camunda
the business process company

Lost in transaction

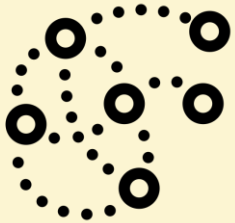
mail@bernd-ruecker.com | @berndruecker
http://bernd-ruecker.com/
Co-Founder & Developer Advocate

REST, SOAP,
Cloud, Saas,
Microservices, SCS,
FaaS, Serverless,
...

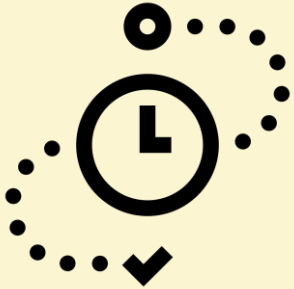
Distributed systems

Distributed systems

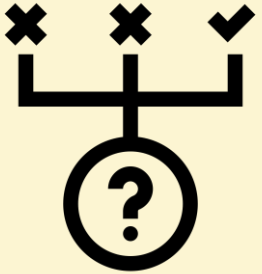
Communication is complex



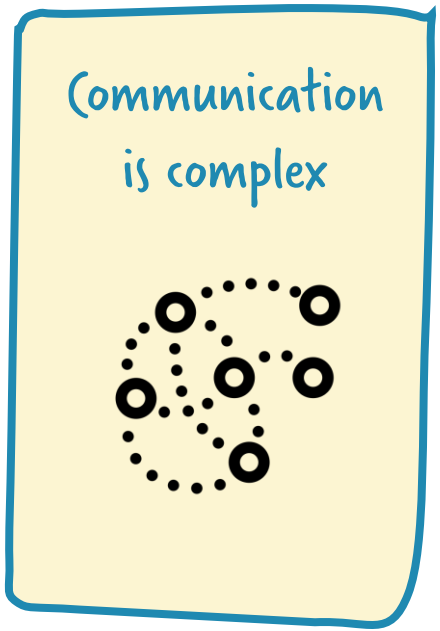
Challenges of asynchronicity

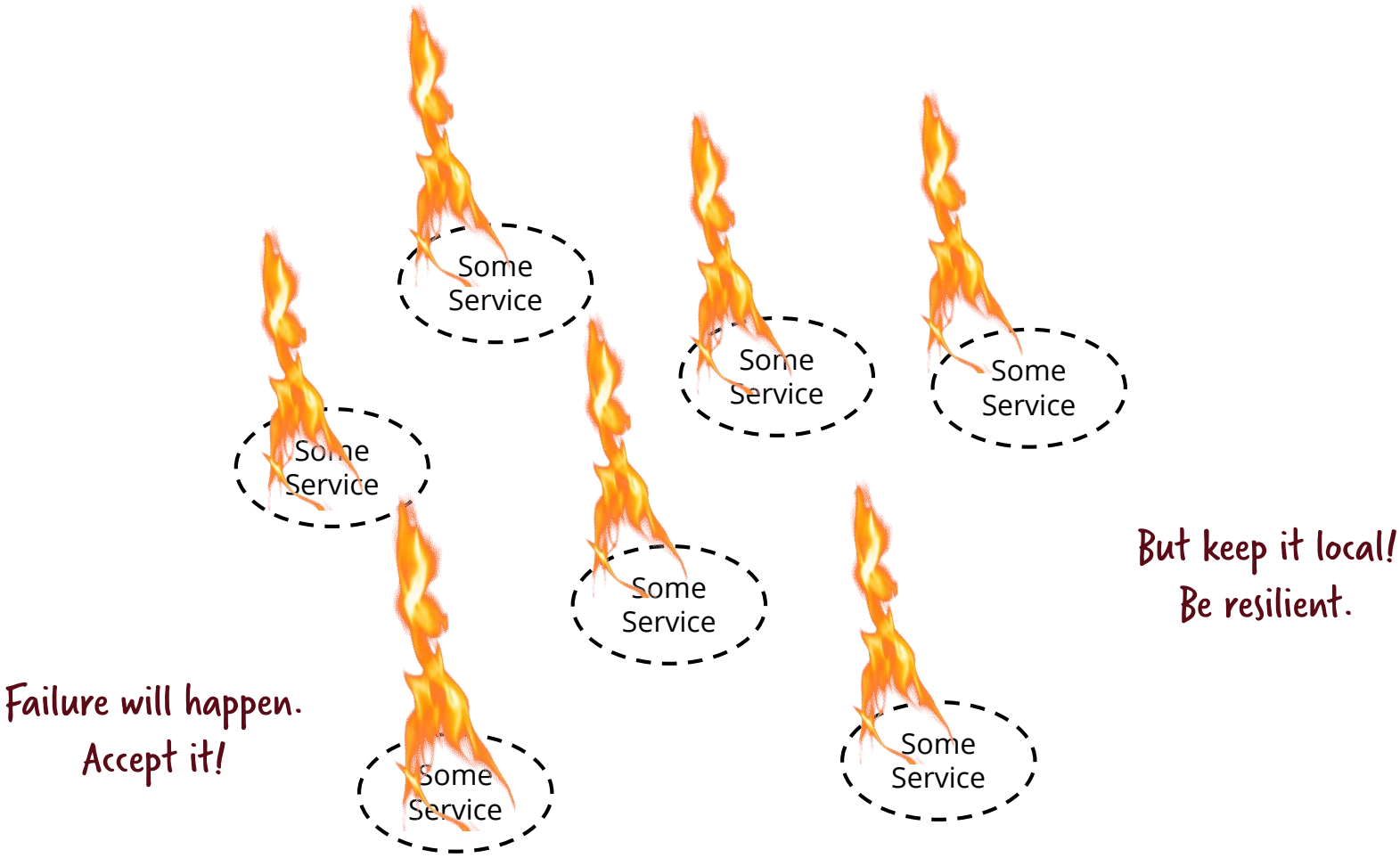


Distributed Transactions

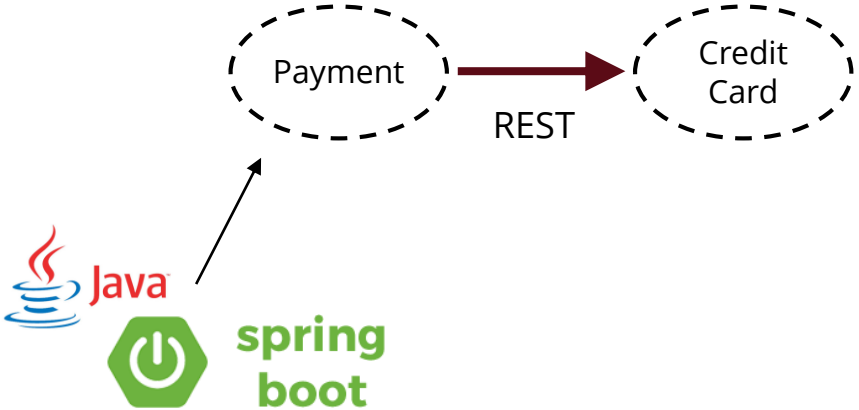


Challenge #1:




Some Service
Some Service
Some Service
Some Service
Some Service
Some Service
Some Service
Failure will happen.
Accept it!
But keep it local!
Be resilient.

Let‘s start with a simple example

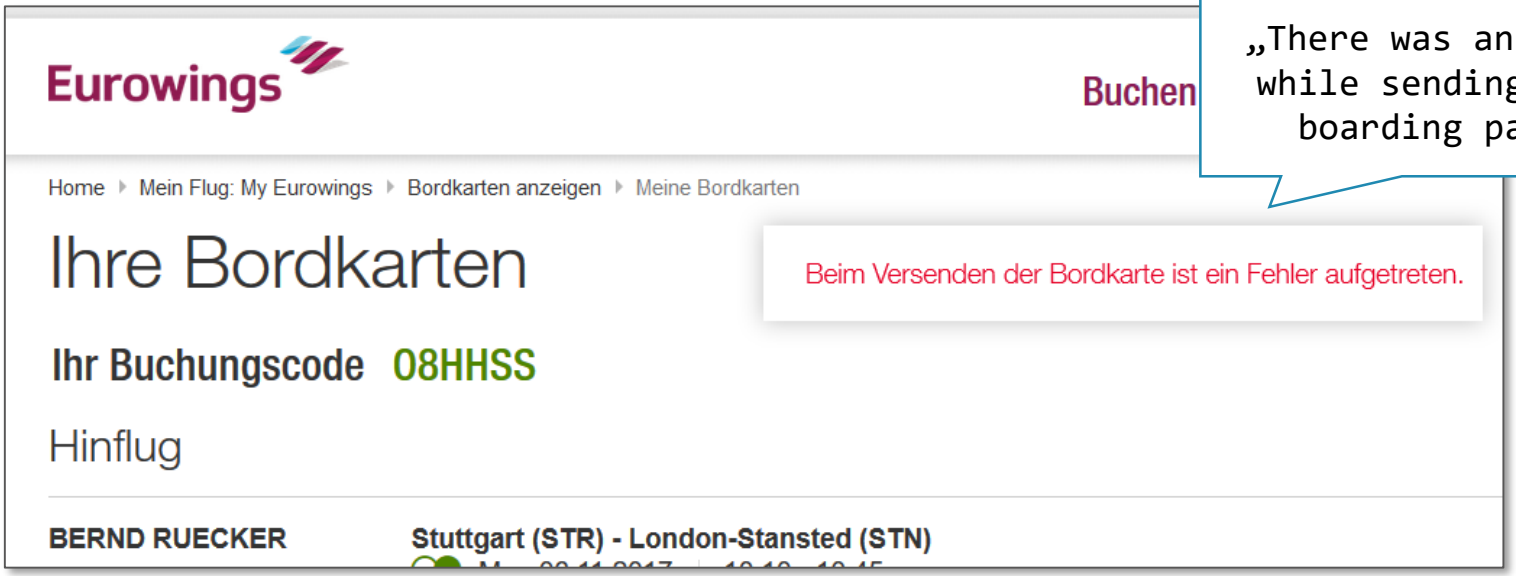


Circuit Breaker

Getting from https://github.com/flowing/flowing-retail/blob/master/payment-rest/src/main/java/io/flowing/retail/payment/port/resthacks/PaymentRestHacksControllerV1.java to V2

