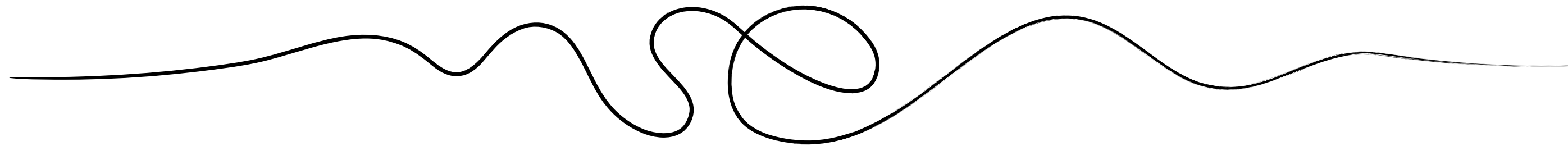


**SIMPLE vs EASY**



**WARNING!**

**BASED SOLELY ON**

**PERSONAL EXPERIENCE**



**Home**



- 1. FULL CONTROL**
- 2. LINEAR EFFORT**



Home

1.8 km, 22 mins walk



**[e]** office



- 1. NO CONTROL**
- 2. EFFORTLESS**

Home



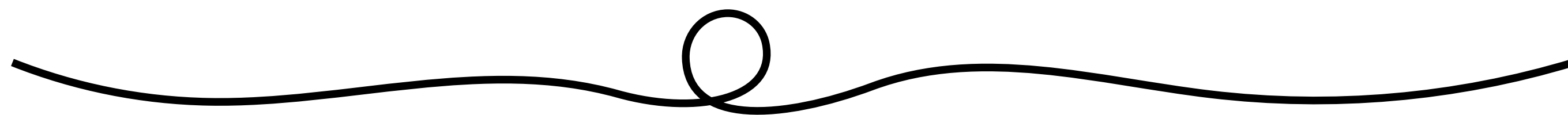
**[e] office**

776 km, 158 hrs walk

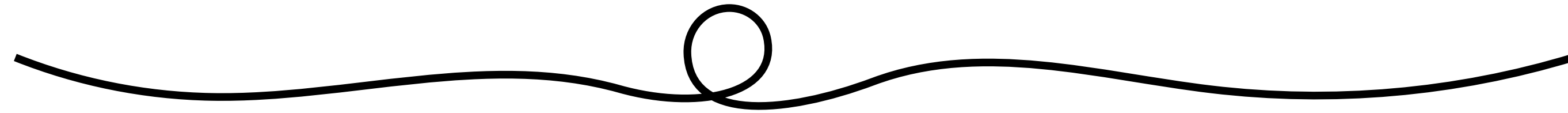


ngBigParty V (Prague)

# SIMPLE



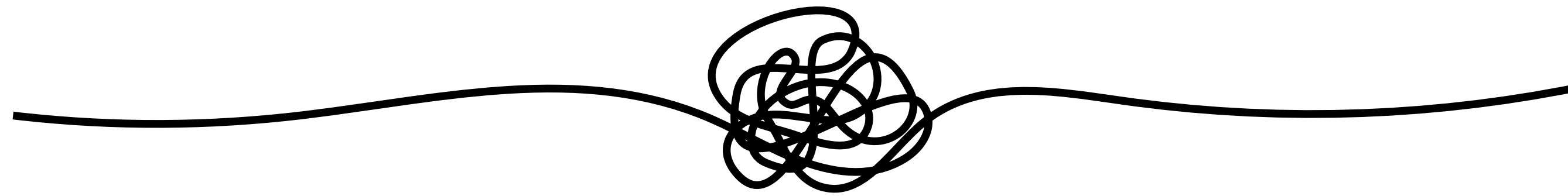
# SIMPLE



*"one fold"*

straightforward, easily understood  
uneducated, unsophisticated, stupid

# COMPLEX

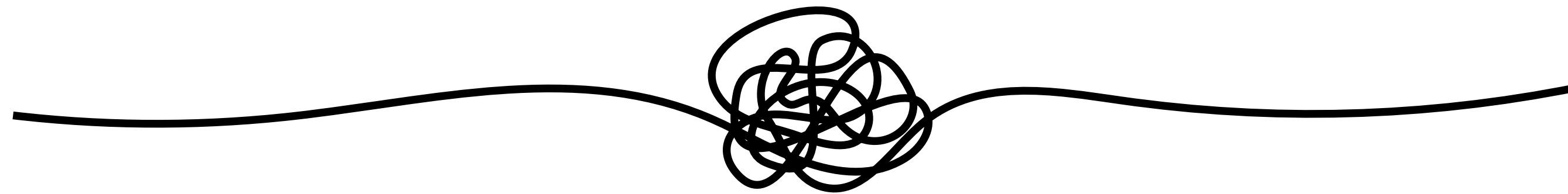


*"composed of interconnected parts"*

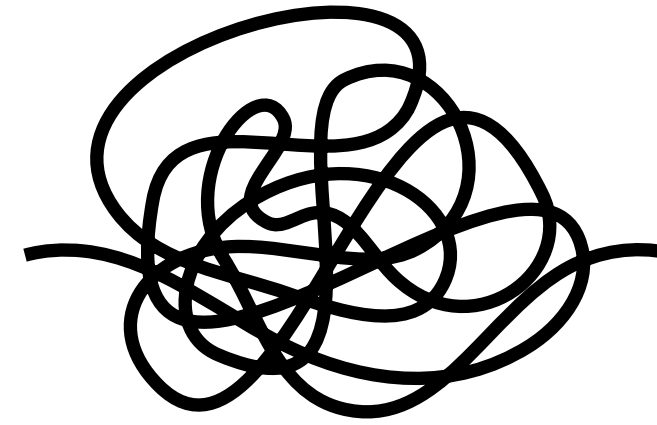
involved, intricate, complicated, not easily analyzed



