EVENT STORMING

A
Super Hero

Story

At the beginning was the
verb...
at past tense

Welcome

Agile Grenoble

Bonitasoft

sopra steria

Groupe SAMSE

VISEO
DIGITAL MAKERS

CGI

KAIZEN

LIGHT UP YOUR FUTURE_
HARDIS GROUP

klaxoon

open
WE EMPOWER YOUR DIGITAL WORLD

Atos

l'Assurance Maladie
Caisse Nationale

zenika

loquant
Solutions Relation Client

Persistent

salesforce

Viveris

photoweb

Schneider Electric

OROS
MEASURING NOISE & VIBRATION

ARaymond
MORE THAN FASTENING

PARX

gfi
NEW CHALLENGES, NEW IDEAS

GRENOBLE ALPES MÉTROPOLE

Cédric Pontet

CTO

Software craftsman

Agile coach / Facilitator

Sketchnoter

@cpontet

ap agilepartner

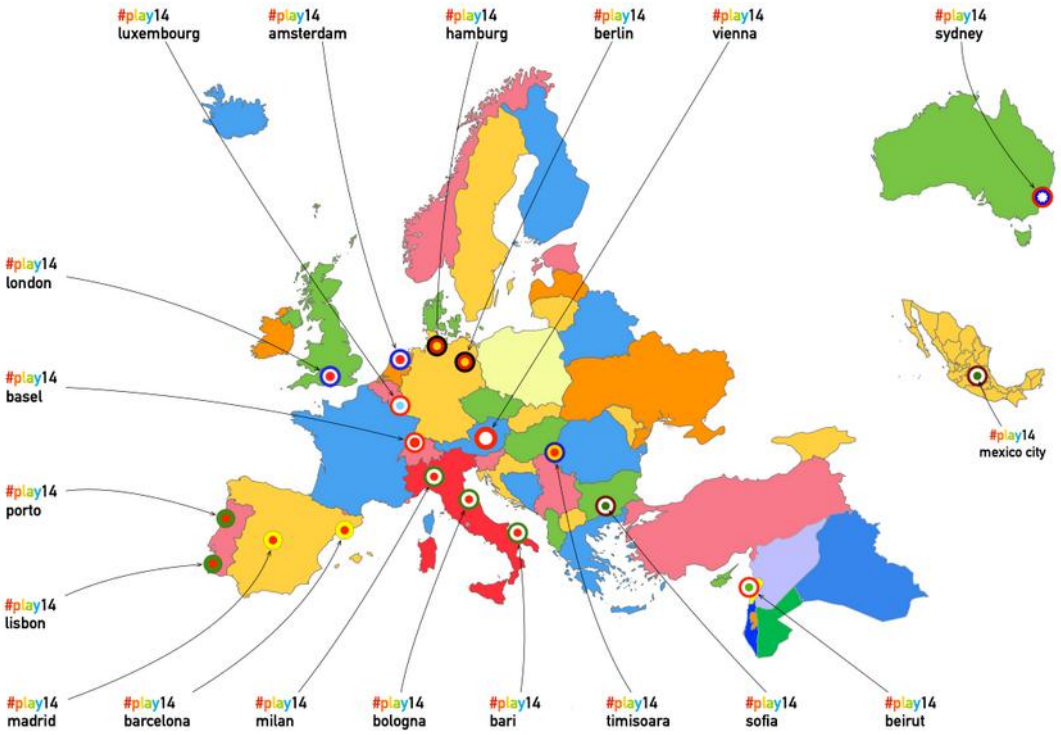
We develop great software and we help others do it right

Cédric Pontet

Founder

Serial Player

Happy Salmon Guru



@play14team

#play14

International gathering of like-minded people who share the common idea that playing is the best way to learn and understand !

Cédric Pontet

Event Stormer

EventStorming summit in Bologna 2018

EVENT STORMING

We just love sticky notes

... especially the orange ones

What is EventStorming?

EventStorming summit in Bologna 2018

A workshop format

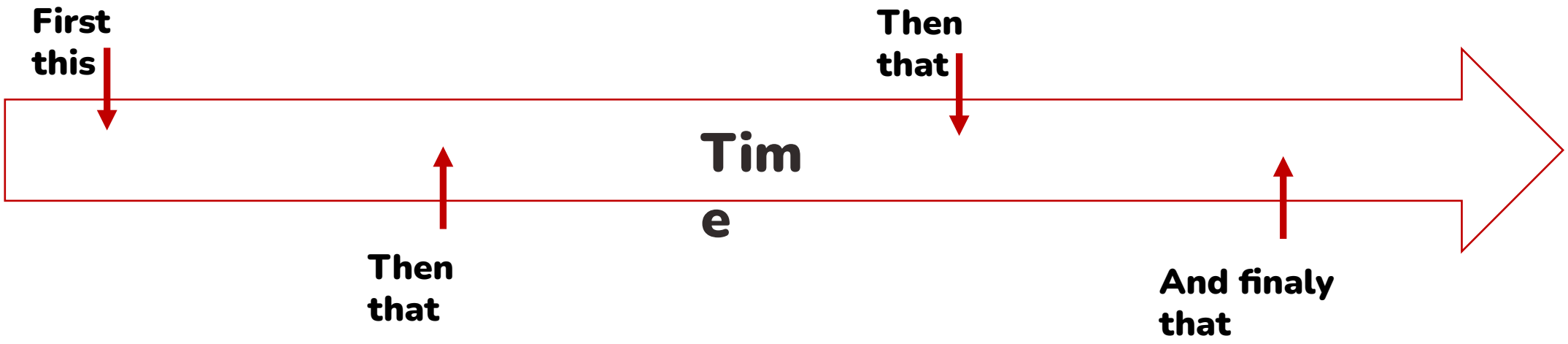
A discovery tool

A communication tool

A learning

What is a complex business domain?

A series of events organized on a time line



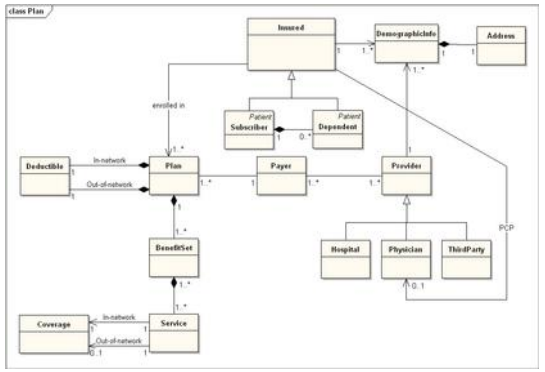
When does EventStorming apply?

Big picture

This Photo by Unknown Author is licensed under CC BY-NC

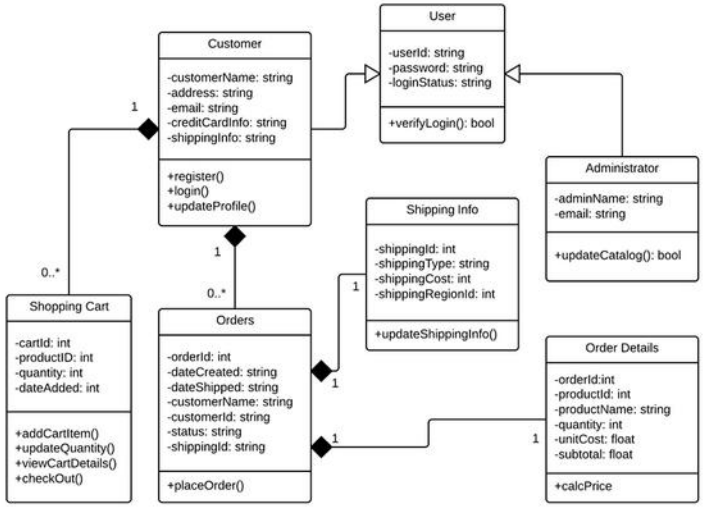
Process modeling

Software design



This Photo by Unknown Author is licensed under CC BY-SA

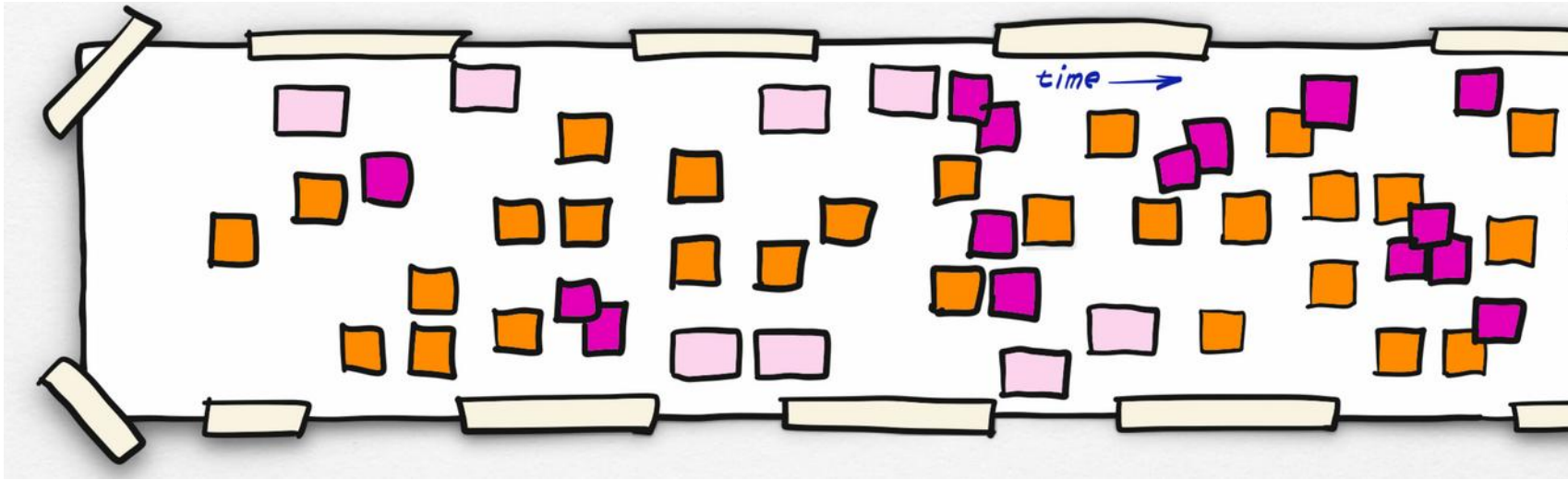
Explore a complex domain at different levels of abstraction



This Photo by Unknown Author is licensed under CC BY-SA

Structural modeling

Temporal modeling



And now…

This Photo by Unknown Author is licensed under CC BY

it's time for a **story**

Imagine
this is **you**

You are a **super hero**

and your
super power is...

EVENT
STORMING

Brandolini San, your **wise** and **revered** **master** has trained you well in the **art of EventStorming**

Alberto Brandolini

http://www.whoisagile.it/wp-content/uploads/2016/03/CaPVtRfWcAAFGSb.jpg

You have honed your skills in colorful stickies and sharpies

This Photo by Unknown Author is licensed under CC BY

This Photo by Unknown Author is licensed under CC BY-SA

And you always cary a paper roll and masking tape with you

This Photo by Unknown Author is licensed under CC BY-NC-ND

This Photo by Unknown Author is licensed under CC BY-SA

Just in case

One day, you receive a call from

This Photo by Unknown Author is licensed under CC BY-NC

They really need your help

They want to create a universal e-commerce system to sell all their gear and swag

This Photo by Unknown Author is licensed under CC BY-NC

But they don't know where to start

The Avengers form a great team

This Photo by Unknown Author is licensed under CC BY-SA

They divided responsibilities according to everyone's skills and abilities

They **outsourced** some work to **external consultants**

This Photo by Unknown Author is licensed under CC BY-SA

Well, not really that kind of consultants

Even though very competent at their job, they only see part of the problem at once

Thor

Sales

Marketing

Community management

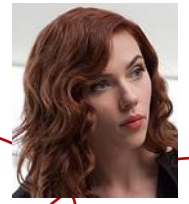
Communication

Peter Parker

Steve Rogers

Product Strategy

Natasha Romanoff



Customer acquisition

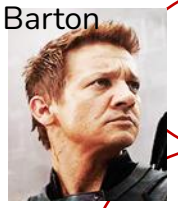
Growth hacking

Use eXperience

Wanda Maximooff

Knowledge is distribu

Clint Barton



Inventory

Shipping

Logistics

Carol Danvers

Stephen Strange

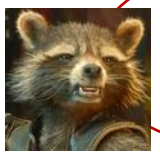
Tony Stark

Operations

Technology

Customer support

Infrastructure



Rocket Racoon

Nebula

Bruce Banner

Finance

Accounting

Selling and shipping products accross the universe is **not easy**

It's by far the **hardest task** they have ever been assigned

This Photo by Unknown Author is licensed under CC BY-NC

Even harder than killing **that guy**

This Photo by Unknown Author is licensed under CC BY-SA

This Photo by Unknown Author is licensed under CC BY-NC

Despite all their powers,

bright minds, and wit

They are unable to reach a common undestanding of what they want

They lack a clear vision
(pun intended)

This Photo by Unknown Author is licensed under CC BY-NC

They need a platfrom that is

Easy to use for both humans and aliens

This Photo by Unknown Author is licensed under CC BY-SA

This Photo by Unknown Author is licensed under CC BY

This Photo by Unknown Author is licensed under CC BY-SA-NC

Ubiquitous, able to cope with multiple cultures, languages and currencies

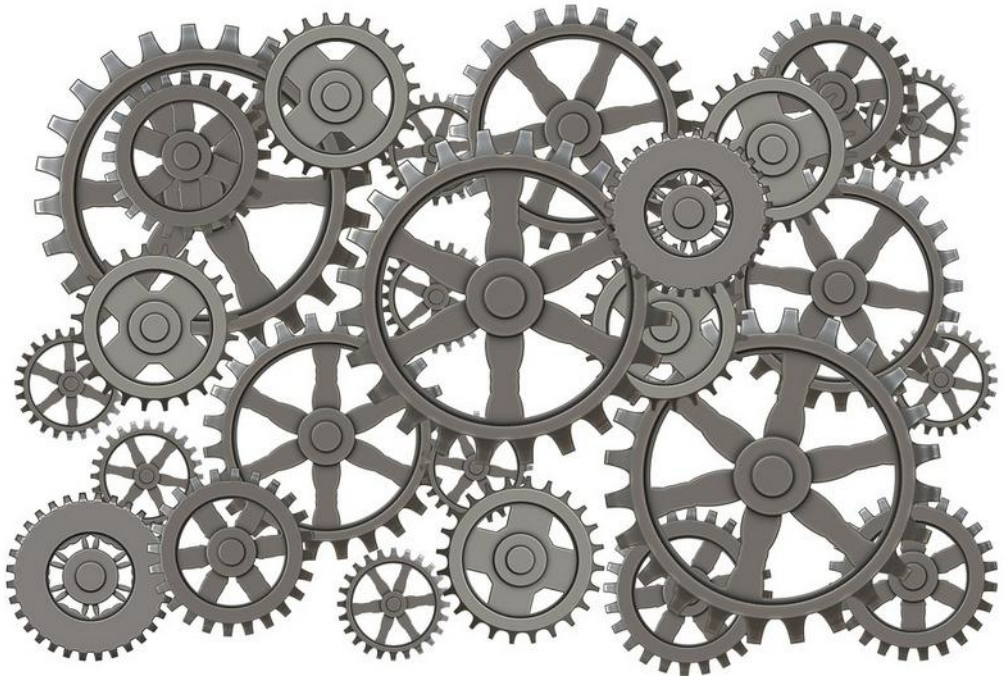
This Photo by Unknown Author is licensed under CC BY-SA

Capable of **scaling** to handle **high demand**

This Photo by Unknown Author is licensed under CC BY-SA

Always **available** and **secure**

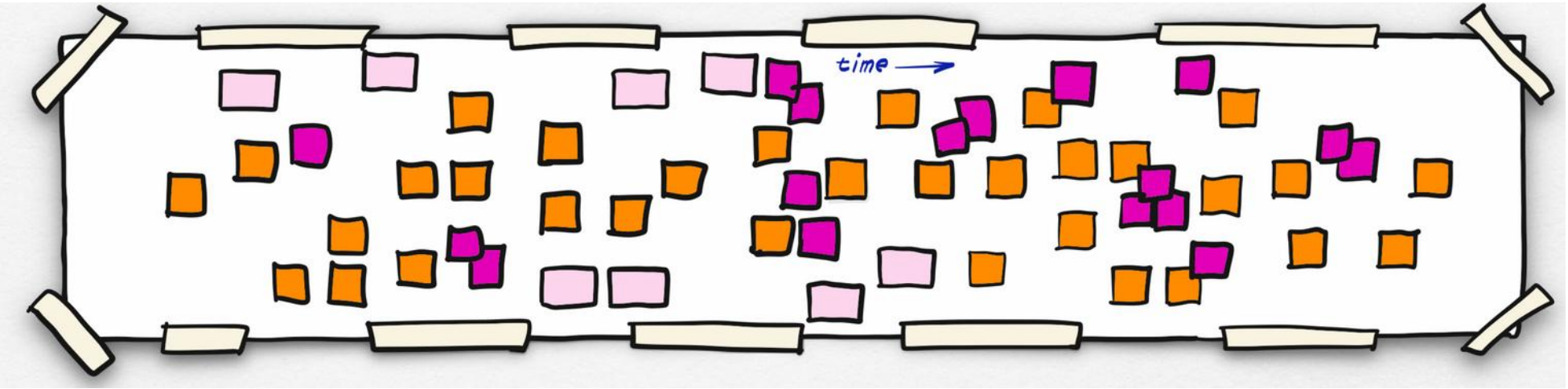
To make sense of that complexity



You decide to organize a
Big Picture
EventStorming

PRO TIP

Use
Big Picture EventStorming
to make sense of a **complex domain** and see it as a whole



It will provide a **global overview** across **corporate silos** and trigger long **over due conversations**

This Photo by Unknown Author is licensed under CC BY-SA

When you arrive at the Avengers facility

On a Quinjet...

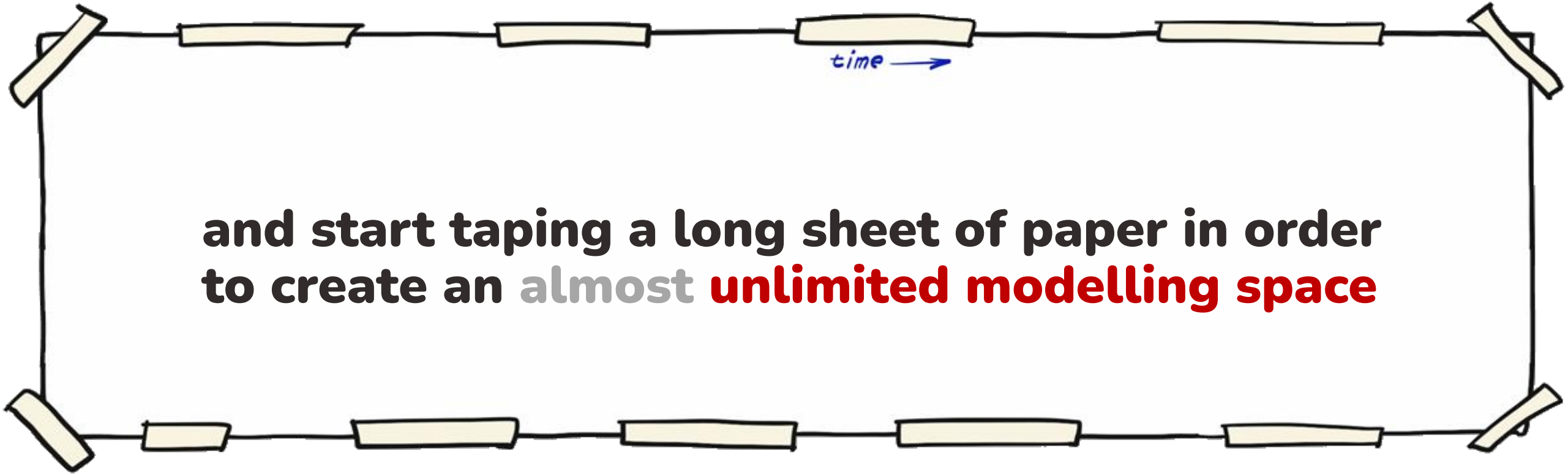
https://marvelcinematicuniverse.fandom.com/wiki/Quinjet

You find a big room and clean up the space by removing all tables and chairs

You need people to be engaged

You want them to actively collaborate, not sit down passively

You spot a nice empty wall on the left



and start taping a long sheet of paper in order
to create an almost unlimited modelling space

You get your **secret weapon** ready

You need **stationery**... plenty of it

And prepare a visible legend

You know you will need **everyone on deck**

So you invite **all** the **key people** to join the workshop

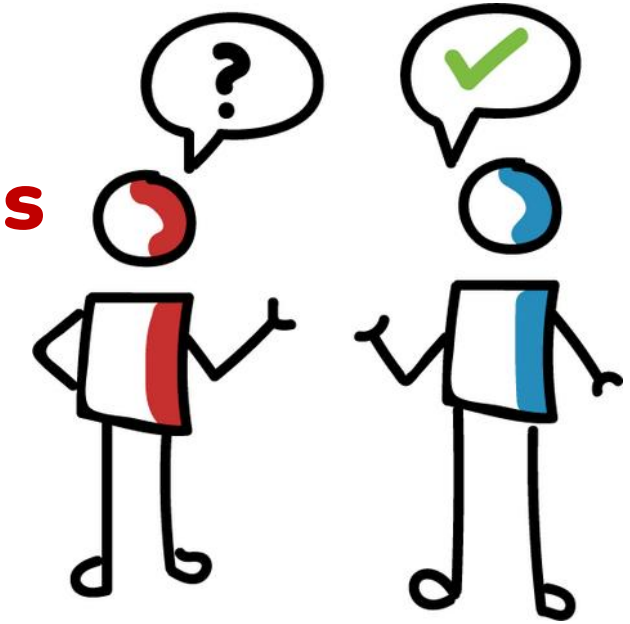
This Photo by Unknown Author is licensed under CC BY-NC-ND

PRO TIP

In order to run a successful **EventStorming** workshop, you need to **invite the right people**

People with questions

- Solution architects
- Software developers
- Change agents
- DevOps



People with answers

- Decision makers
- Business owners
- Domain experts
- Business analysts

Before you begin, you provide some context

Because he is the leader, you ask Steve to set the stage and give a bit of context

Then you remind everyone that the goal for now is to explore the domain in order to understand the big picture

This Photo by Unknown Author is licensed under CC BY-SA

You can then explain some basic rules

One person, one marker

One concept per sticky

Write in CAPITAL letters

This Photo by Unknown Author is licensed under CC BY-NC

This Photo by Unknown Author is licensed under CC BY-SA

Enforce the timeline

Make all the rules explicit

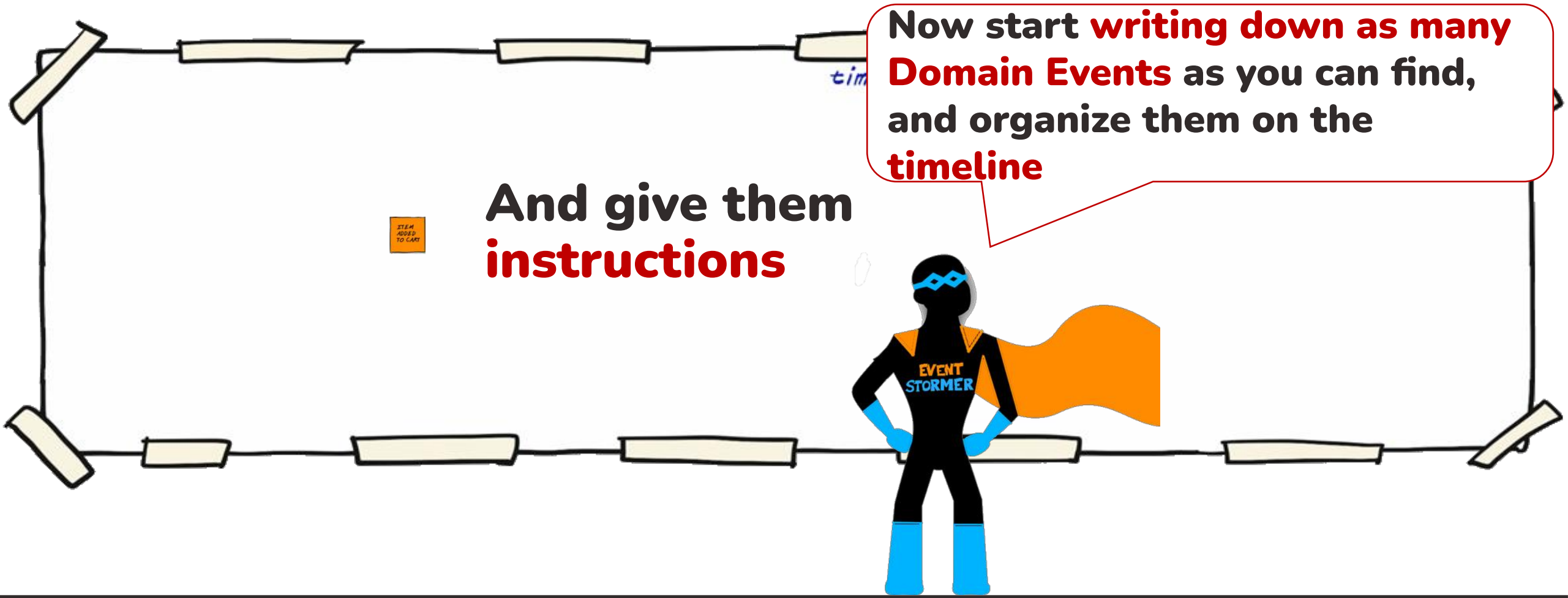
Explain what is a

Domain Event
and add it to the

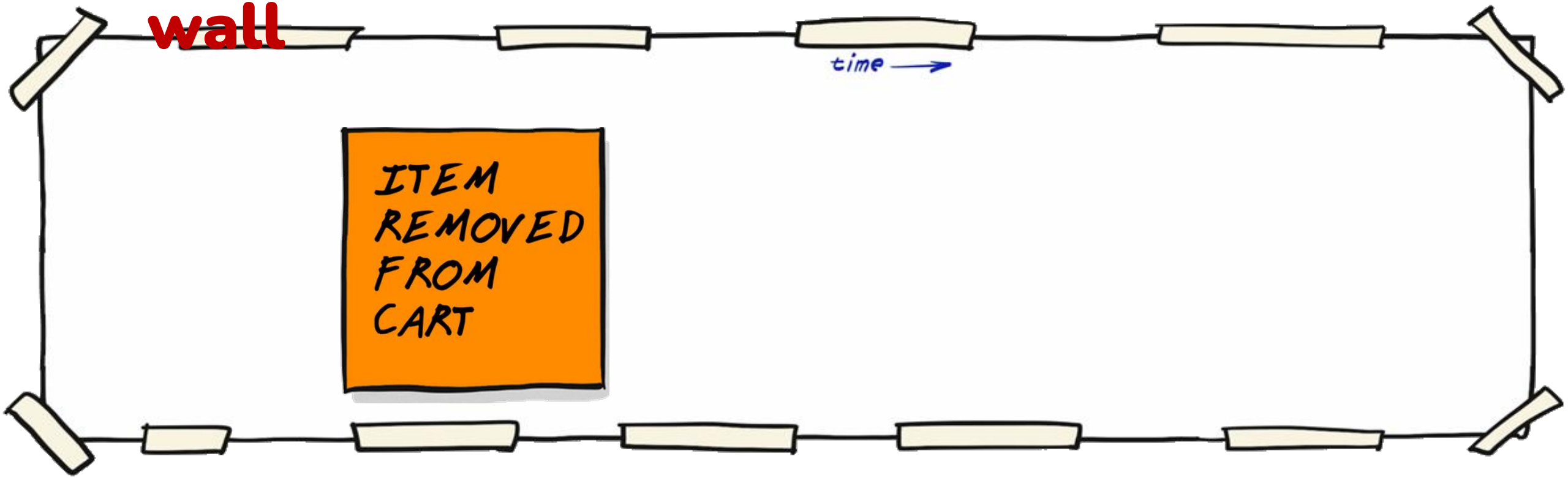
visible legend

Write down the first Domain Event

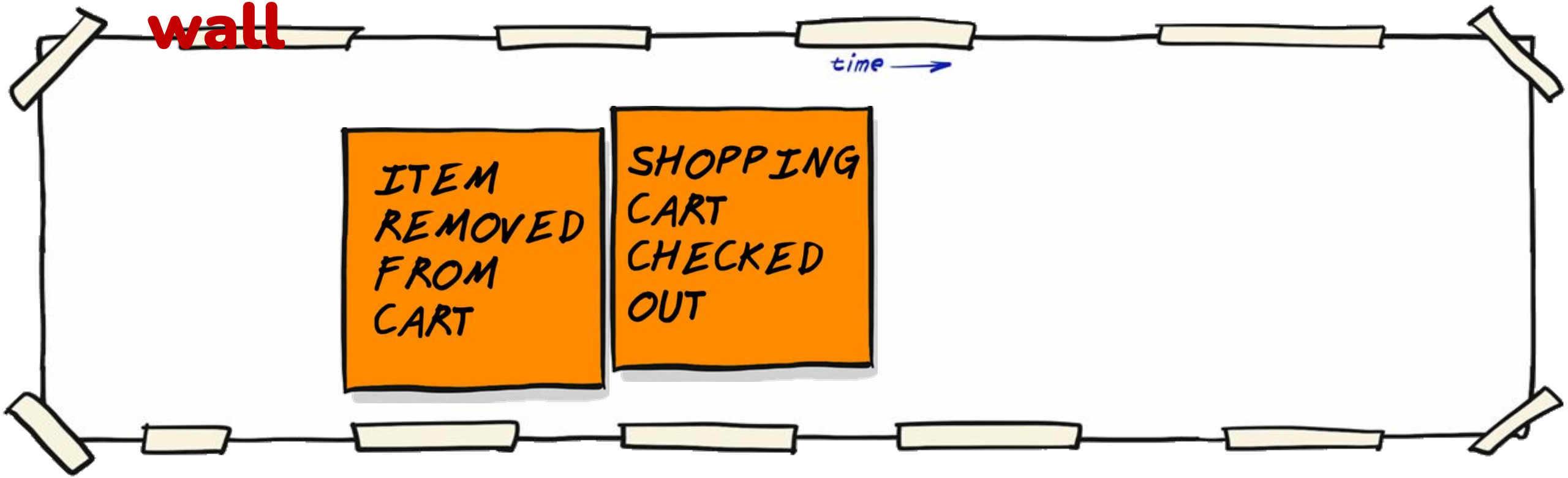
Then stick it in the first third of the **modeling space**



