

JUST-IN-TIME ARCHITECTURE

Macklin Hartley

<https://macklin.me/>





- ** A Software Development Journey
- ** Event-driven Architecture
- ** Communication Patterns



A SOFTWARE DEVELOPMENT JOURNEY



Balance

550,000,000 Coins

Avatar Store



Gorilla

5000 Available

Claim for 500 Coins



Snake

50 Available

Claim for 50,000 Coins



Palm [ULTRA RARE]

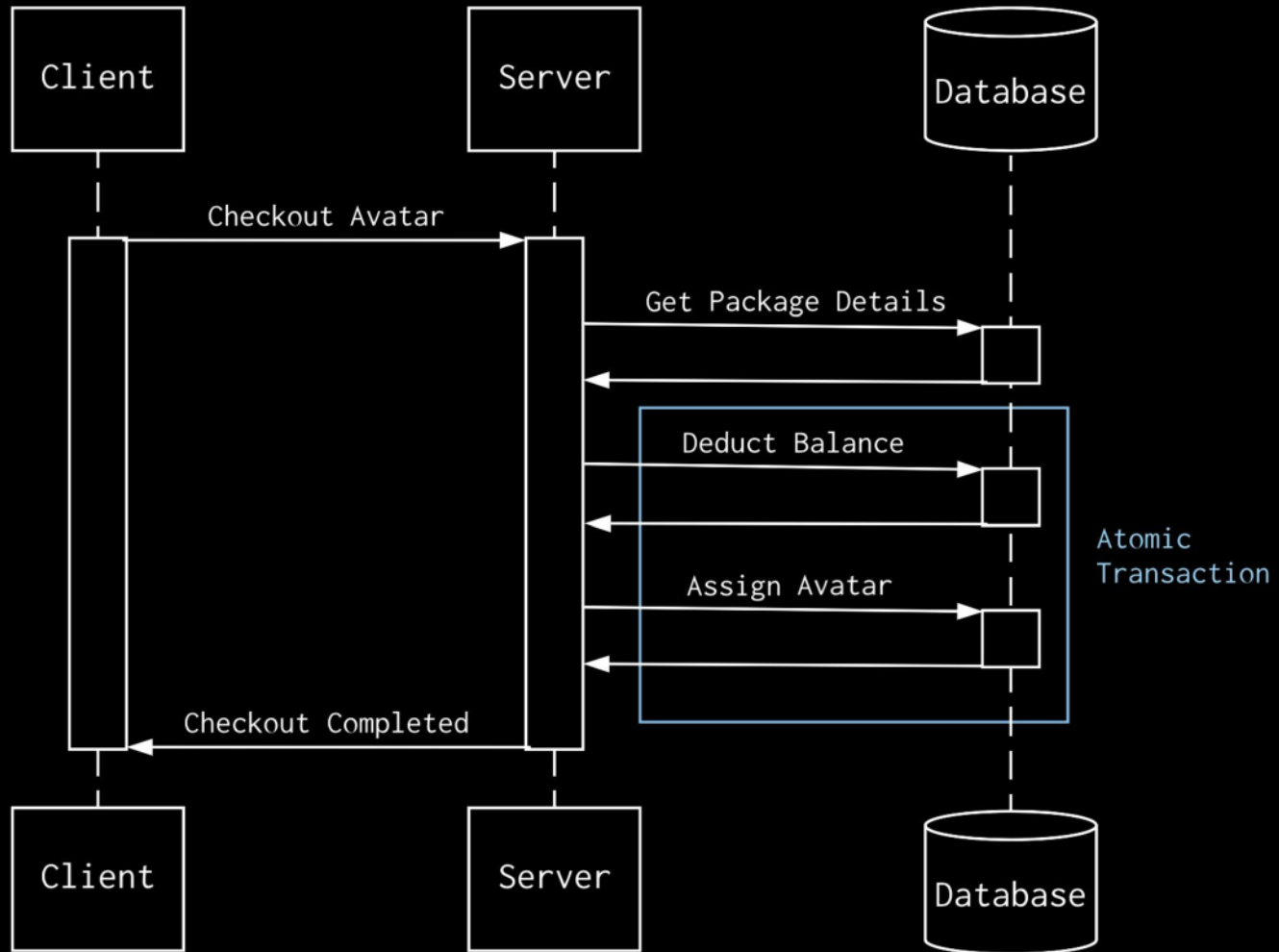
1 Available

Claim for 500,000,000 Coins

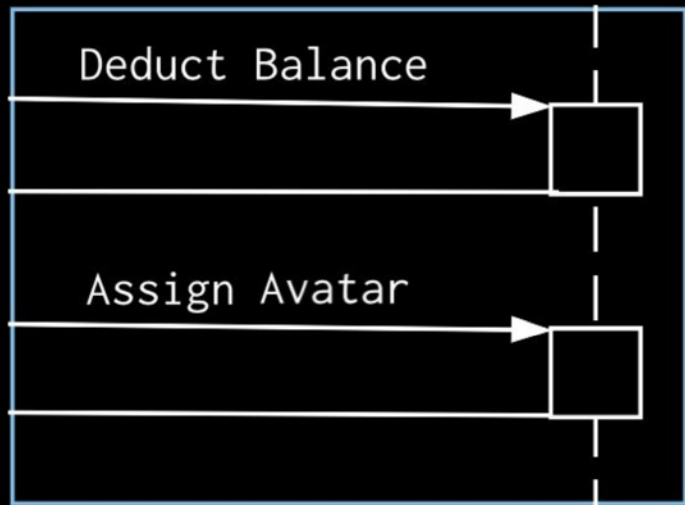


- ** Validate the Checkout Request
- ** Deduct from Balance
- ** Assign Avatar to Inventory





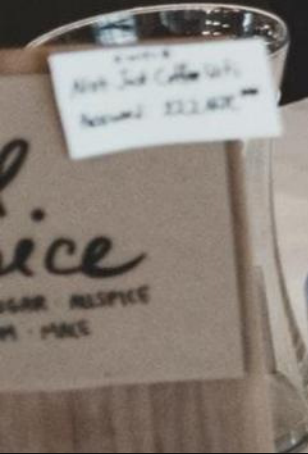
WHAT IS AN ATOMIC TRANSACTION?



Atomic
Transaction

A series of operations such that either all or neither occurs.





Small text on a sign: "New Just Coffee Co. Seasonal \$12.00"

Big Spoon Roasters
Crafted Not Bottled

Fall Spice

AS THE TEAM GREW...



- ** Single shared codebase
- ** Update the entire stack at once
- ** Infrequent, costly releases



- ++ “Just Enough” to Validate Hypothesis
- ++ Atomic Transactions
- Poor Development Experience





NEW CAFE WORKFLOW



Adam

Takes orders



Sarah

Pulls shots



James

Foam + pour milk

Icons from <https://icons8.com>



Microservices!



