



SQS and SNS for .NET developers

Mauro Servienti

Particular Software

 @mauroservienti

SPONSOR

DIAMOND



GUCCI

splunk>



PLATINUM

improve



GOLD



CODICEPLASTICO



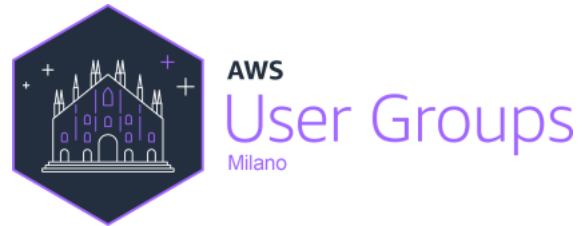
Cubbit

 CLOUD DAY 2022

#CLOUDDAY2022

PARTNER

 CLOUD DAY 2022

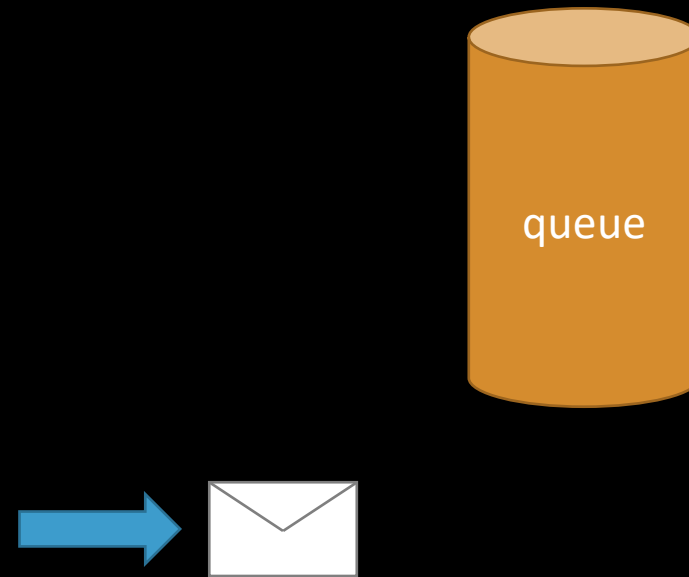


#CLOUDDAY2022

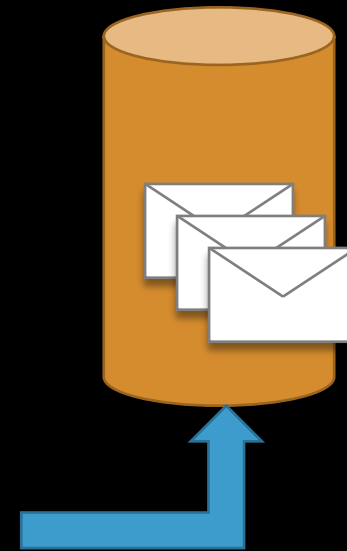
What's SQS?

Simple Queue Service

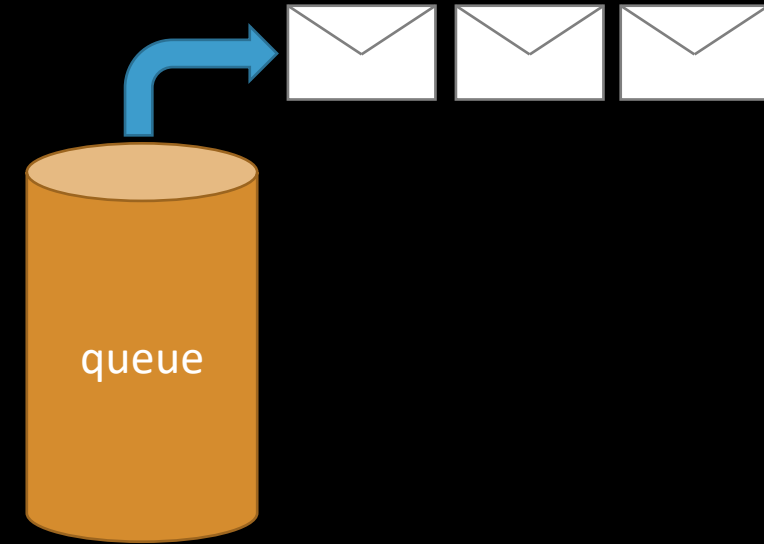
it's a queue
nothing more



it's a queue
nothing more



it's a queue
nothing more



Sure, it's...

- Globally scalable
- Reliable
- Fast, very fast



Maybe it's hard to use

```
var request = new SendMessageRequest(queueUrl, body);  
await sqsClient.SendMessageAsync(request);
```

Maybe batching is scary...

```
var batchEntries = new List<SendMessageBatchRequestEntry>();  
batchEntries.Add(new SendMessageBatchRequestEntry(entry1Id, body1);  
batchEntries.Add(new SendMessageBatchRequestEntry(entry2Id, body2);  
  
var batchRequest = new SendMessageBatchRequest(queueUrl, batchEntries);  
await sqsClient.SendMessageBatchAsync(batchRequest);
```

Test #1: Wed. Sep. 34.4-1.5



then $\cos \theta = \frac{\sqrt{a^2 - x^2}}{a}$
 $a \sqrt{a^2 - x^2} = a \cos \theta$

