Apache Kafka Meets Workflow Engines

Bernd Ruecker

Co-Founder and Chief Technologist

@Camunda

@berndruecker

Überweisungslimit ändern

1 Ihre Angaben 2 Daten prüfen und senden 3 Bestätigung

Legen Sie fest, welchen Maximalbetrag Sie pro Kalendertag überweisen können. So hoch, wie für Sie persönlich nötig. Sie wollen Ihr Überweisungslimit nur für einen Tag ändern? Dann setzen Sie den Haken bei "temporäre Änderung".

Kontoinhaber	Überweisungslimit	temporäre Änderung
Bernd Rücker	5.000 EUR	☐
[illegible]	5.000 EUR	☐

Hinweis

Änderungen des Überweisungslimits erfolgen am nächsten Werktag.

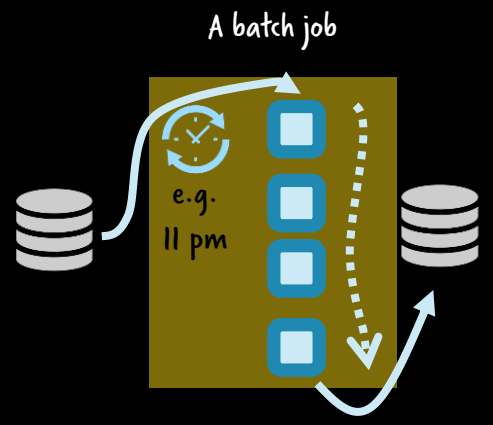
Haben Sie den Haken bei "temporäre Änderung" gesetzt, wird Ihr Überweisungslimit am nächsten Werktag wieder auf den aktuellen Wert zurückgeändert.

Weiter ›


Batch
Processing

Batch jobs

Data at rest

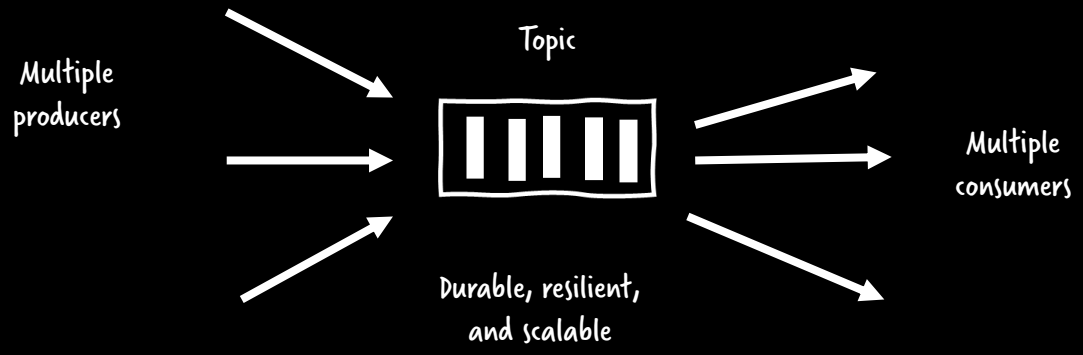


Meet Apache Kafka

Data in motion

Near real-time processing

Apache Kafka



Show me code!

https://github.com/berndruecker/camunda-8-kafka-money-transfer-limit

Überweisungslimit ändern

1 Ihre Angaben 2 Daten prüfen und senden 3 Bestätigung

Legen Sie fest, welchen Maximalbetrag Sie pro Kalendertag überweisen können. So hoch, wie für Sie persönlich nötig. Sie wollen Ihr Überweisungslimit nur für einen Tag ändern? Dann setzen Sie den Haken bei "temporäre Änderung".

Kontoinhaber	Überweisungslimit	temporäre Änderung
Bernd Rücker	5.000 EUR	☐
[redacted]	5.000 EUR	☐

Hinweis

Änderungen des Überweisungslimits erfolgen am nächsten Werktag.

Haben Sie den Haken bei "temporäre Änderung" gesetzt, wird Ihr Überweisungslimit am nächsten Werktag wieder auf den aktuellen Wert zurückgeändert.

Weiter ›

Überweisungslimit ändern

1 Ihre Angaben 2 Daten prüfen und senden 3 Bestätigung

Legen Sie fest, welchen Maximalbetrag Sie pro Kalendertag überweisen können. So hoch, wie für Sie persönlich nötig. Sie wollen Ihr Überweisungslimit nur für einen Tag ändern? Dann setzen Sie den Haken bei "temporäre Änderung".

Kontoinhaber	Überweisungslimit	temporäre Änderung
Bernd Rücker	5.000 EUR	☐
[illegible]	5.000 EUR	☐

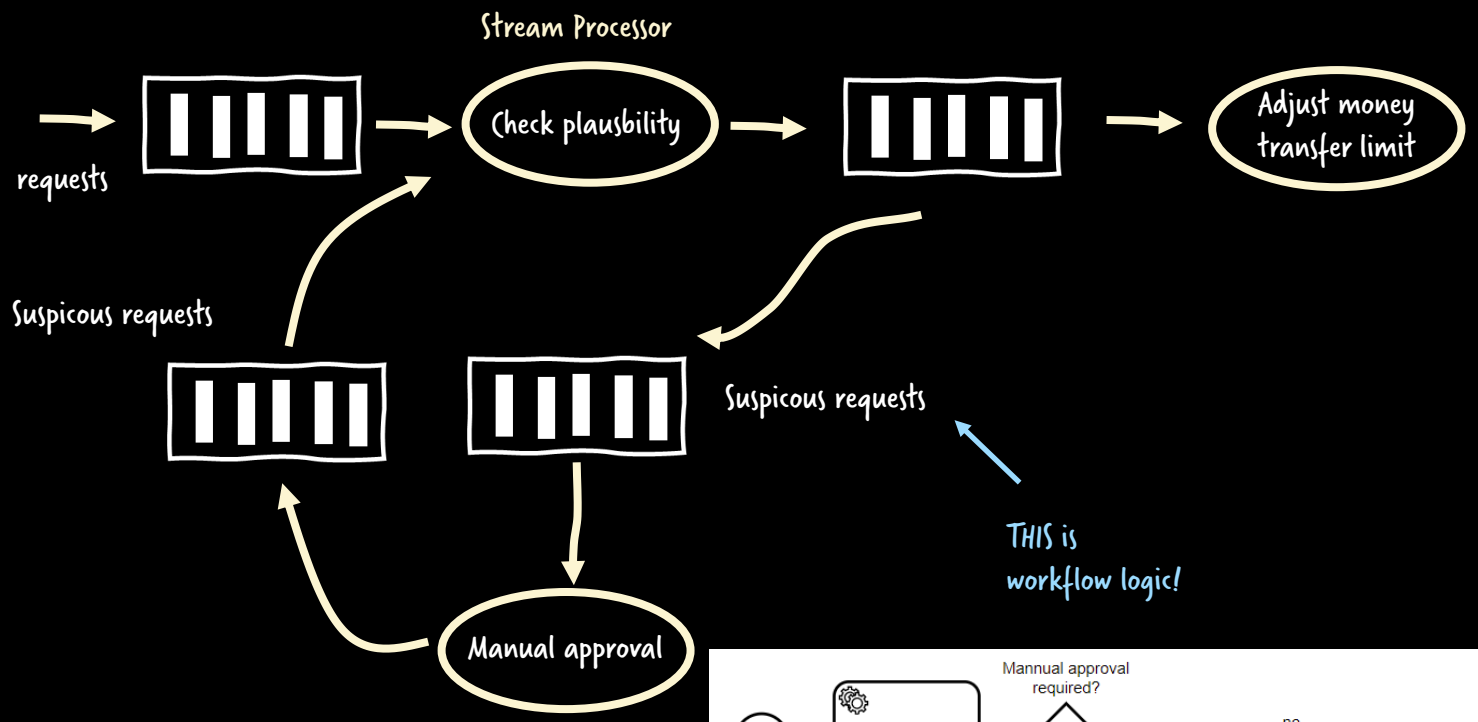
Hinweis

Änderungen des Überweisungslimits werden an Werktagen zwischen 6 Uhr und 17:45 Uhr innerhalb von 60 Minuten bearbeitet. Änderungen nach 17:45 Uhr erfolgen am nächsten Werktag.

Haben Sie den Haken bei "temporäre Änderung" gesetzt, wird Ihr Überweisungslimit am nächsten Werktag wieder auf den aktuellen Wert zurückgeändert.

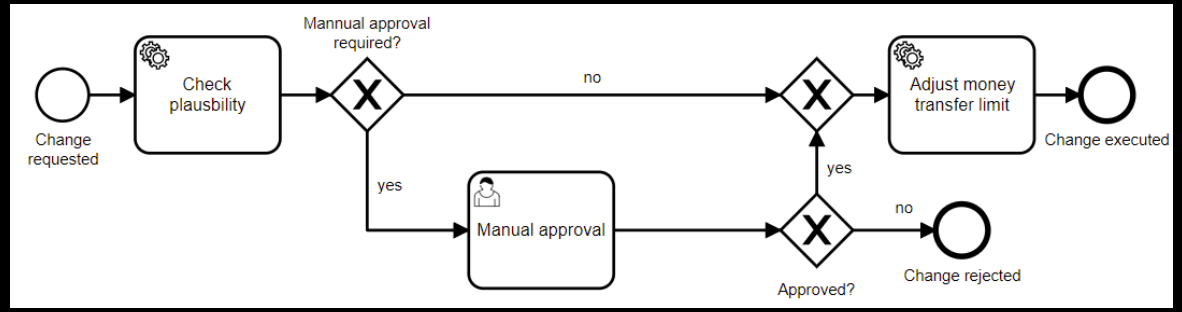
Weiter ›

Further – possibly manual – steps...



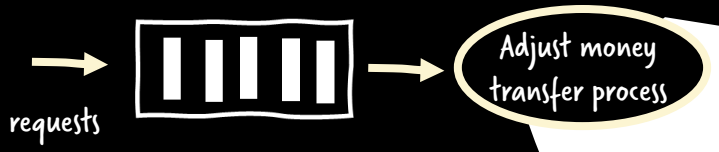
THIS is workflow logic!

