Faculty of Design

2014

From systems to software

Sheiner, Tim

THE DIGITAL MACHINE

Converting Systems to Software

Tim Sheiner

RSD3.Oslo.16.Oct.

Thanks!

My Students

in the Interaction Design Program at California College of Arts who gave me a good reason to organize my thoughts

Jut

who supported my teaching and sponsored my trip to RSD3

Hugh Dubberly

my systems thinking mentor

"SOFTWARE IS EATING THE WORLD"

Marc Andreesen

wsj.com, 2011

Band K.L.King

Copyright 1934 45469 250

1st Edition

12/26/30	500	3280
10/6/38	500	4050
3rd Edition ordered Aug. 48	500	
Reprint 64	500	
Jan 1970	500	

2500

Printed by Otto Zimmerman Co, Cincinnati, O.
Plates at " " 4800

Band Galey K.L.King

Copyright 1934 45966 11/5/35 300

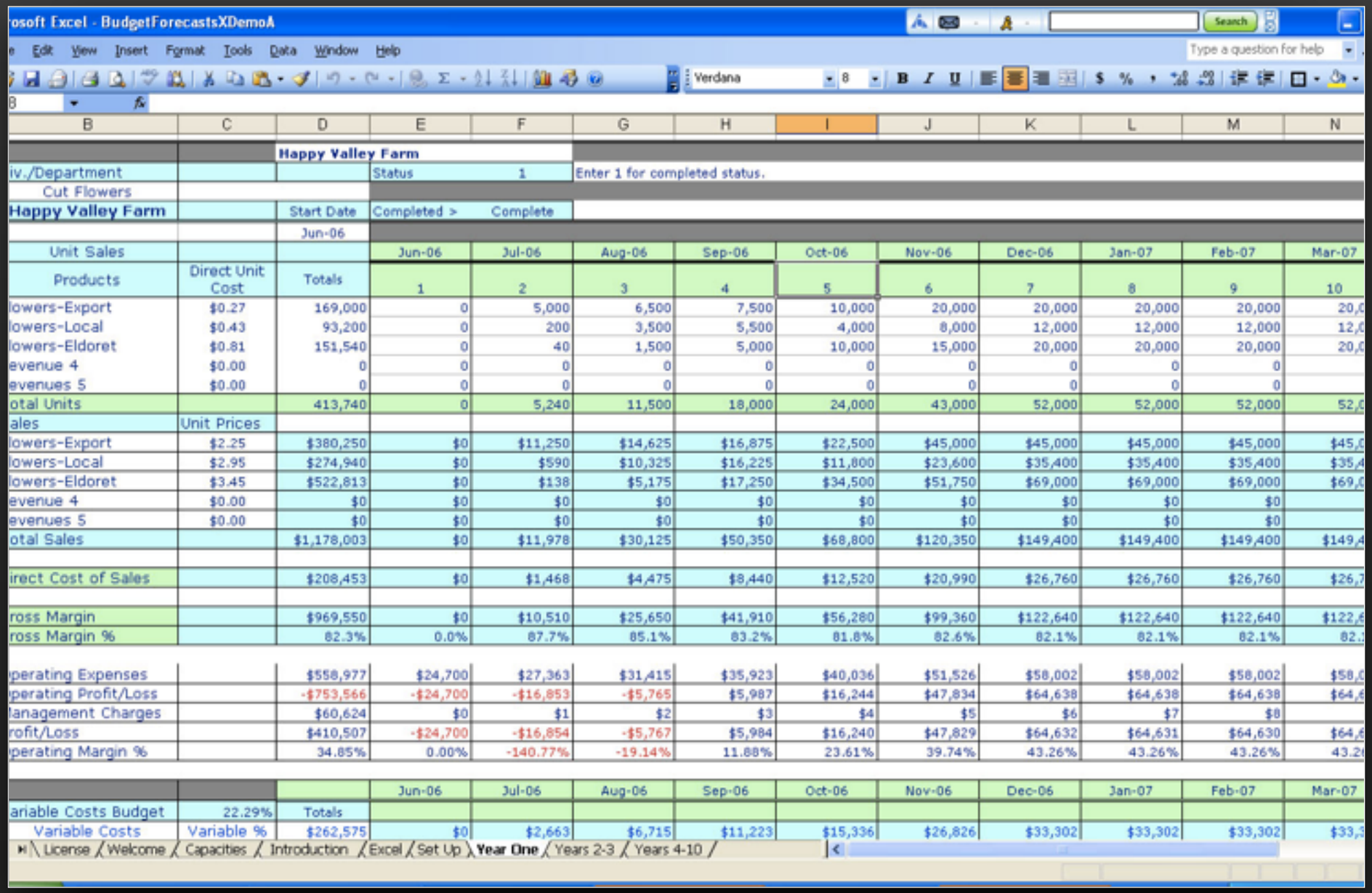
1st Edition

12/24/34 [redacted]	500	4250
2nd Ed. 1942	250	
3rd Ed. 1946	500	
4th Ed. 1950	500	
5th Ed. 1953	500	
6th Ed. 1961	500	
7th Ed. Nov. 1968	500	