# COMPLEX



**SIMPLE**



**COMPLEX**

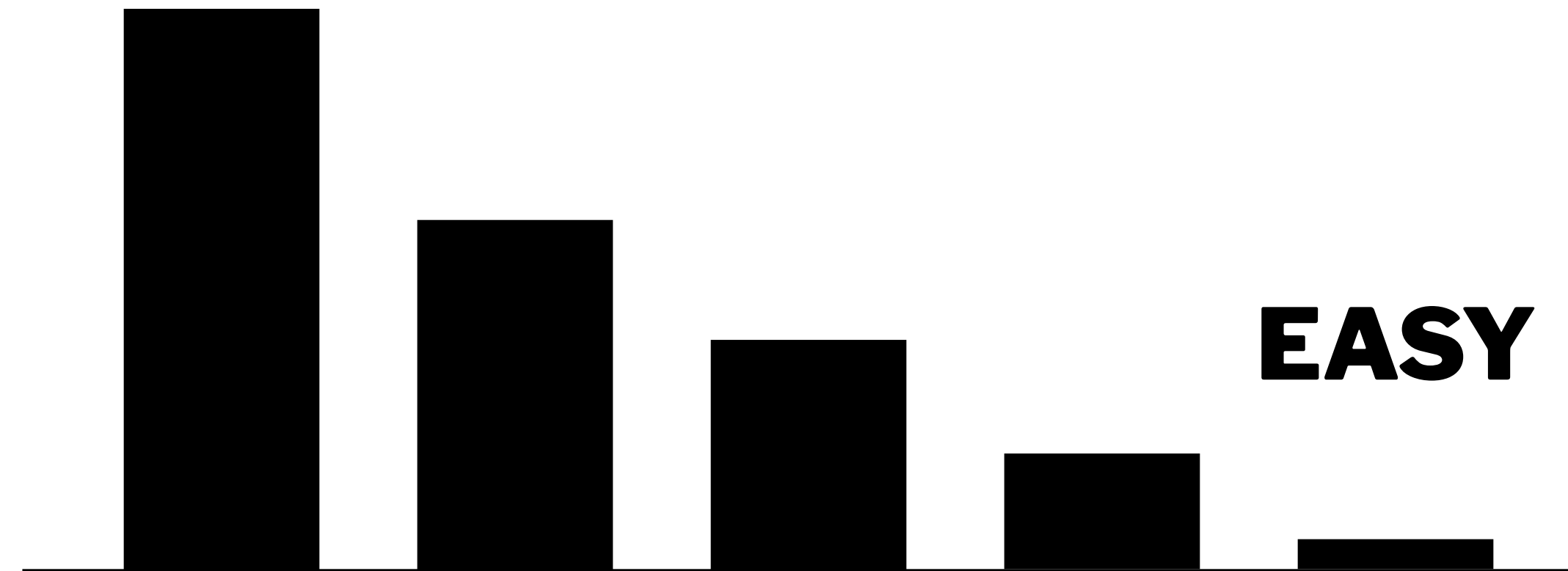
**HOW SOMETHING IS BUILT**

# EASY

*"requiring no great labor or effort"*

comfortable, pleasant

**DIFFICULT**



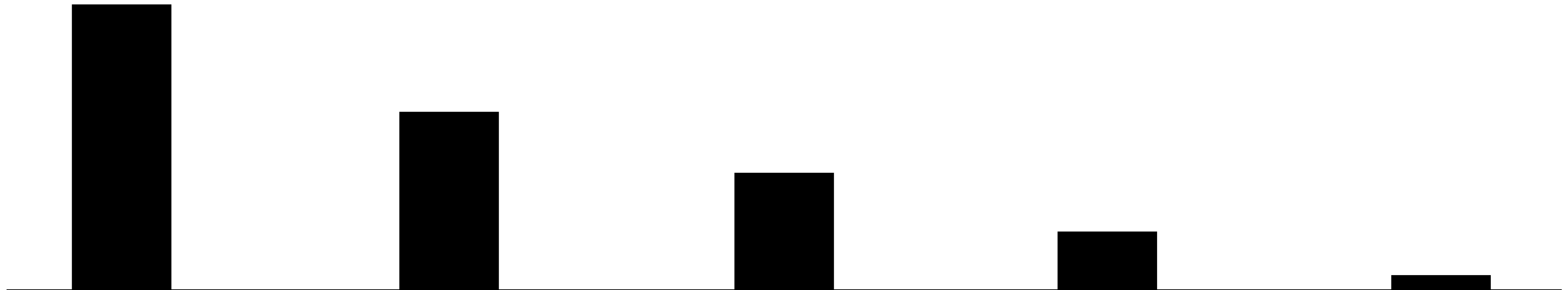
**EASY**

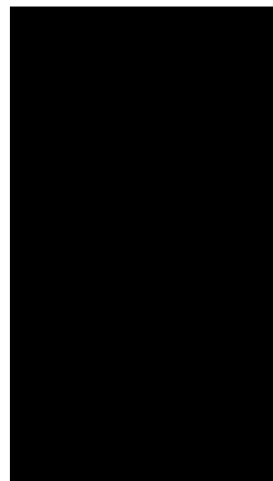
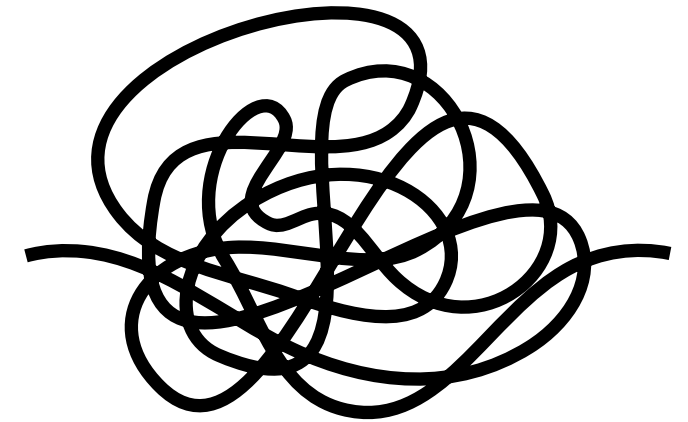
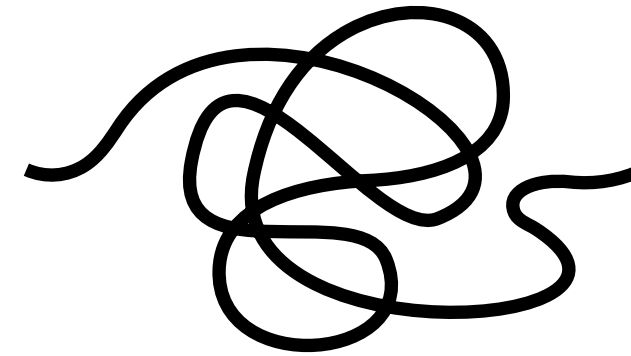
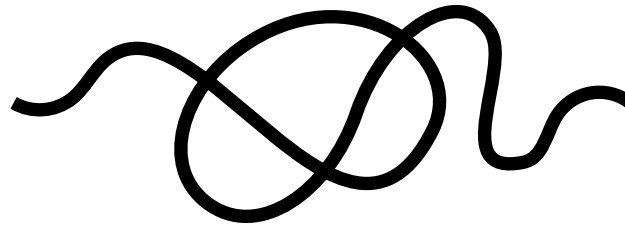
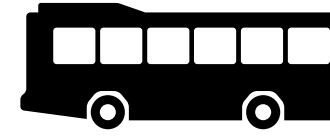
**HOW SOMETHING IS USED**

**DIFFICULT**

**EASY**

**EFFORT**





**EFFORT / CONTROL**

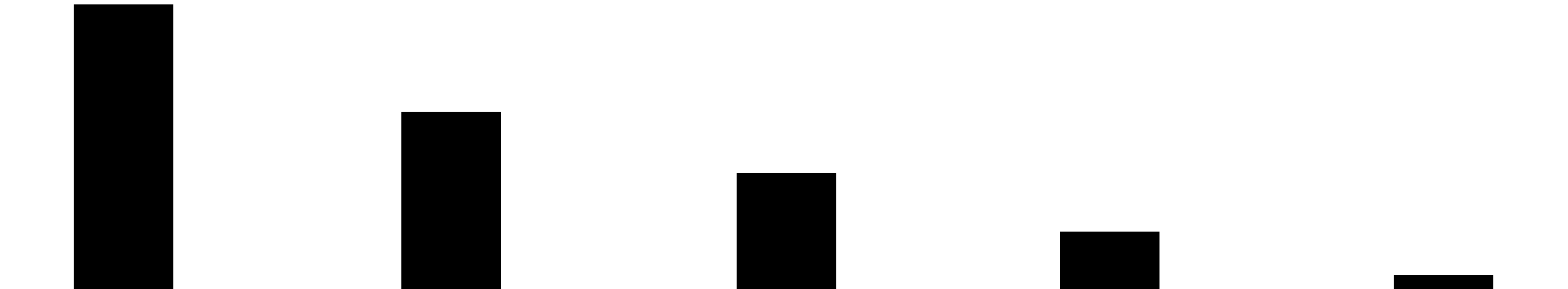
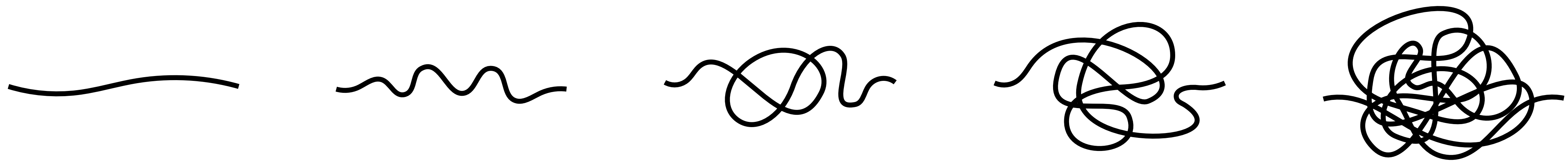
**VANILLA**

**LIBRARIES**

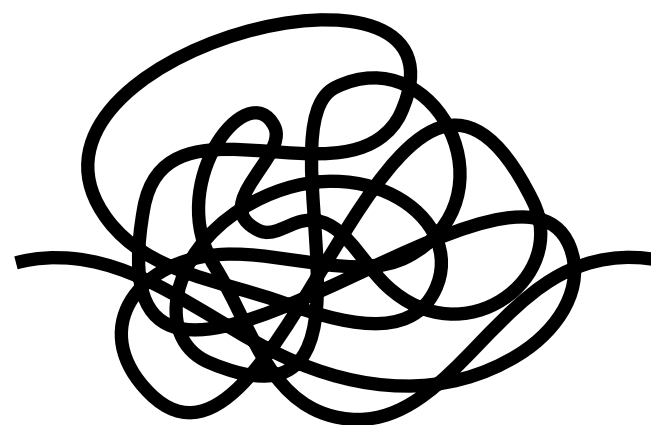
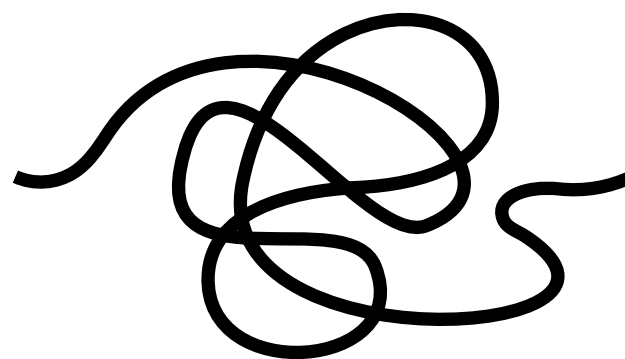
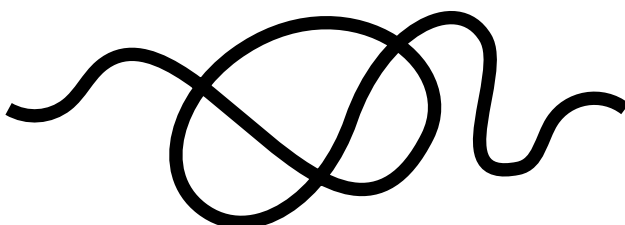
**FRAMEWORKS**