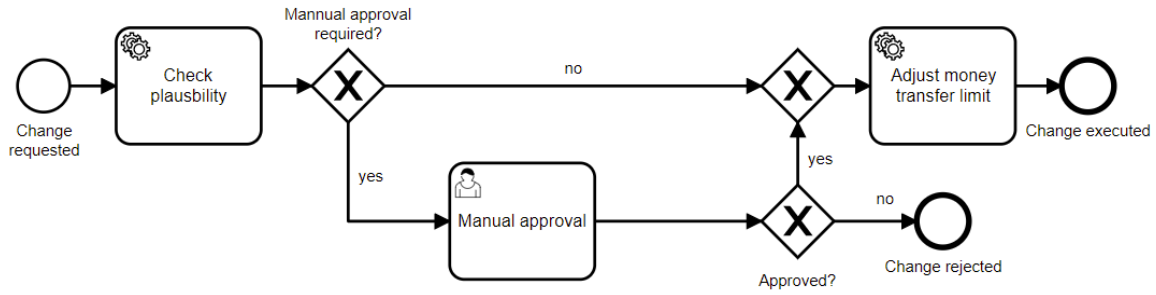
Workflow engines can hold state



Adding workflow logic to a stream processor

of course you can call this also a microservice

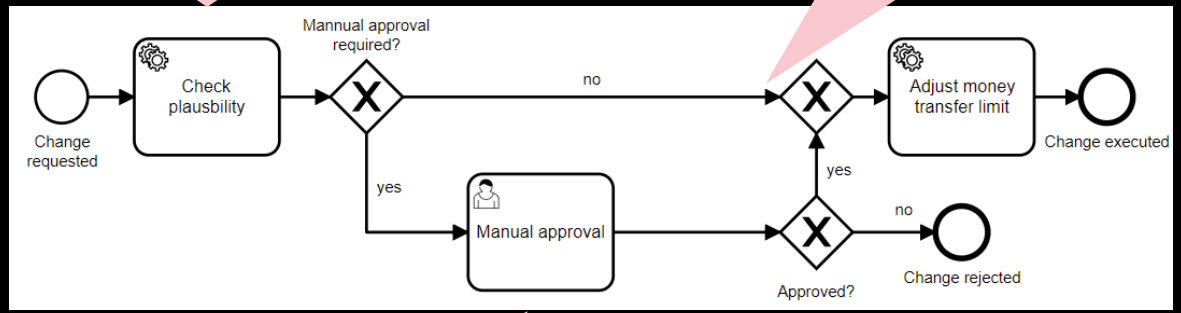




A workflow aka process

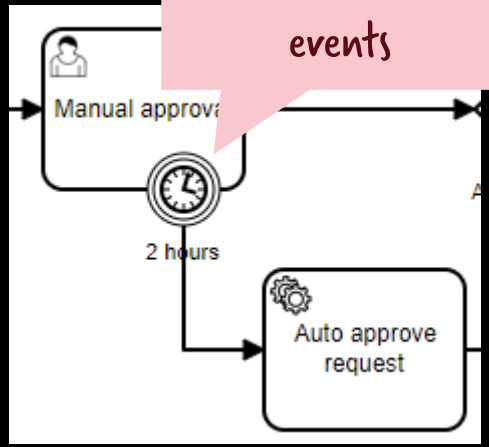
Steps

In a certain sequence

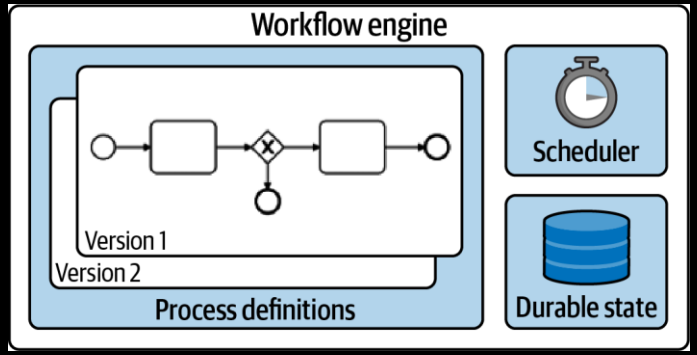


Which might be long running

Time-controlled events



A workflow engine



Process definition table

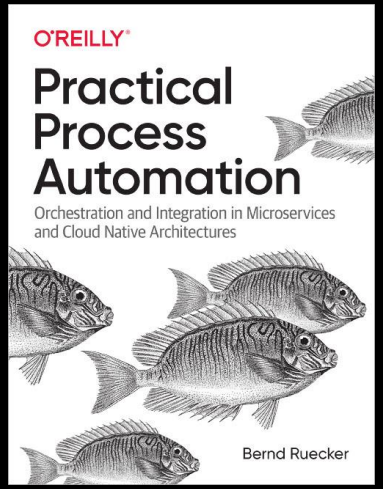
ID	Process definition	Version	Process model
42	Order	1	<bpmn ...
43	Order	2	<bpmn ...
87	Onboarding	1	<bpmn ...

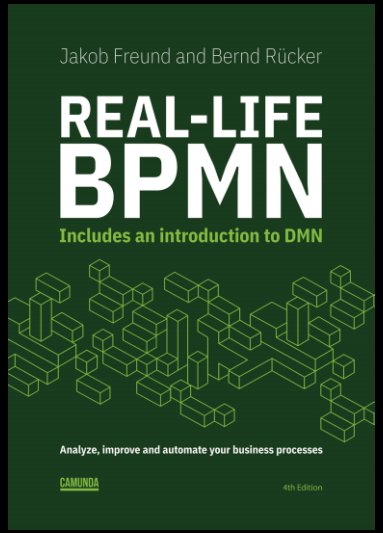
Process instance table

Instance ID	Process definition	Current state	...
...7454	43	Wait for payment	...
...4571	43	Ship goods	...
...4477	87	Customer check	...

Bernd Ruecker

Co-founder and Chief Technologist of Camunda





mail@berndruecker.io
@berndruecker
http://berndruecker.io/

Show me code!

https://github.com/berndruecker/camunda-8-kafka-money-transfer-limit

What the tools bring to the table

Understanding the process

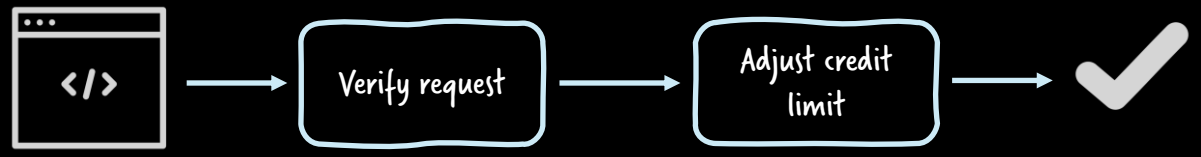
Error Handling

Camunda

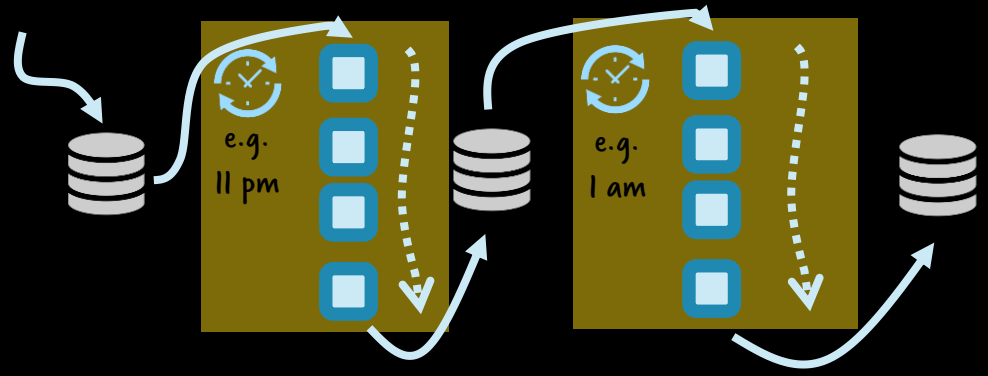
Near real-time processing

Apache Kafka

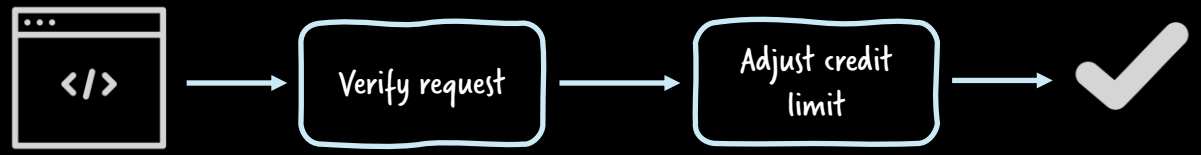
Task vs Process Automation



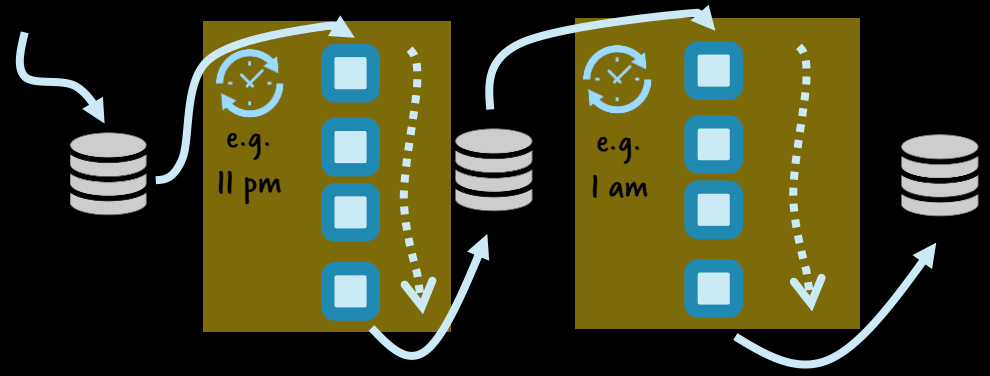
Batch



Task vs Process Automation



Batch

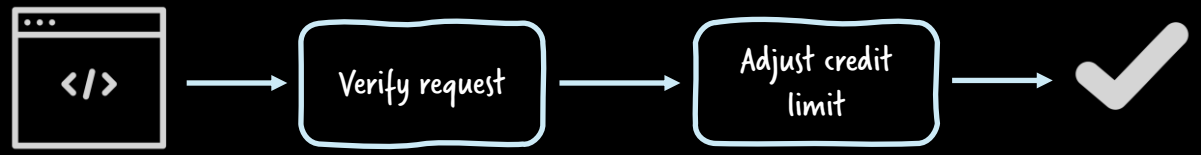


Batches are orthogonal to the real process!