MICROSERVICES



Store



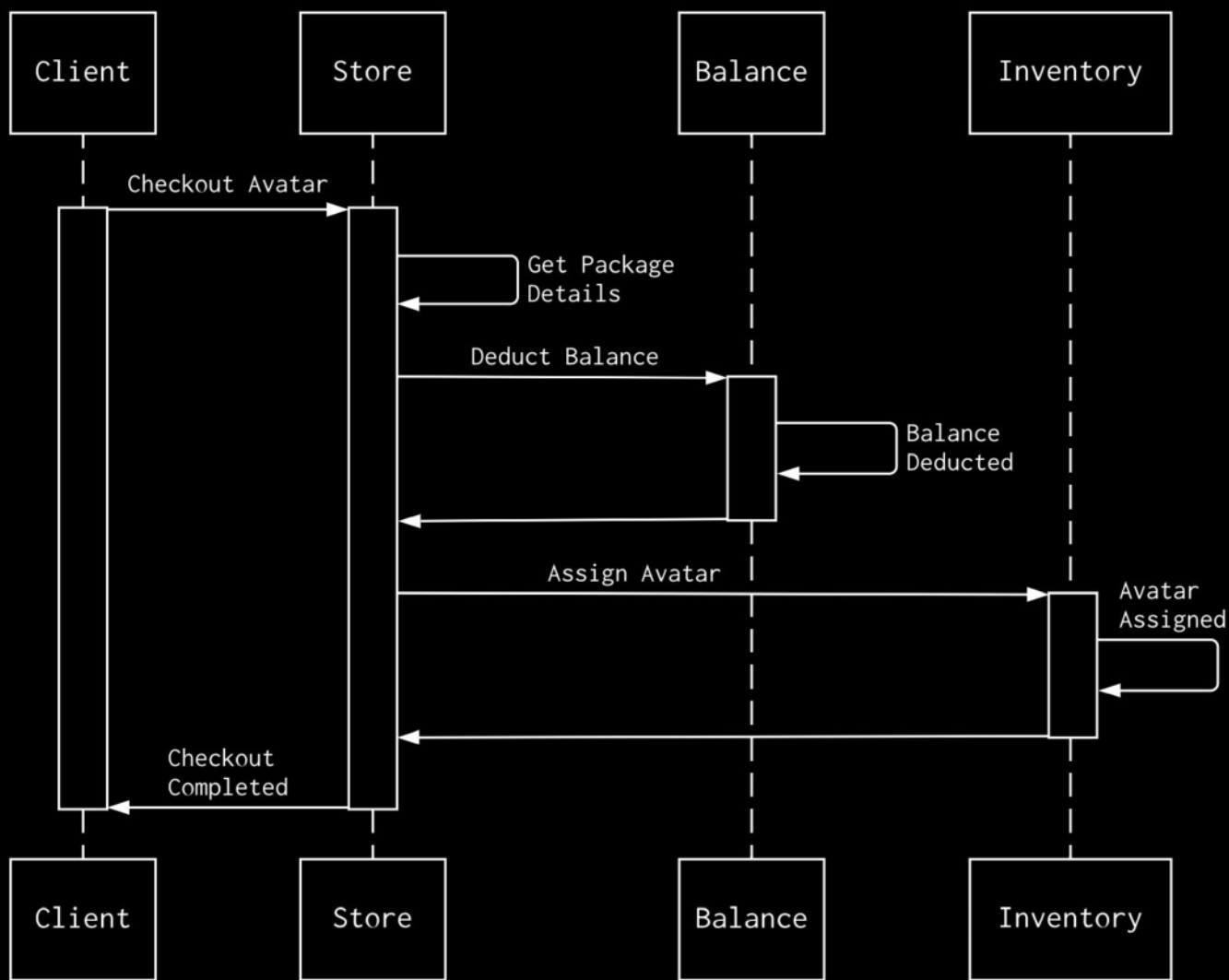
Balance



Inventory

Icons from <https://icons8.com>





SOMETHING ISN'T RIGHT



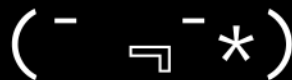
Upset Customers

Customers are not receiving items they have purchased.



Sad Customer Support

Thousands of tickets and complaints every day.



Frustrated Engineers

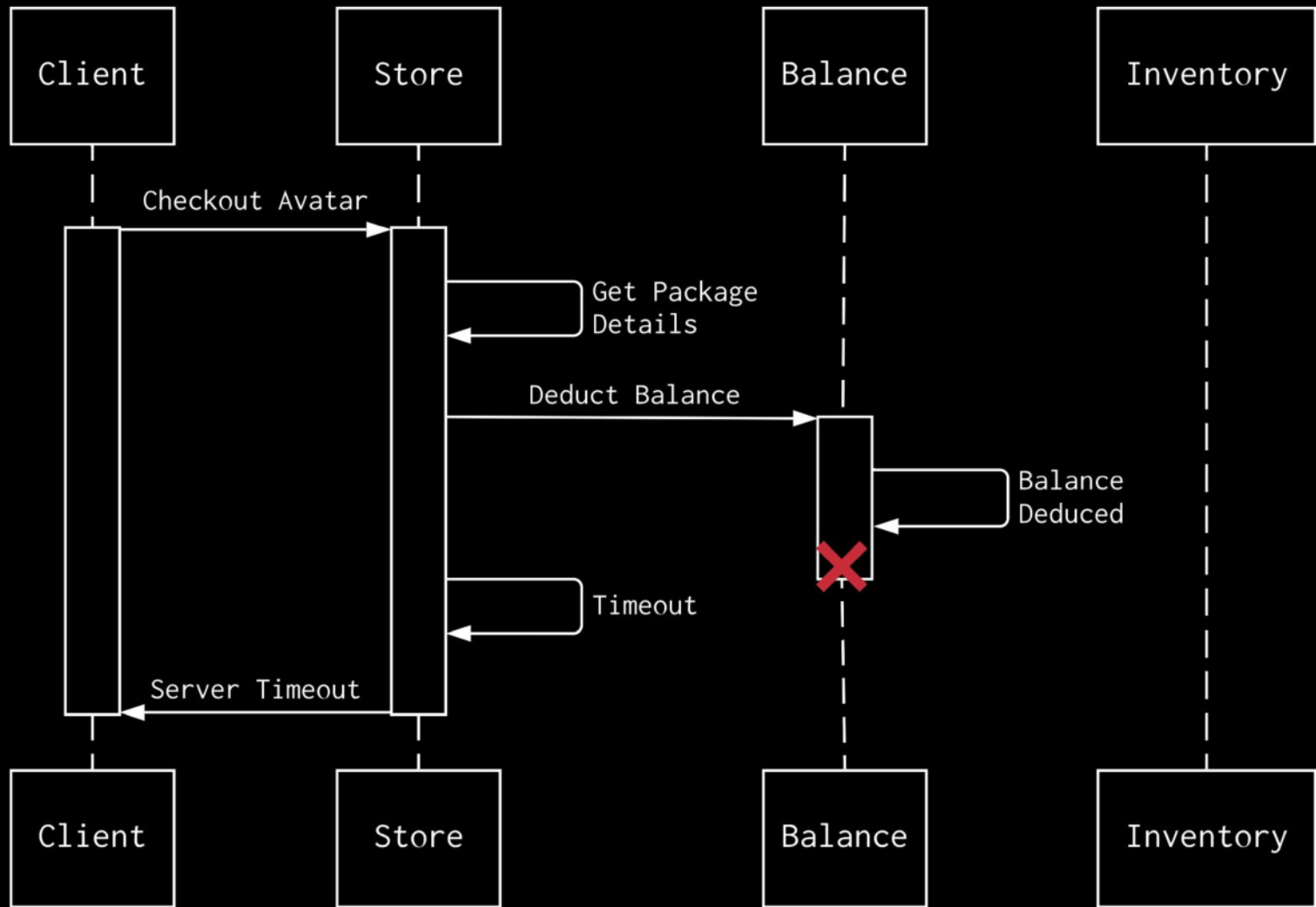
Investigating logs to manually compensate customers.