**PLATFORMS**

**LOW/NO CODE**



**EFFORT / CONTROL**





**KISS**

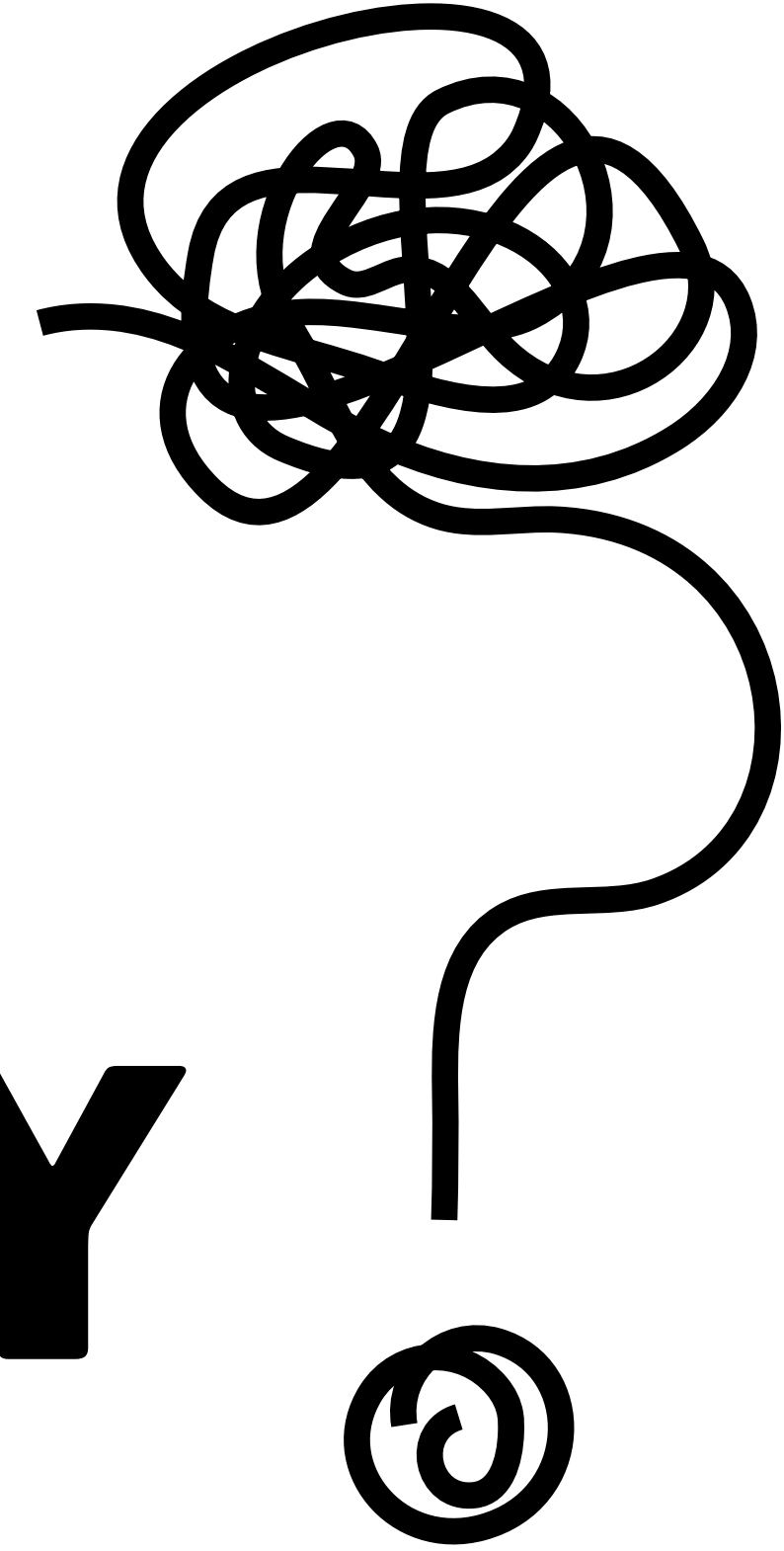
**KEEP**

**IT**

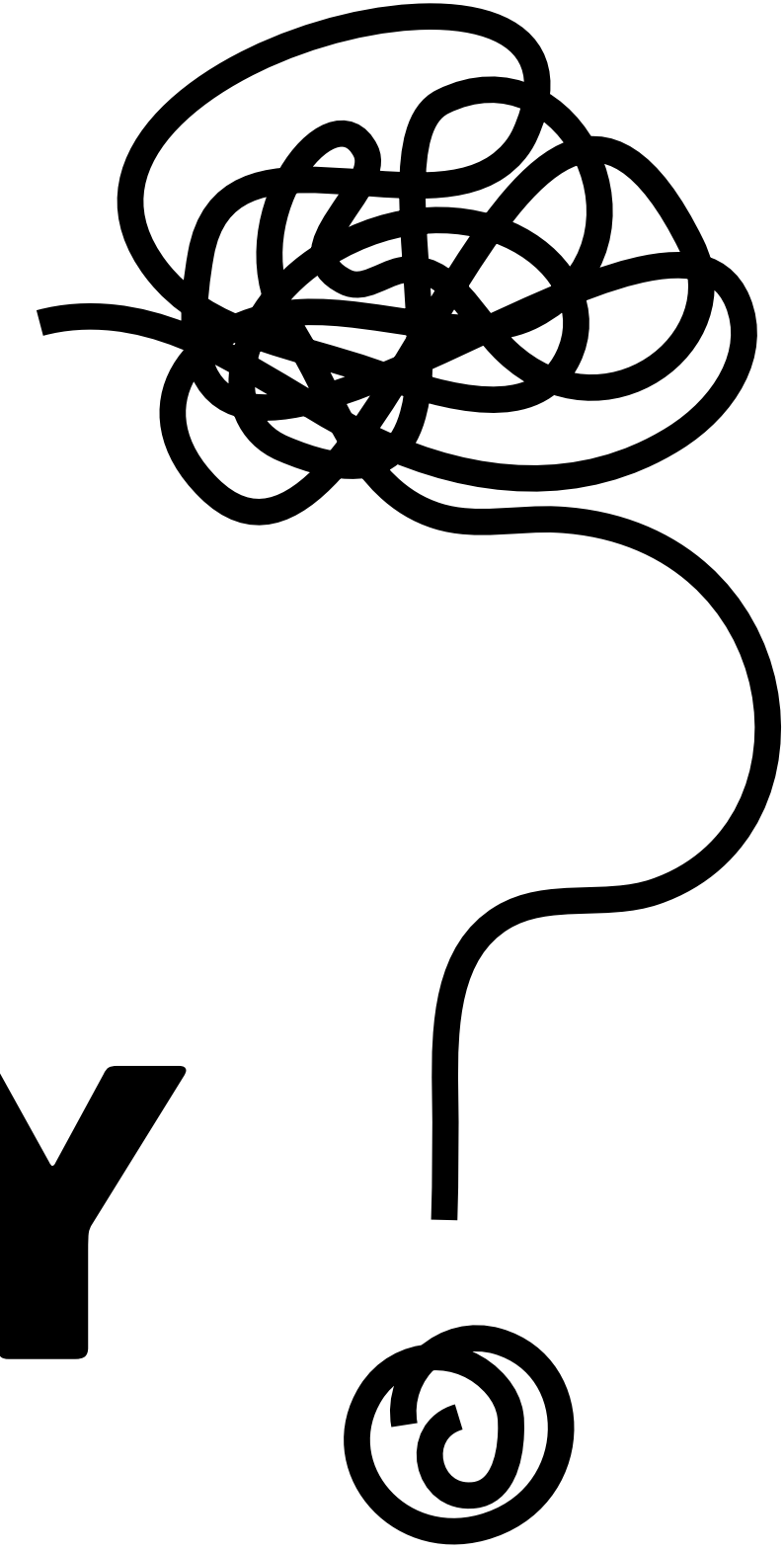
**SIMPLE,**

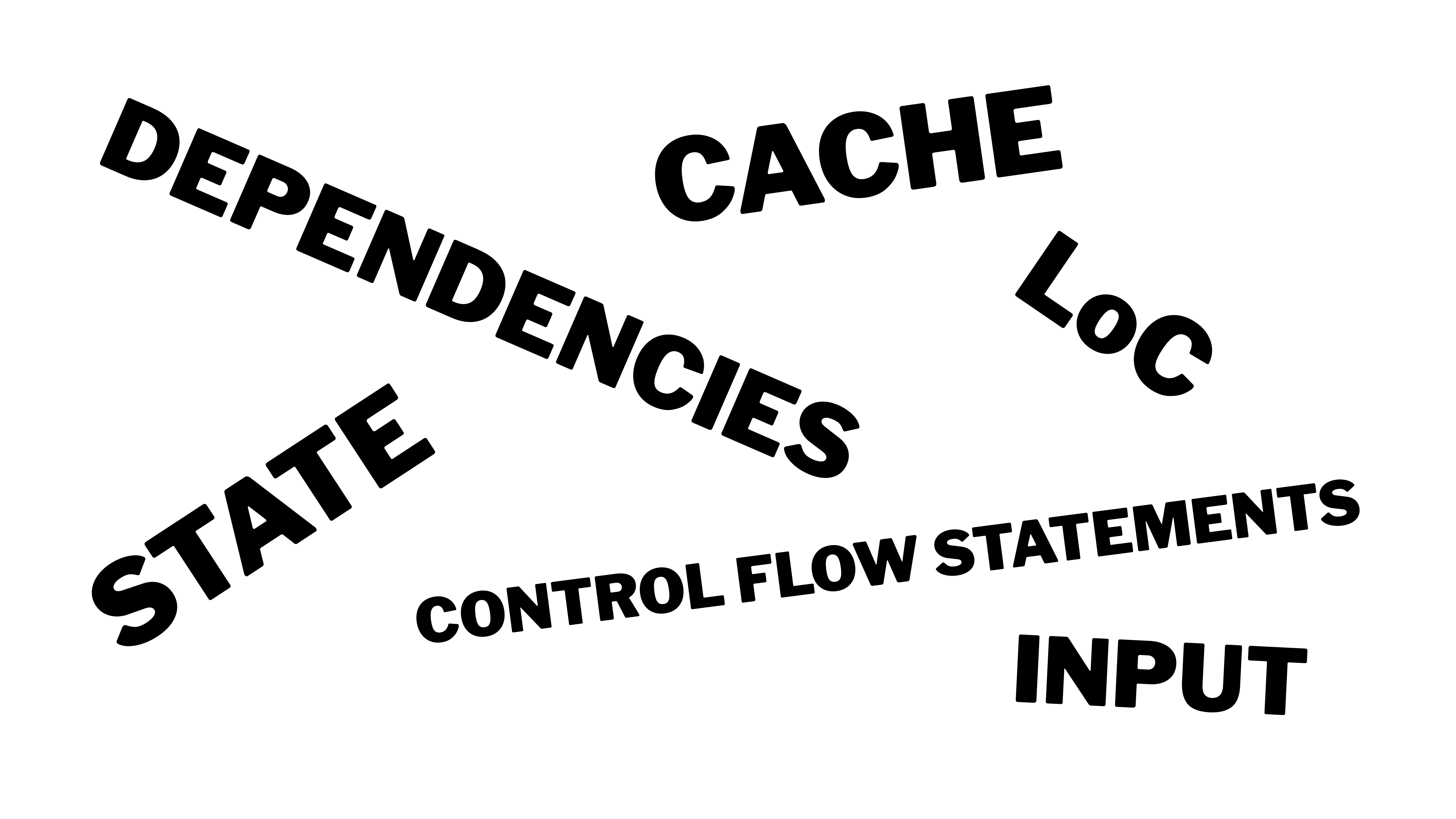
**STUPID**

**WHAT'S WRONG WITH  
COMPLEXITY**



**WHAT IS  
COMPLEXITY**





STATE  
DEPENDENCIES

CACHE

Loc

CONTROL FLOW STATEMENTS

INPUT

**COMPLEXITY**

*building*



**AUTHOR**

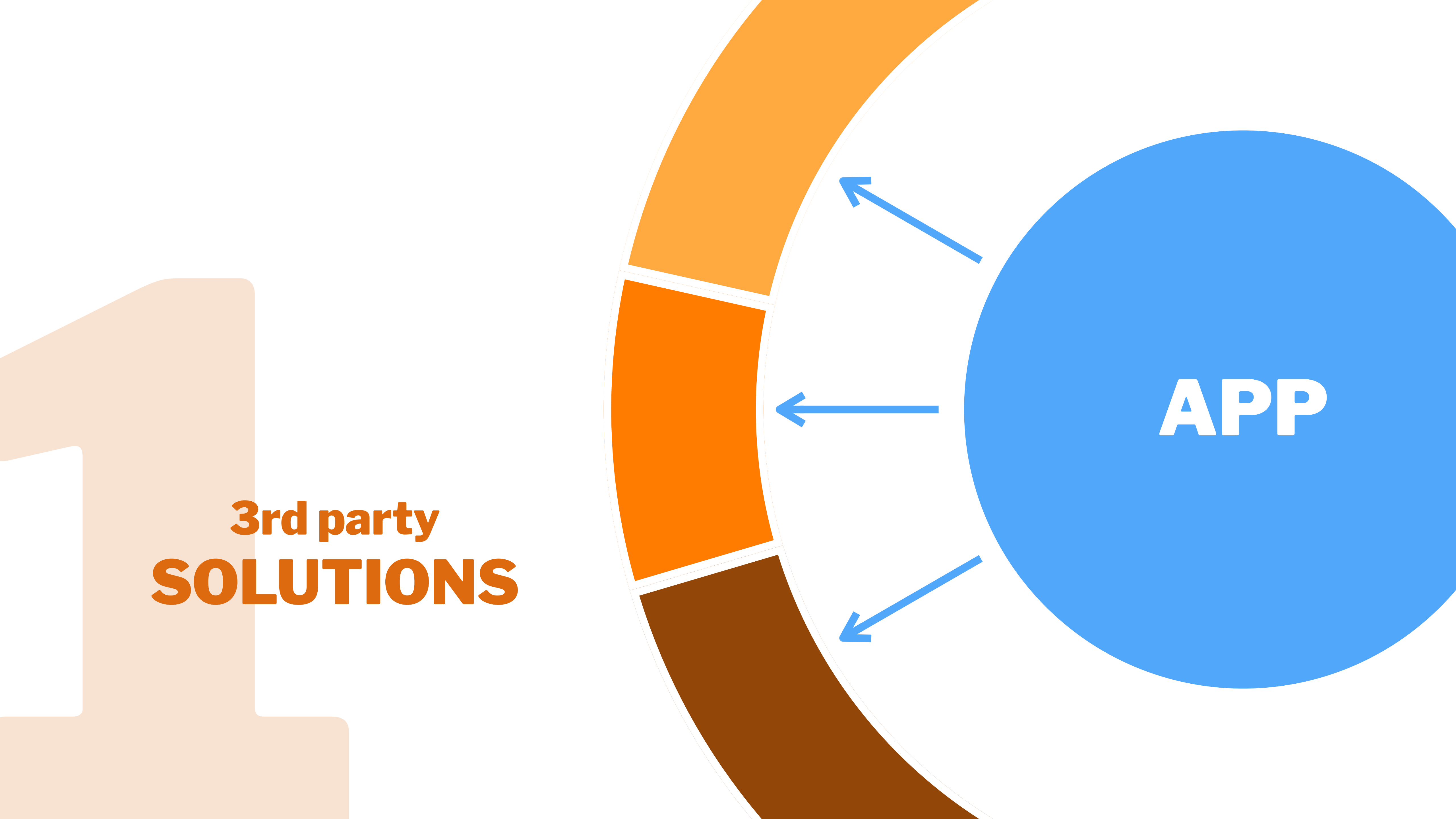
**DIFFICULTY**

*using*



**CONSUMER**





**APP**

**3rd party  
SOLUTIONS**

**#1 PROBLEM**

**WHAT TO CHOOSE ?**









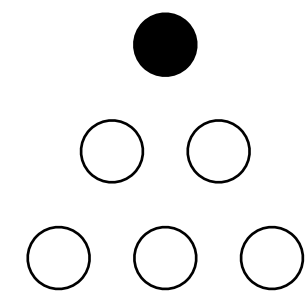
**OR**



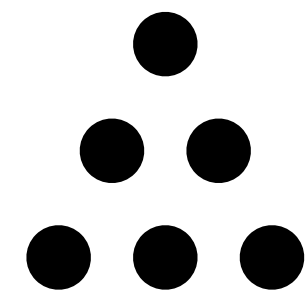
# React Application State Management

- **Redux** w/ many store connections
- **Redux** w/ few store connections
- **MobX**
