

# SOA, Microservices & BPMS

2023-01-17



# Agenda

- Business Process Management & BPMN2
- Microservices & Event Driven Architecture
- ... the real world
- BPM @ tech11



# Business Process Management

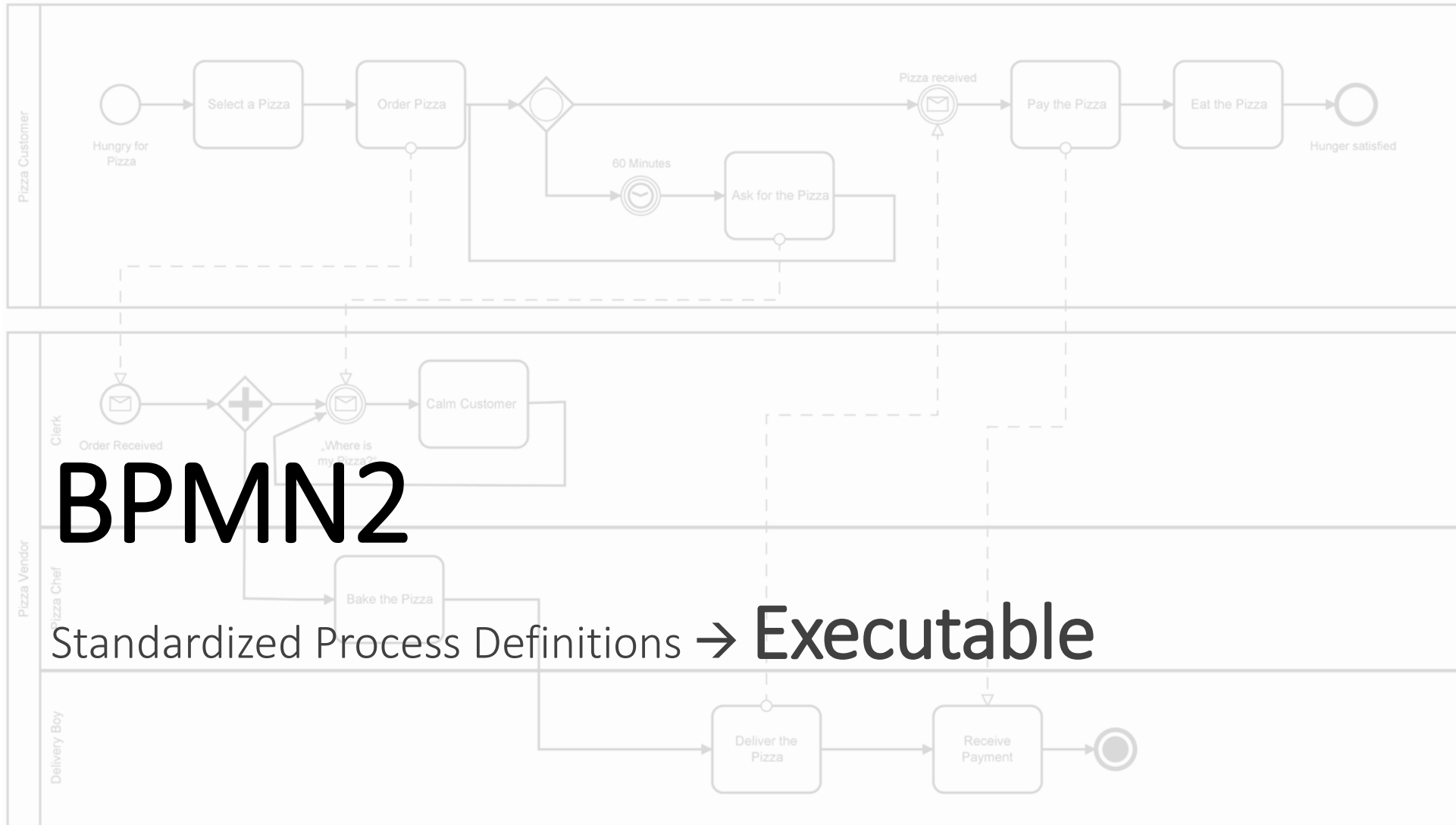
# Process Descriptions

How to describe?

- MS Word – Prosa
- Visio / PowerPoint – Freestyle

Standards:

- EPC – event driven process chain (EPK – Ereignisgesteuerte Prozessketten)
- UML - Unified Modeling Language → Sequence diagrams
- BPMN – Business Process Management and Notation



# BPMN2

Standardized Process Definitions → Executable

# Understanding Business Processes

Simply put, a **Business Process** is a set of tasks that, once completed in a defined order, **accomplishes a defined objective**. Each task in a Business Process has clearly defined inputs and outputs. These tasks may require human intervention or may be completely automated.

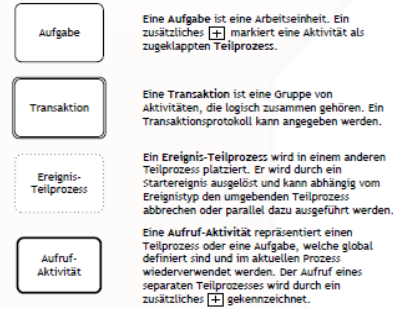
**OMG (Object Management Group)** has defined a standard called [Business Process Model and Notation \(BPMN\)](#) for businesses to define and communicate their processes. BPMN has come to be widely supported and accepted in the industry. The Flowable API fully supports creating and deploying BPMN 2.0 process definitions.

[Source: baeldung → <https://www.baeldung.com/flowable>]

Excursion: **BPM & RPA**

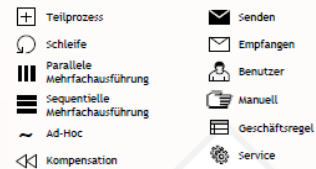
→ <https://weissenberg-group.de/zusammenspiel-von-bpm-und-rpa/>

## Aktivitäten



## Markierungen

Sie beschreiben das Ausführungsverhalten von Aktivitäten:



## Sequenzfluss

definiert die Abfolge der Ausführung.

## Bedingter Fluss

enthält eine Bedingung, die definiert, wann er durchlaufen wird, und wann nicht.

## Standardfluss

wird durchlaufen, wenn alle anderen Bedingungen nicht zutreffen.

## Gateways

## Exklusives Gateway

Bei einer Verzweigung wird der Fluss abhängig von Verzweigungsbedingungen zu genau einer ausgehenden Kante geleitet. Bei einer Zusammenführung wird auf eine der eingehenden Kanten gewartet, um den ausgehenden Fluss zu aktivieren.

## Ereignis-basiertes Gateway

Diesem Gateway folgen stets eintretende Ereignisse oder Empfänger-Aufgaben. Der Sequenzfluss wird zu dem Ereignis geleitet, das zuerst eintritt.

## Paralleles Gateway

Wenn der Sequenzfluss verzweigt wird, werden alle ausgehenden Kanten simultan aktiviert. Bei der Zusammenführung wird auf alle eingehenden Kanten gewartet, bevor der ausgehende Sequenzfluss aktiviert wird (Synchronisation).

## Inklusives Gateway

Es werden je nach Bedingung eine oder mehrere ausgehende Kanten aktiviert bzw. eingehende Kanten synchronisiert.

## Komplexes Gateway

Verzweigungs- und Vereinigungsverhalten, das nicht von anderen Gateways erfasst wird.

**Exklusives Ereignis-basiertes Gateway (Instanziierung)**  
Sobald eines der nachfolgenden Ereignisse eintritt, wird der Prozess gestartet.

**Paralleles Ereignis-basiertes Gateway (Instanziierung)**  
Erst wenn alle nachfolgenden Ereignisse eintreten, wird der Prozess gestartet.

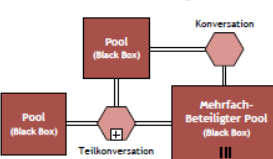
## Konversationen

Eine Konversation definiert einen mehrfachen, logisch zusammengehörigen Nachrichtenaustausch. Ein zusätzliches markiert eine Teilkonversation.

