Celestial Navigation

Dead Reckoning

Event Sourcing

Andriy Drozdyuk

Outline

1. Etymology
2. Definition
3. Problem
4. What are Events?
5. Event Store
6. Benefits
7. Conclusion

Total time: 38 minutes

Etymology

Andriy Drozdyuk
@andriyko

Hi @gregyoung, what does the word "sourcing" in Event Sourcing stand for? Saw mention here as "source the event": eventstore.org/docs/introduct…

3:55 PM - 17 Sep 2017

2 Likes

1 2

Tweet your reply

gregyoung @gregyoung · Sep 17

Replying to @andriyko

I didn't make the name you would have to inquire to @martinfowler

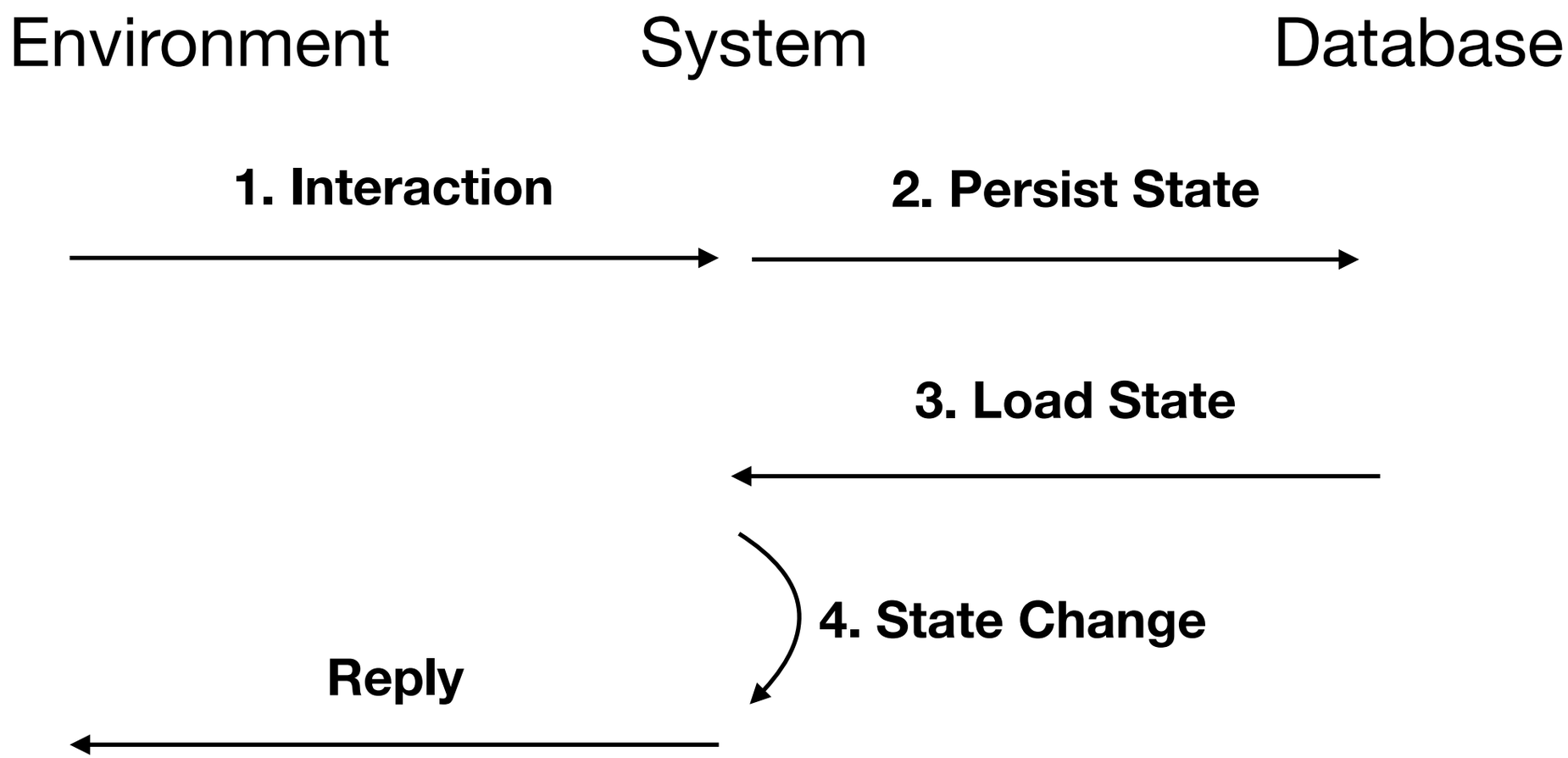
1 1

Definition

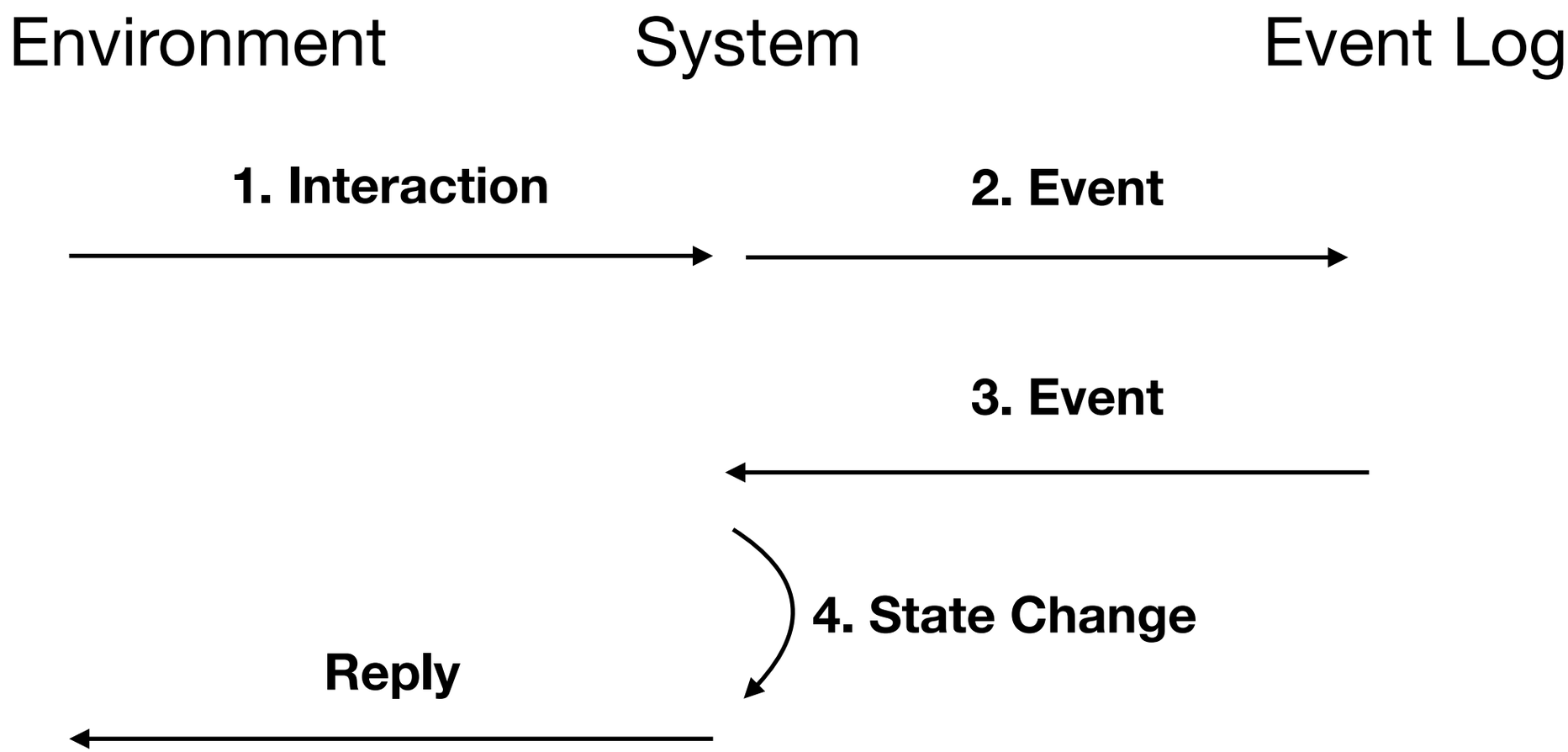
Event Sourcing

Architectural pattern defined by capturing domain events and changing the system's state as a response to those events.

