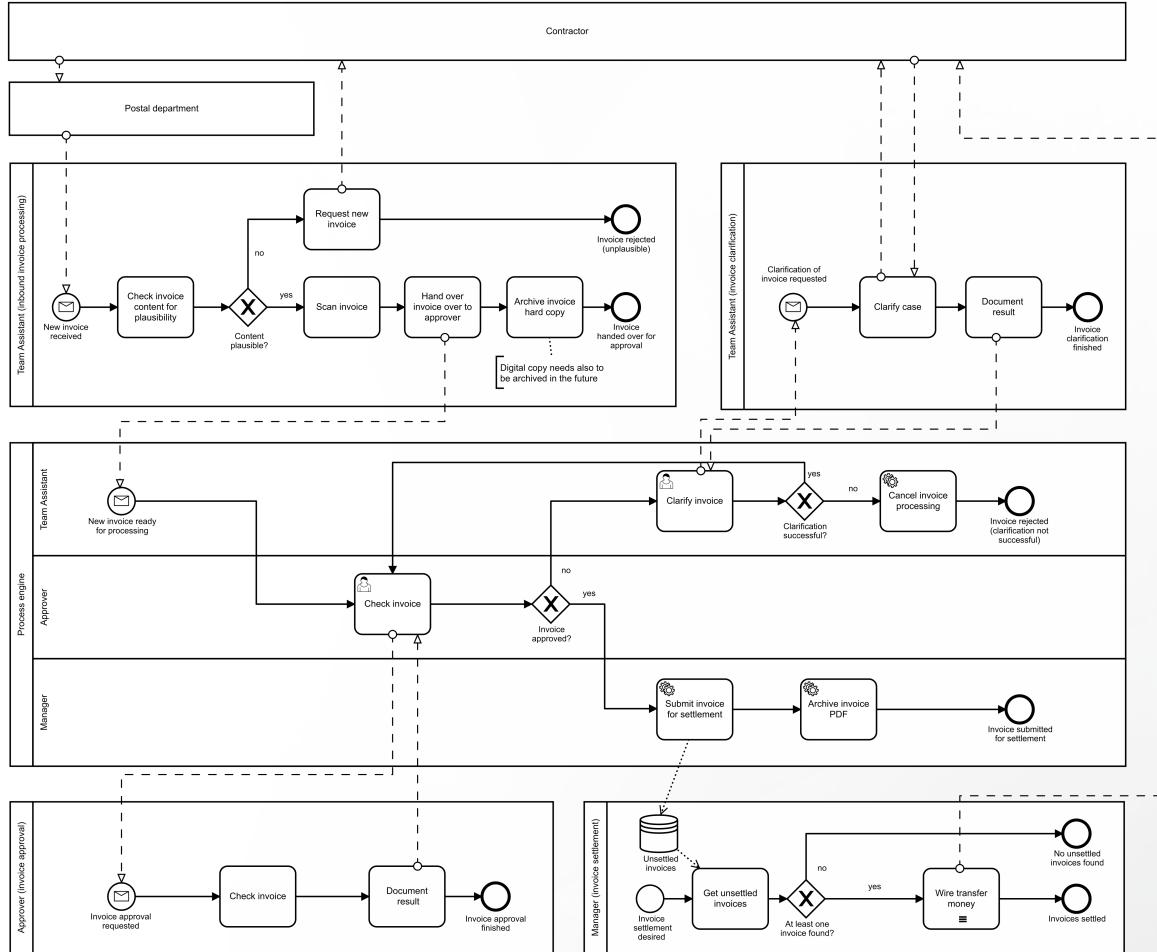
verarbeitet: 96.75 %

Started Instances

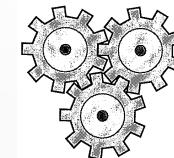
Powered by camunda BPM / v7.6.0-ee

The screenshot shows the Camunda Cockpit interface for a process named 'Rechnungseingang'. On the left, a sidebar displays process details like 'Definition Version: 2', 'Definition ID: rechnungseingang:2:55872f5e-c117-...', and 'Instances Running: current version: 0'. The main area features a process diagram with three lanes: 'Team Assistent', 'Freihandler', and 'Buchhaltung'. The 'Team Assistent' lane starts with 'Rechnung eingegangen' and leads to 'Freigebenden auswählen', then 'Rechnung freigeben'. The 'Freihandler' lane follows 'Rechnung freigeben' through a decision diamond 'Friegabe erfolgt?' to either 'Rechnung freigegeben' or 'Bezahlung vorbereiten'. The 'Buchhaltung' lane follows 'Rechnung freigeben' through another decision diamond 'Klären erfolgreich?' to either 'Rechnung nicht erfolgreich verarbeitet' or 'Rechnung archivieren'. Performance metrics at the bottom include a histogram of cycle times, a donut chart for ratio, and a line chart for started instances.

Process invoice receipt



user

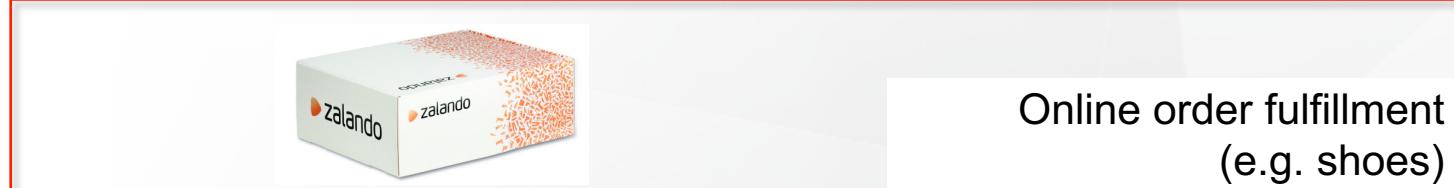
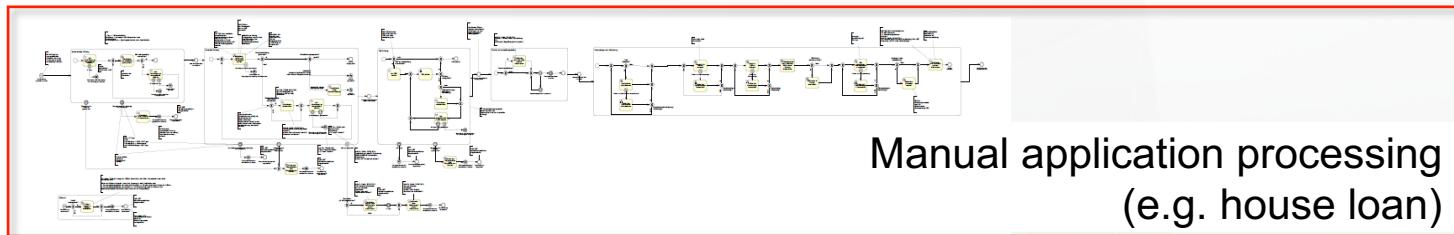
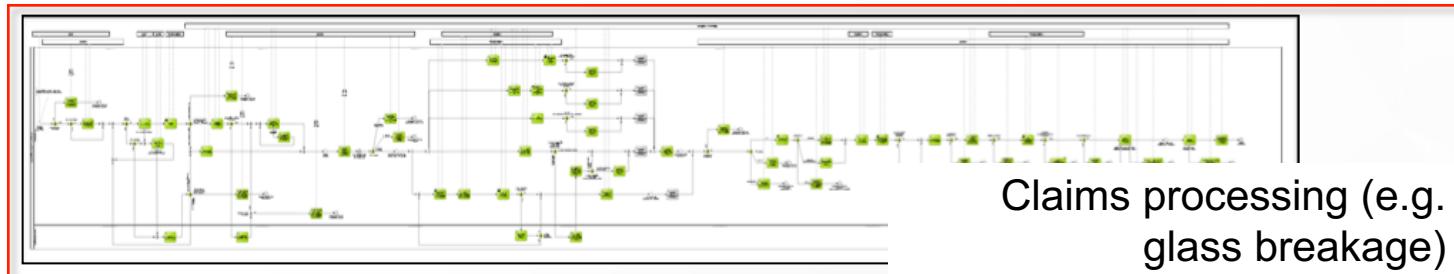
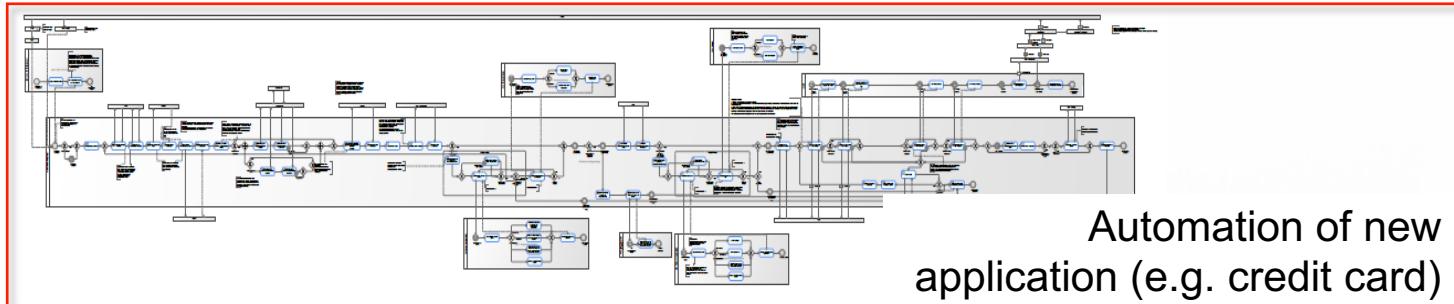


