3 Common Pitfalls in Microservice Integration and How to Avoid Them

@berndruecker

Distributed systems

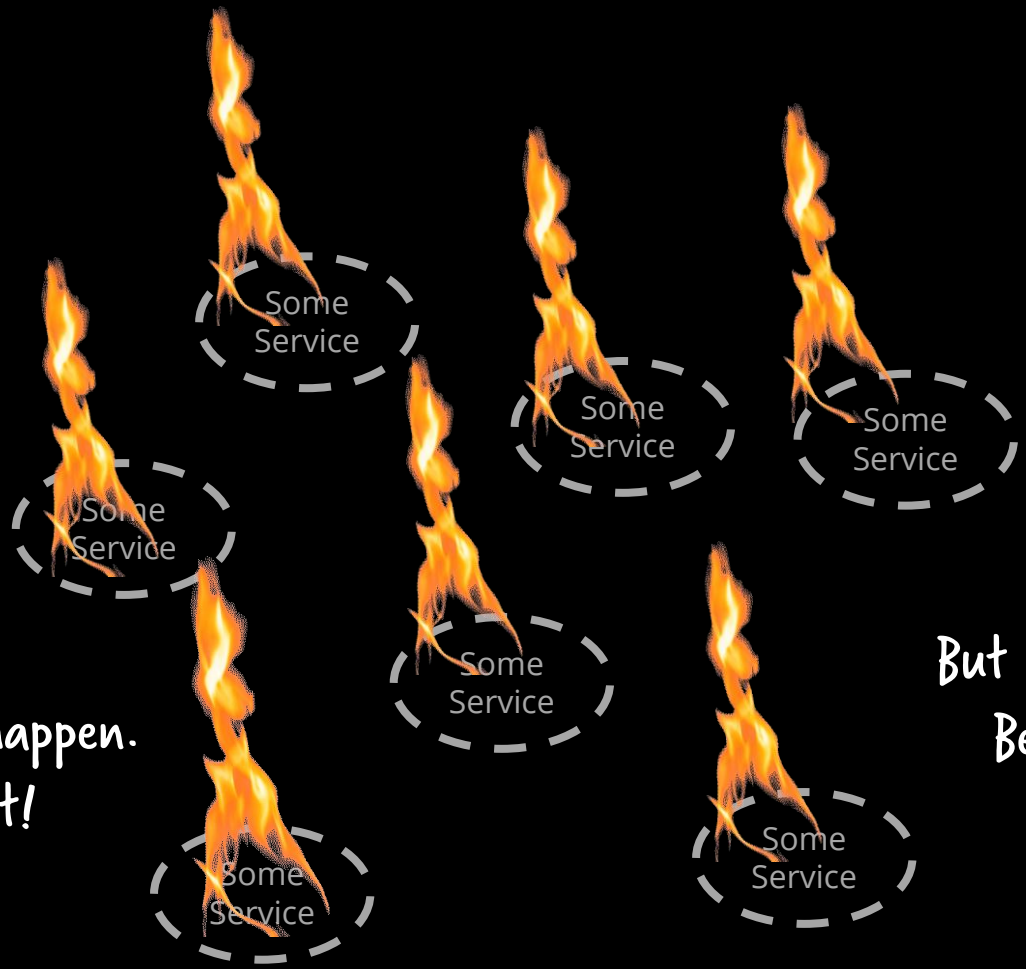
Software Development

Architecture and Design 2019 Q1 Graph

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

InfoQ

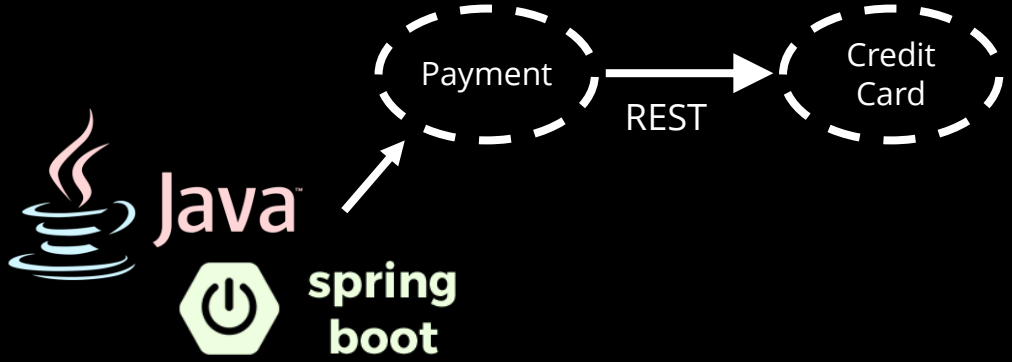
https://www.infoworld.com/article/3254777/application-development/3-common-pitfalls-of-microservices-integrationand-how-to-avoid-them.html



Failure will happen.
Accept it!

But keep it local!
Be resilient.

Let's start with a simple example



https://github.com/berndruecker/flowing-retail

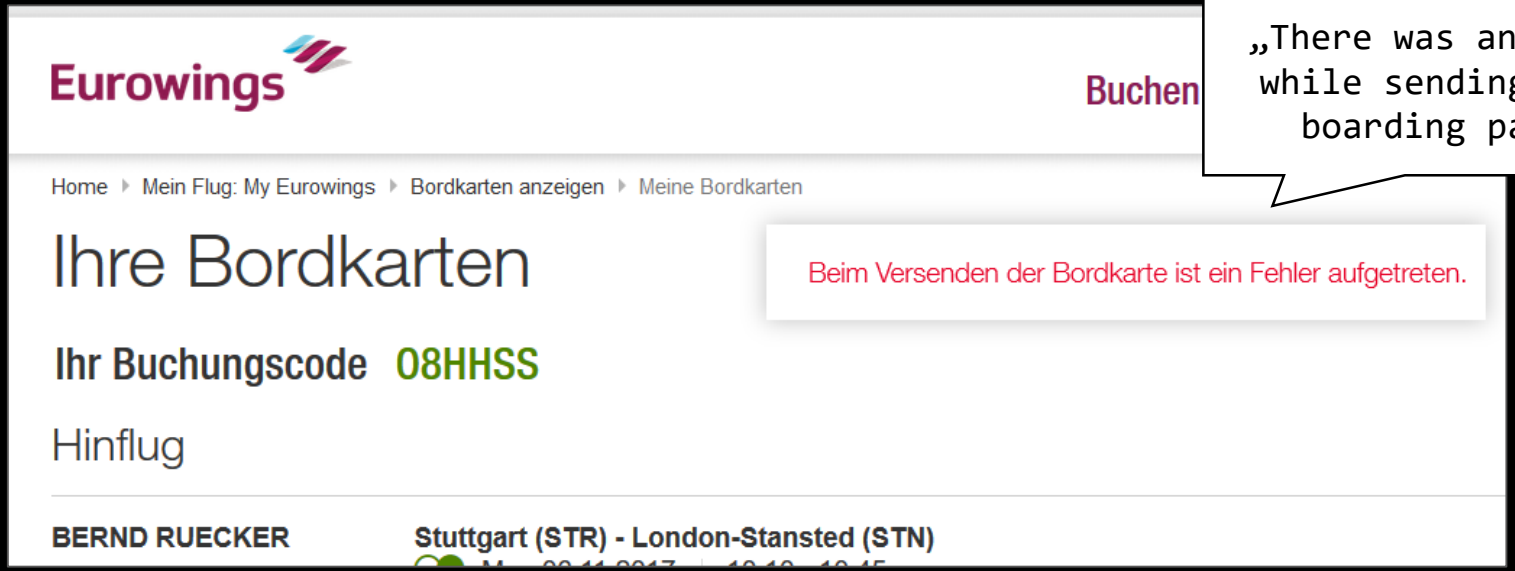
Circuit Breaker

Photo by CITYEDV, available under Creative Commons CC0 1.0 license.

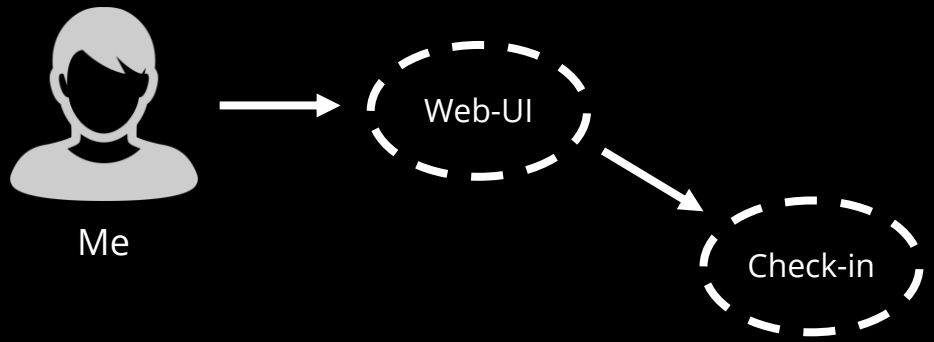
Fail fast
is important

Fail fast
is important
but not enough!

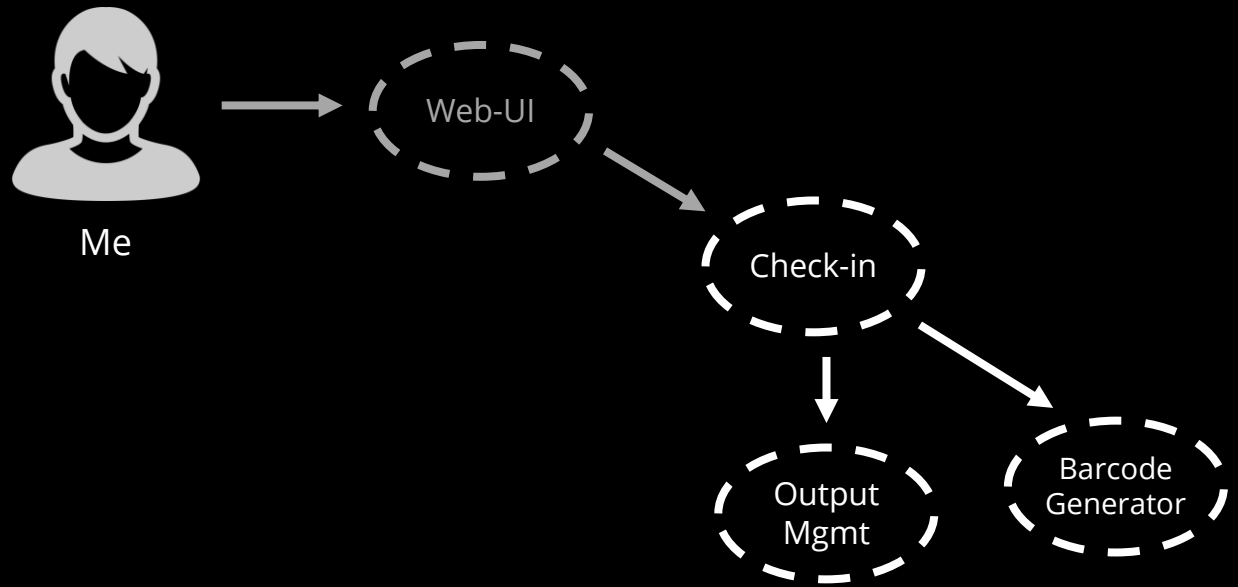
Photo by Tookapic, available under Creative Commons CC0 1.0 license.


