Particular Software

Knock. Knock. Who's there?

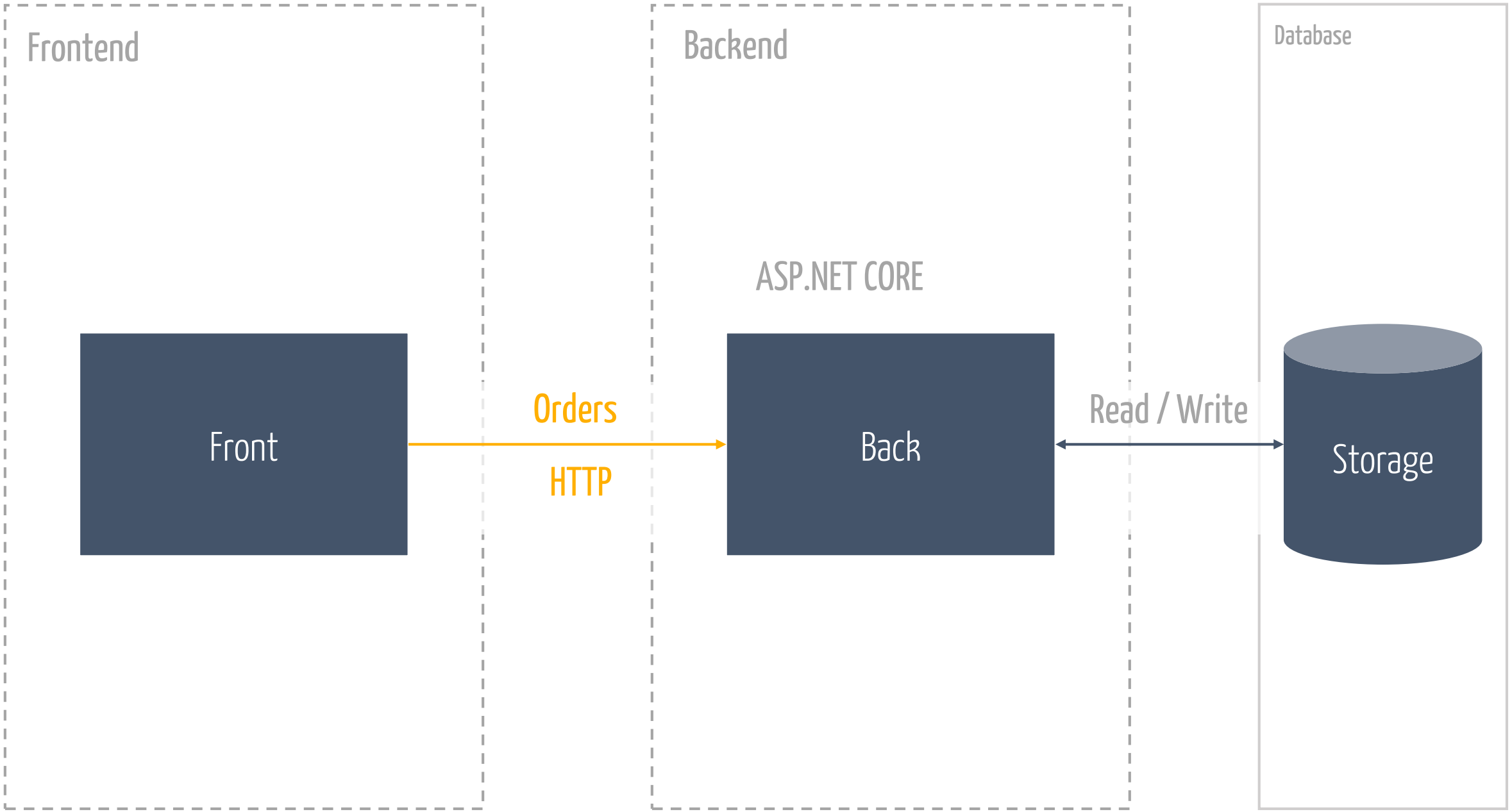
A message from the future


EXIT

Hiding Inconsistencies

Multi user collaboration

Multi user collaboration

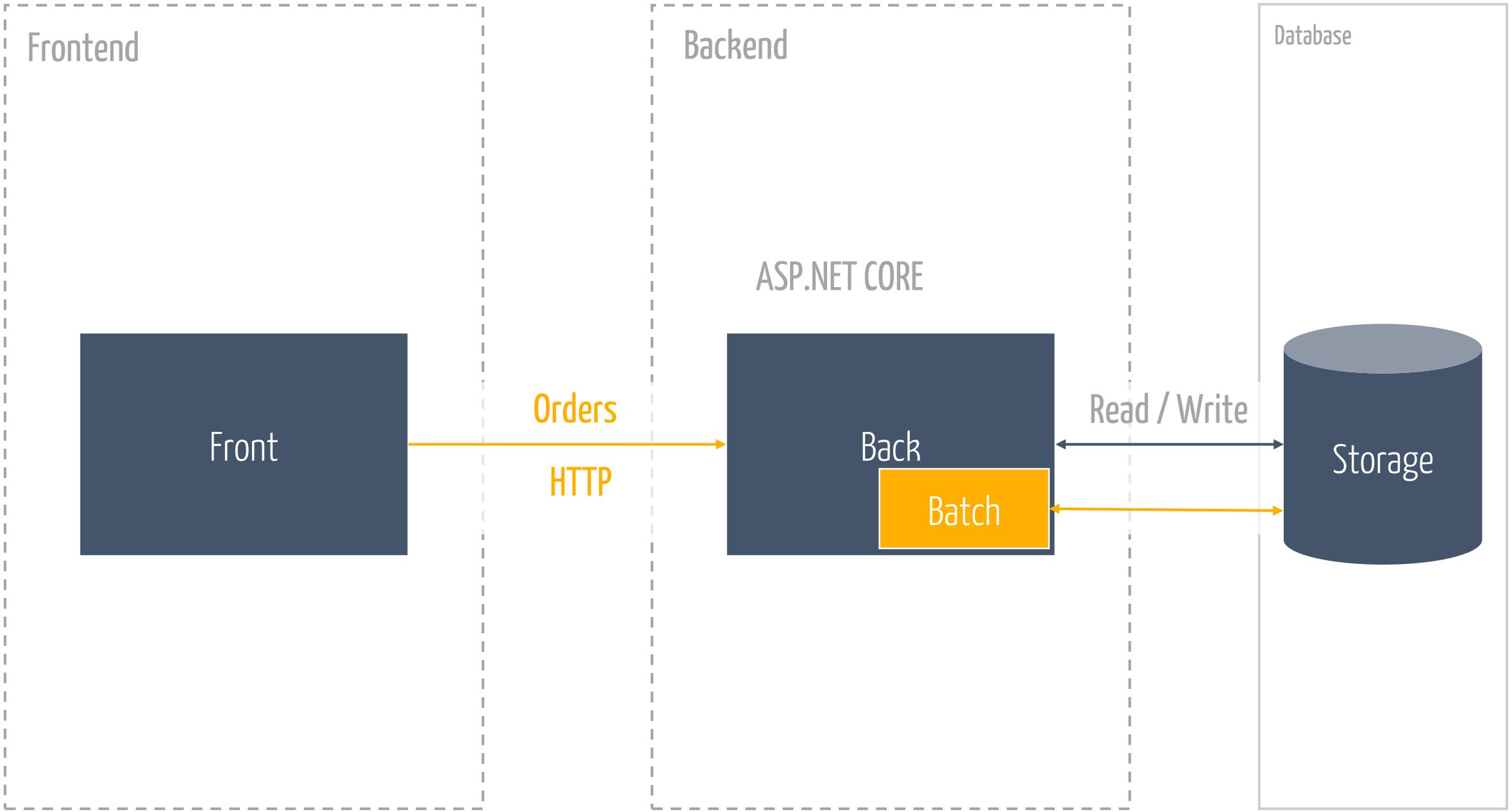

Frontend
Backend
Database
ASP.NET CORE
Front
Orders
HTTP
Back
Read / Write
Storage