technical process



user

Process Examples

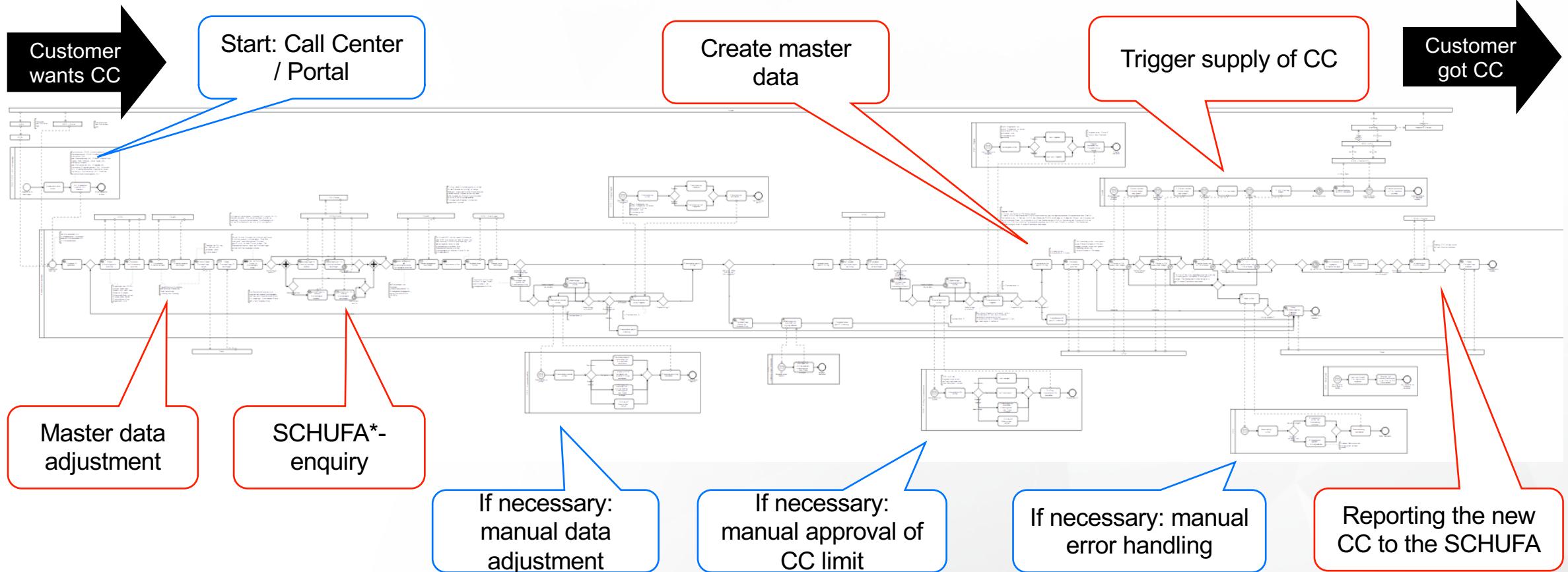


Credit card (CC) application in detail

Legend

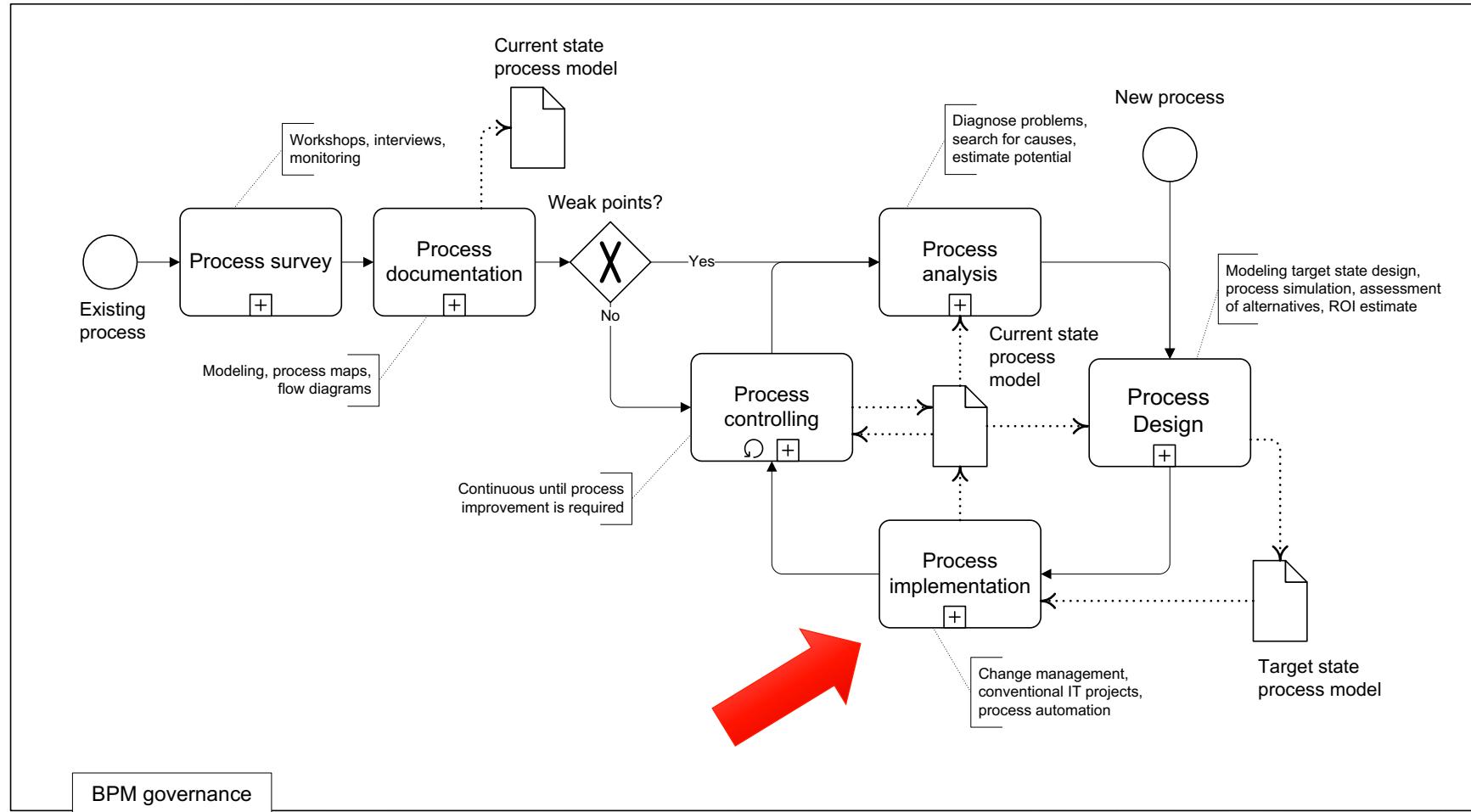
User
Interaction

Service calls

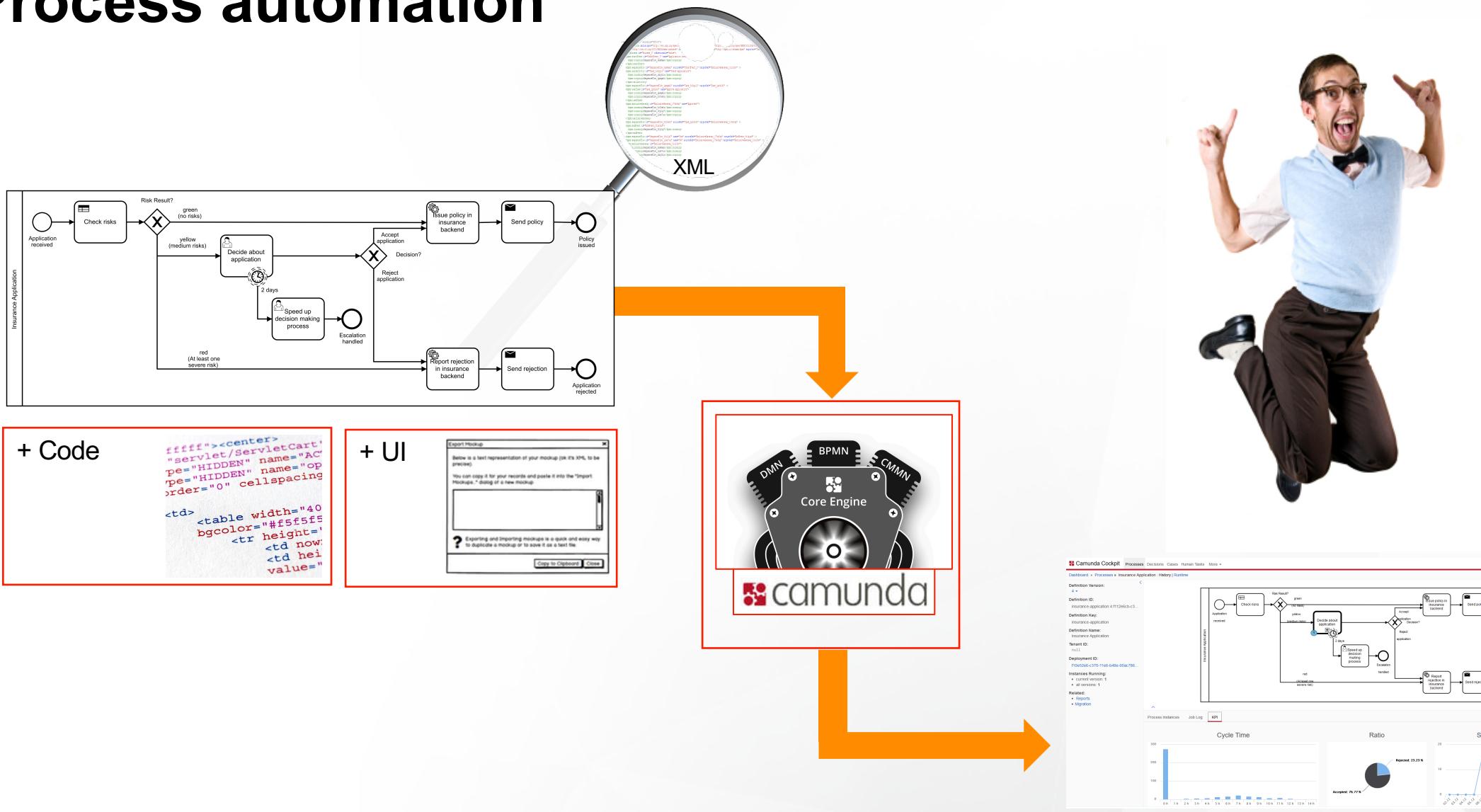


*SCHUFA: German credit protection agency

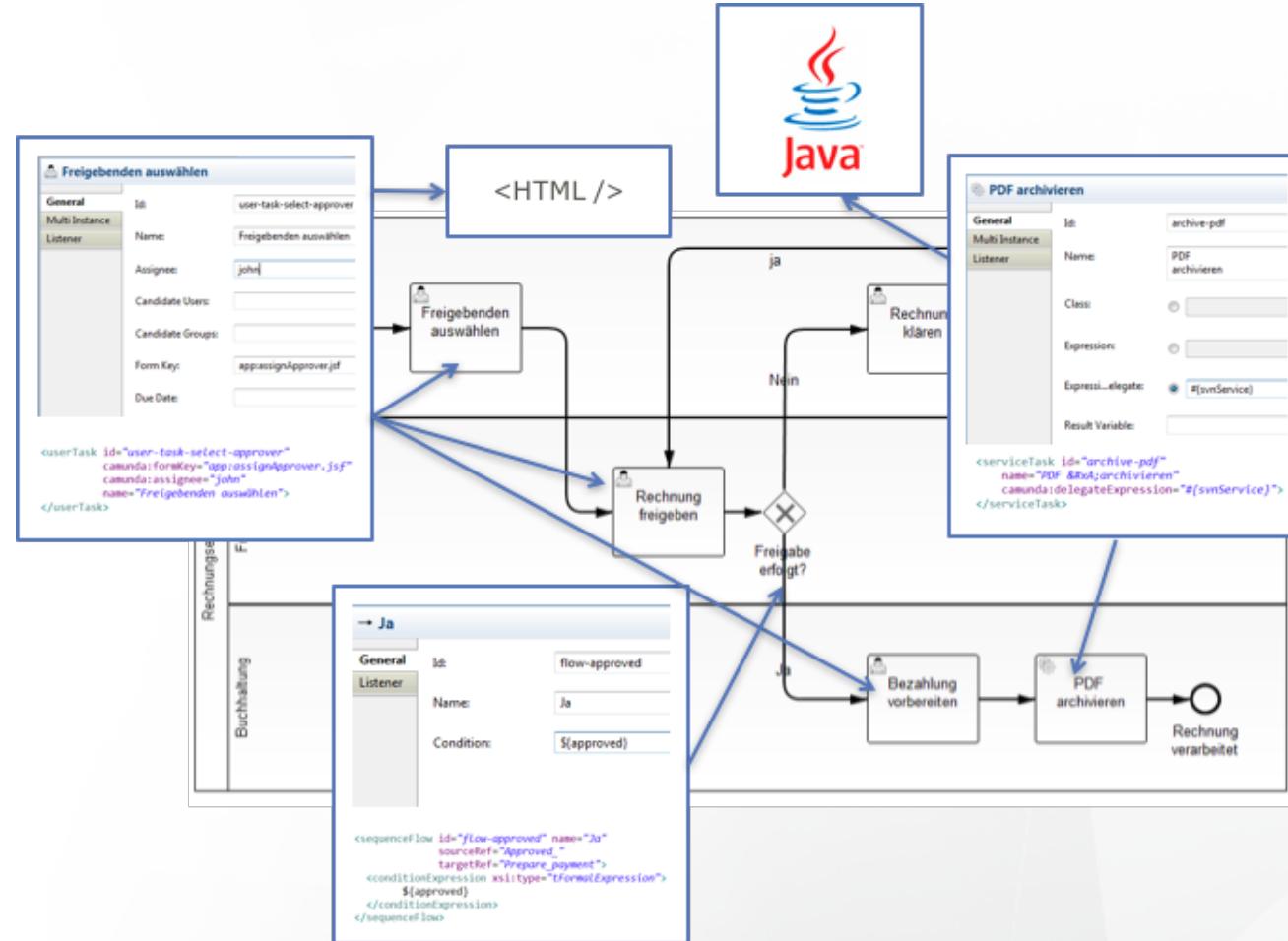
How to automate a BPMN 2.0- process?



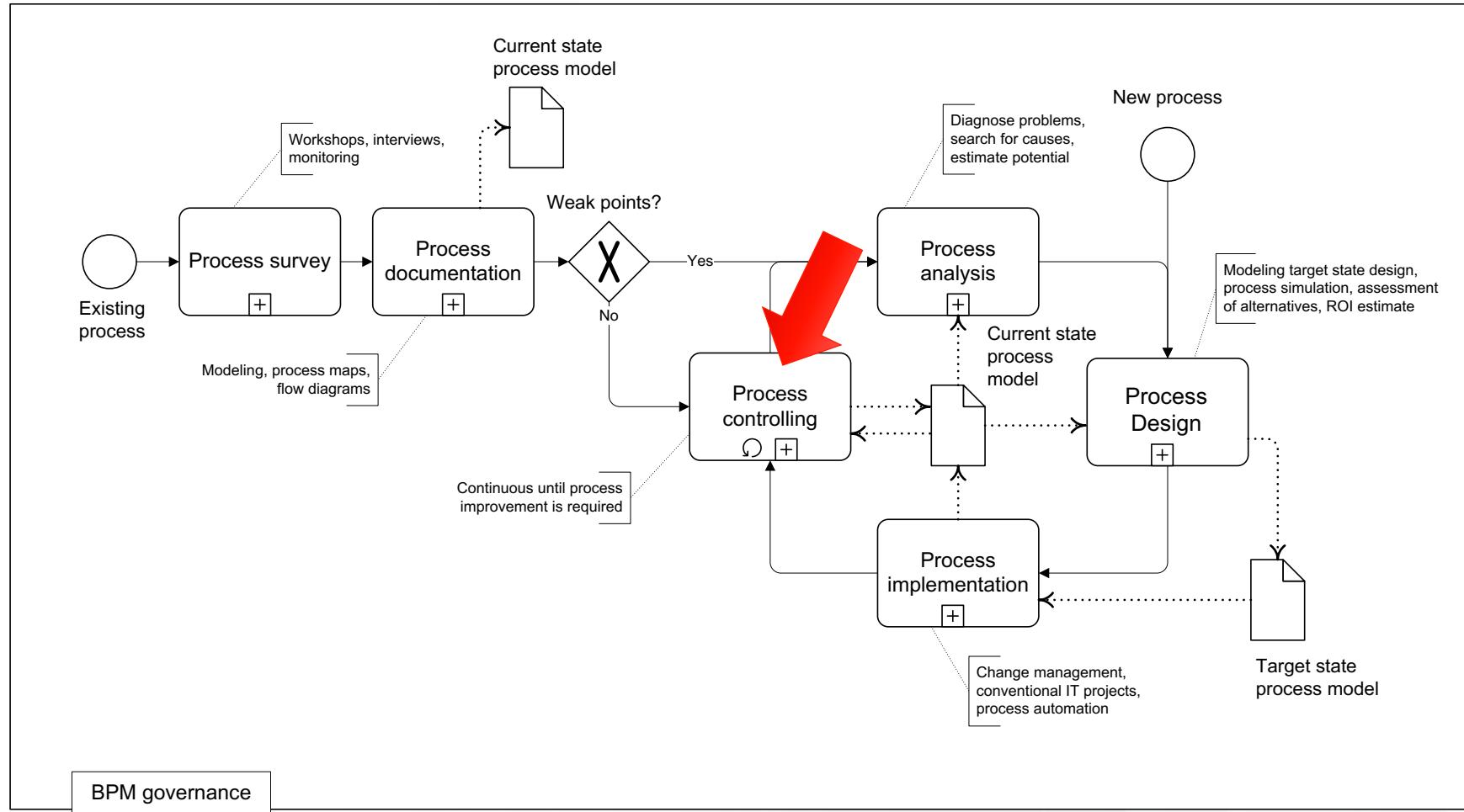
Process automation



Technical attributes of a BPMN model

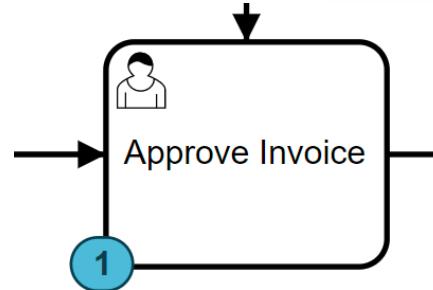


Process controlling as a result of automation



Transparency means quality

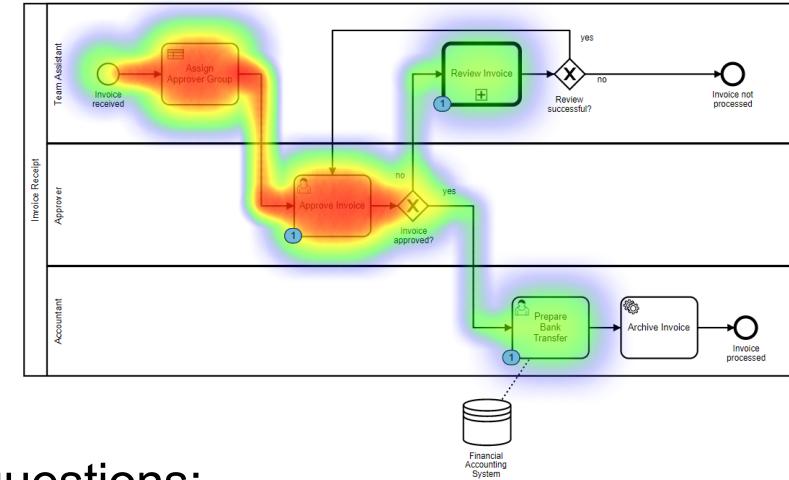
Instance-based



Typical questions:

- Why isn't the invoice paid yet?
- Where is my application?
- When does my shipment arrive?

Process-based



Typical questions:

- Do we keep our service-promise/ SLA?
- Capacity, overload, weak points?
- Where is potential for improvement?
- Does the reality reflect the expectations?

Instance-based: example Cockpit

The screenshot shows the Camunda Cockpit interface for a process instance. The top navigation bar includes 'Processes', 'Decisions', 'Cases', 'Human Tasks', and 'More'. The main area displays the process definition and its variables.

Historical (=audit) data available: Points to the 'History' tab in the top navigation bar.

Where is my process instance at the moment?: Points to the process diagram on the left.

Possibility of intervention: Points to the toolbar on the right.

Data for one process instance: Points to the variable table below.

Process Diagram (BPMN View):

```
graph LR; Start(( )) --> Assign[Assign Approver Group]; Assign --> Decision1{Approve?}; Decision1 -- no --> Review[Review Invoice]; Decision1 -- yes --> Decision2{Invoice approved?}; Decision2 -- no --> End1((( )));
Decision2 -- yes --> Transfer[Prepare Bank Transfer];
Transfer --> Archive[Archive Invoice];
Archive --> End2((( ));
```

Variables Table:

Name	Value	Type	Scope	Actions
invoiceNumber	GPFE-23232323	String	Invoice Receipt	[Edit] [Delete]
invoiceDocument	Download	File	Invoice Receipt	[Edit] [Delete]
amount	30	Double	Invoice Receipt	[Edit] [Delete]
creditor	Great Pizza for Everyone Inc.	String	Invoice Receipt	[Edit] [Delete]
invoiceCategory	Travel Expenses	String	Invoice Receipt	[Edit] [Delete]
approverGroups	java.util.ArrayList	Object	Invoice Receipt	[Edit] [Delete]

Process-based: example Cockpit

Camunda Cockpit Processes Decisions Cases Human Tasks More ▾

Demo Demo

Dashboard » Processes » Rechnungseingang : History | Runtime

Definition Version: 2

Definition ID: rechnungseingang:2:55872f5e-c117-...

Definition Key: rechnungseingang

Definition Name: Rechnungseingang

Tenant ID: null

Deployment ID: 558581ac-c117-11e6-9079-0242ac1...

Instances Running:

- current version: 0
- all versions: 0

Related:

- Reports
- Migration

Rechnungseingang

Freigebender

Buchhaltung

Team - Assistent

Rechnung eingegangen

Freigebenden auswählen

Rechnung freigeben

Rechnung klären

Klärung erfolgreich?

ja

nein

Rechnung nicht erfolgreich verarbeitet

Rechnung freigegeben

Freigabe erfolgreich?

ja

nein

Bezahlung vorbereiten

Rechnung archivieren

Rechnung erfolgreich verarbeitet

Rechnung eingegangen

Rechnung freigeben

Rechnung klären

Klärung erfolgreich?

ja

nein

Rechnung nicht erfolgreich verarbeitet

Rechnung freigegeben

Freigabe erfolgreich?

ja

nein

Bezahlung vorbereiten

Rechnung archivieren

Rechnung erfolgreich verarbeitet

show Heatmap

Process Instances Job Log KPI

Cycle Time

Ratio

Started Instances

RG zurückgewiesen: 3.25

verarbeitet: 96.75 %

Highcharts.com

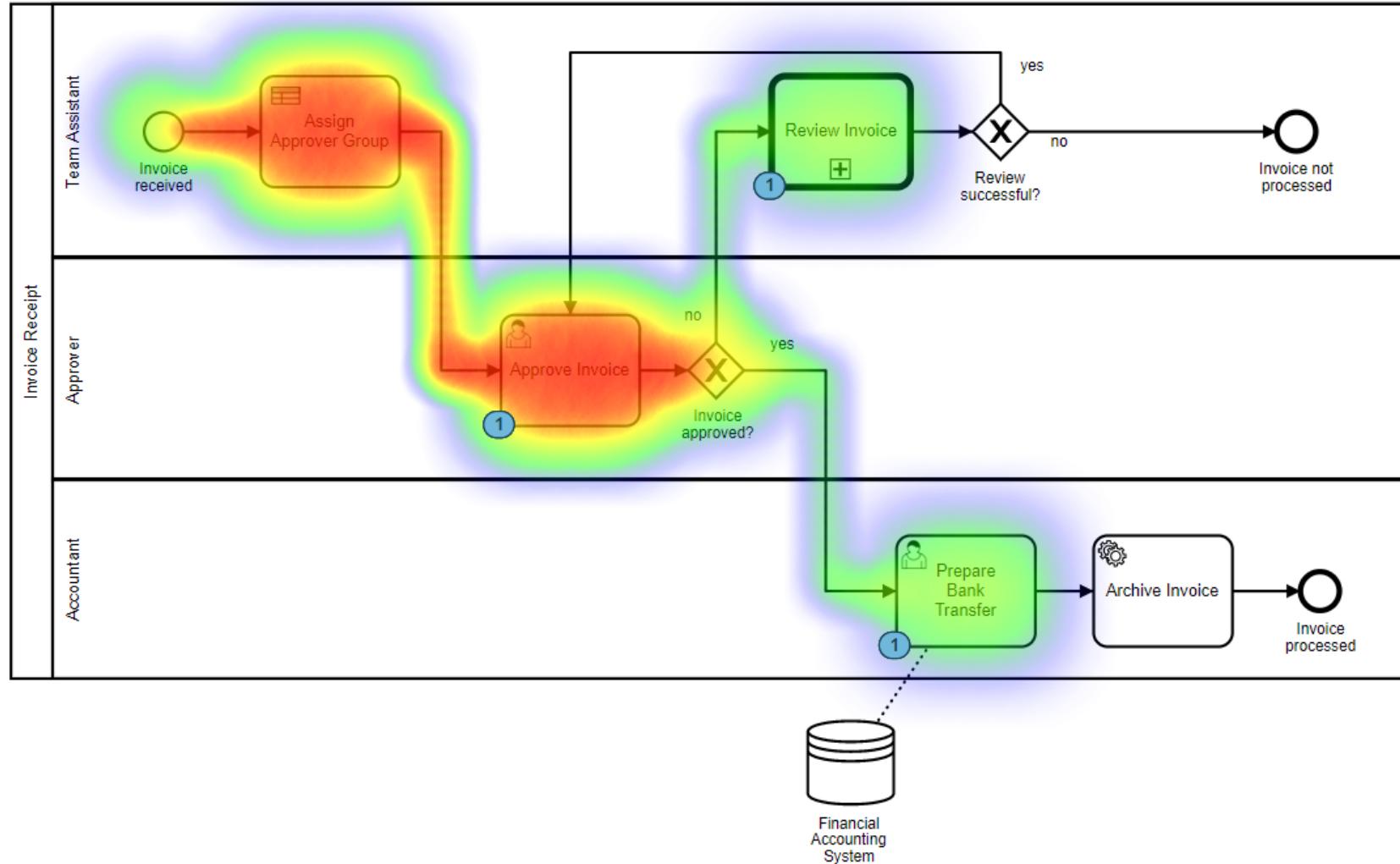
Highcharts.com

Highcharts.com

29.11 30.11 01.12 02.12 03.12 04.12 05.12 06.12 07.12 08.12 09.12 10.12 11.12 12.12 13.12

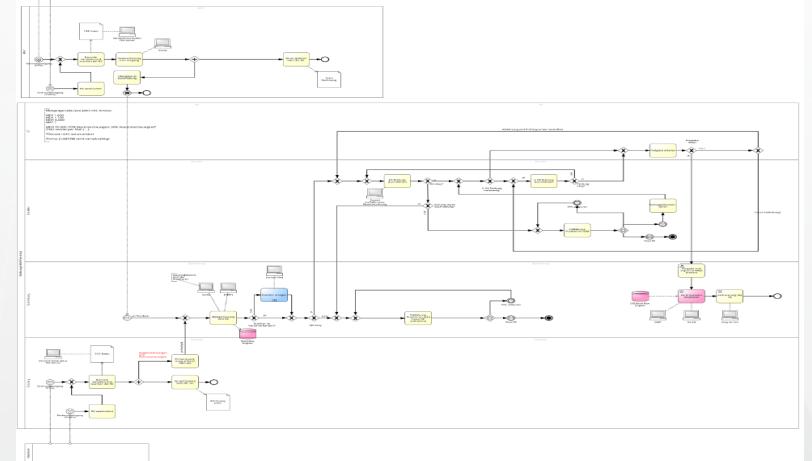
Powered by camunda BPM / v7.6.0-ee

In a nutshell: understanding what is going on technically



Process automation – which tool?

- Process automation can be done in different ways
- This offers different advantages and disadvantages
- Therefore a general recommendations is difficult
- In the following we like to describe a real life customer scenario
 - Big energy supplier
 - Strong growth
 - Invoice receipt process



Possible alternatives

1. No automation
2. To program for themselves
3. To use standard-Software
4. Classical BPM Suite
5. Developer- friendly BPM

Alternative 1: No automation



Process is done manually
(Basic tools like paper, folders,...)



Advantages

- No IT expense



Pain points

- Not scalable
- Slow process run
- High number of process errors
- No process visibility
- Process agility is limited



Cost drivers

- High personnel cost for running the process

Alternative 2: Custom programming



Software solution designed inhouse
(e.g. Java, .NET, ...)



