Accionlabs

Accion INNOVATION SUMMIT 2023

02-05 March 2023,
Sofitel Dubai
The Palm Jumeirah
Dubai

Accion
INNOVATION
SUMMIT 2023

Accionlabs

Finite State Machines to Model and Build Complex User Interfaces

Accion
INNOVATION
SUMMIT 2023

Accionlabs

Vaibhav Satam

Head of frontend application COE

Software architect with extensive experience in designing complex frontend applications.

He enjoys working with startups and enterprises to identify challenging problems and opportunities that can be solved with software and innovation.

Accion
INNOVATION
SUMMIT 2023

Accionlabs

Kejal Shah

Tech Architect

Technical Architect with varied experience in frontend applications.

Dedicated and passionate about creating diverse innovative solutions.

Table of Contents

Accion
INNOVATION
SUMMIT 2023

Accionlabs

Introduction

Accion
INNOVATION
SUMMIT 2023

Accionlabs

Problem Definition

Increasing nature of frontend complexity

Accionlabs

Complexity of User Expectations

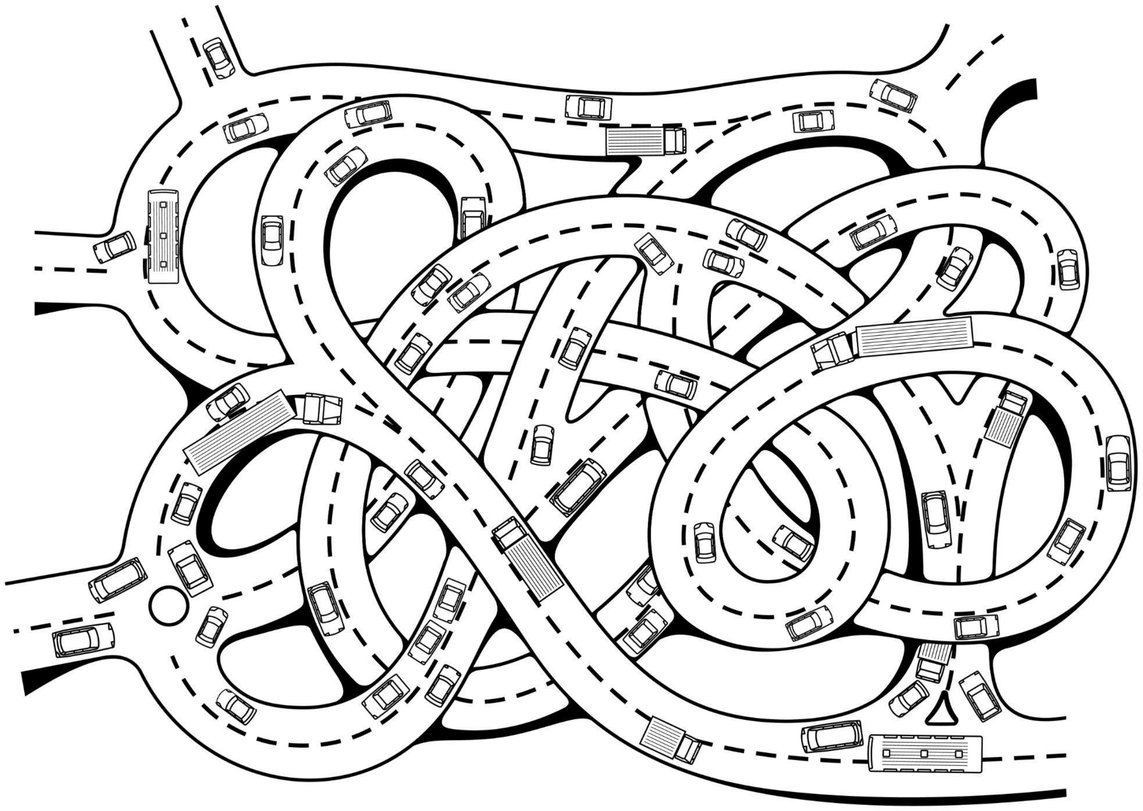
- In the middle of the 2000s, the web was mostly composed of static web pages
- As time went on, the sites we built started to grow in complexity — with features like blog posts and forums
- With each evolution of web UI, we started to build increasingly complex websites and applications
- The web today requires interfaces that are far more complex and event driven

Increasing nature of frontend complexity

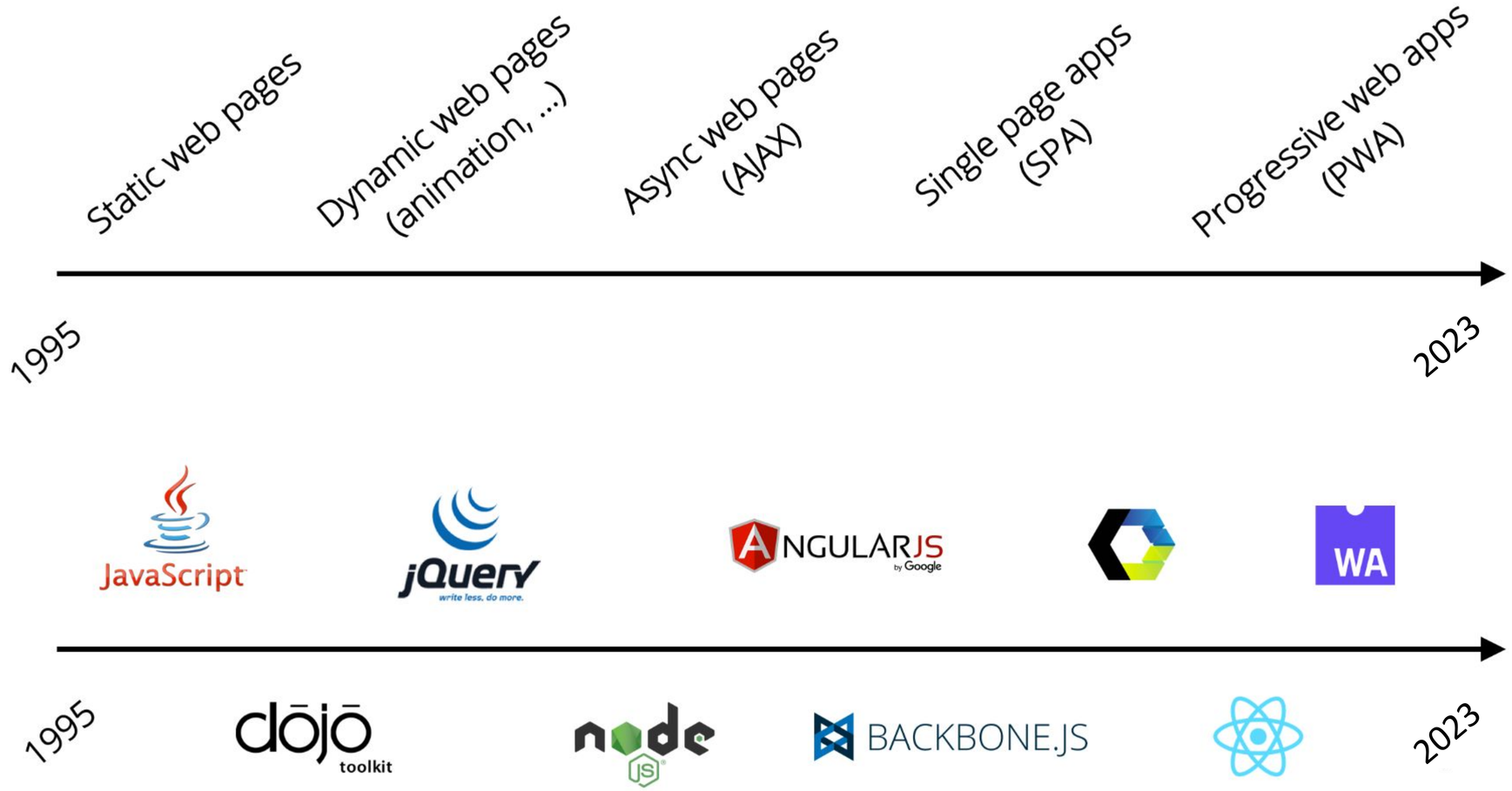
Accionlabs

Complexity of User Expectations

- In the middle of the 2000s, the web was mostly composed of static web pages
- As time went on, the sites we built started to grow in complexity — with features like blog posts and forums
- With each evolution of web UI, we started to build increasingly complex websites and applications
- The web today requires interfaces that are far more complex and event driven



Evolution of Web User Interfaces



Accionlabs

Dealing with Complexities of User Interfaces

- User Interfaces need to address a variety of complexities such as
  - Accessibility
  - Performance
  - Compatibility with multiple devices, browsers and operating systems
  - Varying degree of connectivity

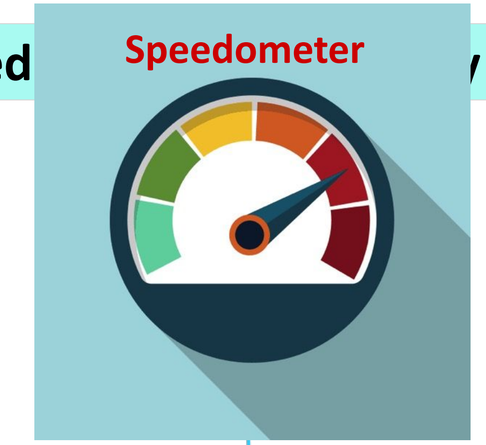
- User Interfaces need to be more **Event Driven**
  - Form based UI is too cumbersome for users
  - UI needs to mimic games, where complexity is hidden under the surface
  - UI need to respond to user events intelligently, and be state aware

Increasing nature of frontend complexity

Dealing with Complexities of User Interfaces

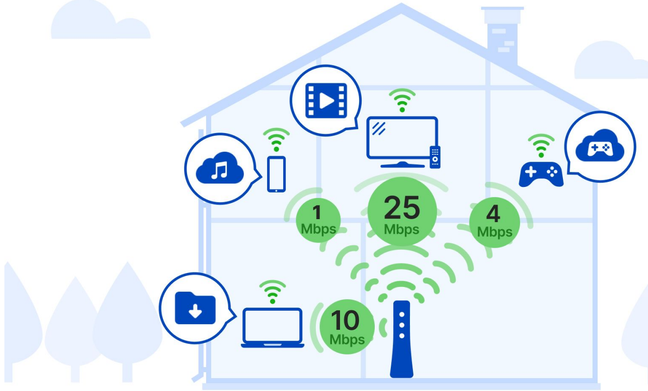
User Interfaces need **y of complexities such as**