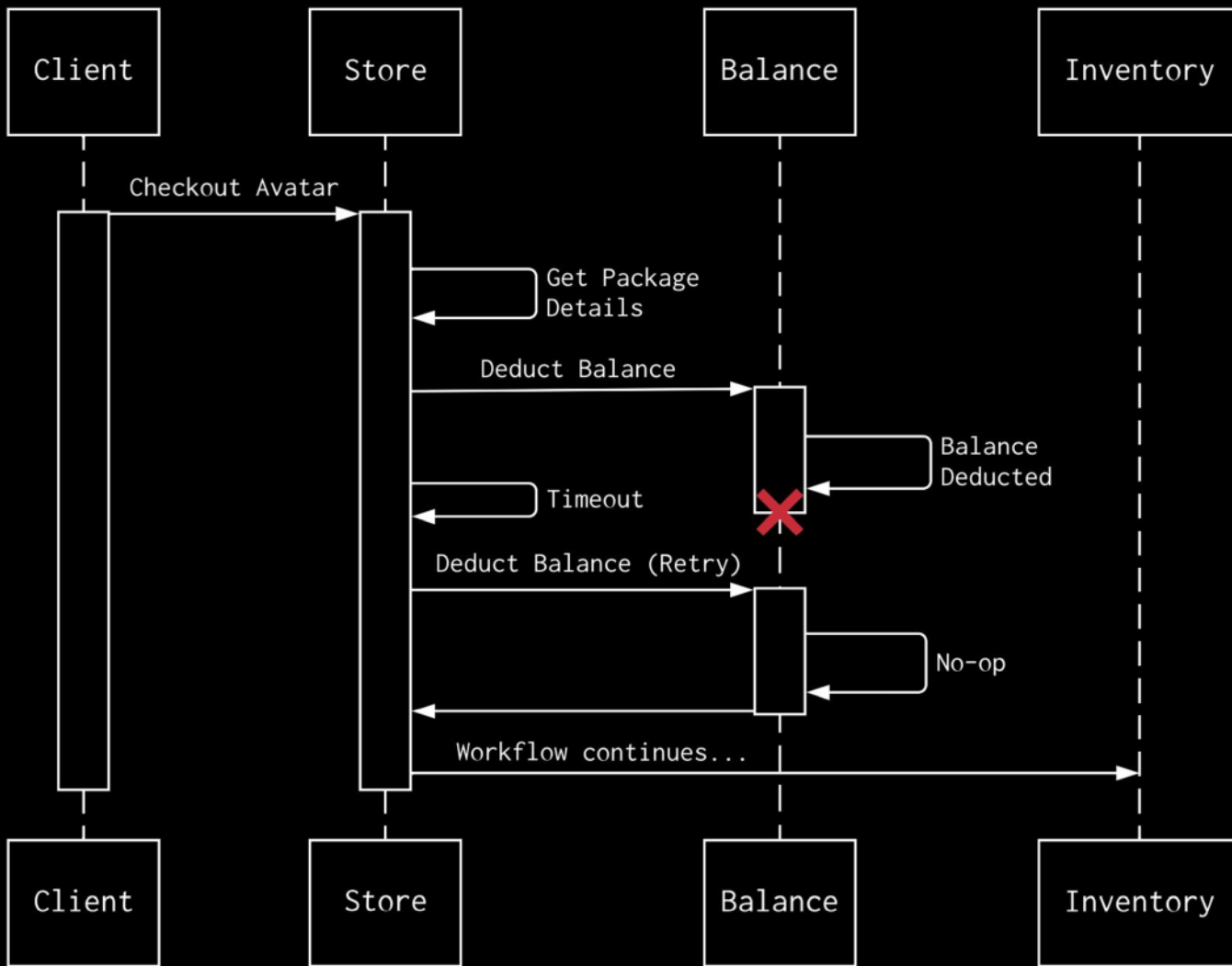


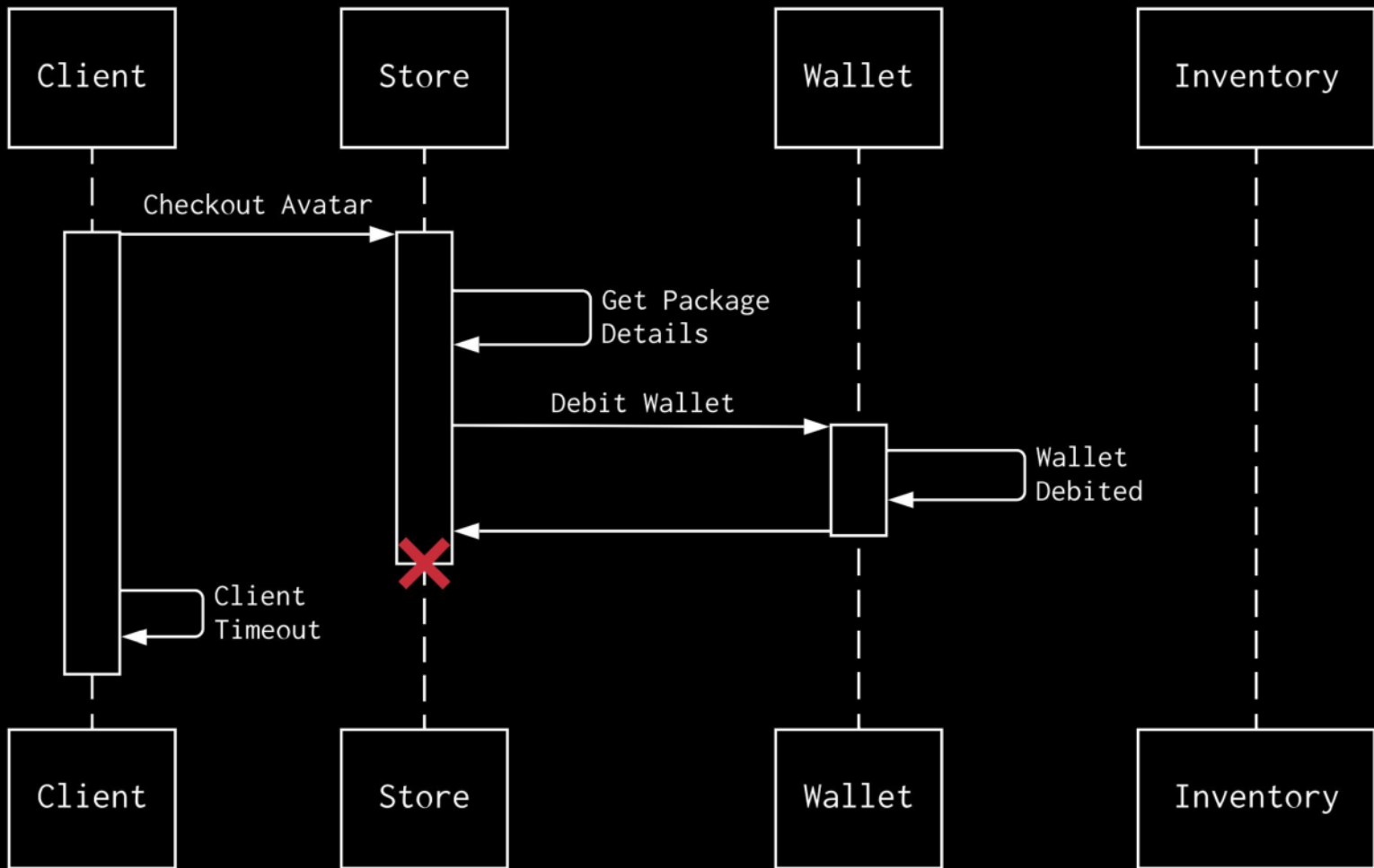
Nervous Stakeholders

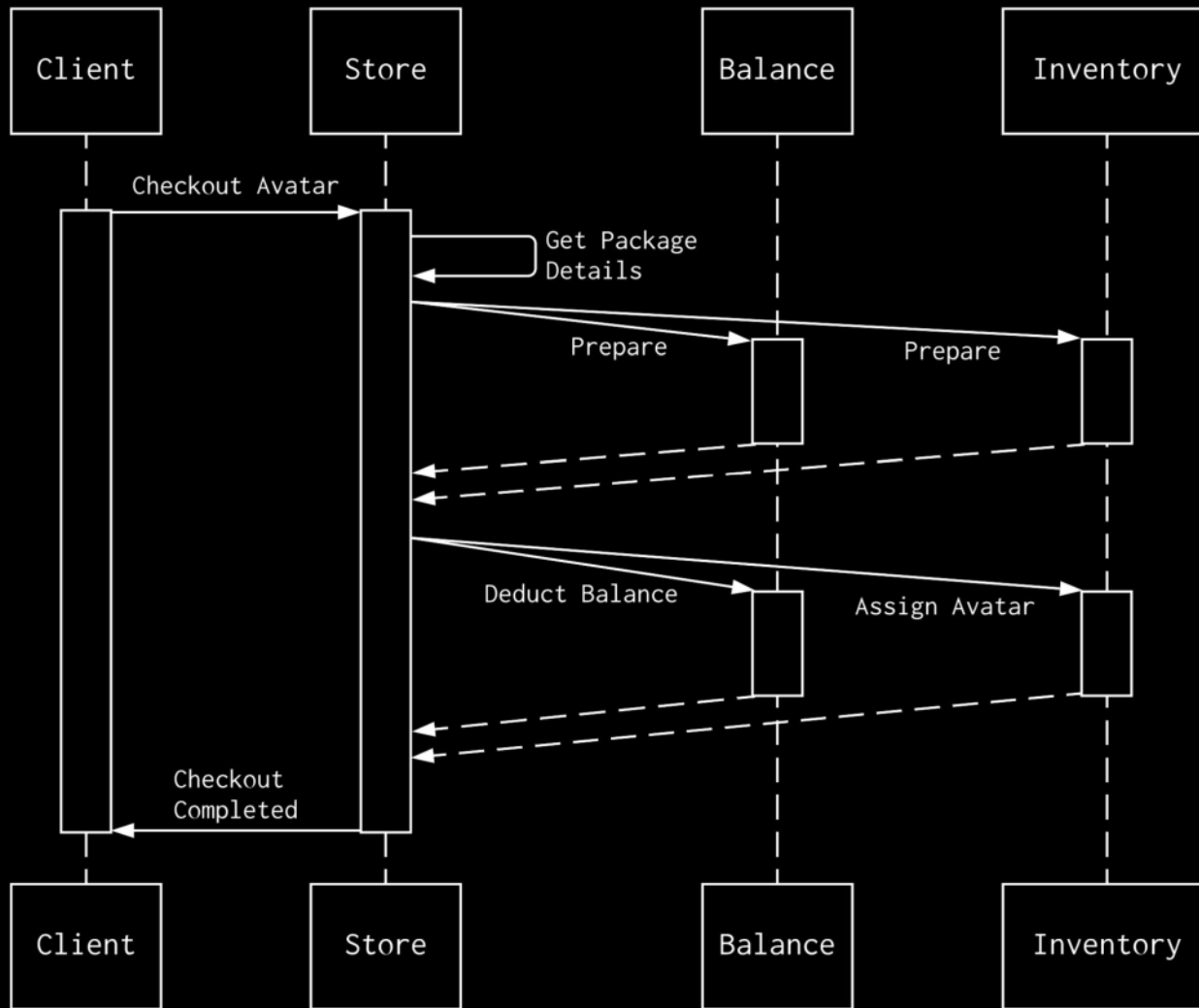
Revenue declining due to loss of trust.











AVAILABILITY COUPLING



Store

99%



Balance

99%



Inventory

99%

Combined Availability

97%

Icons from <https://icons8.com>





- ++ Isolated Boundaries
- ++ Individually Deployable
- Unreliable Workflows
- Availability Coupling



EVENT-DRIVEN ARCHITECTURE

- ** Represents a Change in State
- ** An Immutable Fact
- ** Happened in the Past



“An event-driven architecture uses events to trigger and communicate between decoupled services.”