And then the customer walks into your office


4GIFs.com

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

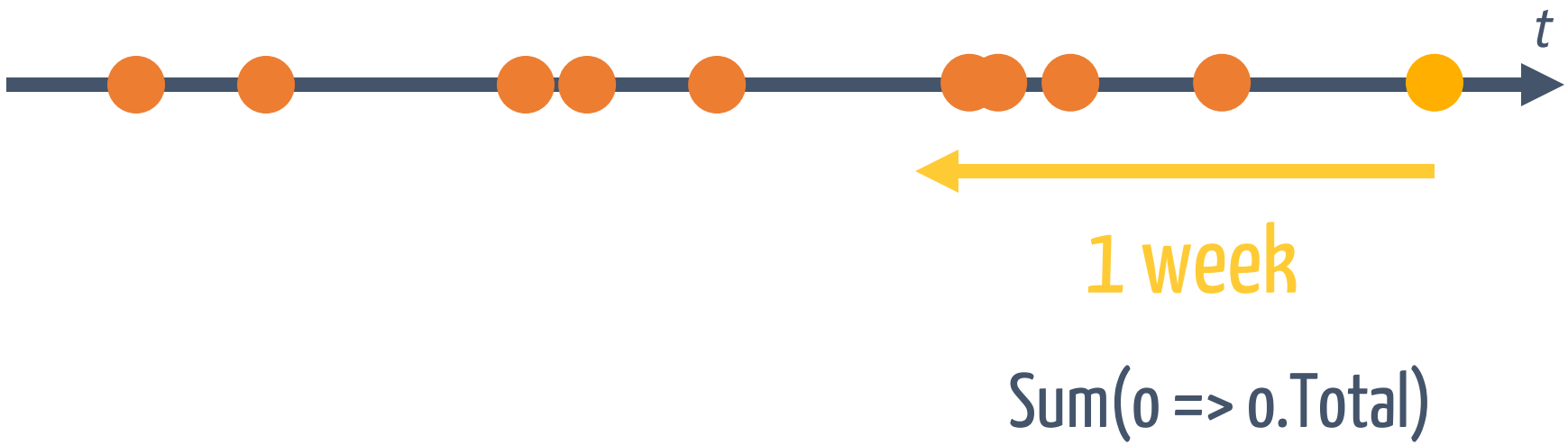
Batch Jobs

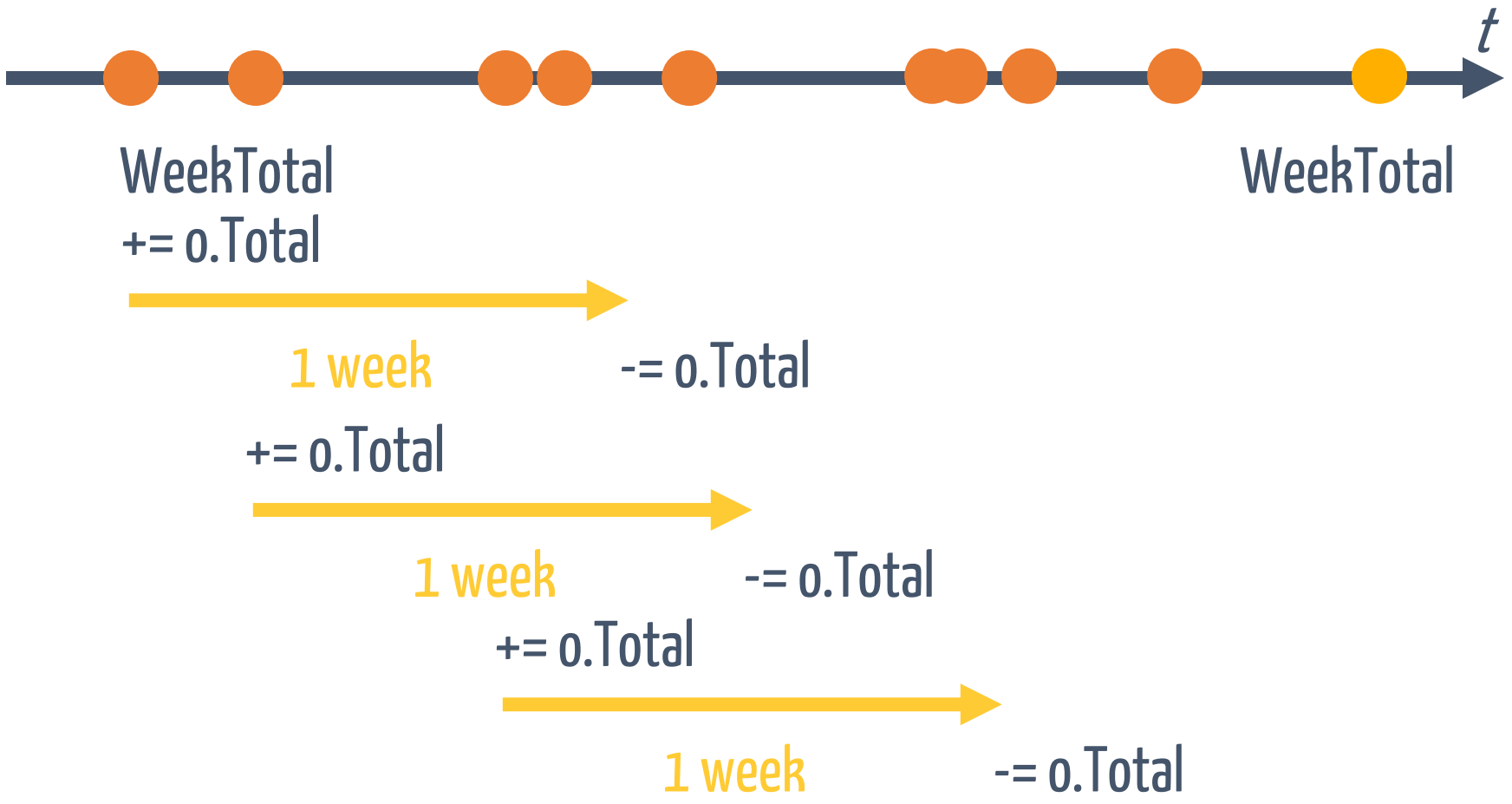