If $x = a \sin \theta$
 then $dx = a \cos \theta d\theta$

$$= 3^7 \int \sin \theta (\sin \theta) \cos \theta d\theta$$

$$= 3^7 \int (1 - \cos^2 \theta) \cos \theta \sin \theta d\theta$$

$$= 3^7 \int (\cos^3 \theta - 2 \cos^5 \theta + \cos^7 \theta) \sin \theta d\theta$$

$$= 3^7 \left(-\frac{1}{3} \cos^3 \theta + \frac{2}{5} \cos^5 \theta - \frac{1}{7} \cos^7 \theta \right) + C$$

eg $\int x^5 \sqrt{9-x^2} dx$ $\left(\begin{array}{l} \text{let } x = 3 \sin \theta \\ dx = 3 \cos \theta d\theta \\ \sqrt{9-x^2} = 3 \cos \theta \end{array} \right)$

$$= \int (3 \sin \theta)^5 (3 \cos \theta) 3 \cos \theta d\theta$$

$$= 3^7 \int \sin^5 \theta \cos^2 \theta d\theta$$

$$= 3^7 \left(-\frac{1}{3} \left(\frac{\sqrt{9-x^2}}{3} \right)^3 + \frac{2}{5} \left(\frac{\sqrt{9-x^2}}{3} \right)^5 - \frac{1}{7} \left(\frac{\sqrt{9-x^2}}{3} \right)^7 \right) + C$$

$$= -27 (9-x^2)^{3/2} + \frac{18}{5} (9-x^2)^{5/2} - \frac{1}{7} (9-x^2)^{7/2} + C$$

EAT

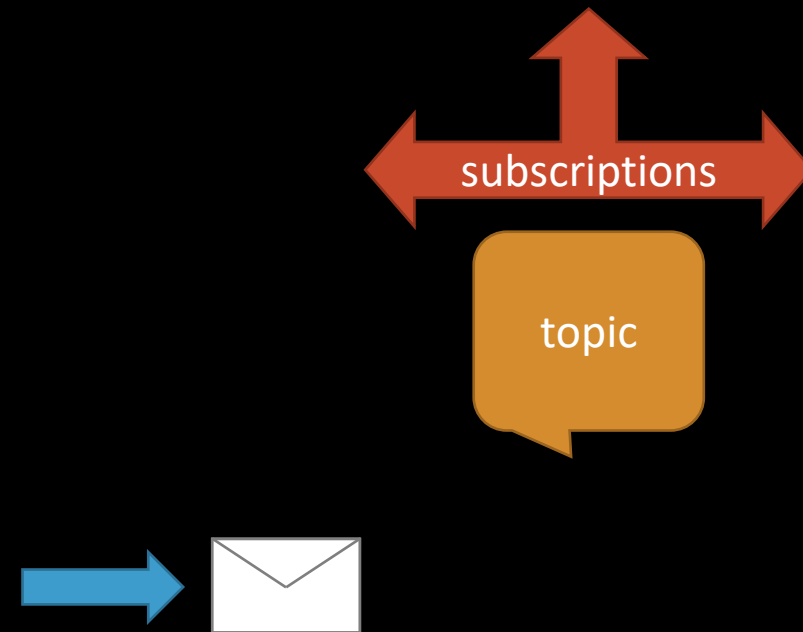
```
var receiveRequest = new ReceiveMessageRequest
{
    MaxNumberOfMessages = numberOfMessagesToFetch,
    QueueUrl = inputQueueUrl,
    WaitTimeSeconds = 20,
    VisibilityTimeout = 30,
    AttributeNames = new List<string> { "SentTimestamp" },
    MessageAttributeNames = new List<string> { "*" },
};
var messages = await sqsClient.ReceiveMessageAsync(receiveRequest)
```



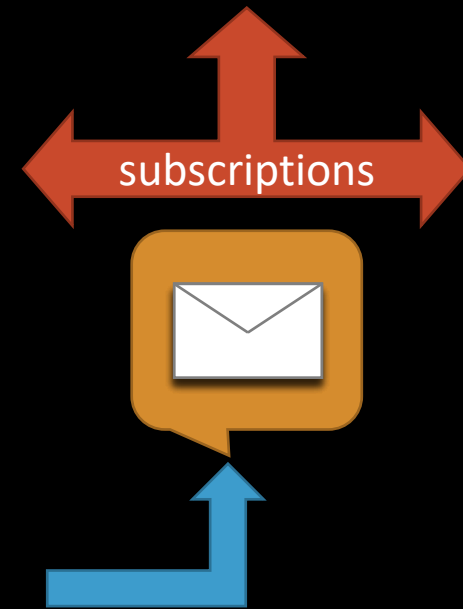
What's SNS?

Simple Notification Service

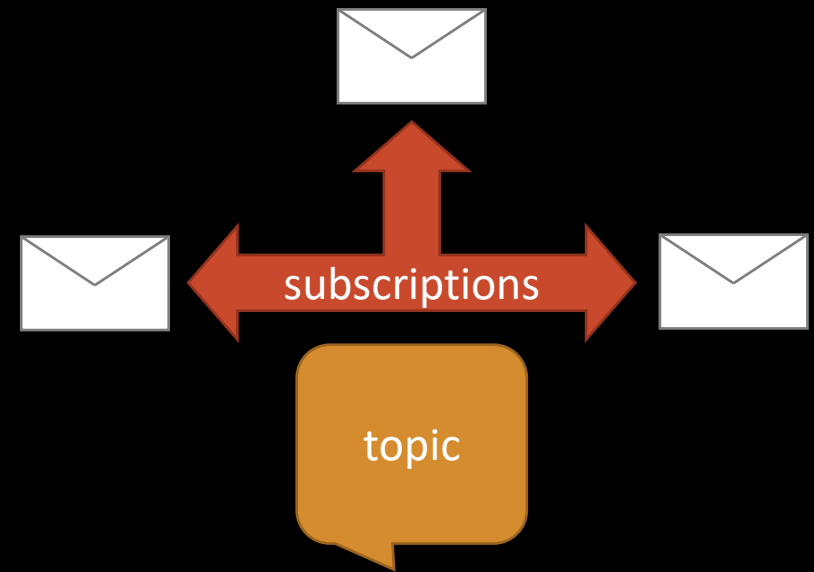
it's a broadcast
mechanism



it's a broadcast
mechanism



it's a broadcast
mechanism



Sure, it's...