<https://aws.amazon.com/event-driven-architecture/>

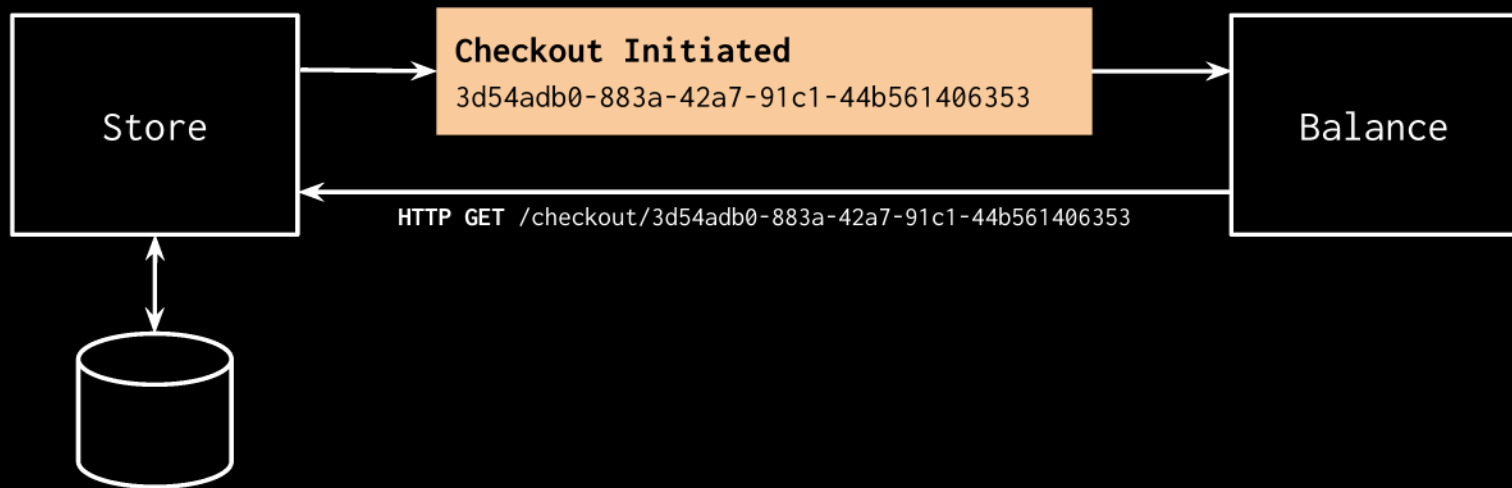


** Event Notification

** Event-carried State Transfer



EVENT NOTIFICATION



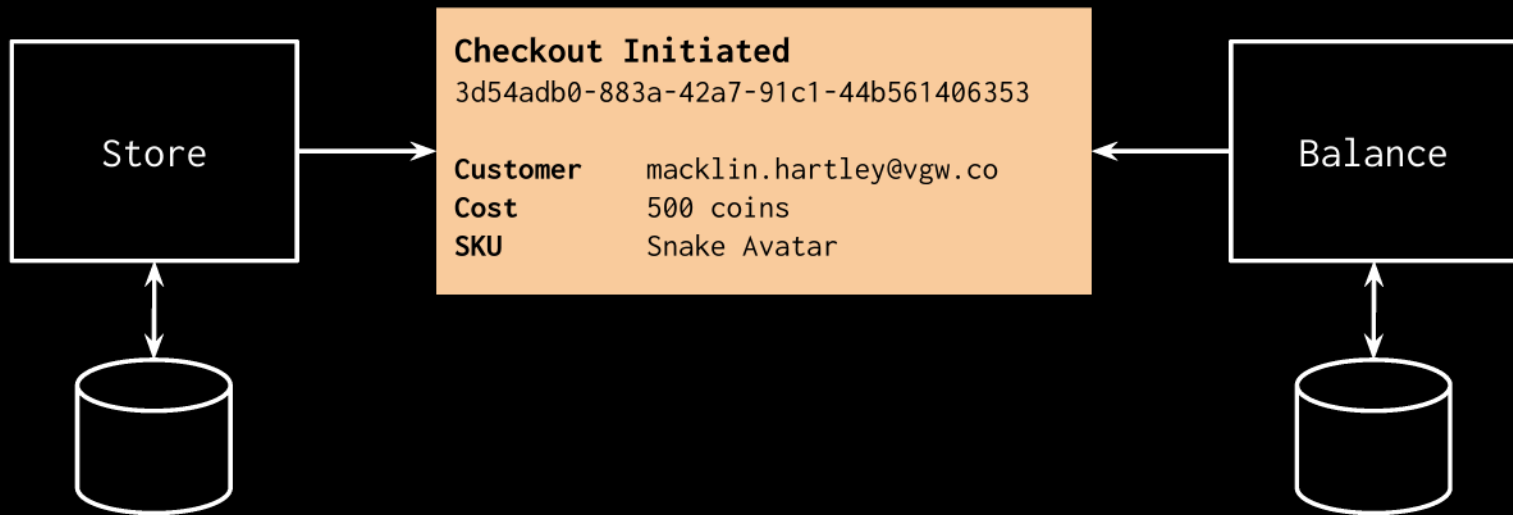
OUTPOST
COFFEE



- ++ Single Source of Truth