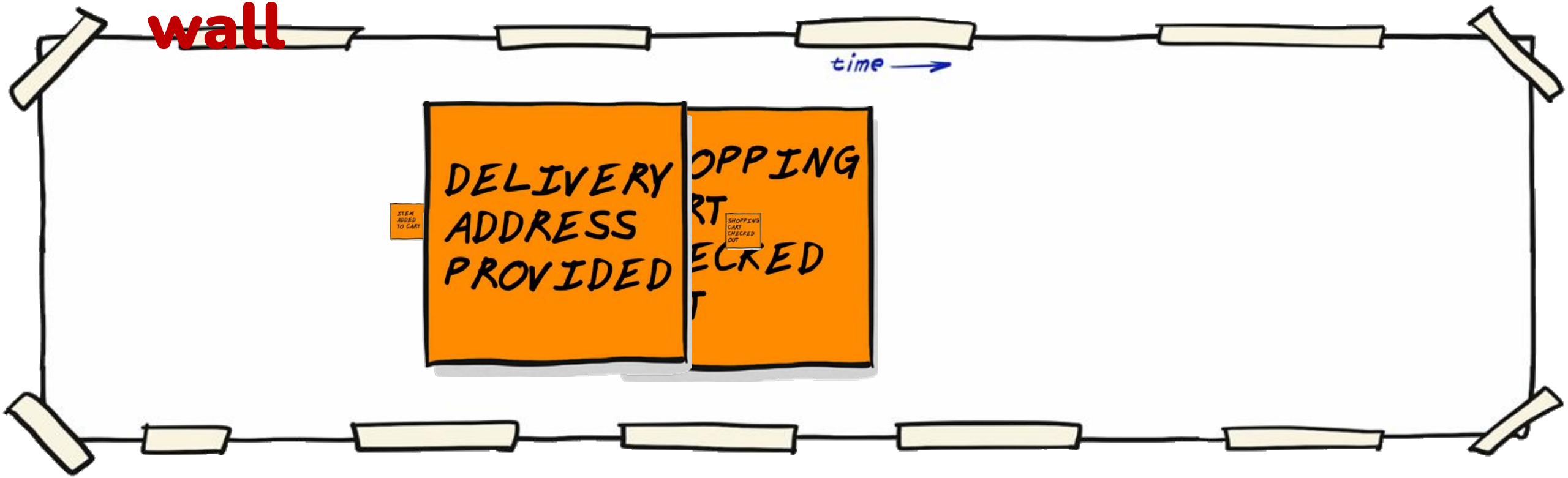
At that point, everyone should start to put stickies on the wall



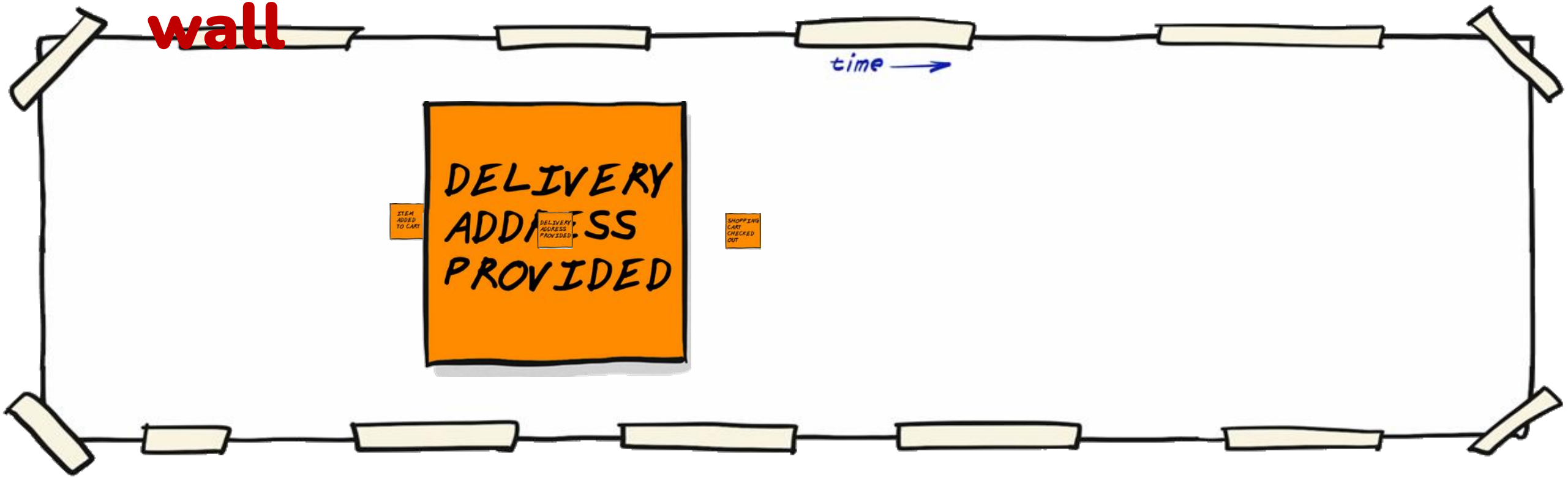
At that point, everyone should start to put stickies on the wall



At that point, everyone should start to put stickies on the wall



At that point, everyone should start to put stickies on the wall



Wait… Aren't we **missing** something here?

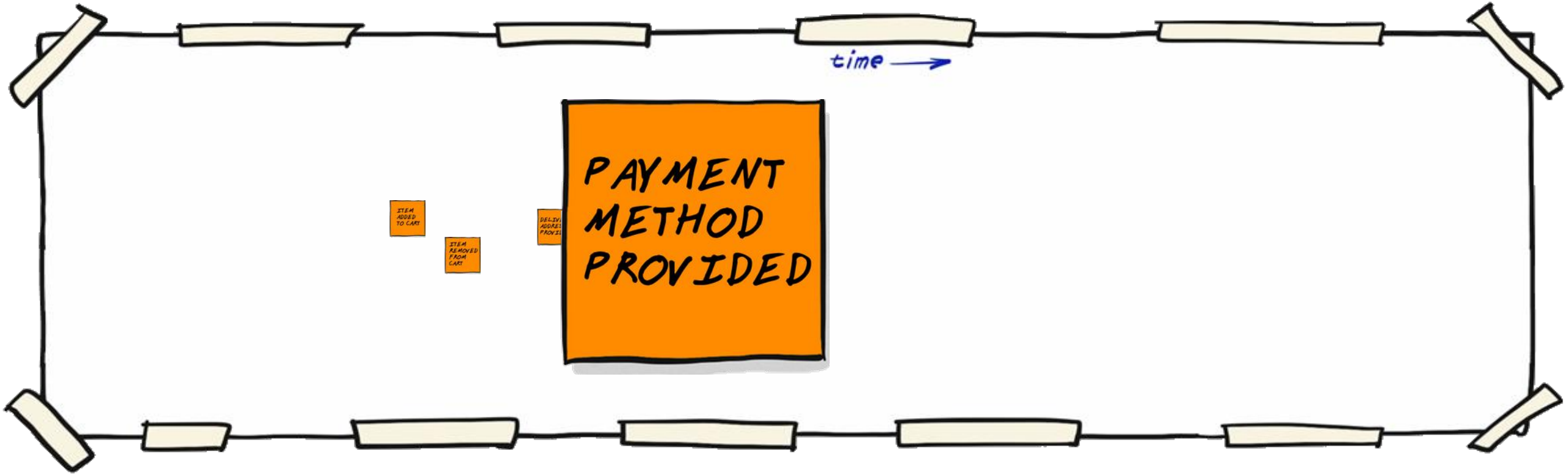
Yes! What about the **money**?

PRO TIP

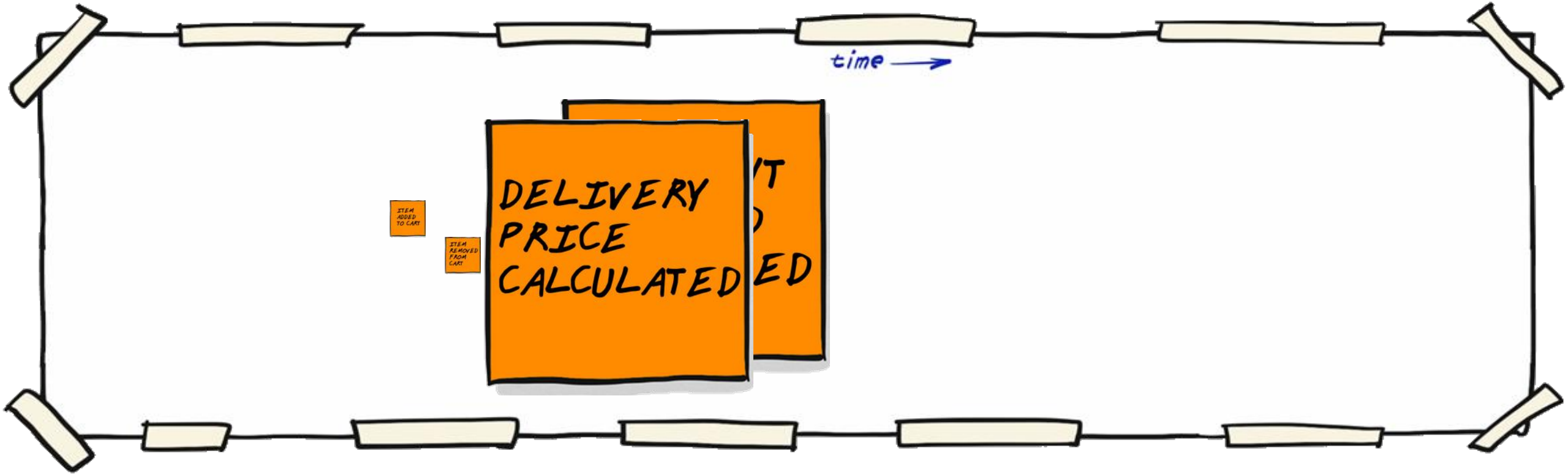
When in doubt, always follow the money

The way money flows usually triggers a lot of very interesting discussions

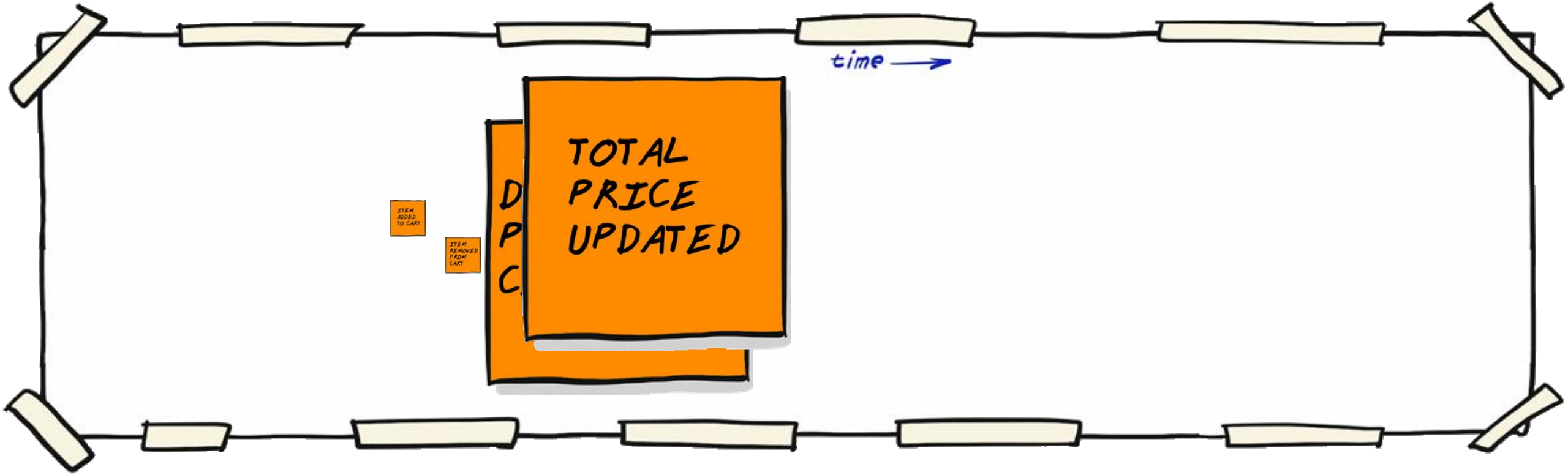
Follow the money



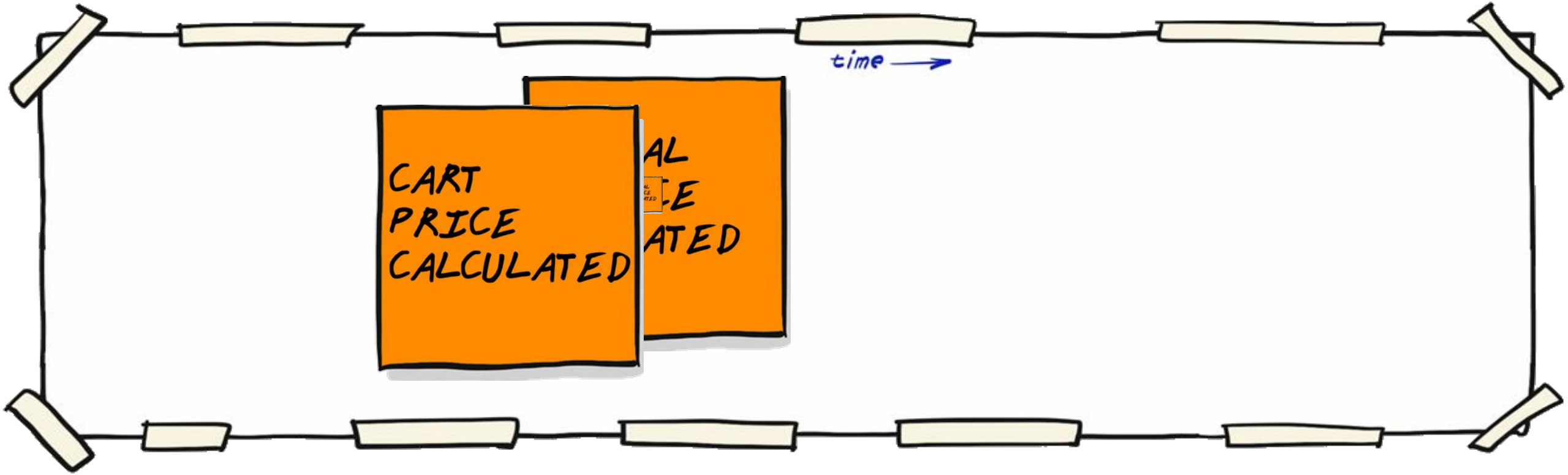
Follow the money



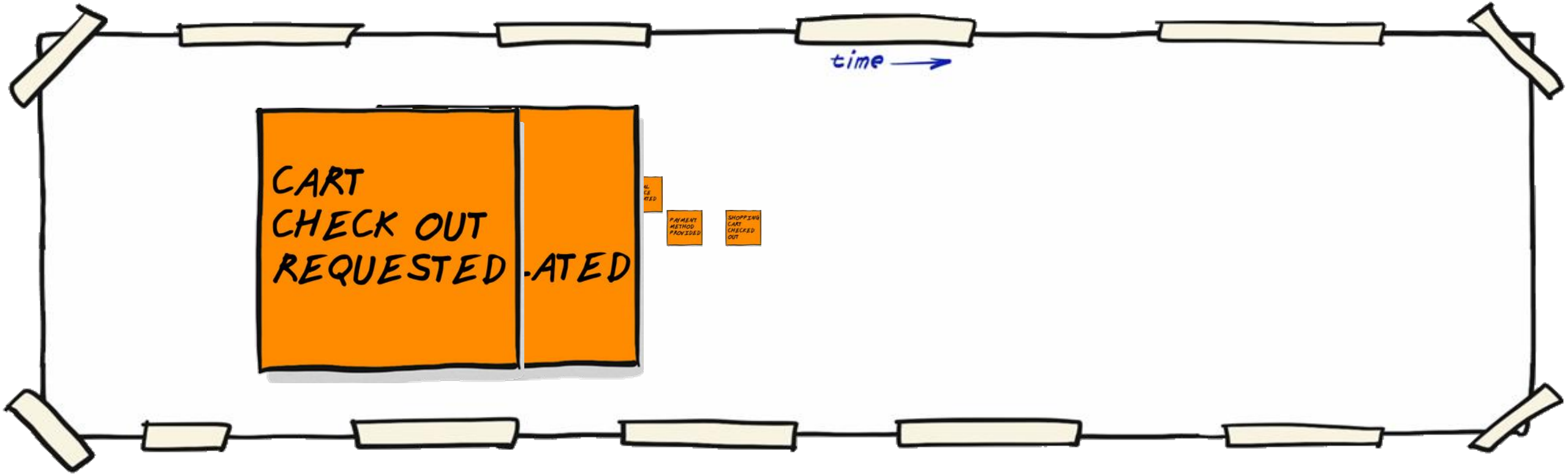
Follow the money



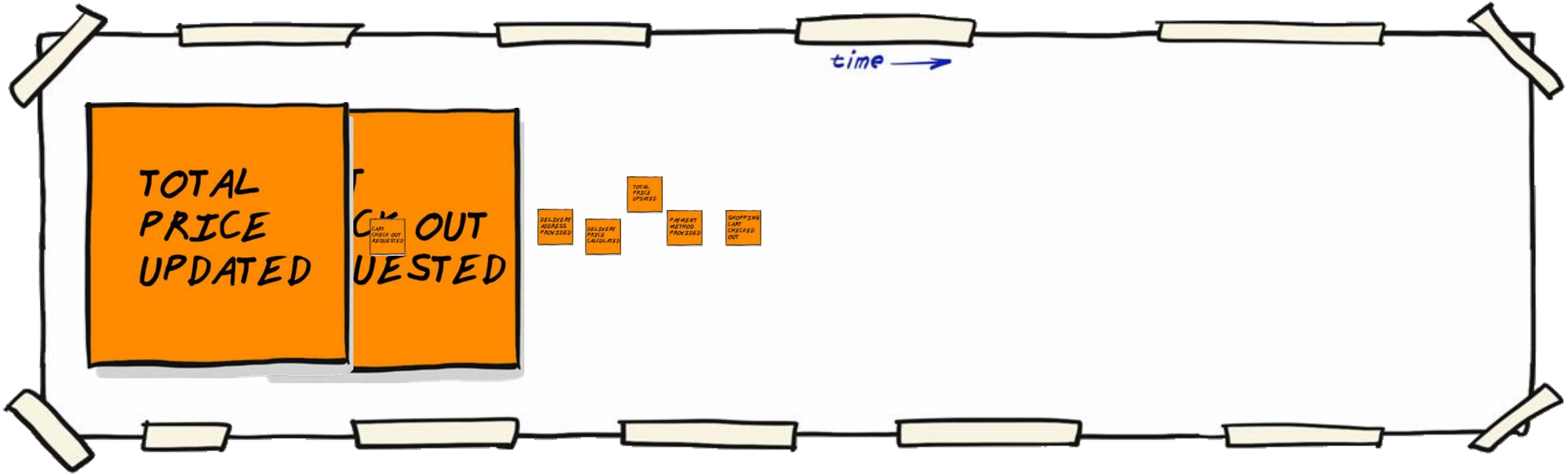
Follow the money



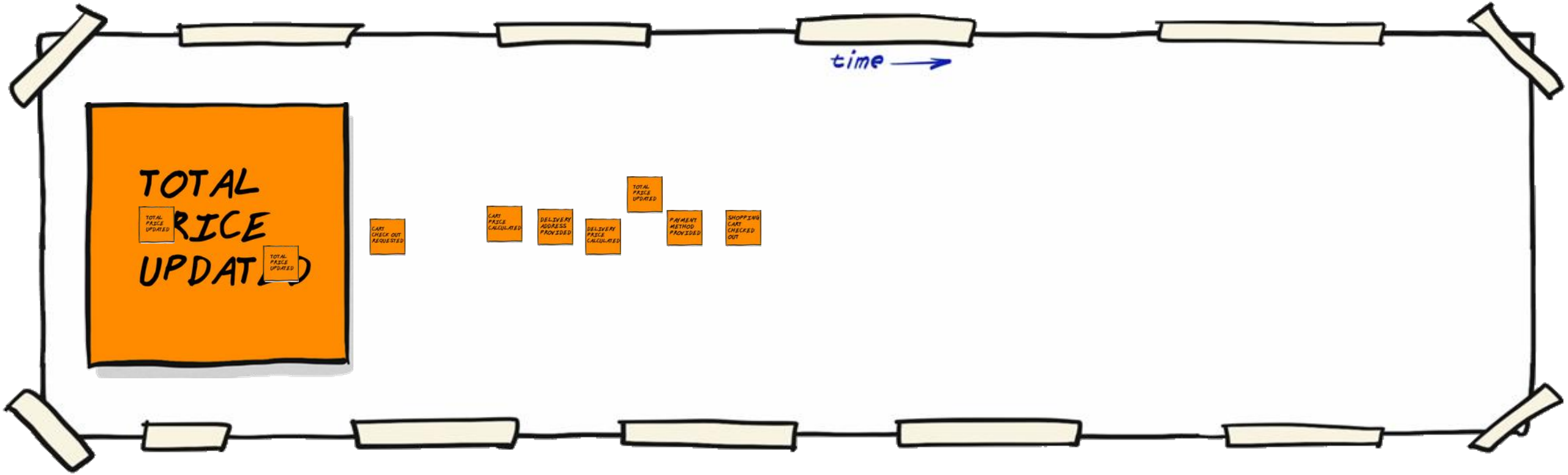
Follow the money



Maintain the timeline



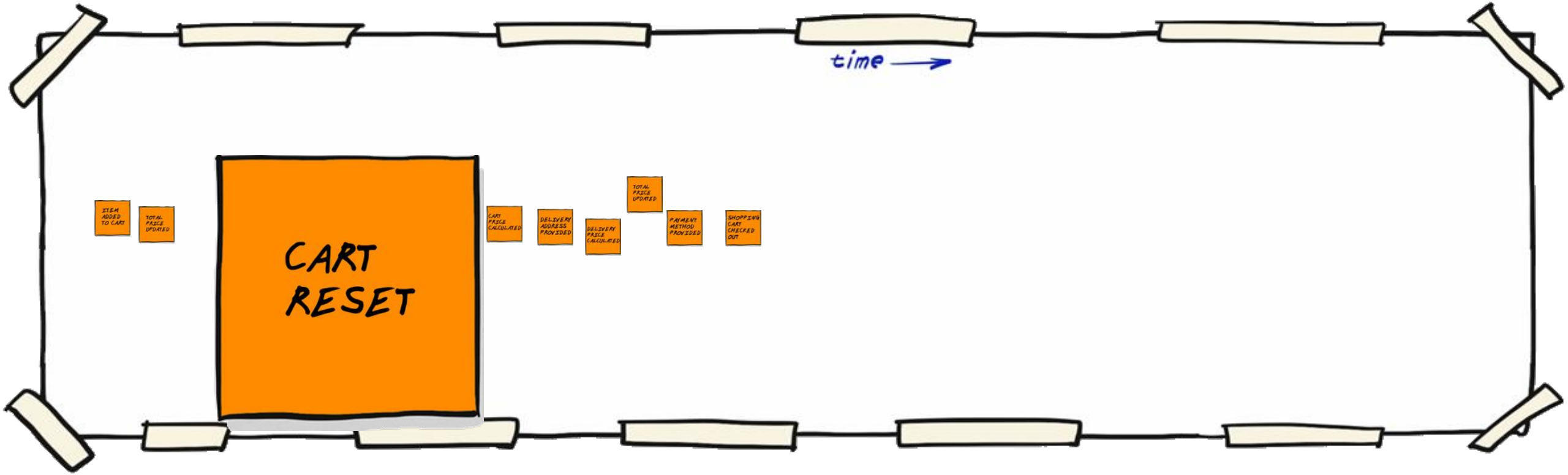
Maintain the timeline



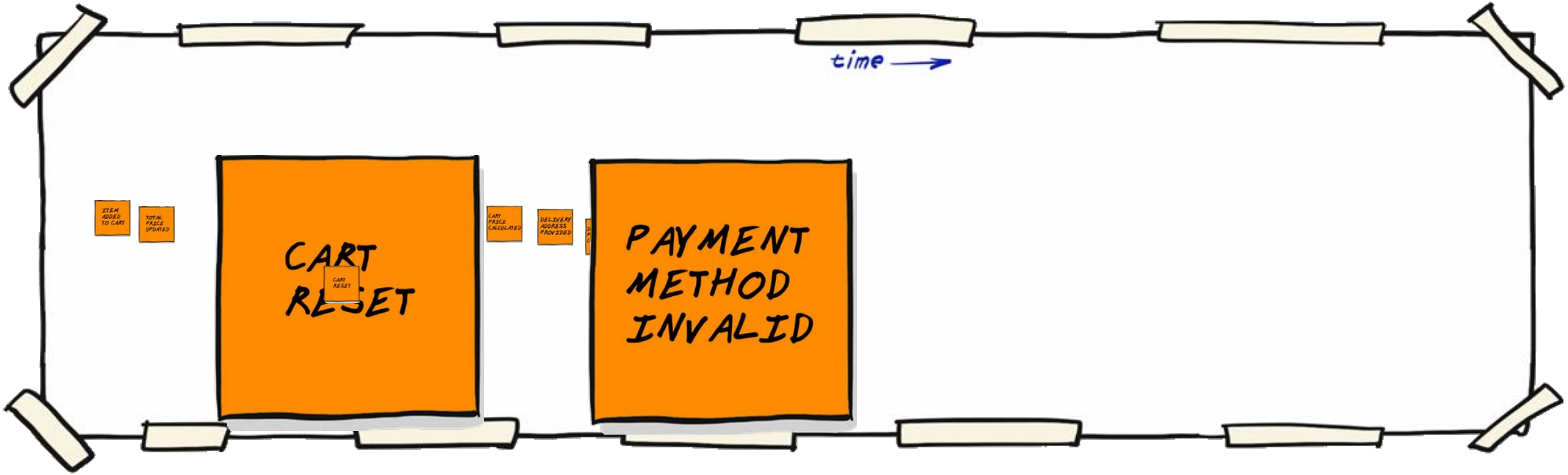
Mmm... Seems we've only followed the **happy path** so far

You're right. From now on, let's also consider what happens when **something goes wrong**.