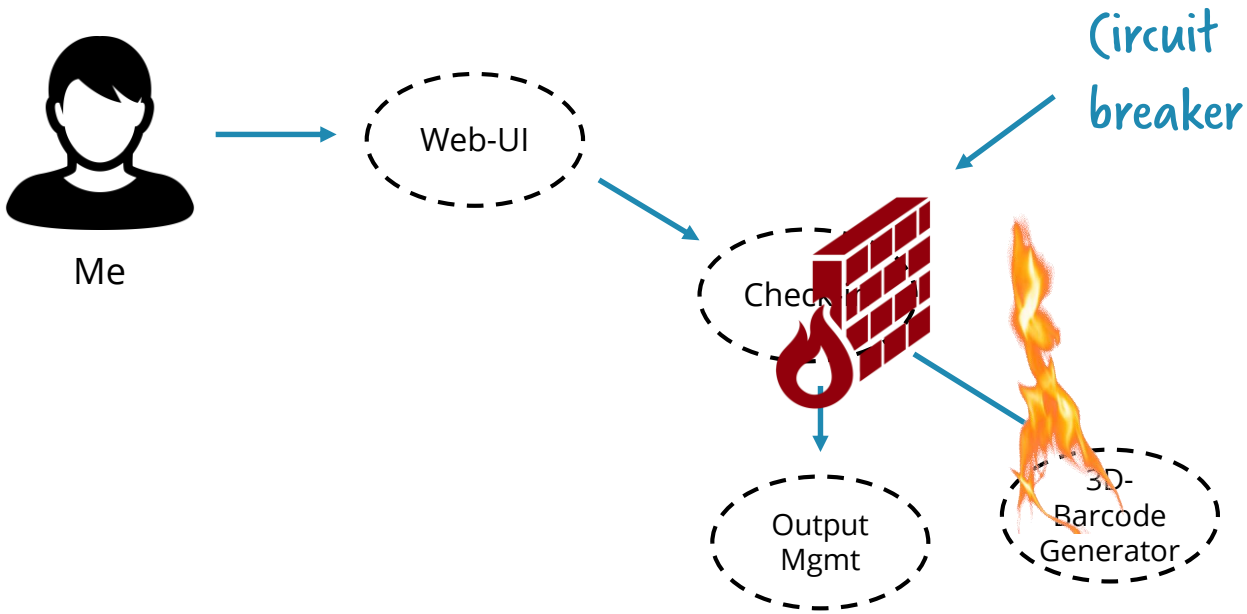
Fail fast
is important

Fail fast
is important
but not enough!

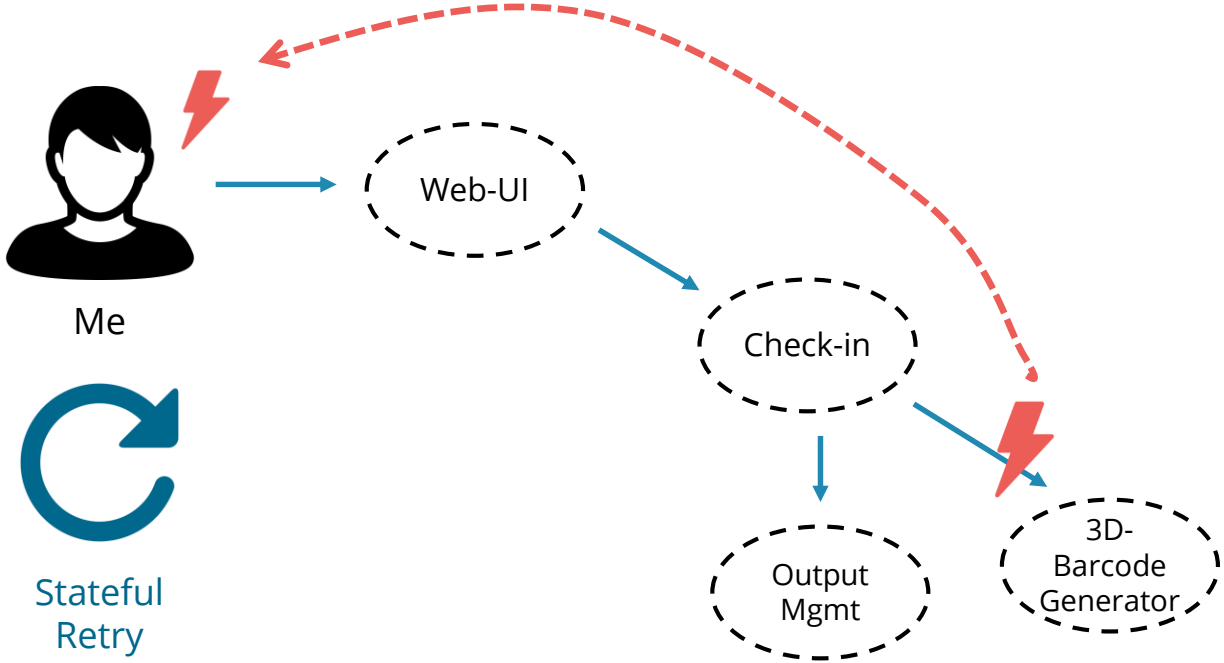


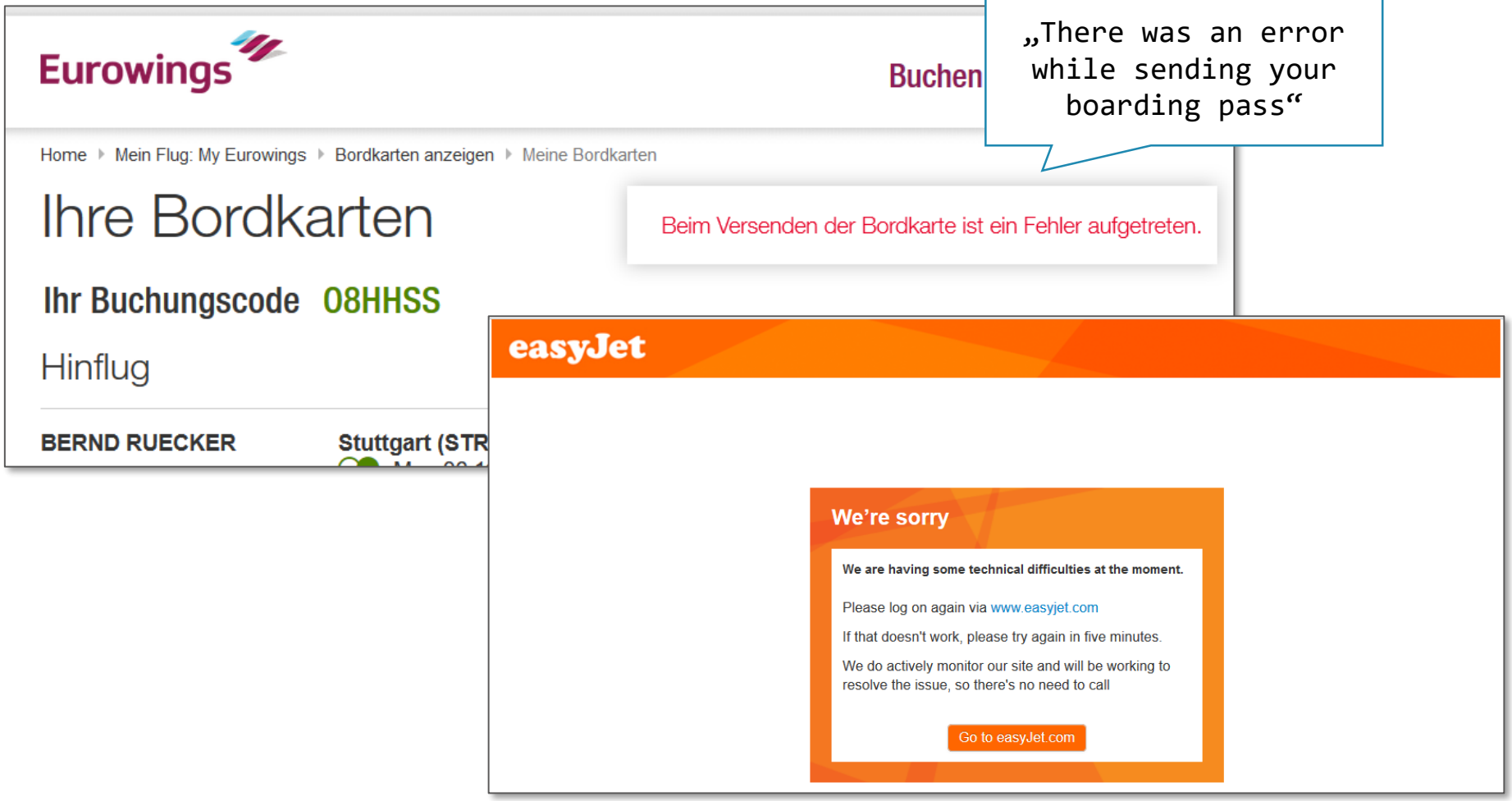
„There was an error while sending your boarding pass“

Current situation



Current situation - the bad part




Eurowings
Buchen
Home ▸ Mein Flug: My Eurowings ▸ Bordkarten anzeigen ▸ Meine Bordkarten
Ihre Bordkarten
Ihr Buchungscode 08HHSS
Hinflug
BERND RUECKER
Stuttgart (STR
„There was an error while sending your boarding pass“
Beim Versenden der Bordkarte ist ein Fehler aufgetreten.
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

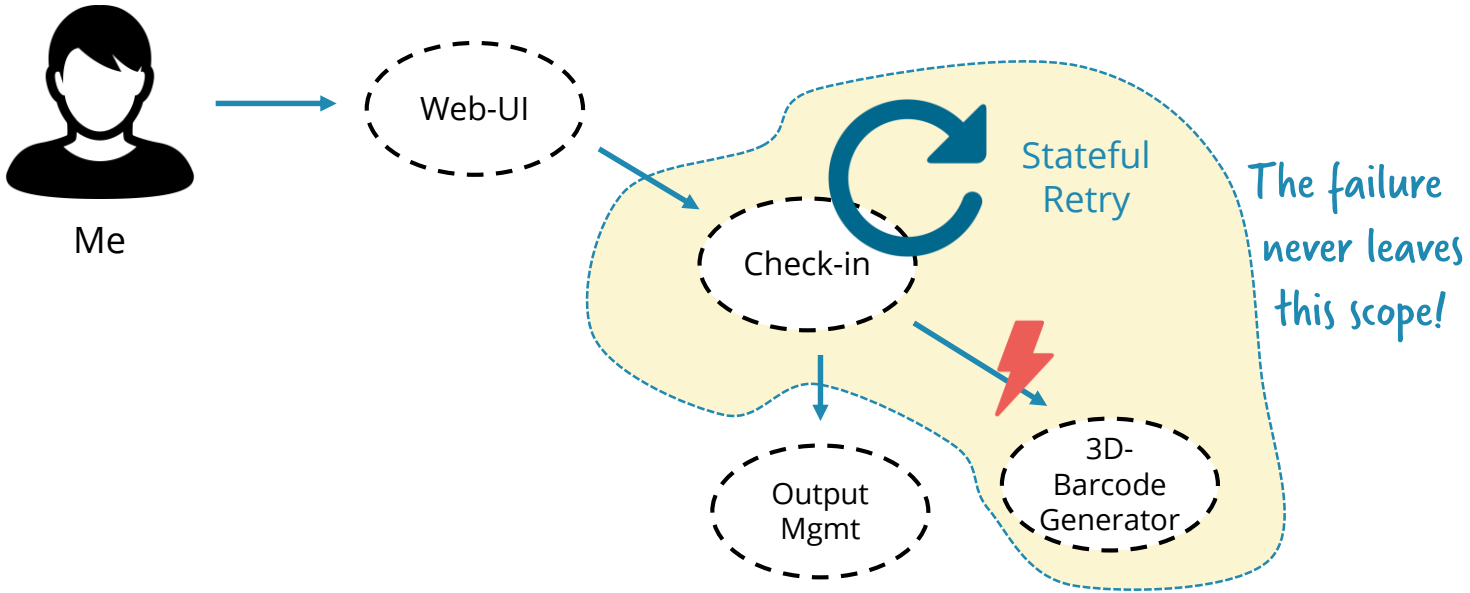
...I just made this up...