3250

Printed by Rayner, Dalheim Co, Chicago, Ill.
Plates at " " " " " "
4800

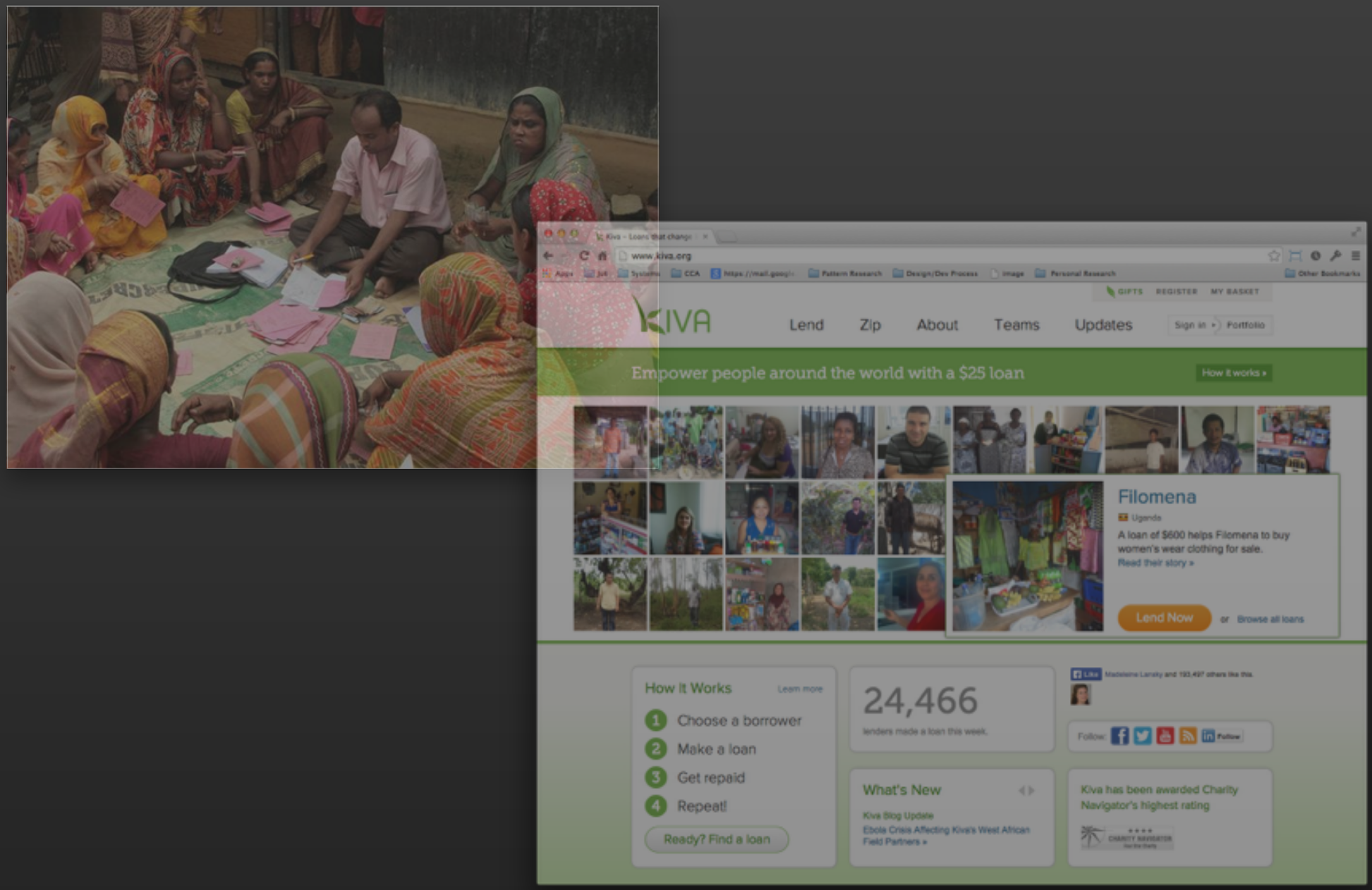
Fully Digested



The New Fast Food

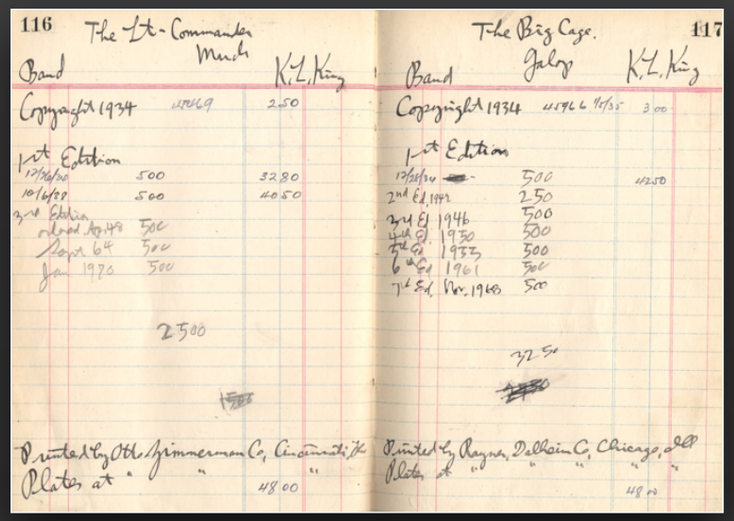
Invent a New Cuisine

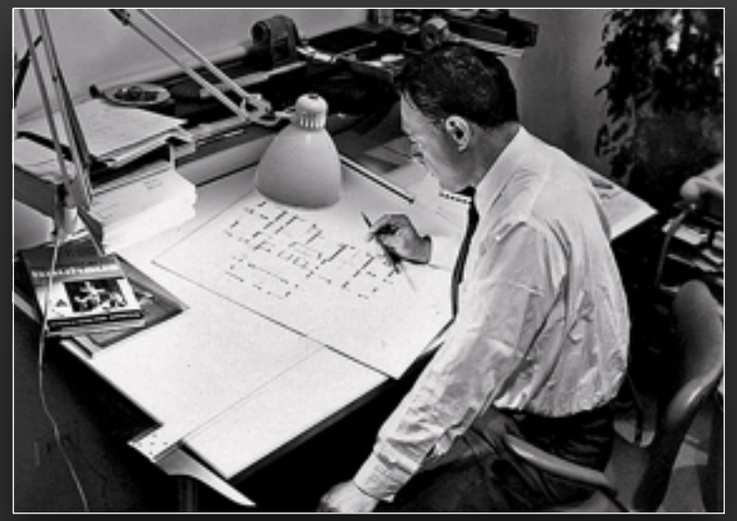
Invent a New Cuisine



HOW?

Analyze The Purposeful Activity[1]



