

Knock. Knock. Who's there?

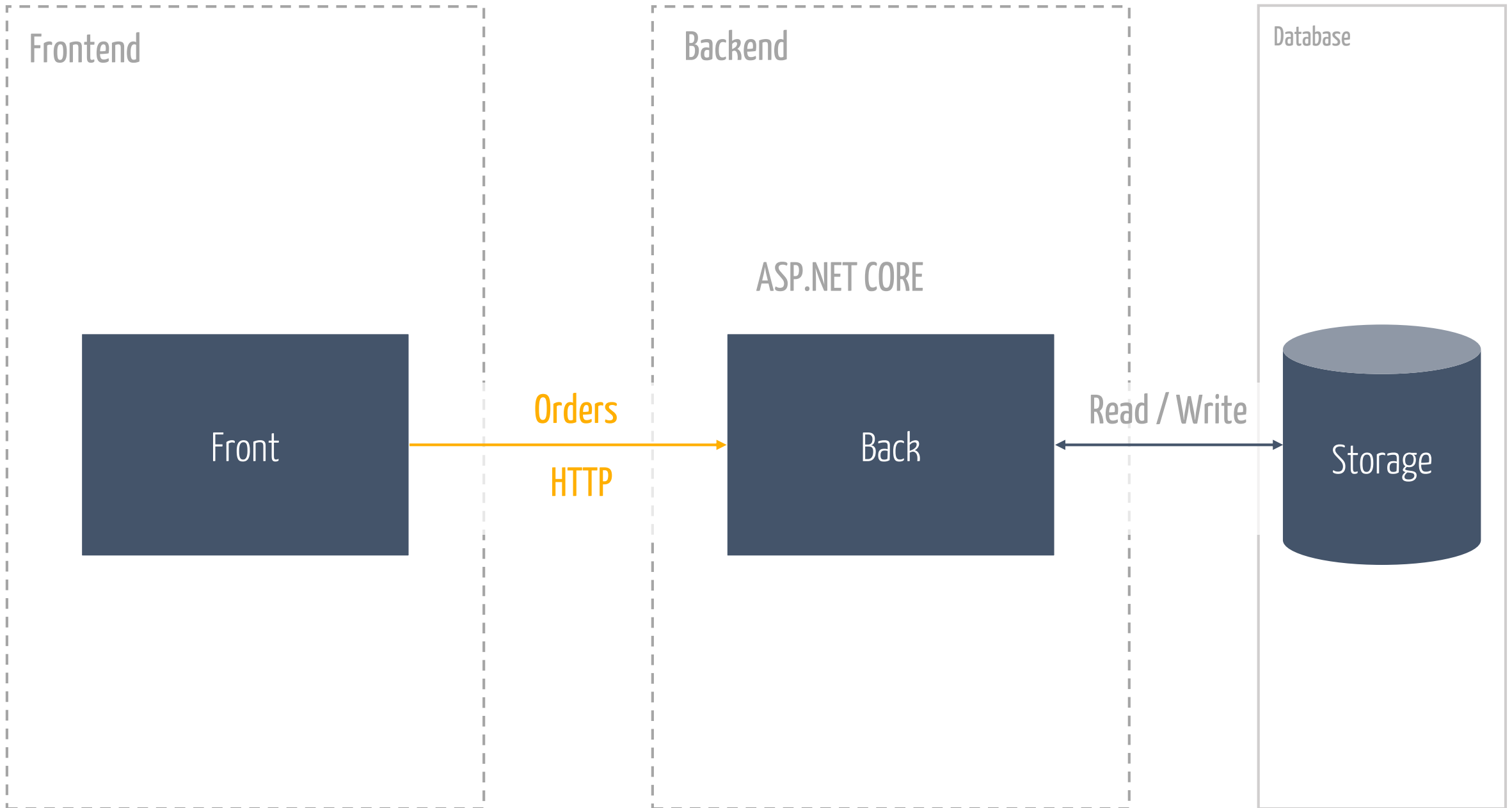
A message from the future



Hiding Inconsistencies

Multi user
collaboration

Multi user
collaboration

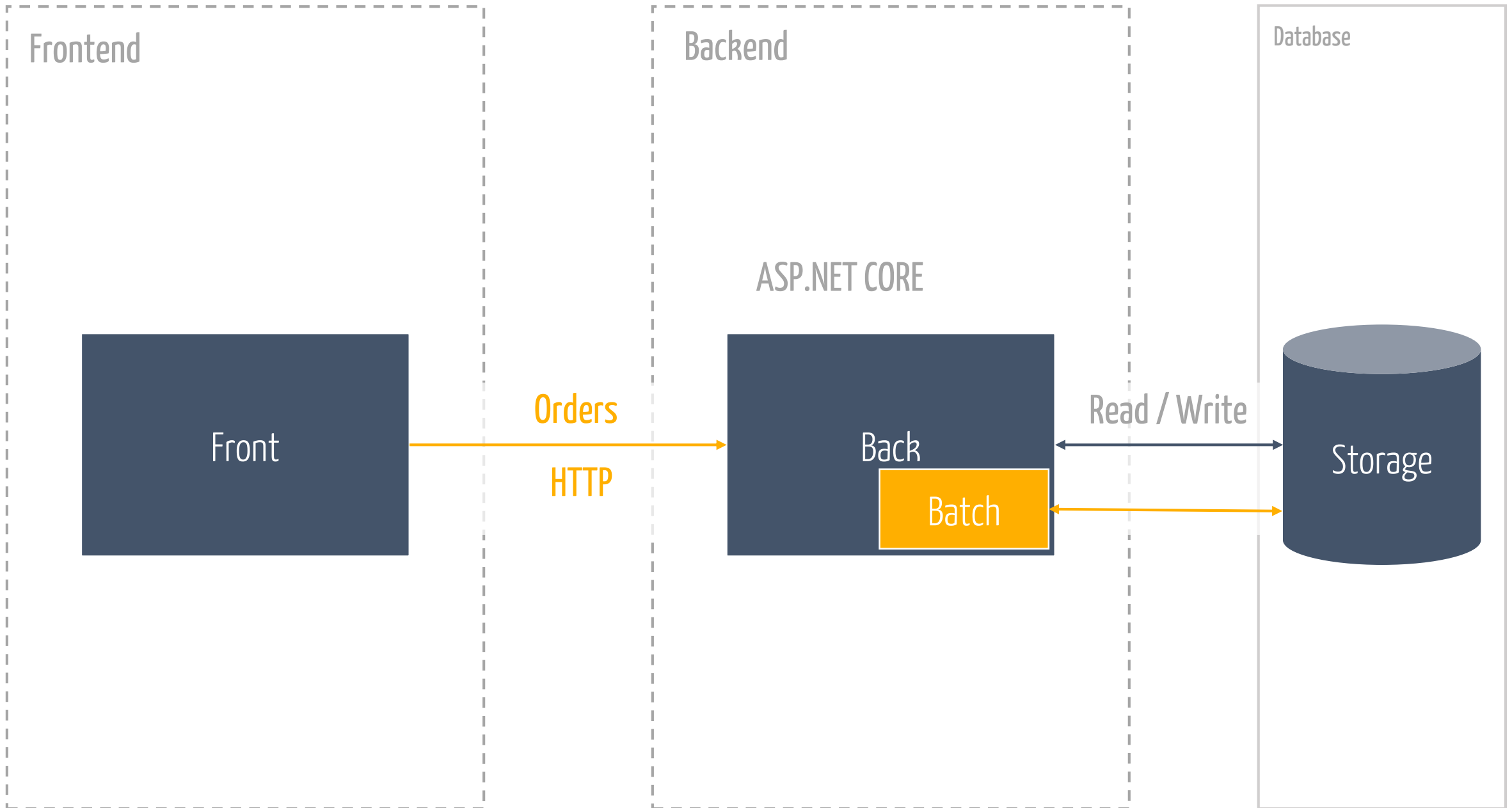


And then the customer
walks into your office



Concurrency and latency in collaborative domains can lead to incorrect application of business rules even in safe architectures