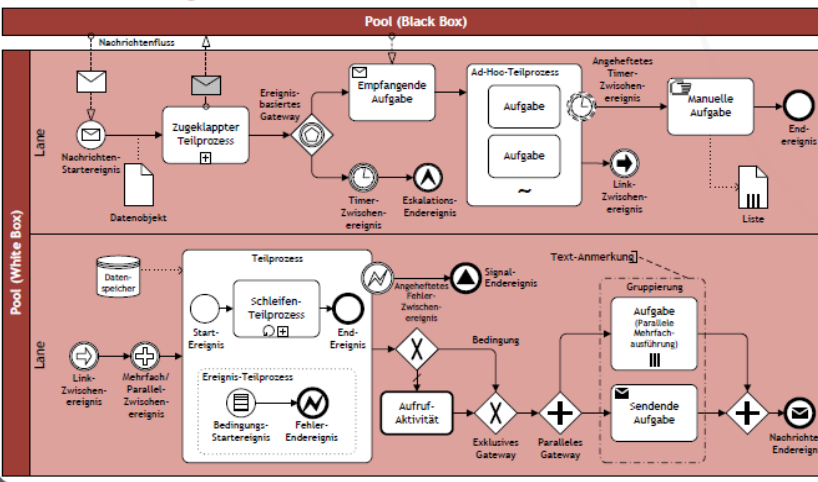
Eine Aufruf-Konversation repräsentiert eine global definierte Konversation oder Teilkonversation. Der Aufruf einer Teilkonversation wird durch ein zusätzliches gekennzeichnet.

Ein Konversationslink verknüpft Kommunikationen und Teilnehmer.

## Konversationsdiagramm

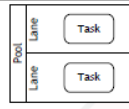


## Kollaborationsdiagramm



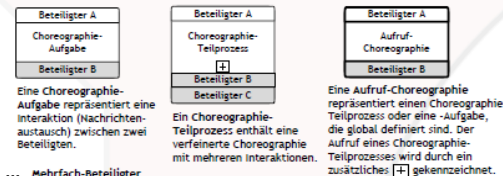
## Swimlanes

**Nachrichtenfluss**  
symbolisiert den Informationsaustausch. Nachrichtenflüsse können an Pools, Aktivitäten und Nachrichtenereignisse andocken. Der Nachrichtenfluss kann mit einem Briefumschlag um den Inhalt der Nachricht angereichert werden.



Pools (Beteiligter) und Lanes repräsentieren Verantwortlichkeiten für Aktivitäten. Ein Pool oder eine Lane können eine Organisation, eine Rolle oder ein System sein.

## Choreographien



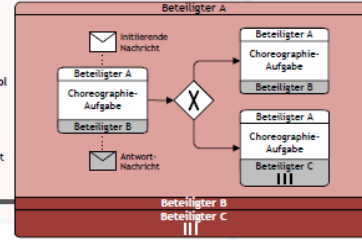
Eine Choreographie-Aufgabe repräsentiert eine Interaktion (Nachrichtenaustausch) zwischen zwei Beteiligten.

Eine Choreographie-Teilprozess enthält eine verfeinerte Choreographie mit mehreren Interaktionen.

**Mehrfach-Beteiligter Markierung**  
beschreibt eine Reihe von Beteiligten derselben Rolle.

**Nachrichtensymbol**  
reichert die Choreographie-Aufgabe um den Inhalt der versendeten Nachricht an.

## Choreographie-Diagramm



## Ereignisse

	Standard	Start	Standard	Start	Zwischen	Ende
	Standard	Start	Standard	Start	Zwischen	Ende
Blanko: Untypisierte Ereignisse, i. d. R. am Start oder Ende eines Prozesses.						
Nachricht: Empfang und Versand von Nachrichten.						
Timer: Periodische zeitliche Ereignisse, Zeitpunkte oder Zeitspannen.						
Eskalation: Meldung an den nächsthöheren Verantwortlichen.						
Bedingung: Reaktion auf veränderte Bedingungen und Bezug auf Geschäftsregeln.						
Link: Zwei zusammengehörige Link-Ereignisse repräsentieren einen Sequenzfluss.						
Fehler: Auslösen und behandeln von definierten Fehlern.						
Abbruch: Reaktion auf abgebrochene Transaktionen oder Auslösen von Abbrüchen.						
Kompensation: Behandeln oder Auslösen einer Kompensation						
Signal: Signal über mehrere Prozesse. Auf ein Signal kann mehrfach reagiert werden.						
Mehrfach: Eintreten eines von mehreren Ereignissen.						
Mehrfach/Parallel: Eintreten aller Ereignisse.						
Terminierung: Löst die sofortige Beendigung des Prozesses aus.						

## Daten



Ein Datenobjekt repräsentiert Informationen, die durch den Prozess fließen, wie z.B. Dokumente, Emails, Briefe oder Datensätze.



Ein Listen-Datenobjekt repräsentiert eine Gruppe von Informationen, wie z.B. eine Liste mit Bestellpositionen.



Ein Dateninput ist ein externer Input für den ganzen Prozess, der von einer Aktivität gelesen wird, und die nach außen sichtbare Eingangs-Prozessschnittstelle.



Ein Datenoutput ist eine Variable, die als Ergebnis eines ganzen Prozesses erzeugt wird, und die nach außen sichtbare Ausgangs-Prozessschnittstelle.



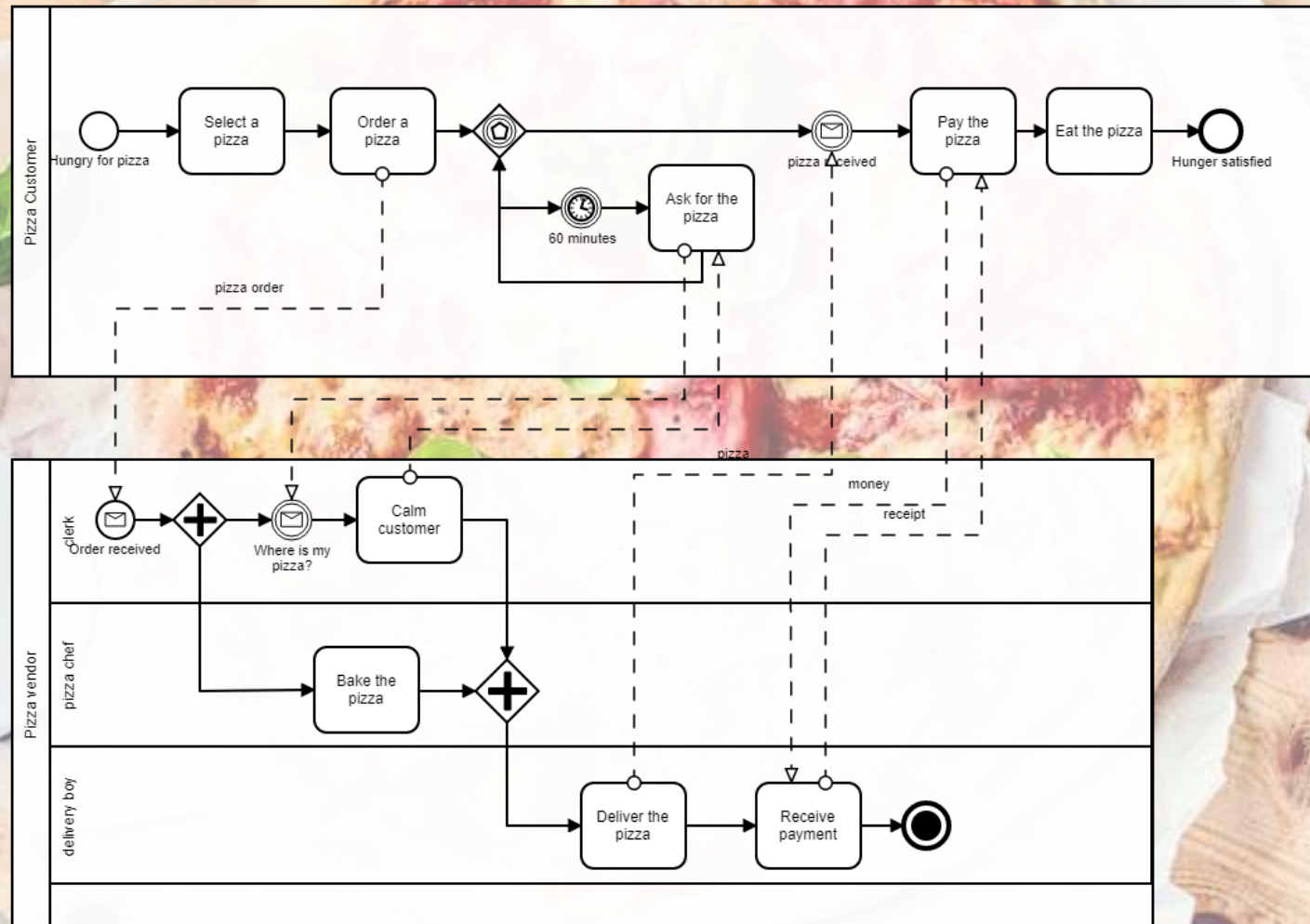
Eine Daten-Assoziation verknüpft Datenobjekte mit Aktivitäten, Prozessen und Aufruf-Aktivitäten.



