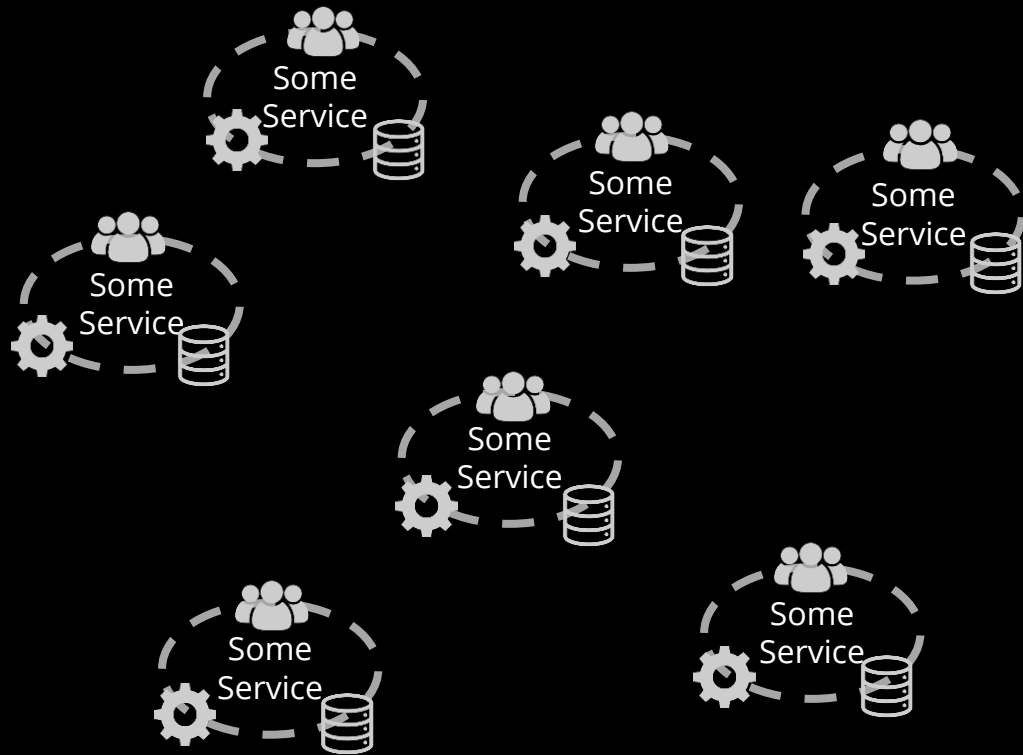


# The Big Picture Monitoring and orchestration of Your Microservices Landscape with Kafka and Zeebe

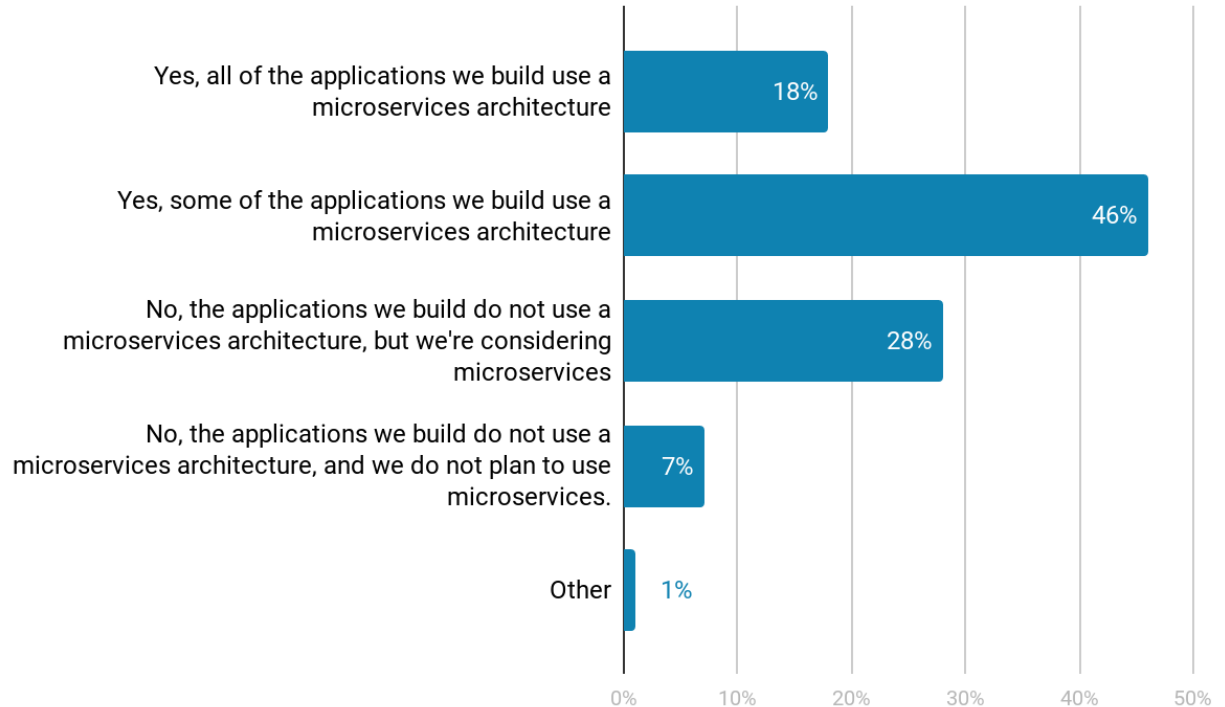
@berndruecker



# Microservices...



# Data, data, data, ...



Source: Microservices orchestration survey,  
July 2018, 354 responses



92 %


A horizontal bar chart is displayed against a black background. The bar is composed of two segments: a dark teal segment on the left and a light blue segment on the right. The dark teal segment represents 92% of the total length, and the light blue segment represents the remaining 8%.

Category	Percentage
Dark Teal	92%
Light Blue	8%



The microservice community favours an alternative approach: smart endpoints and dumb pipes.

[... Microservices ...] are choreographed using simple RESTish protocols rather than complex protocols such as WS-Choreography or BPEL or orchestration by a central tool.

A collage of images related to retail and logistics, including hands holding a card, a laptop with a spreadsheet, a person at a computer, and various retail items like shoes and handbags.

# ORCHESTRATING A HIGHLY-SCALABLE FULFILLMENT PROCESS

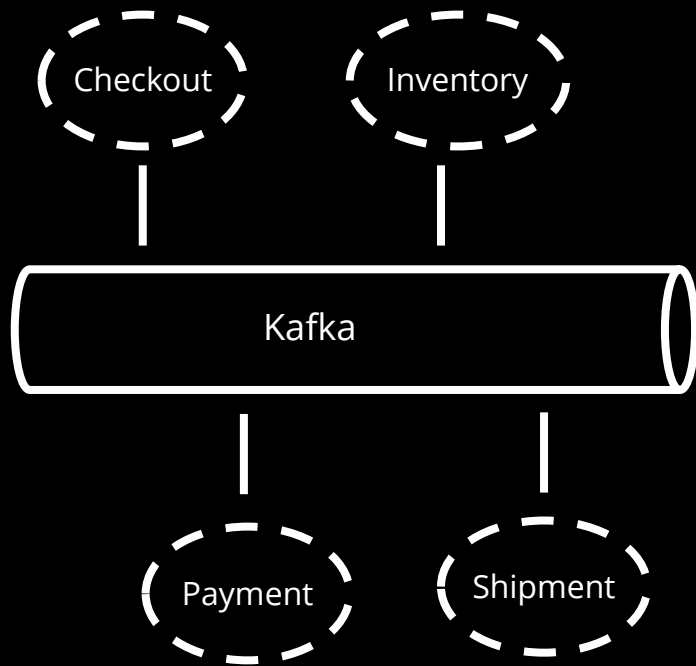
JÖRN HORSTMANN  
LUKAS NIEMEIER

2017-05-19

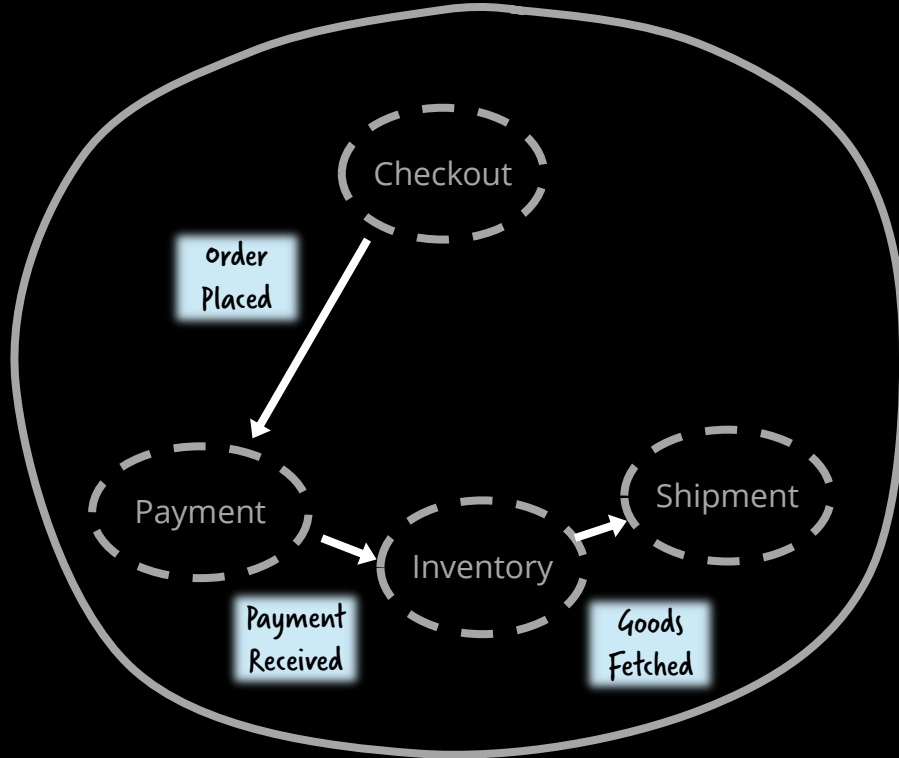


**THE SHUTTLE**  
TECH INNOVATION LAB

# De-coupled services with event backbone

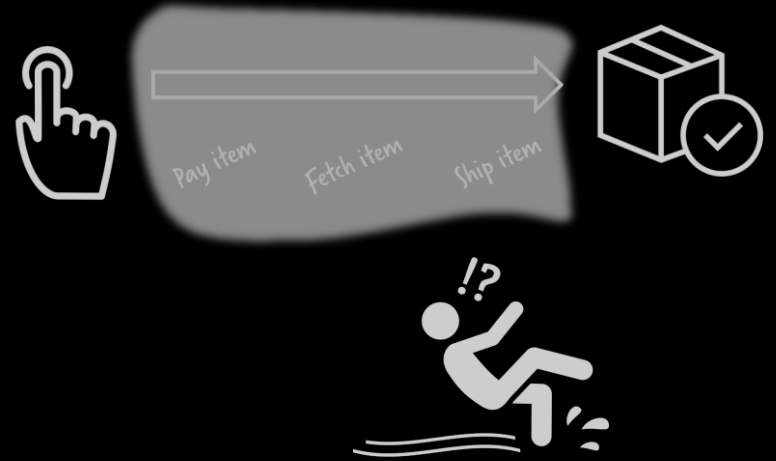
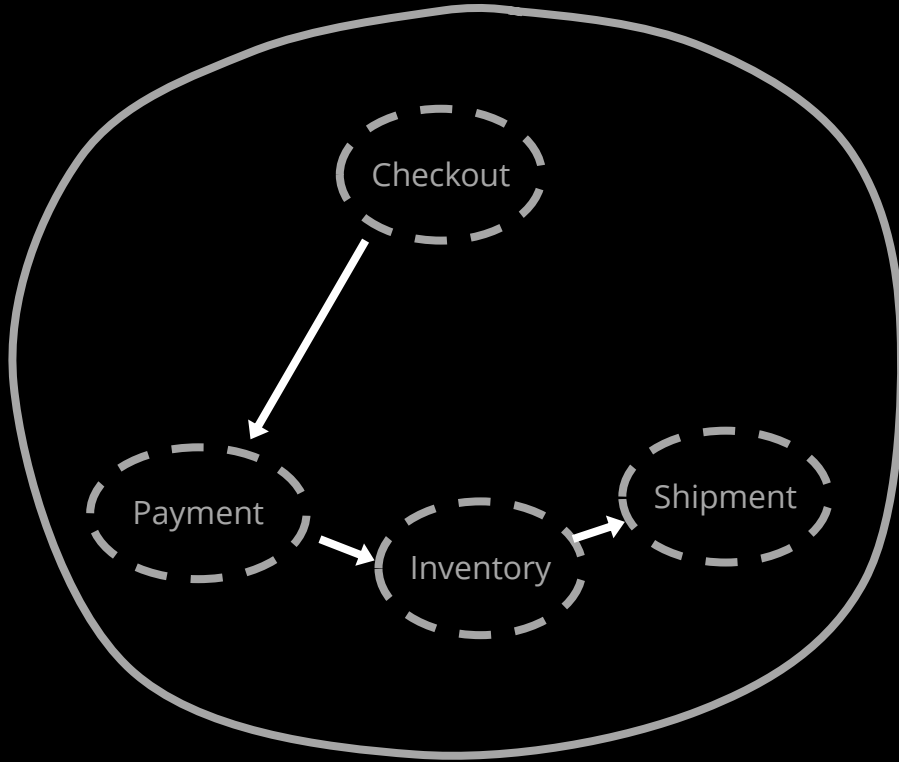


# Event-driven





# Peer-to-peer event chain





The danger is that it's very easy to make nicely decoupled systems with event notification, without realizing that you're losing sight of that larger-scale flow, and thus set yourself up for trouble in future years.