Batch Jobs

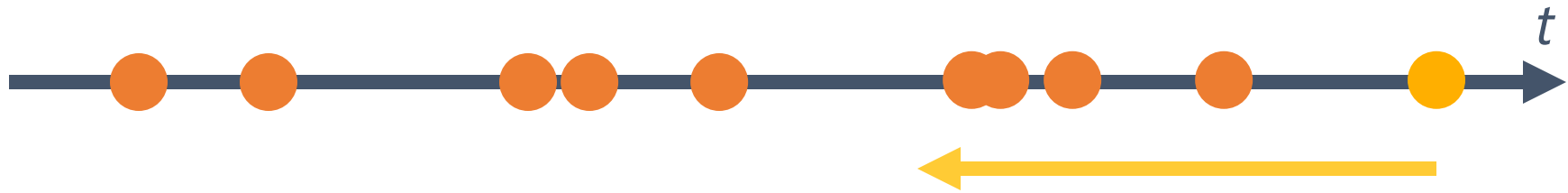


Batch Jobs increase the window of
consistency problems

Batch Jobs for time based business rules
are the **worst enemy** for growth

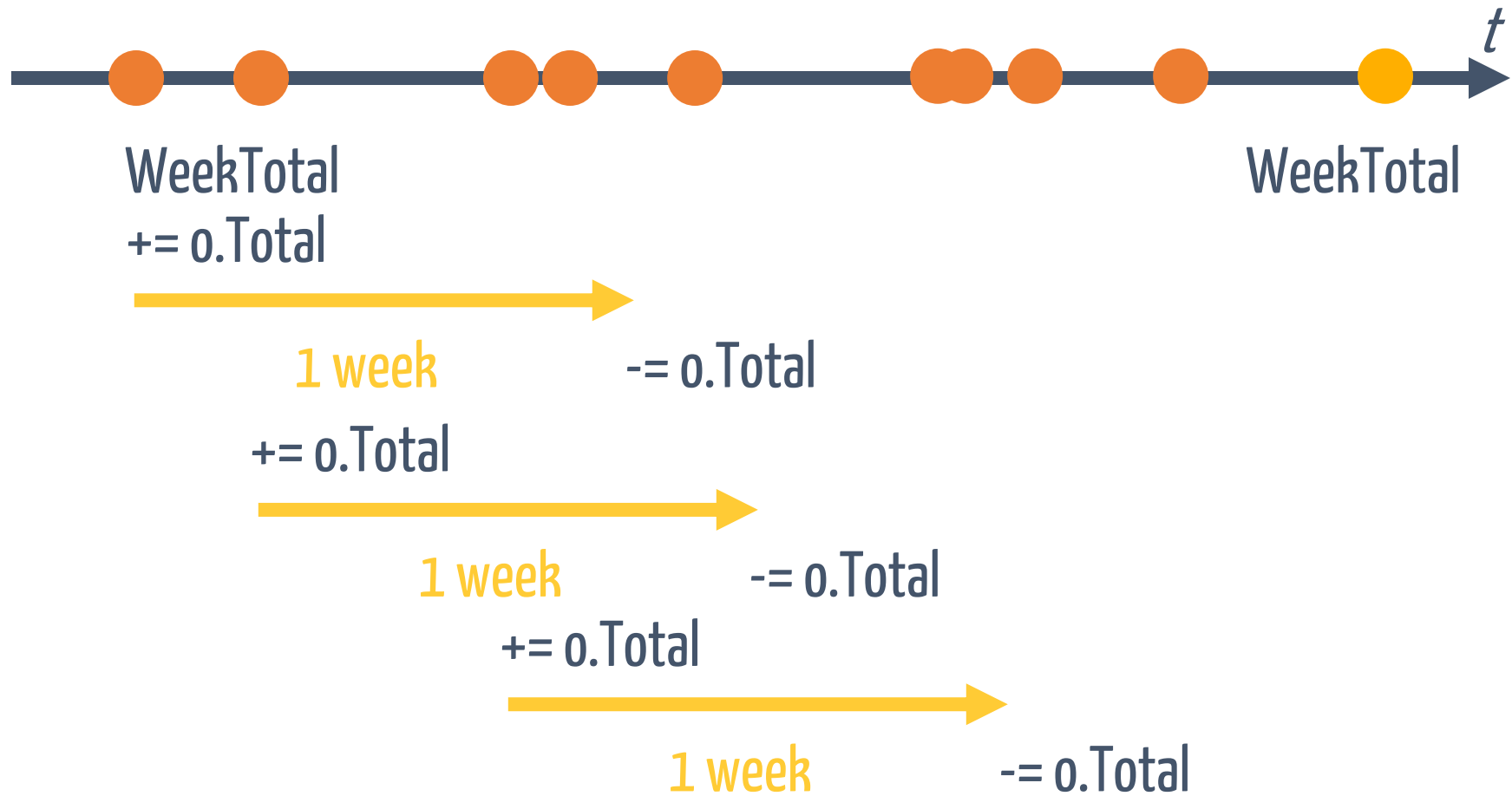
The background consists of several overlapping clock faces in various shades of gray. Some clock faces show Roman numerals, while others show Arabic numerals. The clocks are arranged in a way that they appear to be layered, with some partially obscured by others. The overall effect is a sense of time and movement.