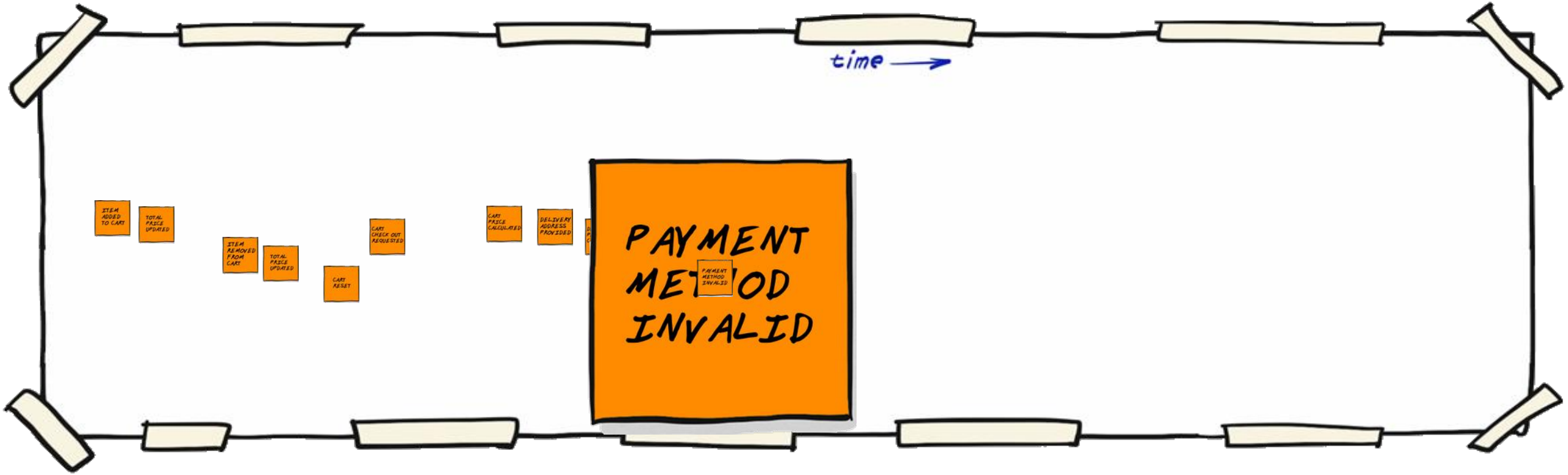
The not so happy path



The not so happy path



The not so happy path



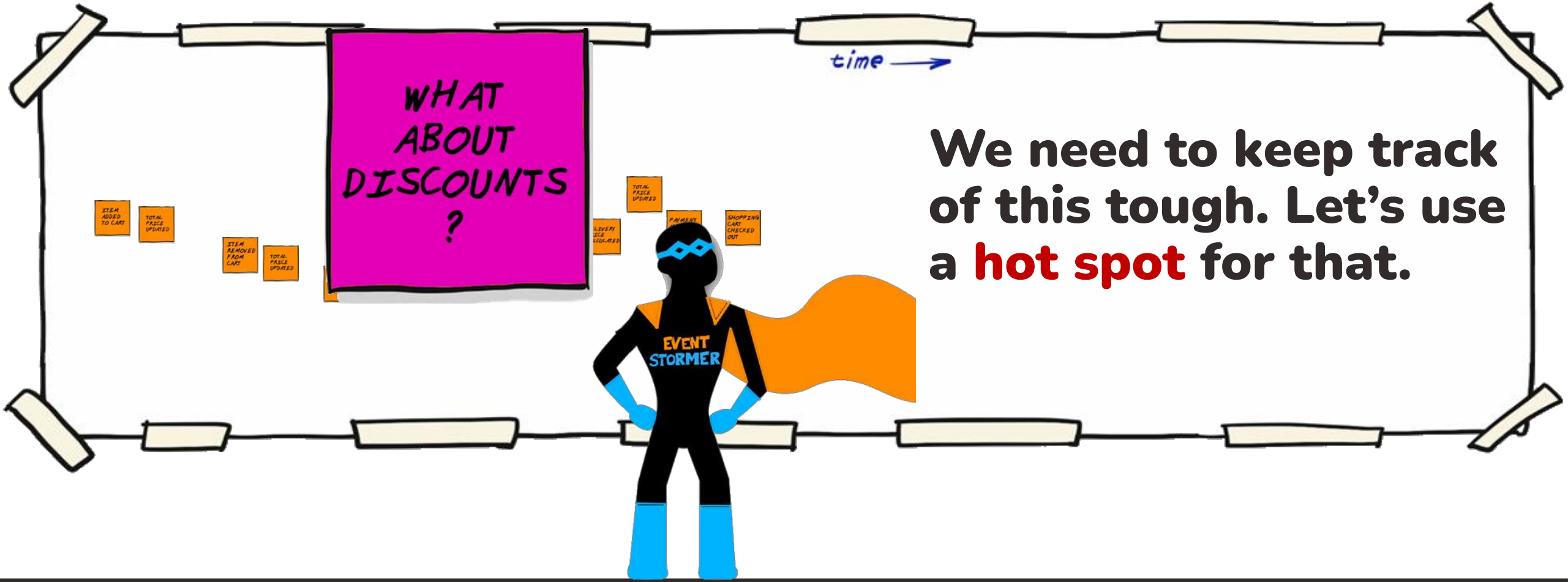
What about **discounts** or **vouchers**?
Do we want any of that?

It's probably **too early** to decide on that.

It would add a lot of **complexity.**

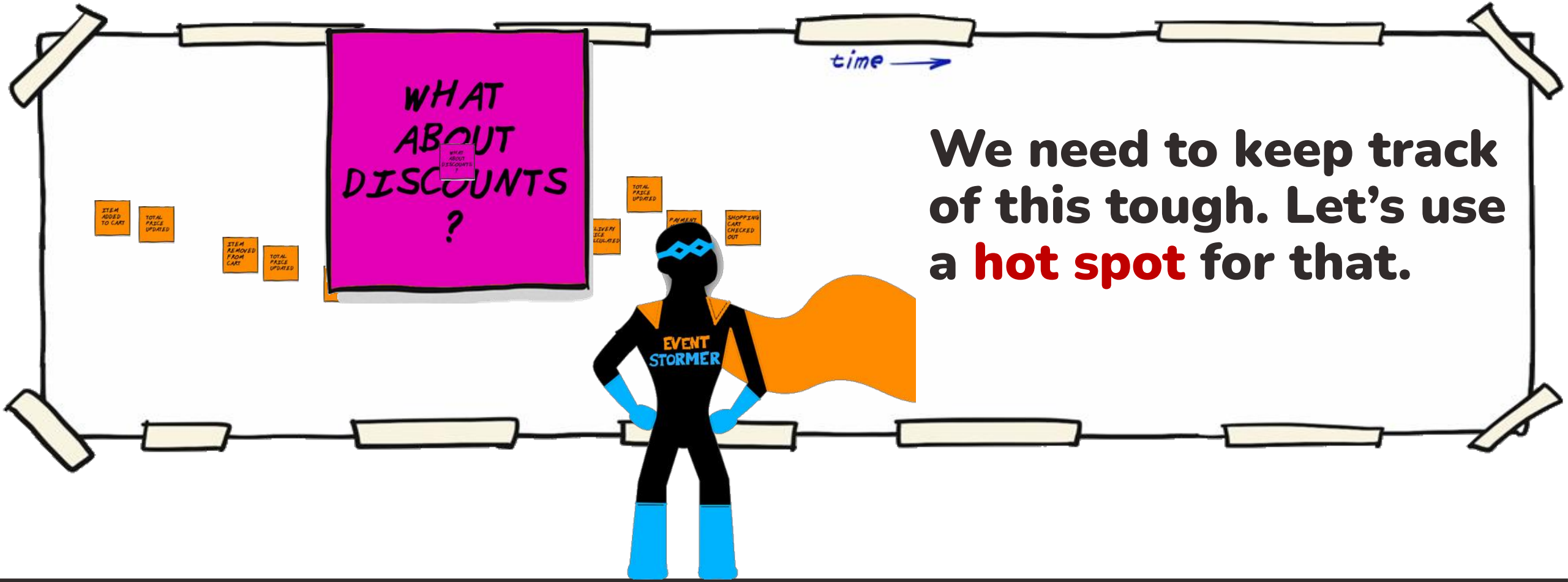
True, but it would be useful to **attract people** on the platform.

We can delay this decision for later



We need to keep track of this tough. Let's use a hot spot for that.

We can delay this decision for later

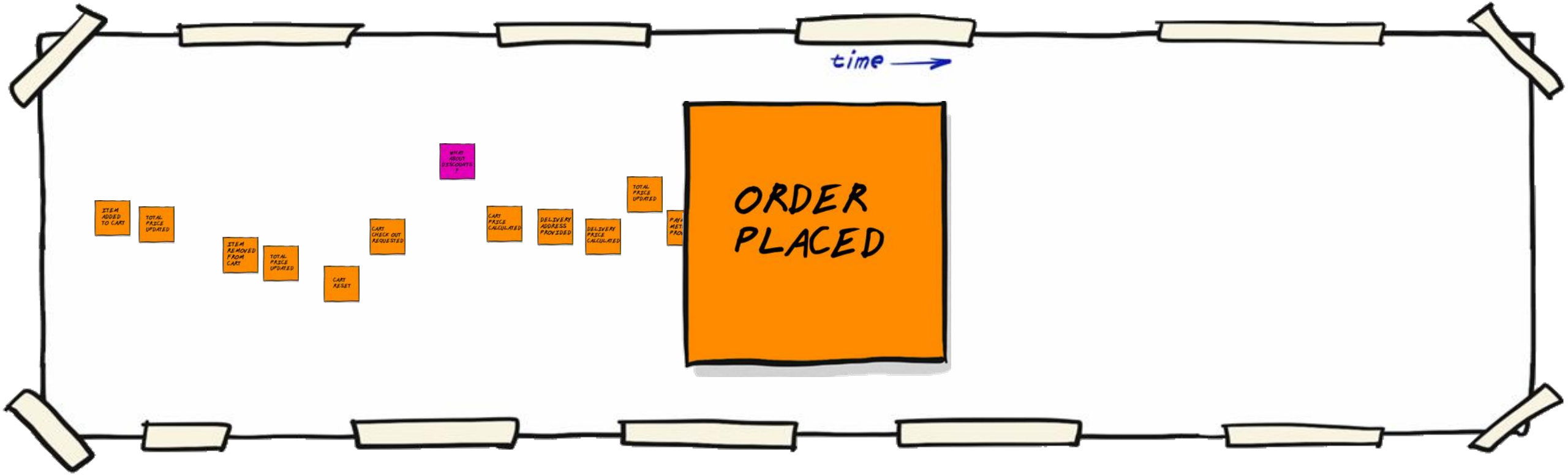


We need to keep track of this tough. Let's use a hot spot for that.

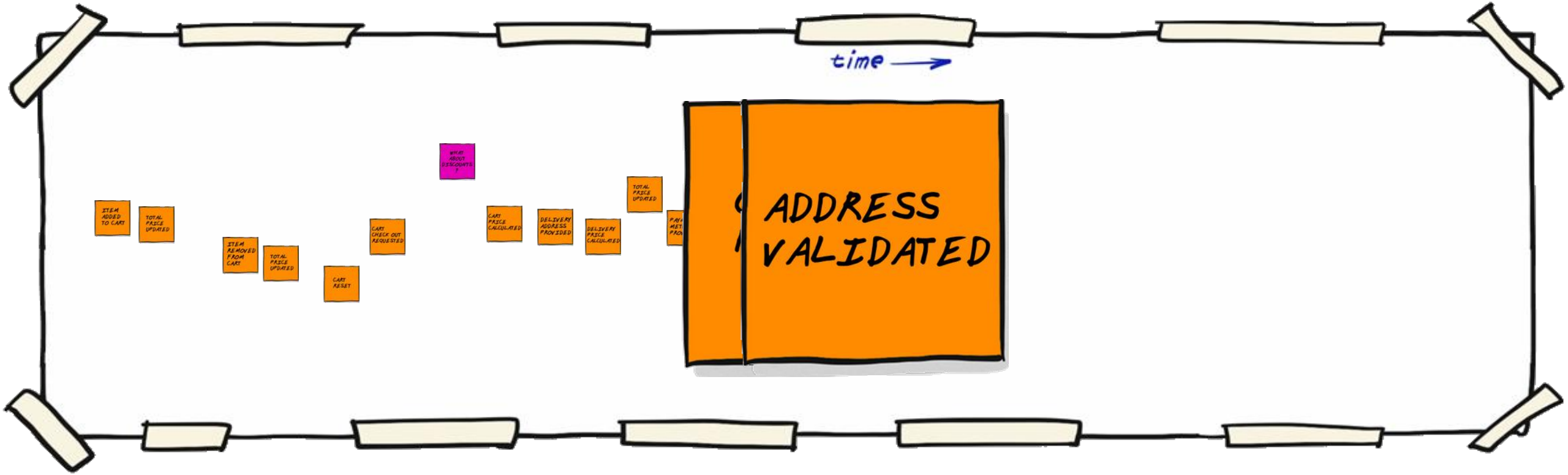
Add the **Hot Spot** sticky to the **legend**

We are using an incremental notation

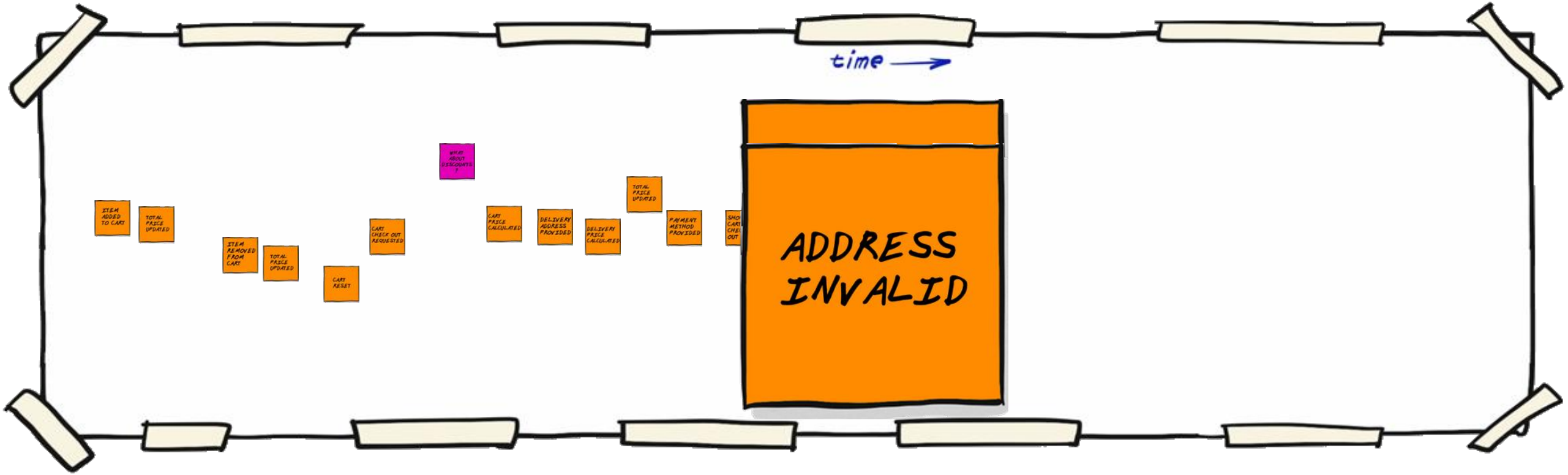
Now... what happens **after** the cart is **checked out**



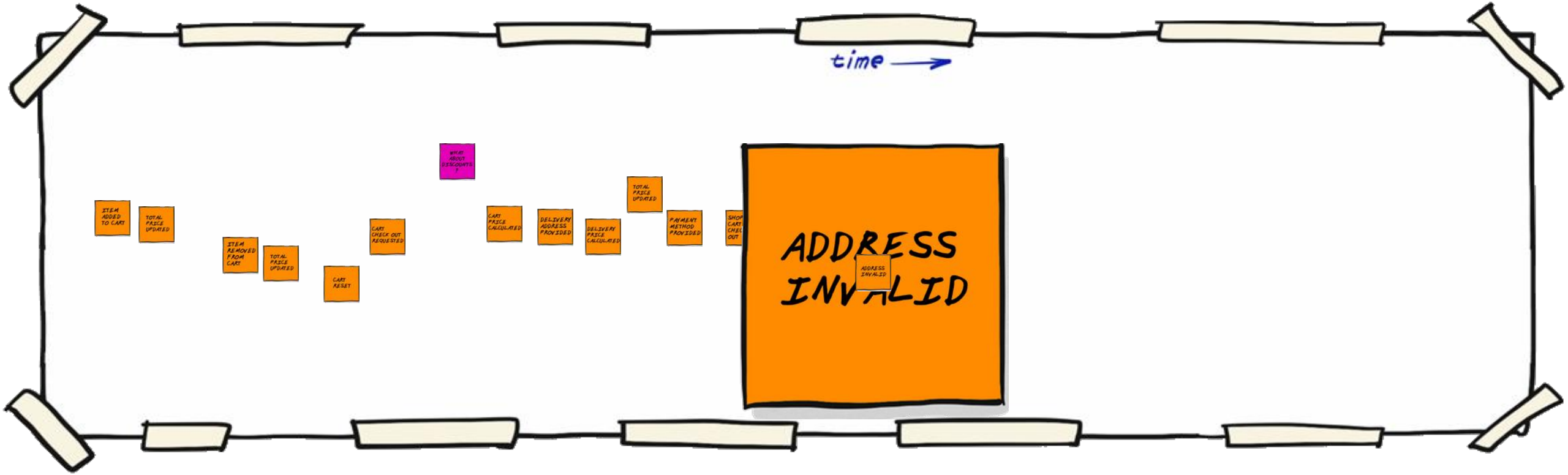
First we need a valid address



Or not

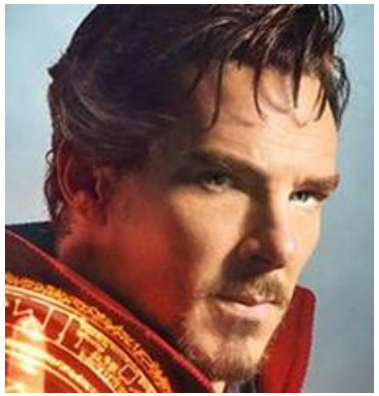

time
ITEM ADDED TO CART
TOTAL PRICE UPDATED
ITEM REMOVED FROM CART
TOTAL PRICE UPDATED
CART RESET
CART CHECK OUT REQUESTED
WHAT ABOUT DISCOUNTS?
CART PRICE CALCULATED
DELIVERY ADDRESS PROVIDED
DELIVERY PRICE CALCULATED
TOTAL PRICE UPDATED
PAYMENT METHOD PROVIDED
ADDRESS INVALID

Or not


time
ITEM ADDED TO CART
TOTAL PRICE UPDATED
ITEM REMOVED FROM CART
TOTAL PRICE UPDATED
CART RESET
CART CHECK OUT REQUESTED
WHAT ABOUT DISCOUNTS?
CART PRICE CALCULATED
DELIVERY ADDRESS PROVIDED
DELIVERY PRICE CALCULATED
TOTAL PRICE UPDATED
PAYMENT METHOD PROVIDED
ADDRESS INVALID
ADDRESS INVALID

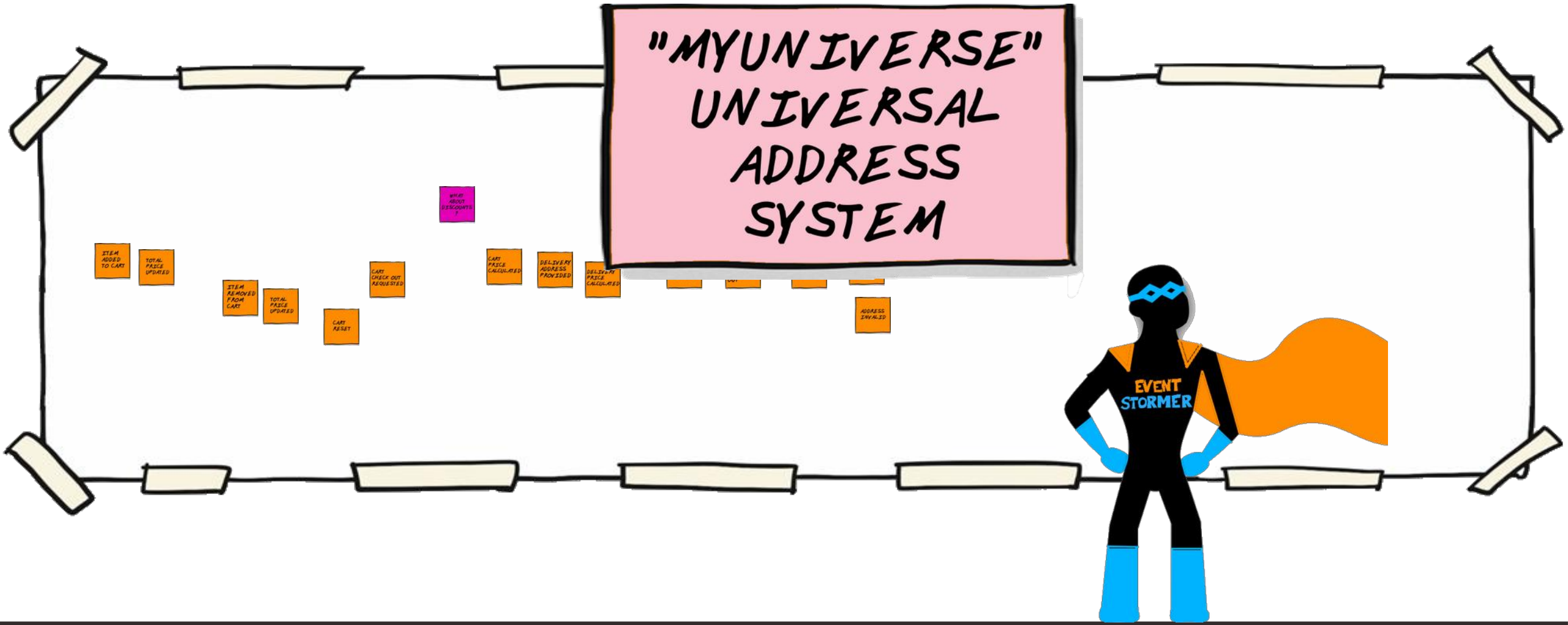
Em... how do we **validate** the address? Against what **source of truth**?

Good question. The **universe** is a pretty **big** place...

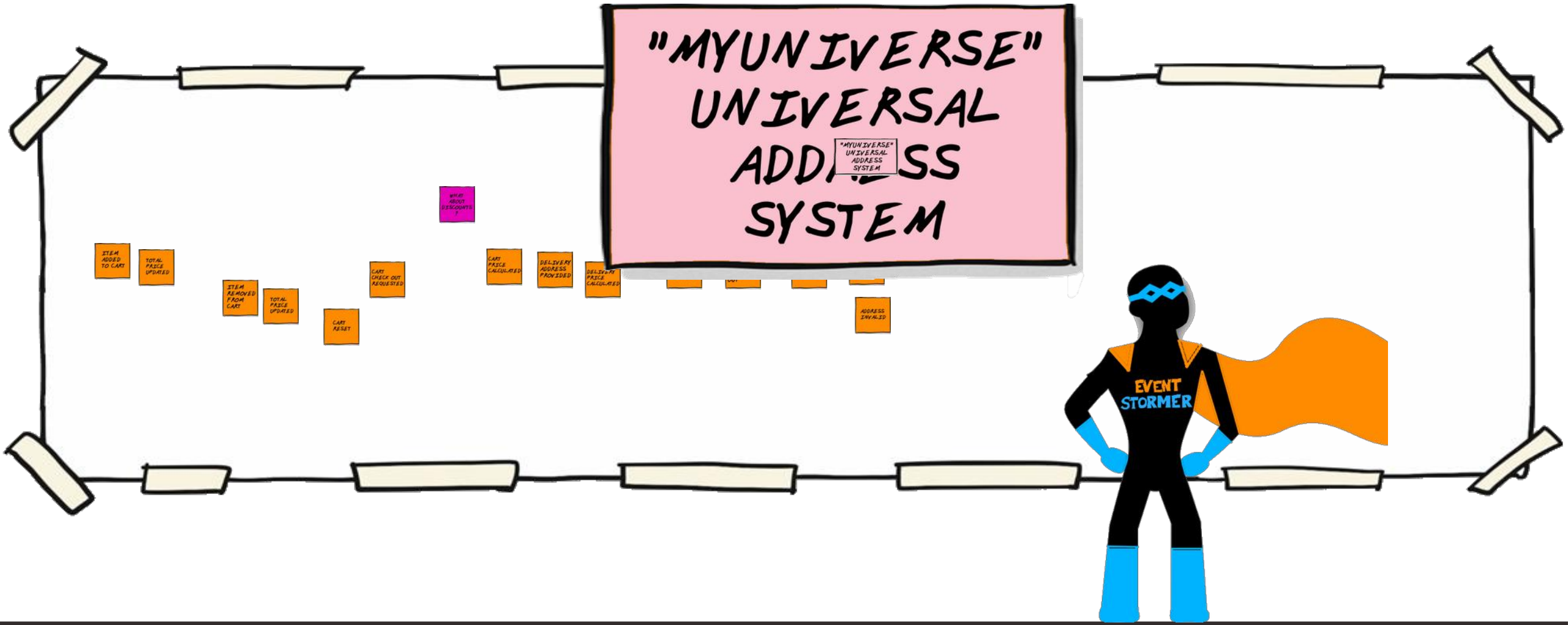


We could use an external repository called **MyUniverse** to make sure an **address is valid**

