HOCHSCHULE
HANNOVER
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS
–
Fakultät IV
Wirtschaft und
Informatik

CC_ITM Competence Center
Information Technology and Management
Institute at the University of Applied Sciences
and Arts Hannover | cc_itm@hs-hannover.de

Service Computation 2023
June 26 – 30, 2023 – Nice, Saint-Laurent-du-Var, France

Towards Patterns for Choreography of Microservices-based Insurance Processes

A. Koschel, A. Hausotter, A. Link, H. Meyer, C. Schulze

Faculty of Business and Computer Science
University of Applied Sciences and Arts, Hanover
Ricklinger Stadtweg 120 30459 Hanover
{arne.koschel | andreas.hausotter}@hs-hannover.de

Presenter

Dr. ANDREAS HAUSOTTER is a professor emeritus for distributed information systems and database systems at the University for Applied Sciences and Arts, Hanover, Germany, Faculty of Business and Computer Science. His area of specialization comprises service computing – including service-oriented Architectures (SOA) and microservices – Java EE, webservices, distributed information systems, business process management, business rules management, and information modeling.

In 1979 he received his PhD in mathematics at Kiel University, Faculty of Mathematics and Natural Sciences. After graduation he started his career with KRUPP ATLAS ELEKTRONIK, Bremen, as a systems analyst and systems programmer in the area of real time processing. In 1984 he was hired as systems engineer and group manager SNA Communications for NIXDORF COMPUTER, Paderborn. After that, he worked for HAAS CONSULT, Hanover, as a systems engineer and product manager for traffic guidance systems.

In 1996 he was appointed professor of operating systems, networking and database systems at the University of Applied Sciences and Arts, Hanover. He has been retired since March 2018.

From the beginning he was involved in several research projects in cooperation with industry partners. During his research semester he developed a Java EE / EJB application framework. Based on this framework a web-based simulation software for securities trading was implemented by his research group to train the apprentices of the industry partner.

In 2005, the Competence Center IT & Management (CC_ITM) was founded in cooperation with industry partners. Different ambitious research projects have since then been carried out in the context of service-computing, microservices, cloud computing, business process management, and business rules management.

Agenda

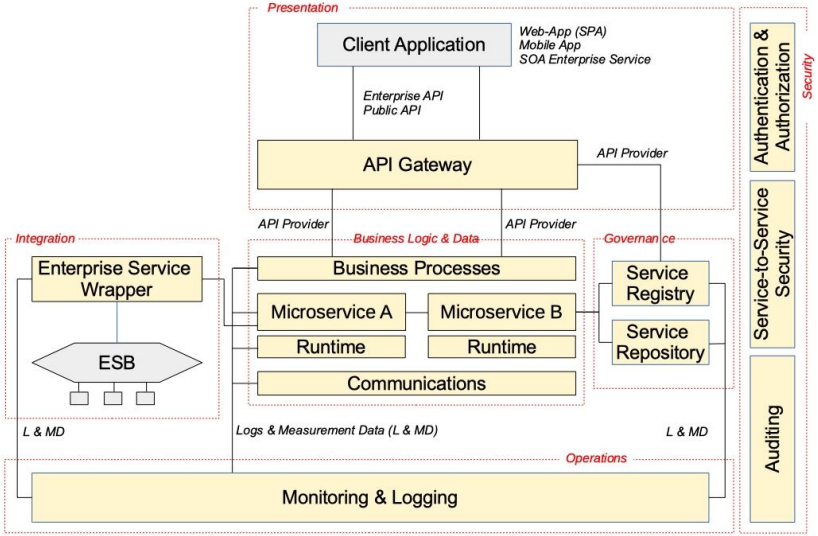
1. Introduction
2. Choreography
3. Choreography Patterns
4. Conclusion and Future Work

CC_ITM@HsH

- **Competence Center Information Technology & Management (CC_ITM)**
  - Institute at the University of Applied Sciences and Arts, Hanover;
  - Founded in 2005 by colleagues from the departments of **Business Information Systems and Computer Science;**
  - Members: **Faculty staff and industry partners** (practitioners) of different areas of businesses.
- Main objective
  - **Knowledge transfer** between university and industry.
- Research topics
  - Management of information processing;
  - Service computing, including Microservices, Service-oriented Architectures (SOA), Business Process / Rules Management (BPM/BRM);
  - Cloud Computing.