Stereotypical Architecture



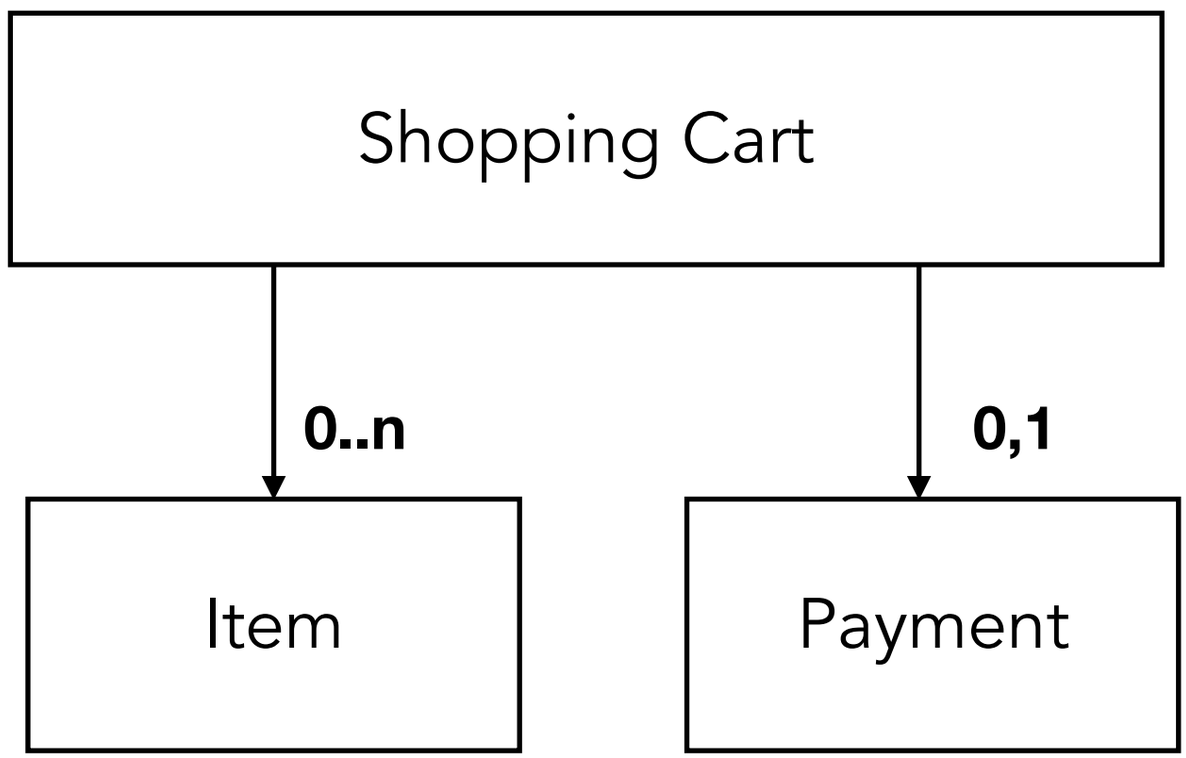
Event Sourcing



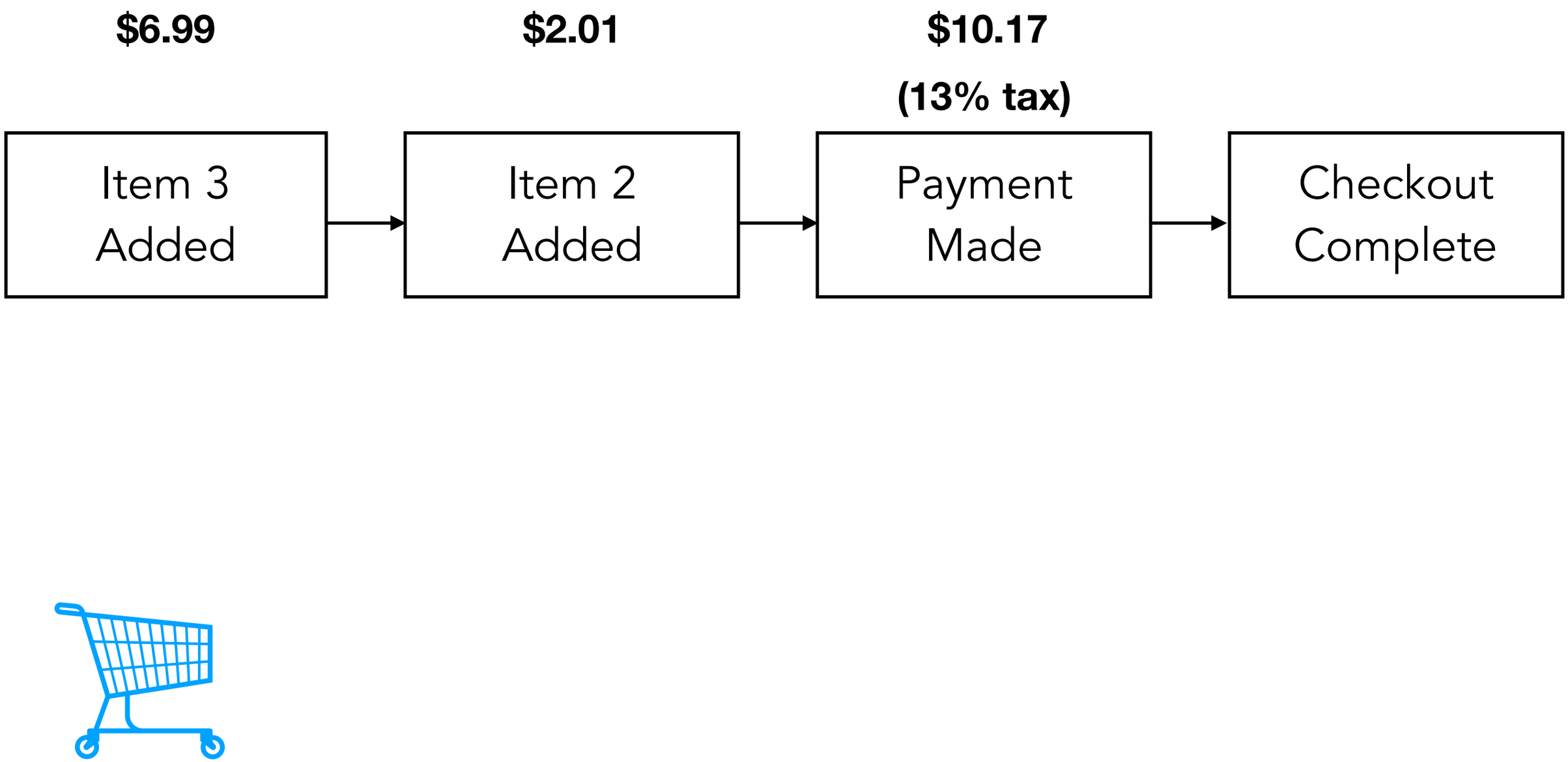
Problem

Shopping cart checkout

Shopping Cart
- Items
- Payment

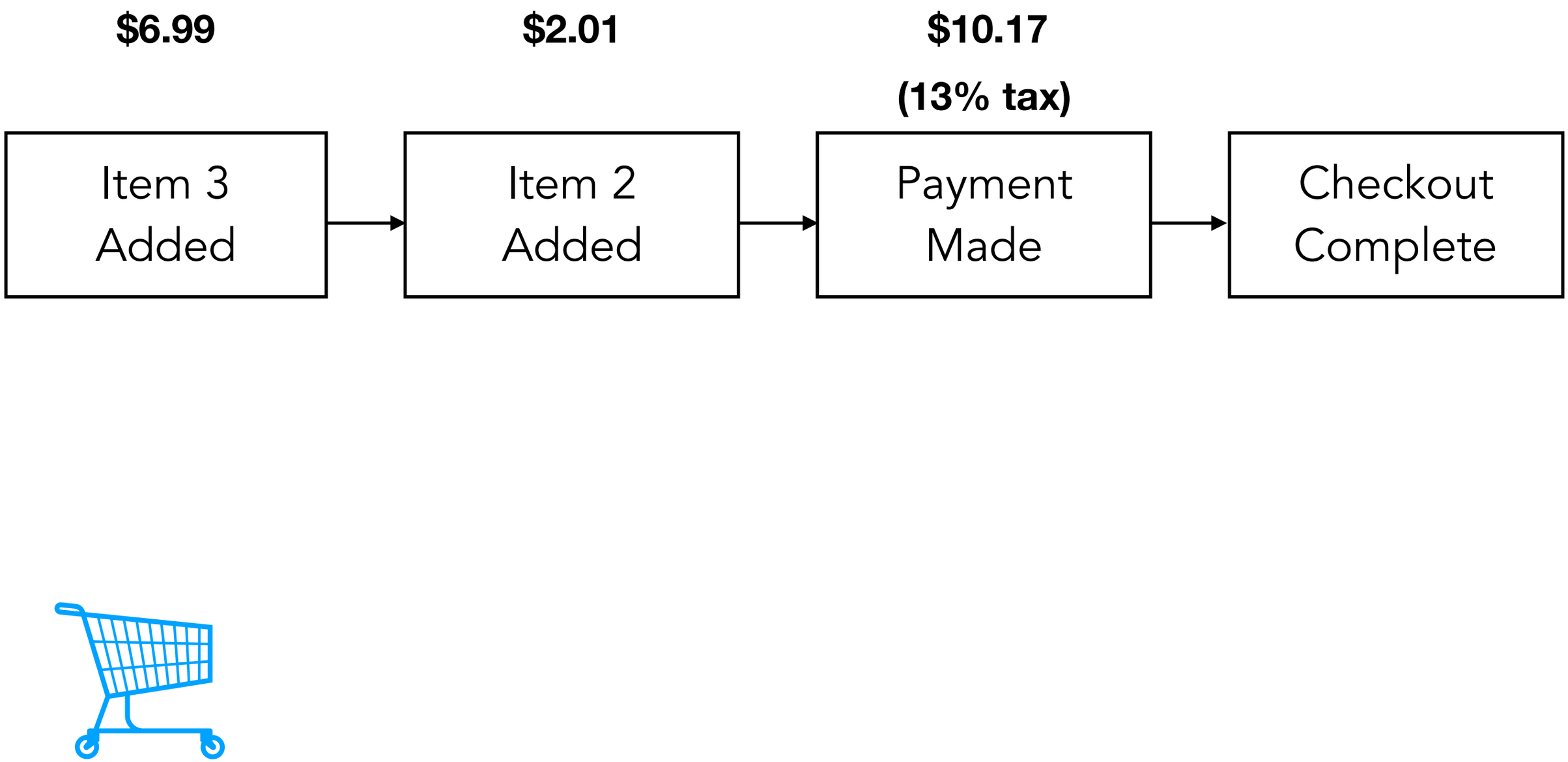


DATE		FOLIO	DEBIT $	CREDIT $
Apr 1	Dr Bank		15,000	
	Cr Capital			15,000
	Mr. Burnham deposited $15,000 in the bank account.			
7	Dr Bank		5,000	
	Cr Loan			5,000
	Loan of $5,000 from XYZ Bank.			
9	Dr Baking equipment		12,000	
	Cr Bank			12,000
	Baking equipment purchased from BB Machines CC.			
15	Dr Drawings		500	
	Cr Bank			500
	Mr. Burnham withdrew funds for personal use.			
17	Dr Bank		10,500	
	Cr Services rendered (income)			10,500
	Catering services for the Davidson's' wedding.			
19	Dr Salaries (expense)		4,000	
	Cr Bank			4,000
	Salary to W. Thistlespoon.			


$6.99
$2.01
$10.17
(13% tax)
Item 3 Added
Item 2 Added
Payment Made
Checkout Complete

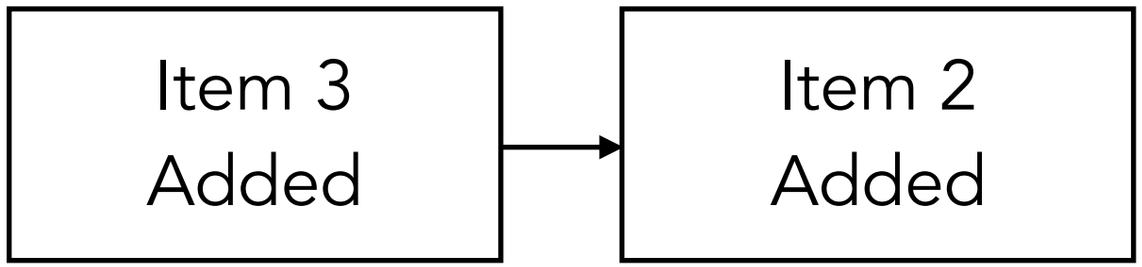
What are Events?

History


$6.99
$2.01
$10.17
(13% tax)
Item 3 Added
Item 2 Added
Payment Made
Checkout Complete

Current state?