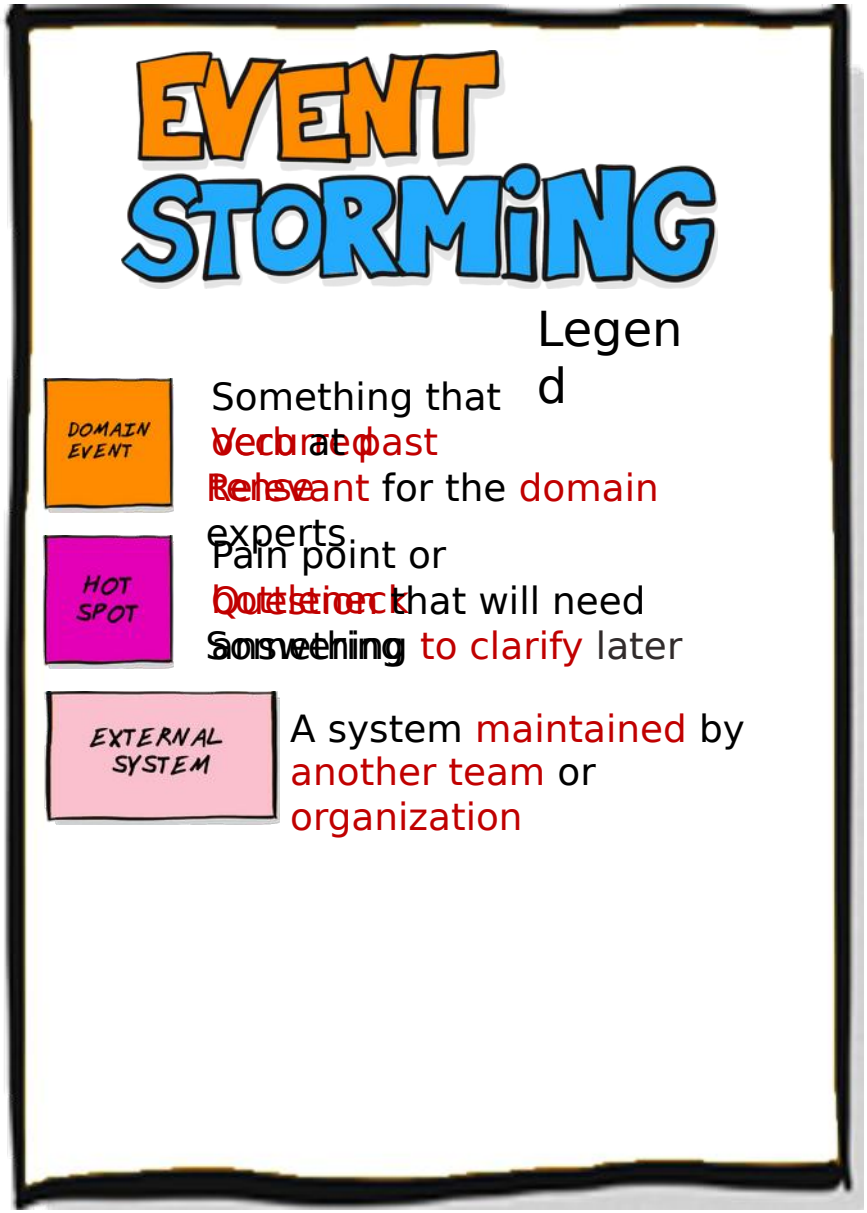
Let's increment our notation again



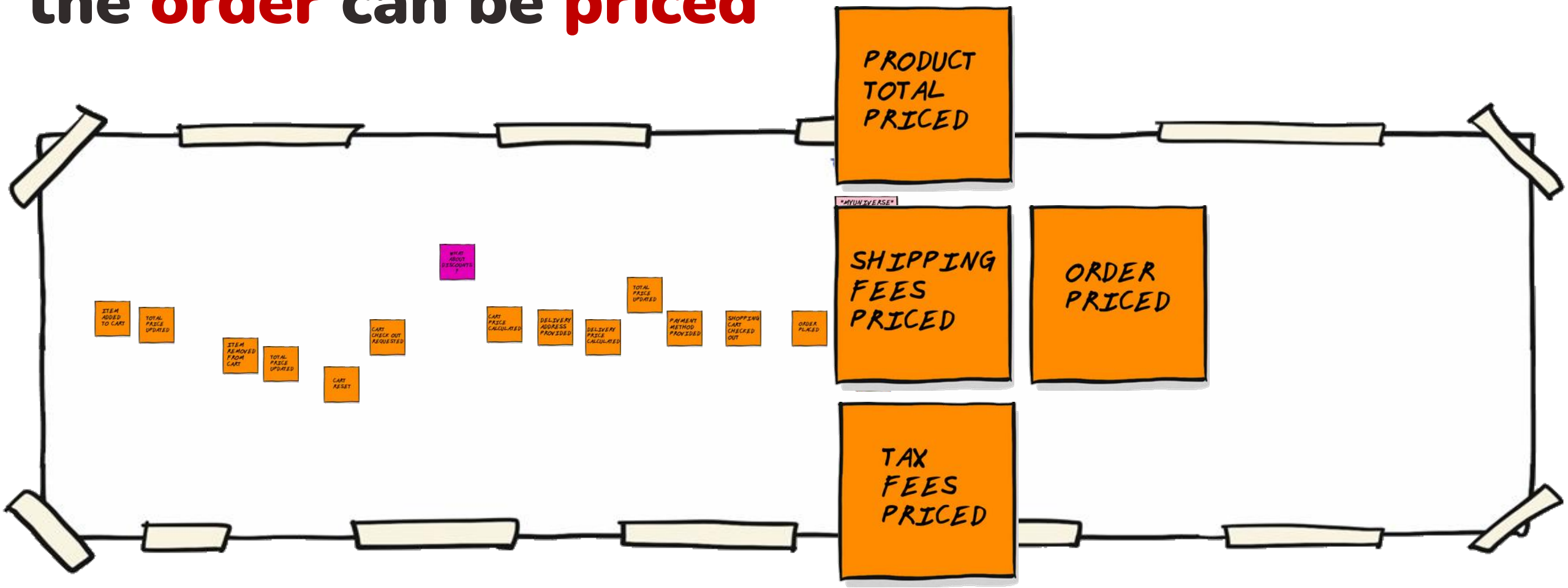
Let's increment our notation again



Add the **External System** sticky to the **legend**

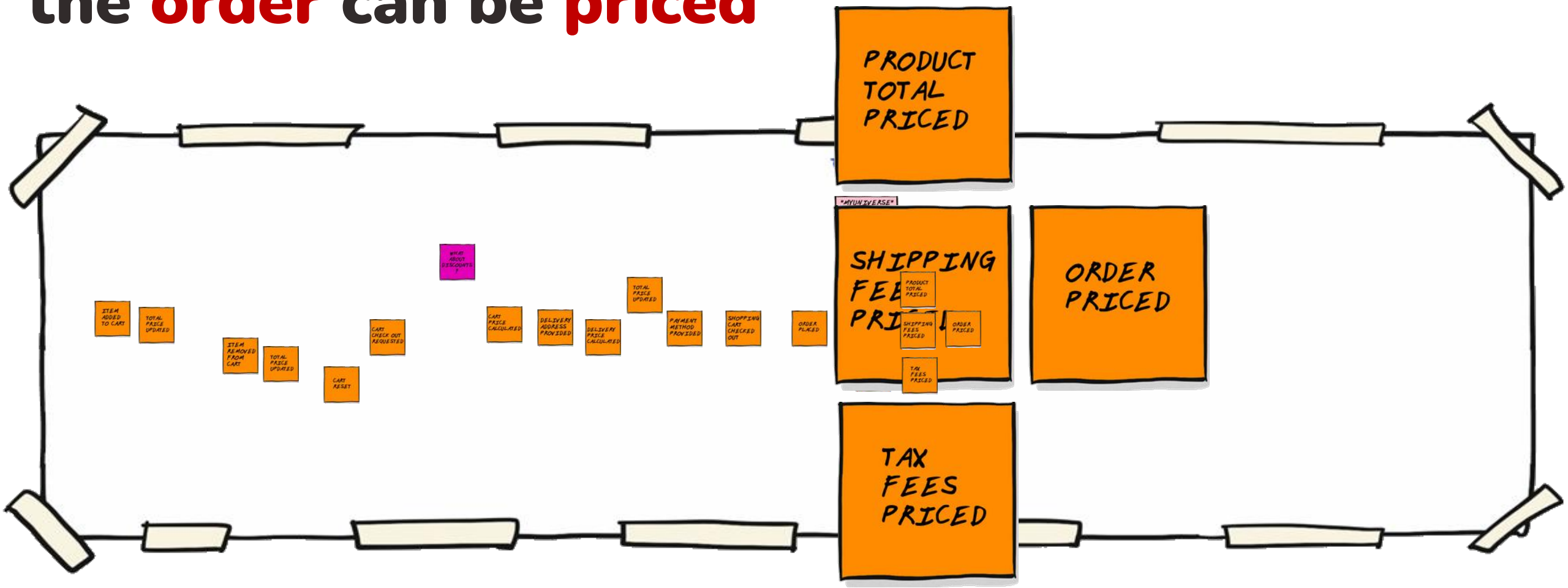


With a valid address, the order can be priced



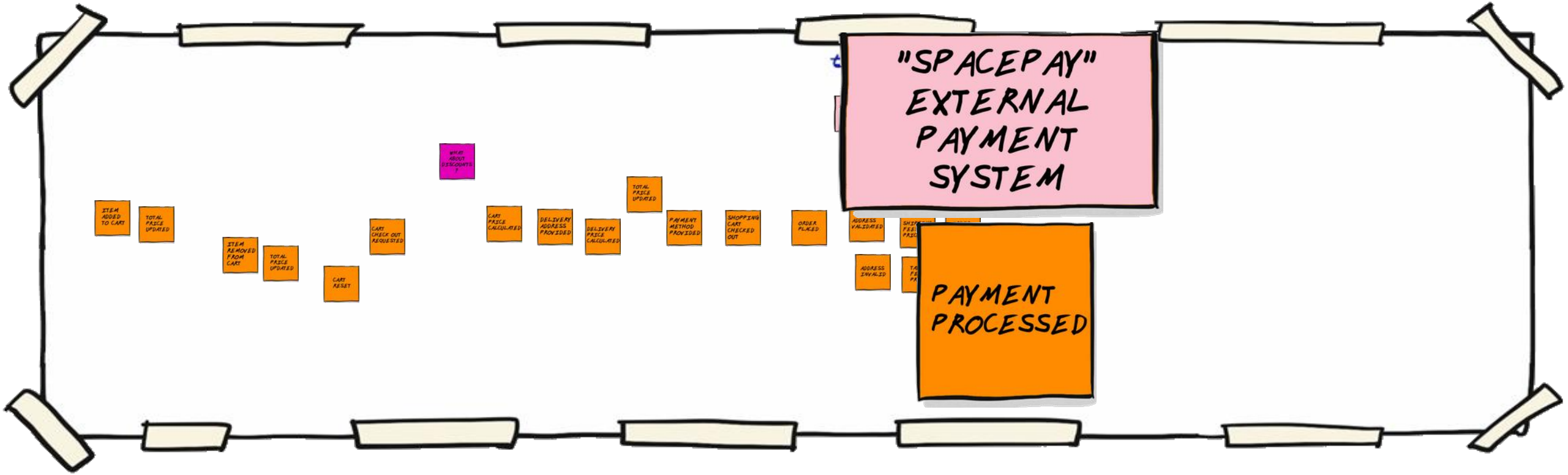
Including shipping and tax fees

With a valid address, the order can be priced

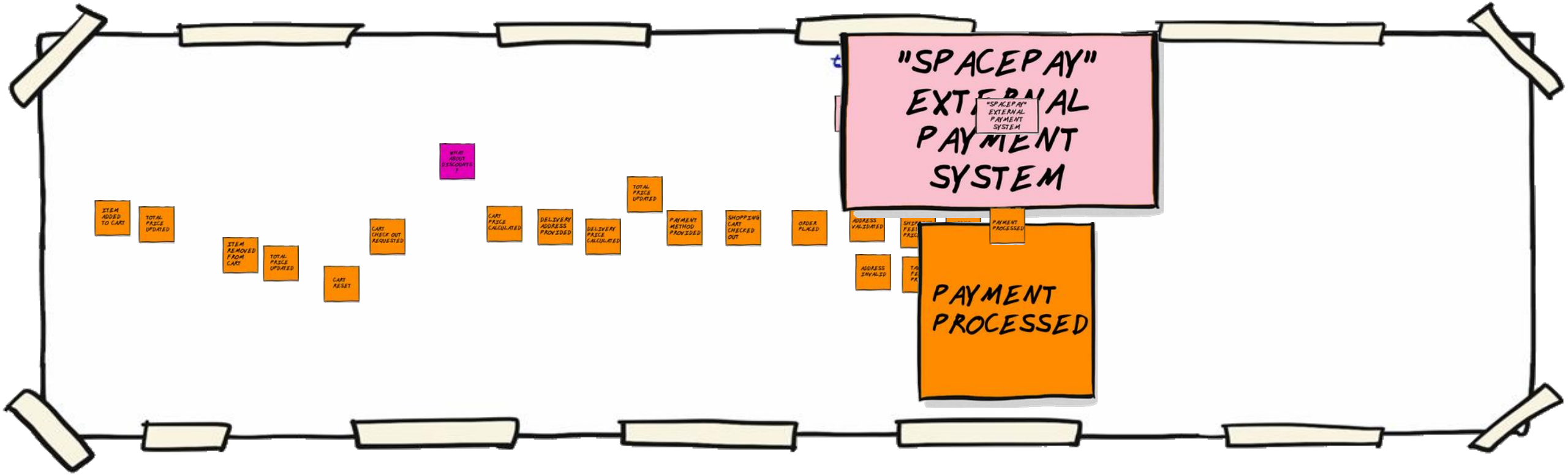


Including shipping and tax fees

Now, let's proceed with payment



Now, let's proceed with payment



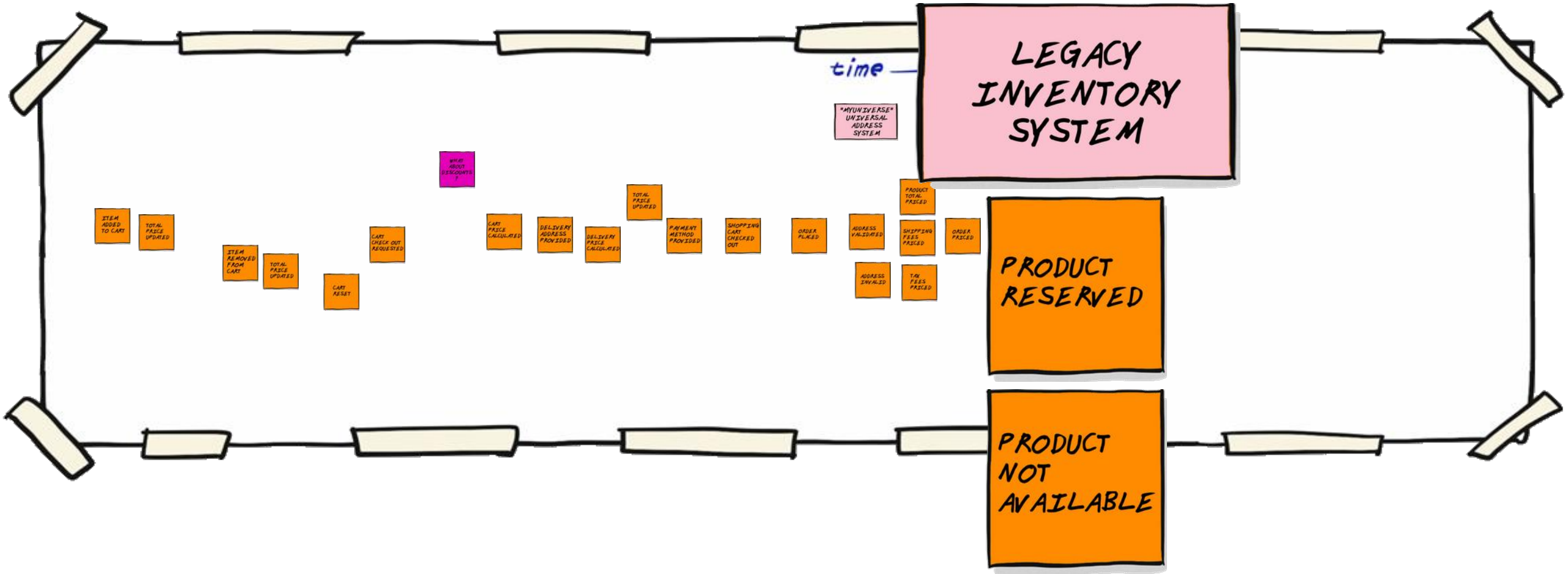
Don’t we need to **check** that the products are **actually available** before we proceed with payment?

Right, what happens if the customer **pays** for something we **don’t have in stock** anymore?

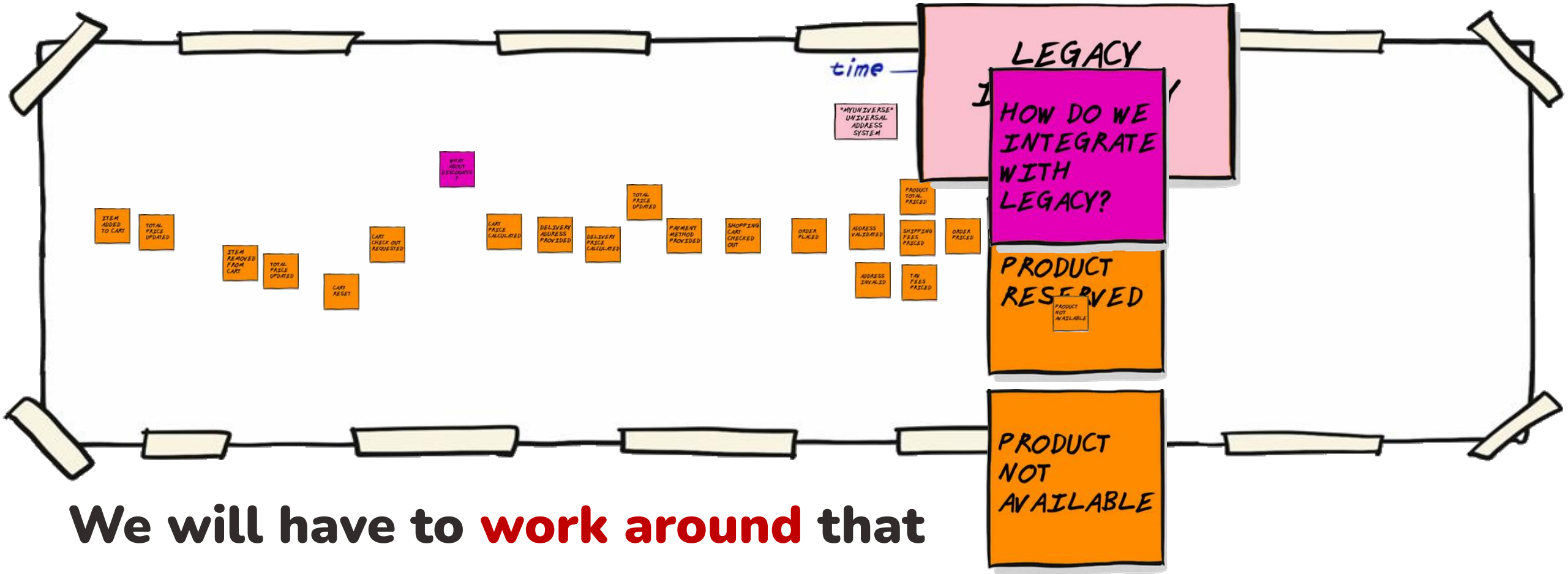
I see two options : either we **refund** the customer for the product that is unavailable, or the customer will **wait** for us to have it in stock again and **ship it** to him **later**.

Ok. Then we need to **ask the customer** about his **decision**. How do we do that?

First, we check for availability of the products

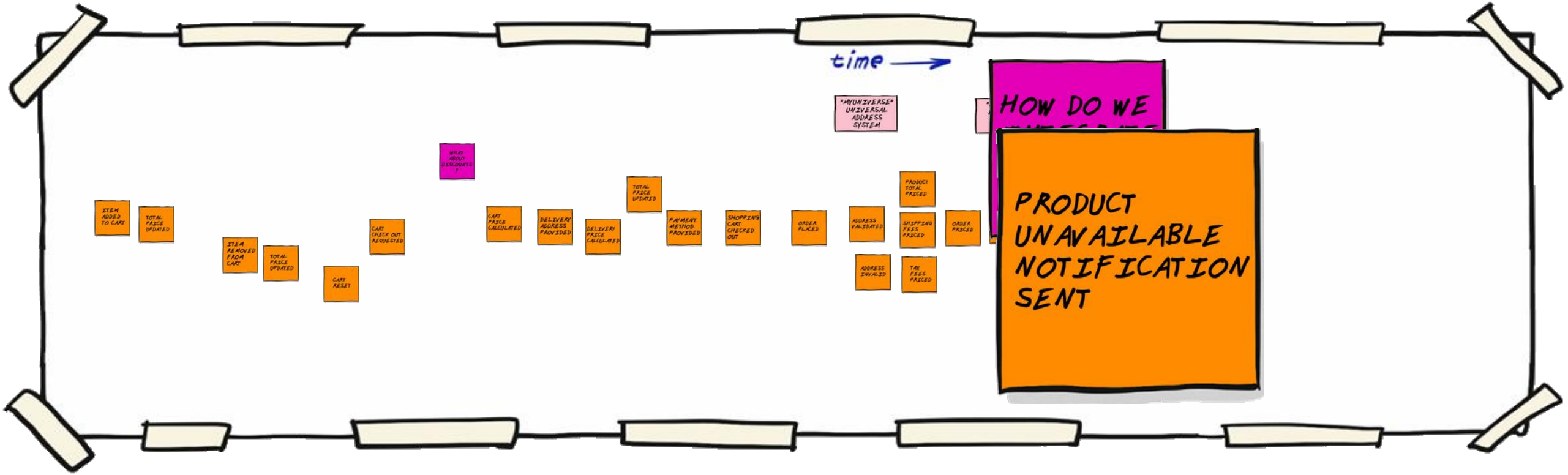


Our inventory is a legacy system

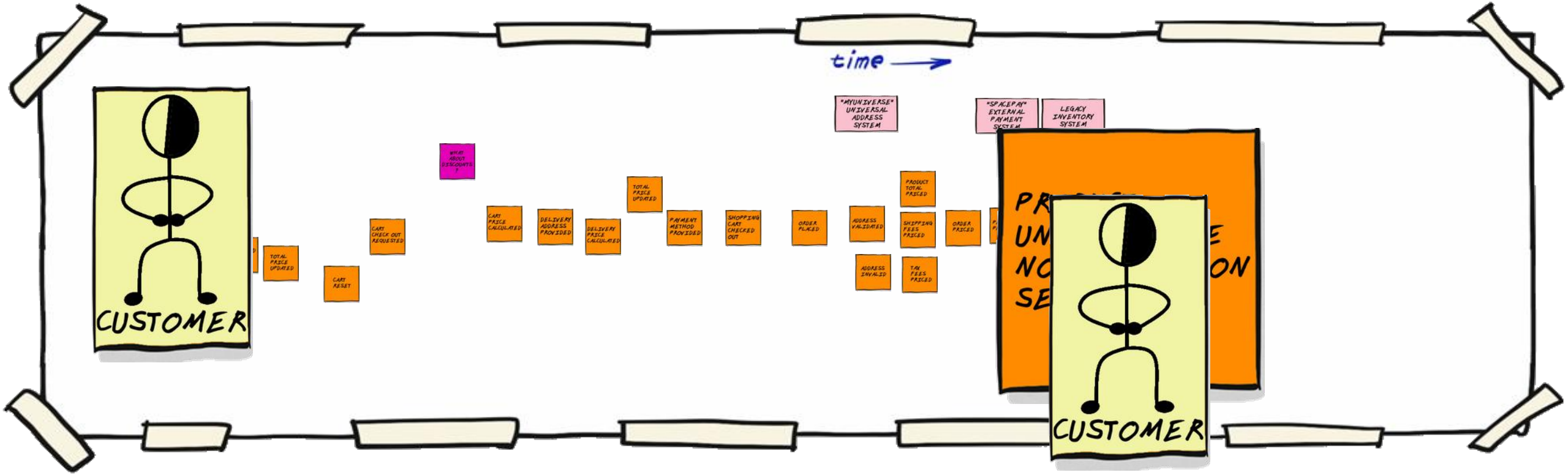


We will have to work around that

We need to **notify** the customer when a product is **unavailable**

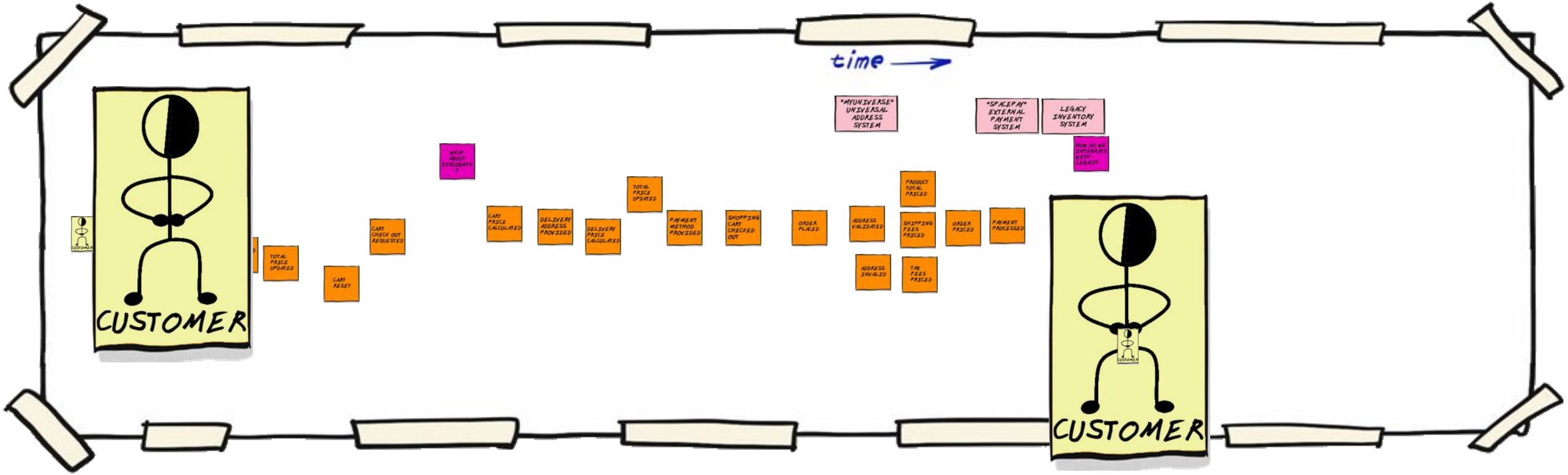


The customer is involved again here



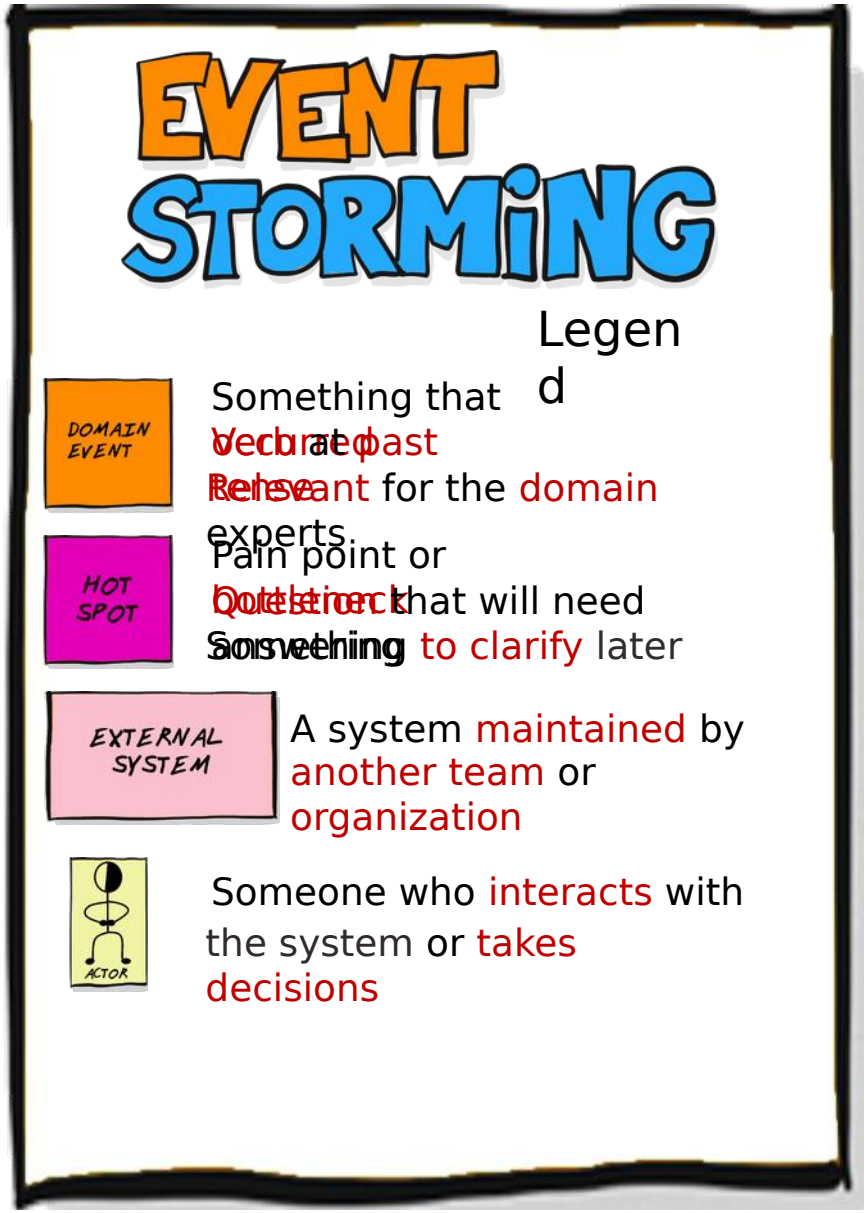
Let's make that *"infinity stone"* clear

The customer is involved again here

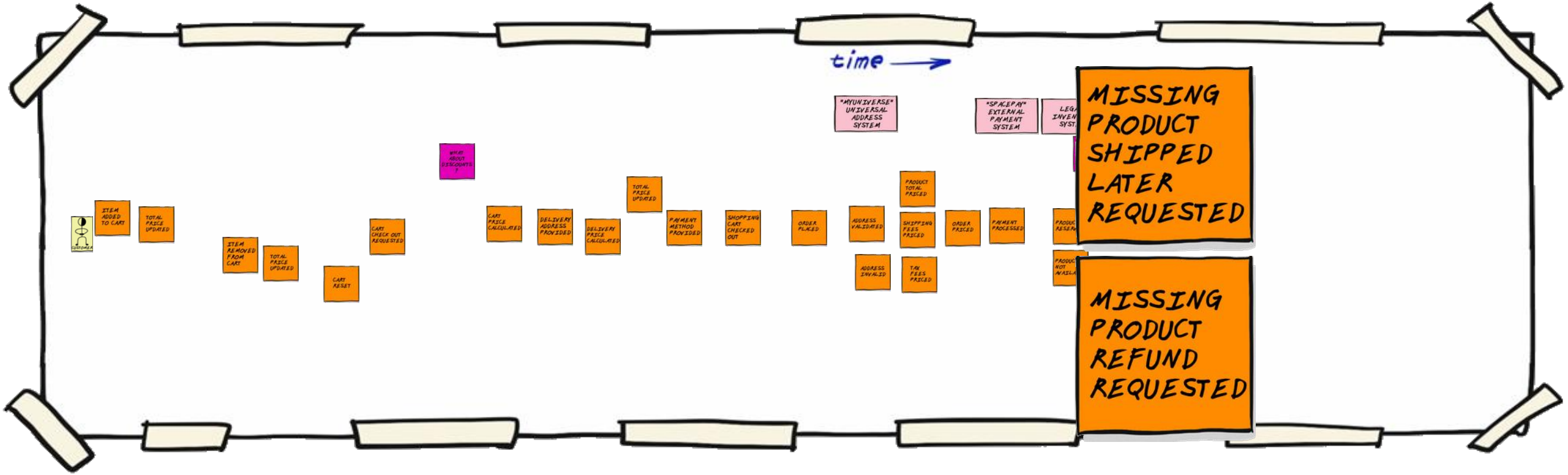


Let's make that *infinity stone* clear

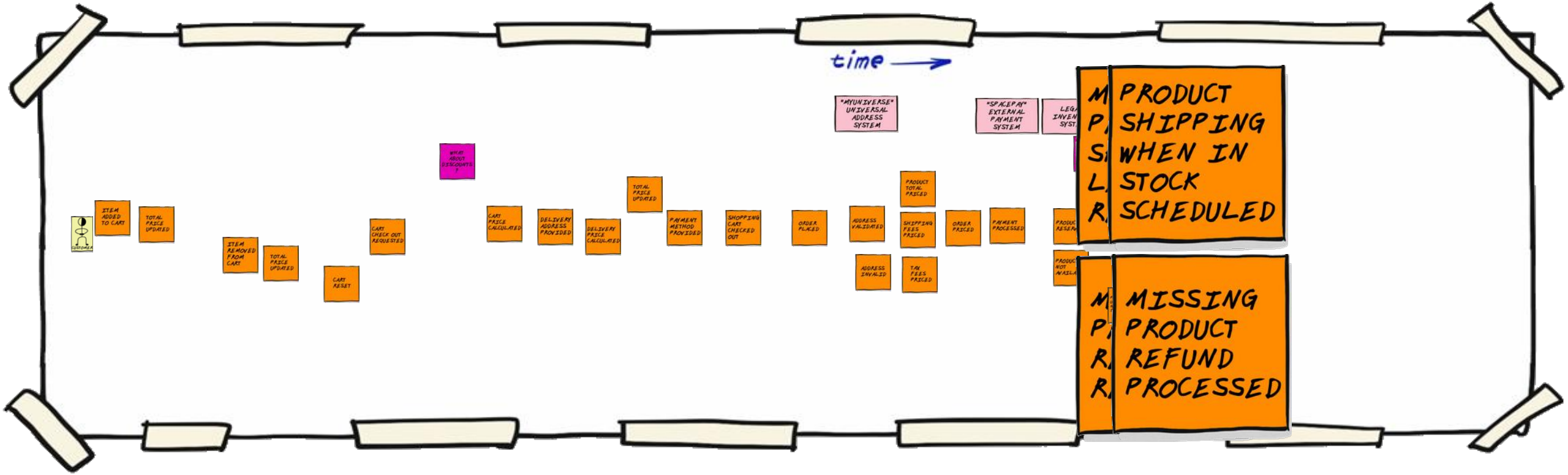
Add the **Actor** sticky to the **legend**