- Requests for Additional Information



EVENT-CARRIED STATE TRANSFER

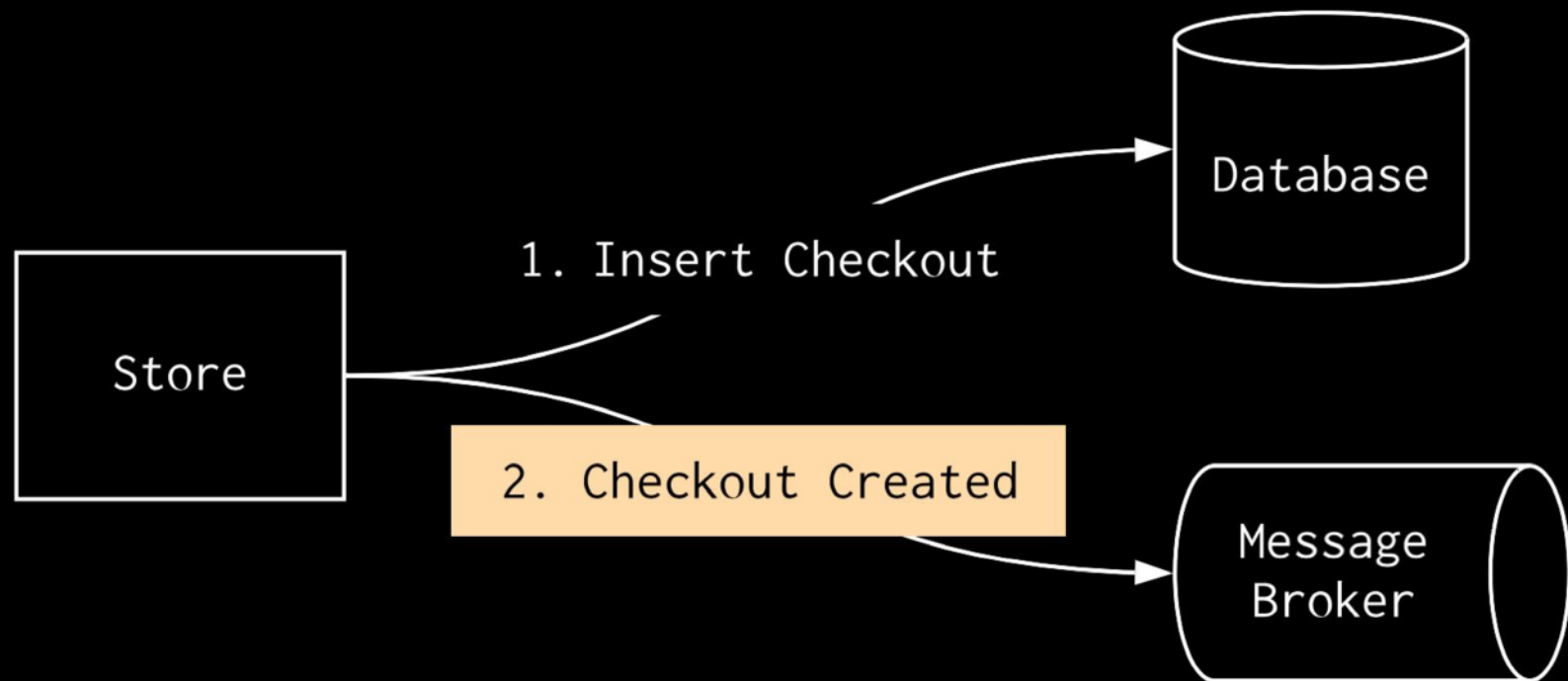




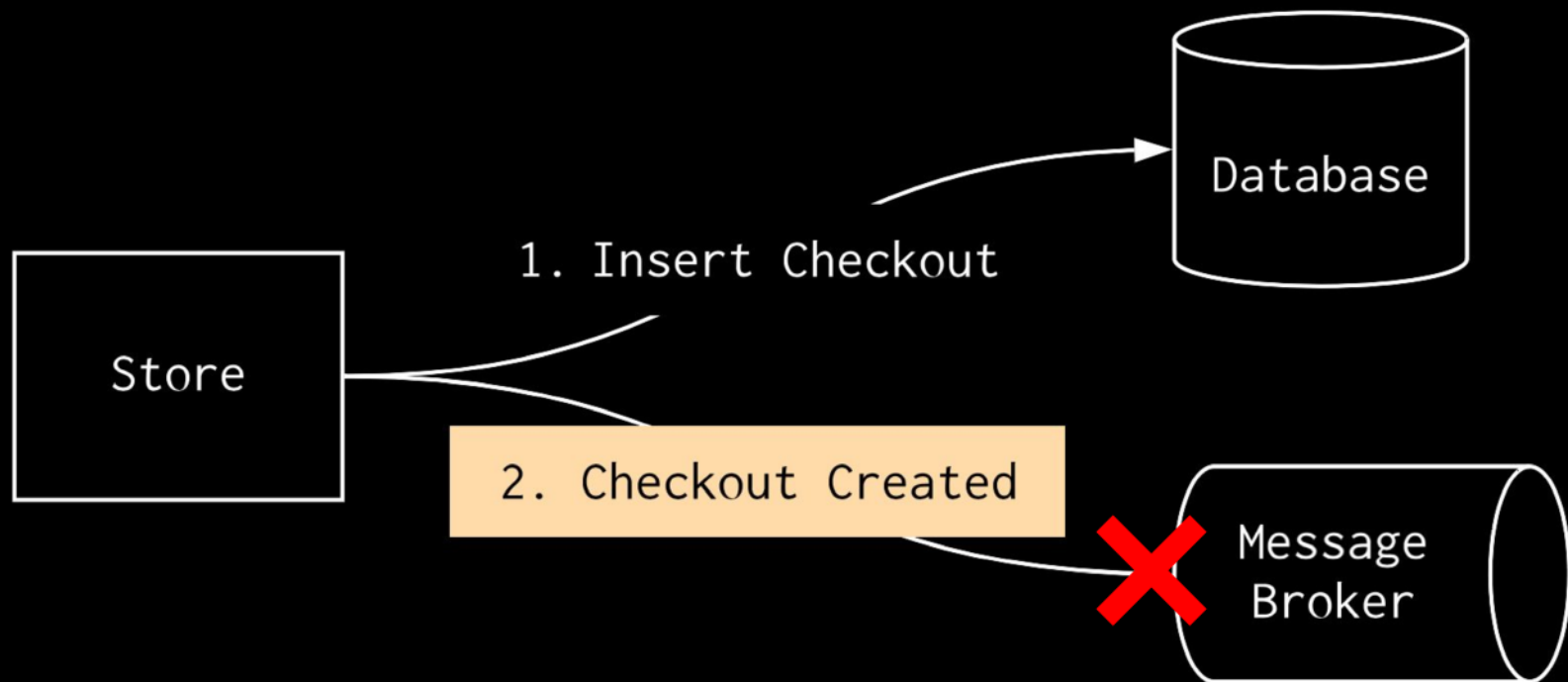
- ++ No Requests for Additional Information
- Events Contain Stale Information
- Eventually Consistent



HOW TO PRODUCE AN EVENT?