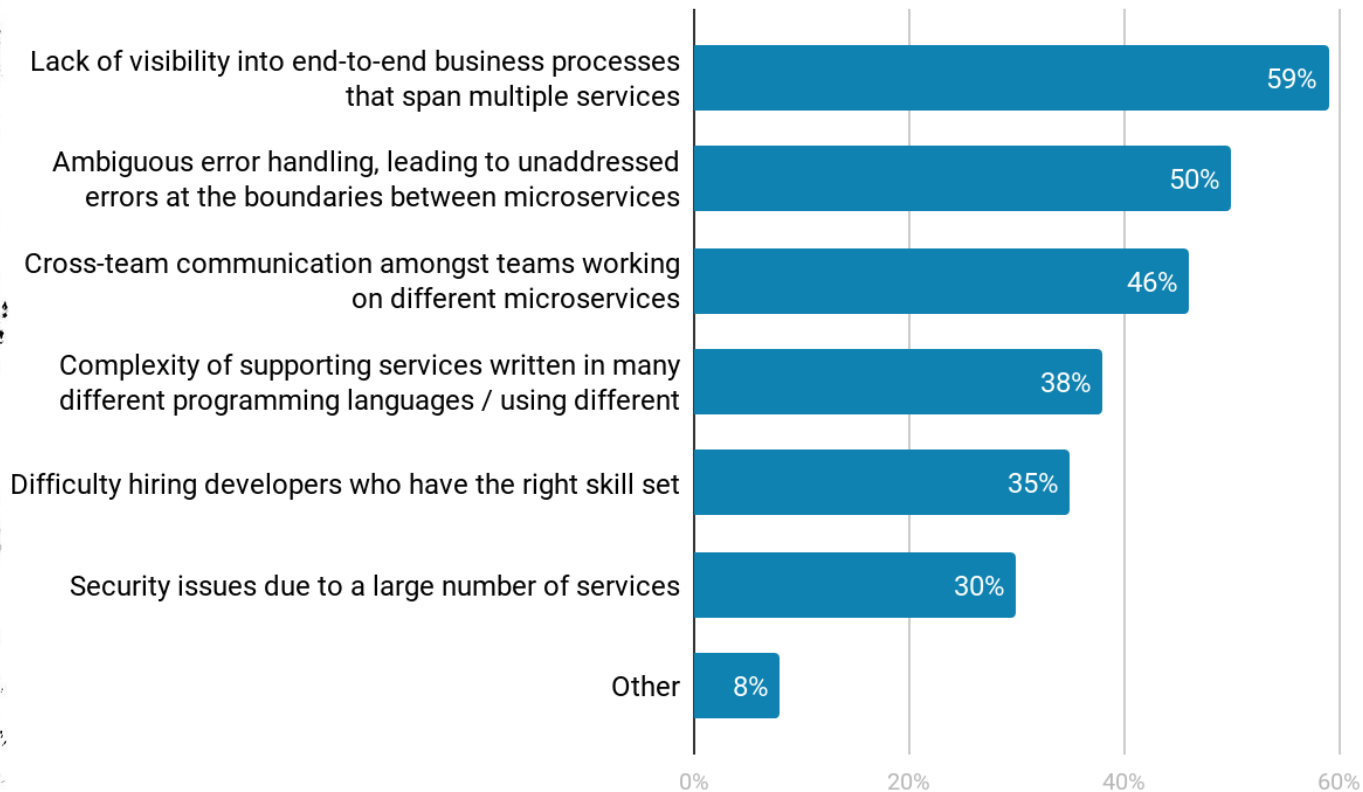


The danger is that it's very easy to make nicely decoupled systems with event notification, without realizing that you're losing sight of that larger-scale flow, and thus set yourself up for trouble in future years.



The danger is that it's very easy to make nicely decoupled systems with event notification, without realizing that you're losing sight of that larger-scale flow, and thus set yourself up for trouble in future years.

# Challenges?



# Microservice pioneers have become aware

Traditionally, some of these processes had been orchestrated in an ad-hoc manner using a combination of pub/sub, making direct REST calls, and using a database to manage the state. However, as the number of microservices grow and the complexity of the processes increases, getting visibility into these distributed workflows becomes difficult without a central orchestrator.



Netflix Technology Blog [Follow](#)

Learn more about how Netflix designs, builds, and operates our systems and engineering organizations

Dec 12, 2016 · 7 min read

## Netflix Conductor: A microservices orchestrator

The Netflix Content Platform Engineering team runs a number of business processes which are driven by asynchronous orchestration of tasks executing on microservices. Some of these are long running processes spanning several days. These processes play a critical role in getting titles ready for streaming to our viewers across the globe.

A few examples of these processes are:

- Studio partner integration for content ingestion
- IMF based content ingestion from our partners
- Process of setting up new titles within Netflix
- Content ingestion, encoding, and deployment to CDN

Traditionally, some of these processes had been orchestrated in an ad-hoc manner using a combination of pub/sub, making direct REST calls, and using a database to manage the state. However, as the number of microservices grow and the complexity of the processes increases, getting visibility into these distributed workflows becomes difficult without a central orchestrator.

What we wanted



vs. what we got

# Finding the right balance

## observe Microservice Paradigms

- Loose Coupling
- Autonomy
- Event Collaboration



## Ensure Flow Manageability


- End-To-End Visibility
- Meet Business Process SLA
- Support Continuous Process Improvement



Workflow automation | orchestration  
is an essential building block  
in microservice architectures

of course I am opinioated – but I truly believe this!

orchestration is not a central thing!  
„It is not BPEL“



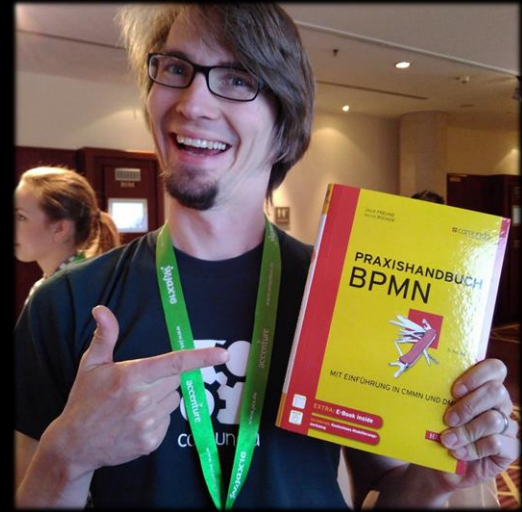
Workflow automation | orchestration  
is an essential building block  
in microservice architectures

of course I am opinioated – but I truly believe this!

**Warning:  
Contains Opinion**



Bernd Ruecker  
Co-founder and  
Developer Advocate of  
Camunda



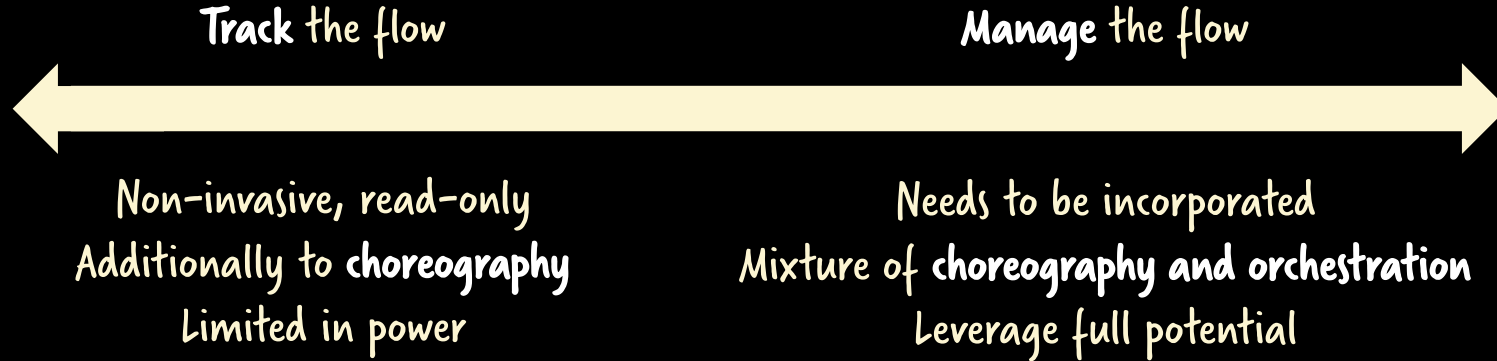
HQ Germany (Berlin) + offices in US



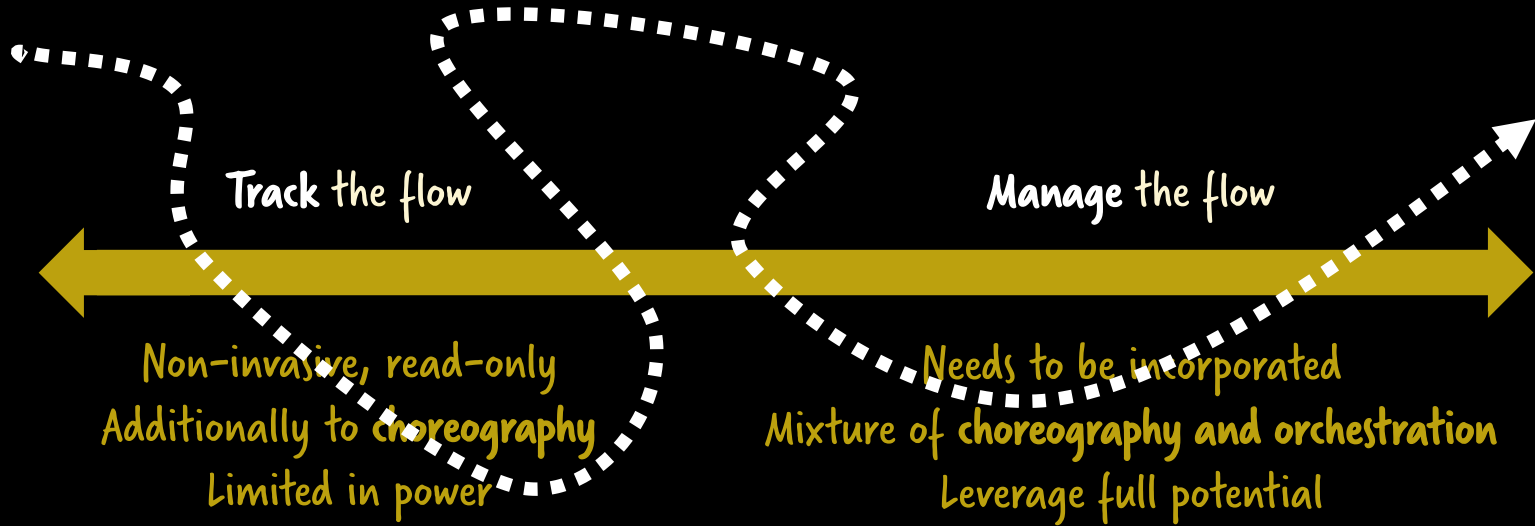
bernd.ruecker@camunda.com  
@berndruecker



