The
**SOFTWARE
ESSENTIALIST**

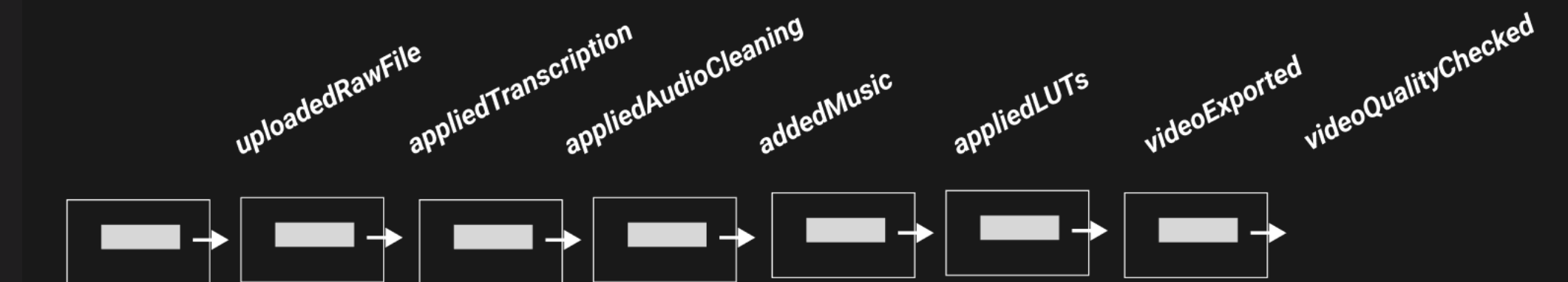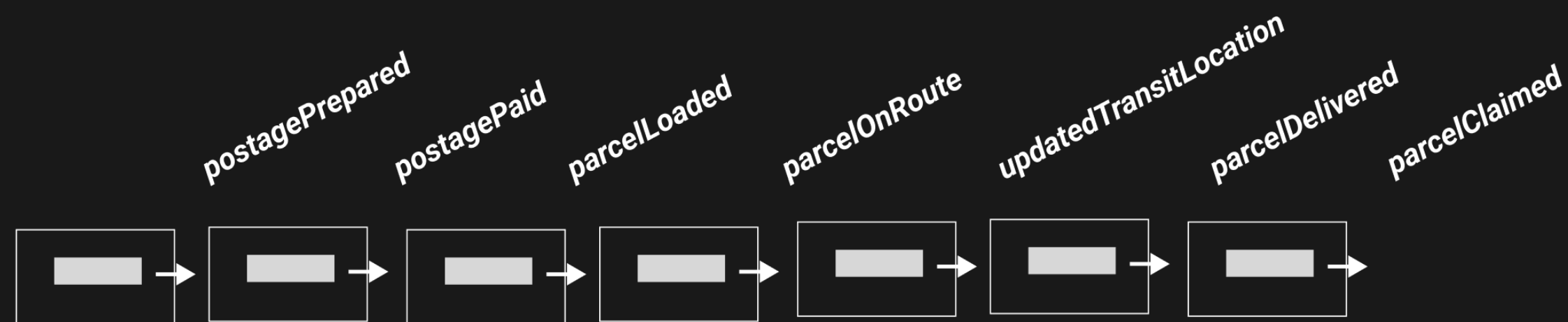# Systems Thinking (The Most Important Mental Model for Developers)

What does mailing a package and editing a video have in common?

essentialist.dev

They can both be broken down into systems...

...and Systems Thinking is vital for every software developer

essentialist.dev

What we'll cover

- *What is Systems Thinking & how does it work?*
- *Why is it so powerful to us as developers?*

What is a system?