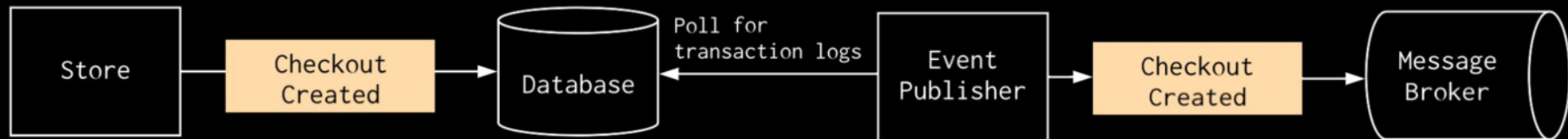
HOW TO PRODUCE AN EVENT?



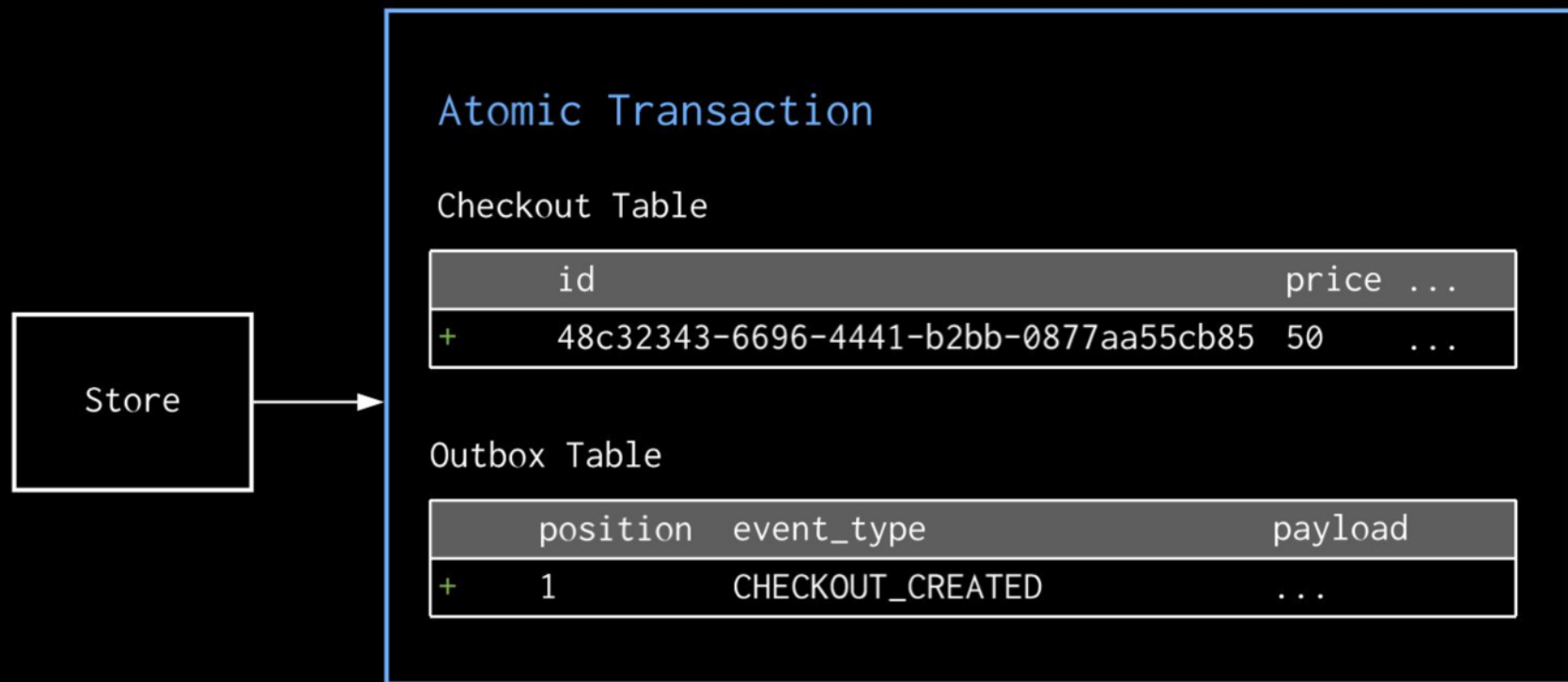
- ** Transaction Logs
- ** Transactional Outbox Pattern
- ** Event Sourcing