Accessibility



Performance

Responsive design



Data connectivity

Varying degree of connectivity

User Interfaces need to be more Event D

e devices, systems

Form-based UI is too cumbersome for users

UI needs to mimic games, where complexity is hidden under the surface

UI need to respond to user events intelligently, and be state aware

Increasing nature of frontend complexity

Accionlabs

Dealing with Complexities of User Interfaces

User Interfaces need to address a variety of complexities such as

Accessibility

Performance

Compatibility with multiple devices, browsers and operating systems

Varying degree of connectivity

User Interfaces need to be more Event Driven

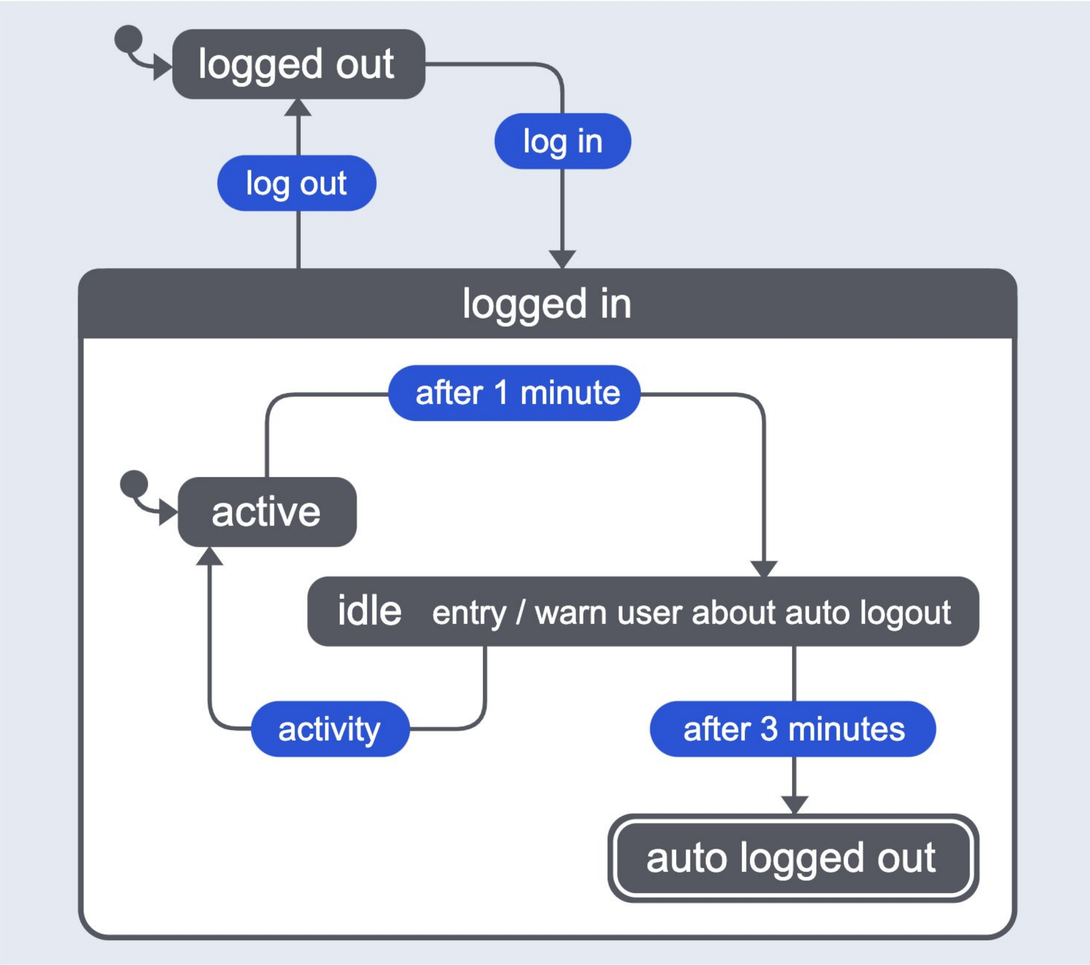
Form-based UI is too cumbersome for users

UI needs to mimic games, where complexity is hidden under the surface

UI need to respond to user events intelligently, and be state aware

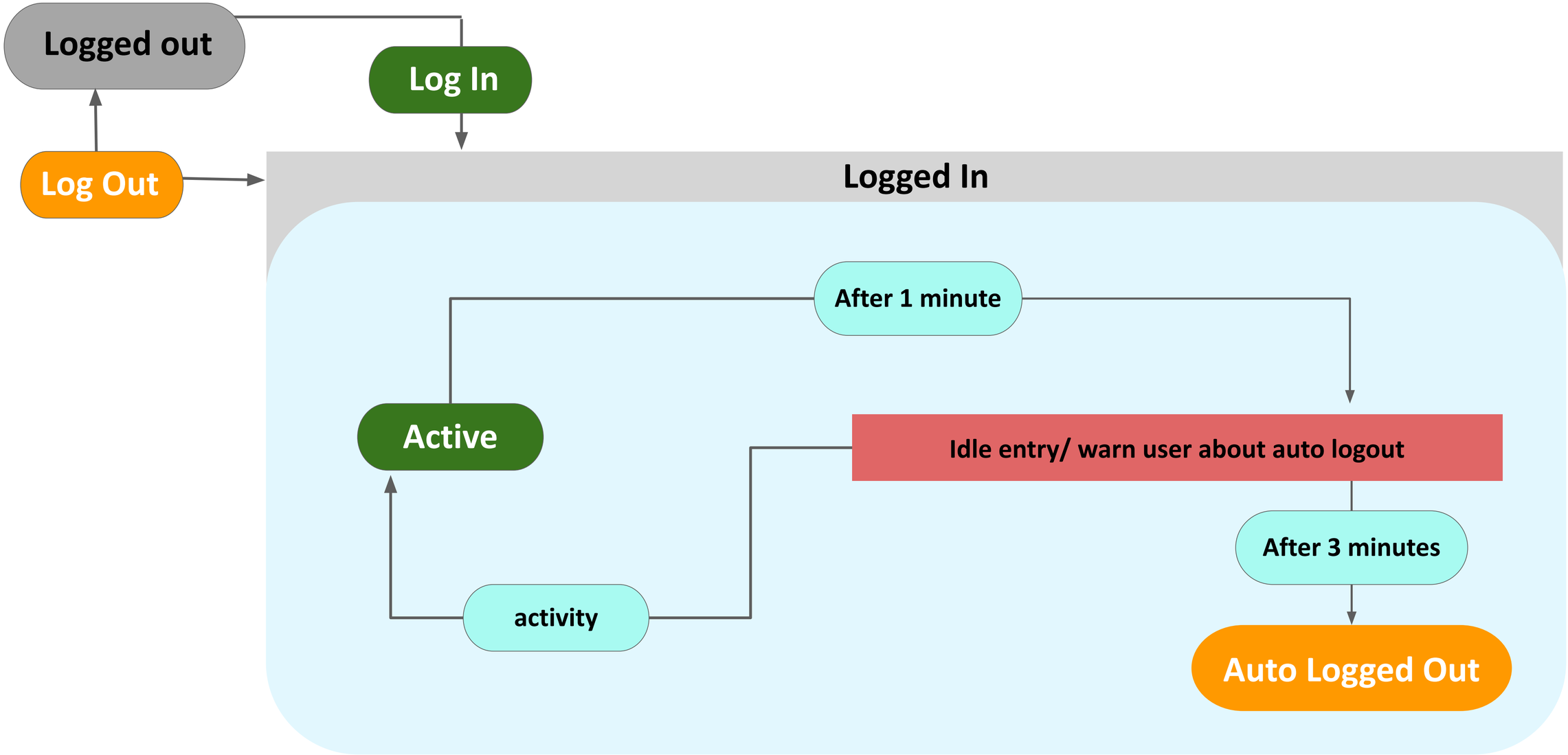
Example of Event Driven v/s Conventional UI

Example: UI need to respond to user events intelligently, and be state aware



Example of Event Driven v/s Conventional UI

Example: UI need to respond to user events intelligently, and be state aware



Accion INNOVATION SUMMIT 2023

Accionlabs

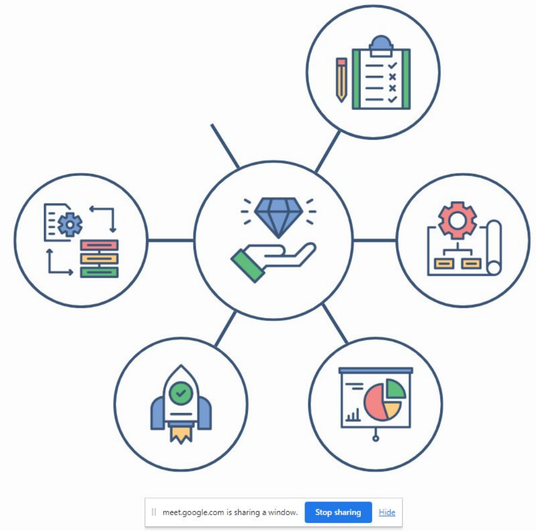
Conventional State Management

Conventional Solutions - State Management

Over the last several years, there has been a ton of innovation in the realm of State Management; from Redux to MobX to Vuex and beyond.

These technologies did not provide a way to design state, but just to manage state

Event driven UI requires much more complex design of states



Accion
INNOVATION
SUMMIT 2023

Accionlabs

State Machines - Simplifying Complex UI

A brief introduction to state machines and statecharts

Accionlabs

- **Finite Number of States**
  - A state machine, per the academic definition, is any abstract machine that can be in exactly one of a finite number of states at a given time.

- **State Charts**
  - With statecharts, we can model each discrete piece of logic in our application with a state machine, then visualize how each of those pieces fit together to form a complete application.

- **State Machines for Javascript**
  - State machines have been a mainstay in other areas of computer programming for many, many years, primarily due to their predictability. They've only recently started to make their way into the zeitgeist of javascript developers.

A brief introduction to state machines and statecharts

Accionlabs

- **State Machines for Complex Use Cases**
  - NASA uses state machines to model the Space Launch System solid rocket boosters. Video game developers have leaned on state machines for decades.

- **State Machines for Design and Modeling**
  - State machines provide clear instructions on when and how that state should change, giving strong control over the behavior of applications.