Reference Architecture for Microservices

- **Coexistence**: Legacy applications, SOA and MSA based applications will be operated in parallel for a longer transition period.
- **Business processes** are critical elements in an insurance company's applications landscape.
- To keep their competitive edge, the enterprise must **change their processes** in a flexible and agile manner.
- For more information, see our past paper [7].

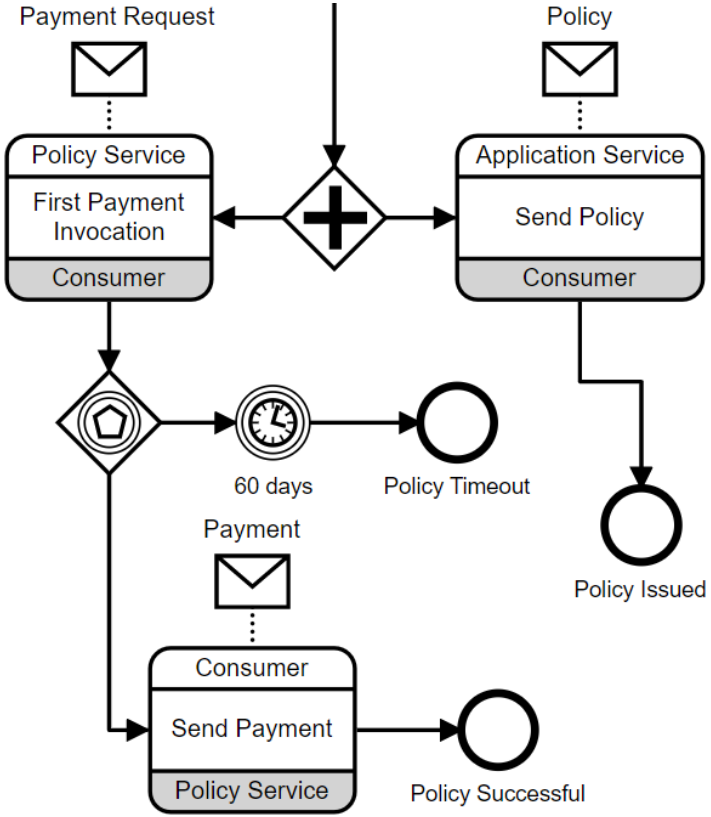


Building Blocks of the Logical Reference Architecture RaMicsV [own representation].

Application Scenario

- Customer requests a policy.
- The policy is sent to the customer.
- Additionally, a payment request is sent to the customer.
- If the customer pays within 60 days, everything is well.
- If the no payment is received, policy timeout occurs.

- How can we implement this using choreographed microservices?



Example Application Scenario

Questions to be answered

- Which Business processes / workflows can/do utilize a choreography-based approach?
- What common scenarios occur during diverse use cases?
- How can we define these occurrences? What patterns can be made from them?
- How would a grammar for these patterns work?

