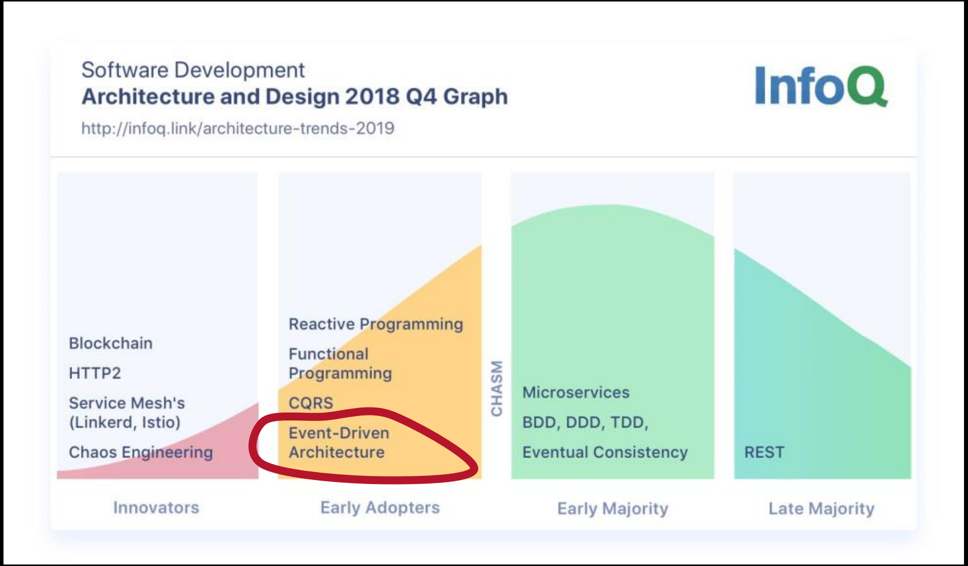
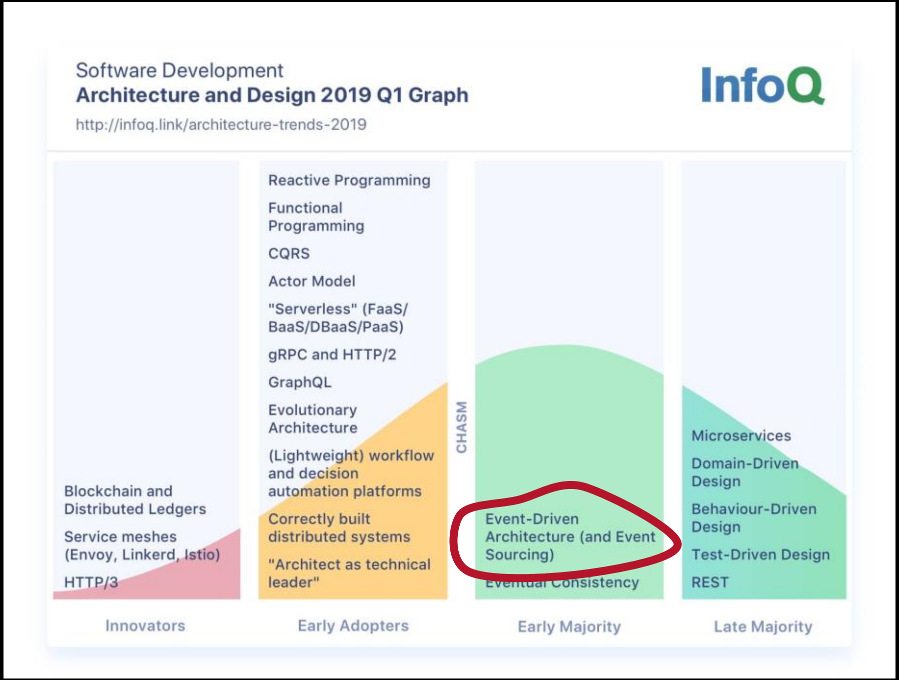
Opportunities & Pitfalls of Event-Driven Utopia

@berndruecker

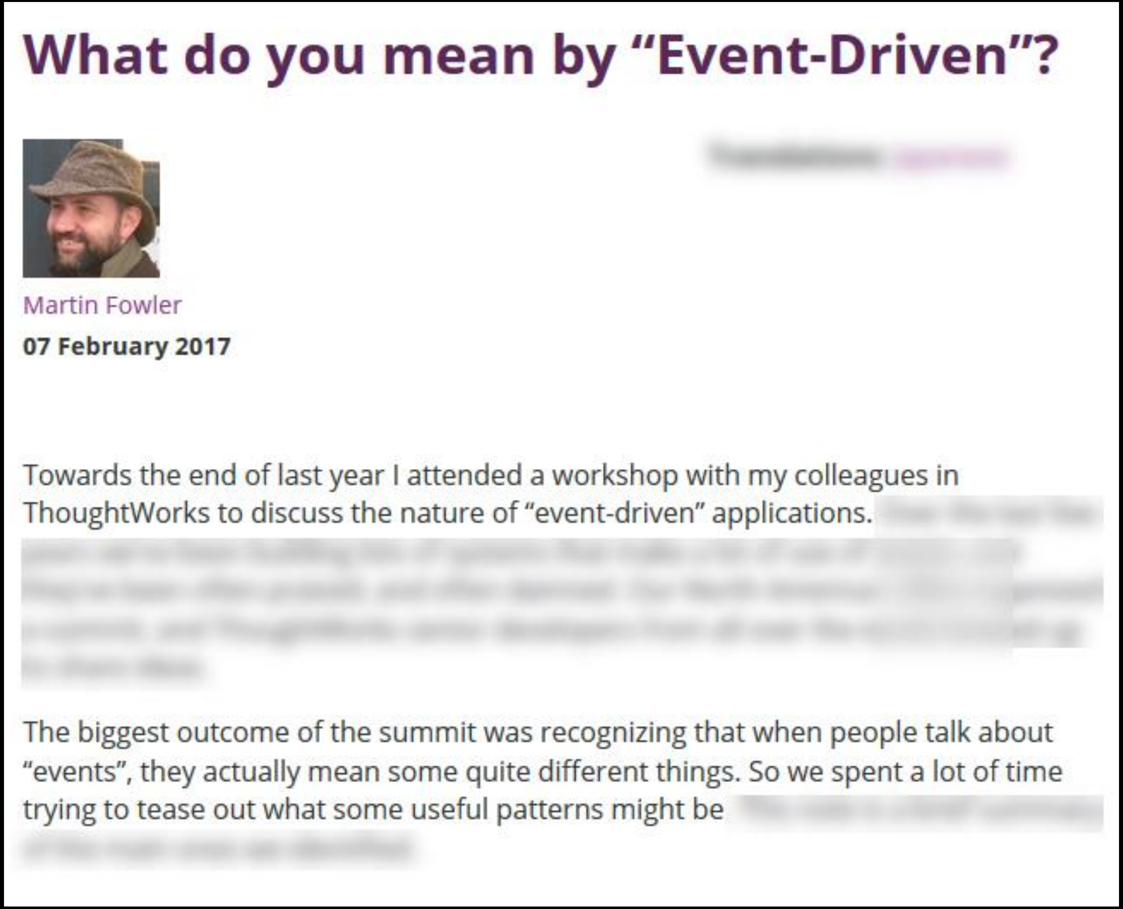
LES UTOPIES DE LA NAVIGATION AÉRIENNE AU SIÈCLE DERNIER.
COLLECTION 476
2ème Série (N° 2)
ROMANET & Cie IMP. EDIT. PARIS.

Why this talk





Why this talk



What do you mean by “Event-Driven”?

[illegible]

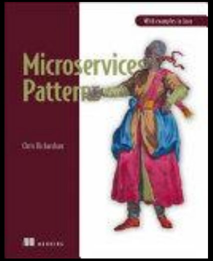
Martin Fowler

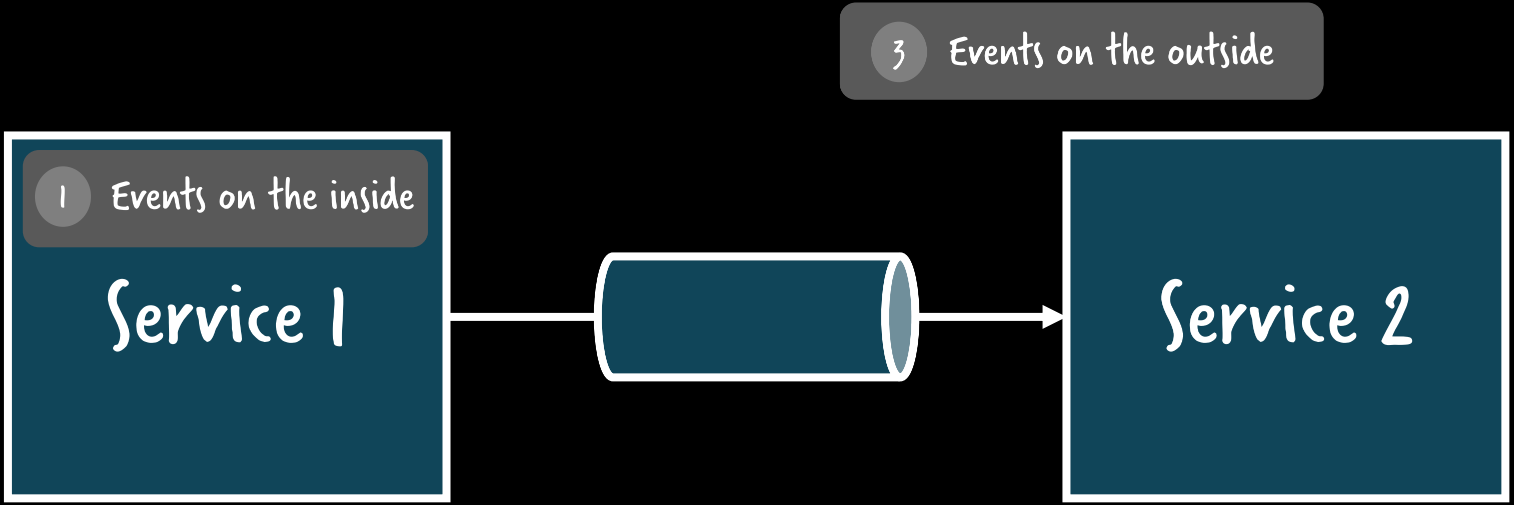
07 February 2017

Towards the end of last year I attended a workshop with my colleagues in ThoughtWorks to discuss the nature of “event-driven” applications. [illegible]

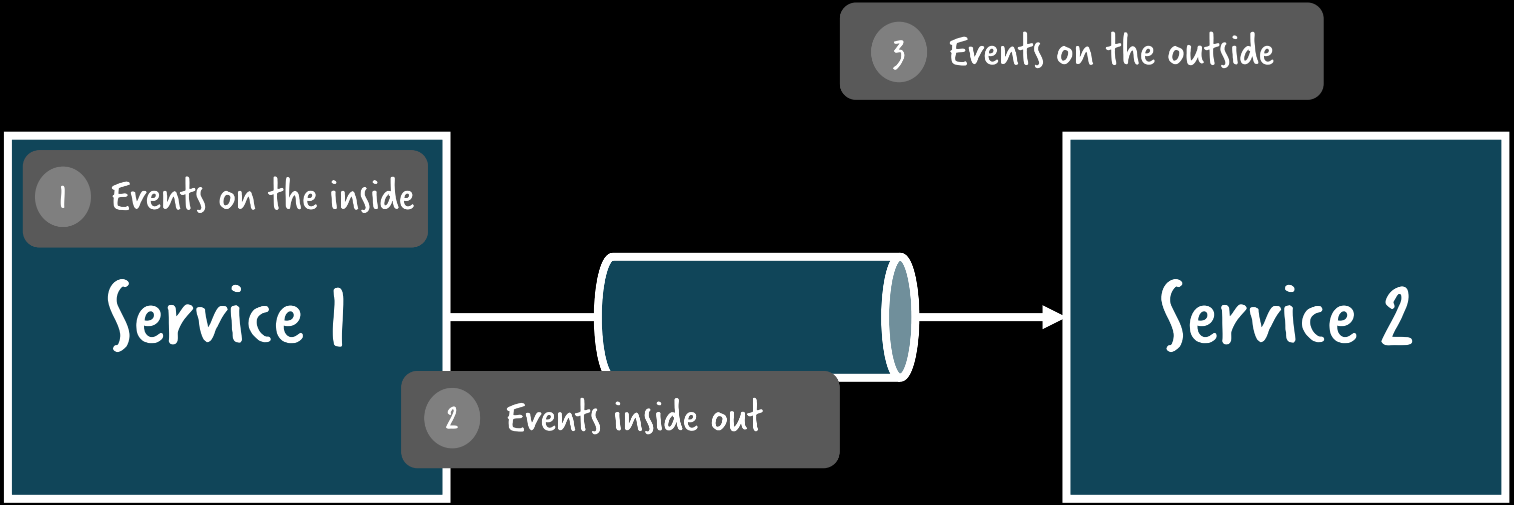
The biggest outcome of the summit was recognizing that when people talk about “events”, they actually mean some quite different things. So we spent a lot of time trying to tease out what some useful patterns might be [illegible]

Agenda

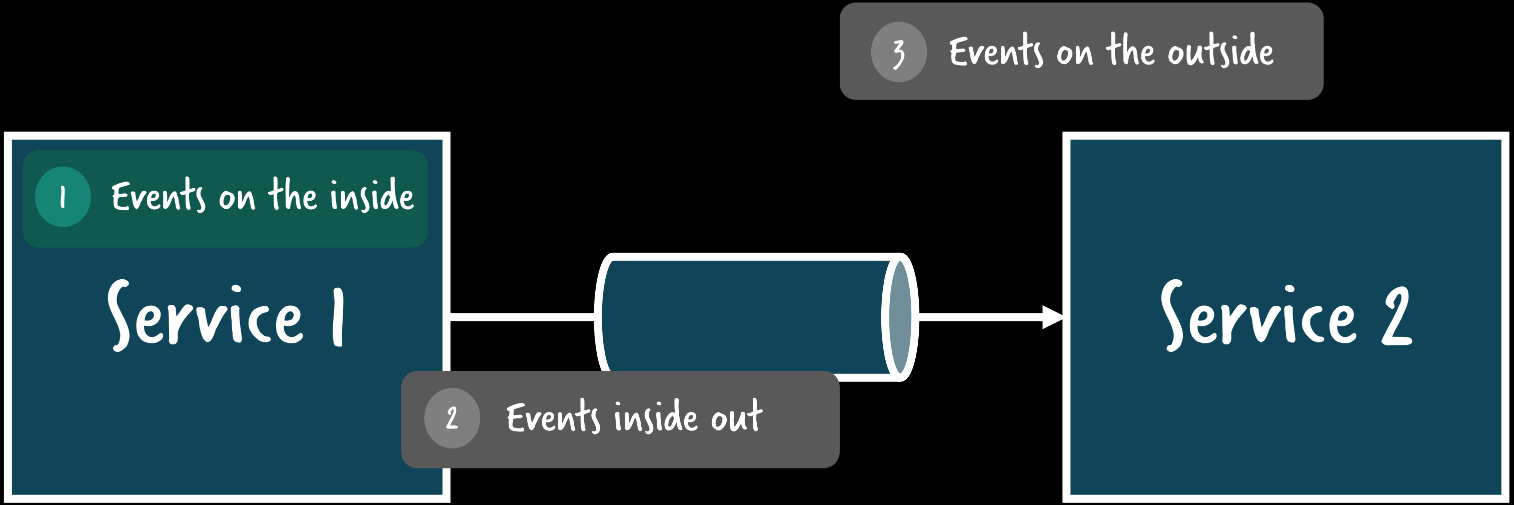



3 Events on the outside
1 Events on the inside
Service 1
Service 2

Agenda


1 Events on the inside
Service 1
2 Events inside out
3 Events on the outside
Service 2

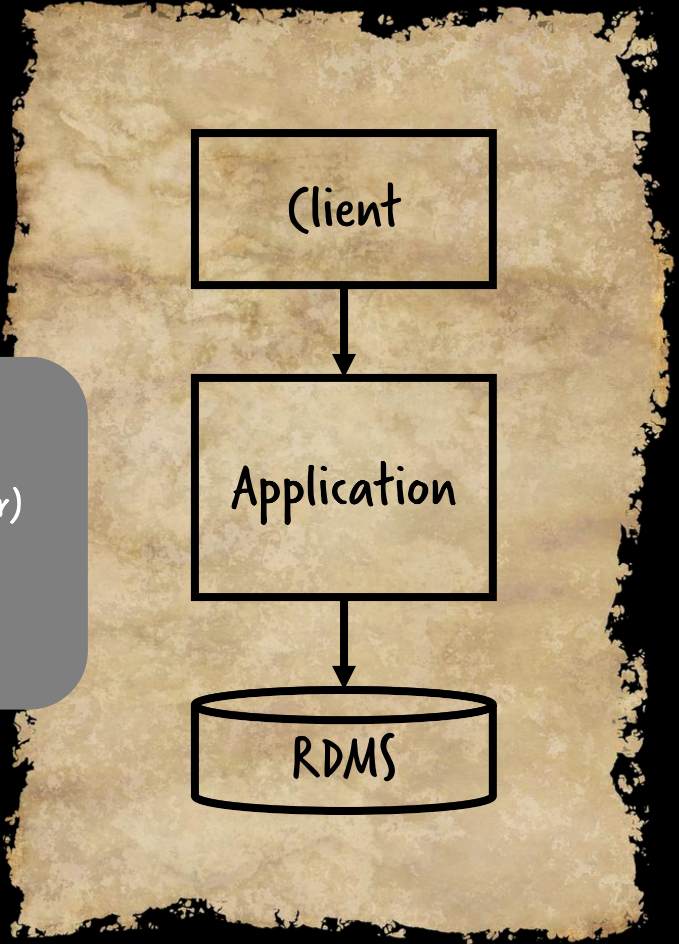
Agenda


3 Events on the outside
1 Events on the inside
Service 1
Service 2
2 Events inside out

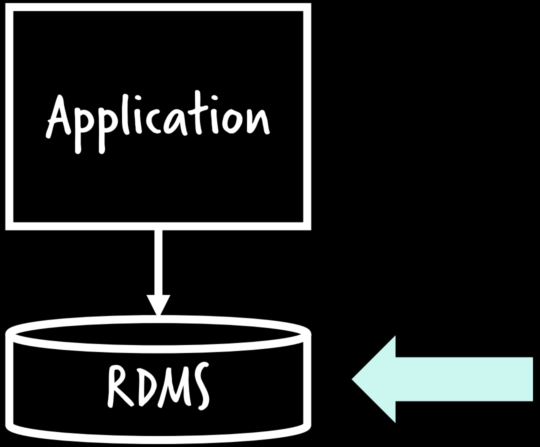
once upon a time...

Ted Neward
@tedneward
Computational philosopher: techno-jargon for I am a big geek.
Redmond, WA
about.me/TedNeward
Joined June 2008

BBC architecture
(box - arrow – box – arrow - cylinder)
Every architecture diagram
you'll ever need

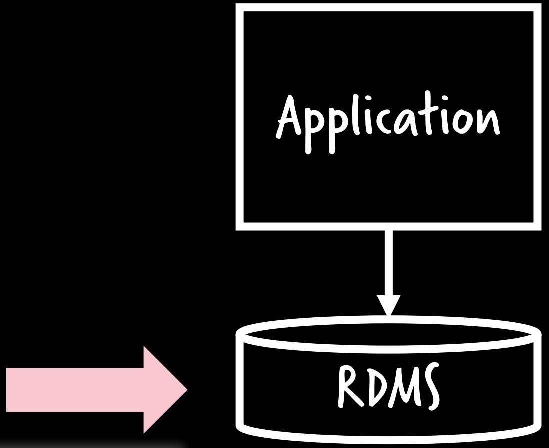


The great thing about this architecture



DB gurantees
(e.g. ACID)

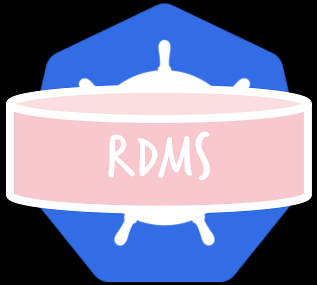
The problem



not cloud-scale

resiliency is expensive

Does not fit in Kubernetes:

