

**HOCHSCHULE
HANNOVER**
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS

–
Fakultät IV
Wirtschaft und
Informatik

Choreographed microservices

*Towards the automation of BPMN 2.0
choreography diagrams*

C. Schulze und A. Link, April 2024

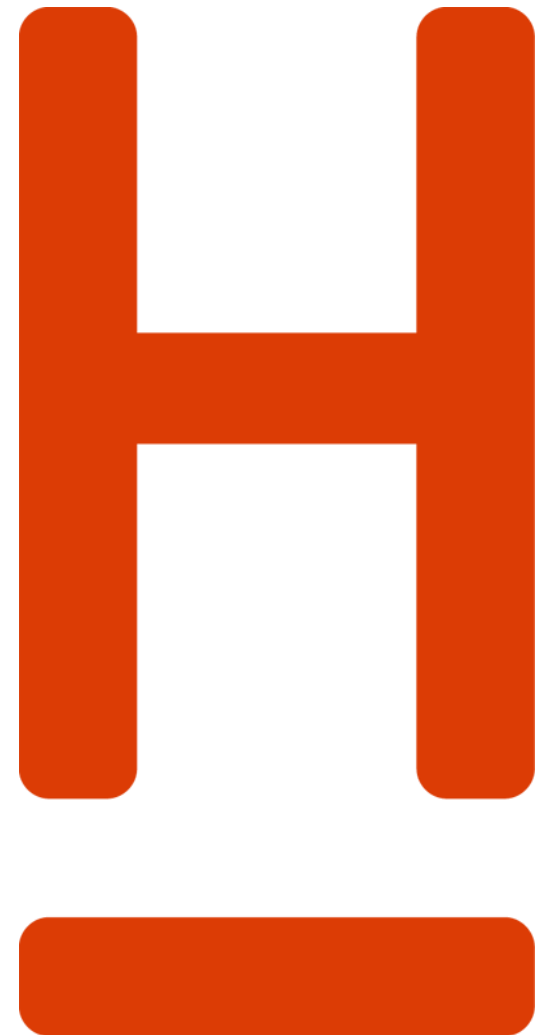


Table of contents

Chapter 1	Presentation
Chapter 2	Recap
Chapter 3	Choreography patterns
Chapter 4	Insight into the choreography grammar
Chapter 5	Outlook and next steps



Table of contents

Chapter 1	Presentation
Chapter 2	Recap
Chapter 3	Choreography patterns
Chapter 4	Insight into the choreography grammar
Chapter 5	Outlook and next steps



Presentation

CC_ITM@HsH

Competence Center Information Technology and Management (CC_ITM):

- Institute at the University of Applied Sciences and Arts, Hanover with a focus on research,
- Cooperation between HsH and regional companies in the insurance industry,
- Practical and scientific: topics of information processing and information technology.
- Topics covered:
 - Service Computing,
 - Rule Management,
 - **Workflows/business processes,**
 - Security,
 - etc.



Presentation

Presenters

Christin Schulze (B.Sc.):

- Master's student at the University of Applied Sciences and Arts, Hanover the computer science department
- Employee of the CC_ITM research institute
- Tutor for Software Engineering 3, Database Systems 1 & 2, Distributed Information Systems
- Tutor for bachelor projects in the field of distributed systems, cloud applications, UI/UX

Alexander Link (B.Sc.):

- Master's student at the University of Applied Sciences and Arts, Hanover the computer science department
- Employee of the CC_ITM research institute
- Cool guy



Table of contents

Chapter 1	Presentation
Chapter 2	Recap
Chapter 3	Choreography patterns
Chapter 4	Insight into the choreography grammar
Chapter 5	Outlook and next steps



Recap

Microservices Reference Architecture for Insurance Companies

Current work:

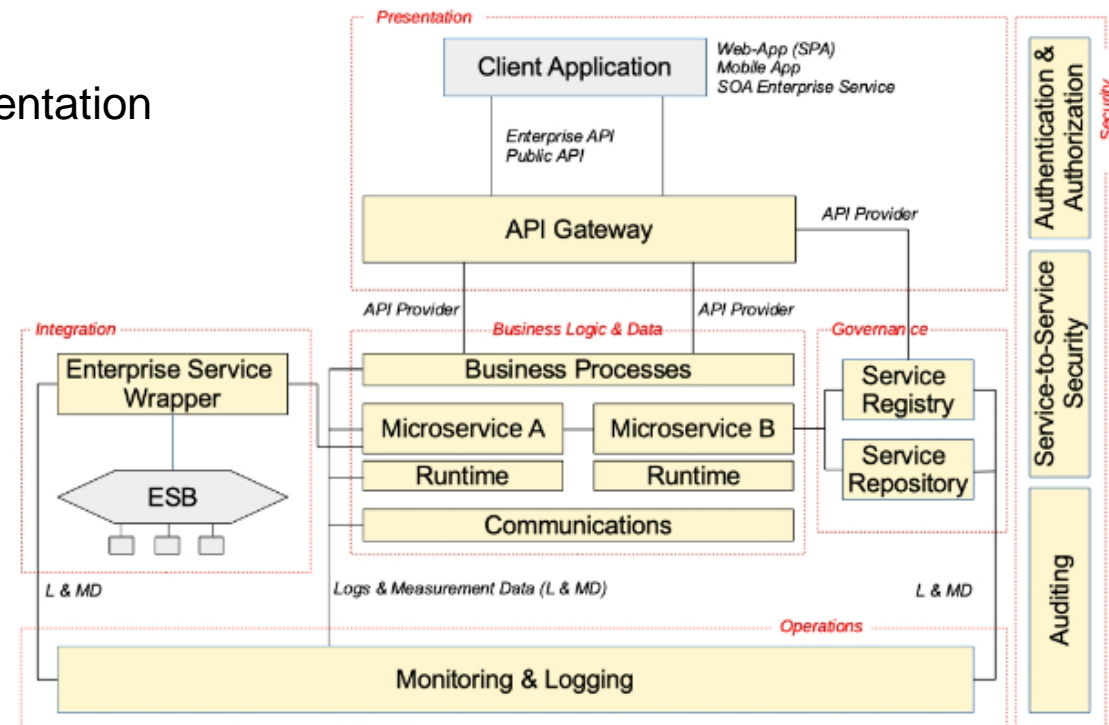
Further development of the "Microservices Reference Architecture for Insurance Companies (RaMicsV)"

Development, evaluation and prototypical implementation of the topic area Business Logic & Data

Problem:

How can business processes/workflows be implemented using multiple microservices?