We're sorry

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

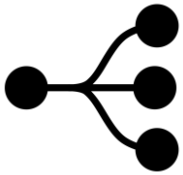
But we do actively retry ourselves, so lean back, relax and we will send it on time.

Possible situation – much better!



Handling State

Persist thing
(Entity, Actor, ...)



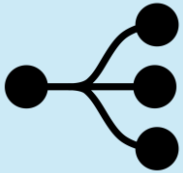
State machine or
workflow engine

Typical concerns

DIY = effort, accidental complexity

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

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



State machines or workflow engines

UBER CADENCE

zeebe

AWS Step Functions

camunda

jBPM

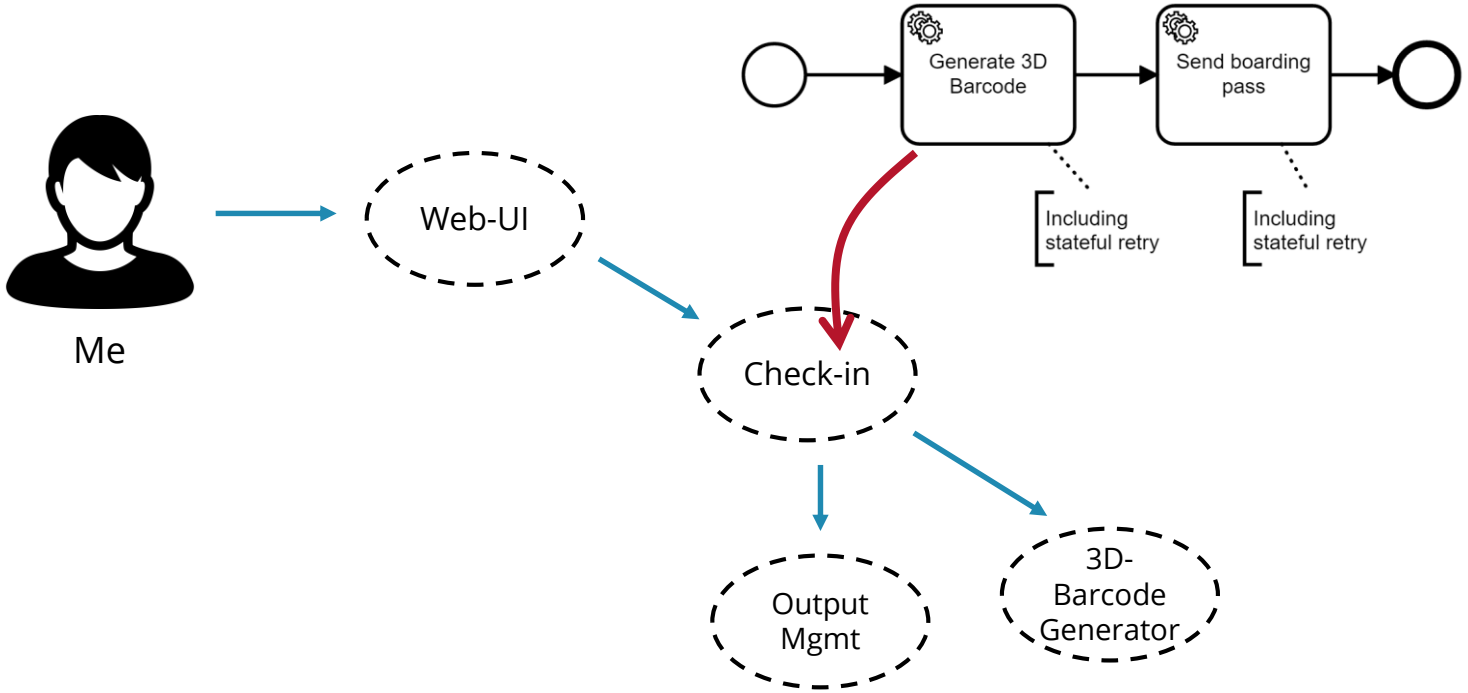
conductor

NETFLIX OSS

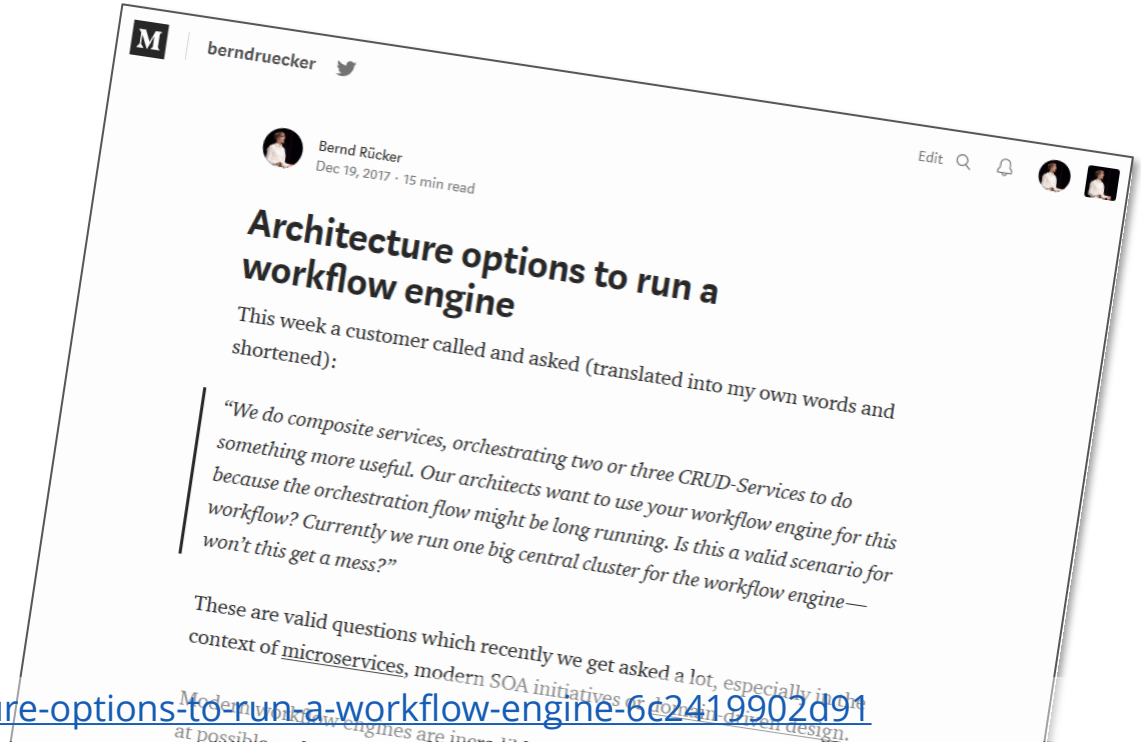
Activiti™

Getting to https://github.com/flowing/flowing-retail/blob/master/payment-rest/src/main/java/io/flowing/retail/payment/port/resthacks/PaymentRestHacksControllerV3.java

Workflows live inside service boundaries



Manigfold architecture options



berndruecker

Bernd Rücker
Dec 19, 2017 · 15 min read

Edit

Architecture options to run a workflow engine

This week a customer called and asked (translated into my own words and shortened):

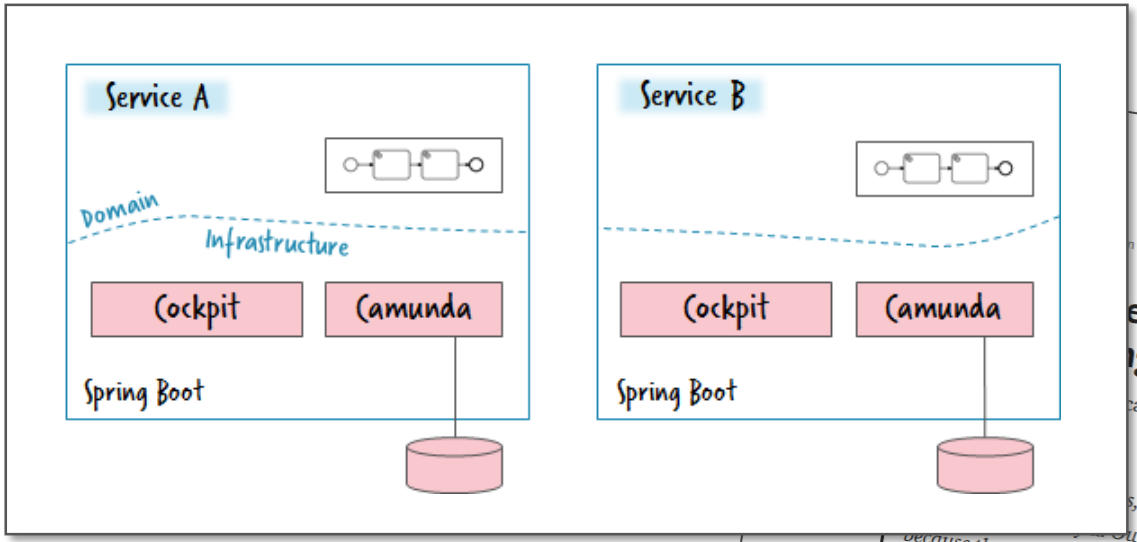
> *"We do composite services, orchestrating two or three CRUD-Services to do something more useful. Our architects want to use your workflow engine for this because the orchestration flow might be long running. Is this a valid scenario for workflow? Currently we run one big central cluster for the workflow engine—won't this get a mess?"*