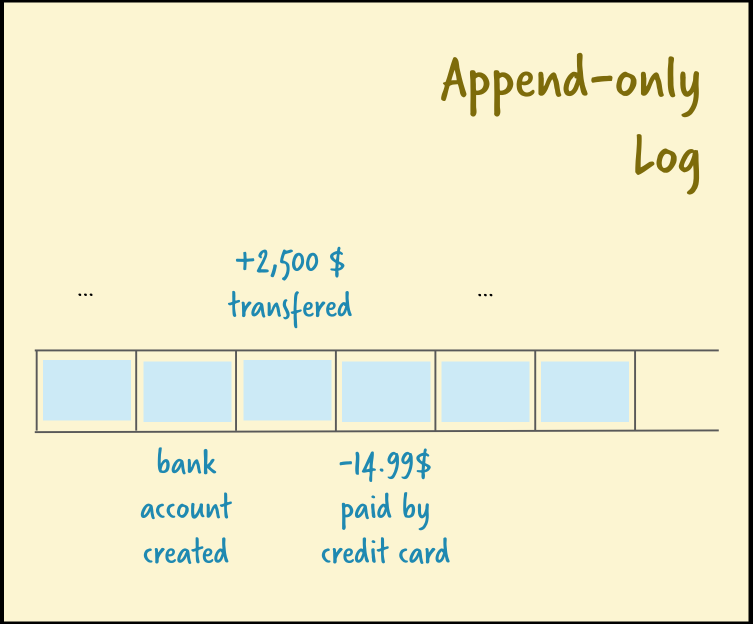


Pat Helland

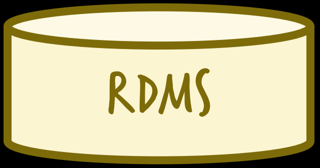
https://vimeo.com/52831373

Persistent change

Persistent state

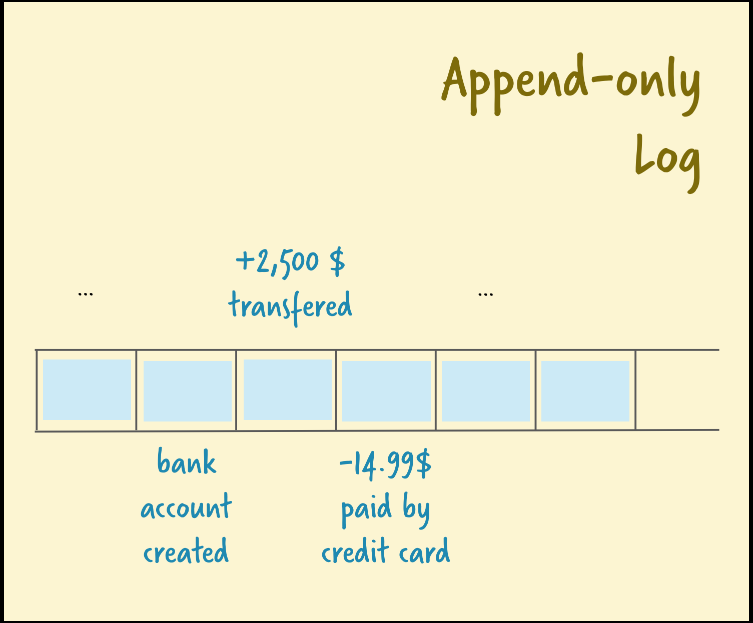


Current Balance = 2,485.01 $



Account #	Balance
12345	2,500$

Persistent change



Event

Bank Account Created
2019/04/16 11:00
# 12345

Event

Money Transfer Received
2,500$
2019/04/16 11:00
# 12345

An example from my world

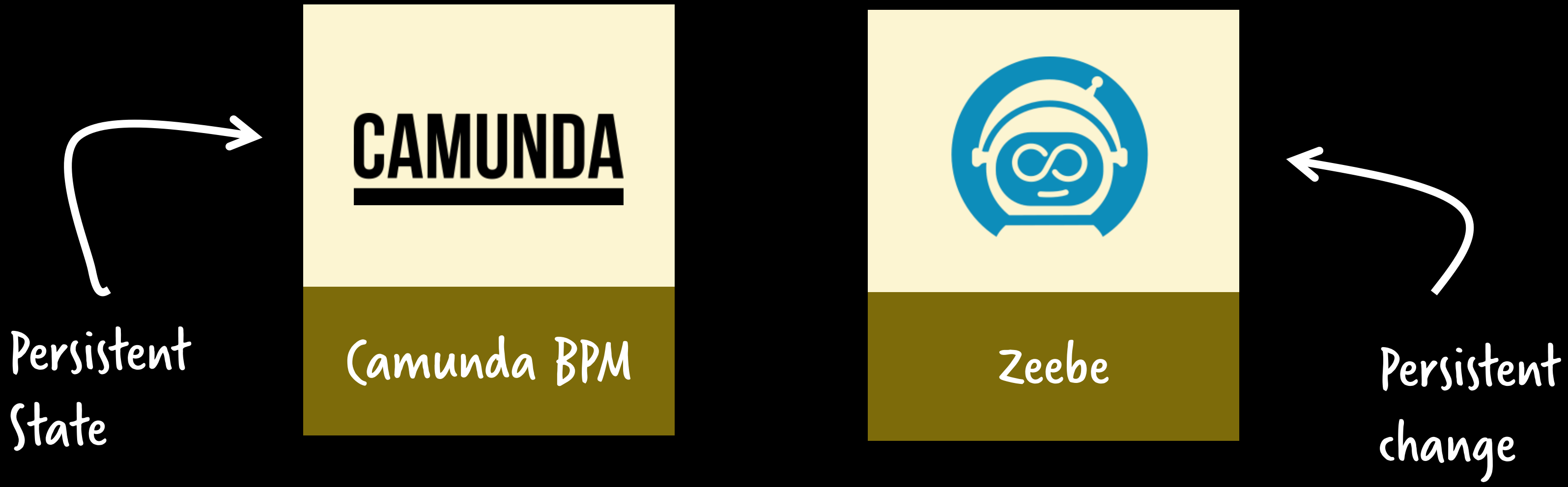
Bernd Ruecker

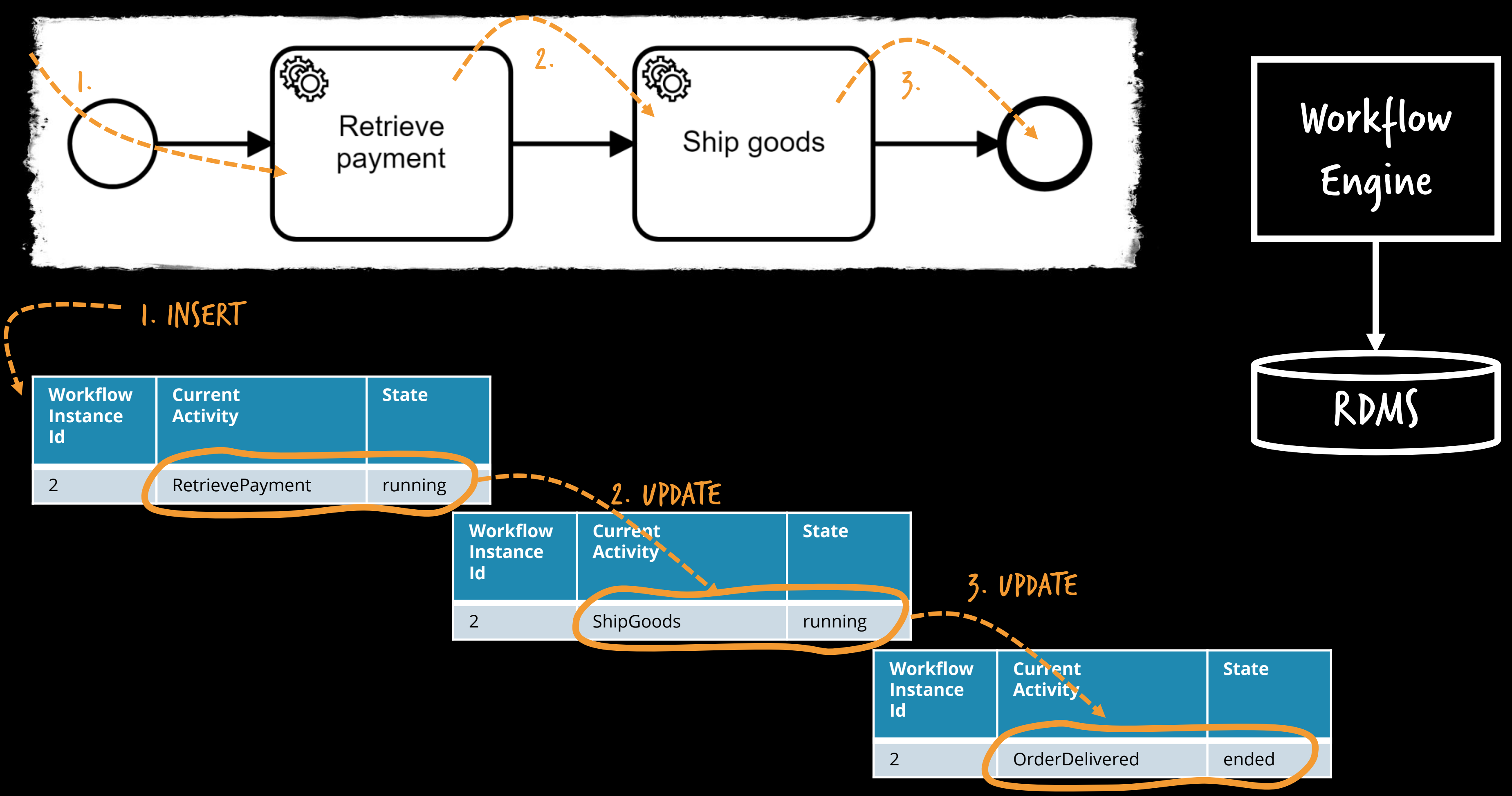
Co-founder and
Chief Technologist of
Camunda

mail@berndruecker.io
@berndruecker

Warning:
Contains oppinion!

We build two different workflow engines. Why?

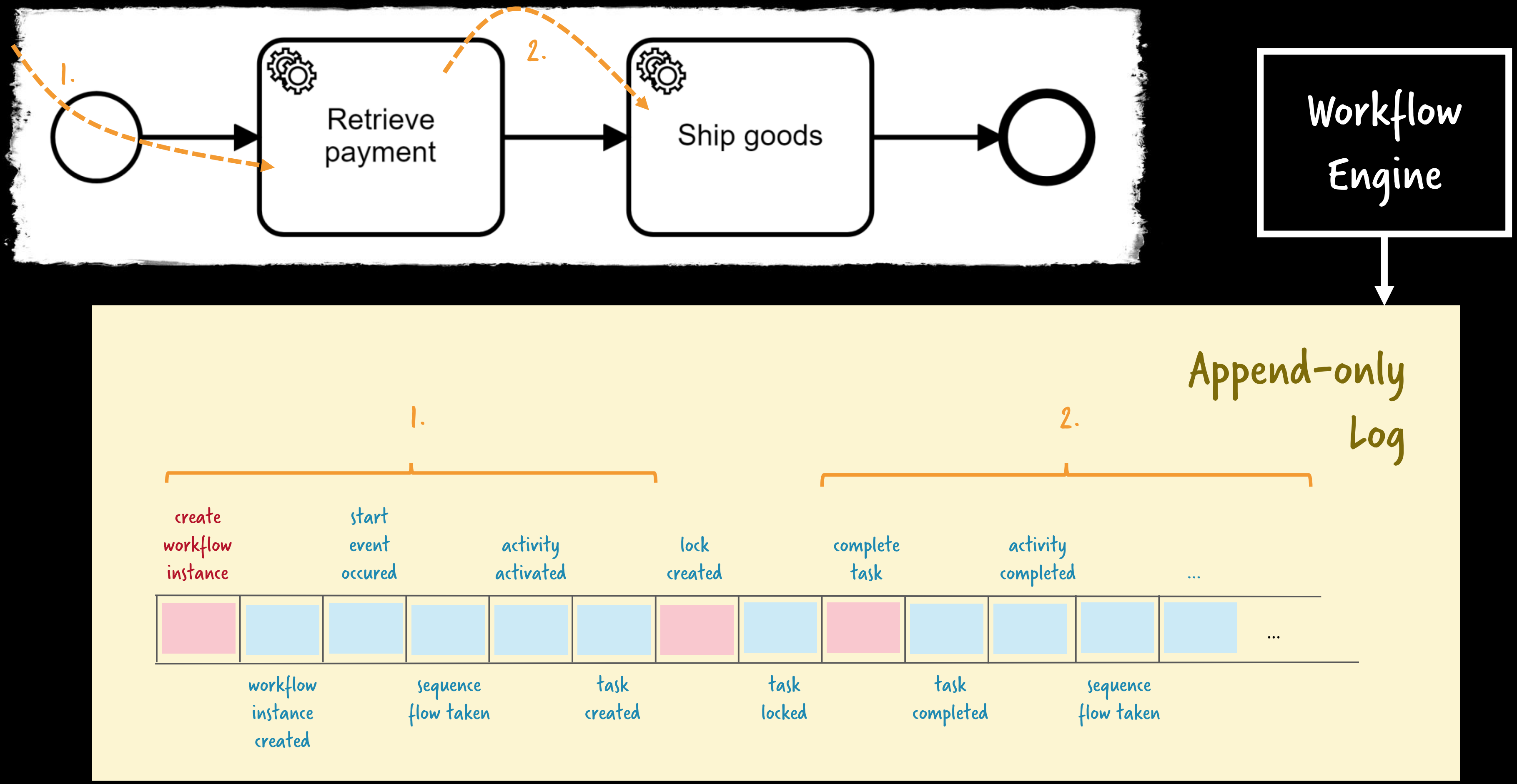




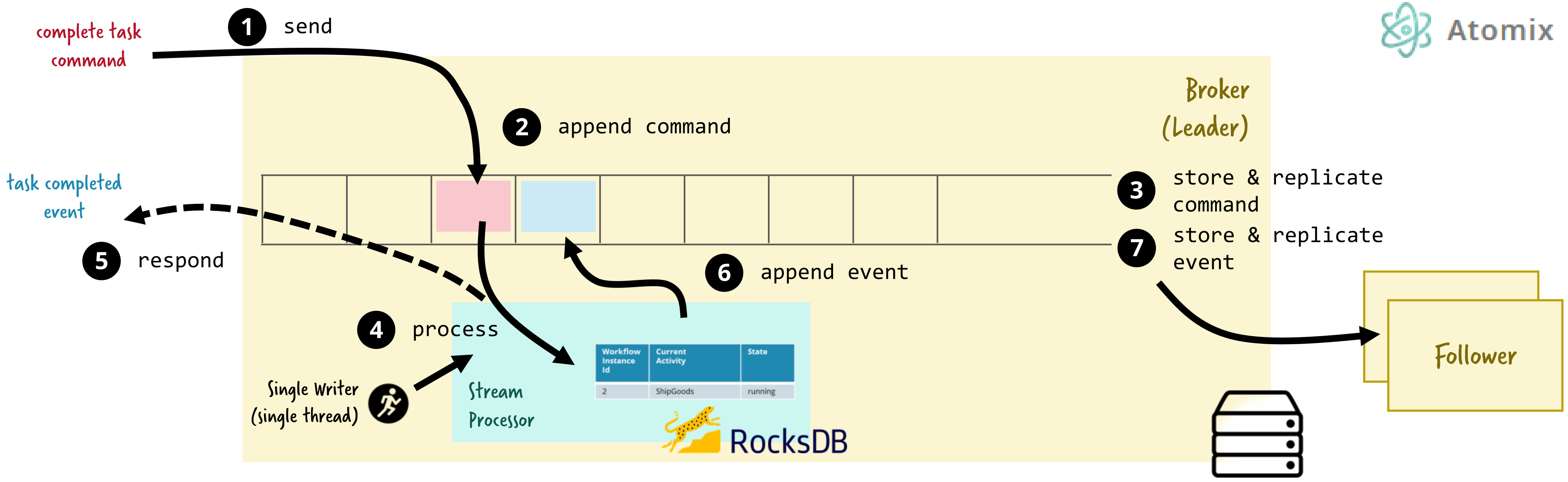
Workflow Instance Id	Current Activity	State
2	RetrievePayment	running

Workflow Instance Id	Current Activity	State
2	ShipGoods	running

Workflow Instance Id	Current Activity	State
2	OrderDelivered	ended


1.
Retrieve payment
2.
Ship goods
Workflow Engine
Append-only Log
1.
2.
create workflow instance
workflow instance created
start event occured
sequence flow taken
activity activated
task created
lock created
task locked
complete task
task completed
activity completed
sequence flow taken
...
...

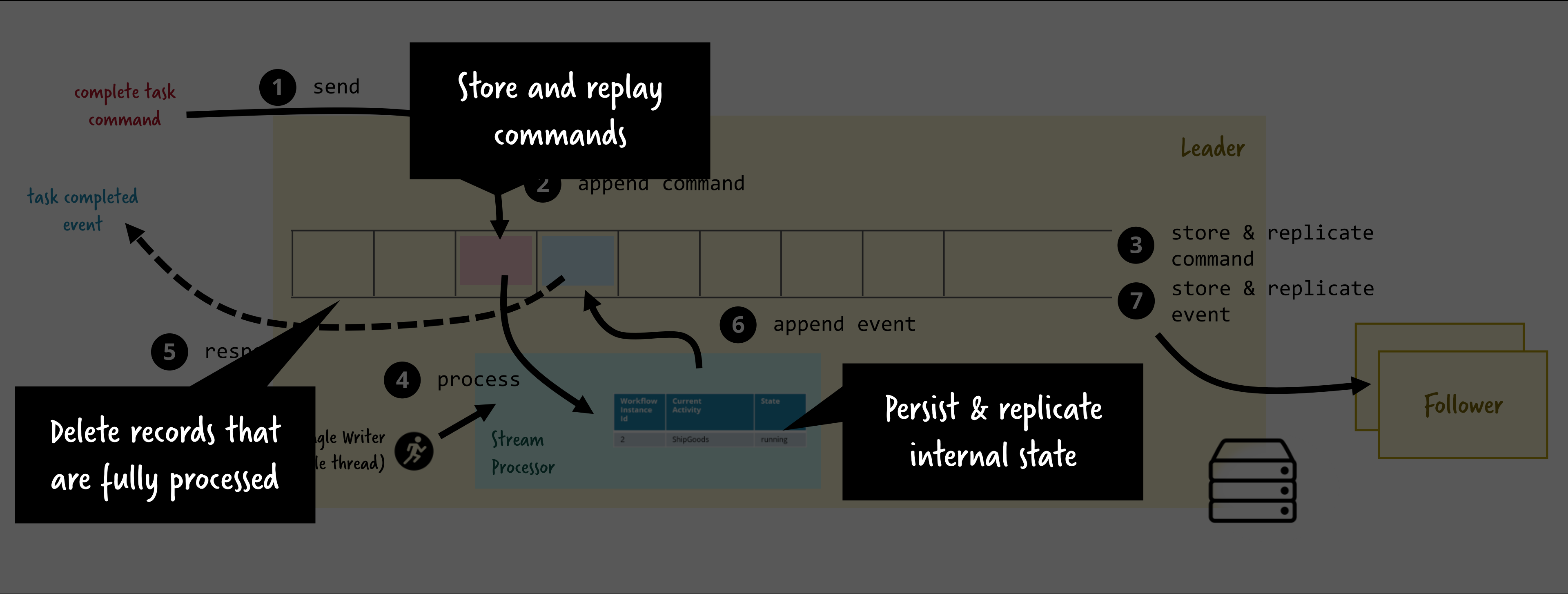
Event Handling, Replication & Single Writer



Workflow Instance Id	Current Activity	State
2	ShipGoods	running

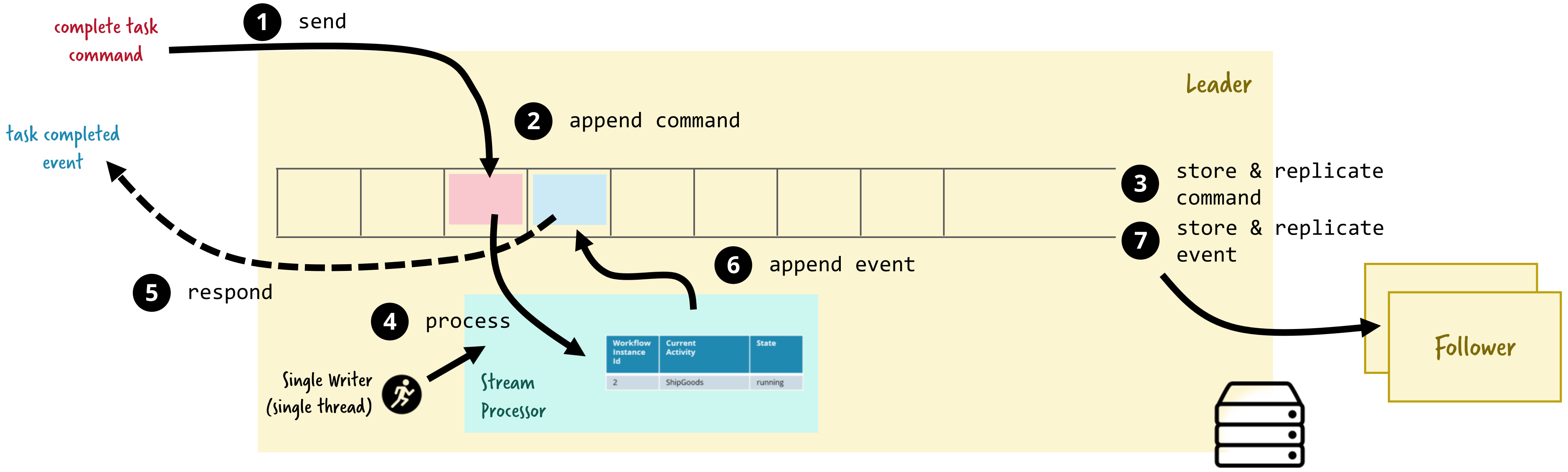
Atomix

What we do different



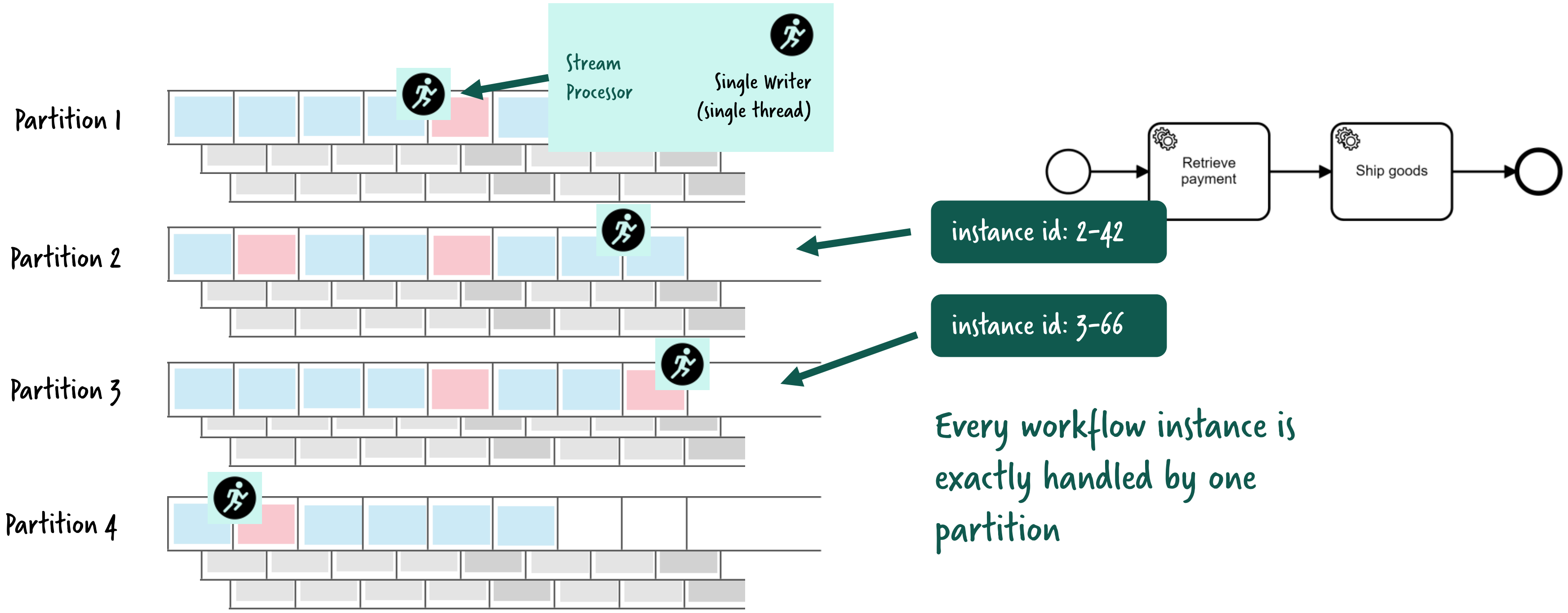
Consistency
Availability
Partition

Zeebe is CP



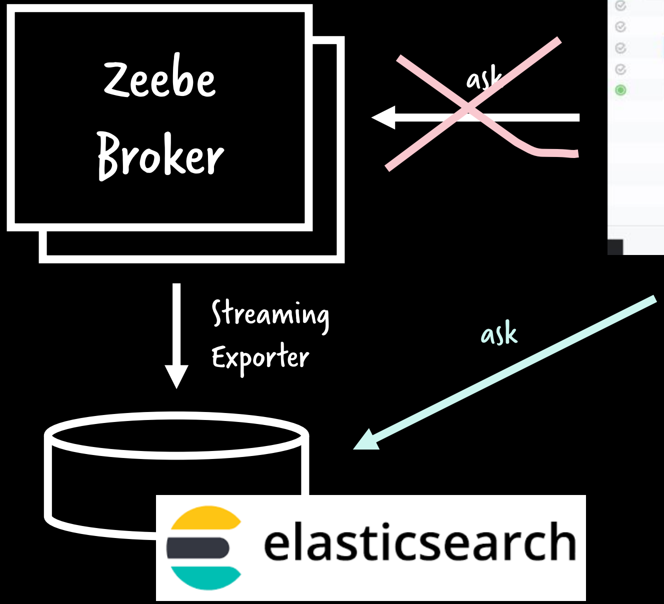
Workflow Instance Id	Current Activity	State
2	ShipGoods	running