- Globally scalable
- Reliable
- Fast, very fast



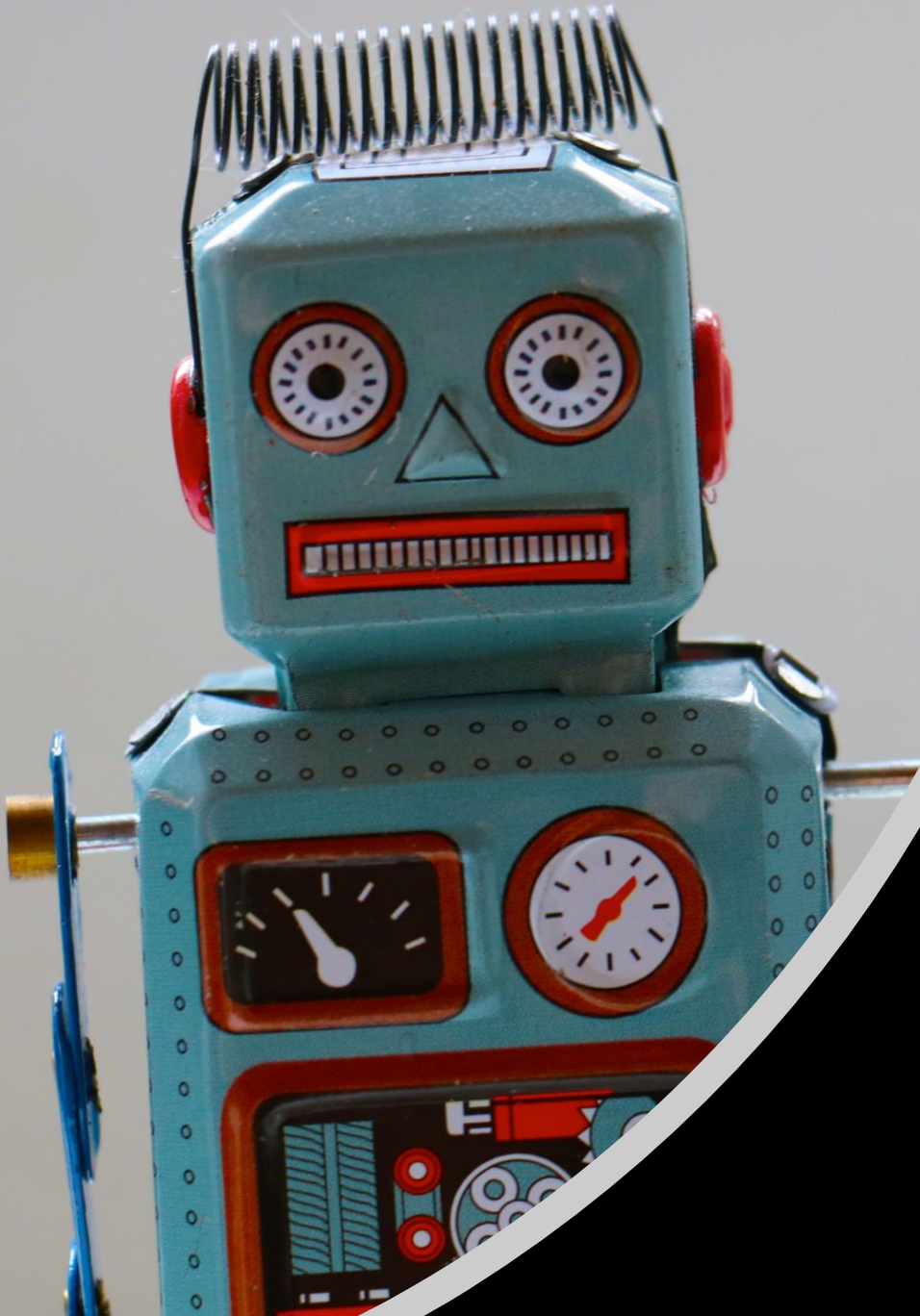
Maybe publishing is hard


```
var request = new PublishRequest(destinationTopic, body);  
await sqsClient.PublishAsync(request);
```



they are simple





why are
we here?



why do we
want to use
messaging?

A business requirement



Important messages about items in your Cart:

3 items in your Cart have changed price.

Items in your Shopping Cart will always reflect the most recent price displayed on their product detail pages.

- [A Dog's Purpose: A Novel for Humans](#) ▼ has increased from \$9.10 to \$9.36
- [A Dog's Purpose](#) ▼ has decreased from \$8.92 to \$8.46
- [Samsung Internal SSD \(MZ-75E1T0B/AM\)](#) ▼ has increased from \$324.99 to \$349.99

most recent price displayed
increased from \$9.10 to \$9.36
decreased from \$8.92 to \$8.46
[MZ-75E1T0B/AM](#) ▼