$6.99 **$2.01**

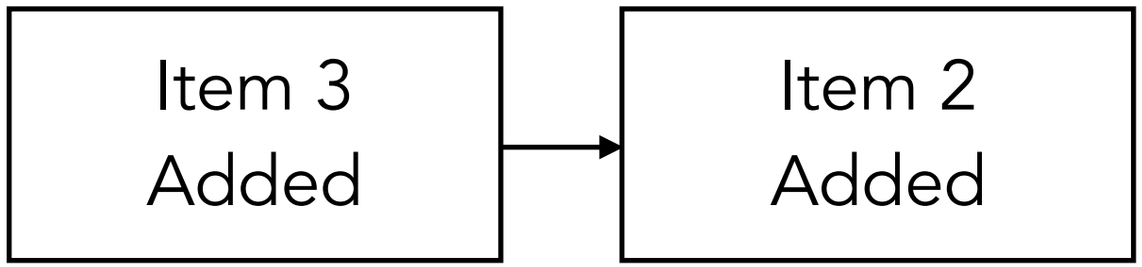


$6.99

Item 3
Added

$6.99

$6.99 **$2.01**

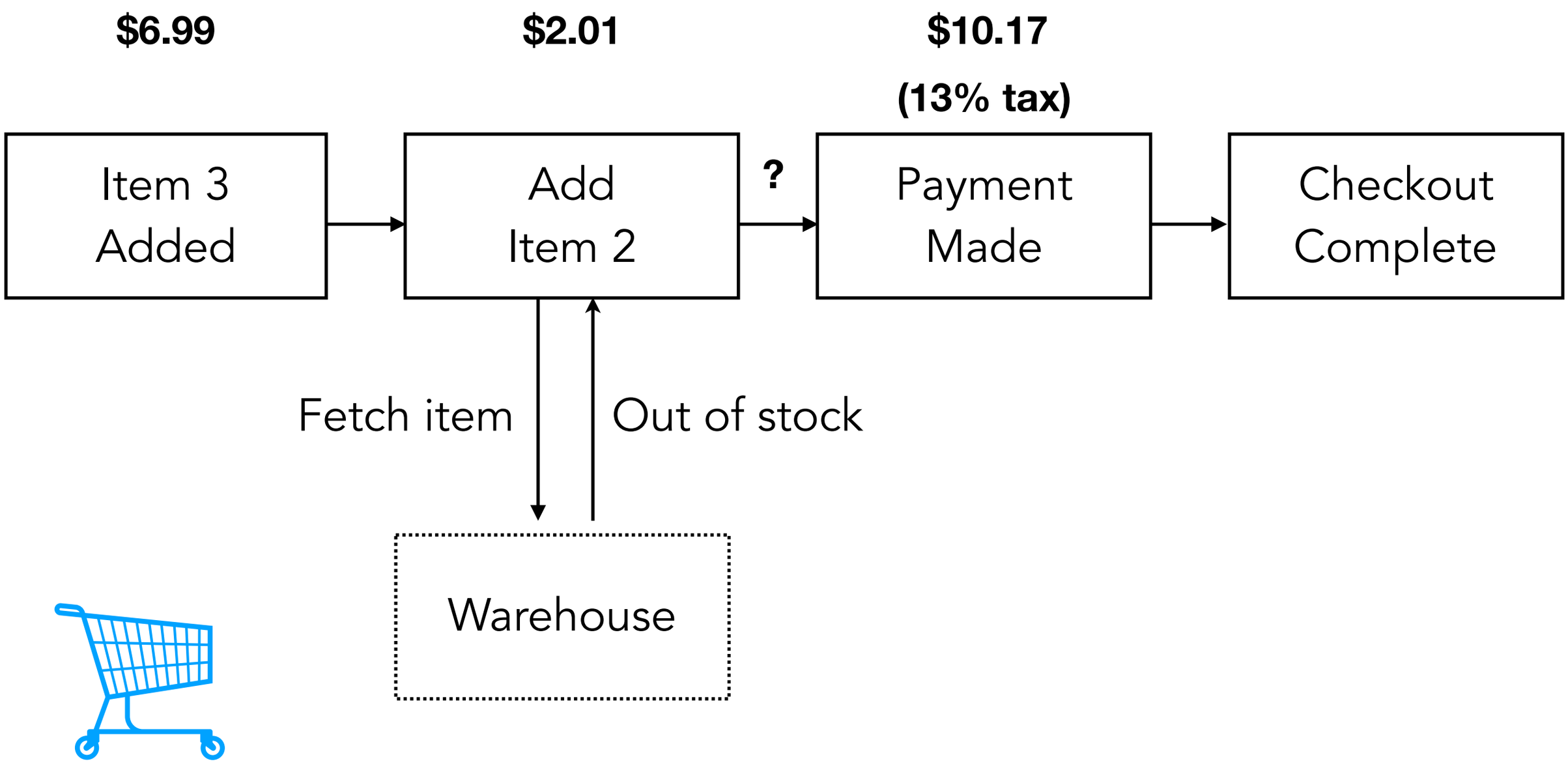


$6.99 + $2.01 = $9.00

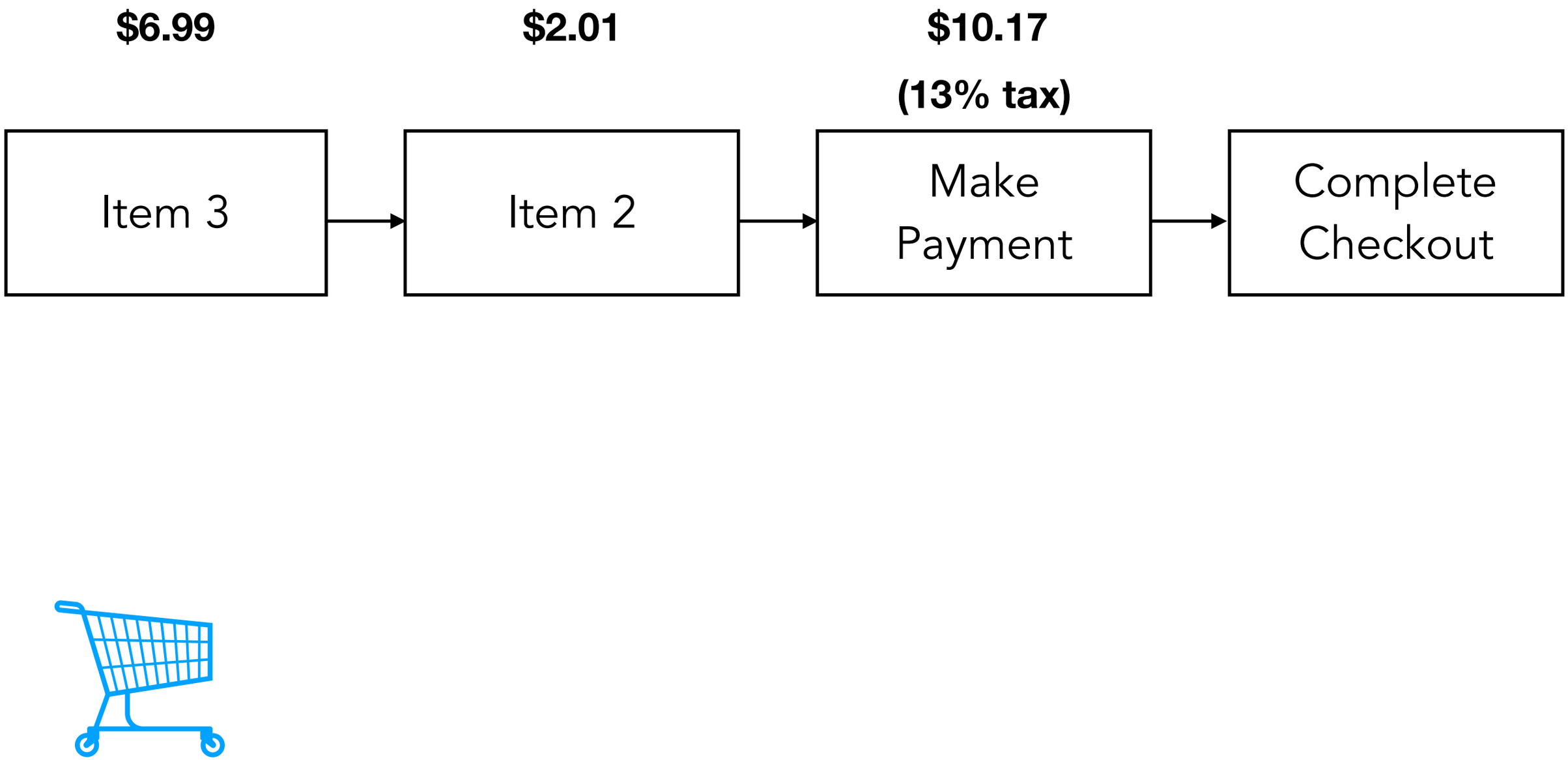
Projection

$9.00

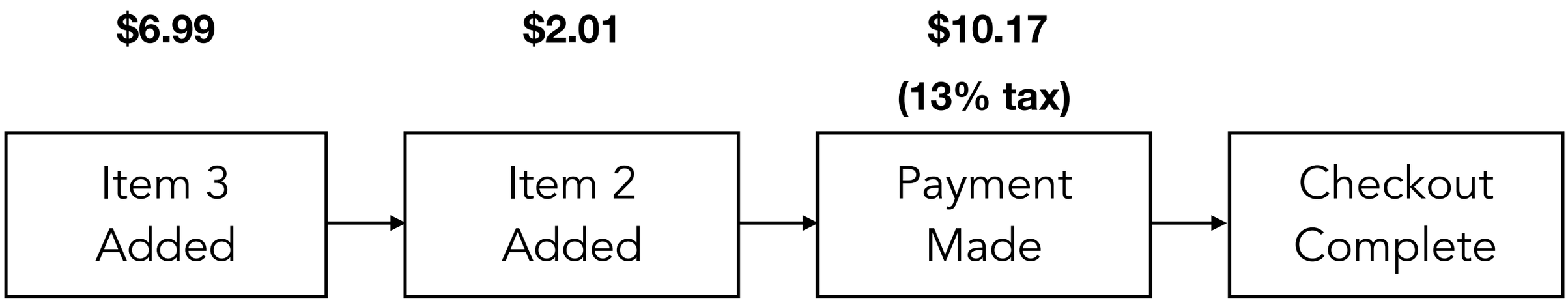
No side effects


$6.99
$2.01
$10.17
(13% tax)
Item 3 Added
Add Item 2
?
Payment Made
Checkout Complete
Fetch item
Out of stock
Warehouse

