

# CO-DESIGNING THE DIGITAL MACHINE

How to use models to get the entire product team on the same page about what you are building

Tim Sheiner

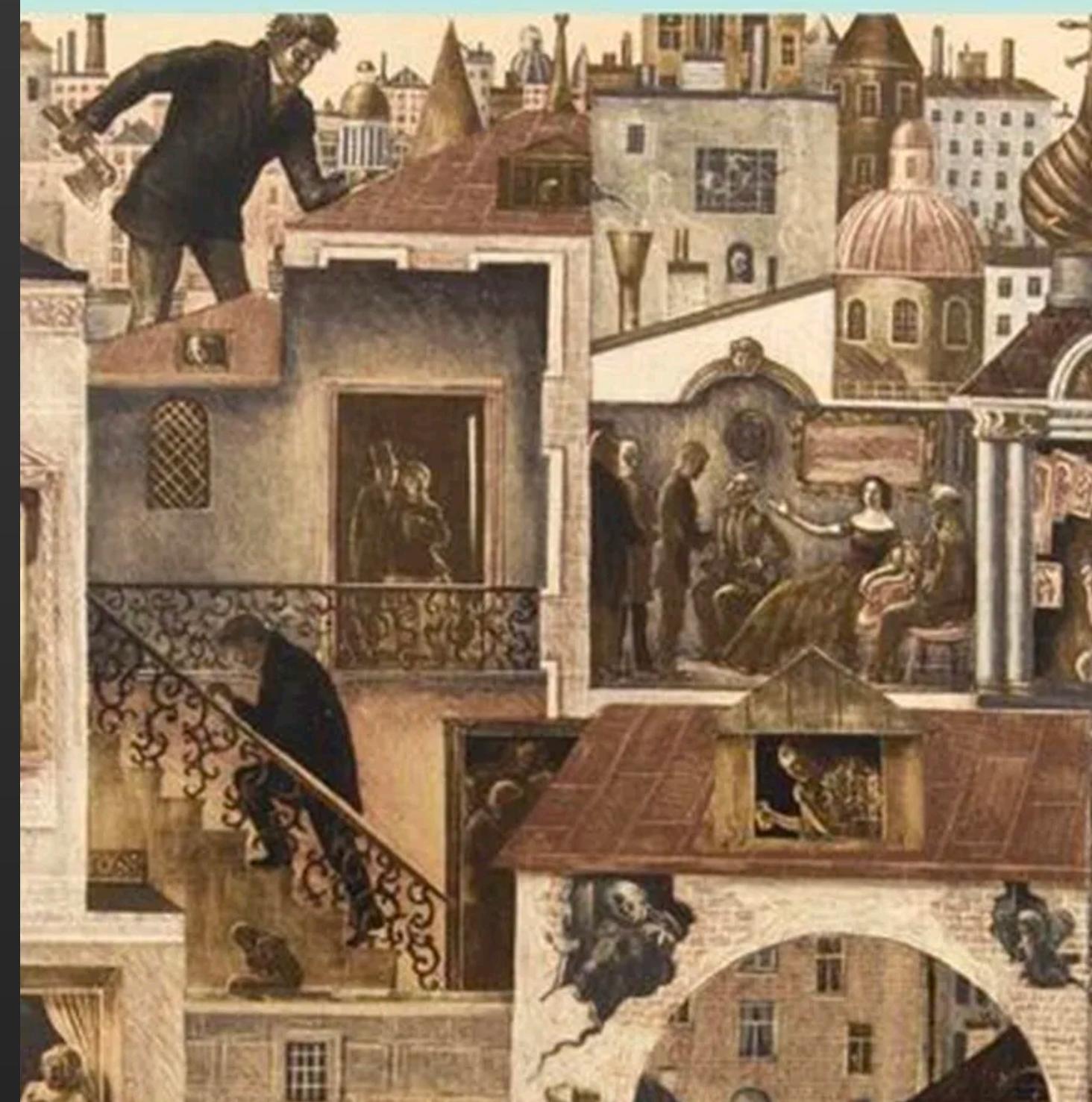
# In 1988...

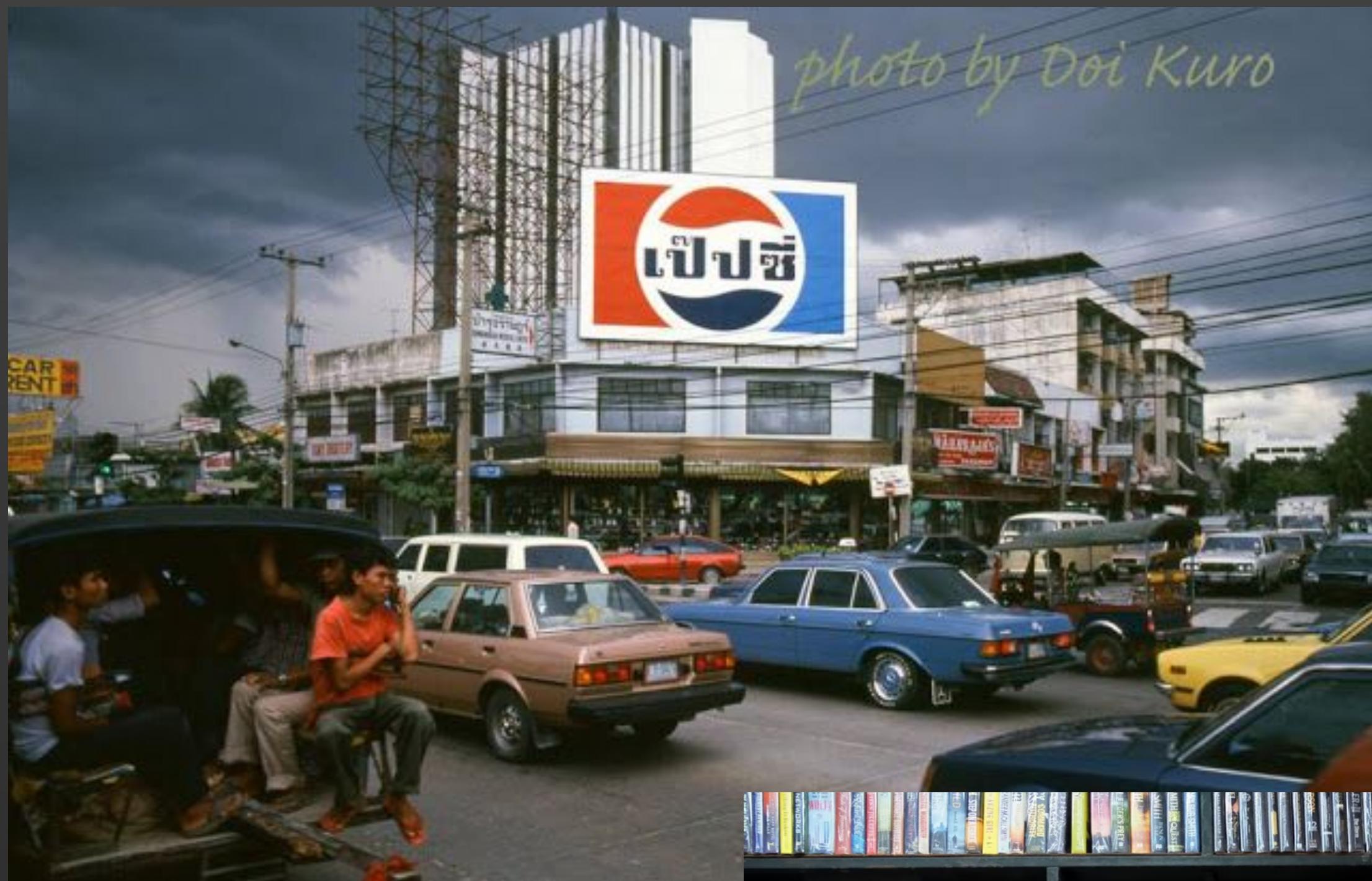




# Crime and Punishment

FYODOR DOSTOEVSKY





# **TRANSLATION IS HARD**

1. Difficult to move ideas from one form to another
2. Only an expert can tell when it is done badly

What's translation got to do  
with building software?

