

opportunities & Pitfalls of Event-Driven Utopia

@berndruecker

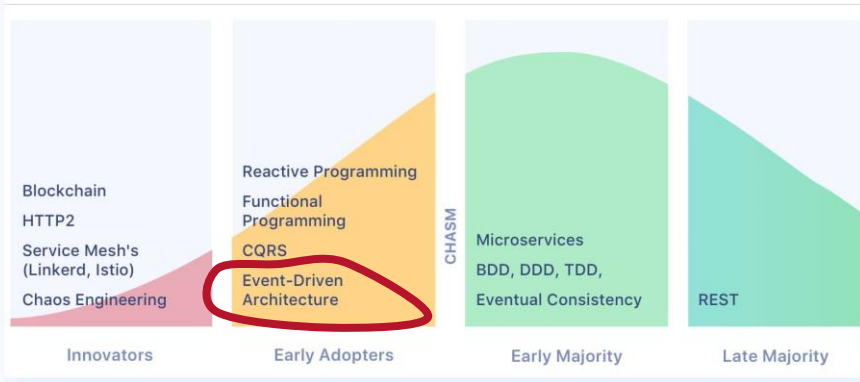


Why this talk

Software Development Architecture and Design 2018 Q4 Graph

<http://infoq.link/architecture-trends-2019>

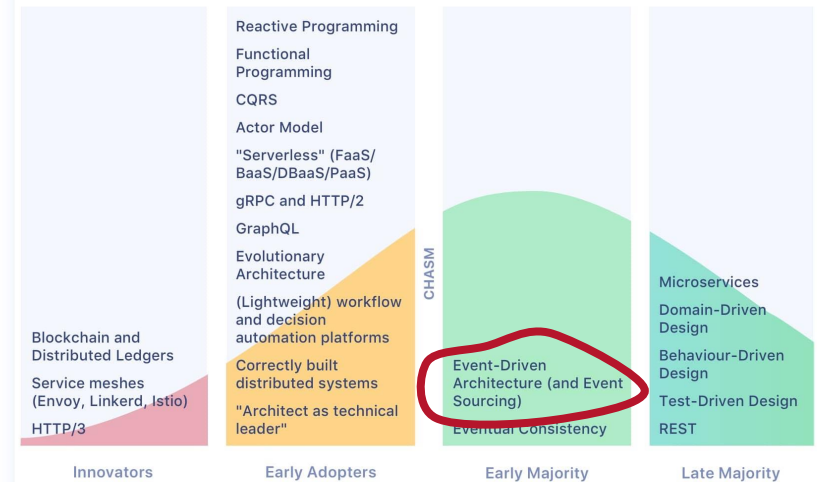
InfoQ



Software Development Architecture and Design 2019 Q1 Graph

<http://infoq.link/architecture-trends-2019>

InfoQ



Why this talk

What do you mean by “Event-Driven”?



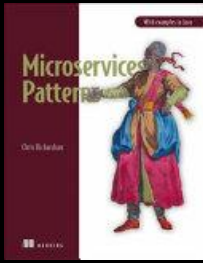
Martin Fowler

07 February 2017

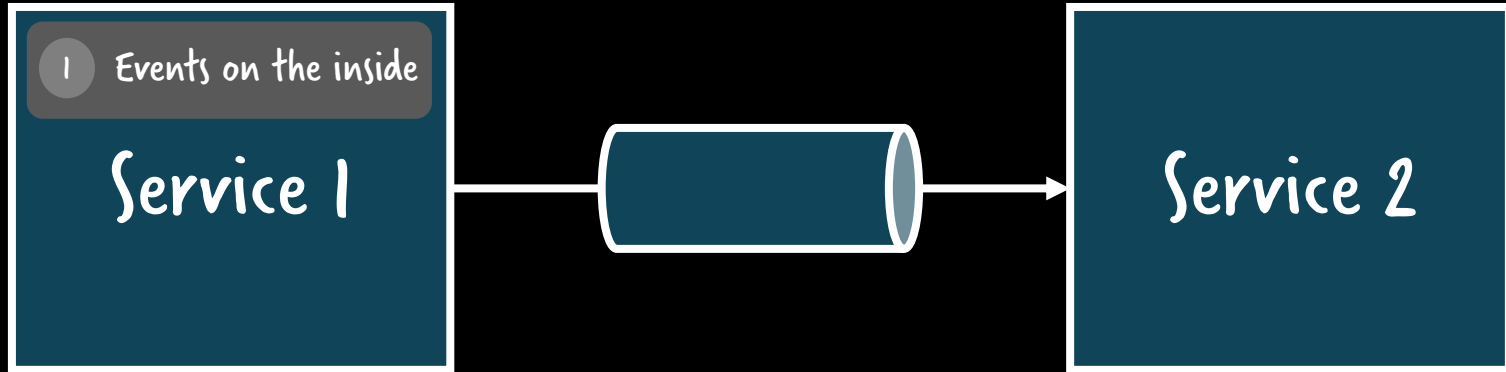
Towards the end of last year I attended a workshop with my colleagues in ThoughtWorks to discuss the nature of “event-driven” applications.

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

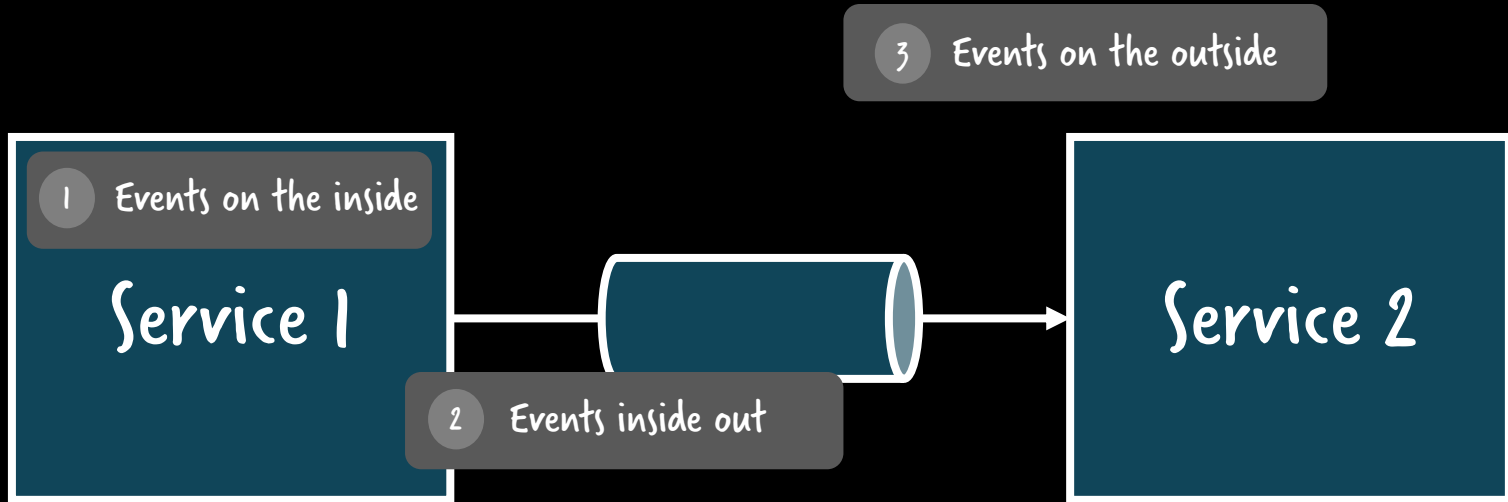
Agenda



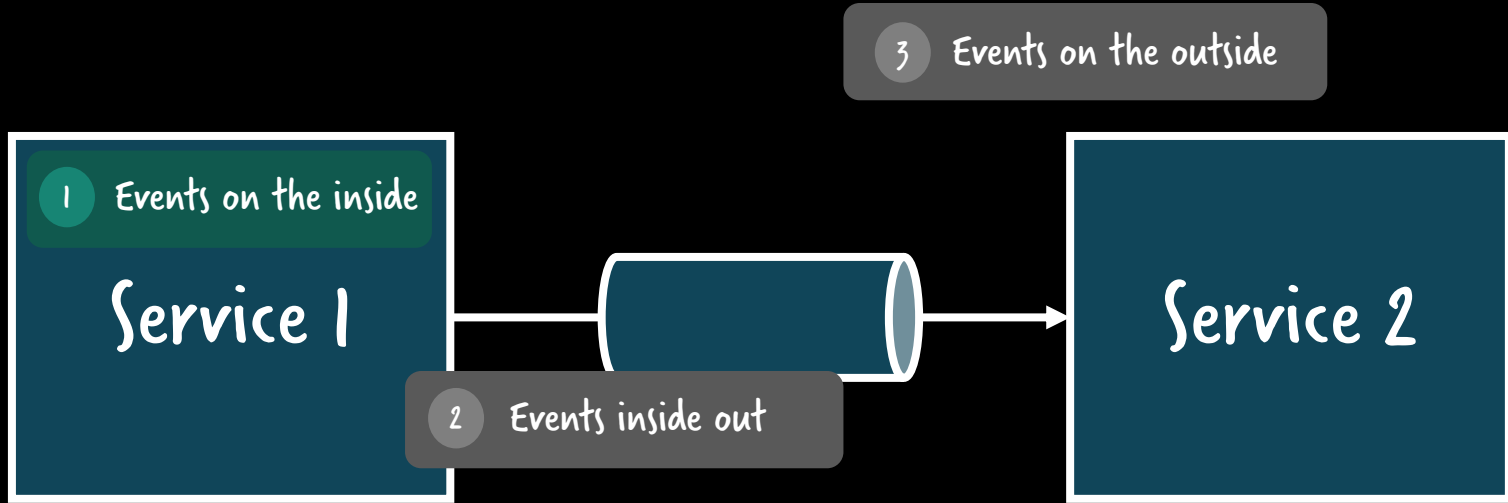
3 Events on the outside



Agenda



Agenda



once upon a time...



Ted Neward

@tedneward

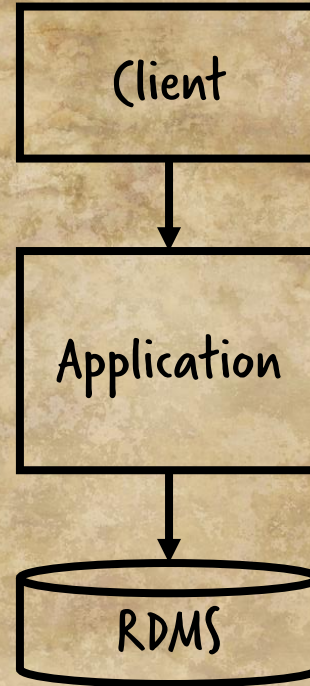
Computational philosopher: techno-jargon for I am a big geek.

📍 Redmond, WA

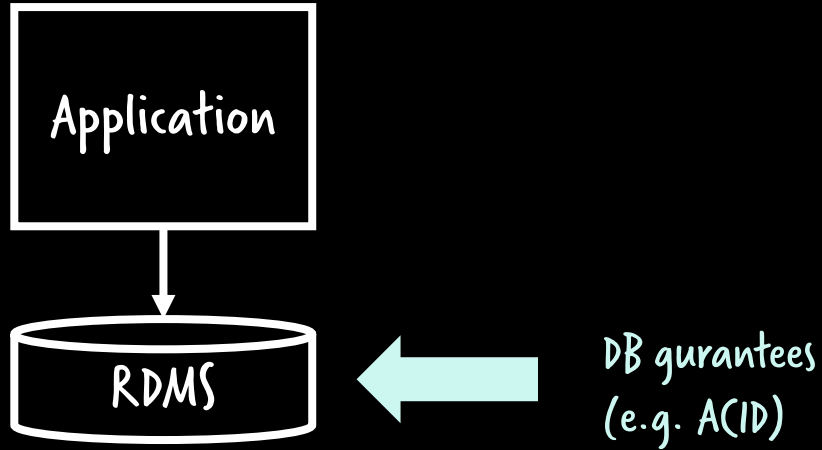
🔗 about.me/TedNeward

📅 Joined June 2008

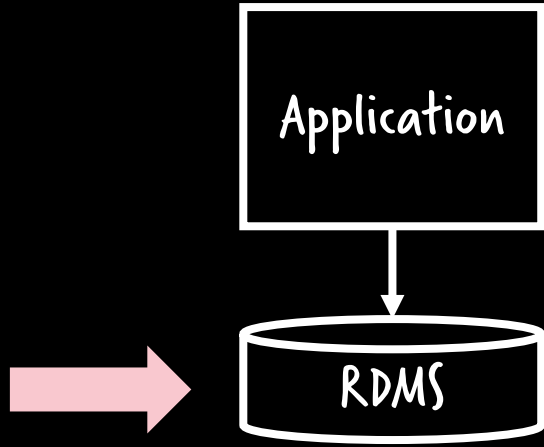
BBC architecture
(box - arrow - box - arrow - cylinder)
Every architecture diagram
you'll ever need



The great thing about this architecture

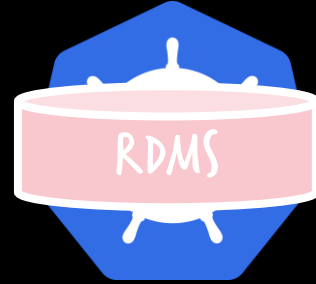


The problem



not webscale
resiliency is expensive

Does not fit in Kubernetes:

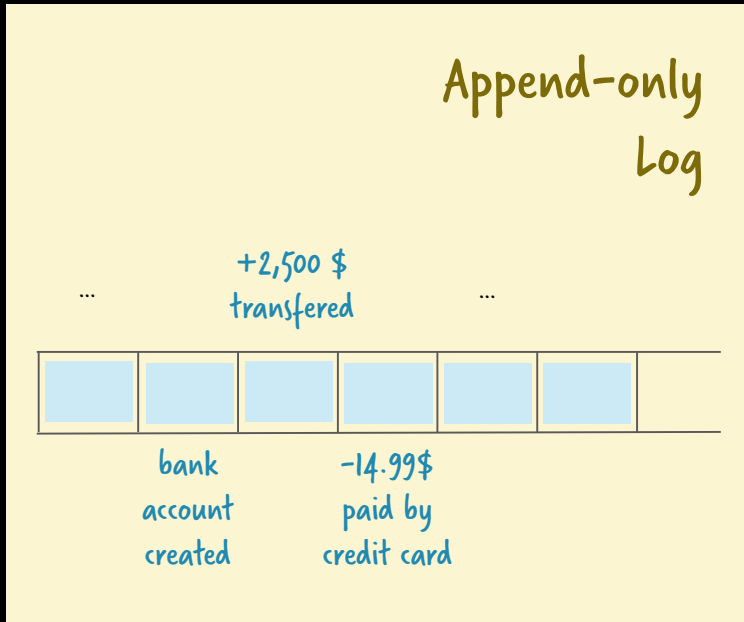


Pat Helland



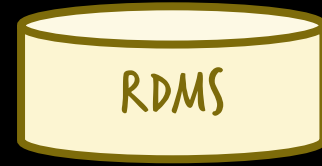
*Immutability
Changes
Everything!*

Persistent change



➔ Current Balance = 2,485.01 \$

Persistent state

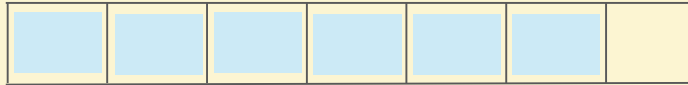


Account #	Balance
12345	2,500\$

Persistent change

Append-only Log

... +2,500 \$
transferred ...