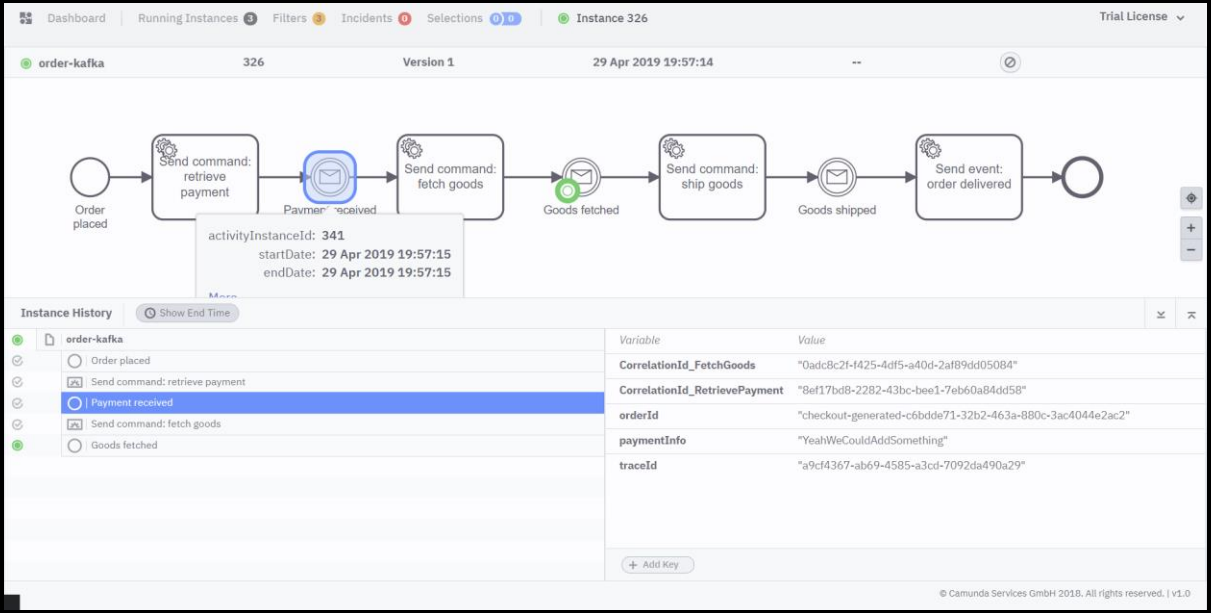
Horizontal scalability by partitioning



Every workflow instance is exactly handled by one partition

Queries and read models





order-kafka | 326 | Version 1 | 29 Apr 2019 19:57:14 | --

activityInstanceId: 341
startDate: 29 Apr 2019 19:57:15
endDate: 29 Apr 2019 19:57:15

Instance History

- order-kafka
  - Order placed
  - Send command: retrieve payment
  - Payment received
  - Send command: fetch goods
  - Goods fetched

Variable	Value
CorrelationId_FetchGoods	"0adc8c2f-f425-4df5-a40d-2af89dd05084"
CorrelationId_RetrievePayment	"8ef17bd8-2282-43bc-bee1-7eb60a84dd58"
orderId	"checkout-generated-c6bdde71-32b2-463a-880c-3ac4044e2ac2"
paymentInfo	"YeahWeCouldAddSomething"
traceId	"a9cf4367-ab69-4585-a3cd-7092da490a29"

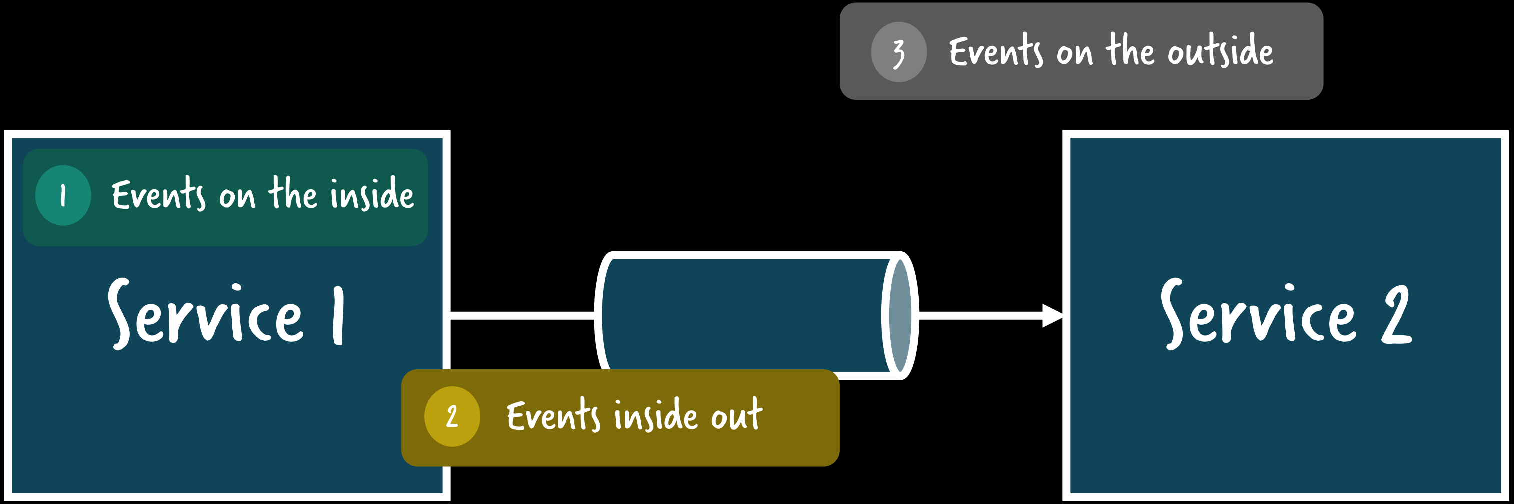
© Camunda Services GmbH 2018. All rights reserved. | v1.0

Recap 1 – Events on the inside

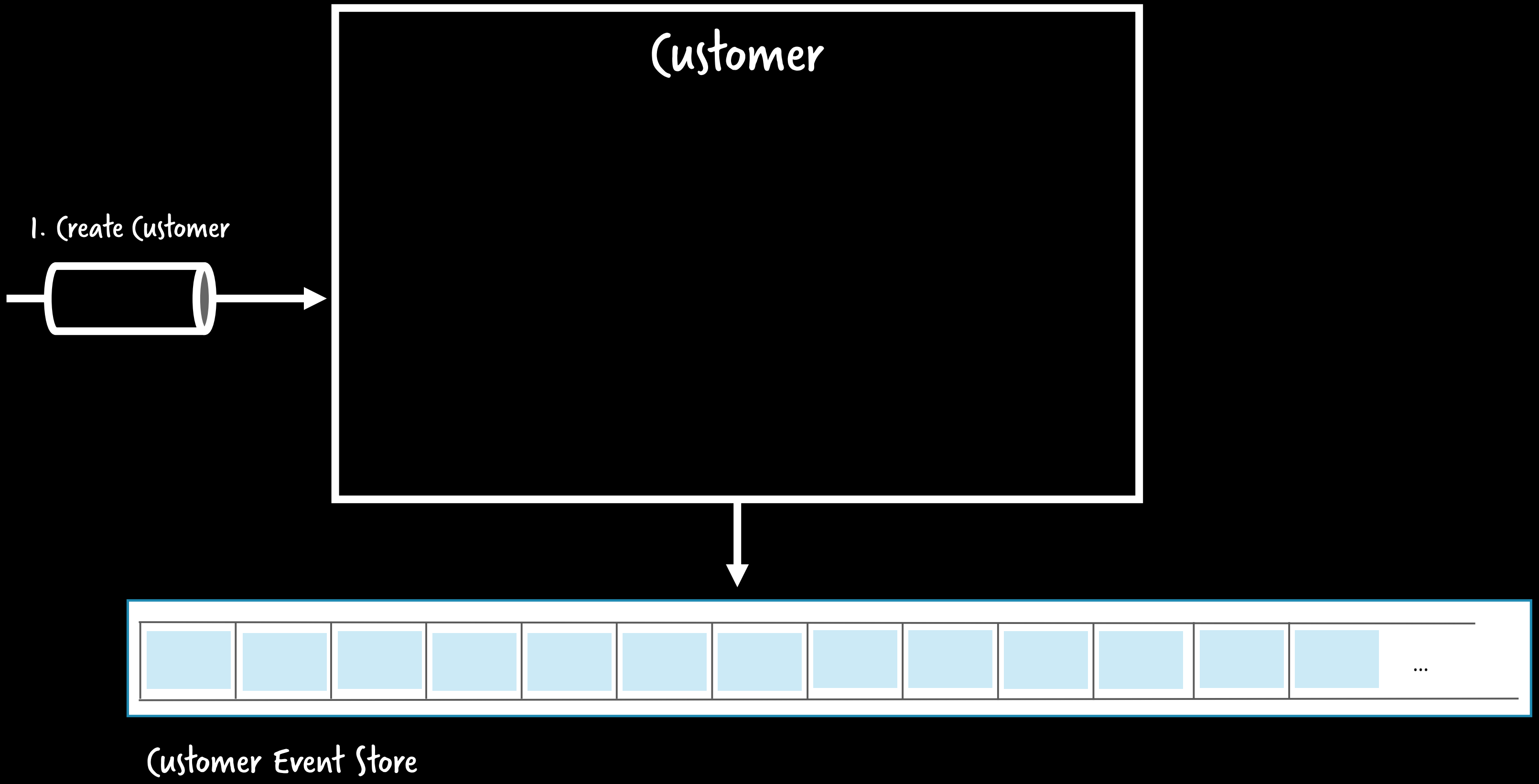
# Natural mechanism to build scalable services in distributed systems (with outbox & co included)

But
# You have to think about reads, queries & eventual consistency
# Few industry experience available

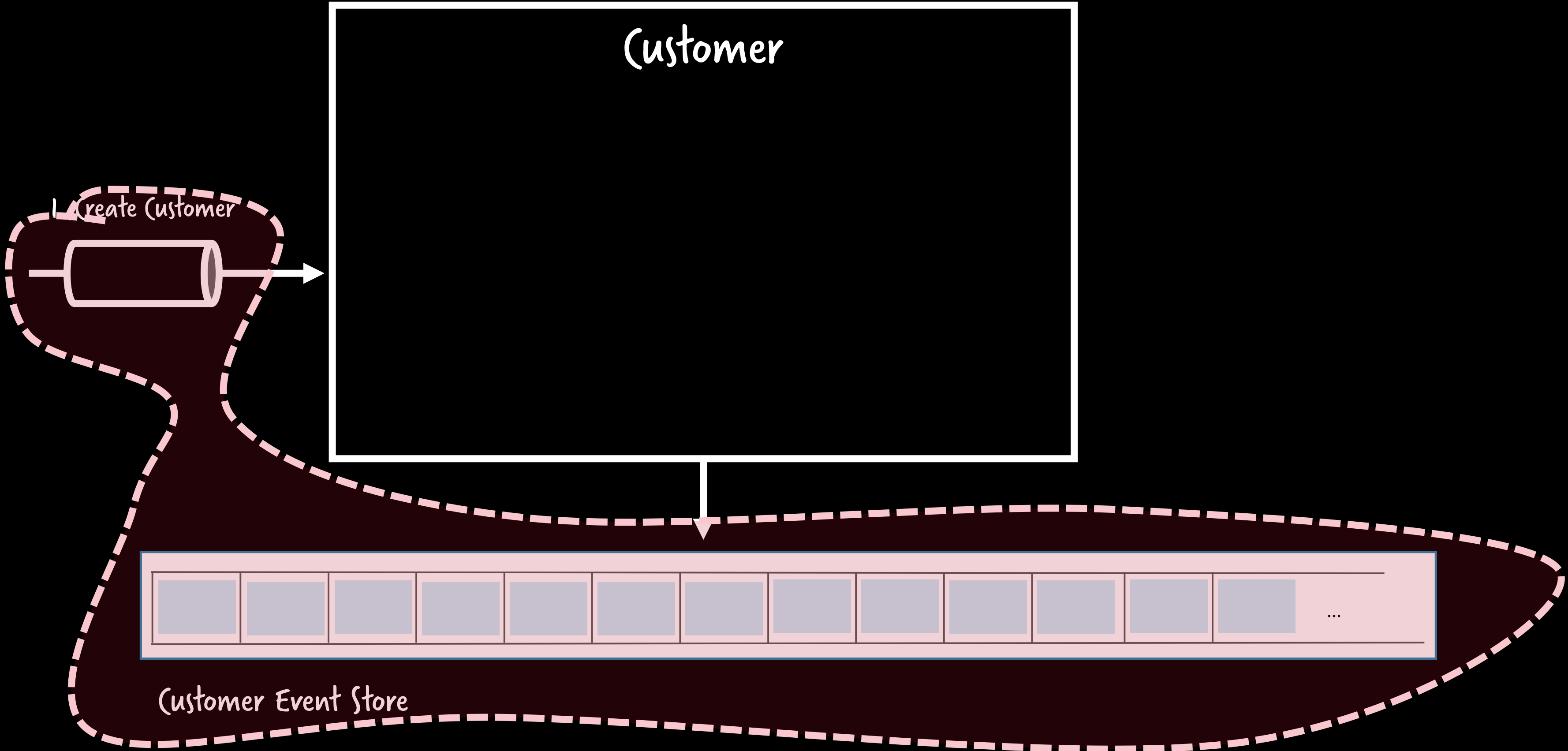
Agenda


3 Events on the outside
1 Events on the inside
Service 1
Service 2
2 Events inside out

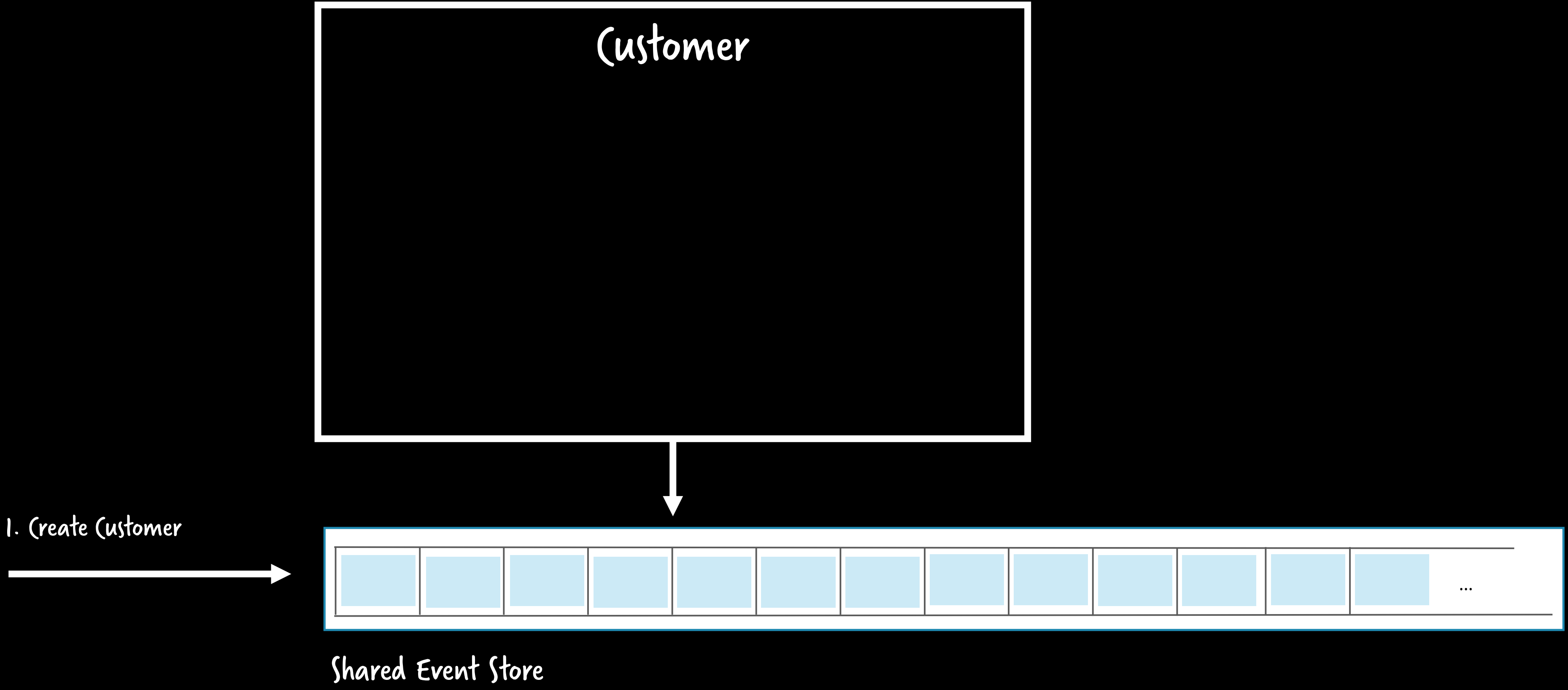
Event Store and Messaging



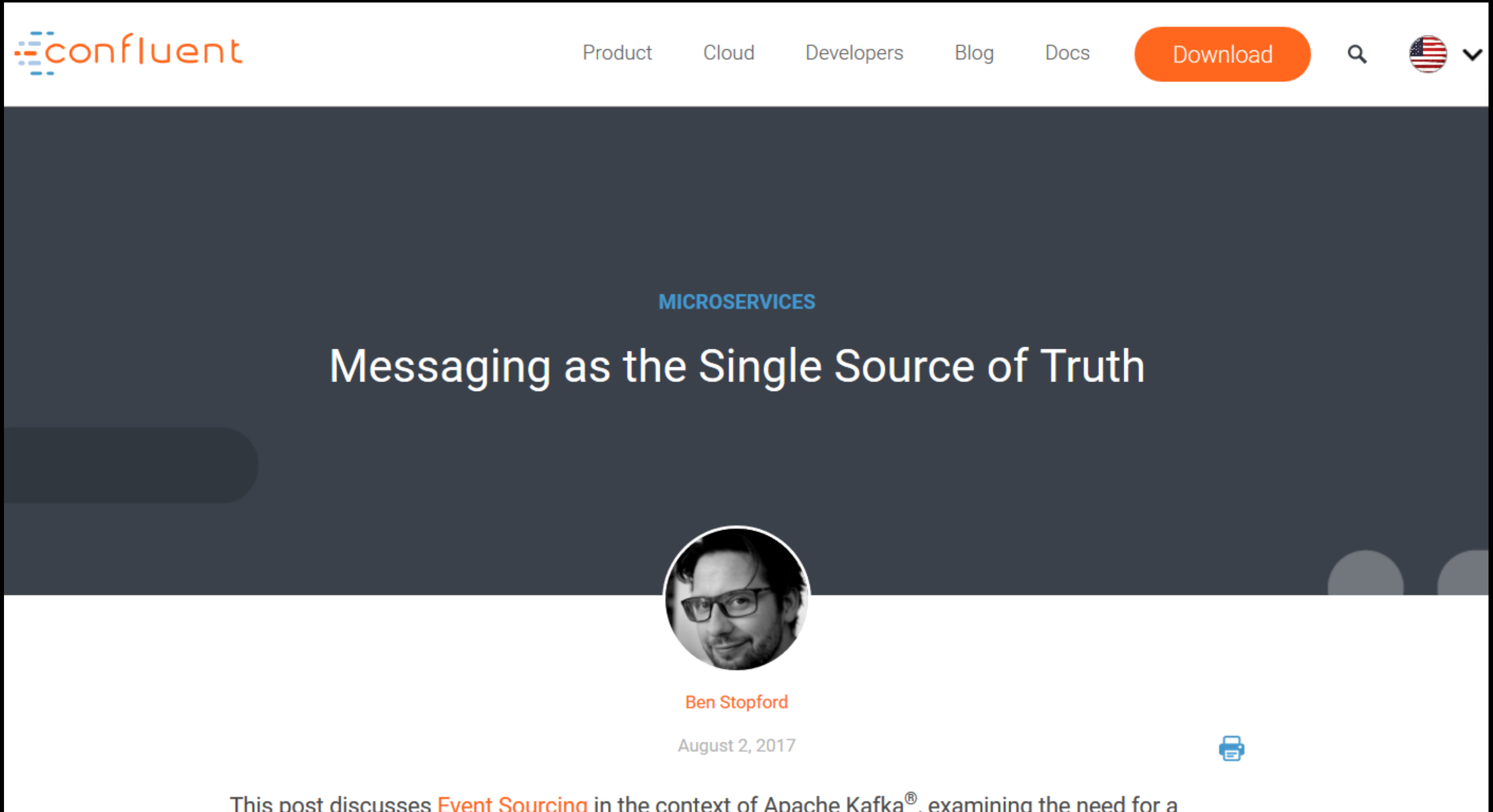
Merge Messaging and Event Store



Merge messaging and event store



Enter the world of Kafka...



MICROSERVICES

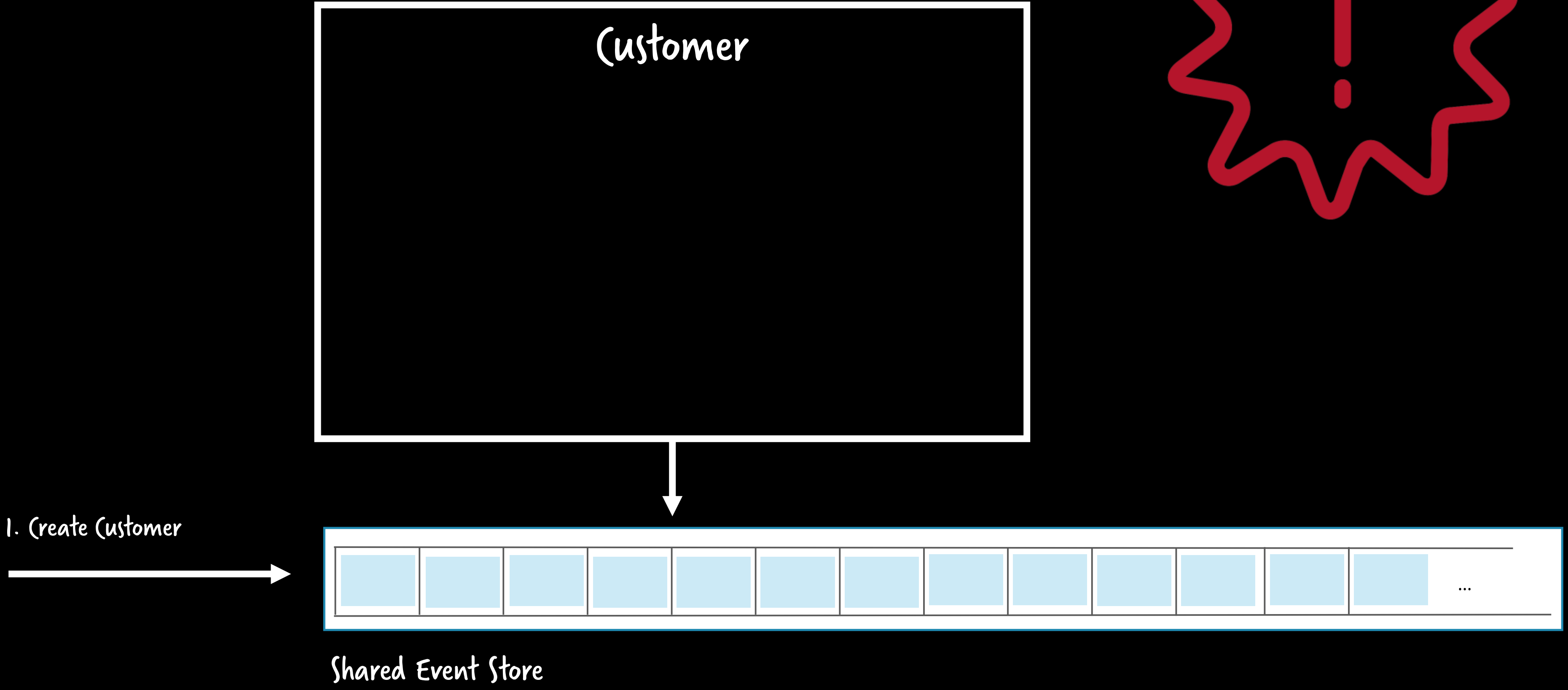
Messaging as the Single Source of Truth

Ben Stopford

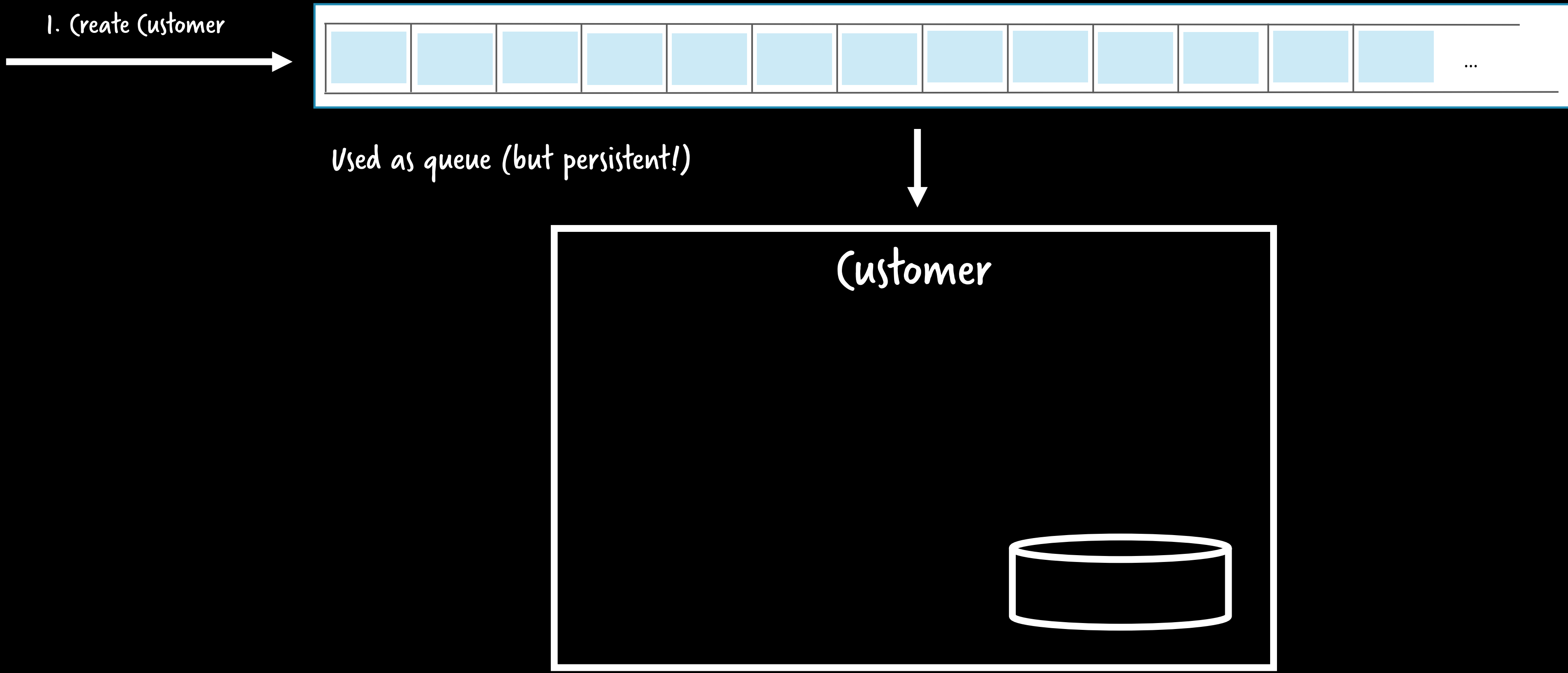
August 2, 2017

This post discusses Event Sourcing in the context of Apache Kafka®, examining the need for a