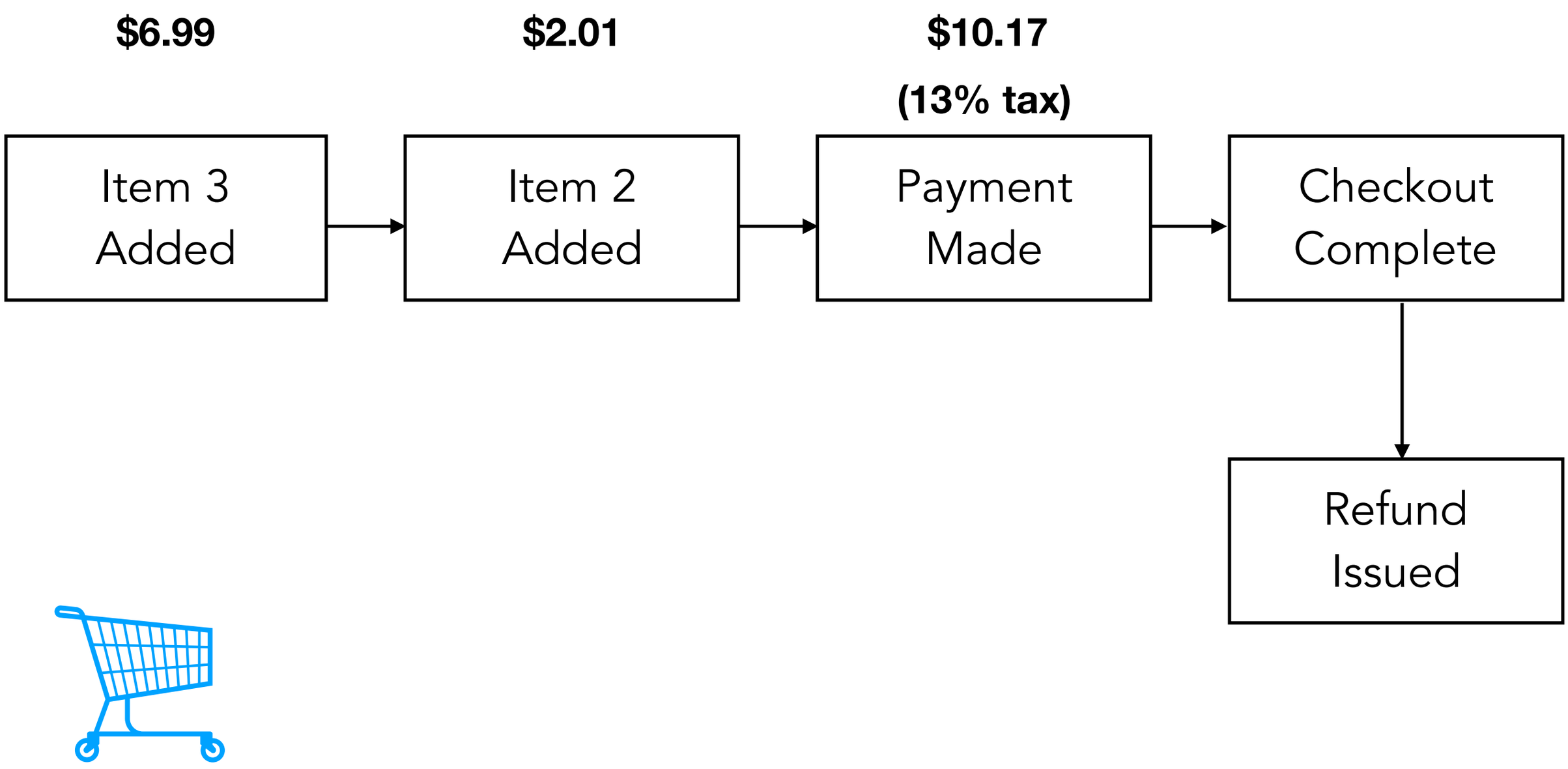
Past tense


$6.99
$2.01
$10.17
(13% tax)
Item 3
Item 2
Make Payment
Complete Checkout

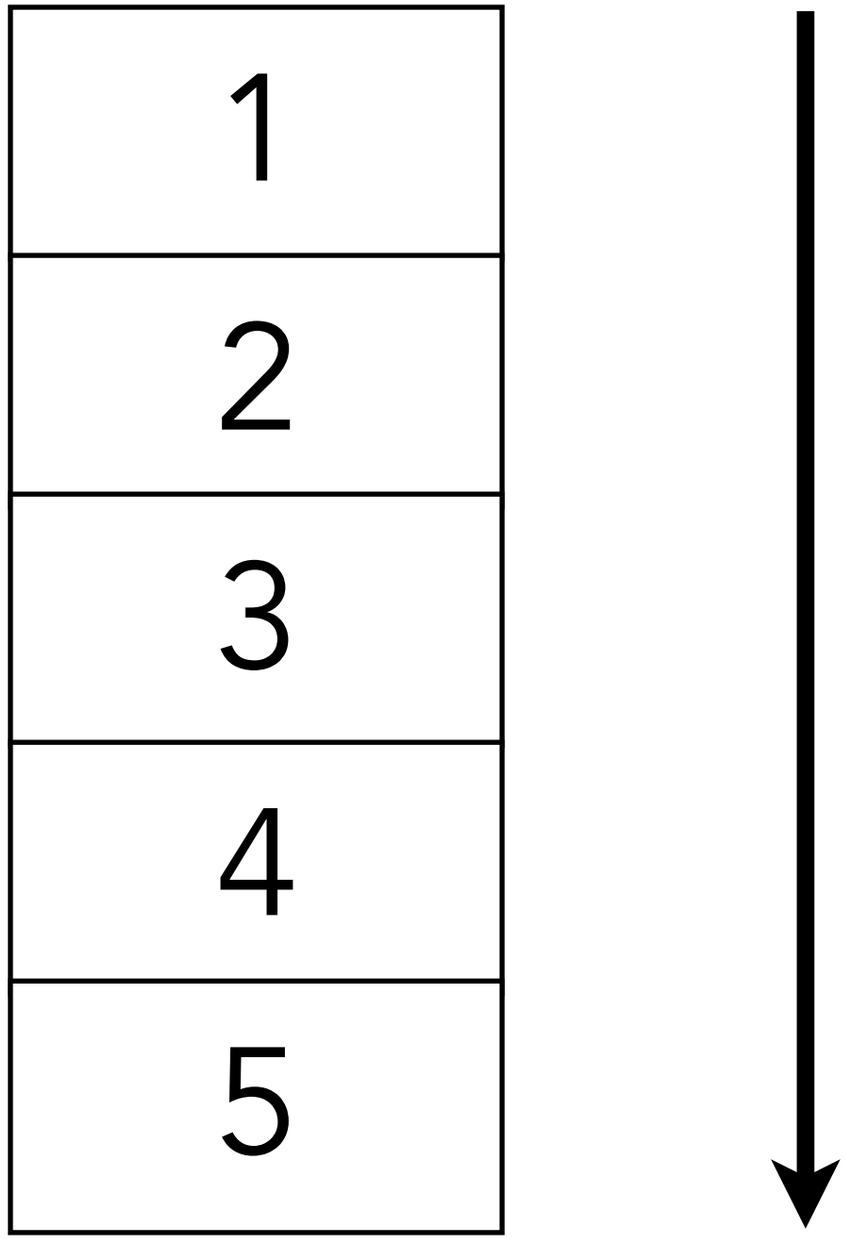
No delete



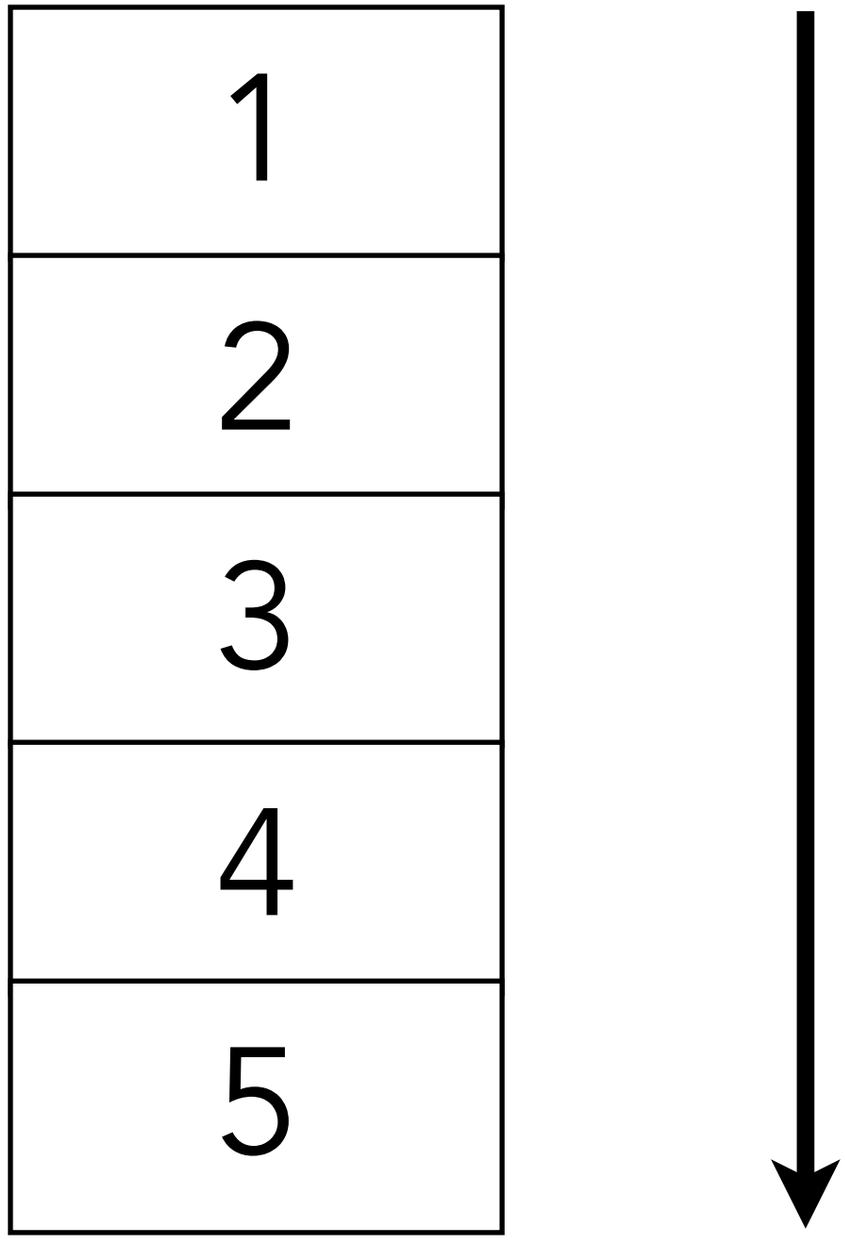
Delete customer order?