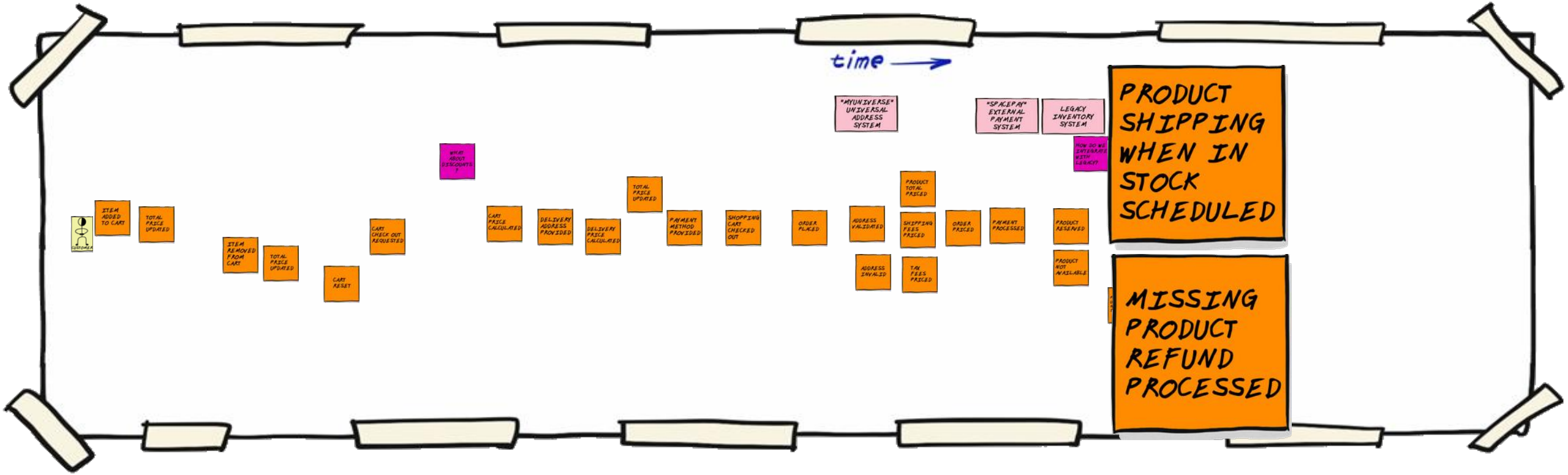
Then we take note of the customer's decision



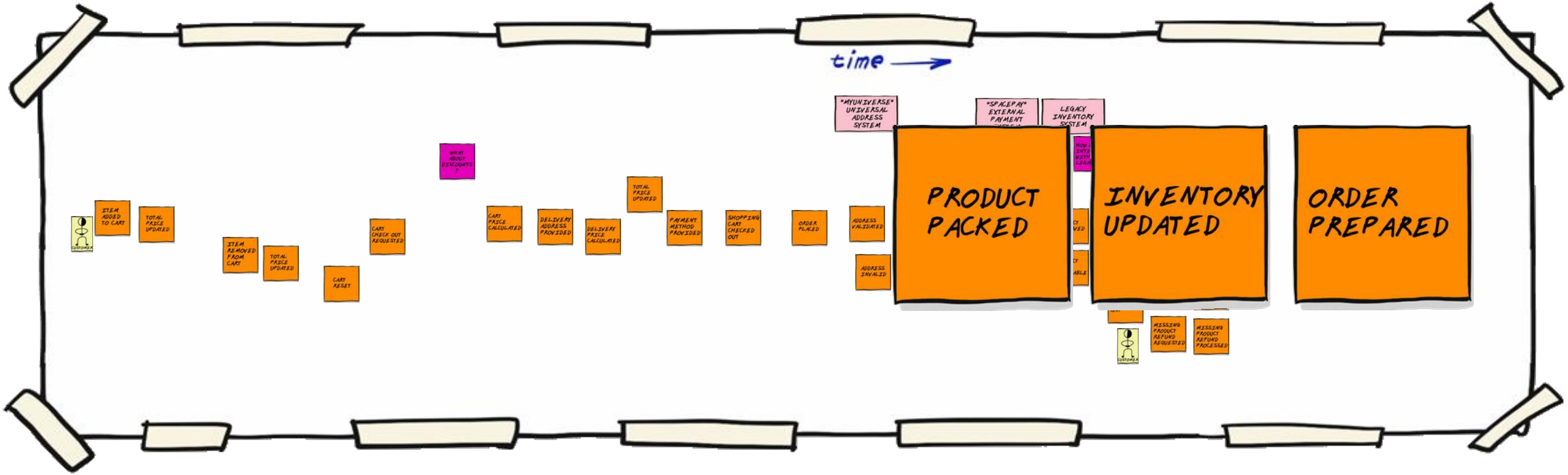
And we react accordingly



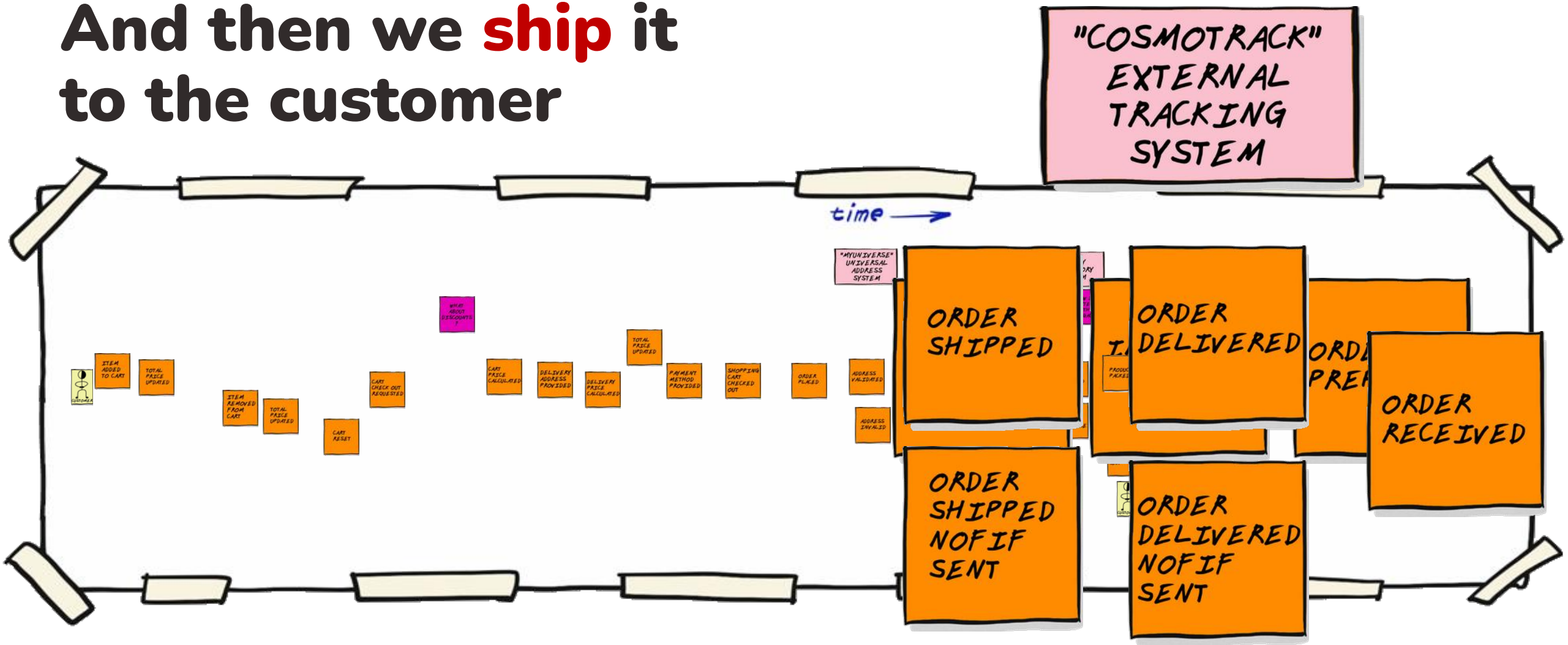
And we react accordingly



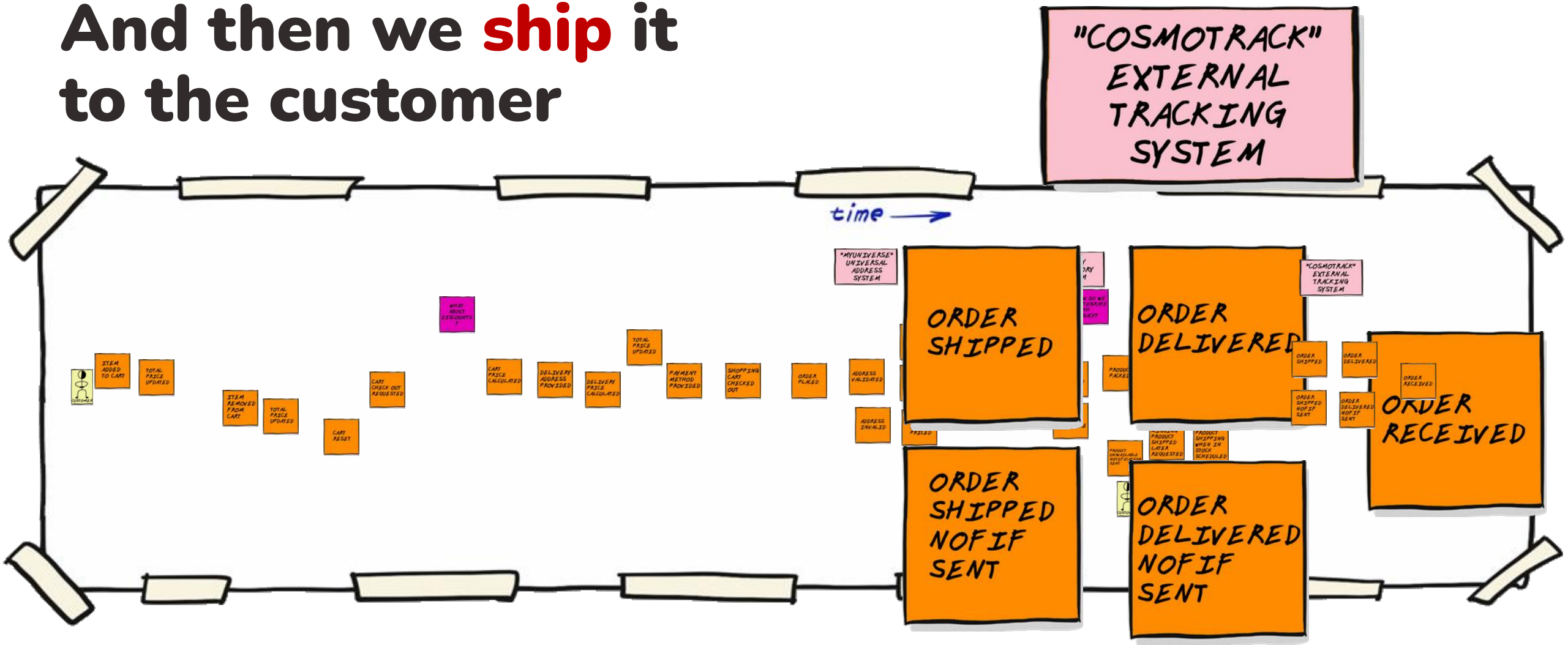
We move on with preparing the order



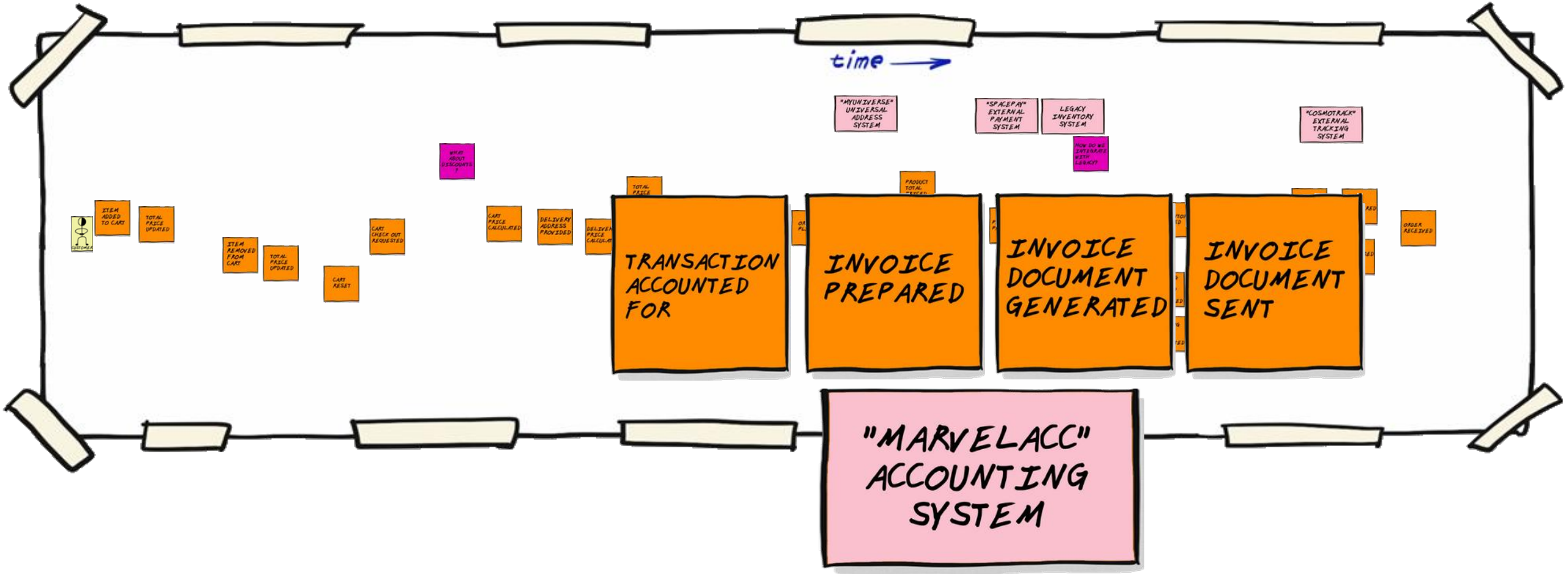
And then we ship it to the customer



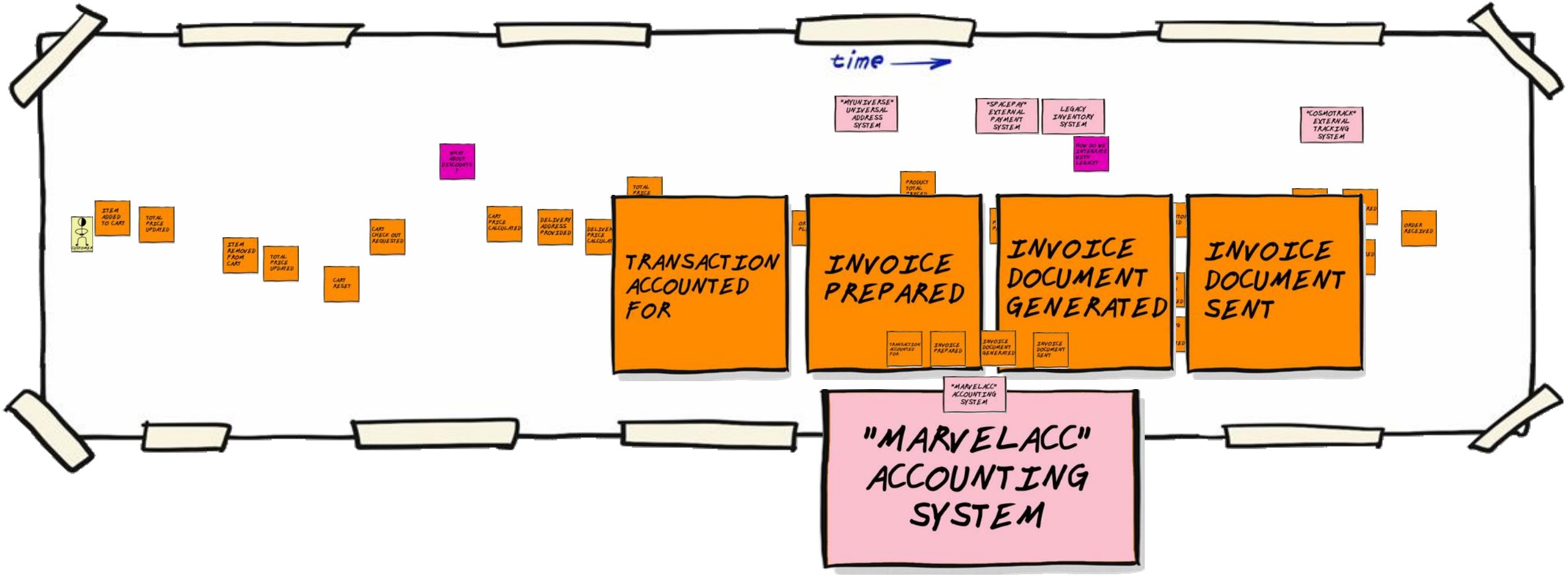
And then we ship it to the customer



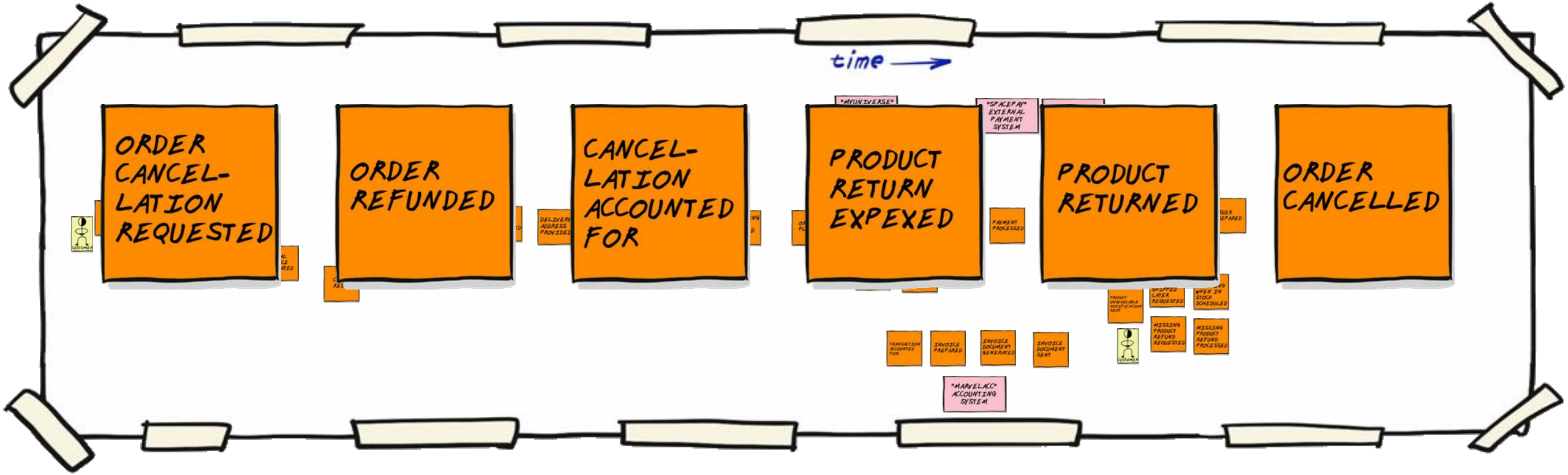
We also need to take care of **invoicing** and **accounting**



We also need to take care of **invoicing** and **accounting**

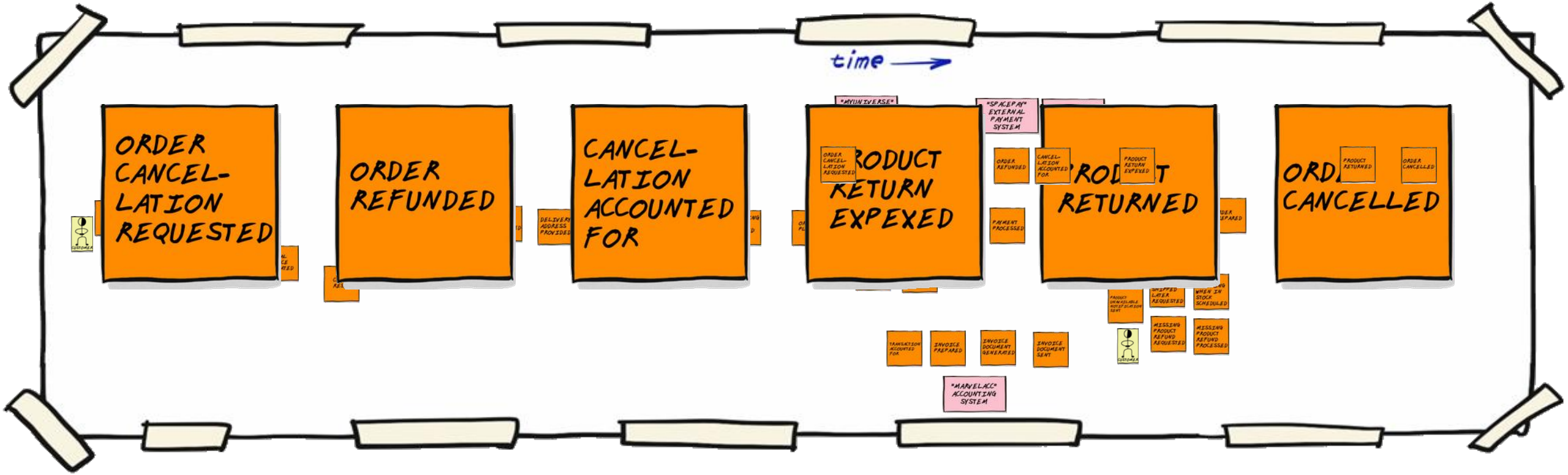


The customer can also cancel his order



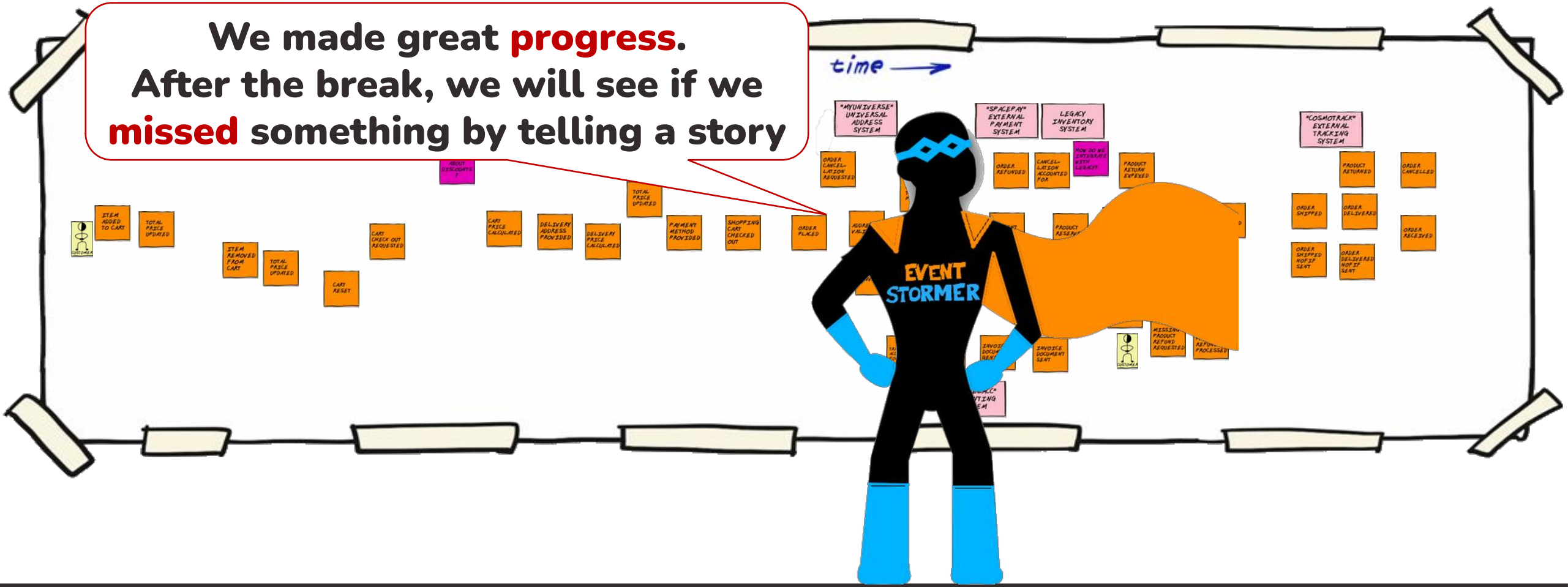
We need some kind of compensating actions

The customer can also cancel his order



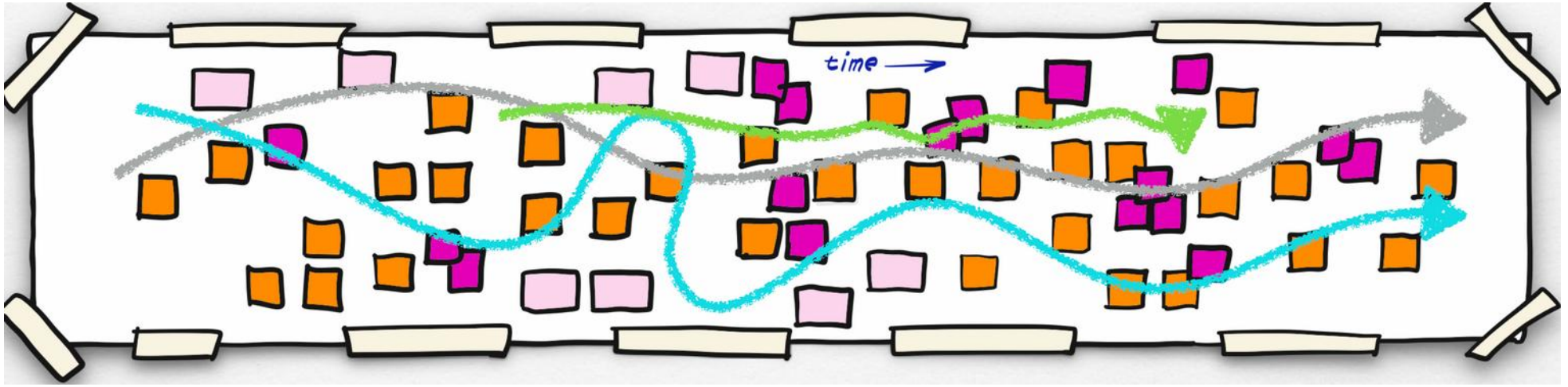
We need some kind of compensating actions