Advantages

- Faster process executions
- Lower process costs
- Less process errors



Pain points

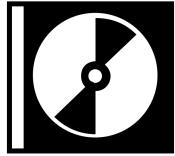
- No process visibility
- Process agility is limited strongly



Cost drivers

- Costs for development and maintenance
- Project costs for modification (processes hidden inside the implementation)

Alternative 3: Off-the-shelf-software



Purchased software with e.g. invoice receipt module, document management systems



Advantages

- Faster process executions
- Less process costs
- Less process errors



Pain points

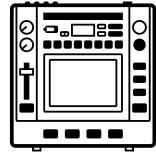
- Processes in standard software aren't customizable for own company
- Software is a foreign body in the company's IT landscape
- Agility of process is limited
- Dependence on software supplier



Cost drivers

- Costs for license
- Proprietary know-how (specialized consultants)

Alternative 4: Classical BPM-Suite



Black-Box-BPM-Suite with „Zero-Code“-promise
(e.g. BPM-Suites of big software producers)



Advantages

- Faster process executions
- Less process costs
- Less process errors
- Improvement of process visibility (dependent on software producer)
- End-to-End-processes possible



Pain points

- Software is a foreign body in the company's IT landscape
- Dependence on software supplier



Cost drivers

- Costs for license
- Proprietary know-how (specialized consultants and developers)

Alternative 5: Developer-friendly BPM

C

Open Source Platform with a „developer-friendly“ approach
(e.g. Camunda Plattform)



Advantages

- Faster process executions
- Less process costs
- Less process errors
- Improved process visibility
- End-to-End-processes possible
- Embeddable in IT- landscape
- Standard technology and procedures (e.g. Java, CI, Scrum)



Pain points

- Taking a risk on doing it differently
- Need for developers



Cost drivers

- Cost for development and integration

Summary of the alternatives

See also
Whitepaper:

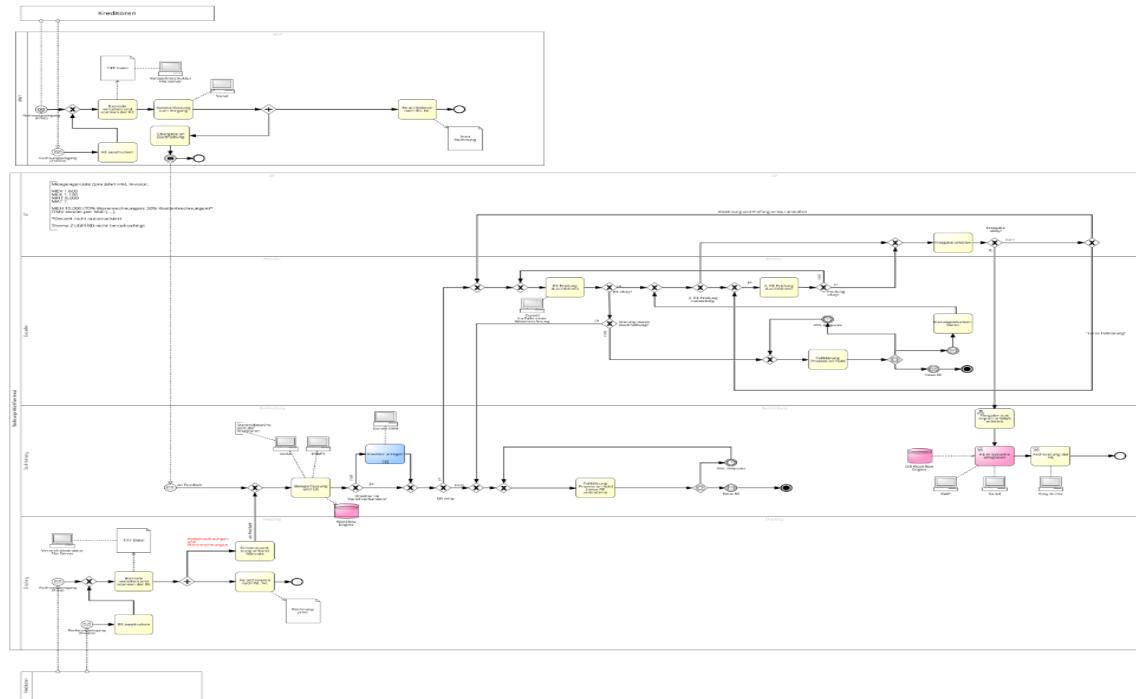
Camunda in comparison
with other alternatives



	No automation	Custom programming	Off-the-shelf-software	Classical BPM-Suite	Developer-friendly BPM
Low process cost		X	X	X	X
Faster process executions		X	X	X	X
Less process errors		X	X	X	X
Real process visibility			possible	possible	X
High agility of process				possible	X
Less programming cost	X		X		X
Custom-tailored processes	X	X		X	X
Embeddable with existing IT		X			X
No special developers needed		X	X		X
No dependence on software supplier/ producer	X	X			X
End-to-End-processes		possible		X	X

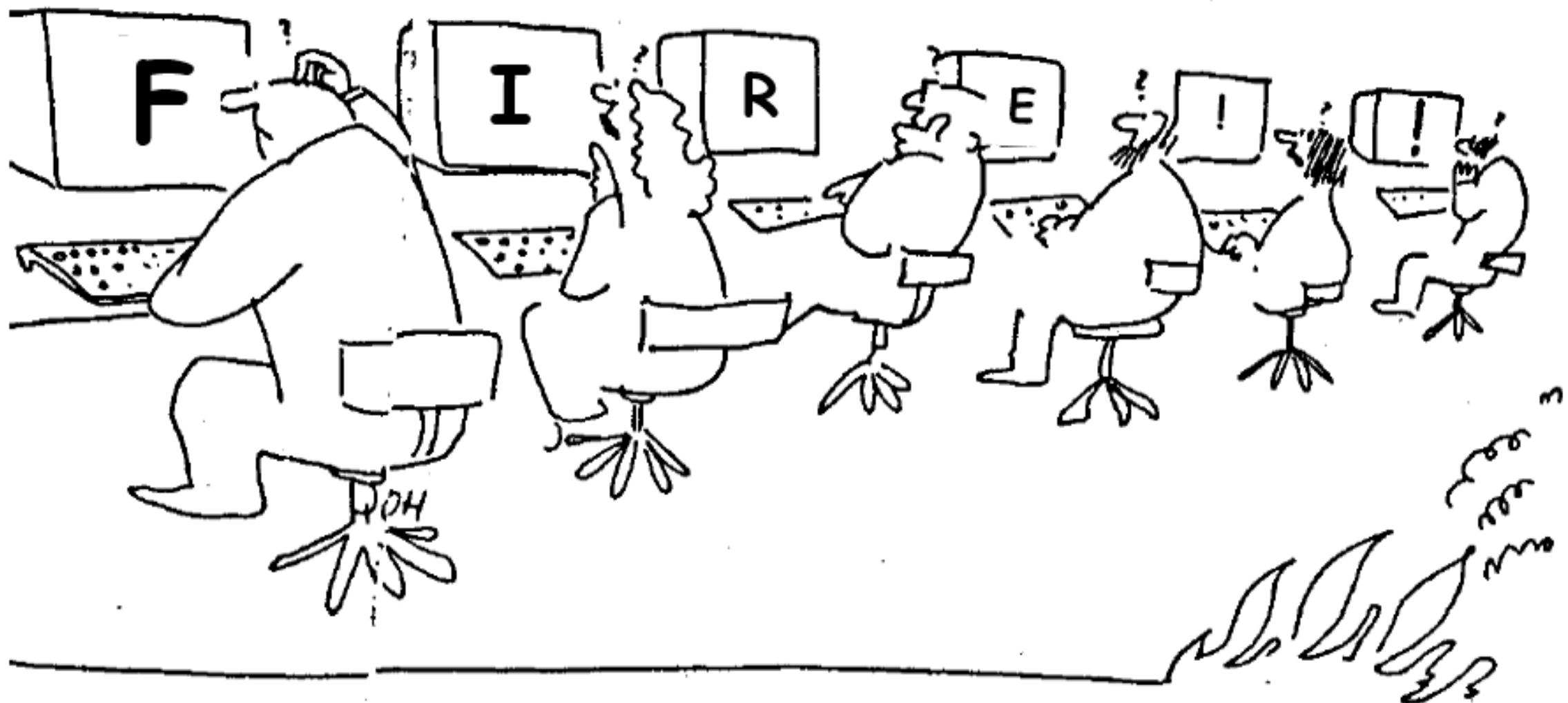
The energy supplier

- Started with manual processes (alternative 1)
- Got by with self made little helpers (alternative 2) when the load increased

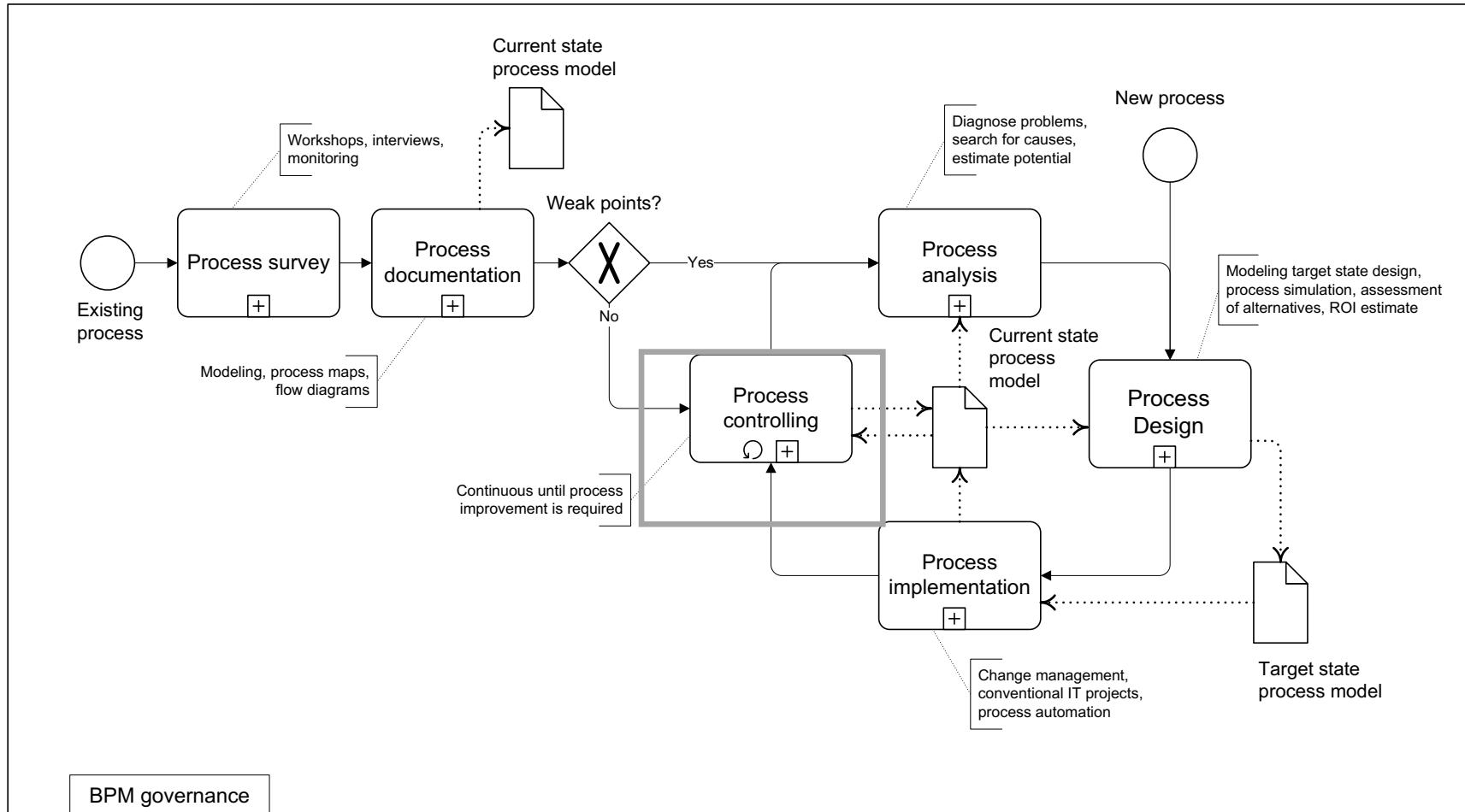


- Evaluated standard software (alternative 3) and decided not to buy it because of missing features and therefore high involved customizing costs
- Introduced a development friendly BPM (alternative 5) because:
 - The end-to-end process
 - With all its special characteristics,
 - With BPMN 2.0 synchronized, implemented and monitored as well as
 - Could be implemented with existing inhouse developers
- The solution is growing along company growth.

Do you know your processes?



The camunda BPM lifecycle



Richard is “Process Owner”

- Richard is driving projects for process improvement, often involving process automation with Camunda BPM
- Continuously improving business process performance, reporting to (senior) line managers.
- Working closely with both business users and IT.



Typical questions Richard wants to get answered for his process

Is the process producing the intended outcome?

Is the process error-prone?

Is the process taking too long?

Is the process too expensive?



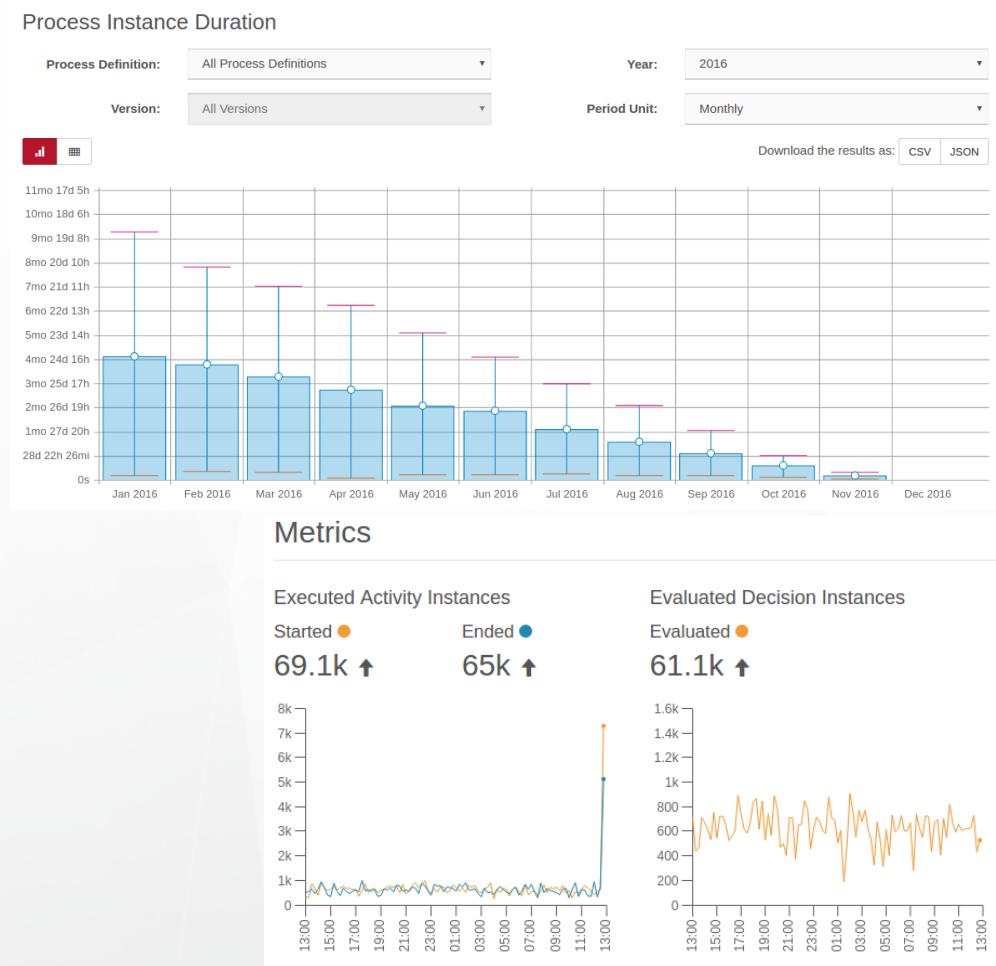
What to measure?

Key performance indicators...

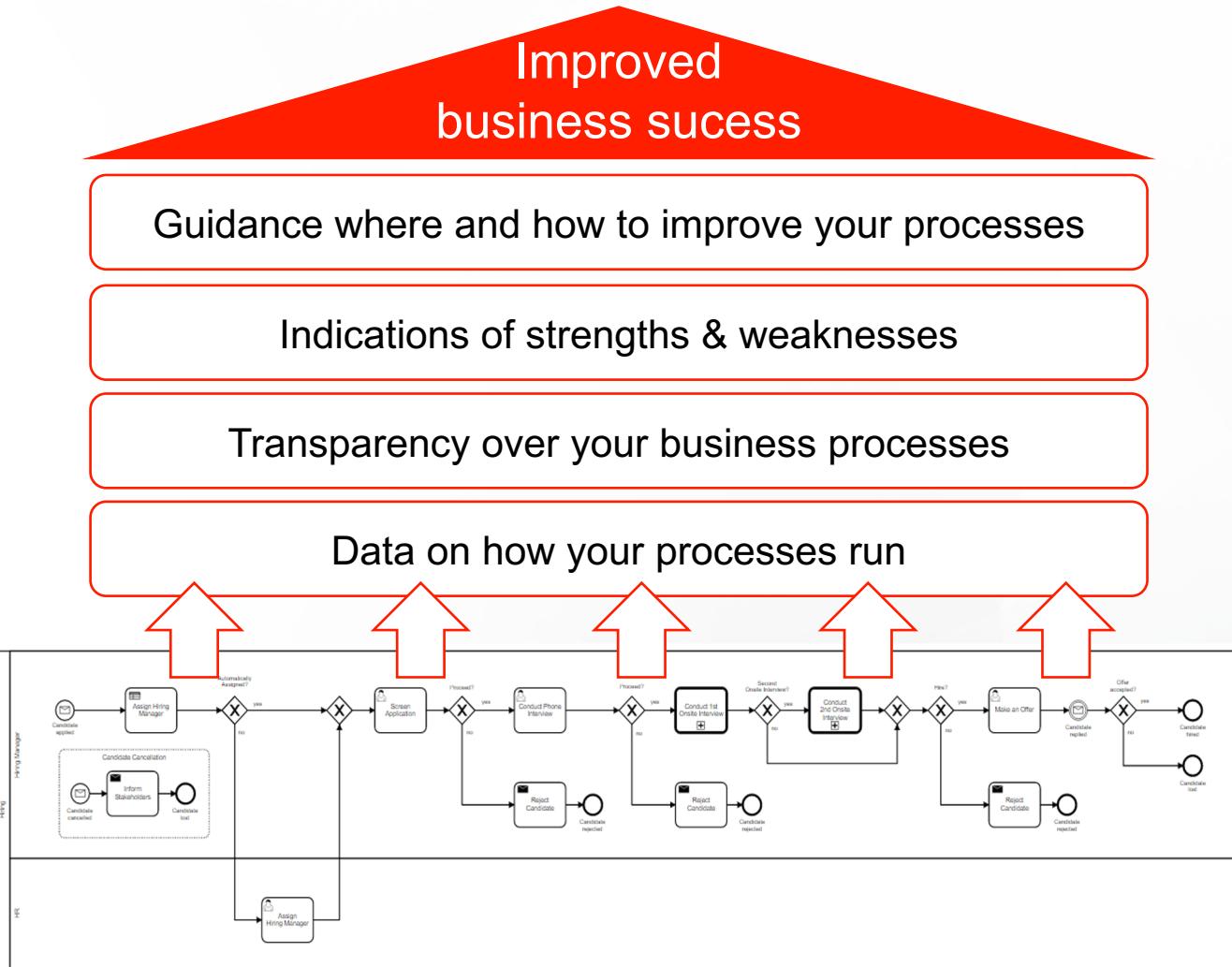
- ...evaluate the success of an organization or a particular activity.
- ...support organizations to define and measure those activities that are critical for achieving a certain goal
- ...can be very individual depending on the organization or domain