Ein Datenspeicher ist ein Ort, auf den der Prozess lesend und schreibend zugreifen kann, z.B. eine Datenbank oder ein Altersschrank. Er existiert unabhängig von der Lebensdauer der Prozessinstanz.



# Example: Pizza Ordering Process



[Source: <https://camunda.com/de/bpmn/> ]



# BPM / Process Automation „Types“

- No-Code

- <https://zapier.com/>
- <https://powerautomate.microsoft.com/>
- <https://kissflow.com/low-code/no-code/>
- <https://www.gbtec.com/resources/no-code-and-low-code-automation/>

- Low-Code

- <https://www.pega.com/de/products/platform/case-management>
- <https://kissflow.com/low-code/>
- <https://www.flokzu.com/>

- Developer Friendly

- <https://camunda.com/>
- <https://www.flowable.com/>
- <https://www.activiti.org/>
- <https://www.redhat.com/en/resources/process-automation-overview>

# Demo I

<https://github.com/MatthiasReining/thws2301-bpm>

## Screenplay

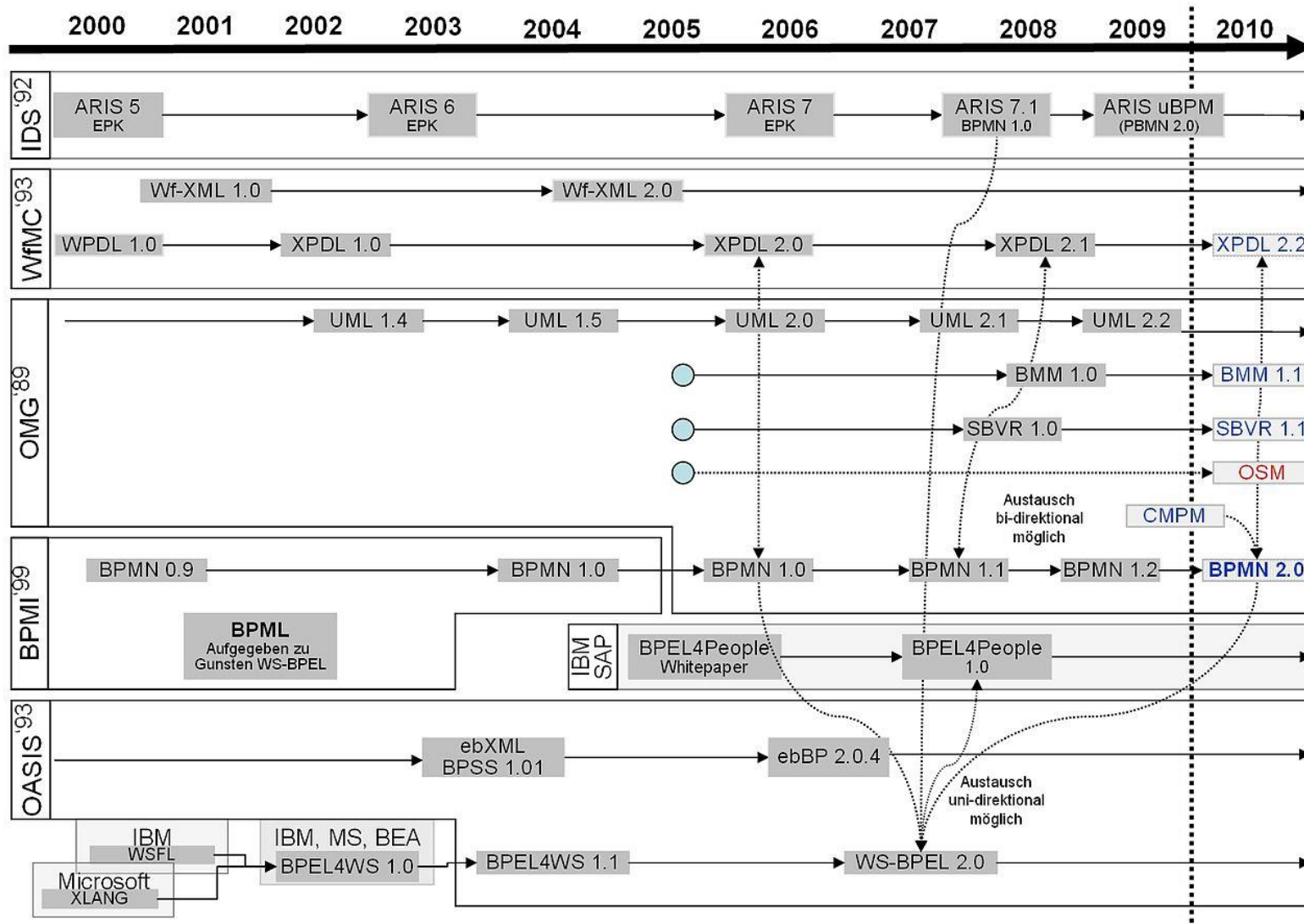
- Setup BPMS (Camunda)
- Camunda Demo Processes
- New THWS Processes
- Script Tasks
- Java Service Tasks
- REST API
- Change Service Task Order
- Add Gateway



[illegible][illegible]

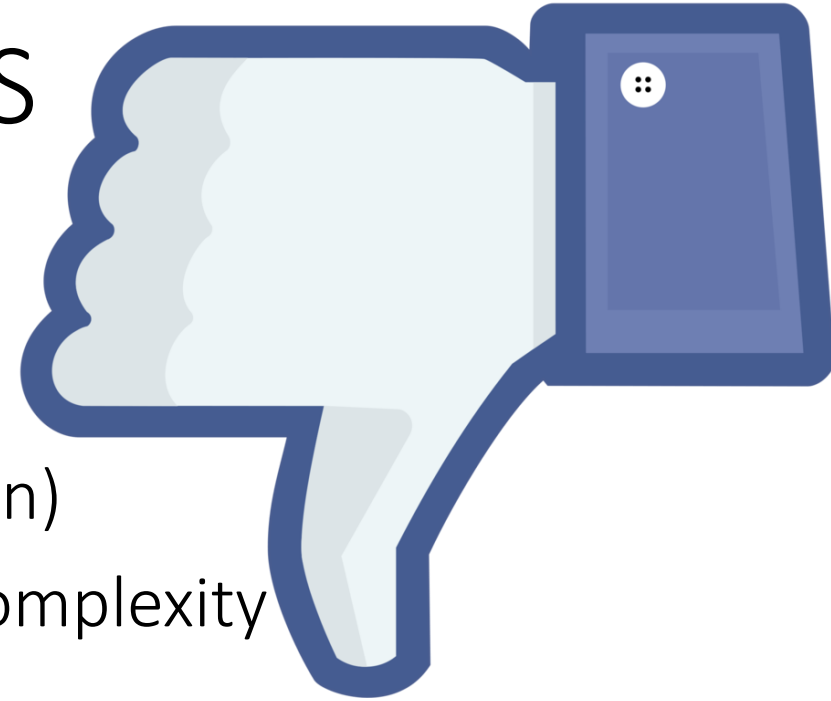


# Process Modelling History...



# Disadvantages when using BPMS

- Centralized business logic
    - Single point of failure
  - Finding the right detail level (in your BPMN design)
  - “Wrong” modelling will lead to unmanageable complexity
- 
- Sounds like “old stuff” (legacy)
  - Often used with ESB (enterprise service bus) → the bad “monster”
  - No-code BPMS solutions often do not fulfil what they promise  
OR the requirements grow over time

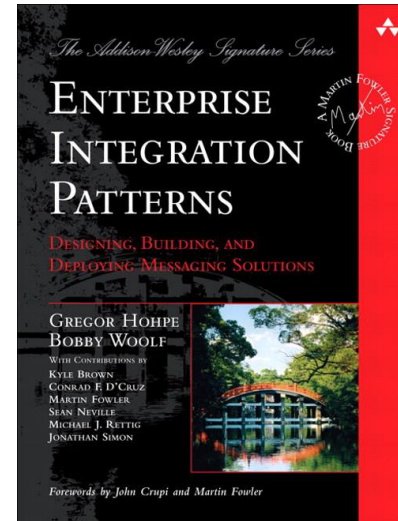




# Microservices & Event Driven Architecture



# Martin Fowler



... 2003

- Refactoring
- Unit Testing
- CI / CD
- Definition of Microservices
- ...