bank
account
created

-14.99\$
paid by
credit card

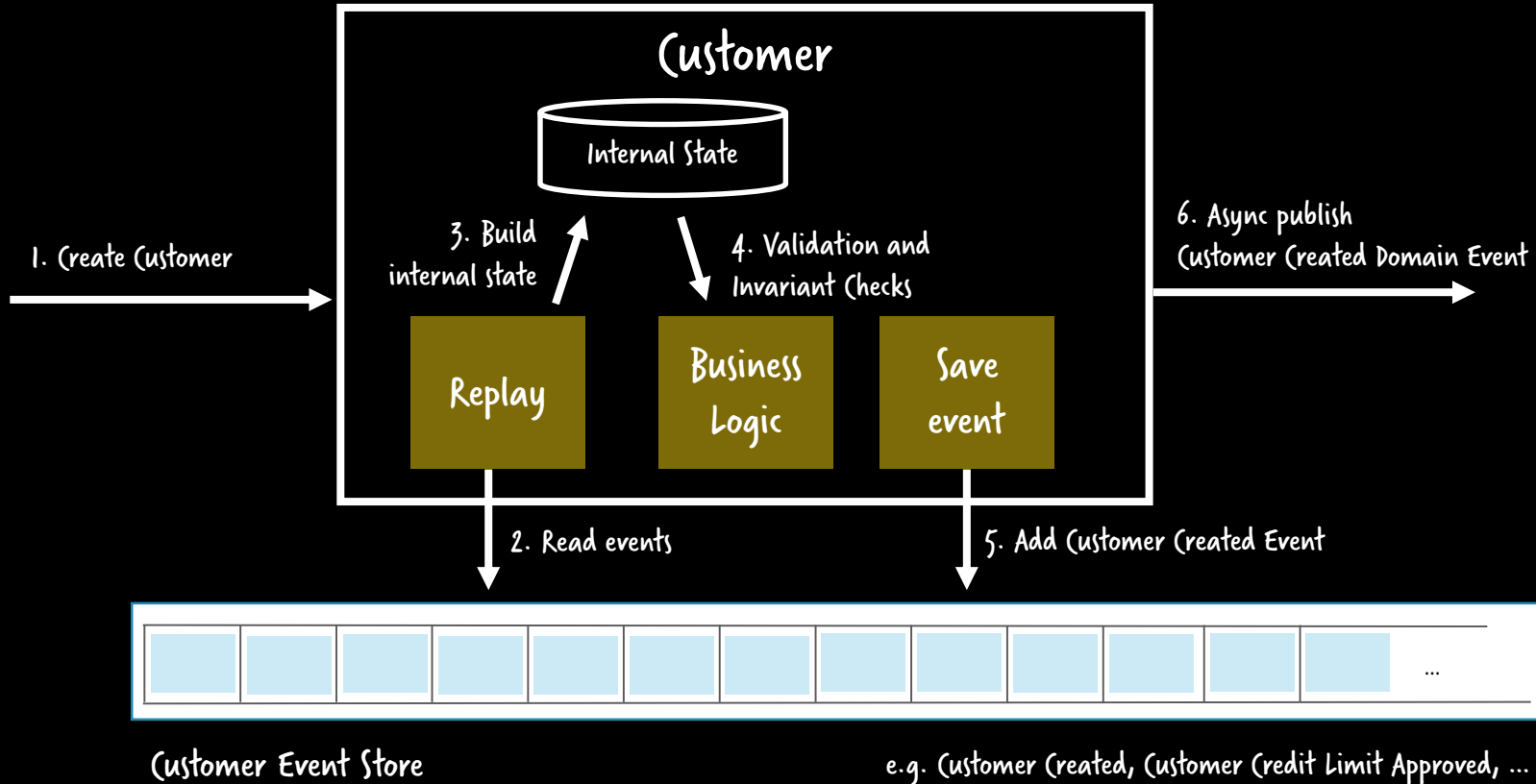
Event

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

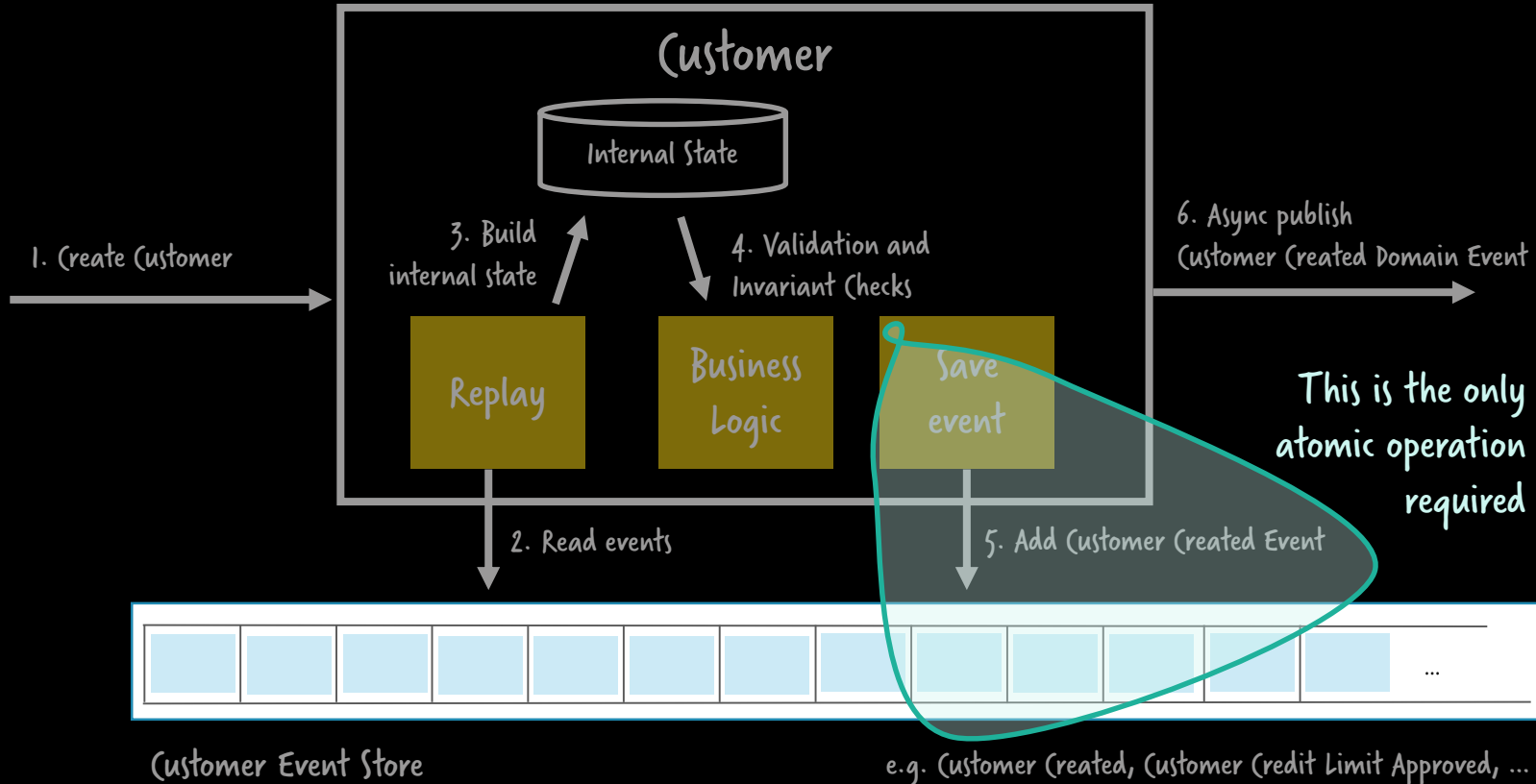
Event

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

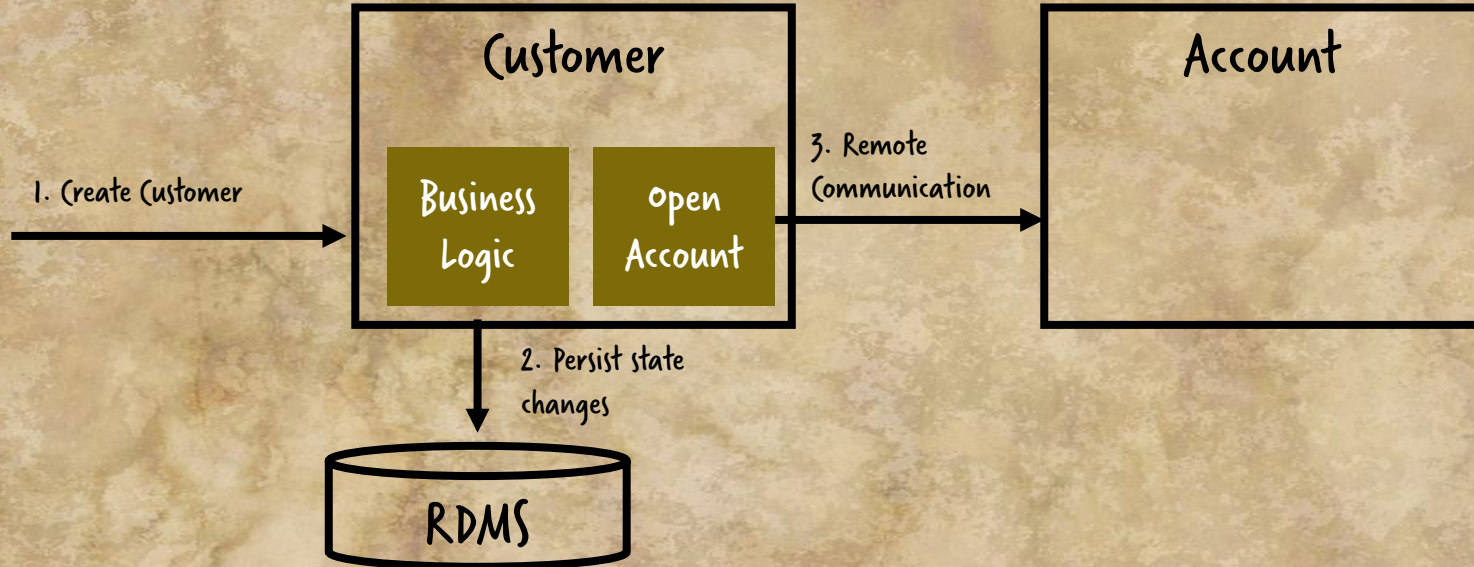
Event Sourcing in a nutshell



Working without distributed transactions



Traditional Architecture





Pat Helland

Distributed Systems Guru
Worked at Amazon,
Microsoft & Salesforce

Life beyond Distributed Transactions: an Apostate's Opinion

Position Paper

Pat Helland

Amazon.Com
705 Fifth Ave South
Seattle, WA 98104
USA

PHelland@Amazon.com

The positions expressed in this paper are personal opinions and do not in any way reflect the positions of my employer Amazon.com.

ABSTRACT

Many decades of work have been invested in the area of distributed transactions including protocols such as 2PC, Paxos, and various approaches to quorum. These protocols provide the application programmer a façade of global serializability. Personally, I have invested a non-trivial portion of my career as a strong advocate for the implementation and use of platforms

Instead, applications are built using different techniques which do not provide the same transactional guarantees but still meet the needs of their businesses.

This paper explores and names some of the practical approaches used in the implementations of large-scale mission-critical applications in a world which rejects distributed transactions. We discuss the management of fine-grained pieces of application data which may be repartitioned over time as the application grows. We also discuss the design patterns used in sending messages between these repartitionable pieces of data.

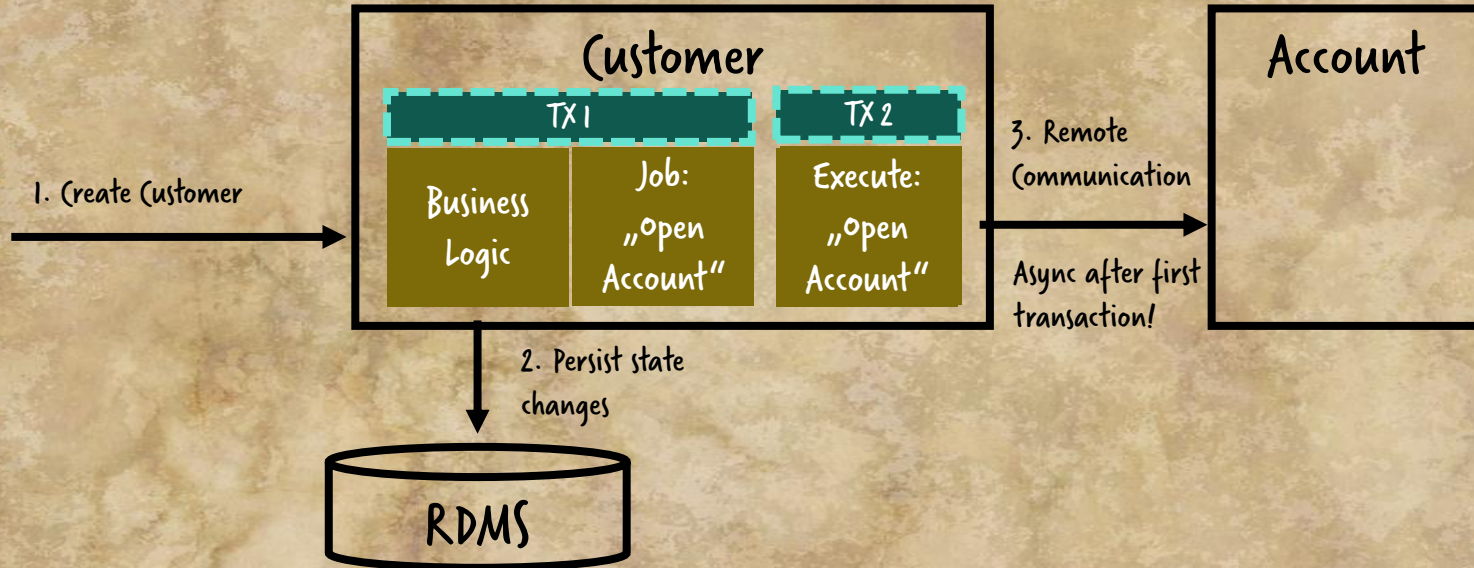


Pat Helland

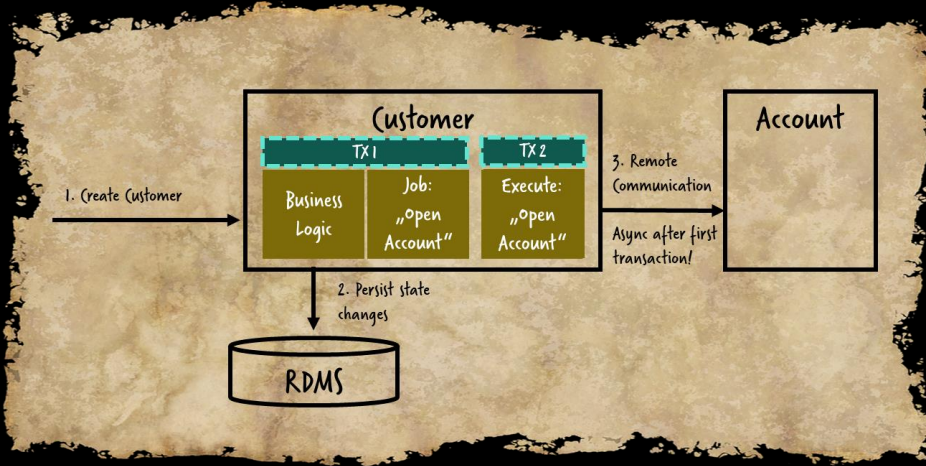
Distributed Systems Guru
Worked at Amazon,
Microsoft & Salesforce

“
Grown-Ups Don't Use
Distributed Transactions

outbox pattern in traditional architectures



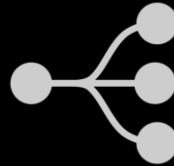
outbox pattern – Implementation Approaches



