NServiceBus Workshop

HCSS

CLEAR MEASURE

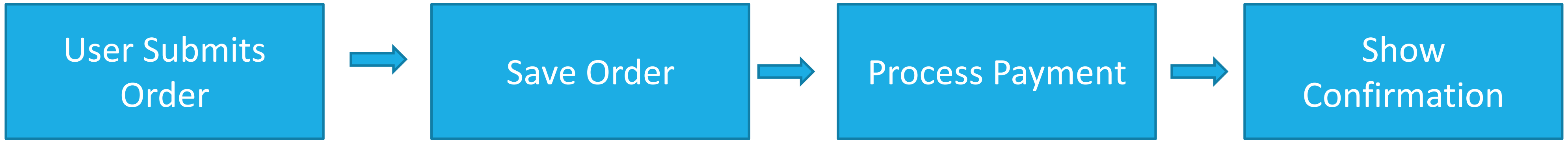
Justin Self

- Principal Solution Architect @ Clear Measure
- Theater Major
- Terrible Wood Worker
- Austin
- Family
- WesternDevs.com
- Justinself.com

Rough Outline

- Intro – 20 minutes
- Creating Commands – 90 minutes
- Creating Events – 90 minutes
- Break – 60 minutes
- Sagas – 10 minutes
- Customer VIP Saga – 90 minutes
- Shipping Saga – 90 minutes