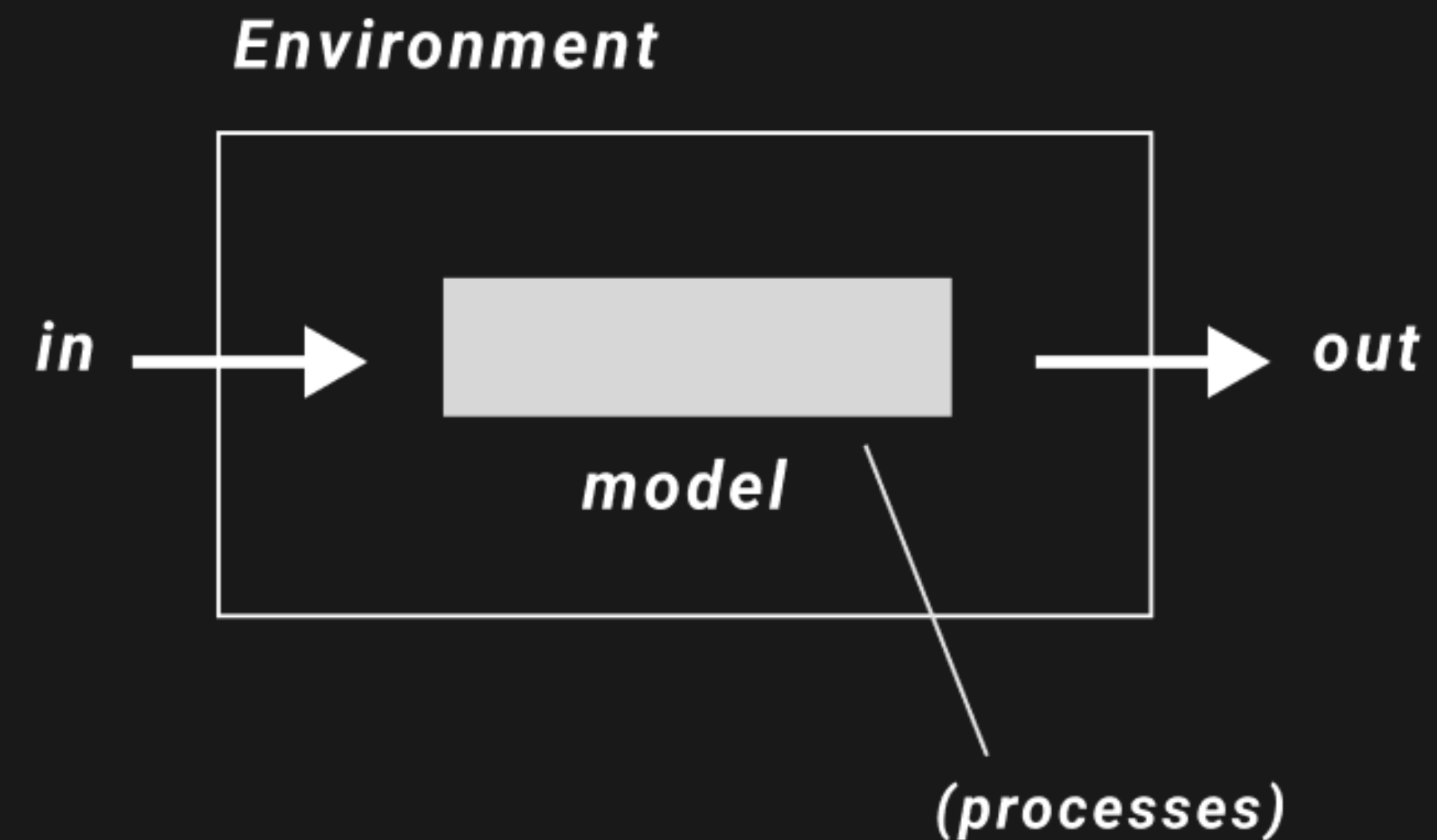