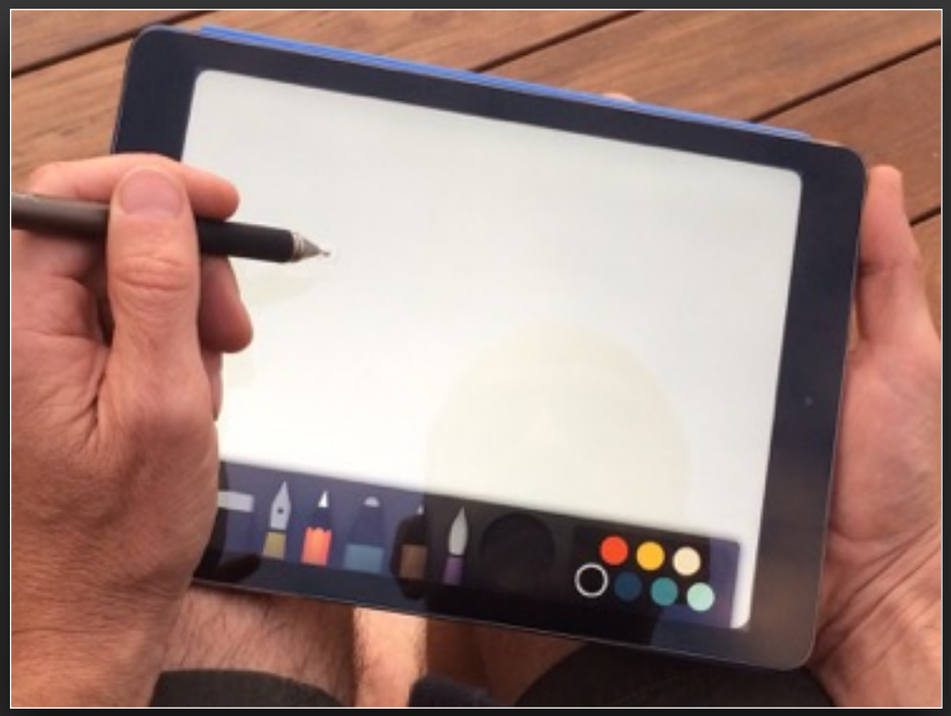


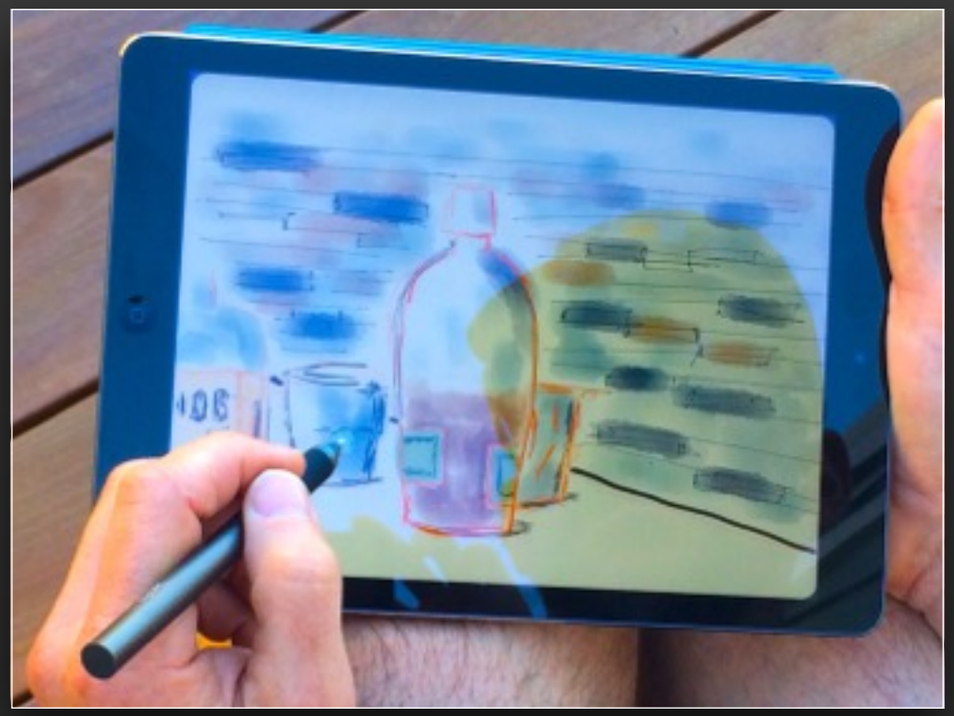
[1]Checkland, Learning for Action

Describe it as a Transformation

$$\frac{d}{dt}(\text{system})$$

Convert the Transform to Algorithms





$$\frac{d}{dt}(\text{system})$$

User Process

Observational Model

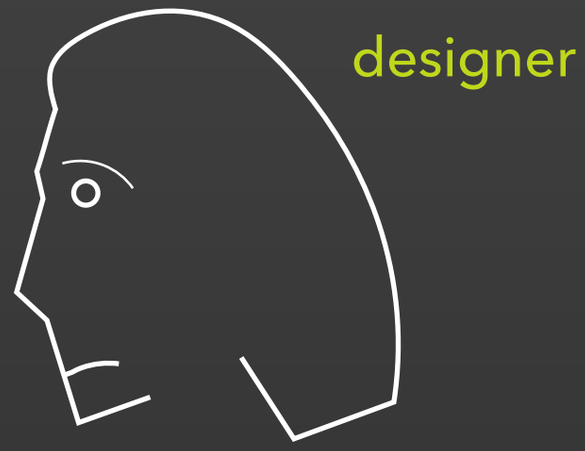
designer

developer

Algorithmic Model

Software

User Process



Observational Model

Software

User Process

designer

Observational Model

developer

Software

Algorithmic Model

Observational Model

Digital Machine

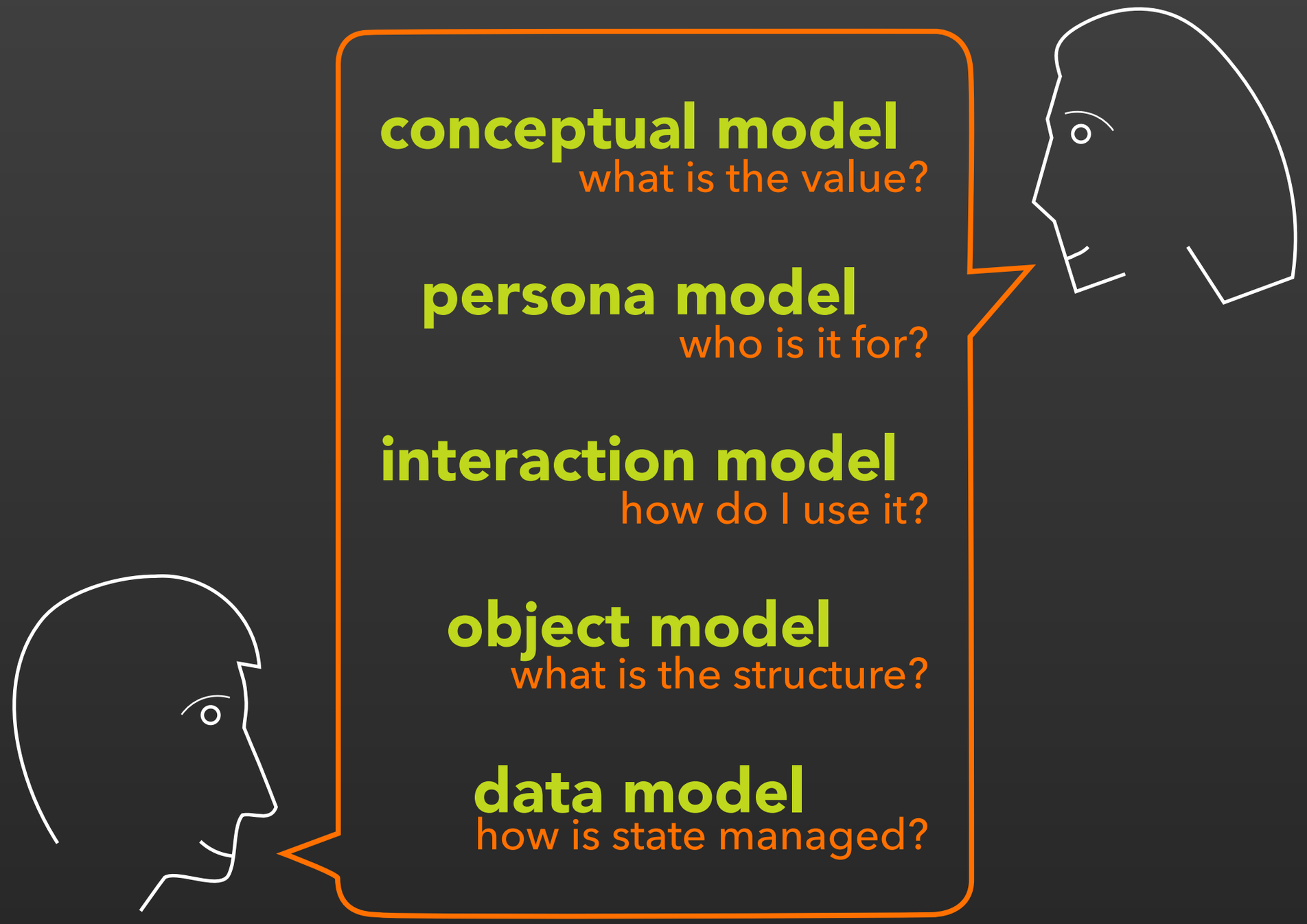
conceptual model

persona model

interaction model

object model

data model



the digital machine

conceptual model

object model

output

logic

input

storage

persona model

interaction model

data model

name:value
name:value
name:value

THE MODELS

conceptual model

object model

output

logic

input

storage

persona model

interaction model

data model

name:value
name:value
name:value

Conceptual Model

is a positioning statement

For target customers

Who have a particular need

This product is a category of solution

That provides a key benefit

Unlike the competition

conceptual model

object model

output

logic

input

storage

persona model

interaction model

data model

name:value
name:value
name:value

Persona Model

Design Personas

THE TECHNICIAN

Kyle Ratcliffe

Born and educated in Britain, Kyle trained in computer science before taking his first job as an operations technician with Barclays bank. Following an impetuous move across the Atlantic, Kyle has landed at DropBox as a member of the ops team.

Challenges

- Lots of context switching between monitoring tools
- Moving from a perspective of worrying about machines to worrying about the delivered experience
- Understanding if a problem is in the application or the infrastructure
- Being overwhelmed by continually increasing data sources and volume

Needs

Kyle's primary concern is system stability so he wants a system that tells him about change: has it happened, and if so, where and how much?

Kyle needs to trust that he will only hear from his system when there is a problem or, preferably, when a problem might be starting.

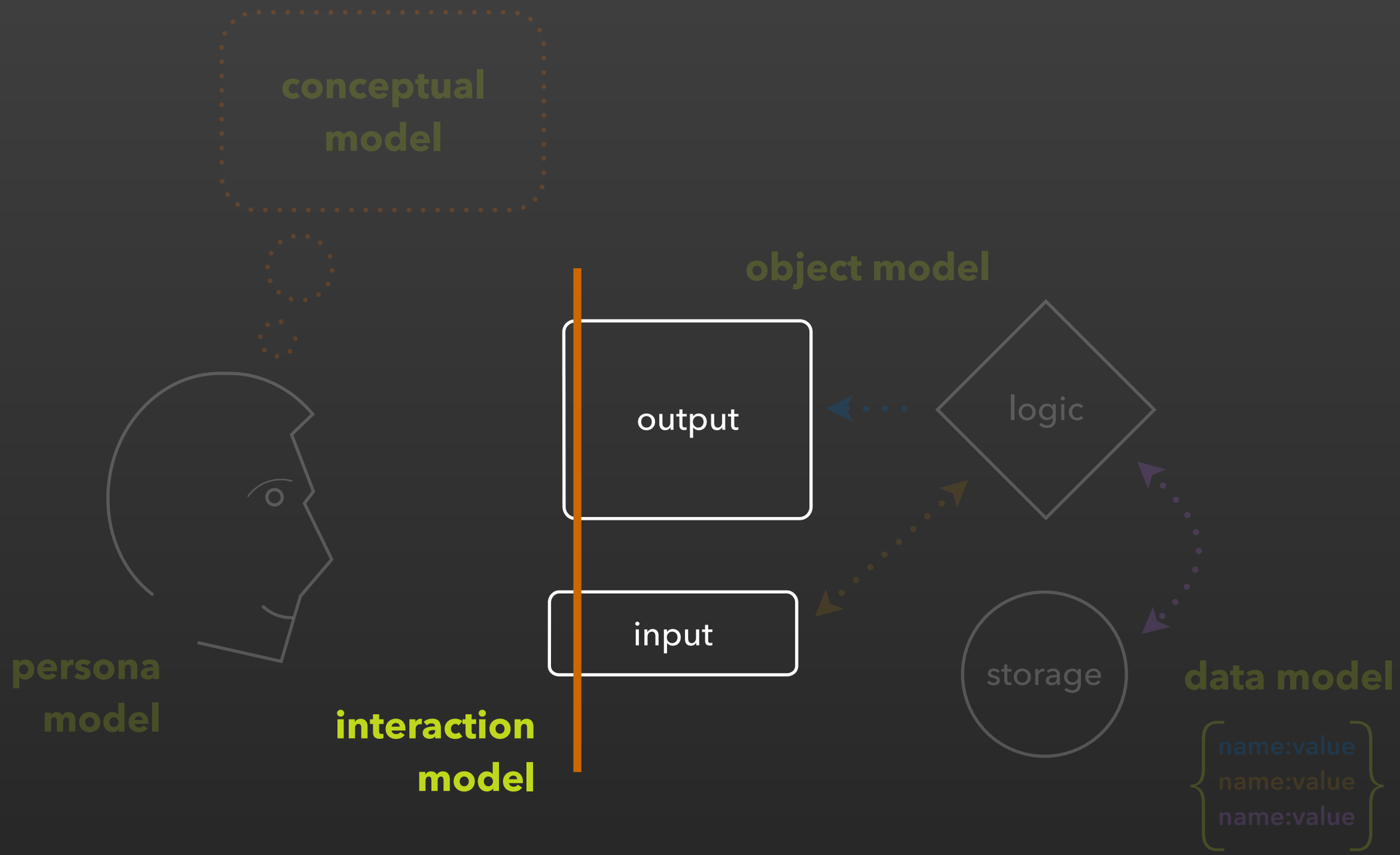
When there is trouble Kyle's particular challenges are

- visualizing trends across his system
- seeing log events as measurable metrics

Kyle's Favorite Tool: Chef

Likes

- he can create a logical recipe that describes



Scenario 1:

First thing in the morning, Kyle checks out hacker news and reads an article about a new data analytics tool called Jut.

Being [illegible] clicks a link to Jut's homepage and ends up on Jut.io

Upon arriving at Jut, Kyle immediately sees a visualization of real time and historical data of [traffic for hacker news]. Also within view is a code editor window pre - loaded with code in a language that he is told is Juttle.

Kyle also sees a short description of what exactly the code is doing along with challenge for changes he can try to make to the code and see how it effects the visualization.