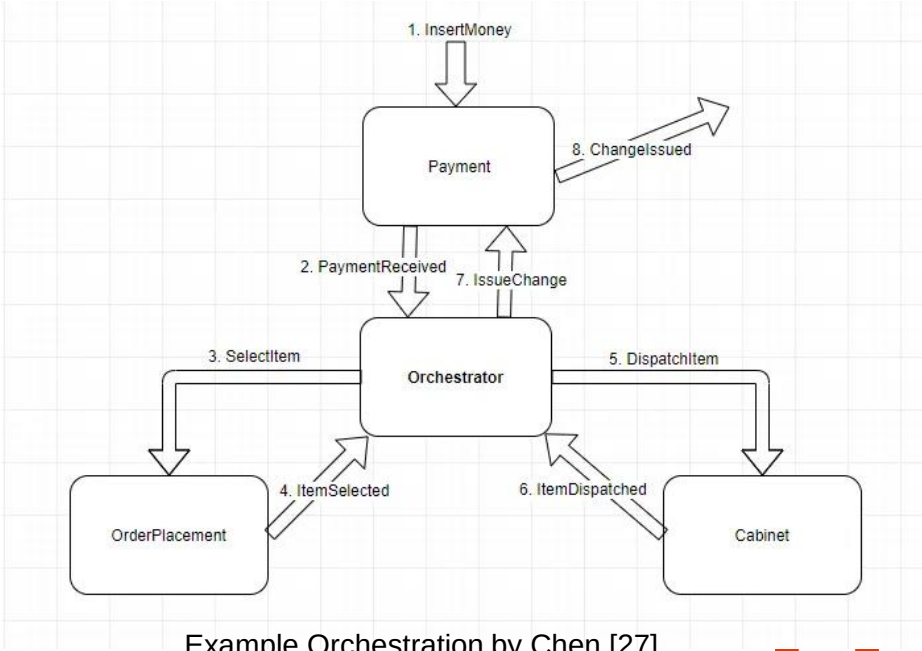
Agenda

1. Introduction
2. Choreography
3. Choreography Patterns
4. Conclusion and Future Work

Choreography vs. Orchestration

Orchestration:

- Central coordinator / orchestrator needed,
- Explicit modeling of a workflow,
- Mapped by the orchestrator,
- Responsibility lies within the orchestrator,
- Decisions are made the orchestrator.

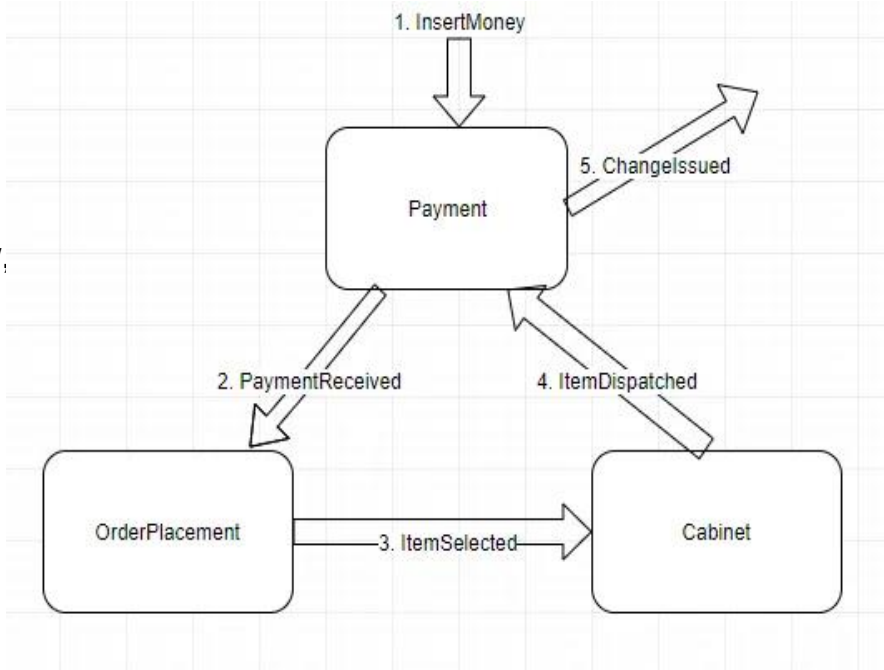


Example Orchestration by Chen [27]

Choreography vs. Orchestration

Choreography:

- No central coordinator / orchestrator,
- No explicit modeling or monitoring of a workflow,
- Mapped by the sequence of actions,
- Responsibility lie within the services,
- Decisions are made by the participants.



Example Choreography by Chen [27]

BPMN

- **B**usiness **P**rocess **M**odel and **N**otation
- **BPMN 2.0 collaboration** demonstrates the **workflow,** using Participants (in our case, synonymous to one or more microservices of the same Bounded Context)**;**
- **BPMN 2.0 choreography** focusses on **the order of messages and alternatives between interactions;**
- **No clear realization and implementation rules from the OMG BPMN standard**, to map **BPMN 2.0 choreography diagram to choreographed microservices** yet.

Problems with Implementing a Choreography

- The automated transfer from choreography diagrams to choreographed micrservices is not researched.
- Engines like **Camunda** only offer automated orchestration deployment, using BPMN 2.0 collaboration diagrams.
- **Requirement** of our project partners: **Construct a choreographed microservices from a given choreography diagram.**
  - **Orchestration is not an option.**

Agenda

1. Introduction
2. Choreography
3. Choreography Patterns
4. Conclusion and Future Work

Why Patterns?