Time for a **break**

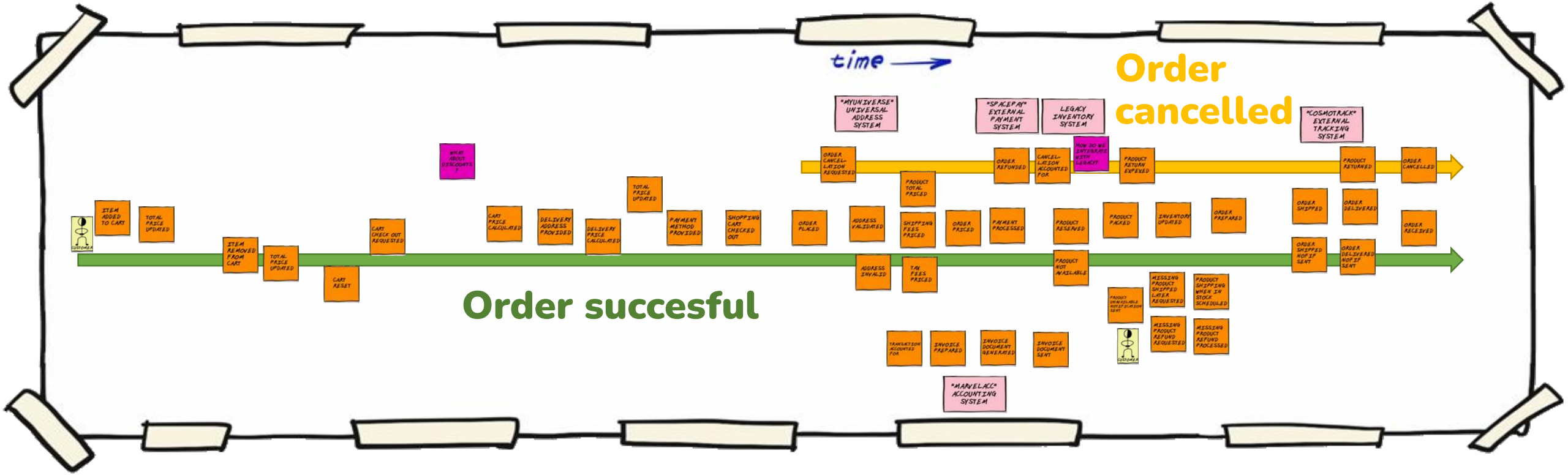


PRO TIP

Using **story telling**, you can go through the domain events and create different **business narratives** to verify that you haven't missed anything

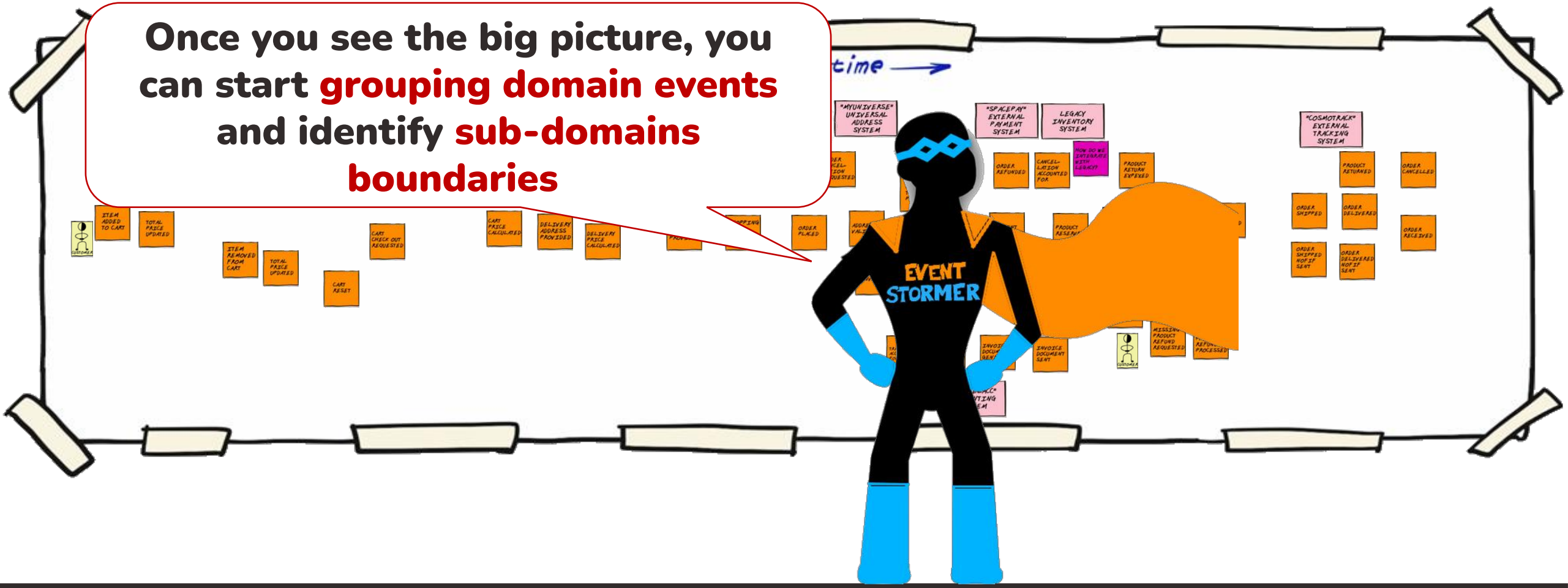


Different business narratives



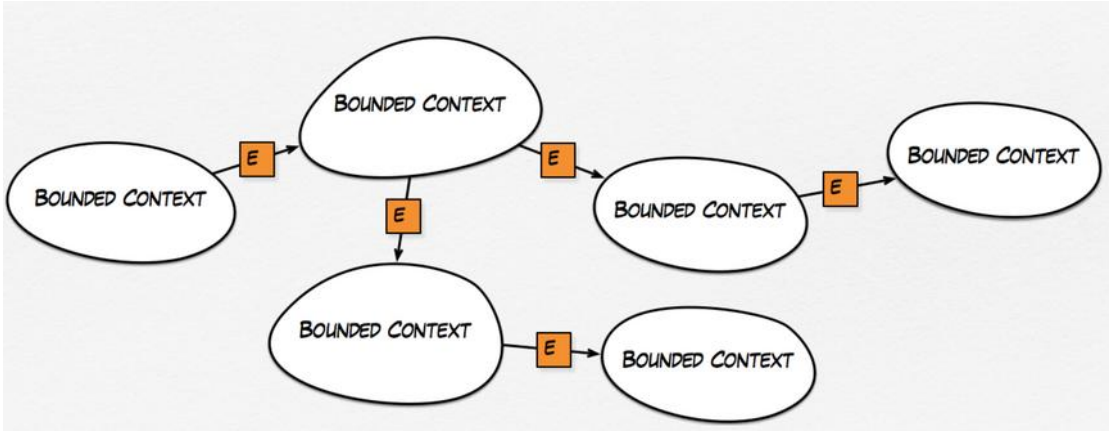
Finding sub-domain boundaries

Once you see the big picture, you can start grouping domain events and identify sub-domains boundaries



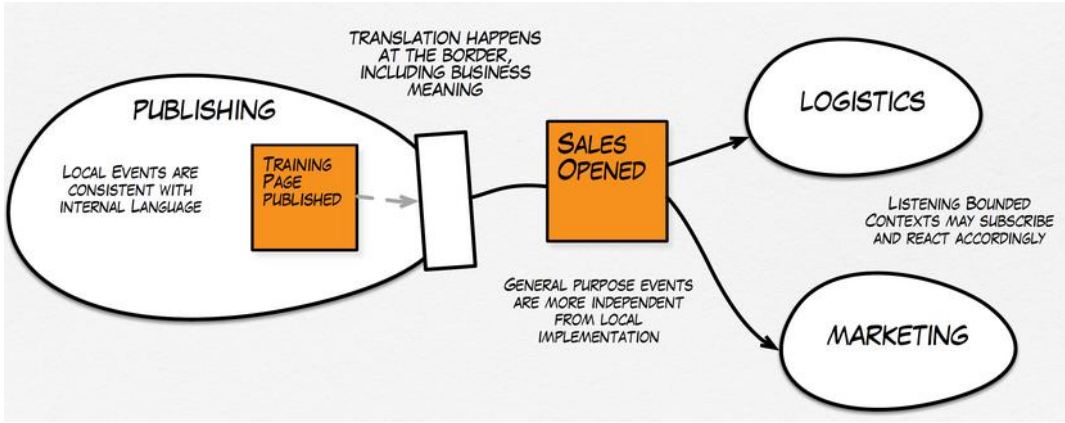
PRO TIP

Finding sub-domain boundaries therefore identifying clear bounded contexts is one of the key benefits of Domain-Driven Design

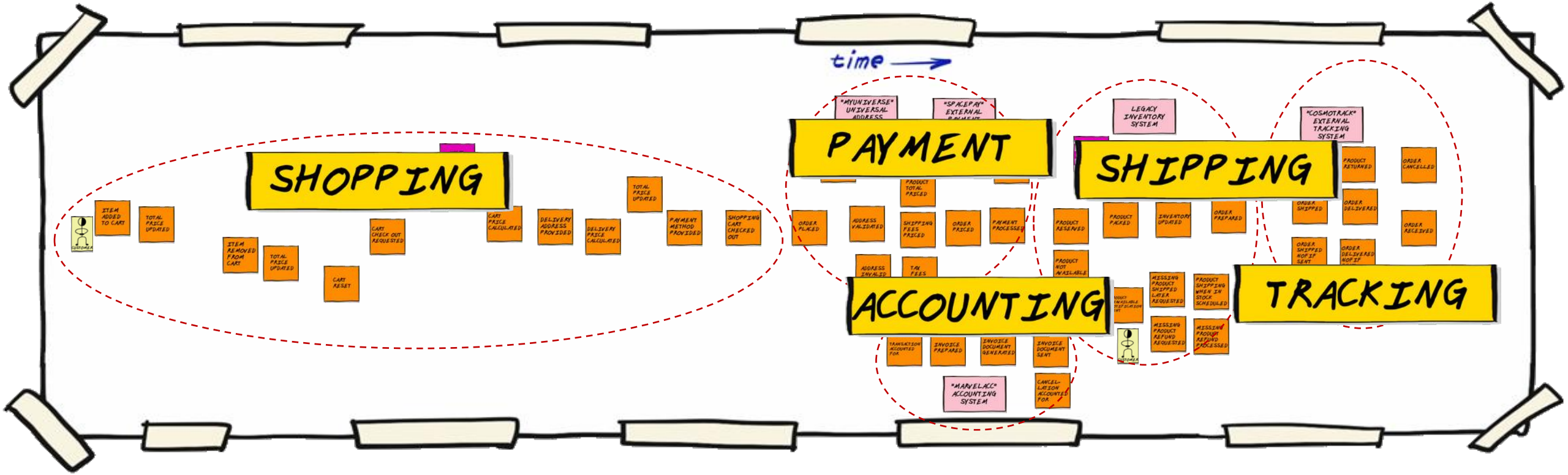


Events can flow between bounded contexts to ensure synchronization

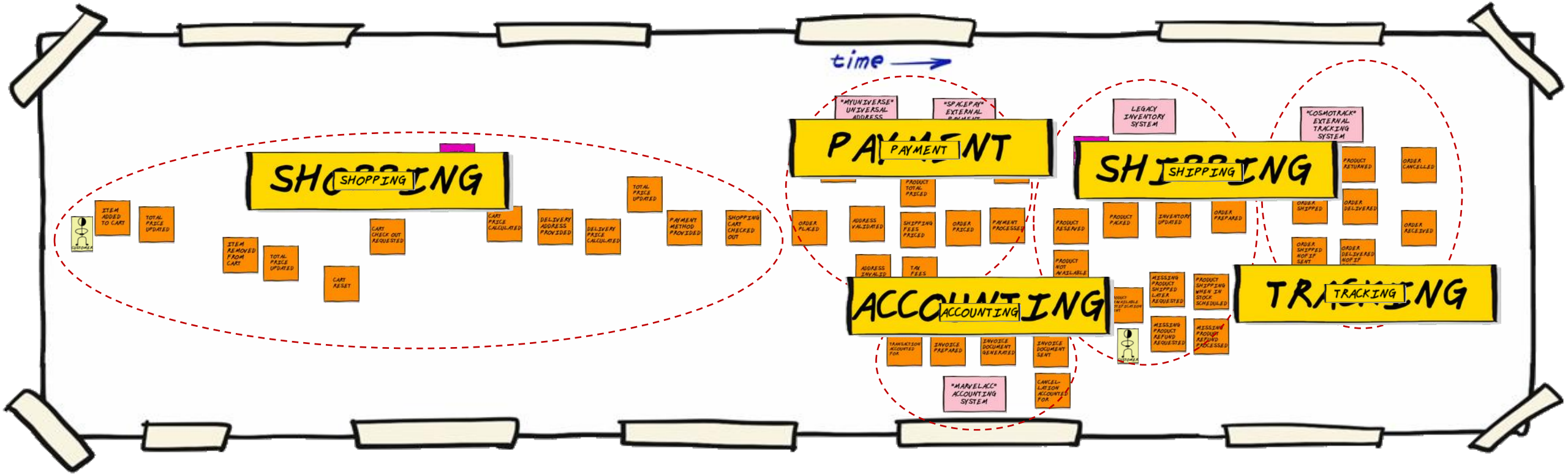
Domain events can be considered as a published language



Finding sub-domain boundaries



Finding sub-domain boundaries



Big Picture EventStorming

Rich discussions and interactions
Everyone engaged
Different point of views
Focused on the business
Building a shared language
(a.k.a. ubiquitous language)

Visualization helps
Incremental notation
Fuzzy by design

Exploring together
Building a shared vision
Keeping uncertainty in check
(hot spot)

Clarity

This Photo by Unknown Author is licensed under CC BY-NC-ND

Some time later...

This Photo by Unknown Author is licensed under CC BY-ND

Guys! You put me in charge of customer service. But I have no idea what the **process to return a product** should look like.

You're right. Let's try to **work** on that now.

It should normaly start by a customer **requesting to return a product**. But how do we visualize that?

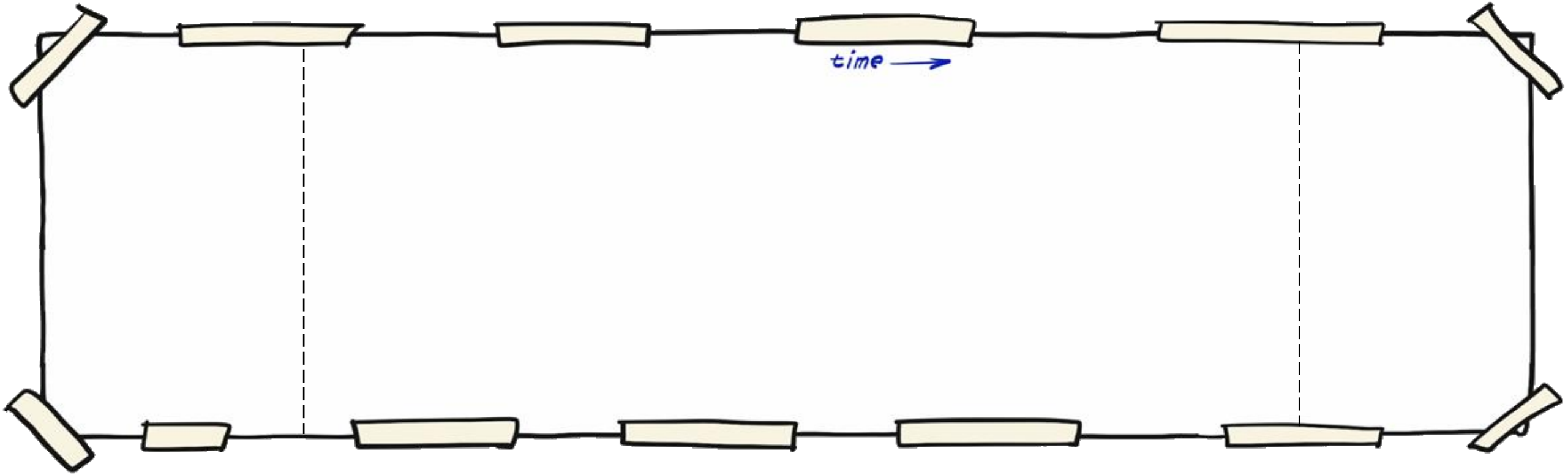
PRO TIP

Process Modeling
EventStorming can be used to model an epic or a set of features

BIG PICTURE	EVENTS	HOT SPOTS, SYSTEMS, PEOPLE	CONFLICTS, GOALS, BLOCKERS, BOUNDARIES
PROCESS MODELLING	EVENTS	+ POLICIES, COMMANDS, READ MODELS	VALUE PROPOSITION, POLICIES, PERSONAS, INDIVIDUAL GOALS

EventStorming can be used to map things at different levels of abstraction

Process Modeling EventStorming

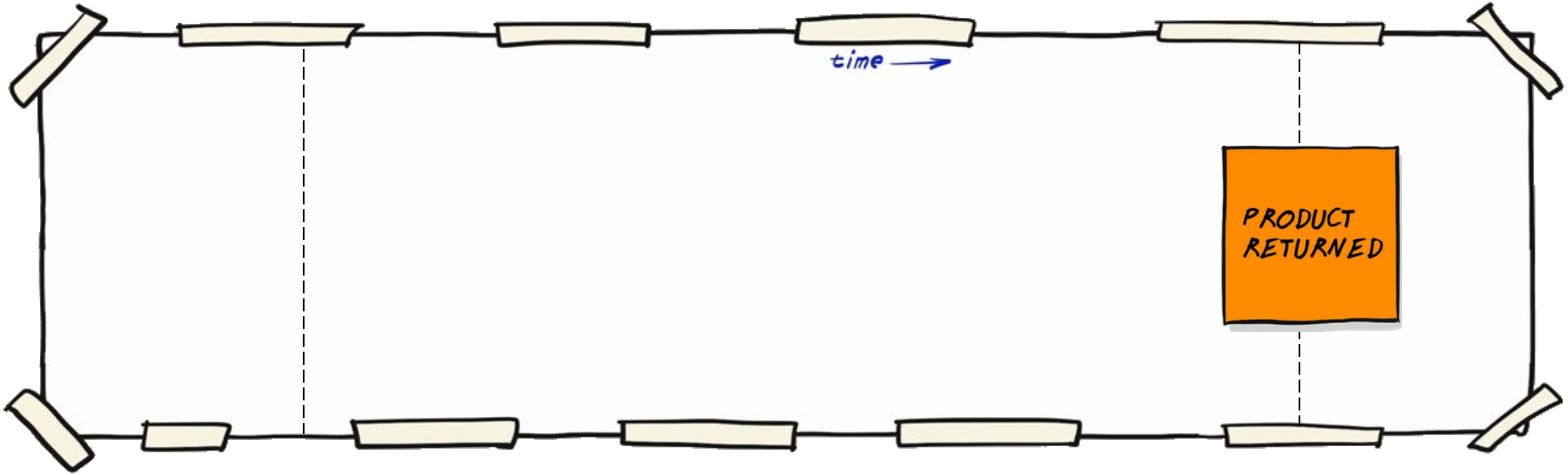


Preconditions

The flow to discover

Outcomes

The process finishes with a domain event

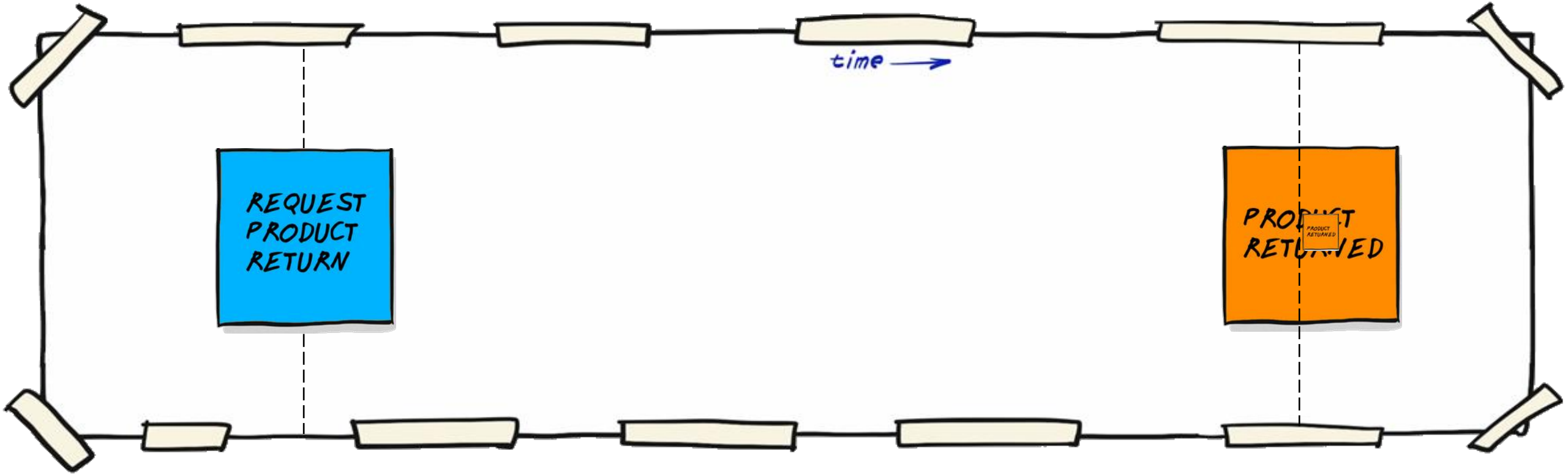


Preconditions

The flow to discover

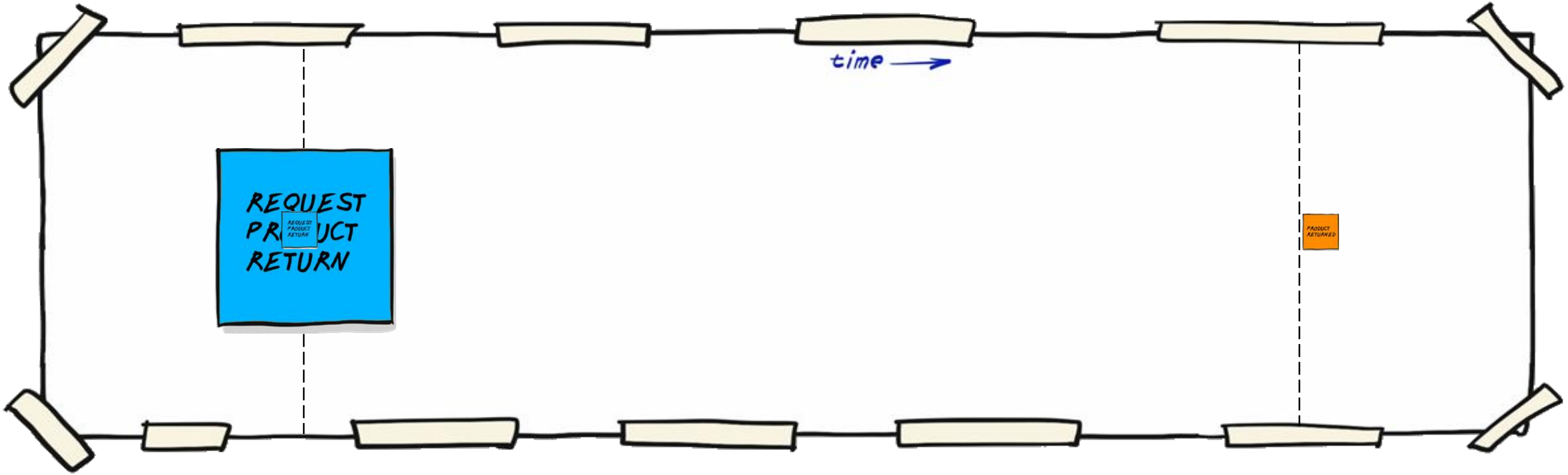
Outcomes

And starts with a command



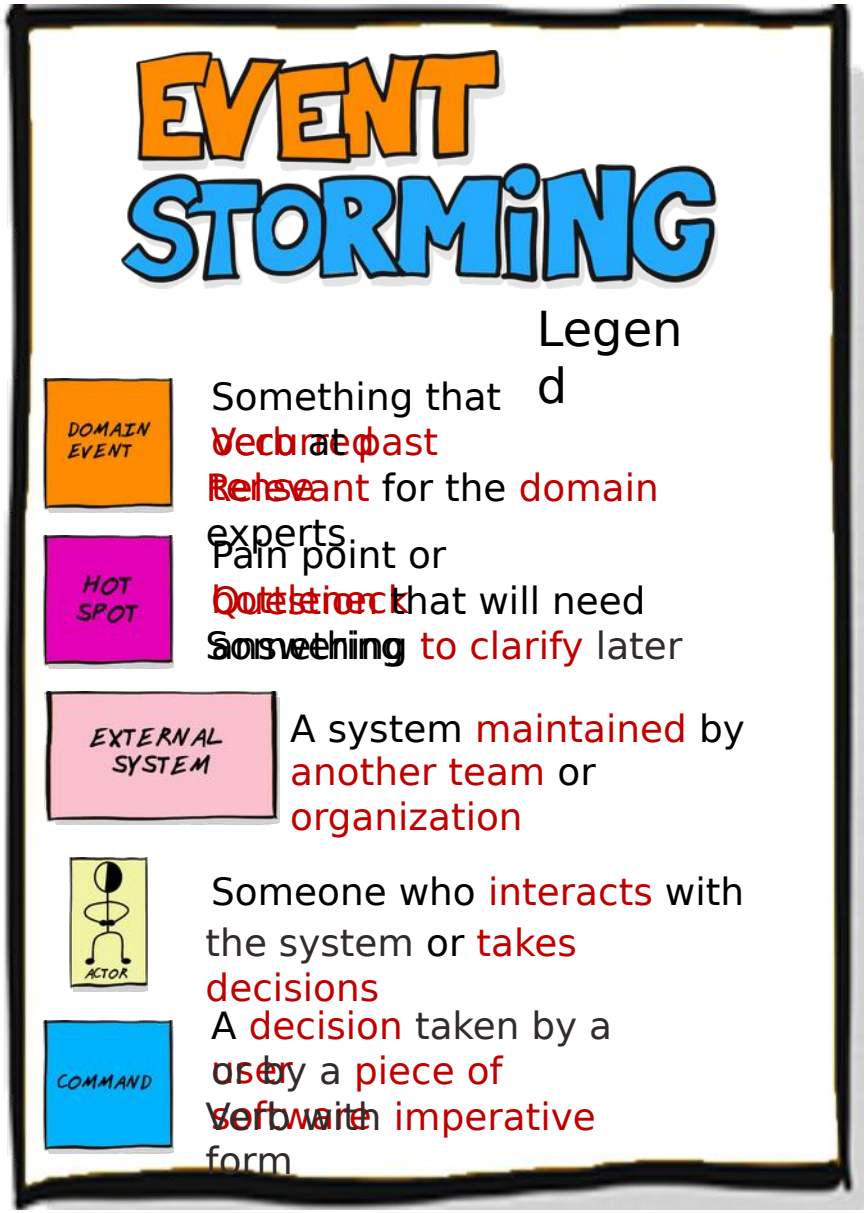
Preconditions **The flow to discover** **Outcomes**