**Building software requires 2  
kinds of translation!**

# 1. Analog to Digital



User

analog value  
↔  
digital workflow



Product

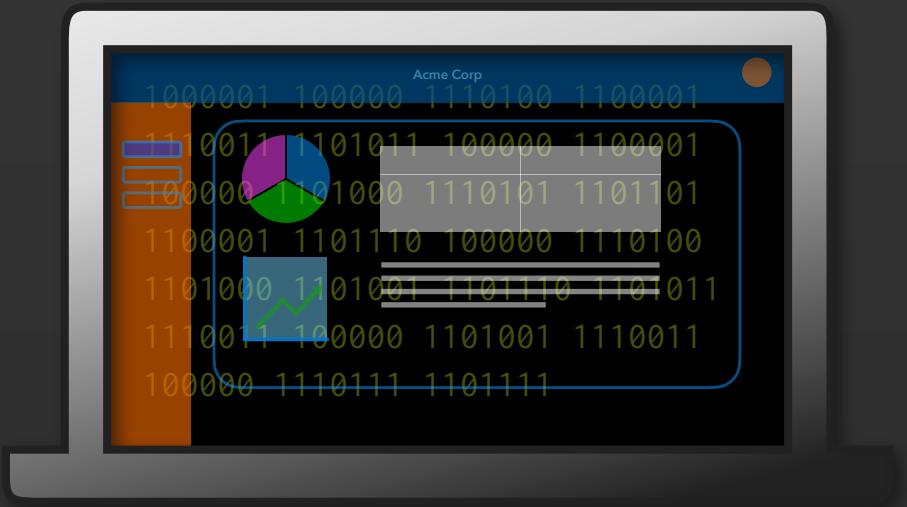
## 2. Stakeholder to Stakeholder



Product Mgr



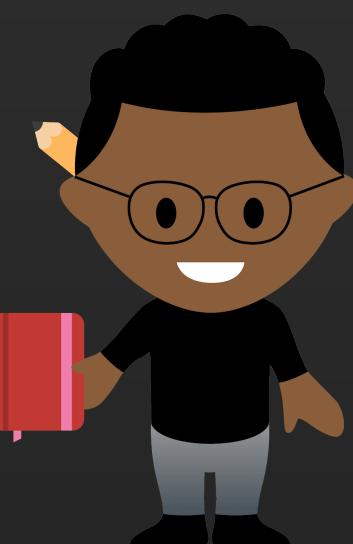
Engineer



Product



User

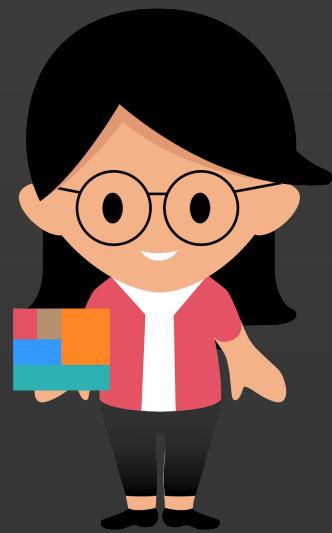


Designer

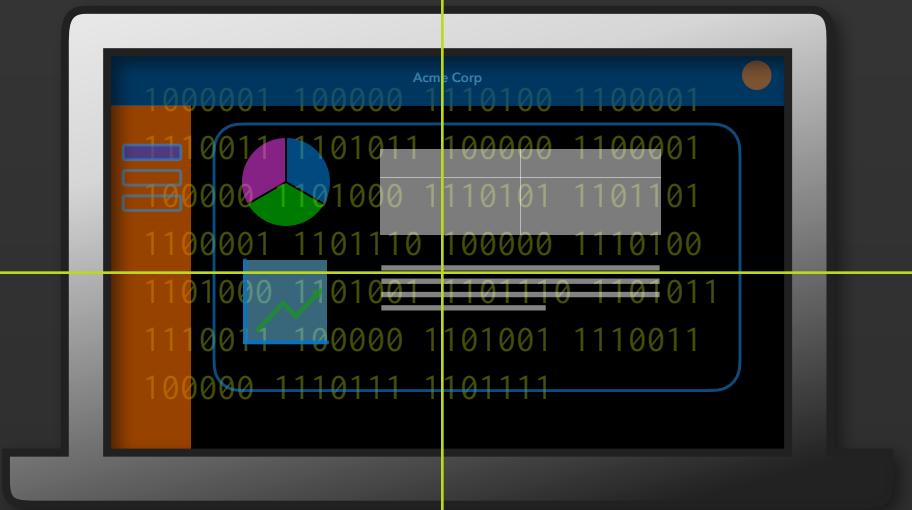
## 2. Stakeholder to Stakeholder



Product Mgr



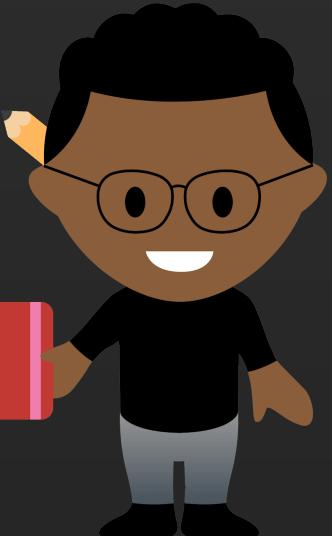
Engineer



Product



User

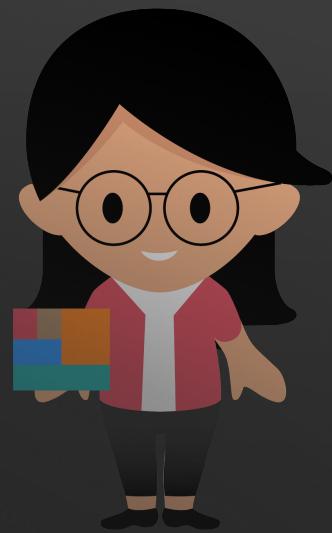


Designer

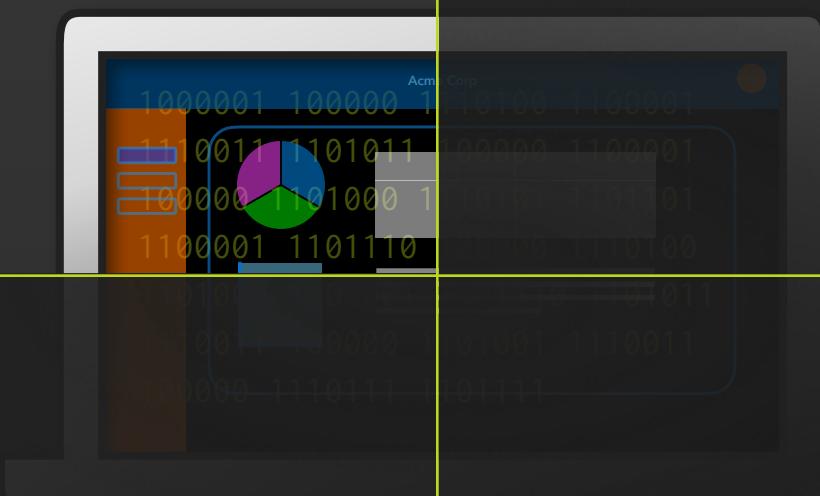
## 2. Stakeholder to Stakeholder



Product Mgr



Engineer



Product



User



Designer

# DIFFERENT CONCERNS

is why stakeholder translation is hard.