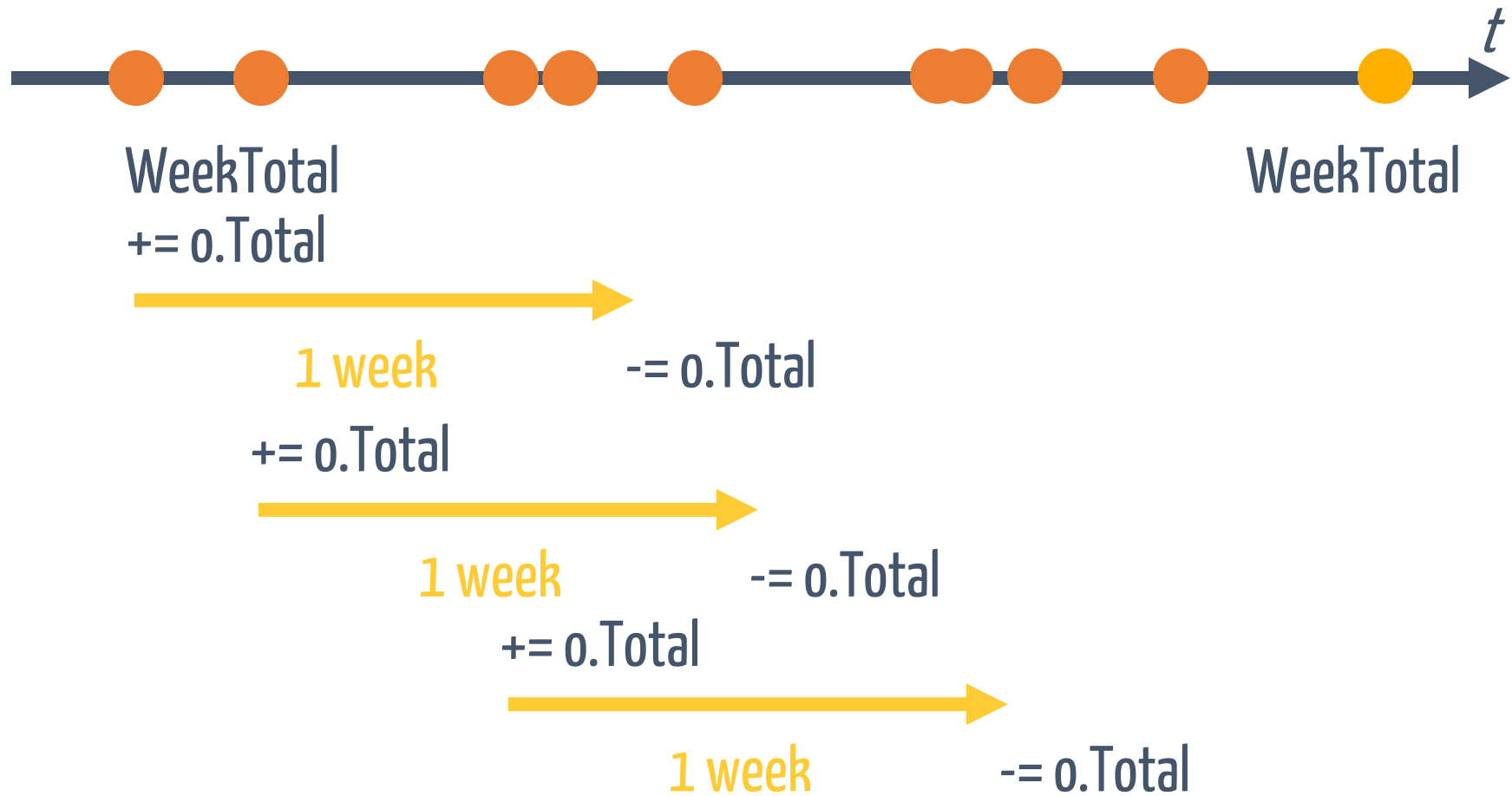
Rethink time



1 week