Scheduler

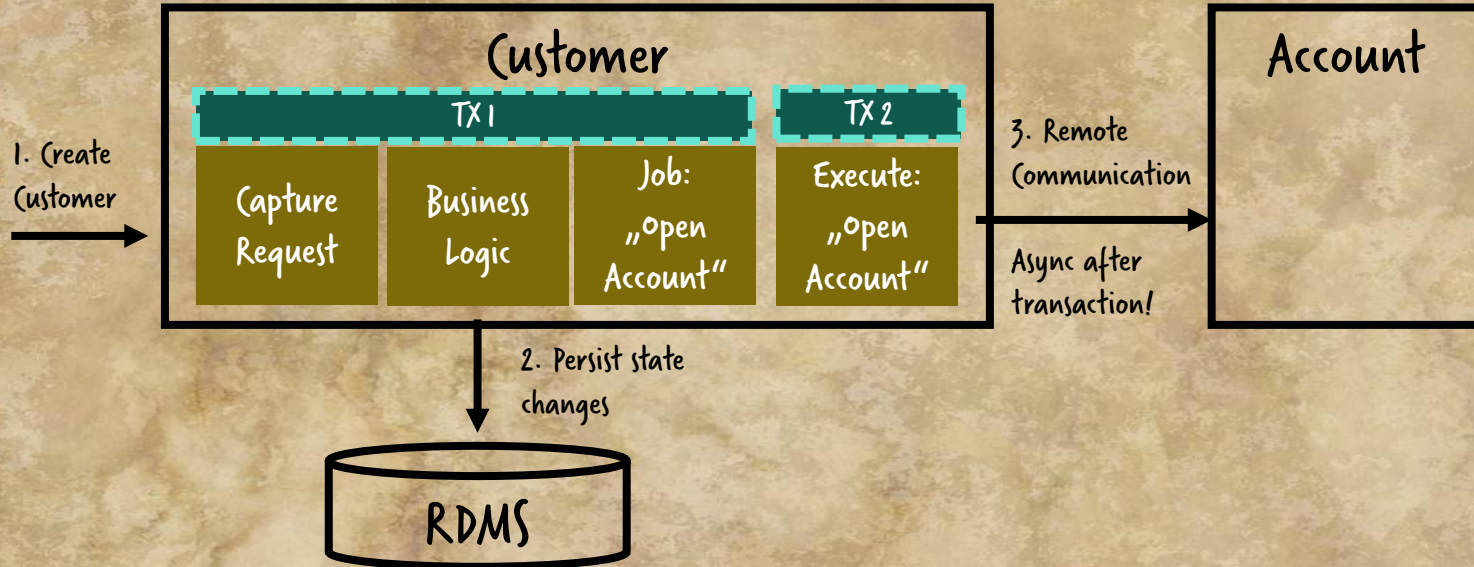


Database
Transaction Log

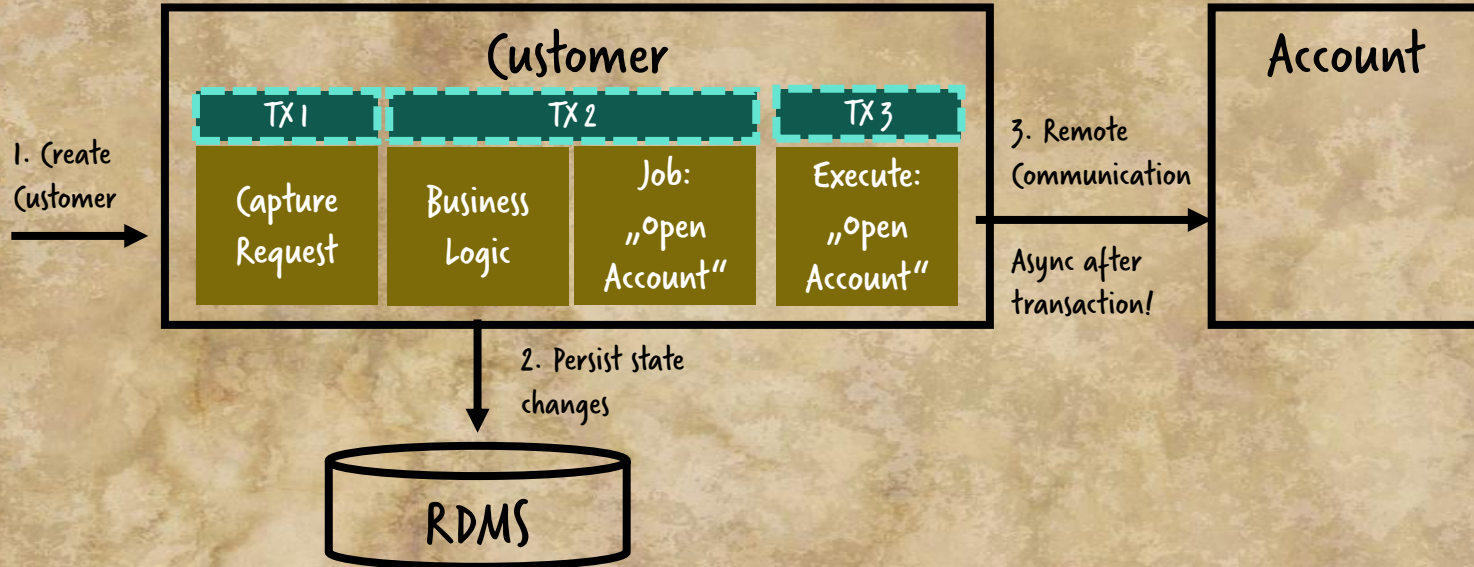


Workflow
Automation

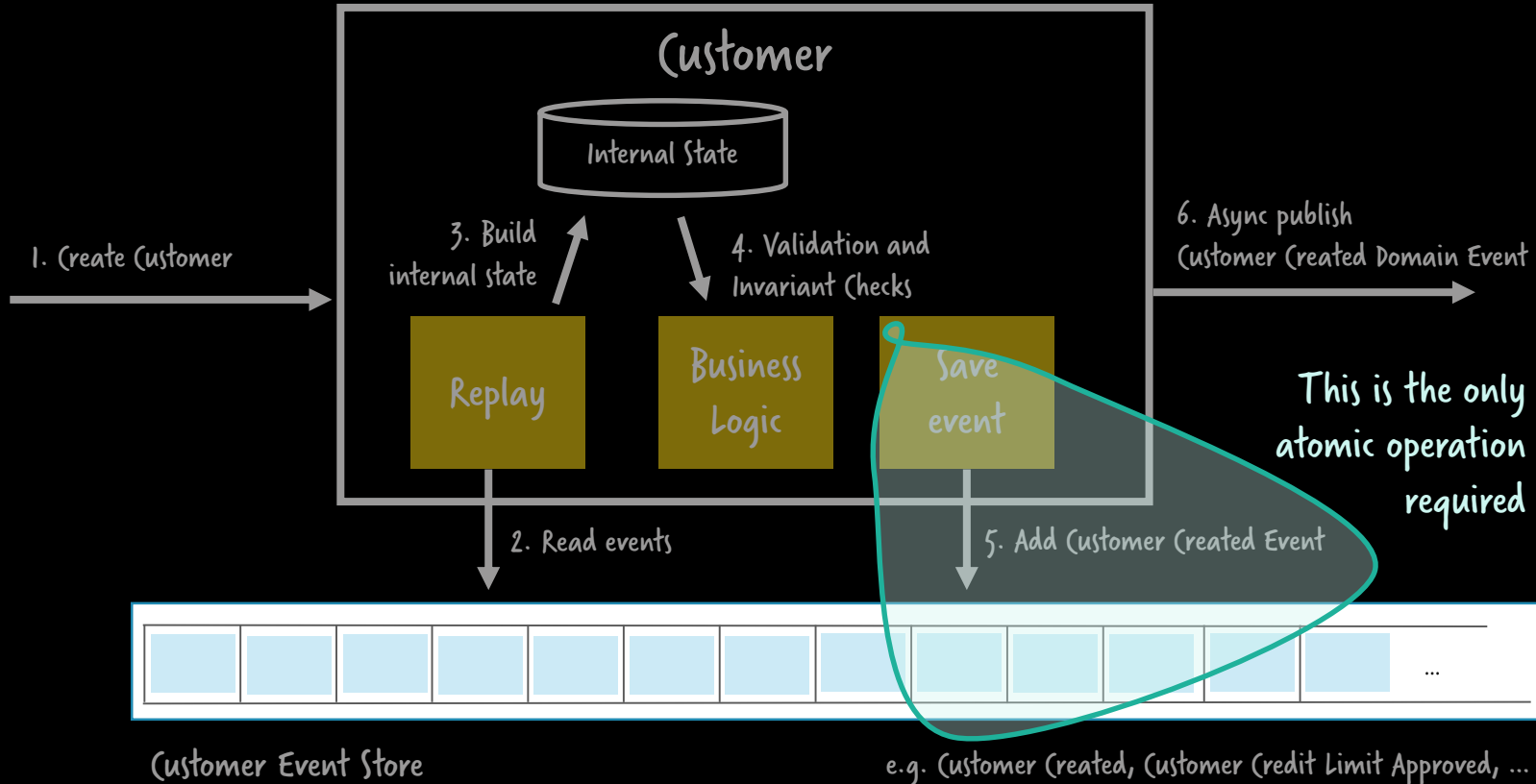
Idempotency



Idempotency



Working without distributed transactions



Events on the inside.

An example from my world



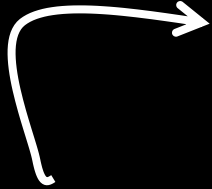
Bernd Ruecker
Co-founder and
Chief Technologist of
Camunda

mail@berndruecker.io
[@berndruecker](#)

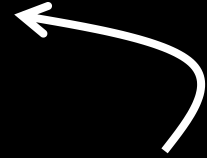
Warning:
Contains opinion!

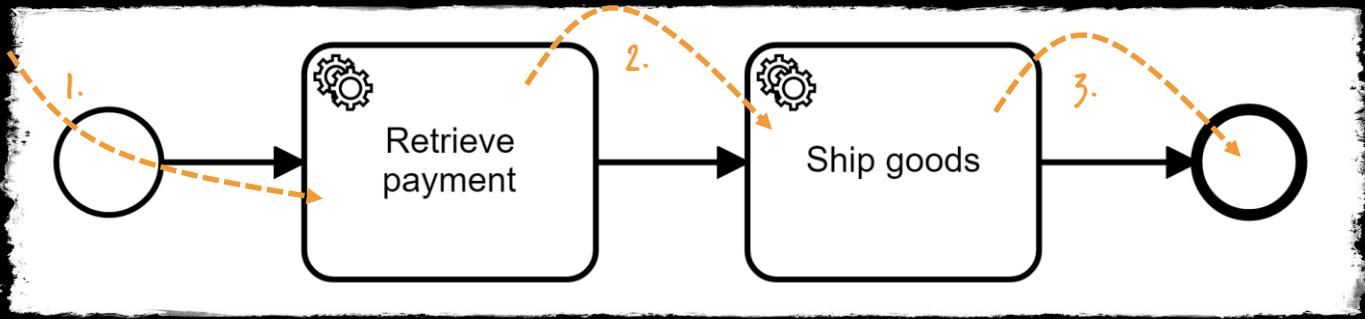
We offer two different workflow engines. Why?

Persistent
State



Persistent
change





Workflow Engine



1. INSERT

Workflow Instance Id	Current Activity	State
2	RetrievePayment	running

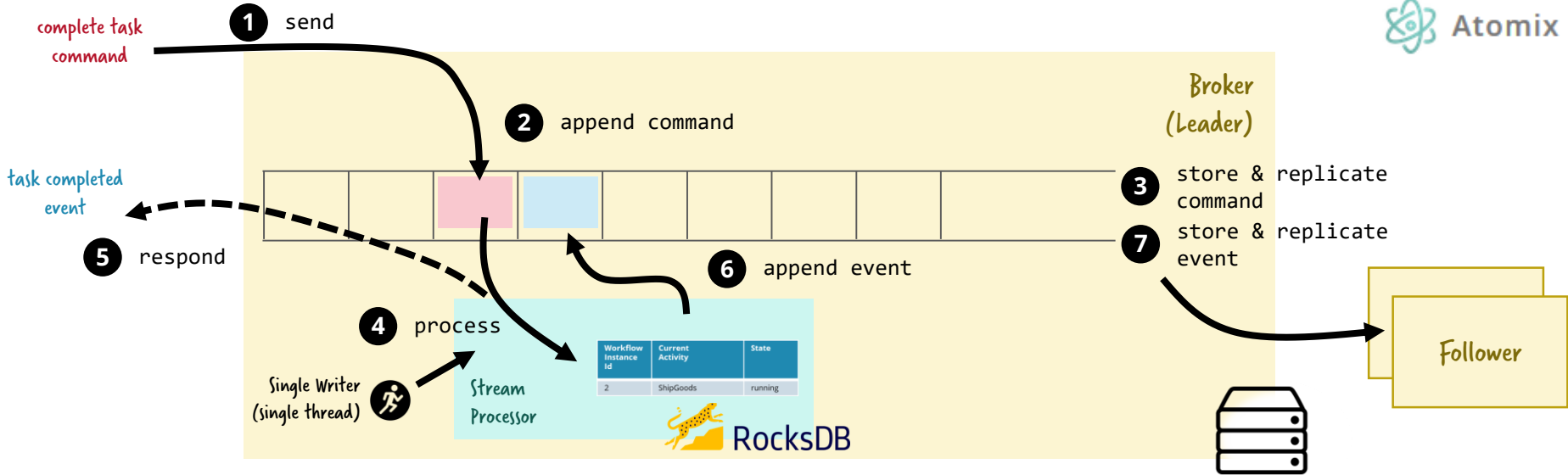
2. UPDATE

Workflow Instance Id	Current Activity	State
2	ShipGoods	running

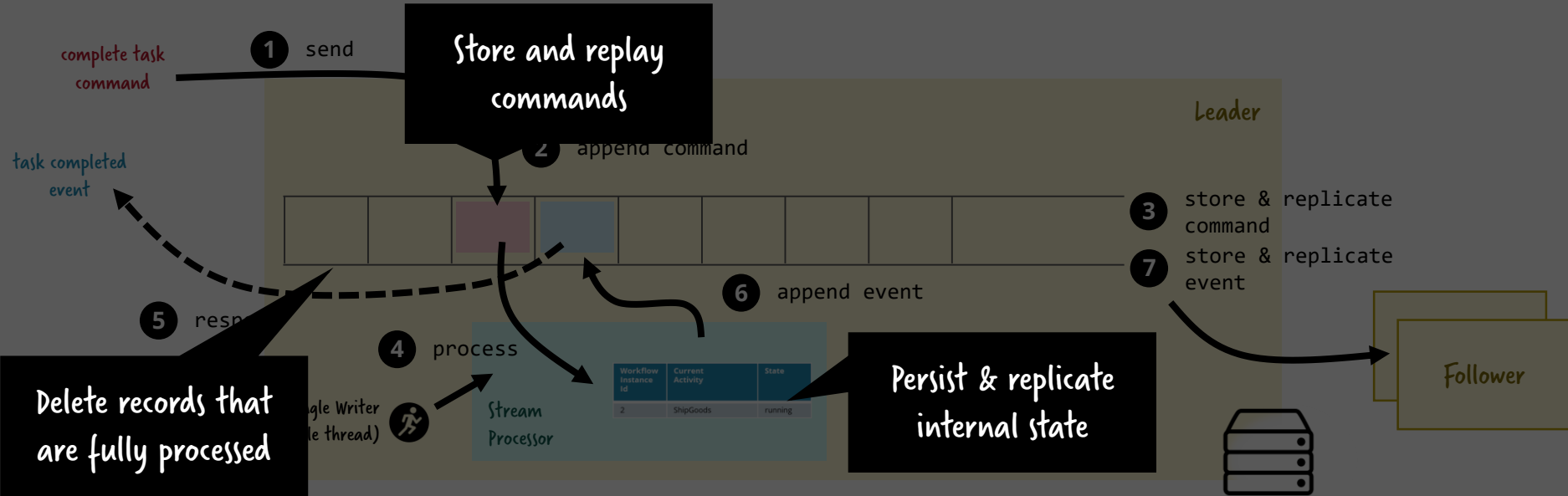
3. UPDATE

Workflow Instance Id	Current Activity	State
2	OrderDelivered	ended

Event Handling, Replication & Single Writer



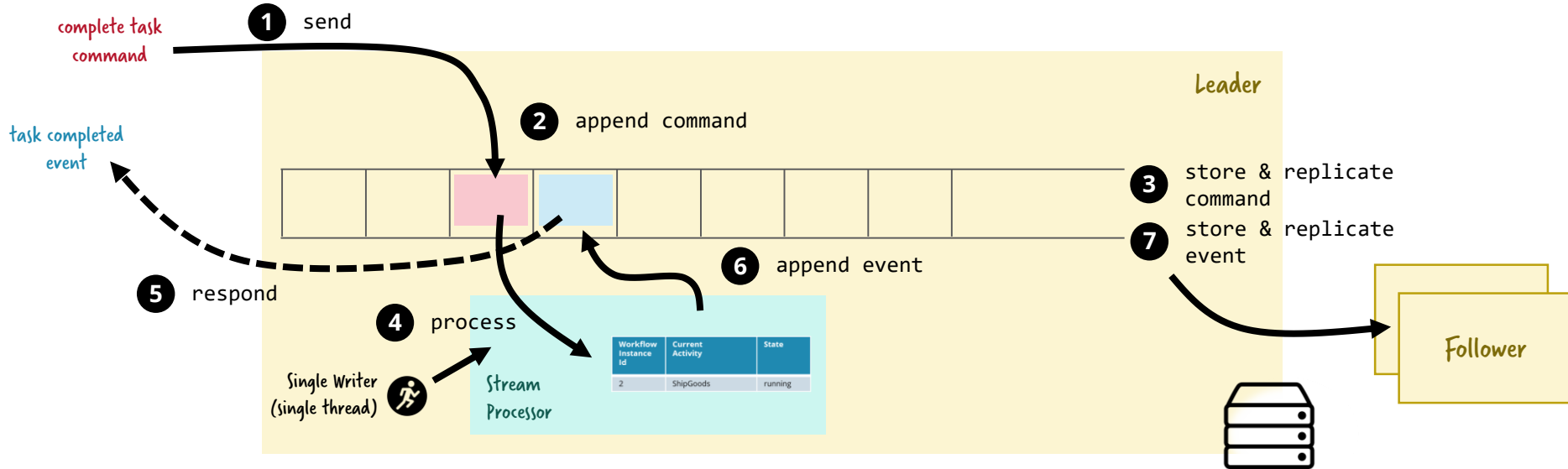
What we do different



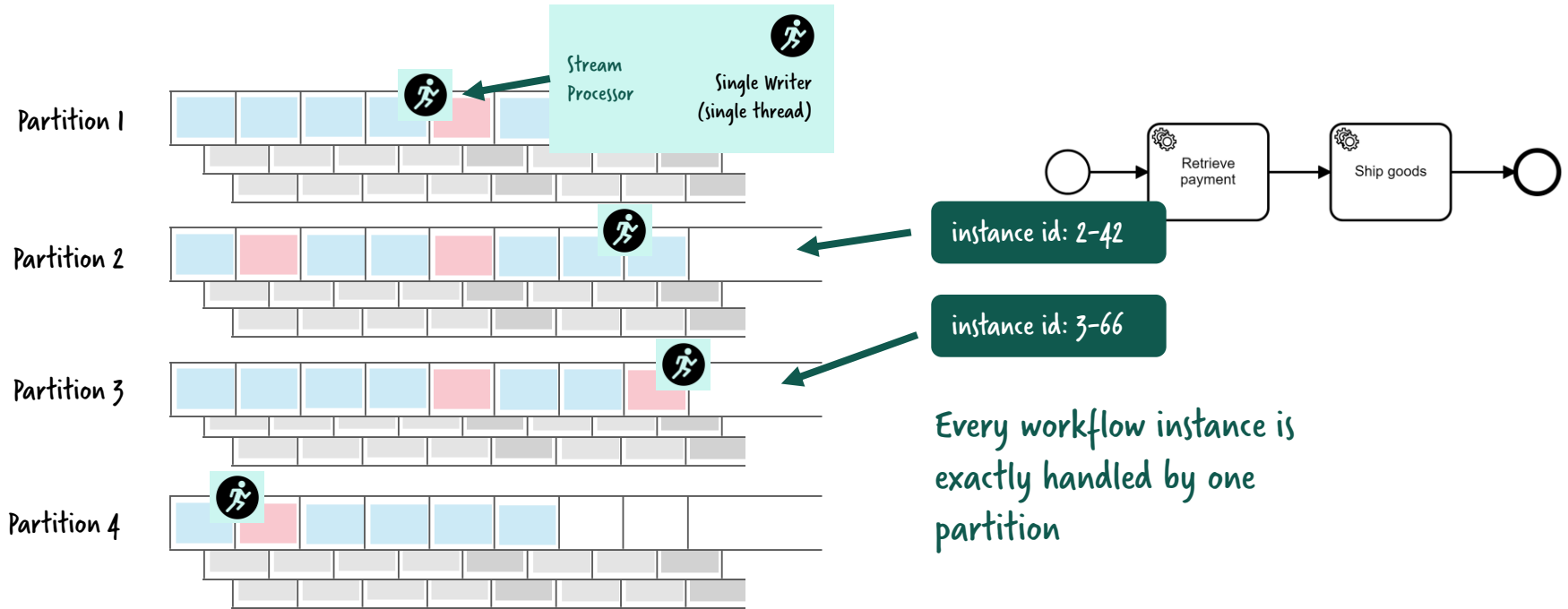
Consistency
Availability
Partition



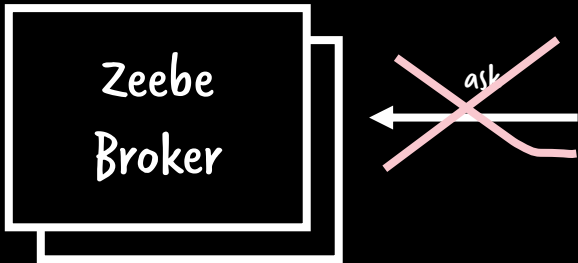
Zeebe is CP



Horizontal scalability by partitioning



Queries and read models



The screenshot shows the Camunda Cloud interface for a process instance named "order-kafka" (Instance 326, Version 1, started on 29 Apr 2019 19:57:14). The process flow diagram includes steps: Order placed, Send command: retrieve payment, Payment received, Send command: fetch goods, Goods fetched, Send command: ship goods, Goods shipped, and Send event: order delivered. A tooltip for the "Payment received" step shows activityInstanceID: 341, startDate: 29 Apr 2019 19:57:15, and endDate: 29 Apr 2019 19:57:15. Below the flow is the "Instance History" table.