$6.99
$2.01
$10.17
(13% tax)
Item 3 Added
Item 2 Added
Payment Made
Checkout Complete
Refund Issued

Event store

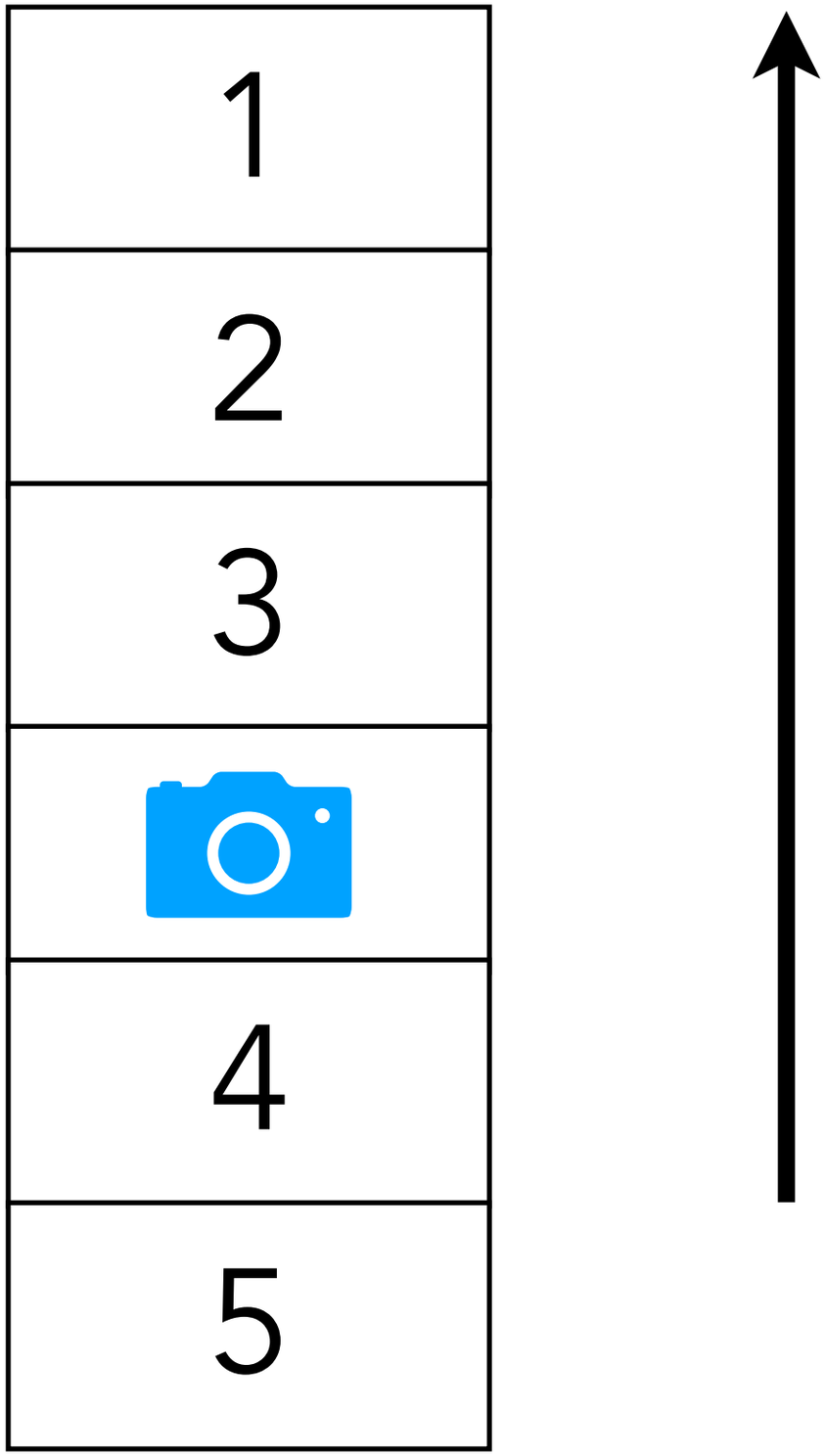

1
2
3
4
5

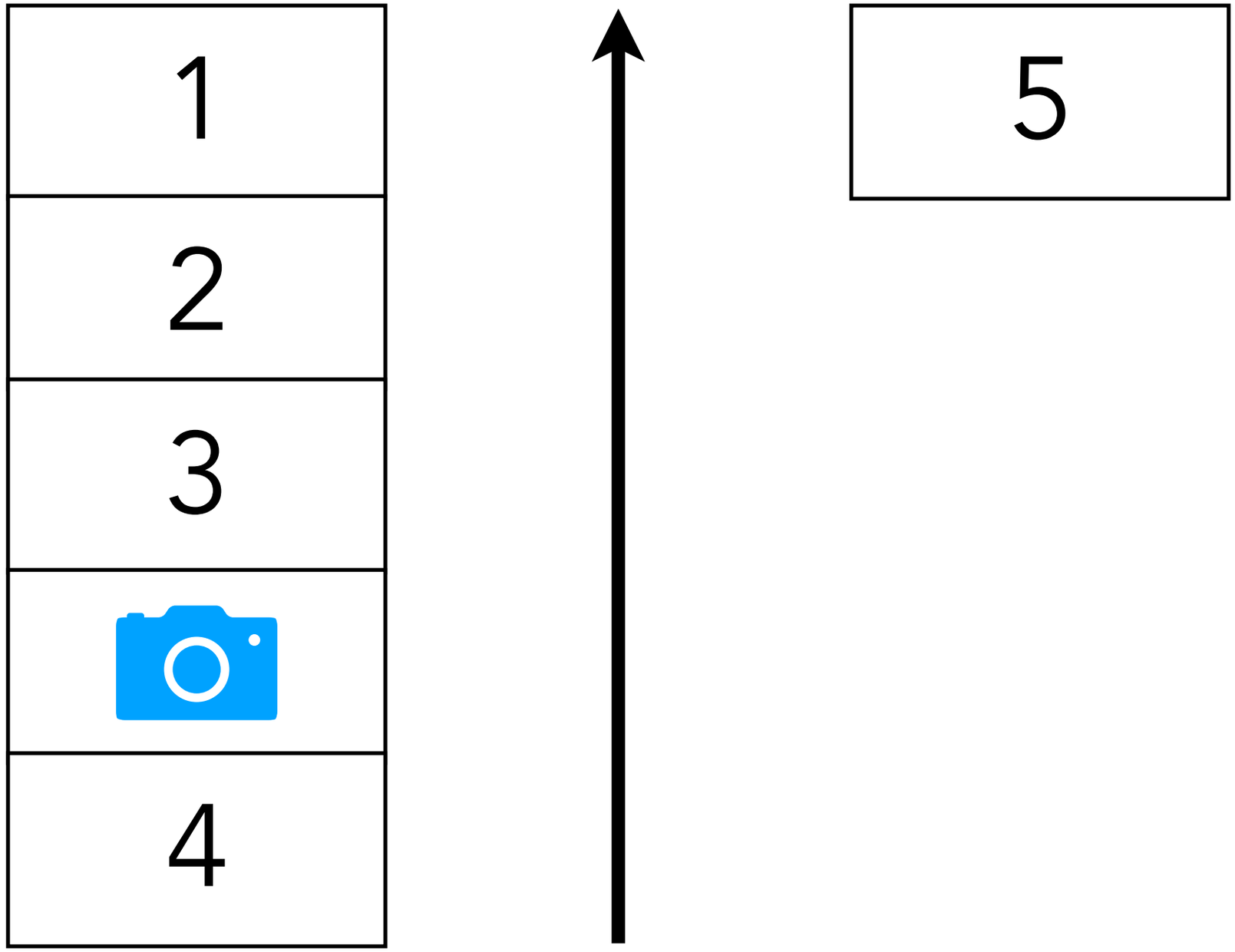
Snapshots

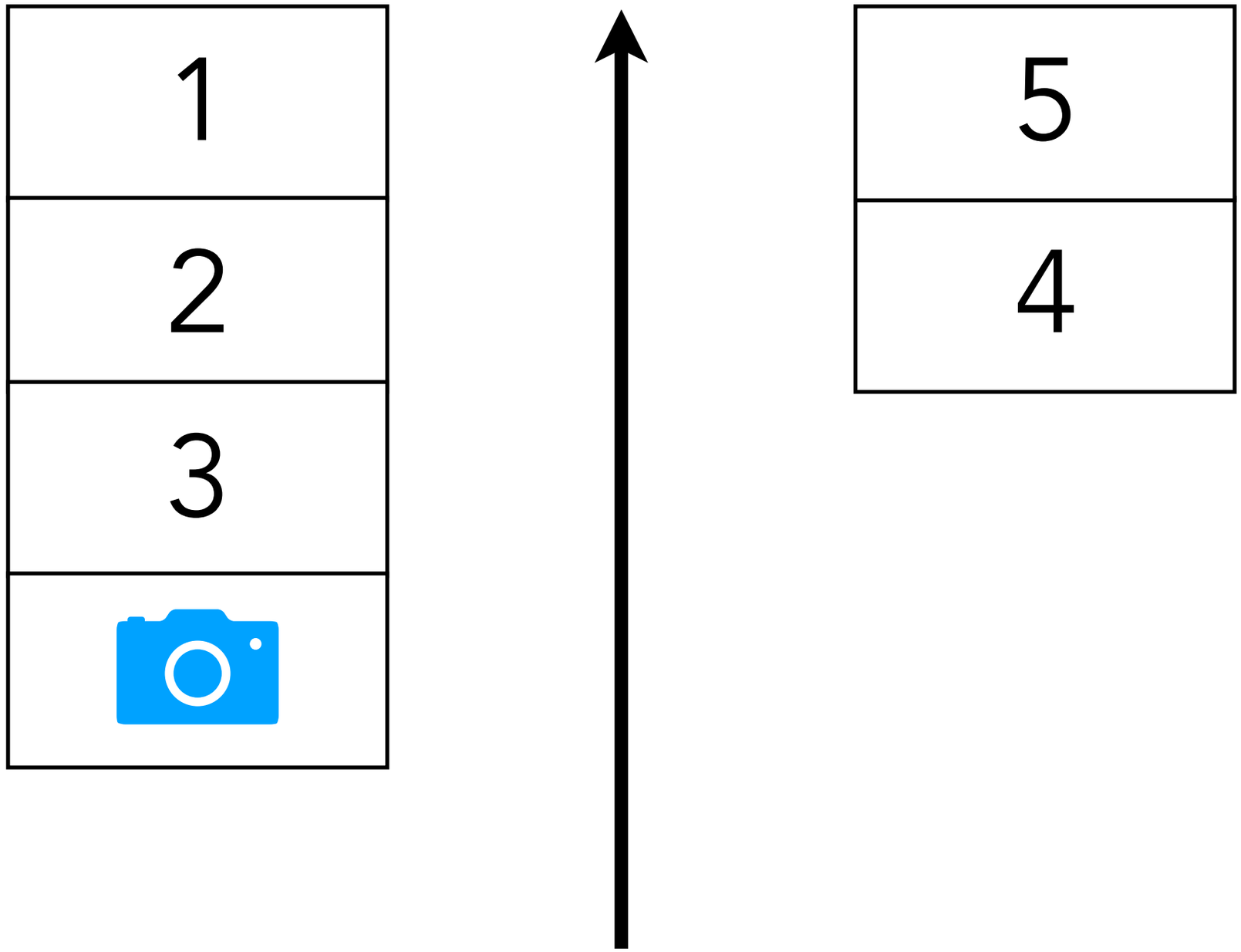

1
2
3
4
5

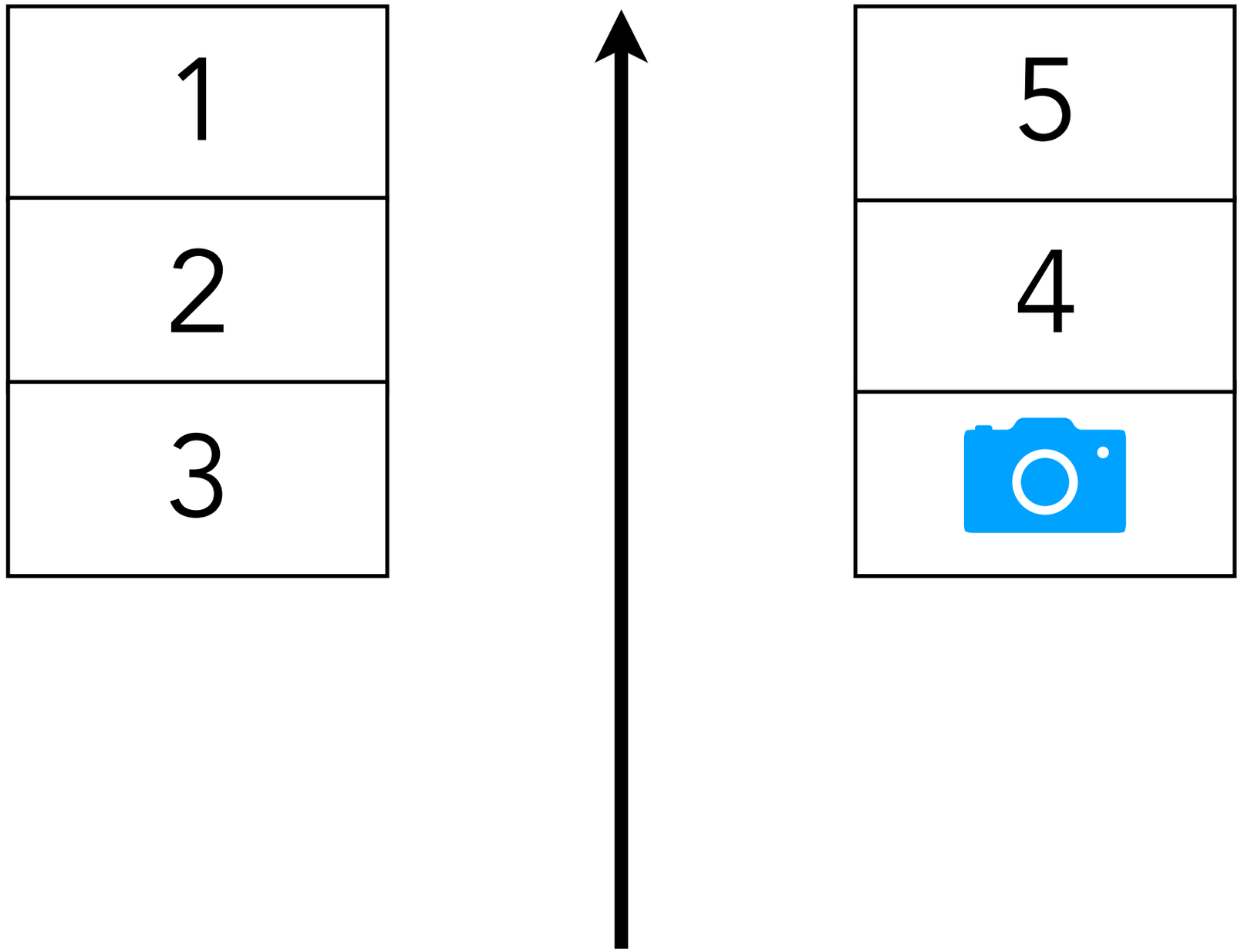
1
2
3

4
5


1
2
3
4
5


1
2
3
4
5


1
2
3
5
4


1
2
3
5
4

Benefits

Enables Domain Driven Design.

Eliminates impedance mismatch

	Units	Intrerface	Concerns	Foundation	Types
OO	Objects	Behaviour	Extensibility, Reusability, Safety	Graph theory	Yes

	Units	Intrerface	Concerns	Foundation	Types
OO	Objects	Behaviour	Extensibility, Reusability, Safety	Graph theory	Yes
Relational	Rows, tables	Data	Efficiency, Fault tolerance, Liveness	Set theory	No

	Units	Intrerface	Concerns	Foundation	Types
OO	Objects	Behaviour	Extensibility, Reusability, Safety	Graph theory	Yes
Relational	Rows, tables	Data	Efficiency, Fault tolerance, Liveness	Set theory	No
Events	Events	Behaviour	Carefree!	Functions	Yes

Eliminates impedance mismatch

OO - objects, tranversal, behaviour as interface.

Relational - rows and tables, joins, data as interface.