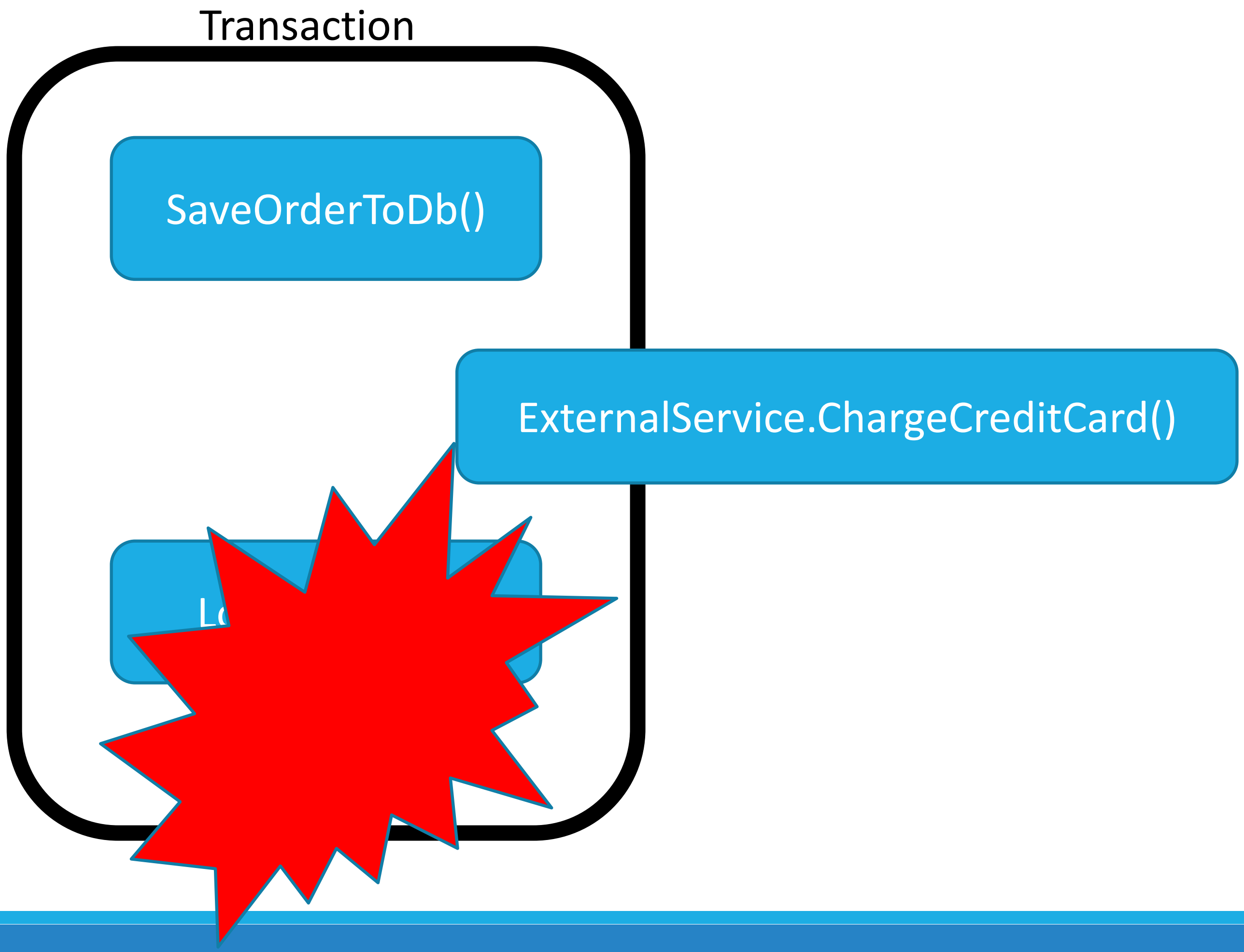
A Common Scenario



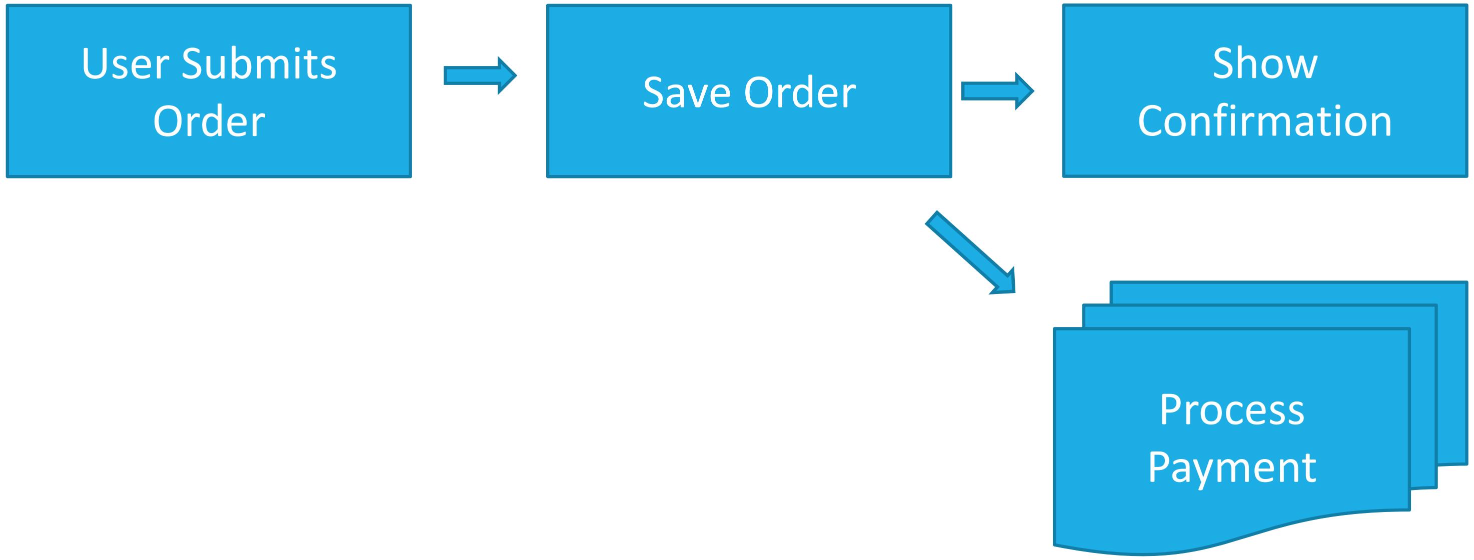
A Common Problem

```
 using (var scope = new TransactionScope())
 {
    SaveOrderToDb();
    ExternalService.ChargeCreditCard();
    LogPaymentProcess();
    scope.Complete();
}
```

A Common Problem: Now with shapes!



A Common Solution



A Common Solution: But With Questions

- How to handle failure?
- How to re-queue?
- How to reprocess?
- How to poll queue?
- How to host process?
- What happens if queue is offline?
- What happens if the application crashes when processing?
- What happens if network fails during polling?

NServiceBus

- What is NServiceBus?
- Why is there an ‘N’ in NServiceBus?
- What’s with this logo?

What is NServiceBus?

A light weight messaging framework that focuses on ‘bilities’ in your distributed systems:

- Durability
- Reliability
- Scalability
- Extensibility
- Performance-bility
- Make me look good when things go wrong-bility
- Take care of things so I don’t need to write them myself and can focus on my custom software-bility

What is a Bus?

It’s a reliable way of moving messages from one process to another.

What is a Bus?

It’s not a broker.