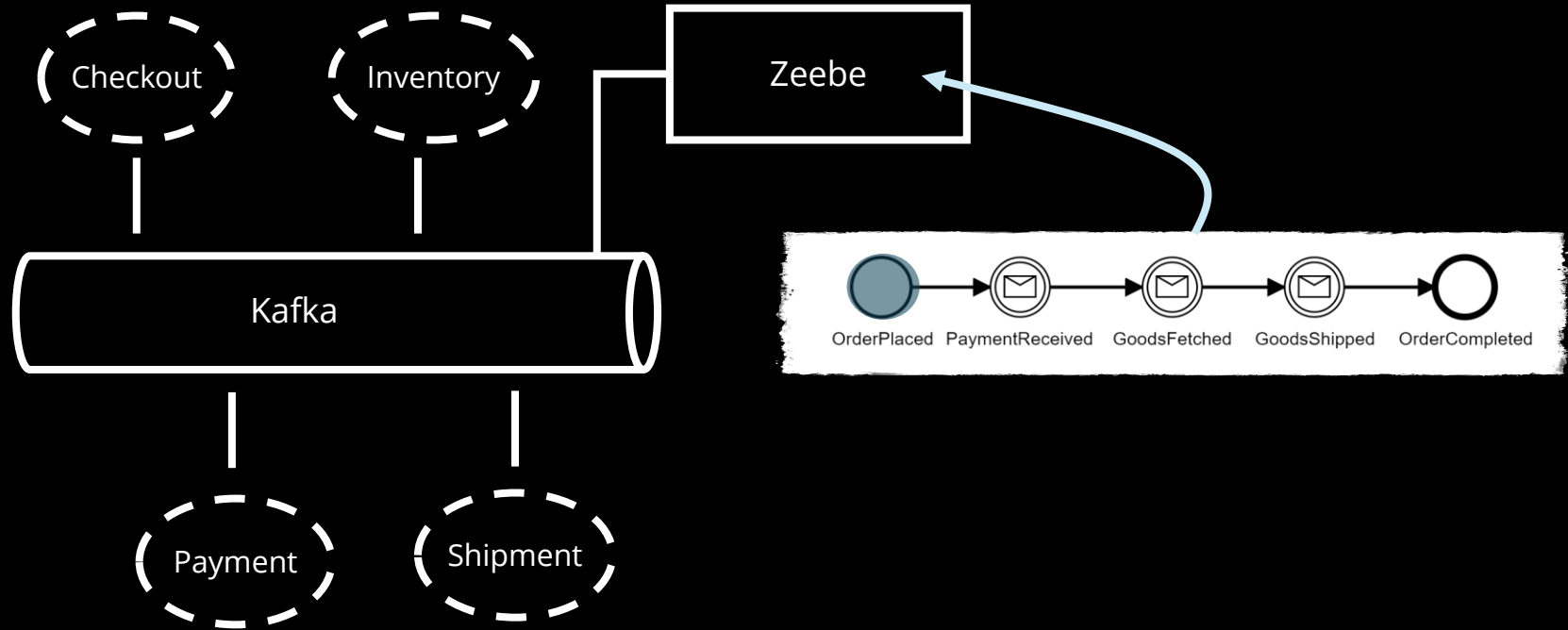
# Different levels of leveraging workflow automation