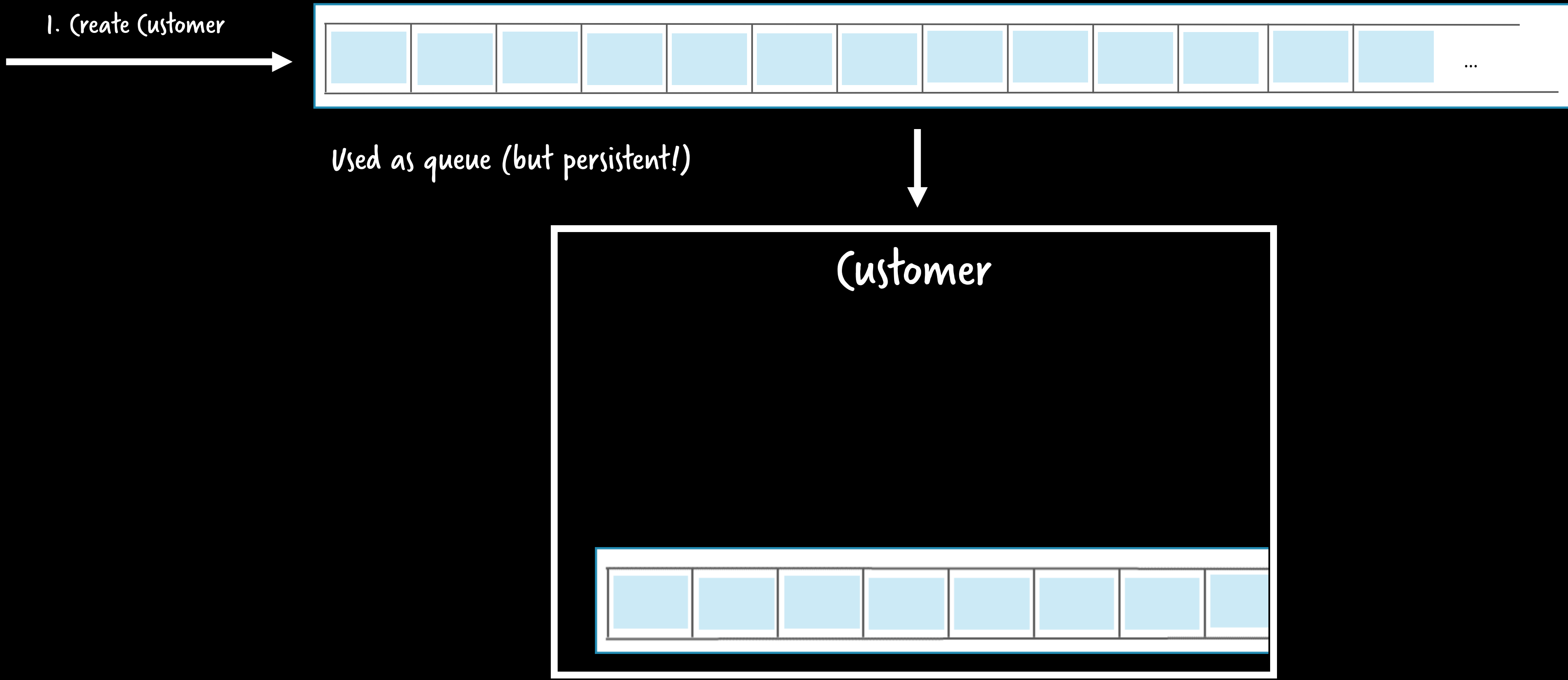
Merge messaging and event store



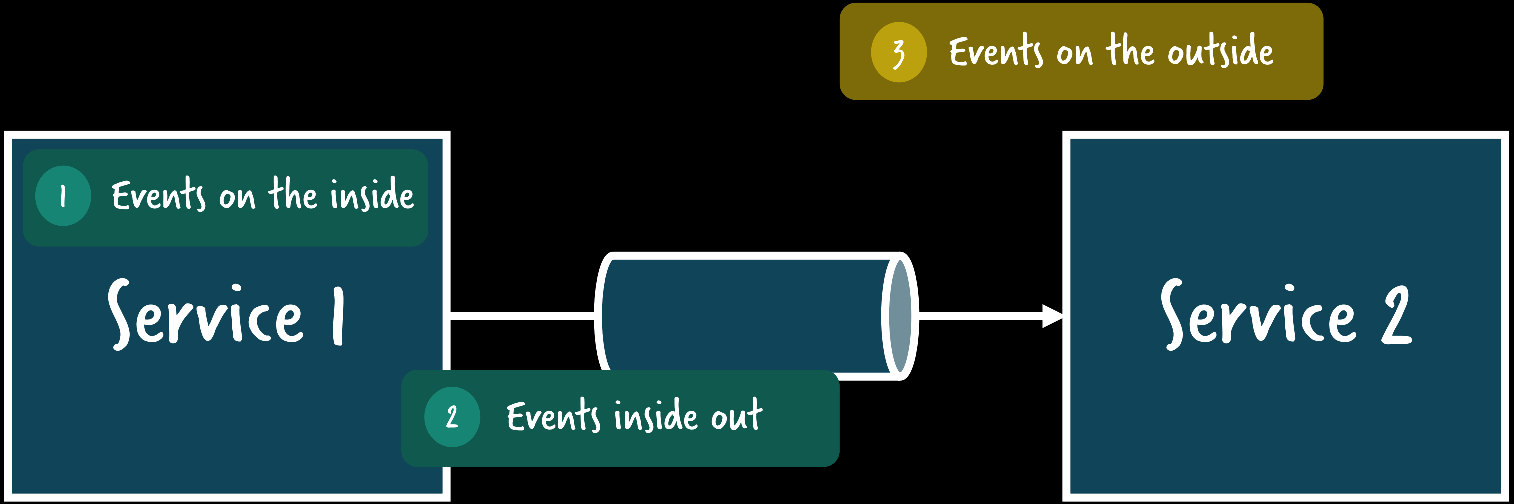
Kafka as transport



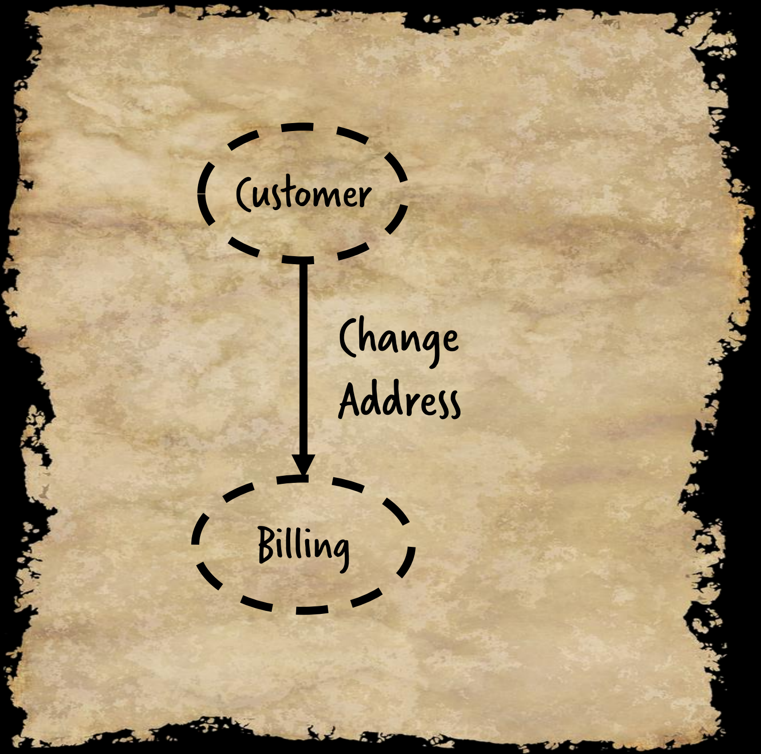
Kafka as transport



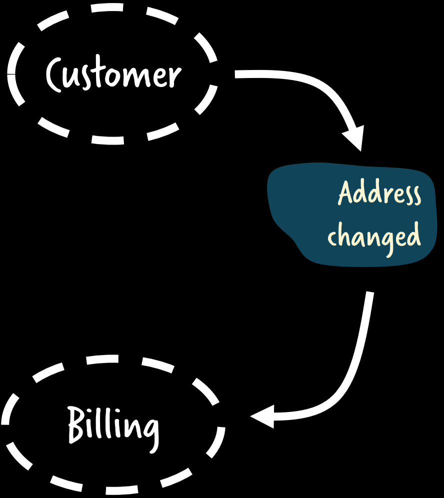
Agenda


3 Events on the outside
1 Events on the inside
Service 1
2 Events inside out
Service 2

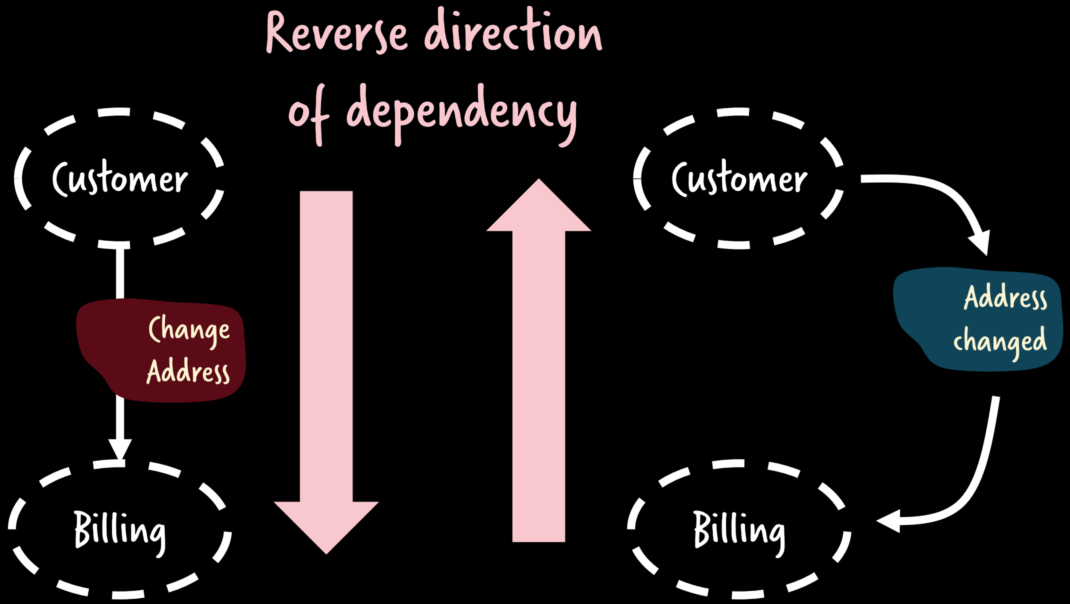
once upon a time



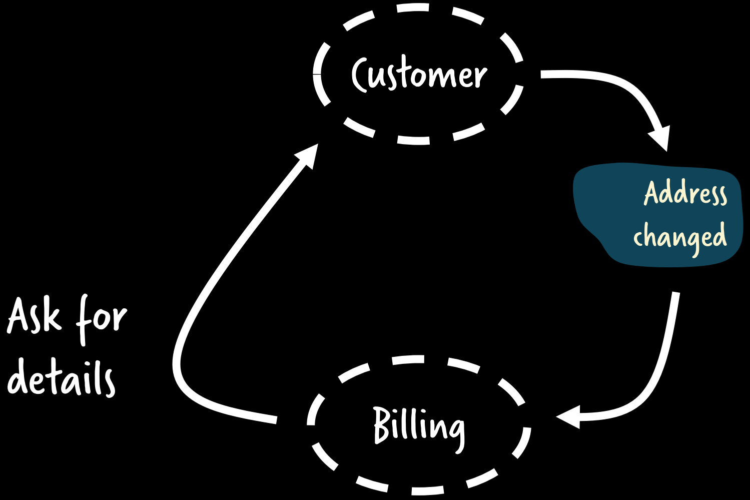
Event Notification



Event Notification

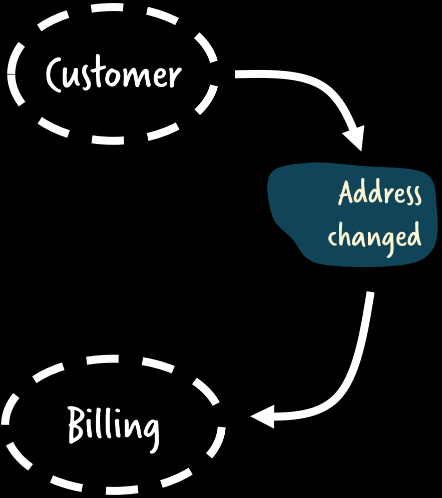


Event Notification



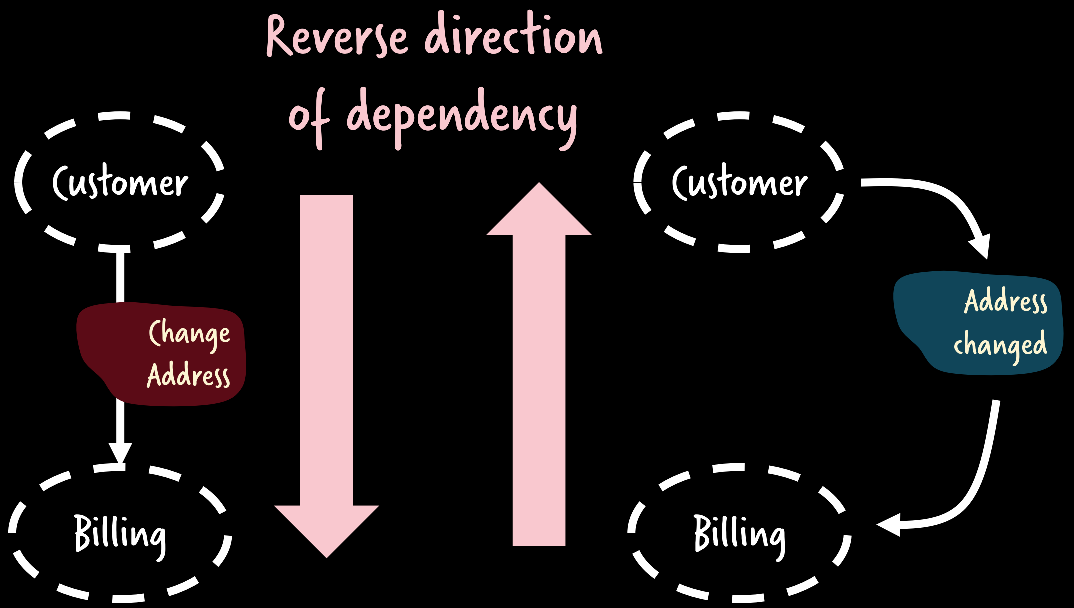
```
AdressChanged
{
  customerId: 42
}
```

Event-carried State Transfer

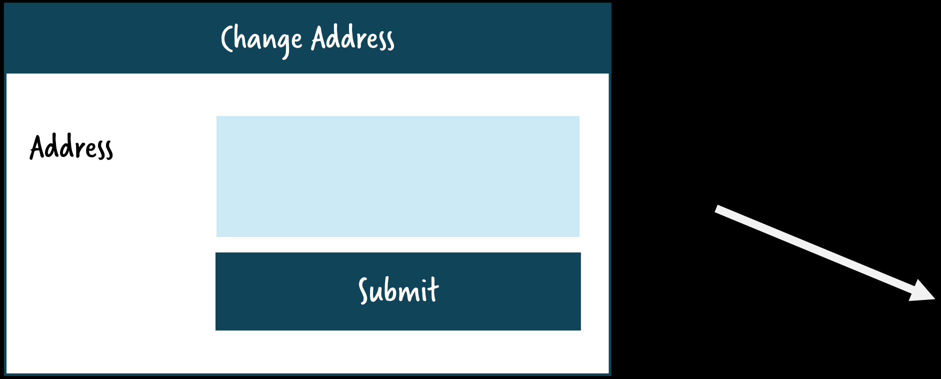


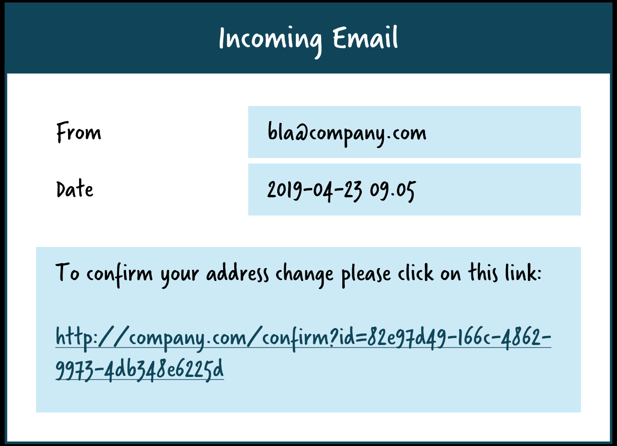
```
AddressChanged
{
  customerId: 42,
  oldAddress: ...
  newAddress: ...
}

AddressChanged
{
  customerId: 42,
  address: ...
}

CustomerChanged
{
  customerId: 42,
  status: A,
  address:
  ...,
}

CustomerMoved
{
...,
```

This decision is complex



Example



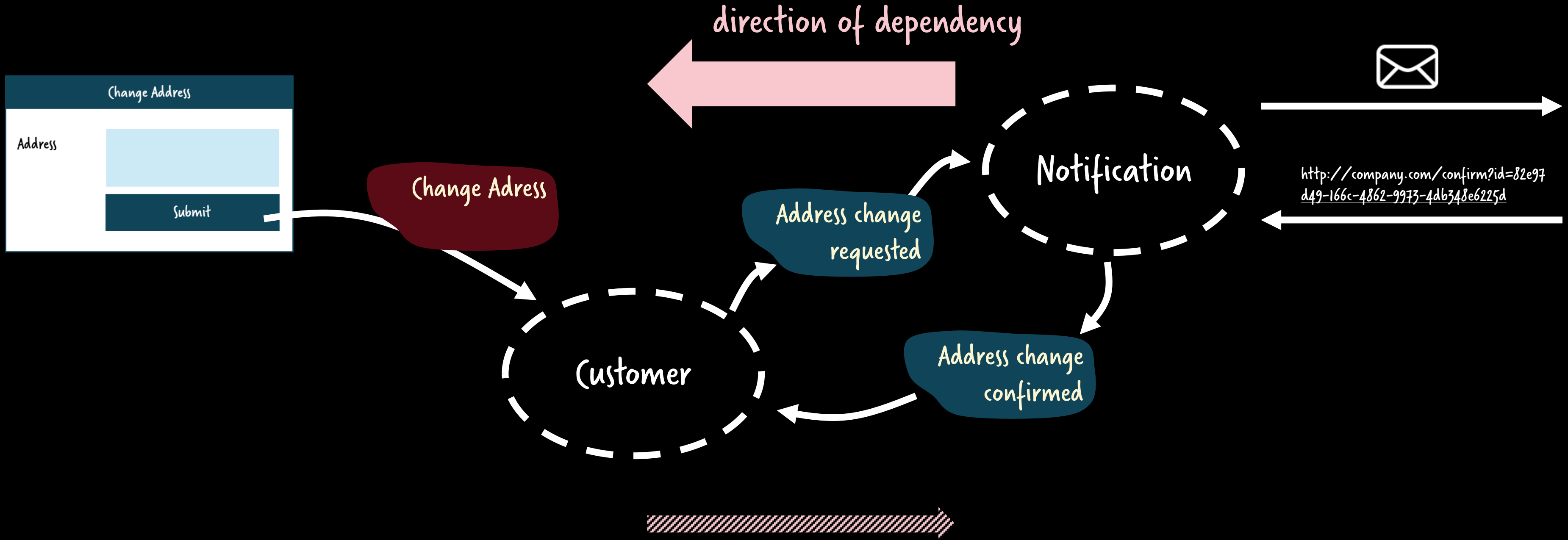

Incoming Email

From	bla@company.com
Date	2019-04-23 09.05

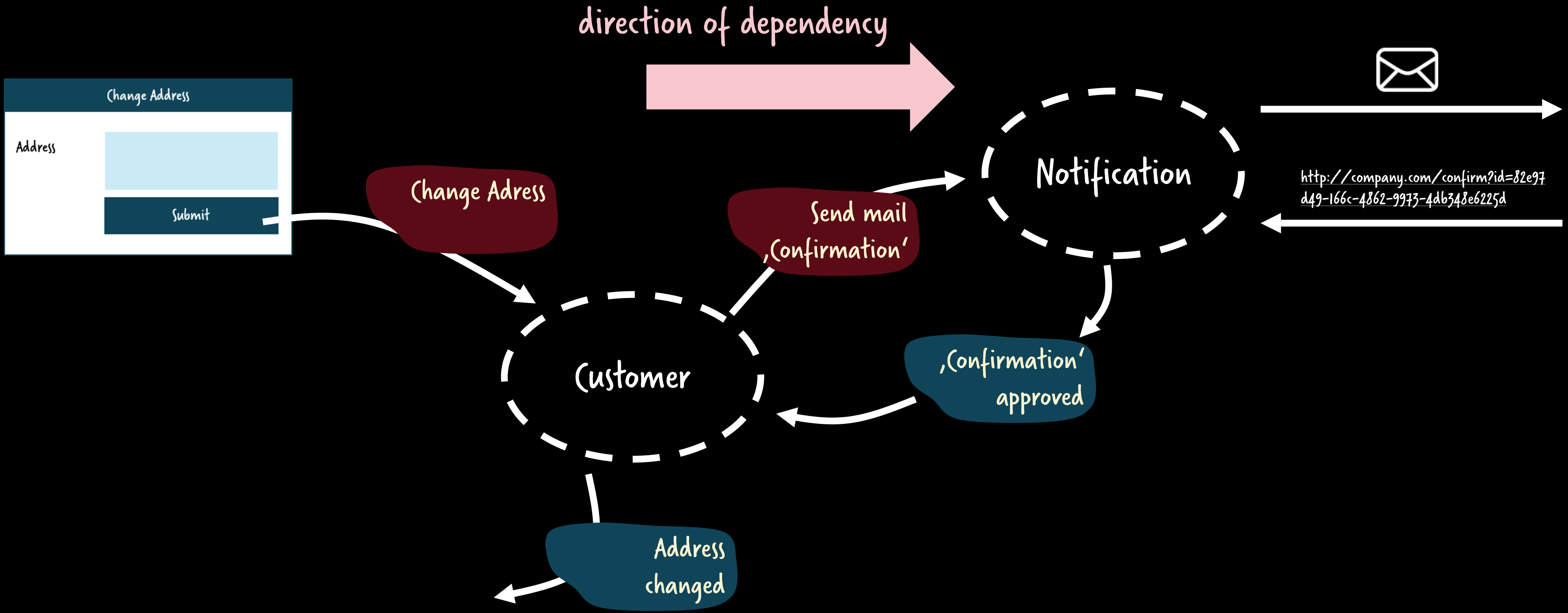
To confirm your address change please click on this link:

http://company.com/confirm?id=82e97d49-166c-4862-9973-4db348e6225d

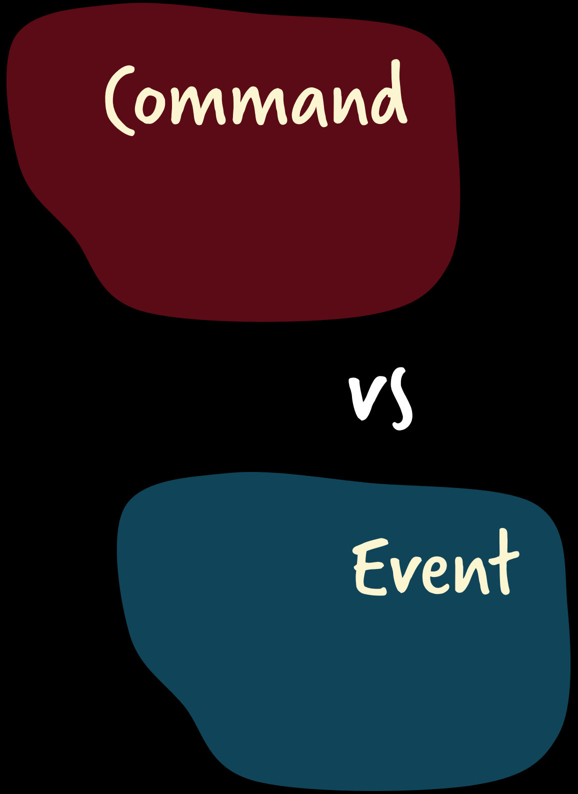
Example


direction of dependency
Change Address
Address
Submit
Change Adress
Customer
Address change requested
Notification
Address change confirmed
http://company.com/confirm?id=82e97d49-166c-4862-9973-4db348e6225d