- **MobX** with mobx-state-tree
- **Unstated**

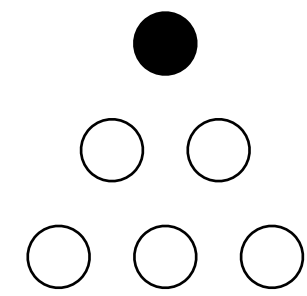
*[andreipfeiffer.github.io/react-state-management/](https://andreipfeiffer.github.io/react-state-management/)*



**SINGLE COMPONENT UPDATE**



**MANY COMPONENTS UPDATE**

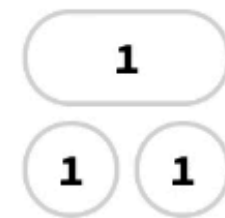


**CONTINUOUS UPDATES**

0

+1 all

☐ continuous update







**SINGLE COMPONENT  
UPDATE**





**ALL COMPONENTS  
UPDATE**

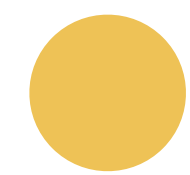
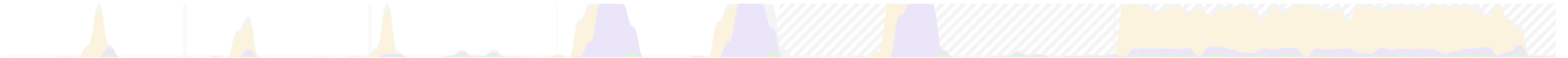




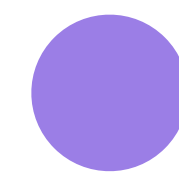


**CONTINUOUS  
UPDATE**



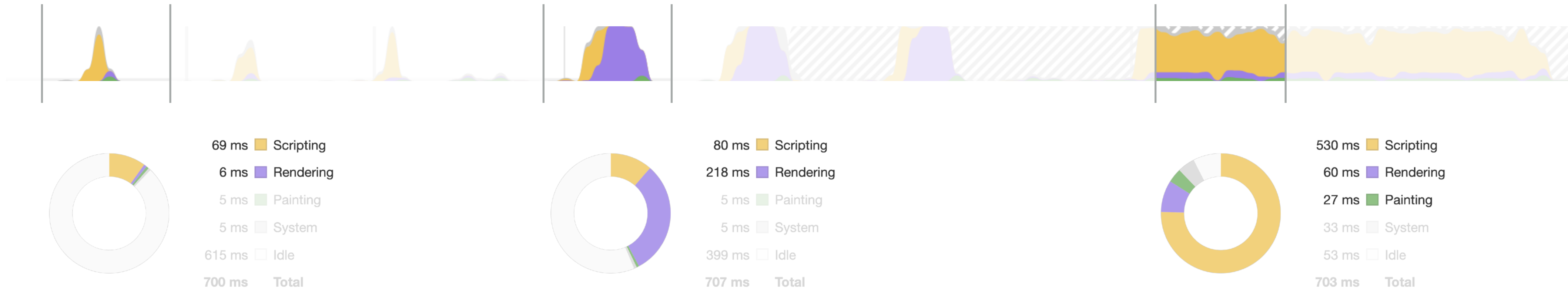


**Scripting**  
Framework effort, JS execution (React rendering)



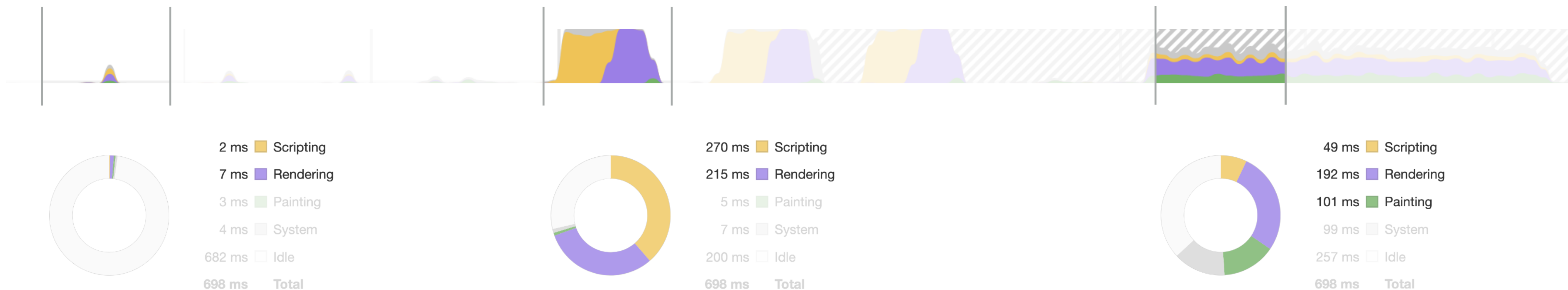
**Rendering**  
Browser effort (Layout / Reflow / Recalculate)