Products & Shopping Cart Microservices

Products

- Item ID
- Price
- Inventory

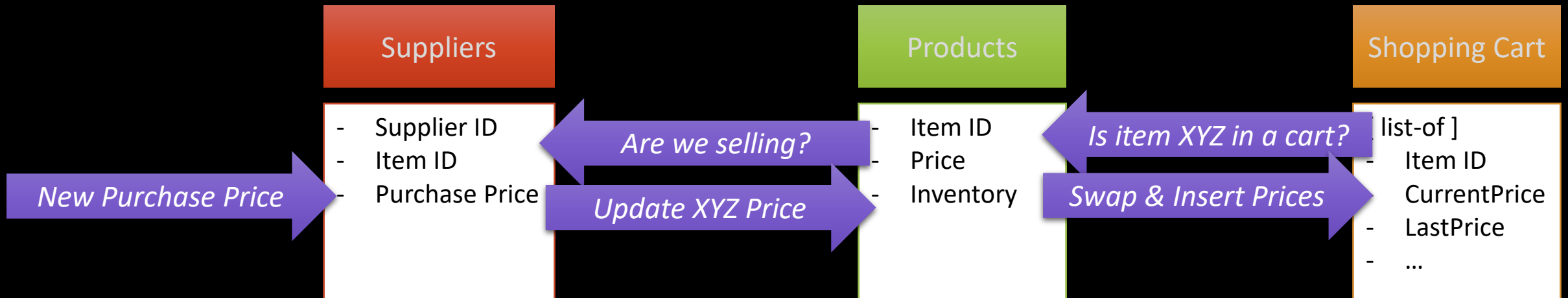
Shopping Cart

- [list-of]
- Item ID
- CurrentPrice
- LastPrice
- ...

Products & Shopping Cart Microservices



Suppliers & Products & Shopping Cart

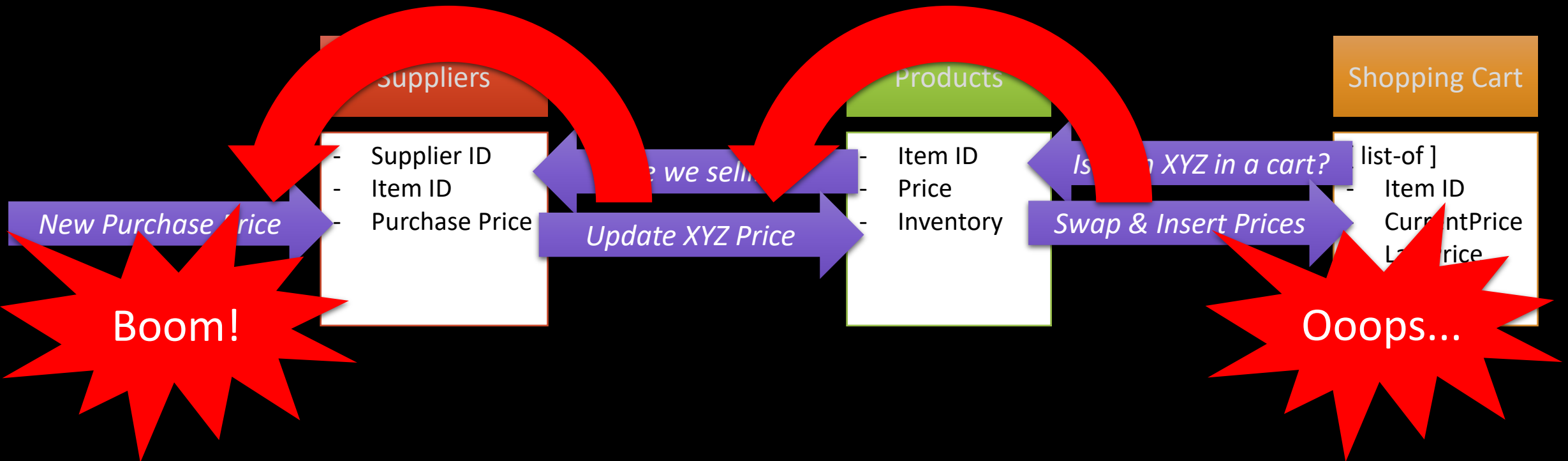


“the microservices way” (trolling)

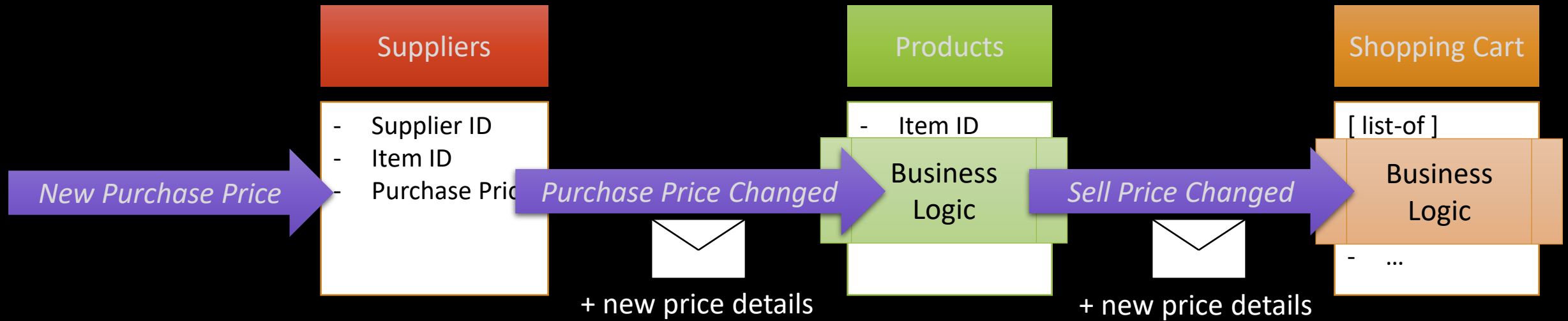
- Incoming “external” HTTP Post
- HTTP Query from Suppliers to Products
- HTTP Post from Suppliers to Products
- HTTP Query from Products to Shopping Cart
- HTTP Post from Products to Shopping Cart