# Microservices

a definition of this new architectural term

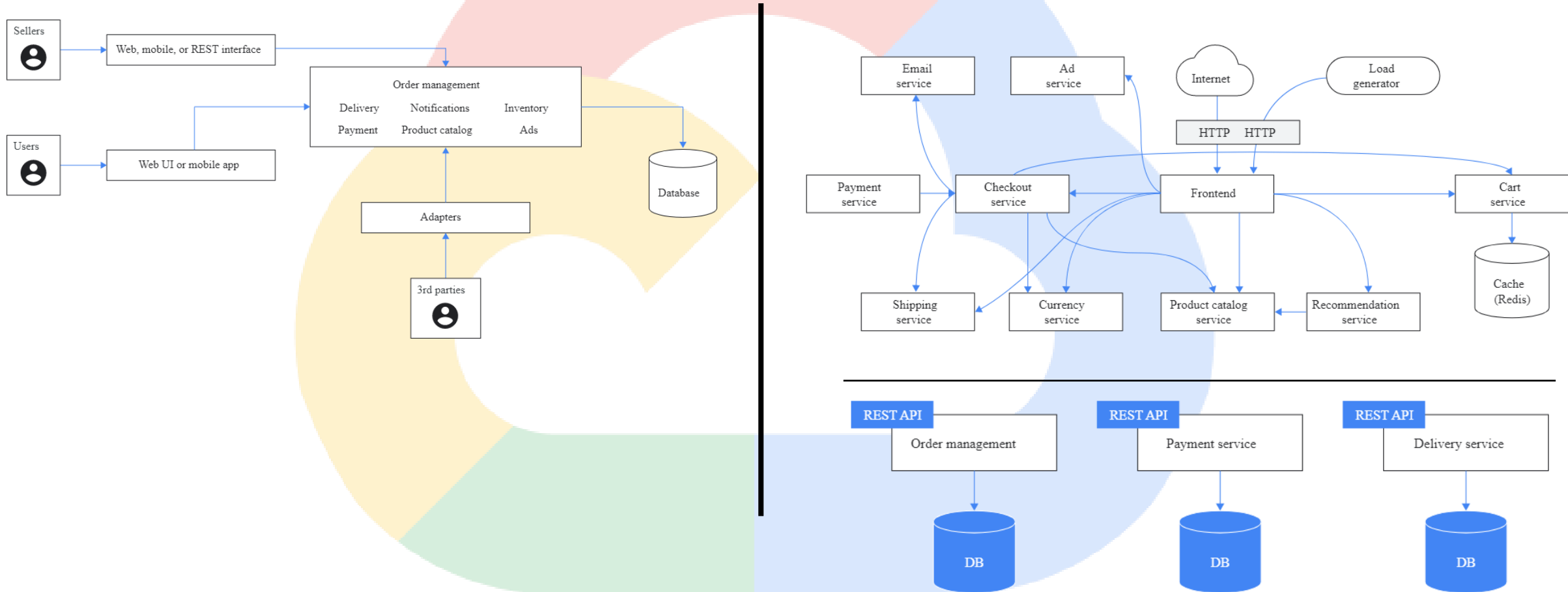
*The term "Microservice Architecture" has sprung up over the last few years to describe a particular way of designing software applications as suites of independently deployable services. While there is no precise definition of this architectural style, there are certain common characteristics around organization around business capability, automated deployment, intelligence in the endpoints, and decentralized control of languages and data.*

25 March 2014

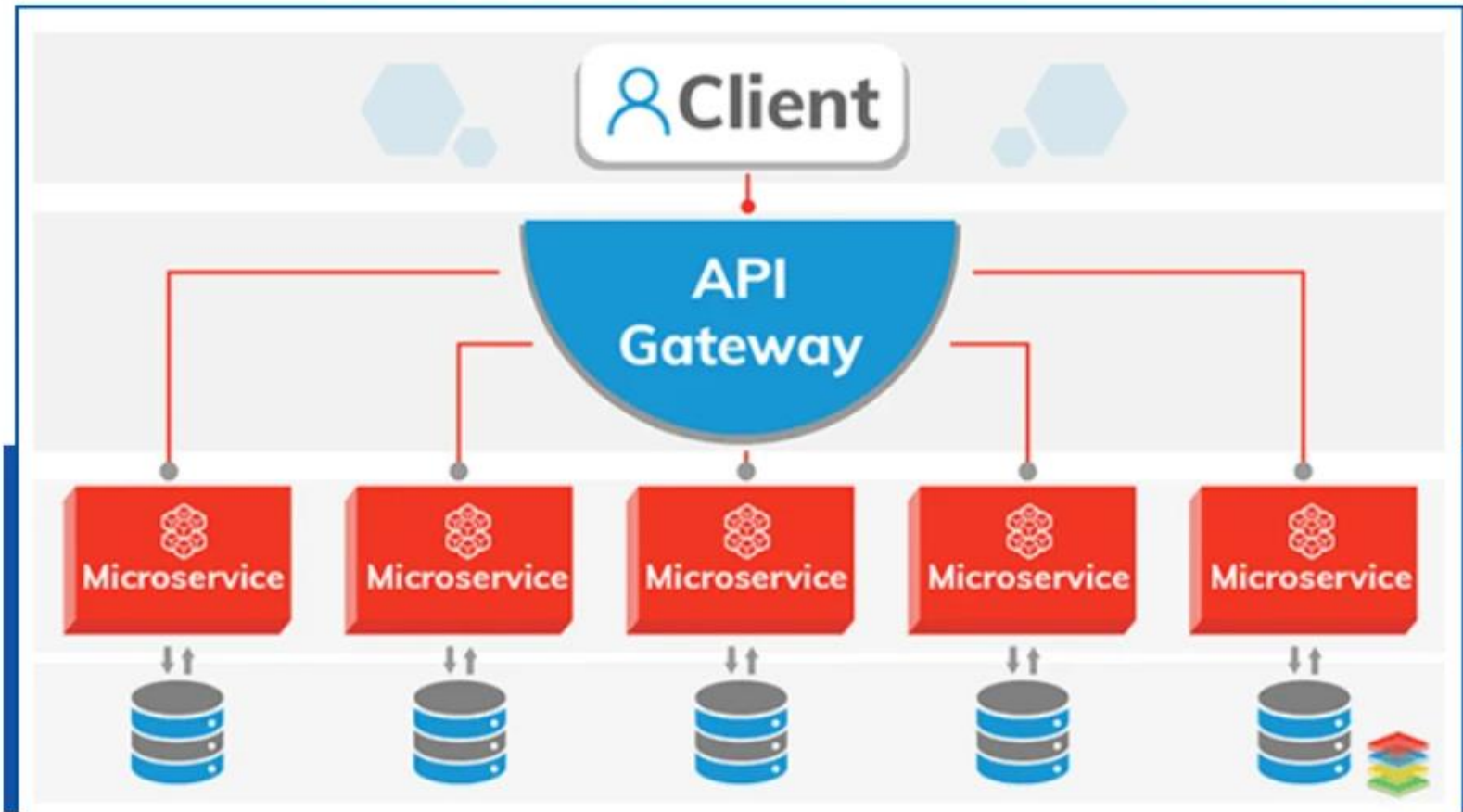
→ <https://martinfowler.com/articles/microservices.html>