Frontend
Backend
Database
ASP.NET CORE
Front
Orders
HTTP
Back
Batch
Read / Write
Storage

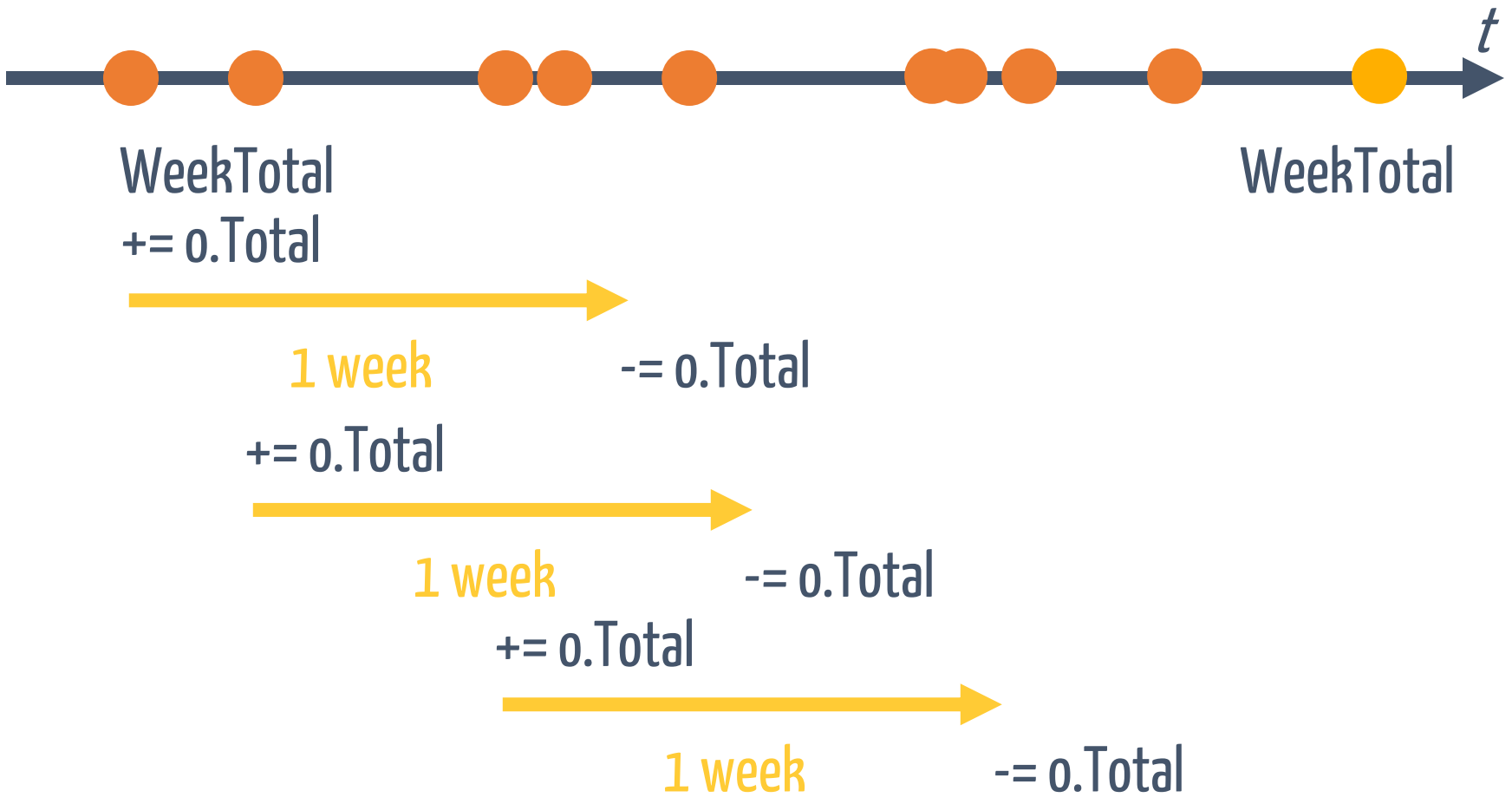
Batch Jobs increase the window of consistency problems