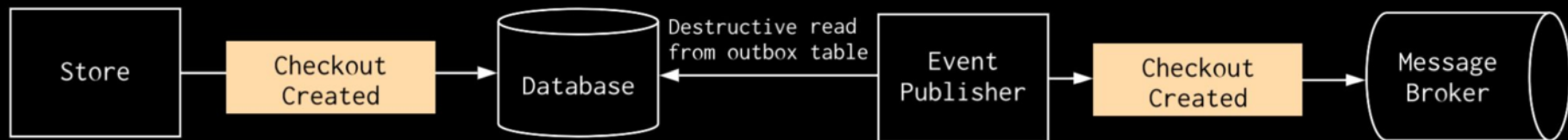
TRANSACTION LOGS



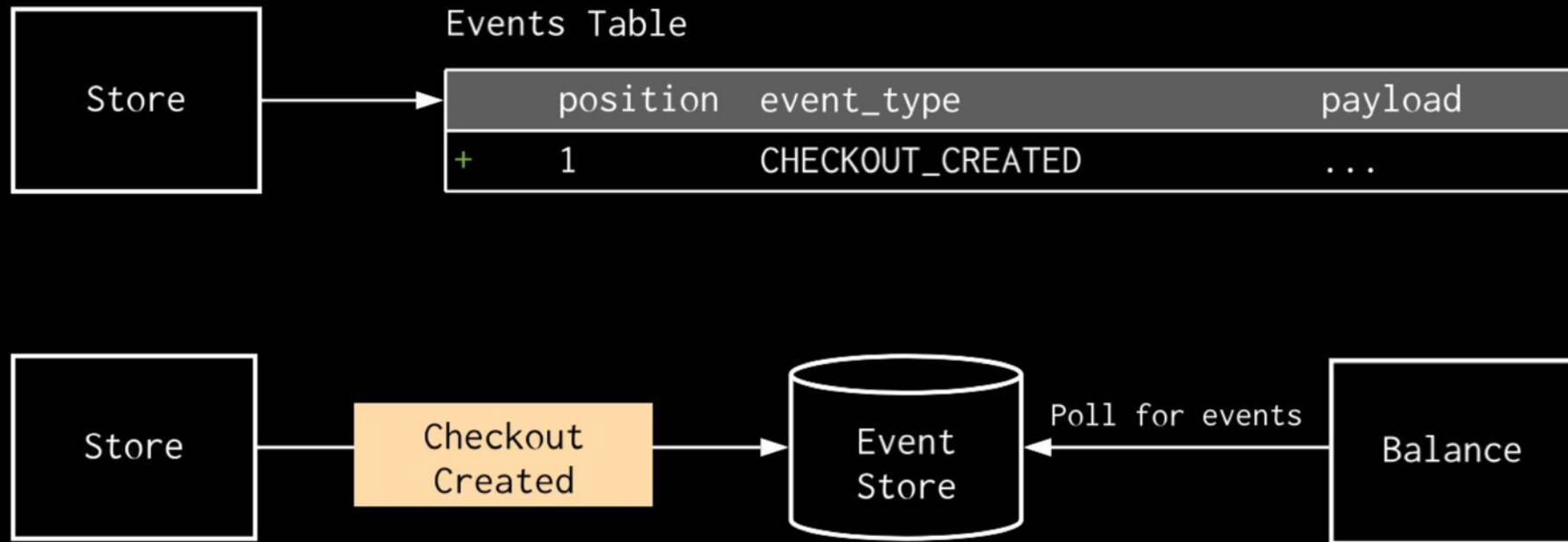
TRANSACTIONAL OUTBOX PATTERN



TRANSACTIONAL OUTBOX PATTERN



EVENT SOURCING



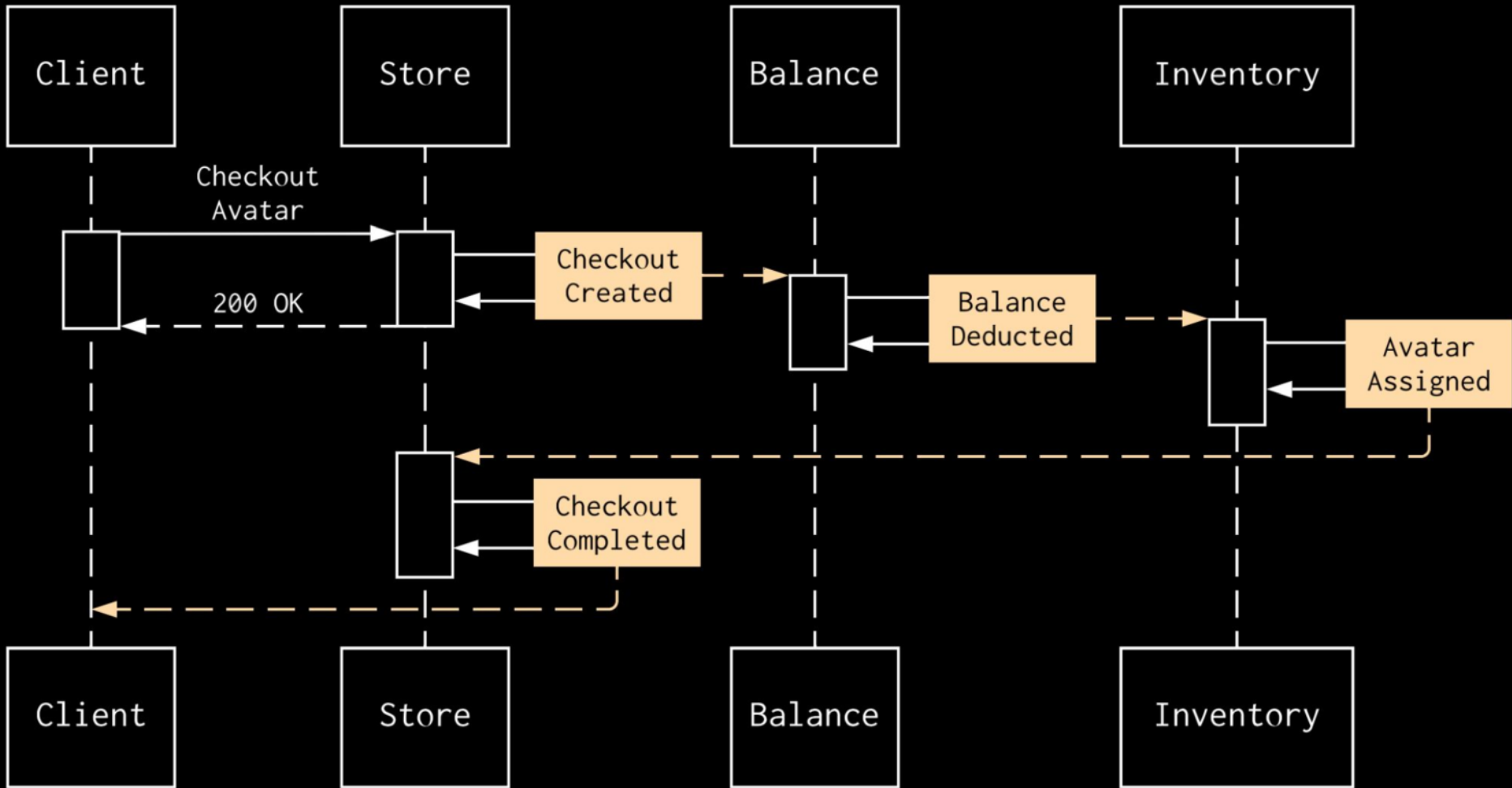
COMMUNICATION PATTERNS

```
/* Bringing it all together */
```