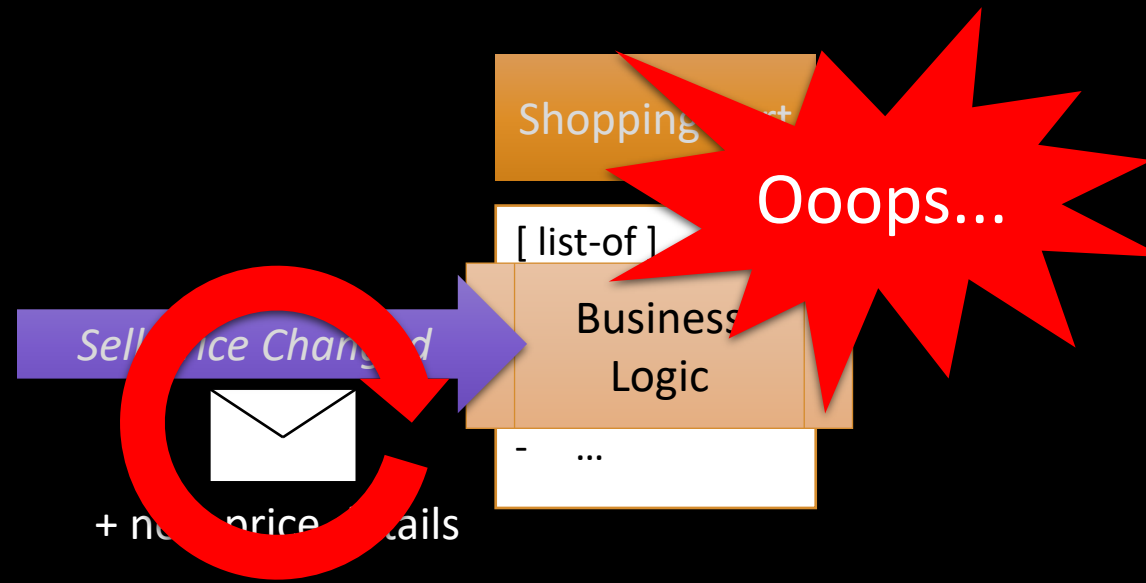
The snowball effect



Messages at play



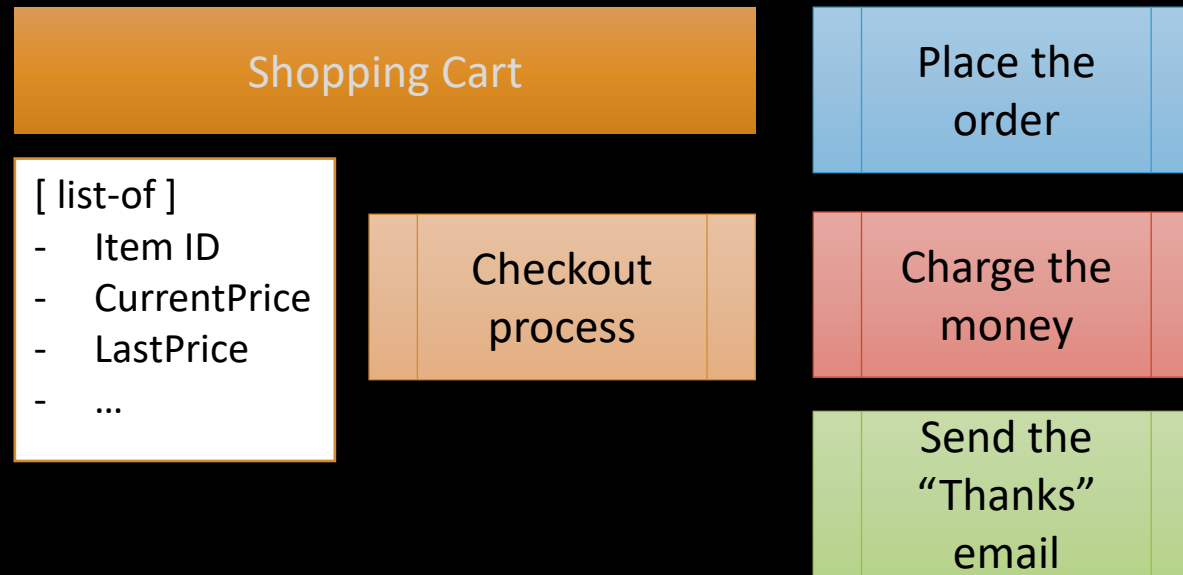
Failures



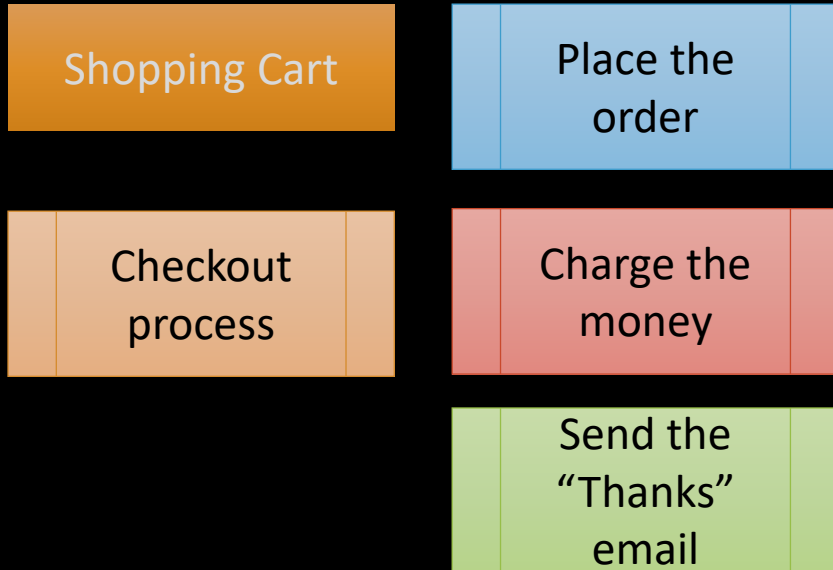
Checking out...