# what to do?

if you can't fix it, feature it

# Concern => Defensiveness & Protection

**My primary concern  
is value**



Product Mgr

**I am the object  
of everyone's  
concerns**



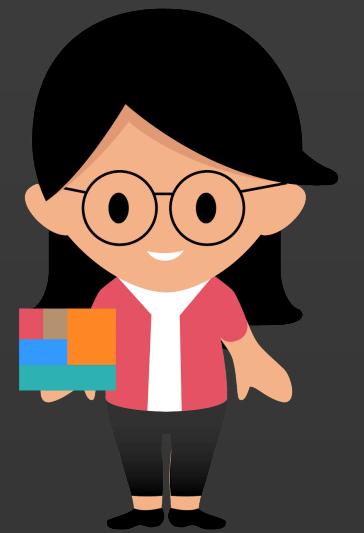
Product

**My primary concern  
is my needs**



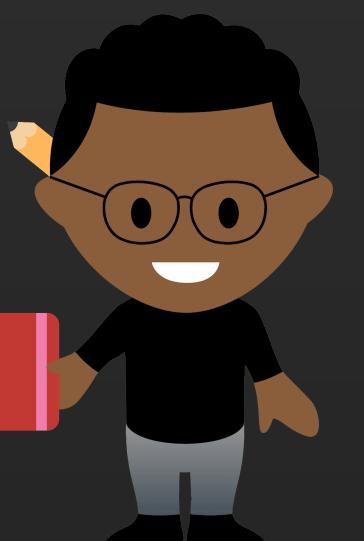
User

**My primary concern  
is data**



Engineer

**My primary concern  
is interactivity**



Designer

# Ownership => Partnership & Trust

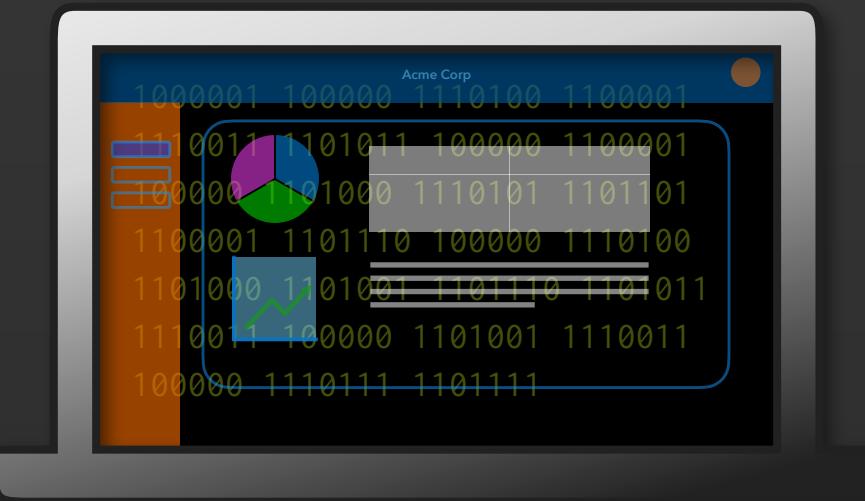
I own the  
**Value Model**



Product Mgr

Everyone owns the

**Object Model**



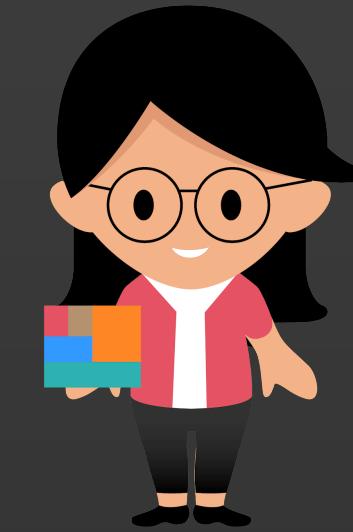
Product

I am the  
**Persona Model**



User

I own the  
**Data Model**



Engineer

I own the  
**Interaction Model**



Designer

# The Digital Machine

## Value Model

why is it useful?

## Data Model

what is its state?

## Object Model

how is it organized?

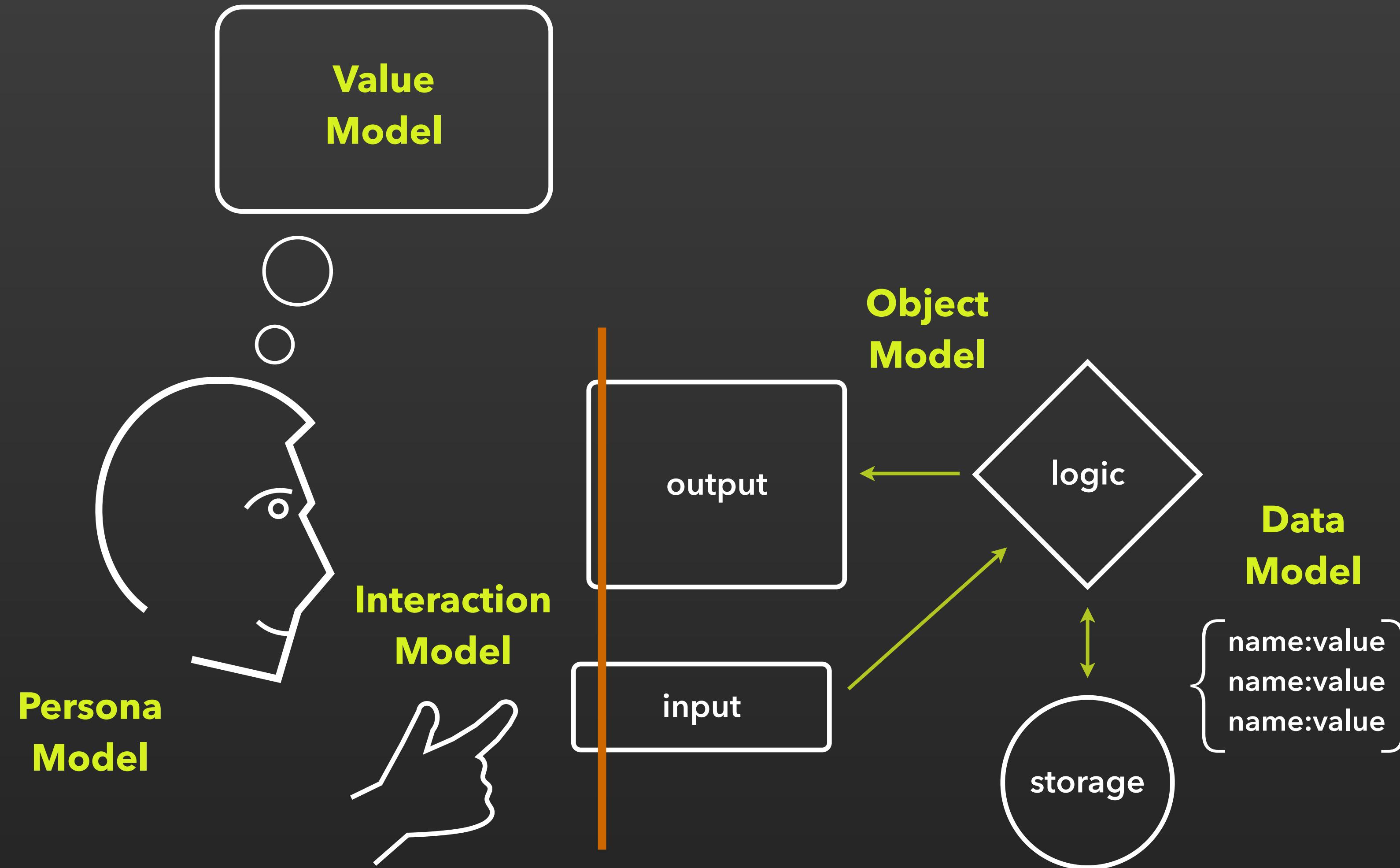
## Persona Model

who is it for?

## Interaction Model

how do I use it?

# The Digital Machine



# Persona Model

## Thomas



### EVANGELIZER

Represents the user who has brought Jut into his organization, set it up and is working to evangelize it. He measures his success in terms of how effectively he helps others do their jobs well.

#### Details

- Despite the job title, Thomas is the architect of the company infrastructure
- Primary responsibility is keeping the system running smoothly and efficiently
- Top skills for his job is being a UNIX guru and having a gut feel for troubleshooting

### I care about stuff before it breaks... most people don't.

No one understands our company infrastructure as well as I do. I see it like a living system, with quirks, habits and behaviors. Even so I could really use one tool that would organize and centralize all of the monitoring I do—not having an easy way to get a high level view does wears on me over time.

I also need to be able to share my understanding because the most important part of my job is making it easier for other people to do their jobs. The ops guys need dependable and meaningful alerts and dashboards. The devs need to understand how to optimize their code to run most efficiently on our systems.

I love helping these people but I need to make their access to system behavior more self-serve because there just aren't enough hours in the day.

#### Needs

- Ways to help dev and ops do their jobs better
- Being able to see system change
- A way to correlate cause and effect, events and metrics
- Making access to an understanding of system behavior be self-serve

#### Frustrations

- Constant task switching
- Never being sure what tomorrow's top priority will be
- Dealing with system failures caused by devs not understanding what will break the system

The persona model engages stakeholder

**EMPATHY**

# Value Model

**Product**

**is a category of experience**

**that provides me with these benefits**

**(and is better than the alternative because...)**

# Value Model

Amazon  
is an e-commerce website  
that provides me with immediate gratification  
(and is better than individual product websites because  
I don't have to create a new account and learn a new  
website)