** Choreography

** Orchestration



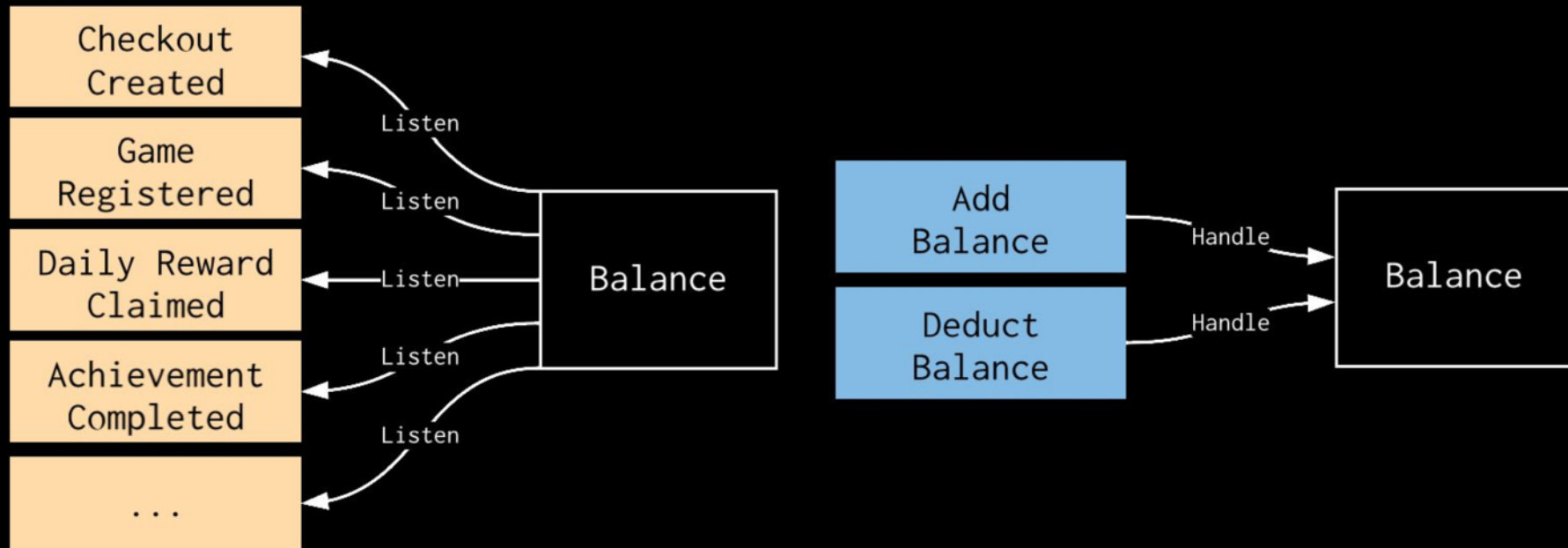


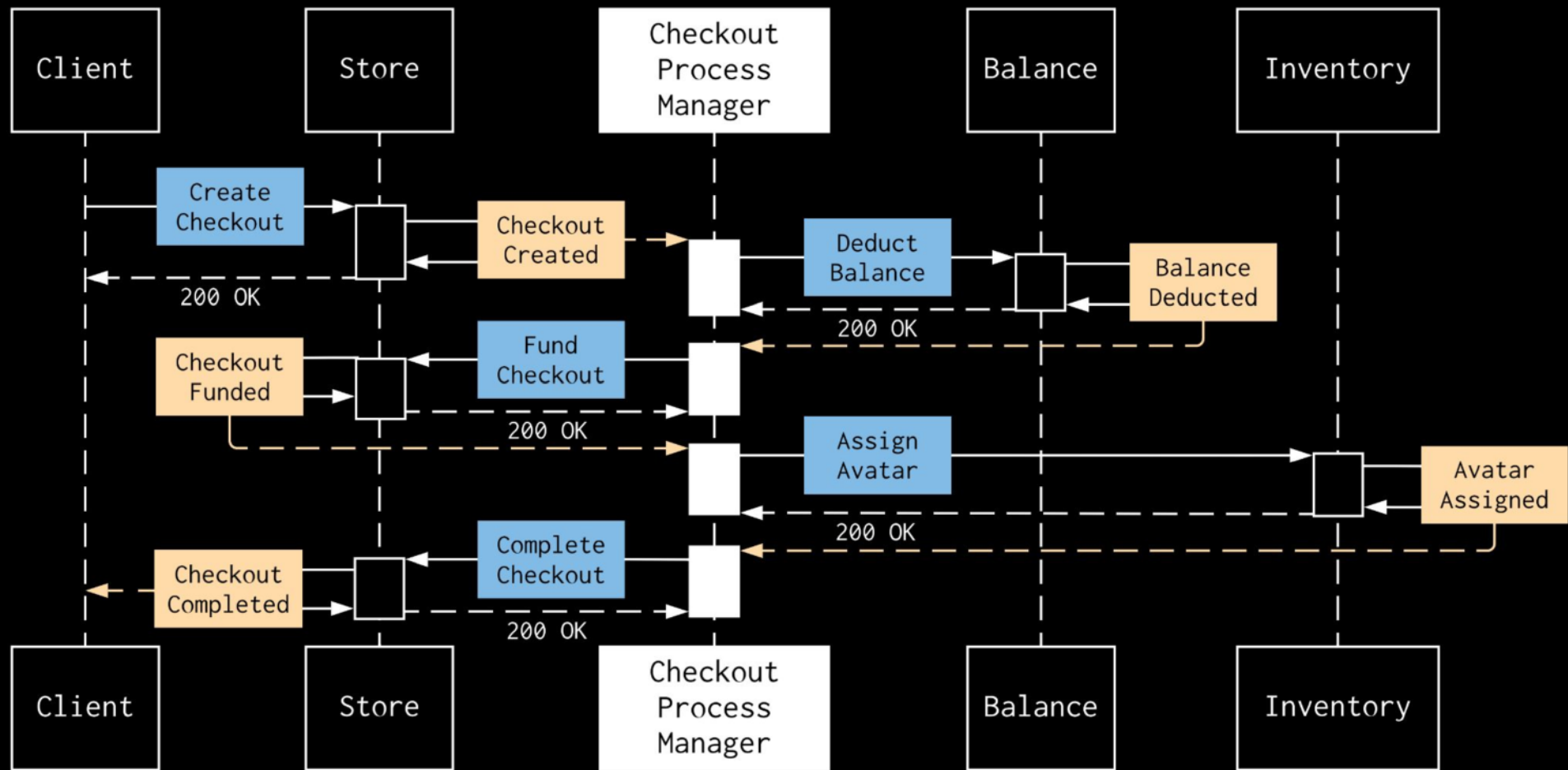


- ++ Elegant
- Difficult to Observe
- Optimal for Sequential Workflows



EVENTS VS. COMMANDS





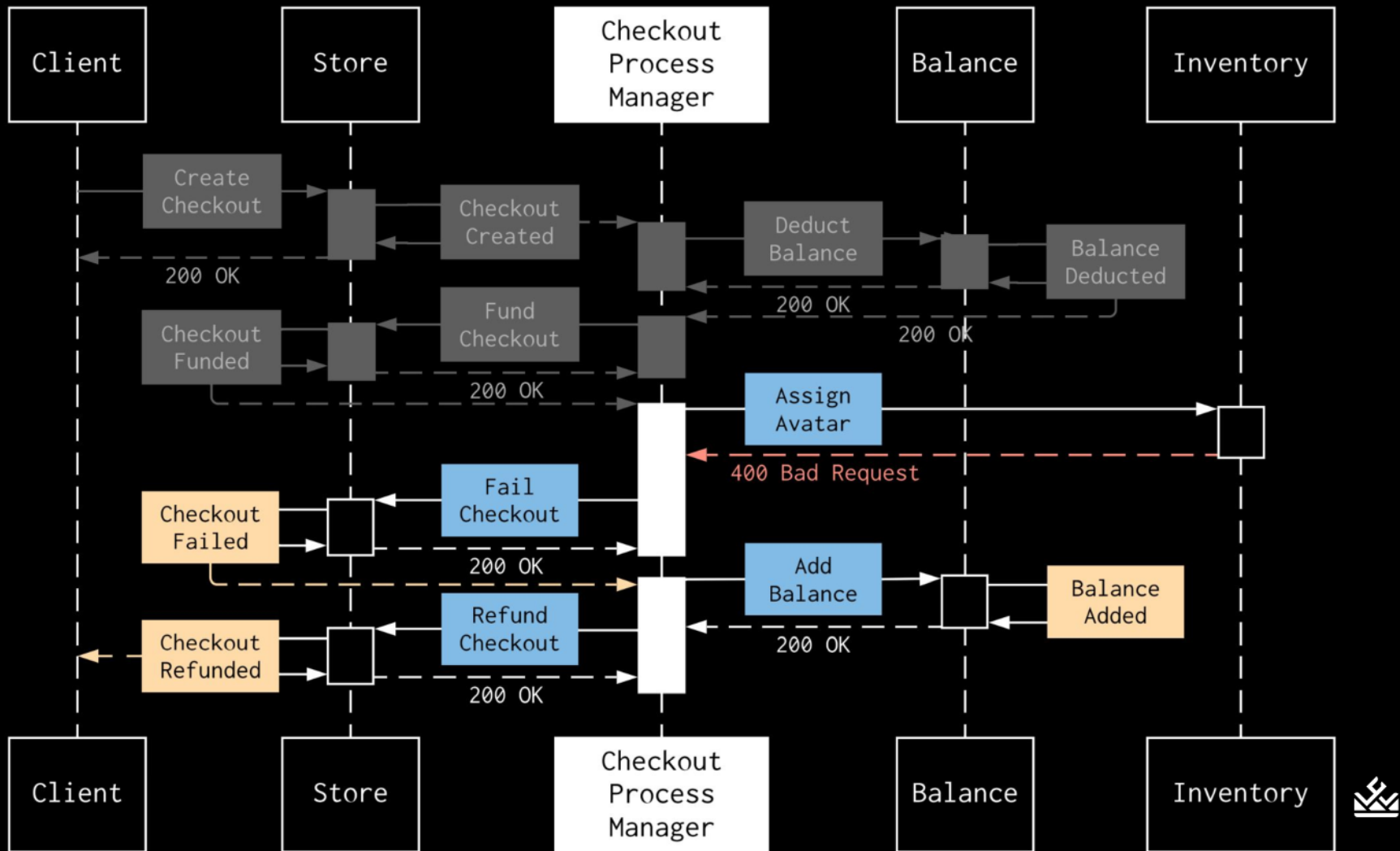




Photo by Tim Douglas from Pexels

- ++ Centralised workflow logic
- Single source of failure



There is no one-size
fits all architecture



- ** A Software Development Journey
- ** Event-driven Architecture
- ** Communication Patterns



THANK YOU!

Macklin Hartley

<https://macklin.me/>