Kyle sees a series of other scenarios that he can explore. As he clicks through each scenario, he can see how short segments of Juttle code allow him to analyze data in a way that is more effective and more efficient than h can with other applications.

Kyle begins to make edits in the editor window and is provided with code hints to help him better understand what methods and functions are available to him in Juttle.

Kyle follows the directions given in this first scenario and makes the suggested changes. Afterward, he sees the visualization immediately react to the changes he made.

After completing the recommended changes, Kyle clicks the 'submit' code button and is congratulated for taking his first step for correctly completing the challenge.

Kyle is then given an explanation of how juttle is designed with dev-ops engineers in mind along some suggestions for other ways he can make more advanced changes to his code in this scenario. He decides to pass on these suggestions and move on to the next scenario.

Kyle finishes a couple more scenarios and, on the last one, is given the option to import his own live <twitter feed>. After loading his data in, he makes some changes to the Juttle code that allow him to visualize it in a unique way. He decides he wants to share this visualization in a blog post about the advent of dev-ops and its effects on traditional operations.

Kyle copies a link to the visualization he's just made along with the data that he's uploaded and embeds the link in his blog.

Kyle sees how Jut could be useful for a couple future projects at work and decides to sign up for the beta while copying a few co-workers on the sign-up to loop them into the conversation and make sure they all know when Jut is ready for use.

Interaction Model

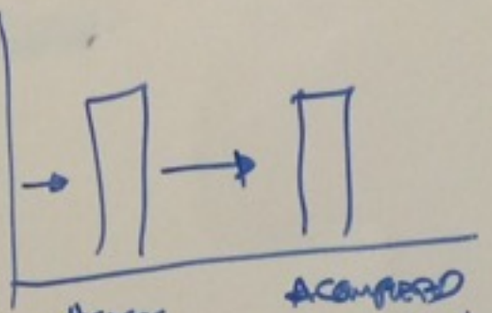
is a story about change

(EXPLORE) → ANOMALY DETECTED → VISUALIZE METRICS → COMPARISON
- VS STD INTERVAL
- VS NAMED EVENTS
→ (NO PRECEDENT)

- PROBLEM
- CONFIRM TRUE PROBLEM
- LOOK FOR ERRORS
- FIND BAD MACHINE
- REPLACE PROBLEM IS CHECK IN

- STOPPING FILES
- CAN'T CHECKOUT

BROKEN STEP

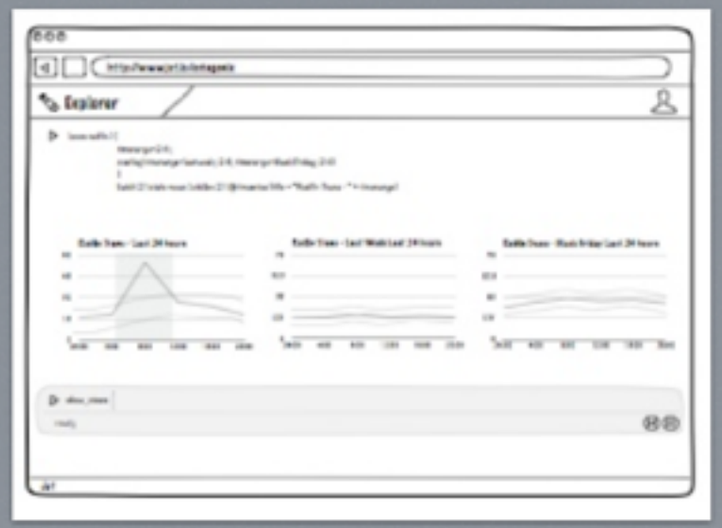


SEARCH FOR ERRORS IN LOGS

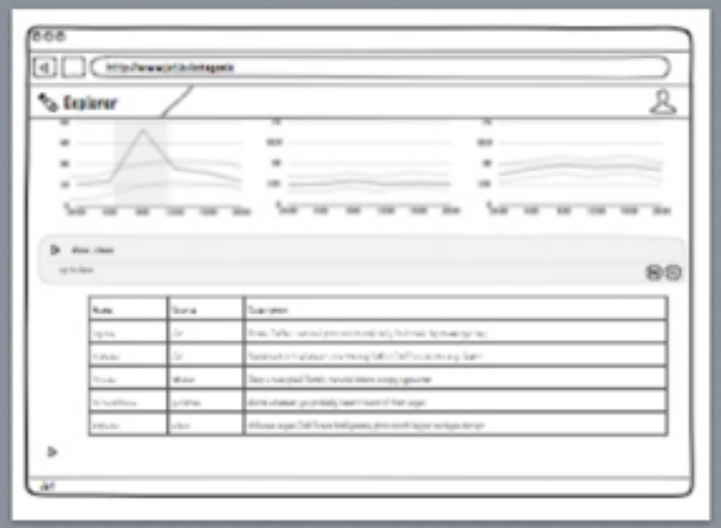
OVERLAY EVENTS

OVERLAY GRAPHS

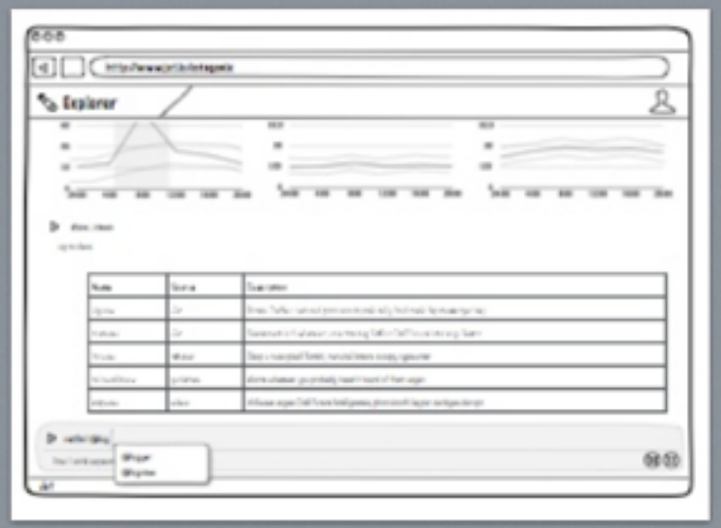
NARROW PROBLEM TO A HOST → SEE METRICS OF HOST PERFORMANCE → IDENTIFY TIME PROBLEM BEGAN → IDENTIFY CAUSE