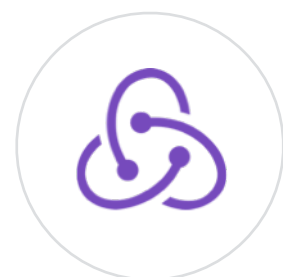
Requires manual optimizations

Same amount of JS execution



Optimized out-of-the-box

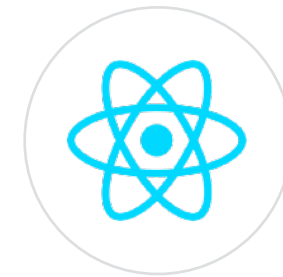
Diminishing returns, unoptimizable



Redux



MobX



React



Vue



Angular



Express



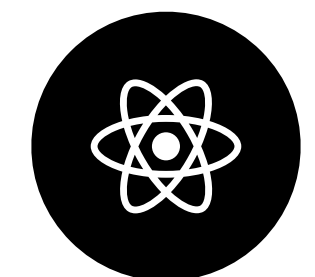
Nest



Next



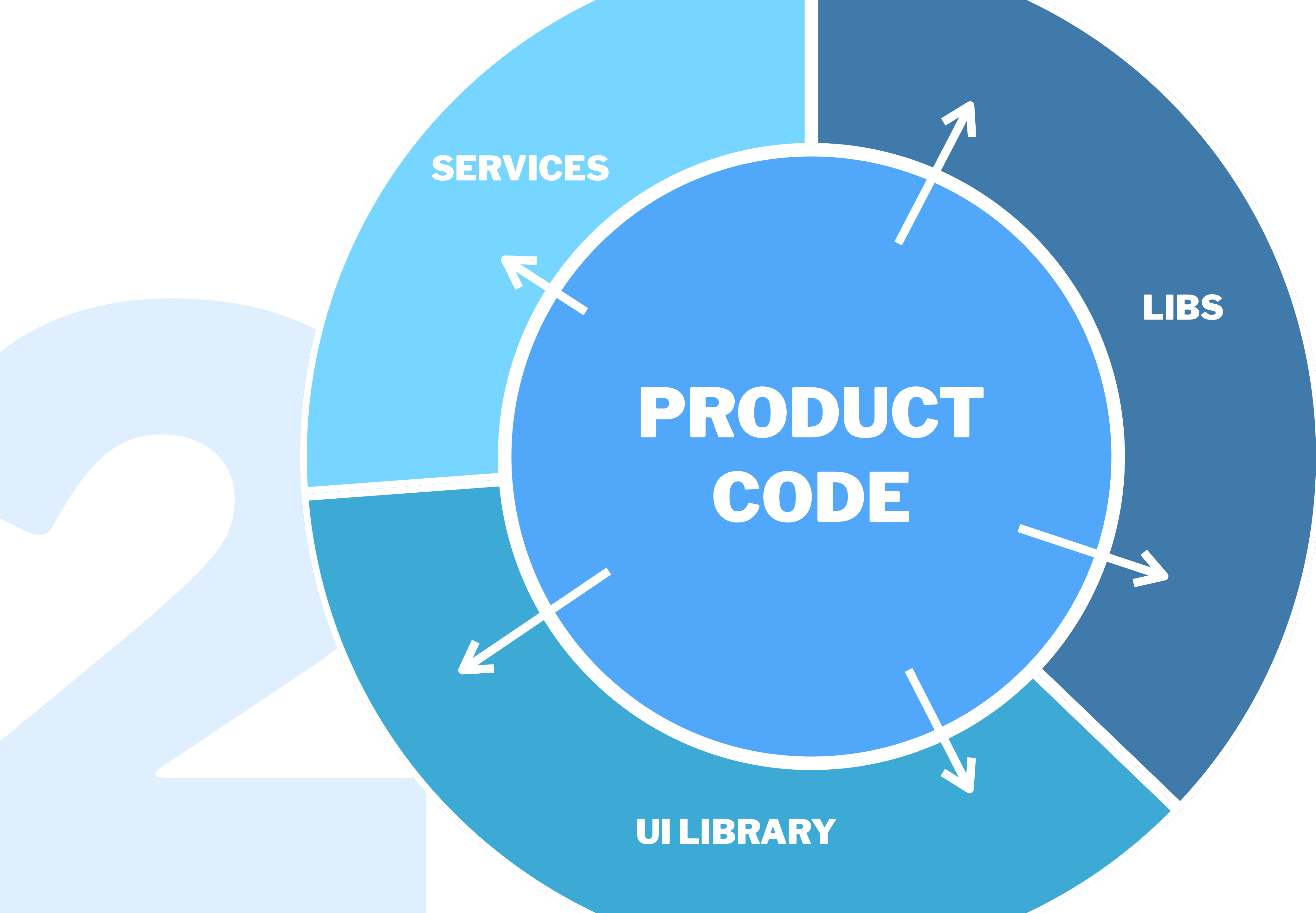
Gatsby

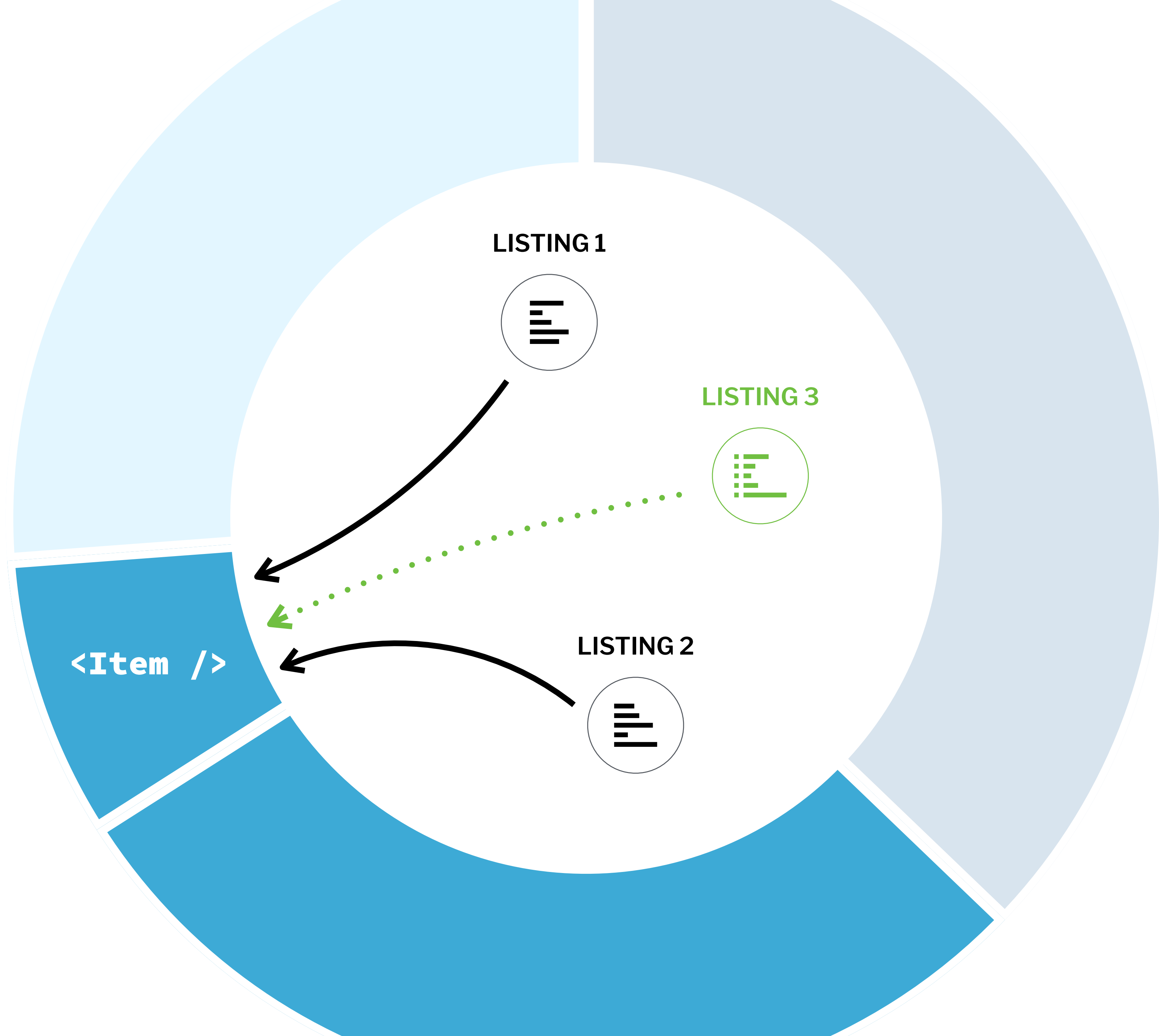


React Native



Expo























<Item />

---

**Content**

---

```
interface Props {
  body: string;
}
```

AUTHOR

CONSUMER

Content

```
<Item body="Content" />
```

```
interface Props {
  body: string;
  arrow?: boolean;
}
```

AUTHOR

CONSUMER

Content



```
<Item body="Content" arrow />
```

```
interface Props {
  body: string;
  icon?: 'arrow' | 'check';
}
```

AUTHOR

CONSUMER

Content	>
Enabled	✓

```
<Item body="Content" icon="arrow" />
<Item body="Enabled" icon="check" />
```

```
interface Props {
  body: string;
  icon?: 'arrow' | 'check';
  count?: number;
}
```

AUTHOR

CONSUMER

Content	>
Enabled	✓
Amount	5

```
<Item body="Content" icon="arrow" />
<Item body="Enabled" icon="check" />
<Item body="Amount" count={5} />
```

```
interface Props {
  body: string;
  icon?: 'arrow' | 'check';
  count?: number;
  selection?: string;
}
```

AUTHOR

CONSUMER

Content	>
Enabled	✓
Amount	5
Currency	EUR

```
<Item body="Content" icon="arrow" />
```

```
<Item body="Enabled" icon="check" />
```

```
<Item body="Amount" count={5} />
```

```
<Item body="Currency" selection="EUR" />
```



```
interface Props {
  body: string;
  extra?: React.ReactChild;  // <- inversion of control
}
```

AUTHOR

CONSUMER

Content	>	<Item body="Content" extra={<Icon name="arrow" />} />
Enabled	✓	<Item body="Enabled" extra={<Icon name="check" />} />
Amount	5	<Item body="Amount" extra={<Badge count={5} color={BLUE} />} />
Currency	EUR	<Item body="Currency" extra={<Text color={MUTED}>EUR</Text>} />

// simple implementation / interface

```
interface Props {  
  body: string;  
  extra?: React.ReactChild;  
}
```

.....

// increased usage effort  
// consumer has high control

```
<Item body="Amount" extra={  
  <Badge count={5} color={BLUE} />  
}/>
```