Variable	Value
CorrelationId_FetchGoods	"0adc8c2f-1425-4df5-a40d-2af89dd05084"
CorrelationId_RetrievePayment	"8ef17bd8-2282-43bc-bee1-7eb60aB4dd58"
orderId	"checkout-generated-c6bdde71-32b2-463a-880c-3ac4044e2ac2"
paymentInfo	"YeahWeCouldAddSomething"
traceId	"a9cf4367-ab69-4585-a3cd-7092da490a29"

Recap 1 – Events on the inside

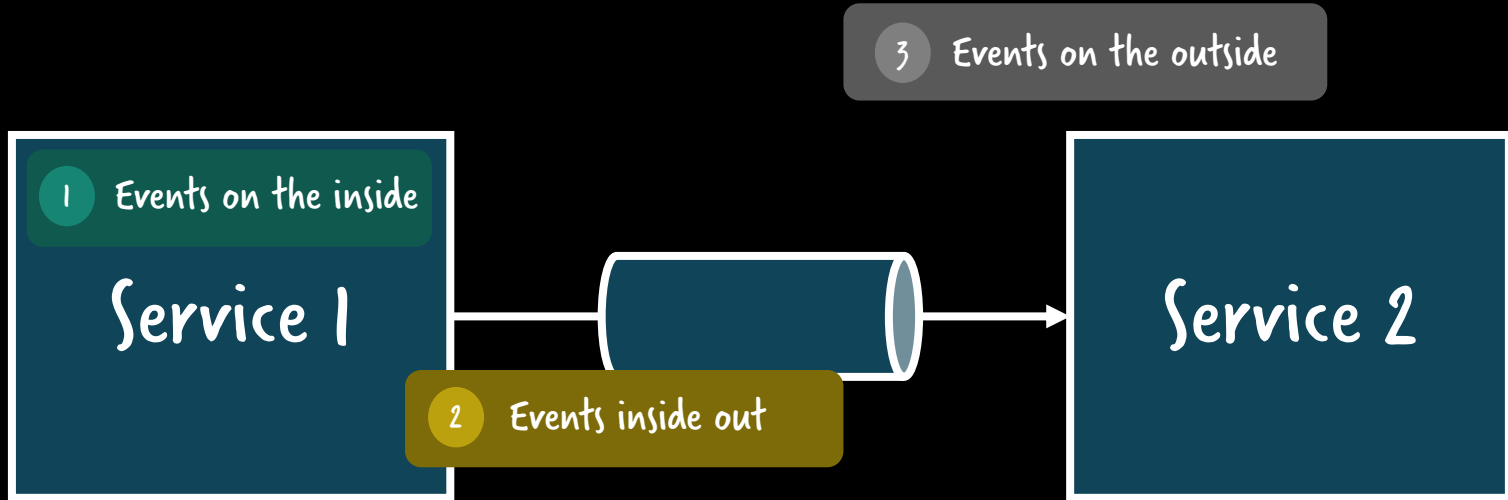
Natural mechanism to build scalable services in distributed systems (with outbox & co included)

But

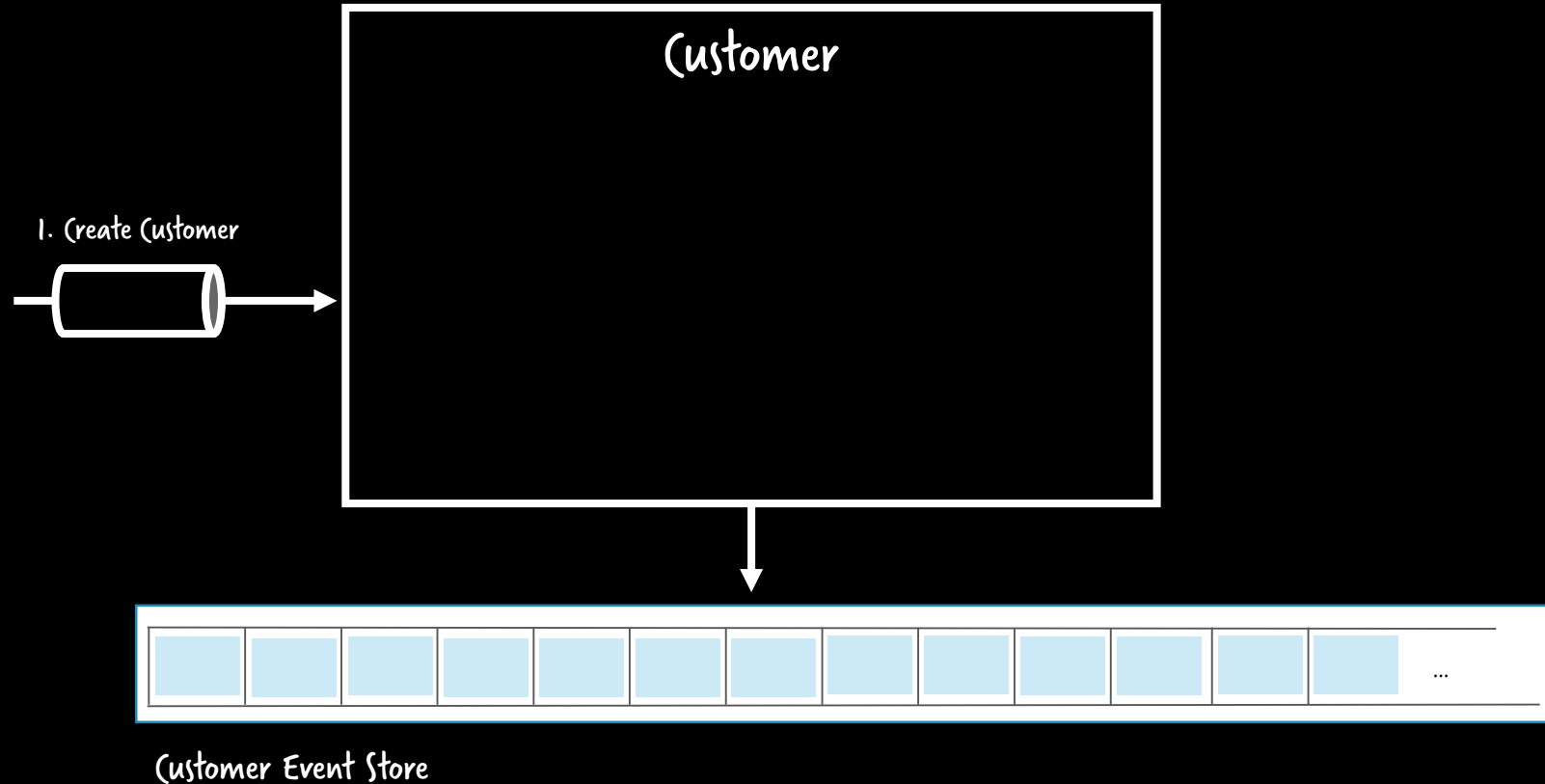
You have to think about reads, queries & eventual consistency

Much less industry experience available

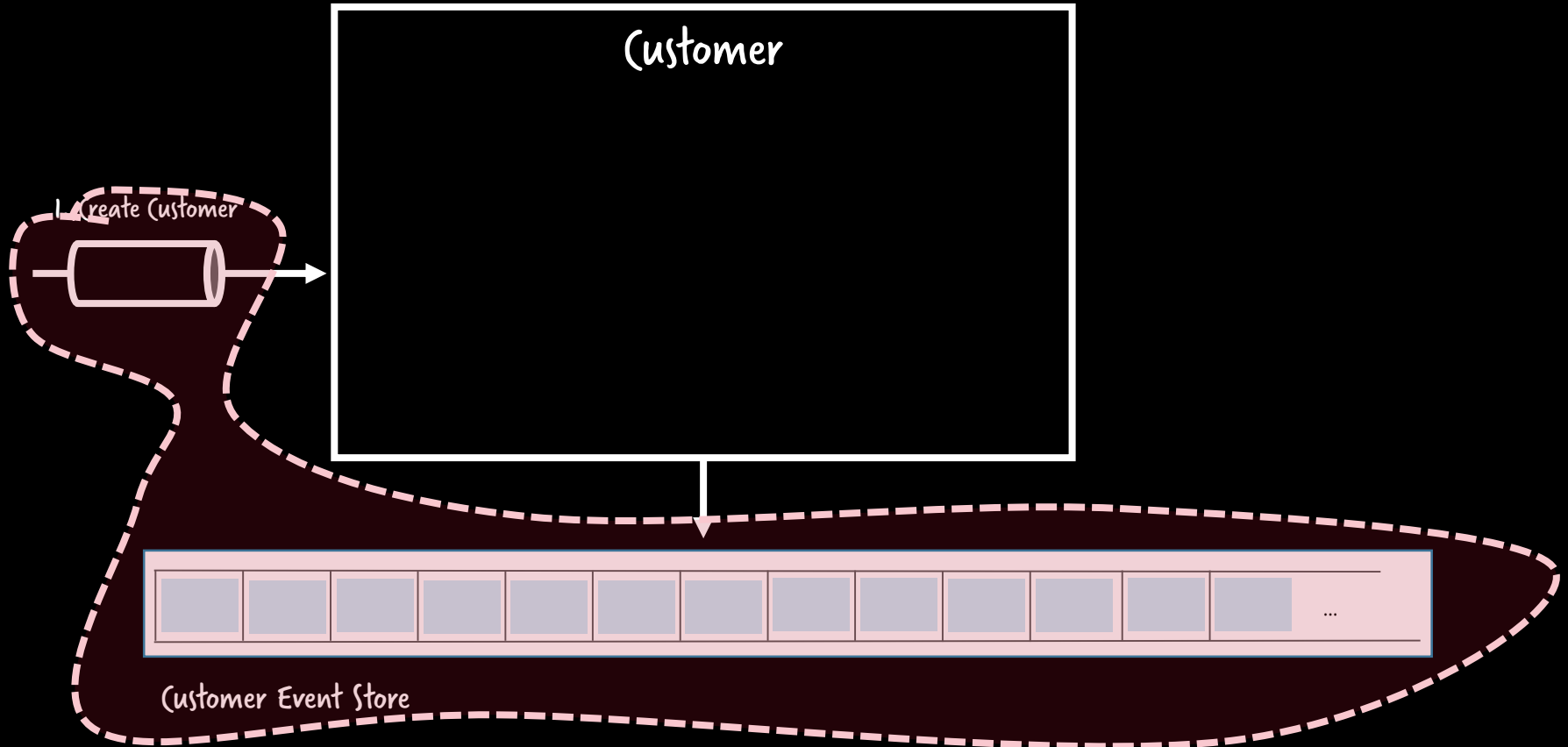
Agenda



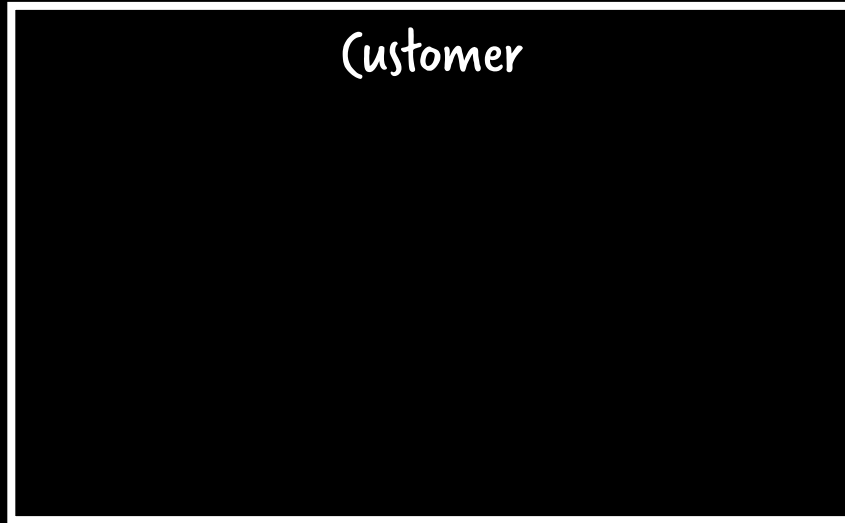
Event Store and Messaging



Merge Messaging and Event Store



Merge messaging and event store



1. Create Customer



Shared Event Store

Enter the world of Kafka...


The image shows a screenshot of a Confluent blog post. At the top left is the Confluent logo. The navigation menu includes 'Product', 'Cloud', 'Developers', 'Blog', and 'Docs'. A prominent orange 'Download' button is on the right, along with a search icon and a US flag. The main content area has a dark background with the word 'MICROSERVICES' in blue. The title 'Messaging as the Single Source of Truth' is in large white text. Below the title is a circular profile picture of Ben Stopford. His name 'Ben Stopford' is written in orange, and the date 'August 2, 2017' is in grey. A blue print icon is visible on the right. The beginning of the article text is visible at the bottom: 'This post discusses Event Sourcing in the context of Apache Kafka® examining the need for a'.