Task Automation

Process Automation

Task vs Process Automation

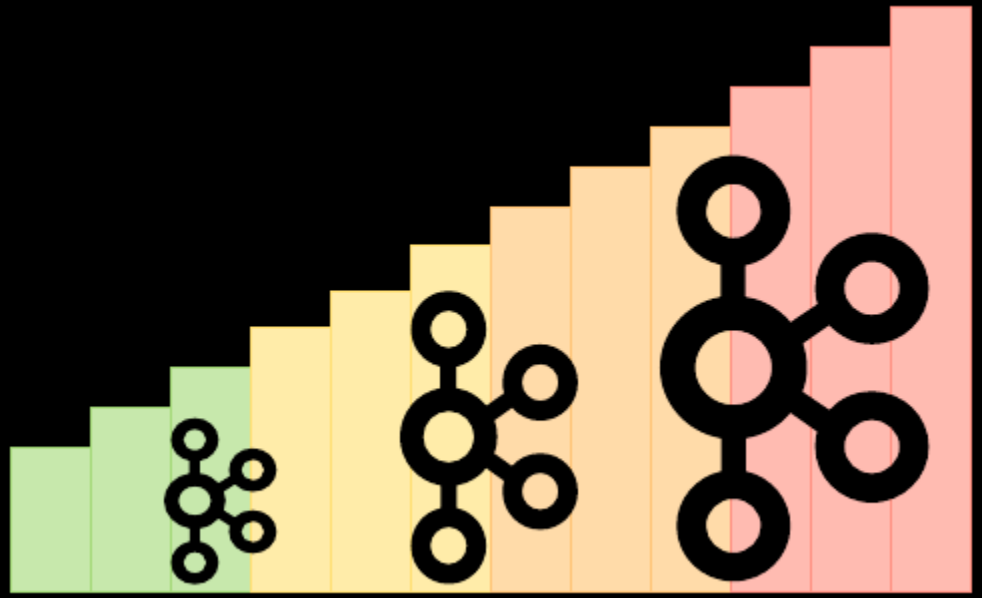


Stream processors are orthogonal to the real process!

Task Automation

Process Automation

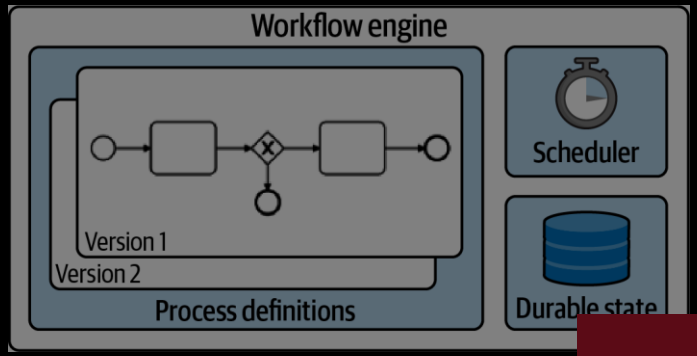
Kafka scales horizontally



Taken from https://mbukowicz.github.io/kafka/2020/06/22/scaling-kafka.html

https://www.flickr.com/photos/dullhunk/28251201308

A workflow engine



CAMUNDA

We went from RDBMS to event sourcing...

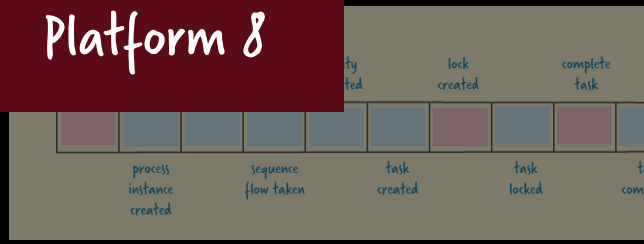
Camunda Platform 7

Camunda Platform 8

Process definition table

ID	Process definition	Version	Process model
42	Order	1	<bpmn ...
43	Order	2	<bpmn ...
87	Onboarding	1	<bpmn ...

...7454	43	Wait for payment	...
...4571	43	Ship goods	...
...4477	87	Customer check	...



https://blog.bernd-ruecker.com/zeebe-io-a-horizontally-scalable-distributed-workflow-engine-45788a90d549

“Zeebe” is the workflow engine offered within Camunda 8

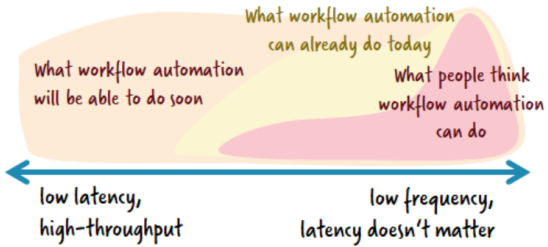
berndruecker

Zeebe.io — a horizontally scalable distributed workflow engine

Say hello to cloud-native workflow automation — part 1

Bernd Rücker
Jul 17, 2019 · 10 min read

There are many use cases for workflow automation out there. Many people think that workflow automation is only used for slow and low frequency use cases like human task management. Despite the fact that this is not true (see e.g. 24 Hour Fitness or Zalando) I do see limitations of current workflow technology in terms of scalability, but on a very different order of magnitude. As traditional engines are based on relational databases they are naturally limited in scale to what that database can handle. Even if this is sufficient for most companies, I know there are definitely interesting use cases requiring more performance and scalability, e.g. to process financial trades which need soft real-time guarantees under a very high load.



Workflow engines at scale

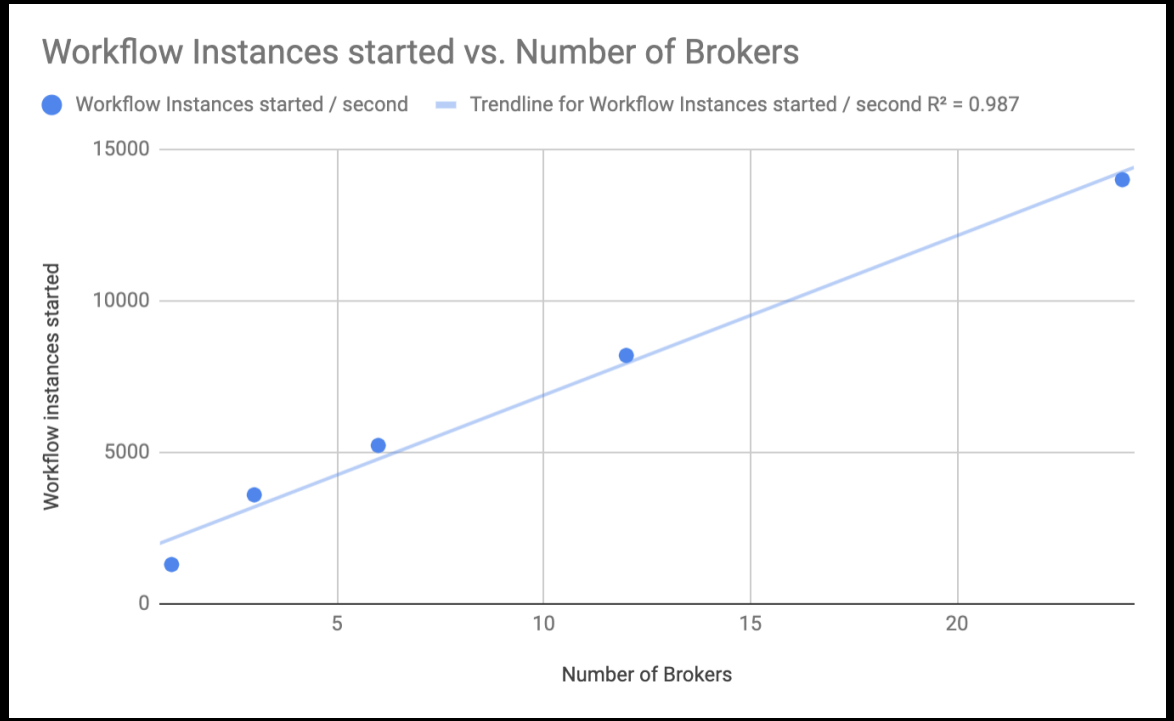
CAMUNDA CLOUD, PROCESS AUTOMATION AS A SERVICE, SCALABILITY

AUGUST 15, 2019

Scaling Zeebe Horizontally: A Simple Benchmark

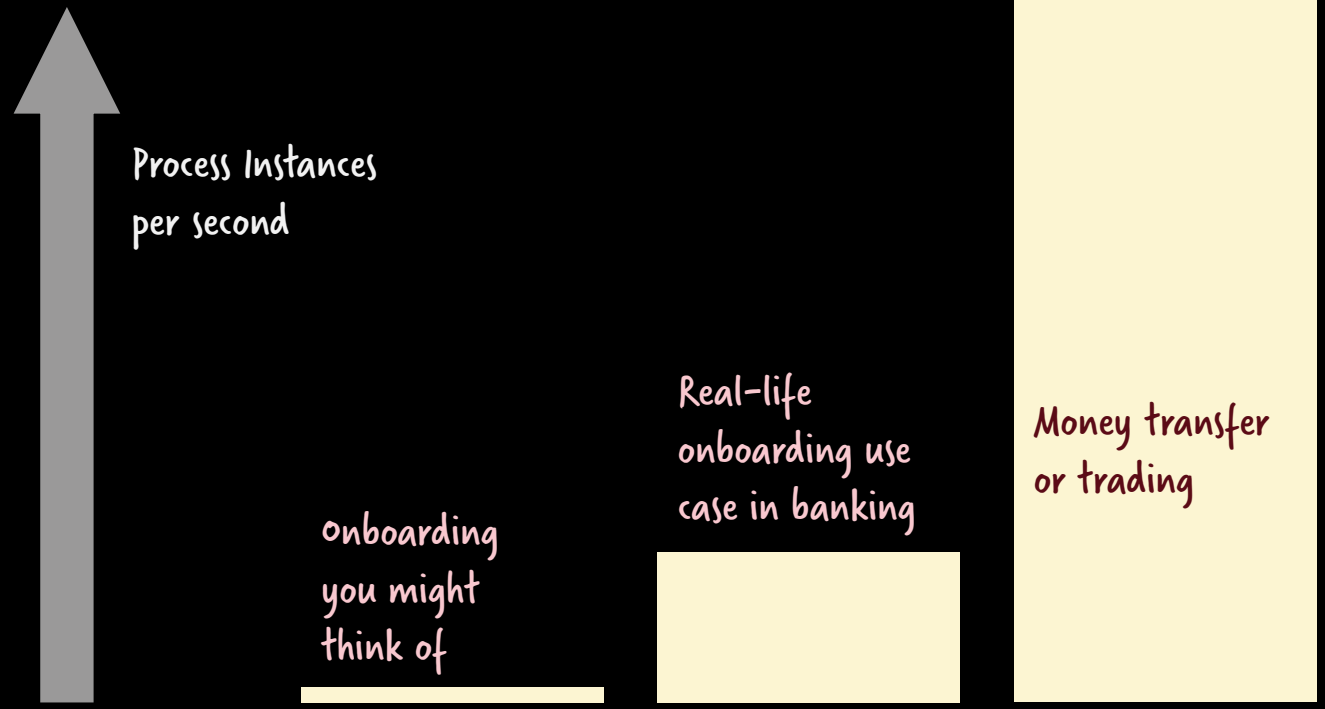
Note: The specific performance metrics in this blog post are from an earlier release of Zeebe. Since this post was published, work has been done to stabilise Zeebe clusters, and this has changed the performance envelope. You can follow the steps in this blog post to test the current release of Zeebe yourself, and derive the current performance envelope. Zeebe advertises itself as being a "horizontally-scalable workflow engine". In this post, we cover what that means and...