A brief introduction to state machines and statecharts

Accionlabs

Finite Number of States

A state machine, per the academic definition, is any abstract machine that can be in exactly one of a finite number of states at a given time

State Charts

With statecharts, we can model each discrete piece of logic in our application with a state machine, then visualize how each of those pieces fit together to form a complete application

State Machines for Javascript

State machines have been a mainstay in other areas of computer programming for many, many years, primarily due to their predictability. They've only recently started to make their way into the zeitgeist of javascript developers

State Machines for Complex Use Cases

A state machine, per the academic definition, is any abstract machine that can be in exactly one of a finite number of states at a given time

State Machines for Design and Modeling

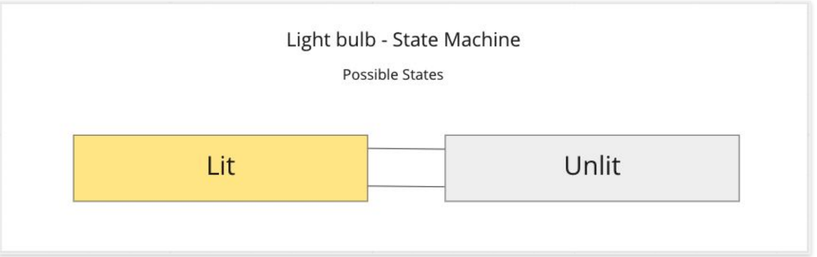
With statecharts, we can model each discrete piece of logic in our application with a state machine, then visualize how each of those pieces fit together to form a complete application

Modeling complex user interfaces with finite state machines

Here are a few specific ways that FSMs can be useful for modeling complex user interfaces:

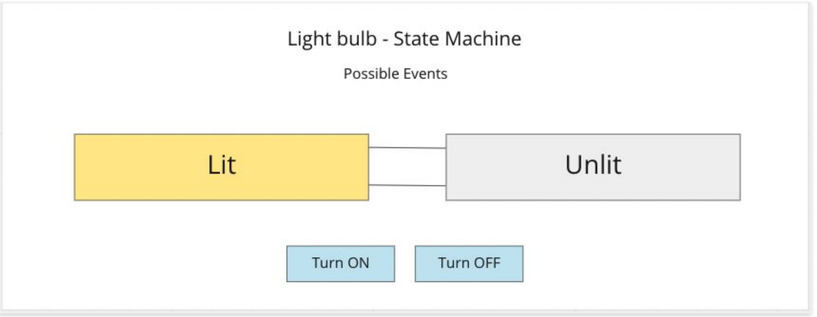
Describing the different states of the interface

A user interface can have many different states, depending on factors like user input, system state, and external events. FSMs provide a way to formally define these different states and how they relate to each other.



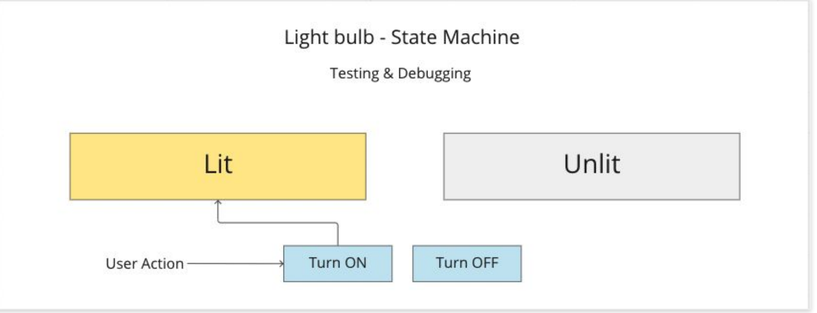
Defining the events that trigger state transitions

In an FSM, each transition between states is triggered by a specific event. By defining these events explicitly, designers can ensure that they have considered all the possible ways that the interface might change state.



Supporting testing and debugging

FSMs provide a clear and formal representation of the behavior of the user interface. This can be useful when testing and debugging the interface.

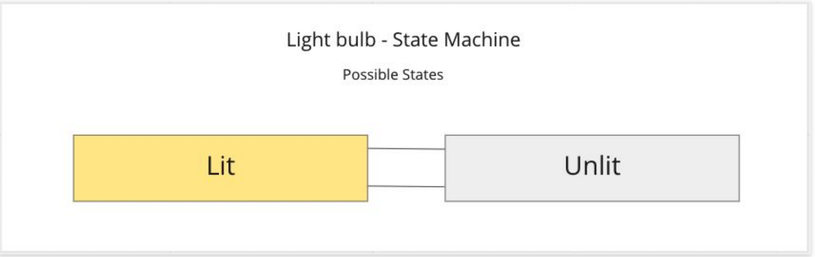


Modeling complex user interfaces with finite state machines

Here are a few specific ways that FSMs can be useful for modeling complex user interfaces:

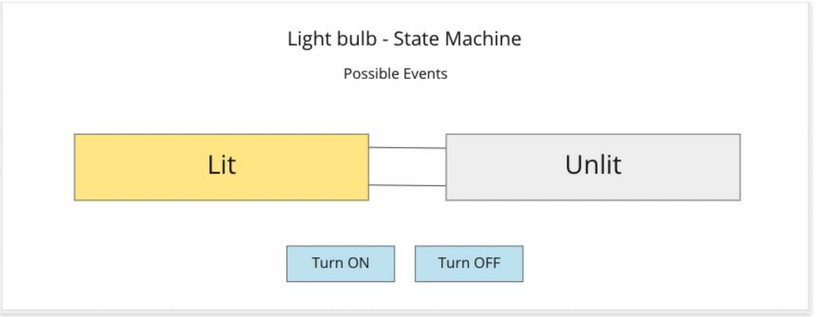
Describing the different states of the interface

A user interface can have many different states, depending on factors like user input, system state, and external events. FSMs provide a way to formally define these different states and how they relate to each other.



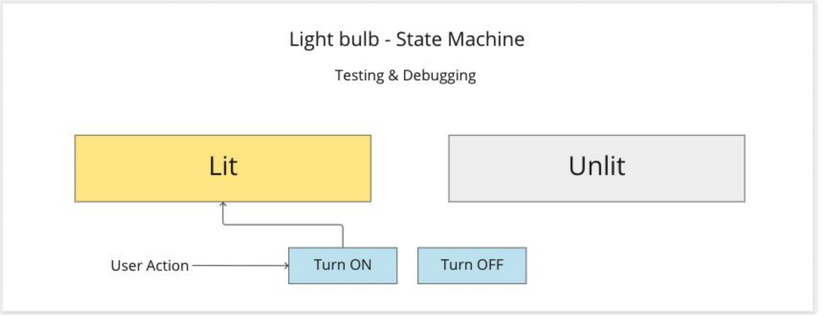
Defining the events that trigger state transitions

In an FSM, each transition between states is triggered by a specific event. By defining these events explicitly, designers can ensure that they have considered all the possible ways that the interface might change state.



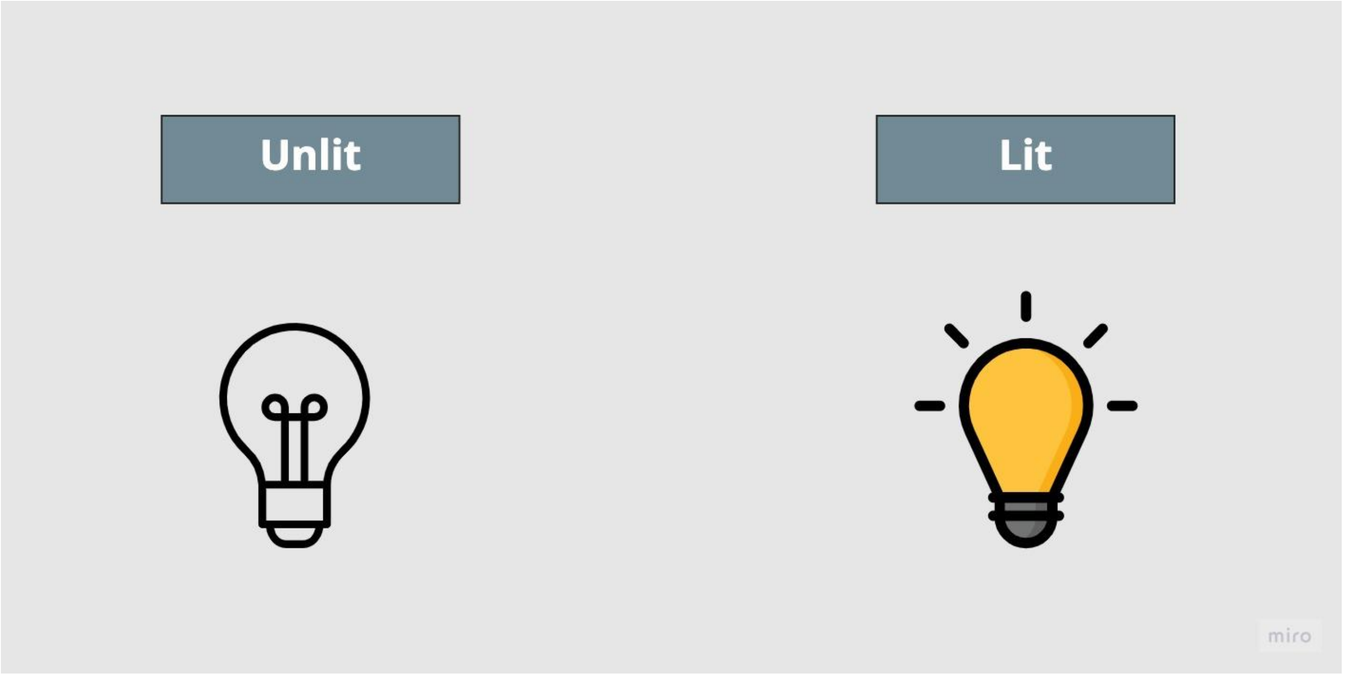
Supporting testing and debugging

FSMs provide a clear and formal representation of the behavior of the user interface. This can be useful when testing and debugging the interface.