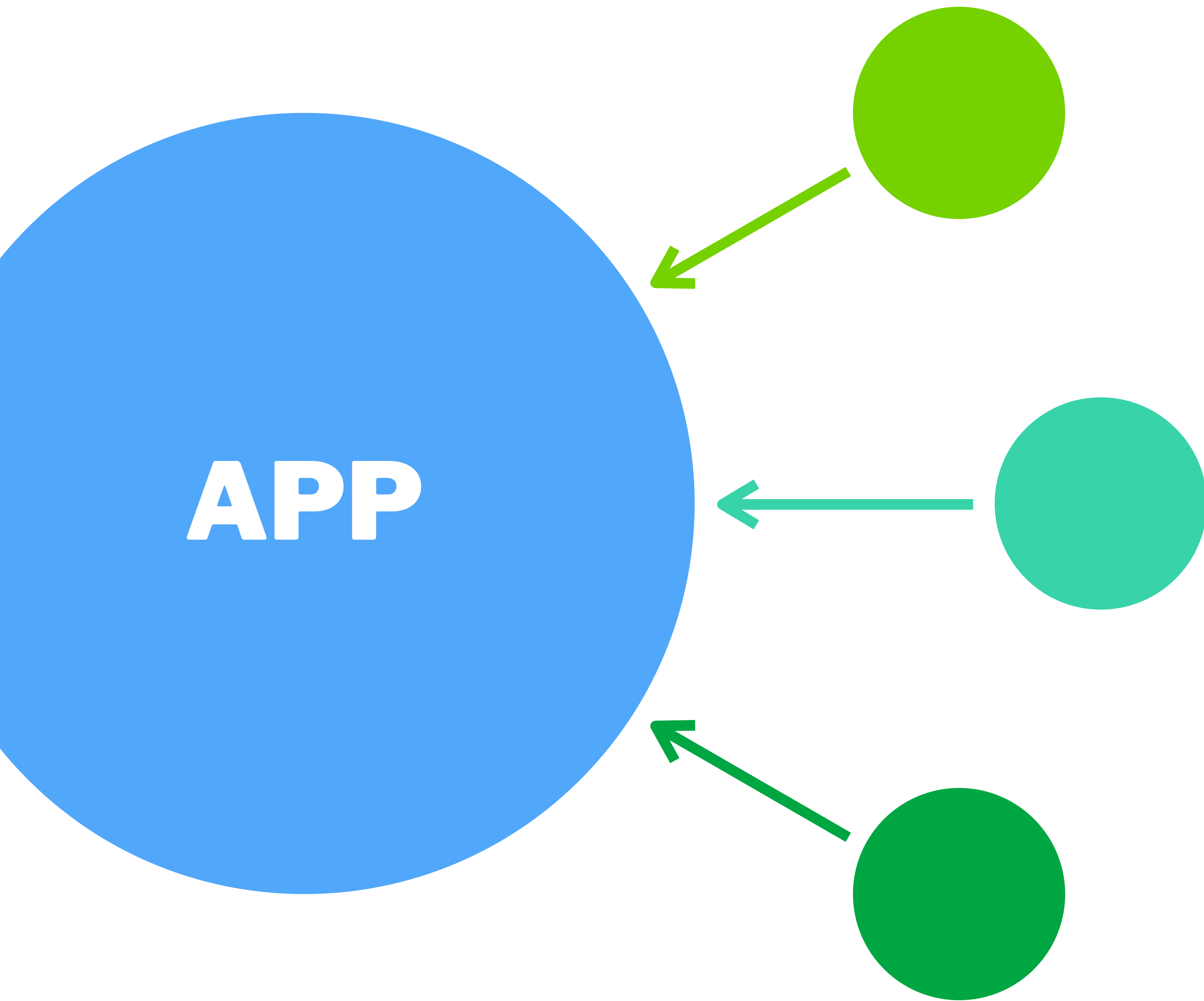
// complex implementation / interface

```
interface Props {  
  body: string;  
  icon?: 'arrow' | 'check';  
  count?: number;  
  selection?: string;  
}
```

.....

// easy to use  
// consumer has low control

```
<Item body="Amount" count={5} />
```



**END CUSTOMERS**



## Price Range

**\$500 - \$10 200**

Average price: \$1 200



---

<https://dribbble.com/cuberto>



min price  
lei 44

—

max price  
lei 4500+

Show 300+ stays

**TAKEAWAYS**

**WE ARE BOTH**  
**AUTHORS AND CONSUMERS**

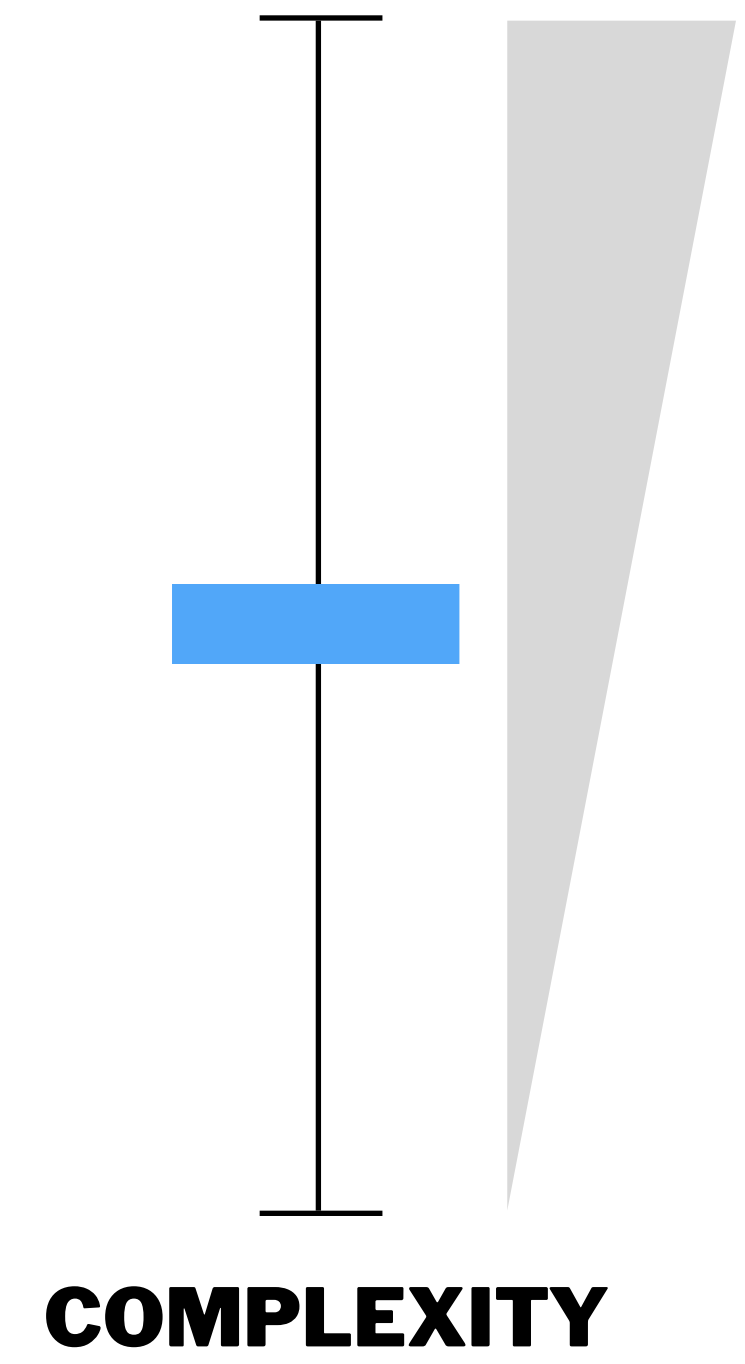
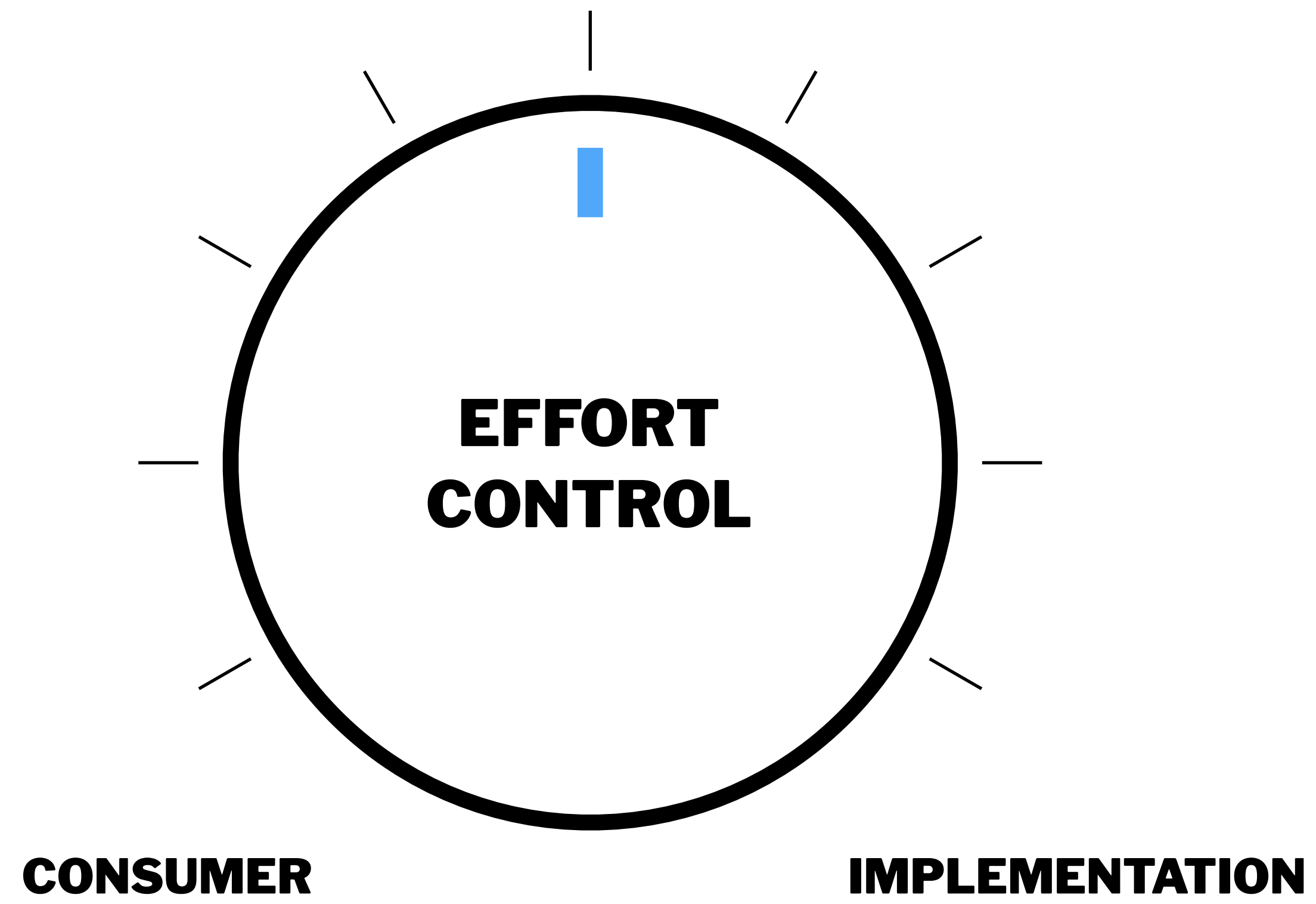
**COMPLEXITY**  
**IS INHERENT**