Read more

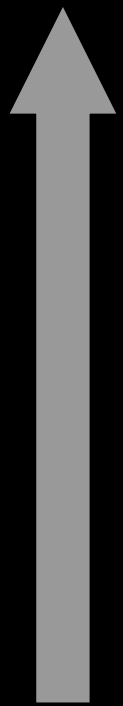


https://camunda.com/blog/2019/08/zeebe-horizontal-scalability/

Scale


Process Instances
per second
Onboarding
you might
think of
Real-life
onboarding use
case in banking
Money transfer
or trading

Scale



Process Instances
per second

Telco
onboarding

Making phone
calls

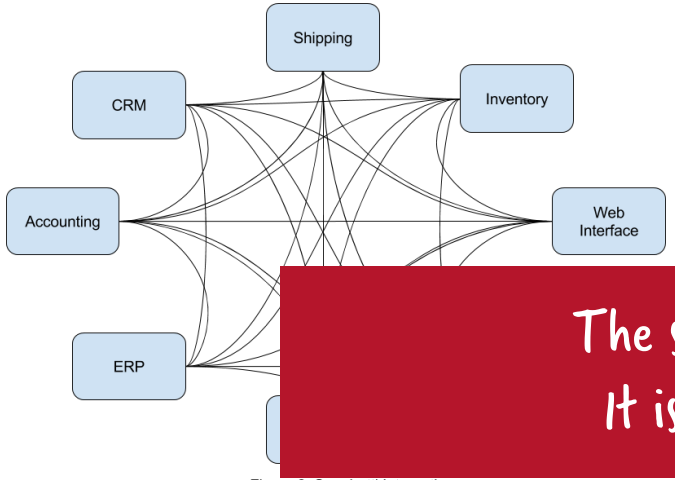
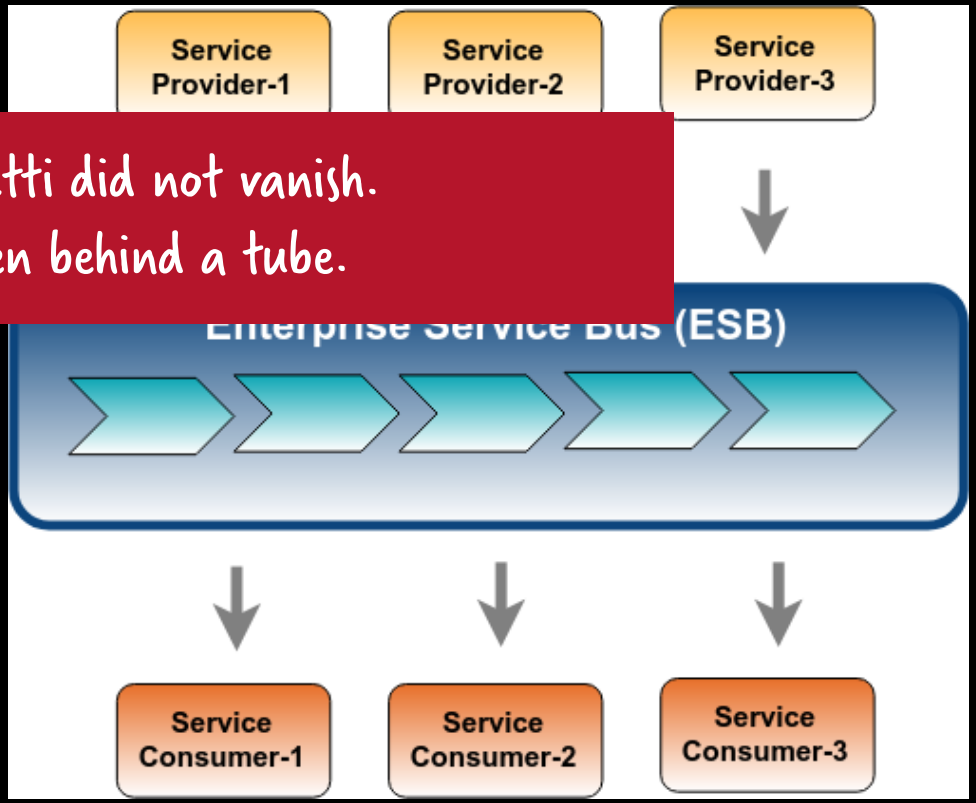


Figure 2: Spaghetti Integration

The spaghetti did not vanish.
It is hidden behind a tube.



Smart endpoints vs dumb pipes

Smart endpoints and dumb pipes

When building communication structures between different processes, we've seen many products and approaches that stress putting significant smarts into the communication mechanism itself. A good example of this is the Enterprise Service Bu (ESB), where ESB products often include sophisticated facilities for message routing, choreography, transformation, and applying business rules.

The microservice community favours an alternative approach: *smart endpoints and dumb pipes*. Applications built from microservices aim to be as decoupled and as cohesive as possible - they own their own domain logic and act more as filters in the classical Unix sense - receiving a request, applying logic as appropriate and producing a response. These are choreographed using simple RESTish protocols rather than complex protocols such as WS-Choreography or BPEL or orchestration by a central tool.

The two protocols used most commonly are HTTP request-response with resource API's and lightweight messaging[8]. The best expression of the first is

> *Be of the web, not behind the web*
>
> -- *Ian Robinson*

Microservice teams use the principles and protocols that the world wide web (and to a large extent, Unix) is built on. Often used resources can be cached with very little effort on the part of developers or operations folk.

https://martinfowler.com/articles/microservices.html#SmartEndpointsAndDumbPipes

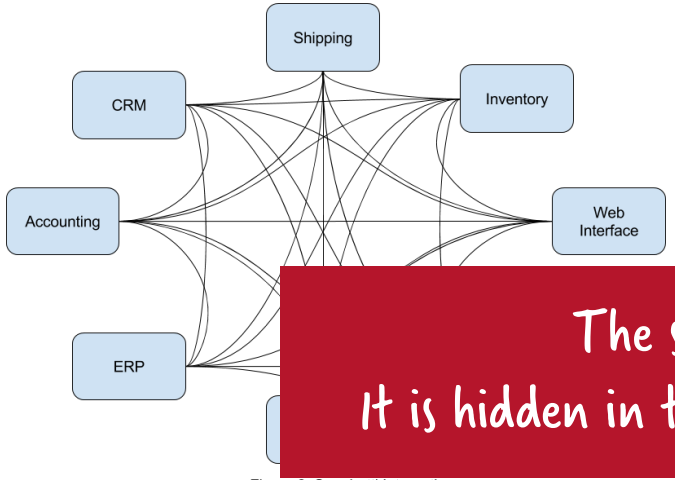
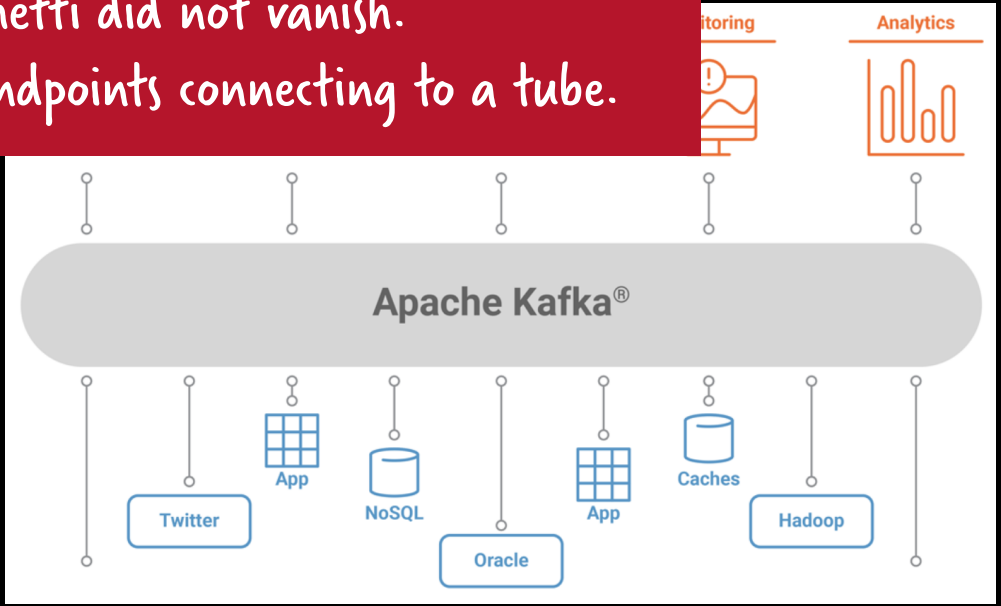
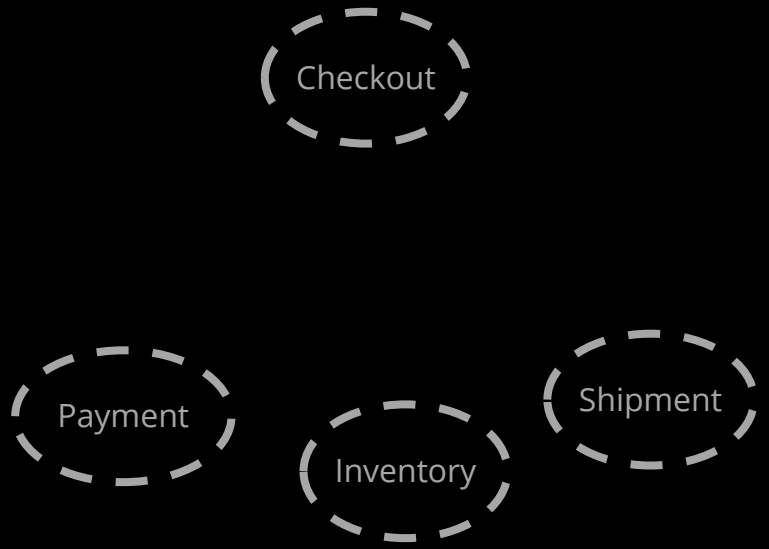


Figure 2: Spaghetti Integration

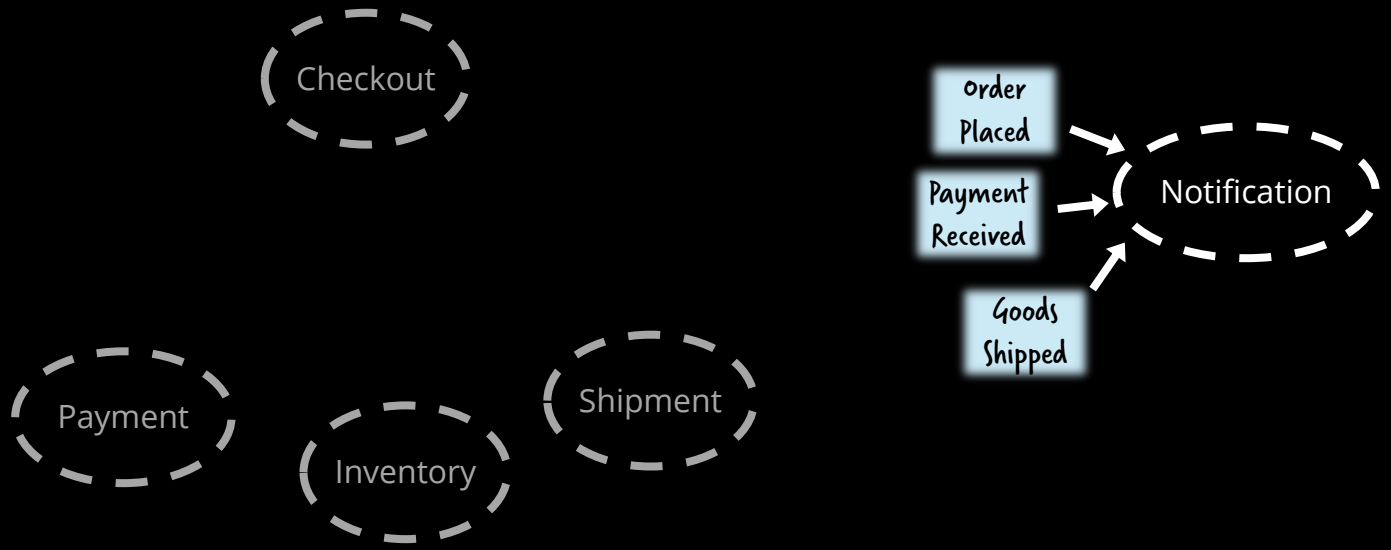
The spaghetti did not vanish.
It is hidden in the endpoints connecting to a tube.