Typical KPIs for business processes:

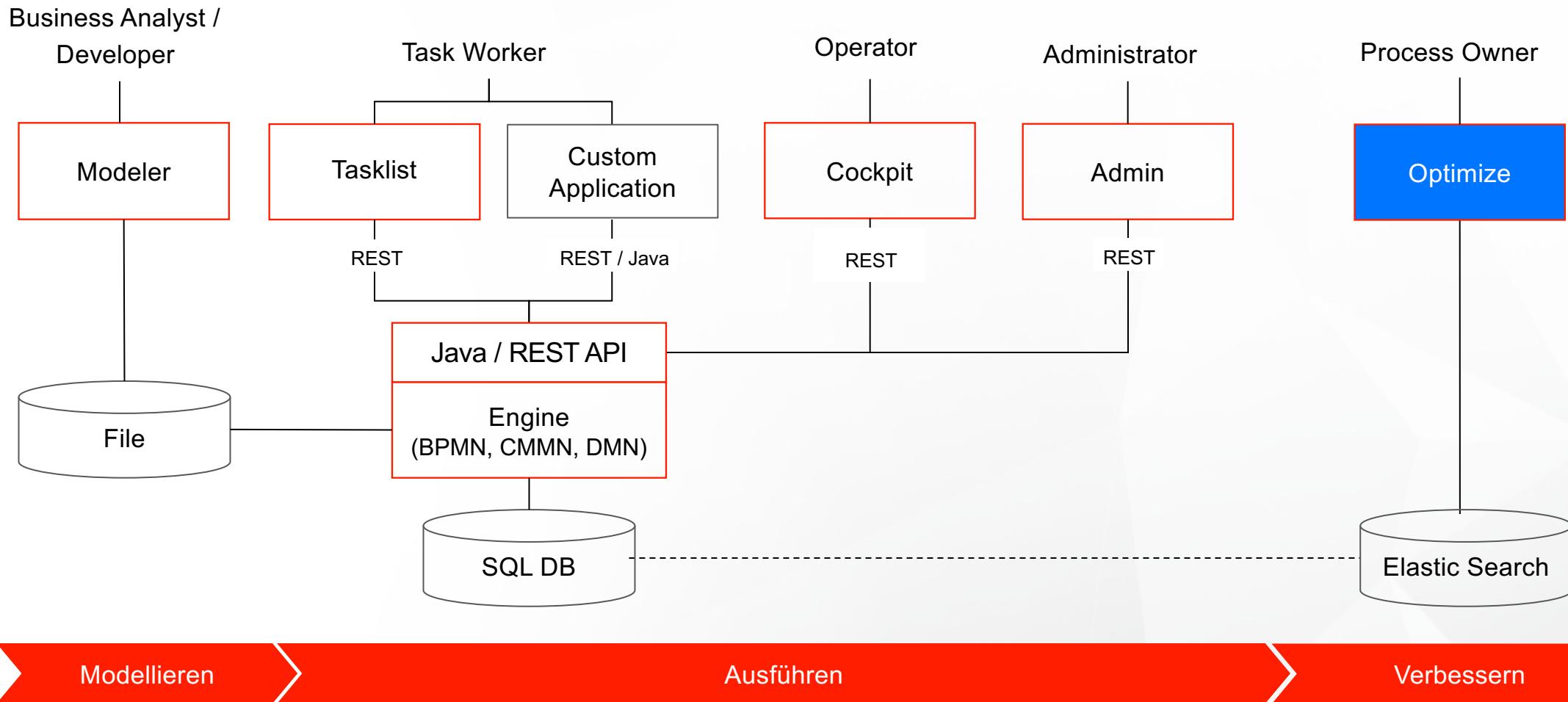
- Durations
- Output capacity
- Costs
- Resource consumption



Benefits - Reporting based on automated processes



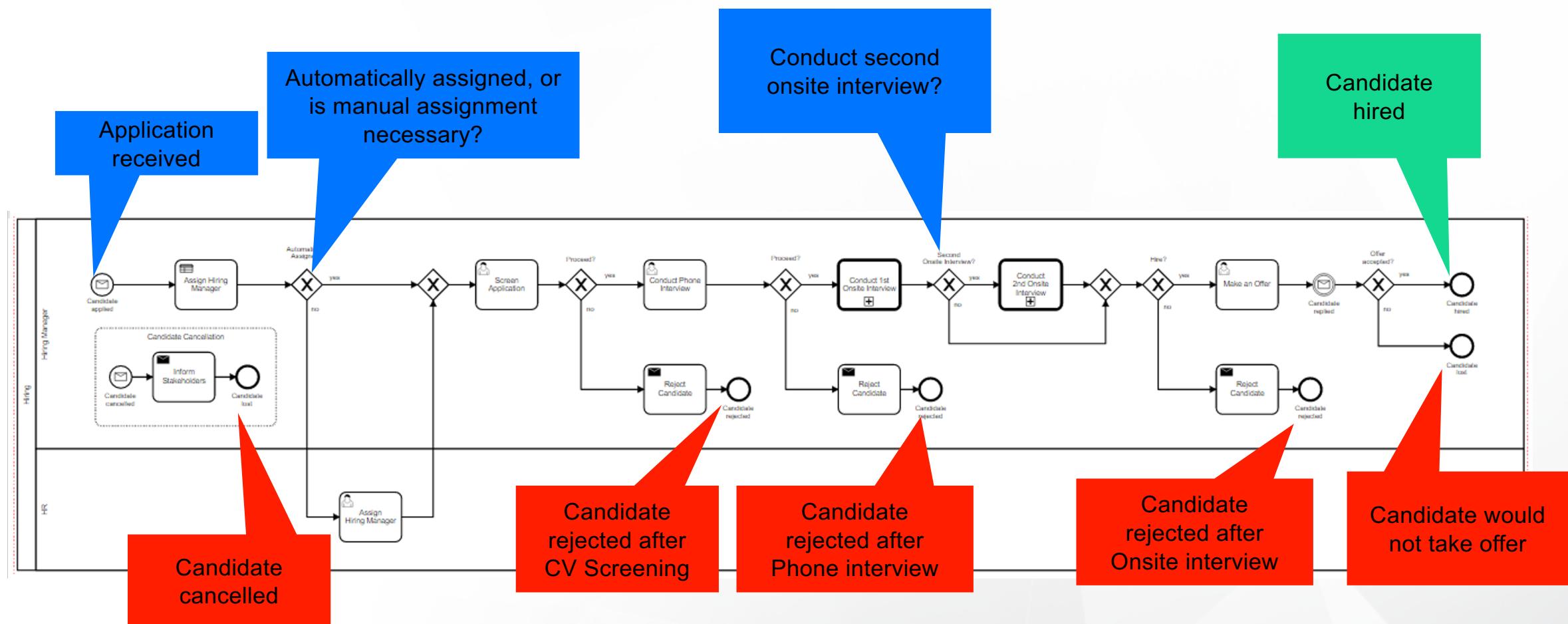
Camunda Platform und Optimize



Demo Example: Hiring



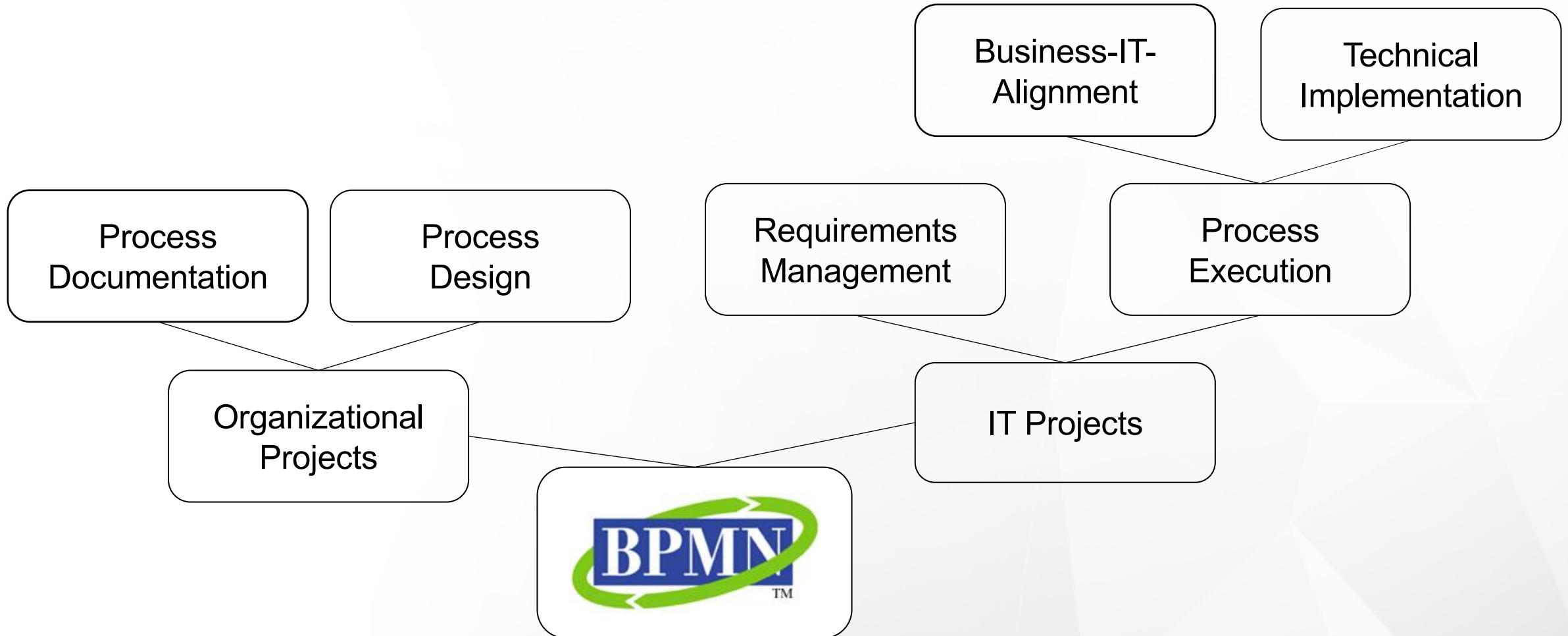
Hiring Example in Detail





Methods: BPMN Guidelines

Question: For which scenario do I want to use BPMN?

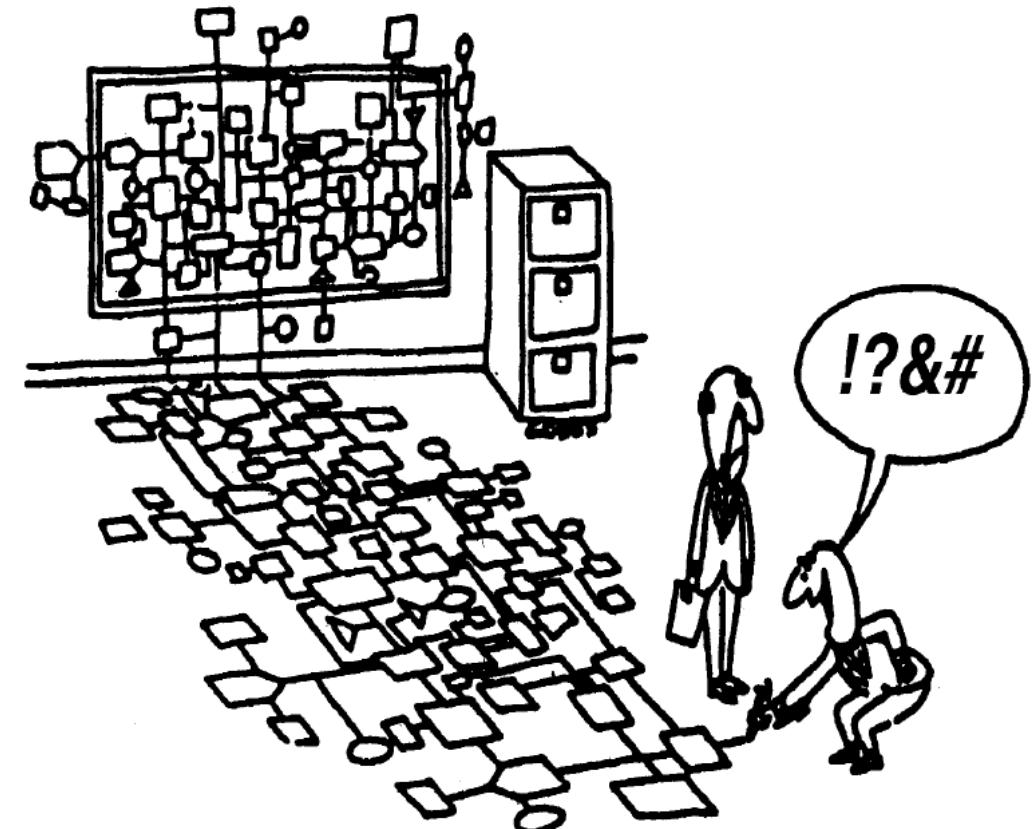


Why do I need a modeling guideline?

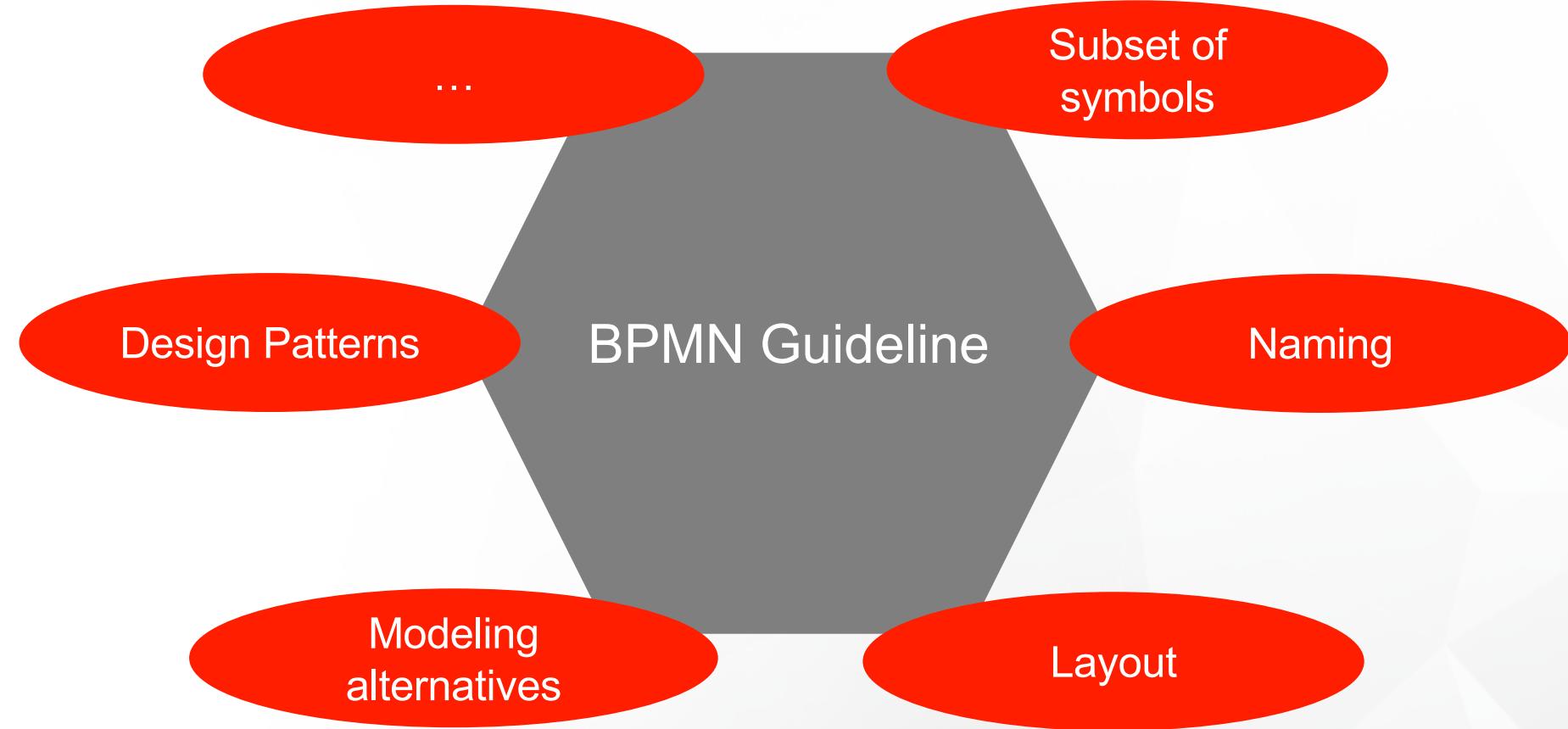
- BPMN is complex
- BPMN has many degrees of freedom
- A guideline provides a framework and orientation
- It helps to model faster and with better quality
- It helps to make models more consistent and therefore better comparable and easier to understand

In total: better readability and acceptance and at the same time formal quality and usability is assured.

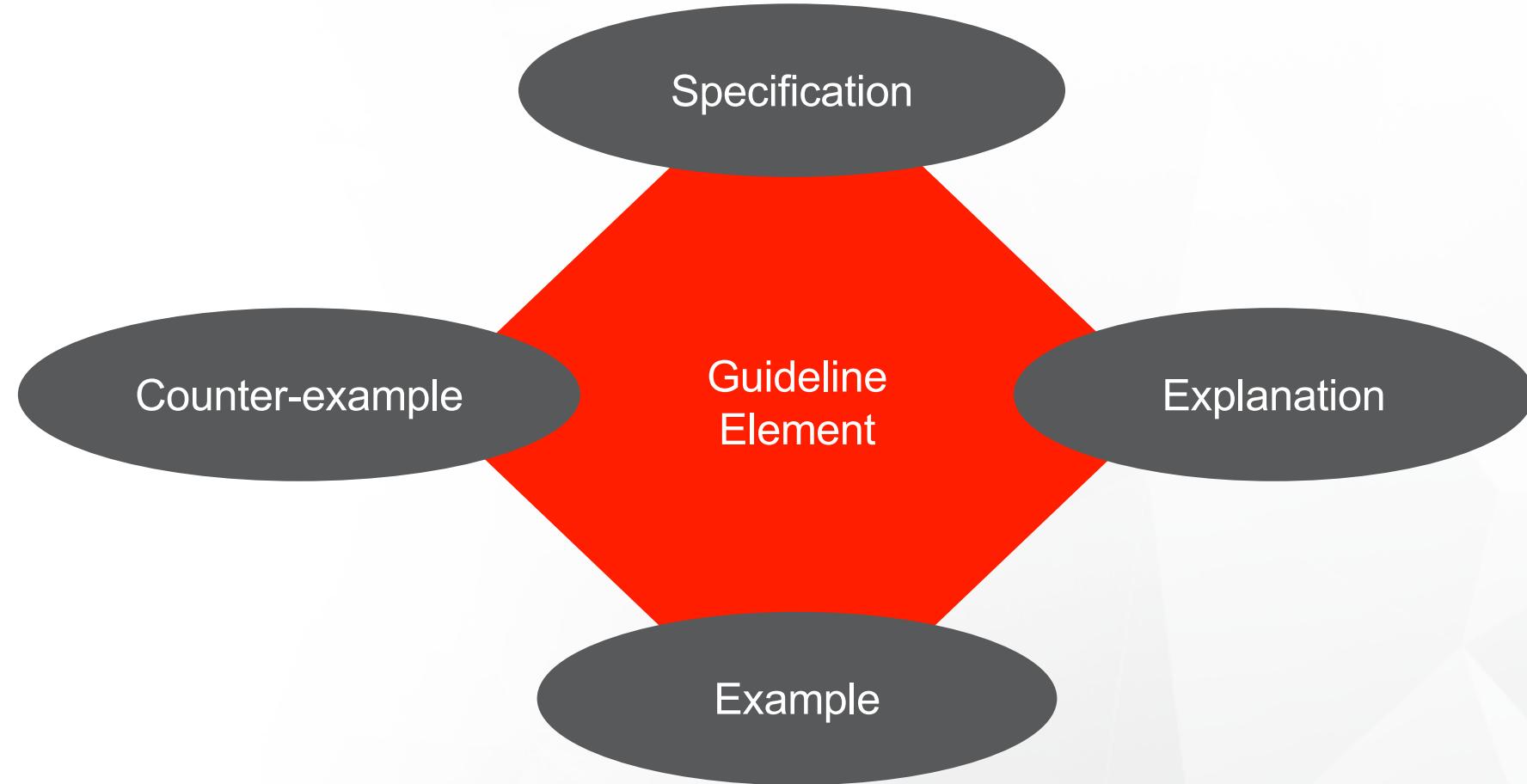
Important: The guideline depends on your goals.