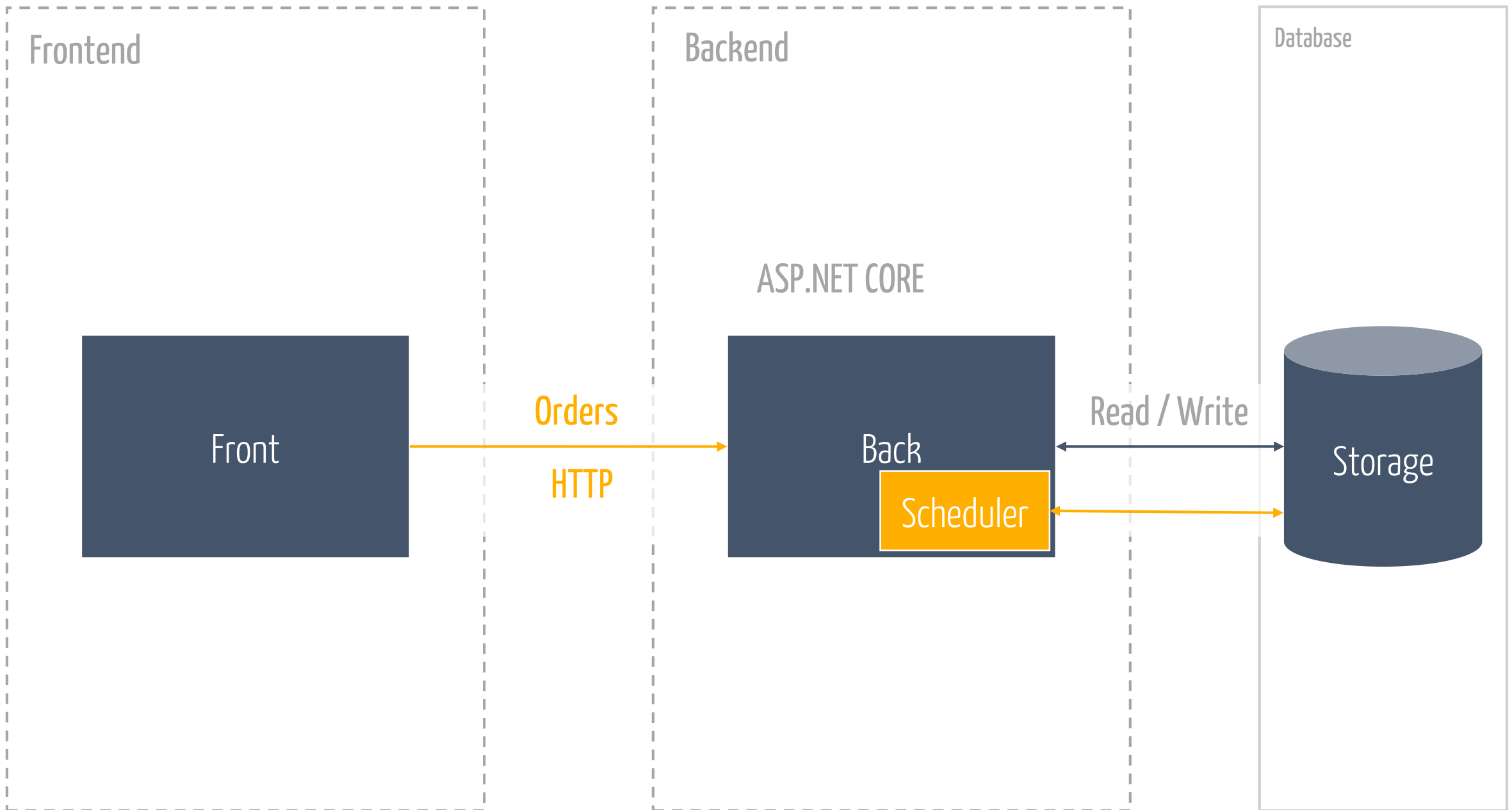
$\text{Sum}(o \Rightarrow o.\text{Total})$