# A possible journey from pogo to beauty...

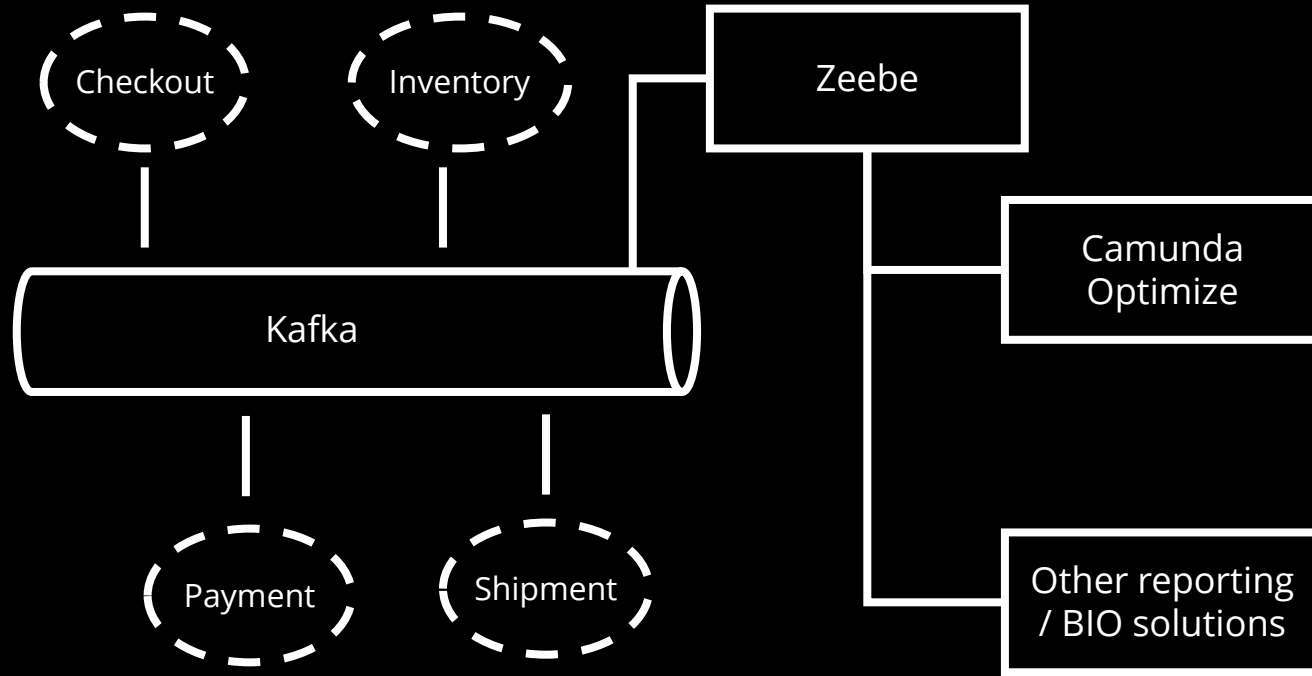


# Tracking

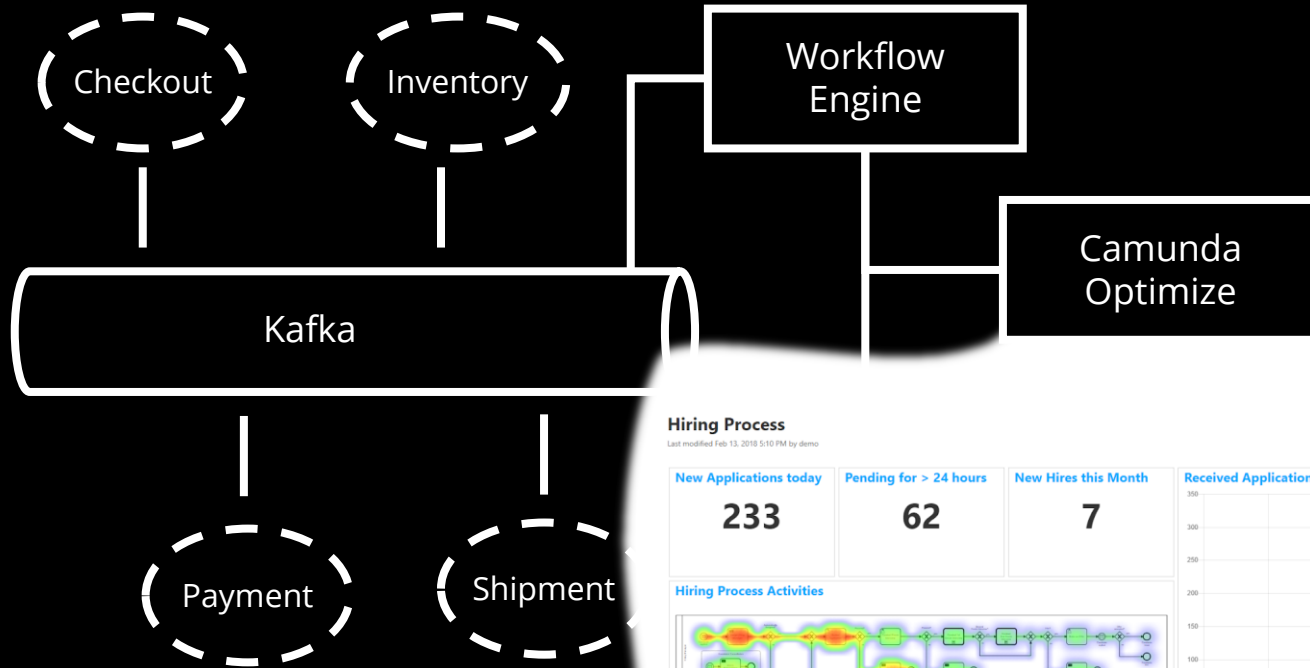




# Tracking + BI



# Tracking + BI



## Hiring Process

Last modified Feb 13, 2018 5:10 PM by demo

New Applications today

233

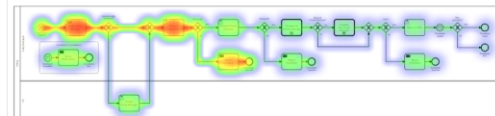
Pending for > 24 hours

62

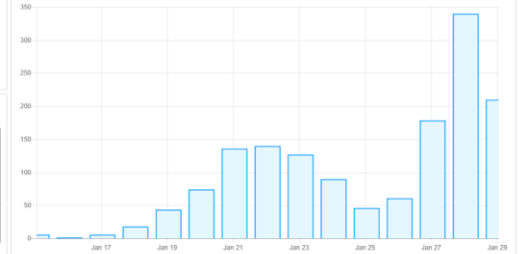
New Hires this Month

7

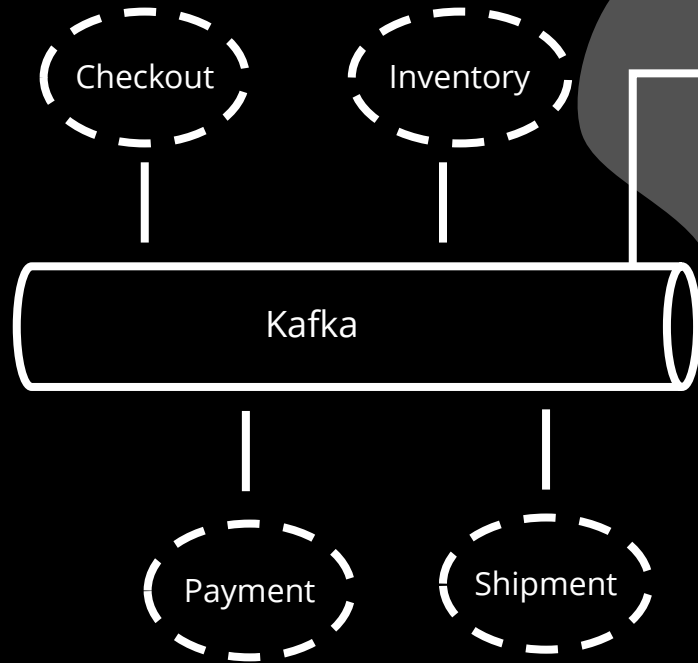
Hiring Process Activities



Received Applications per Day

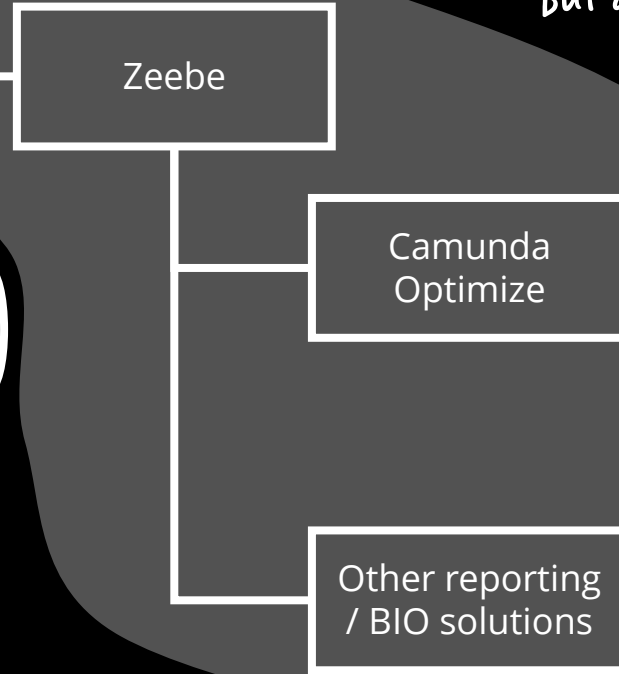


# Tracking + BI

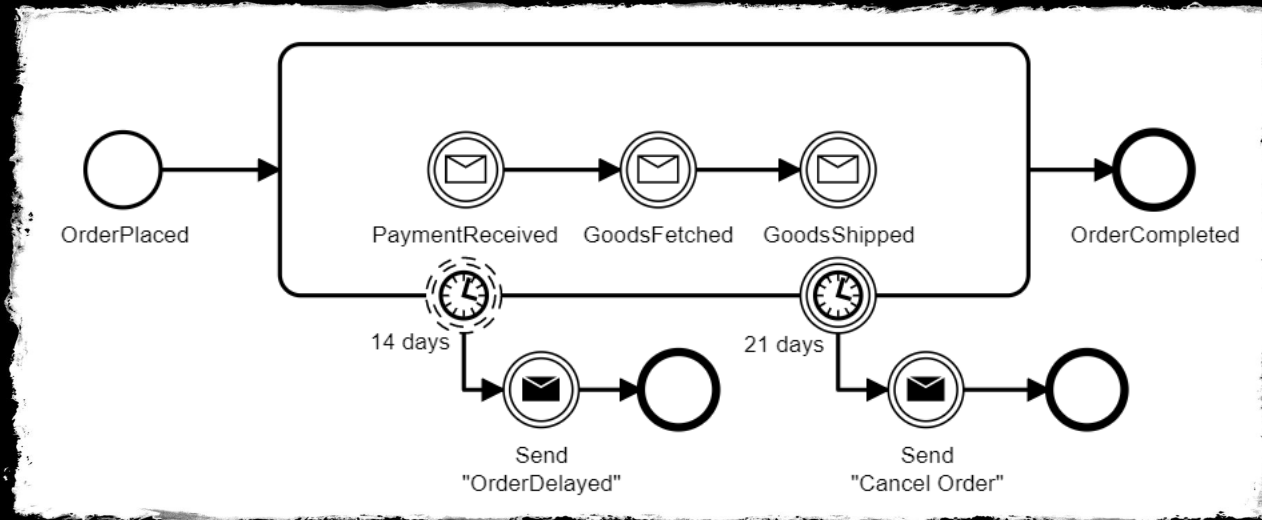


*This is totally non-invasive*

*But also limited in power*



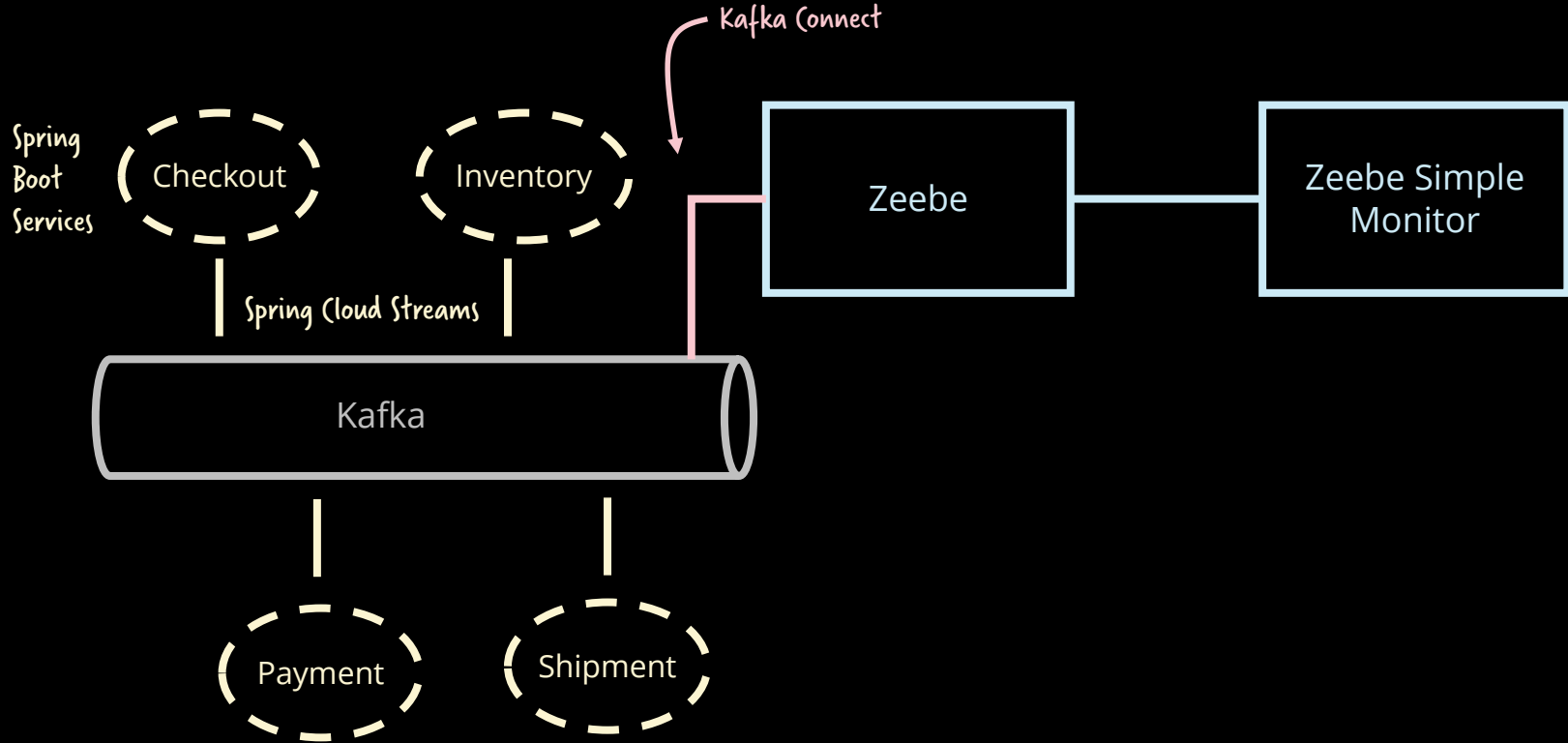
But you can start acting upon certain events



Live hacking



# Demo architecture

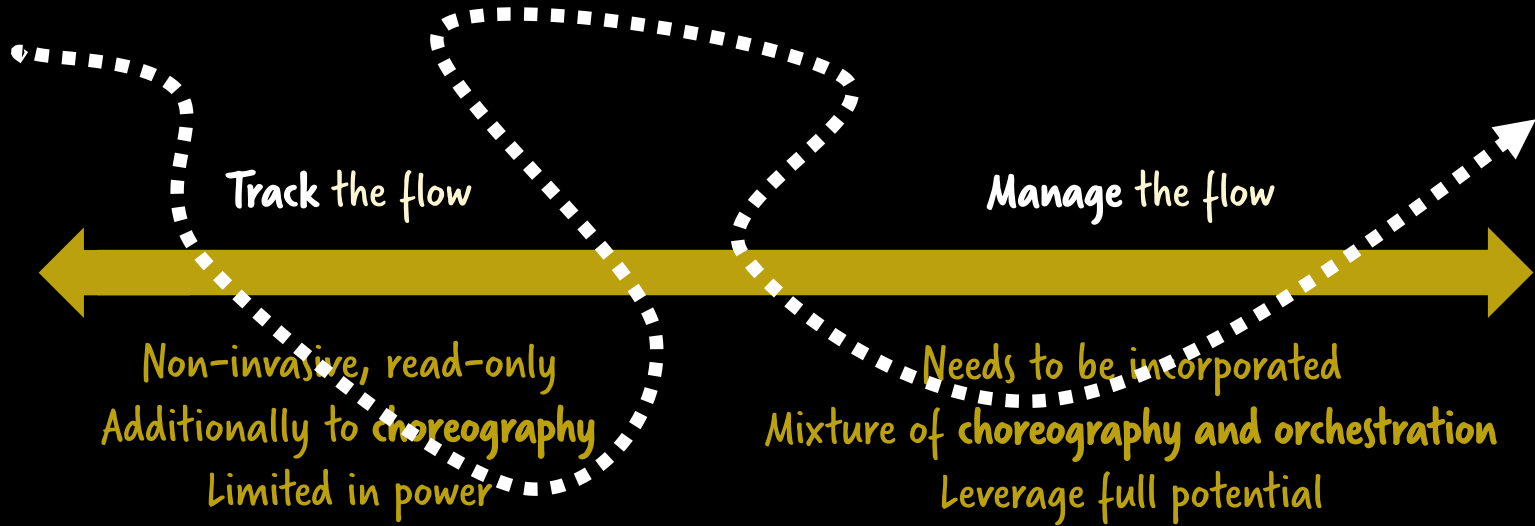


Why so shy?



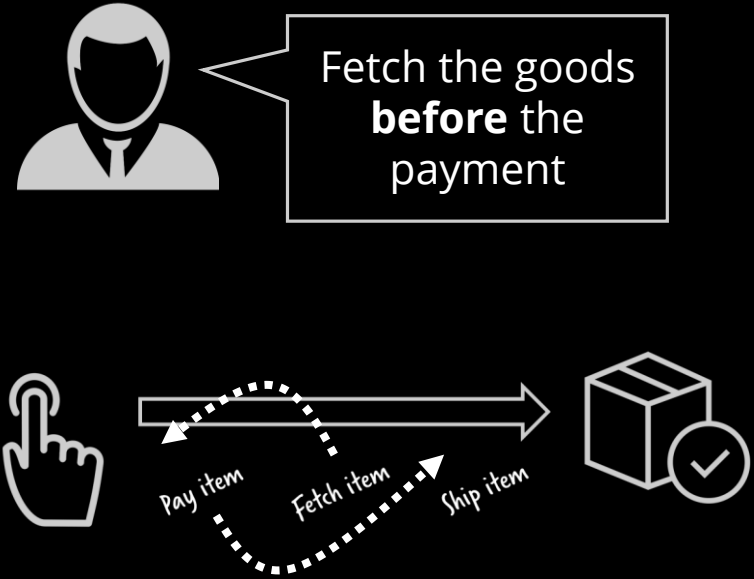
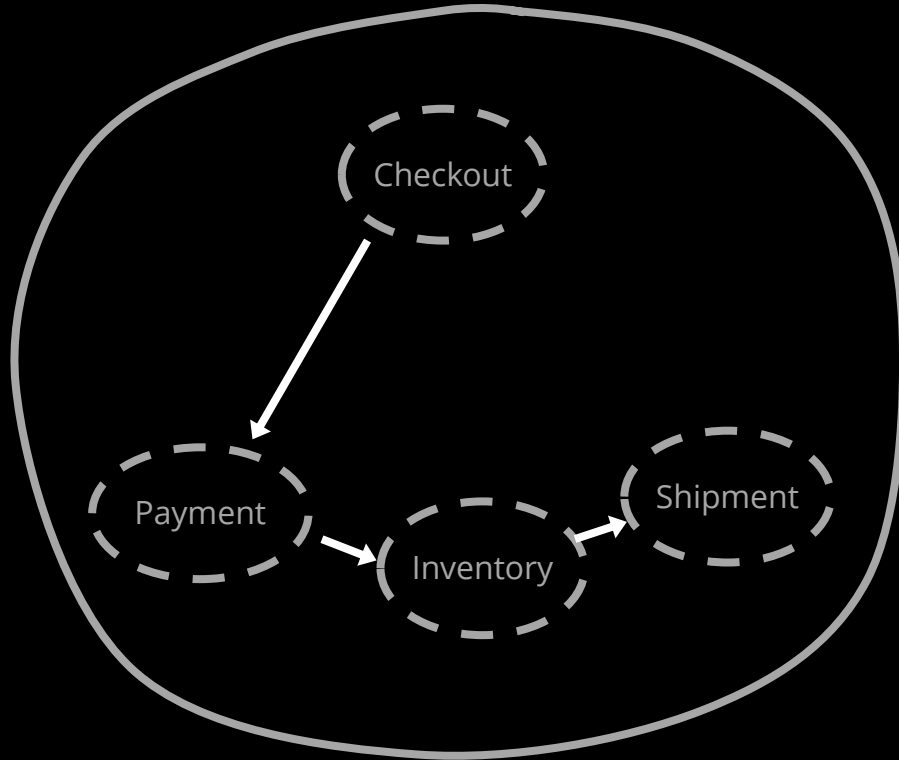
Photo by [LuidmilaKot](#), available under [Creative Commons CC0 1.0 License](#).