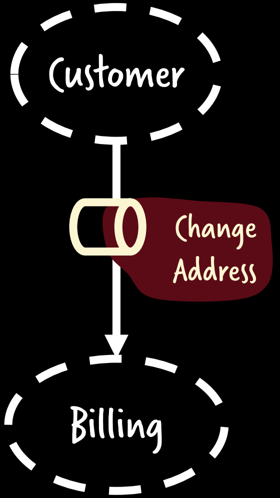
Example

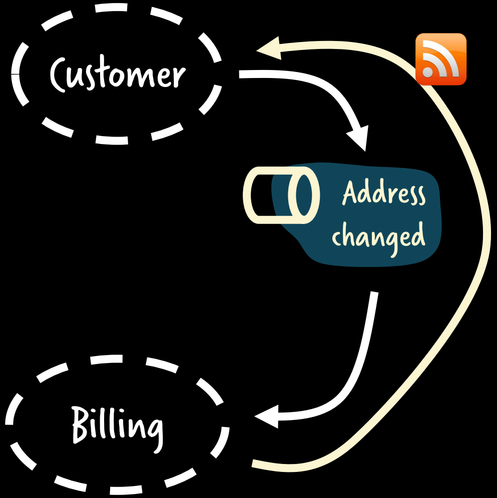

direction of dependency
Change Address
Address
Submit
Change Adress
Customer
Send mail ‚Confirmation'
Notification
http://company.com/confirm?id=82e97d49-166c-4862-9973-4db348e6225d
‚Confirmation' approved
Address changed

Challenge: Command vs. Event



It is NOT about communication protocols





It can be messaging, REST, whatever, ...

Manifold ways of transport

APACHE kafka®

webhooks

RabbitMQ

...

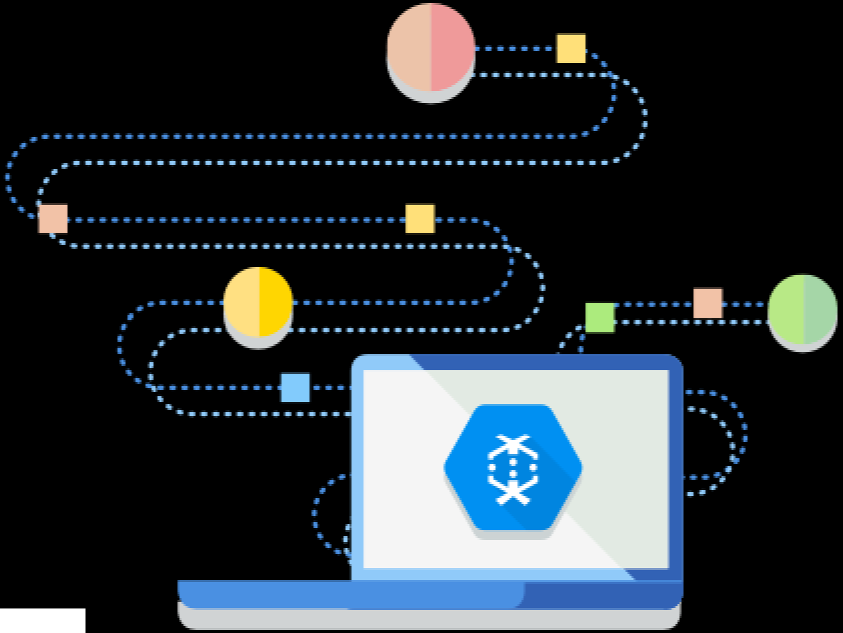
Manifold ways of transport

amazon Kinesis

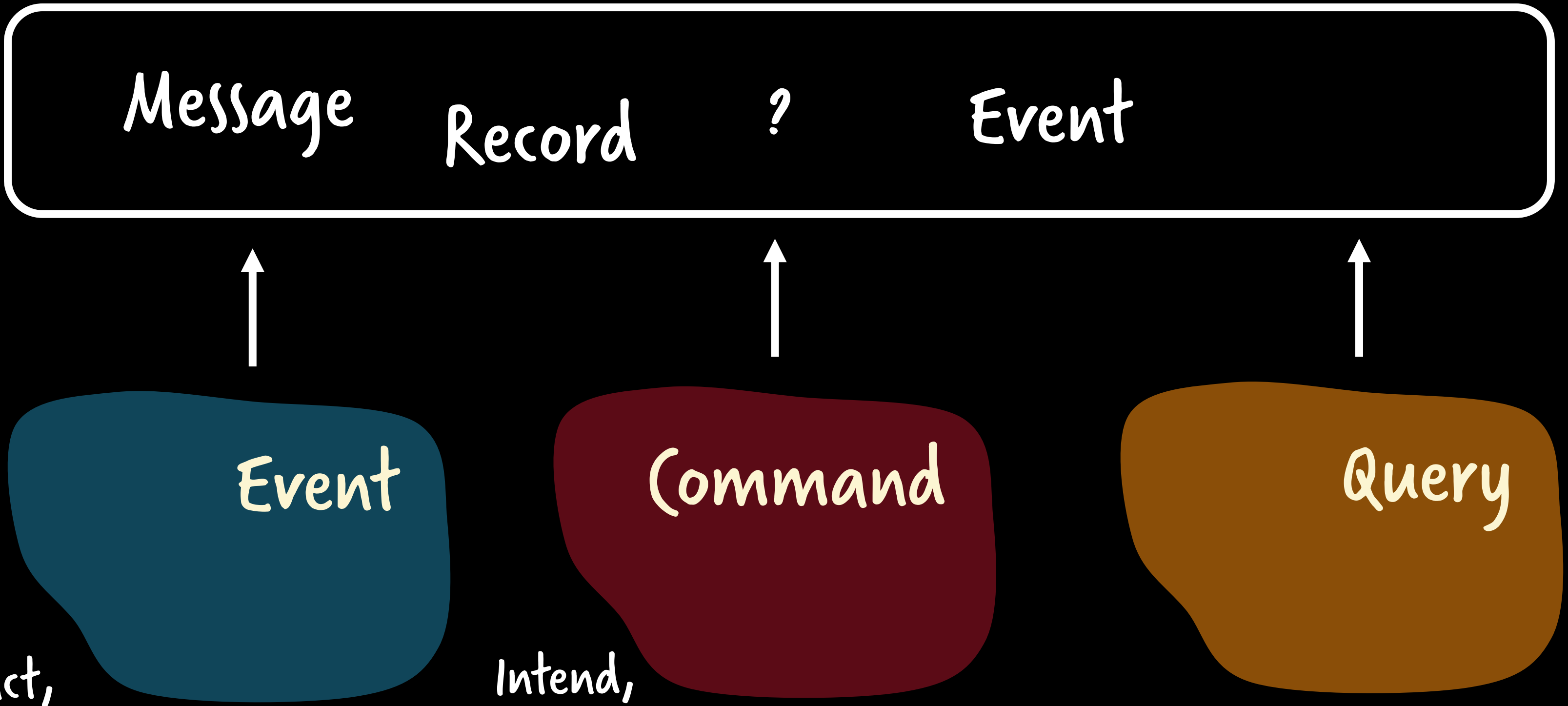
Azure Service Bus

Azure Event Grid

Azure Event Hub

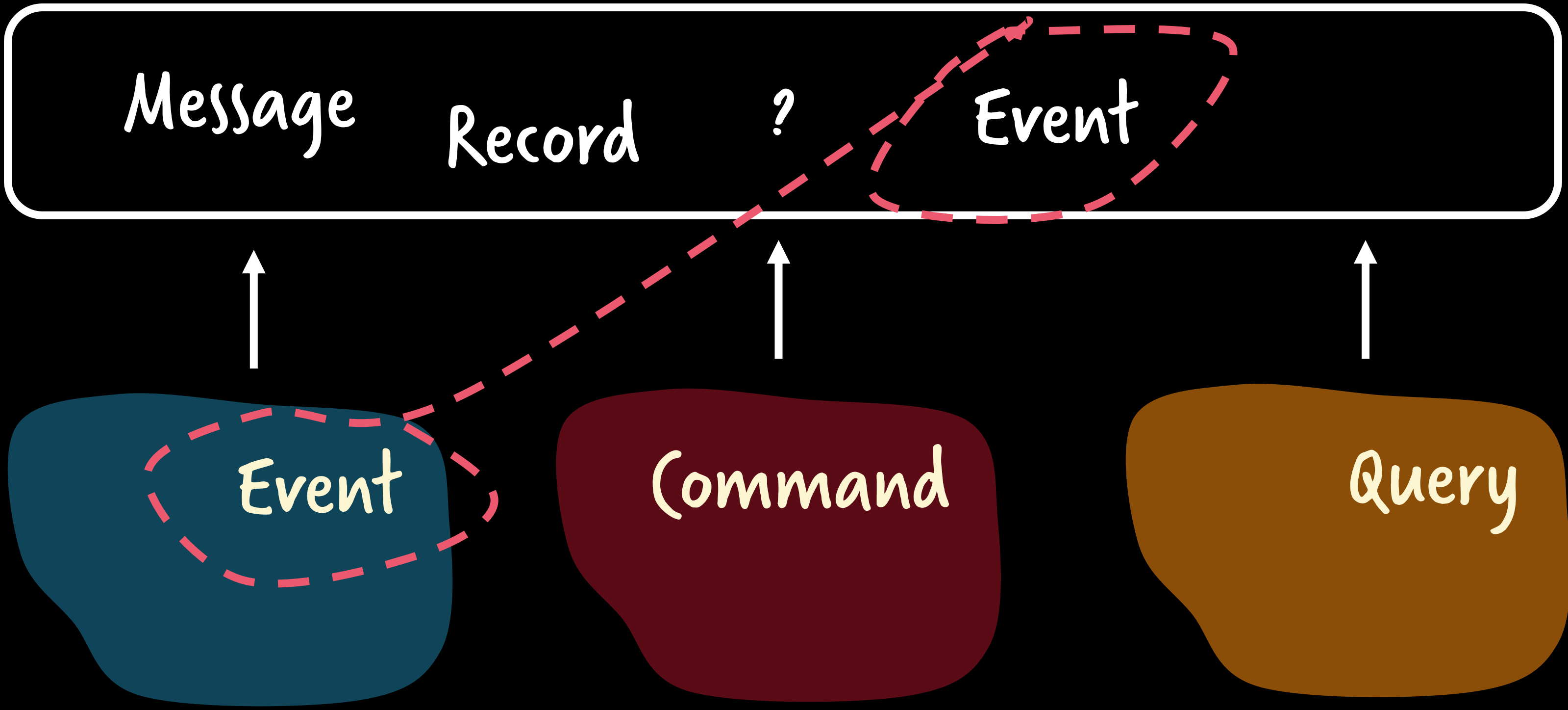


...



Fact,
happened in the past,
immutable

Intend,
Want s.th. to happen,
The intention itself is a fact


Message
Record
?
Event
Event
Command
Query

Commands in disguise

Wording of recipient

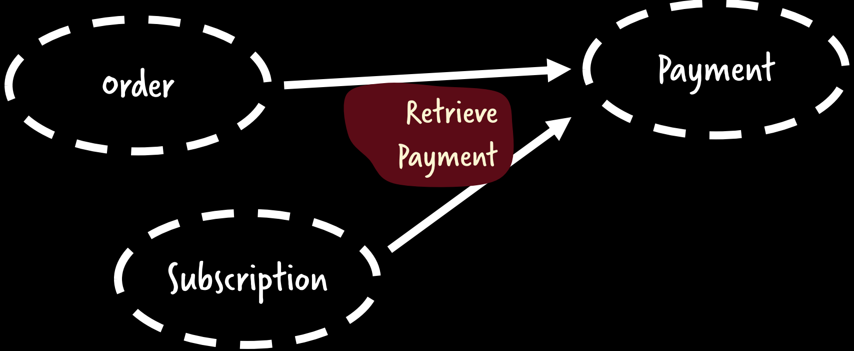
Send Message

The Customer Needs To Be Sent A Message To Confirm Address Change Event

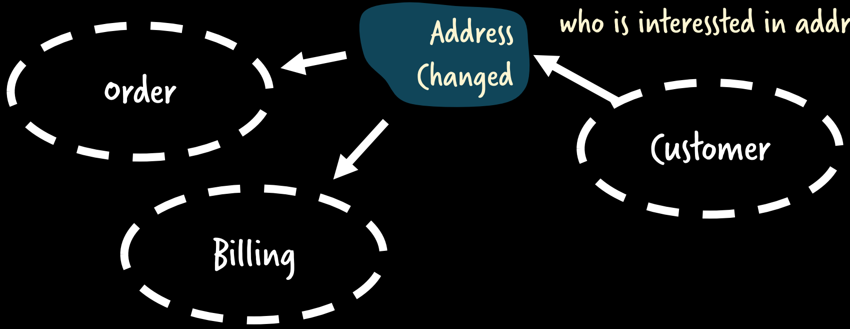
Wording of Sender

Examples

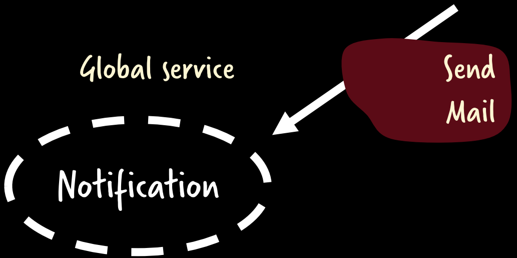
More general,
does not need to know
who is retrieving payments