Checking out requires multiple steps



Interesting facts

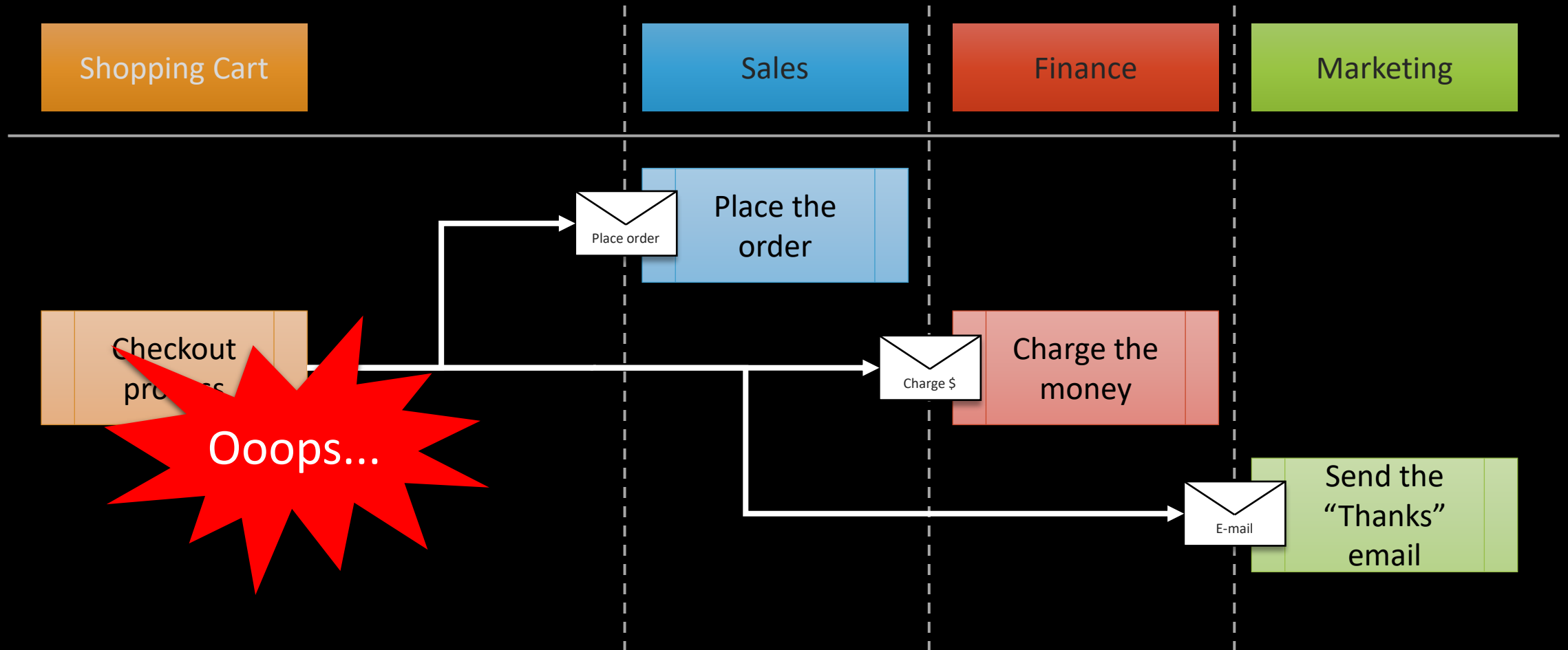


The order doesn't matter
It's all or nothing