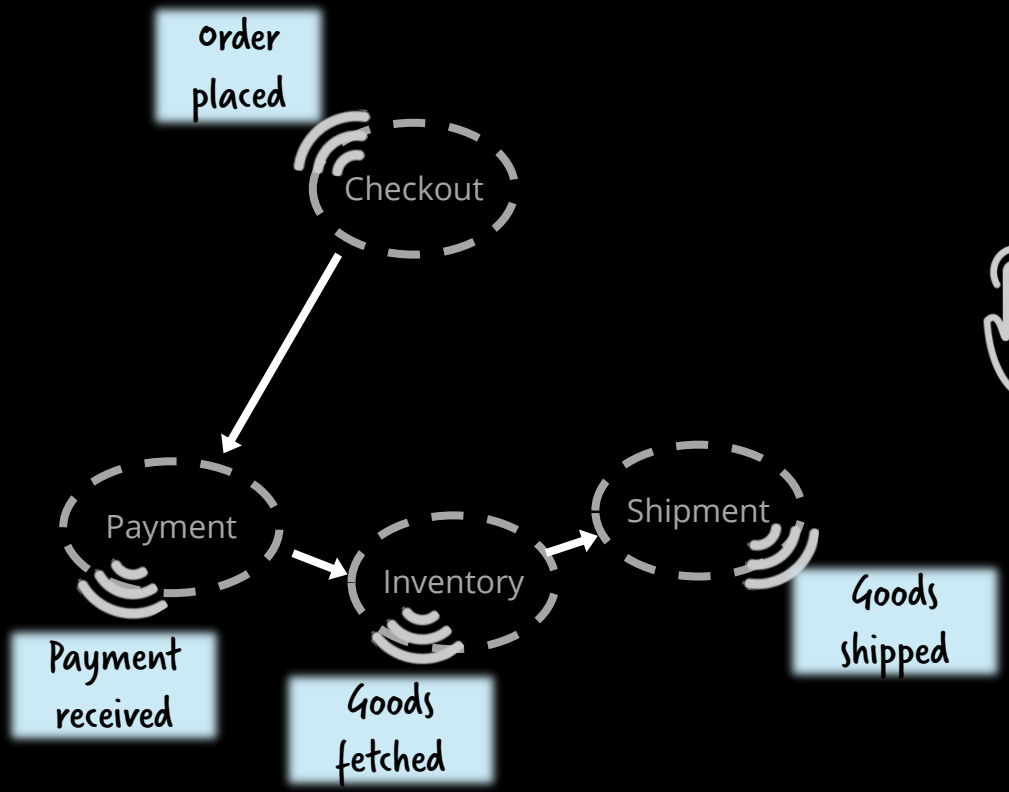
Microservices



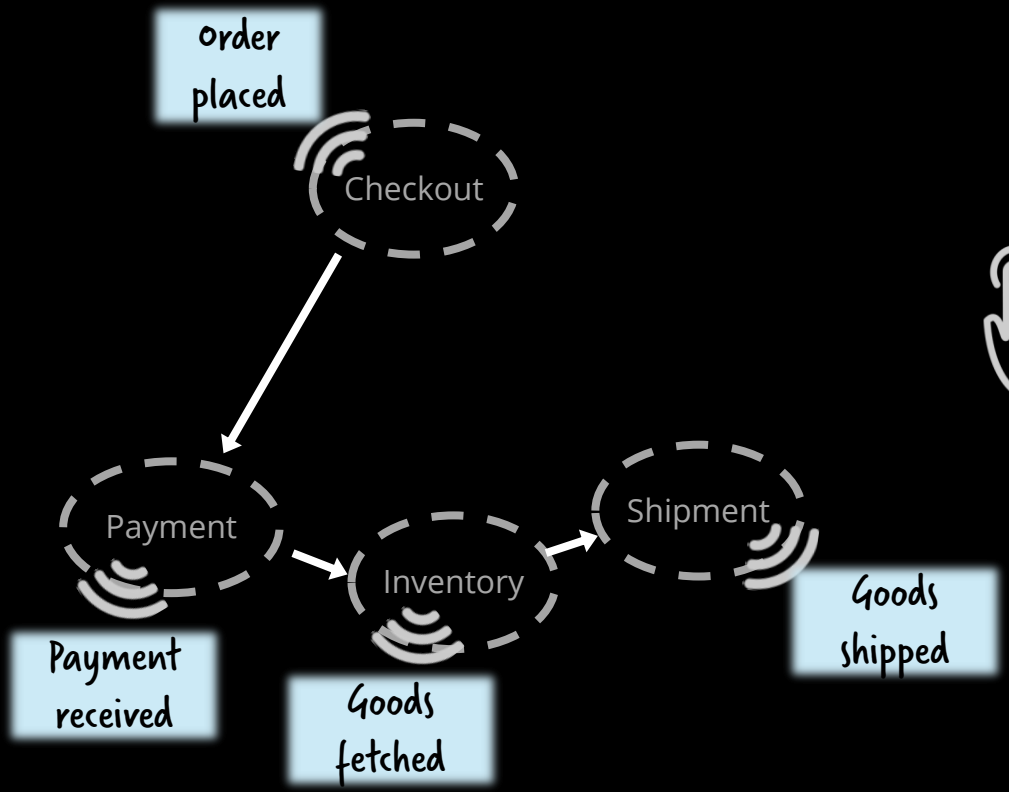
Event-driven & Reactive

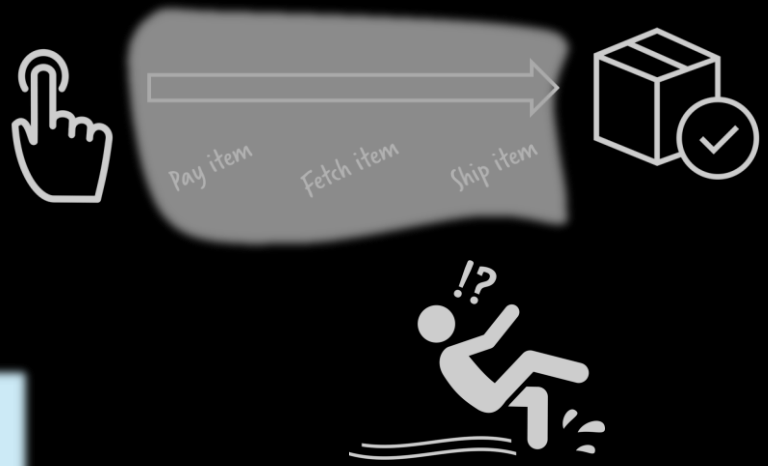


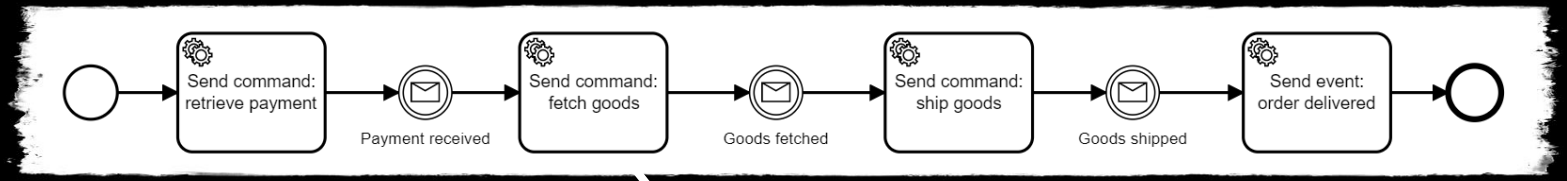
Peer-to-peer event chains

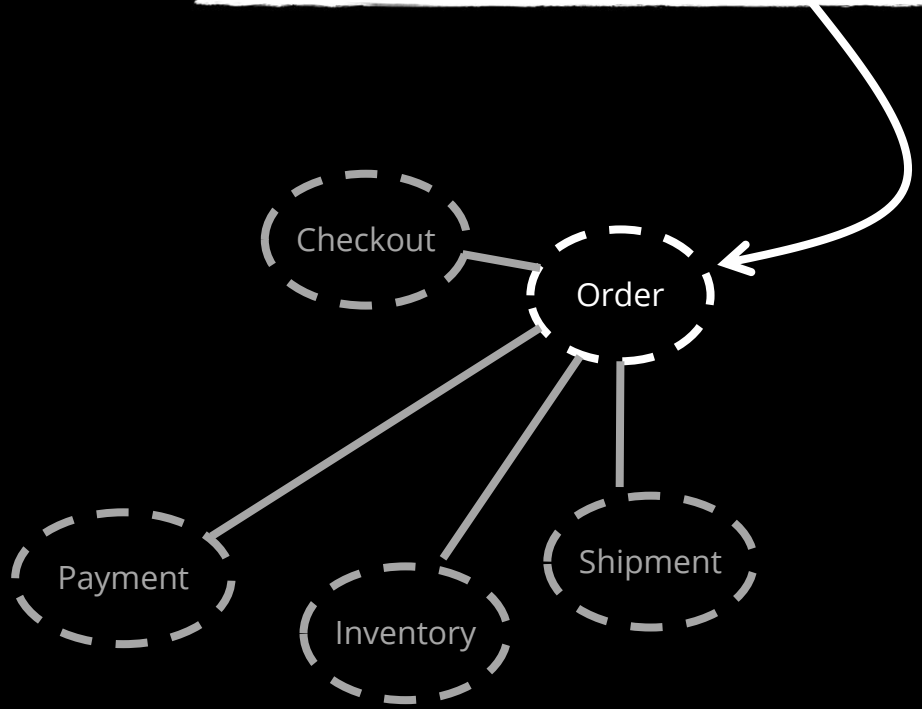


Peer-to-peer event chains









@berndruecker

Some code?