confluent

Product Cloud Developers Blog Docs Download

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

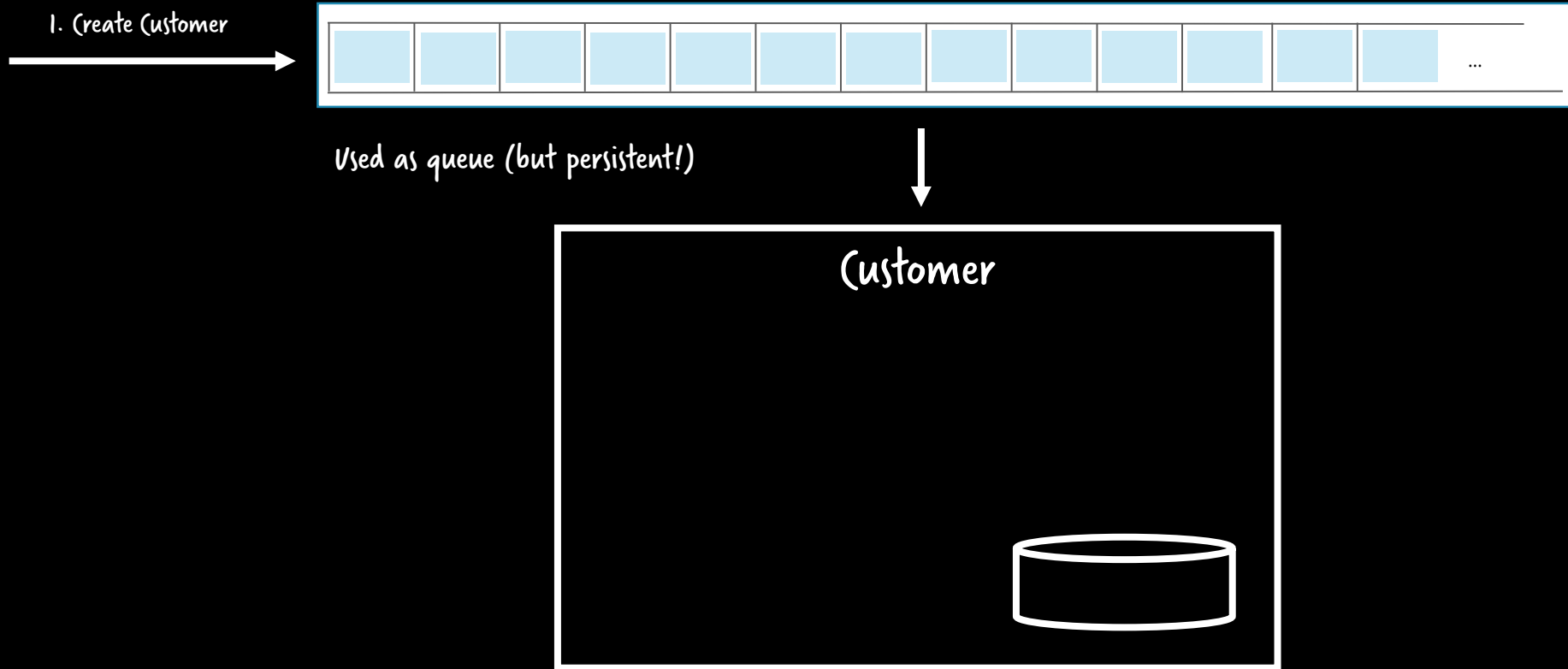


1. Create Customer

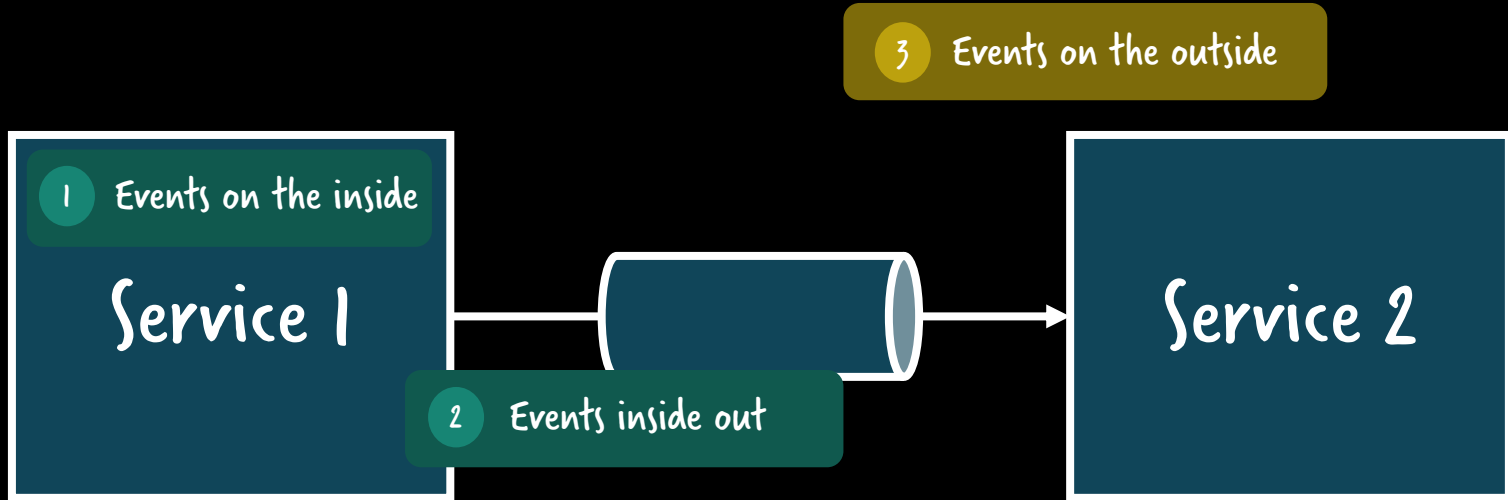


Shared Event Store

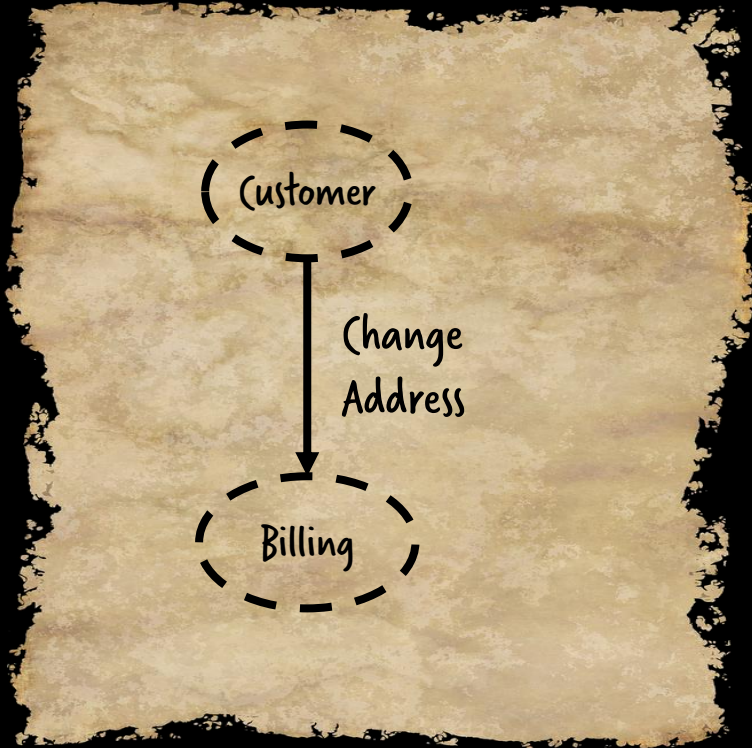
Kafka as transport



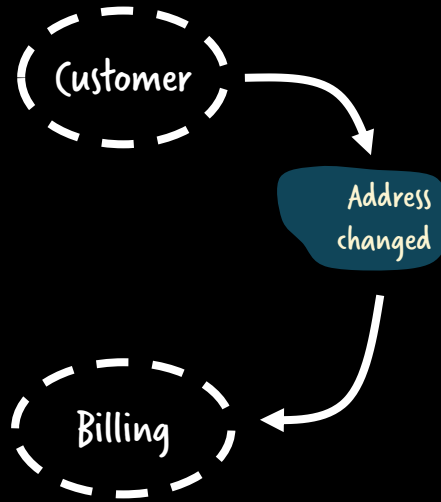
Agenda



once upon a time

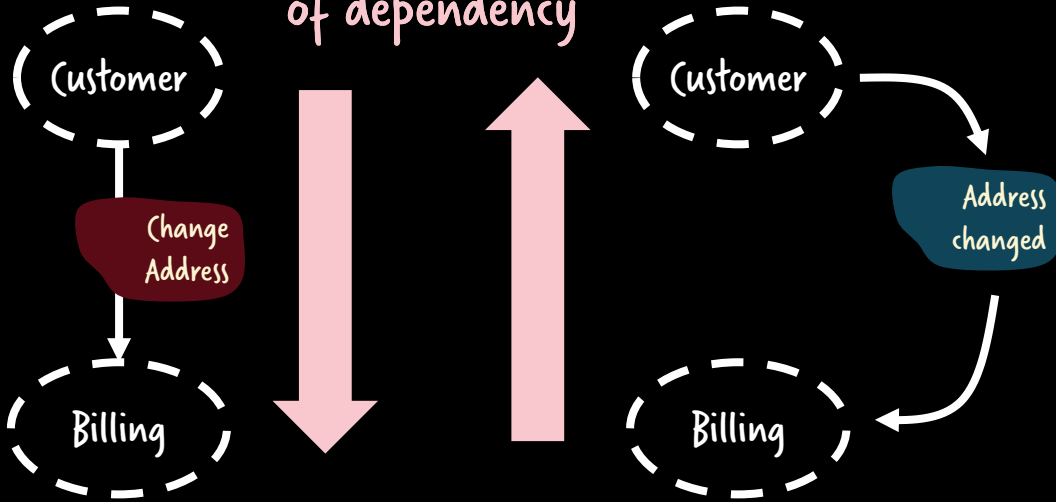


Event Notification

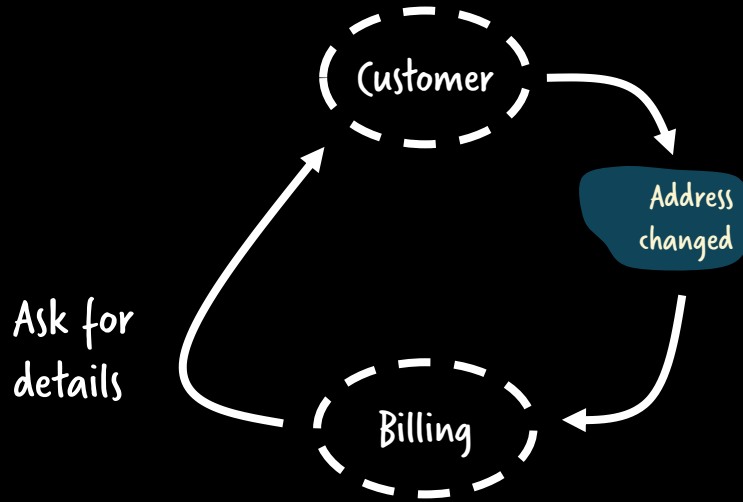


Event Notification

Reverse direction
of dependency

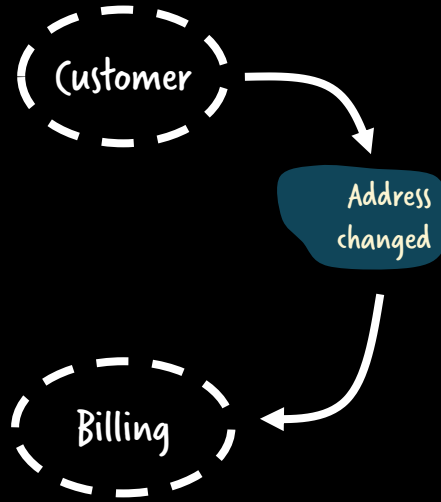


Event Notification



```
AdresChanged  
{  
  customerId: 42  
}
```

Event-carried State Transfer



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

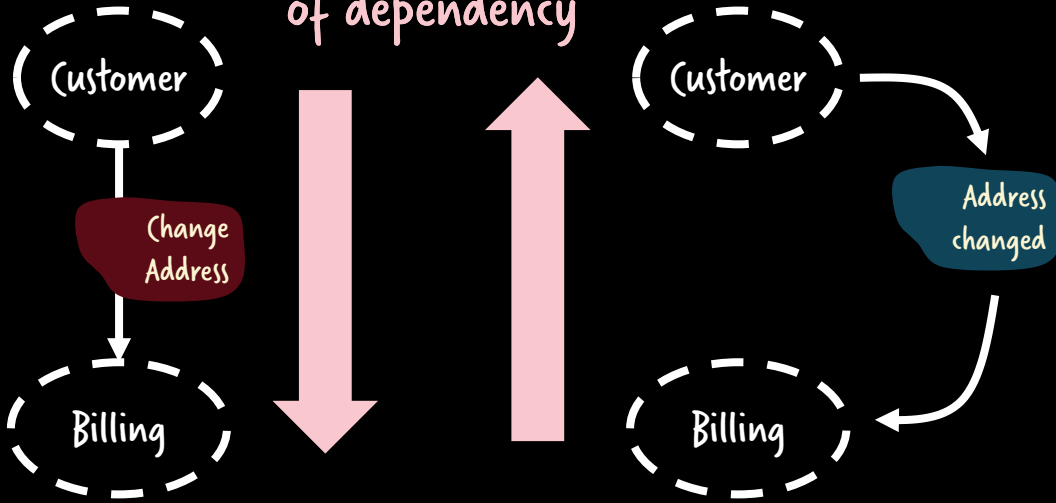
```
AddressChanged  
{  
  customerId: 42,  
  address: ...  
}
```

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

```
CustomerMoved  
{  
  ...,  
}
```

This decision is complex

Reverse direction
of dependency



Example

Change Address

Address

Submit



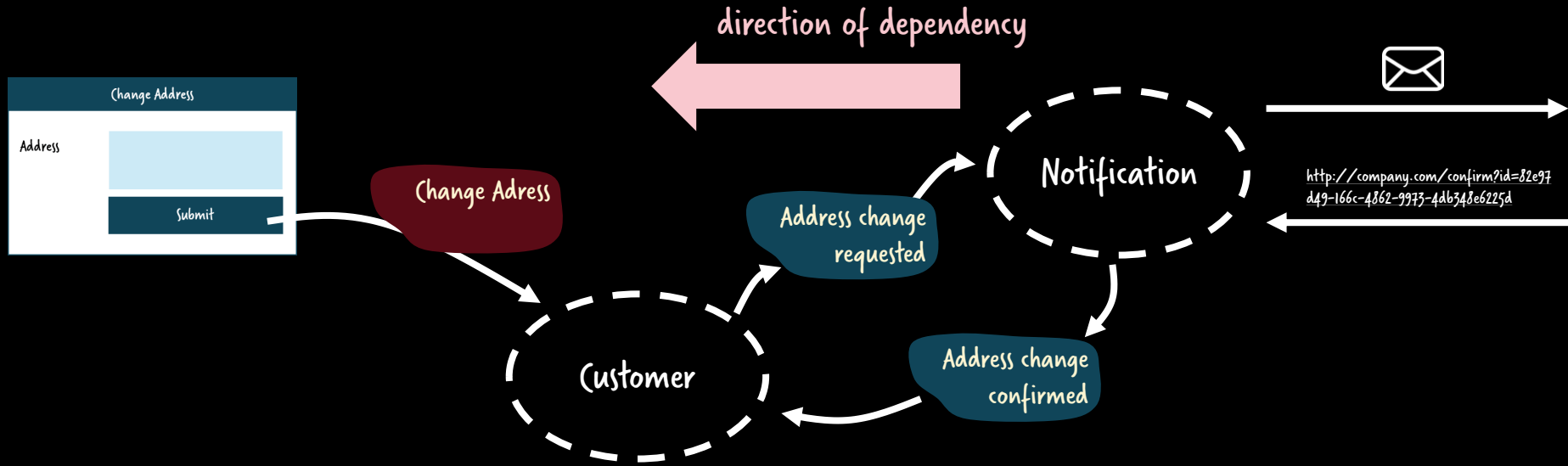
Incoming Email

From	bla@company.com
Date	2019-04-23 09:05

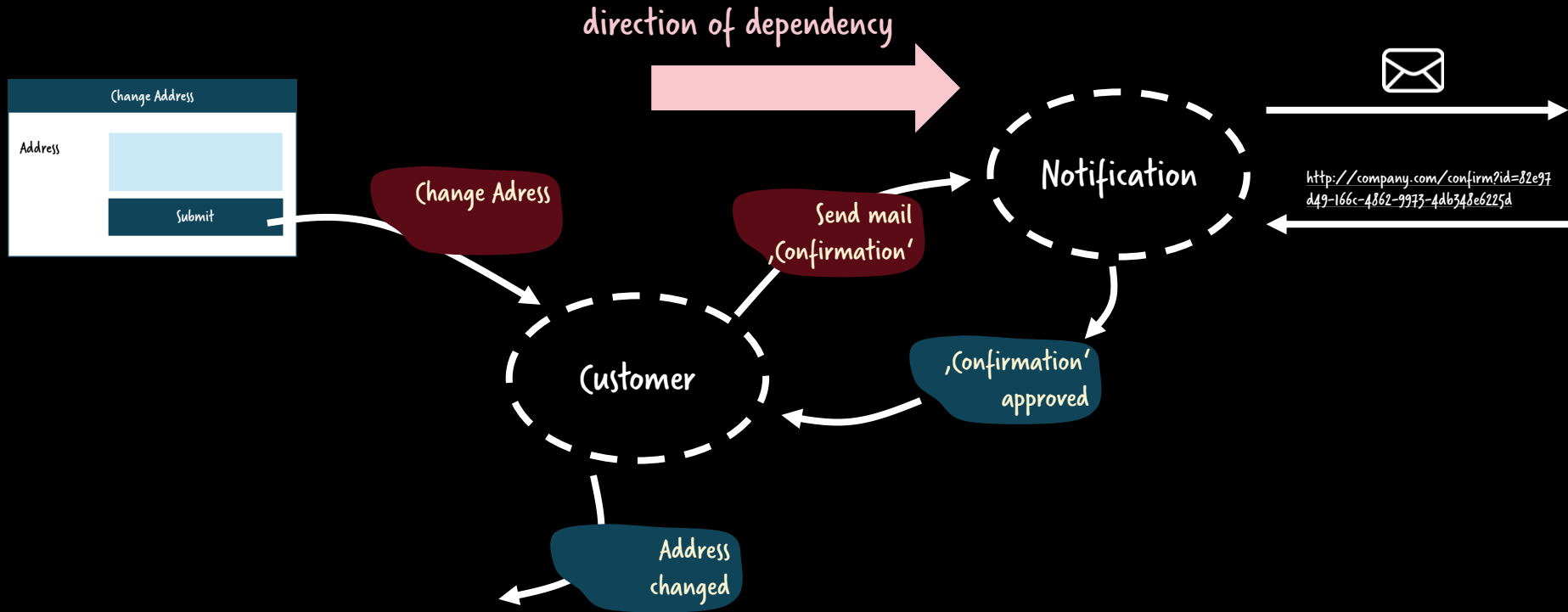
To confirm your address change please click on this link:

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

Example



Example



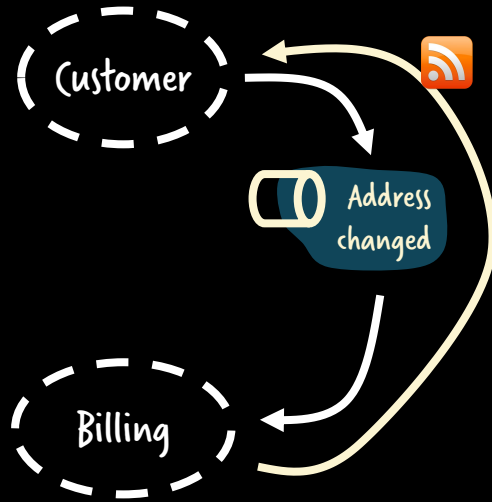
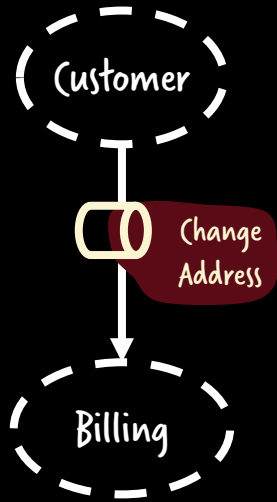
Challenge: Command vs. Event

Command

vs

Event

It is NOT about communication protocols



It can be messaging,
REST, whatever,

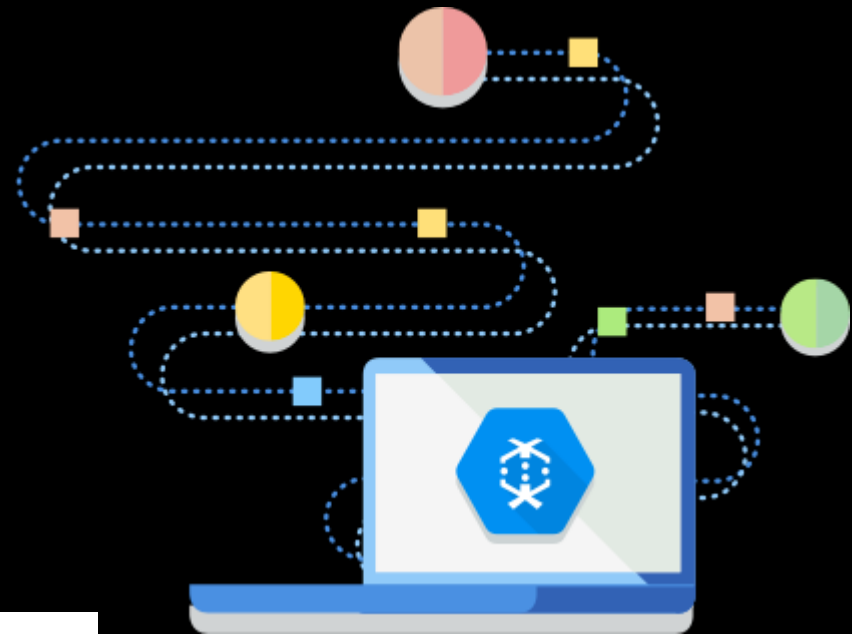
Manifold ways of transport



 RabbitMQ

...

Manifold ways of transport



Message

Record

?