# A possible journey away from pogo...





# Changing the flow



# Changing the flow

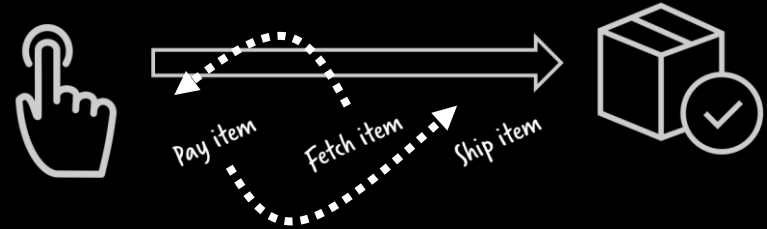
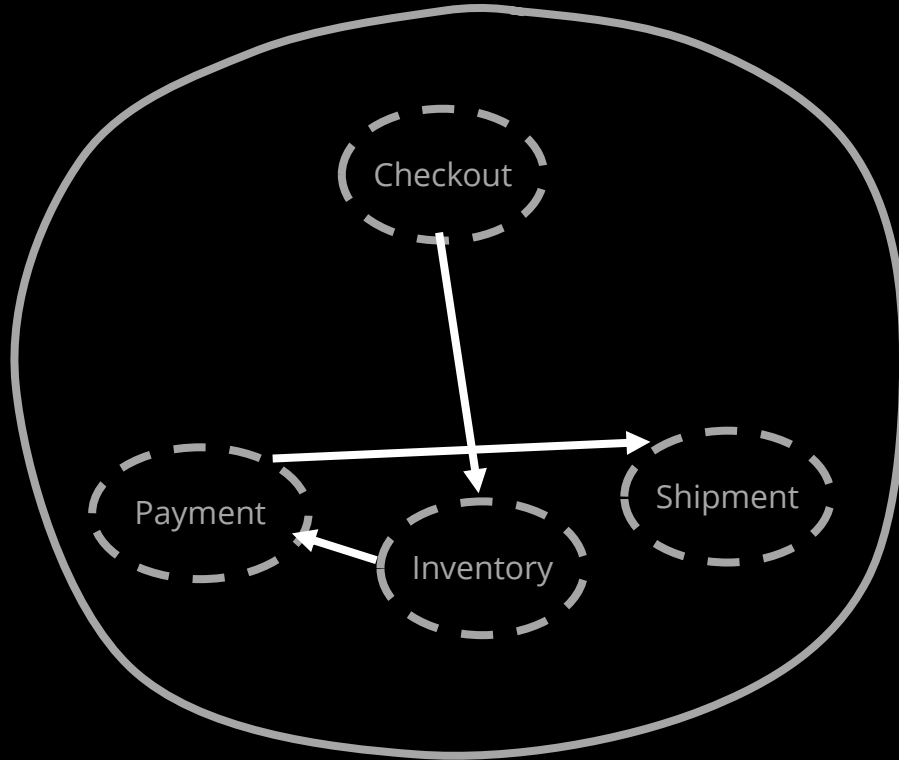
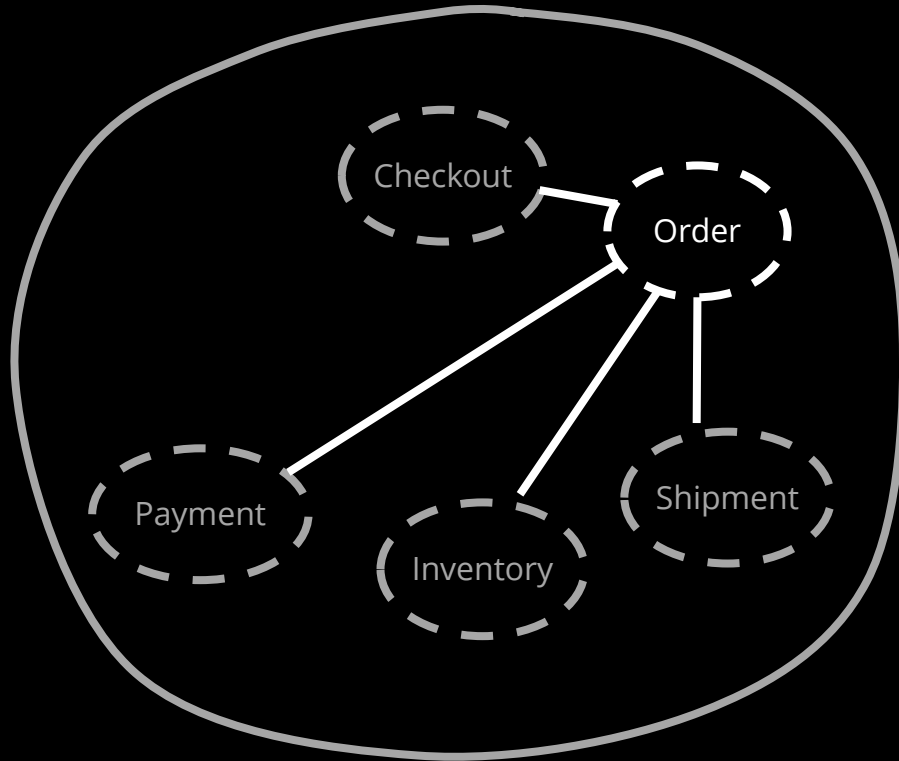




Photo by born1945, available under [Creative Commons BY 2.0 license](#).

Important responsibilities need a home



# Distributed systems





# Distributed systems



WIKIPEDIA  
The Free Encyclopedia

[Main page](#)  
[Contents](#)  
[Featured content](#)  
[Current events](#)  
[Random article](#)  
[Donate to Wikipedia](#)  
[Wikipedia store](#)

Interaction

[Help](#)  
[About Wikipedia](#)  
[Community portal](#)  
[Recent changes](#)  
[Contact page](#)

Tools

[What links here](#)  
[Related changes](#)  
[Upload file](#)

Article [Talk](#)

## Fallacies of distributed computing

From Wikipedia, the free encyclopedia

The **fallacies of distributed computing** are a set of assertions made by [L Peter Deutsch](#) and others at [Sun Microsystems](#)

### Contents [\[hide\]](#)

- [1 The fallacies](#)
- [2 The effects of the fallacies](#)
- [3 History](#)
- [4 See also](#)
- [5 References](#)
- [6 External links](#)

## The fallacies [\[ edit \]](#)

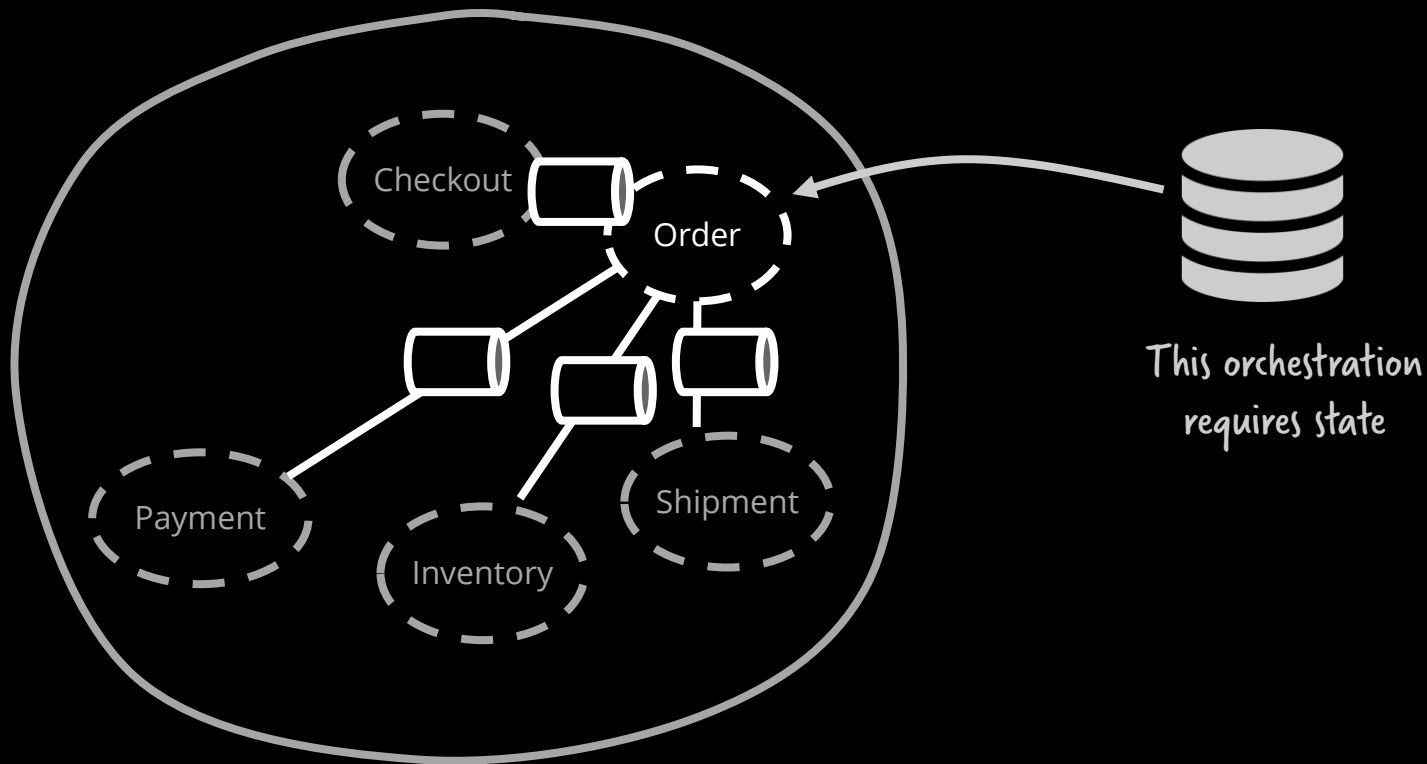
The fallacies are:<sup>[1]</sup>

1. The network is reliable.
2. Latency is zero.
3. Bandwidth is infinite.

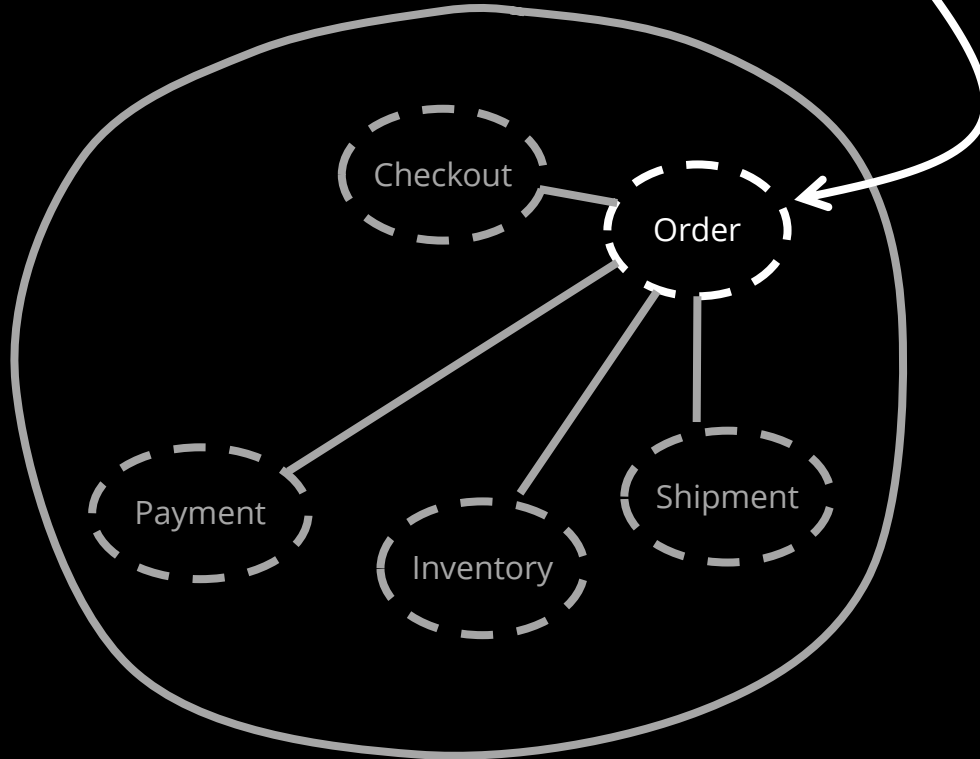
# Distributed systems



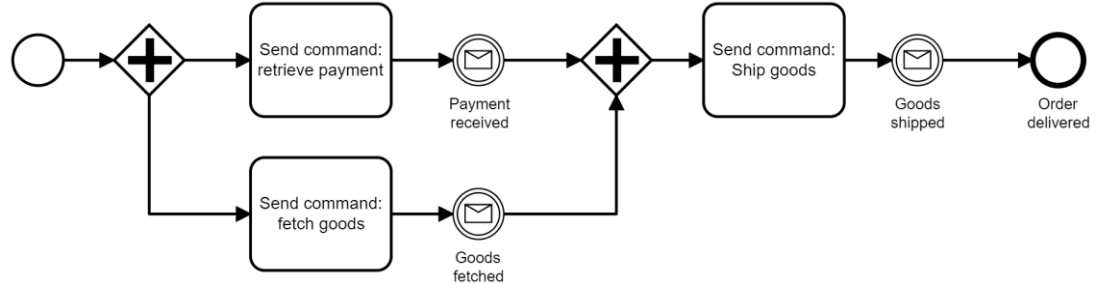
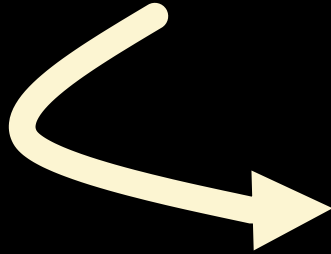
# Stateful orchestration