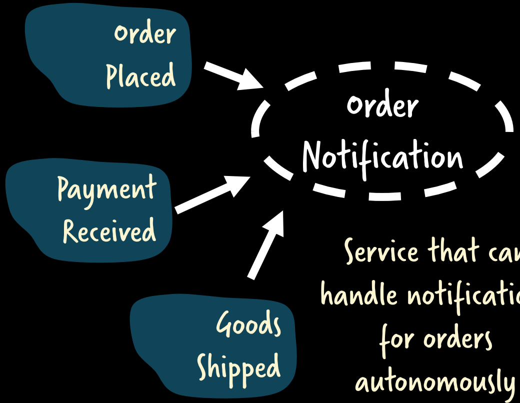


More general,
does not need to care about
who is interessted in address changes



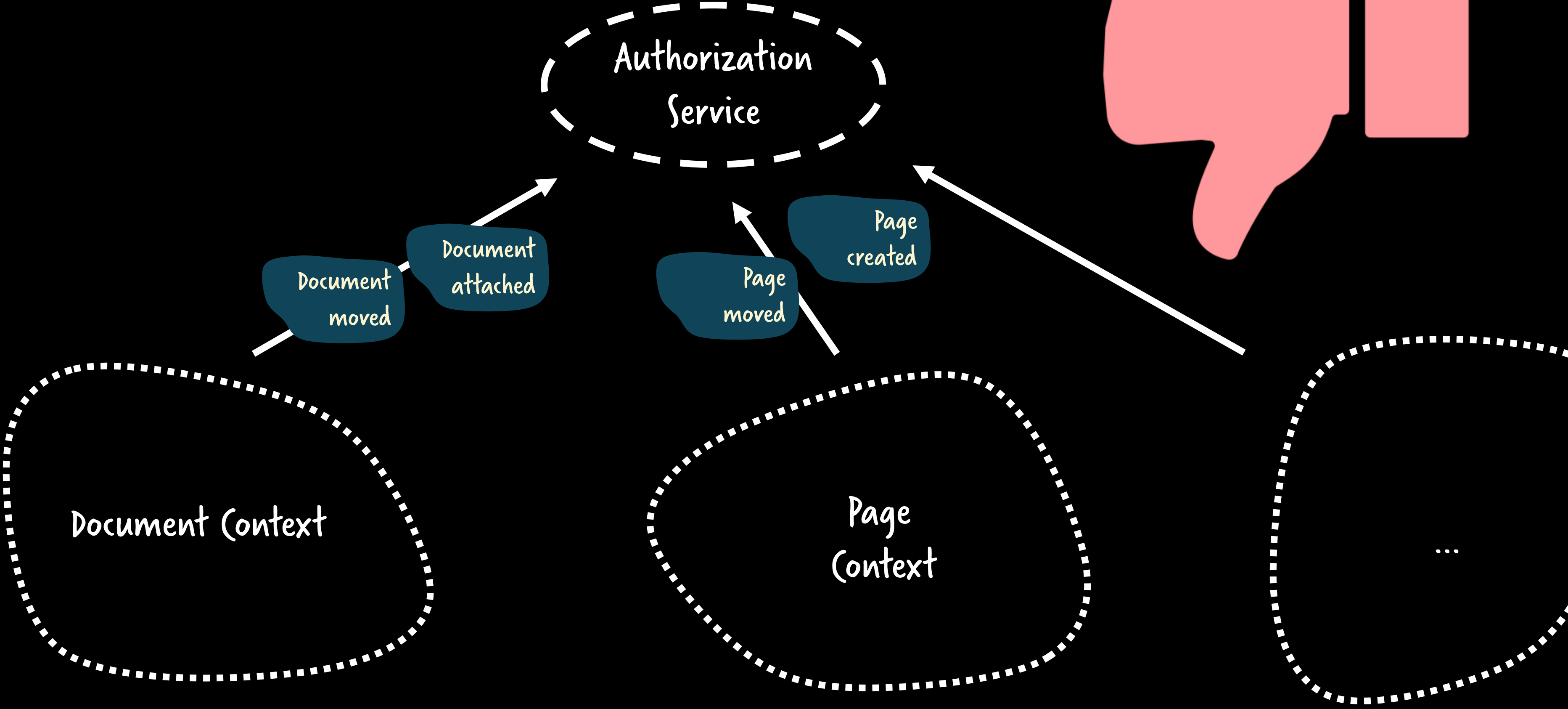
Global service



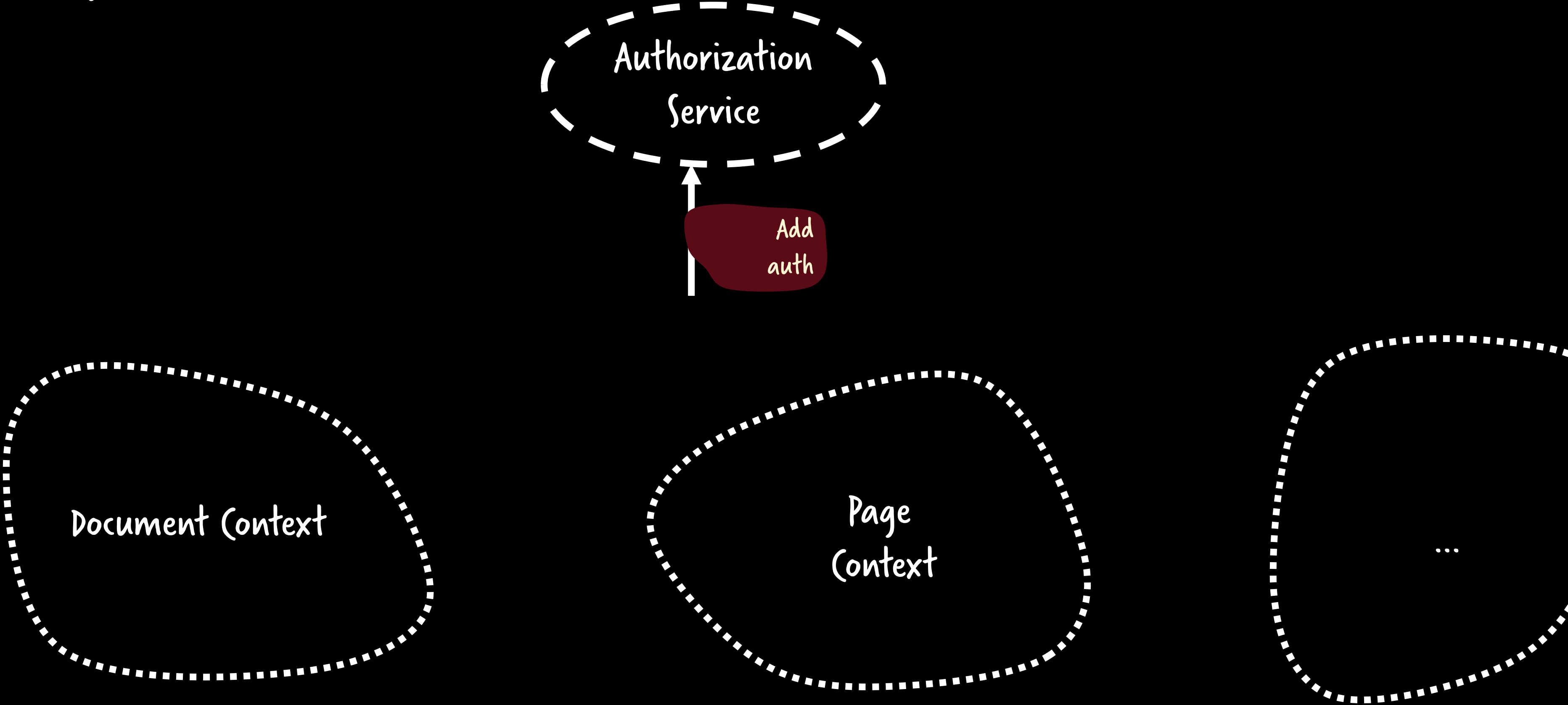


Service that can
handle notifications
for orders
autonomously

Distributed Monoliths

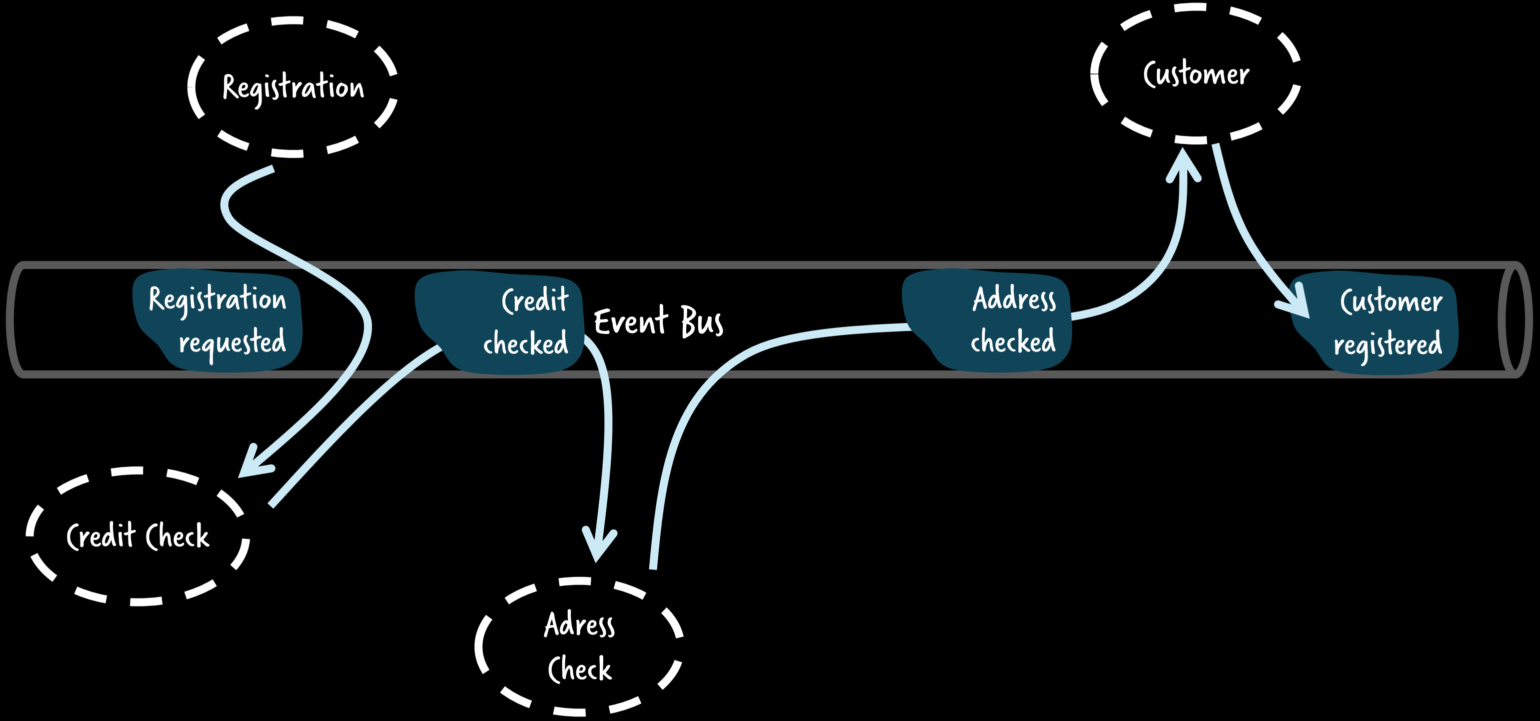


Define stable contract/API instead

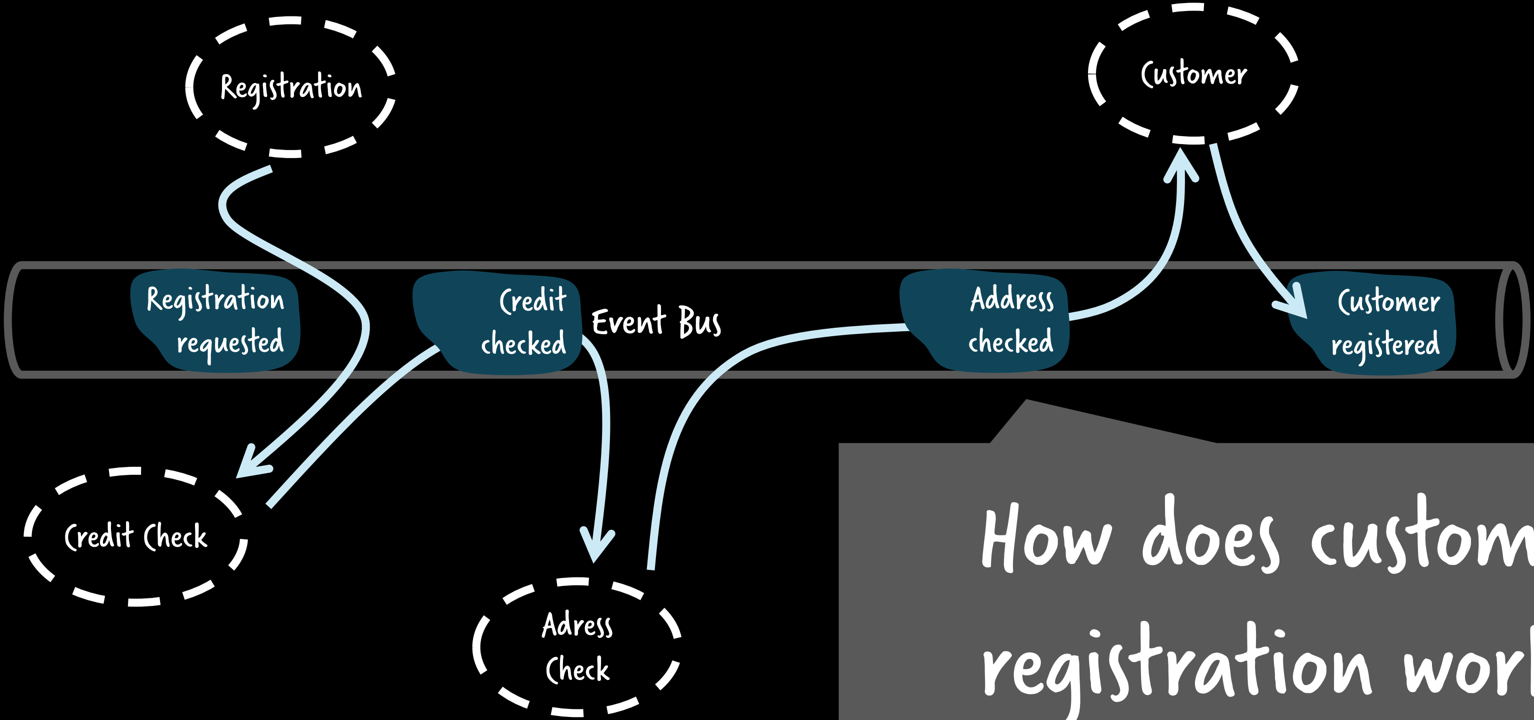


Next challenge:
Event chains

Event Chains



Event Chains



How does customer registration work?

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.

https://martinfowler.com/articles/201701-event-driven.html

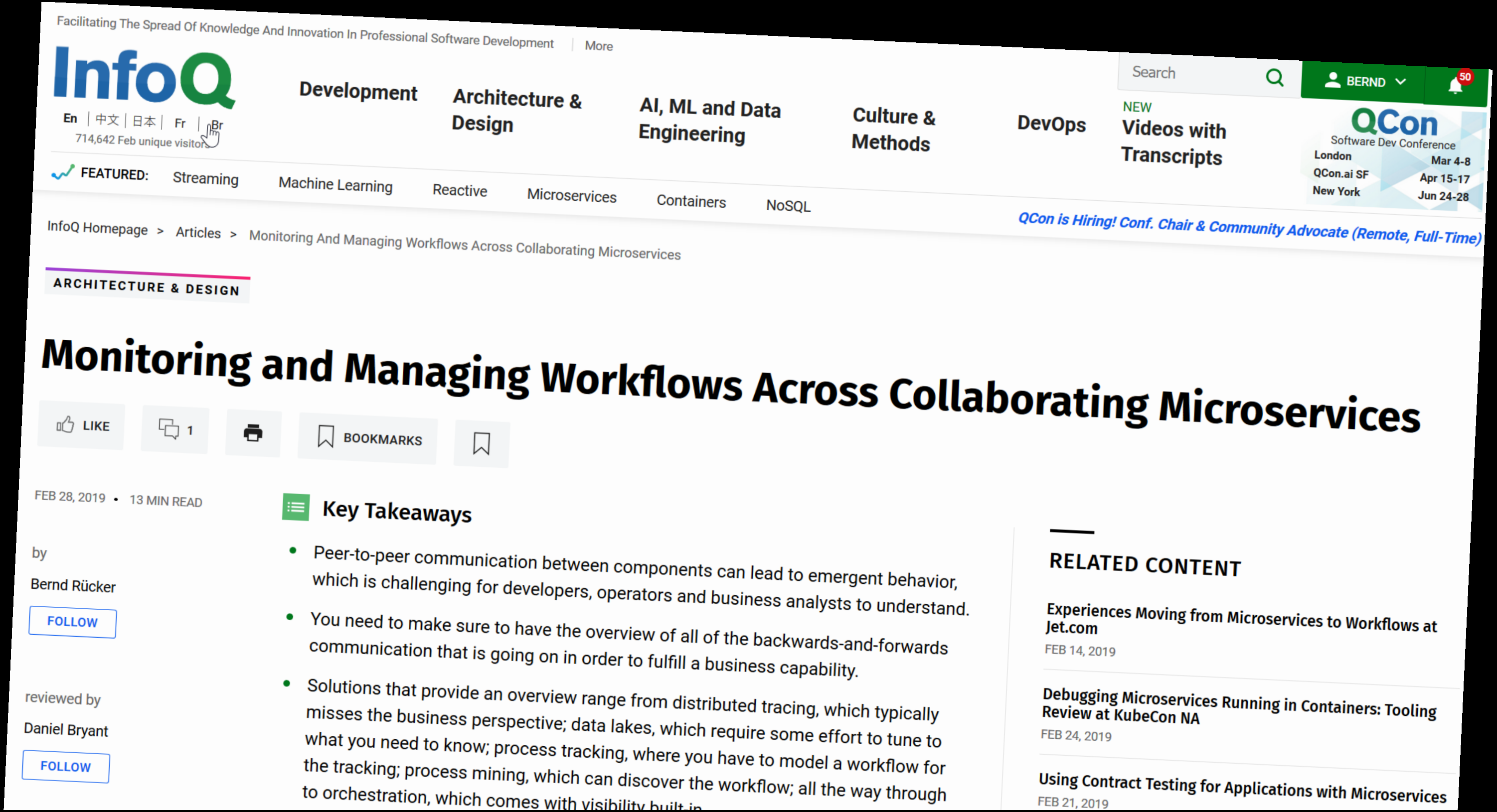
The danger is that it's very easy to make nicely decoupled systems with event notification, **without realizing that you're losing sight of that larger-scale flow**, and thus set yourself up for trouble in future years.

https://martinfowler.com/articles/201701-event-driven.html

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**.

https://martinfowler.com/articles/201701-event-driven.html

Monitoring Workflows Across Microservices



Facilitating The Spread Of Knowledge And Innovation In Professional Software Development | More

InfoQ

En | 中文 | 日本 | Fr | Br

714,642 Feb unique visitors

Development | Architecture & Design | AI, ML and Data Engineering | Culture & Methods | DevOps | NEW Videos with Transcripts

Search | BERND | 50

QCon Software Dev Conference: London Mar 4-8, QCon.ai SF Apr 15-17, New York Jun 24-28

FEATURED: Streaming | Machine Learning | Reactive | Microservices | Containers | NoSQL

InfoQ Homepage > Articles > Monitoring And Managing Workflows Across Collaborating Microservices

QCon is Hiring! Conf. Chair & Community Advocate (Remote, Full-Time)

ARCHITECTURE & DESIGN

Monitoring and Managing Workflows Across Collaborating Microservices

LIKE | 1 | BOOKMARKS

FEB 28, 2019 • 13 MIN READ

by Bernd Rücker FOLLOW

reviewed by Daniel Bryant FOLLOW

Key Takeaways

- Peer-to-peer communication between components can lead to emergent behavior, which is challenging for developers, operators and business analysts to understand.
- You need to make sure to have the overview of all of the backwards-and-forwards communication that is going on in order to fulfill a business capability.
- Solutions that provide an overview range from distributed tracing, which typically misses the business perspective; data lakes, which require some effort to tune to what you need to know; process tracking, where you have to model a workflow for the tracking; process mining, which can discover the workflow; all the way through to orchestration, which comes with visibility built in

RELATED CONTENT

Experiences Moving from Microservices to Workflows at Jet.com
FEB 14, 2019

Debugging Microservices Running in Containers: Tooling Review at KubeCon NA
FEB 24, 2019

Using Contract Testing for Applications with Microservices
FEB 21, 2019

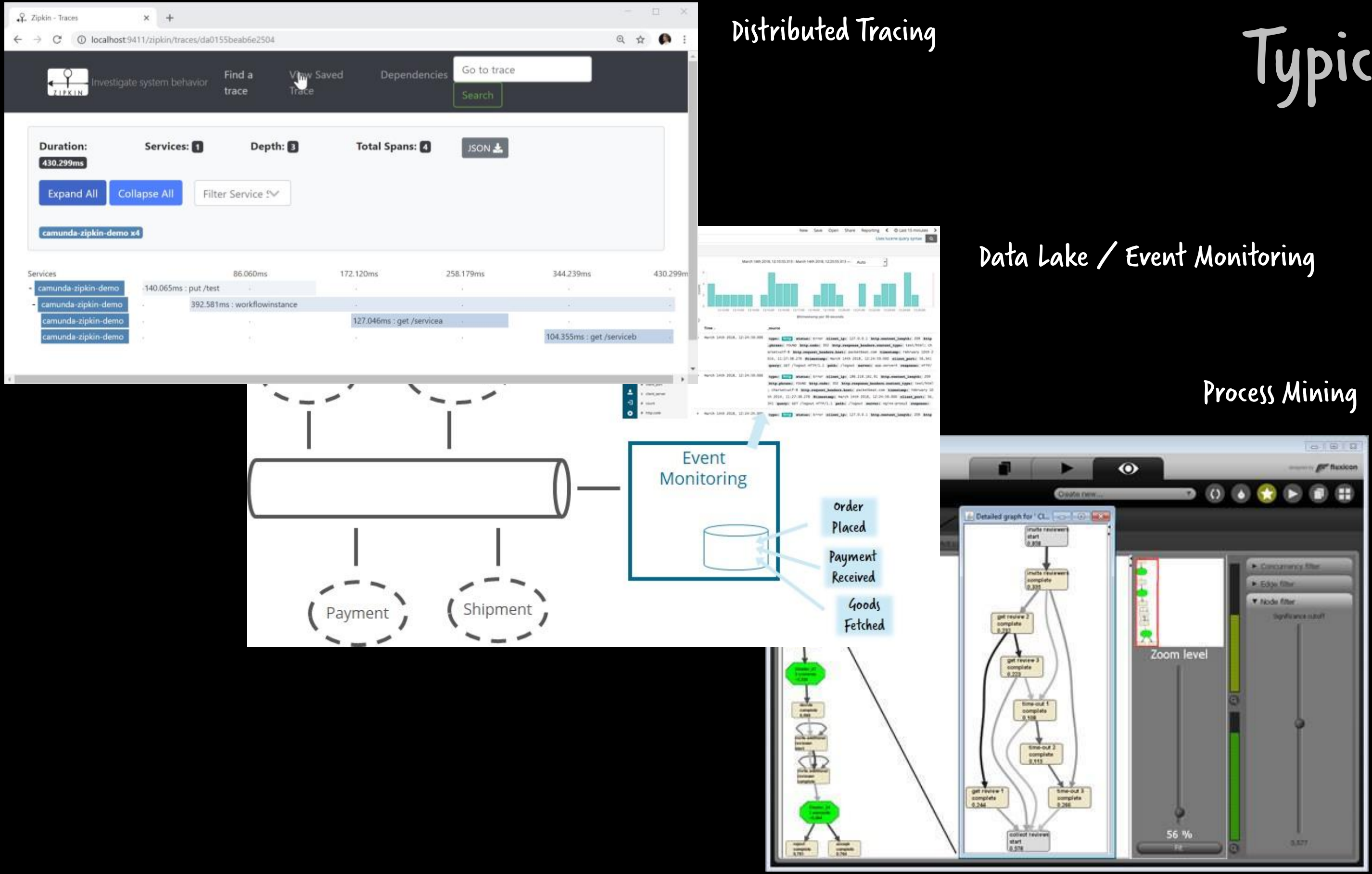
https://www.infoq.com/articles/monitor-workflow-collaborating-microservices

Typical approaches

Distributed Tracing

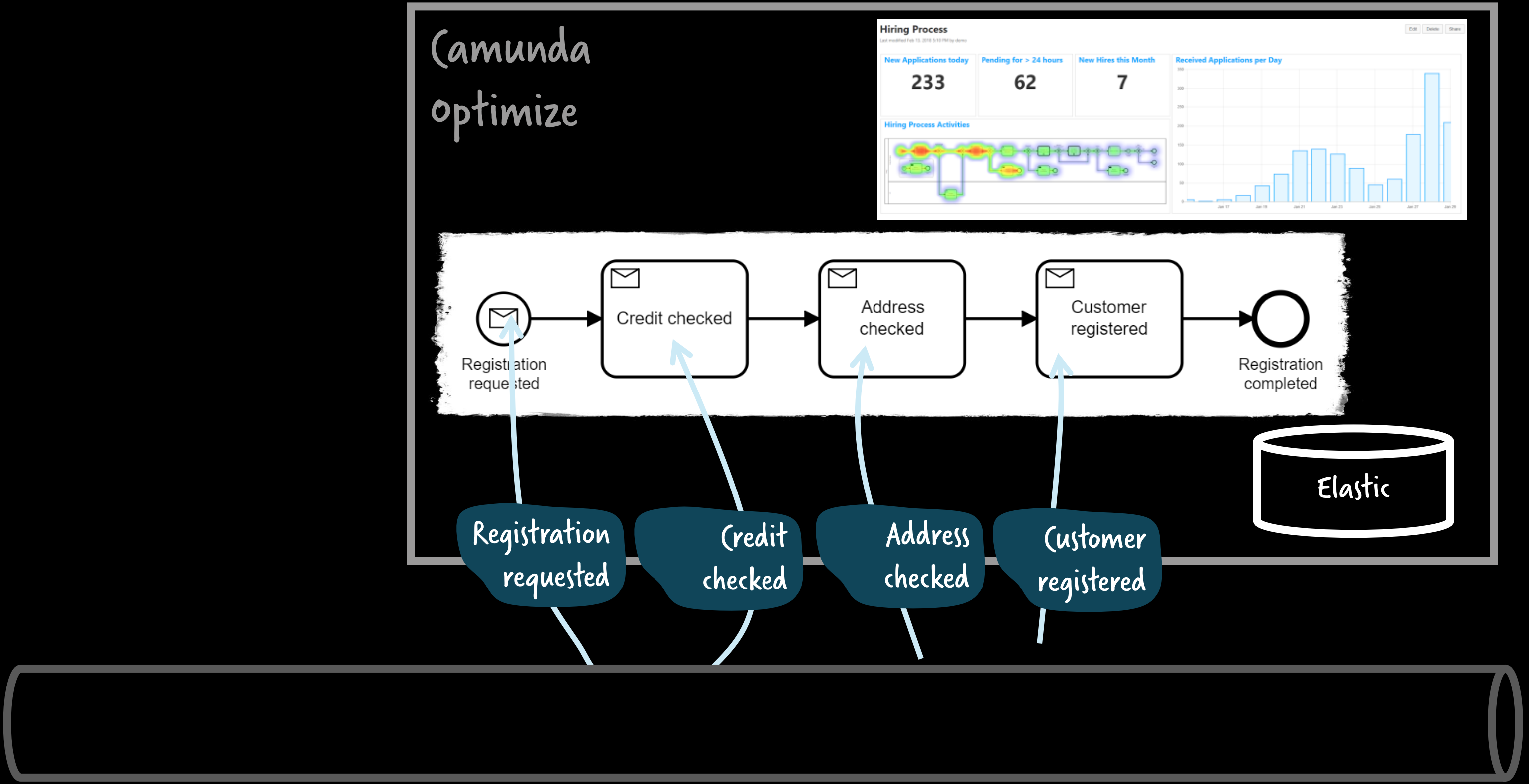
Data Lake / Event Monitoring

Process Mining

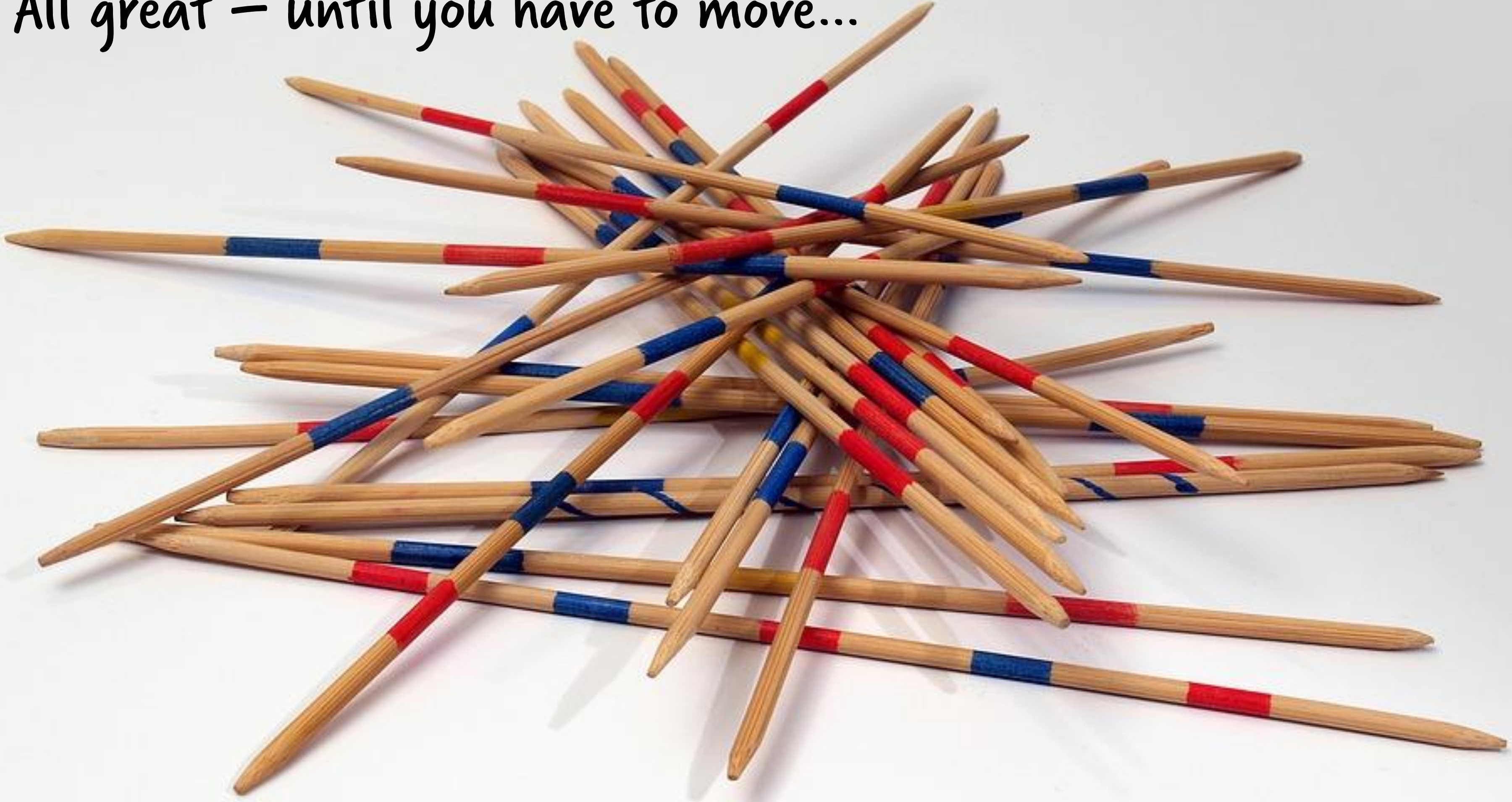


@berndruecker

What we currently build with customers...