Aggregate?

Model that represents your domain logic.

Like a “class” but more general and domain specific.

E.g.: “Shopping Cart" aggregate for a session. Contains all the rules for adding & removing items, payment and checking out.

Easy to partition

Partitioning is based on an aggregate id.

Easier to model Aggregates

Easy to load an aggregate.

No lazy loading required.

Business Value

Provides insight into the system.

“If a customer buys X he is likely to buy Y at some time later?”

“What is the most number of reactor-core failures that ever happened in one hour period?”

“Did the change we make to code three months ago really increase the number of successful orders placed?”

Future-proofs the system

Events — record every action ever taken.

Future-proofs the system

Events — record every action ever taken.

Model — simplified representation of the system.

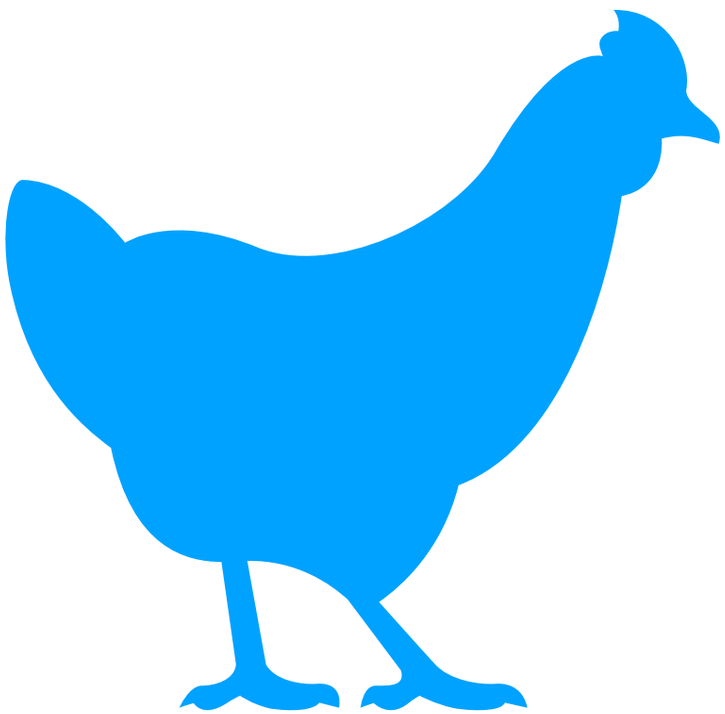
Future-proofs the system

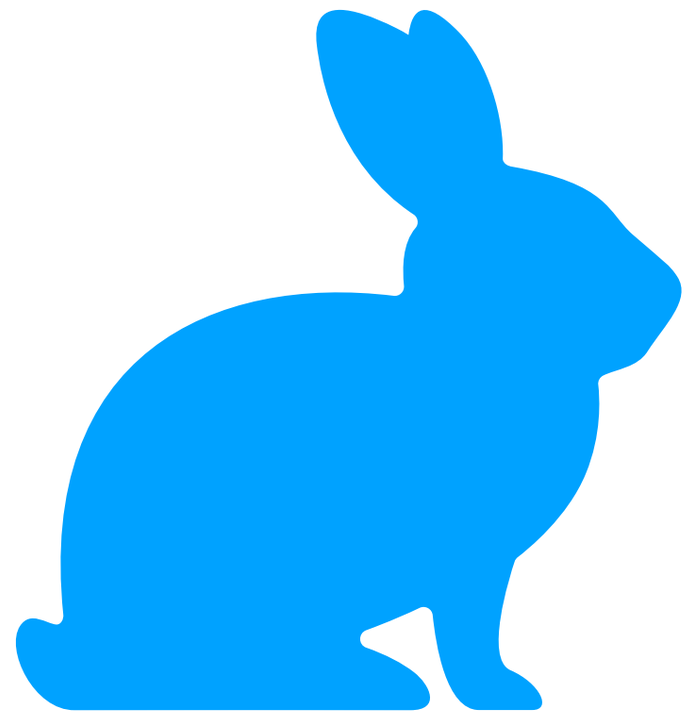
Events — record every action ever taken.

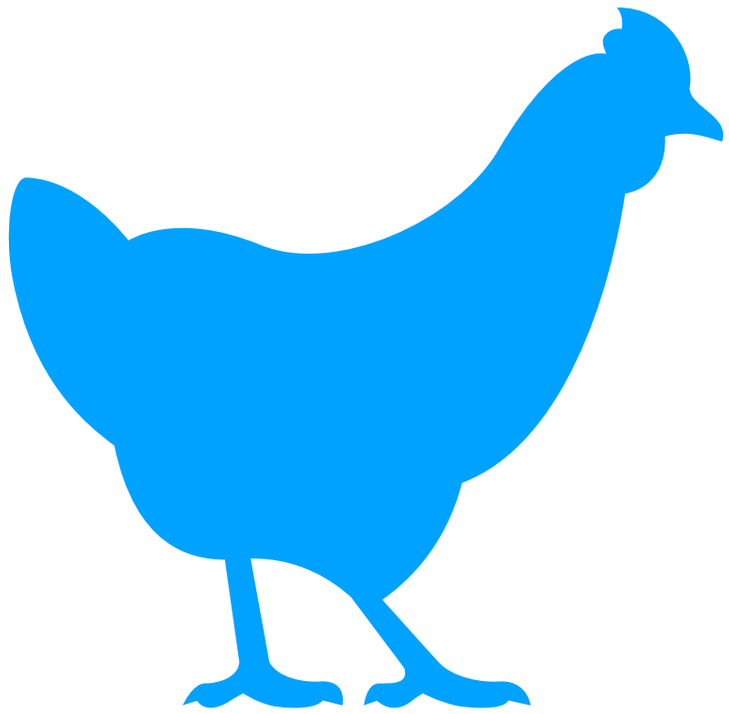
Model — simplified representation of the system.