Important parts of a BPMN guideline

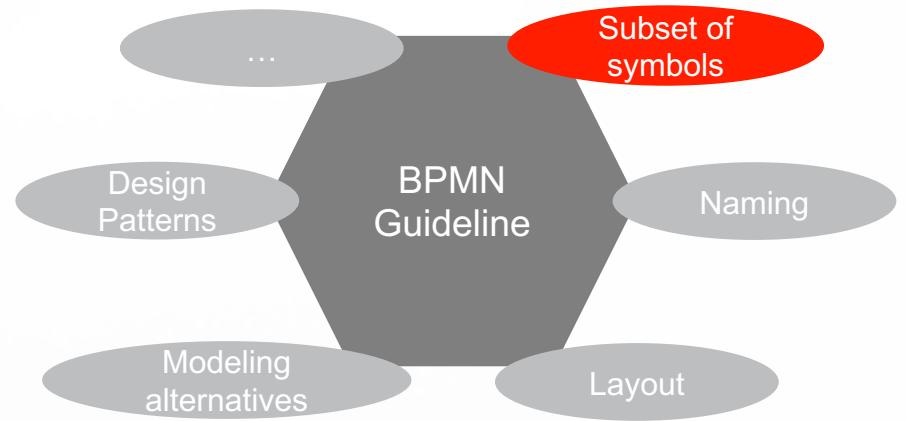


Aspects for describing a guideline element



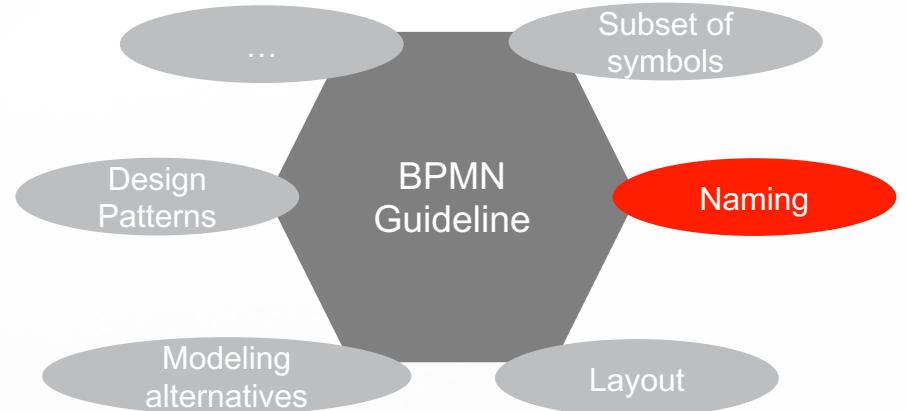
Subset of Symbols

- Define a subset of BPMN symbols
- Extend with own artefacts, if needed



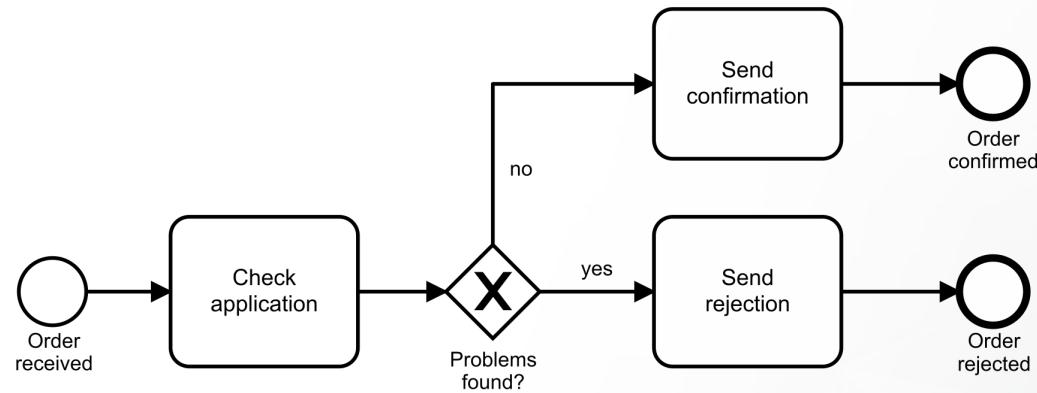
Naming

- Helps to make models more consistent
- Supports better readability
- Minimizes room for misinterpretation

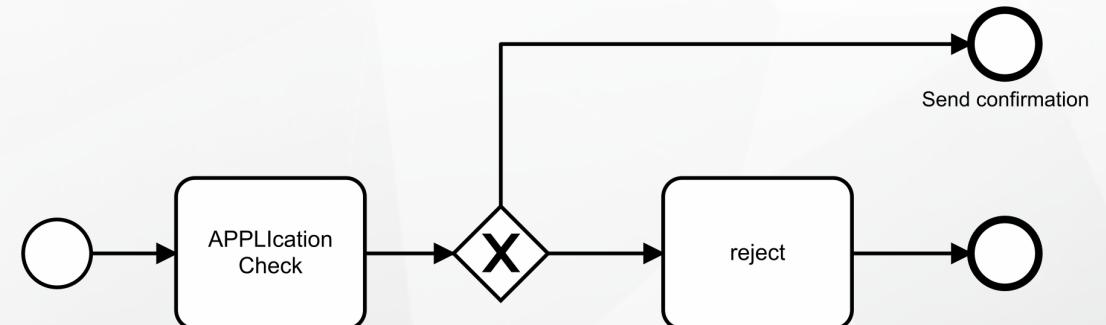


Example: Naming

- **Specification:** For naming elements in a BPMN model following rules shall apply:
Task: [object] + [verb]
Event: [object] + [state]
- **Explanation:** Consistent naming significantly improves readability and can prevent errors.
- **Example:**

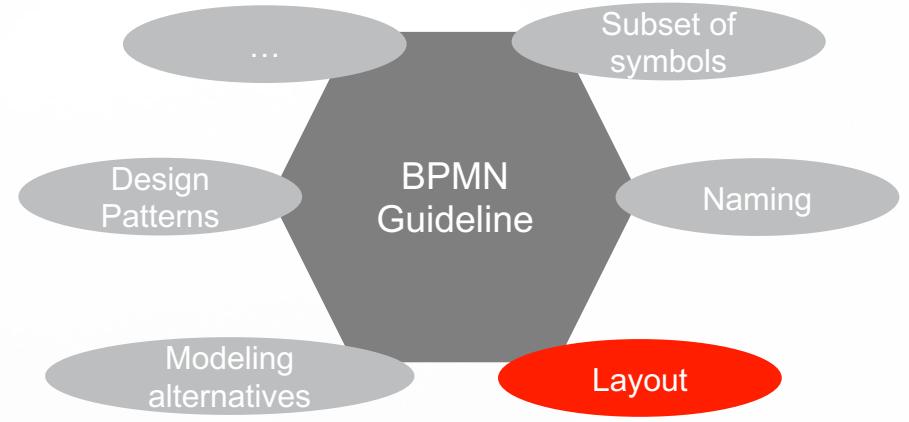


- **Counterexample:**



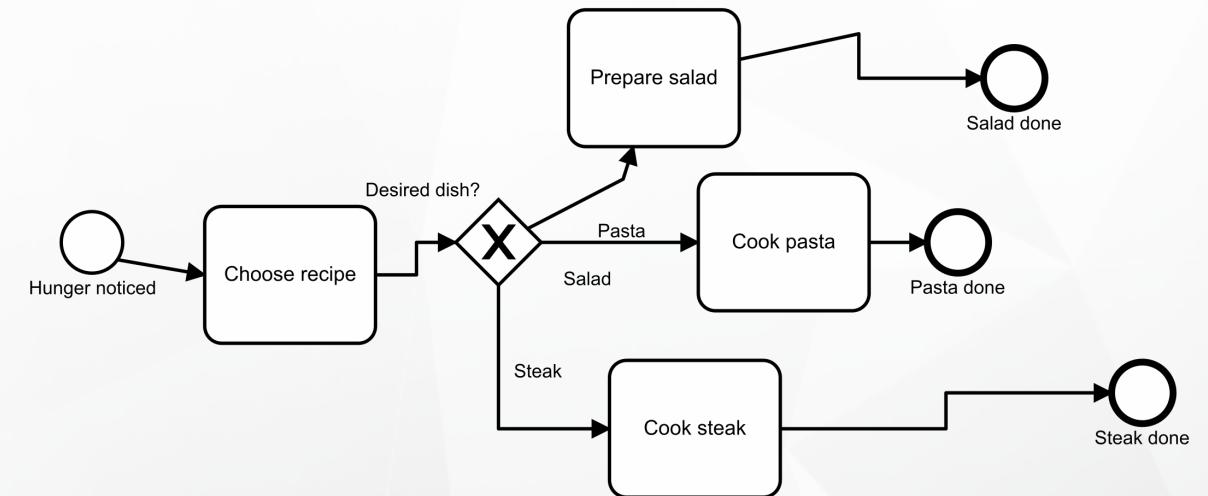
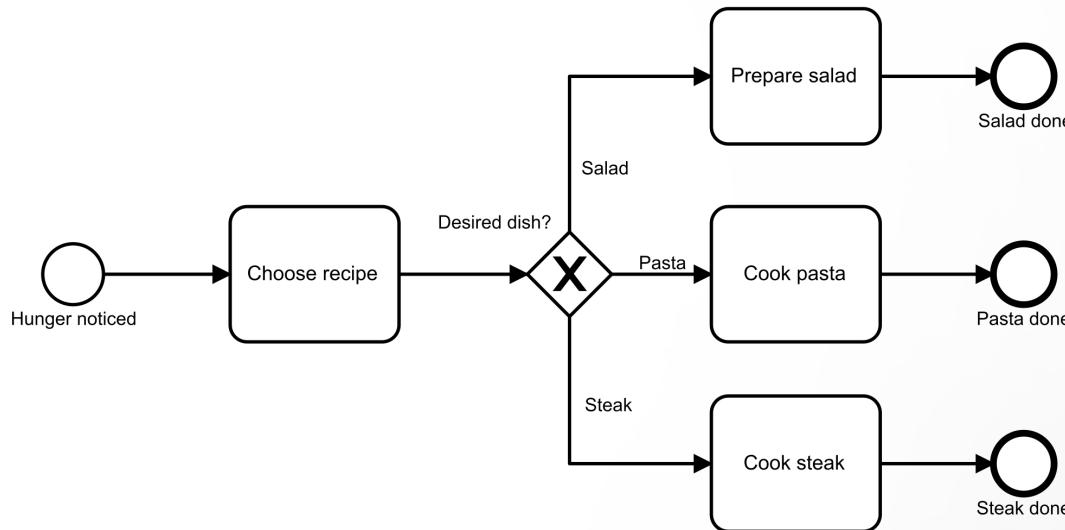
Layout

- The layout deals with the visual appearance of process diagrams
- It makes diagrams more uniform and more readable



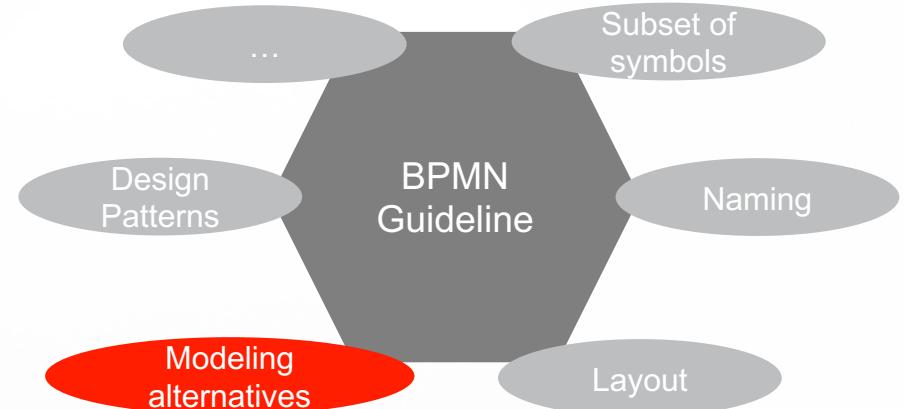
Example: Layout

- Specification: Always model in a grid layout
- Explanation: The arrangement of the elements in a grid layout increases the readability
- Example:
 - Counterexample:



Modeling alternatives

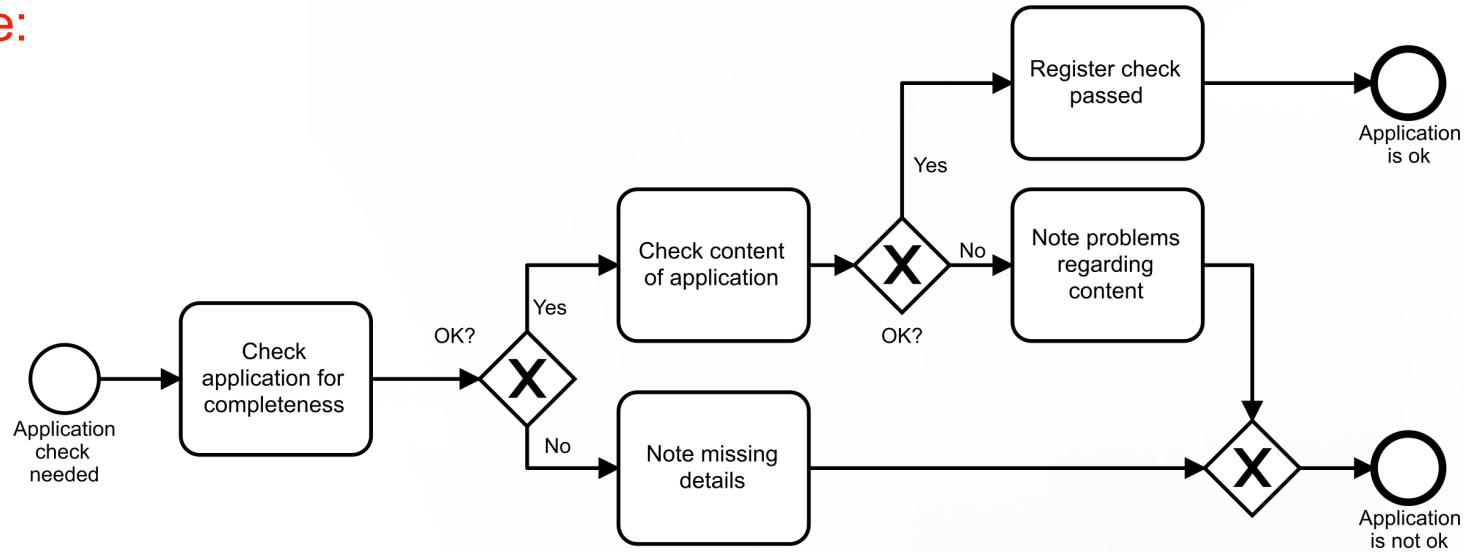
- A guideline that explains how certain situations should be modeled
- Basic situations with no reference to concrete content-related processes.



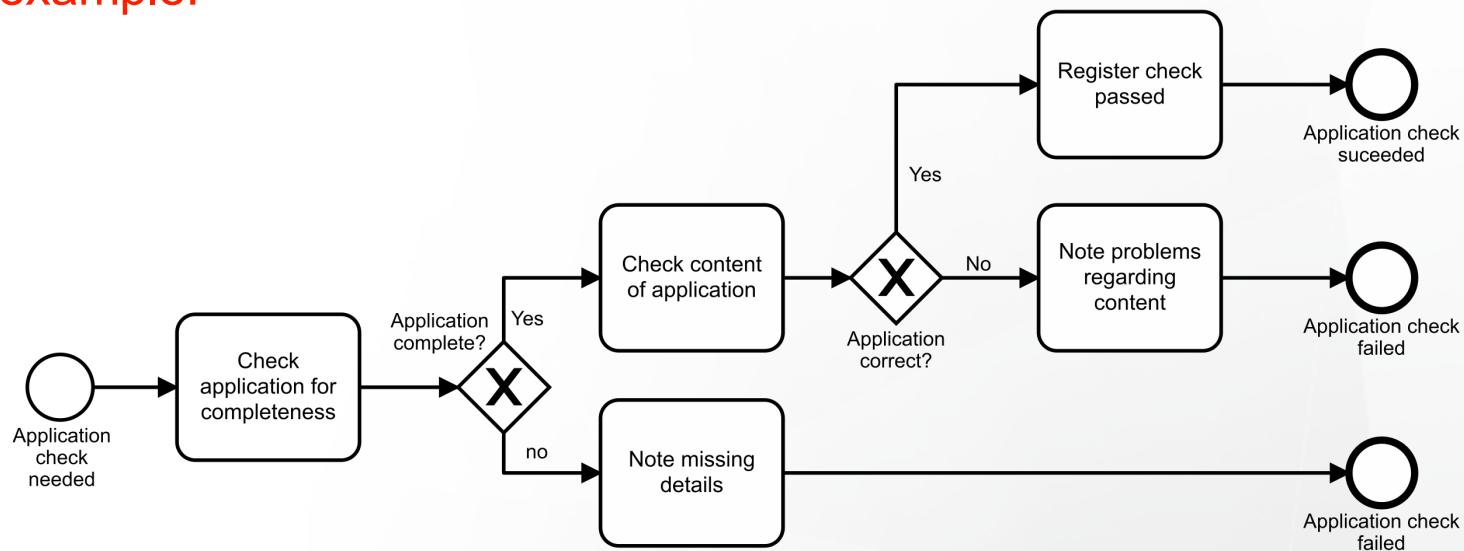
Example: Modeling alternatives

- **Specification:** End events with similar meaning should be summarized in one symbol. End events with differing meaning should be modeled separately.
- **Explanation:** This way, the viewer will know faster that different end states are possible, and he or she will recognize the respective state sooner.