- Choreographies can be hard to maintain and monitor, because the workflow is spread across the participants.
- Patterns allow for **better understanding of structure**.
- They also guarantee a **working** choreography of microservices.
- Finding every possible pattern is **not possible**, therefore we orient ourselves on **real world workflows.**

Pattern Language

- A full language is currently work in process.
- Currently only for uniform description of patterns.
- Grammar still needs to be developed.
- In this presentation: an atomic- and an event based-pattern.
- More Patterns will be presented in future work.

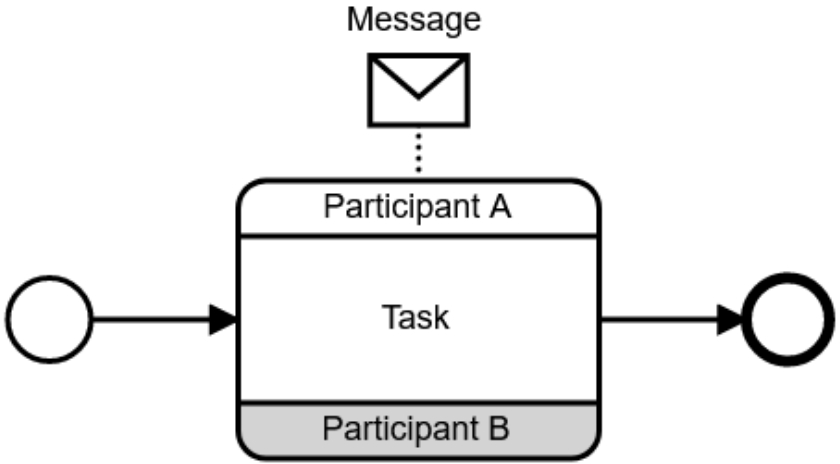
How we describe the patterns

- **ID** of the pattern,
- **Name** of the pattern,
- **Figures** that visualize the pattern,
- a **Description,**
- **Rules** and conditions for the use of the pattern,
- (A list of **Used BPMN Elements,)***
- **Used Patterns,**
- **Synonyms** from literature and industry,
- **Variations,**
- **Typical Combinations,**
- Example **Use Cases**

***** Not presented in detail

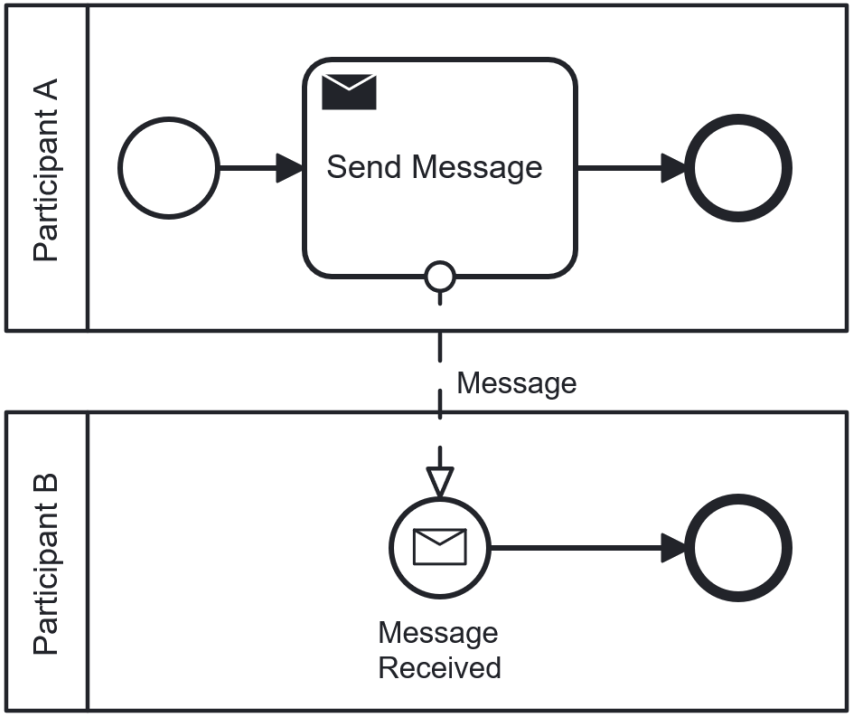
One-Way Task

- **ID:** BPMNChor01
- **Description:** An initiator sends a message to the receiver.
- **Rules:** None
- **Used Patterns:** None
- **Synonyms:** Fire-and-Forget, One-Way Notification.
- **Variations:** None
- **Typical Combinations:** Combinable with every pattern.
- **Use Cases:** E-Mail or SMS