Eurowings
Buchen
„There was an error while sending your boarding pass“
Home ▸ Mein Flug: My Eurowings ▸ Bordkarten anzeigen ▸ Meine Bordkarten
Ihre Bordkarten
Beim Versenden der Bordkarte ist ein Fehler aufgetreten.
Ihr Buchungscode O8HHSS
Hinflug
BERND RUECKER
Stuttgart (STR) - London-Stansted (STN)

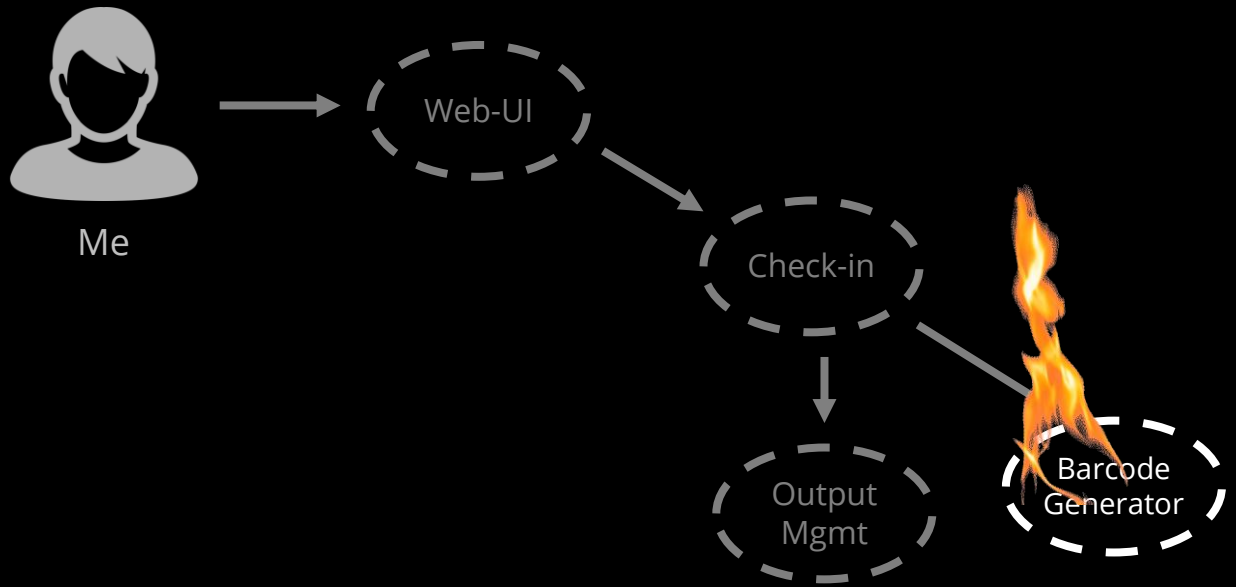
Current situation



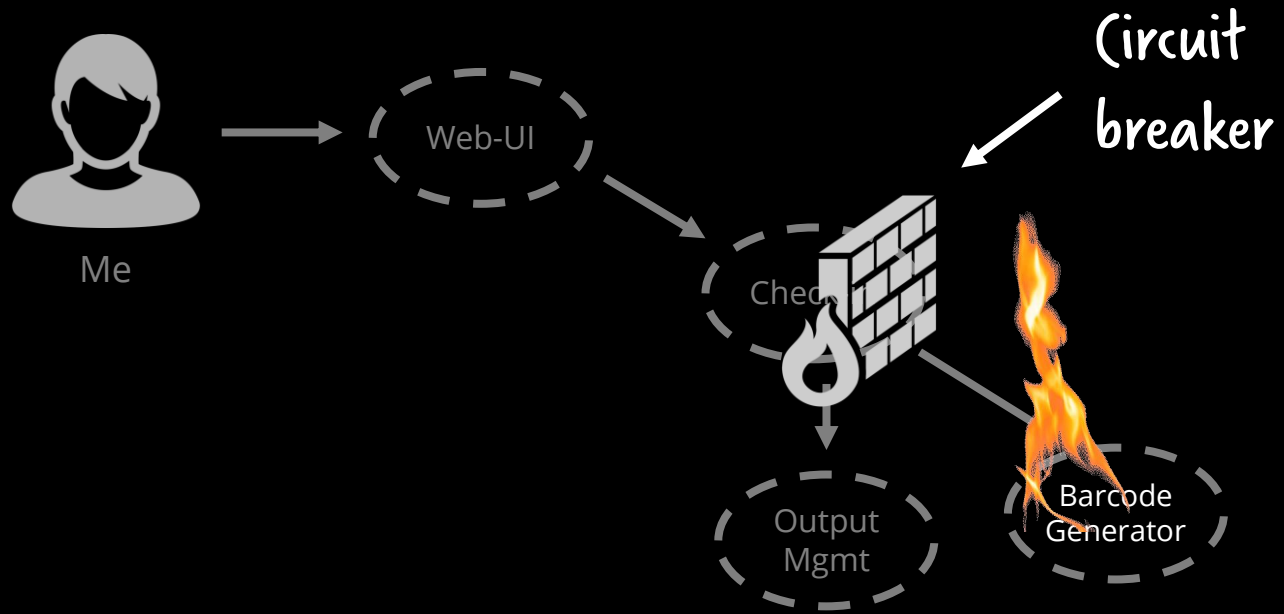
Current situation



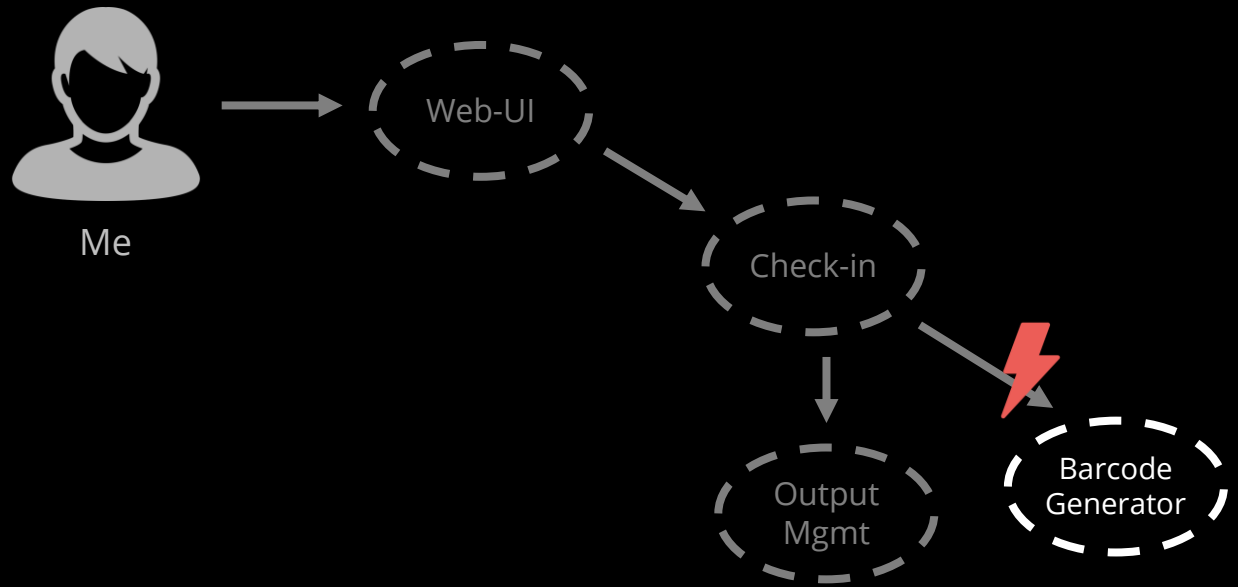
Current situation



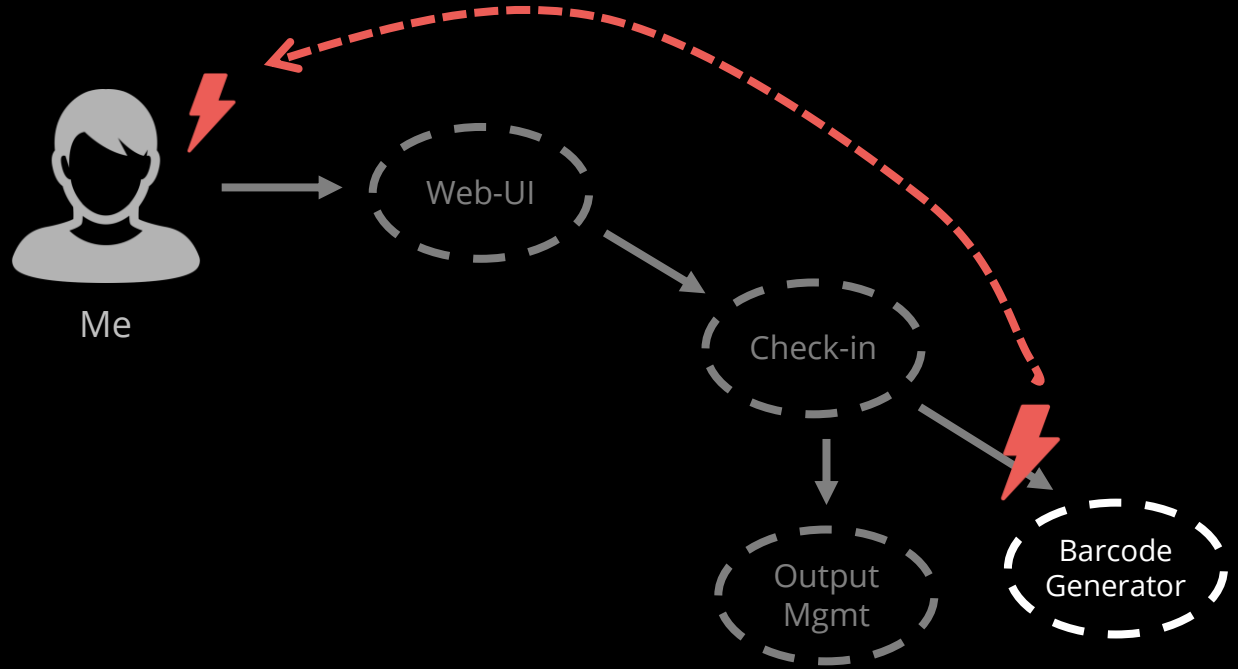
Current situation



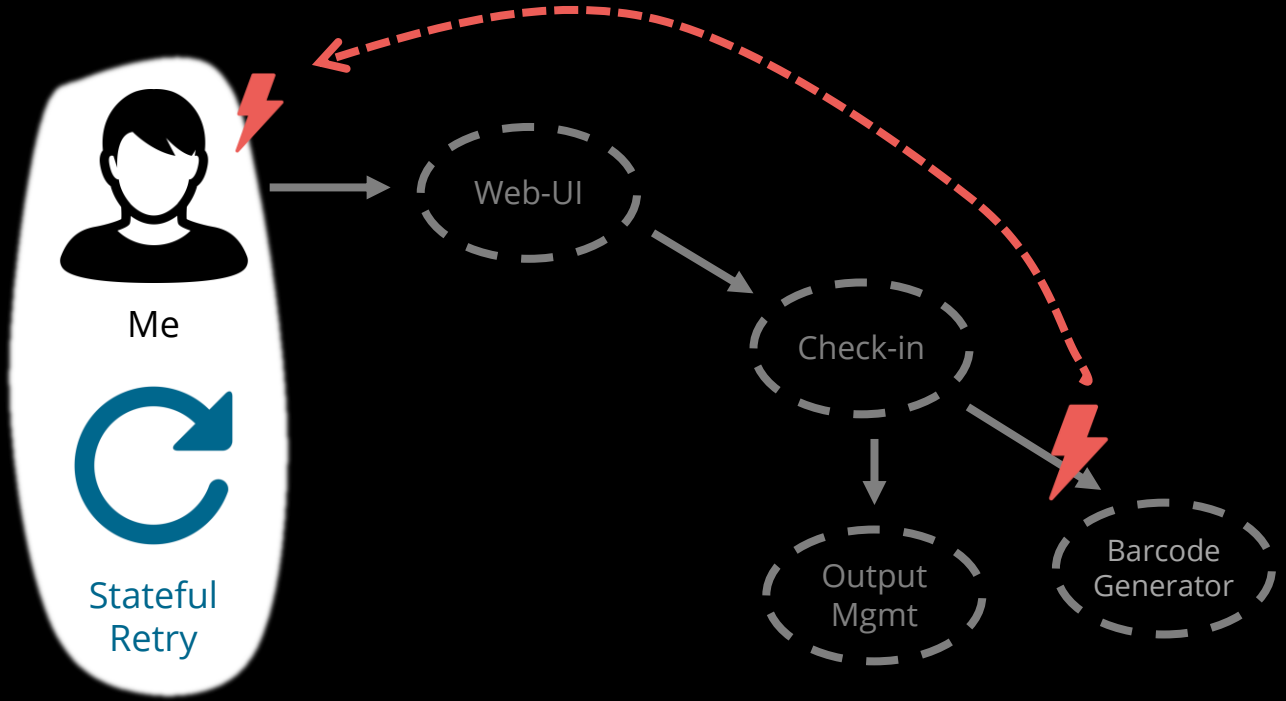
Current situation – the bad part

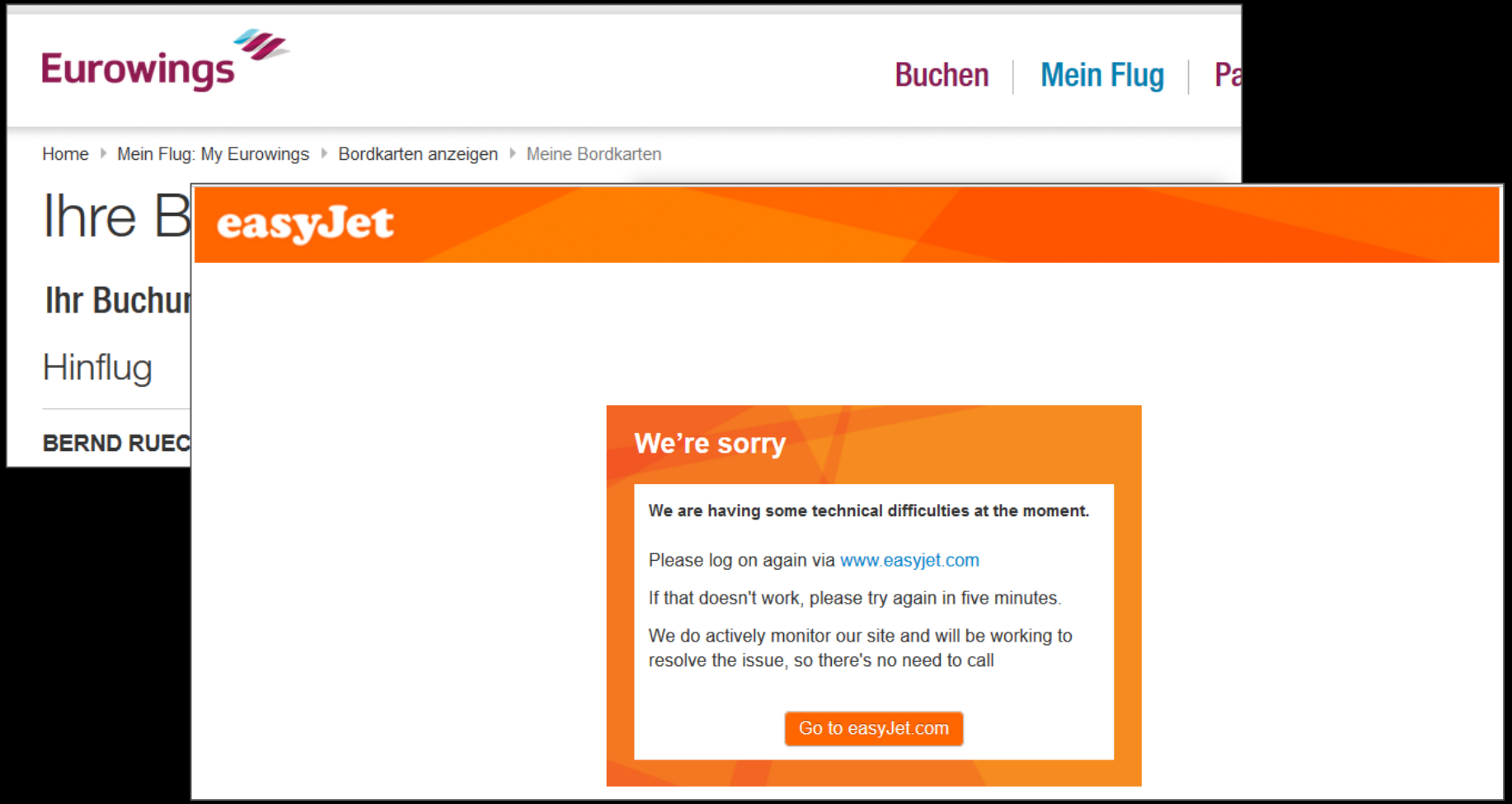


Current situation – the bad part



Current situation – the bad part




Eurowings
Buchen
Mein Flug
Home
Mein Flug: My Eurowings
Bordkarten anzeigen
Meine Bordkarten
Ihre B
Ihr Buchu
Hinflug
BERND RUEC
easyJet
We're sorry
We are having some technical difficulties at the moment.