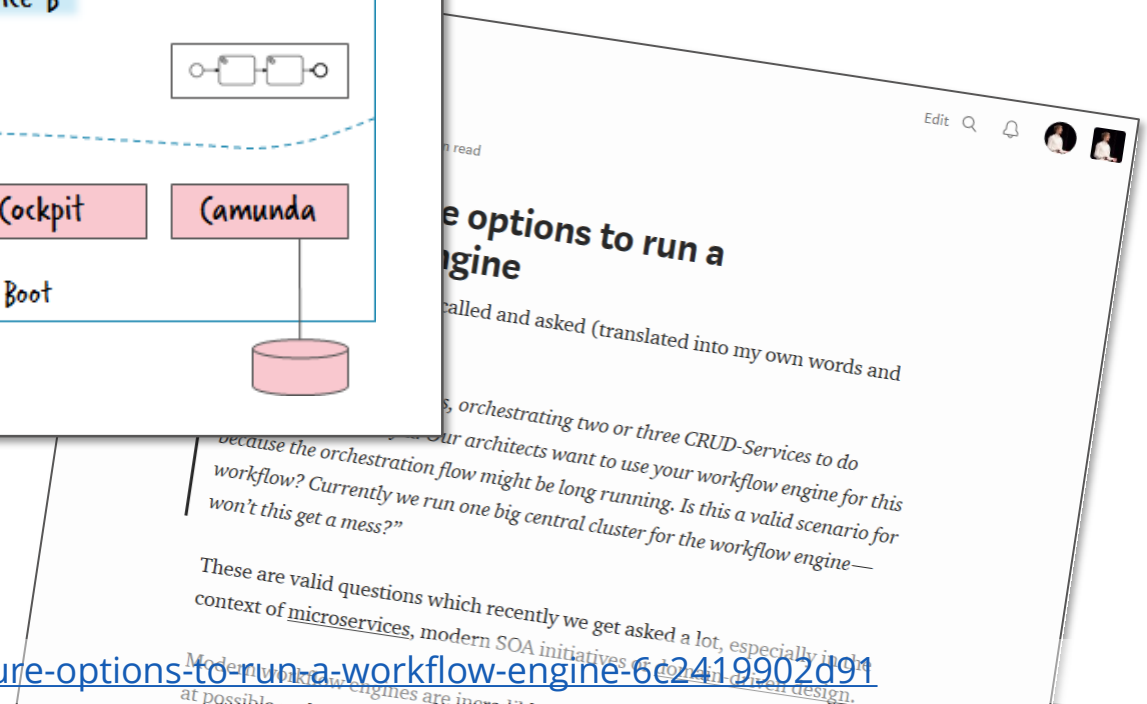
These are valid questions which recently we get asked a lot, especially in the context of microservices, modern SOA initiatives or domain-driven design.

Modern workflow engines are ingra... at possibl...

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

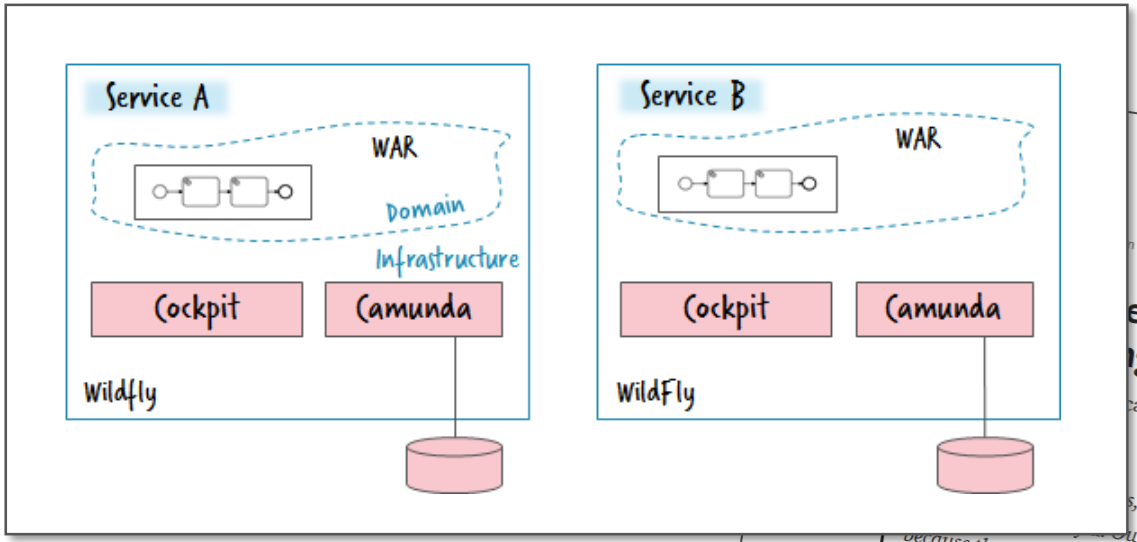
Manigfold architecture options

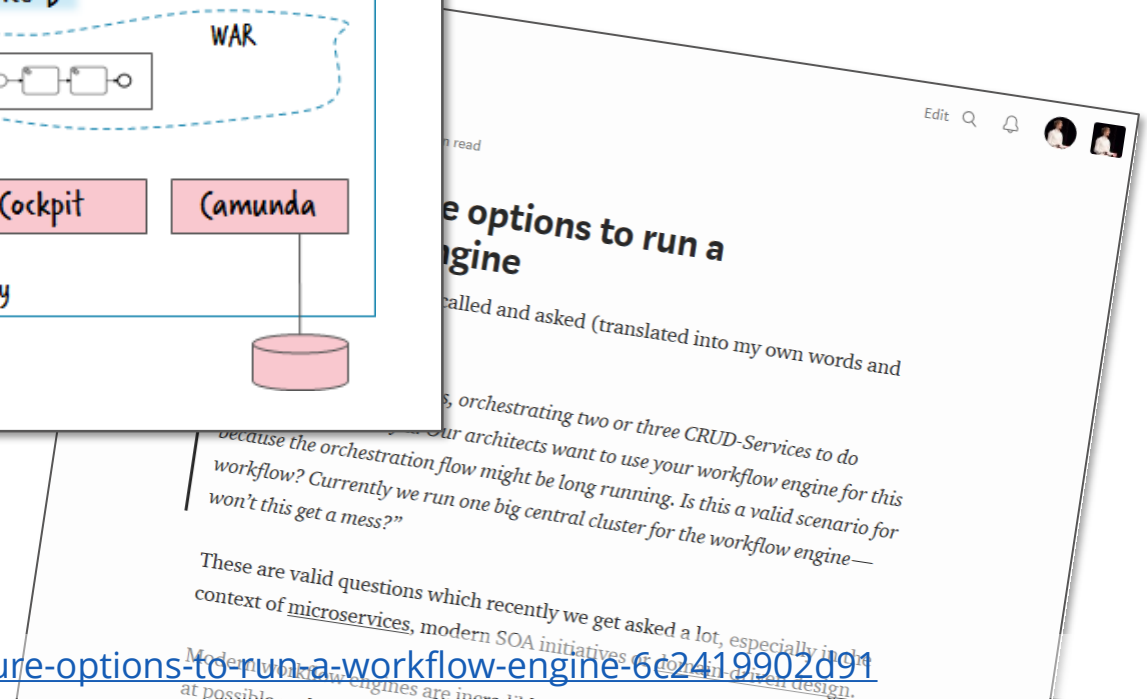




https://blog.bernd-ruecker.com/architecture-options-to-run-a-workflow-engine-6c2419902d91