- Example:

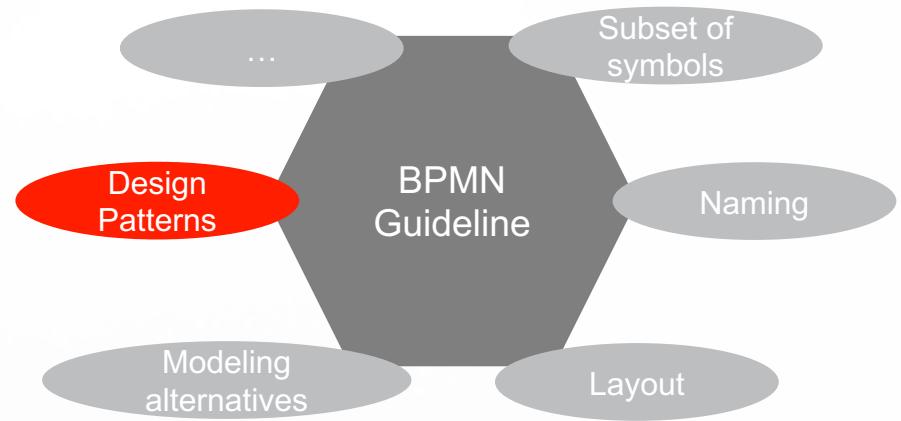


- Counterexample:



Design Patterns

- Like a recipe: guide for certain modeling situations
- Recommendatory character (more flexible than other guidelines)
- Different structure than other guidelines:
 - **Requirement:** Describes in which situation the design pattern can be helpful
 - **Recommendation:** Refers to a pattern, then recommends it
 - **Available design patterns:** The design patterns suitable for the situation are shown through examples



Example: Design Patterns (two-stage escalation)

Requirement:

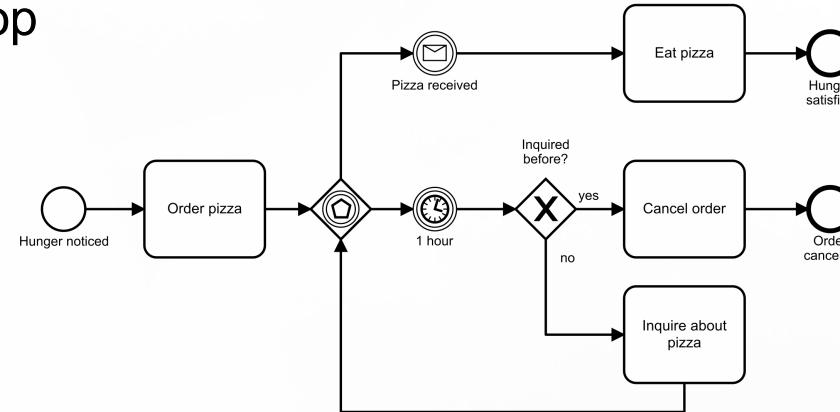
- I want my interaction partner to do something, so I send her a message (e.g.: an invoice to be paid, an item to be delivered, an instruction to be executed, ...)
- My interaction partner does not react.
- I remind her a reminder for my request, and I may set a new deadline (if necessary, I can repeat this several times)
- In case of no reaction: Escalation (e.g.: forwarding invoice to debt collector, cancelling the order, informing a superior of non-performance)

Recommendation:

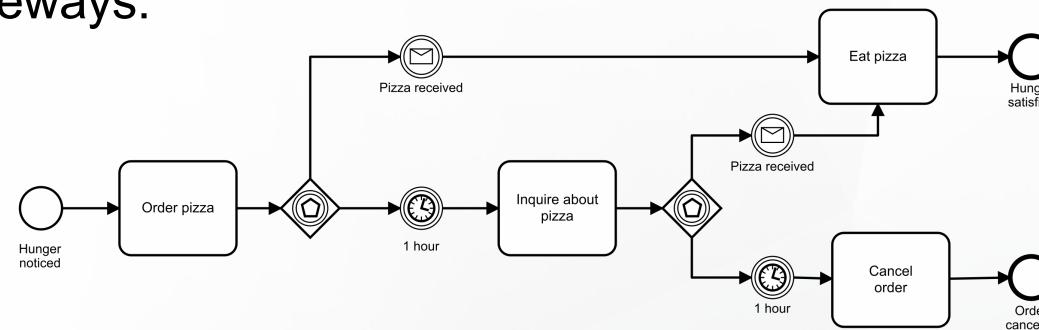
- „Event-based gateway with loop“
 - Clear, easily understood and formally correct
 - Alternative: „Attached timer events“ (if the selected engine cannot interpret the event-based gateway)

Available Design Patterns:

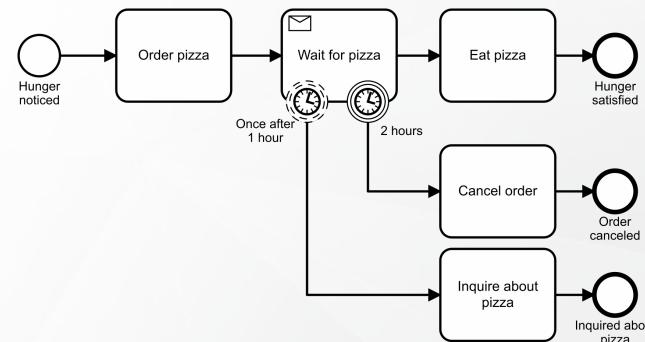
- Event-based gateway with loop



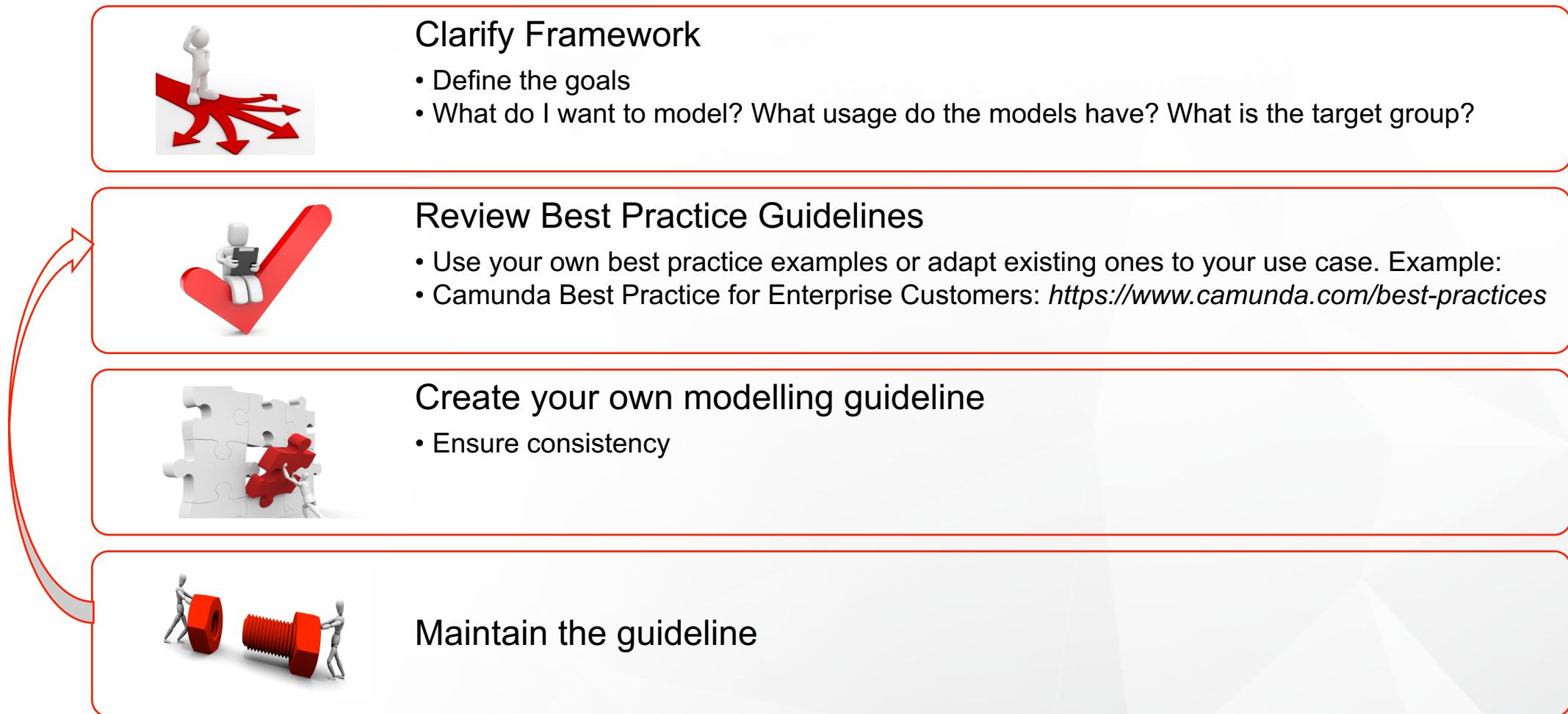
- Chain-link of event-based gateways:



- Attached timer events:



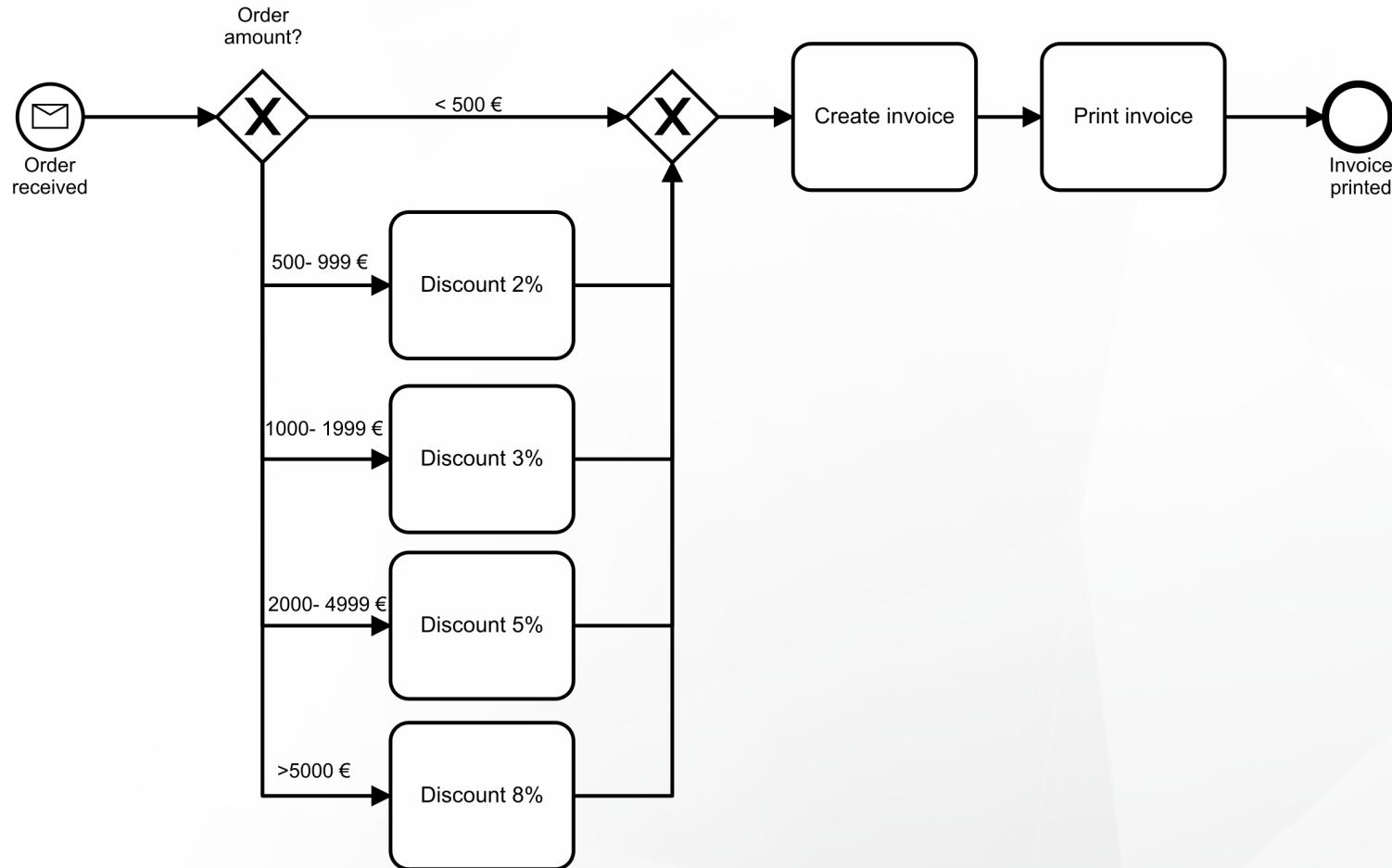
Pathway to own guideline





Business rules with DMN

Discount rules in a BPMN model?



Discount rules in a DMN table

Discount			
discountDetermination			
U	Input +	Output +	Annotation
	order amount	discount	
	long	double	
1	< 500	0	-
2	[500..999]	2	-
3	[1000..1999]	3	-
4	[2000..4999]	5	-
5	> 5000	8	-
+	-	-	-

DMN standard

- Decision Model and Notation
- Standard of OMG for decisions/business rules

„The primary goal of DMN is to provide a **common notation** that is readily understandable by all **business users**, from the business analysts needing to create initial decision requirements and then more detailed decision models, to the **technical developers** responsible for automating the decisions in processes, and finally, to the business people who will manage and monitor those decisions. DMN creates a standardized **bridge for the gap between the business decision design and decision implementation**. DMN notation is designed to be **useable alongside the standard BPMN** business process notation.”

DMN tables can be executed

- Technical attributes enable the direct execution of desicion tables
- Role seperation between structure (columns) and rules (rows) is possible

Discount			
discountDetermination			
U	Input +	Output +	Annotation
	order amount	dicount	
	order.amount	discount	
	long	double	
1	< 500	0	-
2	[500..999]	2	-
3	[1000..1999]	3	-
4	[2000..4999]	5	-
5	> 5000	8	-
	+	-	-

Structure of a DMN table

The diagram illustrates the structure of a DMN table. A red box labeled "Hit Policy" points to the "discountDetermination" section above the table. Another red box labeled "Input Name + Expression" points to the "order amount" entry in the "Input" column. A third red box labeled "Rules" points to the five rows of rules below the header. A red box labeled "Table name + ID" points to the "Discount" header at the top left. A red box labeled "Output Name + Expression" points to the "dicount" entry in the "Output" column.

U	Input +	Output +	Annotation
	order amount	dicount	
	order.amount	discount	
	long	double	
1	< 500	0	-
2	[500..999]	2	-
3	[1000..1999]	3	-
4	[2000..4999]	5	-
5	> 5000	8	-
	+	-	-

Hit Policies

Single

- **U(nique)**: Exactly one rule applies
- **A(ny)**: More rules can apply but don't necessarily provide the same outcome
- **P(riority)**: Rule with the highest priority (result field) is selected
- **F(irst)**: First applicable rule is selected (order of rows)

Multiple

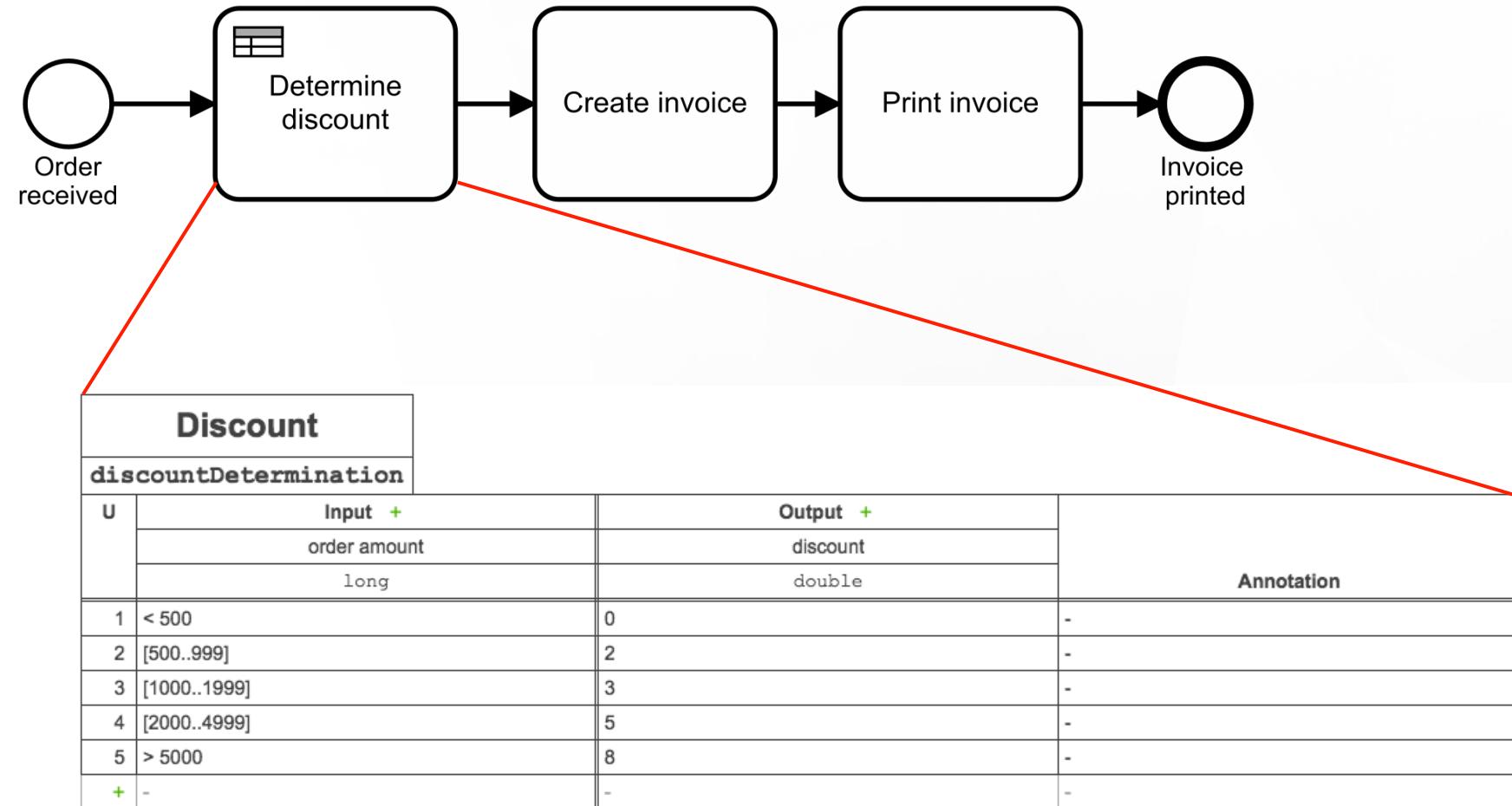
- **O(utput Order)**: List of results in order of priority (result fields)
- **R(ule Order)**: List of results in order of rules
- **Collect**: List of results without an order, can be combined with operators +
(Sum); < (Min); > (Max); # (Amount)

More examples from the DMN specification

Applicant Risk Rating			
U	Applicant Age	Medical History	Applicant Risk Rating
1	> 60	good	Medium
2		bad	High
3	[25..60]	-	Medium
4	< 25	good	Low
5		bad	Medium

Student Financial Package Eligibility				
R	Student GPA	Student Extra-Curricular Activities Count	Student National Honor Society Membership	Student Financial Package Eligibility List
1	> 3.5	>= 4	Yes	20% Scholarship
2	> 3.0	-	Yes	30% Loan
3	> 3.0	>= 2	No	20% Work-On-Campus
4	<= 3.0	-	-	5% Work-On-Campus