Please log on again via www.easyjet.com
If that doesn't work, please try again in five minutes.
We do actively monitor our site and will be working to resolve the issue, so there's no need to call
Go to easyJet.com

easyJet

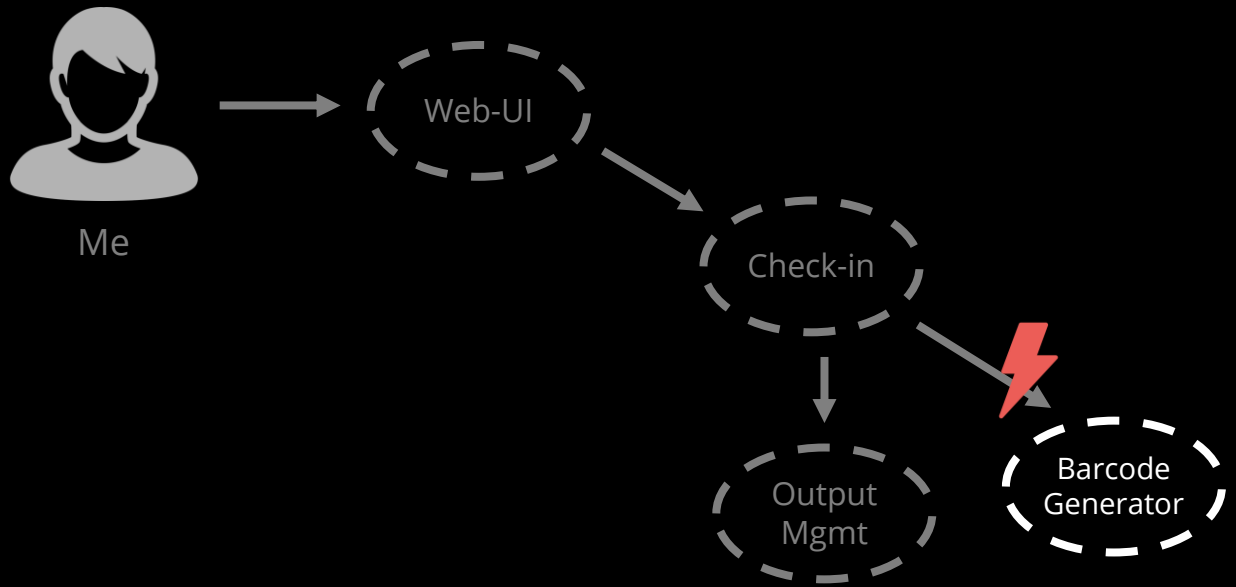
...I just made this up...

We're sorry

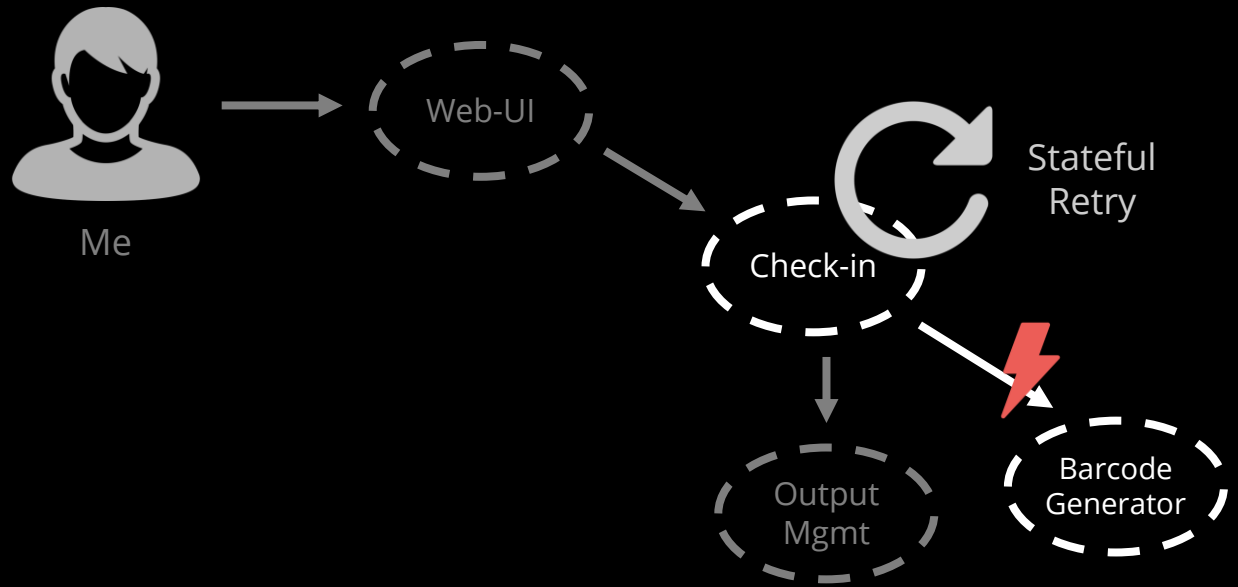
We are having some technical difficulties and cannot present you your boarding pass right away.

But we do actively retry ourselves, so lean back, relax and we will send it on time.

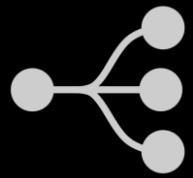
Possible situation – much better!



Possible situation – much better!



Handling State



Persist thing
(Entity, Document, Actor, ...)

State machine or workflow engine

Typical concerns

DIY = effort, accidental complexity

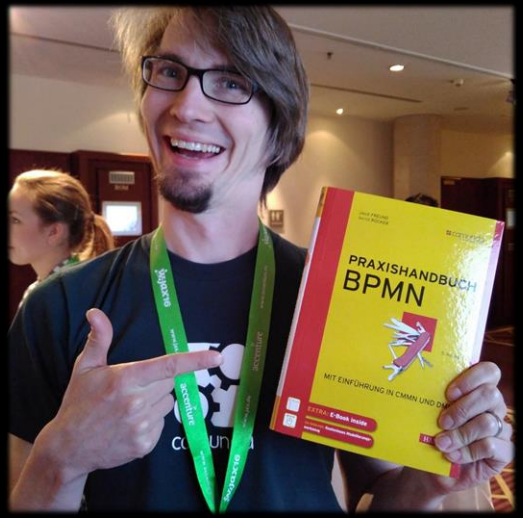
Scheduling, Versioning, operating, visibility, scalability, ...