distributed transactions



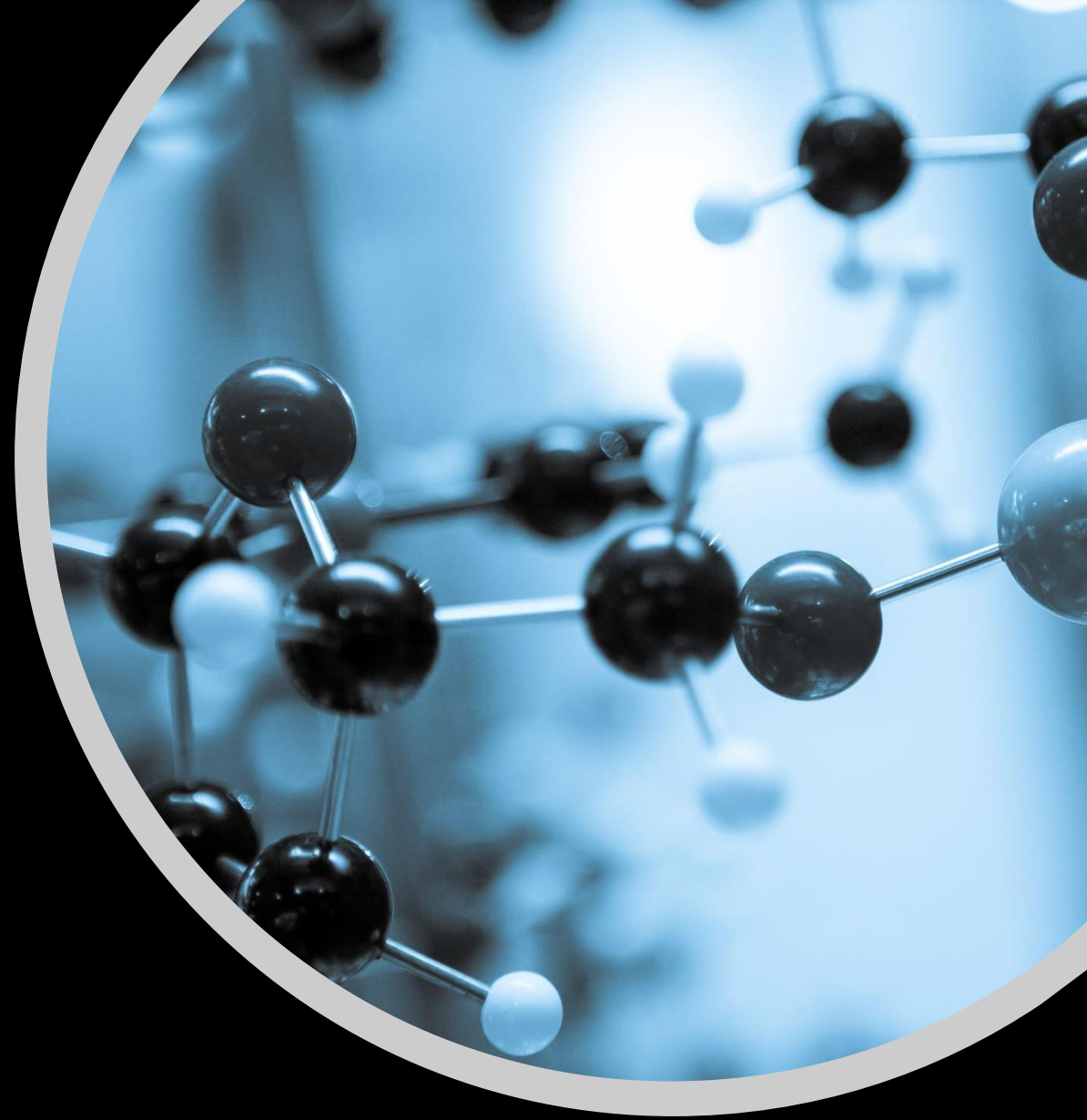
Use messaging

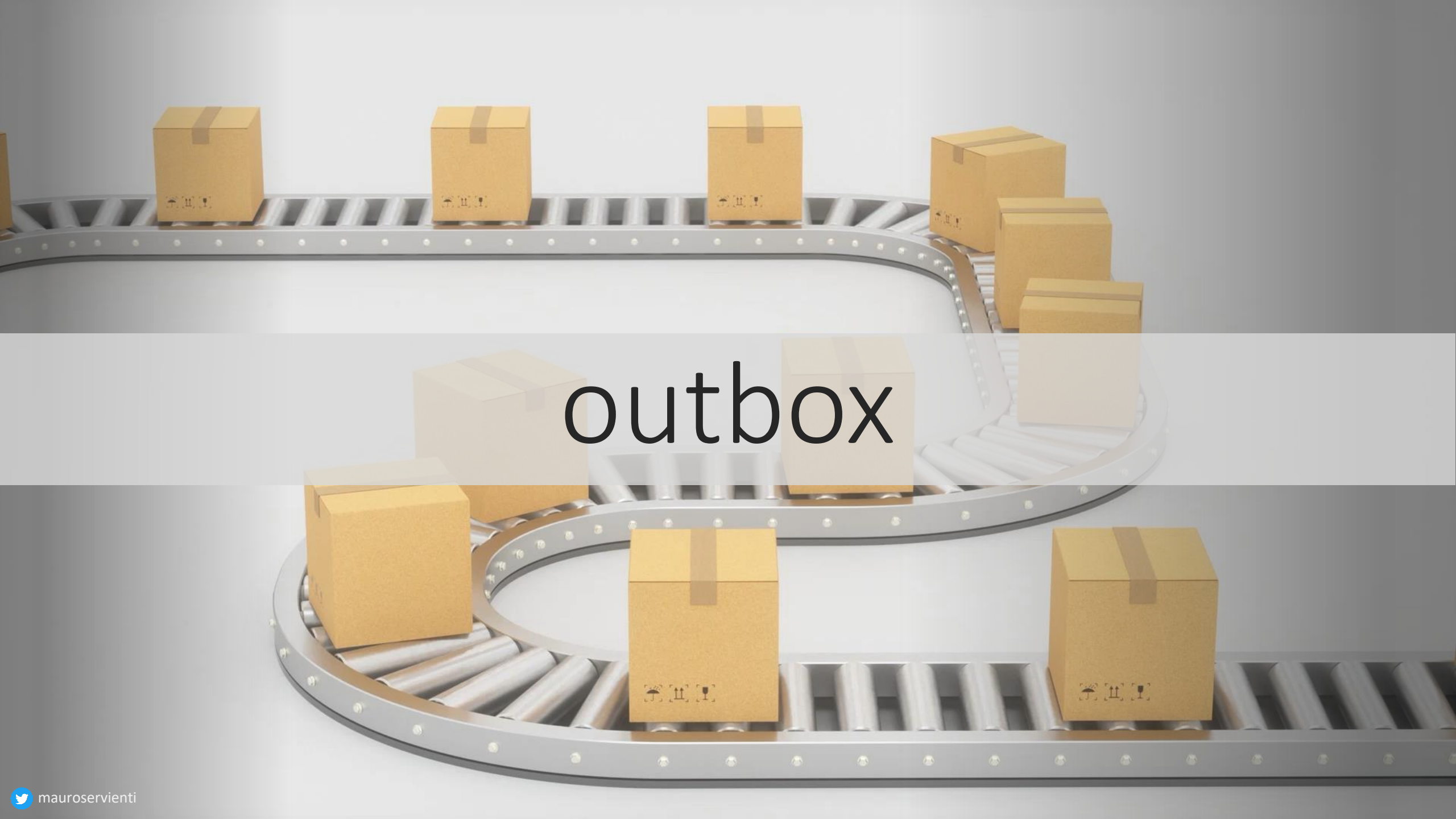


can we use batching?