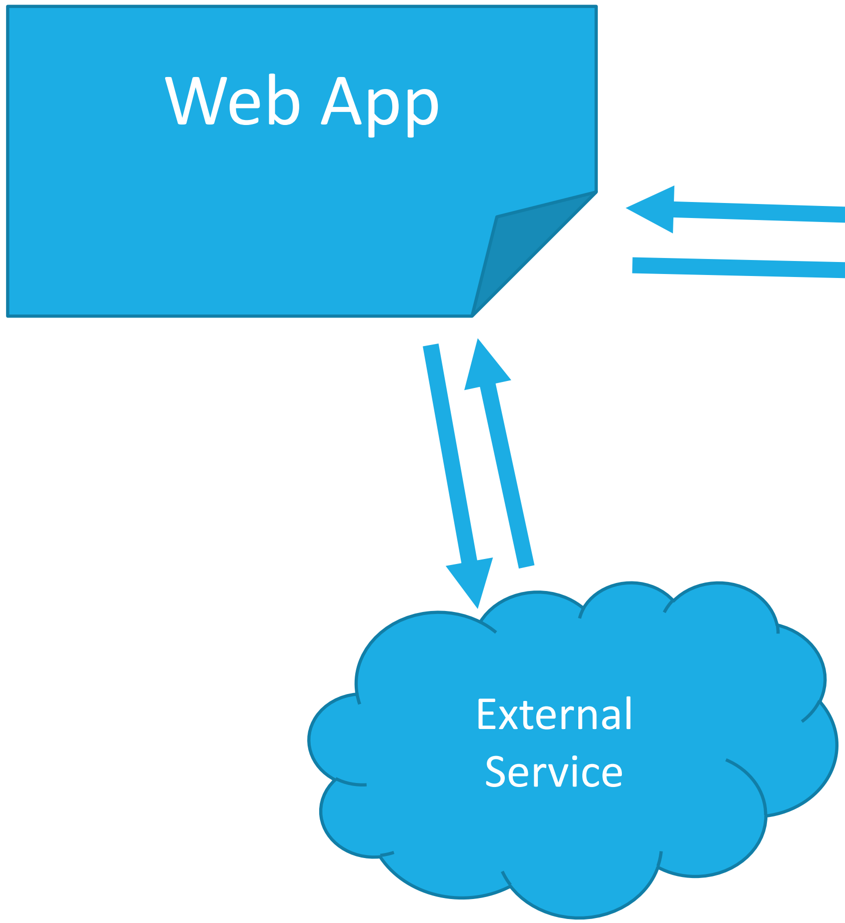
Persistence

1. InMemory
2. RavenDB
3. Nhibernate
4. MSMQ
5. Azure Tables

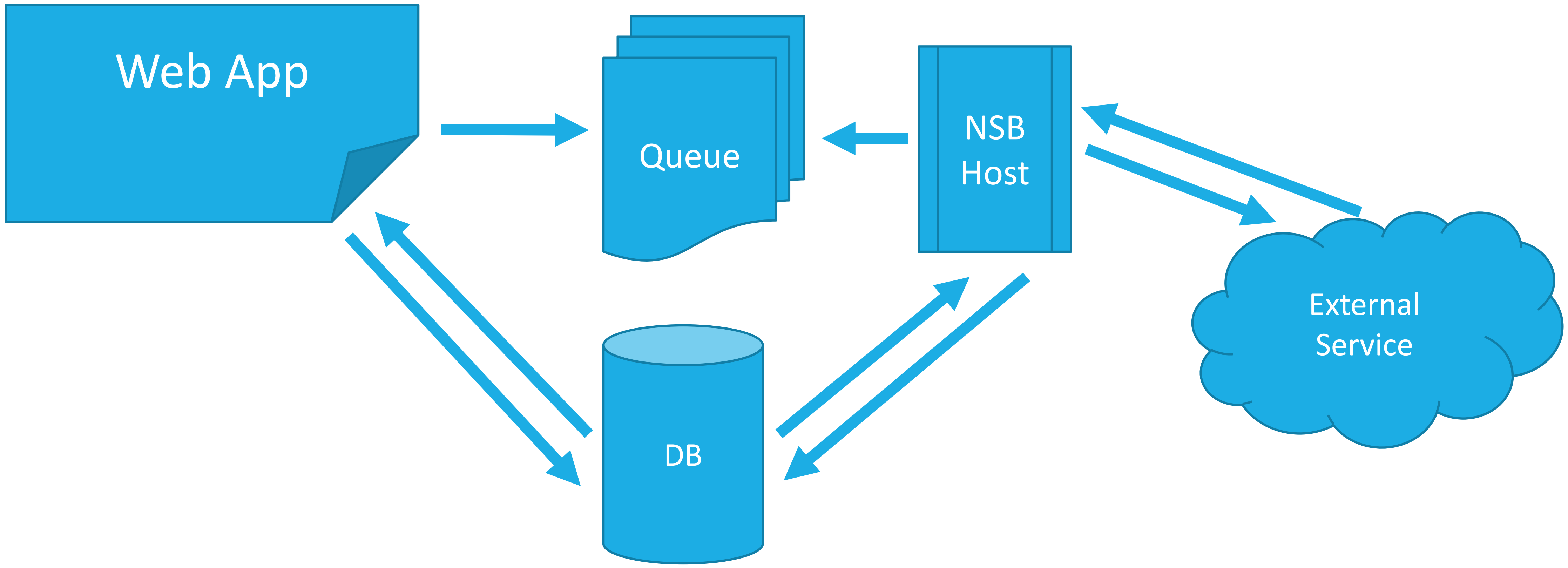
Transports (Queueing Infrastructure)