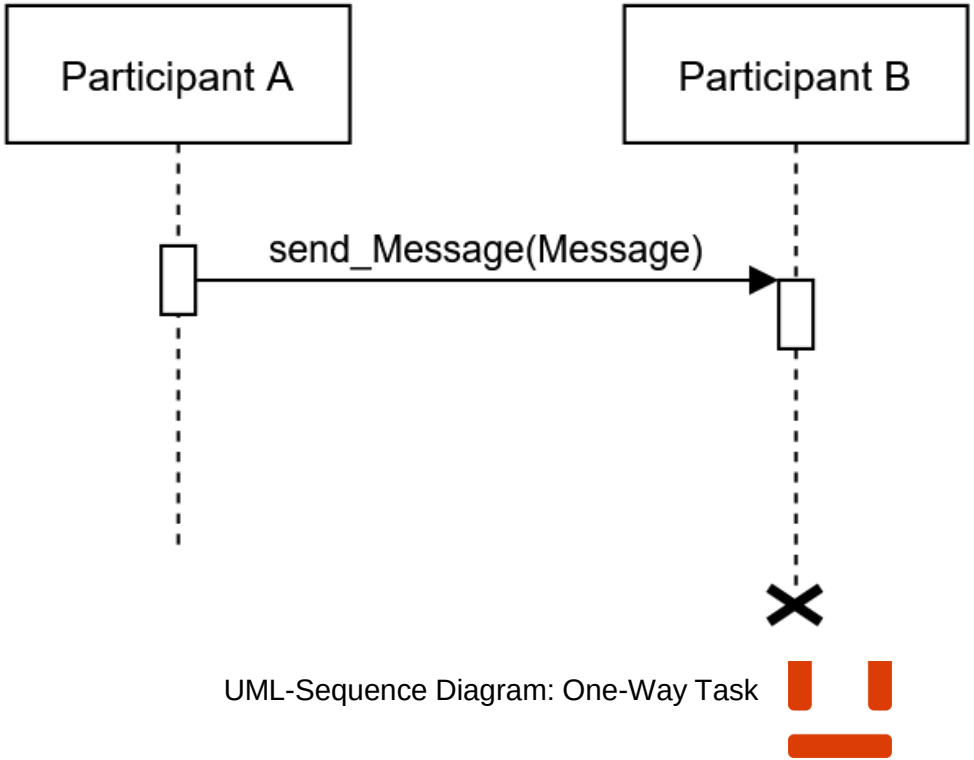


Choreography Diagram: One-Way Task

One-Way Task



Collaboration Diagram: One-Way Task



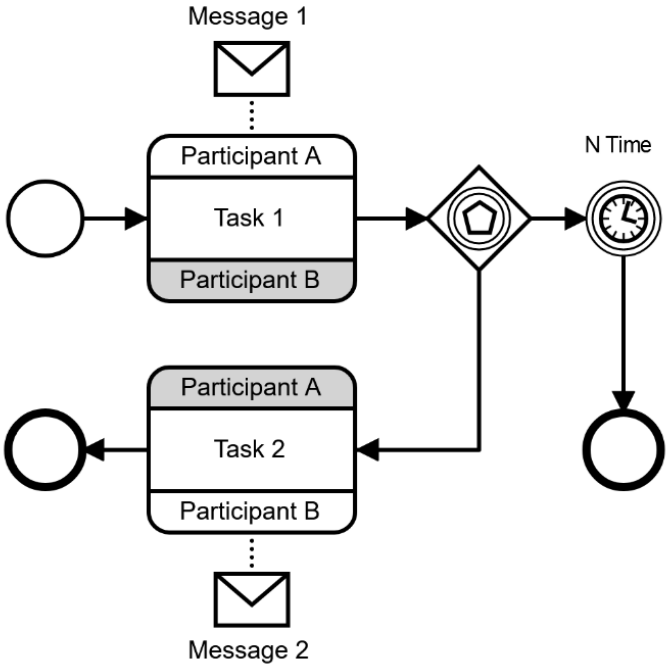
UML-Sequence Diagram: One-Way Task

Event-based Gateway – Deadline

CC_ITM Competence Center Information Technology and Management Institute at the University of Applied Sciences and Arts Hannover | cc_itm@hs-hannover.de

- **ID:** BPMNChor11
- **Description:** The receiver receives a message and only has a certain amount of time to send the reply.