Complex, proprietary, heavyweight, slow, [illegible], developer adverse

Warning:
Contains Opinion

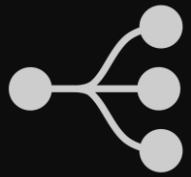
Bernd Ruecker

Co-founder and
Chief Technologist of
Camunda



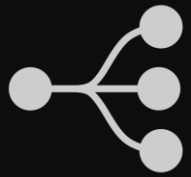
Berlin, Germany

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



Workflow engines,
state machines

It is
relevant
in modern
architectures



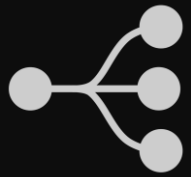
Workflow engines,
state machines

UBER CADENCE

AWS Step Functions

Silicon valley
has recognized

conductor
NETFLIX OSS



Workflow engines, state machines

UBER CADENCE

There are lightweight open source options

zeebe by Camunda

AWS Step Functions

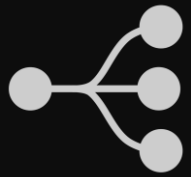
camunda

jBPM

Activiti™

conductor

NETFLIX OSS



Workflow engines,
state machines

UBER CADENCE

also at scale

zeebe
by Camunda

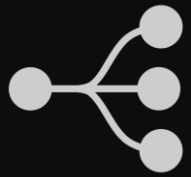
AWS Step Functions

camunda

jBPM

conductor
NETFLIX OSS

Activiti™



Workflow engines,
state machines

UBER CADENCE

for todays demo

zeebe by Camunda

AWS Step Functions

camunda

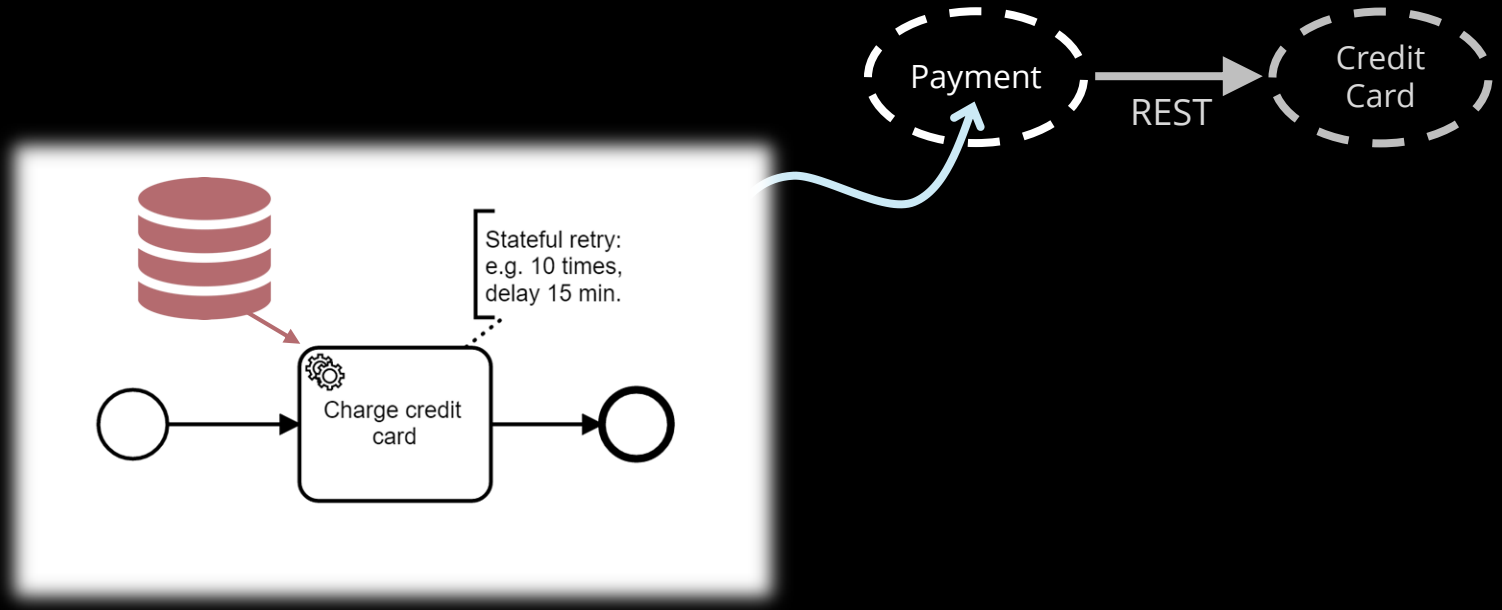
jBPM

conductor
NETFLIX OSS

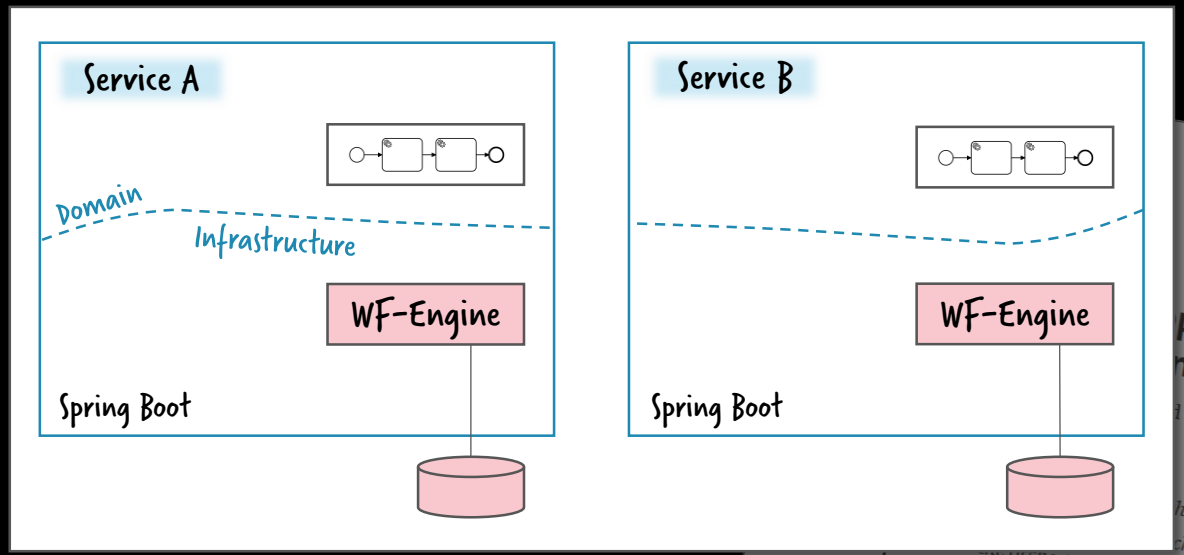
Activiti™

https://github.com/berndruecker/flowing-retail

Now you have a state machine!



Manigfold architecture options



ptions to run a
ne

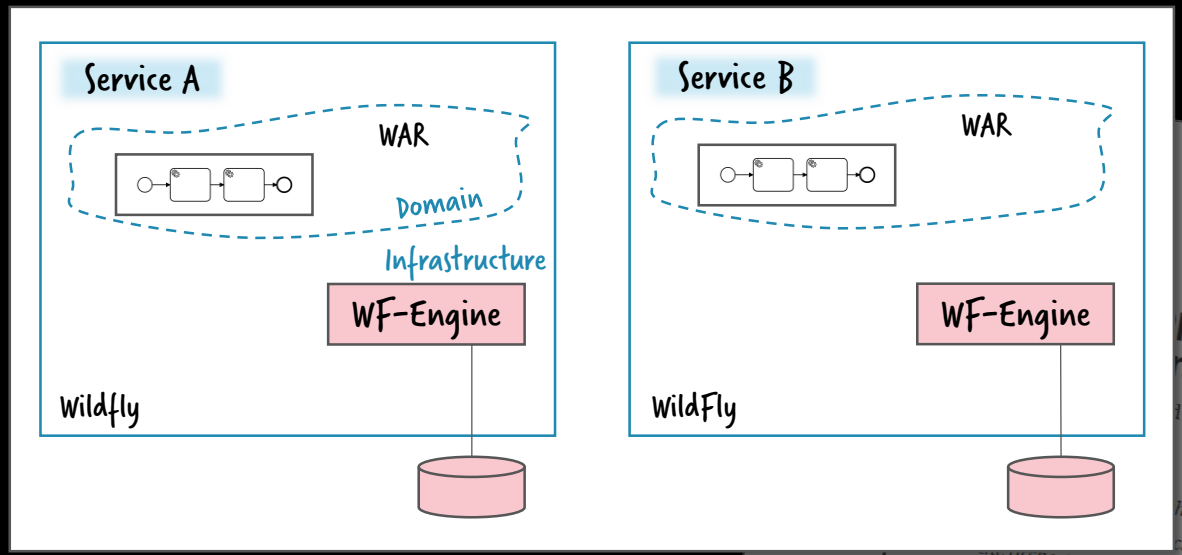
l and asked (translated into my own words and

hestrating two or three CRUD-Services to do
chitects want to use your workflow engine for this
workflow? Currently we run one big central cluster for the workflow engine—
won't this get a mess?"

These are valid questions which recently we get asked a lot, especially
context of microservices, modern SOA initiatives

https://blog.bernd-ruecker.com/architecture-options-to-run-a-workflow-engine-6c2419902d91

Manigfold architecture options



ptions to run a
ne

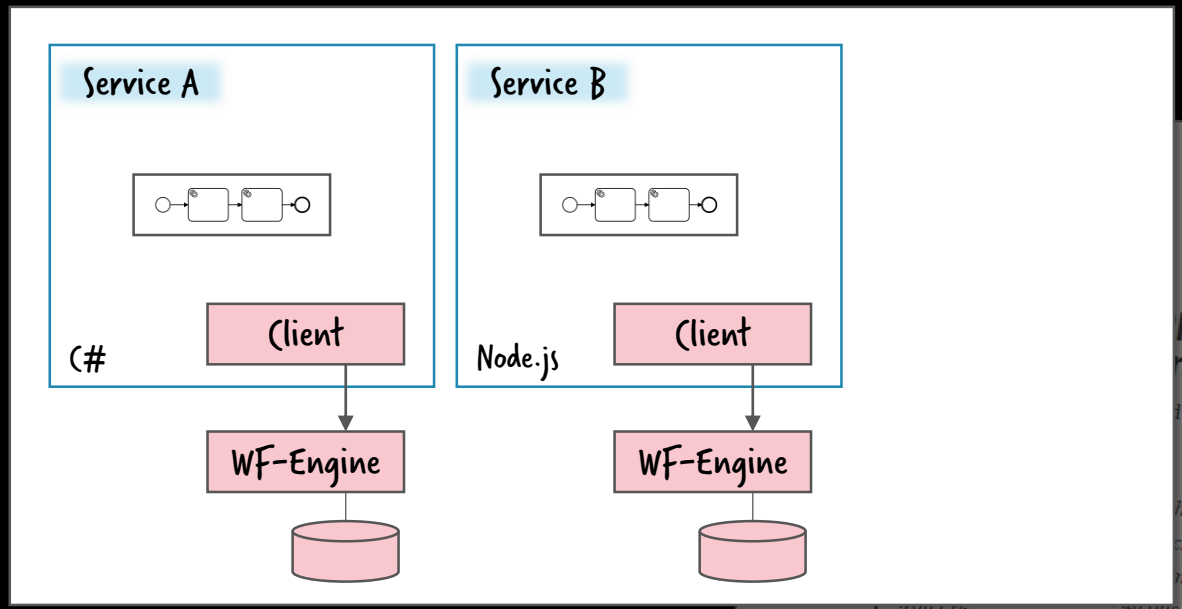
l and asked (translated into my own words and

hestrating two or three CRUD-Services to do
chitects want to use your workflow engine for this
workflow? Currently we run one big central cluster for the workflow engine—
won't this get a mess?"