How can BPMN 2.0 choreography diagrams be executed automatically?



Building: own representation of RaMicsV.

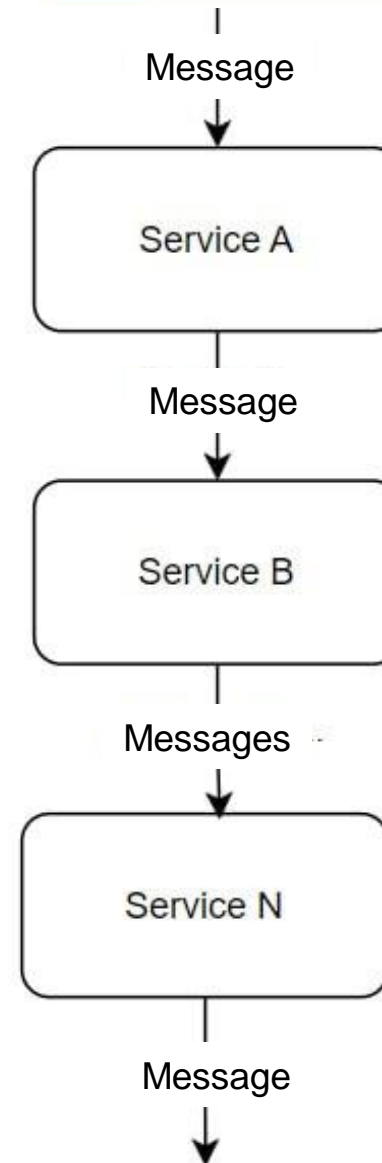
Recap

Definition of choreography

The workflow is **not** monitored, guided or controlled by **any central instance**. The **responsibility** for executing the workflow lies with the **participating services**.

Participating services in a choreography:

- Respond to incoming messages,
- Processing messages,
- If necessary, publish the result after completion of a (partial) task.



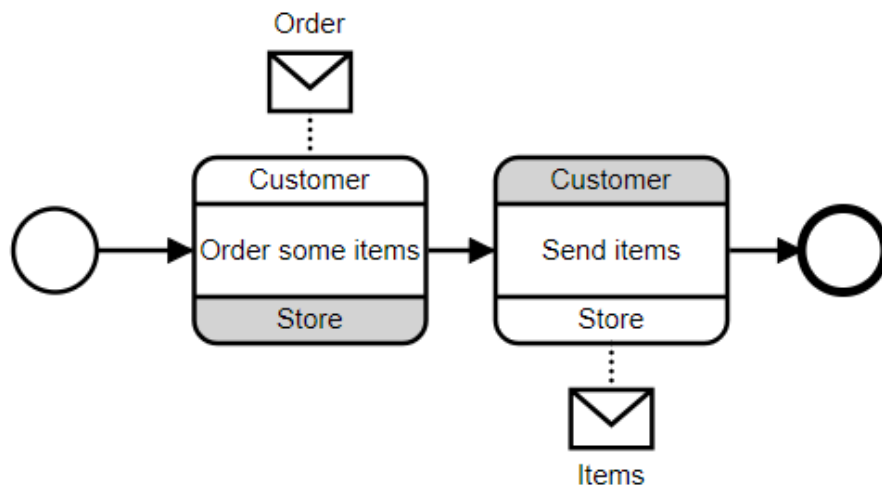
Source: Own representation of choreography.