berndruecker / flowing-retail

Unwatch 119 | Unstar 1.1k | Fork 378

Code | Issues 6 | Pull requests 14 | Actions | Projects | Wiki | Security 20 | Insights | Settings

master | flowing-retail / kafka / | Go to file | Add file

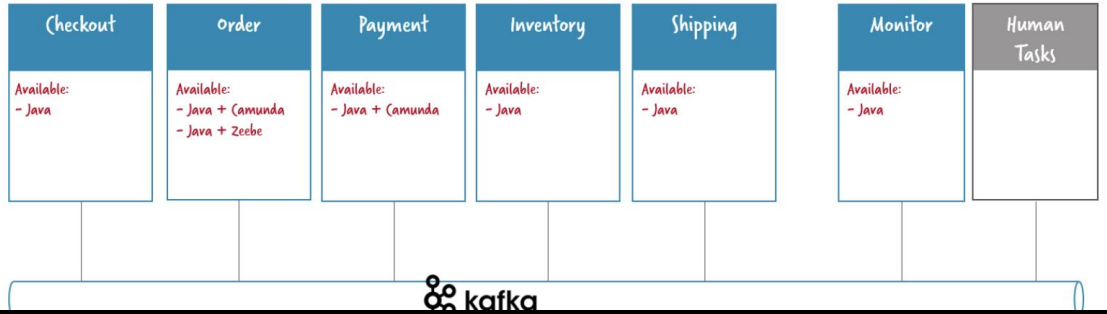
berndruecker Updated to Zeebe 1.0-rc3 — d0f5495 on 11 May — History

java	Updated to Zeebe 1.0-rc3	2 months ago
README.md	adjusted build correctly to Java >= 8	16 months ago

README.md

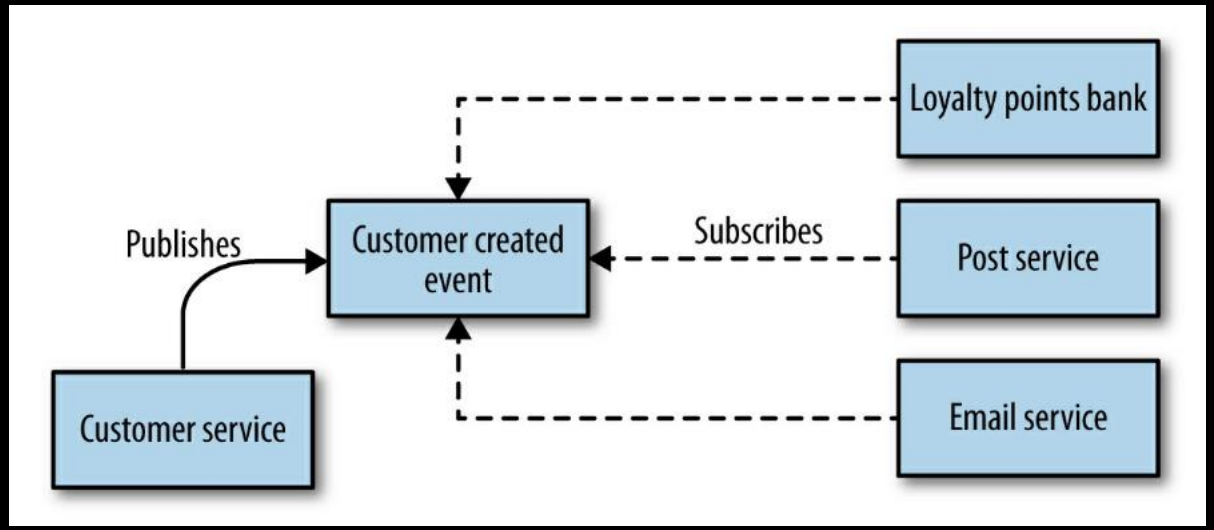
Flowing Retail / Apache Kafka

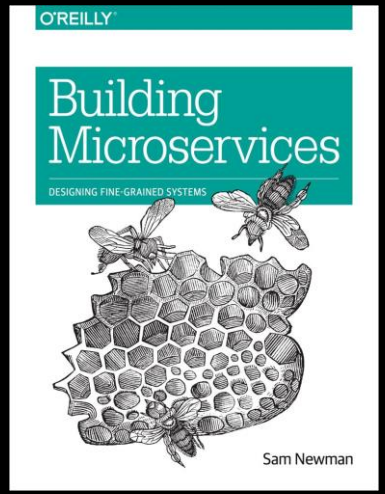
This folder contains services that connect to Apache Kafka as means of communication between the services.



https://github.com/berndruecker/flowing-retail/tree/master/kafka

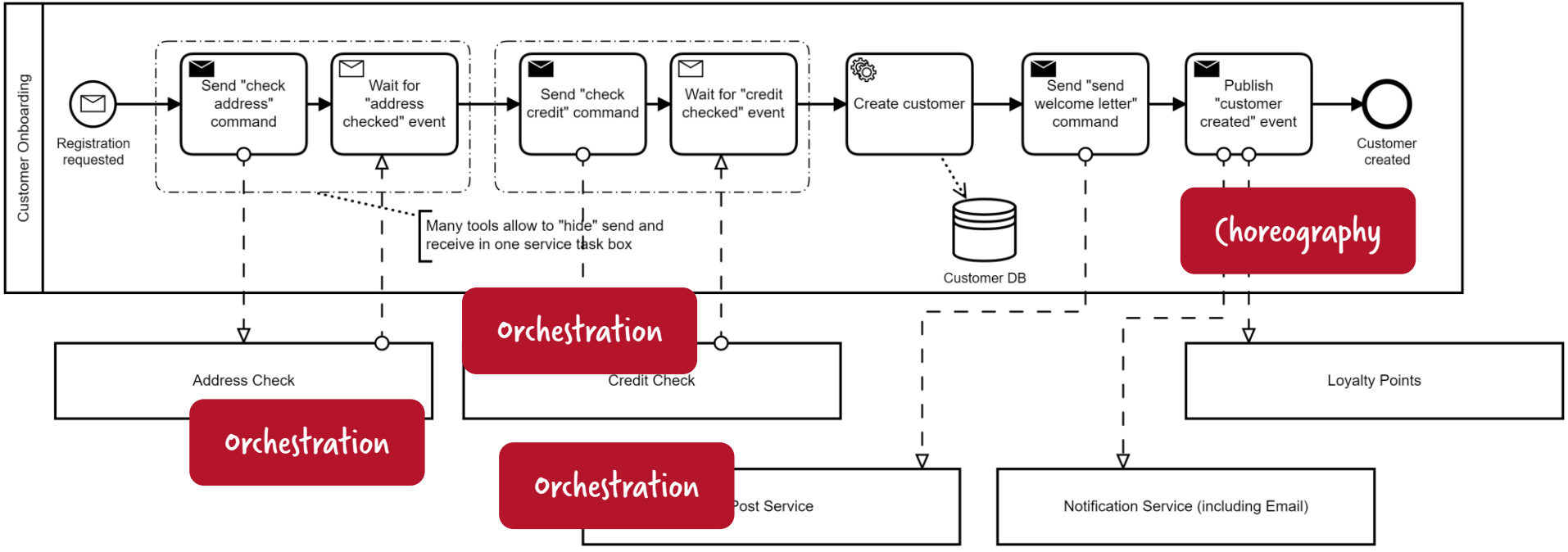
Customer created





Sam Newman: Building Microservices

Mix orchestration and choreography



Loosely or Lousily Coupled?