The value model is

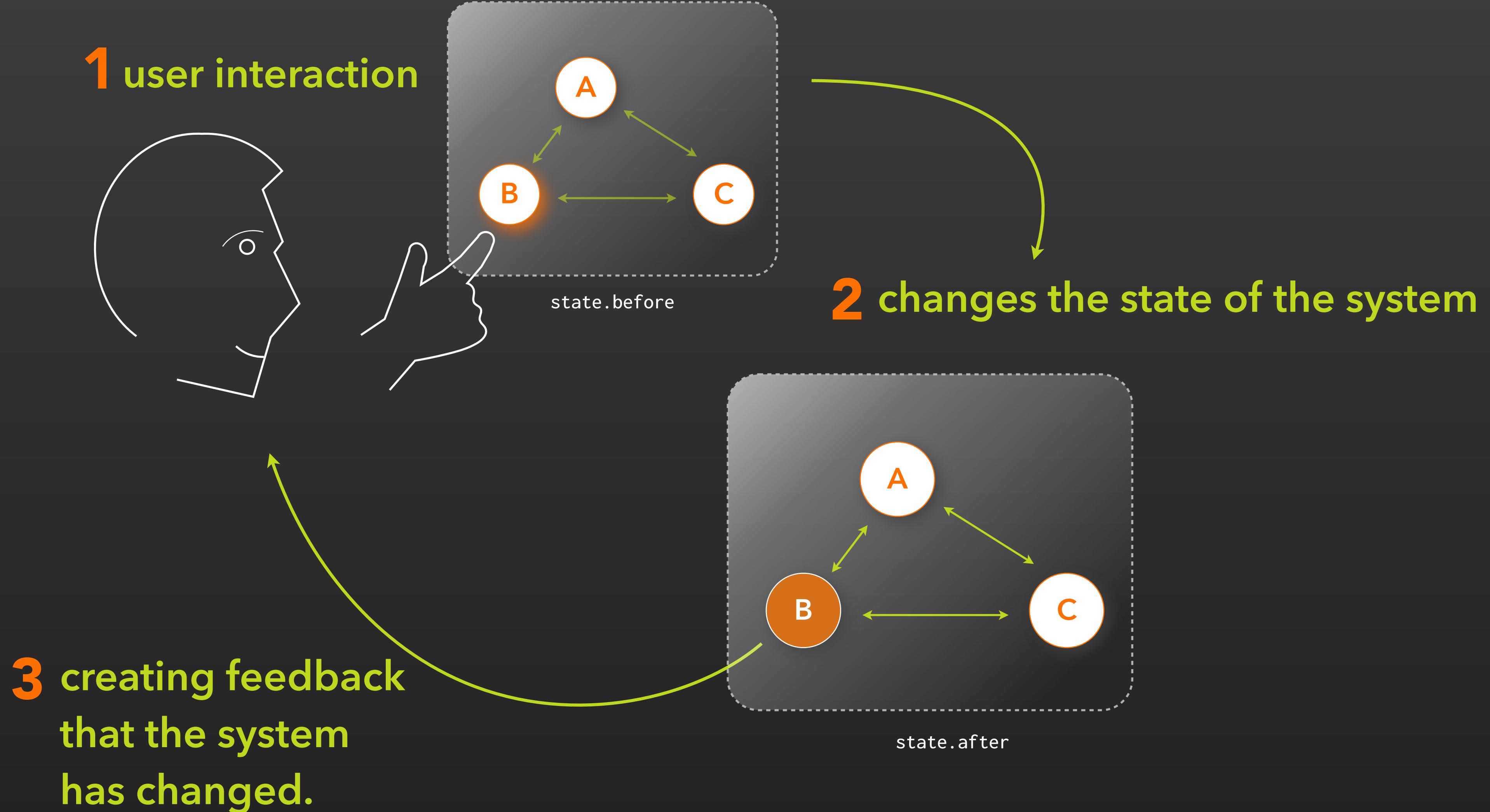
# **VALUE FOR THE PERSONA**

not for the business!

# Interaction Model

# Interaction Model

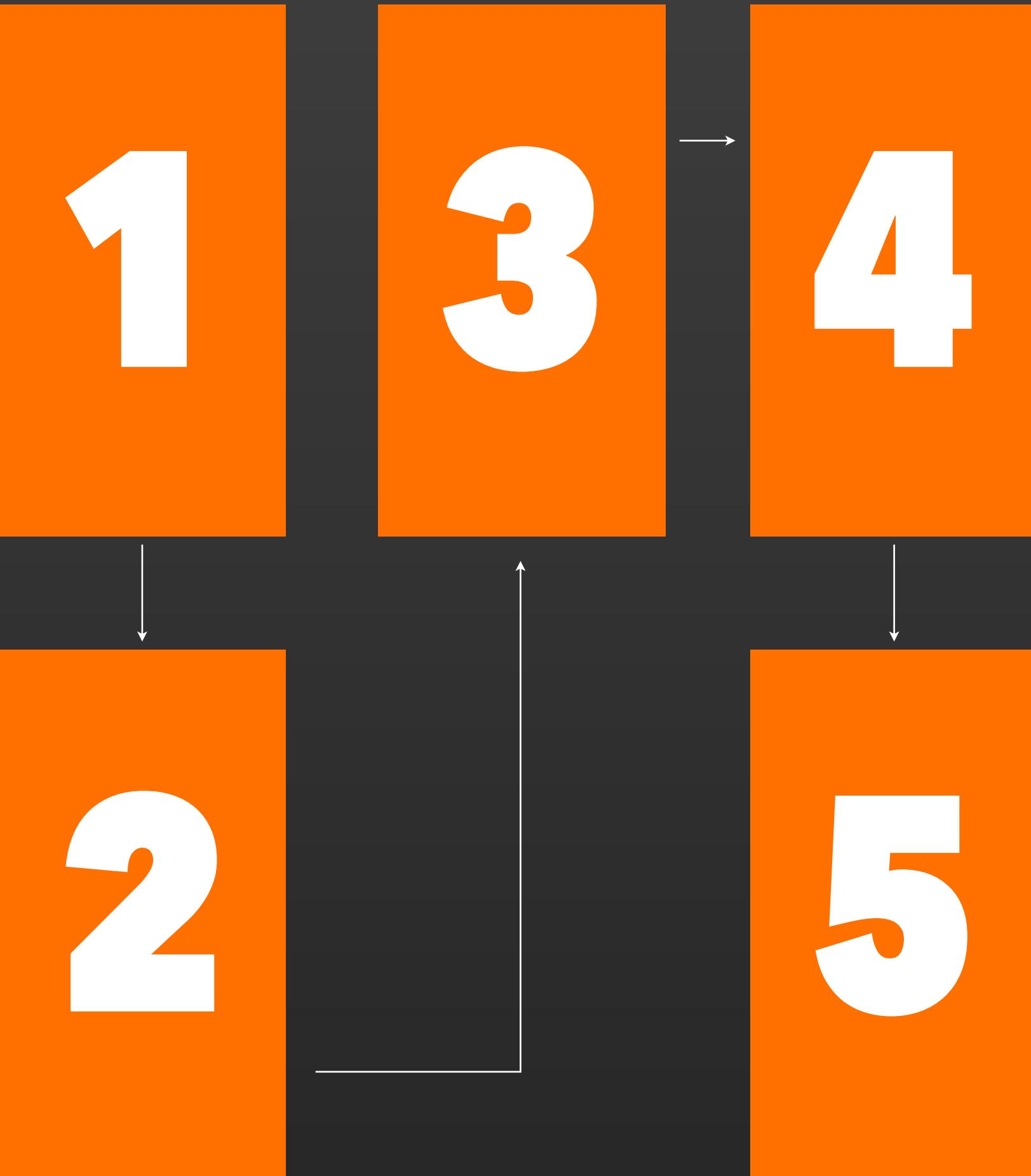
is the mechanism for changing state



# Interaction Model

is designed by telling stories

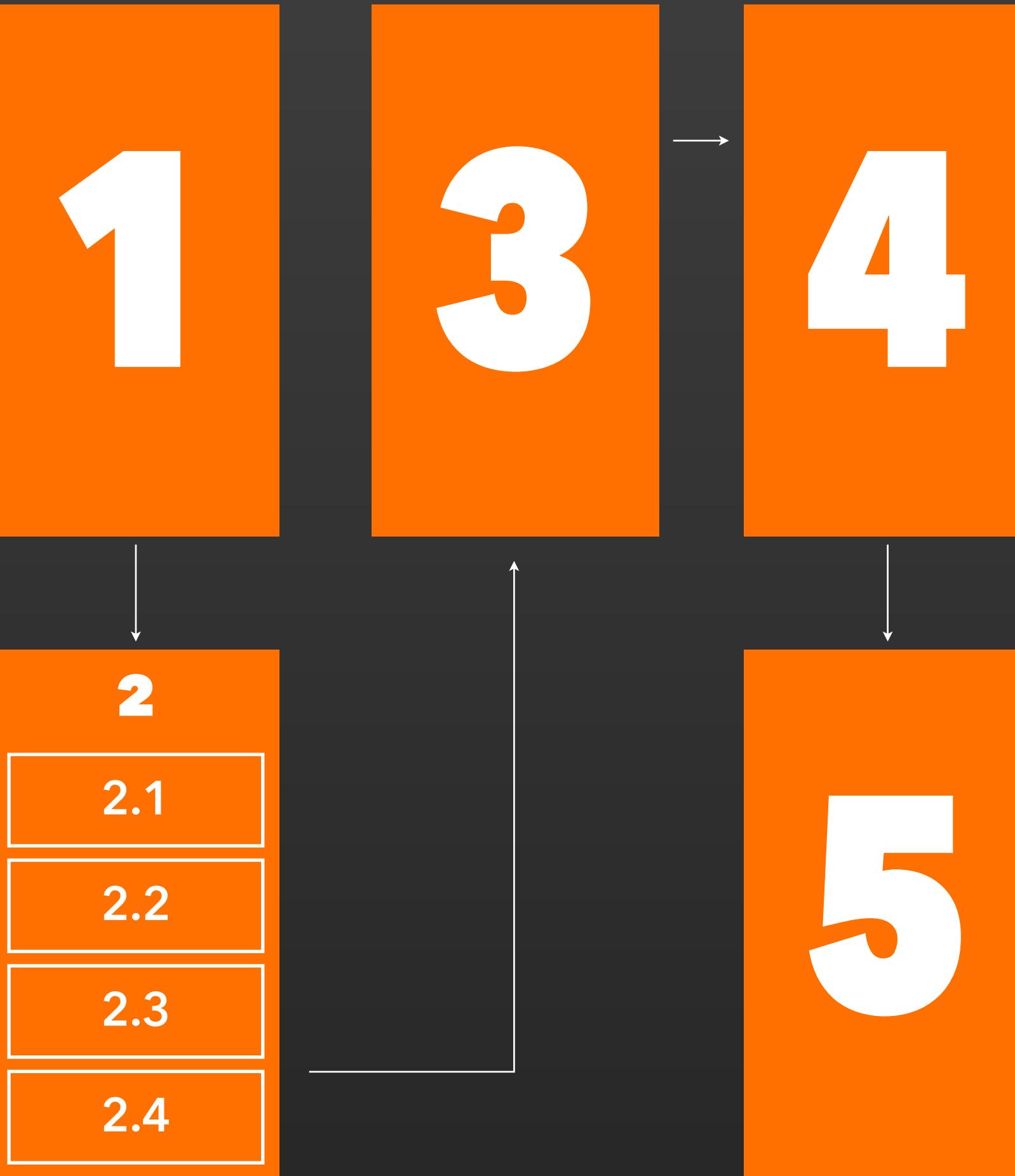
1. Go to amazon.com
2. Find a book
3. View the details for a book
4. Add the book to my shopping cart
5. Checkout