Recap

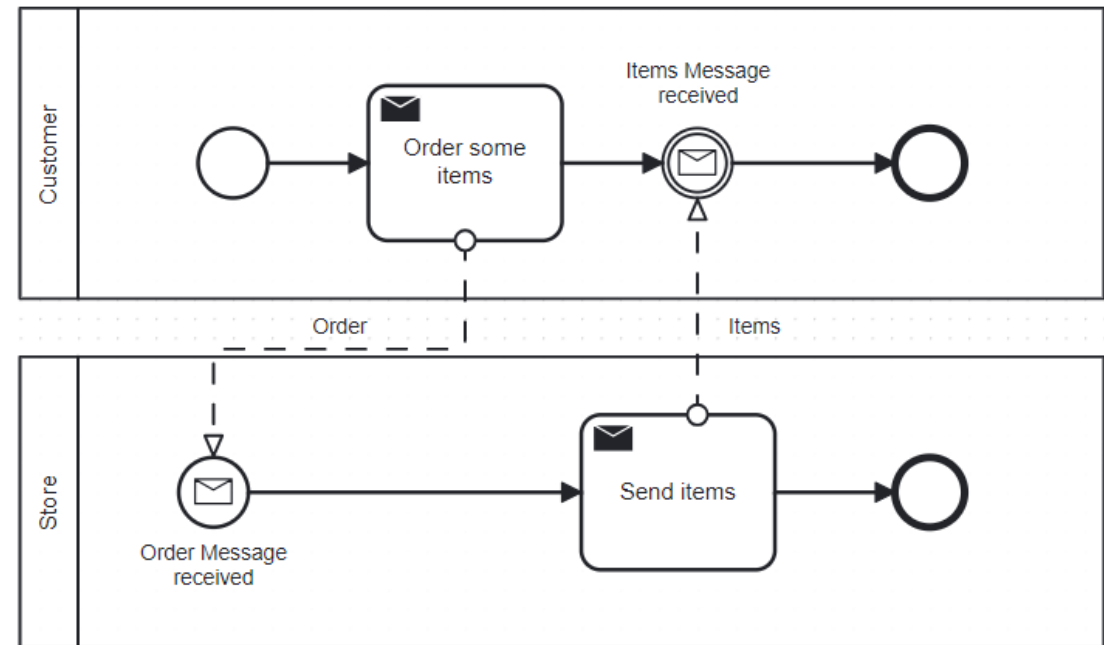
Initial situation: BPMN



BPMN Process



BPMN Choreography



BPMN Collaboration

Recap

BPMN 2.0 Choreography diagrams & problem definition

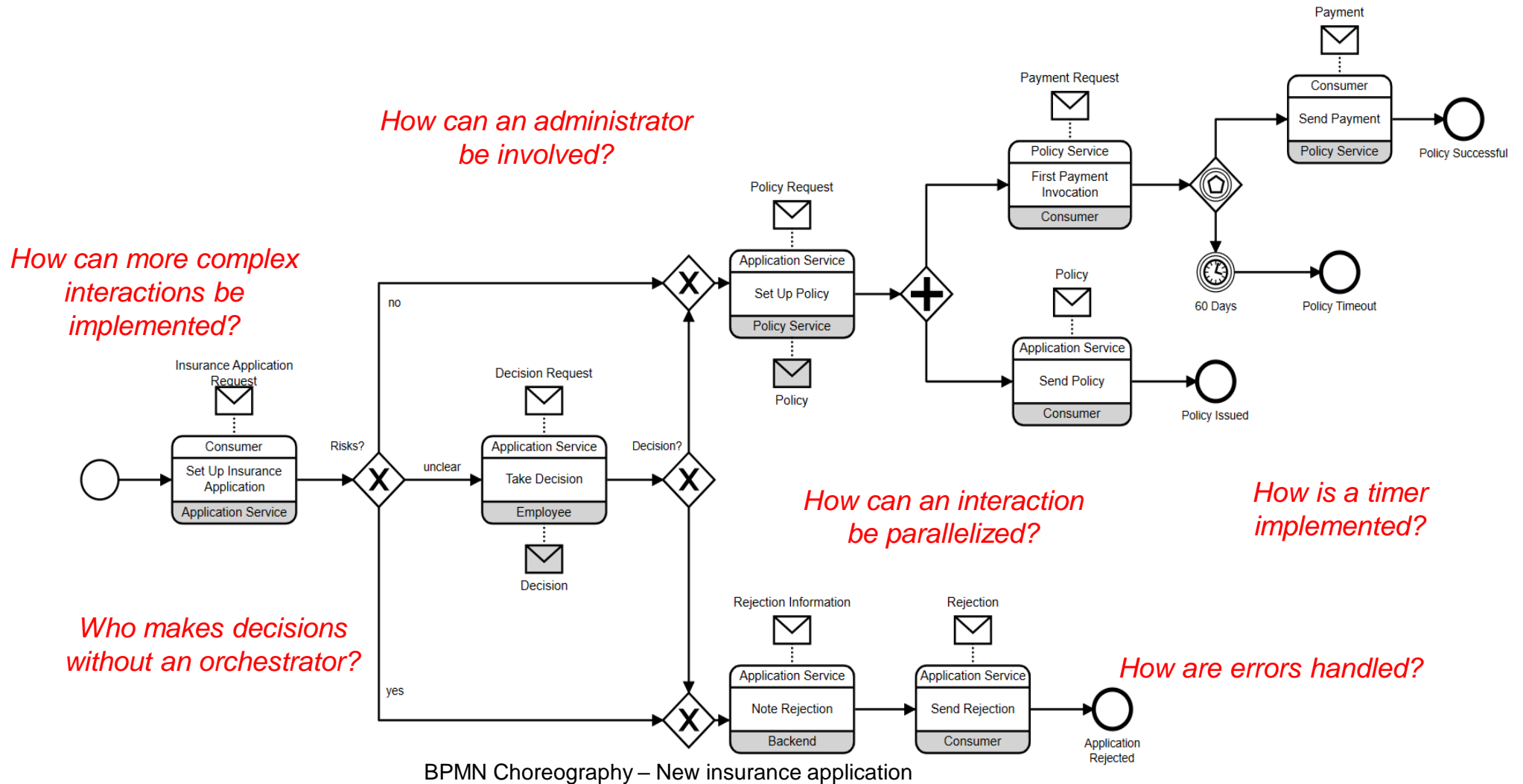


Table of contents

Chapter 1	Presentation
Chapter 2	Recap
Chapter 3	Choreography patterns
Chapter 4	Insight into the choreography grammar
Chapter 5	Outlook and next steps



Choreography patterns

Pattern Language

Idea: Evaluation of frequently occurring patterns in insurance processes in order to define rules and a grammar for the implementation of practically relevant patterns.