Discount rules are referred in a BPMN model



Frequency of modification

Business processes



Modification of business processes need a high effort of change management or process implementation

Seldom
(months or years)

Structure (columns)

Order Discount		orderDiscount	
-	Input +	Output +	Annotation
	Order Amount	Discount	
	order.amount	discount	
	long	int	
1	< 500	0	-
2	[500..999]	2	-
3	[1000..1999]	3	-
4	[2000..4999]	5	-
5	>= 5000	8	-

Modification of columns require effort to integrate input data sources

Seldom
(months)

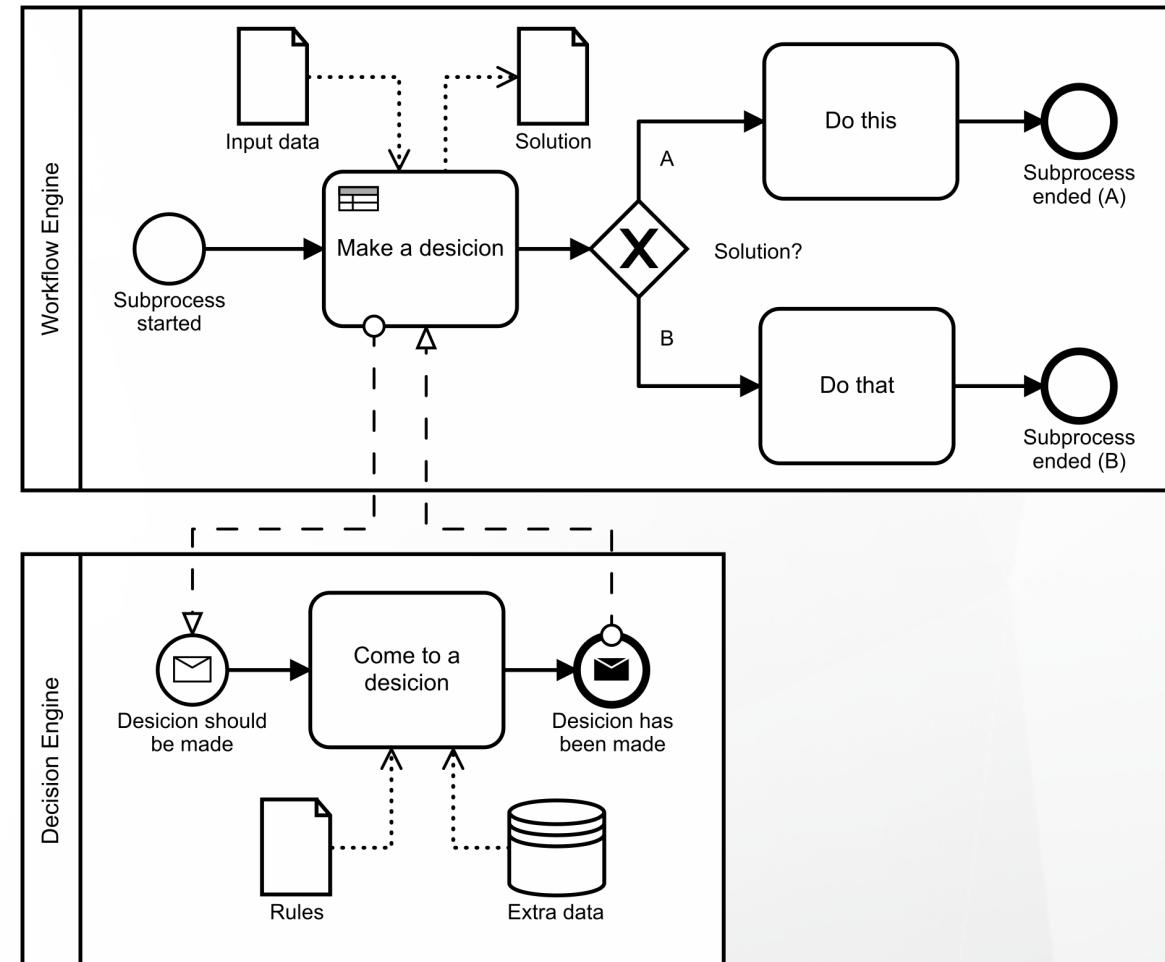
Rules (rows)

Order Discount		orderDiscount	
-	Input +	Output +	Annotation
	Order Amount	Discount	
	order.amount	discount	
	long	int	
1	< 500	0	-
2	[500..999]	2	-
3	[1000..1999]	3	-
4	[2000..4999]	5	-
5	>= 5000	8	-

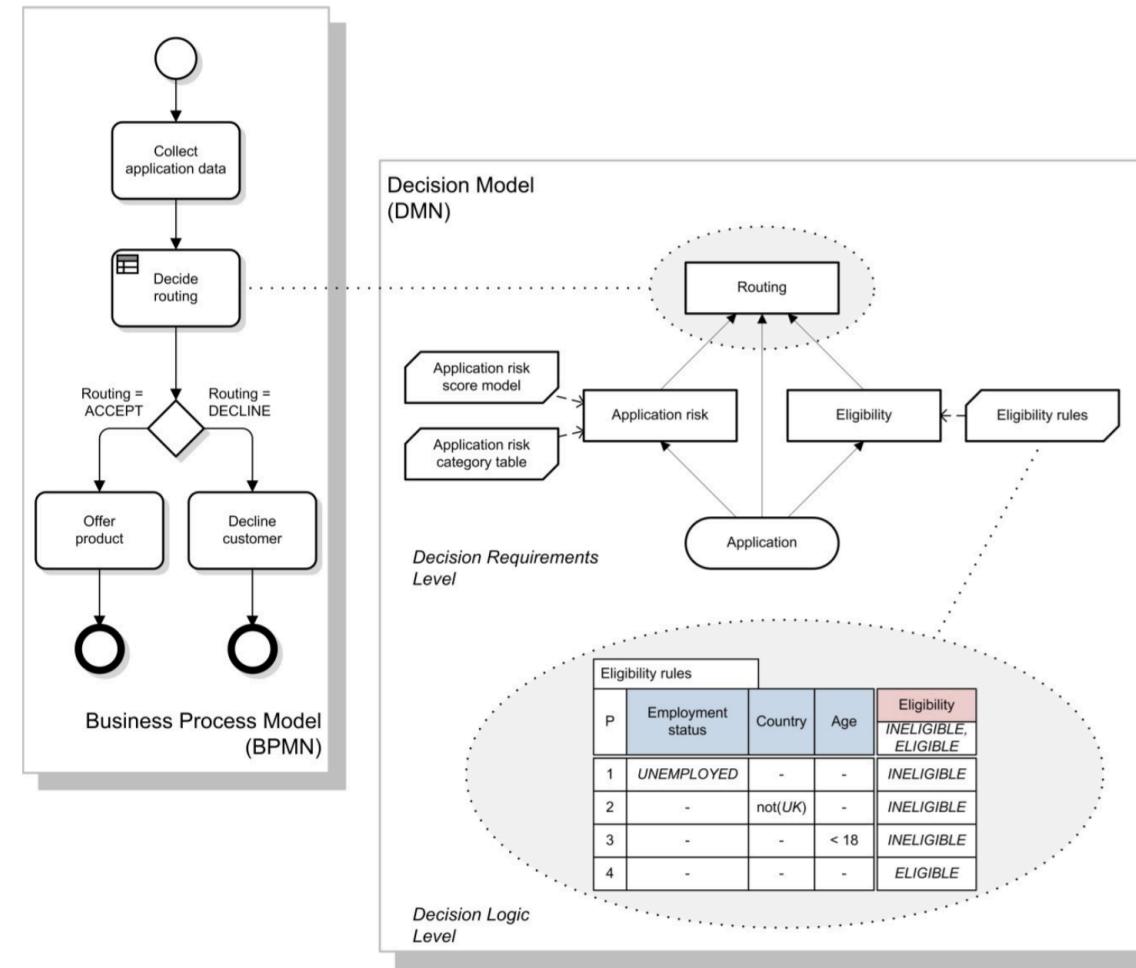
Modification of rules are easy, however validation and testing shouldn't be forgotten

Often
(weeks or months)

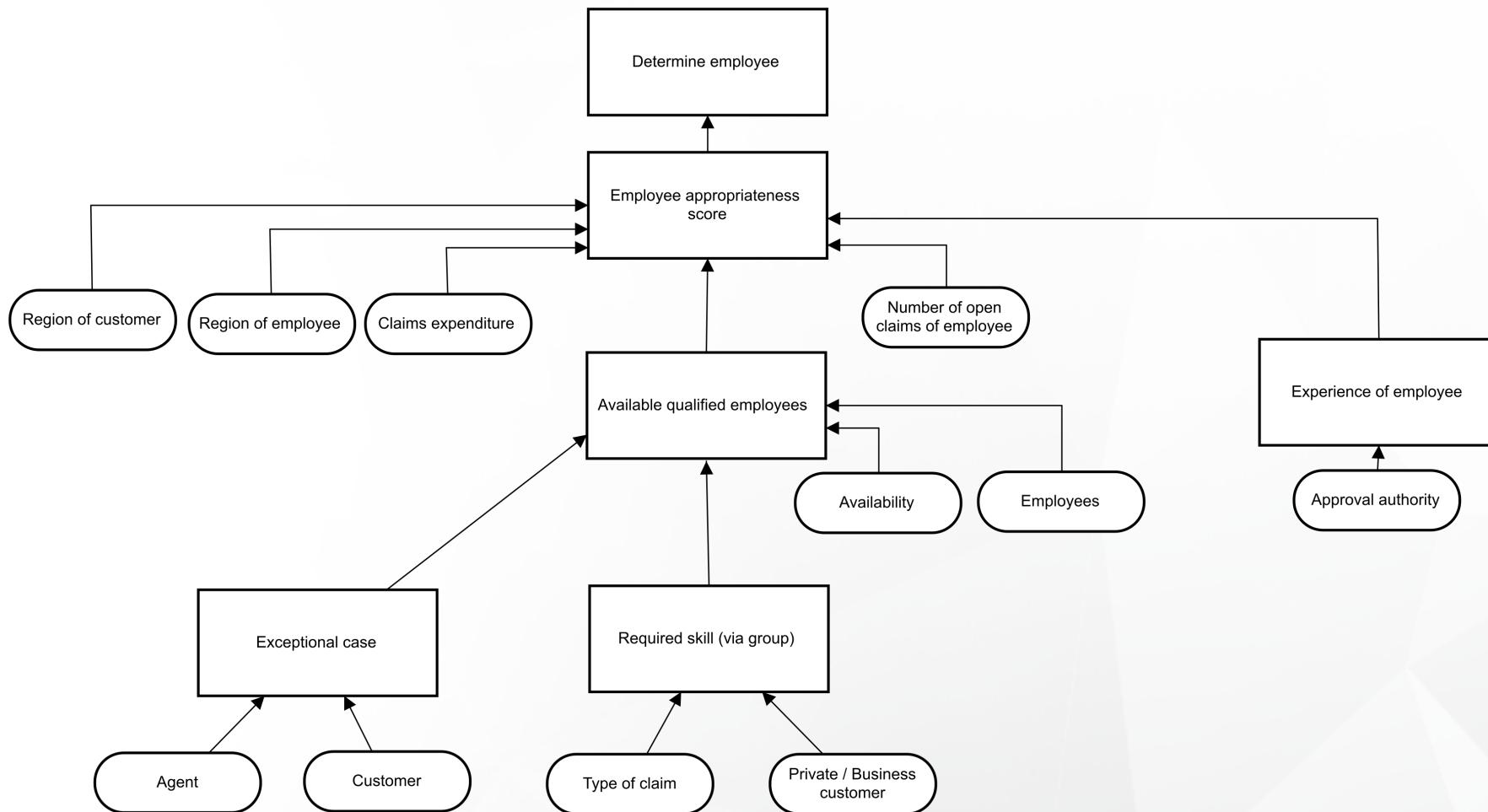
Process engine and decision engine



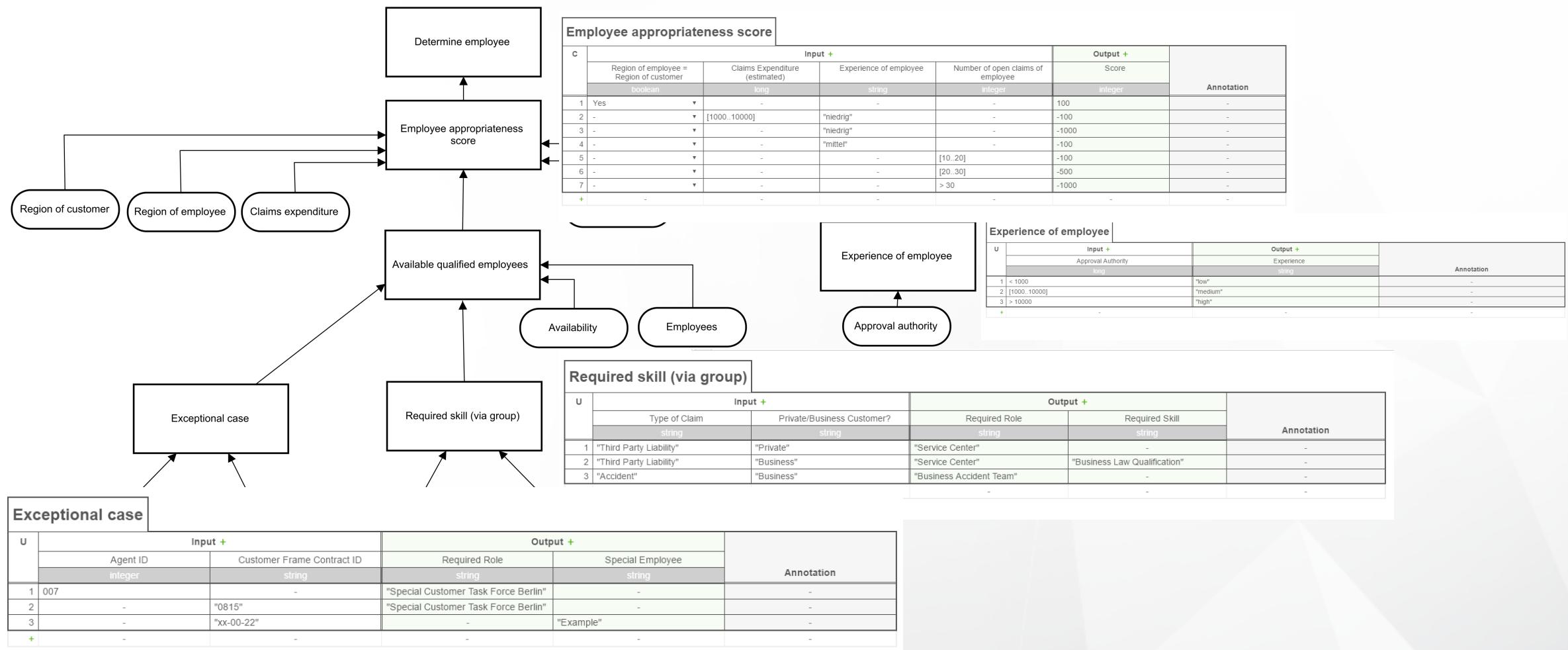
DMN defines „Decision Requirements Levels“



DRD example: skill-based assignment



Decisions in a DRD can be tables





Conversation Models



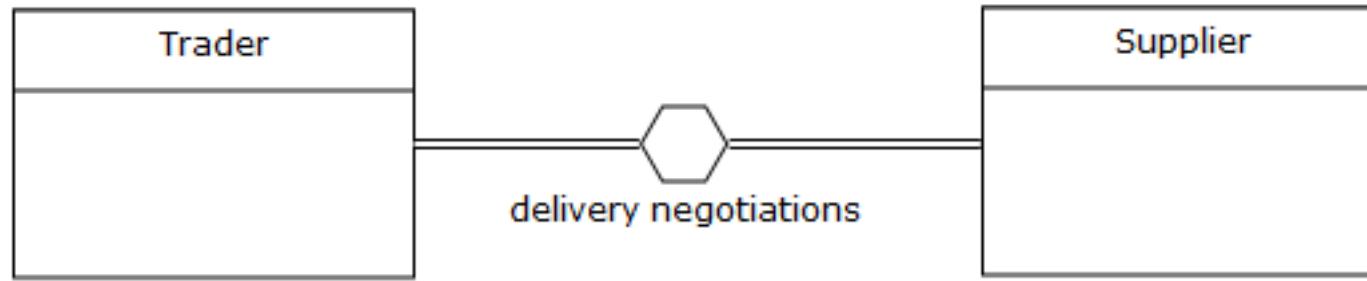
What is a conversation diagram?