States

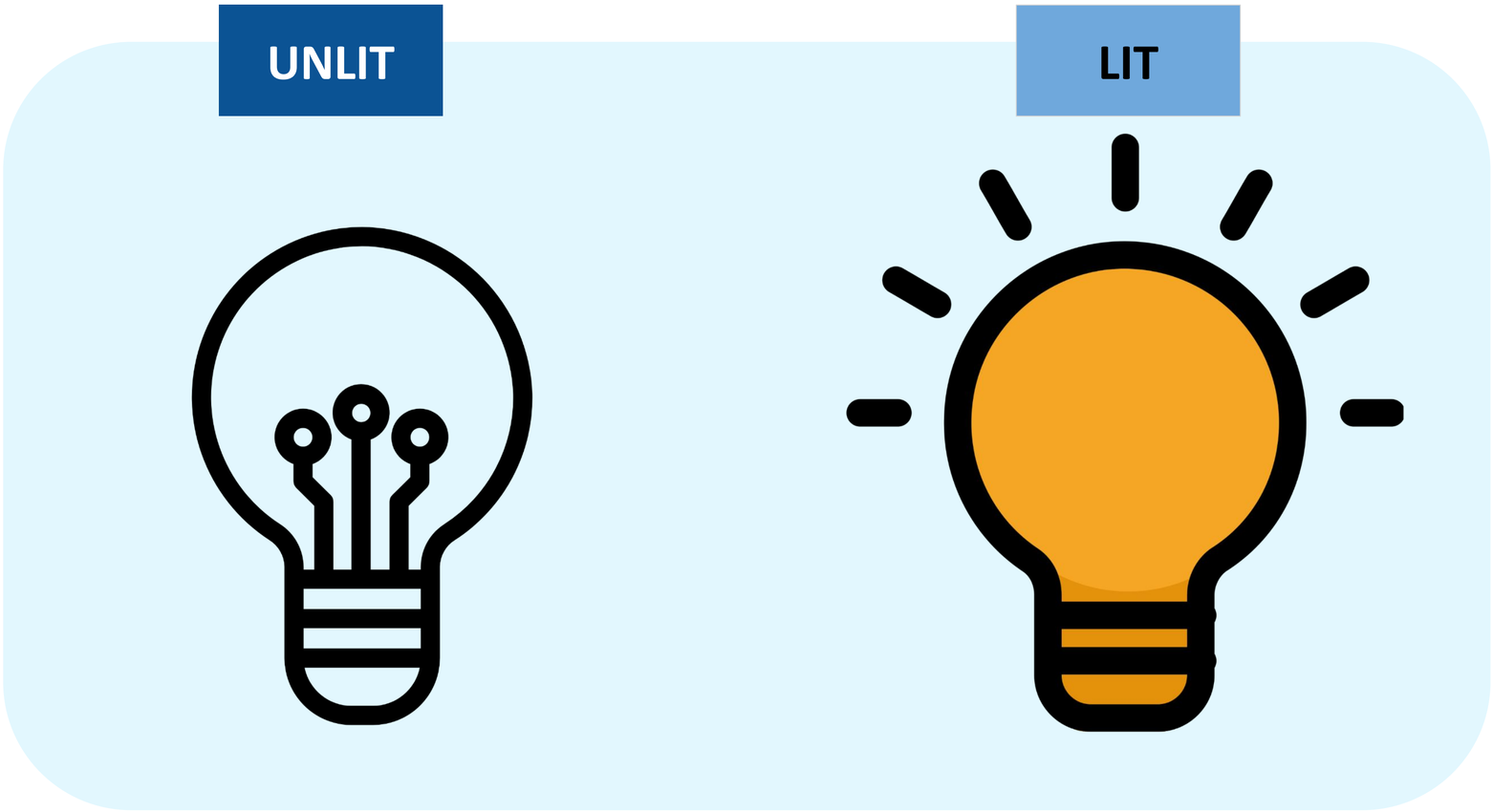
To draw a statechart for the process of a light bulb, there are two states that would first come to mind:



A light bulb is either lit or unlit at all times. The light bulb cannot be both lit and unlit at the same time.

States

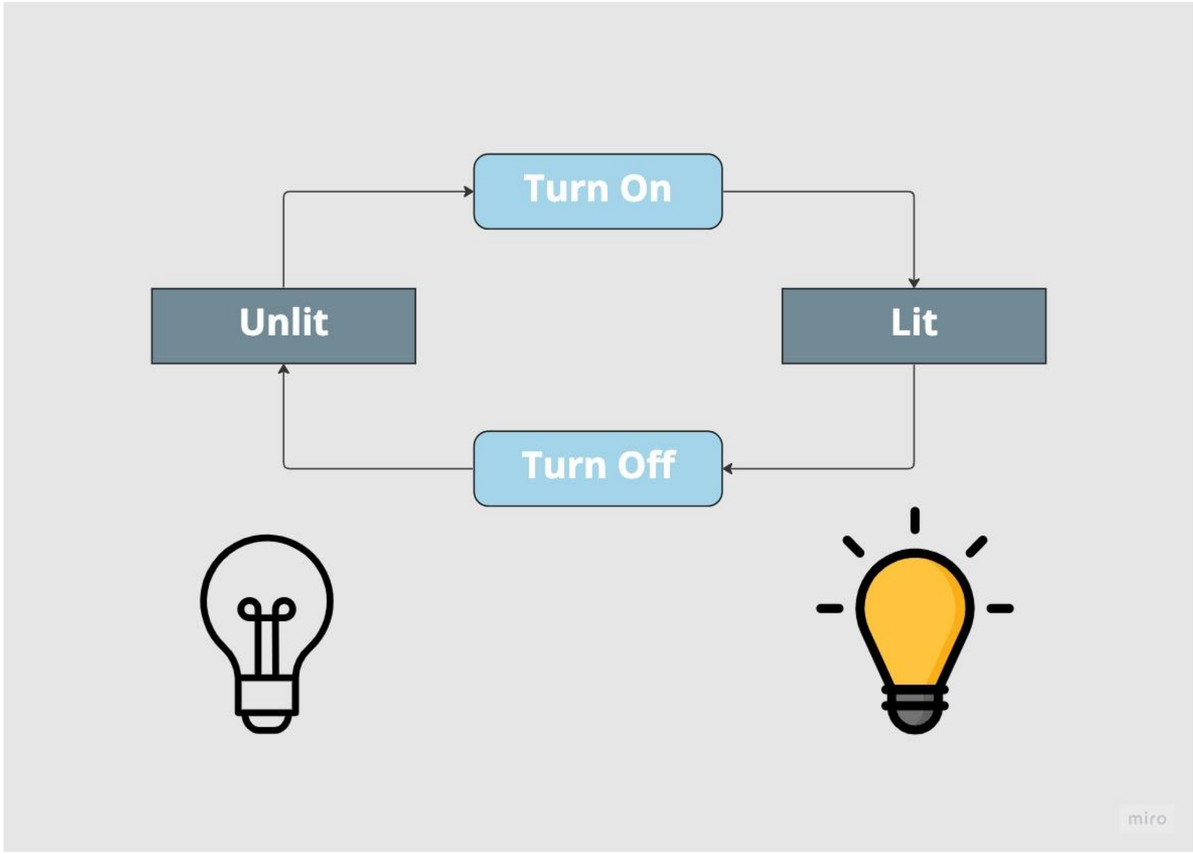
To draw a statechart for the process of a light bulb, there are two states that would first come to mind:



A light bulb is either lit or unlit at all times. The light bulb cannot be both lit and unlit at the same time.

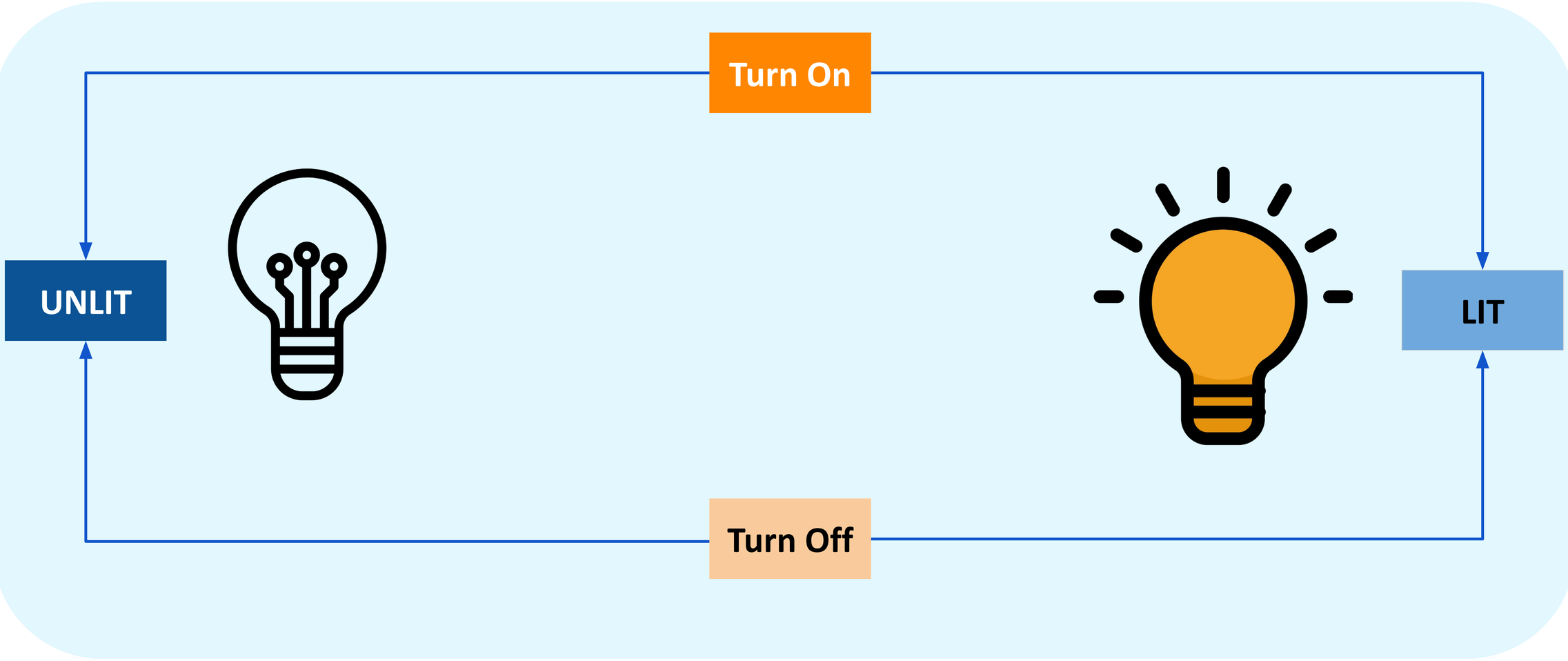
Transitions and events

How the light bulb goes between lit and unlit is through transitions. A transition is caused by an event that results in the change of state.