1. MSMQ – Default
2. RabbitMQ
3. SqlServer
4. Azure (Queues, ServiceBus)

Example Architecture



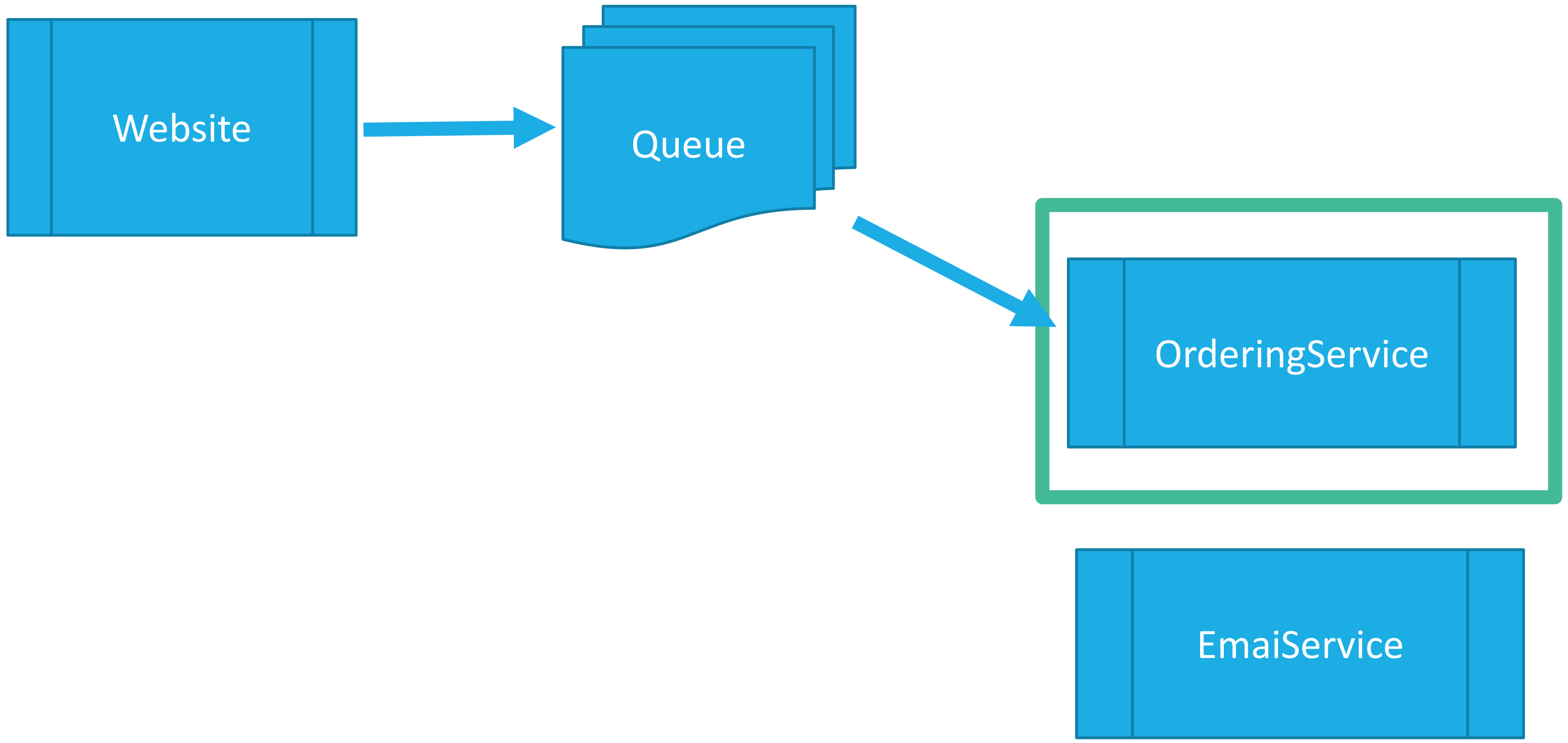
Example Architecture



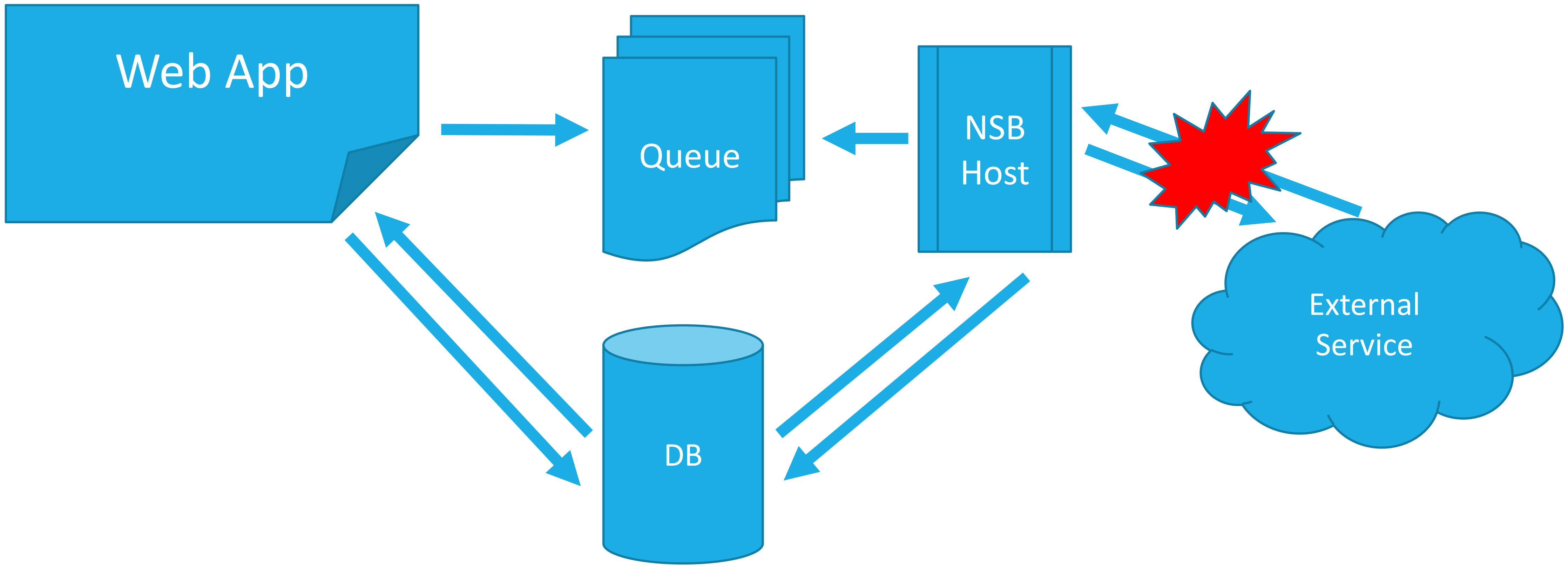
Commands

- Action Oriented, Imperative and Specific (From your Ubiquitous Language)
    - DeleteProfile
    - PlaceOrder
    - UnsubscribeEmail
    - SubmitRefund
- Typically implements ICommand
- Processed by exactly one logical endpoint
- Sent by n logical endpoints