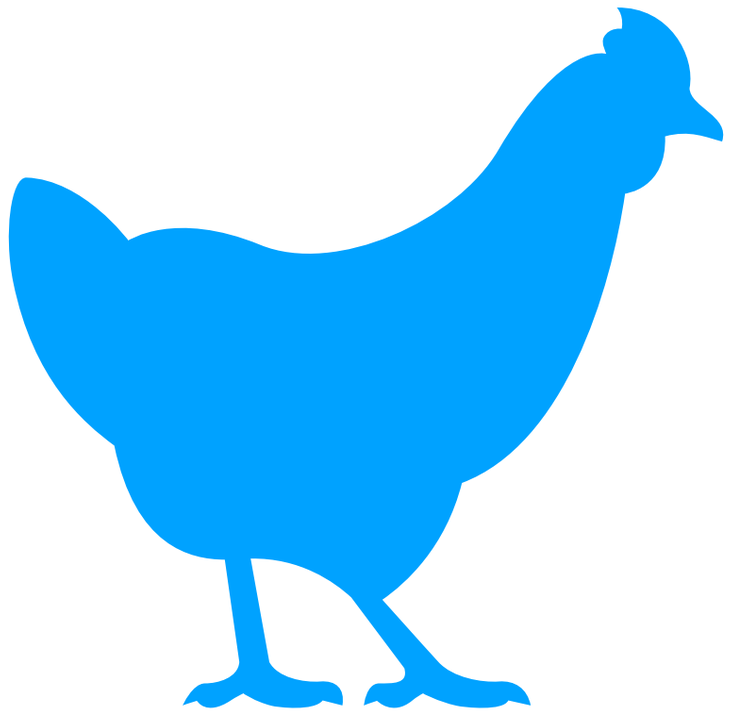
Based on events, we can build any possible model of the system.