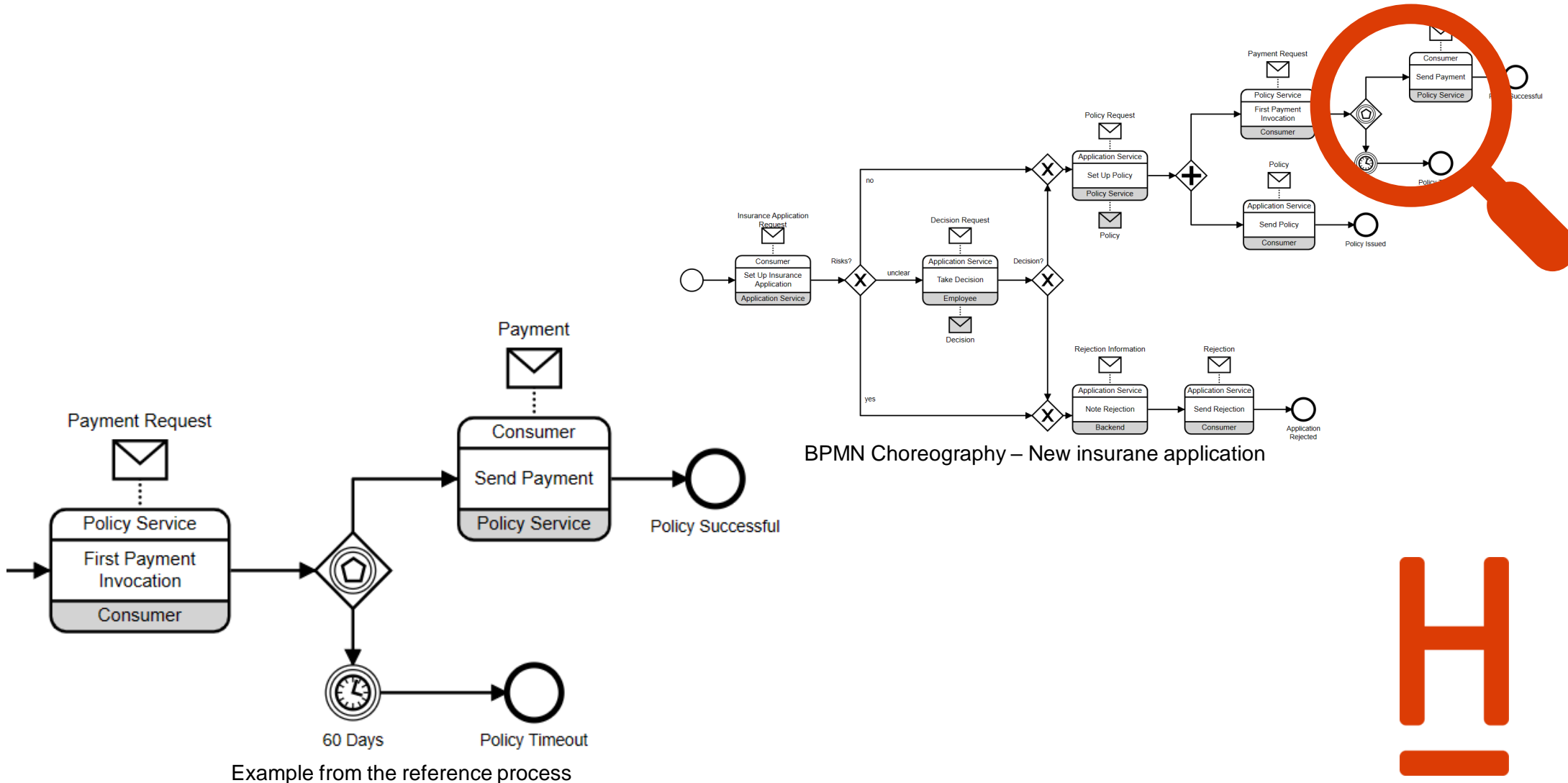
Approach: Pattern Language to define different patterns.

Identification number (ID)	Identification number of the pattern.
Name	Name of the pattern.
Figures	BPMN choreography, BPMN collaboration, and UML sequence diagrams.
Rules	Conditions under which the pattern may be used.
Used BPMN elements	BPMN elements that occur in that pattern.
Used Patterns	Other patterns which this pattern builds upon.
Synonyms	Similar patterns from literature and industry.
Variations	Situational differences for that pattern where the core principal stays the same.
Typical combinations	Other patterns which can be easily combined.
Use-Cases	Practical example where the pattern occurs.