# Monolith

# → Microservice

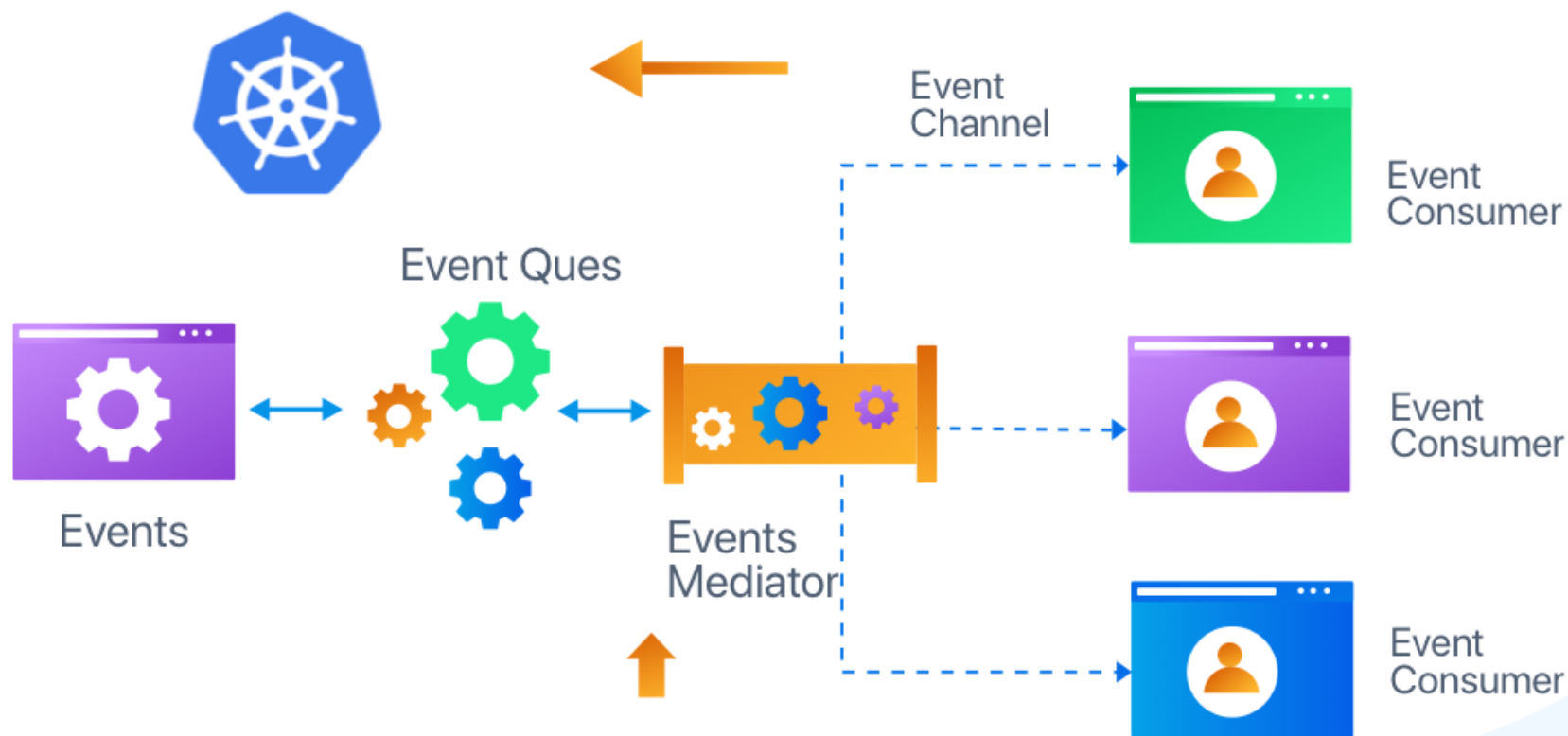


# Microservice Architectures (REST)

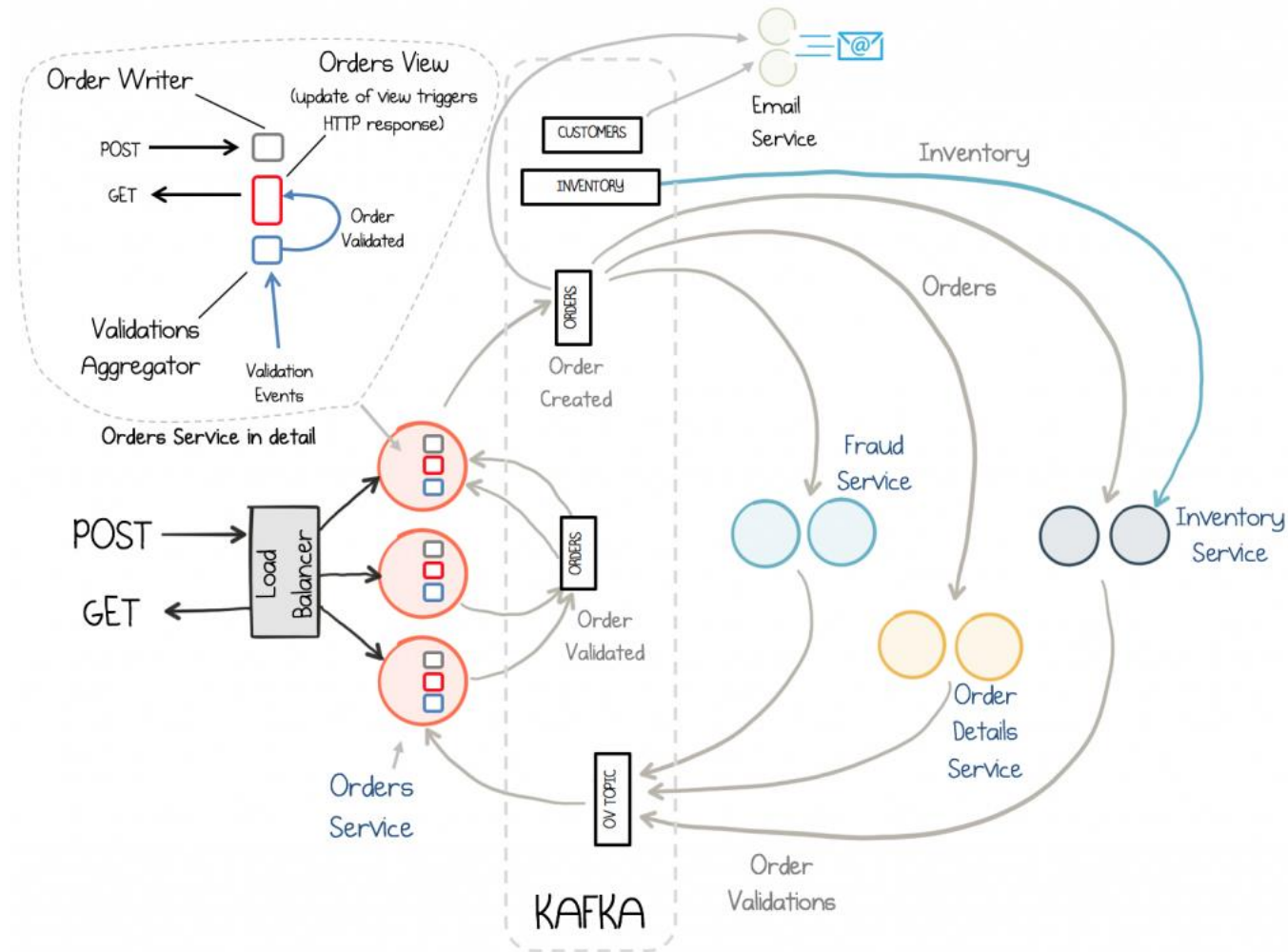




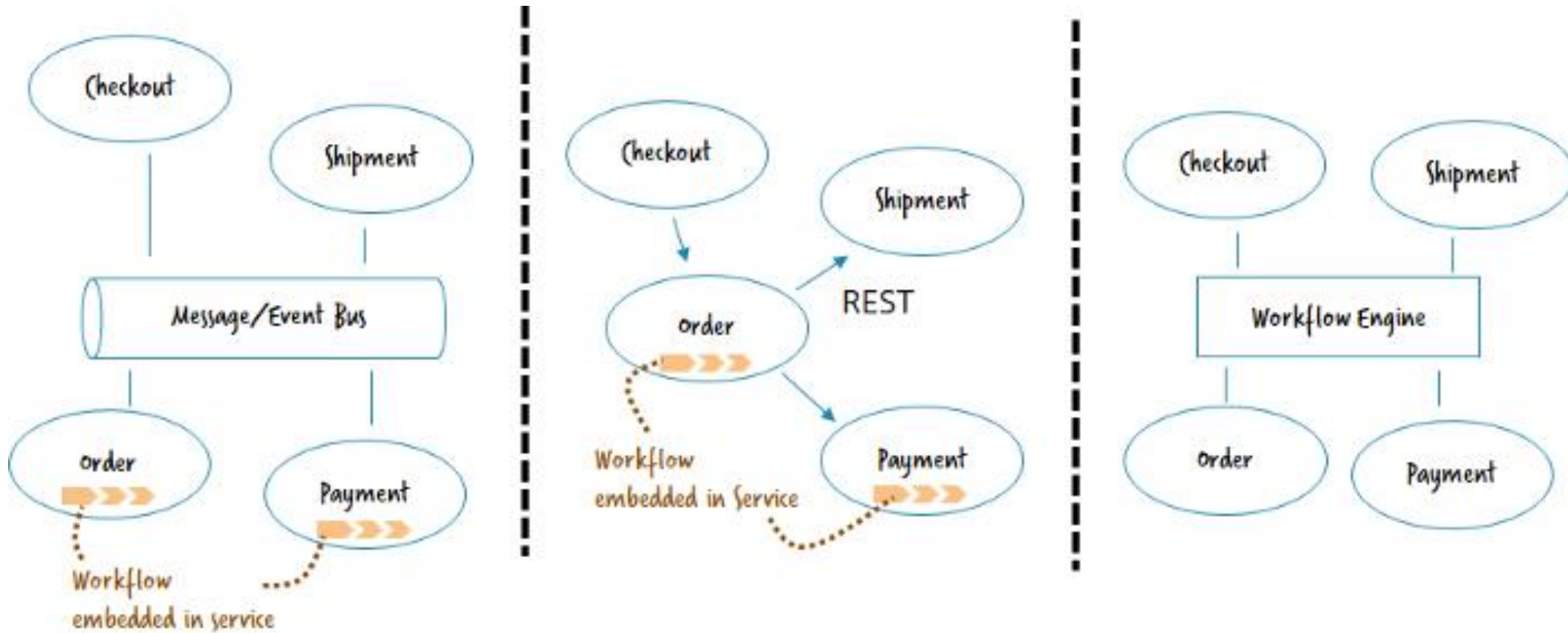
# Event-Driven Architecture for Cloud Native in Kubernetes



# Microservice Architectures (Events)



# The Microservices Workflow Automation Cheat Sheet (by Bernd Rücker)



the real world...