Understanding
Communication Patterns in
Microservices Architectures

@berndruecker

14:30 – 15:15

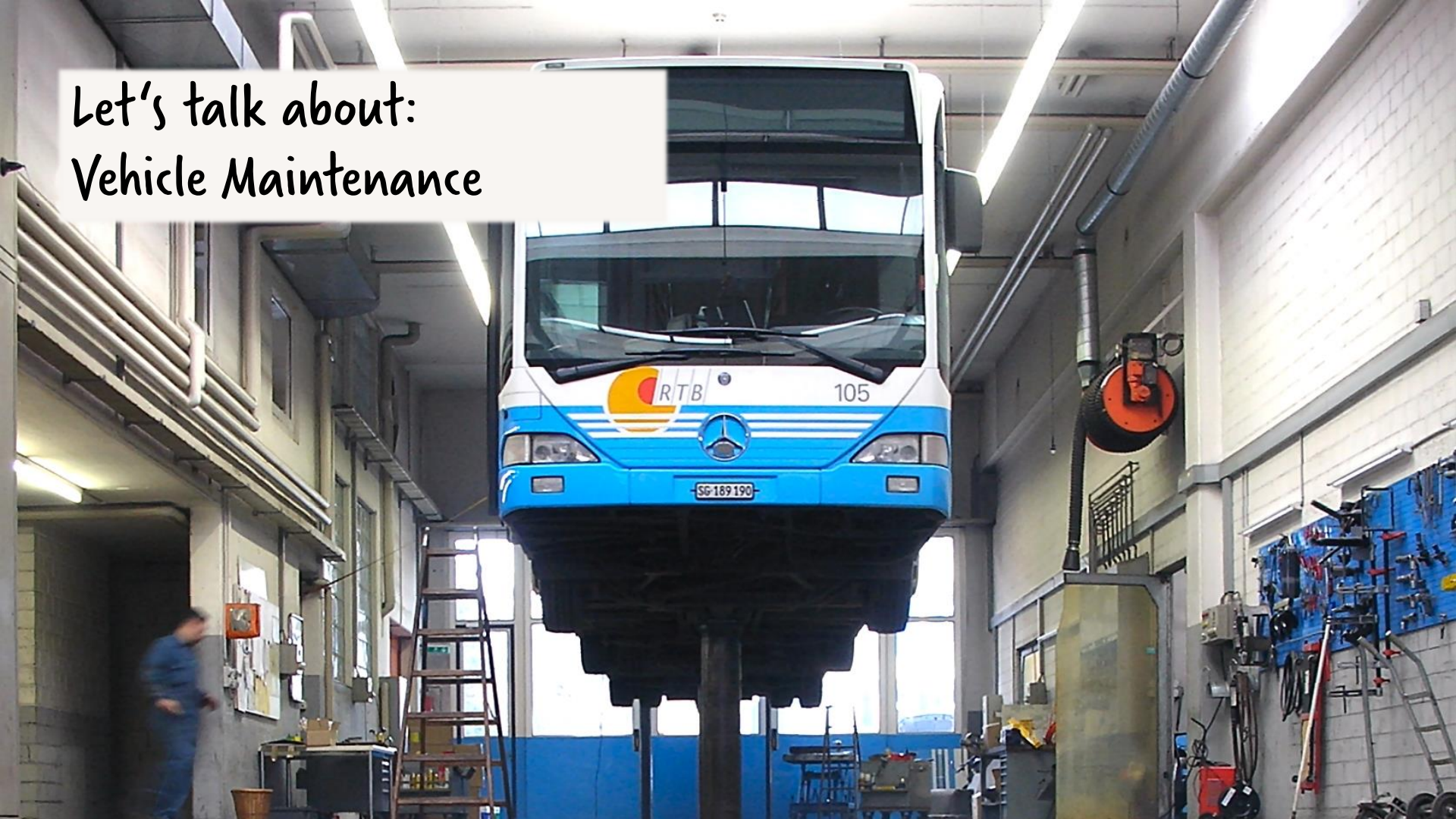
Architektur & Sicherheit

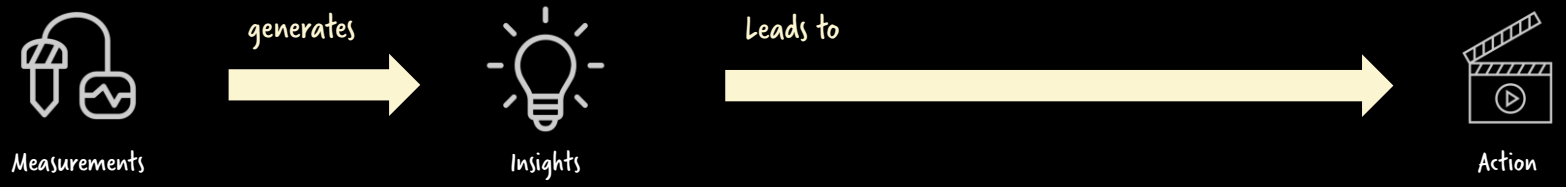
Loosely or lousily coupled? Understanding communication patterns in modern architectures

Bernd Rücker

A5

Let's talk about:
Vehicle Maintenance





oil pressure is 80 psi

oil pressure is critically high

Schedule maintenance

Call driver to stop and inspect

...