Event

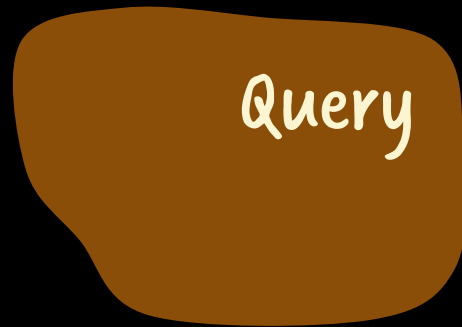
Event

Command

Query

Fact,
happened in the past,
immutable

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



Commands in disguise

Wording of
recipient



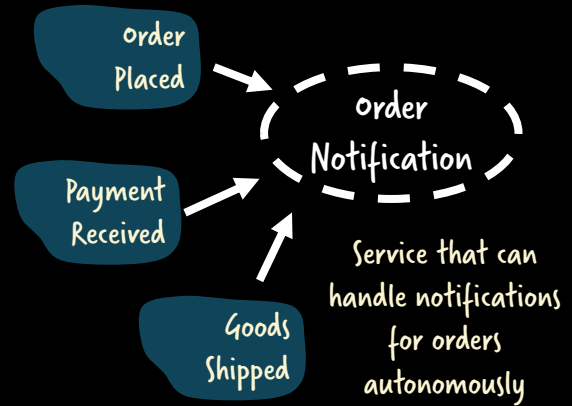
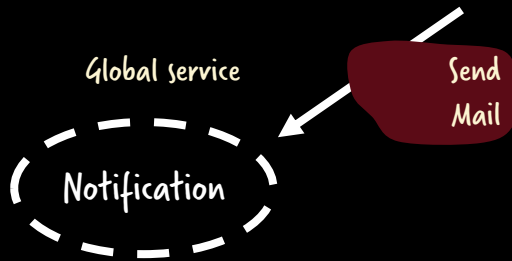
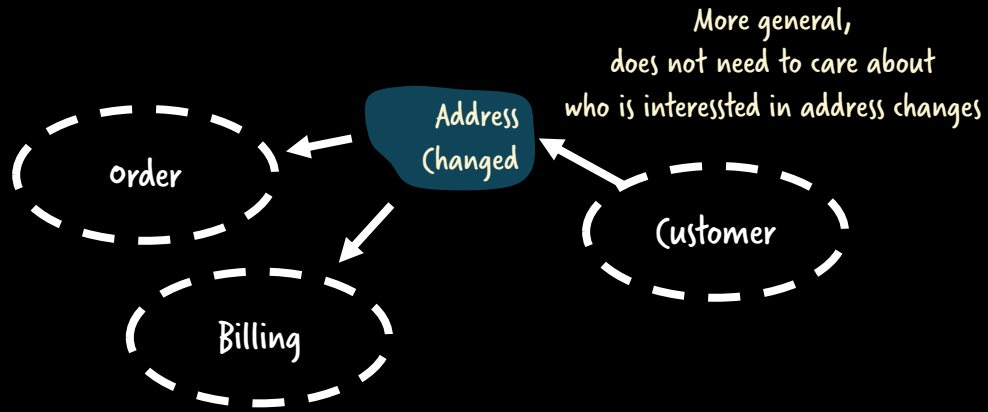
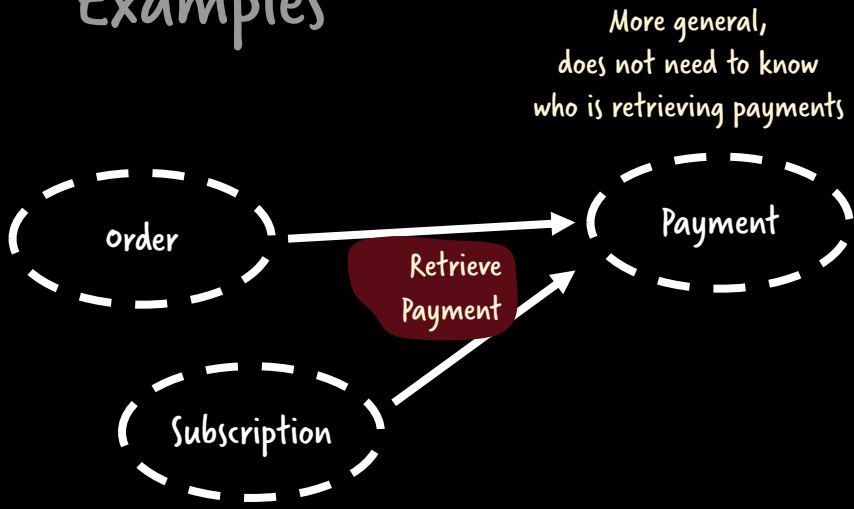
Send
Message

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

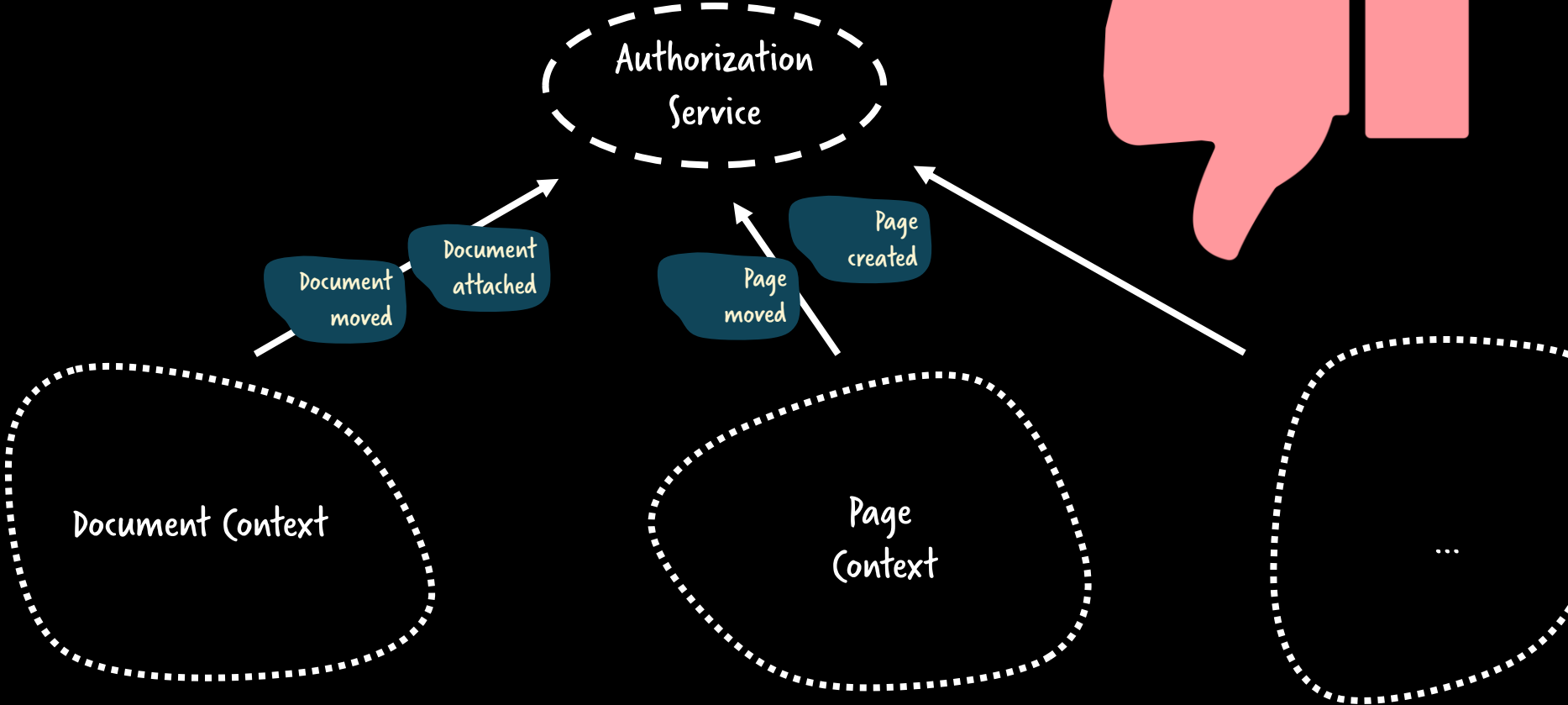
Wording of
Sender



Examples



Distributed Monoliths



Define stable contract/API instead

Authorization
Service



Add
auth

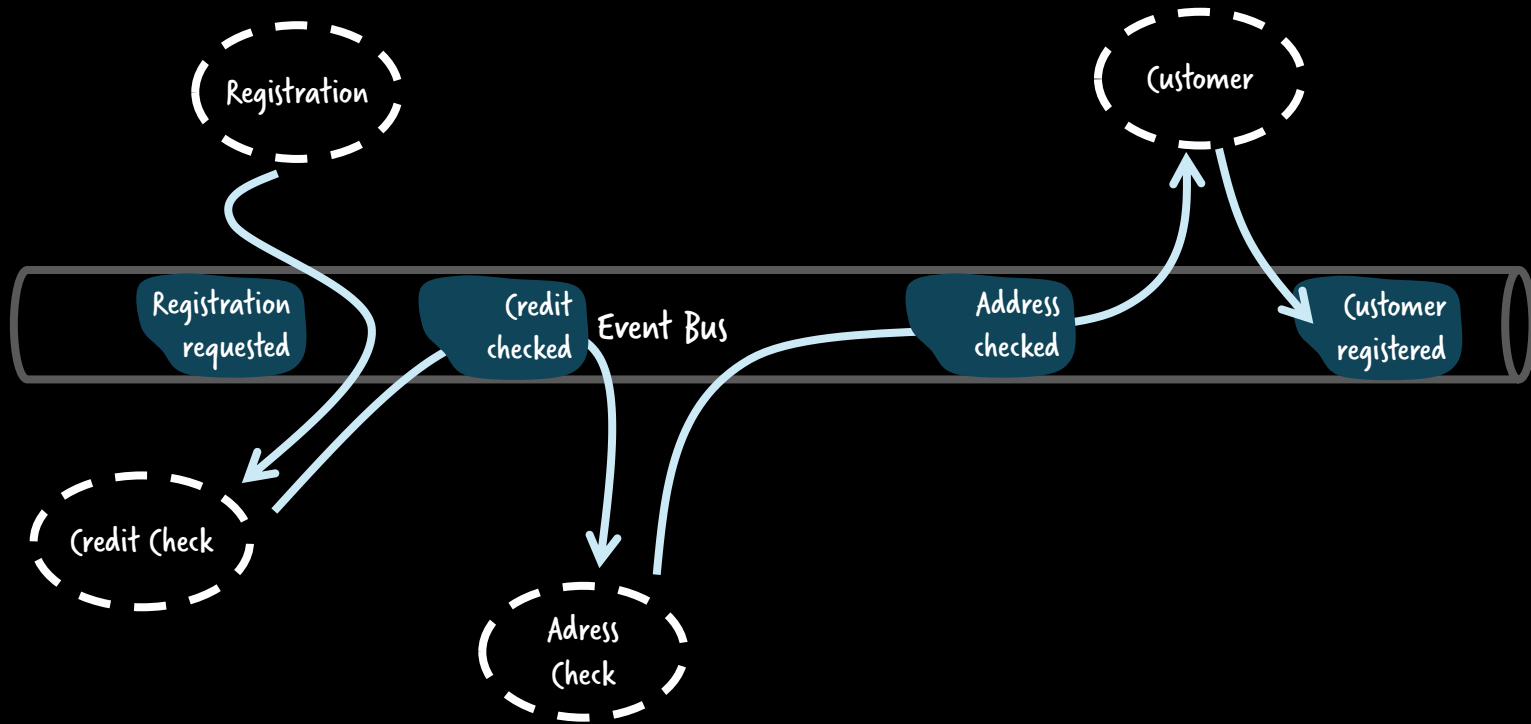
Document Context

Page
Context

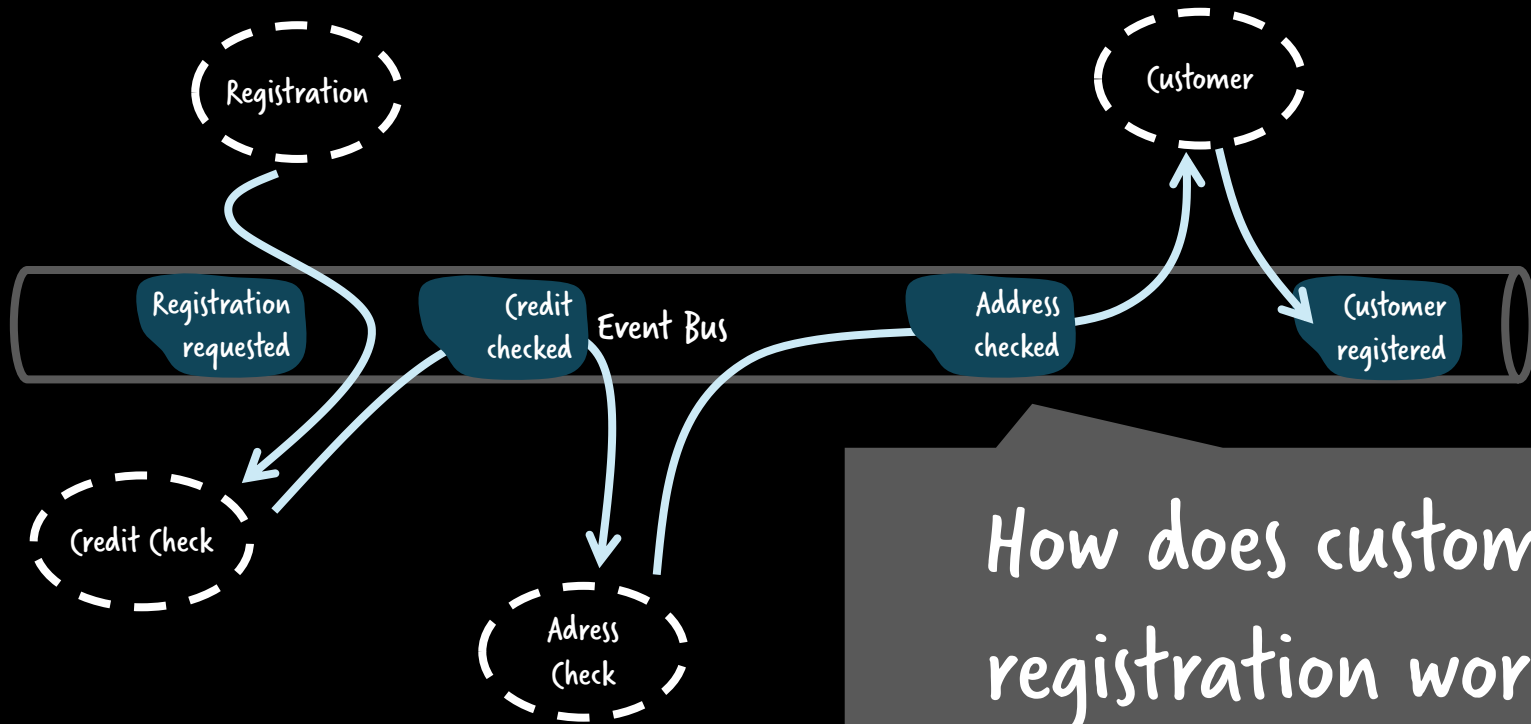
...

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.



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.

Monitoring Workflows Across Microservices

The screenshot shows the InfoQ website interface. At the top, there's a navigation bar with categories like 'Development', 'Architecture & Design', 'AI, ML and Data Engineering', 'Culture & Methods', and 'DevOps'. A search bar and a user profile for 'BERND' are also visible. Below the navigation, there's a featured section with tags like 'Streaming', 'Machine Learning', 'Reactive', 'Microservices', 'Containers', and 'NoSQL'. The main article title is 'Monitoring and Managing Workflows Across Collaborating Microservices' under the 'ARCHITECTURE & DESIGN' category. The author is Bernd Rucker, and it was reviewed by Daniel Bryant. The article includes a 'Key Takeaways' section with three bullet points. On the right, there's a 'RELATED CONTENT' section with three related articles.

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

InfoQ Development Architecture & Design AI, ML and Data Engineering Culture & Methods DevOps

En | 中文 | 日本 | Fr | Br
714,642 Feb unique visitors

Search

BERND

NEW Videos with Transcripts

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 Rucker [FOLLOW](#)

reviewed by Daniel Bryant [FOLLOW](#)

Key Takeaways

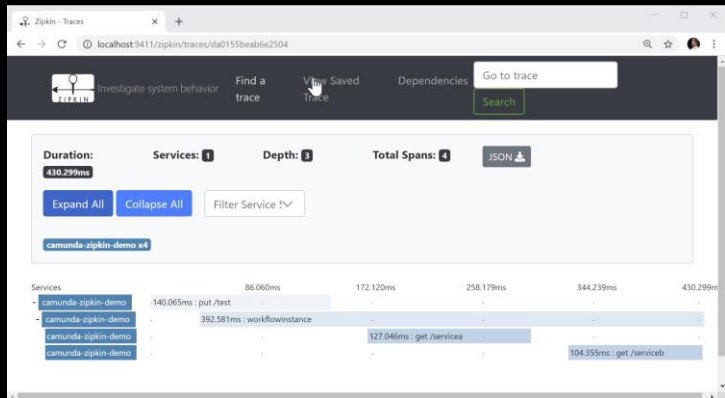
- Peer-to-peer communication between components can lead to emergent behavior, which is challenging for developers, operators and business analysts to understand.
- You need to make sure to have the overview of all of the backwards-and-forwards communication that is going on in order to fulfill a business 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



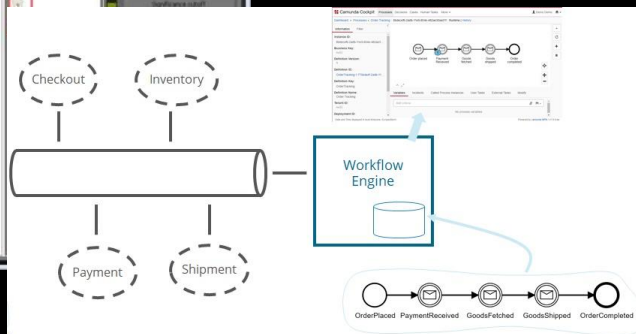
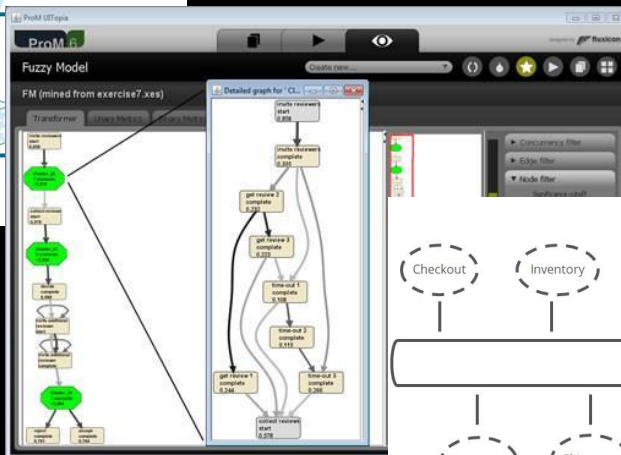
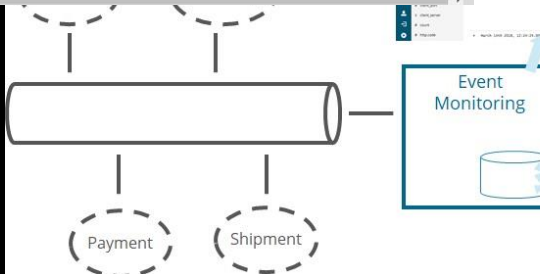
Distributed Tracing

Typical approaches

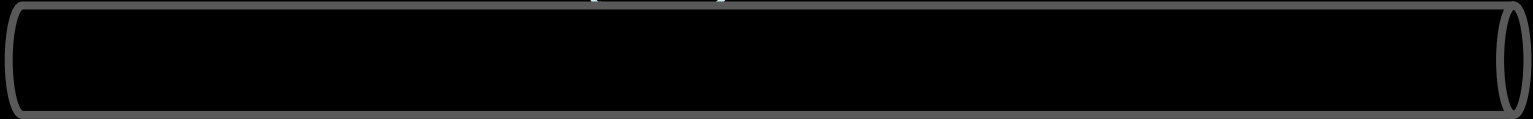
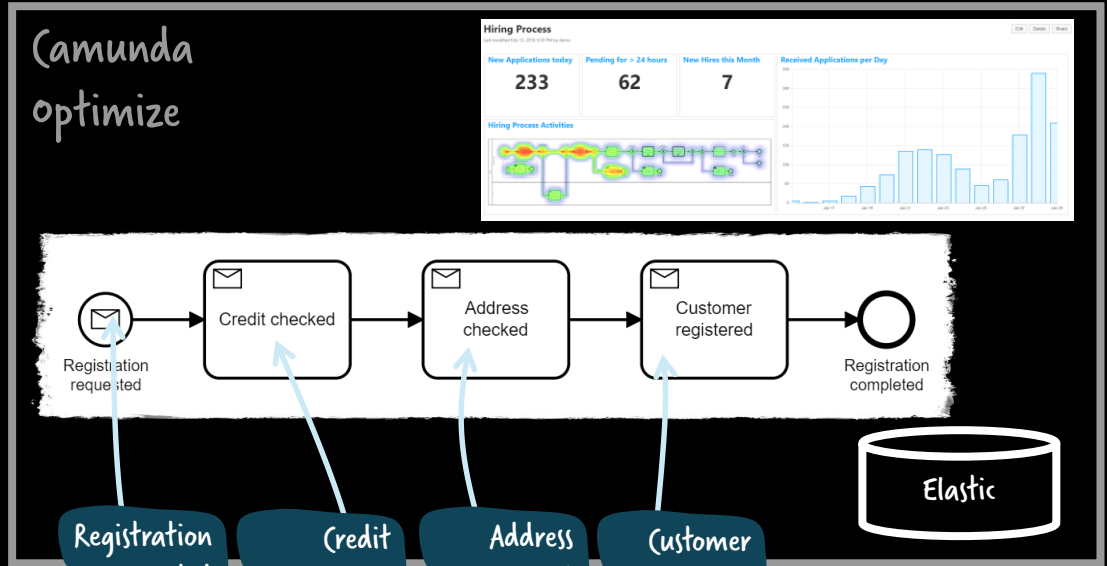
Data Lake / Event Monitoring