- A conversation diagram is kind of a simplified collaboration diagram
- Conversation diagrams show the message flows between process participants in an abstract way
- Conversation diagrams complement collaboration and/or choreography diagrams

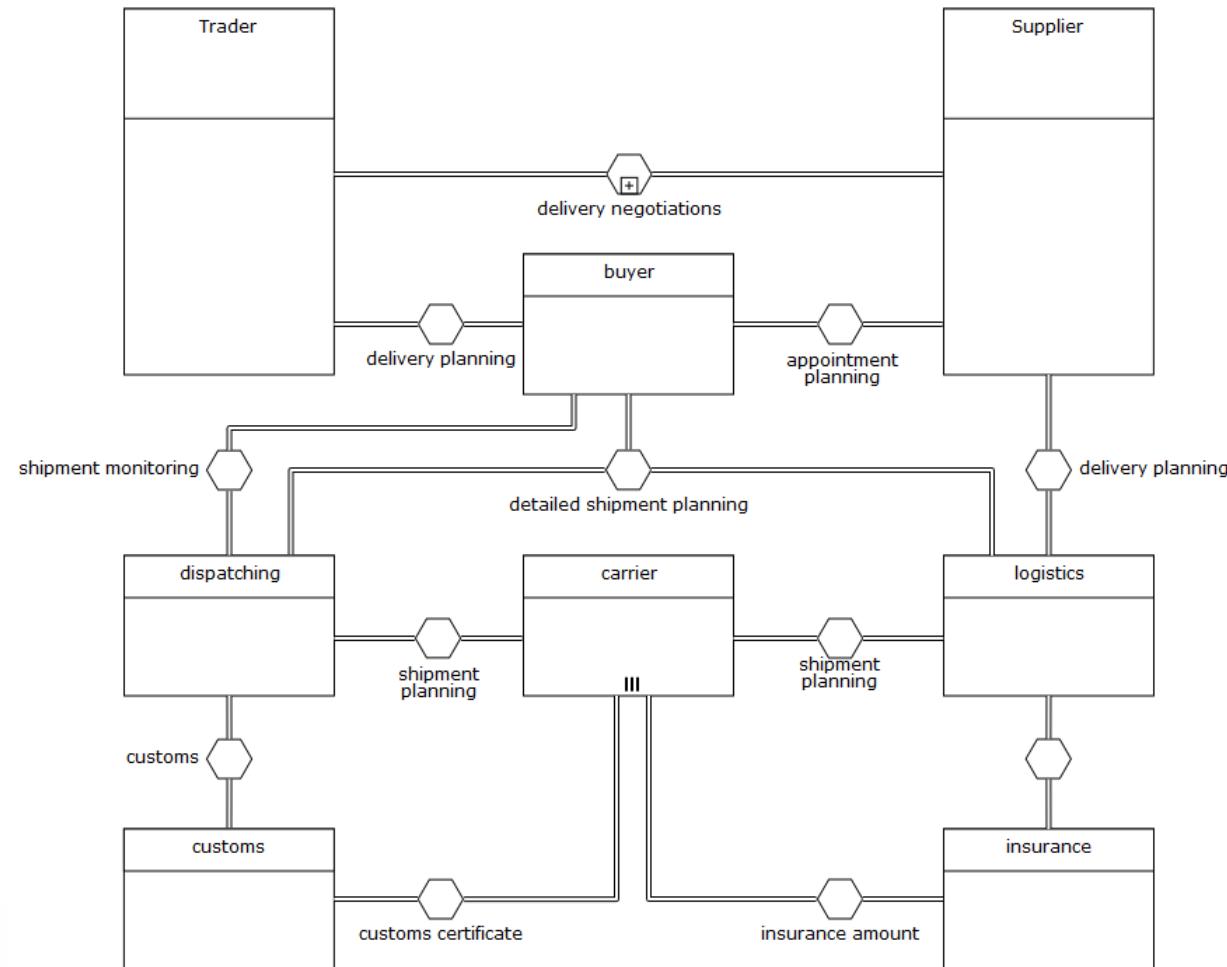
Use of conversation diagrams

- Depicting an abstract and logically correlated message flows
- Abstracting large and complex application scenarios with many message exchanges

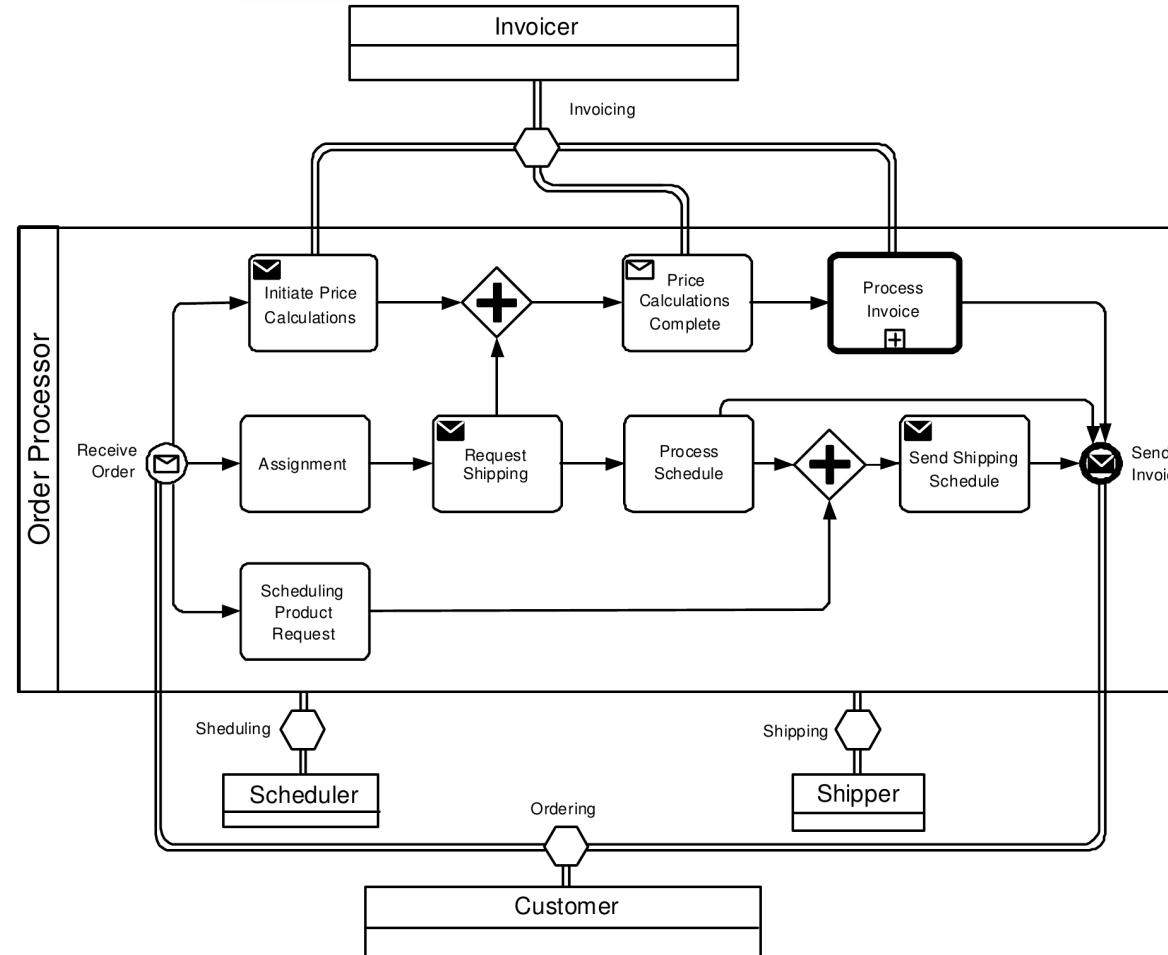
Example 1



Example 2 – shipment of goods



Conversation Diagram = Collaboration Diagram





Choreography diagrams

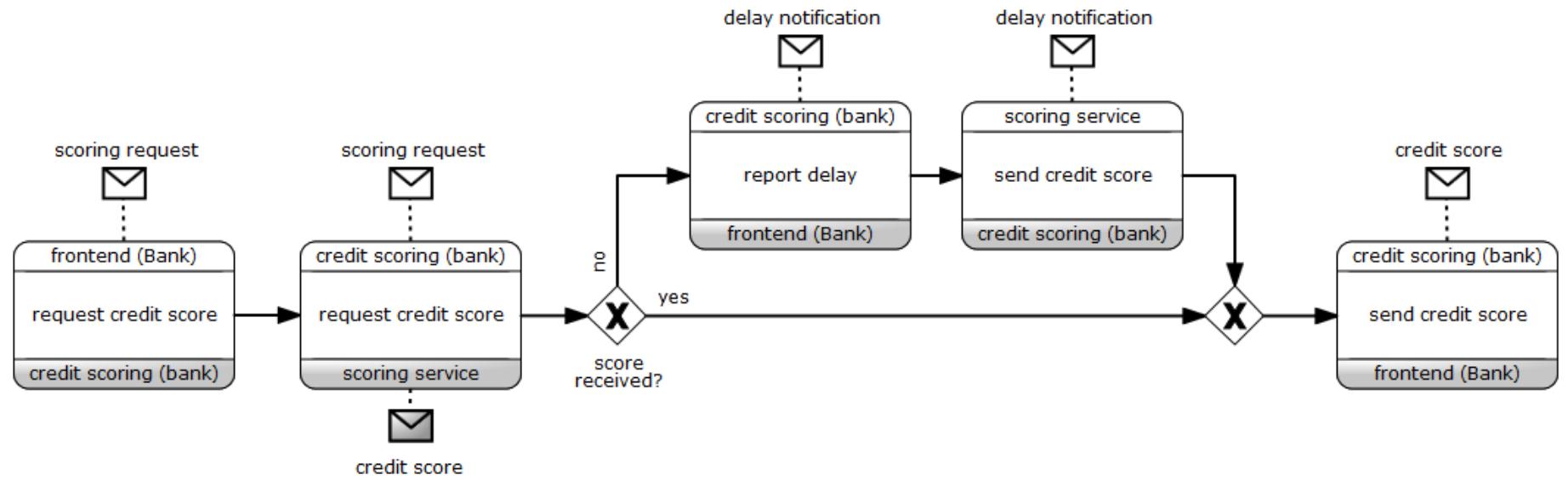
What is a choreography?

- A choreography shows relations between two (or more) process participants that exchange messages
- It describes the desired communication behavior of the parties involved
- It describes the message exchange as well as the logical sequence of the message exchanges

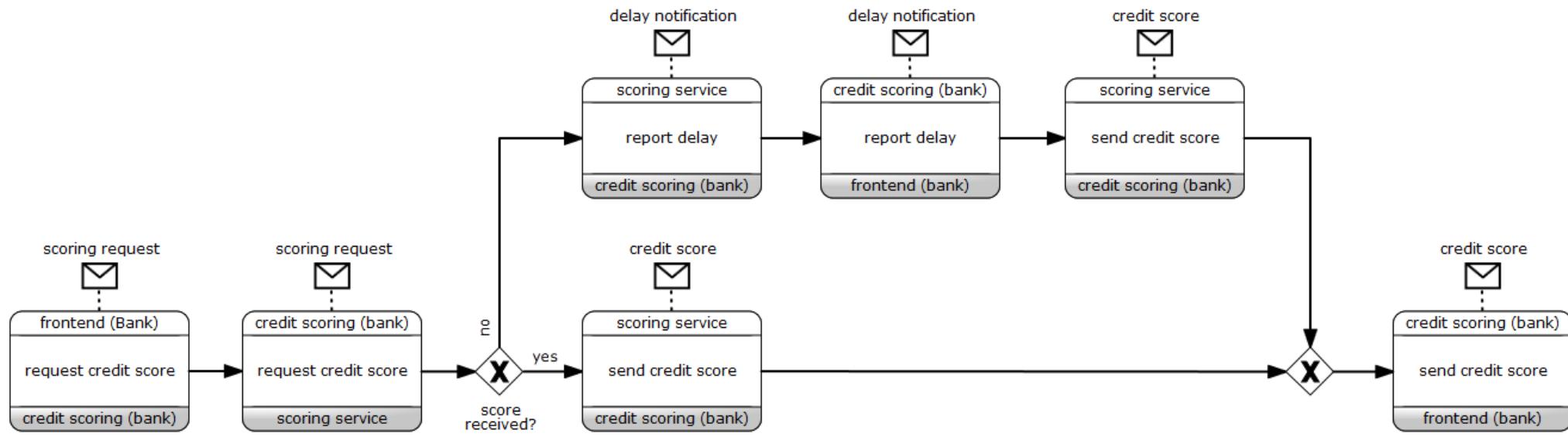
Use of choreography diagrams

- The focus points to message exchanges and all tasks involved in these exchanges
- Process interfaces can be depicted easily
- The diagrams show all parties involved in the exchanges

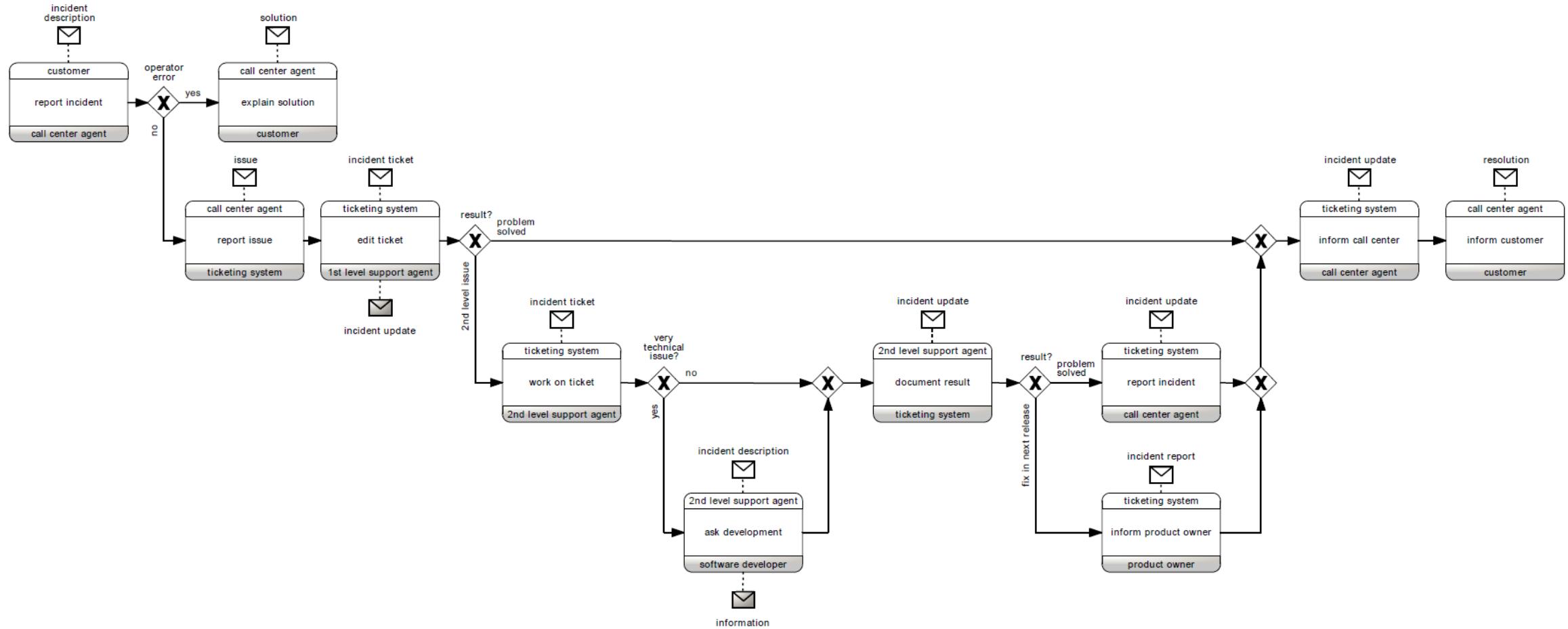
Example 1 - credit scoring (synchronous communication)



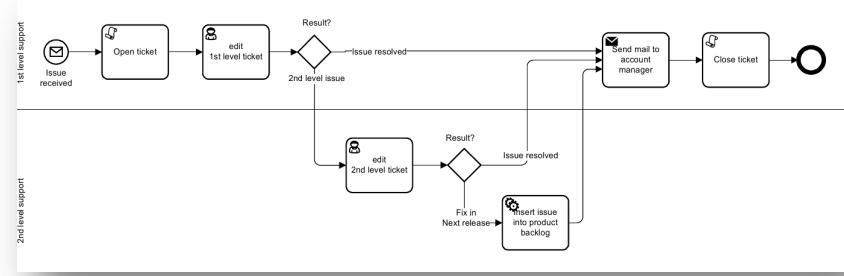
Example 2 - credit scoring (asynchronous communication)



Example 3 - incident management

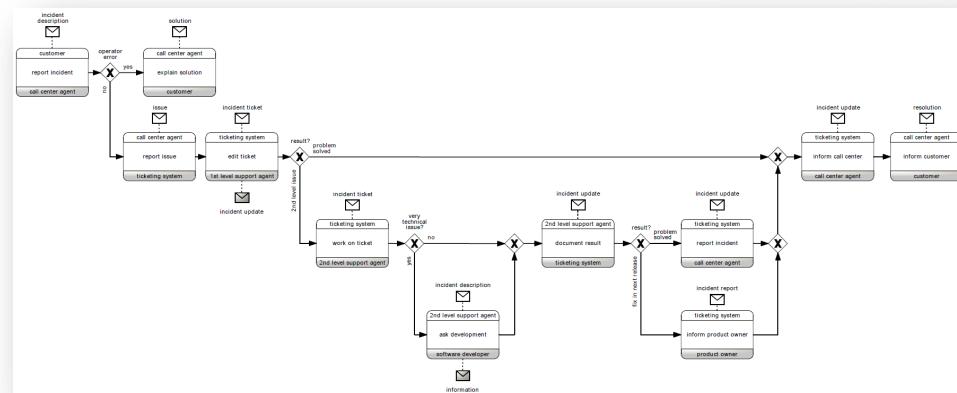


Summary: Basic Types of End-to-End BPMN Models

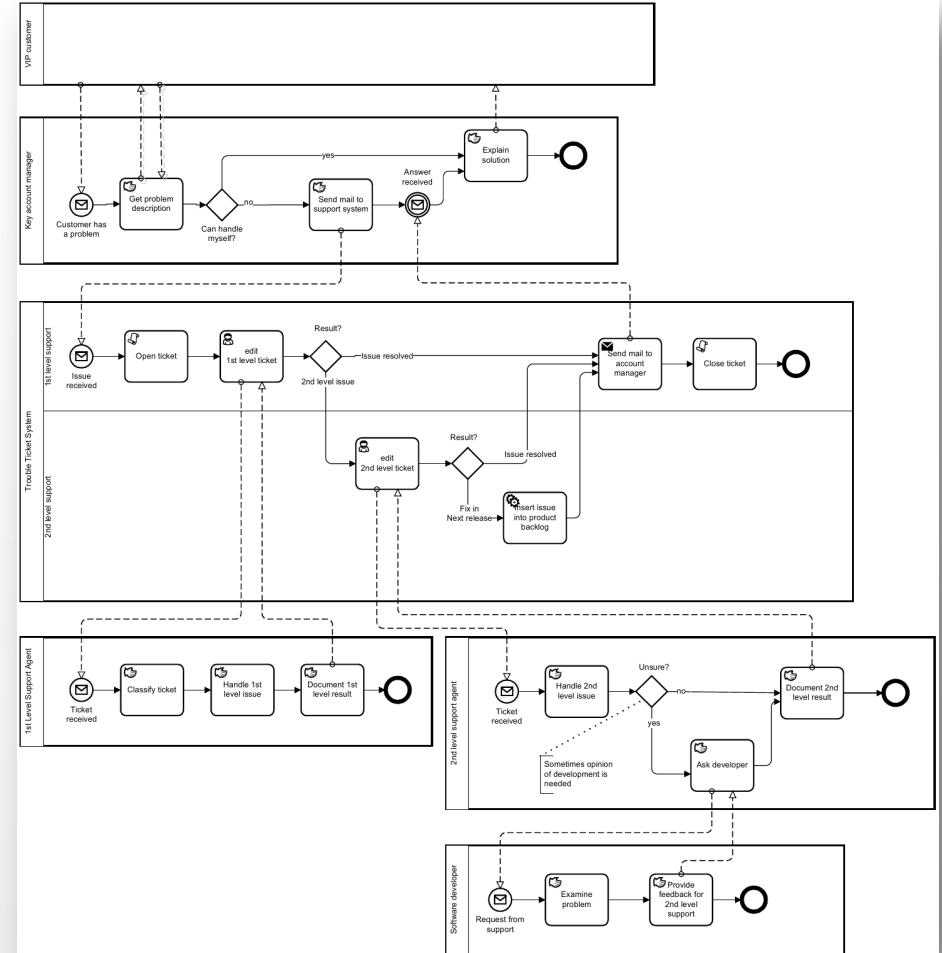


Process

Choreography



Collaboration



Camunda training offering

Overview trainings

Camunda BPM Overview (1 day)

- Introduction into BPMN
- Hands on Camunda BPM
- Monitoring & Reporting
- How to start with Camunda

OCEB training (2 days)

- Prepare for the OCEB BPM certification

Modeling trainings

BPMN (3 days)

- BPMN in detail
- Implementing BPM-projects with BPMN
- BPMN real world examples
- BPMN in context of DMN

DMN (1 day)

- DMN in detail
- Decision design
- Complex decisions with DRDs
- DMN in context of BPMN

Development trainings

Camunda Platform for developers (3 days)

- Introduction BPMN
- Camunda Platform Architecture
- Automating processes with Camunda
- Testing, deployment, versioning

Camunda Platform DevOps (2 days)

- Camunda Platform installation
- Monitoring & alarming
- Process version migration
- Advanced DevOps topics

You'll find further information under
camunda.com/de/services/training

Feedback Time!



<https://camunda.com/services/training/feedback>

CAMUNDA

Training
completed

Thank you
and good luck
with BPMN!