*Tencent* 腾讯

amazon.com<sup>®</sup>



Google



NETFLIX

facebook<sup>®</sup>



Alibaba.com<sup>™</sup>





# Software Architecture

Is these your business?

Do you need the same design principles?

**IT DEPENDS  
ON THE  
CONTEXT**

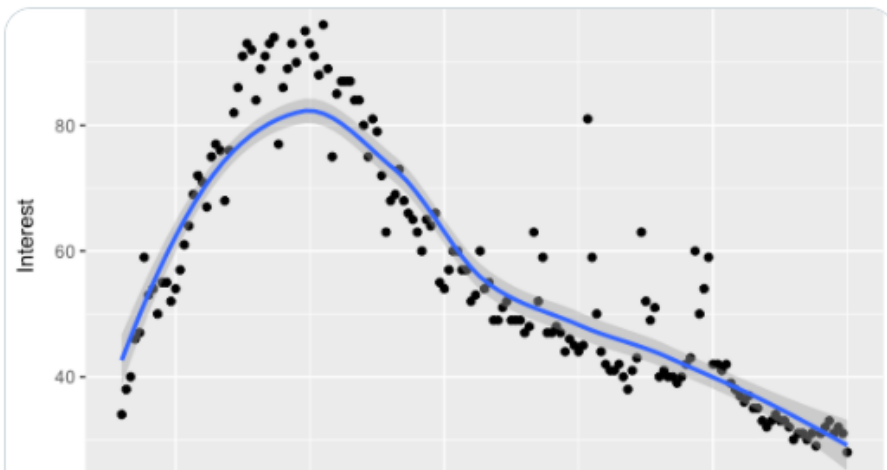


**Martin Fowler** ✓

@martinfowler



from @sogrady - "SOA" was driven by vendors, "microservices" by developers: which is why latter is doing better



The Difference Between SOA and Microservices Isn't Size

For those that have been in the technology industry for some time, there is a tendency to compare or even equate the current microservices ...

redmonk.com

8:11 PM · Jul 20, 2017



712   Reply   Share this Tweet

Read 22 replies

BPM und Microservices

## Wie lässt sich Ordnung in einen Haufen (Micro-)Services bringen?



Bernd Rücker



Daniel Meyer

21. August 2015

Microservices sind hip oder sogar Hype. Und Martin Fowler sagt, **man braucht keine Orchestrie** **ngs-Engine in Microservice-Architekturen**. Sind Workflow-Engines also überflüssig? Und wie gehen wir dann mit Geschäftsprozessen um?

[Source: <https://entwickler.de/software-architektur/wie-lasst-sich-ordnung-in-einen-haufen-micro-services-bringen/> ]



---

**Steht klassisches Business Process Management (BPM) im Widerspruch zur IT-Architektur nach dem Microservices-Ansatz? Der korrekte Microservices-Ansatz erlaubt immerhin keine zentrale unternehmensweite Prozesssteuerungskomponente. Dieser Artikel schildert, wie eine korrekte Implementierung von BPM-Systemen nach Microservices-Ansatz aussehen müsste, und analysiert die Auswirkungen dieser Fusion der Gegensätze.**

Andersherum betrachtet könnte man sagen, dass SOA und BPM richtig am Platz in Unternehmen sind, bei denen jeder Schritt nach zentraler Planung haargenau umgesetzt werden muss, wo keine ungeprüften Optimierungsaktionen zugelassen sind, zum Beispiel in der stark regulierten Pharmaindustrie.

# Macro problems with microservices

Igor Spasic



The microservice architectural style is, as Mr. Fowler defines it, an approach to develop a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and independently deployable by fully automated deployment machinery.

While the idea behind microservice architecture is fine (and not new, after all), it is often neglected how much complexity microservices bring in the development.

 7 minutes to read

[Source: <https://www.zuehlke.com/de/insights/macro-problems-microservices> ]

DIGITAL DELIVERY 5 min

## From Monolithic Pain to Microservices Hell

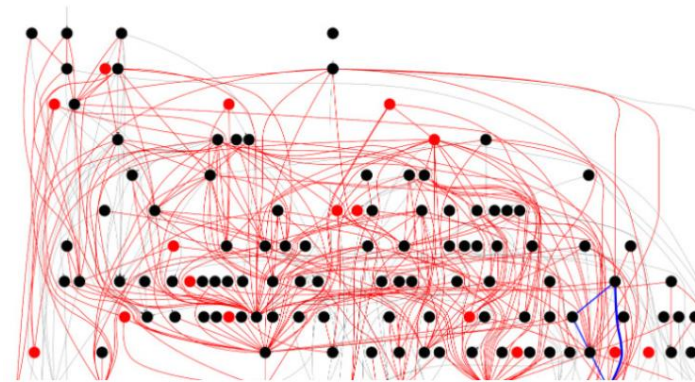
The secret to unlocking business agility and innovation in a competitive market lies with **data**. Harnessing the power in that data requires systems that can help make sense of the information.

Architecting and maintaining such a system is no easy feat. Companies will need to evaluate their application ecosystem to identify opportunities for improvement.

[Source: <https://www.adservio.fr/post/from-monolithic-pain-to-microservices-hell> ]

### Dependency Hell in Microservices and How to Avoid It

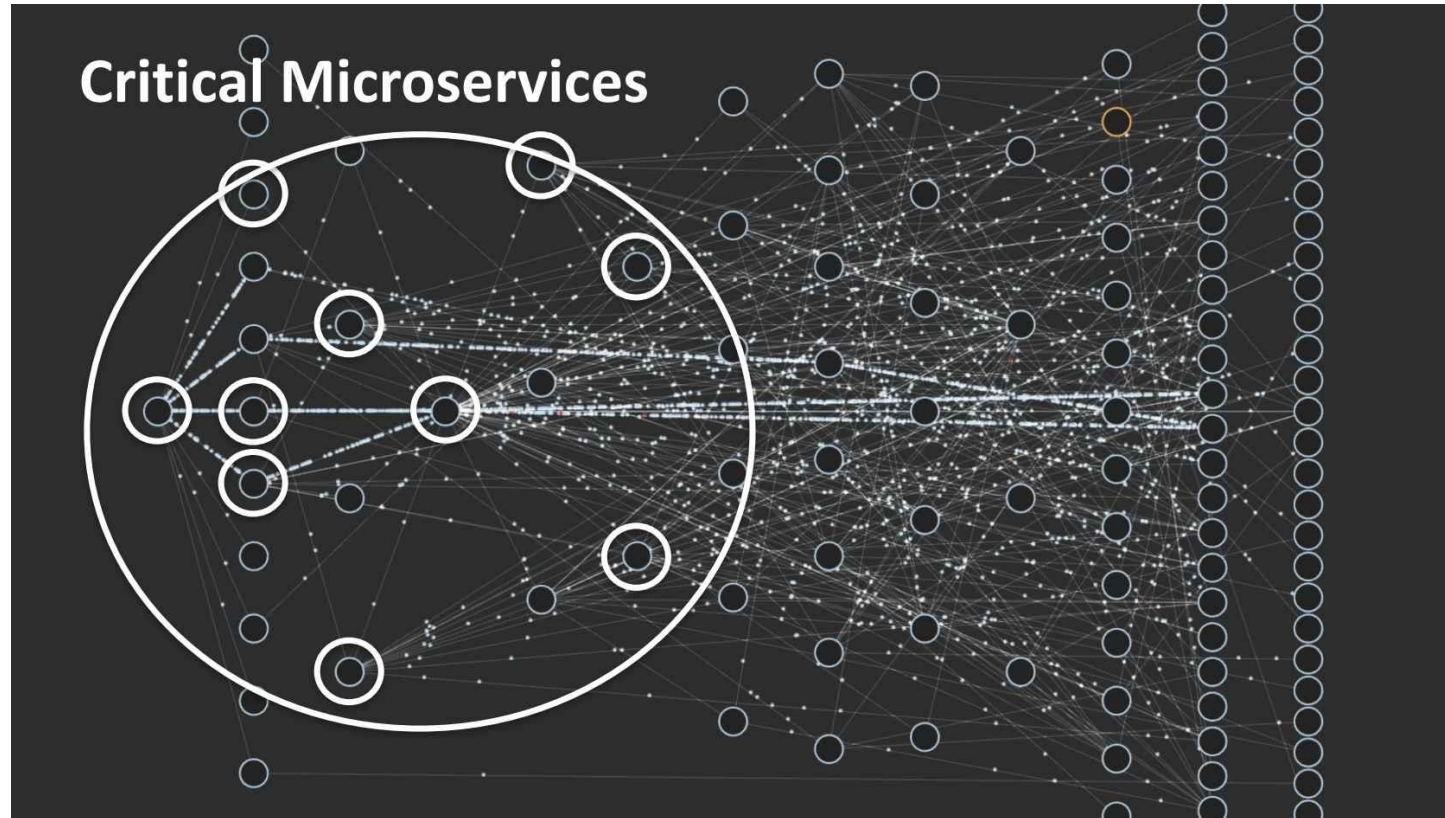
Veröffentlicht am 3. Juli 2016



[Source: <https://www.linkedin.com/pulse/dependency-hell-microservices-how-avoid-nabil-hijazi/> ]