Process Mining

Process Tracking



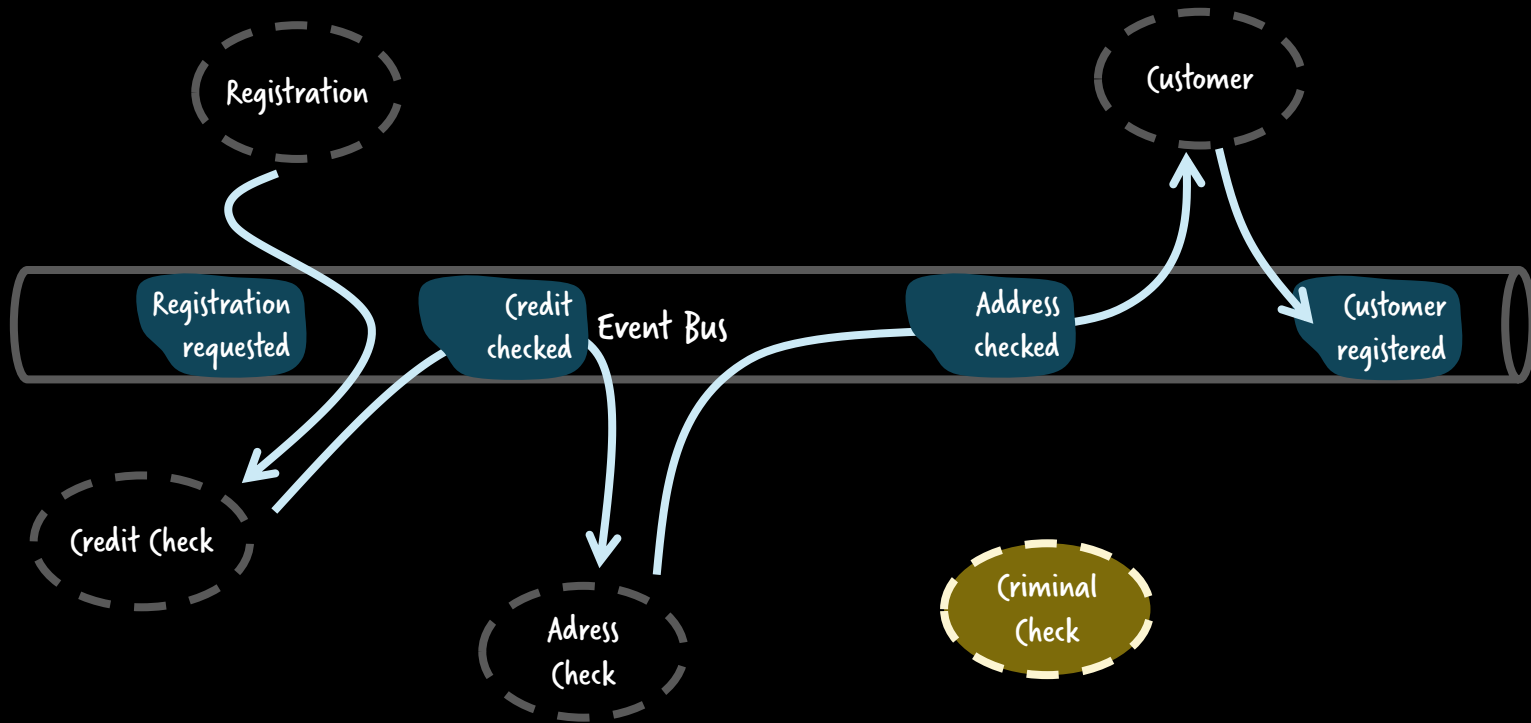
What we currently build with customers...



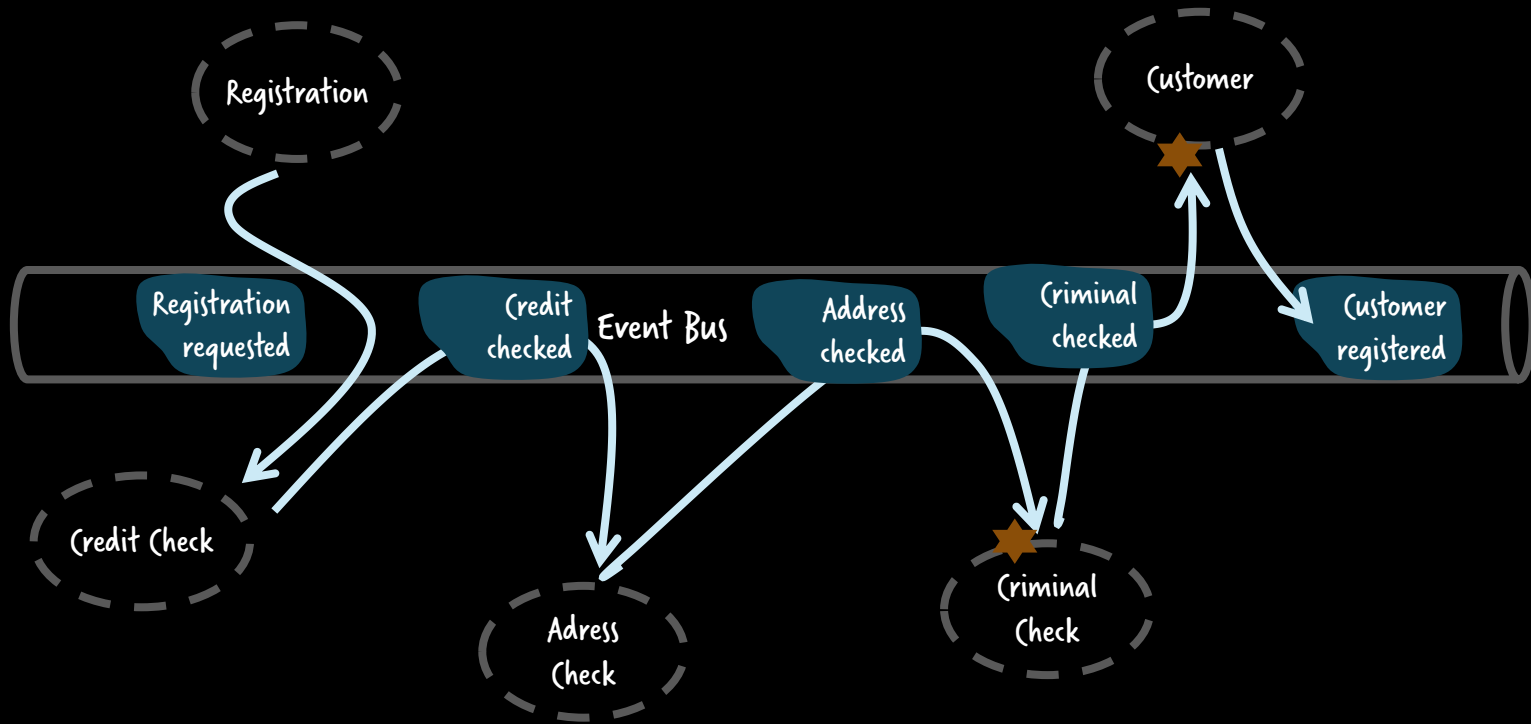
All great – until you have to move...



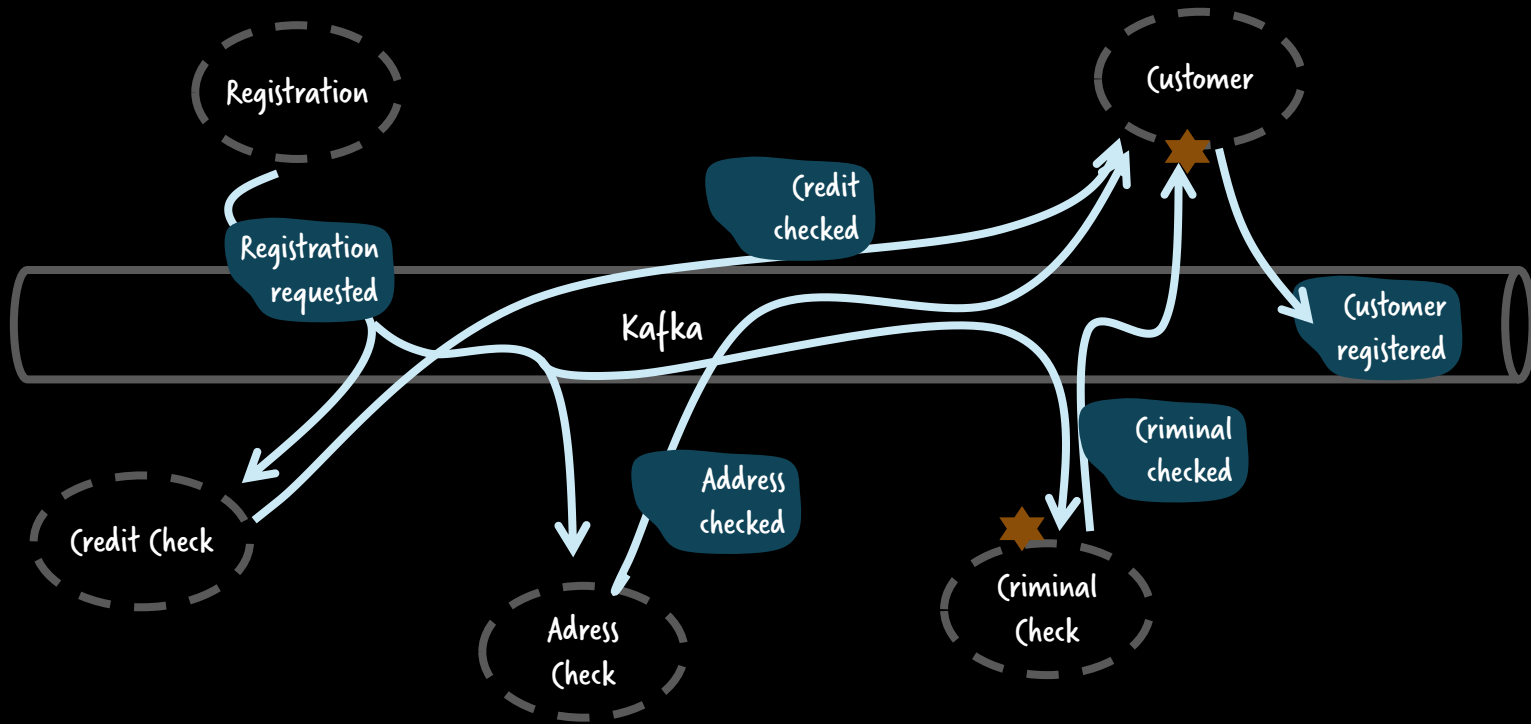
Changes required for an additional check



Changes required for an additional check

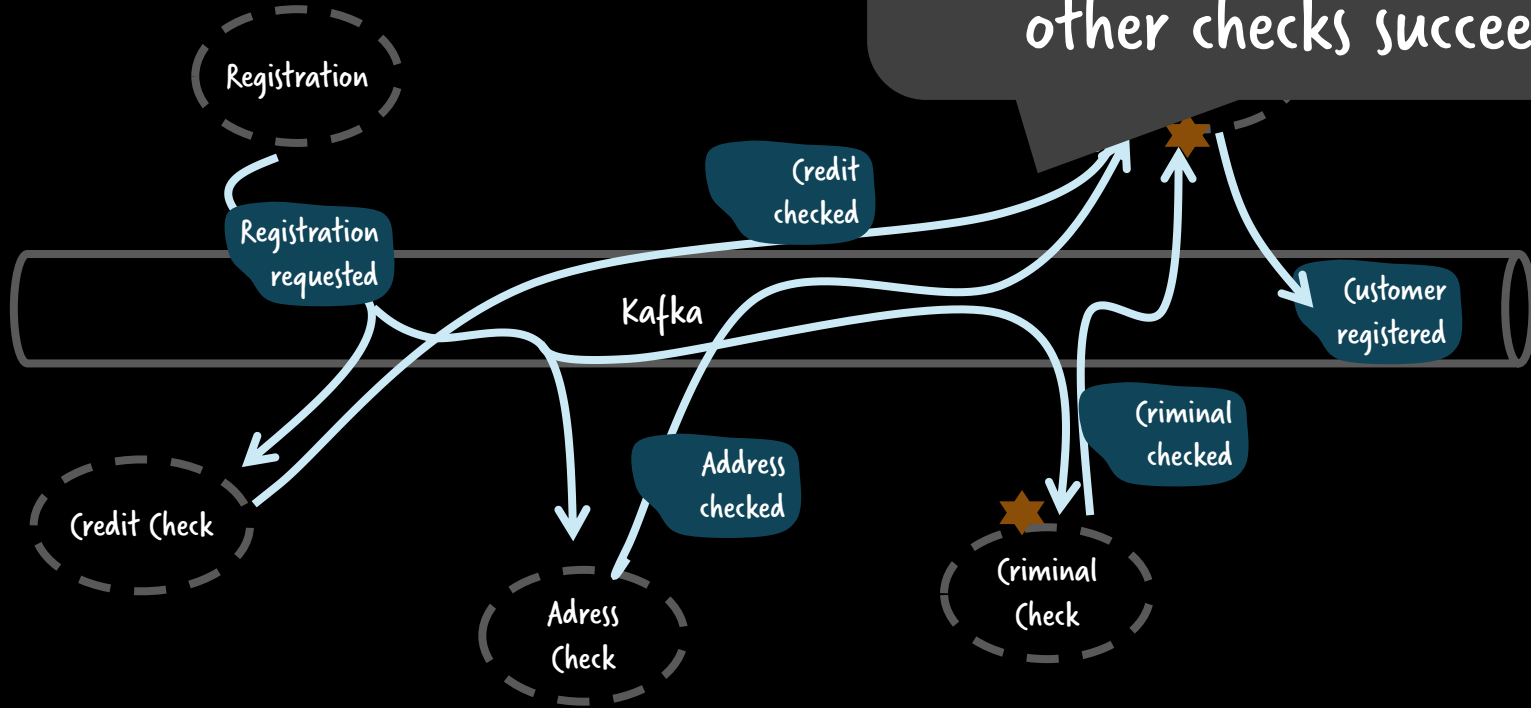



Alternative flow



Alternative flow

„(credit checks got more expensive, do that only if all other checks succeed“

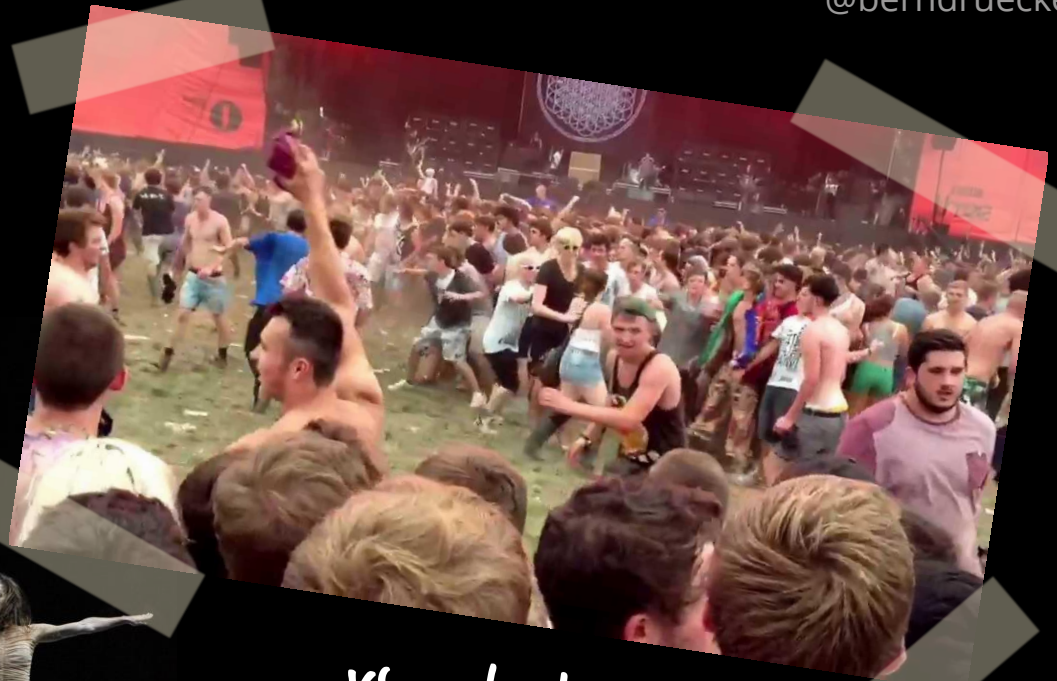




Keep it stable, just move sticks with yellow color to the top.