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

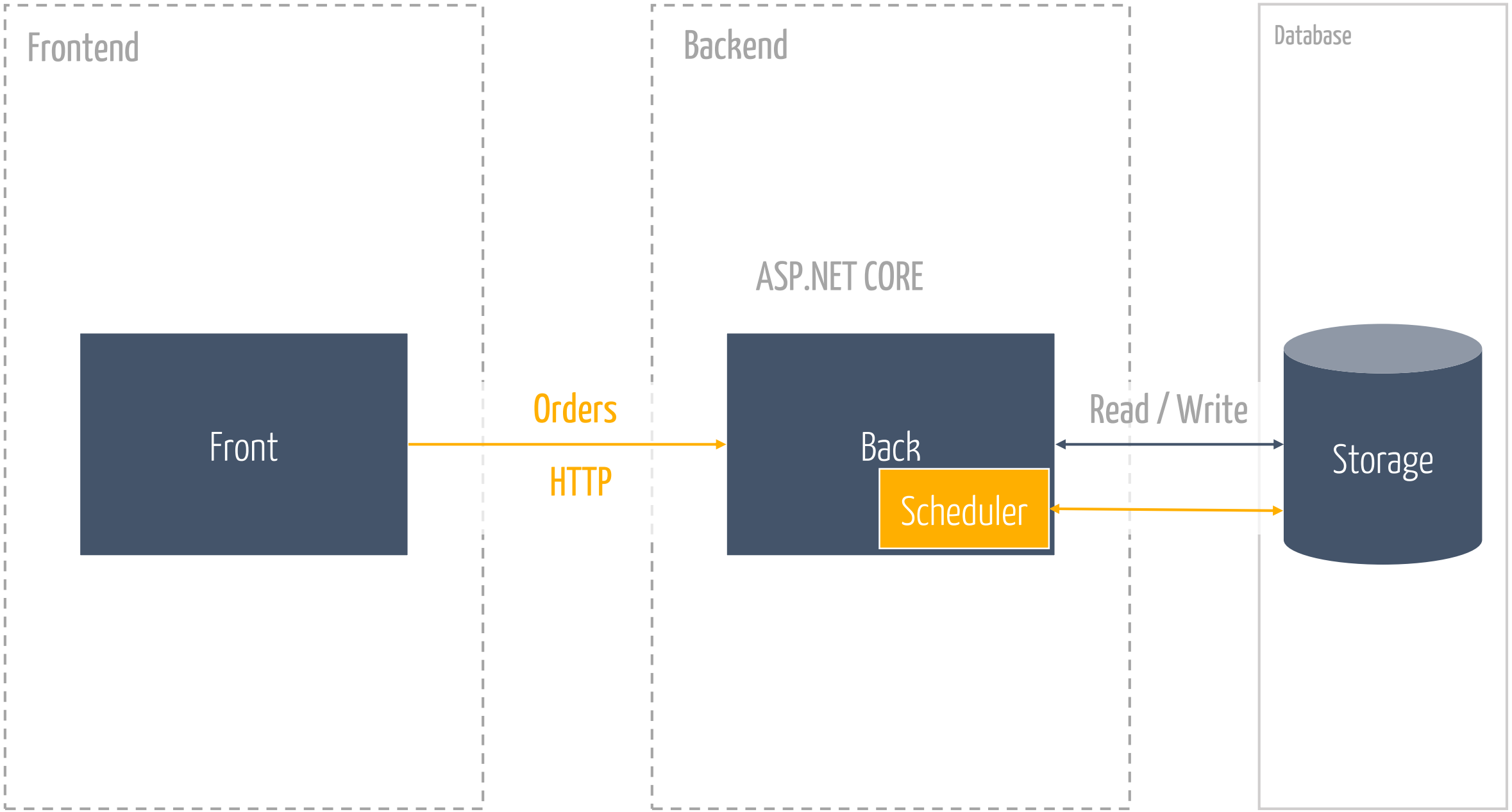

Rethink time


t
1 week
Sum(o => o.Total)


t
WeekTotal
+= o.Total
WeekTotal
1 week
-= o.Total
+= o.Total
1 week
-= o.Total
+= o.Total
1 week
-= o.Total


t
WeekTotal
+= o.Total
WeekTotal
1 week
-= o.Total
+= o.Total
1 week
-= o.Total
+= o.Total
1 week
-= o.Total