# Microservices at Netflix (>1,000)



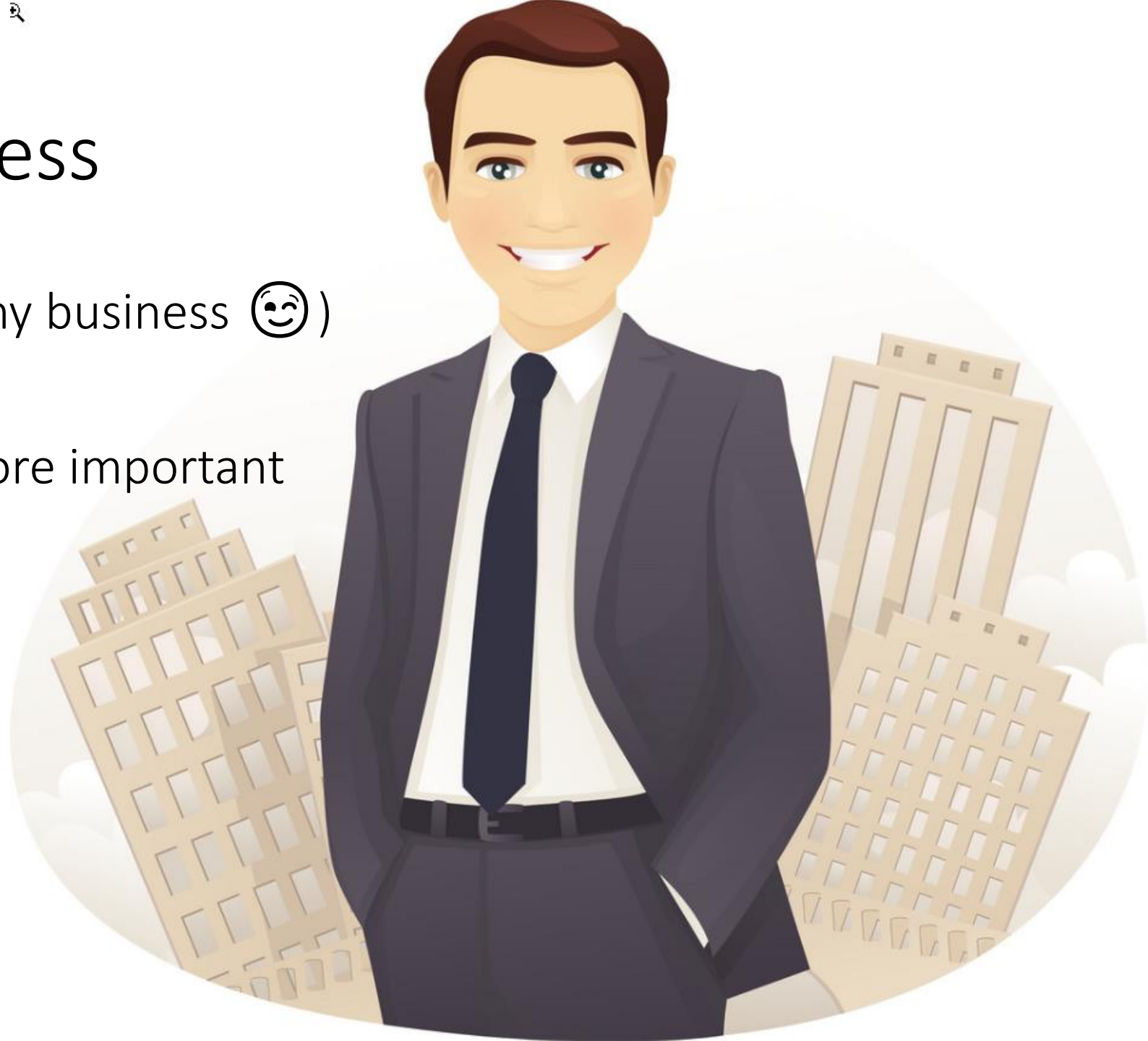
Netflix uses over 1,000 microservices now. Each deployed application controls a specific aspect of the colossal Netflix operation.

[Source: <https://www.cloudzero.com/blog/netflix-aws>]

[Source: <https://www.infoq.com/presentations/netflix-chaos-microservices/>]

# Enterprise Business

- Finance, Insurance, ... (my business 😊)
- Other requirements
- Known processes are more important than auto-scaling / HA

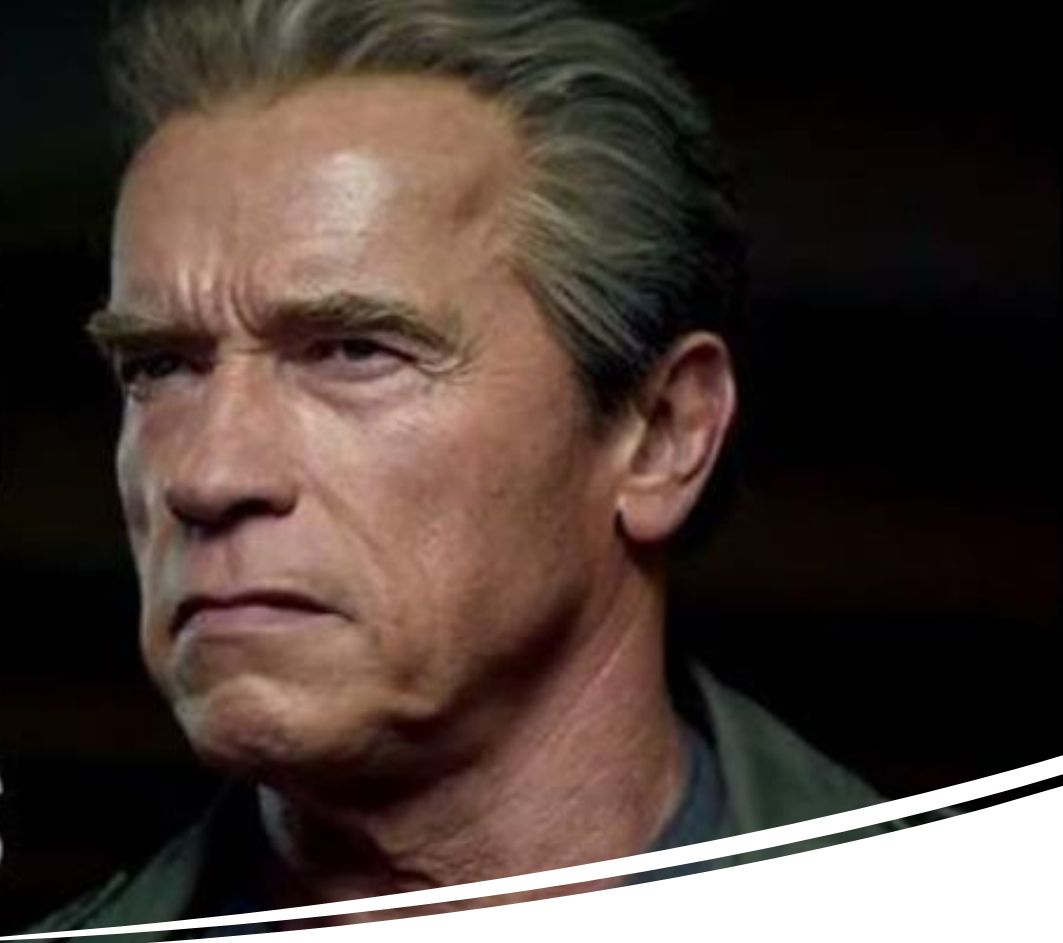


... BPMN2

**“I’M OLD BUT I’M  
NOT OBSOLETE.”**

**- POPS**

Terminator: Genisys



BPMN2.0

- ... still trendy
- ... still growing / becoming more and more popular in the enterprise world.
- e.g., Camunda. <https://camunda.com/about/customers/>

# Demo II

<https://github.com/MatthiasReining/thws2301-bpm>

## Screenplay

- Create Microservices with Quarkus
- Link Microservices to BPMN service tasks
- Error Handling / Service not available
- BPMS as queue
- Error boundary event



An iceberg floating in the ocean, with a large portion of its mass submerged below the water surface. The visible tip is jagged and white, while the submerged part is a massive, dark blue structure. The water is a deep blue, and the sky is a light blue gradient.