# Interaction Model

is designed by telling stories

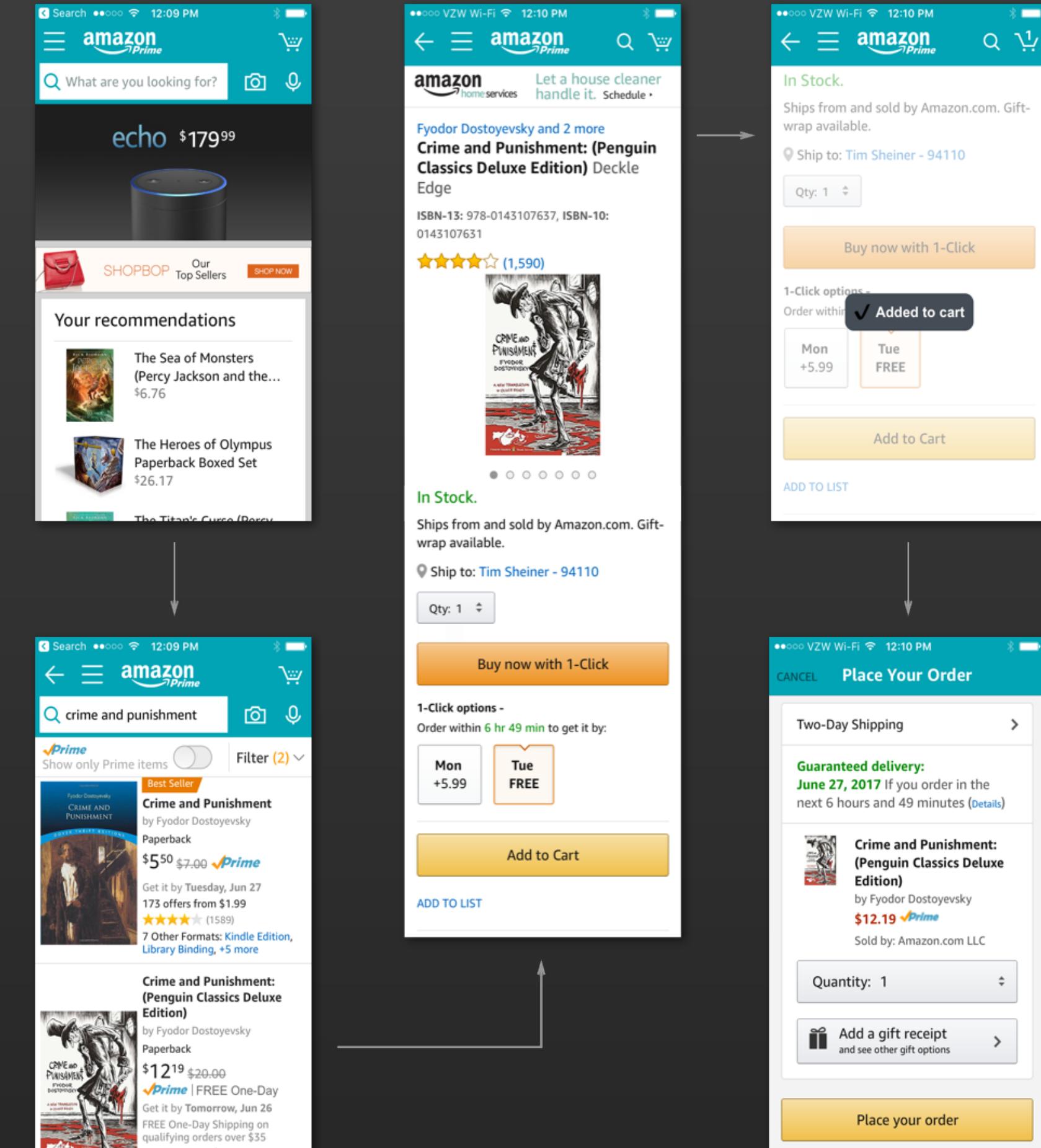
1. Go to amazon.com
2. Find a book
  1. Enter search string
  2. Press Return
  3. View Found Set
  4. Select from Found Set
3. View the details for a book
4. Add the book to my shopping cart
5. Checkout



# Interaction Model

is designed by telling stories

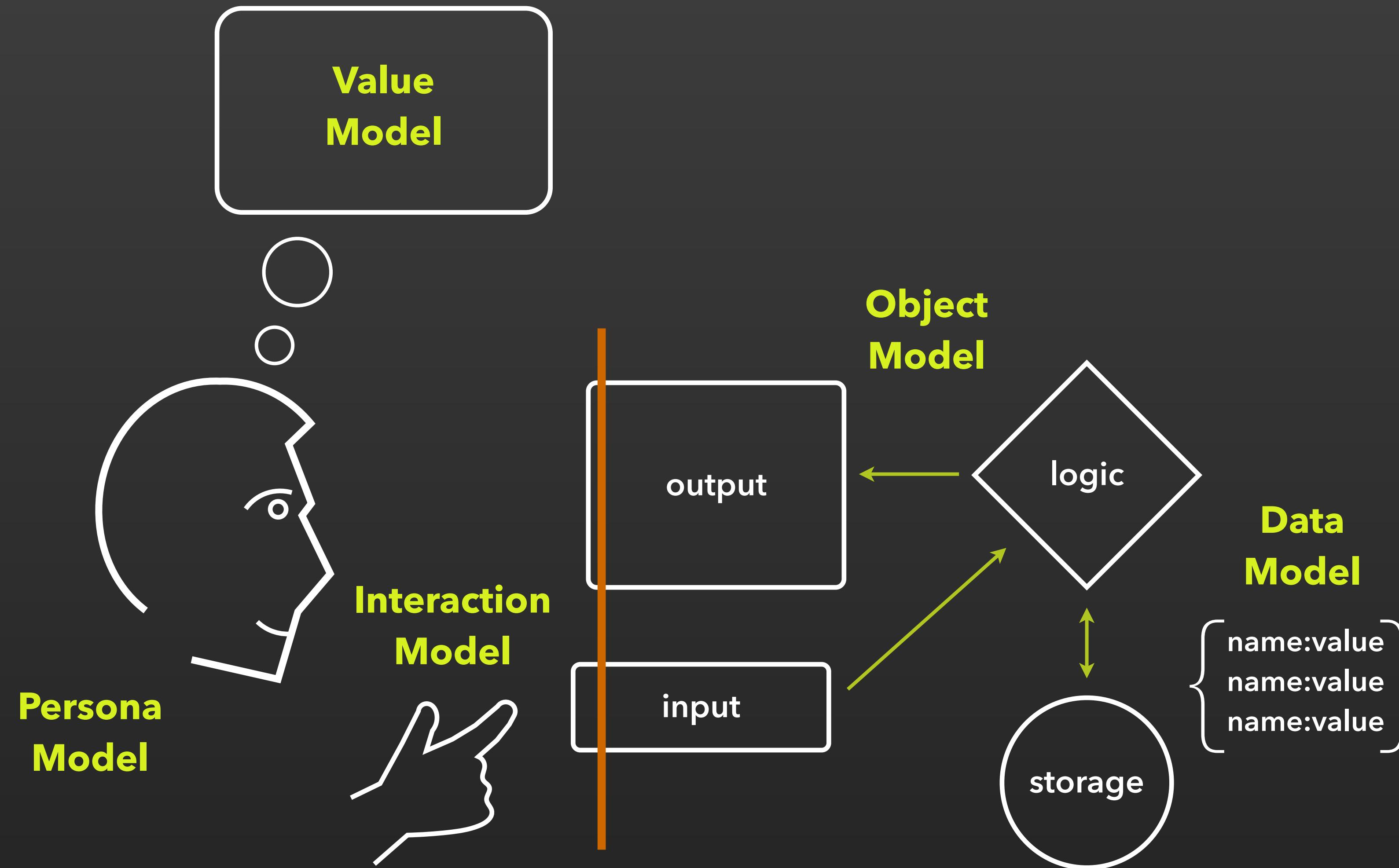
1. Go to amazon.com
2. Find a book
  1. Enter search string
  2. Press Return
  3. View Found Set
  4. Select from Found Set
  3. View the details for a book
  4. Add the book to my shopping cart
  5. Checkout