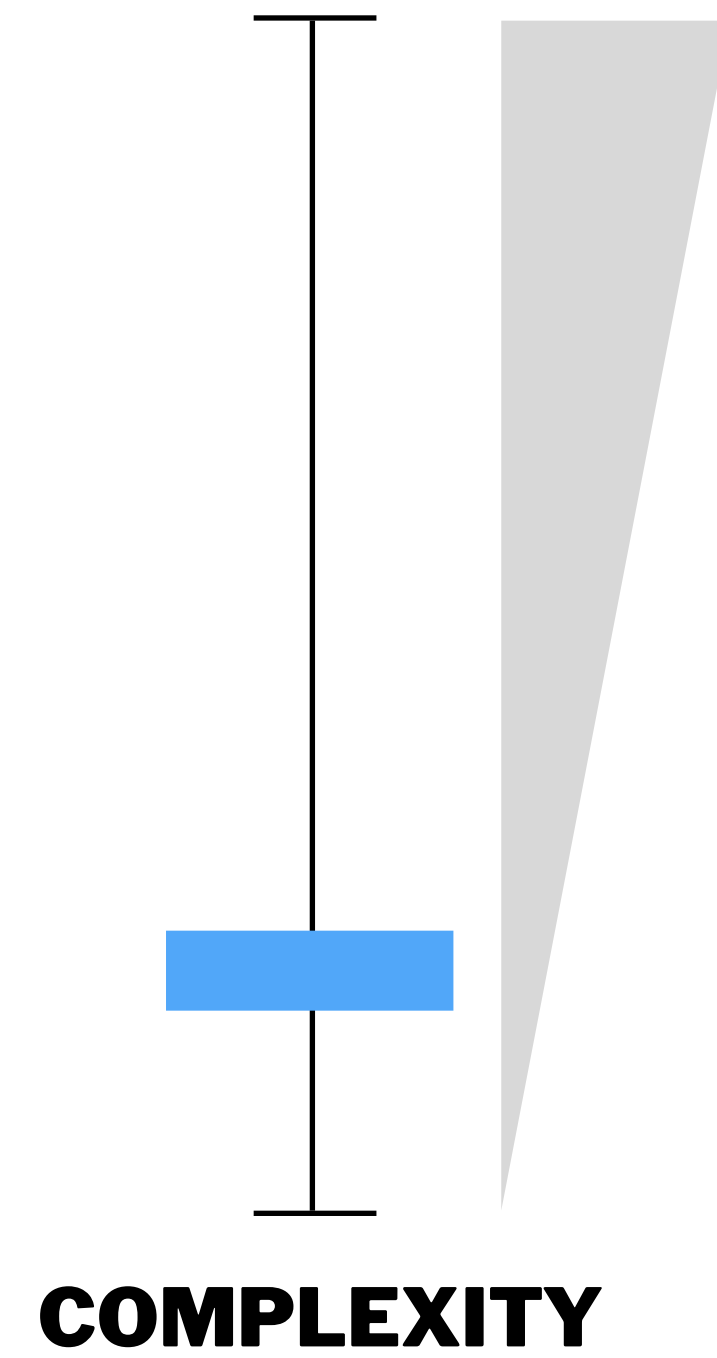
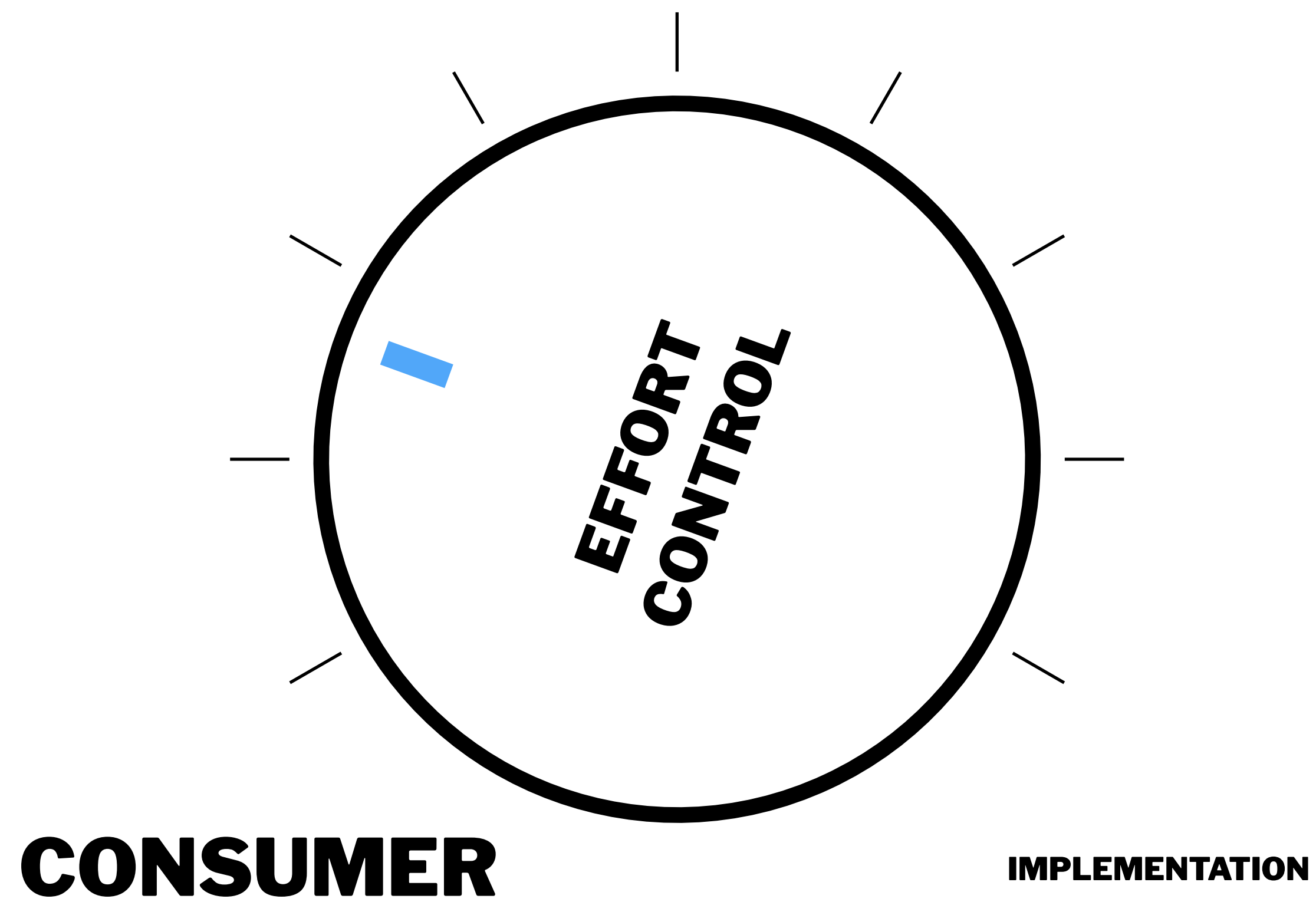


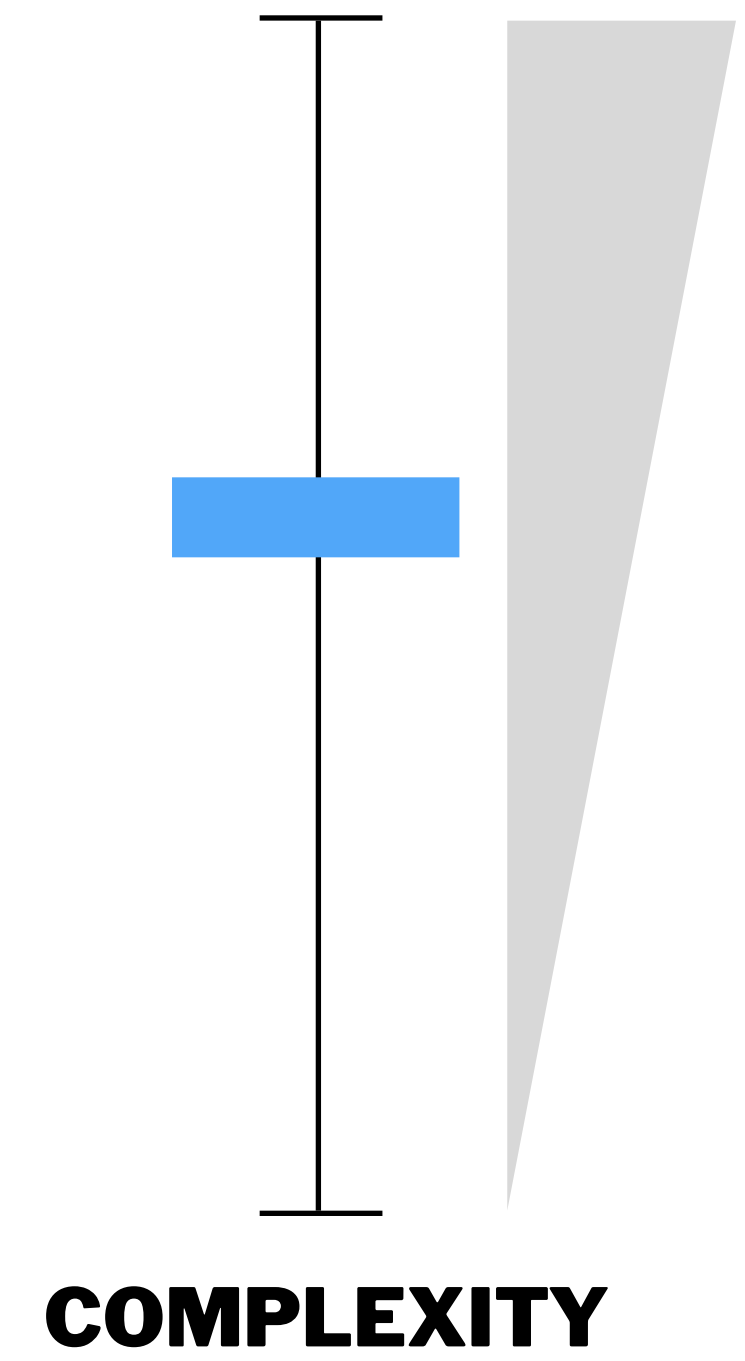
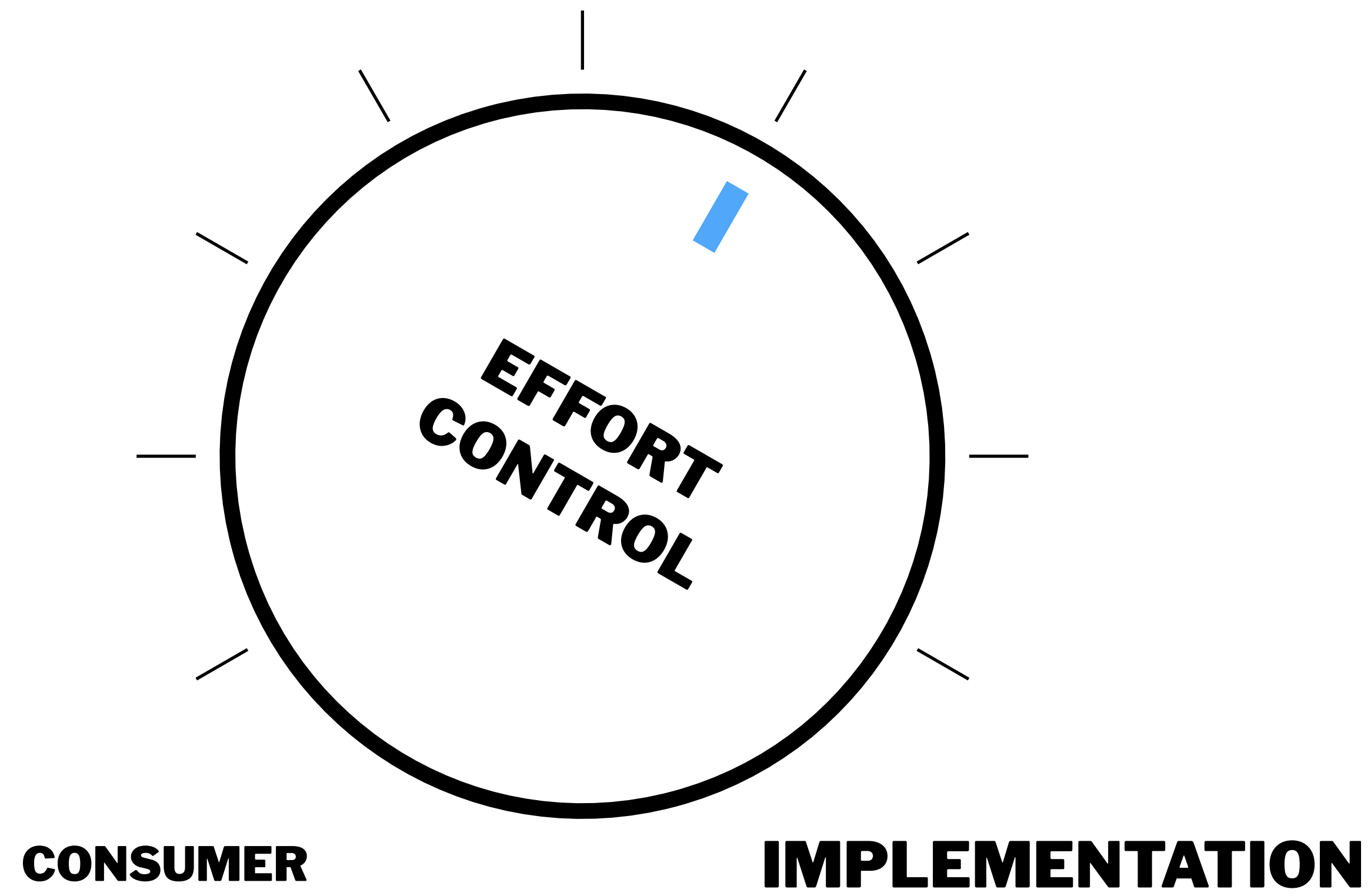
**COMPLEXITY**  
**IS NOT INHERENTLY BAD**