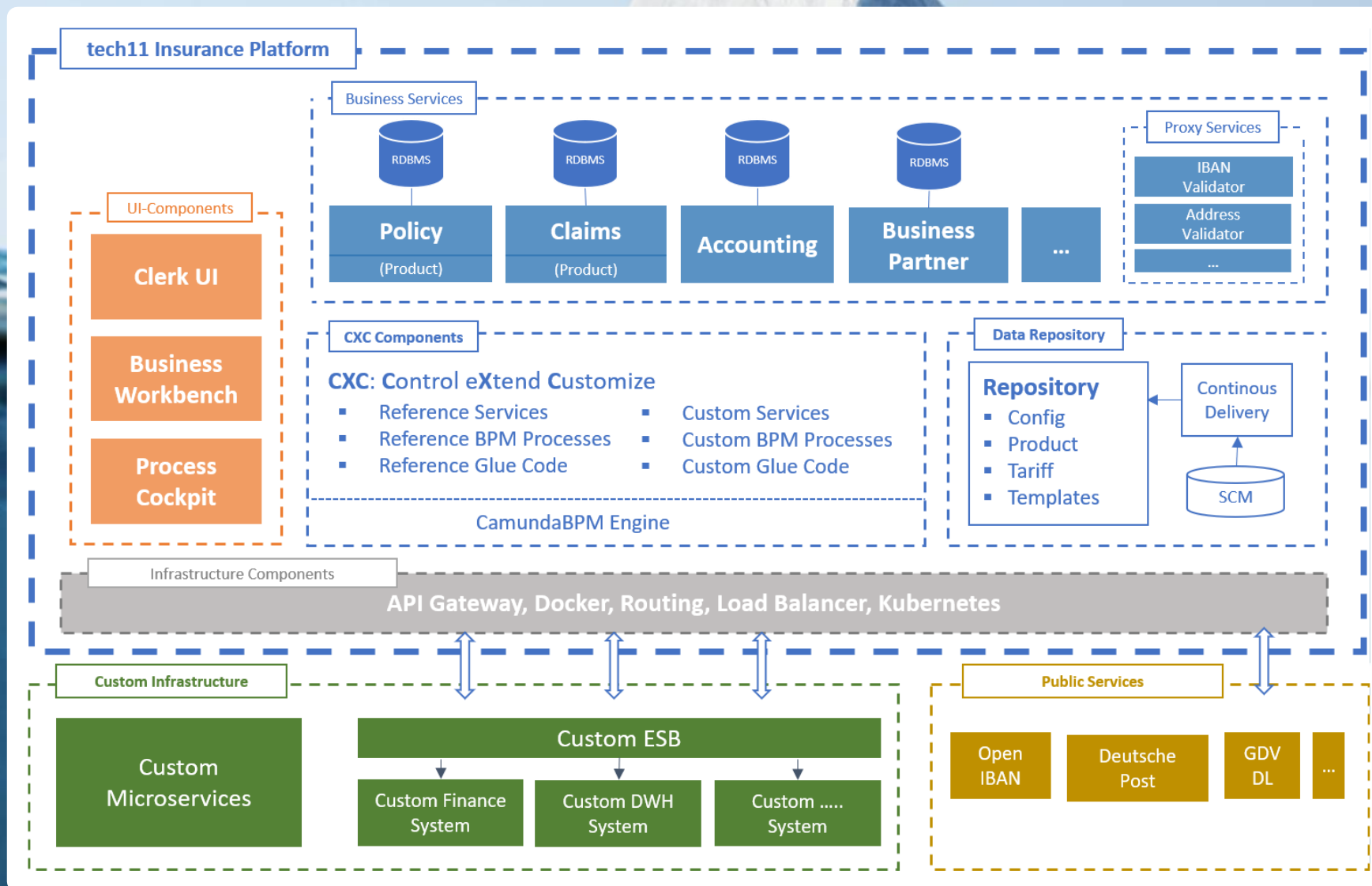
tech11

BPM @ tech11

**GO UNDERSURFACE.**

DIGITAL CORE INSURANCE PLATFORM

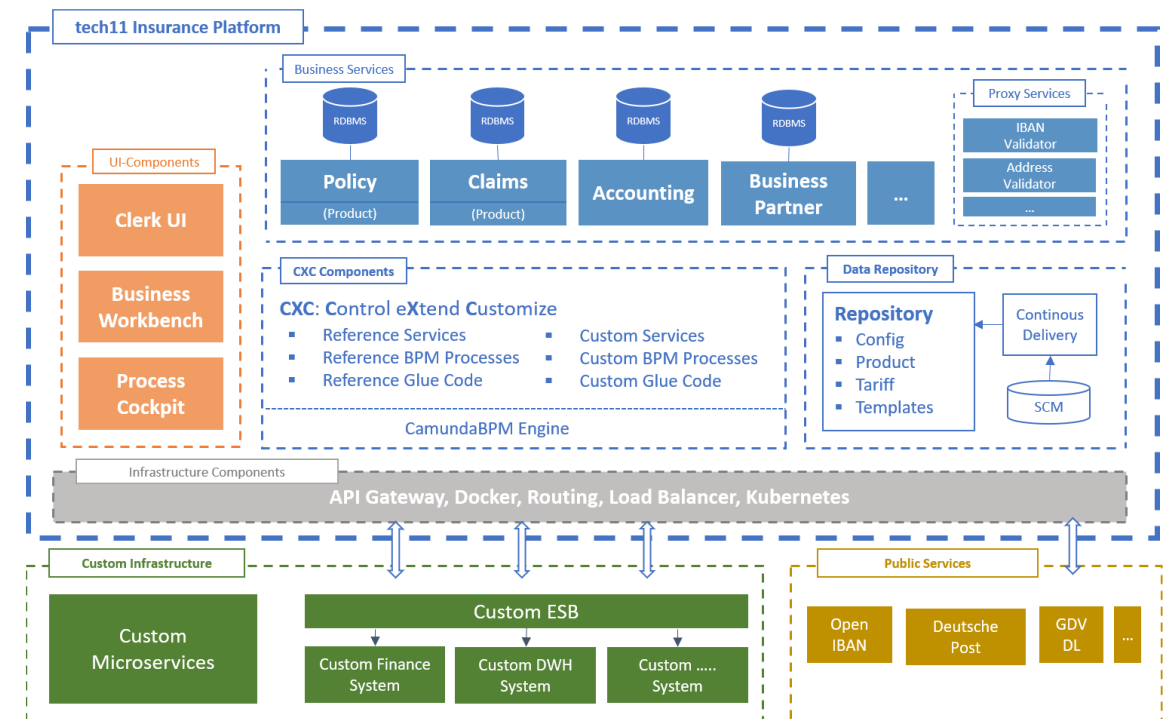
# Architecture: tech11 Insurance Platform



SOA based on  
Microservice Technology

# tech11 Insurance Platform

- Module Size: Business Design
- Modules as Products
- API Driven
- Idempotent Service Design
- Module Orchestration:
  - Camunda BPMS (as reference)
  - BPMS as “Customizing Layer”



# Customizing



Delivery

meets



Platform Development



# Customizing

- Why Customizing?
  - Individual Insurance Products (e.g., the best E-Bike Insurance)
  - Individual Processes (faster, leaner, closer to the customer, ...)
- Product → Customizing → Custom Fit
  - Starting with a GAP analysis (custom need, tech11 reference)
  - Discuss with customers (based on BPM diagrams)



# Summary

- Should I use a BPMS?
  - → „it depends“
  - Custom Solution vs. Product
  - Is an e2e visibility required?
- The “enterprise world” like structured processes
- (Normally) you are not Netflix
- Respect the KISS principle: **keep it simple, stupid**
  - Think about the requirements (including non-functional)
  - Choose your architecture
    - Avoid “cargo cult programming”
  - Depending on the context, BPMS can bring a real added value

sounds interesting? ... join tech11



Thank you for  
your attention