Program time


Frontend
Backend
Database
ASP.NET CORE
Front
Orders
HTTP
Back
Scheduler
Read / Write
Storage

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

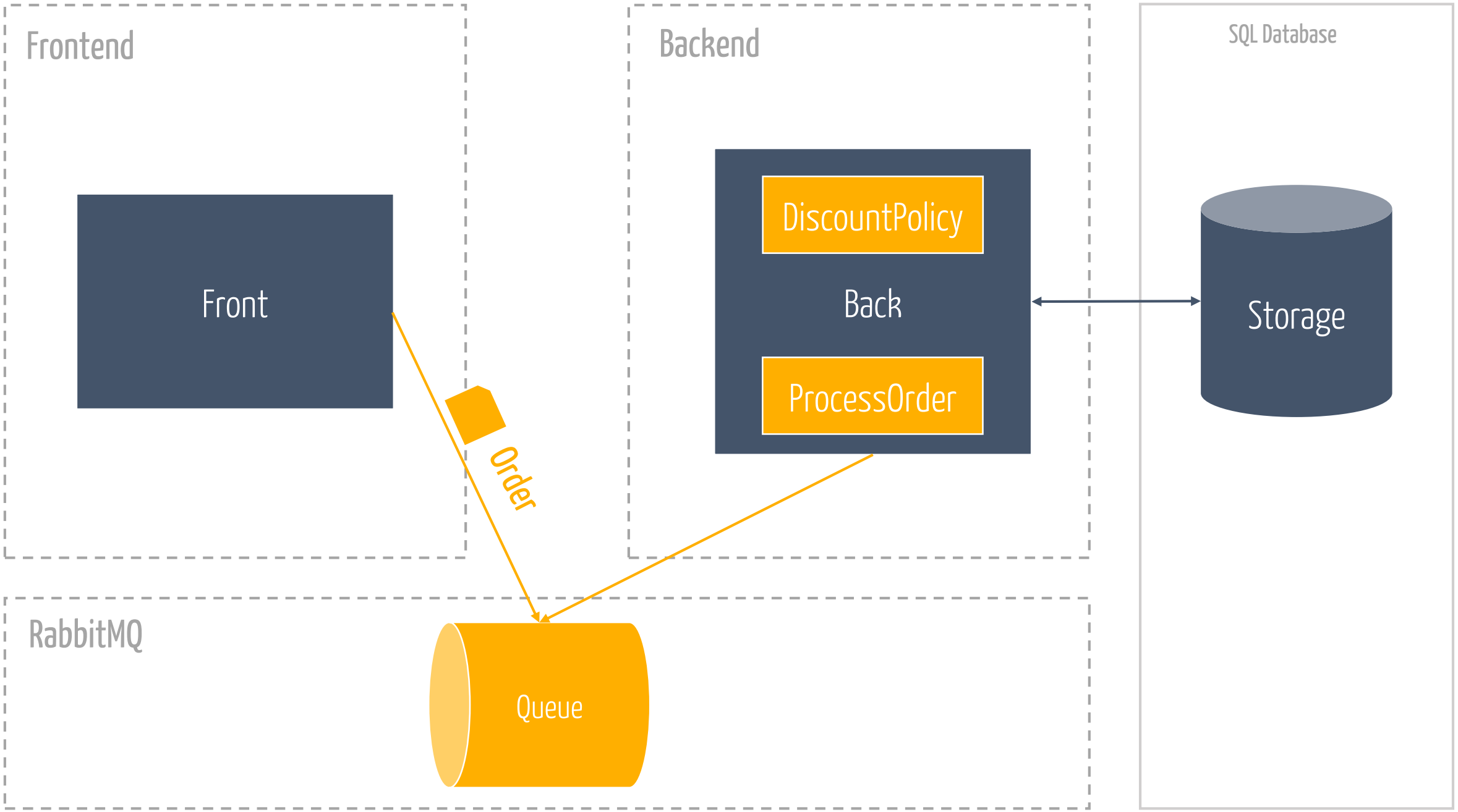