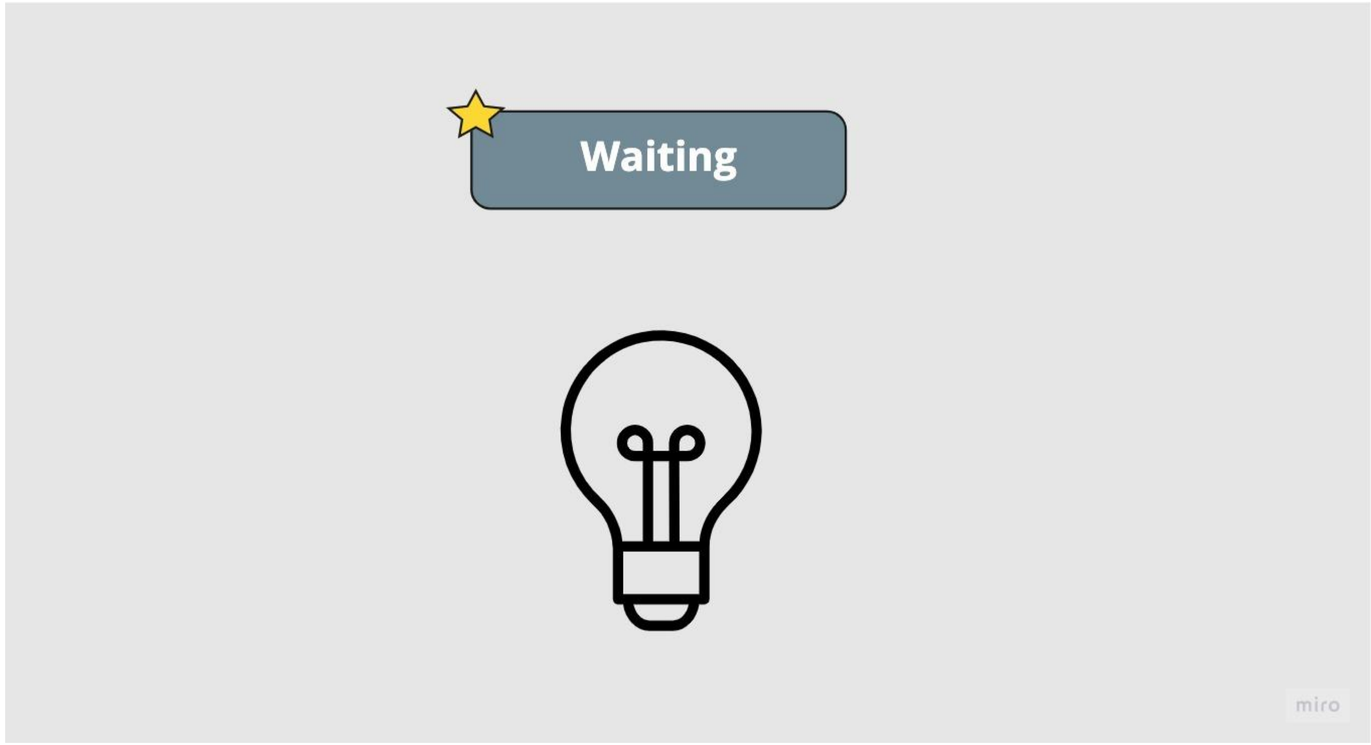
Transitions and events

How the light bulb goes between lit and unlit is through transitions. A transition is caused by an event that results in the change of state.



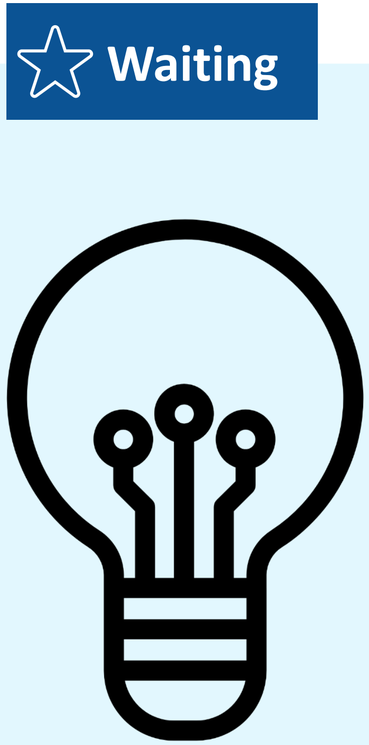
Initial state

Any process that has states will have an initial state, the default state the process exists in until an event happens to change the process's state.



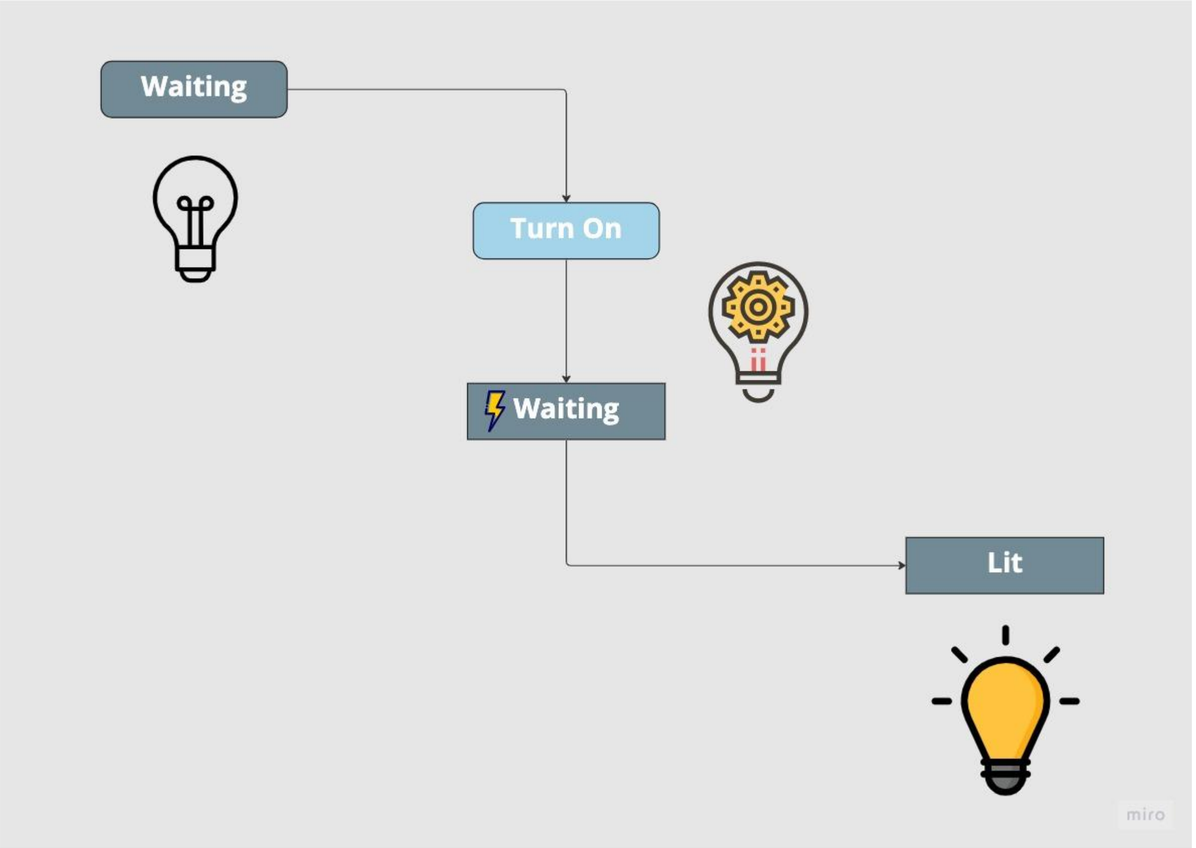
Initial state

Any process that has states will have an initial state, the default state the process exists in until an event happens to change the process's state.



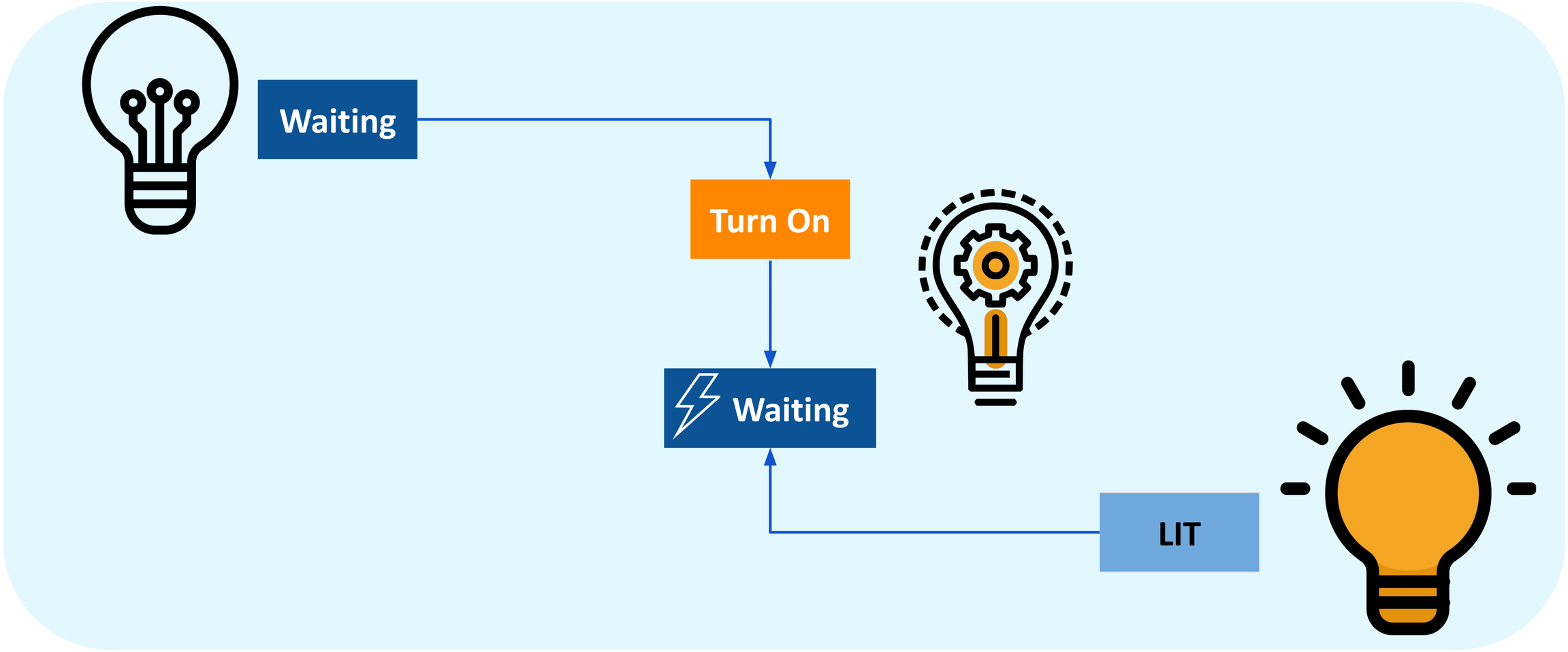
Final state

Most processes with states will have a final state, the last state when the process is finished. The final state is represented by a double border on the state's rounded rectangle box.



Final state

Most processes with states will have a final state, the last state when the process is finished. The final state is represented by a double border on the state's rounded rectangle box.



State Machine Tools

Tools to use finite state machines to model complex user interfaces

- **Figma**
  Figma includes a plugin called "States," which allows to define states and transitions for different components of your design.

- **QT Designer**
  QT Designer is a graphical tool for designing user interfaces in the QT framework. It includes support for designing and simulating finite state machines,

- **Xstate**
  XState is an open-source JavaScript library for creating and managing finite state machines which provides a declarative syntax for defining states and transitions.