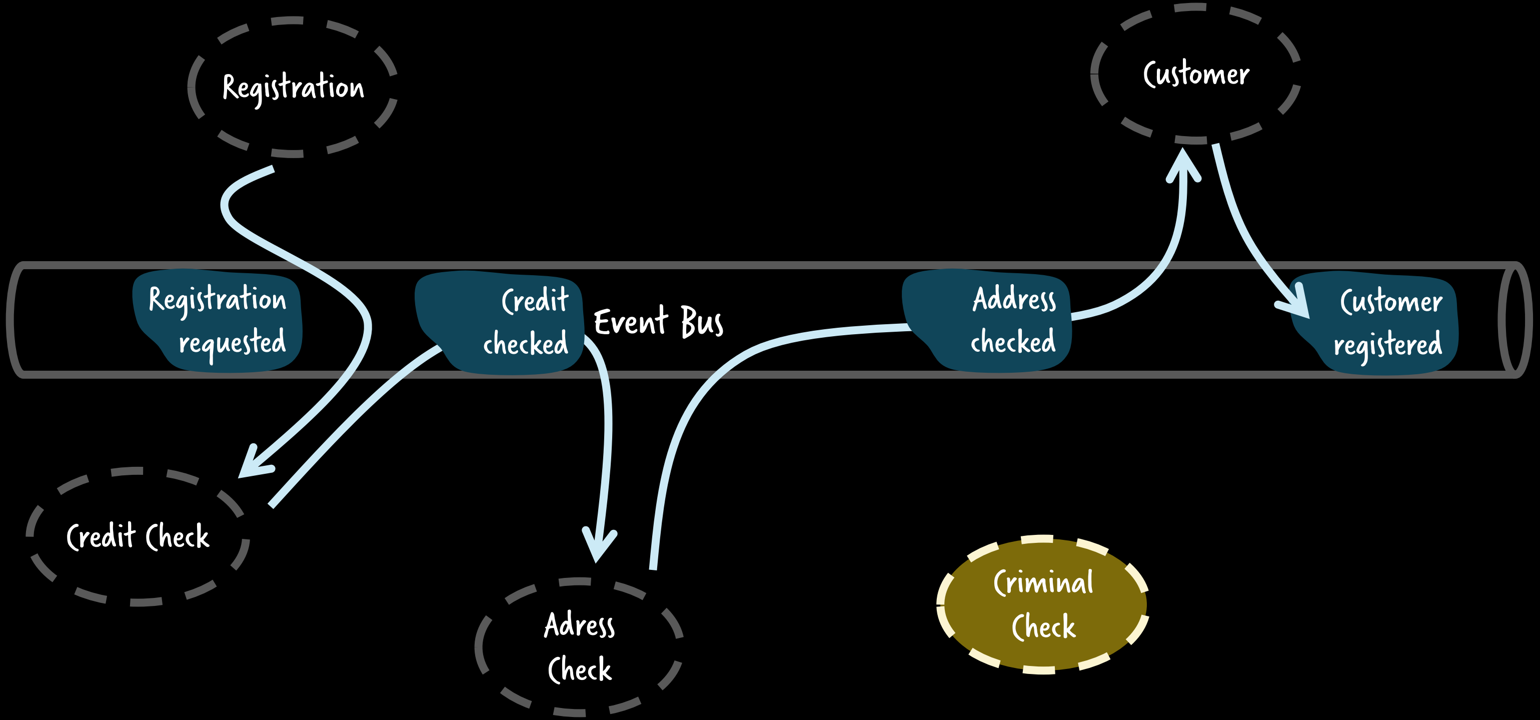


All great – until you have to move...

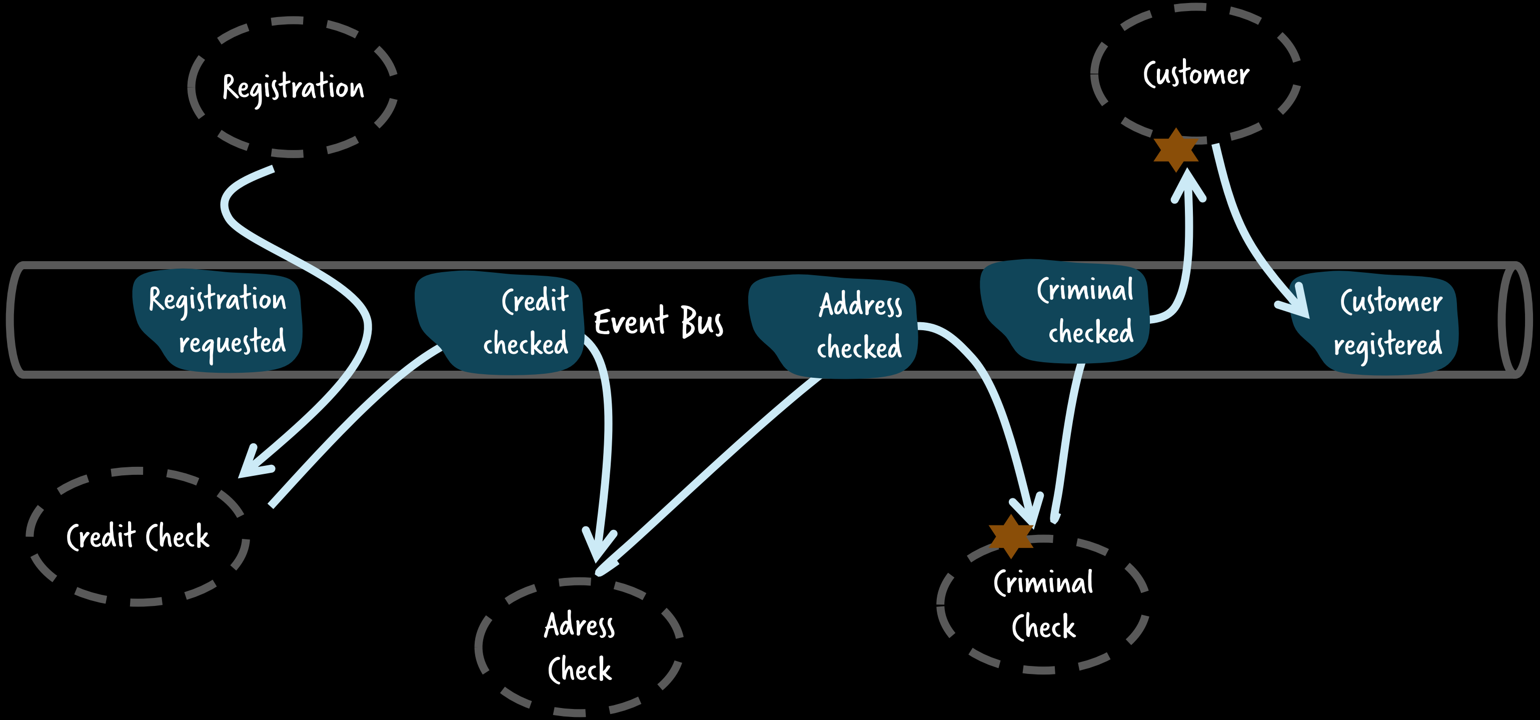


Keep it stable, just move that one stick down there to the top.
How hard can it be?

Changes required for an additional check

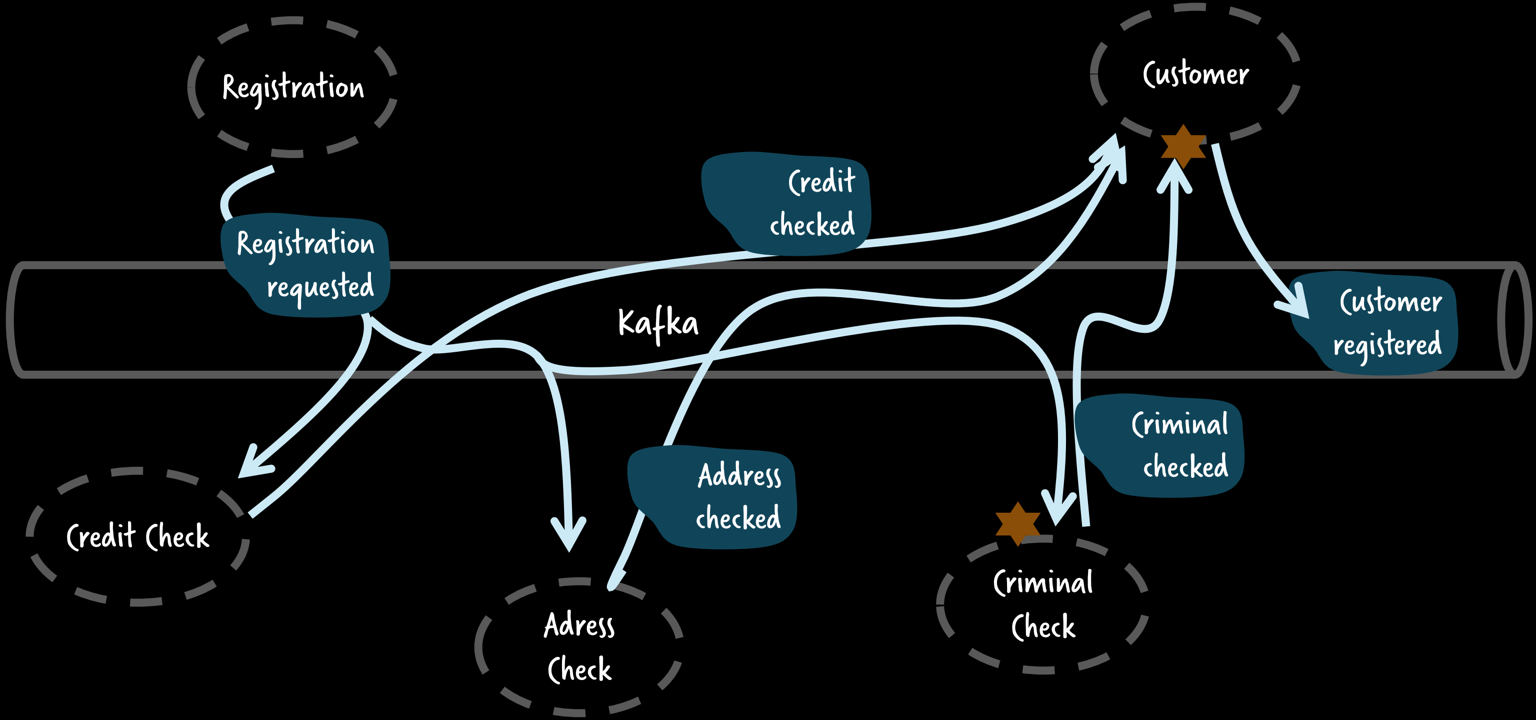


Changes required for an additional check



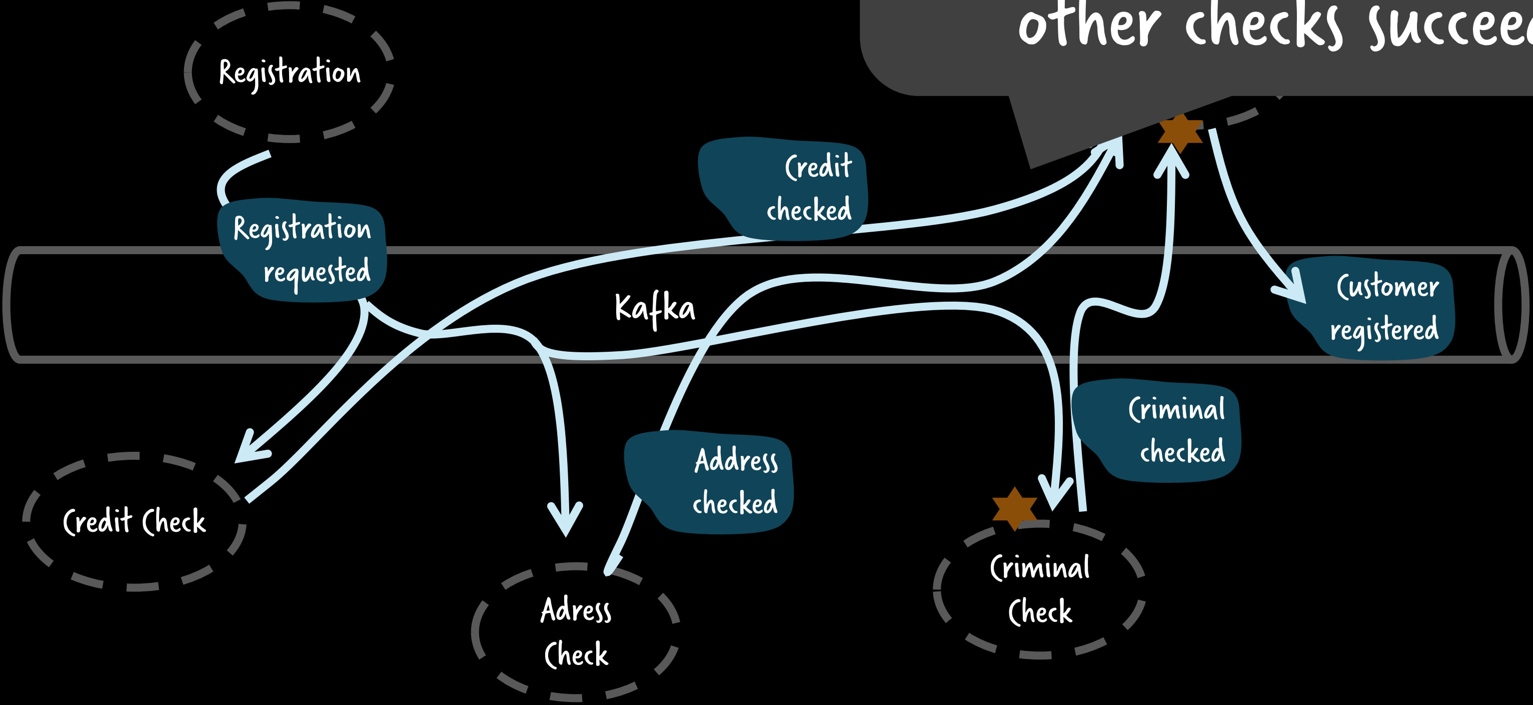
@berndruecker

Alternative flow



Alternative flow

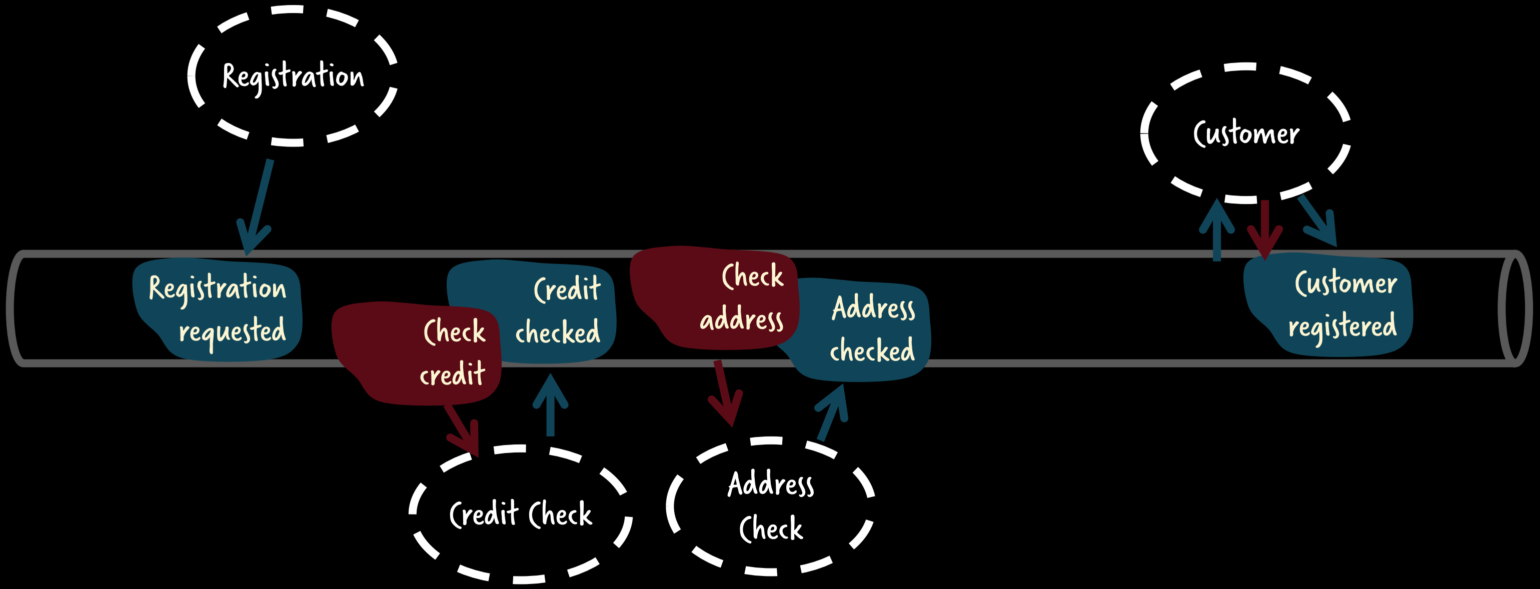
„Credit checks got more expensive, do that only if all other checks succeed"



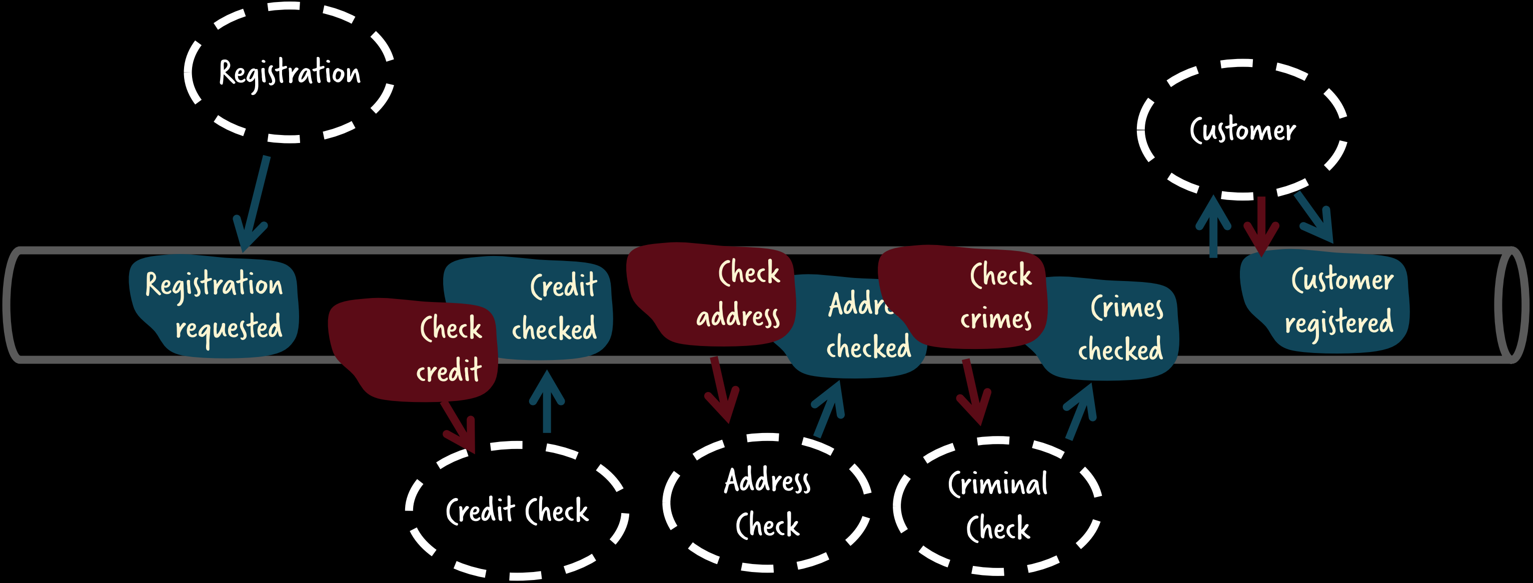
What we wanted

vs. what we got

orchestration



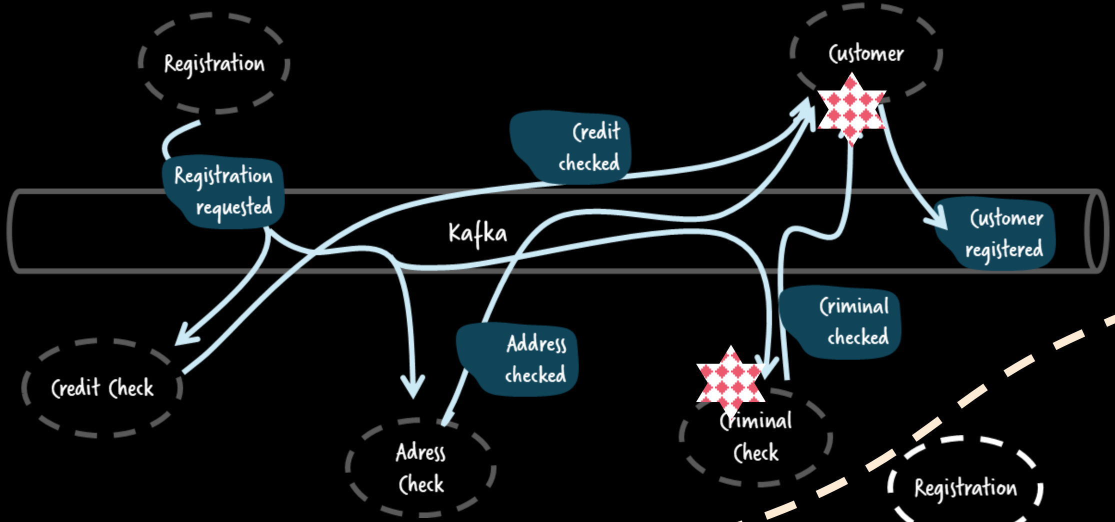
Changes...