Orders are ideal to

Fire & forget

from the frontend perspective

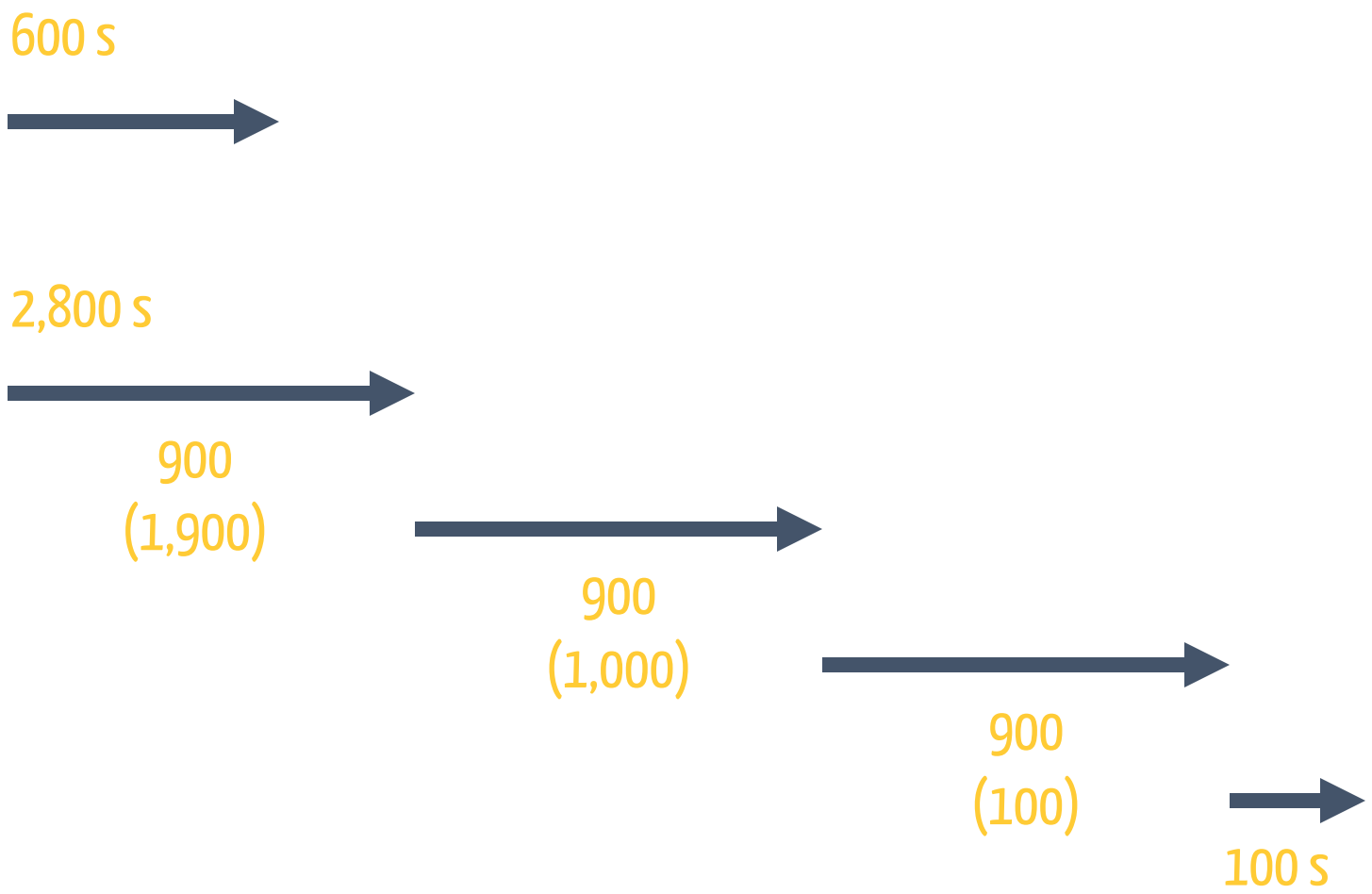
Sagas

Sagas


Frontend
Front
Order
Backend
DiscountPolicy
Back
ProcessOrder
SQL Database
Storage
RabbitMQ
Queue