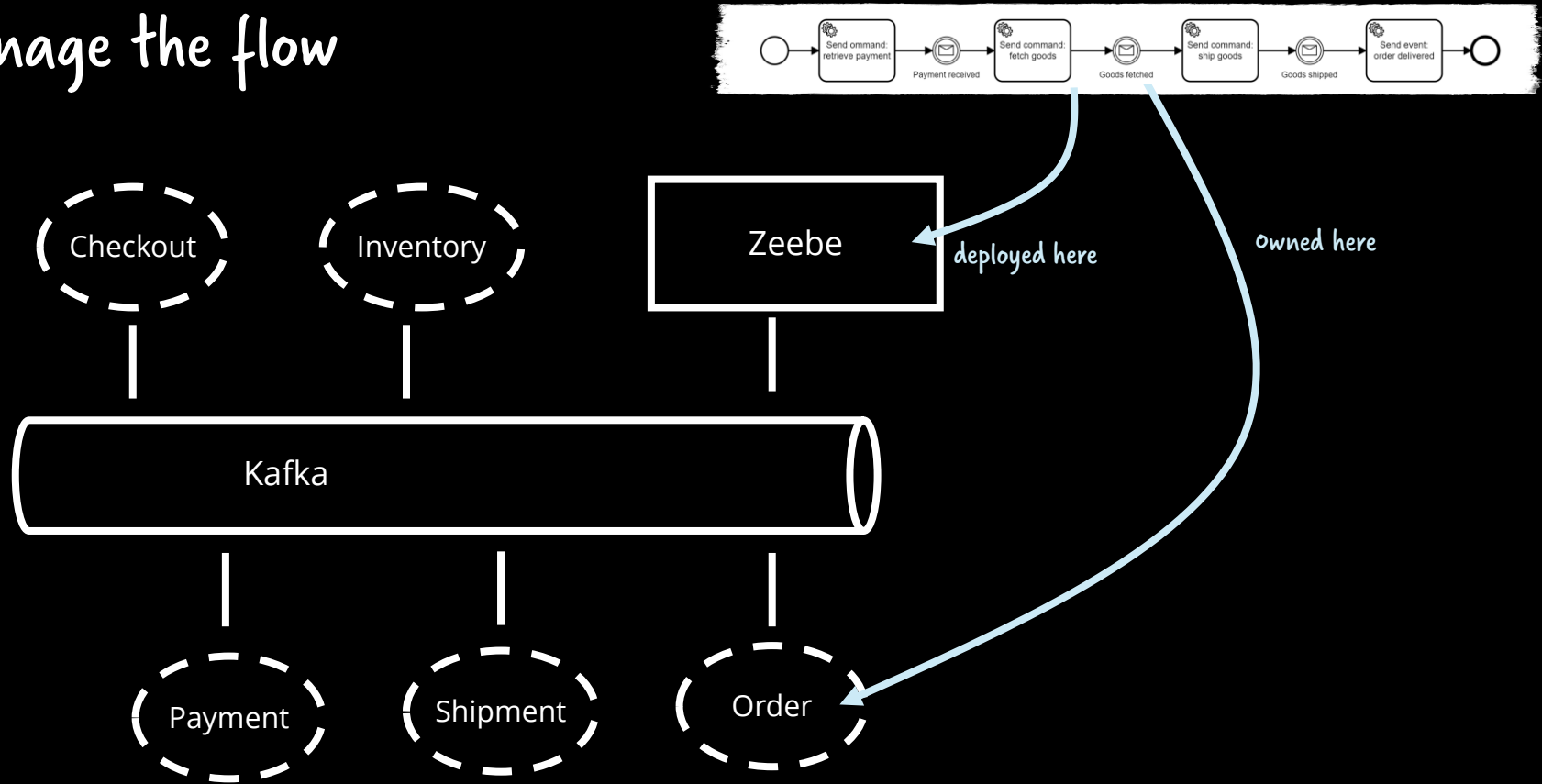




# Now it is easy to change the flow



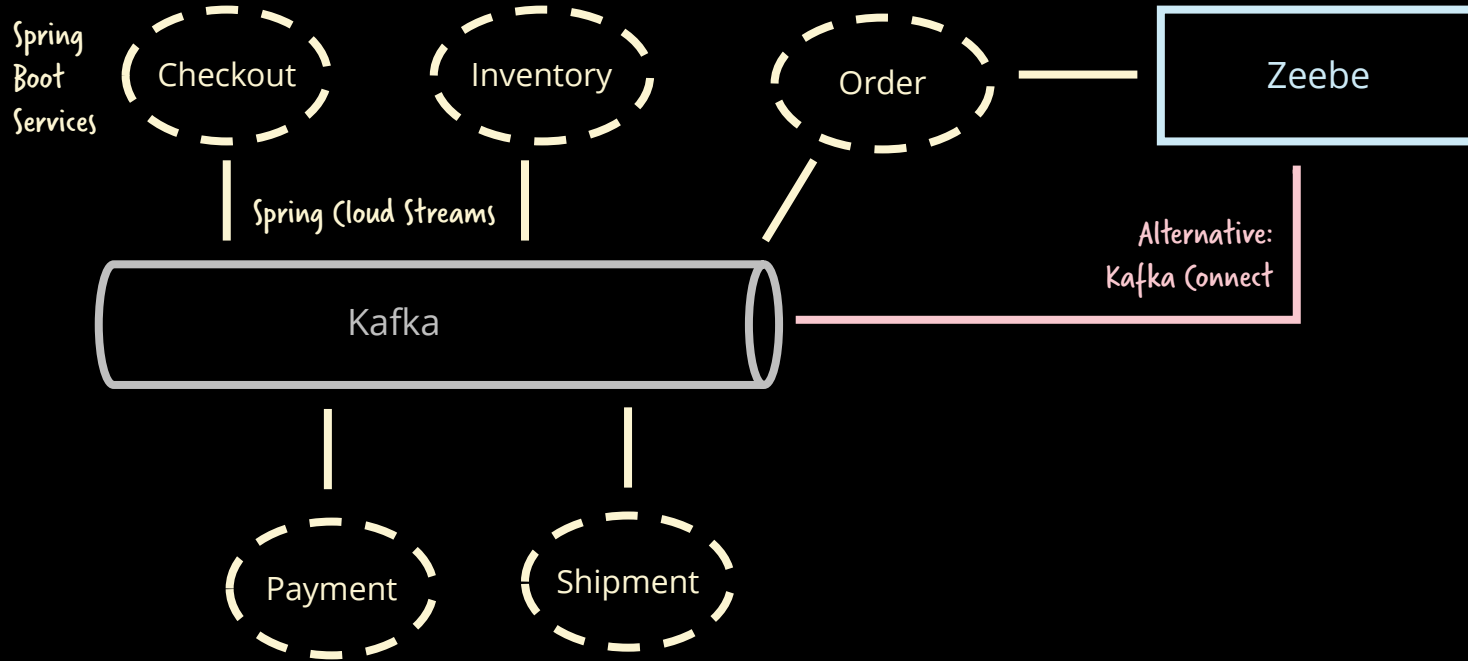
# Manage the flow



Live hacking



# Demo architecture



# BPMN

Business Process  
Model and Notation

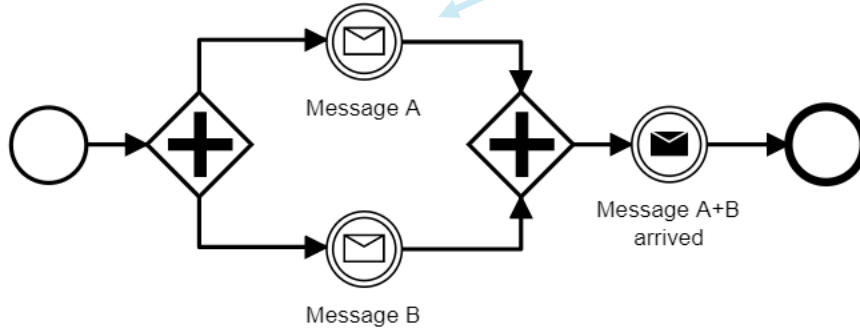
ISO Standard



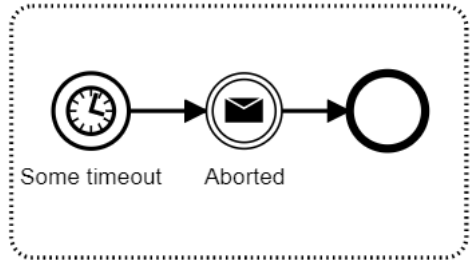
# BPMN supports integration use cases

Zeebe supports Message TTL

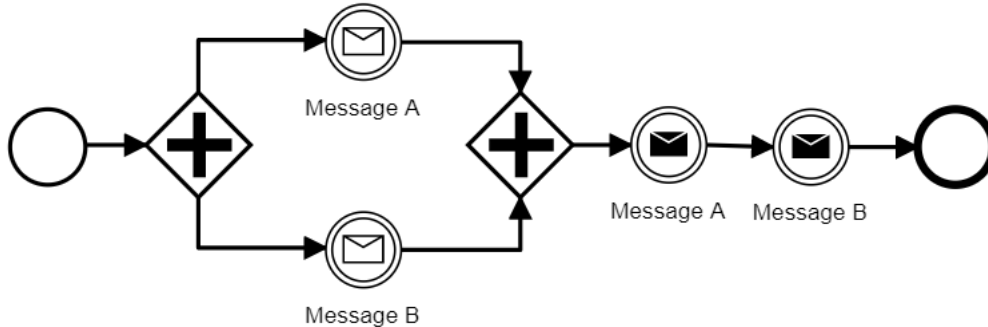
Aggregator



Combined with timeouts:

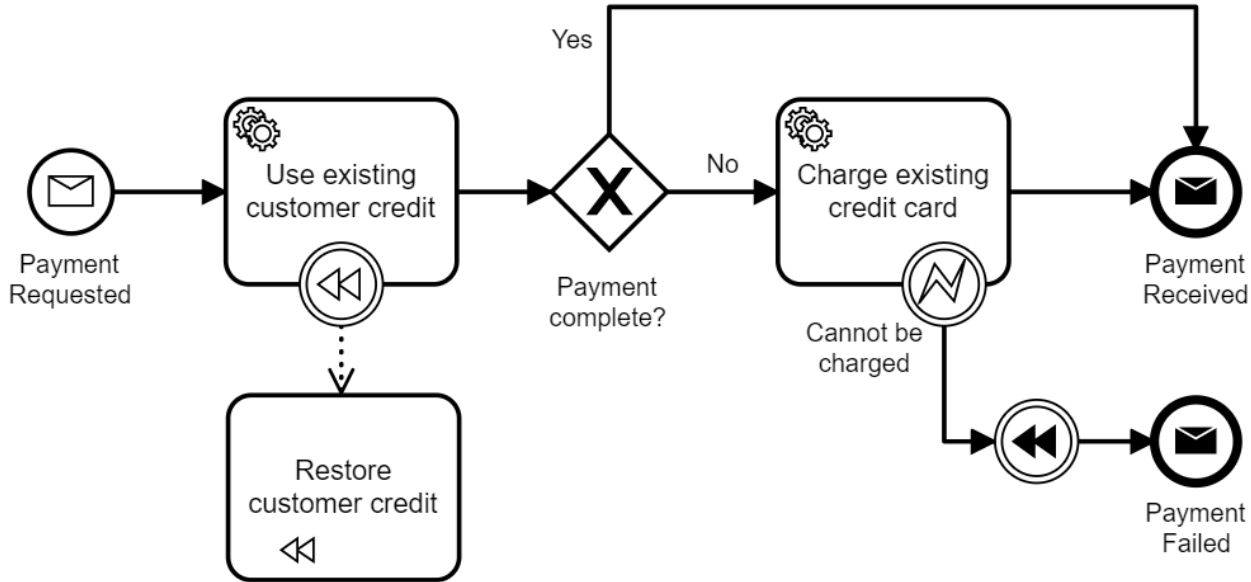


Re-Sequencer



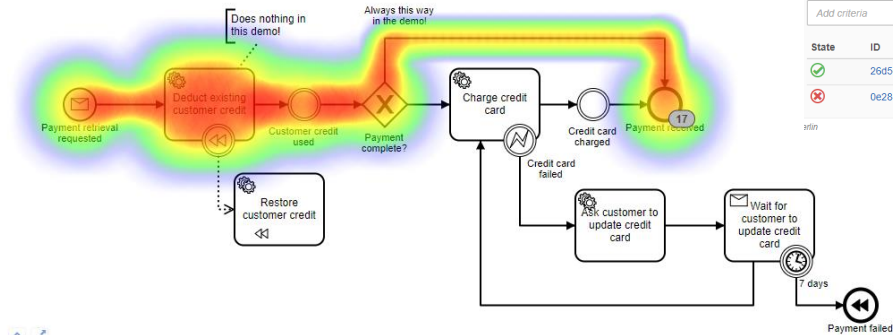
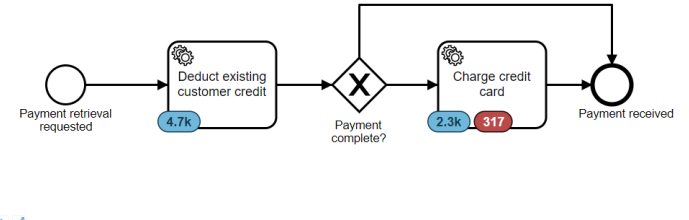
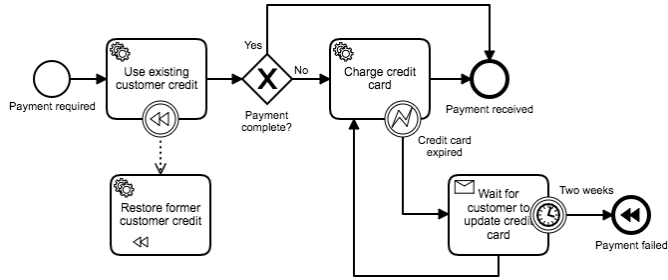
# BPMN supports integration use cases

## Compensation





# Biz Dev Ops



Process Instances Incidents Called Process Definitions Job Definitions Modify

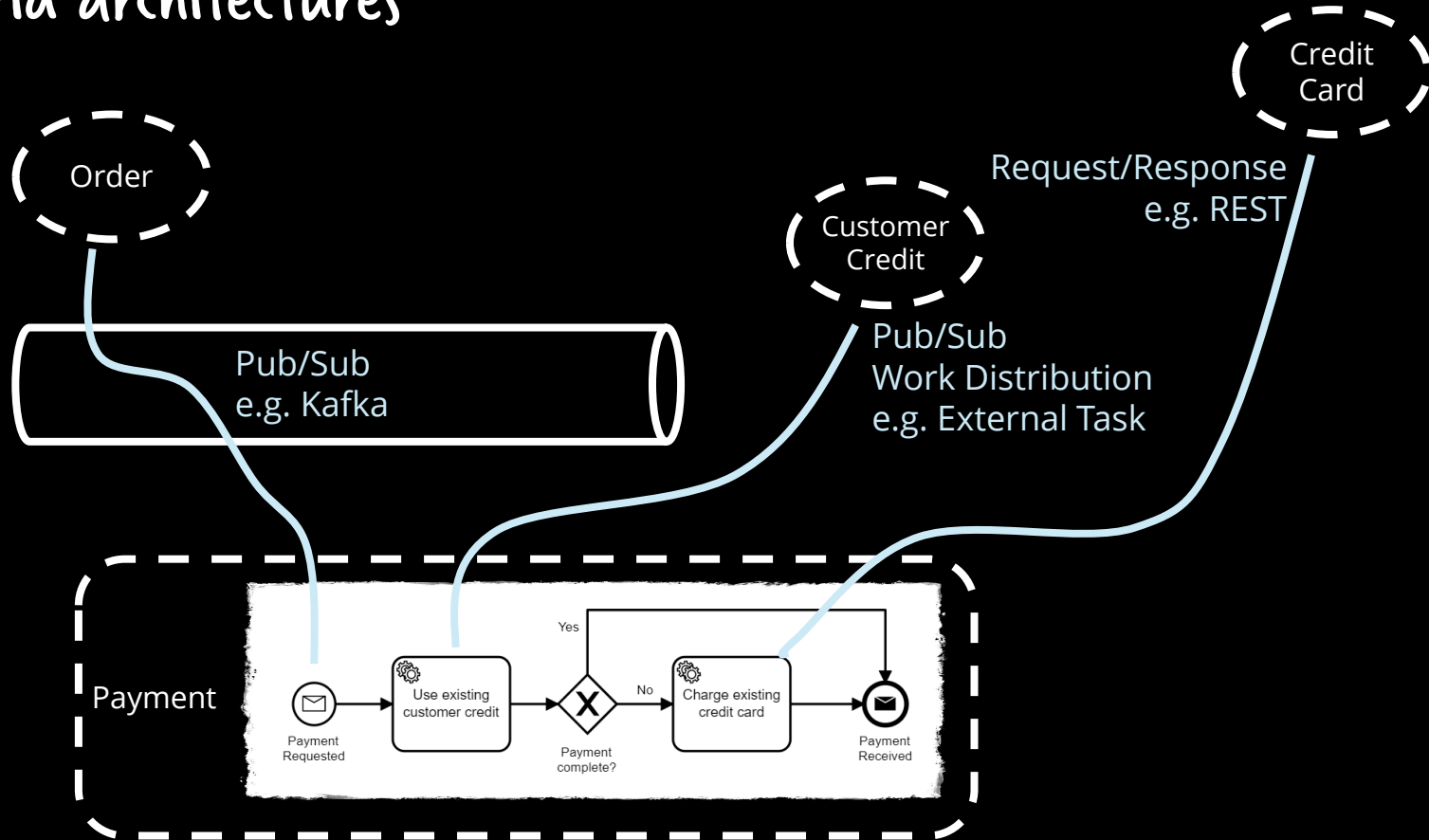
Add criteria 2

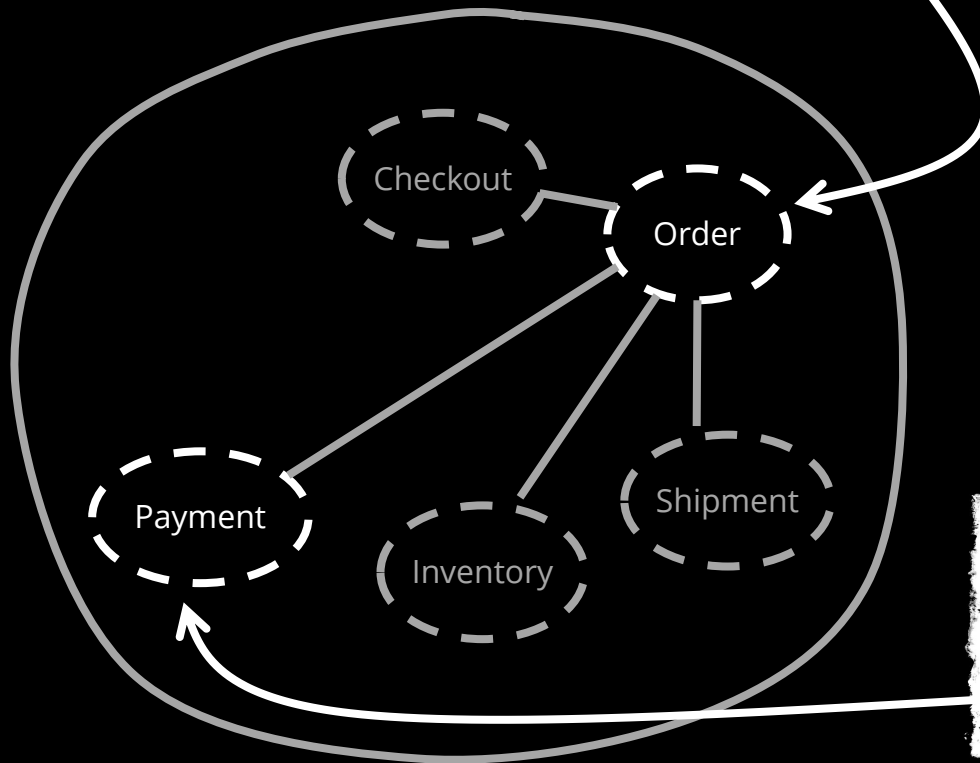
State	ID	Start Time	Business Key
✓	26d5ef21-1ad9-11e8-8882-e4a7a0132825	2018-02-26T10:40:59	
✗	0e28564e-1ad9-11e8-8882-e4a7a0132825	2018-02-26T10:40:18	