These are valid questions which recently we get asked a lot, especially
context of microservices, modern SOA initiatives

https://blog.bernd-ruecker.com/architecture-options-to-run-a-workflow-engine-6c2419902d91

Manigfold architecture options



ptions to run a
ne

l and asked (translated into my own words and

hestrating two or three CRUD-Services to do
chitects want to use your workflow engine for this
might be long running. Is this a valid scenario for
one big central cluster for the workflow engine—

"won't this get a mess?"

These are valid questions which recently we get asked a lot, especially in the context of microservices, modern SOA initiatives

https://blog.bernd-ruecker.com/architecture-options-to-run-a-workflow-engine-6c2419902d91

Client

has to implement

Retry

Service Provider

has to implement

Idempotency

10:37 AM 82%

unitedwifi.com

UNITED

Internet
Processing

YYZ DEN
Landing in 0h 53m

DEN
62°

Your transaction is in progress

It may take up to 60 seconds to complete. Please do not close this window or select the "Back" button on your browser.

10:37 AM 82%

Your transaction is in progress

It may take up to 60 seconds to complete.
Please do not close this window or select the "Back" button on your browser.

10:37 AM 82%

...I just made this up...

Your transaction is in progress

Don't worry, it will happen safely –
even if you loose connection.
Feel free to reload this page any time!

Photo by pixabay, available under Creative Commons CC0 1.0 license.

Requirement: Idempotency of services!

Photo by pixabay, available under Creative Commons CC0 1.0 license.

Requirement: Idempotency of services!

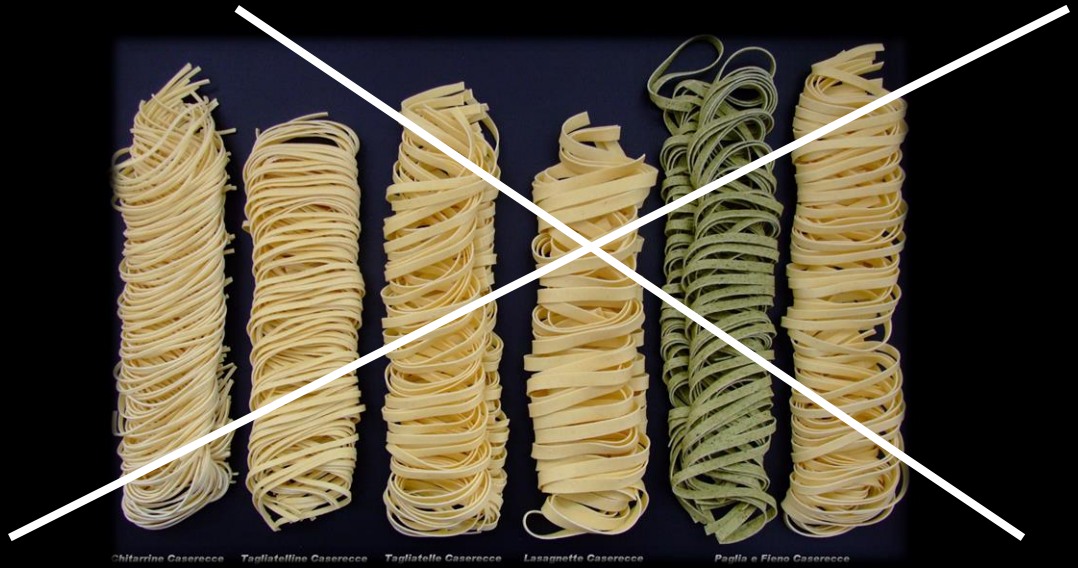
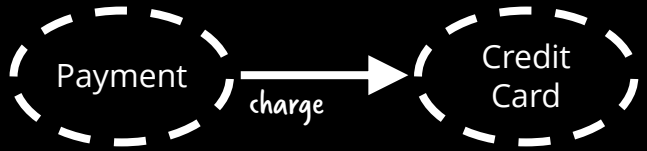


Photo by Chr.Späth, available under Public Domain.

Make every service idempotent!



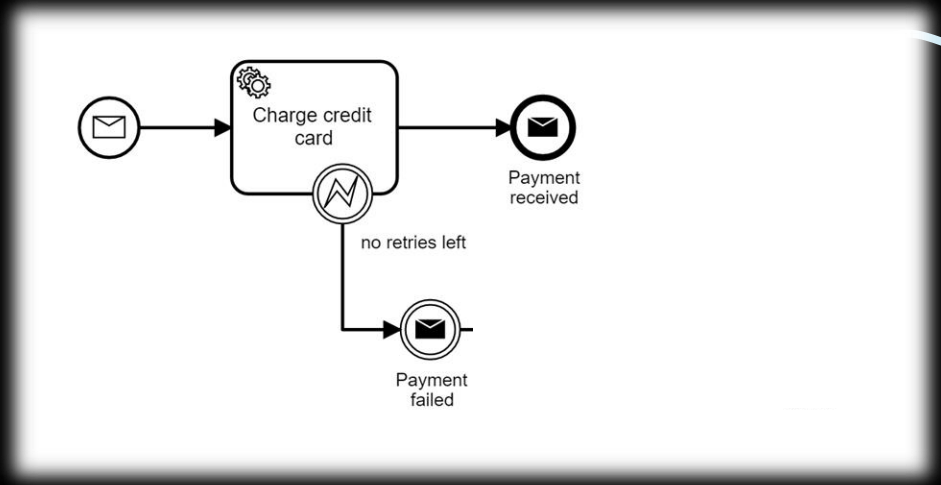
Charge Credit Card
cardNumber
amount

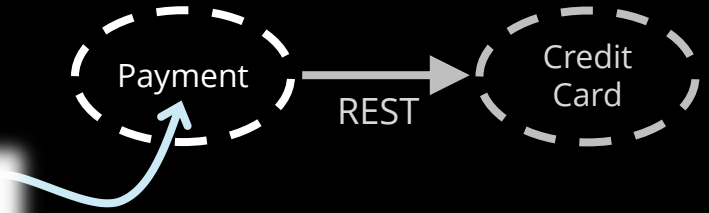
← Not idempotent

Charge Credit Card
cardNumber
amount
transactionId

← Idempotent

Distributed systems introduce complexity you have to tackle!

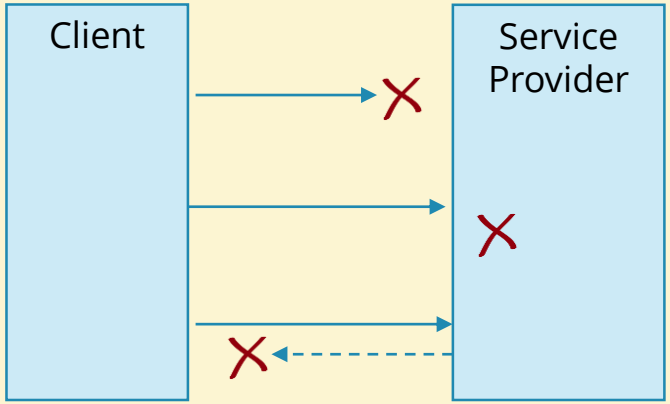




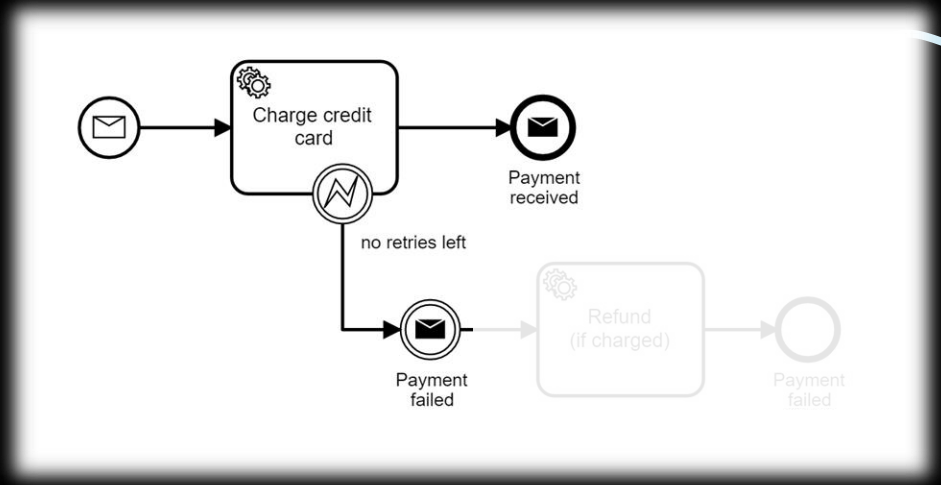
Distributed systems

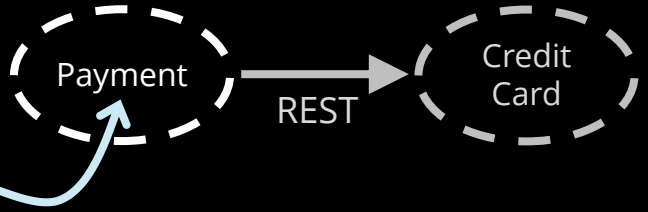
It is impossible to differentiate certain failure scenarios.

Independant of communication style!

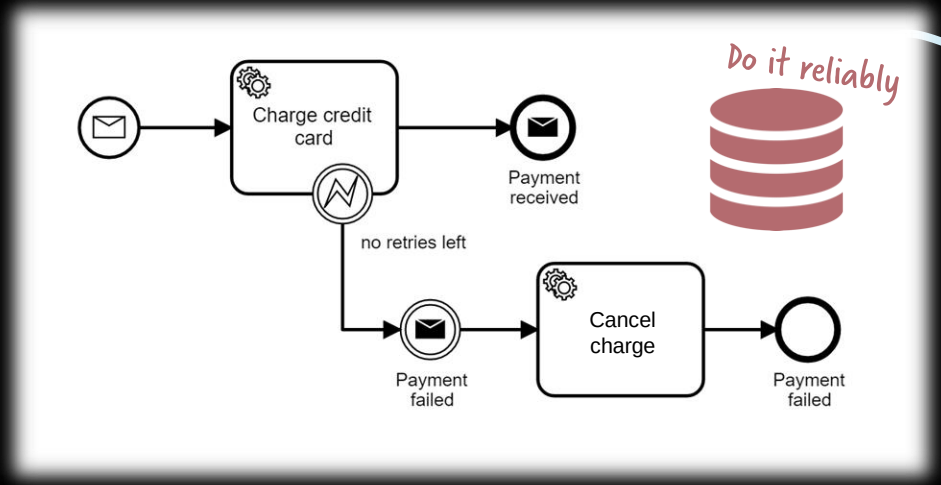


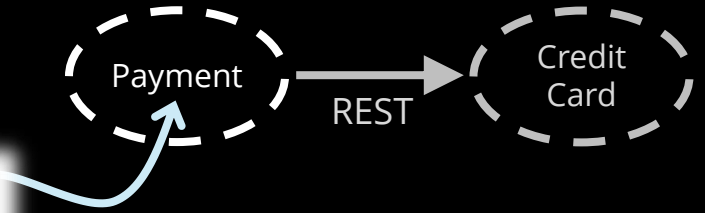
Distributed systems introduce complexity you have to tackle!