Event Streams

Workflows

Measurements

generates

Insights

Leads to

Decision

Action

Kafka Topic Measures

Stateful Stream Processor

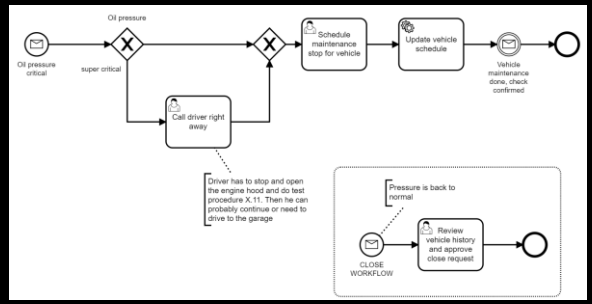
Kafka Topic Insights

Custom Connector

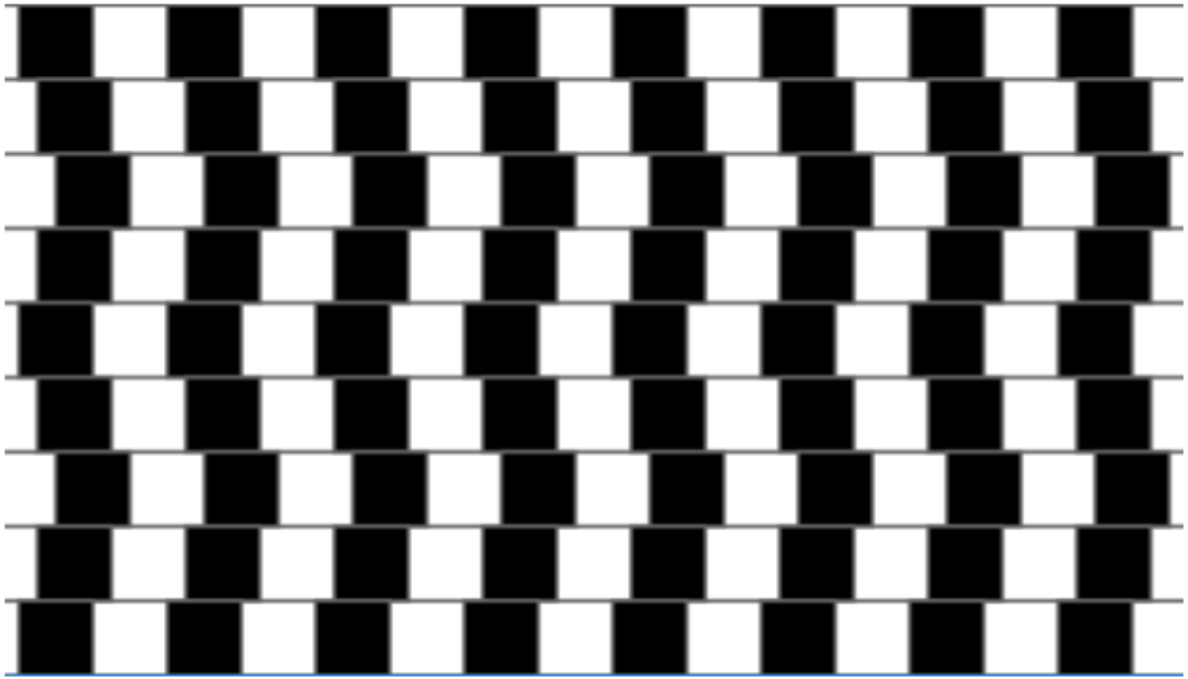
Camunda Workflow Maintenance

oil pressure is 80 psi

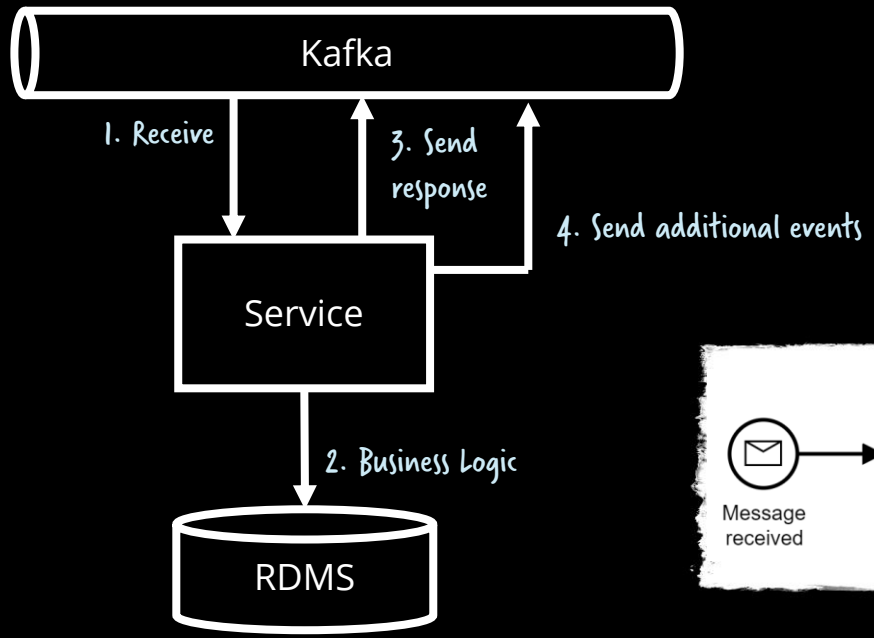
oil pressure is critically high

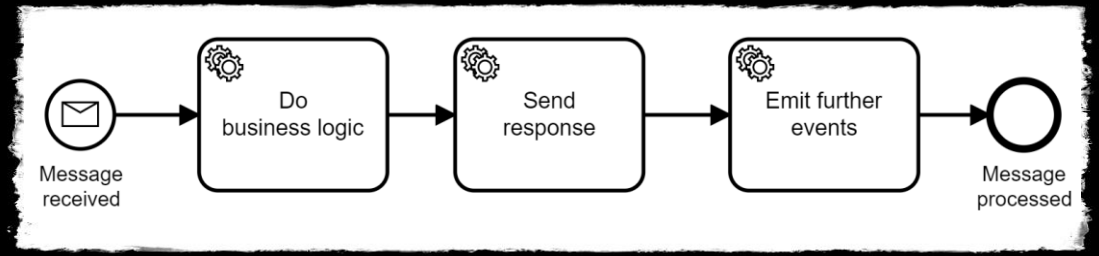


Let's talk about consistency

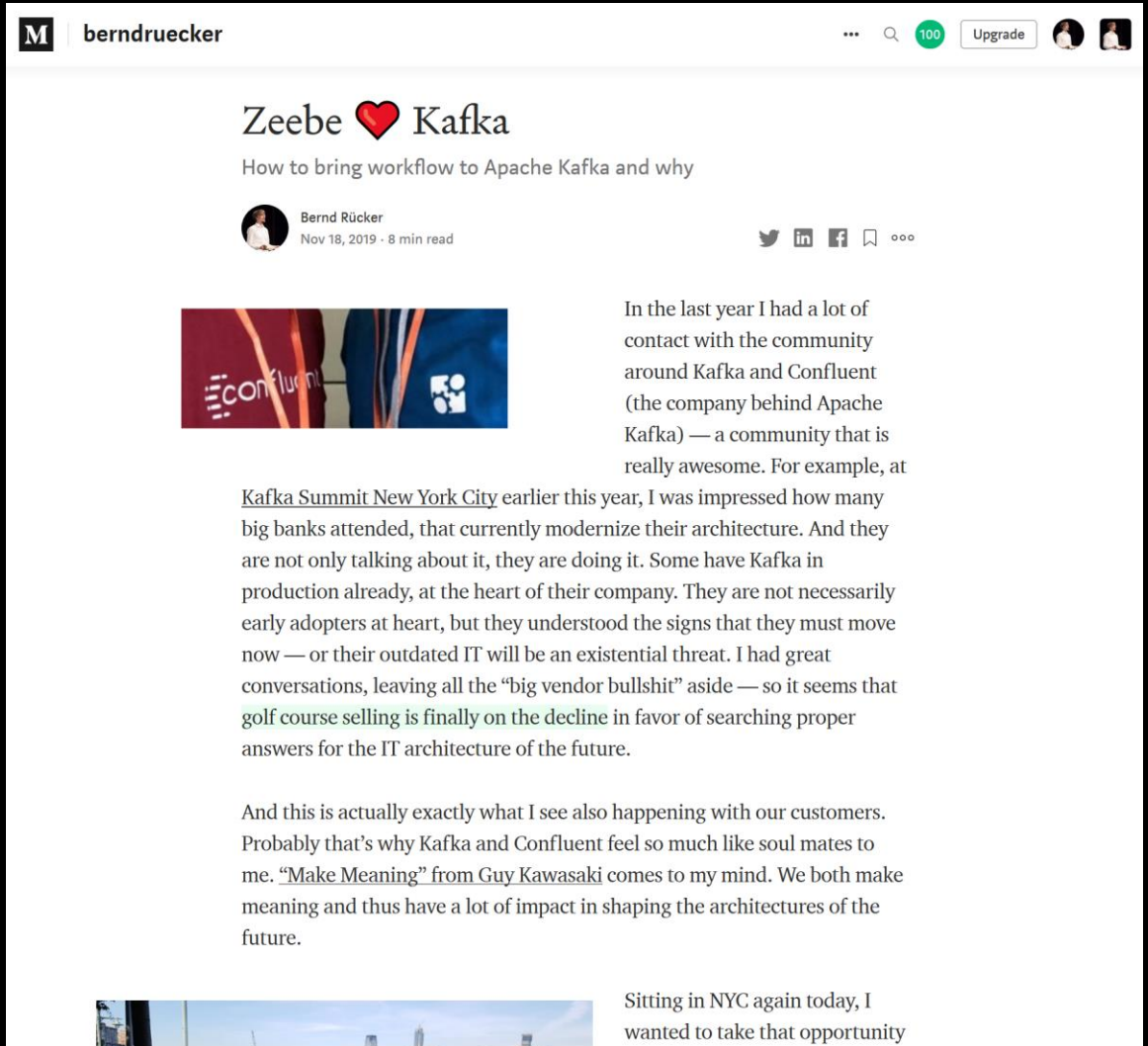


Another use case: Transactions





@berndruecker



berndruecker

Upgrade

Zeebe ❤ Kafka

How to bring workflow to Apache Kafka and why

Bernd Rücker
Nov 18, 2019 · 8 min read

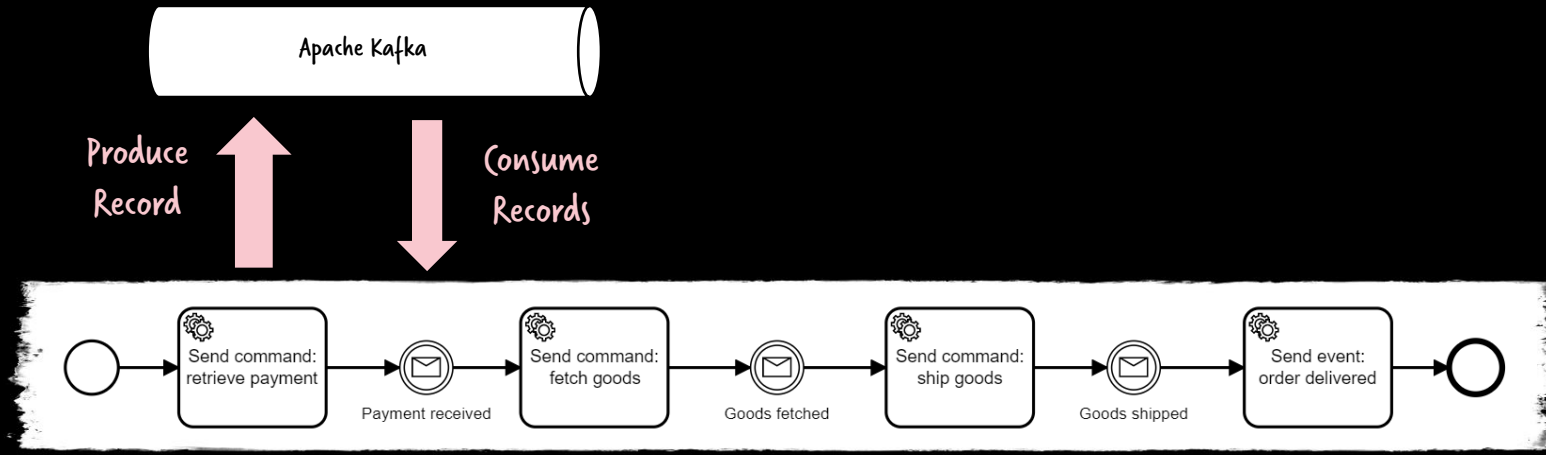
In the last year I had a lot of contact with the community around Kafka and Confluent (the company behind Apache Kafka) — a community that is really awesome. For example, at Kafka Summit New York City earlier this year, I was impressed how many big banks attended, that currently modernize their architecture. And they are not only talking about it, they are doing it. Some have Kafka in production already, at the heart of their company. They are not necessarily early adopters at heart, but they understood the signs that they must move now — or their outdated IT will be an existential threat. I had great conversations, leaving all the "big vendor bullshit" aside — so it seems that golf course selling is finally on the decline in favor of searching proper answers for the IT architecture of the future.

And this is actually exactly what I see also happening with our customers. Probably that's why Kafka and Confluent feel so much like soul mates to me. "Make Meaning" from Guy Kawasaki comes to my mind. We both make meaning and thus have a lot of impact in shaping the architectures of the future.

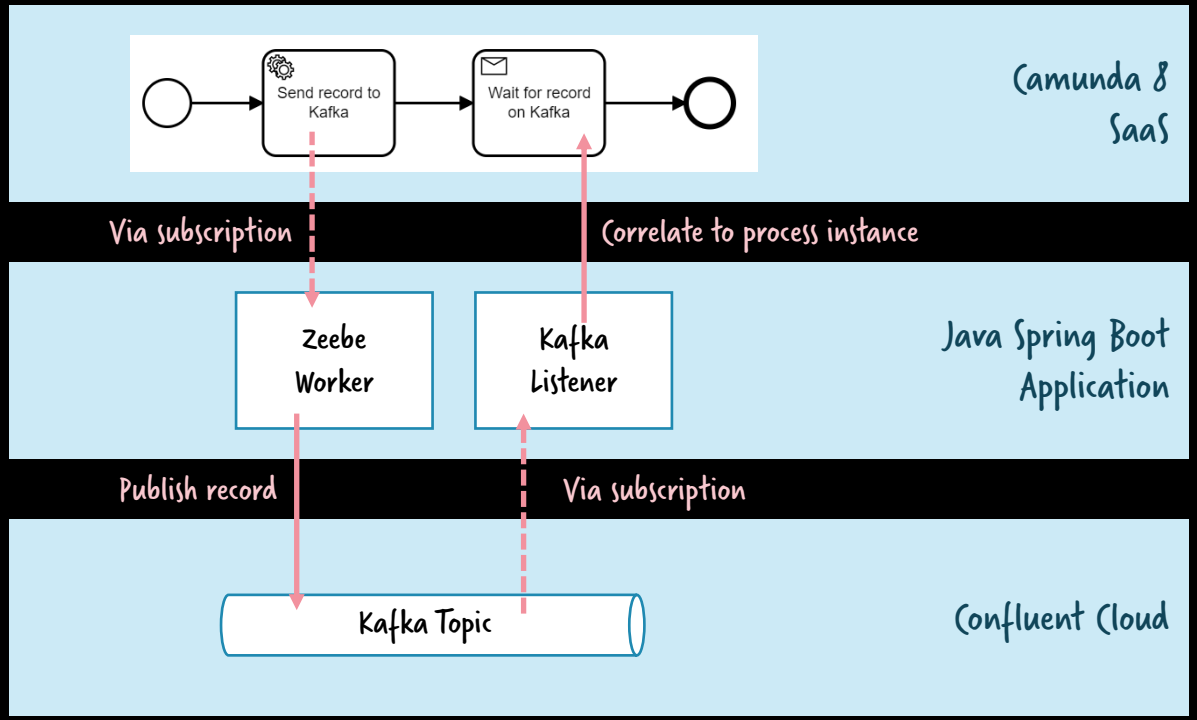
Sitting in NYC again today, I wanted to take that opportunity

https://blog.bernd-ruecker.com/zeebe-loves-kafka-d82516030f99

Camunda ♥ Kafka



Technical Example



https://github.com/berndruecker/kafka-camunda-spring-simple

Kafka Connect

camunda-community-hub / kafka-connect-zeebe

Watch 5 | Unstar 52 | Fork 23

Code | Issues 15 | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

master | kafka-connect-zeebe / README.md | Go to file

berndruecker Switched to 1.1 and introduced cloud region parameter (fixes #59) ✓ Latest commit 9b8541c 21 hours ago History

7 contributors

136 lines (76 sloc) 7.74 KB | Raw | Blame

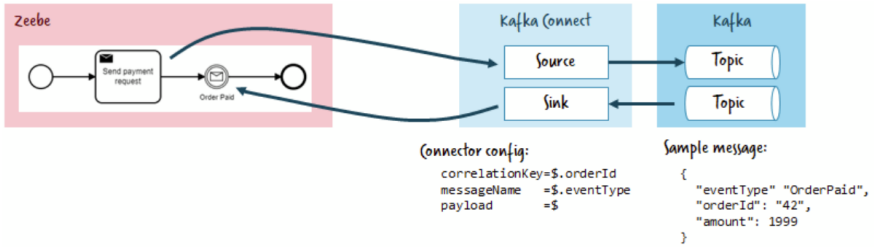
Community Extension | An open source community maintained project | Lifecycle | Incubating

kafka-connect-zeebe

This Kafka Connect connector for Zeebe can do two things:

- **Send messages to a Kafka topic** when a workflow instance reached a specific activity. Please note that a `message` is more precisely a kafka `record`, which is also often named `event`. This is a **source** in the Kafka Connect speak.
- **Consume messages from a Kafka topic and correlate them to a workflow**. This is a Kafka Connect **sink**.

It can work with Camunda Cloud or a self-managed Zeebe broker.



See this blog post for some background on the implementation.

Use Cases

Microservices orchestration (balanced with choreography)

Actions resulting from streams

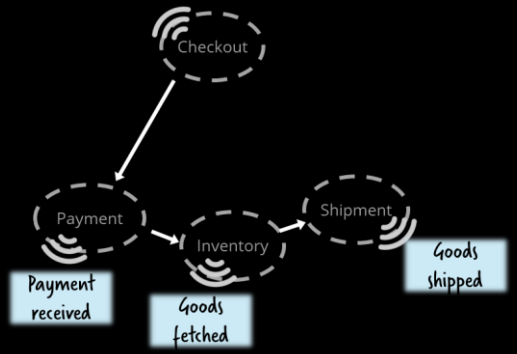
Business transactions & SAGA

Summary

- A workflow engine is complementary to Apache Kafka
- The technical integration is easy and there are tools that scale like Apache Kafka
- Know the playing fields

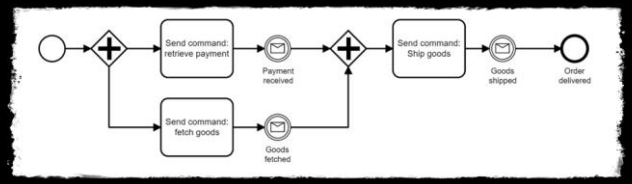
I hope you got a better feeling about...

Streaming



Event-Driven Architecture

Orchestration




Thank you!
@bernduecker

Thank you!

Book: https://ProcessAutomationBook.com/

Contact: mail@berndruecker.io
@berndruecker

Slides: https://berndruecker.io

Blog: https://medium.com/berndruecker

Code: https://github.com/berndruecker