- **Robot**
  Robot is a 1kb sized javascript library inspired by Xstate.

State Machine Tools

Tools to use finite state machines to model complex user interfaces

Figma includes a plugin called "States," which allows to define states and transitions for different components of your design.

QT Designer is a graphical tool for designing user interfaces in the QT framework. It includes support for designing and simulating finite state machines

XState is an open-source JavaScript library for creating and managing finite state machines which provides a declarative syntax for defining states and transitions

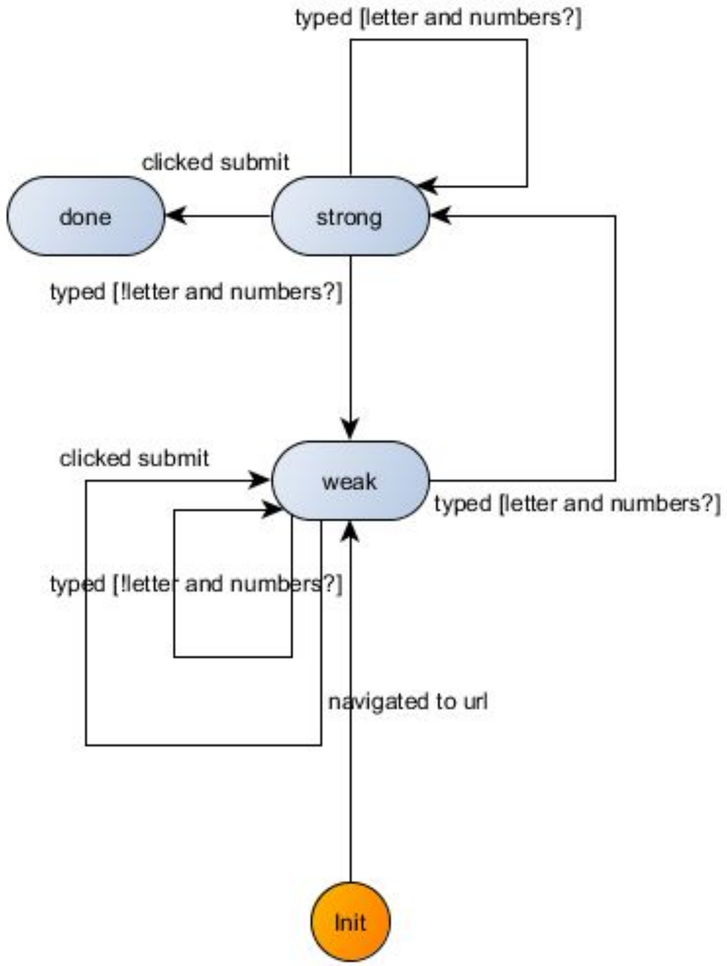
Robot is a 1kb sized javascript library inspired by Xstate

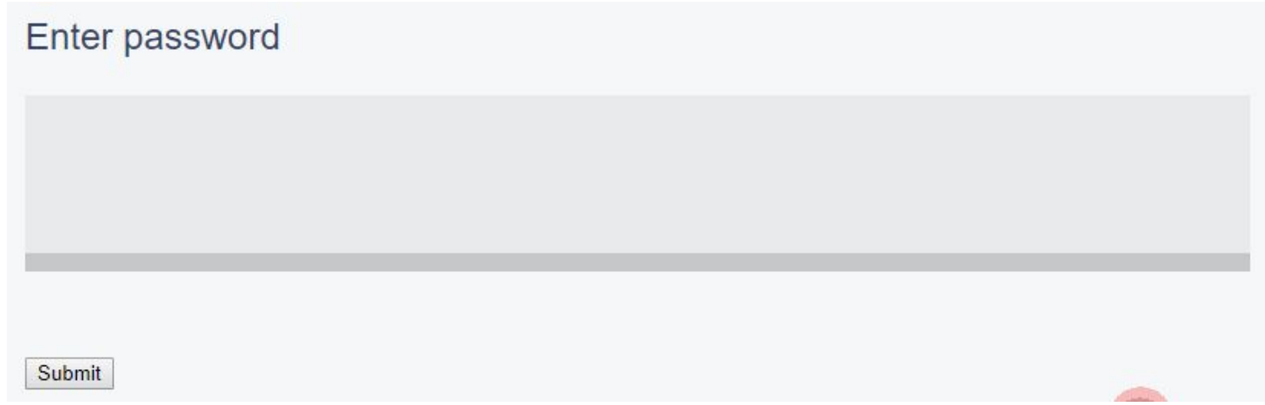
XState - Modeling user interfaces with Javascript

XState is an open-source JavaScript library for creating and managing finite state machines in web applications. It provides a declarative syntax for defining states and transitions, and includes features for hierarchical states, parallel states, and history states.

```
const fetchMachine = Machine( {
    id: 'light_bulb_machine',
    initial: 'unlit',
    states: {
      unlit: {
        on: {
          TURN_ON: {
            target: 'lit'
          }
        }
      },
      lit: {
        on: {
          TURN_OFF: {
            target: 'unlit'
          }
        }
      },
    },
});
```

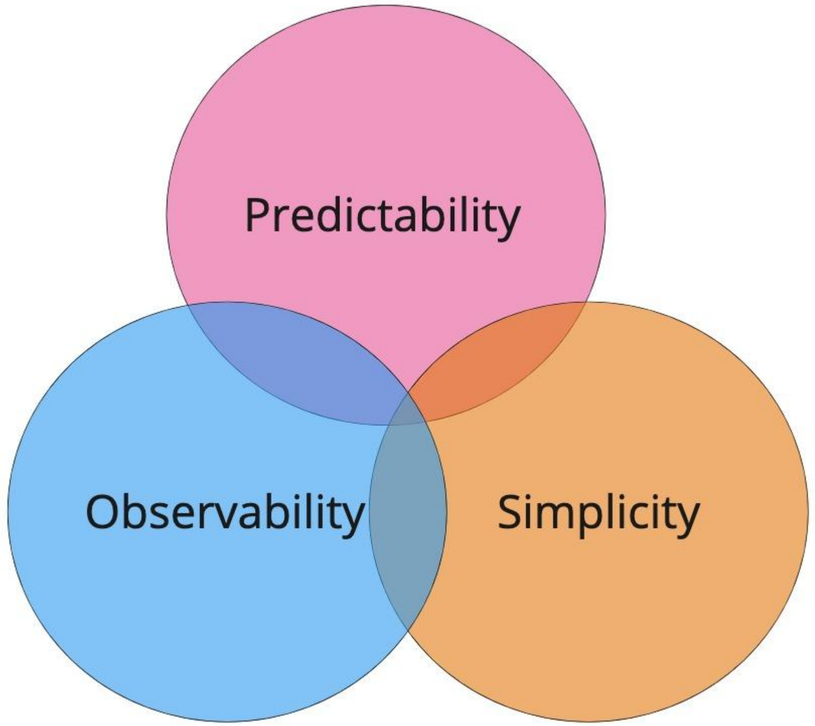
XState - Integrated into Web Application





Advantages of using State Machines

- **Predictability**
  - Find design bugs early
  - Reduce implementation bugs
  - Configuration driven Actions
  - Ease of Changes to States and Actions
- **Observability**
  - Have a clear and automatable piece of documentation of the interface for all members of the development team
  - Ease of Testing
- **Simplicity**
  - Iterate on features faster and more reliably
  - Visualization tools provide ease of design of complex states and actions