Commands



Example Architecture



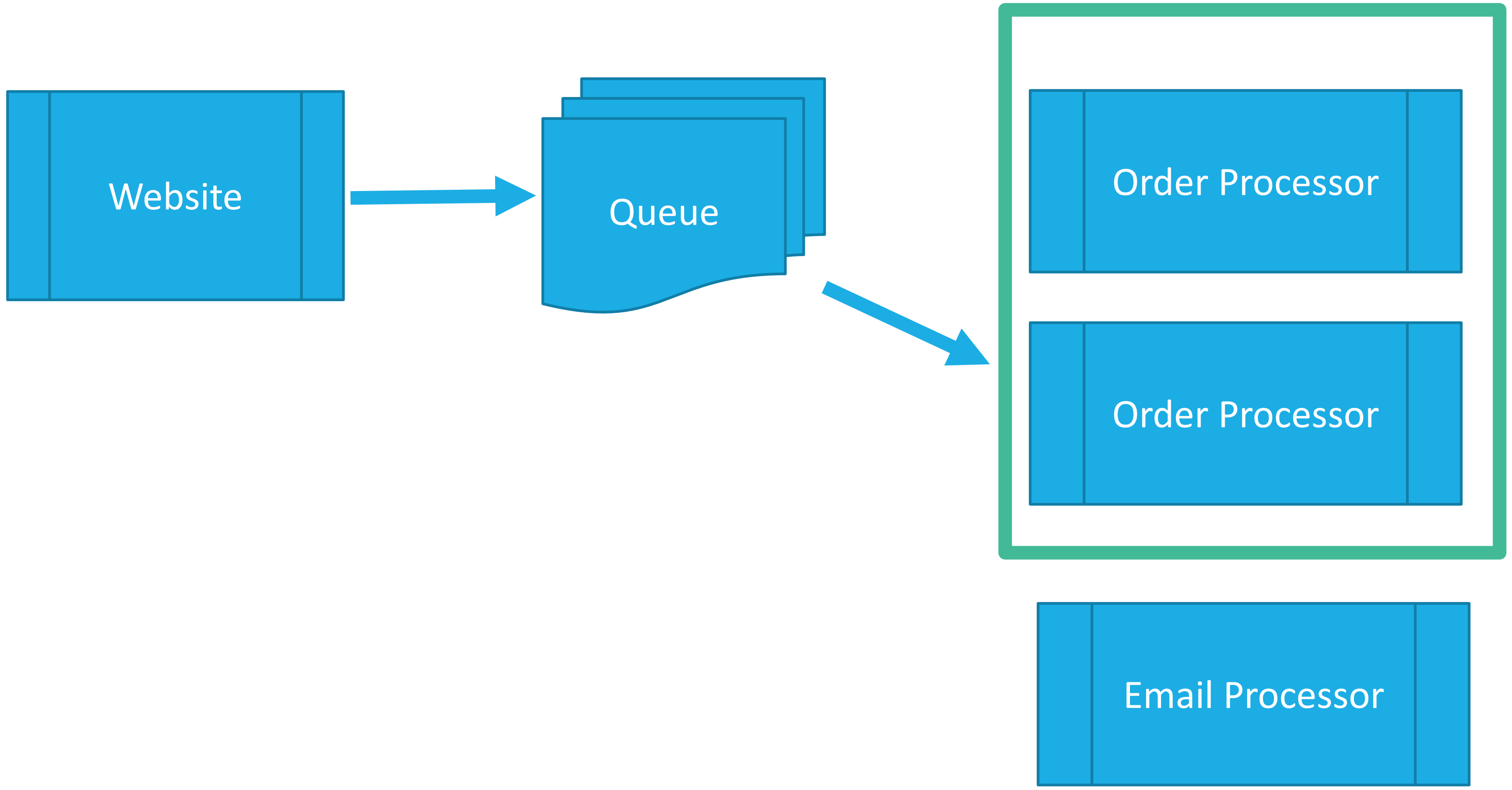
Basic Steps

1. NServiceBus Creates Transaction
2. Pulls message from Queue
3. Find associated Handler for Message
4. Handler calls 3rd party service
5. Call fails, exception thrown
6. Transaction rolls back
7. Message put back on Queue