Choreography patterns

Event-based Gateway – Deadline Pattern



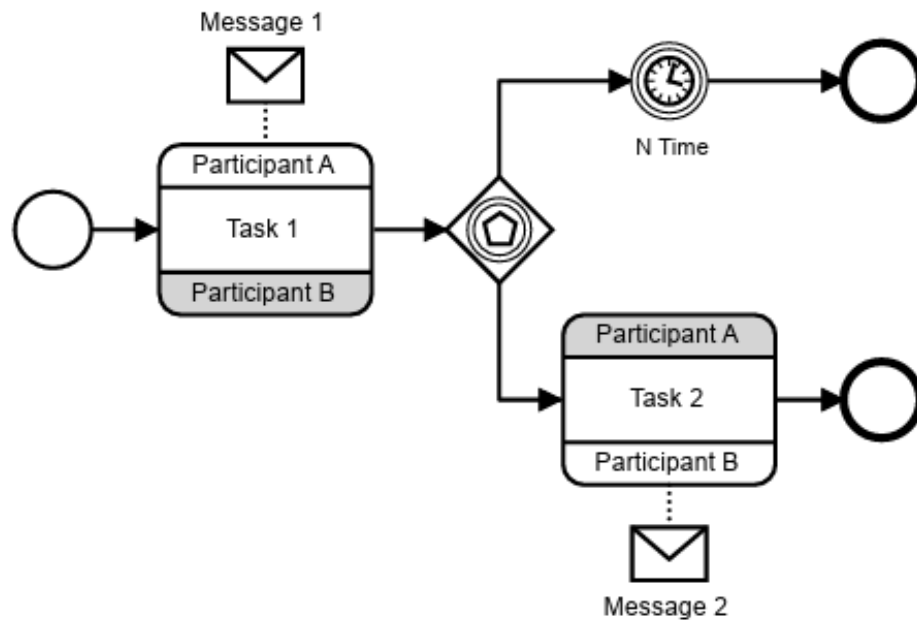
Example from the reference process

Choreography patterns

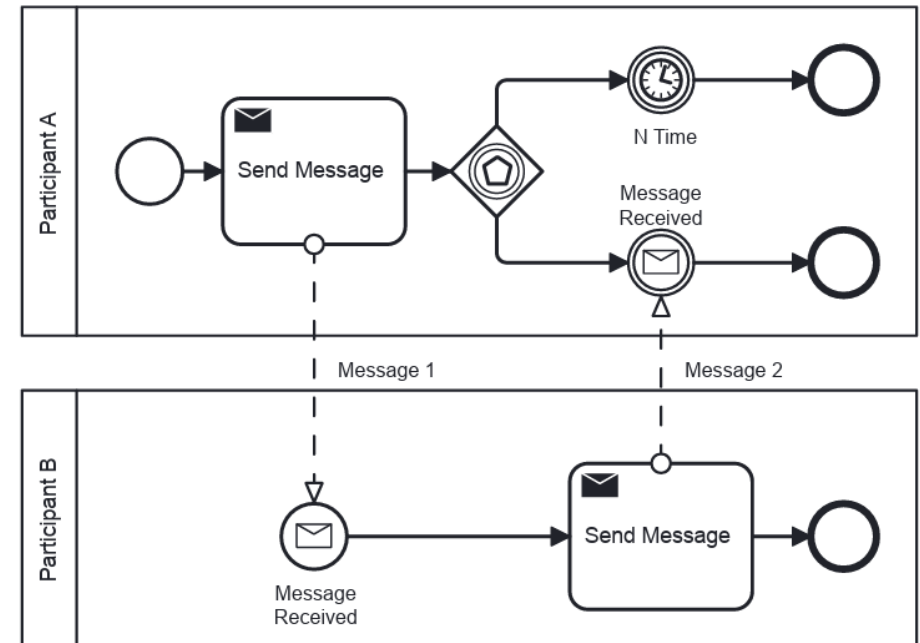
Event-based Gateway – Deadline Pattern

Pattern: Event-based Gateway - Deadline

A time-limited reaction is to be modeled. A participant (B) receives a message from the initiator (A). The recipient (B) must respond within a defined time (N time).



Deadline Choreographie



Deadline Kollaboration

Choreography patterns

Event-based Gateway – Deadline Pattern

ID	BPMNChor11
Name	Event-based Gateway - Deadline
Figures	See previous slide
Rules	The recipient of the first message becomes the new initiator.
Used BPMN elements	startEvent (none), event-based gateway (opening), participant (pool), Messages (send and receive), timerEvent, endEvent (none)
Used patterns	This pattern is based on the <i>Sequence Flow - Two Participants</i> pattern, with the restriction that the participant who has received the message must reply within a certain time.
Synonyms	Asynchronous request-response
Variations	Event-based Gateway - Participant
Typical combinations	With all basic patterns
Use case	Money transfer within a defined period.



Choreography patterns

Outlook

- Evaluation of further patterns in the insurance industry,
- Creation of the pattern catalog,
- Development of the grammar
 (→ Insight to follow shortly 😊)

Pattern Catalog for BPMN Choreography
CC_ITM
 Hochschule Hannover

Christin Schulze, Alexander Link, Arne Koschel & Andreas Hausotter

March 18, 2024
 Version 1.1 WIP VERSION

Contents

1	Introduction	3
2	Basics	4
2.1	General conditions and rules	4
2.2	Reference process	5
2.3	Patterns	5
2.3.1	Pattern Language	5
3	Atomic Patterns	9

Table of contents

Chapter 1	Presentation
Chapter 2	Recap
Chapter 3	Choreography patterns
Chapter 4	Insight into the choreography grammar
Chapter 5	Outlook and next steps



Insight into the choreography grammar

Introduction

Idea: Choreography grammar to implement valid combinations of choreography patterns and exclude non-valid combinations to represent choreographies.

- **Grammar words:** Choreography to represent coherent interaction.
- **Letters:** Choreography patterns and BPMN elements (flow nodes).
- **Currently:** Context-free language, since all relations in R are of the form $N \times (N \cup T)^*$
- **Properties & restrictions:**
 - Language L(G) contains valid choreographies according to pattern catalog,
 - Validation of sequences of patterns possible,
 - No claim to completeness → Guideline for modeling,
 - No checking of initiators and receivers via the grammar → Check rules of the individual patterns.



Insight into the choreography grammar

Formal

Grammar (G) = (non-terminals (N), terminals (T), derivation rules (R), start (S)) with:

- $N = \{C, ANY, N \text{ according to the status of the pattern catalog, capitalized identifier}\}$
- $T = \{T \text{ according to the status of the pattern catalog, lower case identifier}\}$
- $R = \{\text{see next slide – cutout}\}$
- $S = C$

ANY:

- Placeholder for each non-terminal, except start and end event
- Can also be replaced by epsilon (ϵ) \rightarrow Empty word
- No ANY \Rightarrow ANY ANY derivation, as it is precisely defined for each pattern when which patterns can come before or after it



Insight into the choreography grammar

Rules (for Deadline-Pattern)

R =

{

(1) $C \Rightarrow E_s G_{rd}$ | (further derivation rules)

(2) $ANY \Rightarrow$ (all available M) | T_{ow} | ϵ

(3) $T_{ow} \Rightarrow ANY t_{ow} ANY$

(10) $G_{rd} \Rightarrow ANY (t_{ow} | t_{tw})^+ (g_{rd}[ANY E_e]) g_{rd}[e_{it} ANY E_e]$