13



14

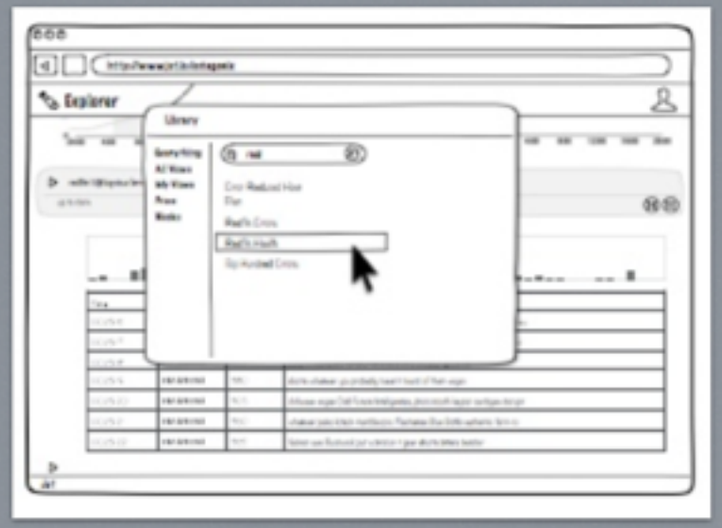


15

16

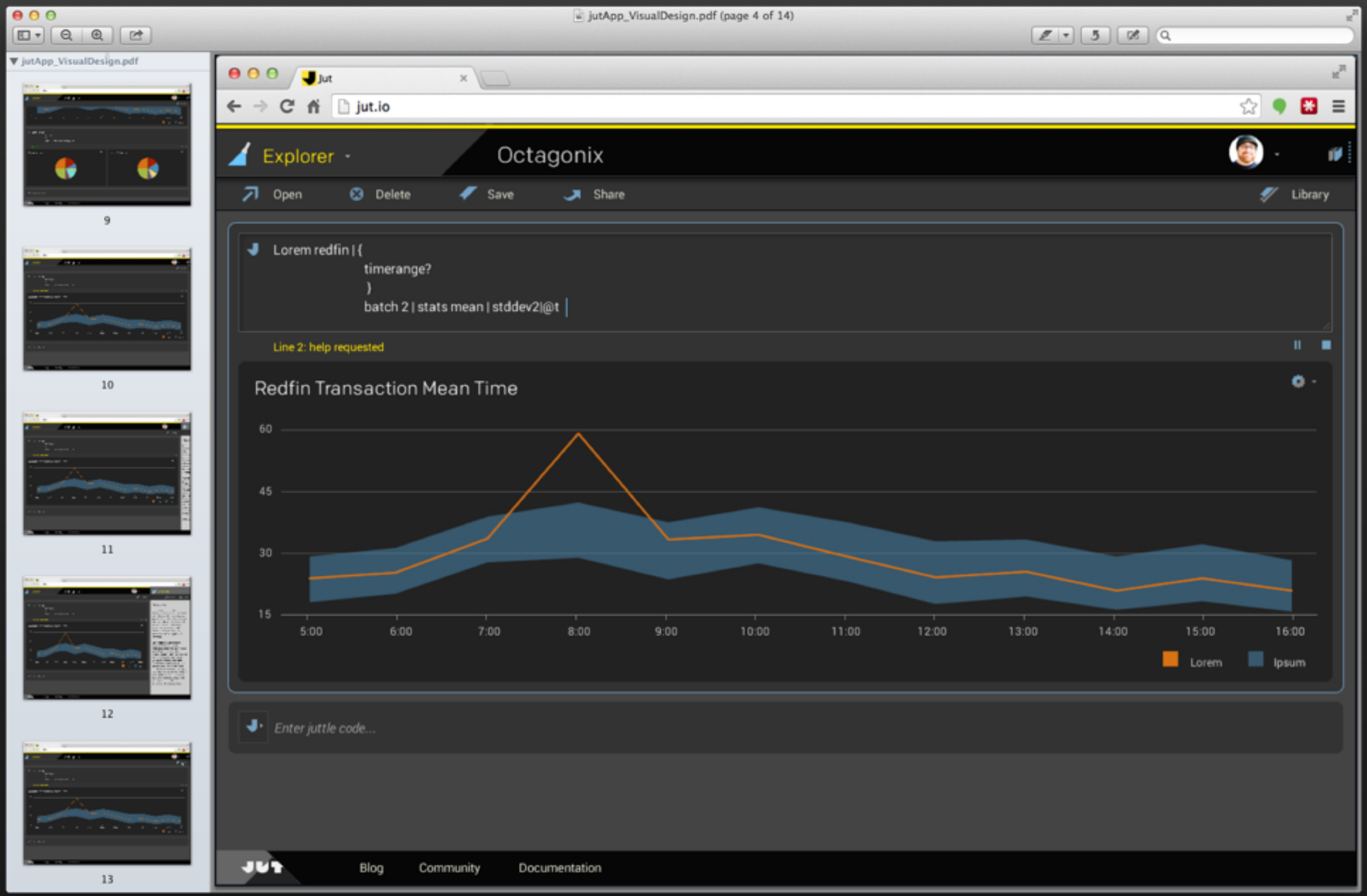
17

18



19

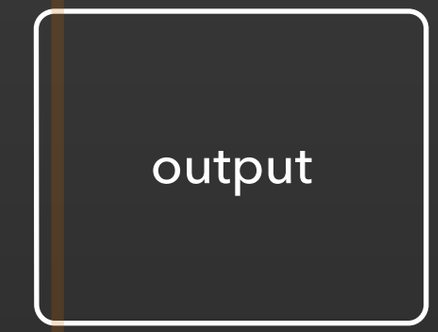
20

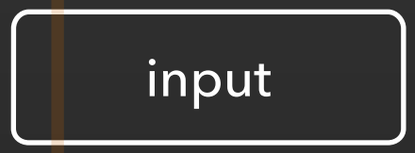

jutApp_VisualDesign.pdf (page 4 of 14)
jutApp_VisualDesign.pdf
9
10
11
12
13
Jut
jut.io
Explorer
Octagonix
Open
Delete
Save
Share
Library
Lorem redfin | {
timerange?
}
batch 2 | stats mean | stddev2|@t
Line 2: help requested
Redfin Transaction Mean Time
60
45
30
15
5:00
6:00
7:00
8:00
9:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
Lorem
Ipsum
Enter juttle code...
Blog
Community
Documentation