A close-up photograph of a person's hand holding a white smartwatch. The watch screen shows the time 4:42, the location Montréal, and a weather forecast for the next few days. The background is a blurred image of a person's arm and a light blue surface.

Program *time*



Durable scheduling introduces reliability on
the server side

Resilient HTTP introduces some reliability
but doesn't survive restarts

Client side **retries** help to resolve transient failures but **increase the latency**

Orders might be **lost** when clients give up on
retries

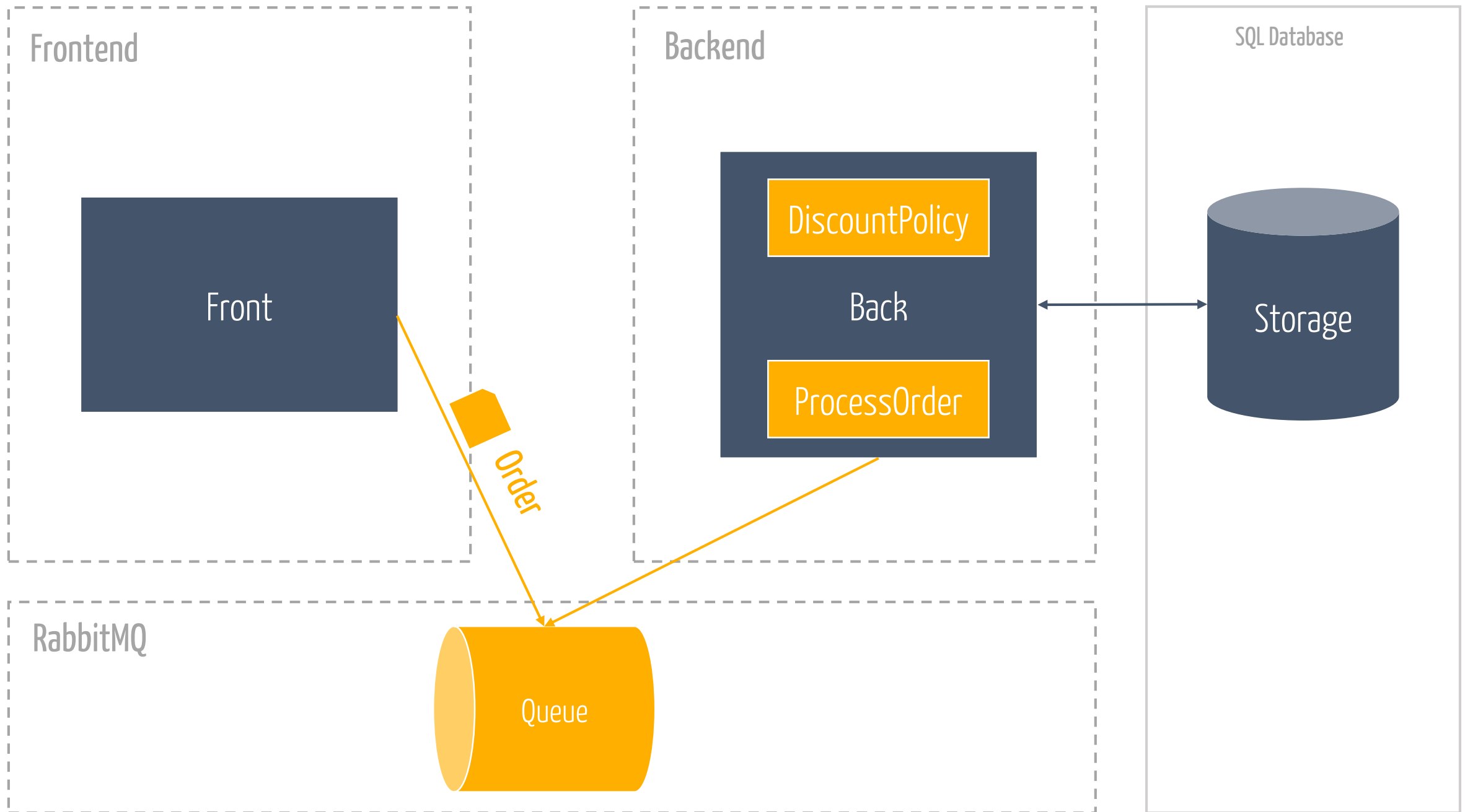
Orders are ideal to

Fire & forget

from the frontend perspective

Sagas

Sagas





*Roads? Where we're going, we don't
need roads.*

AmazonSQS

600 s



2,800 s



900
(1,900)



900
(1,000)



900
(100)



100 s

RabbitMQ

42₁₀ seconds

101010₂ seconds

(42_{10})


32 s  1

16 s  0

8 s  1

4 s  0

2 s  1

1 s  0

Exchange: nsb.delay-level-00

► **Overview**

▼ Bindings

From	Routing key	Arguments	
nsb.delay-level-01	*****.0.#		Unbind



To	Routing key	Arguments
nsb.delay-delivery	*****0.#	Unbind
nsb.delay-level-00	*****1.#	Unbind

Exchange: nsb.delay-level-01

► **Overview**

▼ Bindings

From	Routing key	Arguments	
nsb.delay-level-02	*****.0.#		Unbind

[illegible][illegible]

28 queues

2^{28} seconds

268,435,456 seconds

8.5 years



Sender

Destination-delay.fifo