- **Rules:** The receiver must initiate the reply. Two-Way communication is needed

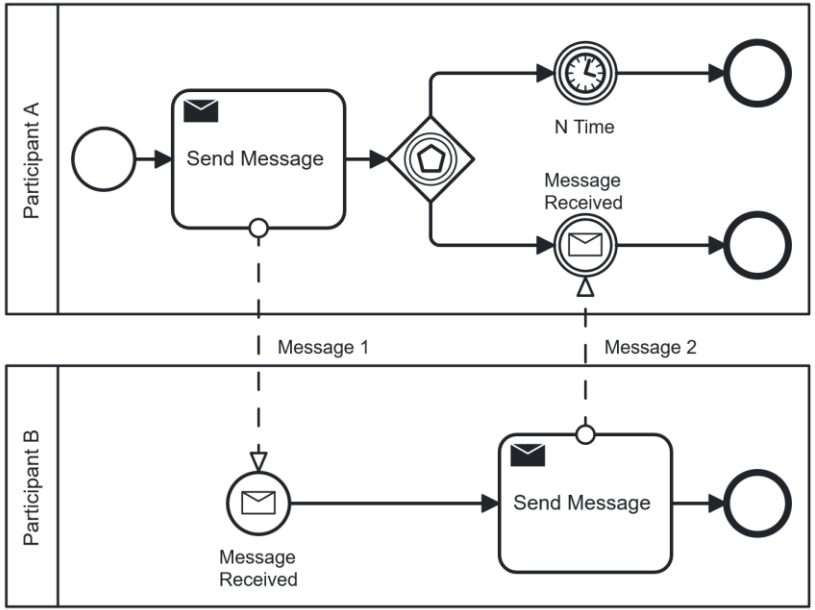
- **Used Patterns:** Sequence Flow – Two Participants (Not in this paper)
- **Synonyms:** Asynchronous Request-Response
- **Variations:** None
- **Typical Combinations:** This pattern may be inserted into any Request-Response workflow.
- **Use Cases:** Deadline for an invoice



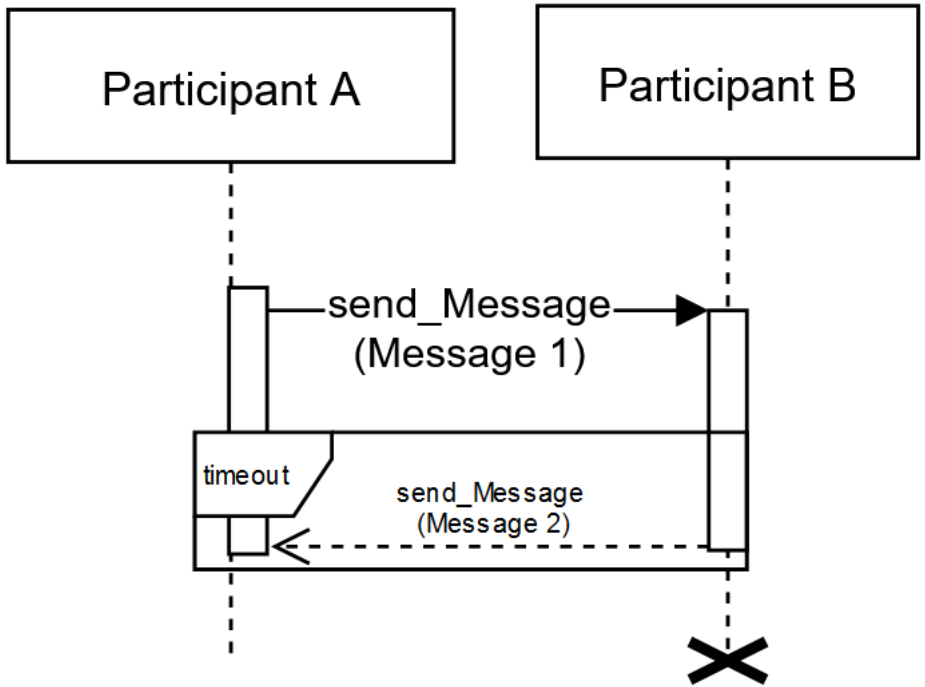
Choreography Diagram: Event-based Gateway - Deadline