conceptual model

object model

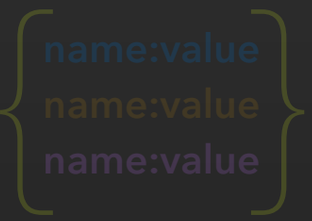
persona model





interaction model

data model



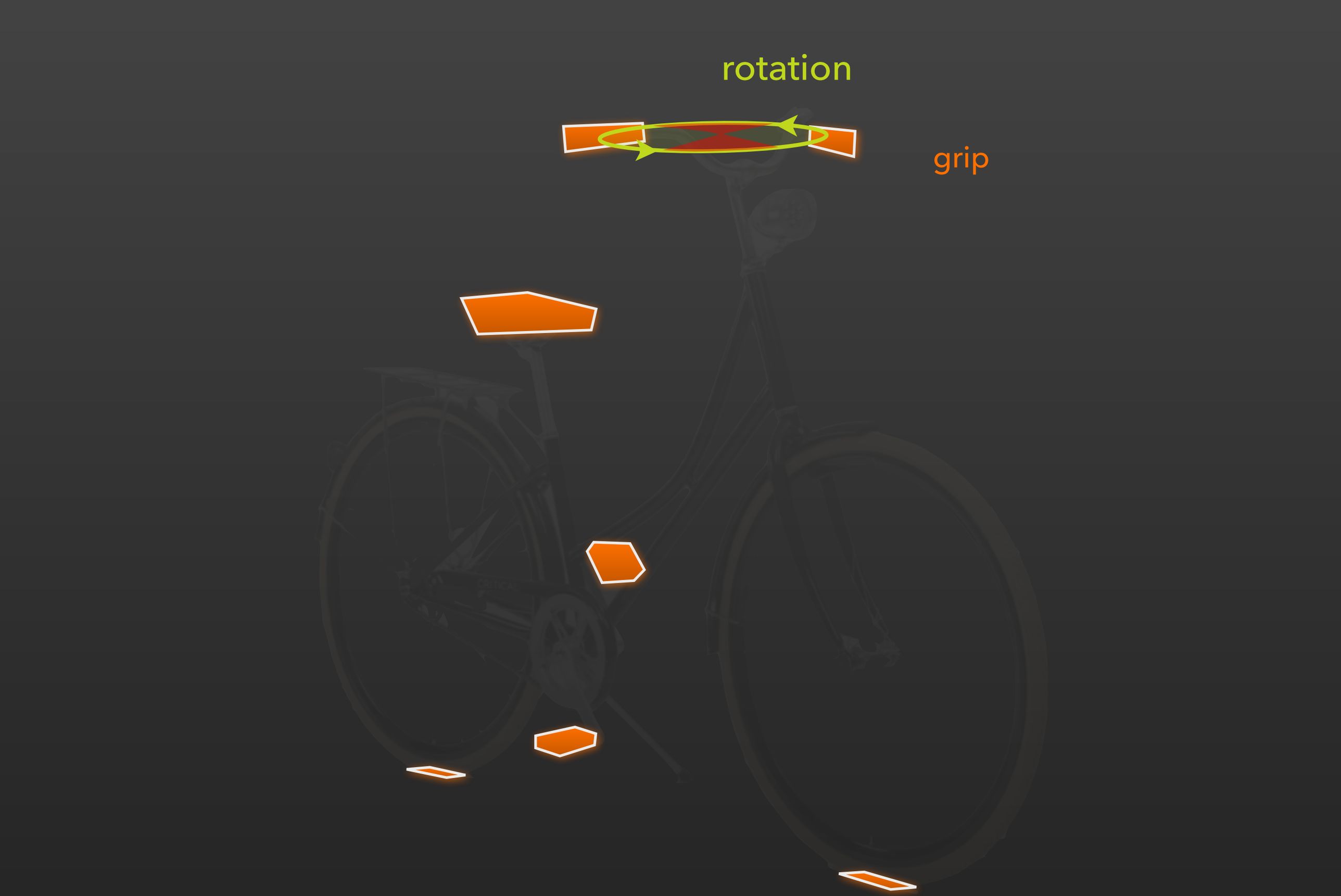

CRITICAL


CRITICAL

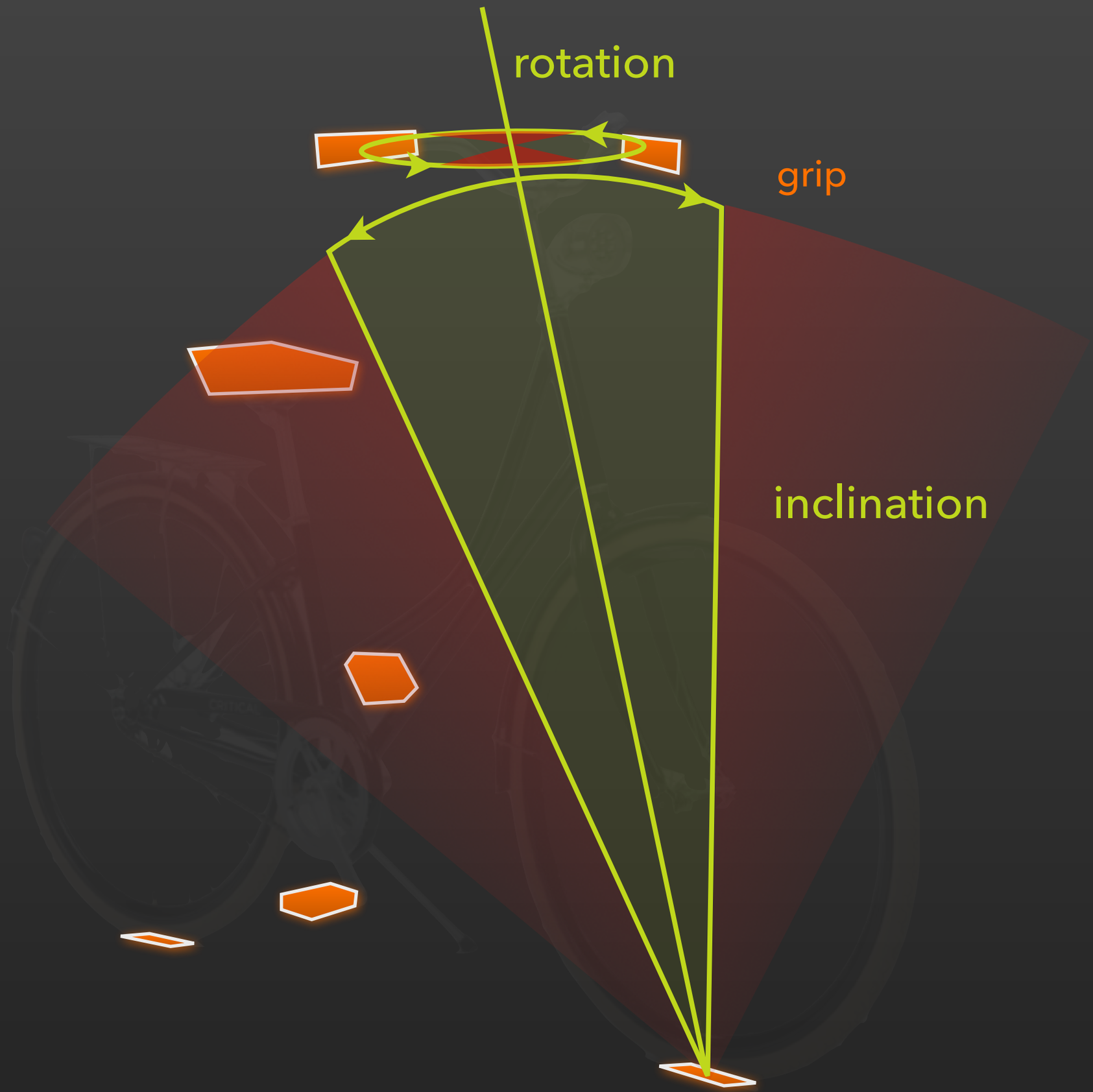
handle bar
handle
grip
handgrip
control

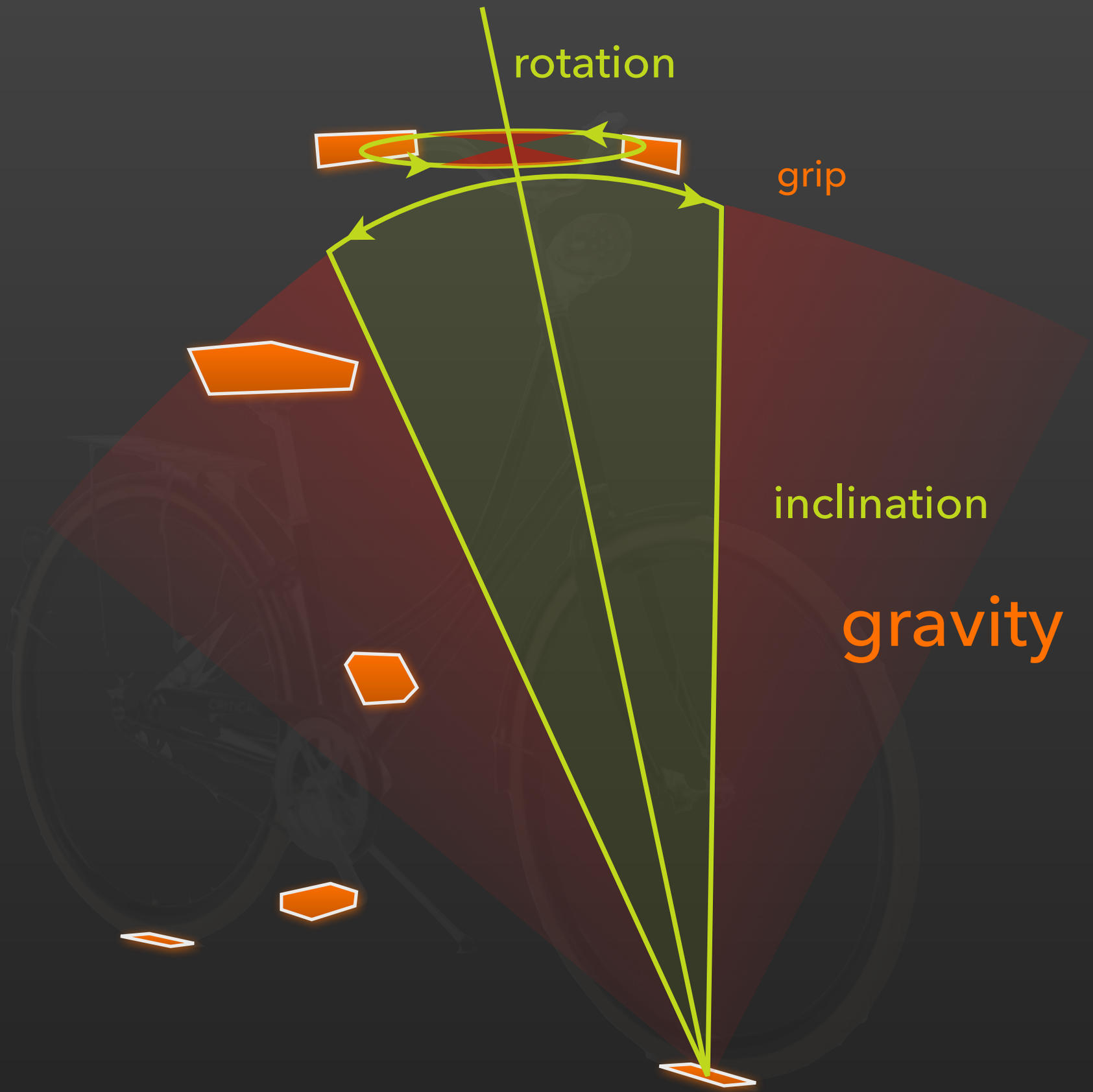
?