The interaction model is a

**STORY**

# The Digital Machine

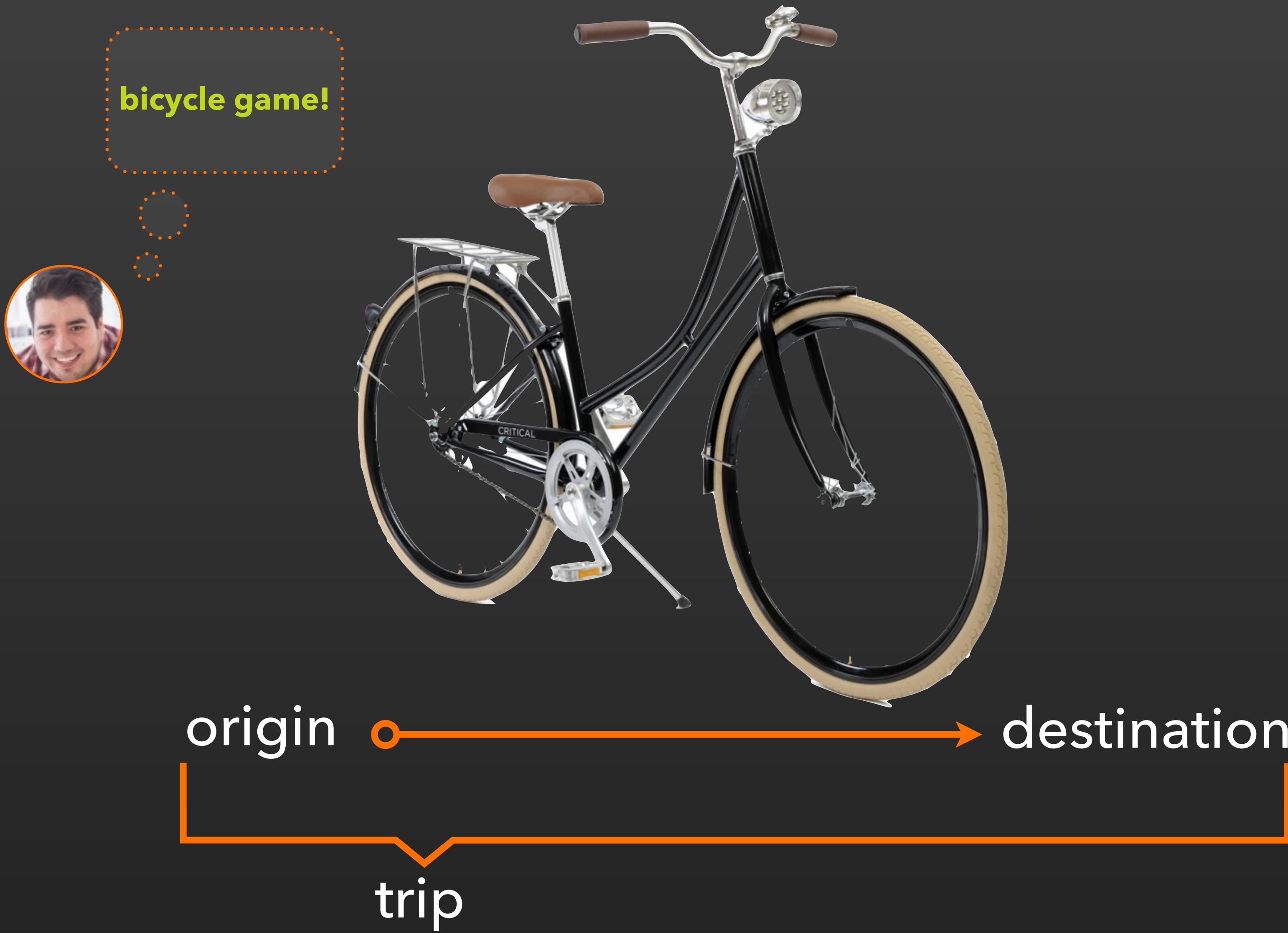


# Object Model

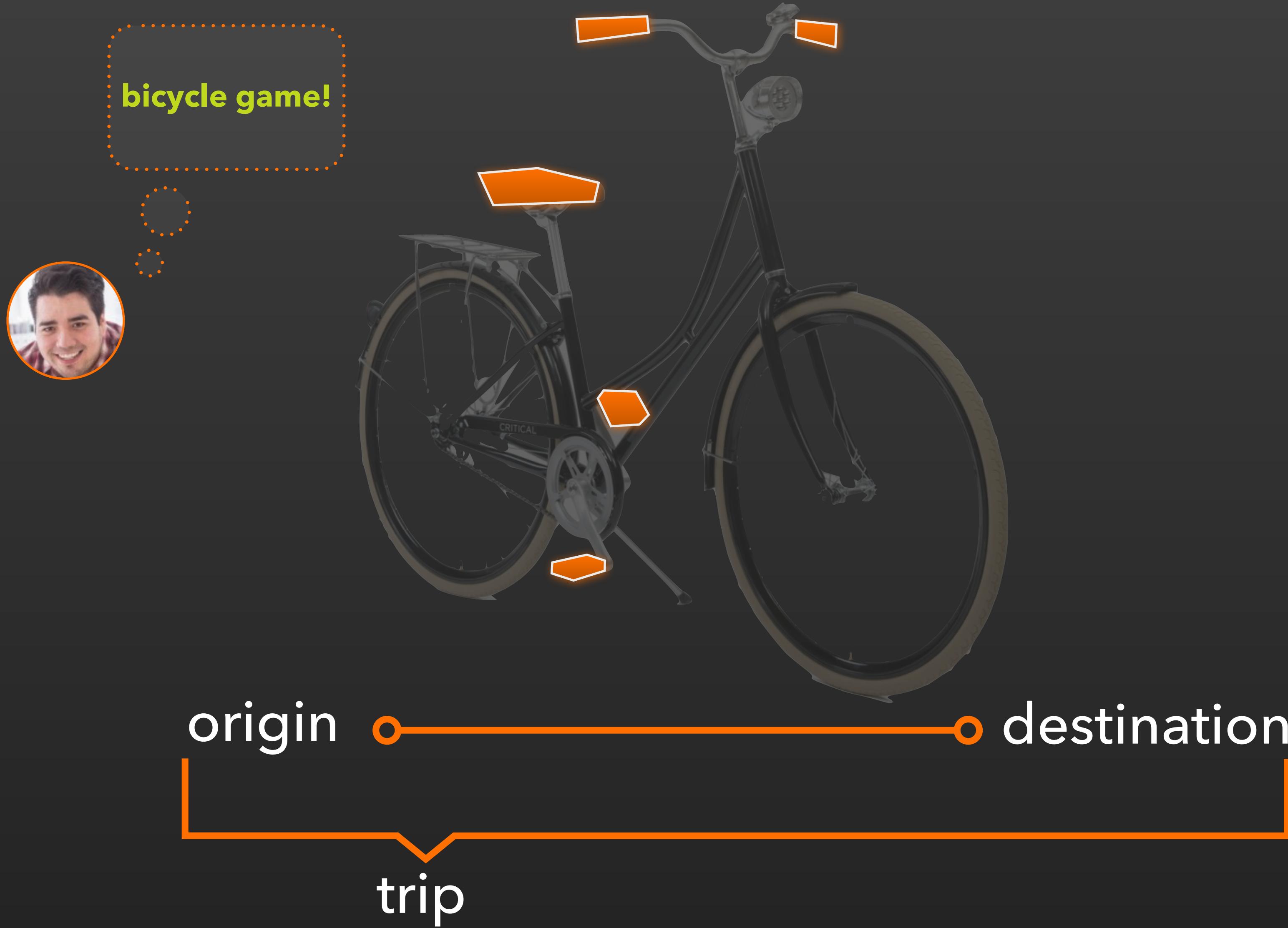
Imagine we are designing a game

about traveling by bicycle.