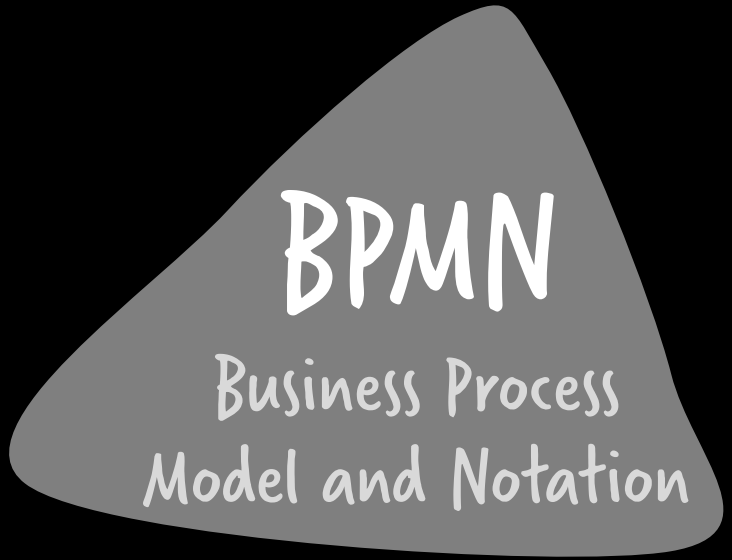


Distributed systems introduce complexity you have to tackle!





ISO Standard
widespread



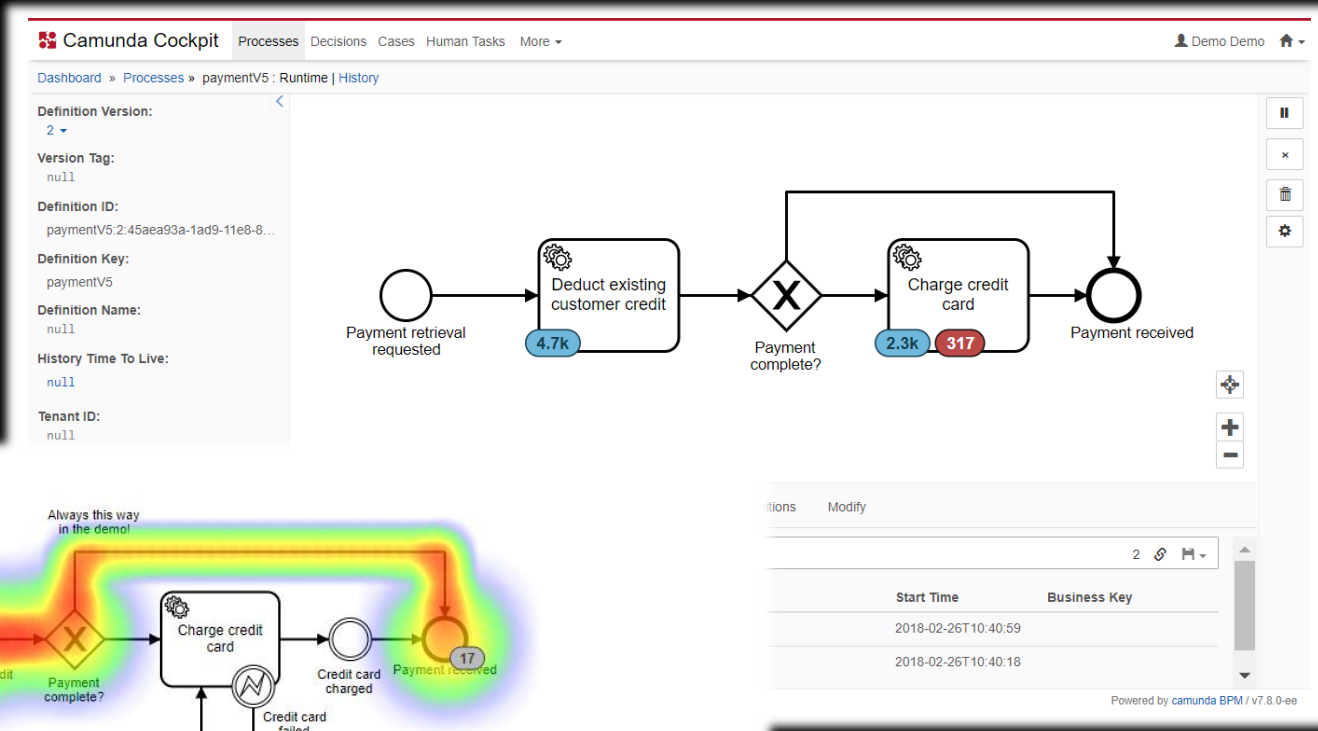
Executable
and mature

Easy to
understand*

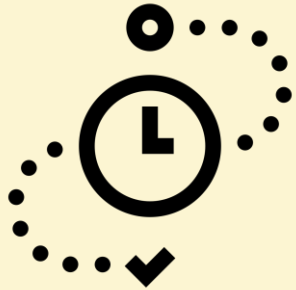
*(for Biz, Dev, and ops)