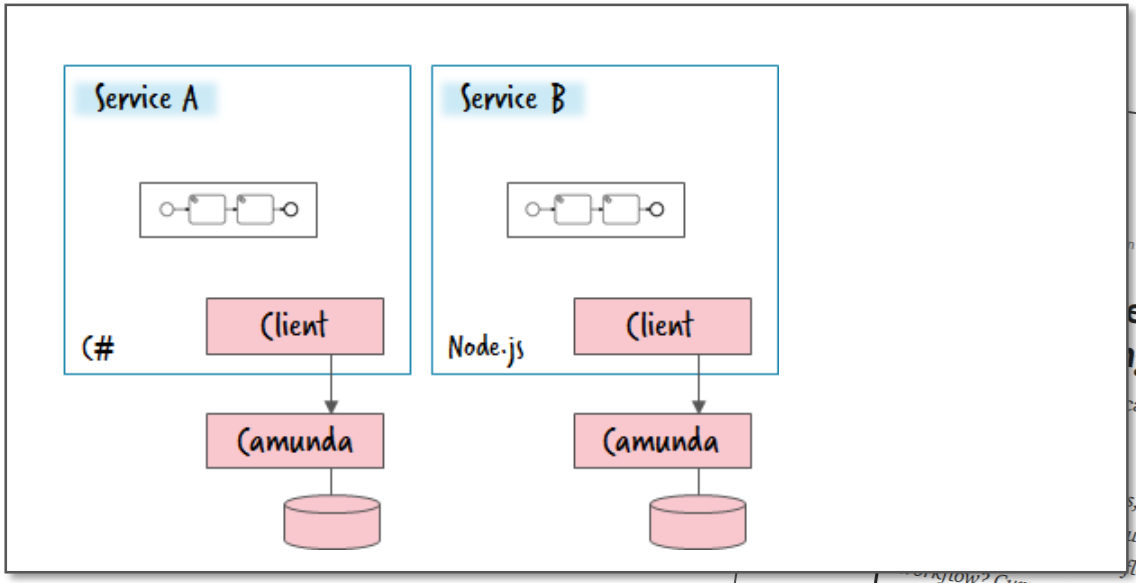
Manigfold architecture options

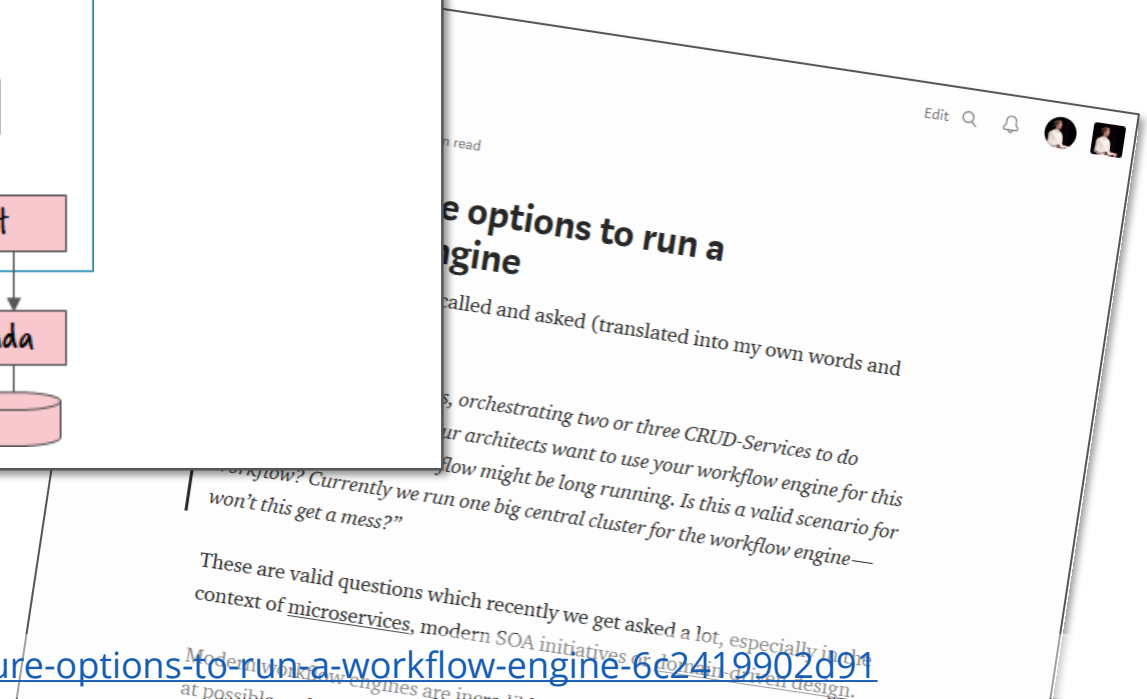




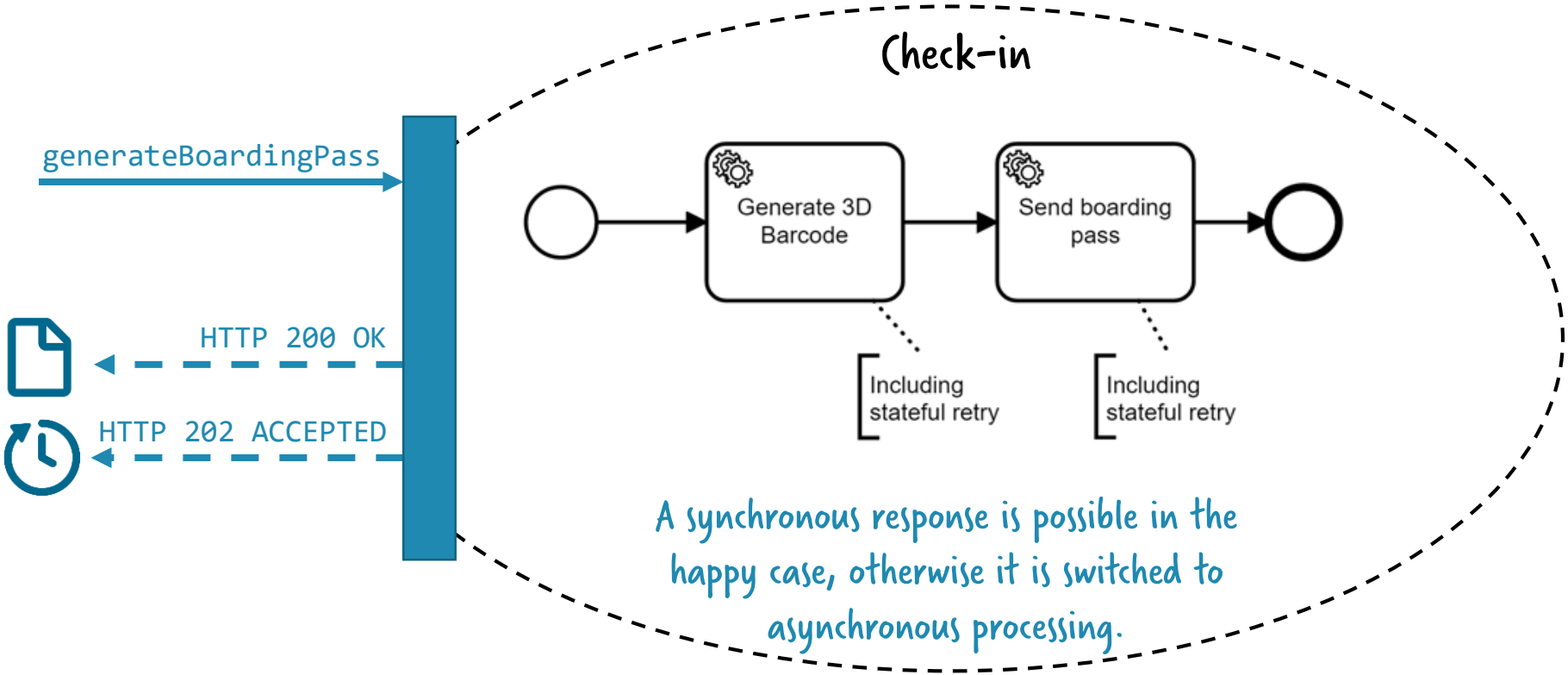
https://blog.bernd-ruecker.com/architecture-options-to-run-a-workflow-engine-6c2419902d91

Manigfold architecture options





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


Check-in
generateBoardingPass
Generate 3D Barcode
Send boarding pass
Including stateful retry
Including stateful retry
HTTP 200 OK
HTTP 202 ACCEPTED
A synchronous response is possible in the happy case, otherwise it is switched to asynchronous processing.

"The customers want synchronous responses!"

Me

„Eh – no!"

Web-UI

Check-in

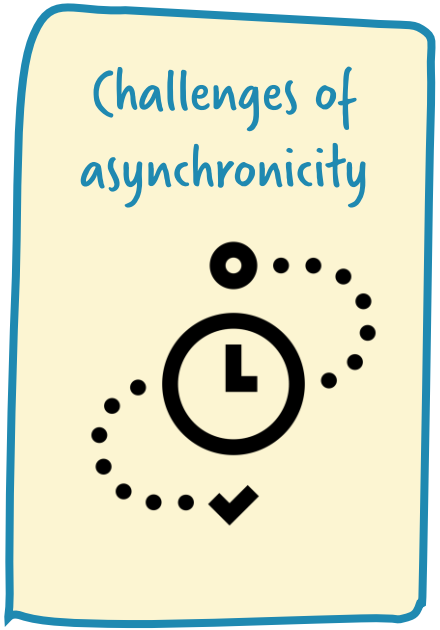
Output Mgmt

3D-Barcode Generator

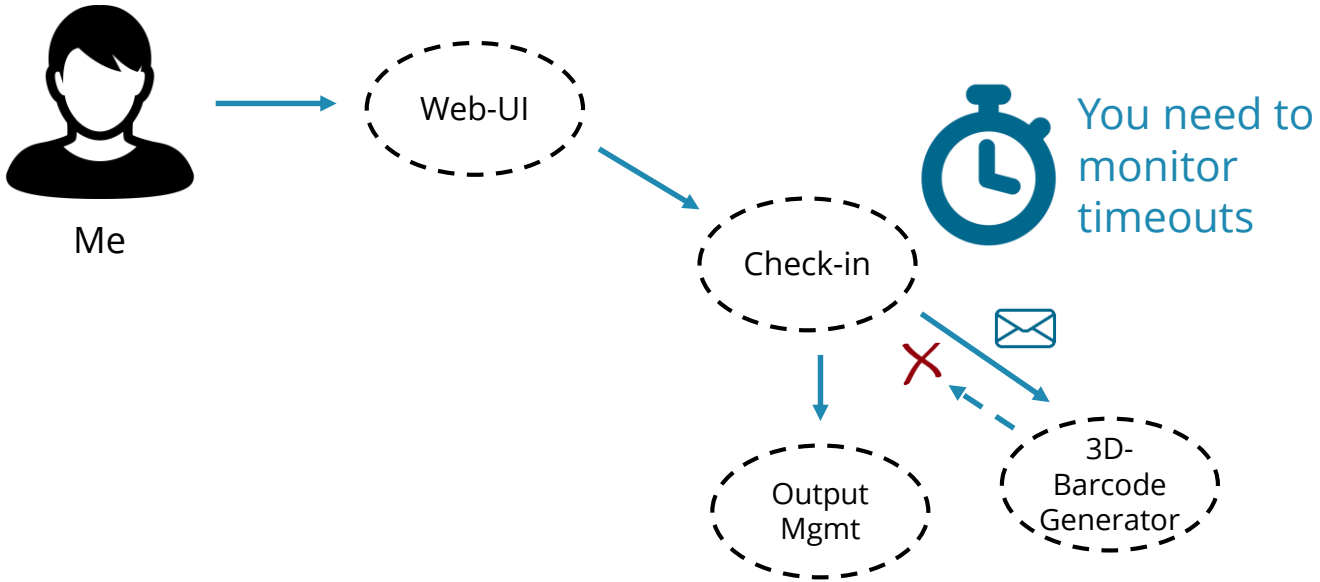
Synchronous communication is the crystal meth of distributed programming

Todd Montgomery and Martin Thompson
in "How did we end up here" at GoTo Chicago 2015

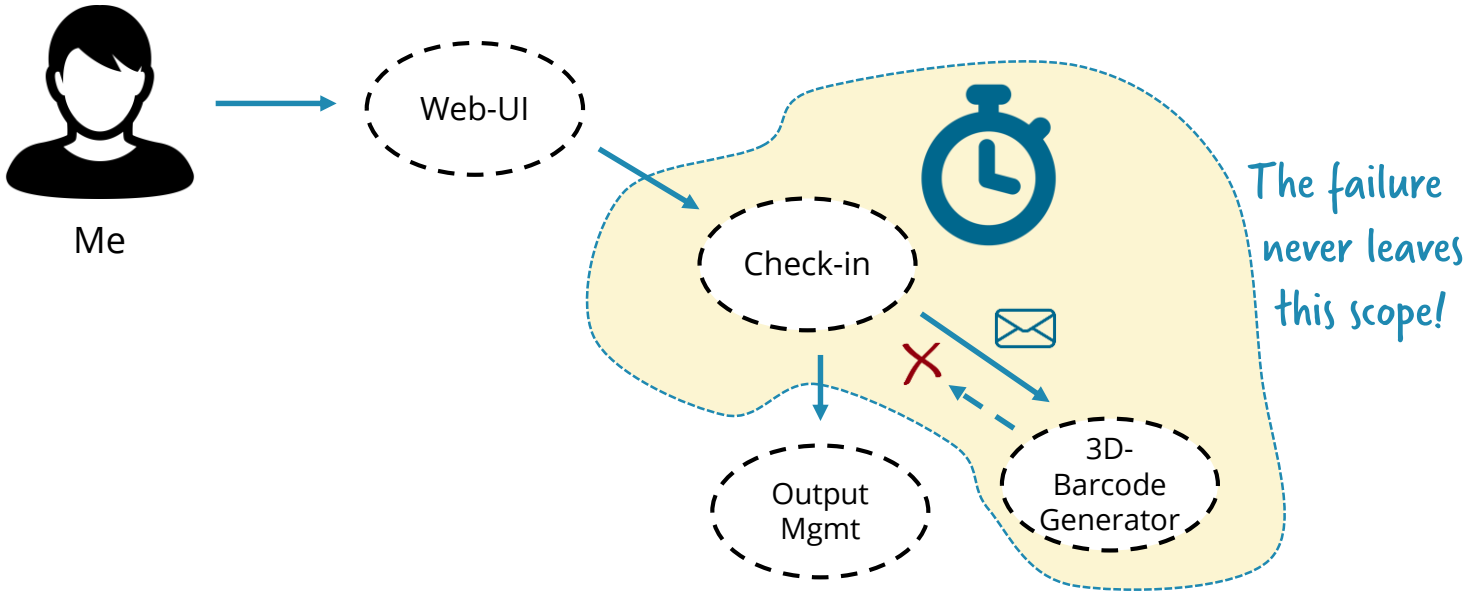
Challenge #2:



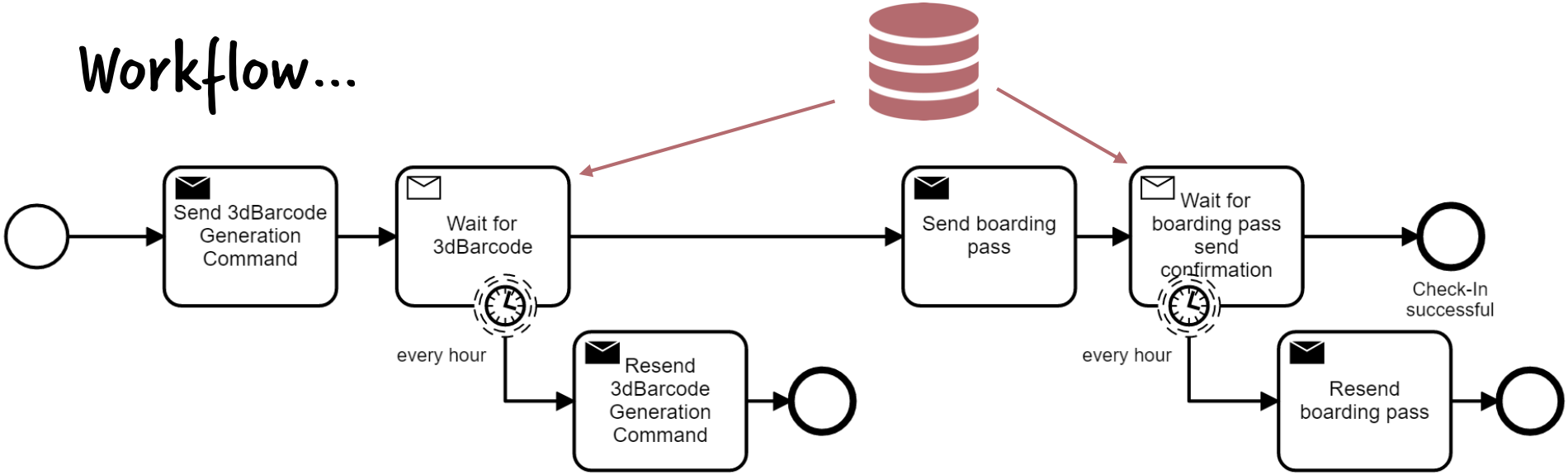
Asynchronous communication



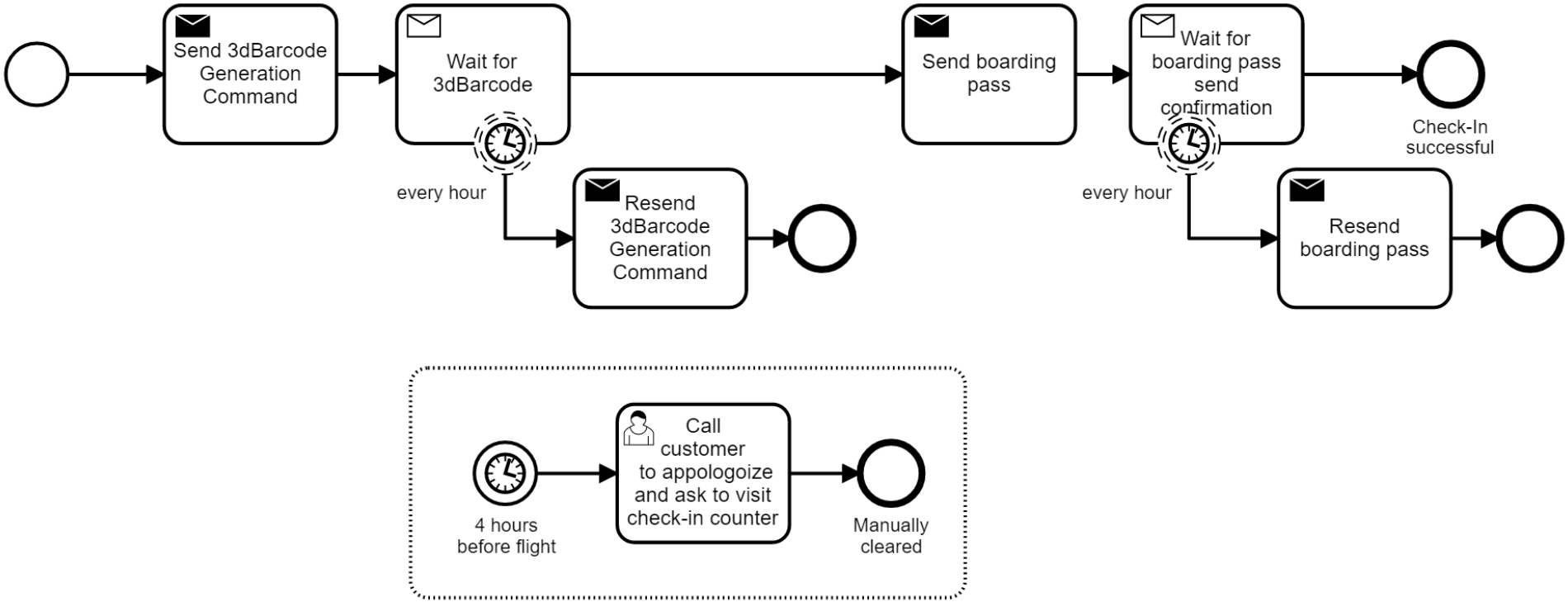
Remember...


Me
Web-UI
Check-in
Output Mgmt
3D-Barcode Generator
The failure never leaves this scope!

Workflow...


Send 3dBarcode Generation Command
Wait for 3dBarcode
every hour
Resend 3dBarcode Generation Command
Send boarding pass
Wait for boarding pass send confirmation
every hour
Resend boarding pass
Check-In successful

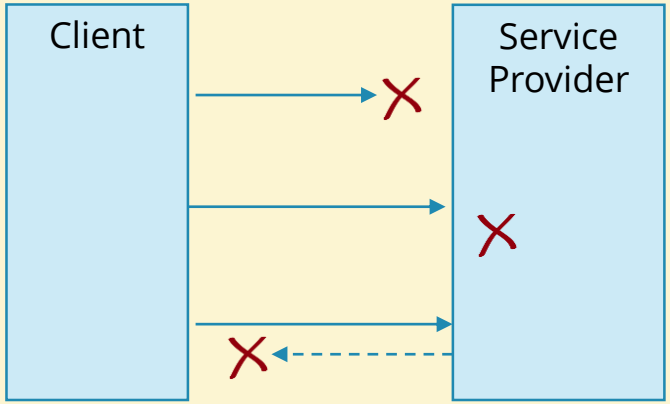
Workflow...


Send 3dBarcode Generation Command
Wait for 3dBarcode
every hour
Resend 3dBarcode Generation Command
Send boarding pass
Wait for boarding pass send 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

Duplicates
Duplicates

It is impossible to differentiate certain failure scenarios.

This is also true for synchronous request/response!



Client

has to implement

Timeout, Retry

Service Provider

has to implement

Idempotency

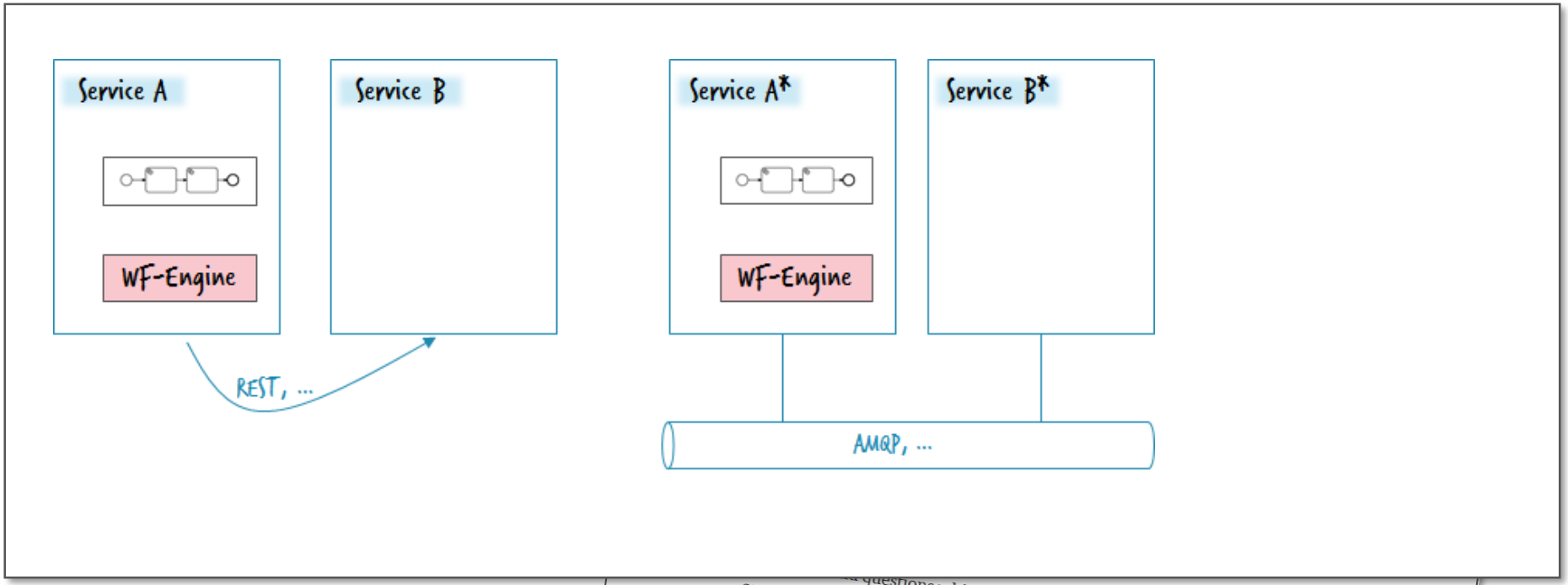
It is a business problem anyway!

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

Who has no problems operating a message bus?