And starts with a command

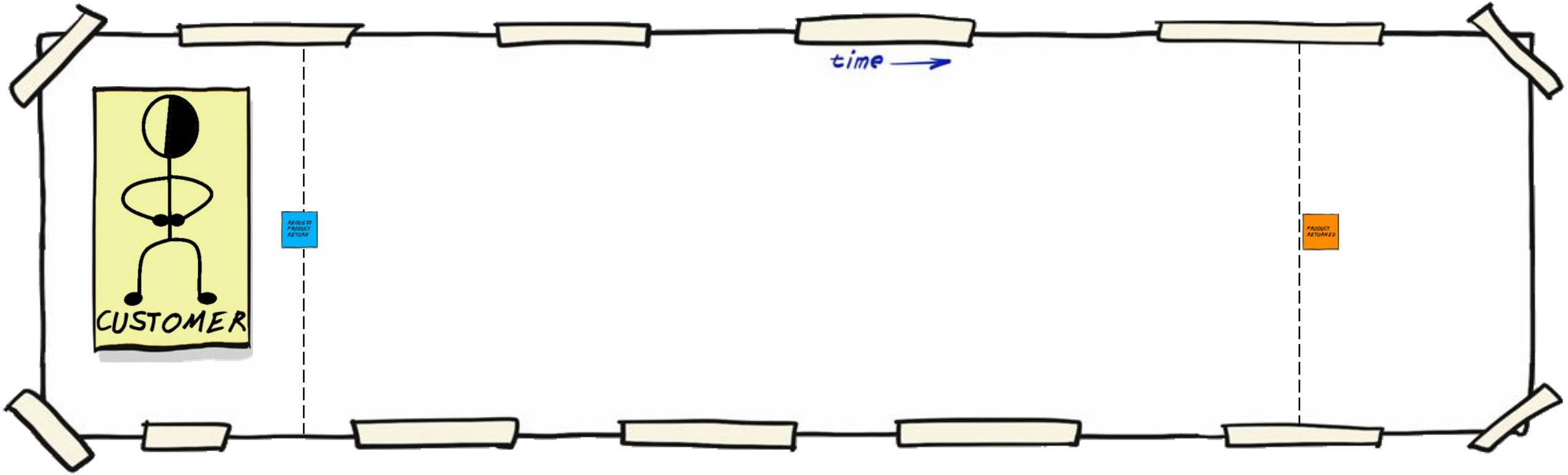


Preconditions **The flow to discover** **Outcomes**

Add the **Command** sticky to the **legend**

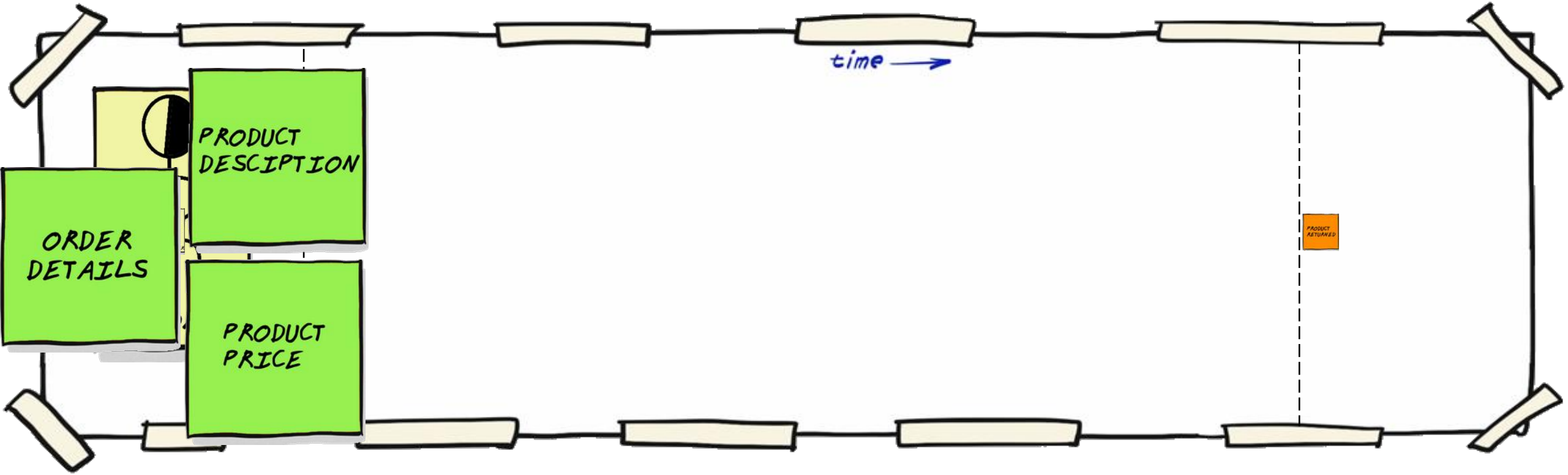


This decision is taken by the customer



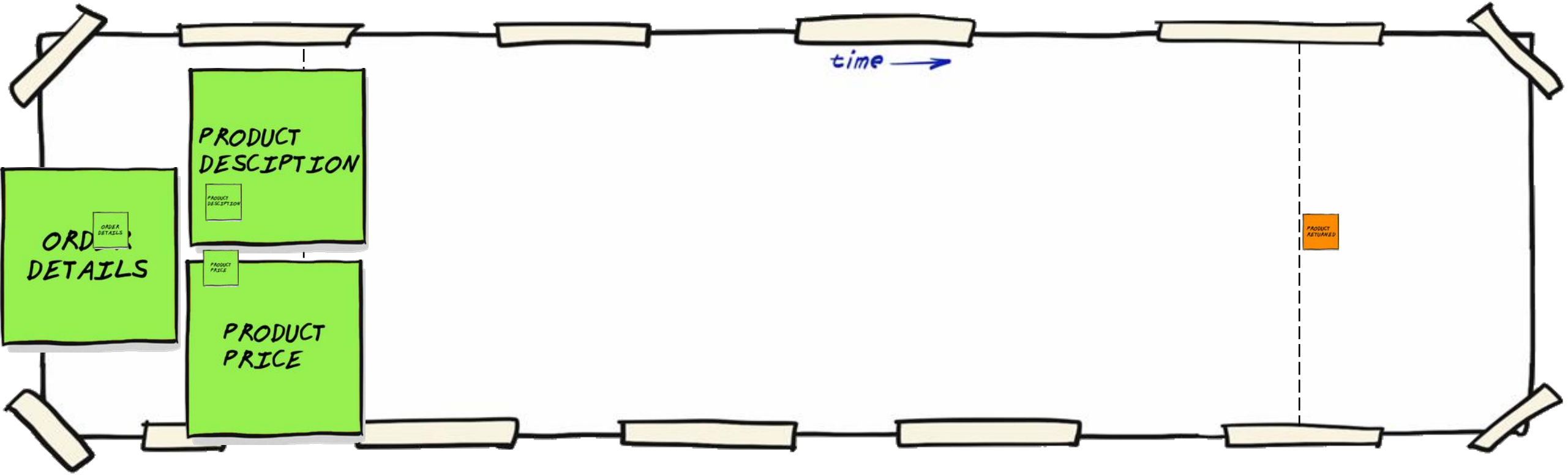
Preconditions **The flow to discover** **Outcomes**

To take this decision, he needs some **information** to be displayed



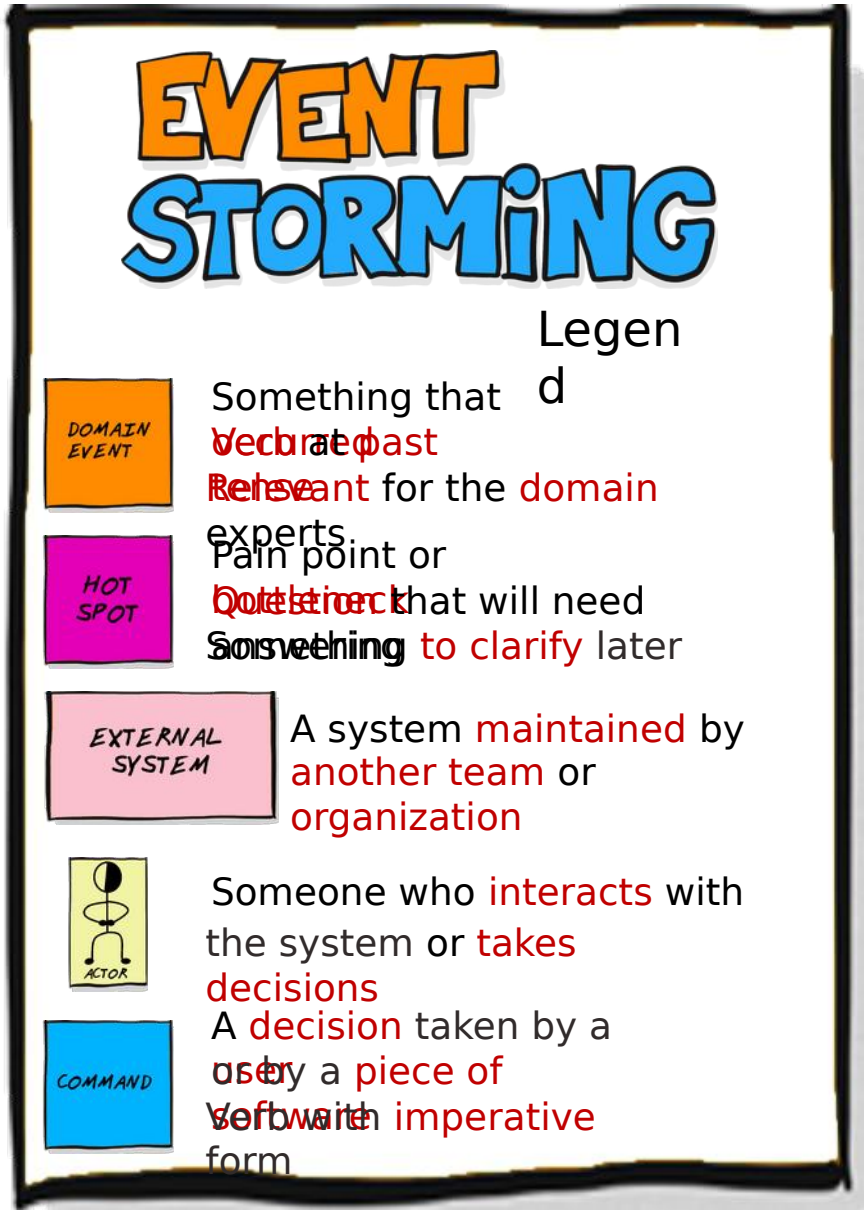
Preconditions **The flow to discover** **Outcomes**

To take this decision, he needs some **information** to be **displayed**

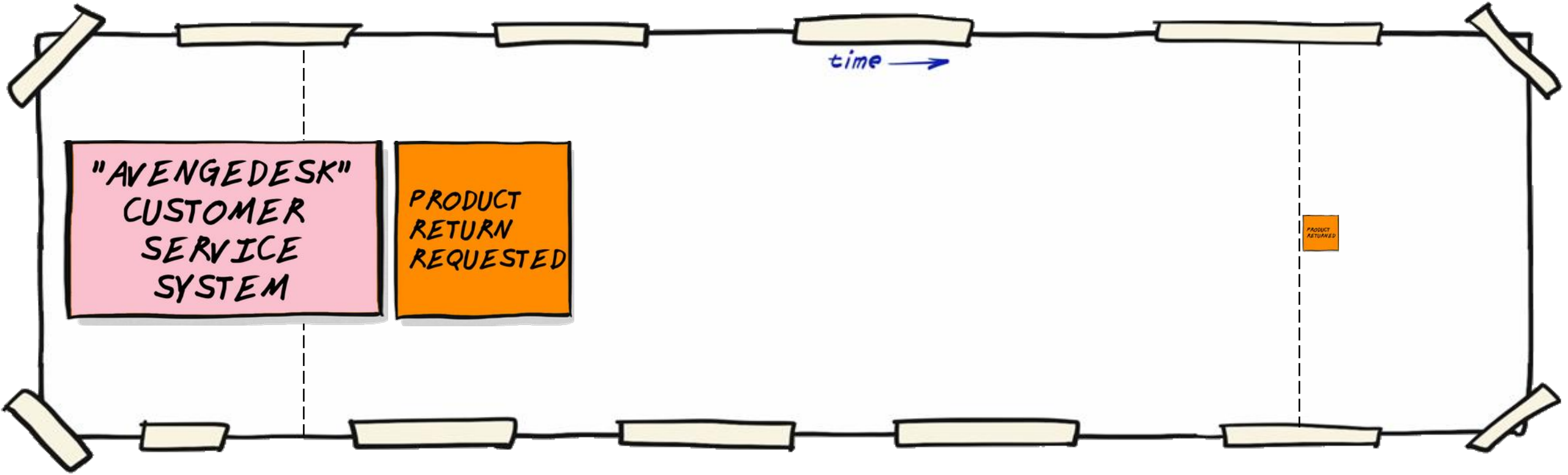


Preconditions **The flow to discover** **Outcomes**

Add the **Read Model** sticky to the **legend**

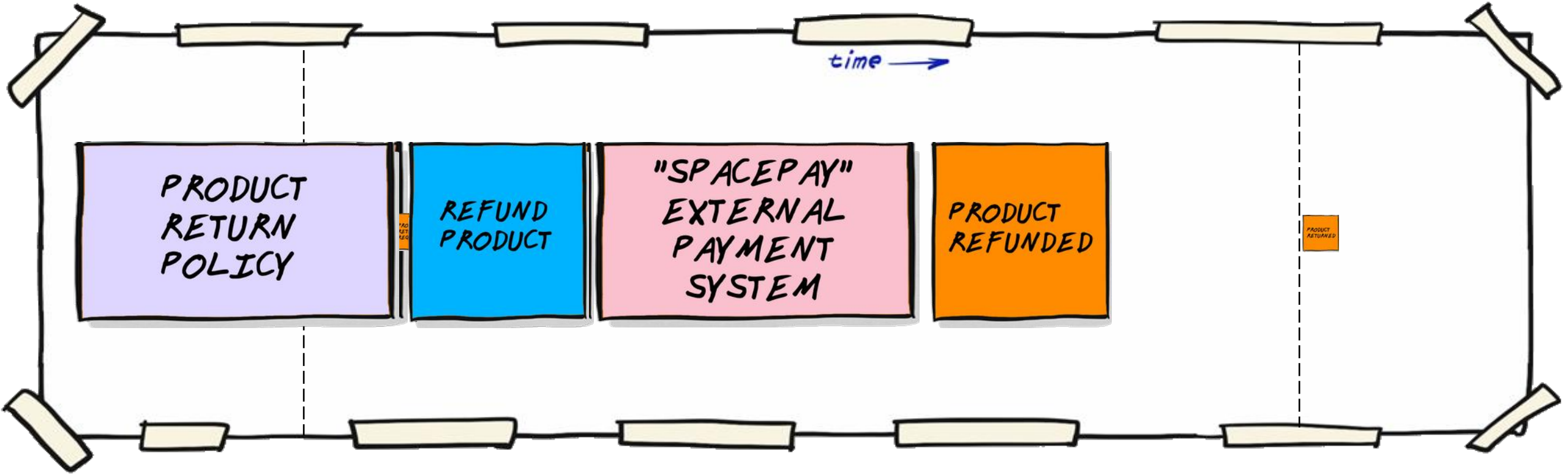


We record the product return request in our customer service system



Preconditions **The flow to discover** **Outcomes**

When the product return is requested, we have to refund the product price

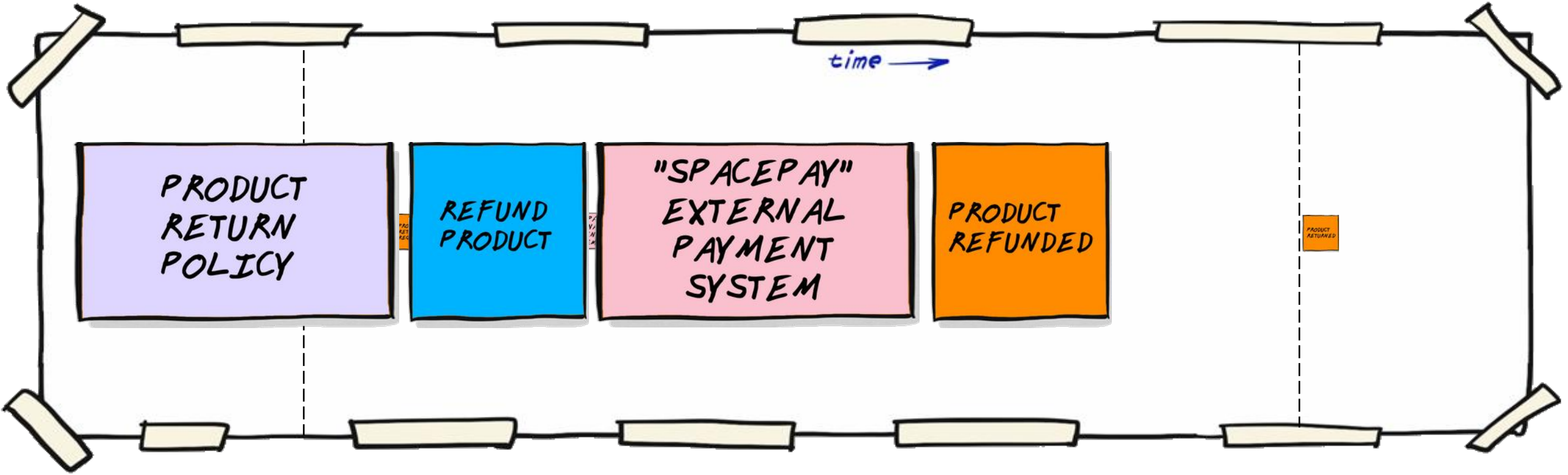


Preconditions

The flow to discover

Outcomes

When the product return is requested, we have to refund the product price

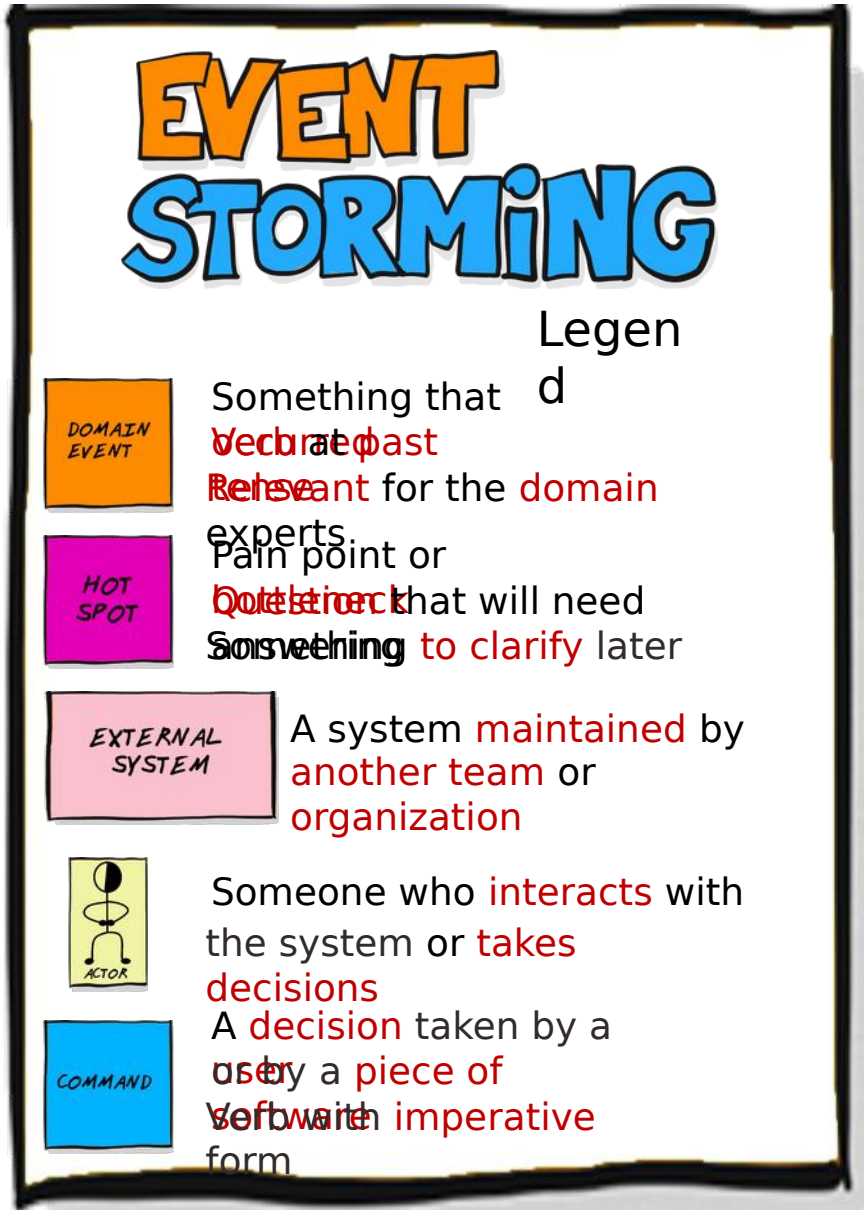


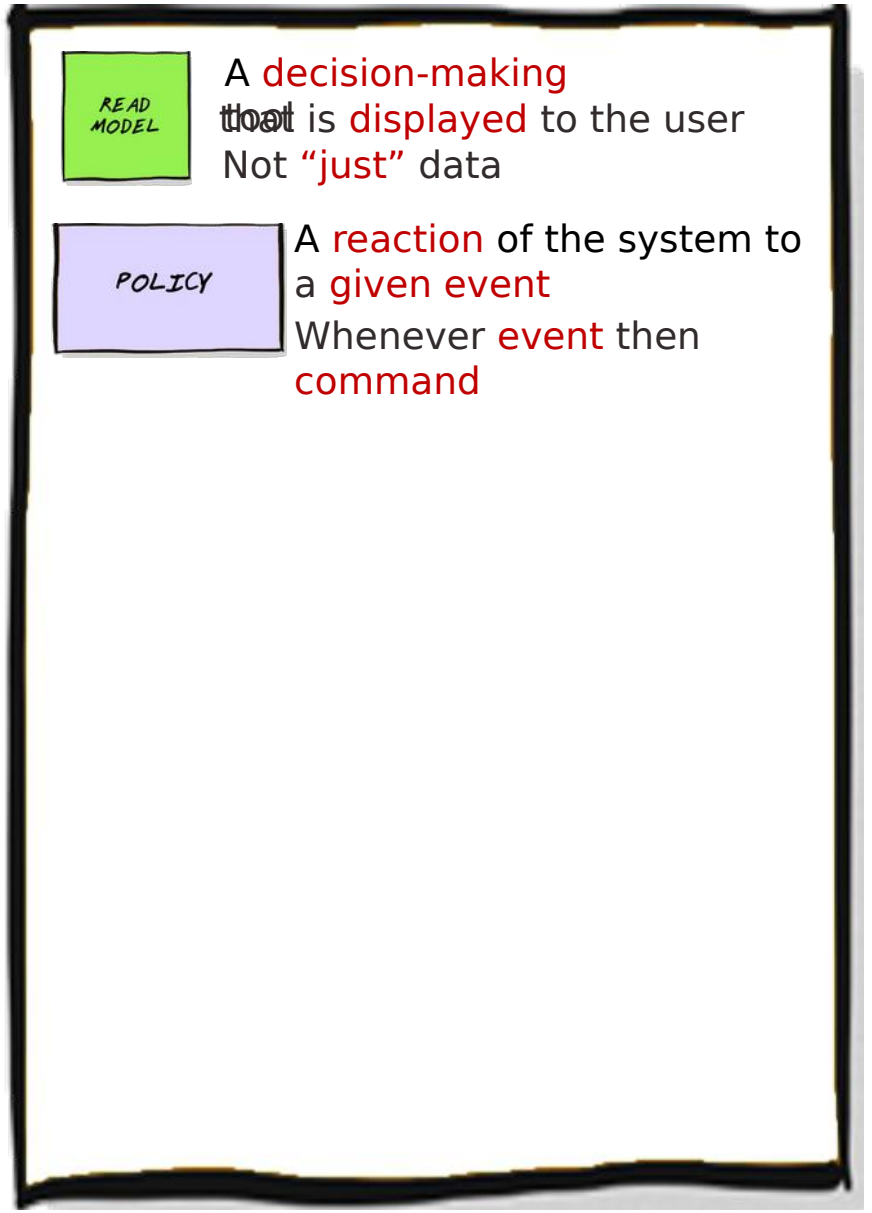
Preconditions

The flow to discover

Outcomes

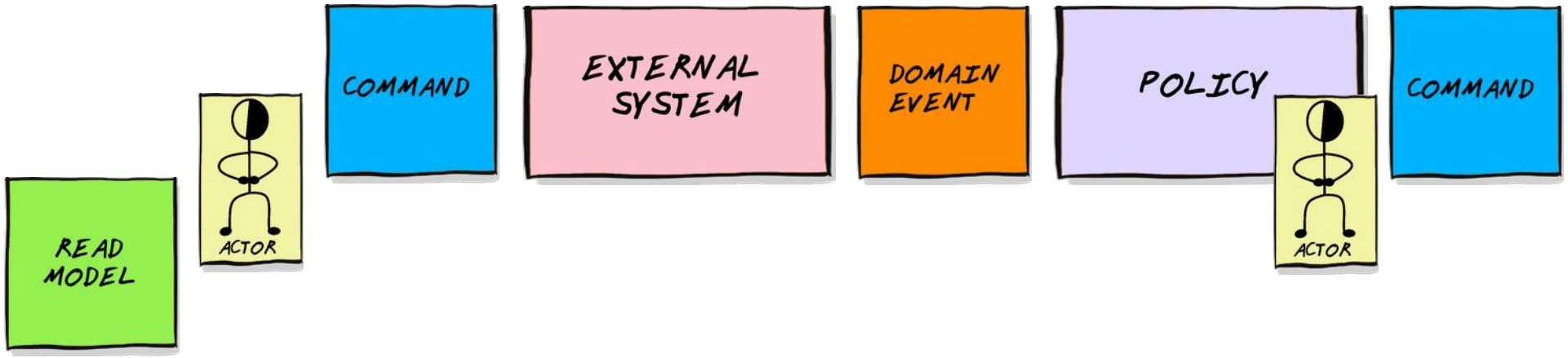
Add the **Policy** sticky to the **legend**





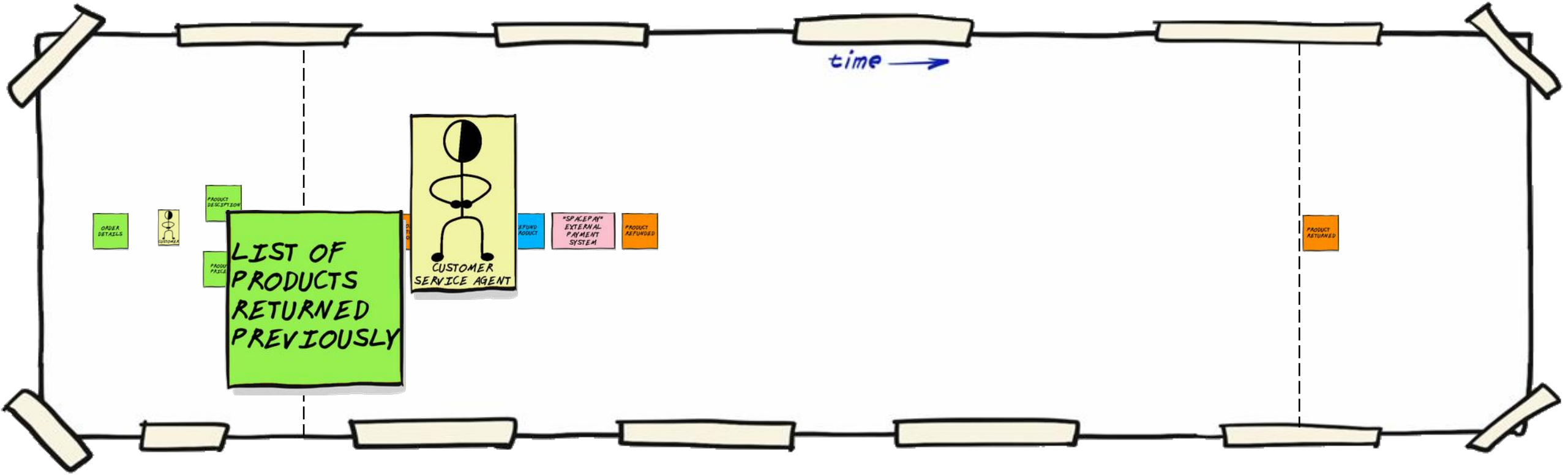
PRO TIP

EventStorming is like a **color-puzzle**



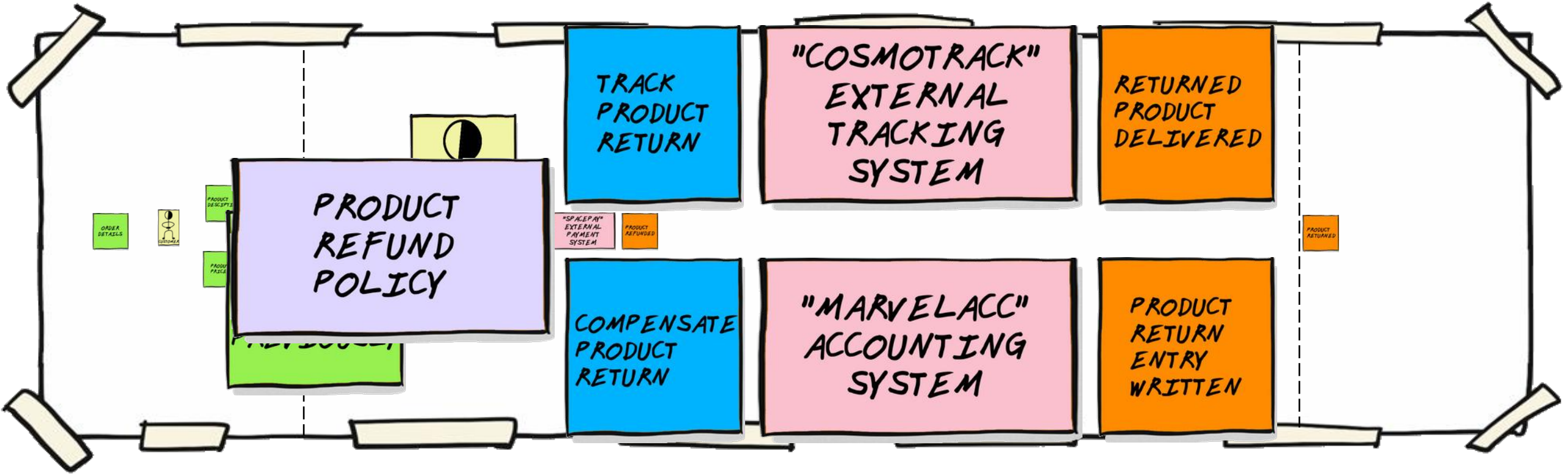
A business process is a chain of **commands handled** by **systems** and **events reacted** to by **automated** or **manual policies**