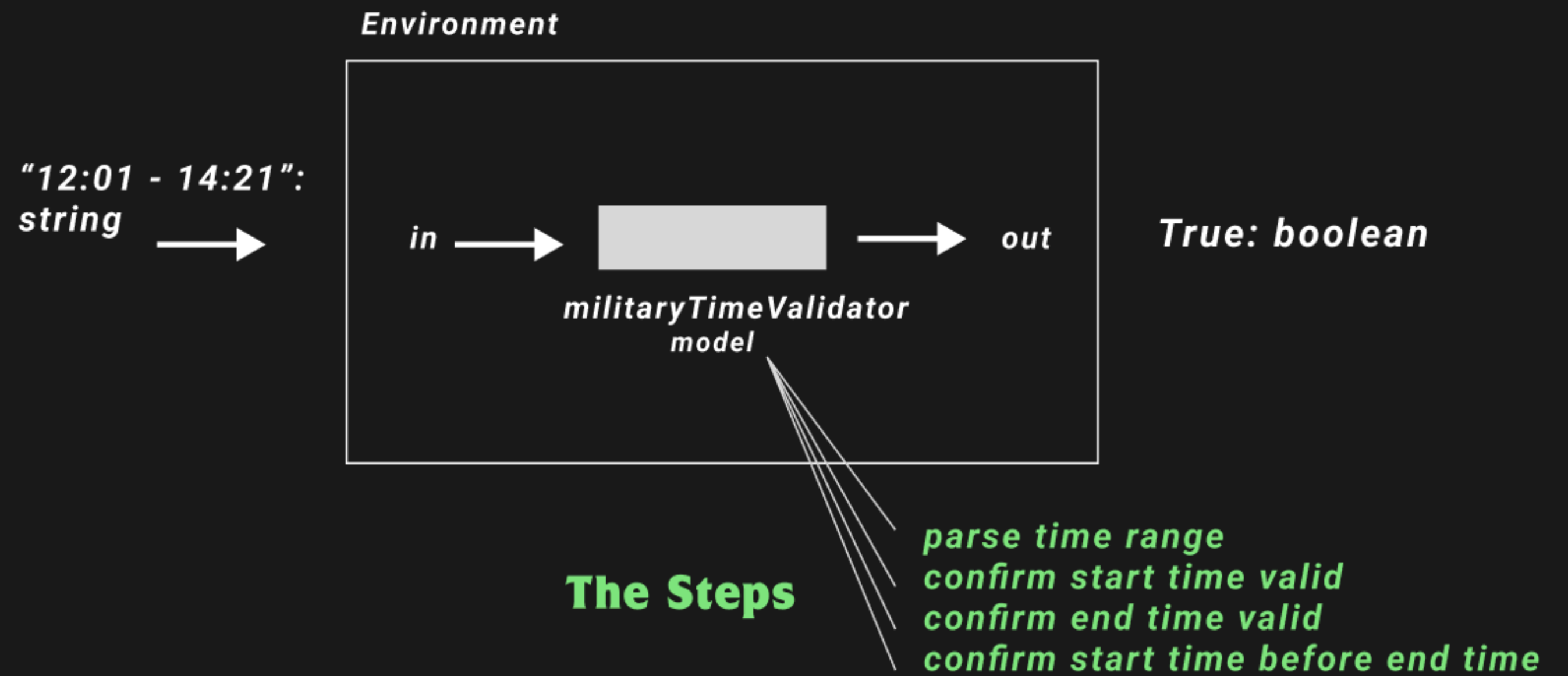essentialist.dev