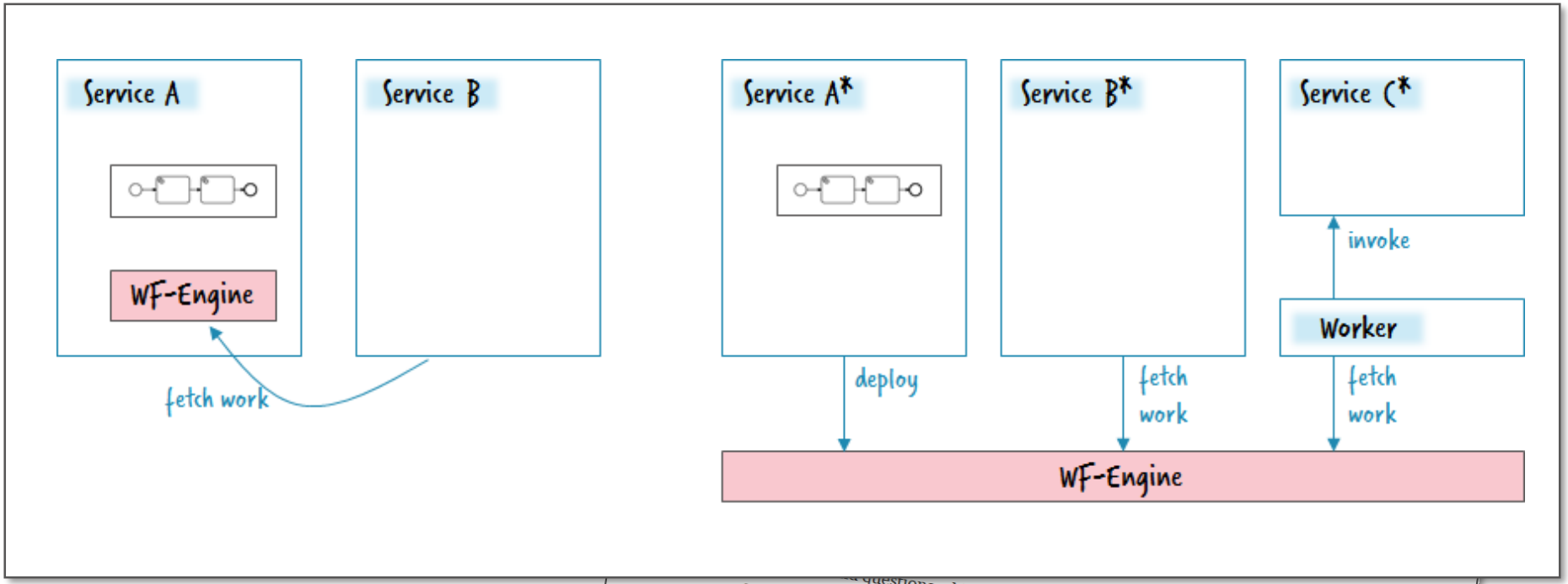
Dead messages | No context | Inaccesible payload | Hard to redeliver | Home-grown message hospitals | ...

Manigfold architecture options



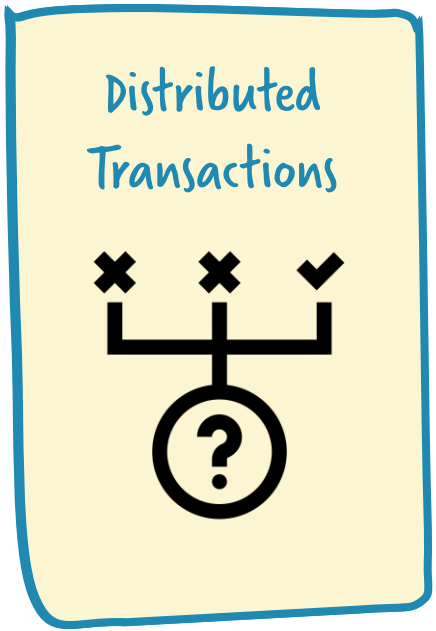
https://blog.bernd-ruecker.com/architecture-options-to-run-a-workflow-engine-6c2419902d91

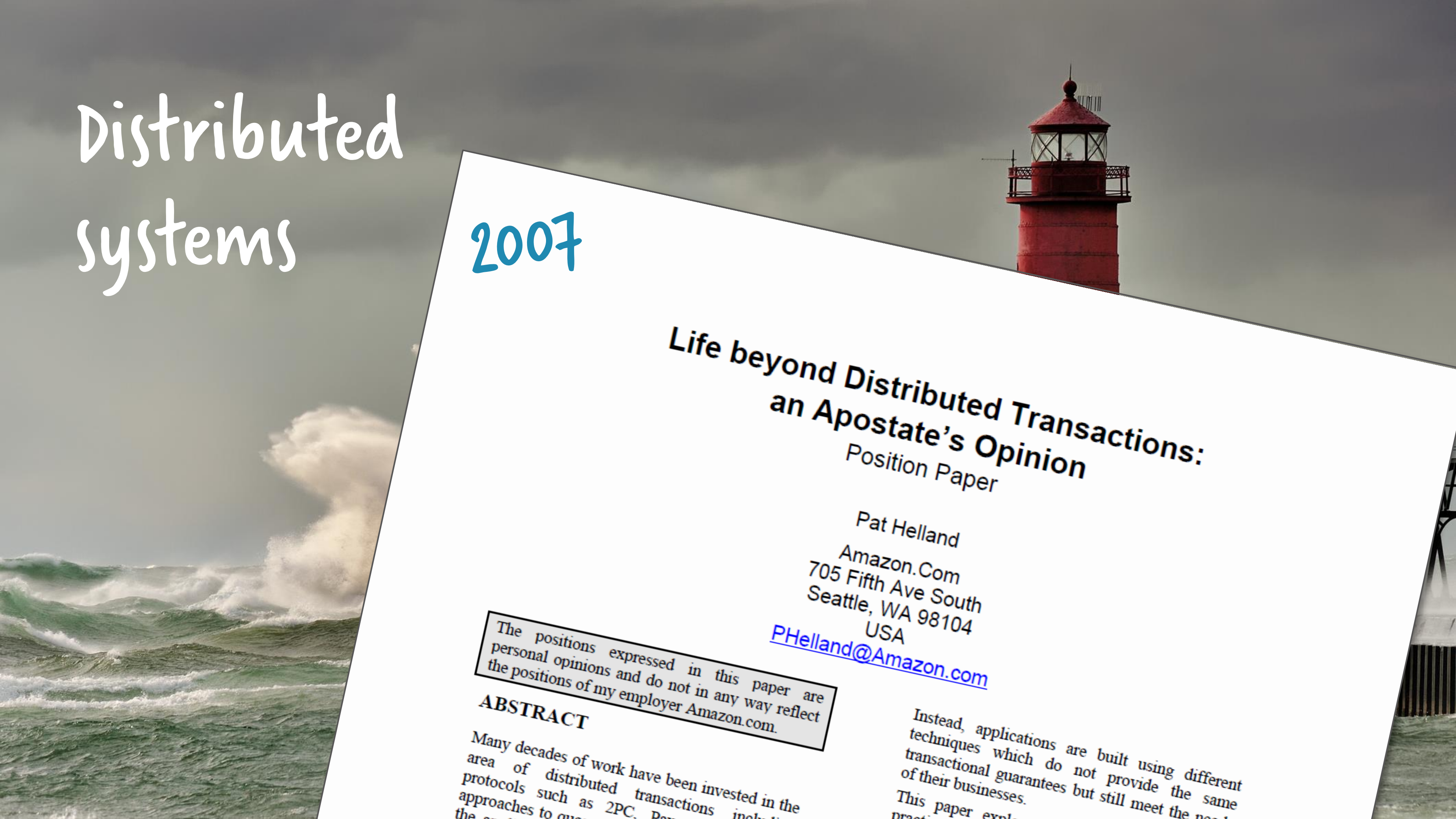
Manigfold architecture options



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

Challenge #3:





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.

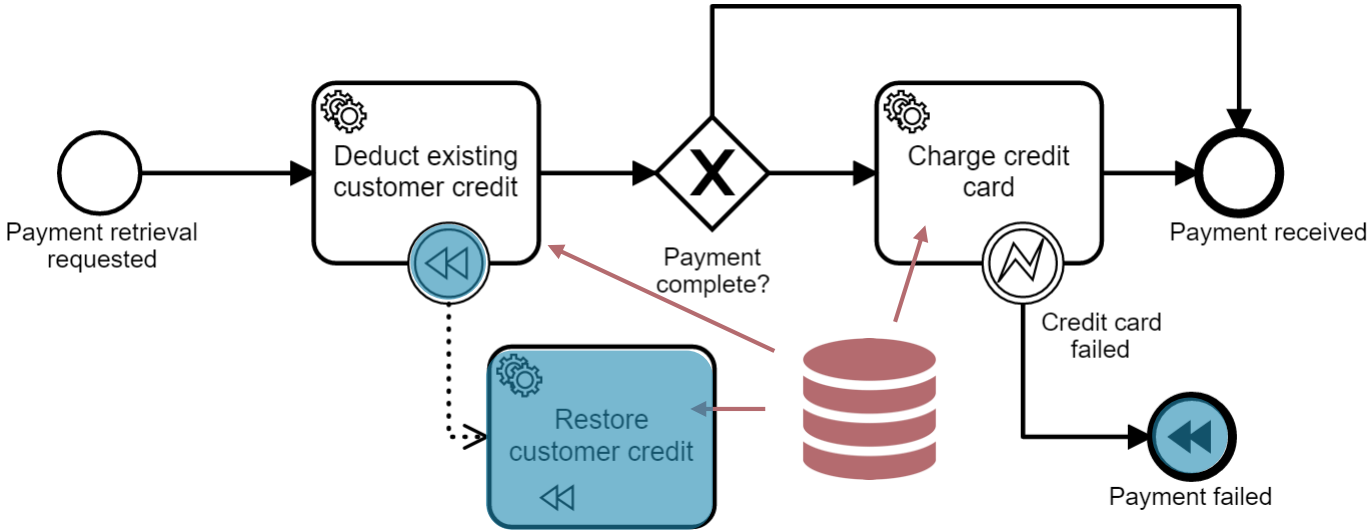
ABSTRACT

Many decades of work have been invested in the area of distributed transactions ...

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

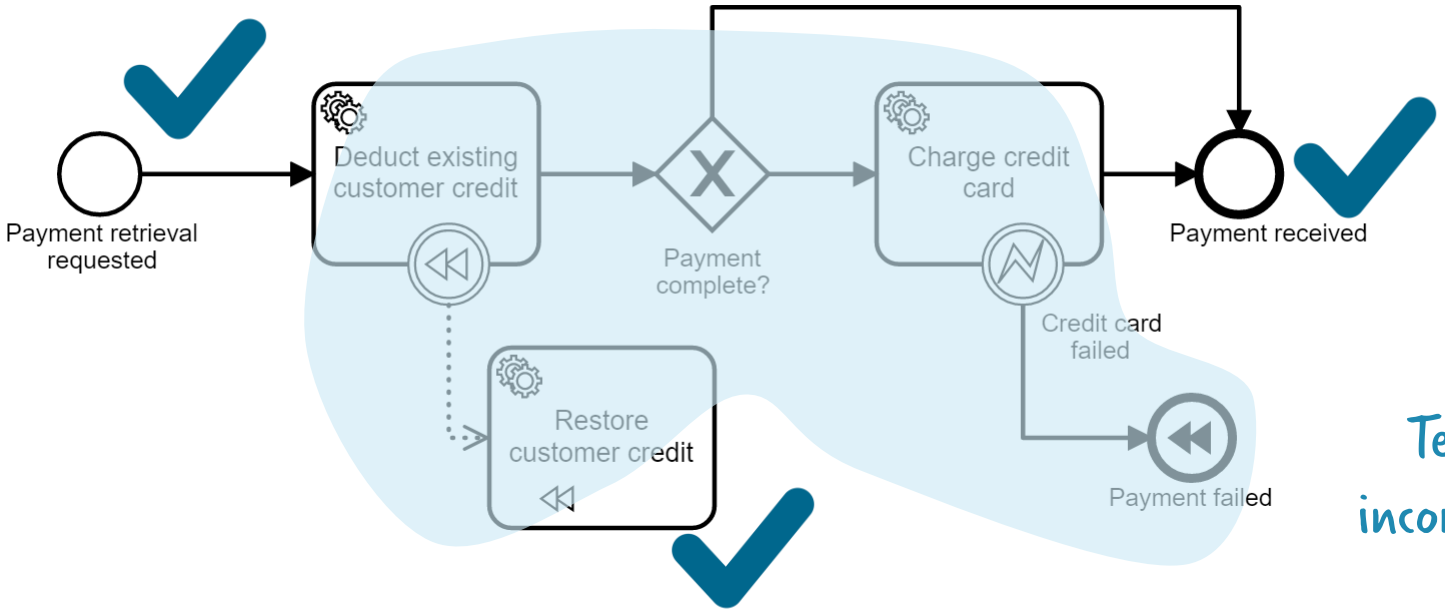
This paper ...