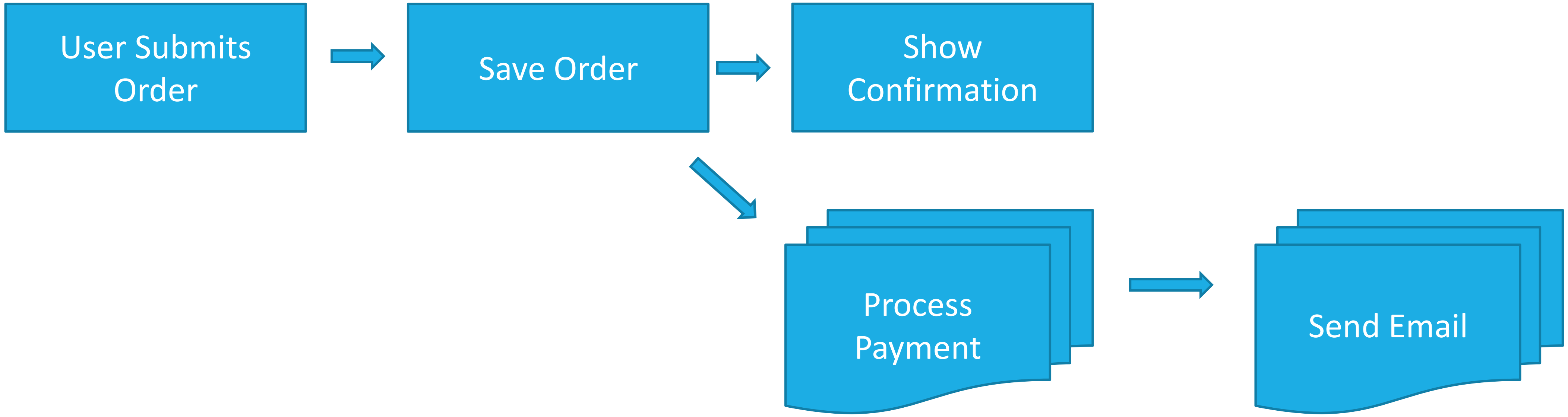
Failure

1. First Level Retry (FLR)
2. Second Level Retry (SLR)
3. Error Queue

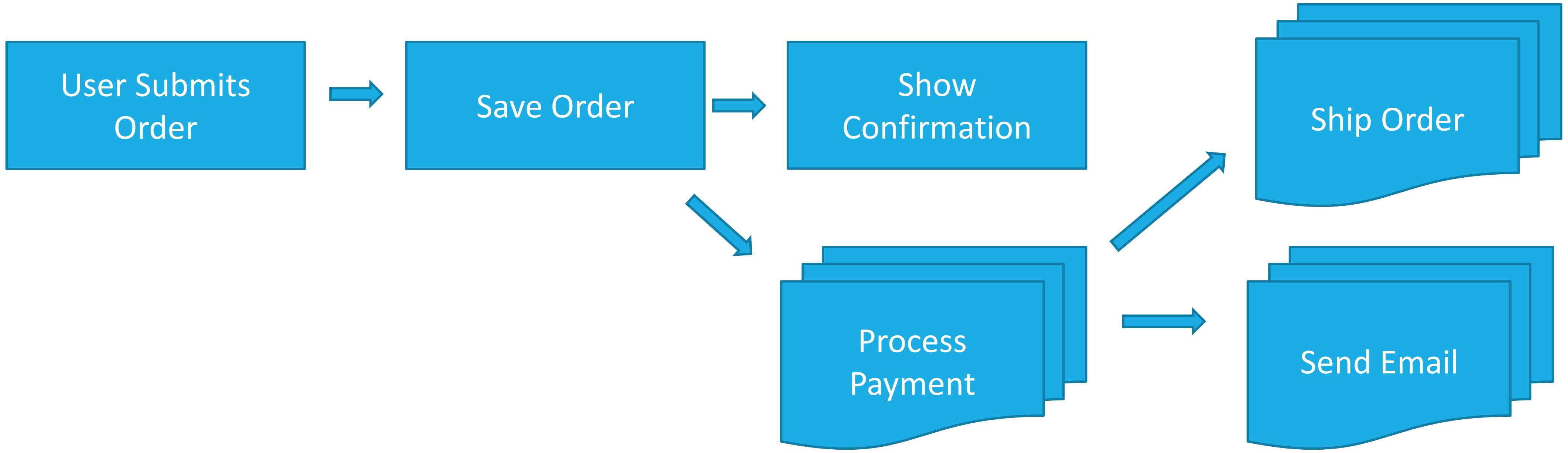
Scaling


Website
Queue
Order Processor
Order Processor
Email Processor

Adding Extensibility



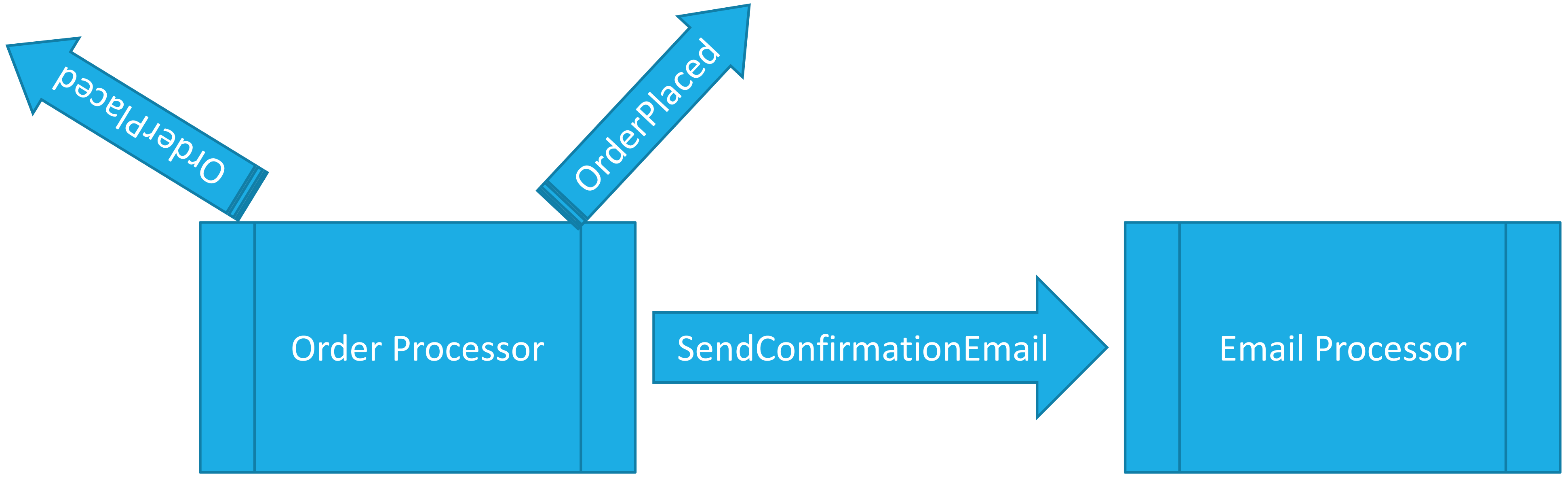
Adding Extensibility



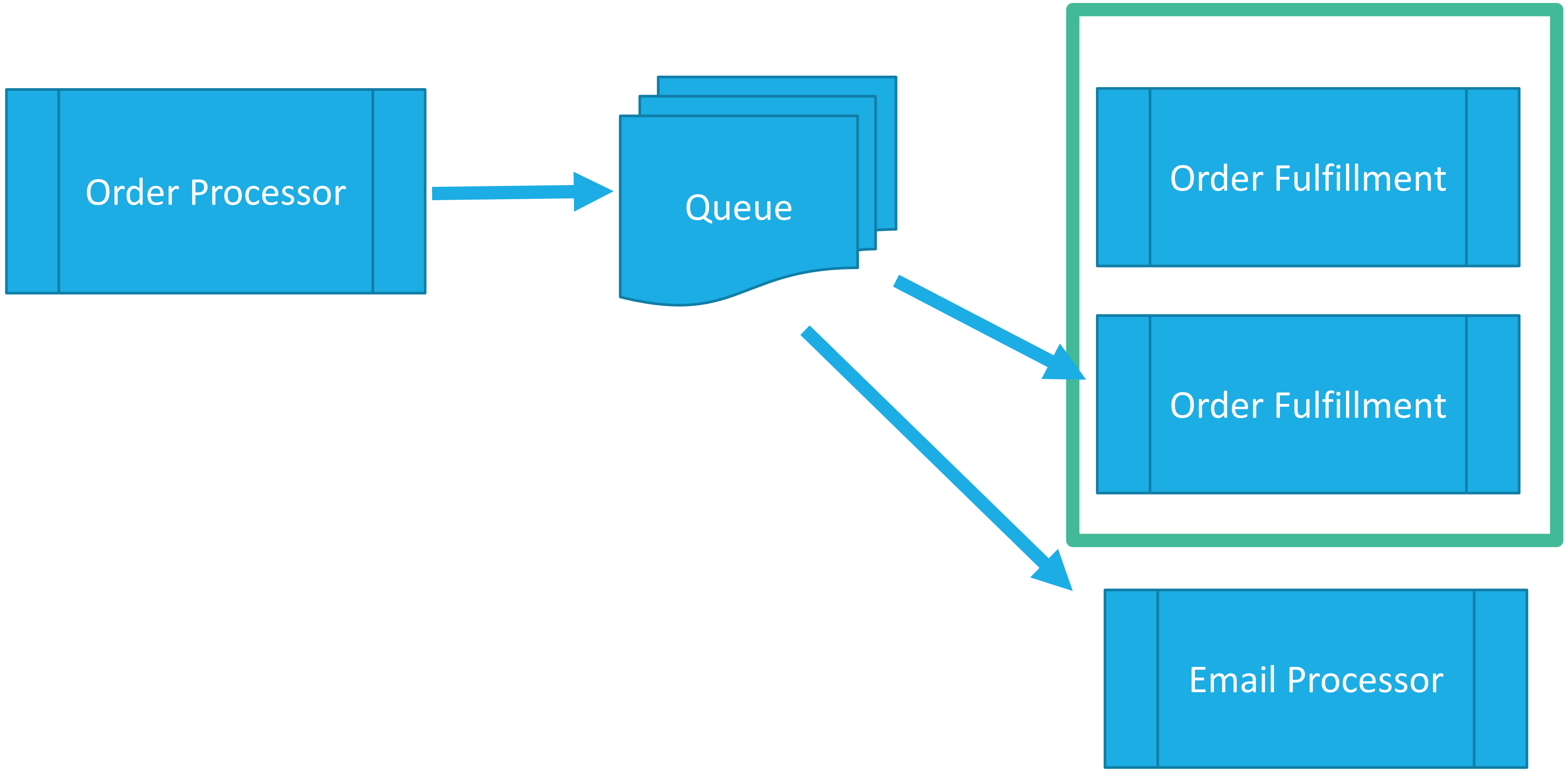
Events

- Past Tense – These have already happened
    - ProfileDeleted
    - OrderPlaced
    - EmailUnsubscribed
    - RefundSubmitted
- Sent by exactly one logical endpoint
- Processed by 0-n logical endpoints
- Typically implements IEvent

DeCoupling with Events



Events


Order Processor
Queue
Order Fulfillment