# Object Model



# Object Model



# Object Model

Describe your system: the nouns are the object model

The **rider** rides  
the **bicycle** by sitting on  
the **seat**,  
holding the **grips**  
and pushing the **pedals** to get from  
an **origin** to  
a **destination** to complete  
a **trip**.

# Object Model

More compactly, without the verbs

```
trip = {  
    vehicle : {  
        rider,  
        bicycle : {  
            grips,  
            seat,  
            pedals  
        },  
    },  
    origin,  
    destination  
}
```

The object model defines the

# **NAMES & RELATIONSHIPS**

of the things in the system

# Data Model

key : value

x: 3

$\pi$ : 3.14

color: 'green'

city: 'san francisco'

```
lat: '37.78',  
long: '-122.42'
```

```
position: {  
  lat: '37.78',  
  long: '-122.42'  
}
```

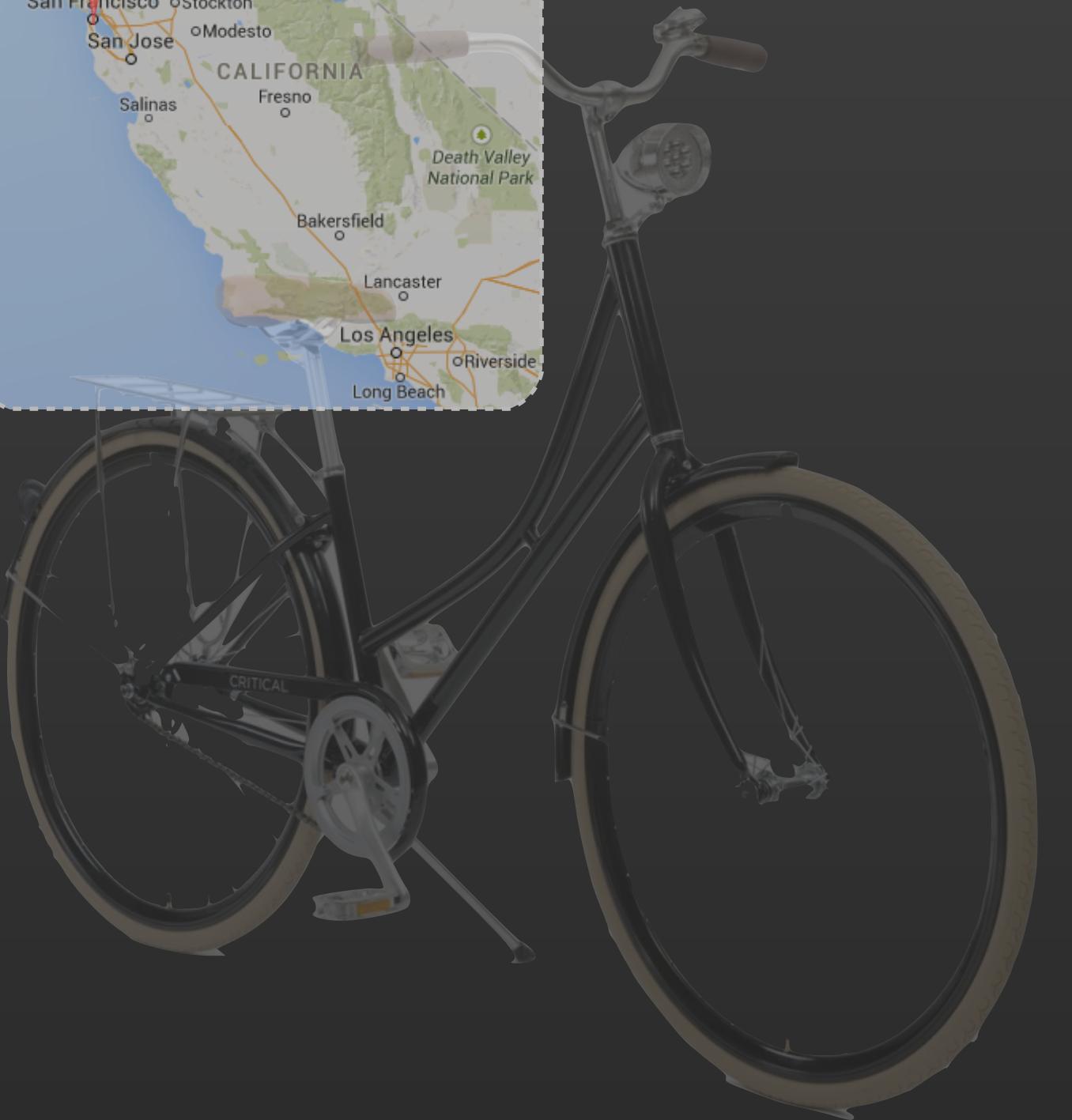
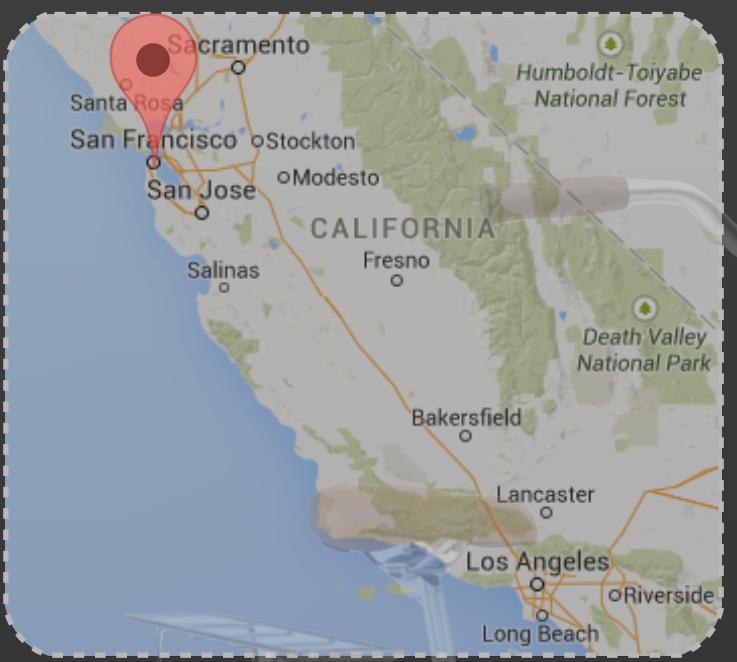
```
trip = {  
    vehicle : {  
        rider : {  
            name : 'Thomas'  
        },  
        bicycle : {  
            grip : {  
                rotation : '12', inclination : '3'  
            },  
            position : {  
                lat : '37.78',  
                long : '-122.42'  
            },  
            origin : {  
                name : 'San Francisco',  
                position : {  
                    lat : '37.78', long : '-122.42'  
                }  
            },  
            destination : {  
                name : 'Los Angeles',  
                position : {  
                    lat : '34.05', long : '-118.24'  
                }  
            }  
        }  
    }  
}
```

The data model keeps

**THE STATE OF EVERY  
OBJECT IN THE SYSTEM**

which is the state of the system

bicycle game!



bicycle game!



1 user interaction



3 creating feedback  
that the system  
has changed.