no

batches are
not atomic



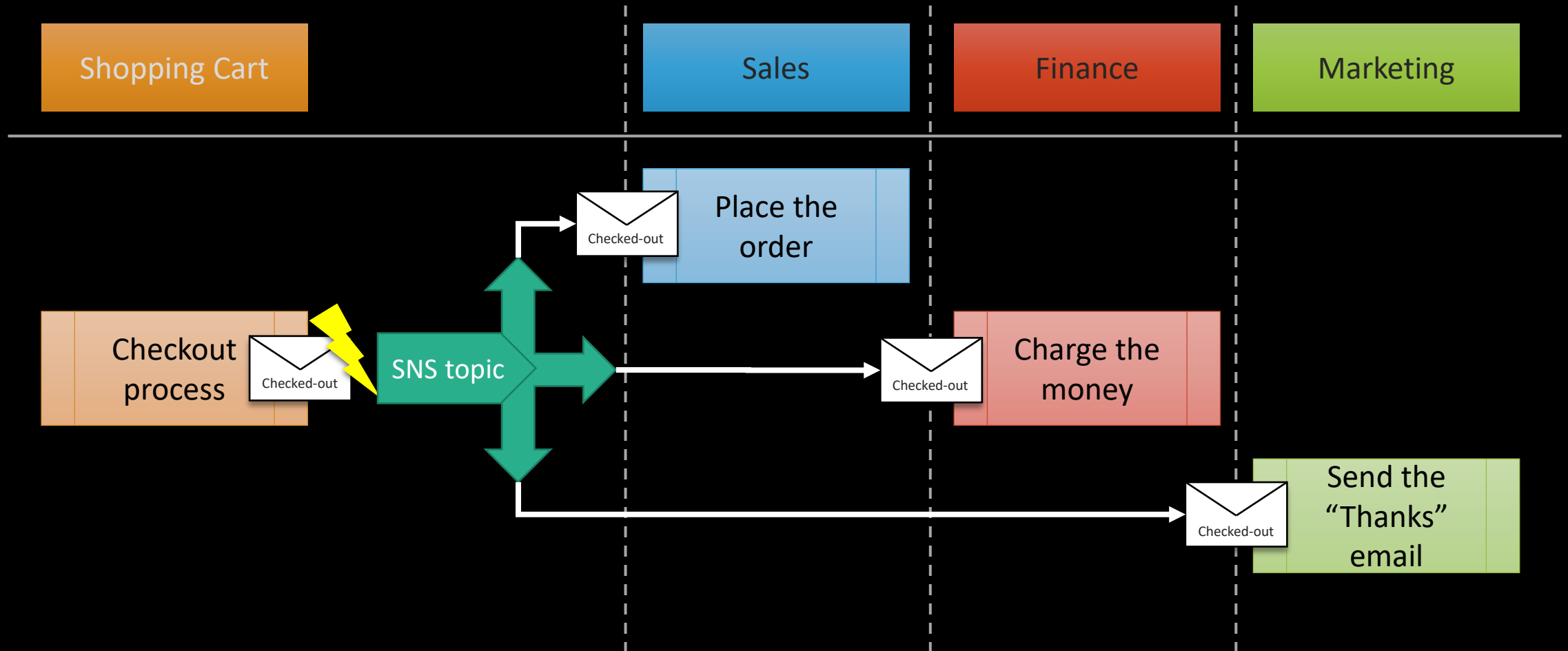


outbox

it's useful, but there's a simpler solution



Use events, Luke



Recap

- Use SQS for one-to-one communication
 - Commands
- Use SNS for one-to-many communication
 - Events
- Use the outbox pattern
 - Local coordination of different operations
- Design retry patterns and idempotency
 - Should we retry charging the credit card?



We need more

- Routing
- First and second level retries
- Serialization
- Outbox and transactional sessions
- Distributed sagas and workflows

Messaging frameworks

- Brighter
- MassTransit
- NServiceBus
- Rebus



aws / dotnet Public

Watch 51

Fork 67

Star 408

<> Code

Issues 3

Pull requests 1

Actions

Projects

Security

Insights

Design Doc: AWS Messaging Framework #42

New issue

Open

ashovlin opened this issue 7 days ago · 14 comments



ashovlin commented 7 days ago · edited by ik130

Member



The AWS .NET team is exploring creating an AWS native framework that simplifies development of .NET message processing applications using AWS services.

The design doc can be viewed and commented on in PR #41 (rendered view).

The purpose of the framework would be to reduce the amount of boiler-plate code developers need to write. The primary responsibilities of the proposed framework

Assignees

No one assigned

Labels

feature-request

needs-discussion

Projects

Slides

[//bit.ly/sqs-sns-net-dev](https://bit.ly/sqs-sns-net-dev)



thank you

Mauro Servienti

Solution Architect — Particular Software, the makers of NServiceBus

mauro.servienti@gmail.com | [@mauroservienti](https://twitter.com/mauroservienti) | [//milestone.topics.it](https://milestone.topics.it)