(12) $E_s \Rightarrow e_s$

(13) $E_e \Rightarrow e_e$

}

with:

$E_s = N$ for Start-Event

$e_s = T$ for Start-Event

$G_{rd} = N$ for Deadline Pattern

$g_{rd} = T$ for Deadline Pattern

$T_{ow} = N$ for One-Way Task

$t_{ow} = T$ for One-Way Task Pattern

$t_{tw} = T$ for Two-Way Task Pattern

$E_e = N$ for End-Event

$e_e = T$ for End-Event

$+ =$ at least once



Insight into the choreography grammar

Example – Deadline Pattern

- **Note:** Adjustment to start event for this example

Create grammar:

- (1) $C \Rightarrow E_s \mathbf{G}_{rd}$
- (10) $\Rightarrow E_s \mathbf{ANY} (t_{ow} | t_{tw})^+ (g_{rd}[\mathbf{ANY} E_e]) g_{rd}[e_{it} \mathbf{ANY} E_e]$
- (2) $\Rightarrow E_s \mathbf{ANY} (t_{ow} | t_{tw})^+ (g_{rd}[T_{ow} E_e]) g_{rd}[e_{it} \mathbf{ANY} E_e]$
- (2) $\Rightarrow E_s (t_{ow} | t_{tw})^+ (g_{rd}[T_{ow} E_e]) g_{rd}[e_{it} E_e]$

Select t_{ow} for the first interaction „First Payment Invocation“,

so that: $\Rightarrow E_s t_{ow} (g_{rd}[T_{ow} E_e]) g_{rd}[e_{it} E_e]$

(3) $\Rightarrow E_s t_{ow} (g_{rd}[t_{ow} E_e]) g_{rd}[e_{it} E_e]$

(13) $\Rightarrow E_s t_{ow} (g_{rd}[t_{ow} e_e]) g_{rd}[e_{it} e_e]$

(12) $\Rightarrow e_s t_{ow} (g_{rd}[t_{ow} e_e]) g_{rd}[e_{it} e_e]$

Complete grammar: $C \Rightarrow e_s t_{ow} (g_{rd}[t_{ow} e_e]) g_{rd}[e_{it} e_e]$

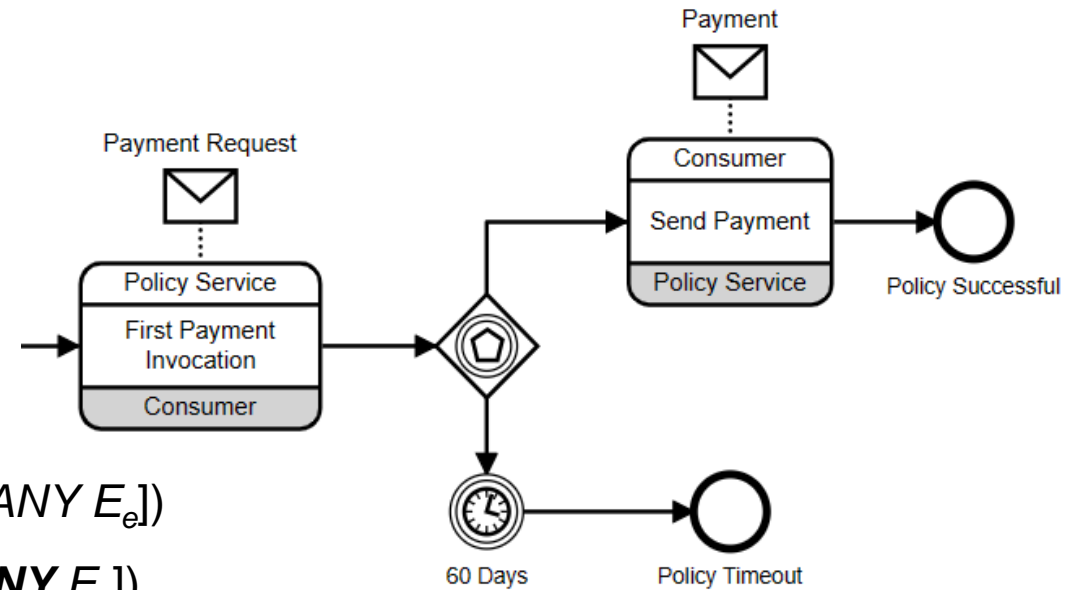


Table of contents

Chapter 1	Presentation
Chapter 2	Recap
Chapter 3	Choreography patterns
Chapter 4	Insight into the choreography grammar
Chapter 5	Outlook and next steps



Outlook and next steps

Long-term goal:

Develop a tool for the creation and automated execution of BPMN 2.0 choreography diagrams for choreographed microservices within the insurance industry.

Steps:

- Serialization of BPMN,
- Extension of the XML schema (BPMN20.xsd for the diagrams and Semantic.xsd for the specification of the BPMN elements),
- parser and generator,
- ...



Current publication

Towards Patterns for Choreography of Microservices-based Insurance Processes:

- https://www.thinkmind.org/index.php?view=article&articleid=service_computation_2023_1_10_10003



Thank you for your attention!

Any questions?