Business Value Example

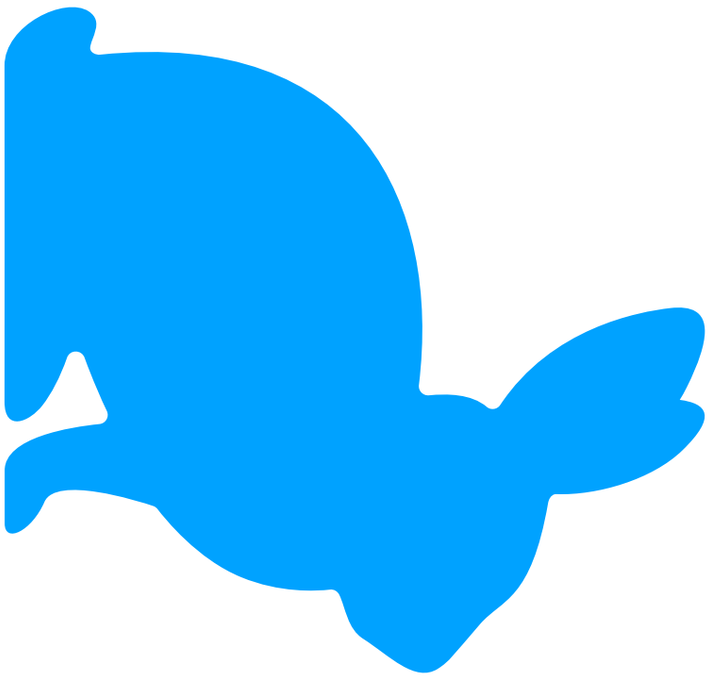


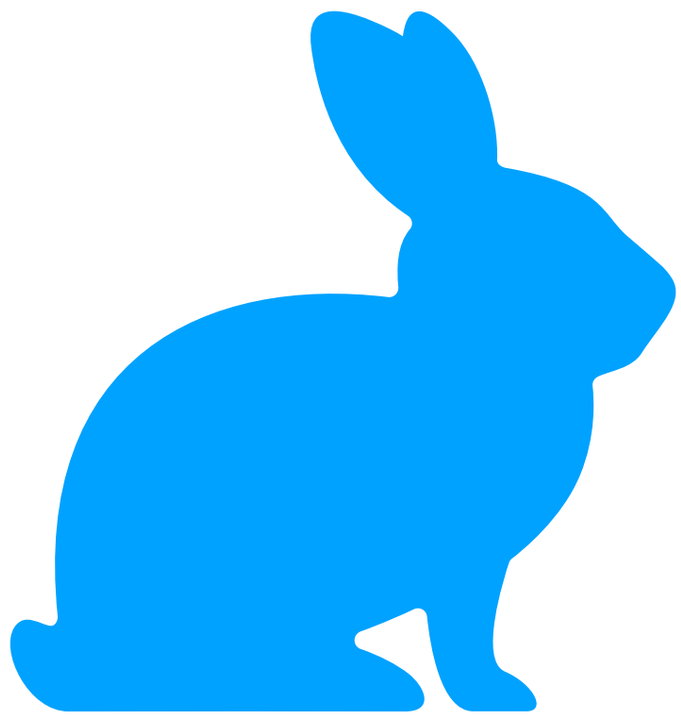




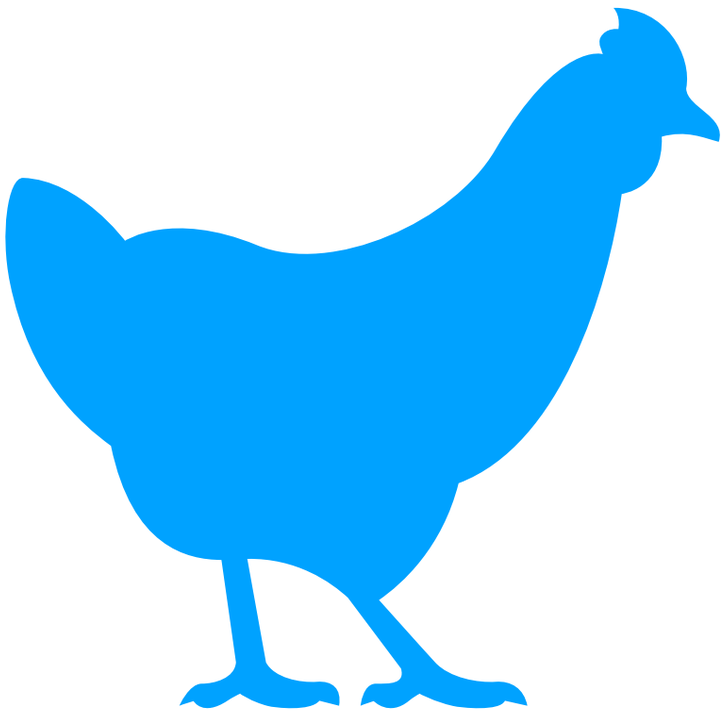


2017—

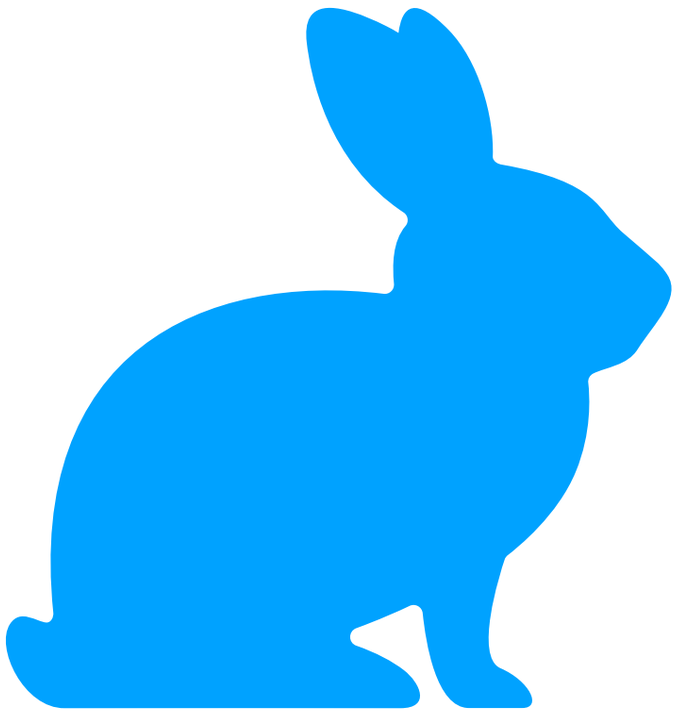




4000 BC —

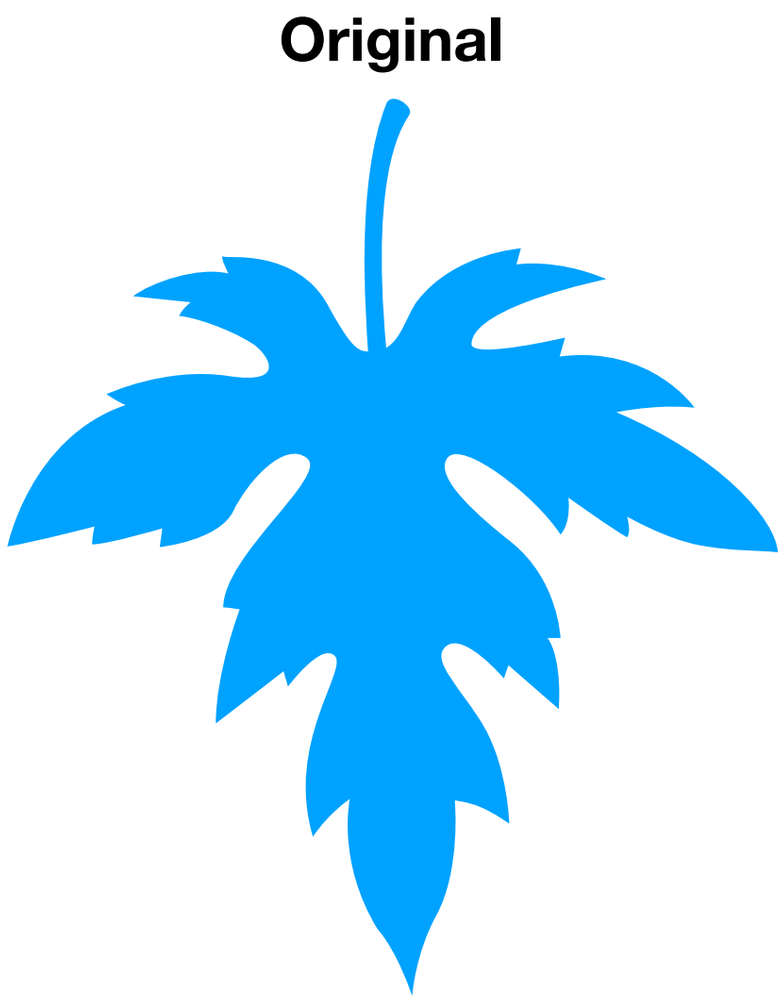


2017—

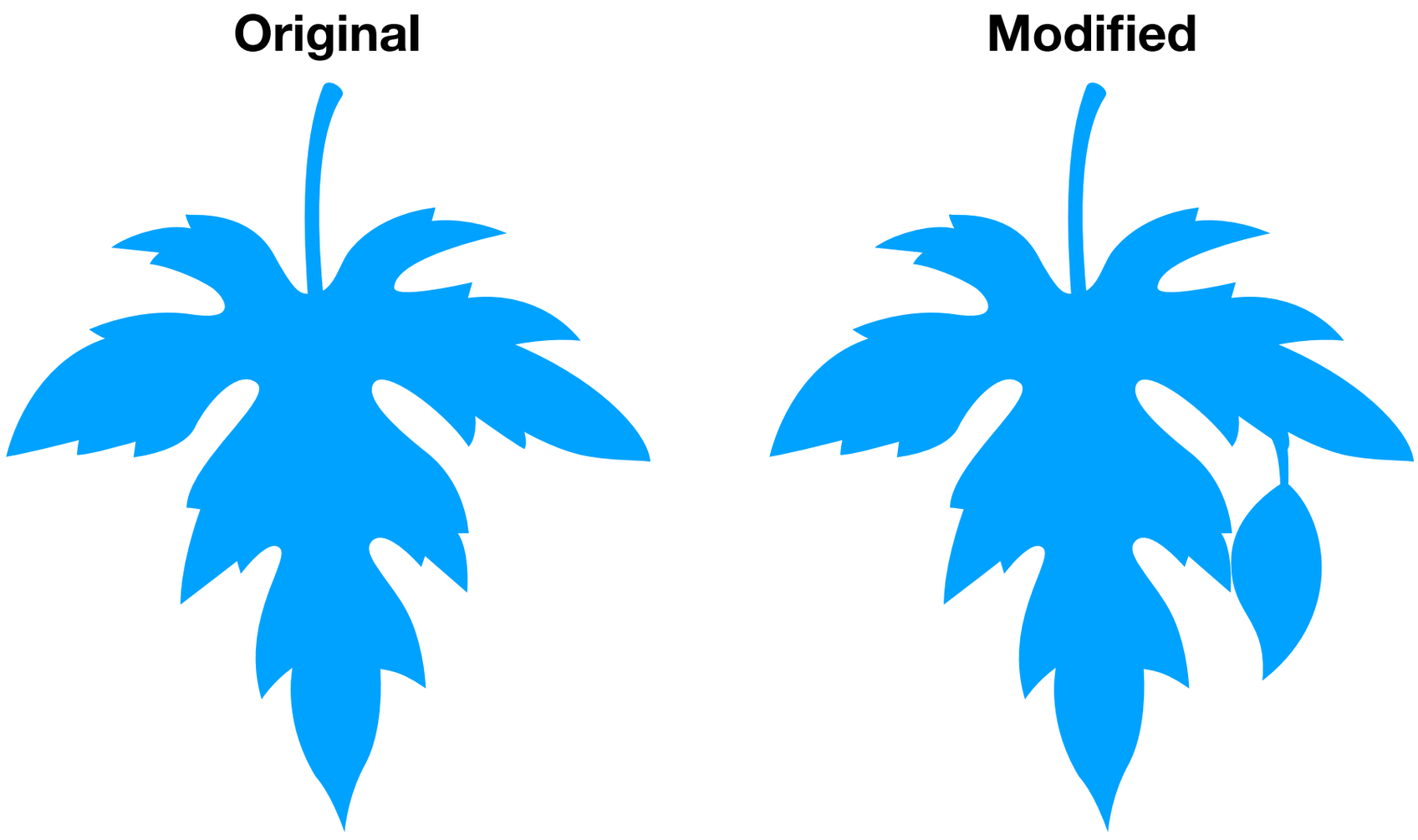


4000 BC—

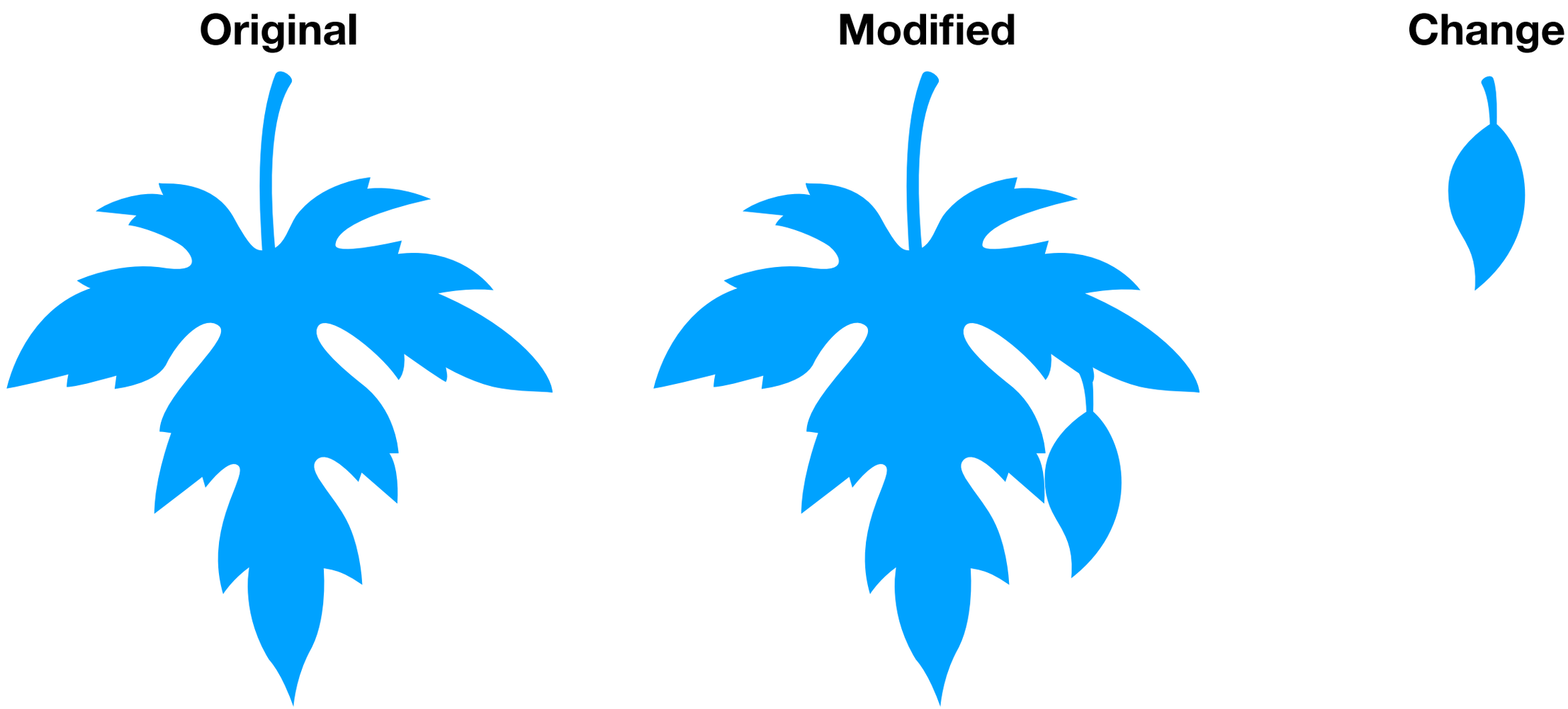
No need for ORM



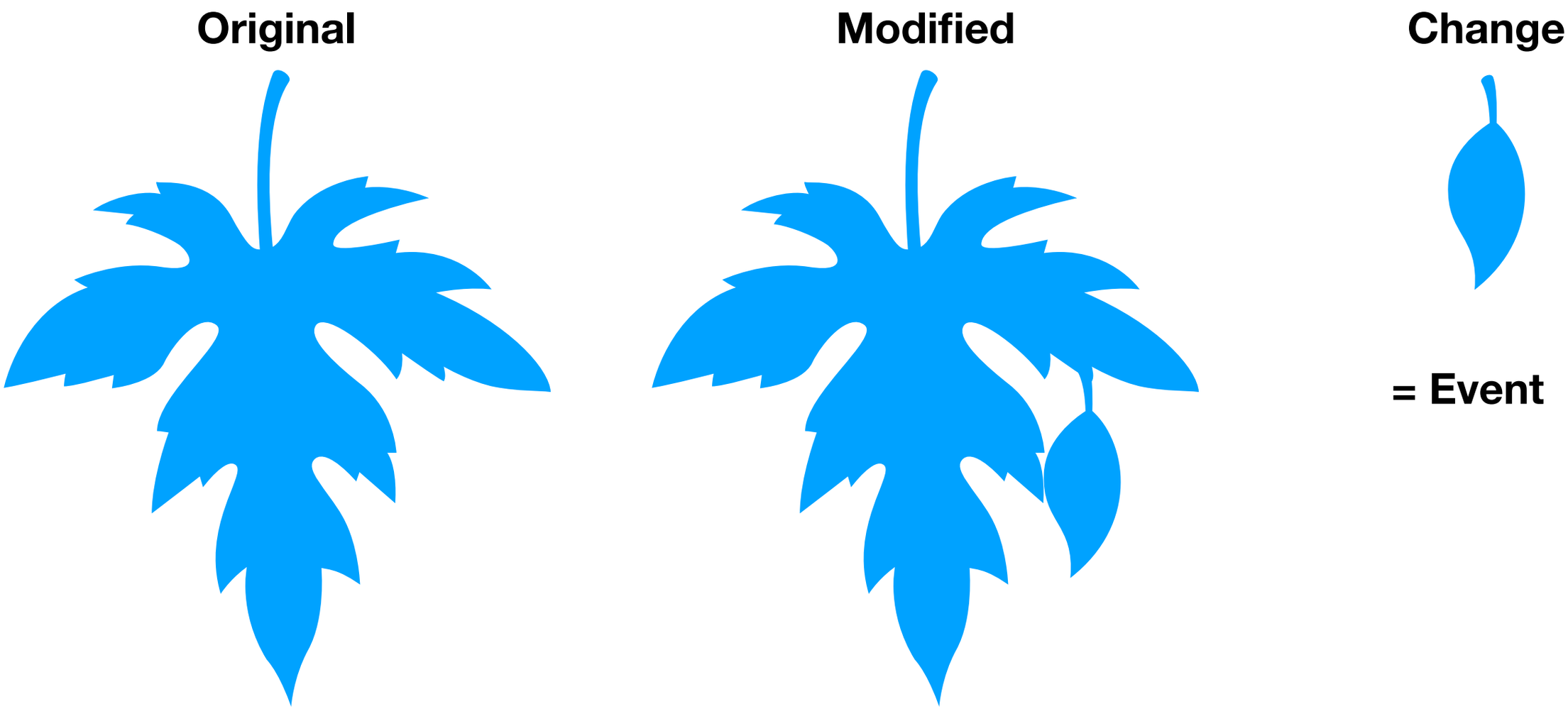
No need for ORM



No need for ORM



No need for ORM



Conclusion

"Don't be a chicken. Use event sourcing."
—Andriy Drozdyuk, 2017

Thank you!

Andriy Drozdyuk
@andriyko on Twitter

The End

References

1. CQRS documents by Greg Young.
2. The Charming Genius of the Apollo Guidance Computer, Brian Troutwine. Video.