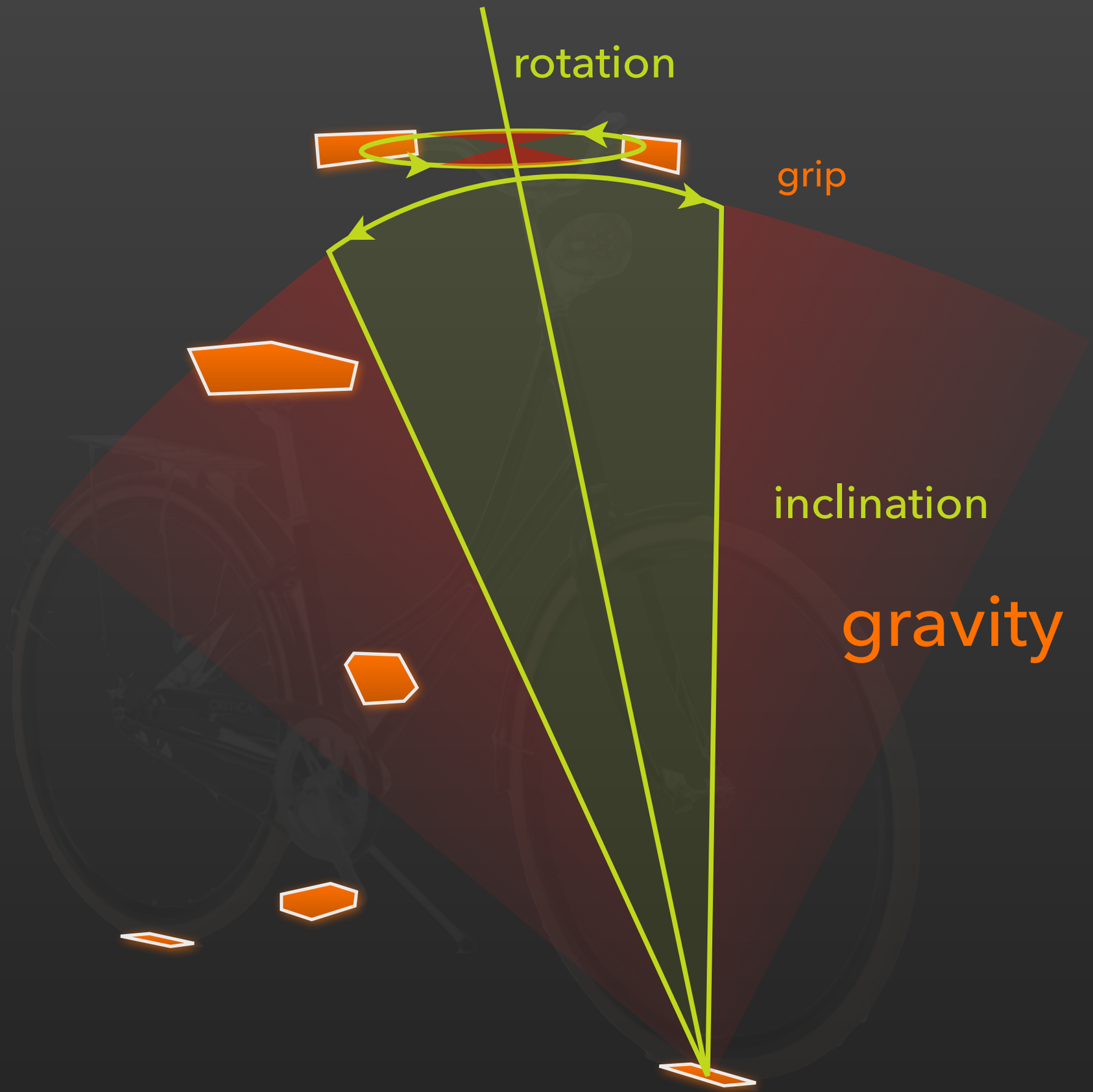

grip


rotation
grip

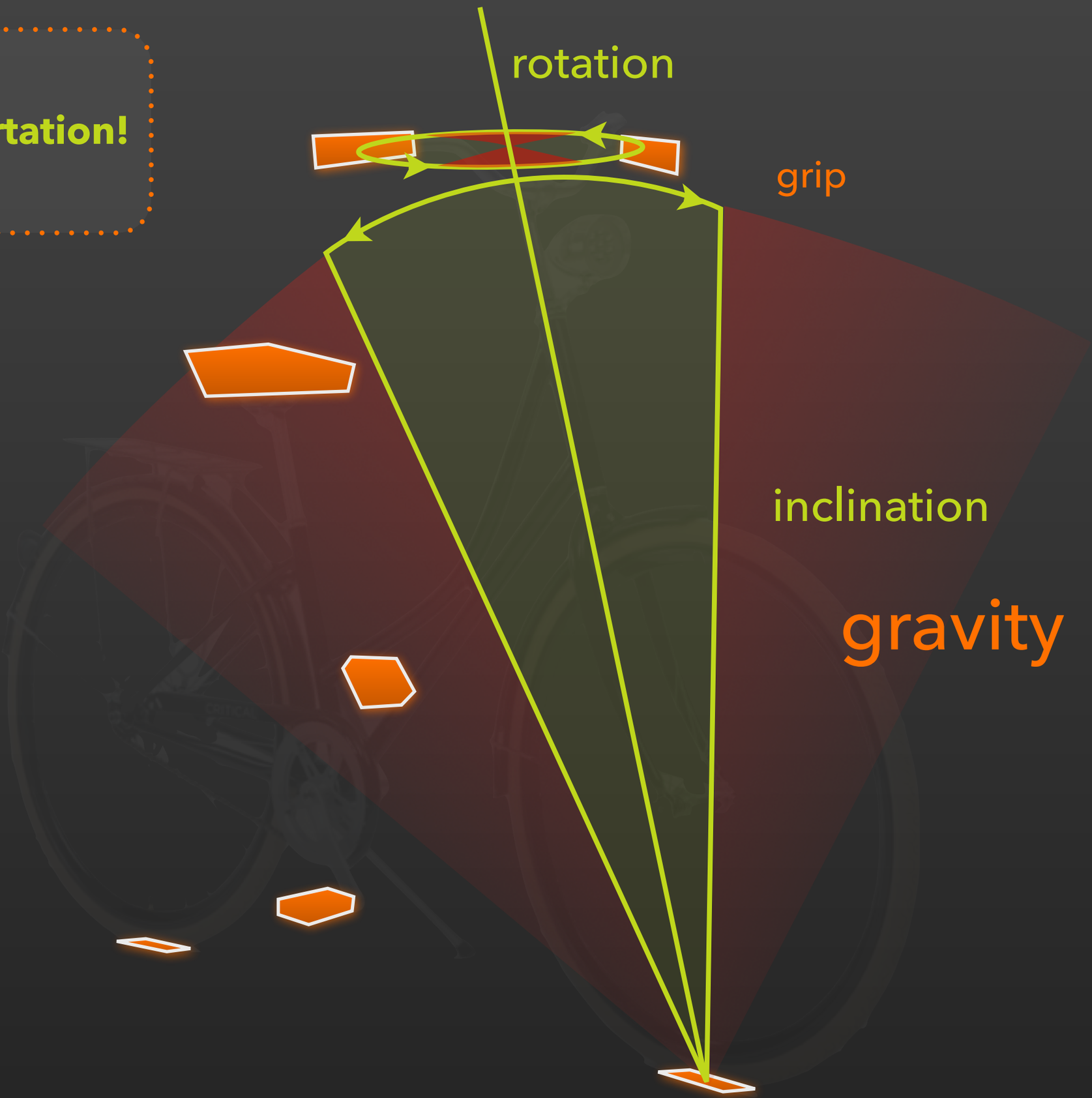

rotation
grip


rotation
grip
inclination

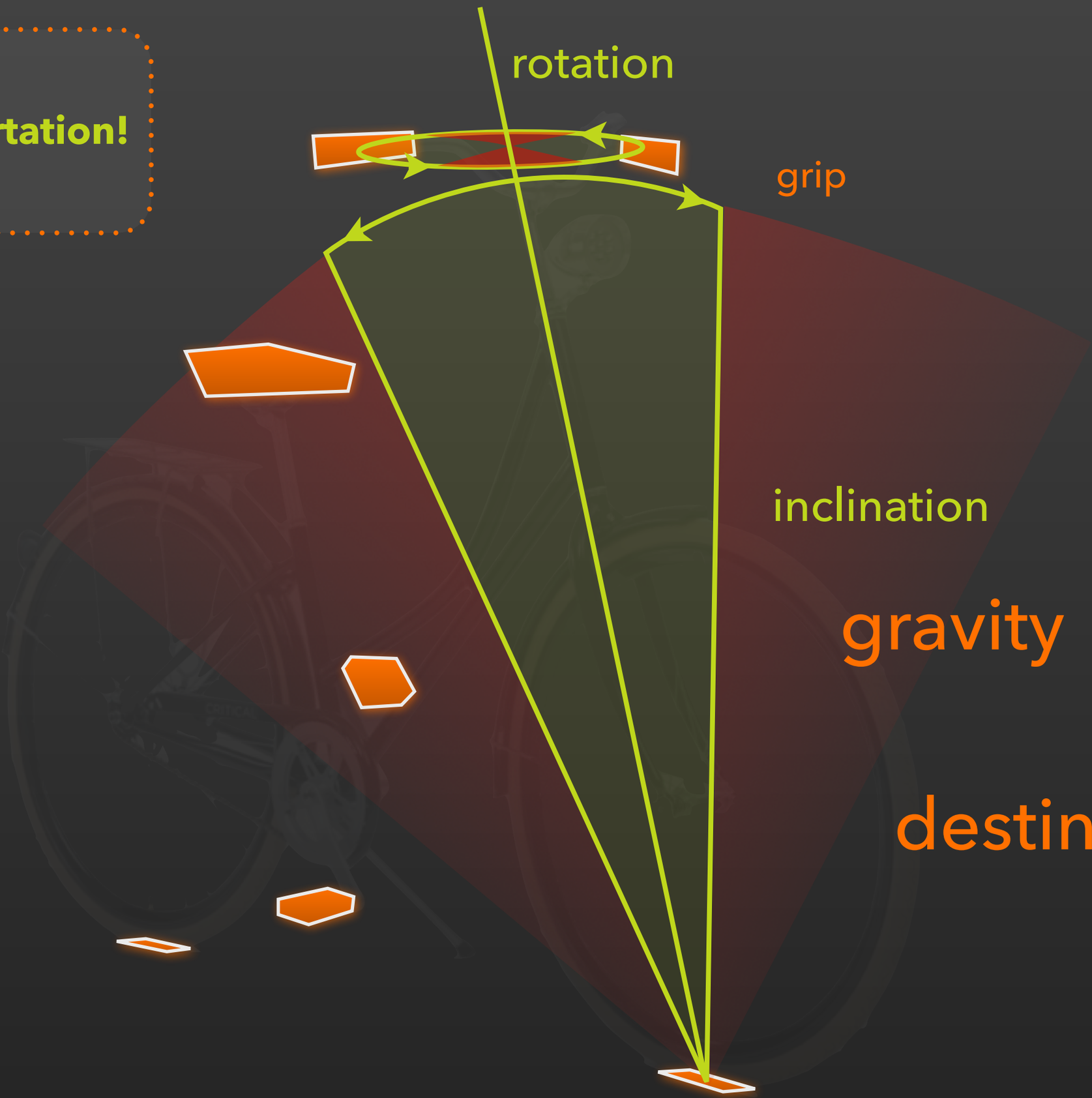

rotation
grip
inclination
gravity


rotation
grip
inclination
gravity

transportation!



transportation!



transportation!

rotation

grip

inclination

gravity

origin

destination

transportation!

rotation

grip

inclination

gravity

origin

position

destination

transportation!

rotation

grip

inclination

gravity

position

origin

destination

trip

Object Model

Object Model

details

rotation

inclination

position

$\frac{d}{dt}$ (system)

?

conceptual model

object model

output

logic

input

storage

persona model

interaction model

data model

name:value
name:value
name:value

WHAT IS DATA?