Contact:

Christin Schulze: christin.schulze@stud.hs-hannover.de

Alexander Link: alexander.link@stud.hs-hannover.de

Andreas Hausotter: andreas.hausotter@hs-hannover.de

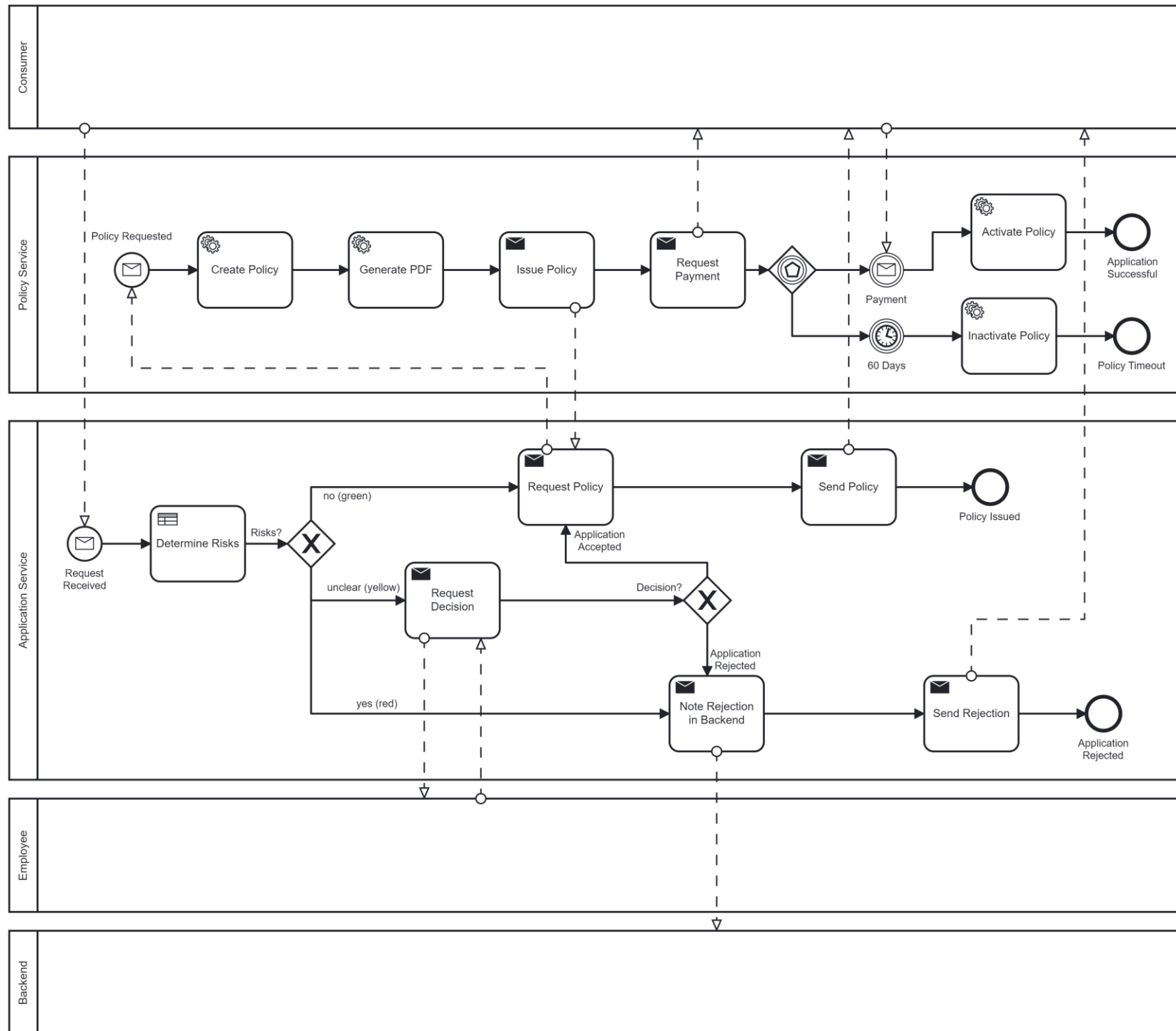
Arne Koschel: arne.koschel@hs-hannover.de



Appendix – Backup Folien



Backup: Reference- process as Collabo- ration



Backup: Initial situation – BPMN

Use of the OMG's BPMN 2.0 specification:

- Acronym for Business Process Model and Notation,
- Graphical specification language for describing business processes and workflows,
- Diagram types: BPMN Process, BPMN Collaboration, BPMN Choreography
- BPMN suites such as Camunda can automatically execute BPMN and BPMN collaboration diagrams. → (So far) no implementation for BPMN choreography diagrams!

Widespread use in the business and insurance industry!