Proper operations

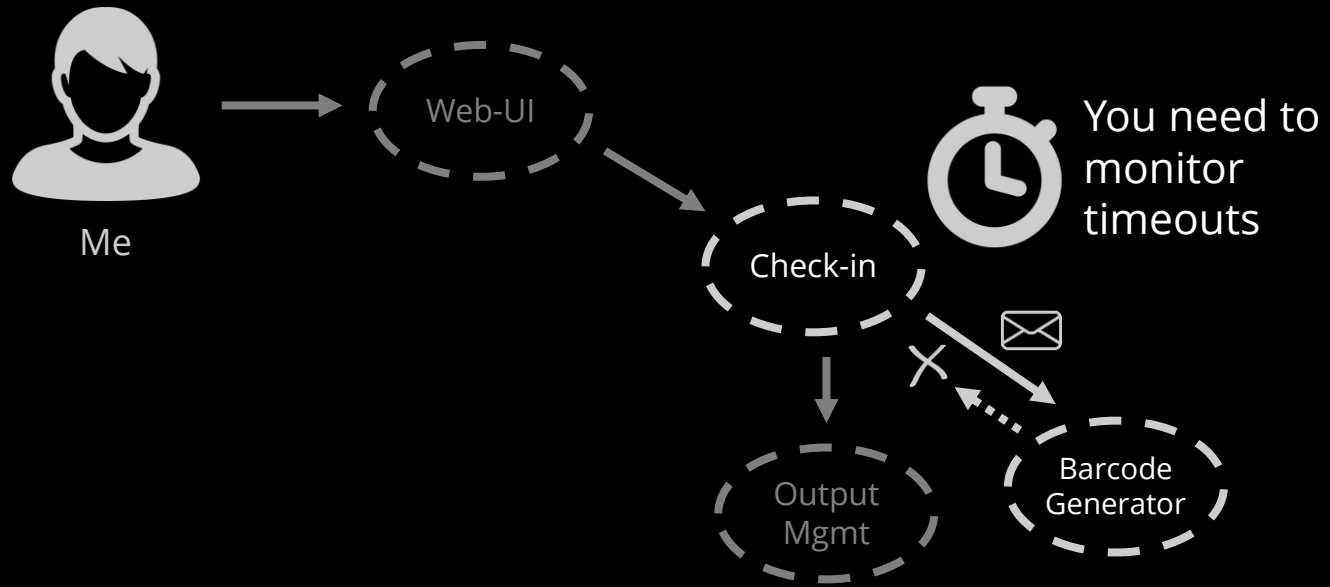
Visibility + Context



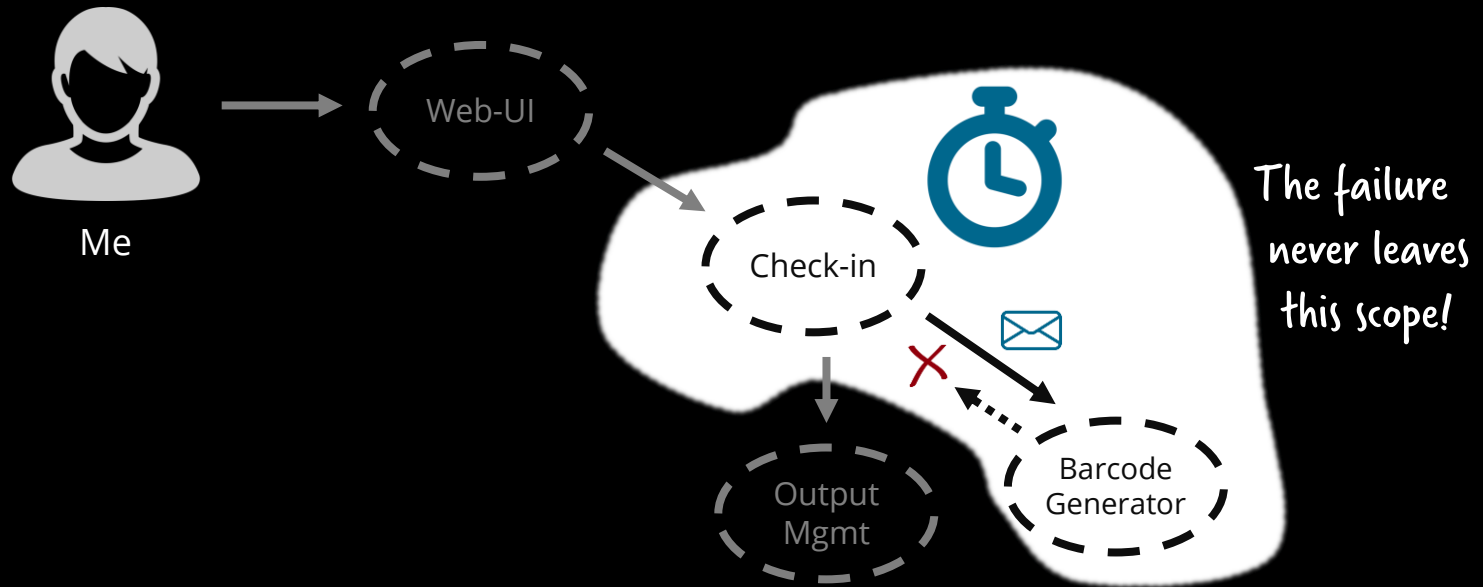
Challenges of asynchronicity



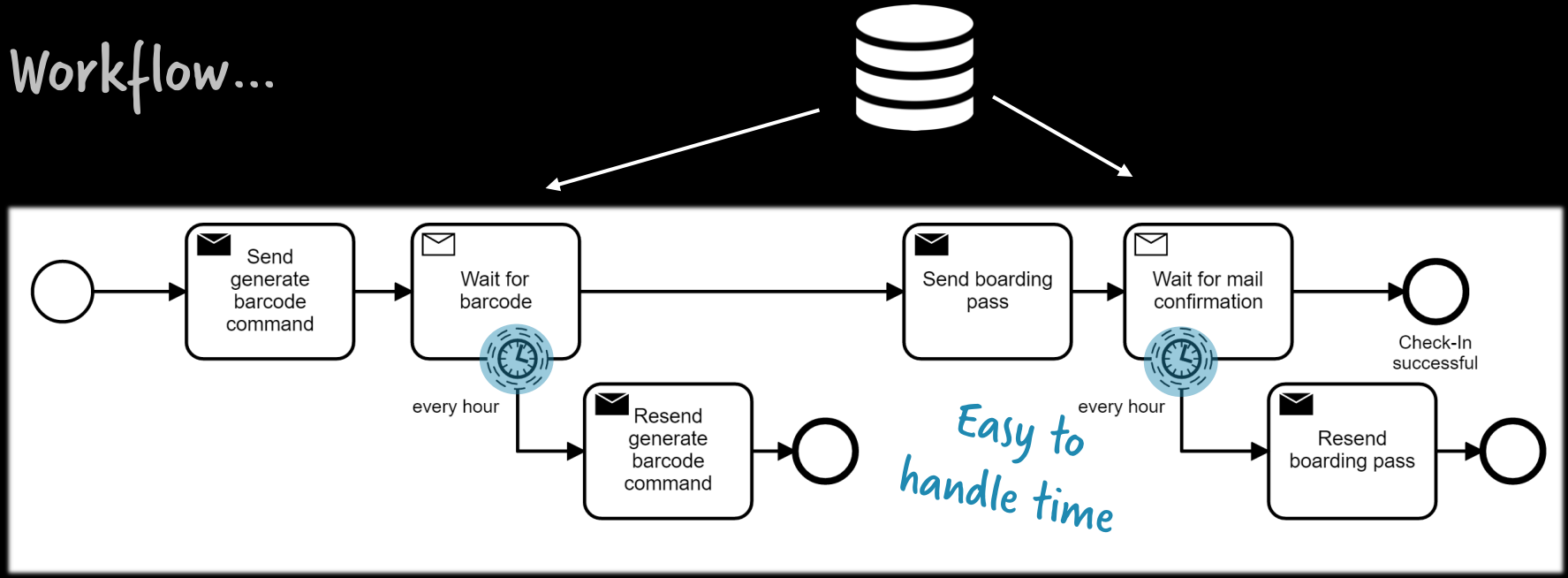
Asynchronous communication



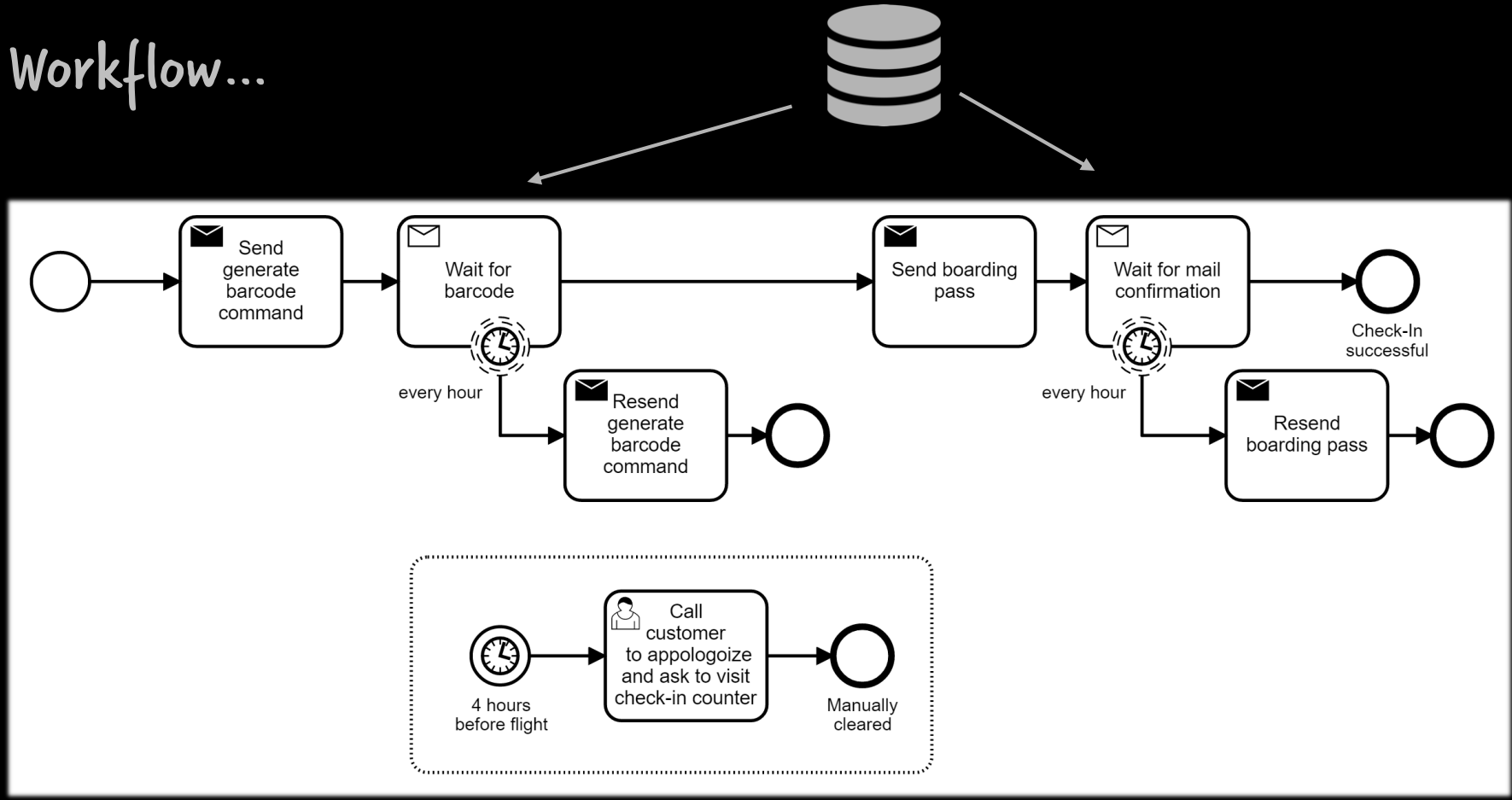
Remember...


Me
Web-UI
Check-in
Output Mgmt
Barcode Generator
The failure never leaves this scope!

Workflow...


Send generate barcode command
Wait for barcode
every hour
Resend generate barcode command
Send boarding pass
Wait for mail confirmation
every hour
Resend boarding pass
Check-In successful
Easy to handle time

Workflow...

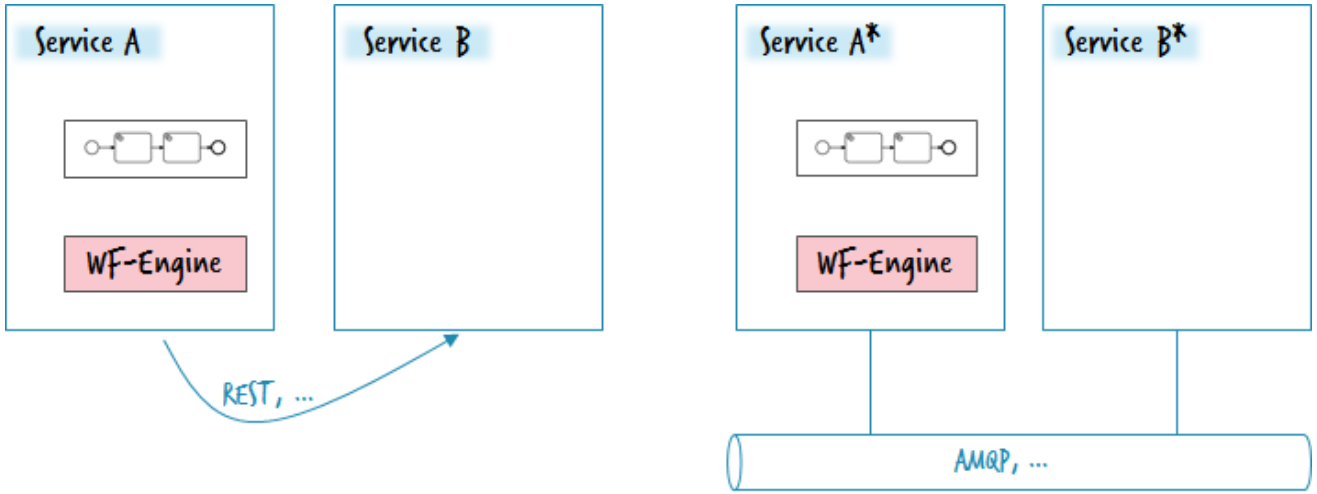

Send generate barcode command
Wait for barcode
every hour
Resend generate barcode command
Send boarding pass
Wait for mail confirmation
every hour
Resend boarding pass
Check-In successful
4 hours before flight
Call customer to appologoize and ask to visit check-in counter
Manually cleared

Who uses a message bus?

Who has no problems operating a message bus?

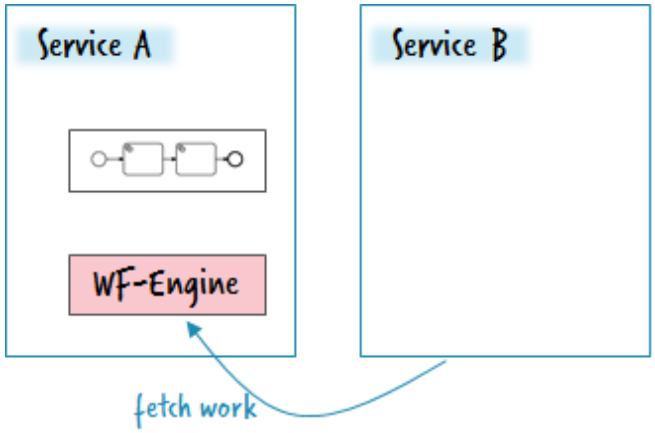
Dead messages | No context | Inaccesible payload | Hard to redeliver | Home-grown message hospitals | ...

Manigfold architecture options



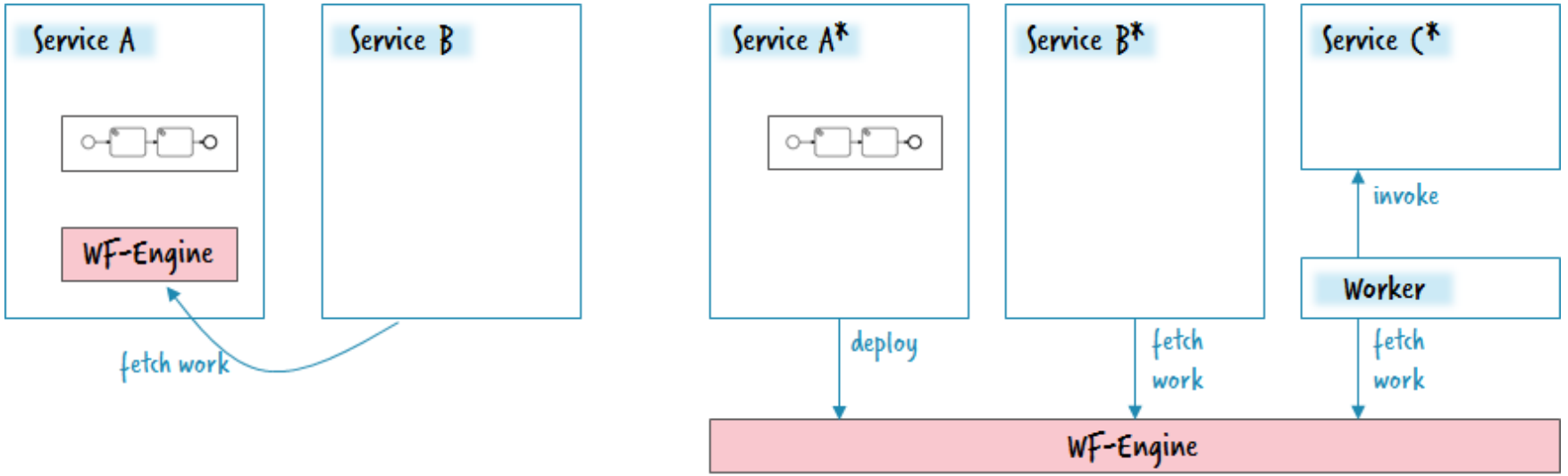
https://blog.bernd-ruecker.com/architecture-options-to-run-a-workflow-engine-6c2419902d91