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

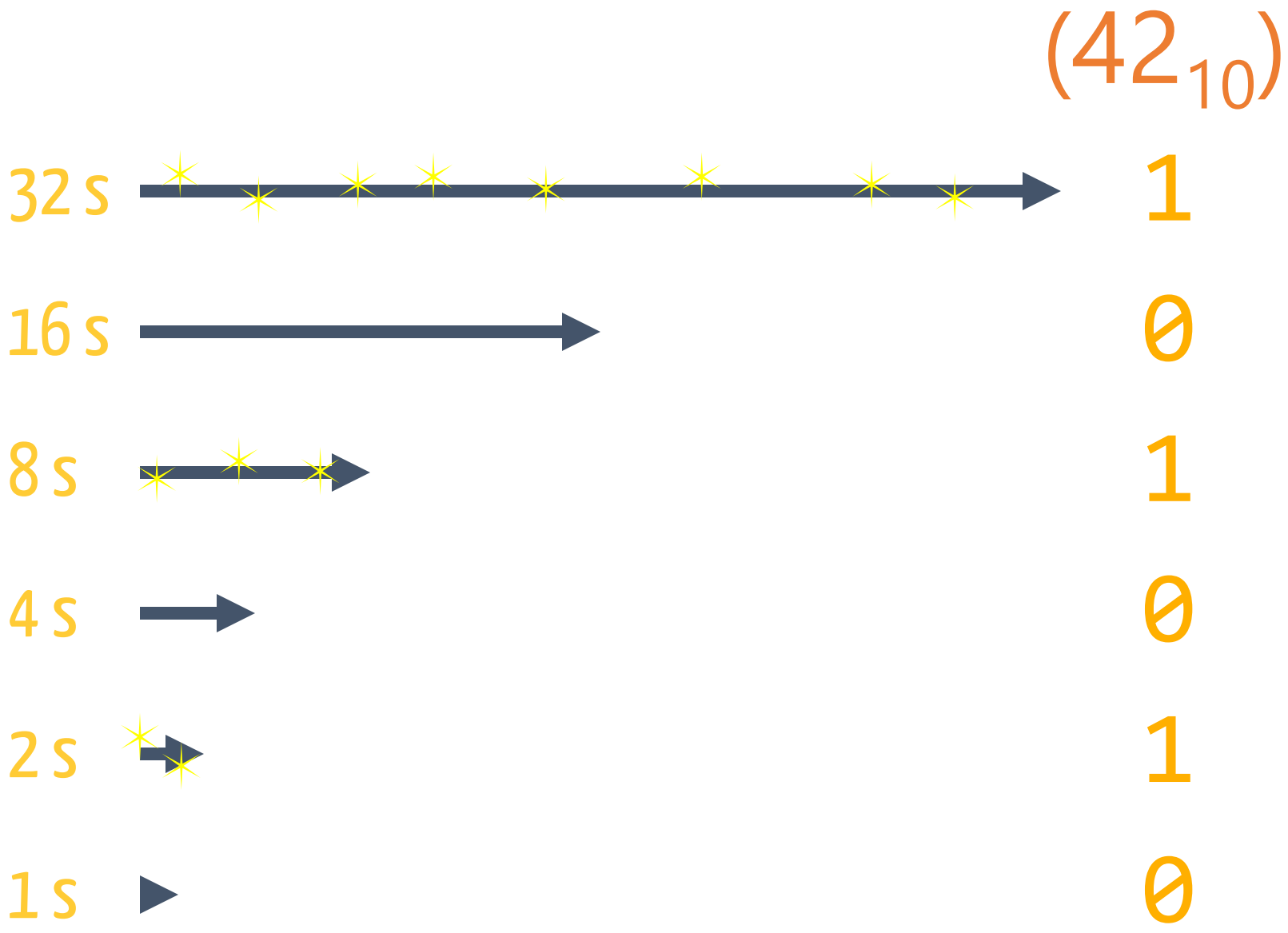
Amazon SQS

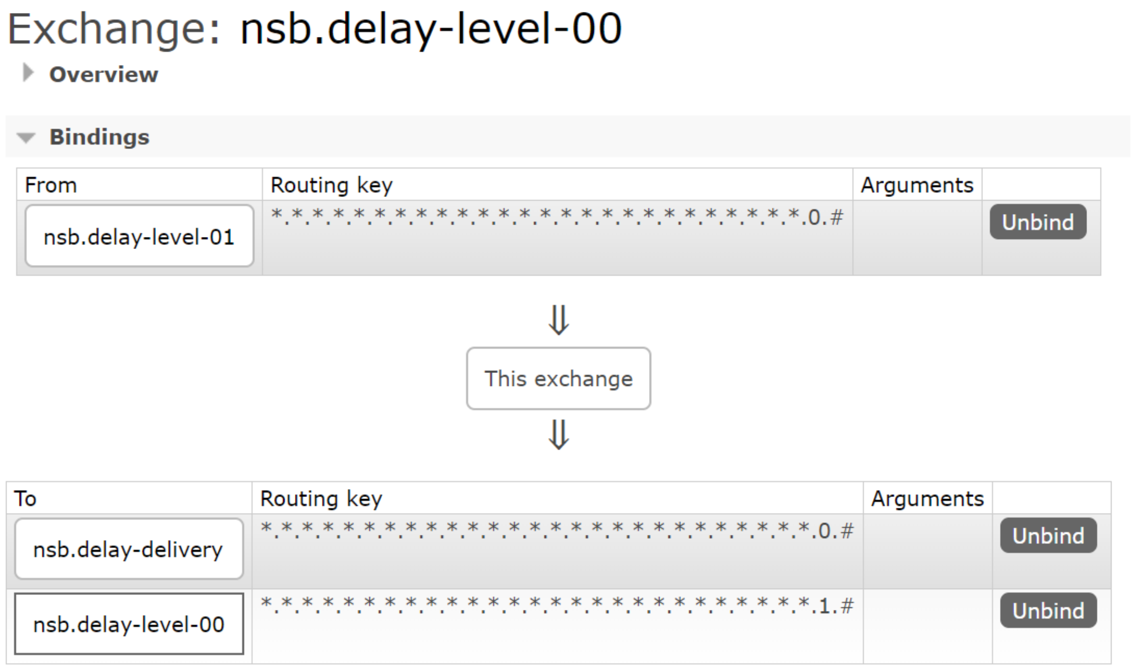

600 s
2,800 s
900
(1,900)
900
(1,000)
900
(100)
100 s

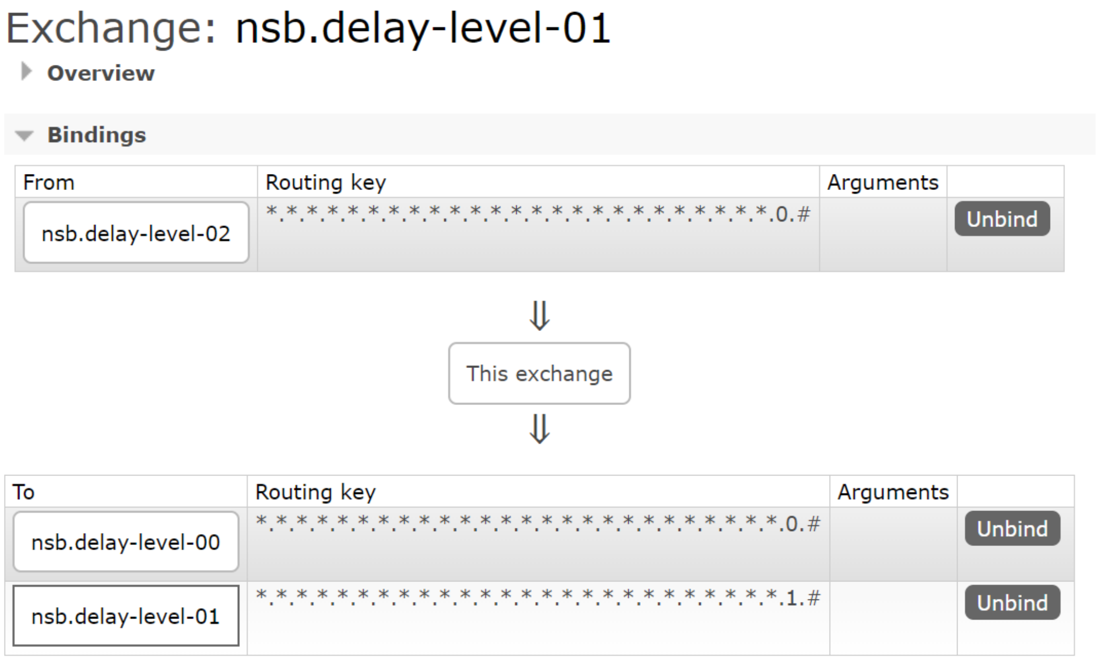
RabbitMQ

42_{10} seconds

101010_{2} seconds


(42_{10})
32 s
1
16 s
0
8 s
1
4 s
0
2 s
1
1 s
0





N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.destination

28 queues

2^{28} seconds

268,435,456 seconds

8.5 years

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

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

N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.N.destination