Event-based Gateway – Deadline

CC_ITM Competence Center Information Technology and Management Institute at the University of Applied Sciences and Arts Hannover | cc_itm@hs-hannover.de



Collaboration Diagram: Event-based Gateway - Deadline



UML-Sequence Diagram:
Event-based Gateway - Deadline

Agenda

1. Introduction
2. Choreography
3. Choreography Patterns
4. Conclusion and Future Work

Conclusion and Future Work

- Insurance companies need a way to map BPMN Workflows to choreographed microservices.
- We showed two patterns of a pattern language as a first step in this direction.
- A grammar will be developed.
- More patterns are to follow.

Thank you for your attention!

References

[1] Gesamtverband der Deutschen Versicherungswirtschaft e.V. – General Association o.t. German Insurance Industry, “VAA Final Edition. Das Fachliche Komponentenmodell (VAA Final Edition. The Functional Component Model),” 2001.

[2] European GDPR, “Complete guide to GDPR compliance,” Online. Available: https://gdpr.eu/ [retrieved: 04, 2023].

[3] Bundesanstalt f¨ur Finanzdienstleistungsaufsicht (BaFin) – Federal Financial Supervisory (BaFin), Versicherungsaufsichtliche Anforderungen an die IT (VAIT) (Insurance Supervisory Requirements for IT (VAIT)) vom 03.03.2023,” 2023, Online. Available: https://www.bafin.de/SharedDocs/Veroeffentlichungen/DE/Meldung/2023/meldung 2023 03 03 Aktualisierung VAIT.html [retrieved: 04,2023].

[4] C. Richardson, Microservices Patterns: With examples in Java. Shelter Island, New York: Manning Publications, 2018.

[5] S. Newman, Building microservices: designing fine-grained systems. Sebastopol, California: O’Reilly Media, Inc., 2015.

[6] C. Fehling, F. Leymann, R. Retter, W. Schupeck, and P. Arbitter, Cloud Computing Patterns Fundamentals to Design, Build, and Manage Cloud Applications. Springer Vienna, 2014.

References

[7] A. Koschel, A. Hausotter, R. Buchta, A. Grunewald, M. Lange, and P. Niemann, “Towards a Microservice Reference Architecture for Insurance Companies,” in SERVICE COMPUTATION 2021, 13th Intl. Conf. on Advanced Service Computing. IARIA, ThinkMind, 2021, pp. 5–9, Online. Available: https://www.thinkmind.org/articles/service computation 2021 1 20 10002.pdf [retrieved: 04, 2023].

[8] A. Hausotter, A. Koschel, M. Zuch, J. Busch, and J. Seewald, “Components for a SOA with ESB, BPM, and BRM – Decision Framework and architectural Details,” Intl. Journal od Advances in Intelligent Systems, vol. 9, no. 3 & 4, pp. 287–297, 2016.

[9] “Workflow and decision automation platform,” Nov 2021, Online. Available: https://camunda.com/ [retrieved: 04, 2023].

[10] A. Koschel, A. Hausotter, R. Buchta, C. Schulze, P. Niemann, and C. Rust, “Towards the Implementation of Workflows in a Microservices Architecture for Insurance Companies – The Coexistence of Orchestration and Choreography,” in SERVICE COMPUTATION 2023, 14th Intl. Conf. on Advanced Service Computing. IARIA, ThinkMind, 2023, pp. 1–5, Online. Available: https://www.thinkmind.org/index.php?view=article&articleid=service computation 2023 1 10 10002 [retrieved: 04, 2023].