(in the software context)

name : value

lat : '59.916'

```
position : {
    lat : '59.916', long : '10.738'
}
```

```
position : {
    lat : '59.916', long : '10.738'
}
```

a set of name : value pairs defines object state

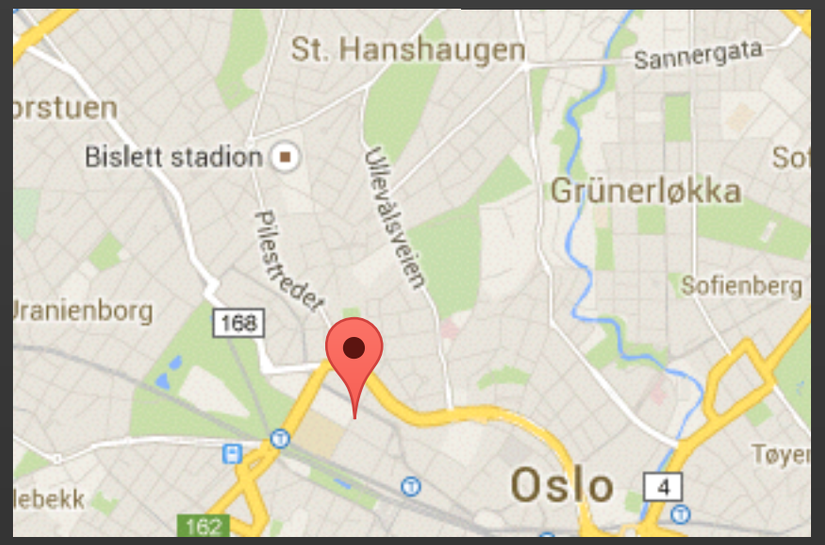
```
position : {
    lat : '59.916', long : '10.738'
}
```

```
bikeTrip = {
    vehicle : {
        rider : {
            name : 'Kyle'
        },
        bicycle : {
            grip : {
                rotation : '12', inclination : '3'
            }
        },
        position : {
            lat : '59.916', long : '10.738'
        },
    },
    origin : {
        name : 'Hotel Savoy',
        position : {
            lat : '59.916', long : '10.738'
        },
    destination : {
        name : 'AHO',
        position : {
            lat : '59.925', long : '10.751'
        }
}
```

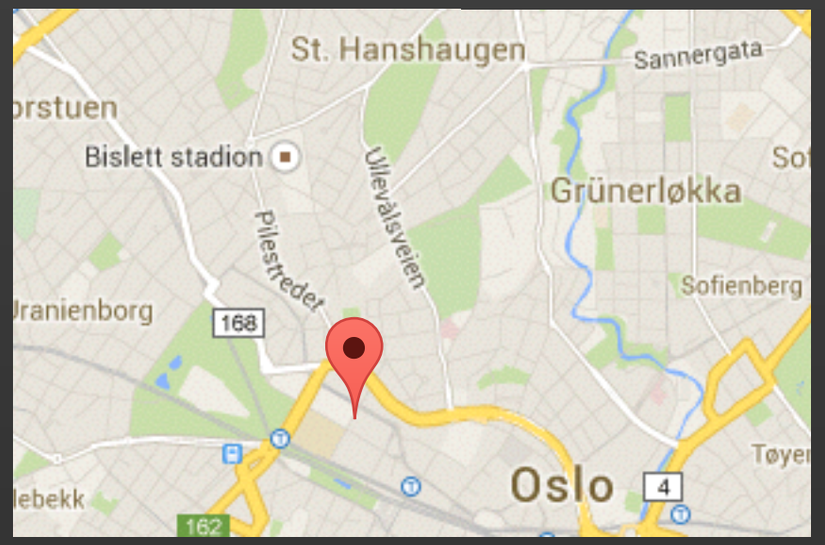
a set of object states defines system state

```
bikeTrip = {
       position : {
          lat : '59.916', long : '10.738'
       }
}
```

```
bikeTrip = {
        position : {
            lat : '59.916', long : '10.738'
        }
}
```



```
bikeTrip = {
        position : {
            lat : '59.916', long : '10.738'
        }
}
```



interaction model

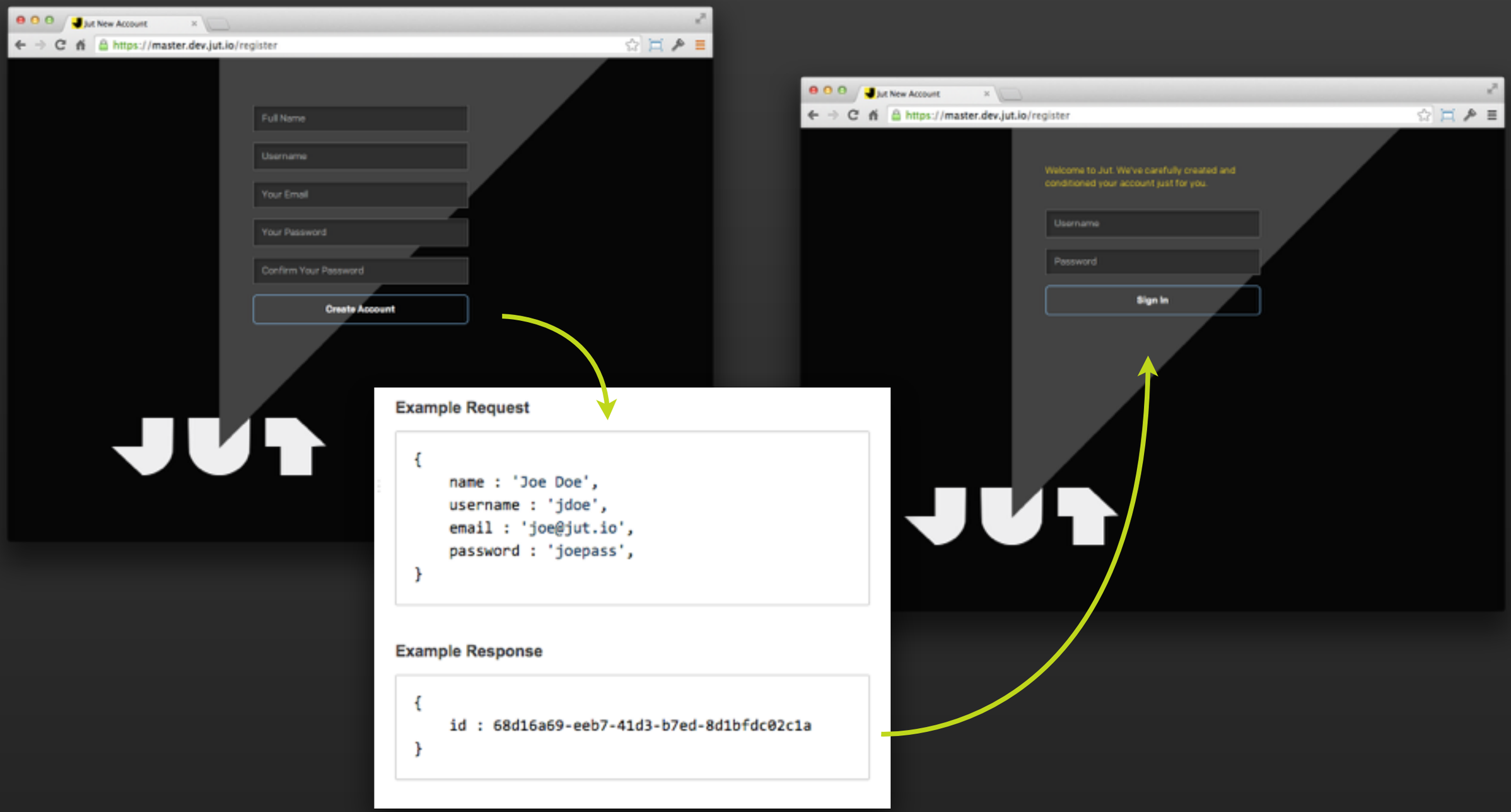
```
bikeTrip = {
        position : {
            lat : '59.925', long : '10.751'
        }
}
```

interaction model

$$\frac{d}{dt}(\text{system})$$

Data Model

underlies every interaction



User Process

Observational Model

the digital machine

conceptual model

object model

output

logic

input

storage

persona model

interaction model

data model

name:value
name:value
name:value

Algorithmic Model

Software

FIN