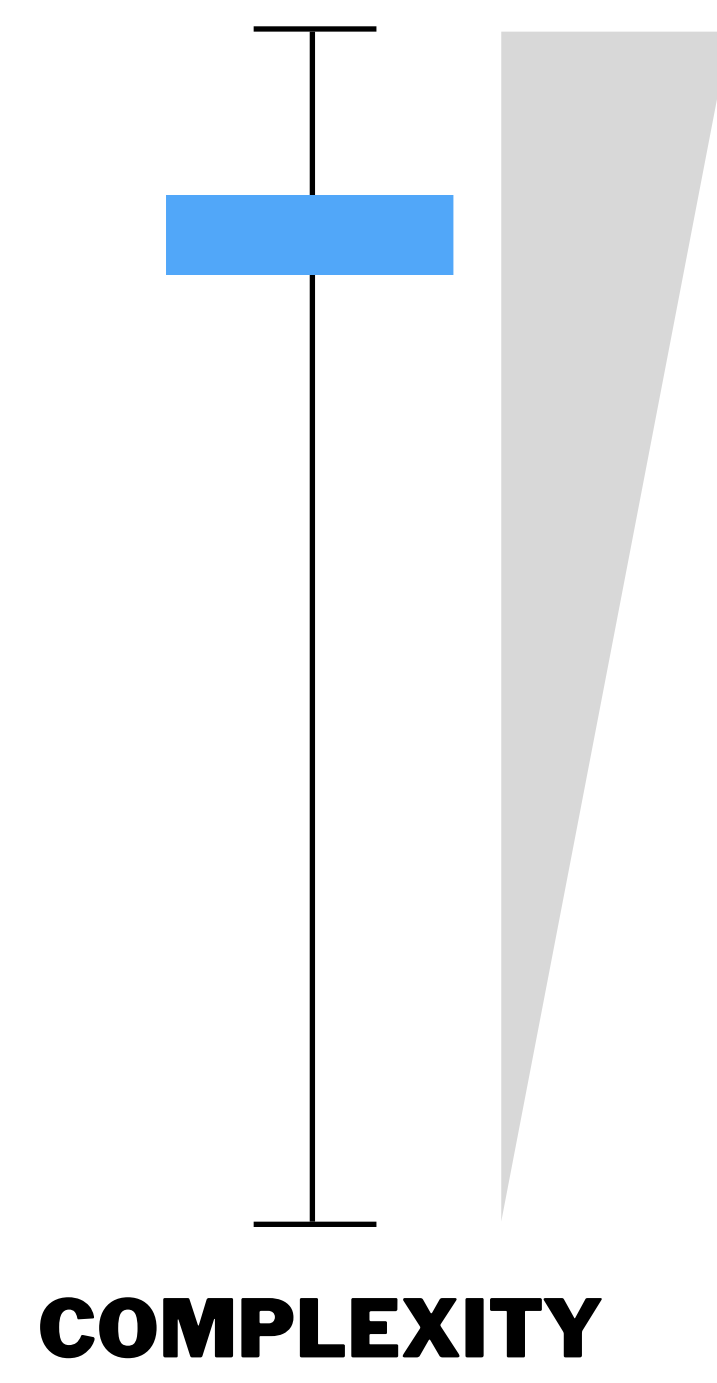
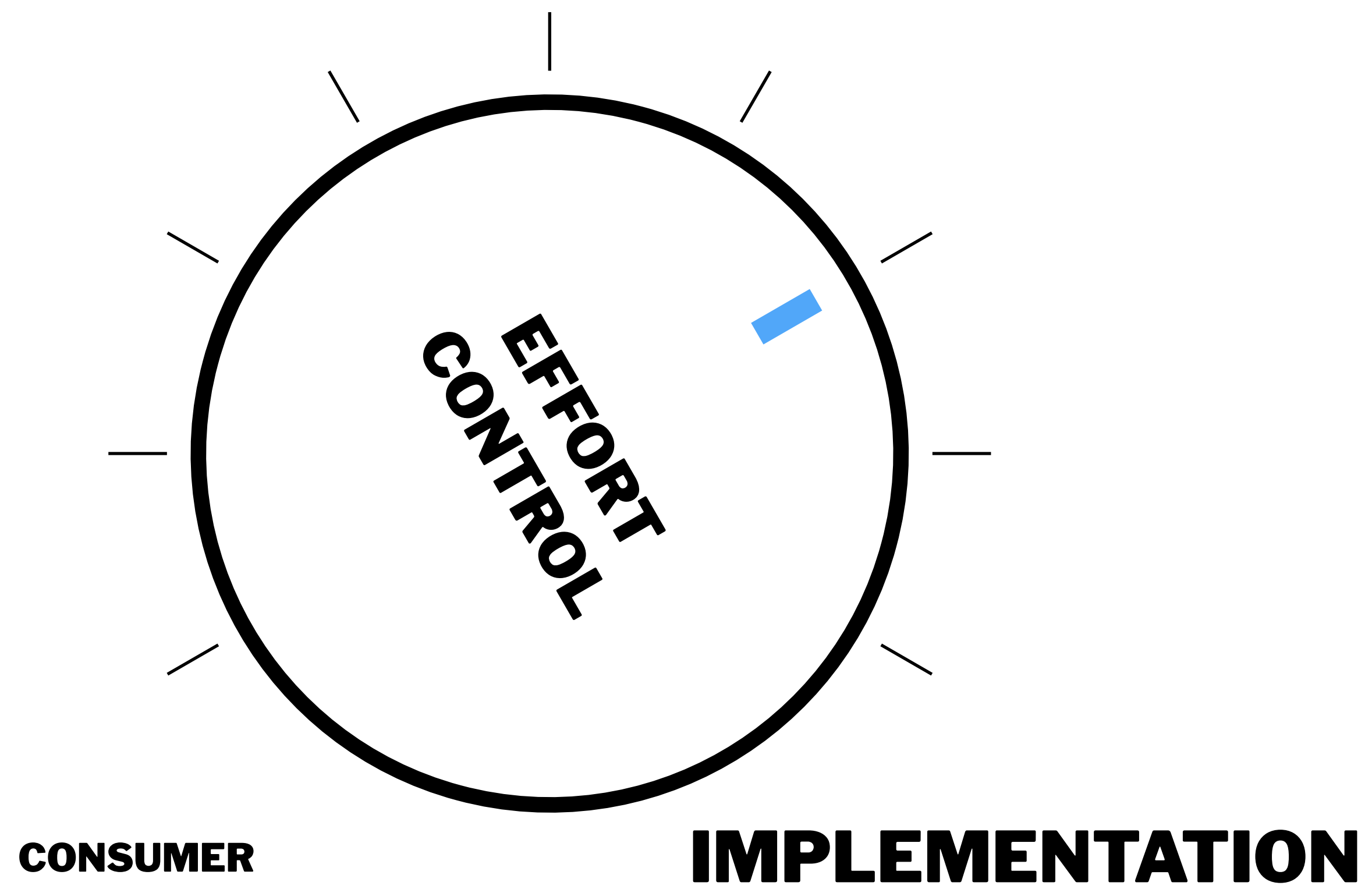
**COMPLEXITY  $\Rightarrow$  EFFORT**

**EFFORT === CONTROL**









**THANK YOU**