2 changes the state of the system

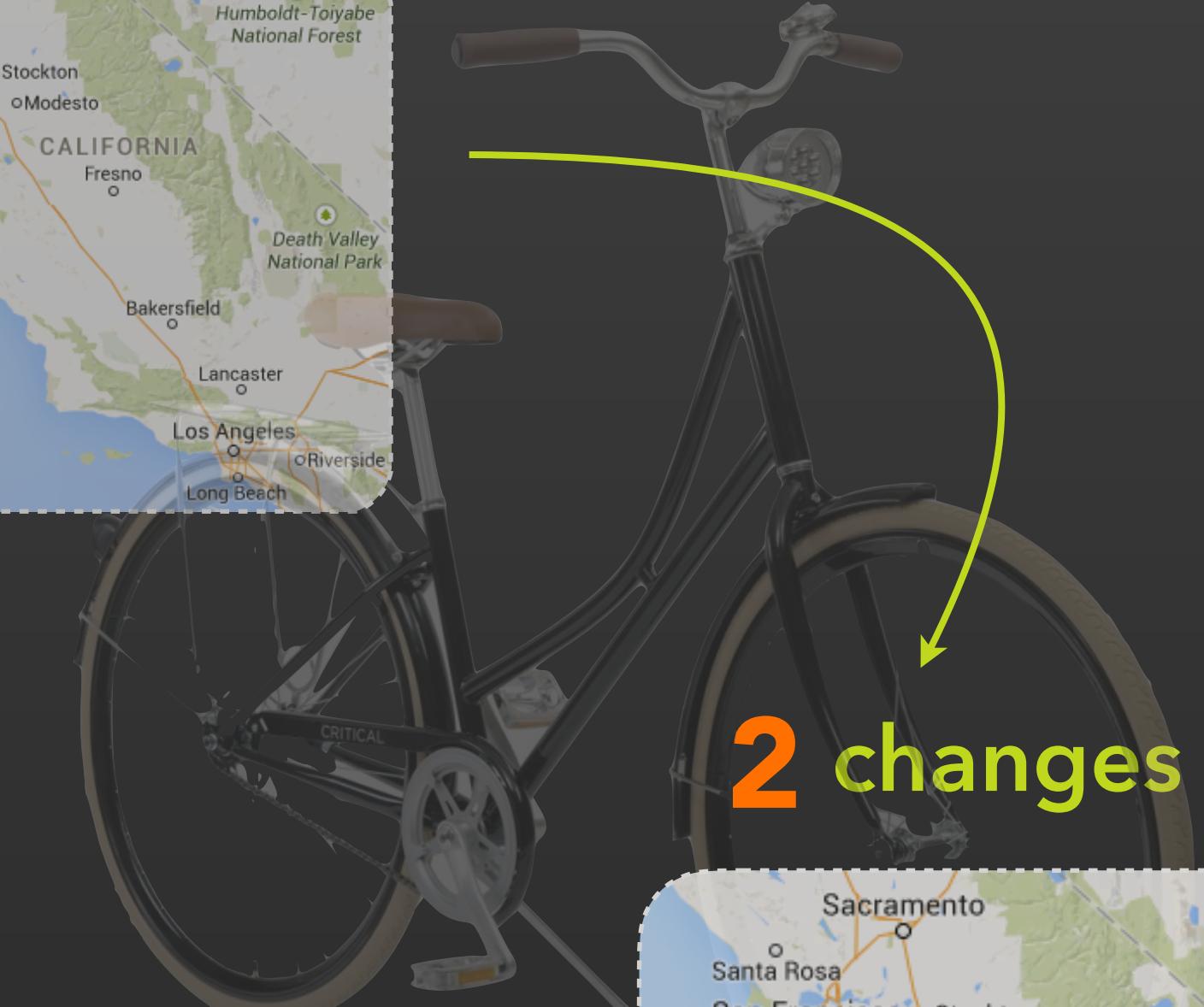
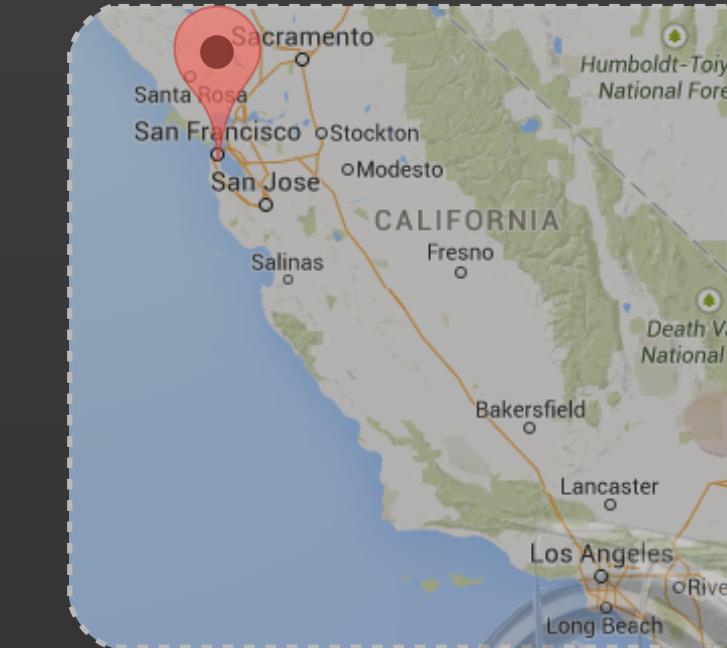
```
position : {  
  lat: '34.058', long: '-118.242'  
}
```

bicycle game!

```
state.before  
position : {  
    lat : '37.78', long : '-122.42'  
}
```



1 user interaction



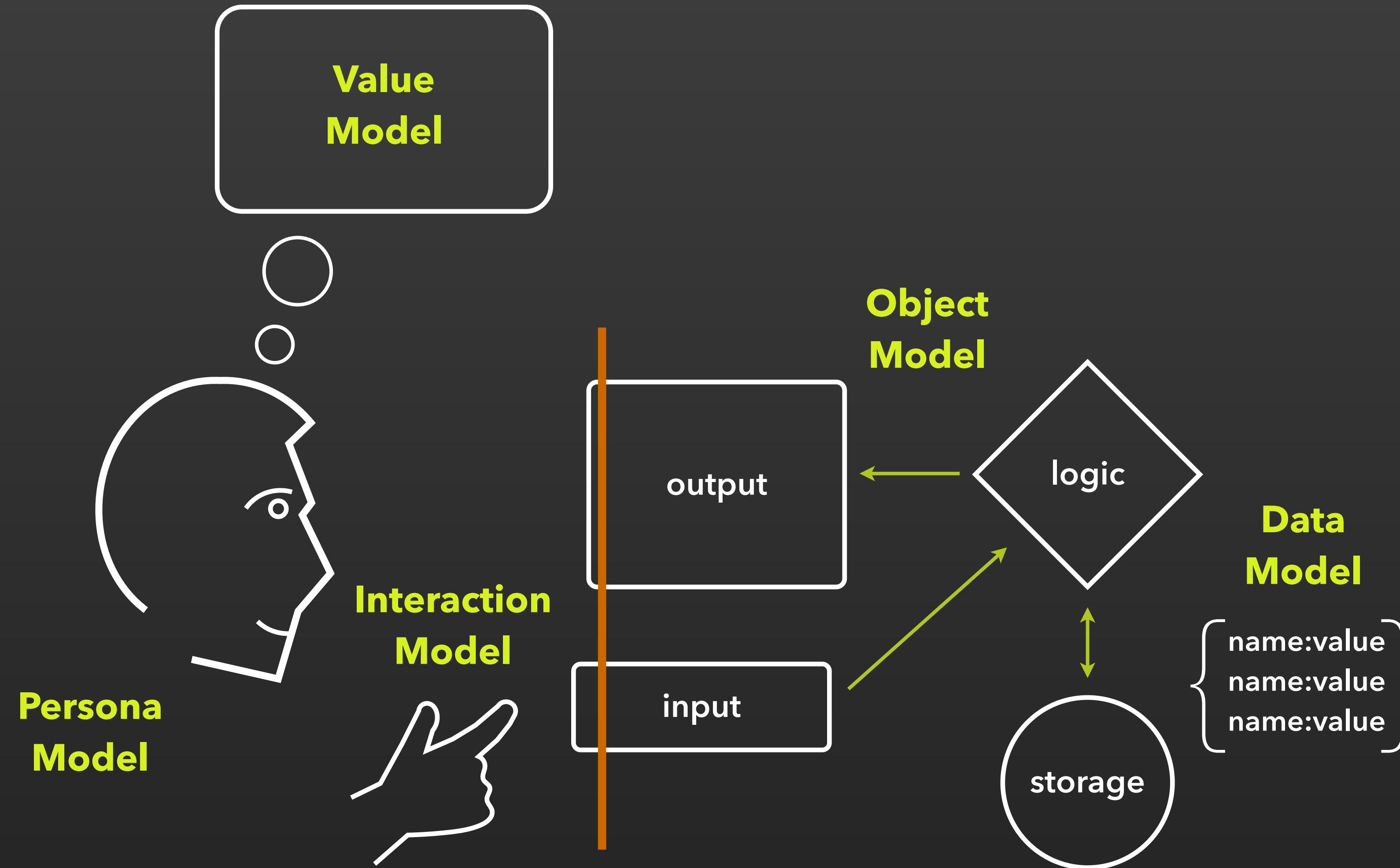
2 changes the state of the system



3 creating feedback  
that the system  
has changed.

```
state.after  
position : {  
    lat : '34.05', long : '-118.24'  
}
```

# The Digital Machine



# The Digital Machine

## Why (I think) it works

1. The communication norms established by the Digital Machine models surface the assumptions that must be aligned across project stakeholders.
2. The design process is made more tractable because it can be pursued in a parallel fashion by working on any of the models independently for some of the time and then bringing them all back together to evaluate as a holistic system.
3. Imagining the Digital Machine together engages the visual and spatial reasoning abilities of the project stakeholders as they develop a shared understanding of what they are building

# BUILD BETTER SOFTWARE FASTER?

Talk about models!

**FIN**