Manigfold architecture options



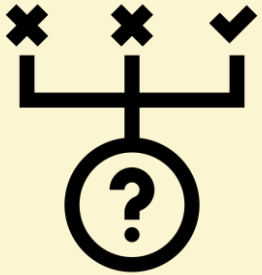
https://blog.bernd-ruecker.com/architecture-options-to-run-a-workflow-engine-6c2419902d91

Manigfold architecture options

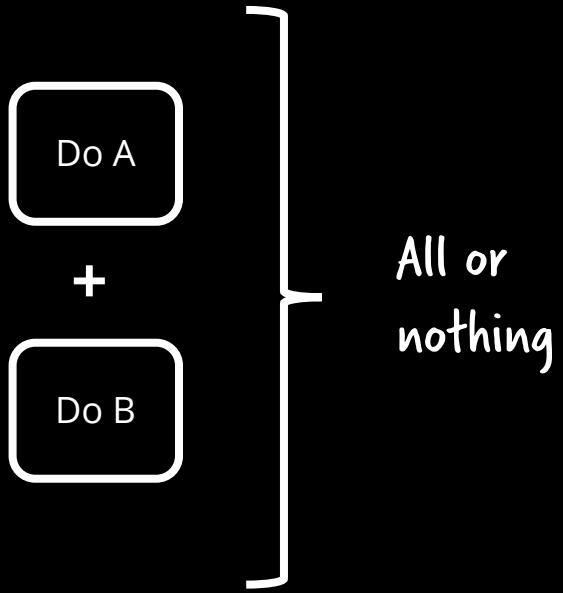


https://blog.bernd-ruecker.com/architecture-options-to-run-a-workflow-engine-6c2419902d91

Distributed Transactions



Transactions...



Distributed systems

2007

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

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

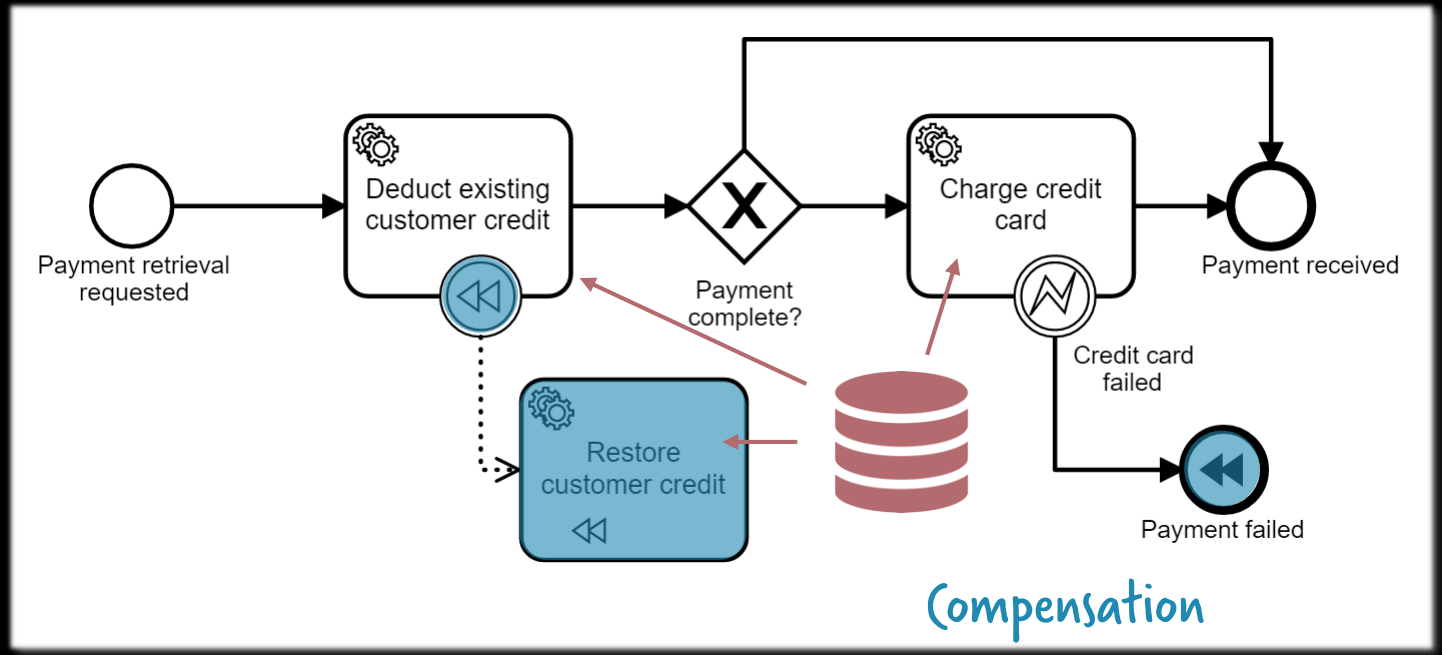
Pat Helland

Distributed Systems Guru
Worked at Amazon,
Microsoft & Salesforce

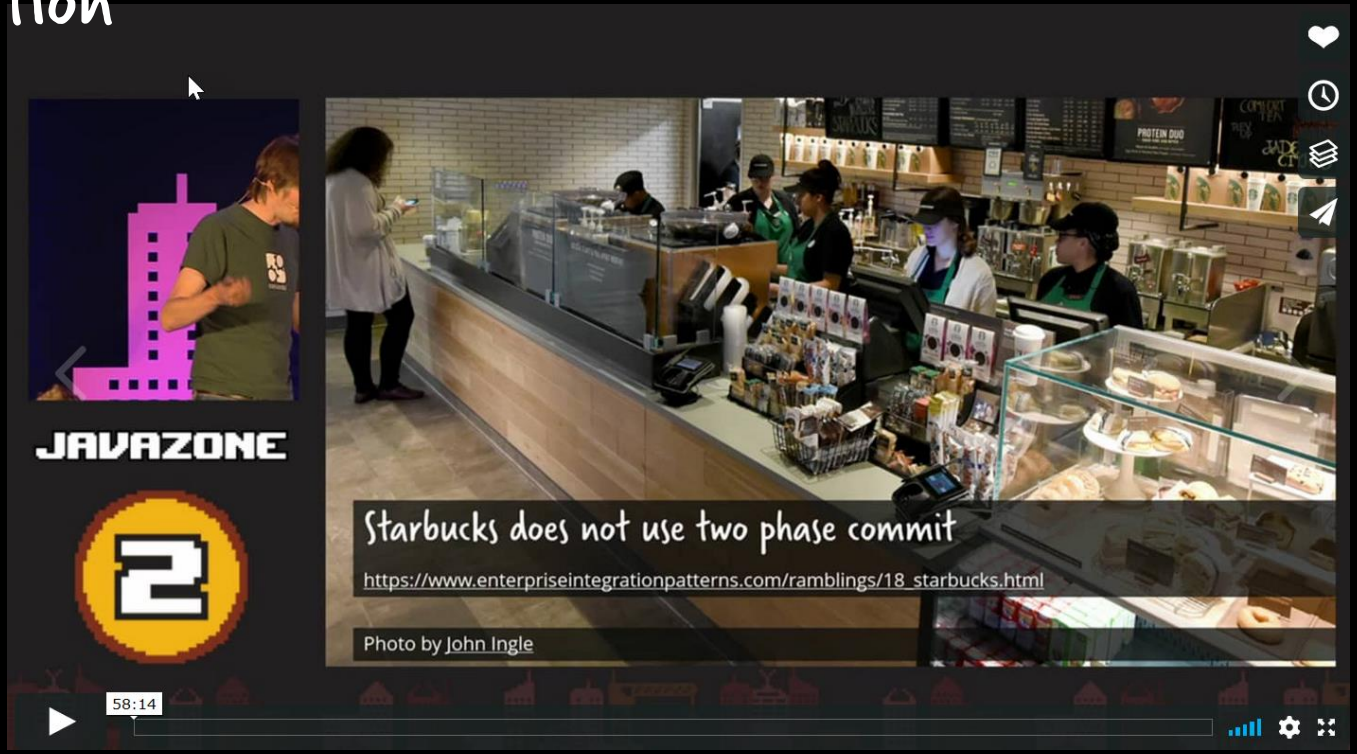
"Grown-Ups Don't Use Distributed Transactions

Distributed transactions using compensation *

*aka Saga pattern



„Lost in transaction“



https://vimeo.com/289508460

https://berndruecker.io/lost-in-transaction/

Client

has to implement

Timeout, Retry, Compensation

Service Provider

has to offer

Compensation

has to implement

Idempotency

Don't forget about state

Client

has to implement

Timeout, Retry,

tion

Service Provider

has to offer

Compensation

has to

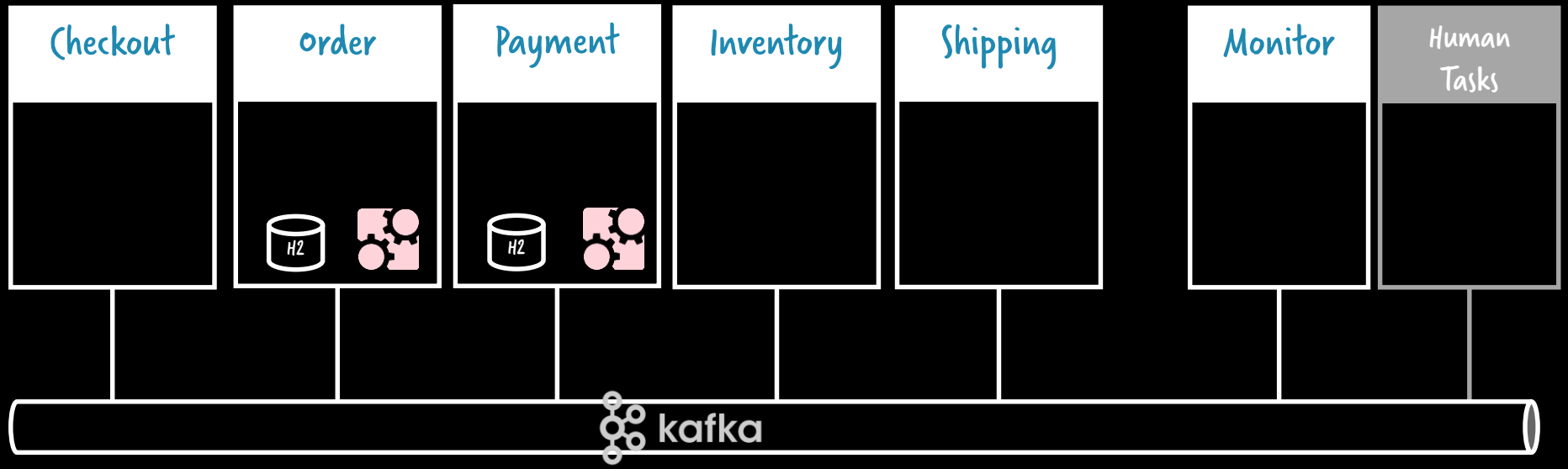
Iden

Communication is complex

Challenges of asynchronicity

Distributed Transactions

Event-driven example also available



https://github.com/berndruecker/flowing-retail/

Be aware of complexity of distributed systems
Know strategies and tools to handle it

e.g. Circuit breaker (Hystrix > Resilience4J)
Workflow engine for stateful retry, waiting, timeout and compensation (Camunda)

Thank you!

Please

Remember to rate this session

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

InfoQ ueue

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

THENEWSTACK

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