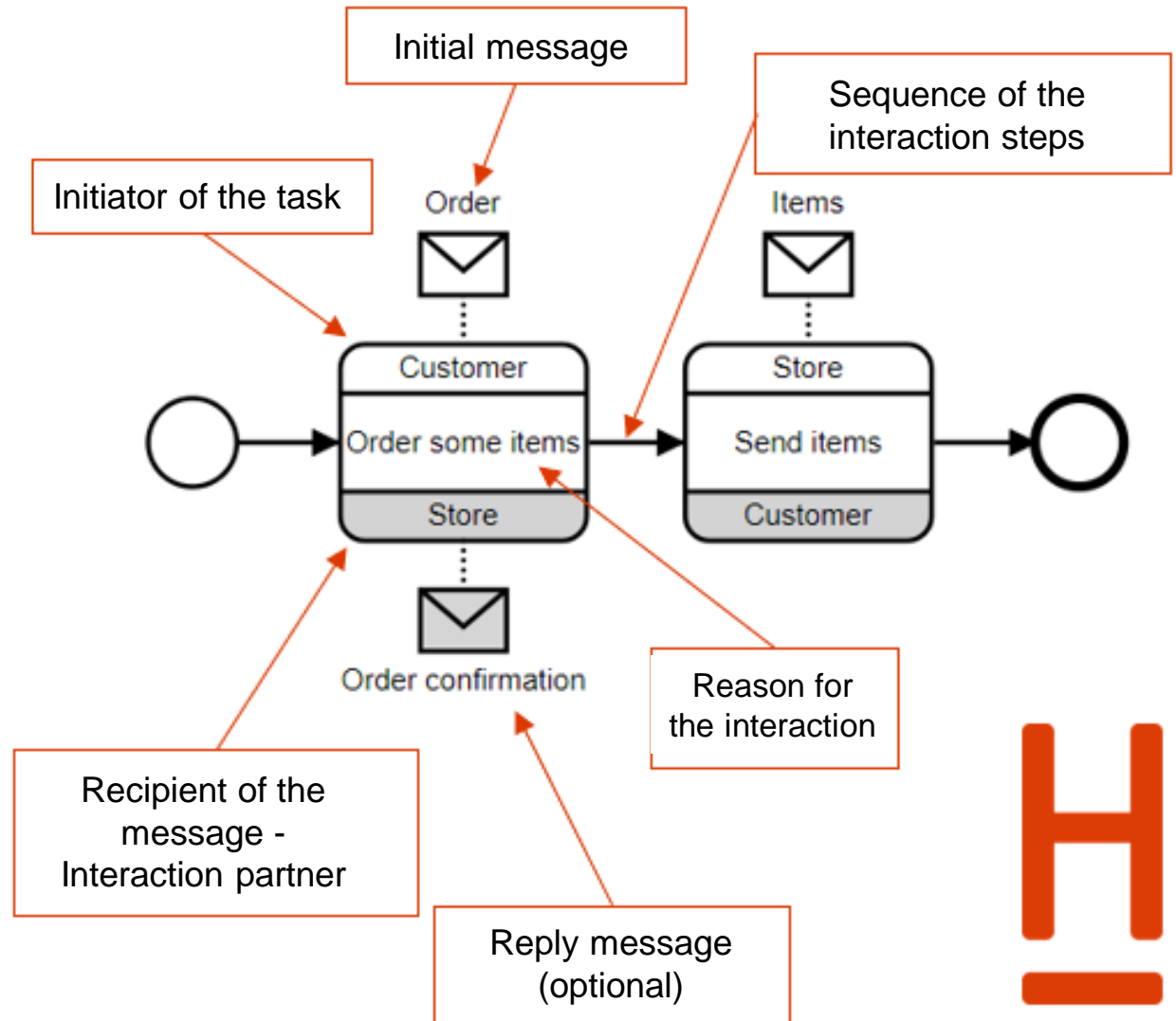


Source Spezifikation: <https://www.omg.org/spec/BPMN/2.0/PDF>

Backup: BPMN 2.0 choreography

BPMN 2.0 Choreography
 OMG specification:

- Comply with and extend the rules of the specification,
- Iteratively add elements for modeling an interaction,
- Evaluation of which elements or patterns frequently occur in the context of insurance.



Source Spezifikation: <https://www.omg.org/spec/BPMN/>

Backup: Grammar - notations

Notation for repetitions:

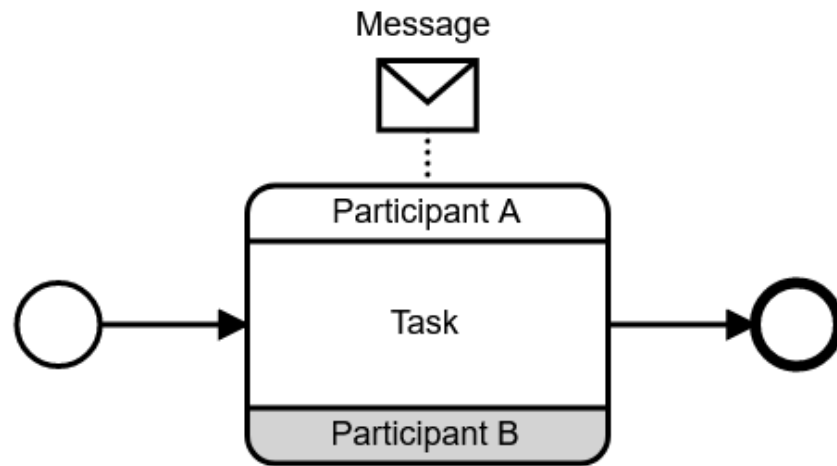
- $*$ = Any number of times, or 0 times
- $+$ = Any number of times, at least 1 time
- $x+$ = Any number of times, at least x times
- $?$ = 0 or 1 time \rightarrow For clarity when using gc

Notation for brackets:

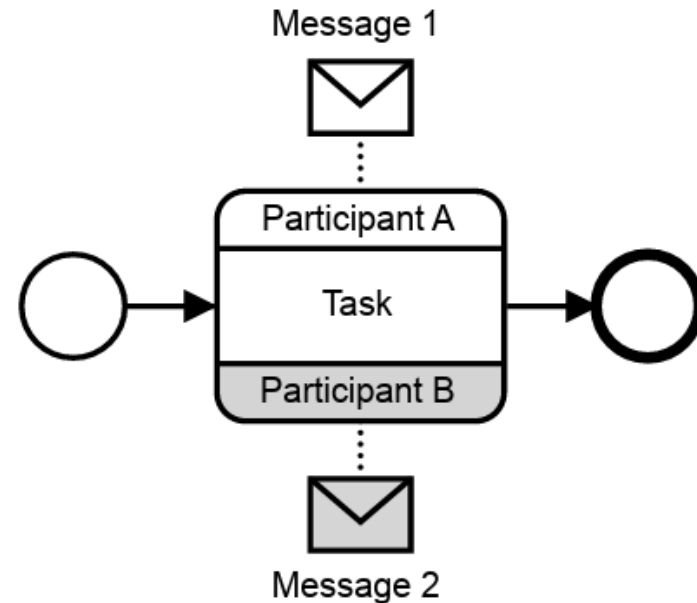
- Round brackets: Decision
- Curly brackets: Paths
- Square brackets: Event-based paths



Backup: Grammar $\rightarrow T_{ow}$ and T_{tw}



One-Way Task Choreography



Two-Way Task Choreography