Systems Thinking
General

You can abstract most things into systems comprised of a set of:
- inputs
- outputs
- the environment it resides
- the model/processes it runs

Environment

in → [model] → out

(processes)

essentialist.dev

The world is a patchwork of systems

essentialist.dev
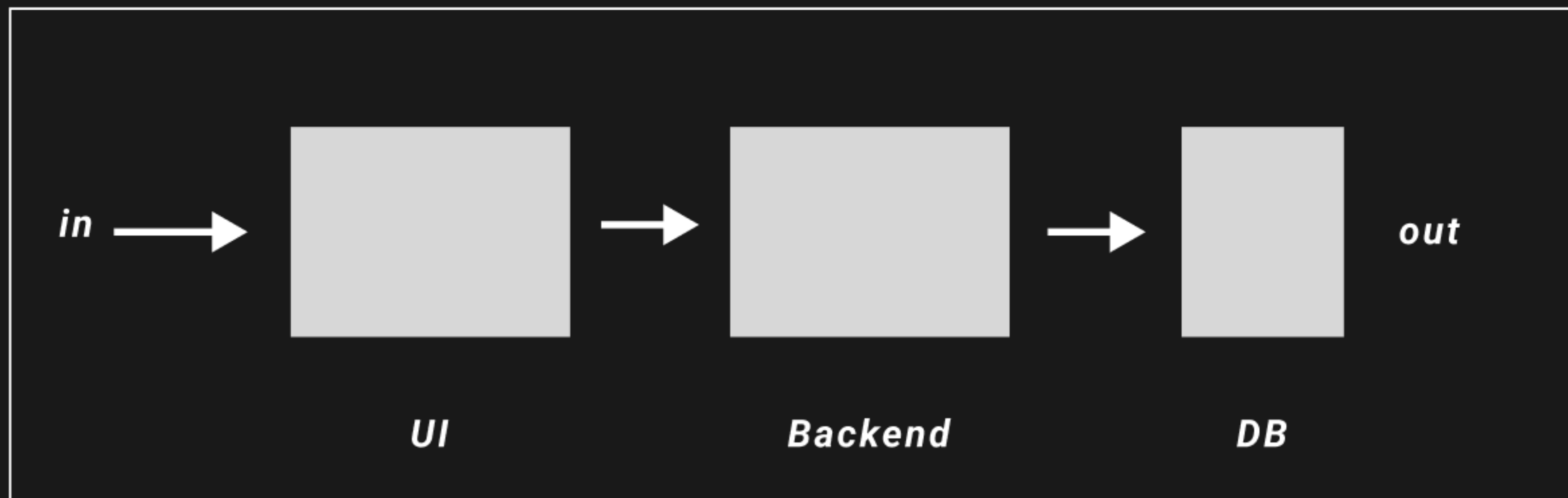
```
function militaryTimeValidator (time: string): boolean {
  let isValid = false;
  // Validate
  return isValid
}
```

Environment

"12:01 - 14:21": string

in → out

militaryTimeValidator
model

True: boolean

The Steps

parse time range
confirm start time valid
confirm end time valid
confirm start time before end time