Destination

`NServiceBus.AmazonSQS.DelaySeconds = delay`

loop

[every 900sec]

all [remaining delay < 900sec]

`NServiceBus.AmazonSQS.DelaySeconds = remaining delay`

[remaining delay <= 900sec]

`DelaySeconds = remaining delay`

docs.particular.net/transports/sqs/delayed-delivery

Exchange: nsb.delay-level-00

► **Overview**

Exchange: nsb.delay-level-01

► **Overview**

docs.particular.net/transport/rabbitmq/delayed-delivery

To	Routing key	Arguments	
nsb.delay-delivery	*****0.#		Unbind
nsb.delay-level-00	*****1.#		Unbind

[illegible][illegible]

Messaging introduces reliability

Retries resolve consistency issues
automatically

Sagas on top of a robust middleware allow to focus on the business logic and stay reactive

For ultra **high contention** domains different approaches might be necessary

Ask the collaborative domain question first

Business consistency rarely ever needs to be
addressed with technical solutions

Favor **simplicity** over complex design
wherever you can

Thanks



go.particular.net/dnd19-parsec





Long Running Processes

Get free access to a section of the Advanced Distributed Systems Design course

Enroll for free

go.particular.net/dnd19-knock-knock

Slides, Links...

github.com/danielmarbach/KnockKnock

Q & A



Software Engineer
Enthusiastic Software Engineer
Microsoft MVP

@danielmarbach
particular.net/blog
planetgeek.ch

Thanks