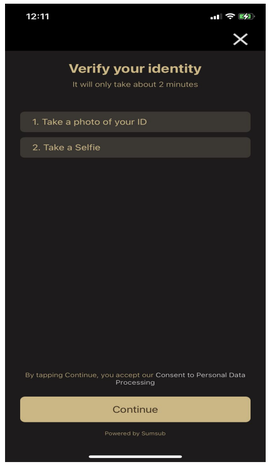


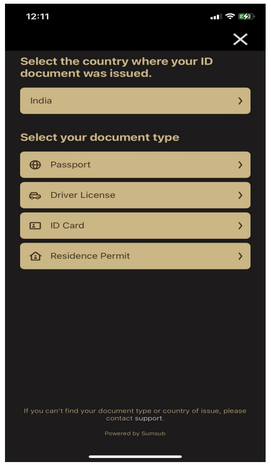
Accion
INNOVATION
SUMMIT 2023

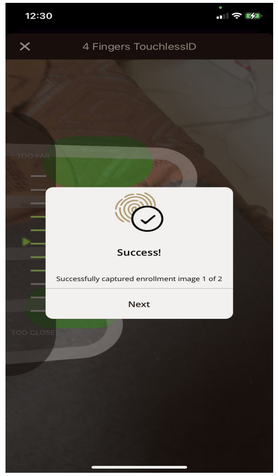
Accionlabs

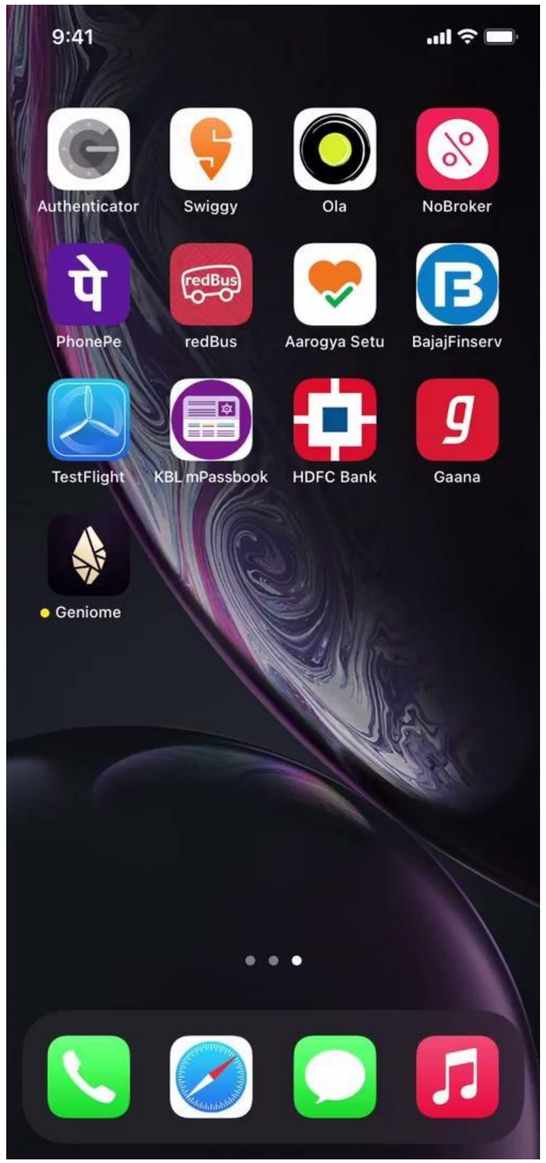
Examples

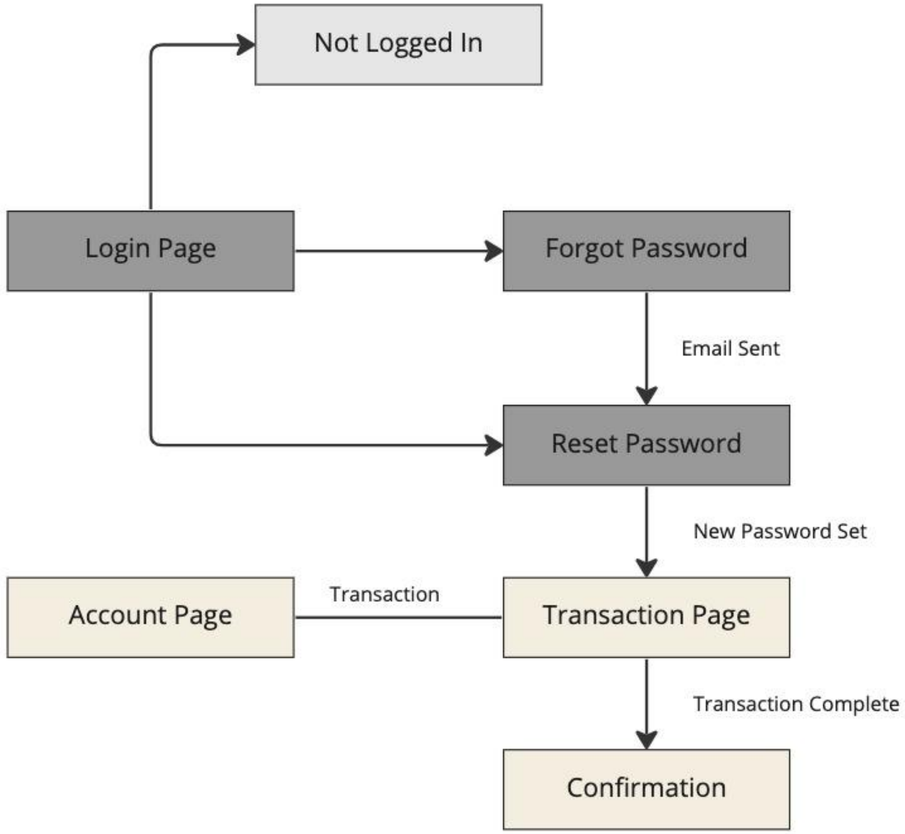
NextGen Wealth Management and Banking App



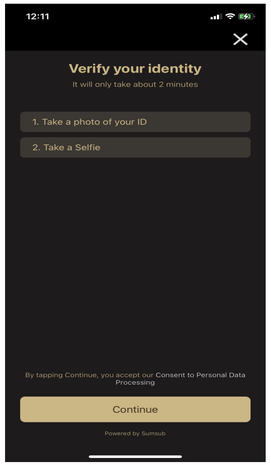


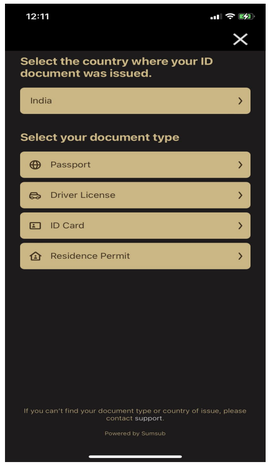


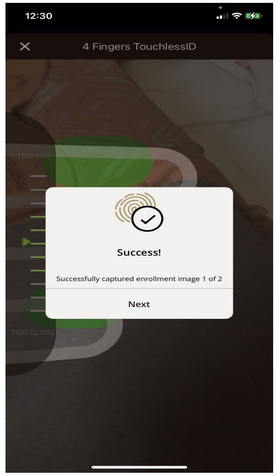


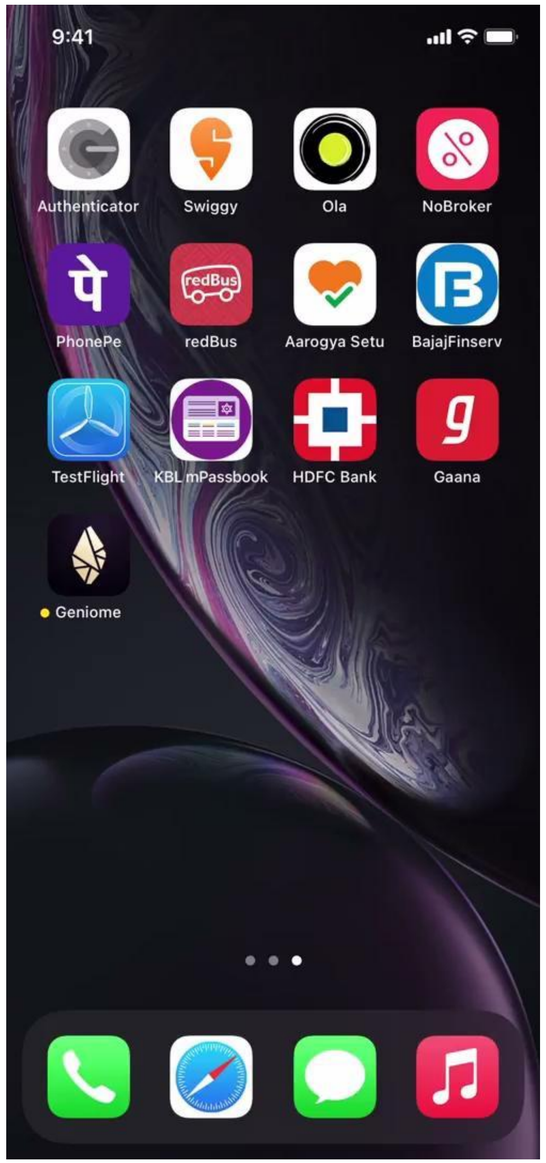


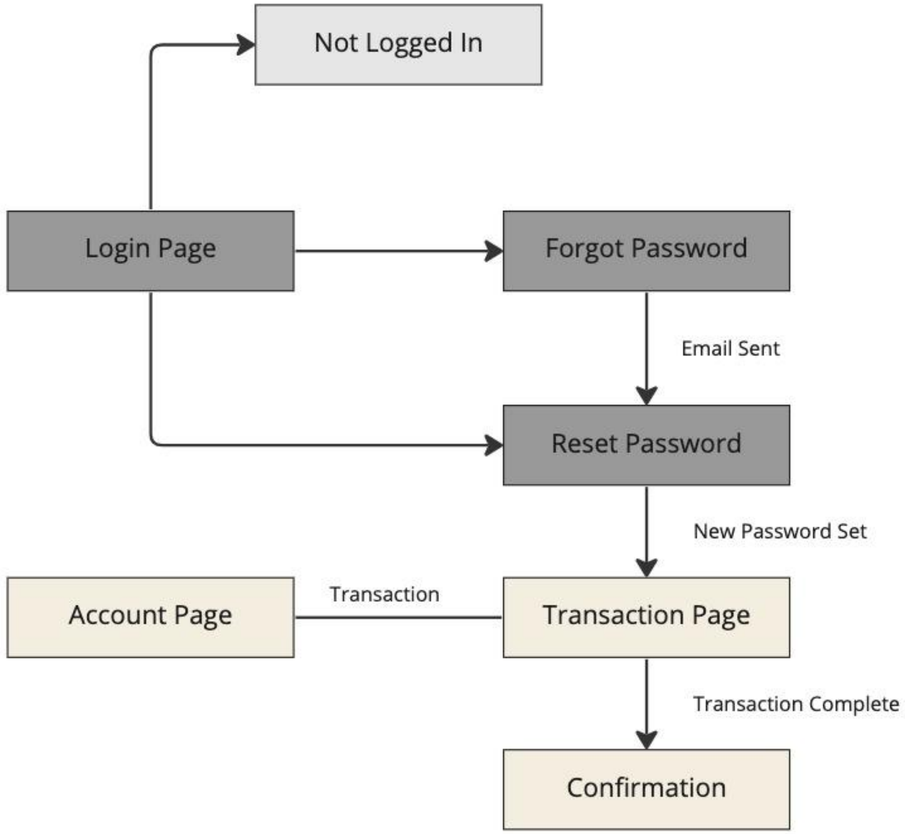
NextGen Wealth Management and Banking App