We want to accept the product return manually before we refund it



Preconditions **The flow to discover** **Outcomes**

When the product is refunded, we track the product return and compensate in accounting

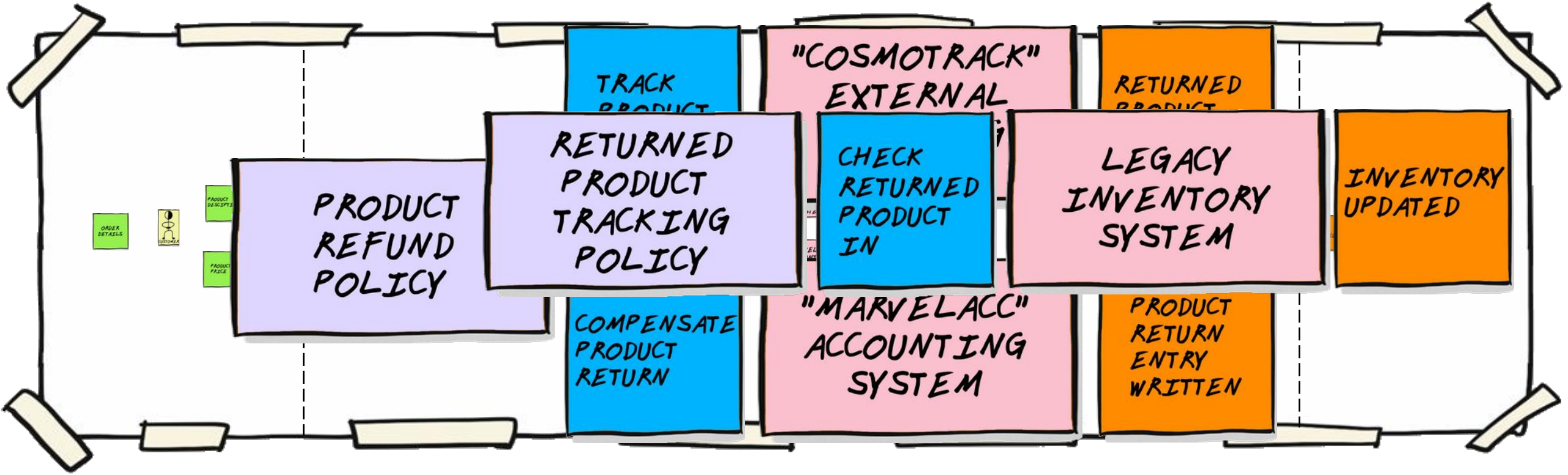


Preconditions

The flow to discover

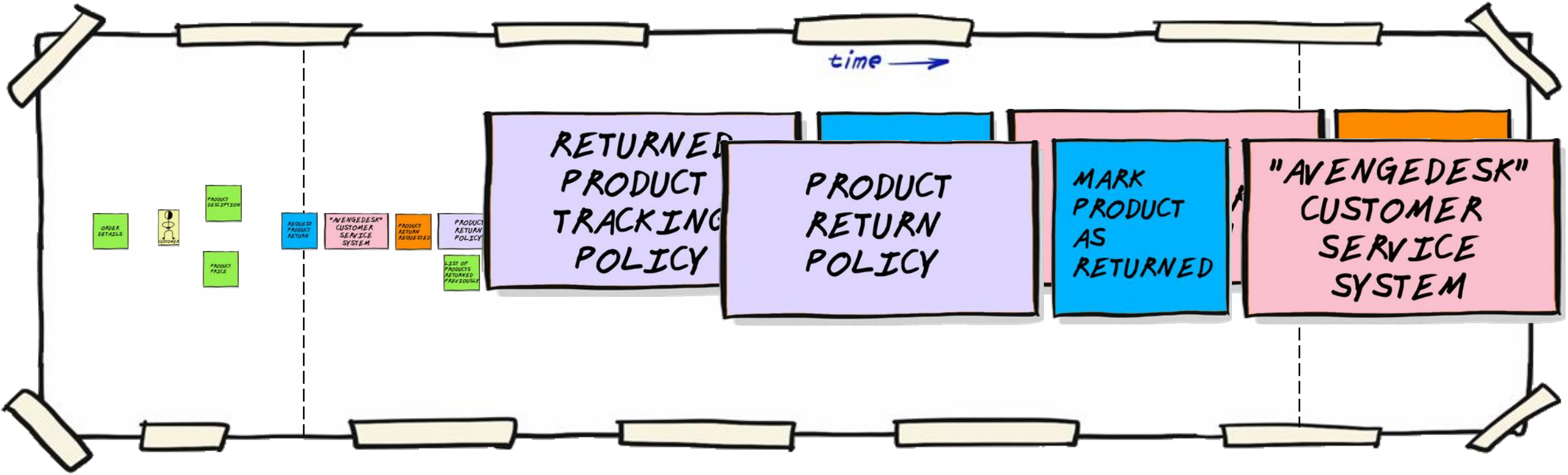
Outcomes

When the returned product is **delivered**, we **update the inventory**



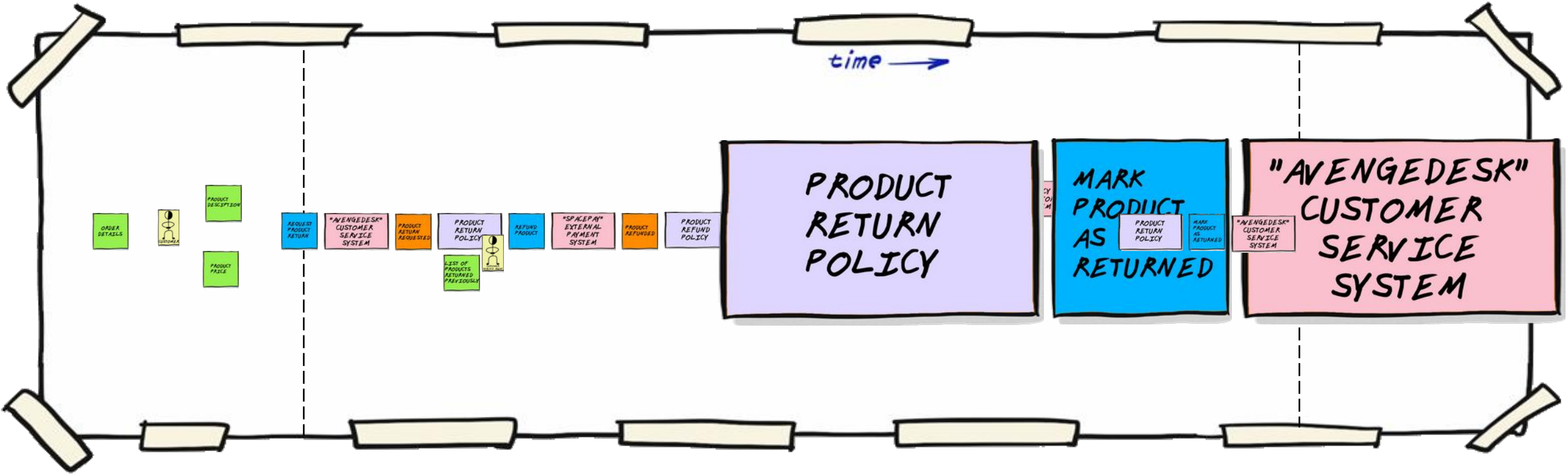
Preconditions **The flow to discover** **Outcomes**

Finally, we mark the product as returned



Preconditions **The flow to discover** **Outcomes**

The whole return process is now modeled



Preconditions **The flow to discover** **Outcomes**

Process Modeling EventStorming

Where everybody **forgets** or **lies**
Ambiguity does not compile
Making the implicit **explicit**

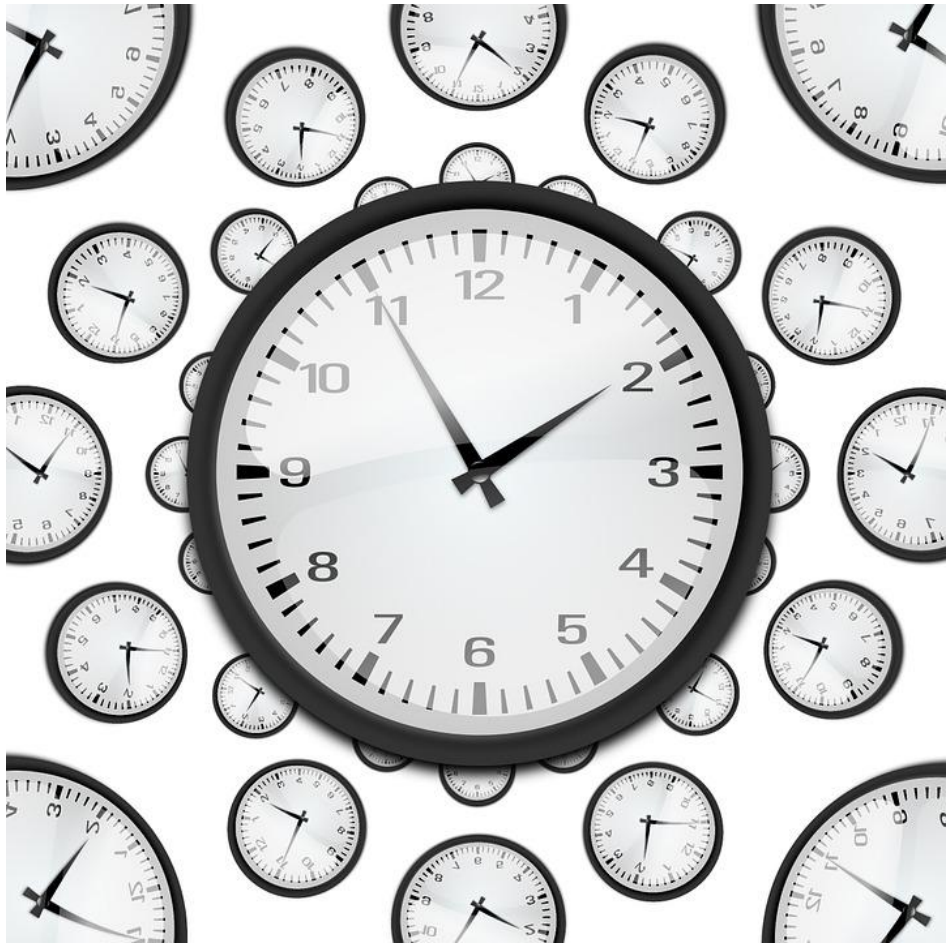
Challenging value
Cross-perspective conversation
Discovering **inconsistencies** and new **opportunities**

Reaching an **agreement**

Consensus

This Photo by Unknown Author is licensed under CC BY-SA-NC

Even later...



Guys! This **product return manual policy** has become a pain in my furry a**! Can't we **do something** about that?

I think that now, we should be able to **automate** part of it.

I think so too. Let's try to **make those rules explicit** and see when the manual policy **is really needed**.

PRO TIP

Software Design

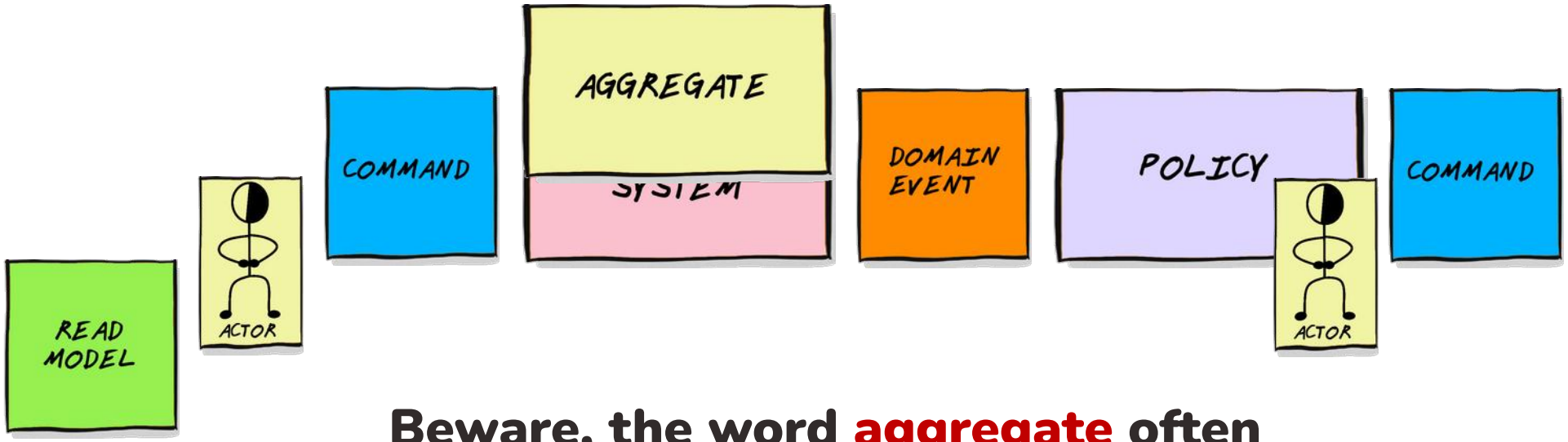
EventStorming is used to design a set of software artifacts that enforce domain logic and business consistency

BIG PICTURE	EVENTS	HOT SPOTS, SYSTEMS, PEOPLE	CONFLICTS, GOALS, BLOCKERS, BOUNDARIES
PROCESS MODELLING	EVENTS	+ POLICIES, COMMANDS, READ MODELS	VALUE PROPOSITION, POLICIES, PERSONAS, INDIVIDUAL GOALS
SOFTWARE DESIGN	EVENTS	+ AGGREGATES	AGGREGATES, POLICIES, READ MODELS, IDS

Remember how EventStorming can be used to map things at different levels of abstraction ?

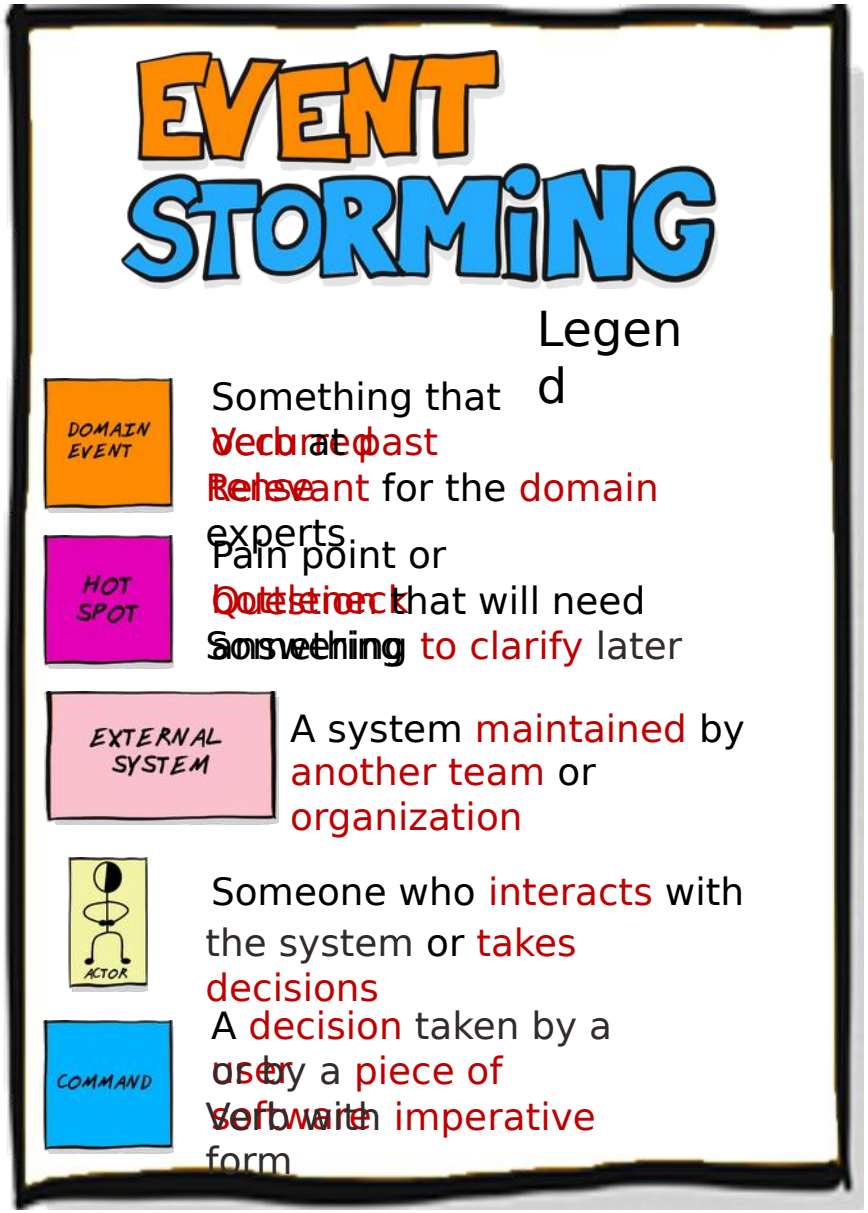
PRO TIP

For that, we add the notion of aggregate to the color-puzzle

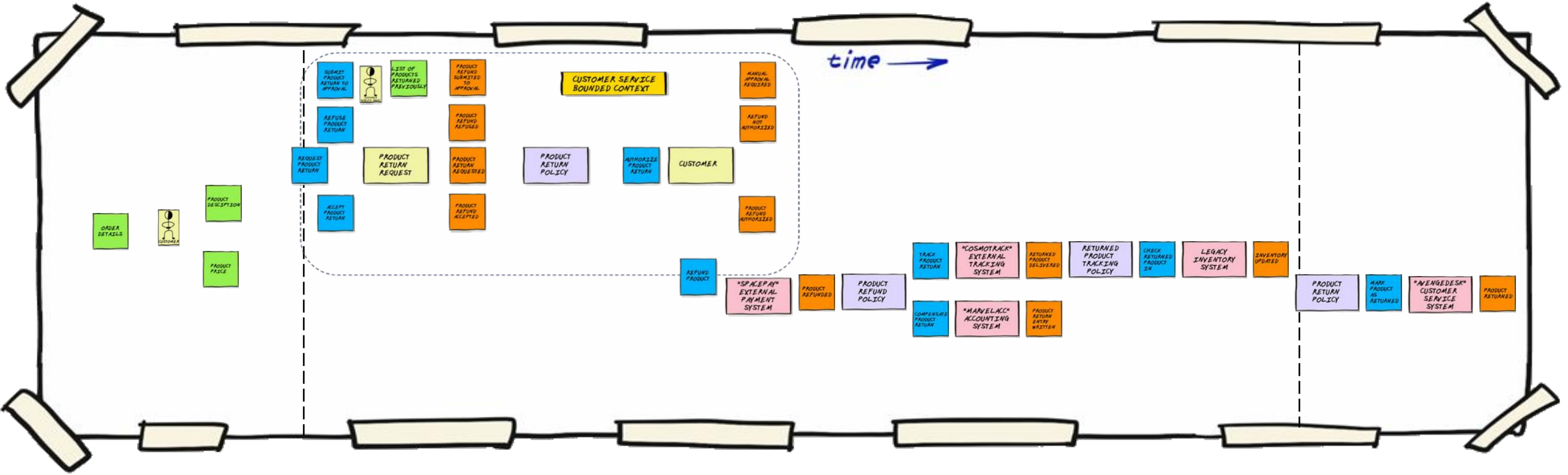


Beware, the word aggregate often scares business experts out of the room

Add the **Aggregate** sticky to the **legend**



Software design with aggregates



Preconditions **The flow to discover** **Outcomes**

CUSTOMER SERVICE BOUNDED CONTEXT

REQUEST PRODUCT RETURN

ACCEPT PRODUCT RETURN

REFUSE PRODUCT RETURN

SUBMIT PRODUCT RETURN TO APPROVAL

PRODUCT RETURN REQUEST

LIST OF PRODUCTS RETURNED PREVIOUSLY

PRODUCT RETURN REQUESTED

PRODUCT REFUND ACCEPTED

PRODUCT REFUND REFUSED

PRODUCT REFUND SUBMITED TO APPROVAL

PRODUCT RETURN POLICY

PRODUCT REFUND AUTHORIZED

REFUND NOT AUTHORIZED

MANUAL APPROVAL REQUIRED

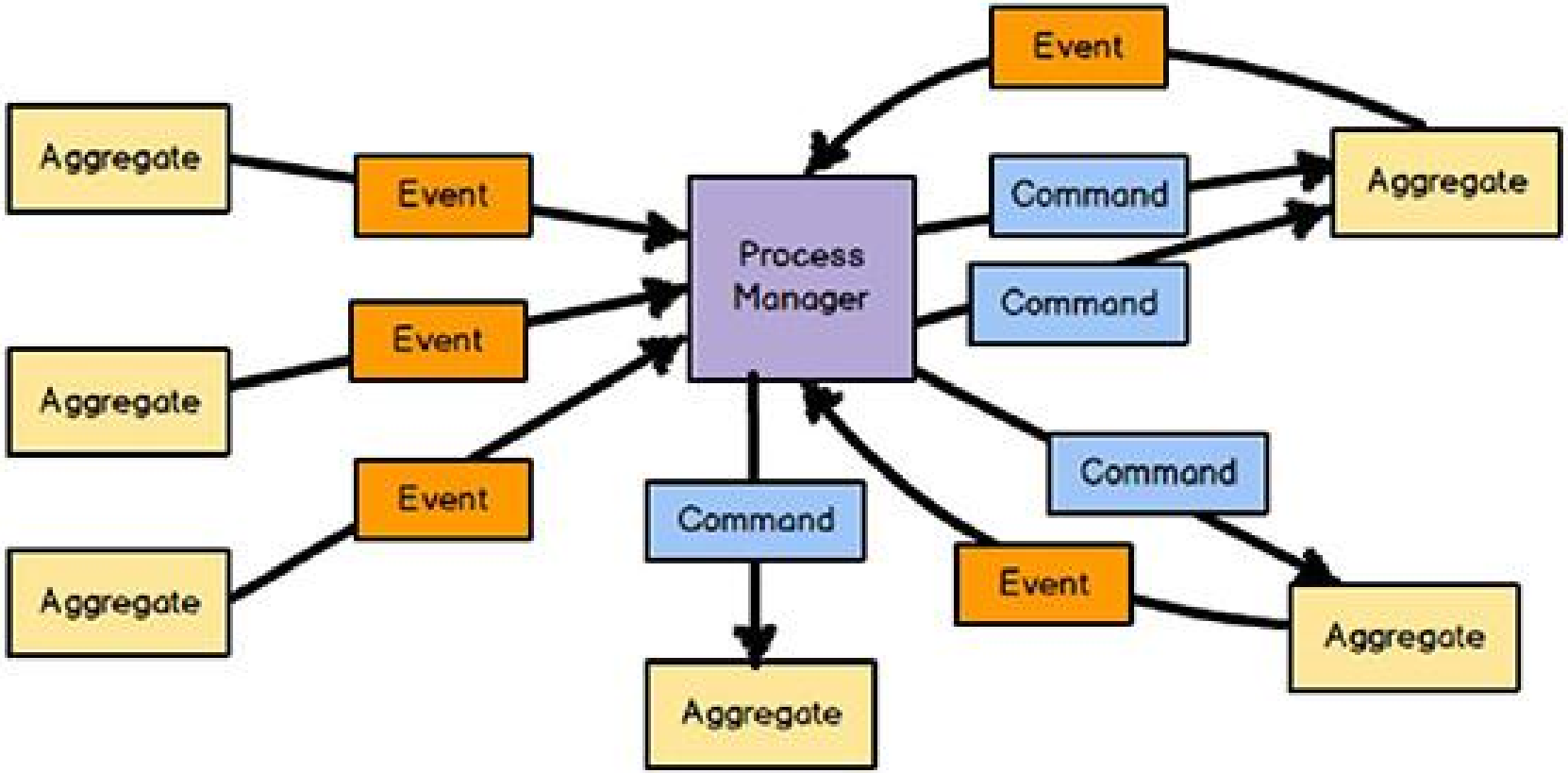
CUSTOMER HISTORY

AUTHORIZE PRODUCT RETURN

REFUND PRODUCT

"SPACEPAY" EXTERNAL PAYMENT SYSTEM

Process Manager and Aggregates



@VaughnVernon https://www.infoq.com/articles/modeling-uncertainty-reactive-ddd/

Software Design EventStorming

Good time to rewrite domain events
Use pedantic semantic precision
Symmetries are telling something

Always challenge the model
There is no over-design
Throwing wrong things away is good
Don't fall in love with your intuition

Hard problems don't have an obvious solution
Badly need a Product Owner

Consistency

This Photo by Unknown Author is licensed under CC BY-NC

Your super power on steroids

Come prepared
Clarify the objective with the sponsor / business owner first

Start with an ice breaker
Don't push your own assumptions
Ask a lot of questions

Ask even more questions

Improve your facilitation skills
(another talk on that subject is coming)
Mix it up with other tools you know

EventStorming comes in different flavors

As-Is vs To-Be
Exploration vs
explaination

Event Modeling
by Adam Dymitruk
https://eventmodeling.org

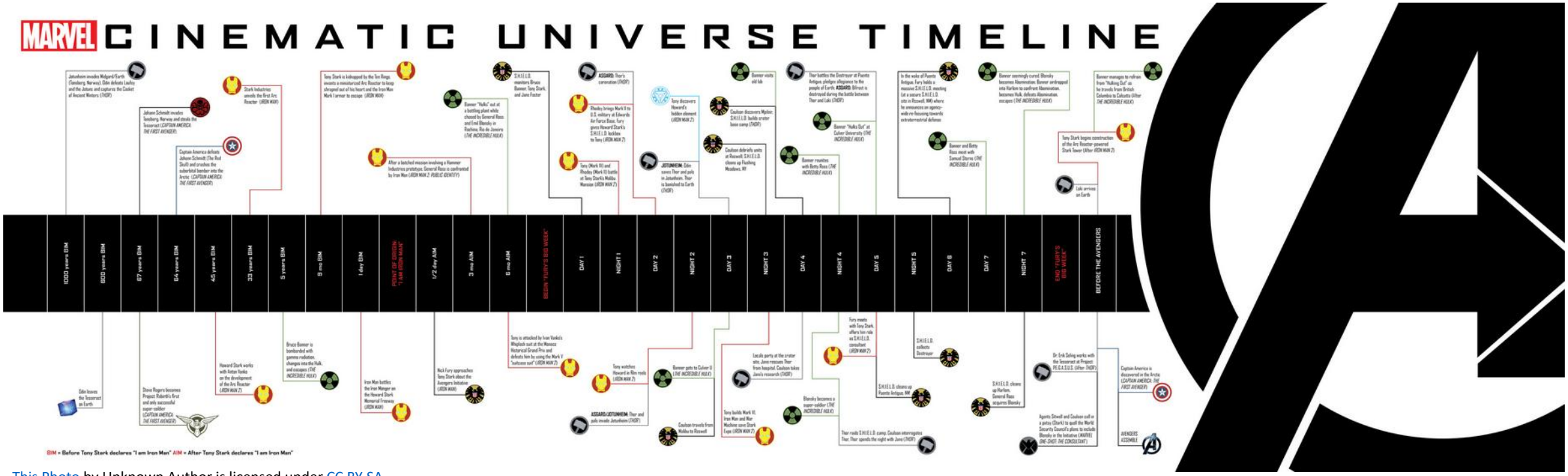
Tweak it
Add your own toppings

EventStorming
Recipes
Book is coming

This Photo by Unknown Author is licensed under CC BY-SA

My personal recipe for gathering feedback

You can even model the entire Marvel Cinematic Universe



This Photo by Unknown Author is licensed under CC BY-SA

Learn more

Official website
https://www.eventstorming.com

LeanPub Book
https://leanpub.com/introducing_eventstorming

Github repo
https://github.com/mariuszgil/awesome-eventstorming

Blog
https://medium.com/@ziobrando
http://ziobrando.blogspot.com
https://blog.redelastic.com/corporate-arts-crafts-modelling-reactive-systems-with-event-storming-73c6236f5dd7
https://medium.com/@wjptak/event-storming-summit-2018-9efd6c0b7d0d

CONTACT

AGILE PARTNER

20A. RUE DU PUITS ROMAIN
L8070 BERTRANGE
Z.A.I BOURMICHT

(+352) 26 37 00 30

contact@agilepartner.net
jobs@agilepartner.net