atm

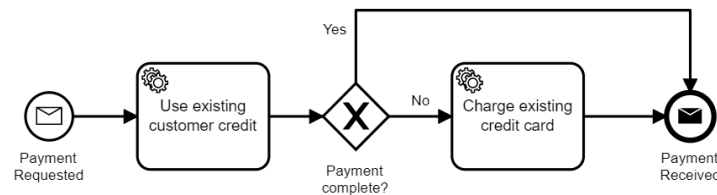
Powered by camunda BPM /

# Hybrid architectures

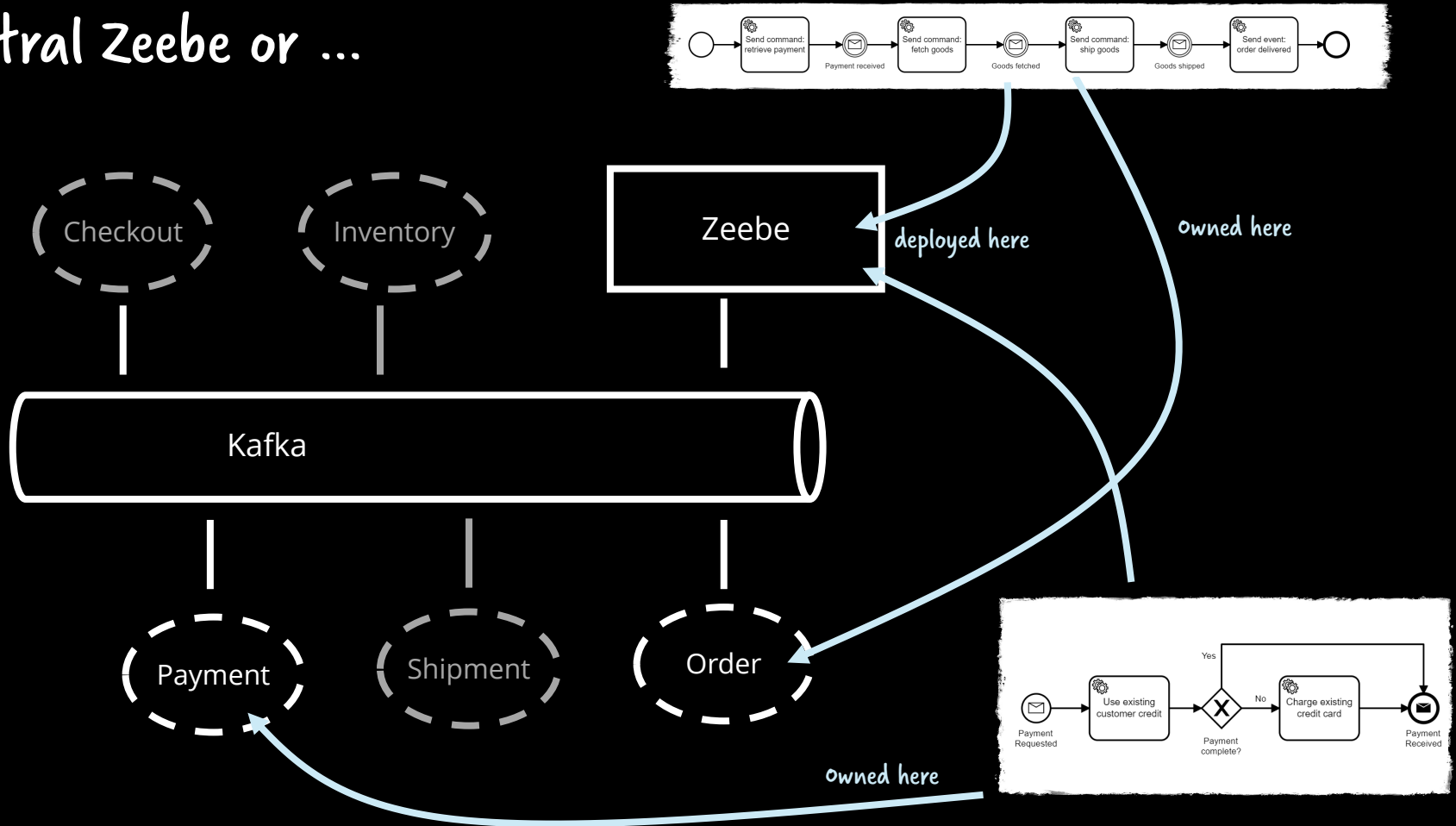




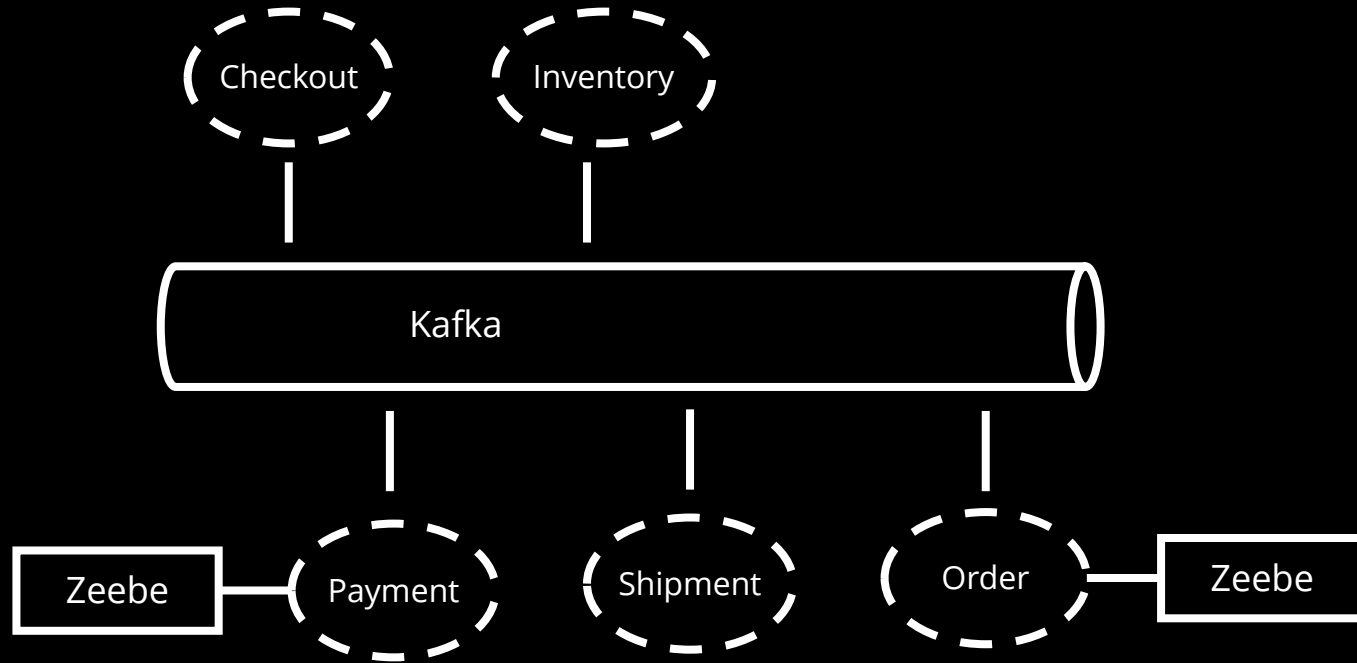
*Every service owns its workflow!*



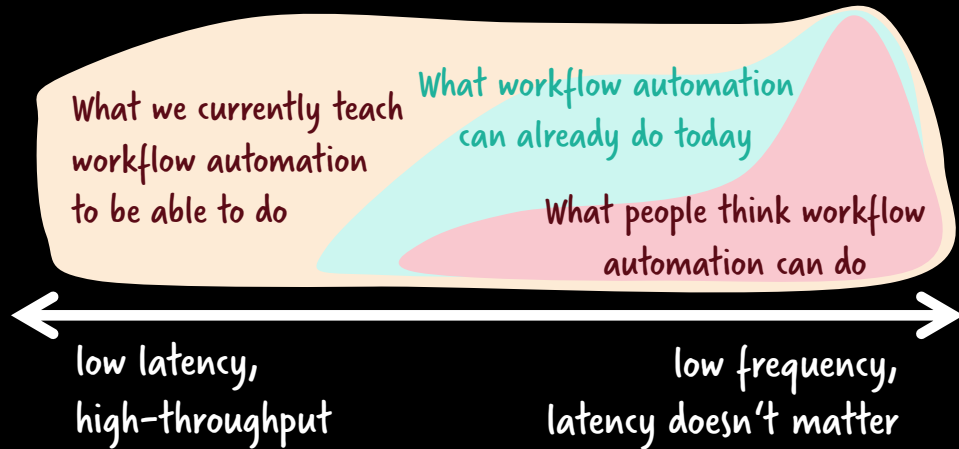
# Central Zeebe or ...



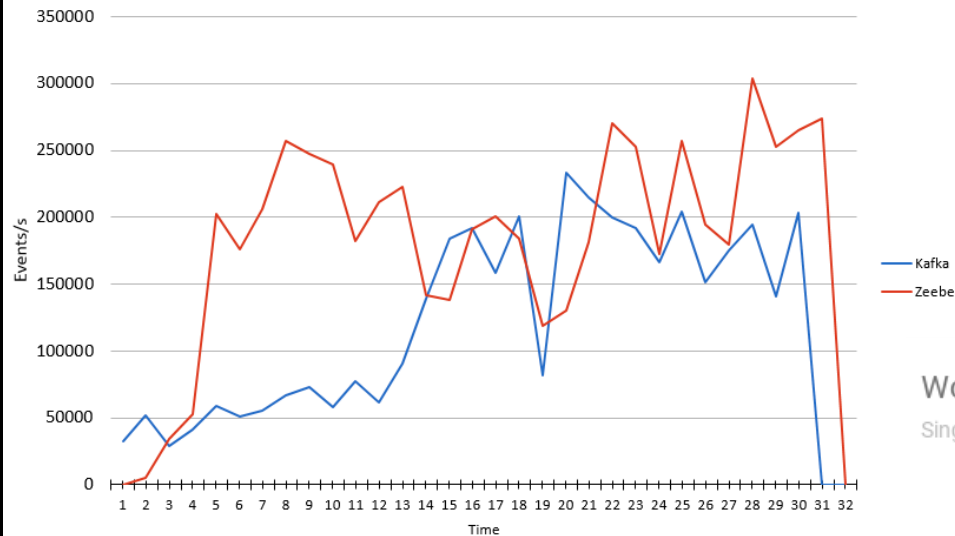
...decentral Zeebe



# Workflow automation at scale!

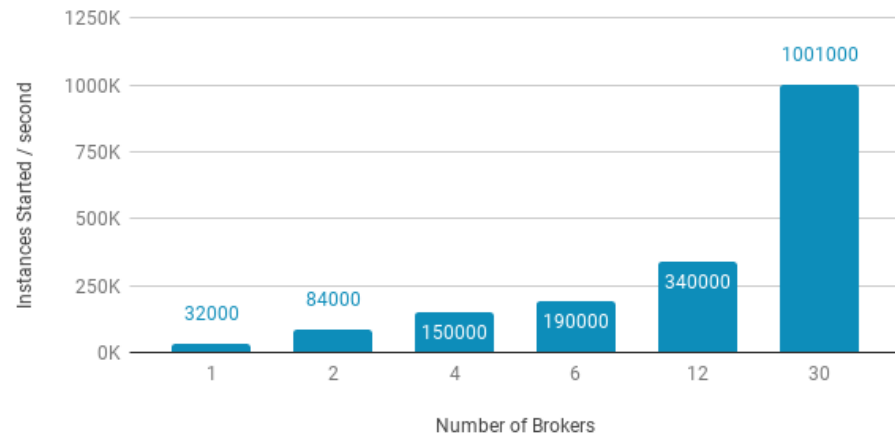


Events written/s Apache Kafka vs. Zeebe

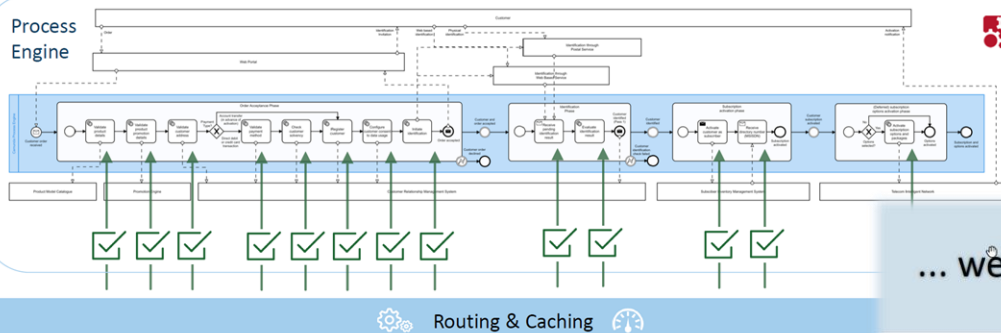


Workflow Instances Started / Second

Single topic, replication factor = 1

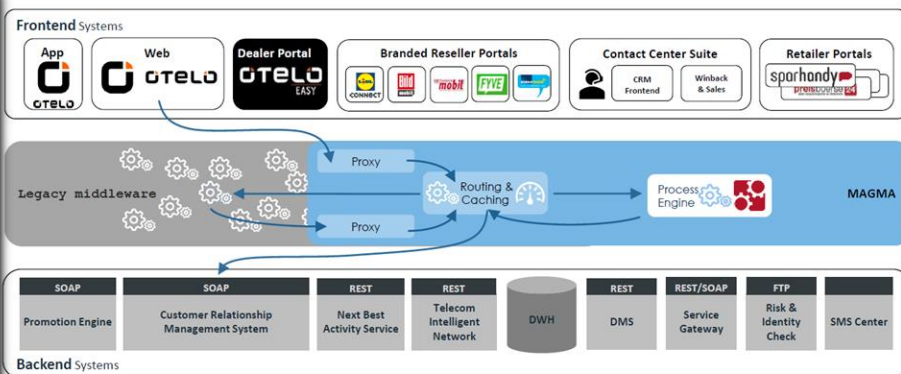


## Using our monitoring abilities for the migration ...



... we could activate the new process step by step

Vodafone, Liongate & WDW  
Presented at CamundaCon Berlin 2018





# Recap

- In event-driven-systems it is easy to loose sight of the flow
- Visibility is essential to survive
- You need to balance orchestration and chorography
- Distributed systems need stateful orchestration
- Track, monitor or manage the flow – it might be a journey
- Workflow automation is an essential building block
- BPMN is very powerful and provides BizDevops

Thank you!



Contact: [mail@berndruecker.io](mailto:mail@berndruecker.io)  
[@berndruecker](https://berndruecker.io)

Slides: <https://berndruecker.io>

Blog: <https://medium.com/berndruecker>

Code: <https://github.com/berndruecker>

**InfoWorld**  
FROM IDG

<https://www.infoworld.com/article/3254777/application-development/3-common-pitfalls-of-microservices-integration-and-how-to-avoid-them.html>

**InfoQ**  
ueue

<https://www.infoq.com/articles/events-workflow-automation>

**THE NEW STACK**

<https://thenewstack.io/5-workflow-automation-use-cases-you-might-not-have-considered/>