[11] OMG, Business Process Model and Notation (BPMN), Version 2.0, Object Management Group Std., Rev. 2.0, January 2011, Online. Available: http://www.omg.org/spec/BPMN/2.0 [retrieved: 04, 2023].
B. Ruecker, Practical Process Automation - Orchestration and Integration in Microservices and Cloud Native Architectures. O’Reilly, 2021.

References

[12] M. Fowler and J. Lewis, “Microservices a definition of this new architectural term,” 2014, Online. Available: https://martinfowler.com/articles/microservices.html [retrieved: 04, 2023].

[13] L. Krause, Microservices: Patterns and Applications: Designing finegrained services by applying patterns. Lucas Krause, 2015.

[14] T. Allweyer, Kollaborationen,Choreographien und Konversationen in BPMN 2.0 - Erweiterte Konzepte zur Modellierung übergreifender Geschäftsprozesse - Collaborations,Choreographies and Conversations in BPMN 2.0 - Advanced Concepts for Modeling Comprehensive Business Processes. Fachhochschule Kaiserslautern, 2009.

[15] T. Allweyer, Gesch¨aftsprozessmanagement: Strategie, Entwurf, Implementierung, Controlling. - Business process management: strategy, design, implementation, controlling. W3l GmbH, 2005.

[16] B. Rücker and J. Freund, Praxishandbuch BPMN 2.0 - Practice Handbook BPMN 2.0. Carl Hanser Verlag München Wien, 2014.

[17] G. Decker, O. Kopp, and A. Barros, An Introduction to Service Choreographies, vol. 50, no 2 ed. Information Technology, 2008.

[18] B. Rücker, “The Microservices Workflow Automation Cheat Sheet,” 2018, Online. Available: https://blog.bernd-ruecker.com/the-microservice-workflow-automation-cheat-sheet-fc0a80dc25aa[retrieved: 04, 2023].

References

[19] D. Mikalkinas, Situation-aware Modelling and Execution of Choreography. Stuttgart University, 2015.

[20] M. Milanovi´c and D. Gasevi´c, “Modeling service choreographies with rule-enhanced business processes,” in 2010 14th IEEE International Enterprise Distributed Object Computing Conference, 2010, pp. 194–203.

[21] J. Ortiz, V. Torres, and P. Valderas, “A catalogue of adaptation rules to support local changes in microservice compositions implemented as choreographies of bpmn fragments,” 2023, Online. available:https://riunet.upv.es/bitstream/handle/10251/181551/CatalogueOfAdaptationRules.pdf?sequence=1 [retrieved: 05, 2023].

[22] C. Richardson, “Eventuate Framework,” 2021, Online. Available: https://eventuate.io/[retrieved: 04, 2023].

[23] ——, “Eventuate Tram,” 2021, Online. Available: https://eventuate.io/abouteventuatetram.html[retrieved: 04, 2023].

[24] ——, “Eventuate Local,” 2023, Online. Available: https://github.com/eventuate-local/eventuate-local[retrieved: 04, 2023].

[25] J. Ladleif, A. von Weltzien, and M. Weske, “chor-js: A modeling framework for bpmn 2.0 choreography diagrams,” 2019.

References

[26] A. Barros, M. Dumas, and H. A.H.M, “Service interaction patterns,” 2005, pp. 302–318, Online. Available: http://www.workflowpatterns.com/documentation/documents/serviceinteraction BPM05.pdf [retrieved: 05, 2023].

[27] C. Chen, “Choreography vs orchestration,” Online. Available: https://medium.com/ingeniouslysimple/choreography-vs-orchestration-a6f21cfaccae [retrieved: 04, 2023].

[28] B. Rücker, “The Microservices Workflow Automation Cheat Sheet,” Online. Available: https://blog.bernd-ruecker.com/the-microservice-workflow-automation-cheat-sheet-fc0a80dc25aa [retrieved: 04, 2023].

[29] ——, Practical Process Automation - Orchestration and Integration in Microservices and Cloud Native Architectures. O’Reilly, 2021.