Business transactions (aka Saga pattern)



Compensation

Eventual consistency



But only temporarily!

Temporarily inconsistent state

Getting to https://github.com/flowing/flowing-retail/blob/master/payment-rest/src/main/java/io/flowing/retail/payment/port/resthacks/PaymentRestHacksControllerV6.java

Client

has to implement

Timeout, Retry,
Compensation

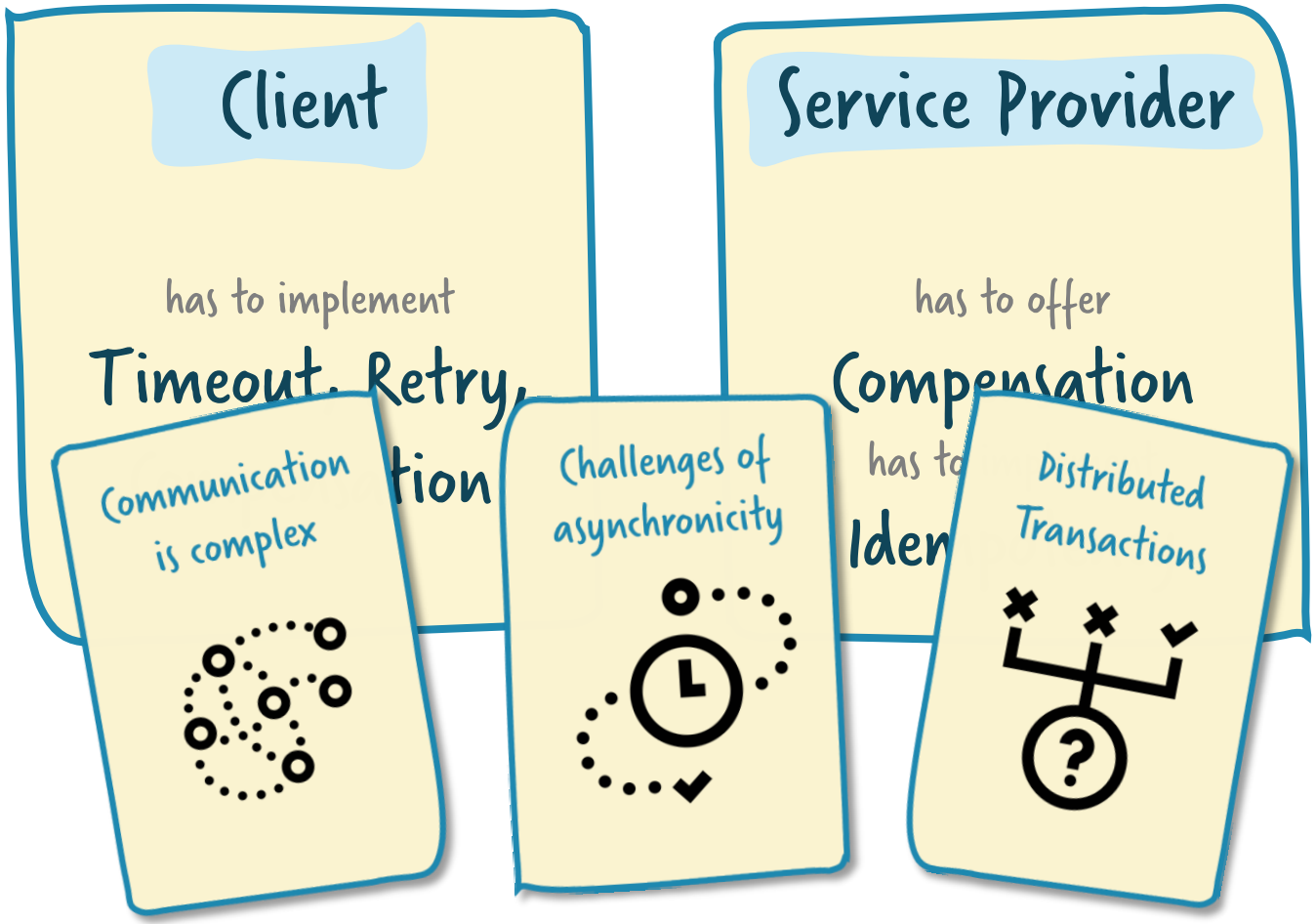
Service Provider

has to offer

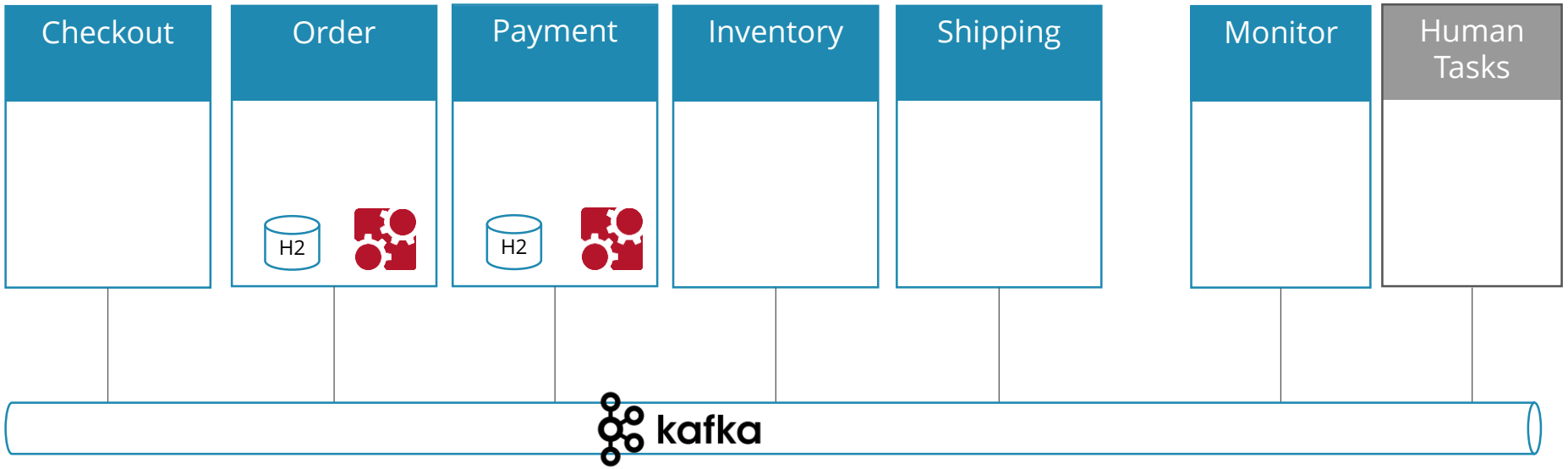
Compensation

has to implement

Idempotency

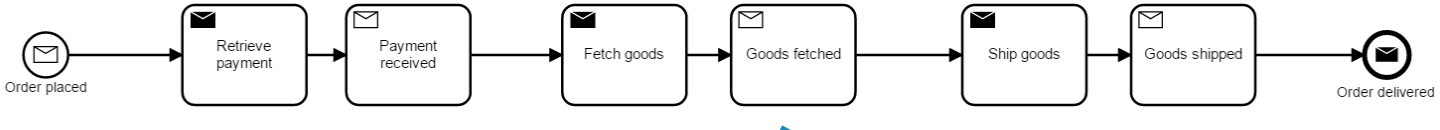

Client
has to implement
Timeout, Retry,
tion
Service Provider
has to offer
Compensation
has to
Iden
Communication is complex
Challenges of asynchronicity
Distributed Transactions

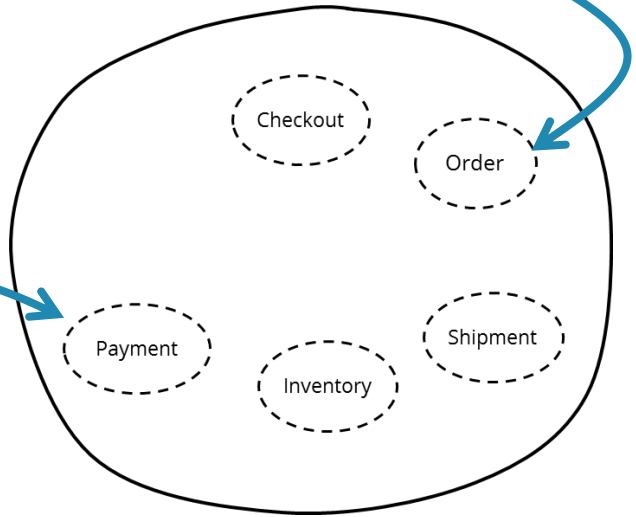
Event-driven example also available

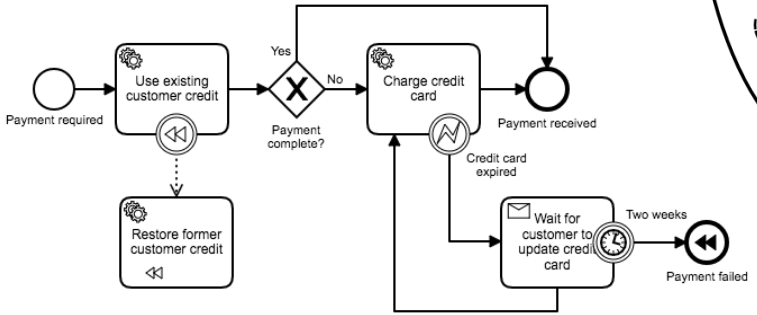


https://github.com/flowing/flowing-retail/

Workflows live inside service boundaries*







*in DDD:
Bounded Contexts

Understand complexity of distributed systems
Know strategies and tools to handle it

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


Thank you!

Contact:
bernd.ruecker@camunda.com
@berndruecker

Slides:
https://bernd-ruecker.com

Blog:
https://blog.bernd-ruecker.com

Code online:
https://github.com/flowing

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

With thoughts from http://flowing.io
@berndruecker | @martinschimak