Registration
Customer
Registration requested
Check credit
Credit checked
Check address
Addre
checked
Check crimes
Crimes checked
Customer registered
Credit Check
Address Check
Criminal Check

Coupling myths

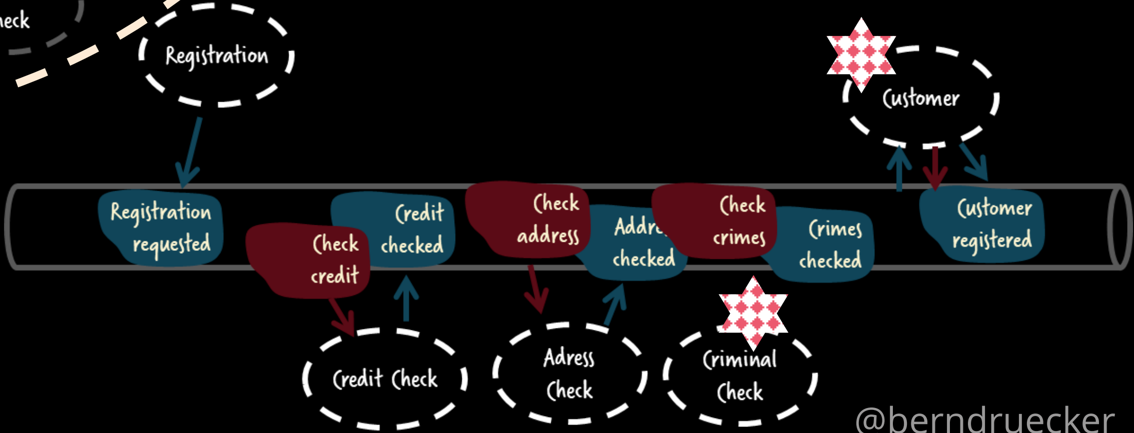
Comparison

See also https://www.infoworld.com/article/3391592/how-to-tame-event-driven-microservices.html

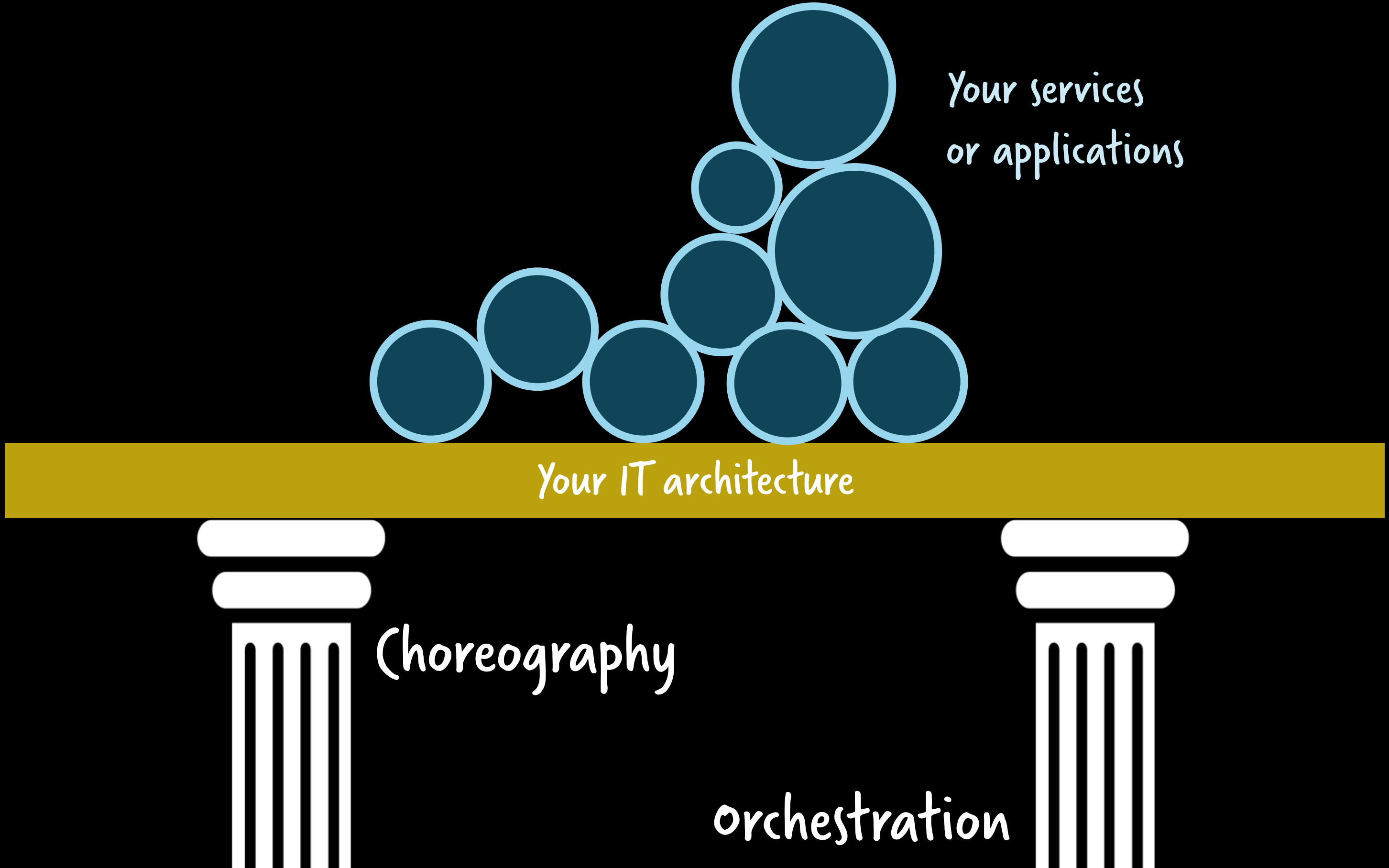


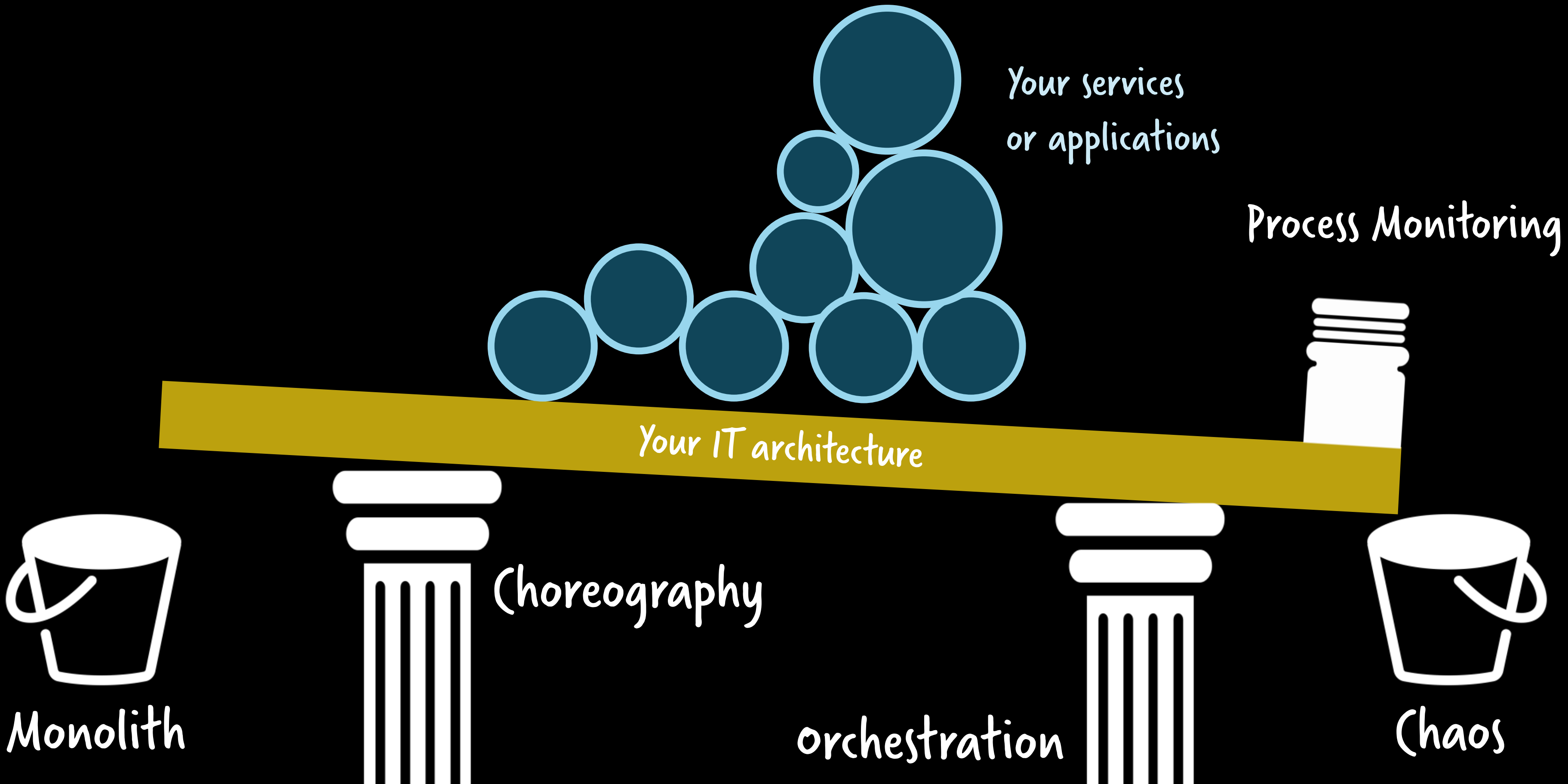
2 changes
criminal check can be deployed first

2 changes,
criminal check can be deployed first

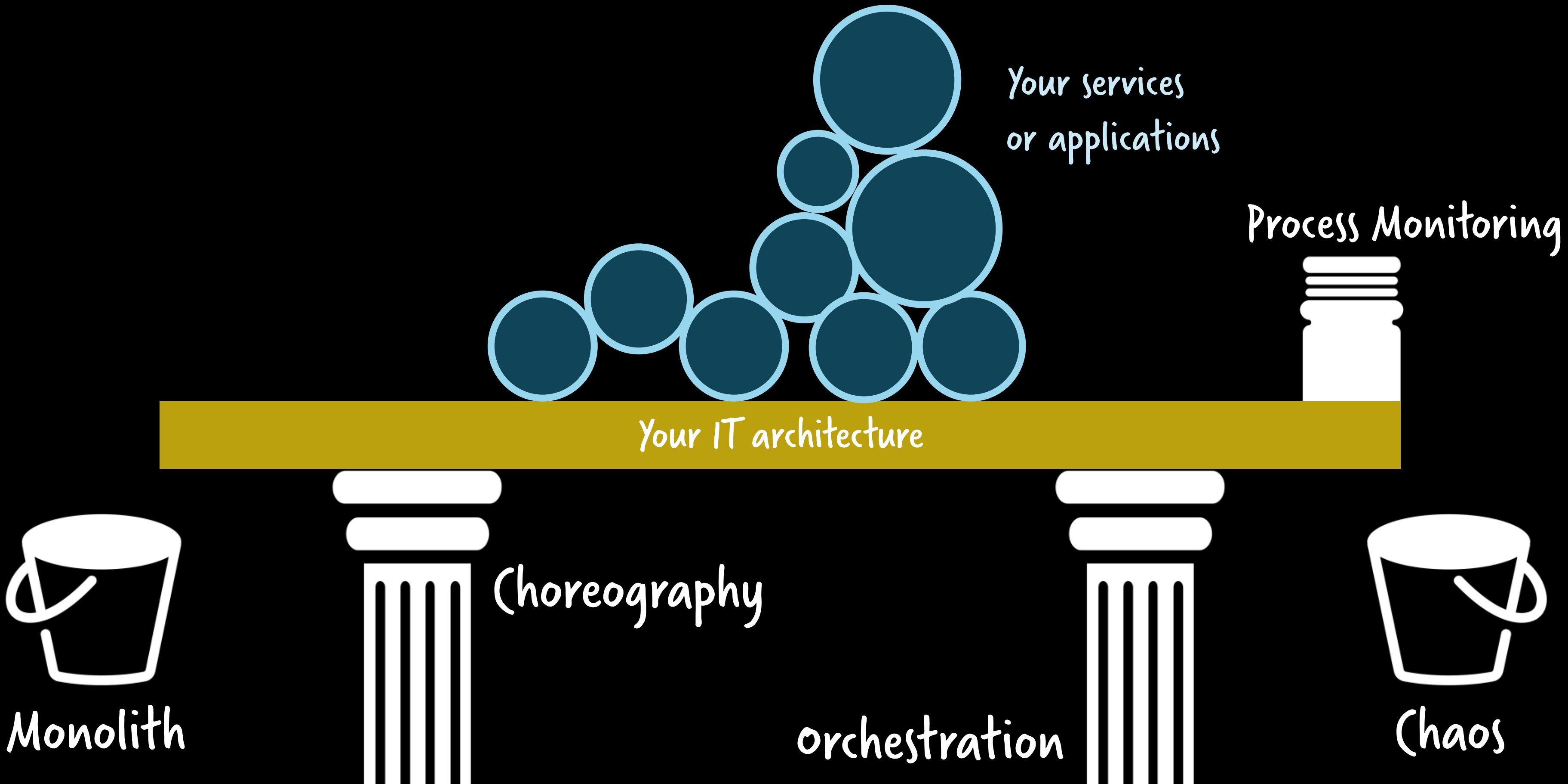


@berndruecker

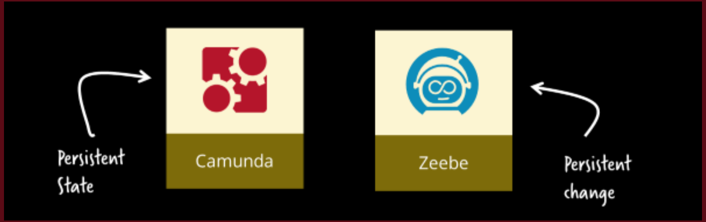

Your services or applications
Your IT architecture
Choreography
Orchestration


Your services
or applications
Process Monitoring
Your IT architecture
Choreography
Monolith
orchestration
Chaos

Balance choreography and orchestration

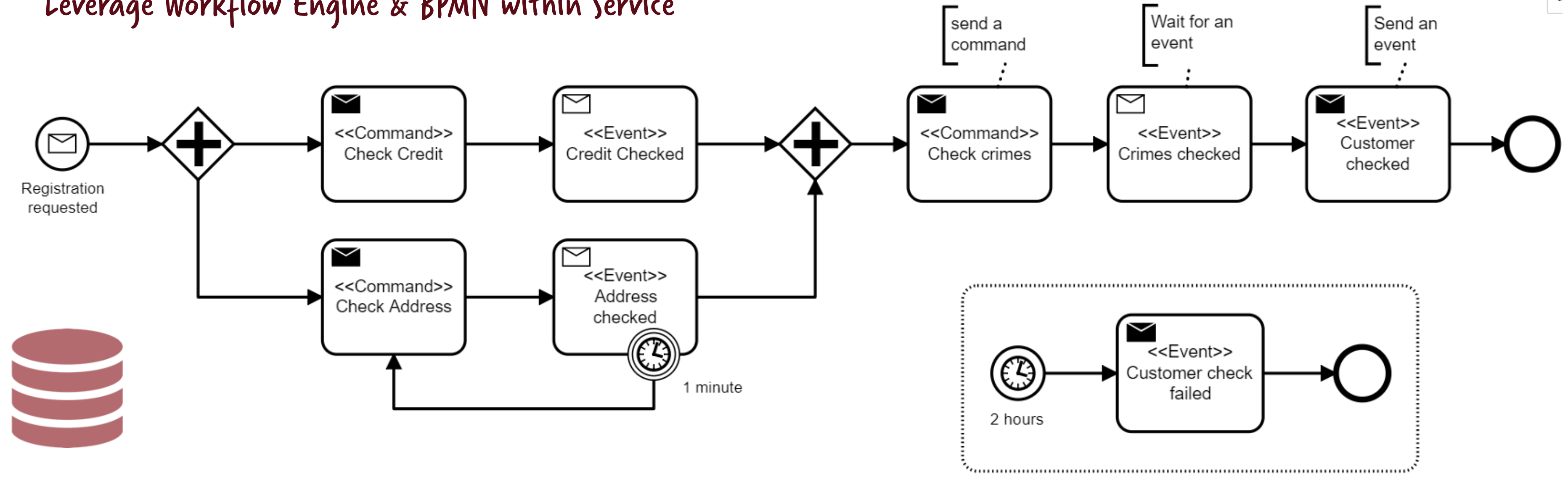


In my world...



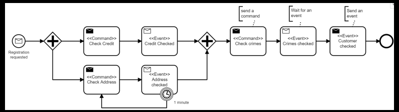
Customer on-boarding

Leverage Workflow Engine & BPMN within Service



Local orchestration

Customer



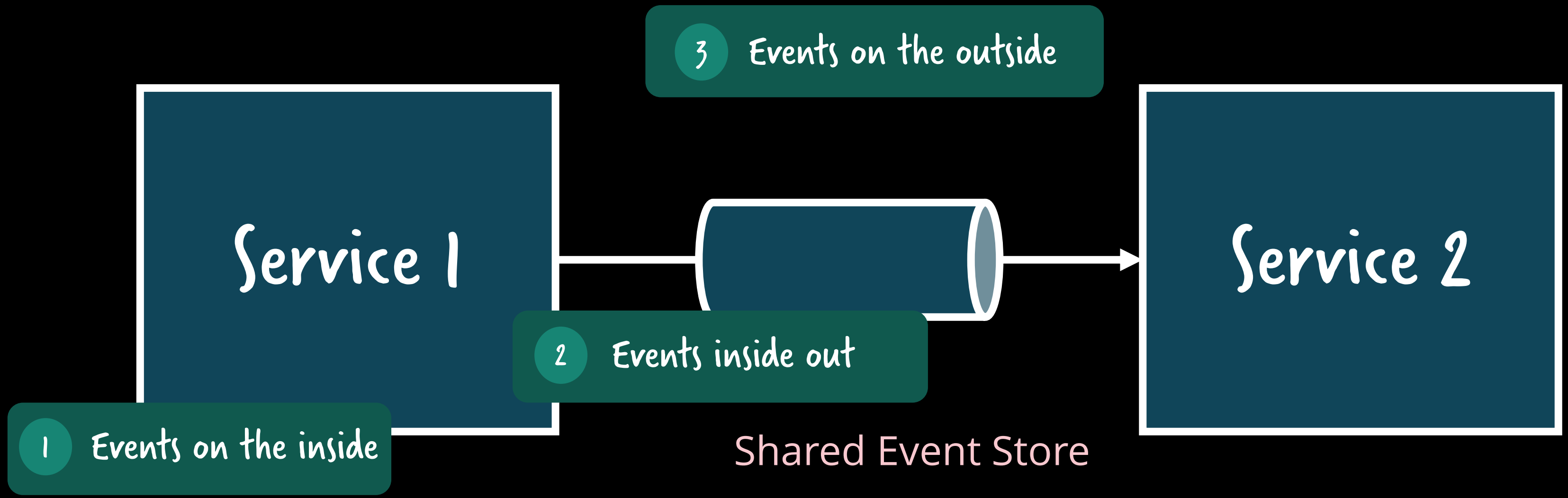
~~Central orchestration Service~~

Recap 2

# Commands vs. Events: Decide about the direction of dependencies
# Beware of event-chains and avoid losing sight
# Balance choreography and orchestration

Recap

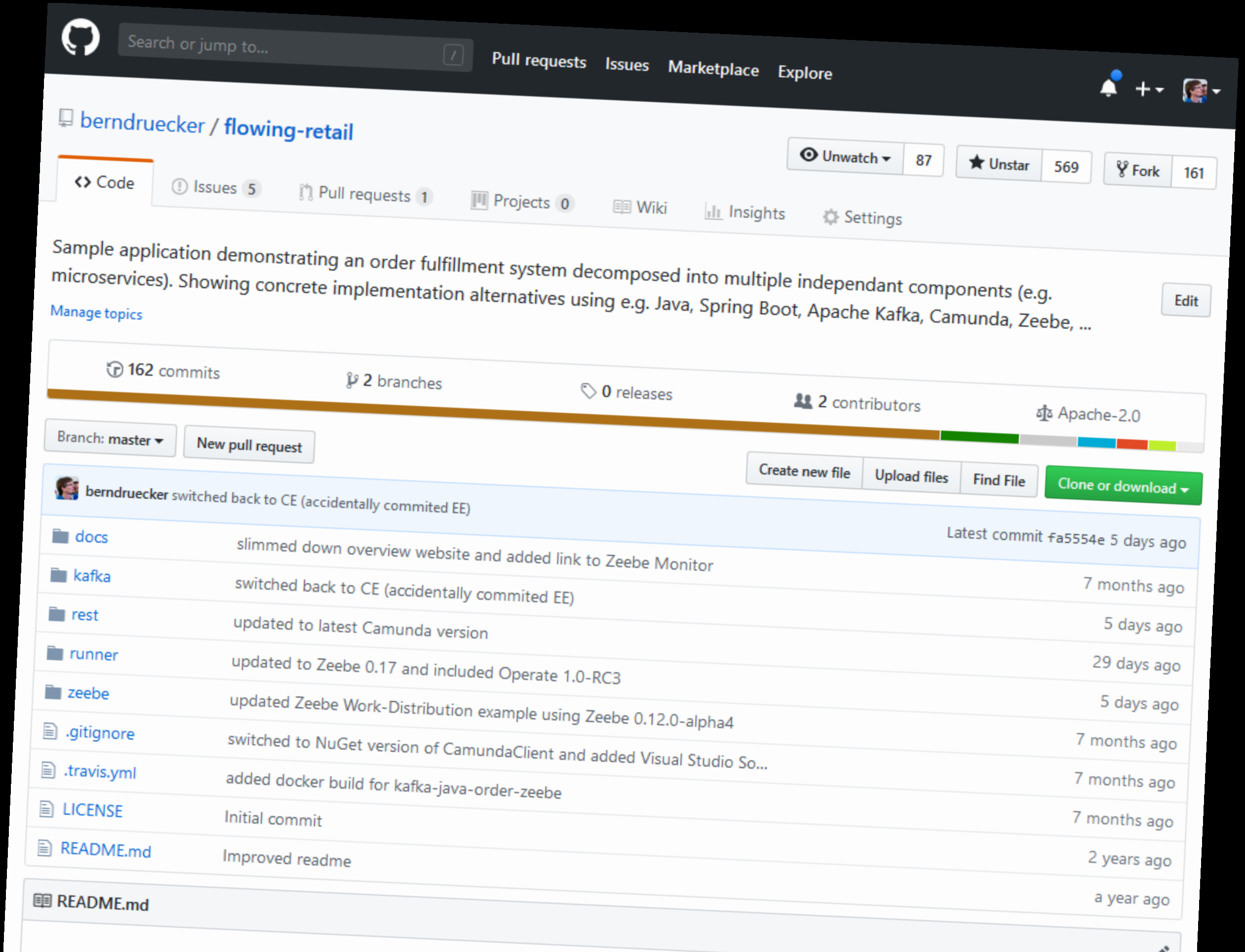
Events as API
Event vs Command
Event chains & visibility
Orchestration vs Choreography



Shared Event Store

Persistent state vs persistent change
Event sourcing & Event Store
Consistency & CAP
Read Models & CQRS

Want to see code?



berndruecker / flowing-retail

Sample application demonstrating an order fulfillment system decomposed into multiple independant components (e.g. microservices). Showing concrete implementation alternatives using e.g. Java, Spring Boot, Apache Kafka, Camunda, Zeebe, ...

162 commits · 2 branches · 0 releases · 2 contributors · Apache-2.0

berndruecker switched back to CE (accidentally commited EE) — Latest commit fa5554e 5 days ago

docs	slimmed down overview website and added link to Zeebe Monitor	7 months ago
kafka	switched back to CE (accidentally commited EE)	5 days ago
rest	updated to latest Camunda version	29 days ago
runner	updated to Zeebe 0.17 and included Operate 1.0-RC3	5 days ago
zeebe	updated Zeebe Work-Distribution example using Zeebe 0.12.0-alpha4	7 months ago
.gitignore	switched to NuGet version of CamundaClient and added Visual Studio So...	7 months ago
.travis.yml	added docker build for kafka-java-order-zeebe	7 months ago
LICENSE	Initial commit	2 years ago
README.md	Improved readme	a year ago

README.md

CAMUNDA CON NEW YORK

Meet practicioners around orchestration & workflow
April 23-24

Capital One,
Cox Automotove,
Nokia Bell Labs,
Goldmann Sachs, …
https://www.camundacon.com/

Nothing for the faint of heart...

Nothing for the faint of heart...

...but doable...

...and worth it

Thank you!

Contact: mail@berndruecker.io
@berndruecker

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-integrationand-how-to-avoid-them.html

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

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