Order Fulfillment
Email Processor

NSB + ASB vs Native ASB

NSB + ASB

1. Development Framework (with different Transport abstractions)
2. Coupled to an abstraction
3. Solid Development Model (error handling, retries, Sagas, serialization, distributed messaging concepts, DataBus)
4. Extensibility
5. Separate DLQ and Error Queue
6. Legitimate Monitoring and ProdOps Tools
7. Developer and Debugging tool
8. Utilize Databus for large message bodies
9. Atomic Send/Receive works OOTB
10. HA and Throttling handled OOTB

RELIABLE Messaging Oriented

Native ASB

1. Transport Only
2. Coupling to a specific service offering
3. No Development Model
4. Mixed Use of DLQ
5. Bland Tooling
6. No Message Correlation
7. Native Batching doesn’t work well
8. Difficult to utilize Atomic Send/Receive
9. HA and Throttling is manually handled
10. Must manage message sizes

Messaging Oriented

Resources

1. Particular.net
2. StackOverflow
3. Learning NServiceBus Second Edition by David Boike
4. Jimmy Bogard on Lostechies.com/jimmybogard

Want more?

Get access to 2 full days of video

from Udi Dahan’s 5 days SOA course.

http://go.particular.net/TXUG

Access code : SELF

Expiration date: September 5th

Particular Software

Contact Info

justinself.com

@thejustinself

justin.self@clear-measure.com

clear-measure.com

github.com/justinSelf/nsbSample