essentialist.dev

Business Systems

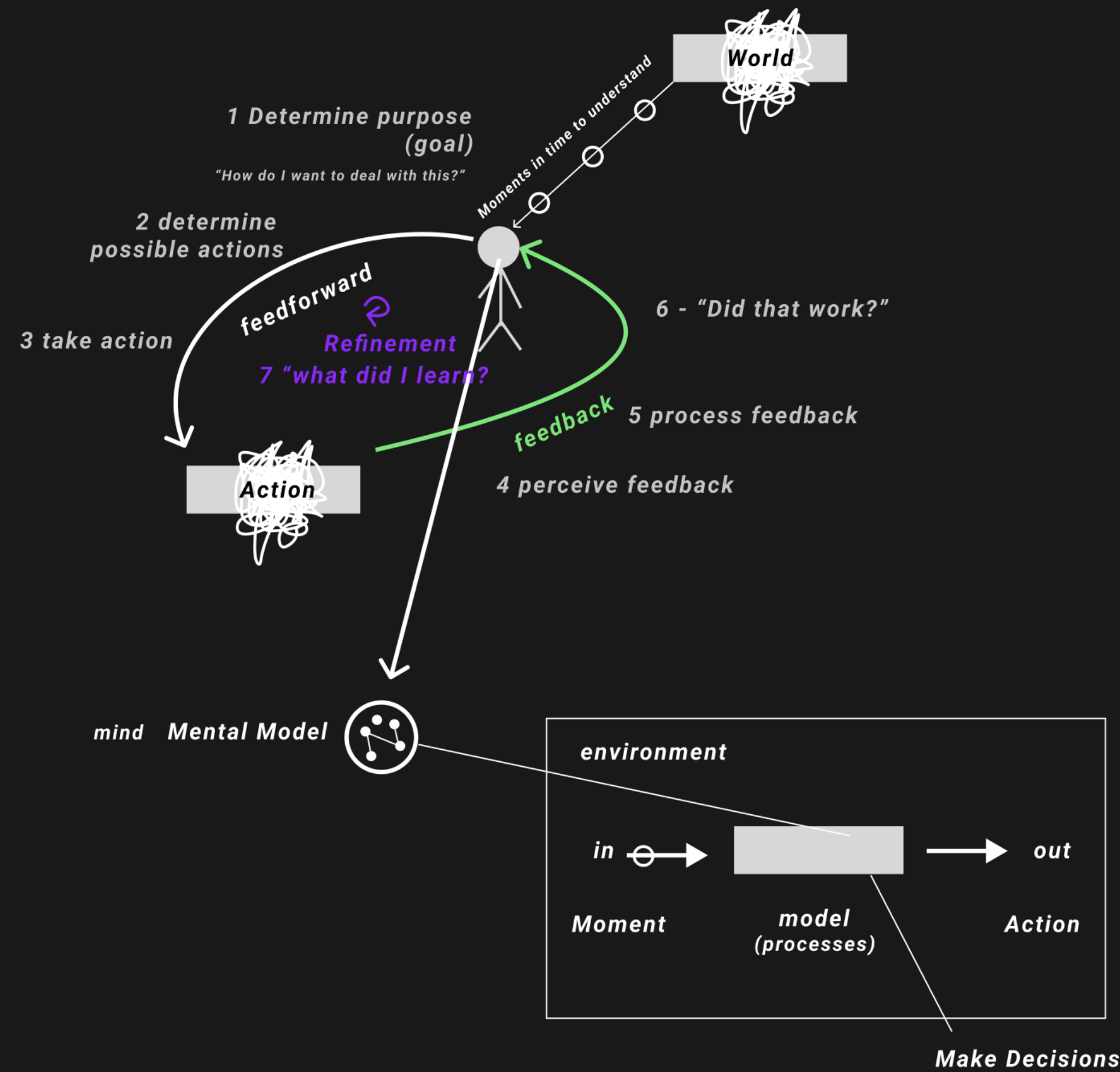*A business is a system where Value is the fuel and it takes on different forms (solutions/ideas, offers, content, fulfillment, payment, etc).*

We're right here :) Changing button colors and font sizes.

Env

in

problems → Leadership → solutions → Value Creation → offer → Marketing → content → Sales → onboarding → Operations → fulfilment → Finance → payment out

all previous inputs

essentialist.dev

World

Moments in time to understand

1 Determine purpose
(goal)

"How do I want to deal with this?"

2 determine
possible actions

feedforward

Refinement
7 "what did I learn?"

3 take action

6 - "Did that work?"

feedback   5 process feedback

4 perceive feedback

Action

mind   Mental Model

environment

in   out

Moment   model
            (processes)   Action

Make Decisions

"MASTERY IS WHEN YOU
CONTINUALLY TAKE
RIGHT ACTION"

essentialist.dev

The purpose of a system is to achieve a goal

- *Mailing a package?*

  - *Package not mailed → packageMailed*

- *Editing a video*

  - *Acquired video files → exportedEditedVideo*

- *Validating text*

  - *Raw text received → textValidated*

essentialist.dev

Why is this so powerful for developers?

- *1 | Problem Decomposition*

- *2 | Debug-ability*

- *3 | Refinement & learning*

essentialist.dev

What we covered

essentialist.dev

What's OUR goal as developers? What's the system there?

Let's get clear on this next.