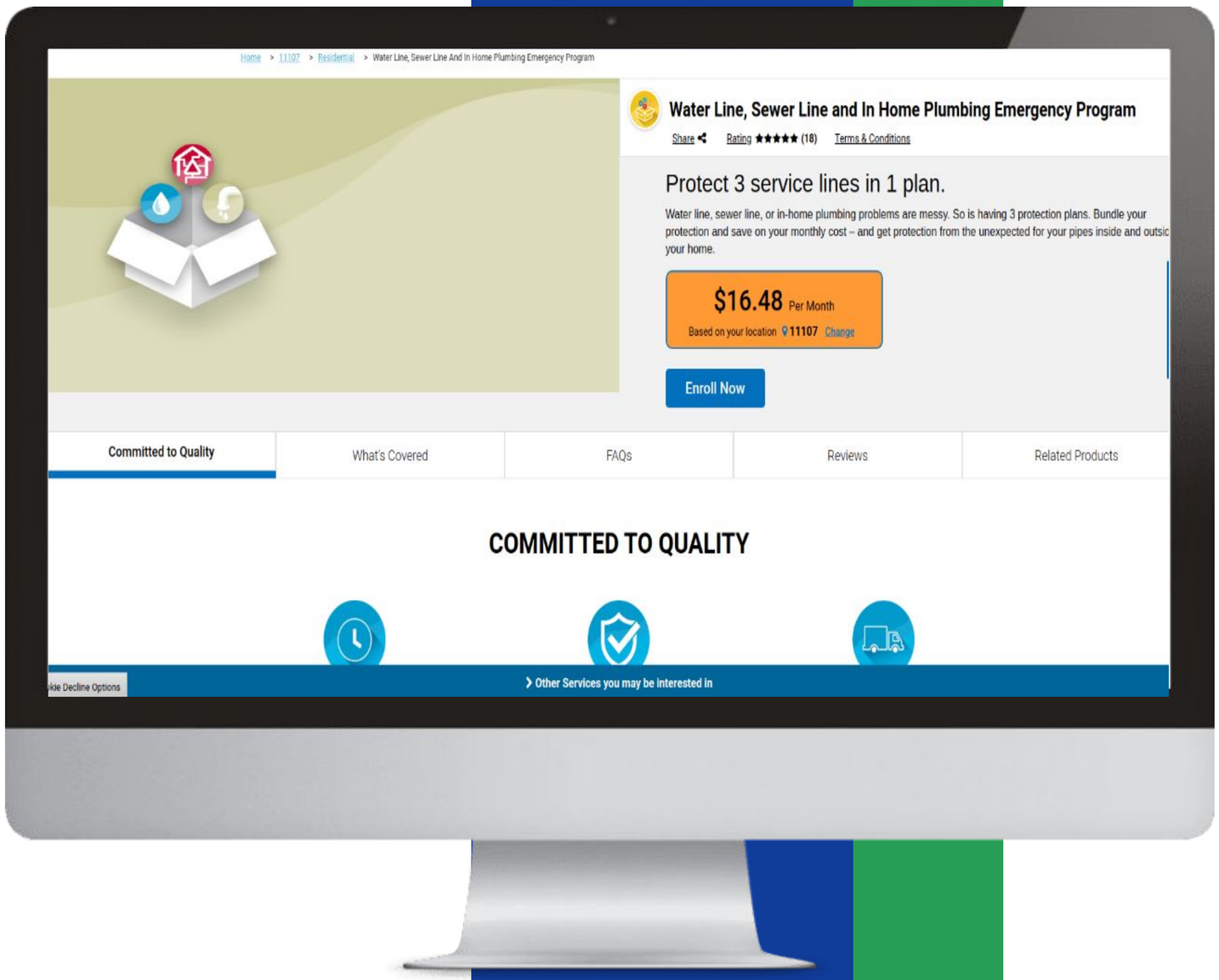
E-Commerce App

Large American public utility company that operates in the United States and Canada which provides safe, clean, affordable, and reliable water services to our customers to make sure we keep their lives flowing.

Here is how we used a finite state machine to model their checkout process:

- The initial state of the checkout process is the shopping cart. The user has the option to proceed to checkout, which triggers a transition to the "enter shipping information" state.

- In the "enter shipping information" state, the user can either enter their shipping information or go back to the shopping cart. If the user enters their information and clicks "continue," the state transitions to "enter payment information." If the user goes back, the state transitions back to the shopping cart.

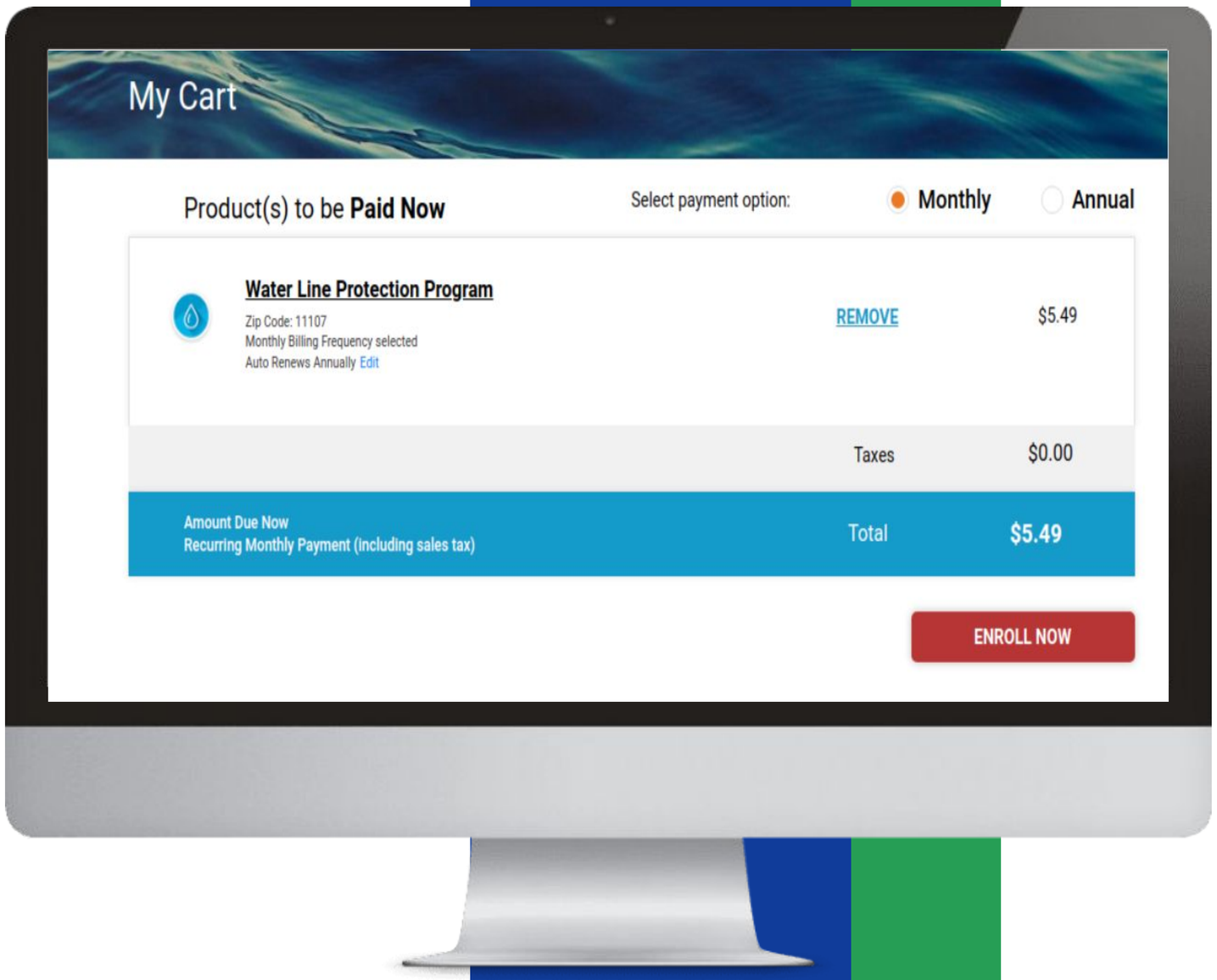


E-Commerce App

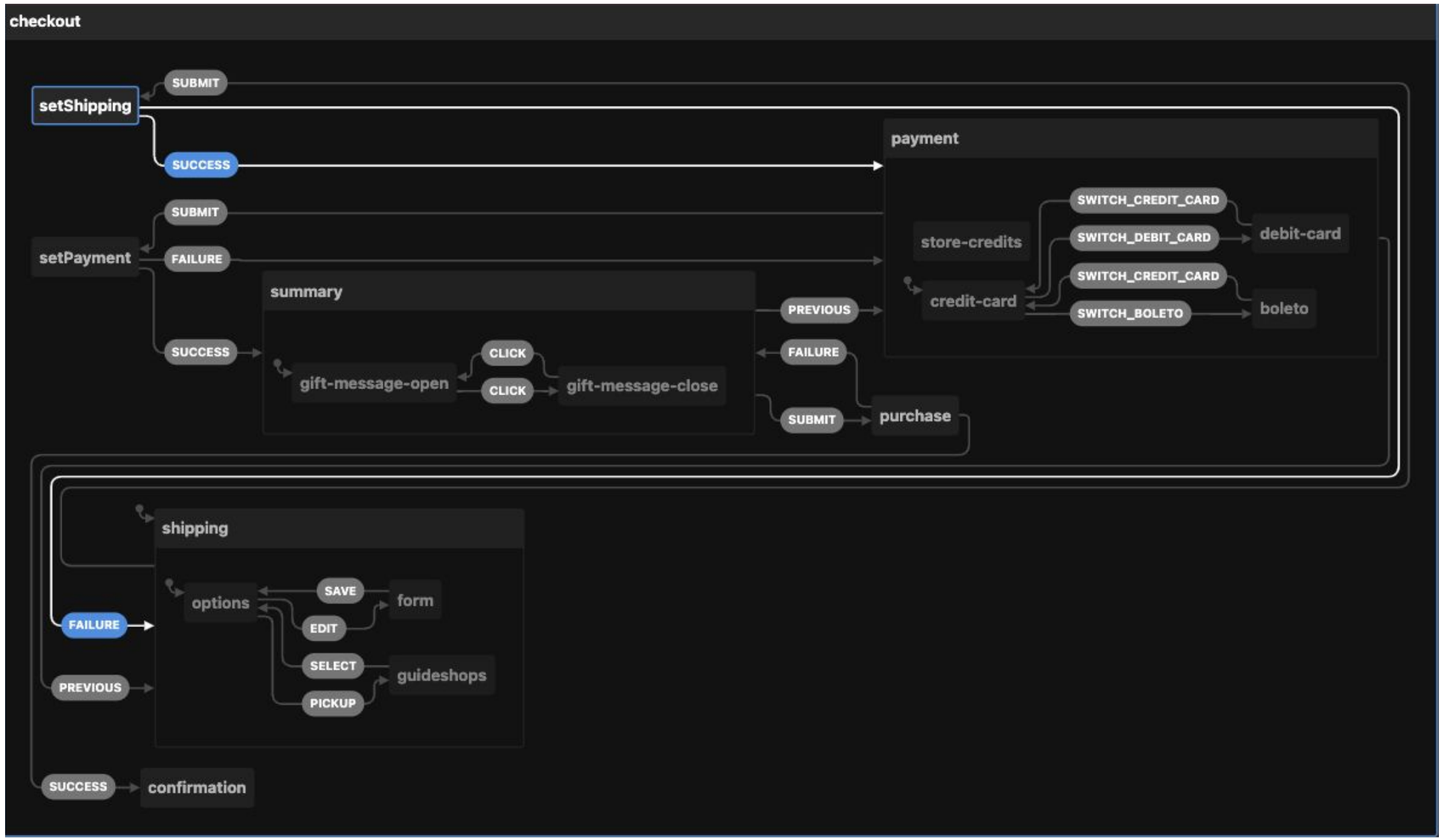
- In the "enter payment information" state, the user can enter their payment details or go back to the previous step. If the user enters their payment details and clicks "continue," the state transitions to the "review order" state. If the user goes back, the state transitions back to the "enter shipping information" state.

- In the "review order" state, the user can review their order details and either confirm the purchase or go back to the previous step. If the user confirms the purchase, the state transitions to the "order confirmation" state. If the user goes back, the state transitions back to the "enter payment information" state.

- In the "order confirmation" state, the user sees a confirmation message and has the option to start a new order or return to the homepage.



E-Commerce App - Checkout State Machine



Accion
INNOVATION
SUMMIT 2023

Accionlabs

Summary

- Frontend evolution and the key problem areas
- Conventional approaches to tackle the problems.
- Innovative approach to solve the problems.
- Use cases and application areas.

Accion
INNOVATION
SUMMIT 2023

Accionlabs

Demo

Accionlabs

Accion
INNOVATION
SUMMIT 2023

Thank You!

vaibhav.satam@accionlabs.com

kejal.shah@accionlabs.com