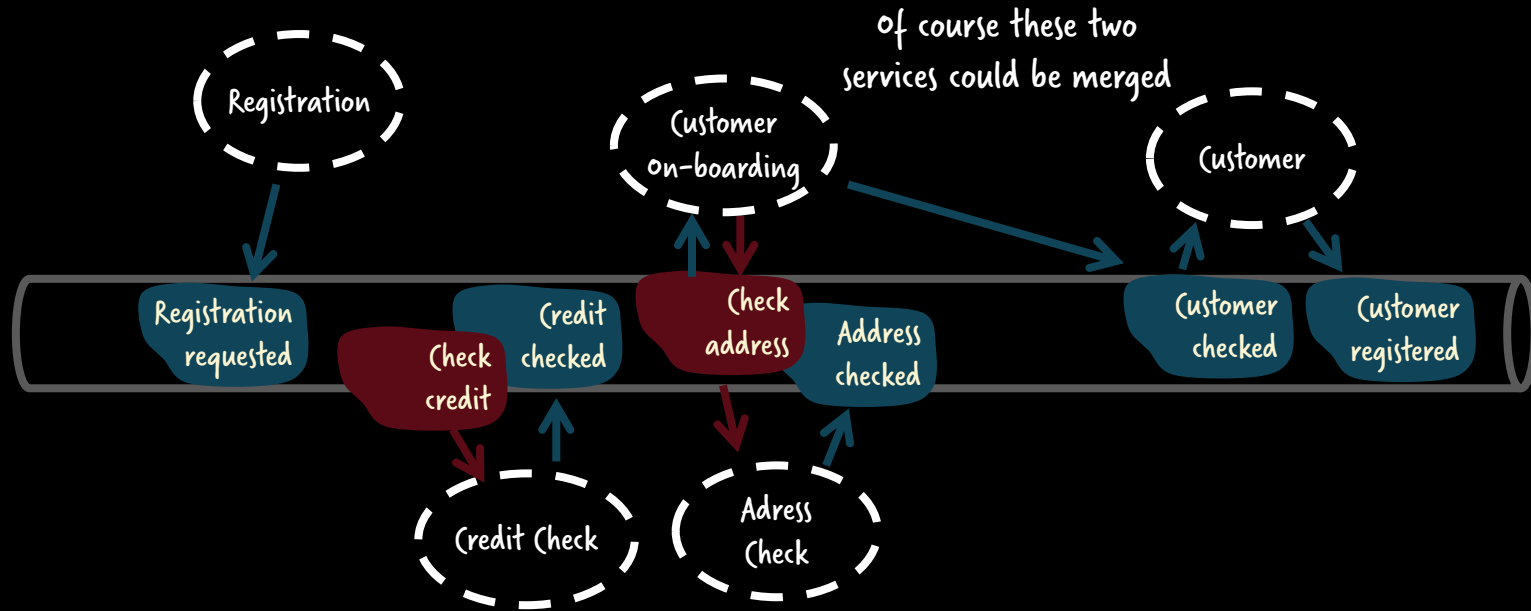
How hard can it be?

What we wanted

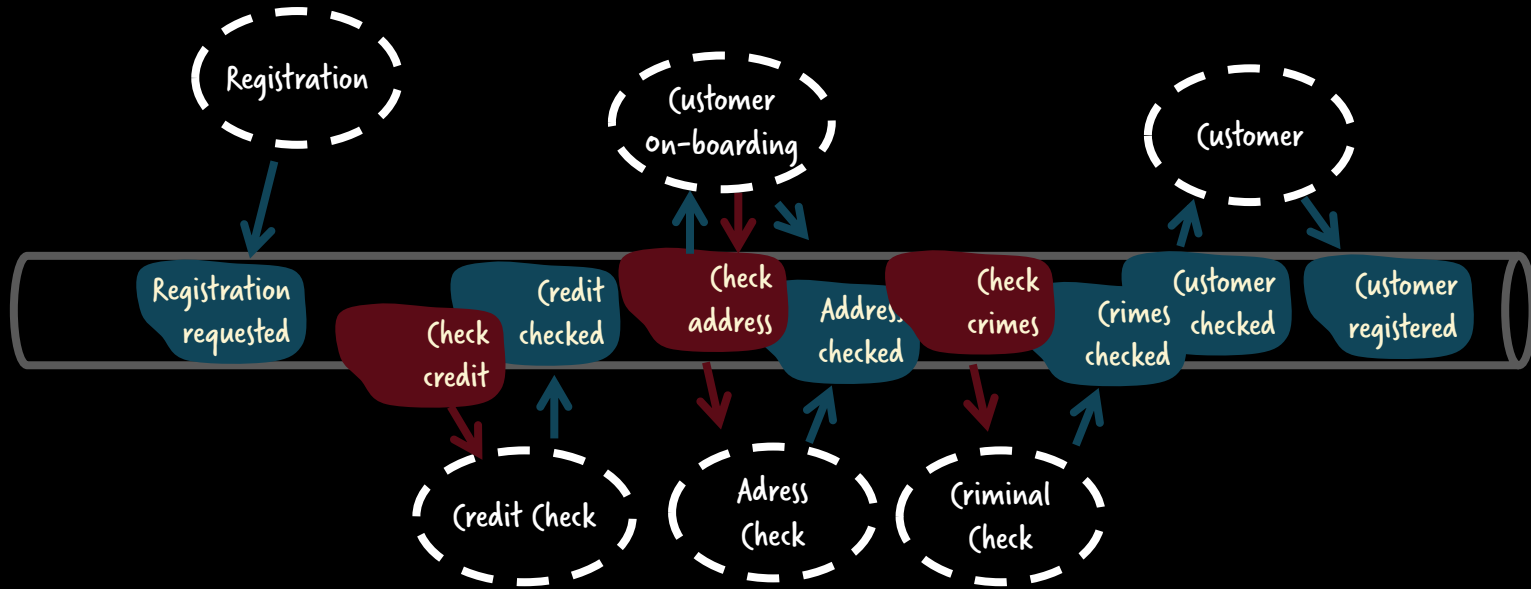


vs. what we got

orchestration

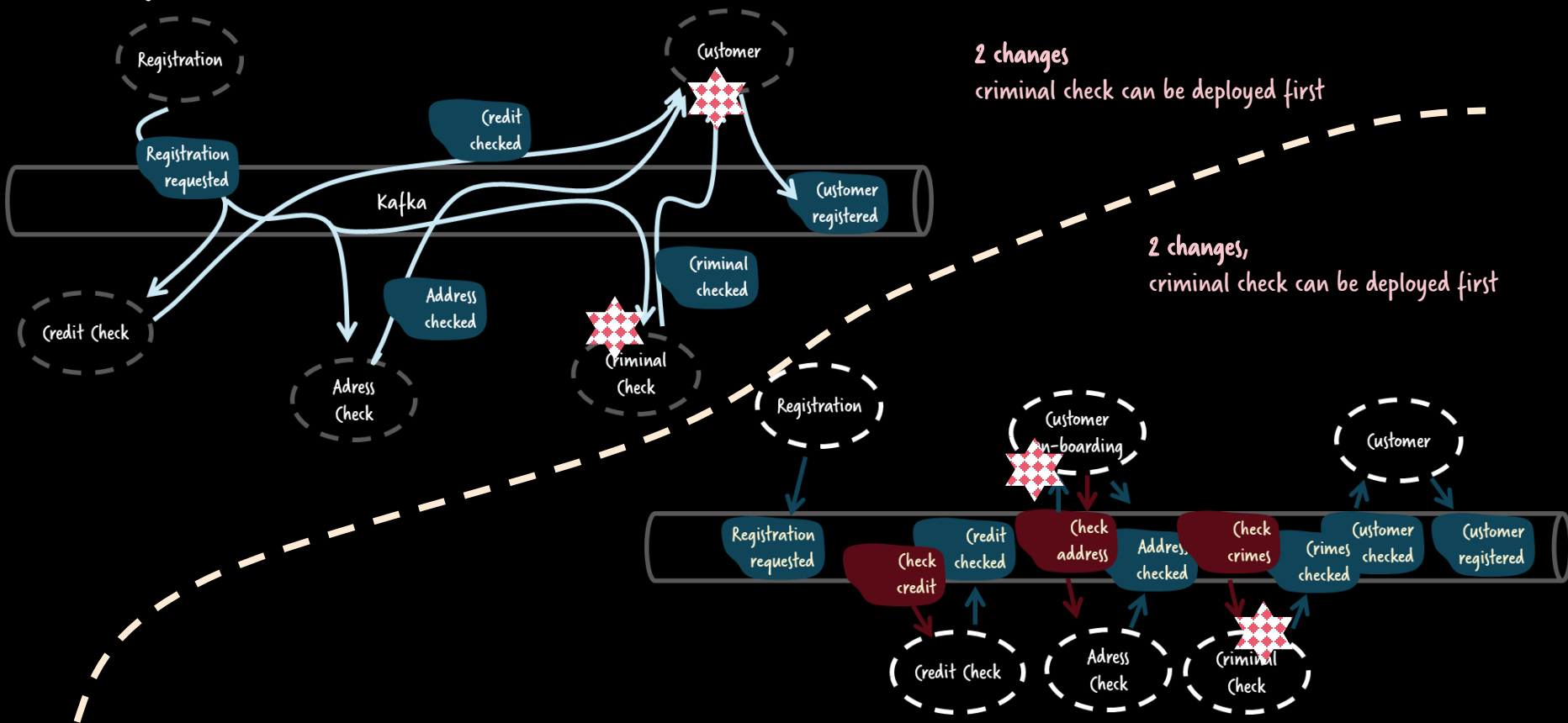


Changes

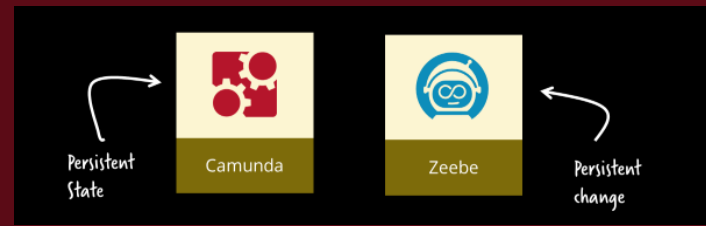


See also <https://www.infoworld.com/article/3391592/how-to-tame-event-driven-microservices.html>

Comparison

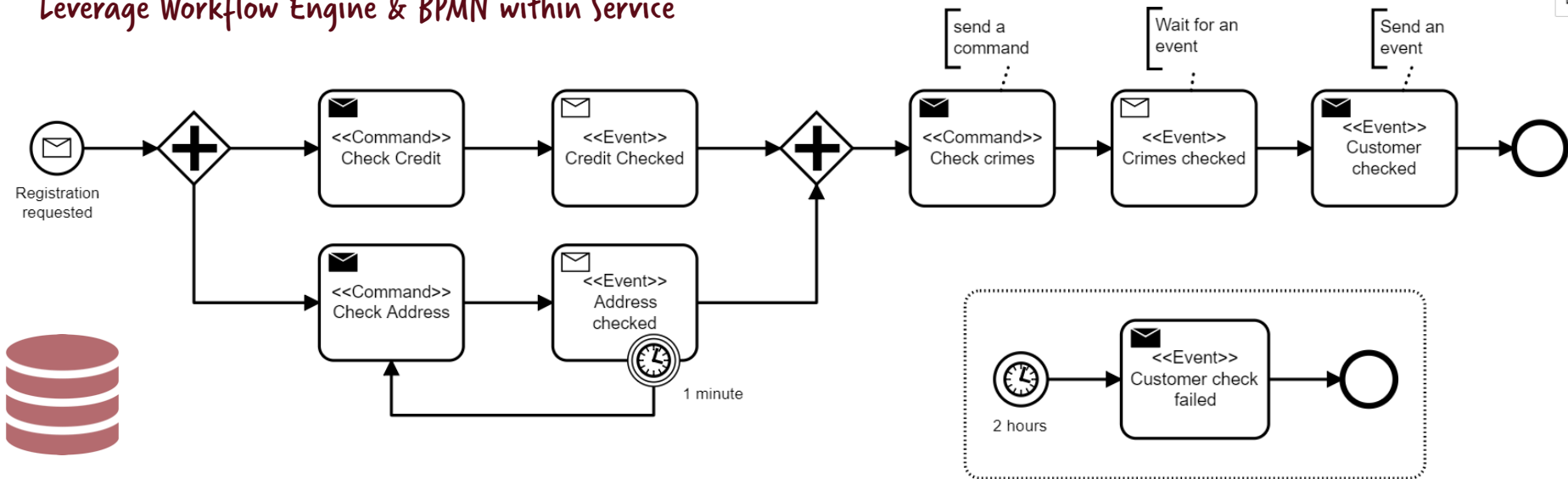


In my world...

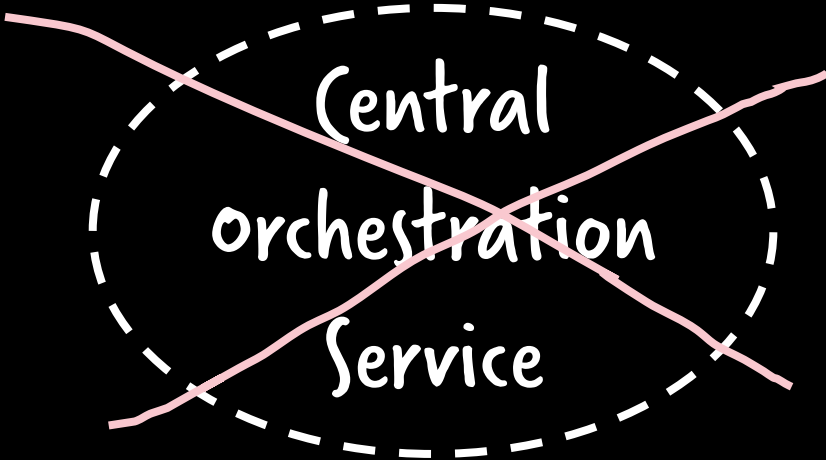
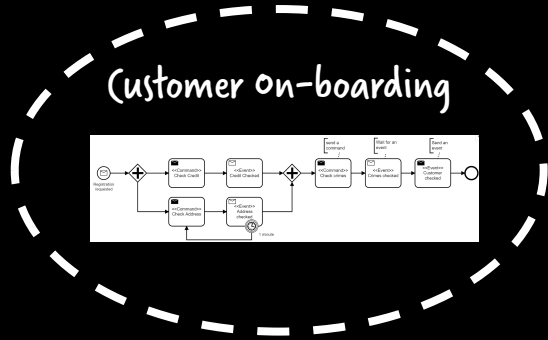


Customer on-boarding

Leverage Workflow Engine & BPMN within Service



Local orchestration



Software Development Architecture and Design 2019 Q1 Graph



<http://infoq.link/architecture-trends-2019>



Recap 2

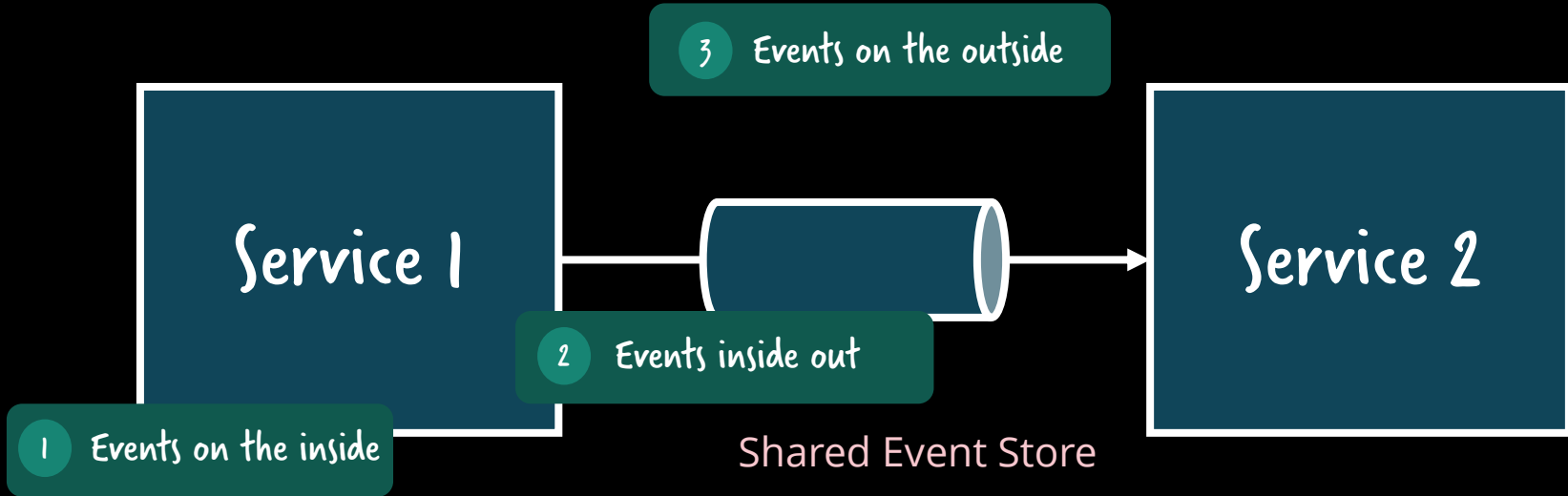
Commands vs. Events: Decide about the direction of dependencies

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



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

Want to see code?

The screenshot shows the GitHub interface for the repository 'berndruecker / flowing-retail'. At the top, there is a search bar and navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. The repository name is displayed in blue. Below the name, there are buttons for 'Unwatch' (87), 'Unstar' (569), and 'Fork' (161). A navigation bar includes 'Code', 'Issues' (5), 'Pull requests' (1), 'Projects' (0), 'Wiki', 'Insights', and 'Settings'. The main description reads: 'Sample application demonstrating an order fulfillment system decomposed into multiple independent components (e.g. microservices). Showing concrete implementation alternatives using e.g. Java, Spring Boot, Apache Kafka, Camunda, Zeebe, ...'. Below this, statistics show '162 commits', '2 branches', '0 releases', and '2 contributors'. A progress bar indicates the repository is Apache-2.0 licensed. The 'Branch: master' dropdown and 'New pull request' button are visible. A commit message from 'berndruecker' is shown: 'switched back to CE (accidentally committed EE)'. A file list follows, including folders like 'docs', 'kafka', 'rest', 'runner', and 'zeebe', and files like '.gitignore', '.travis.yml', 'LICENSE', and 'README.md', each with a brief description and commit date.

berndruecker / flowing-retail

Unwatch 87 Unstar 569 Fork 161

Code Issues 5 Pull requests 1 Projects 0 Wiki Insights Settings

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

Manage topics

162 commits 2 branches 0 releases 2 contributors Apache-2.0

Branch: master New pull request

Create new file Upload files Find File Clone or download

berndruecker switched back to CE (accidentally committed 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 committed 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

Events on the inside

Nothing for the faint of heart...

Events on the outside



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-integration-and-how-to-avoid-them.html>

InfoQ
neue

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

THE NEW STACK

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