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

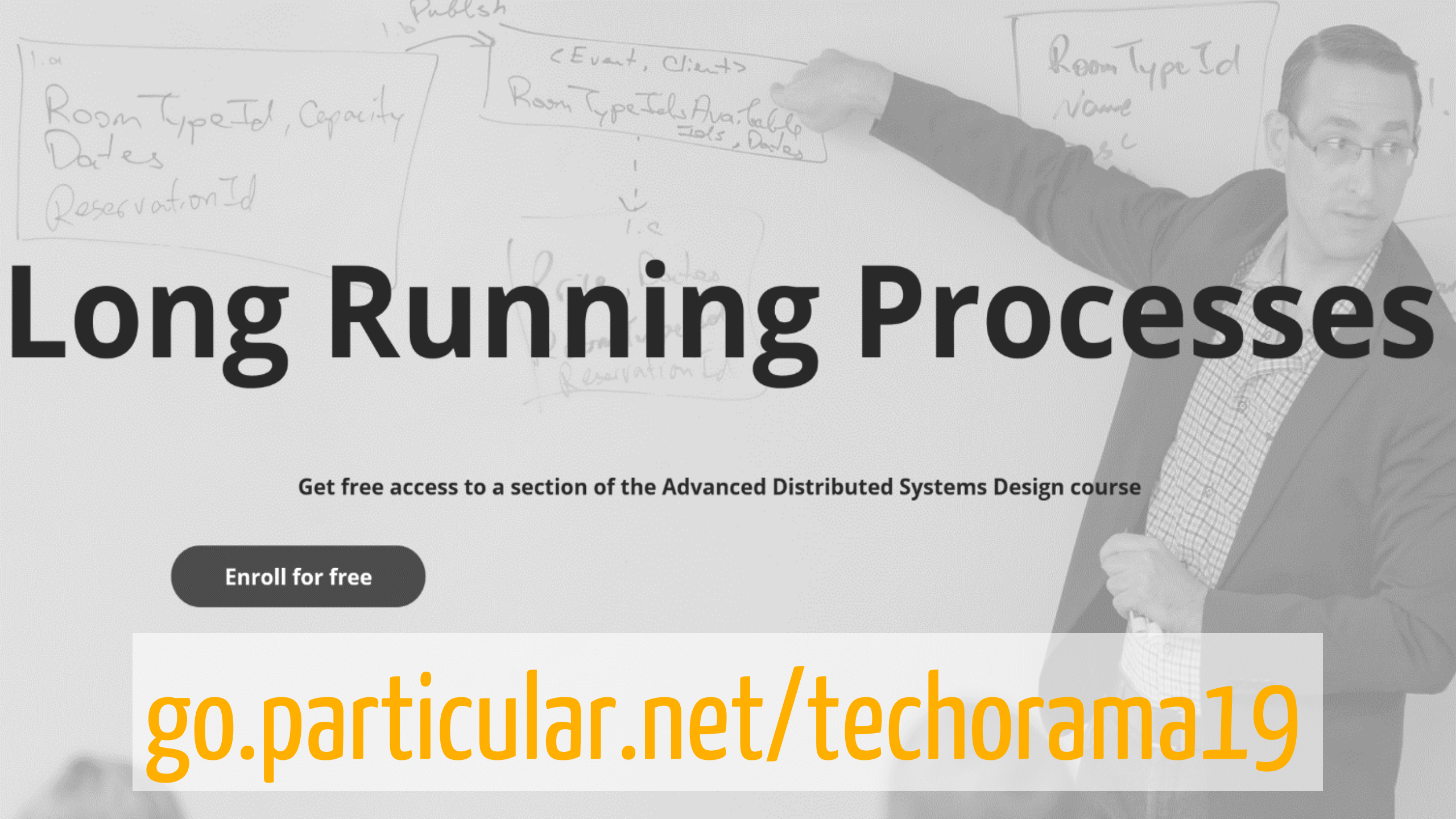
Ask the collaborative domain question first

Business consistency rarely ever needs to be addressed with technical solutions

Favor simplicity over complex design wherever you can

Particular
Software

Thanks



Long Running Processes

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

Enroll for free

go.particular.net/techorama19

Slides, Links...

github.com/danielmarbach/KnockKnock

Particular
Software

Q&A

Particular Software

Software Engineer
Enthusiastic Software Engineer
Microsoft MVP

@danielmarbach
particular.net/blog
planetgeek.ch

Particular
Software

Thanks