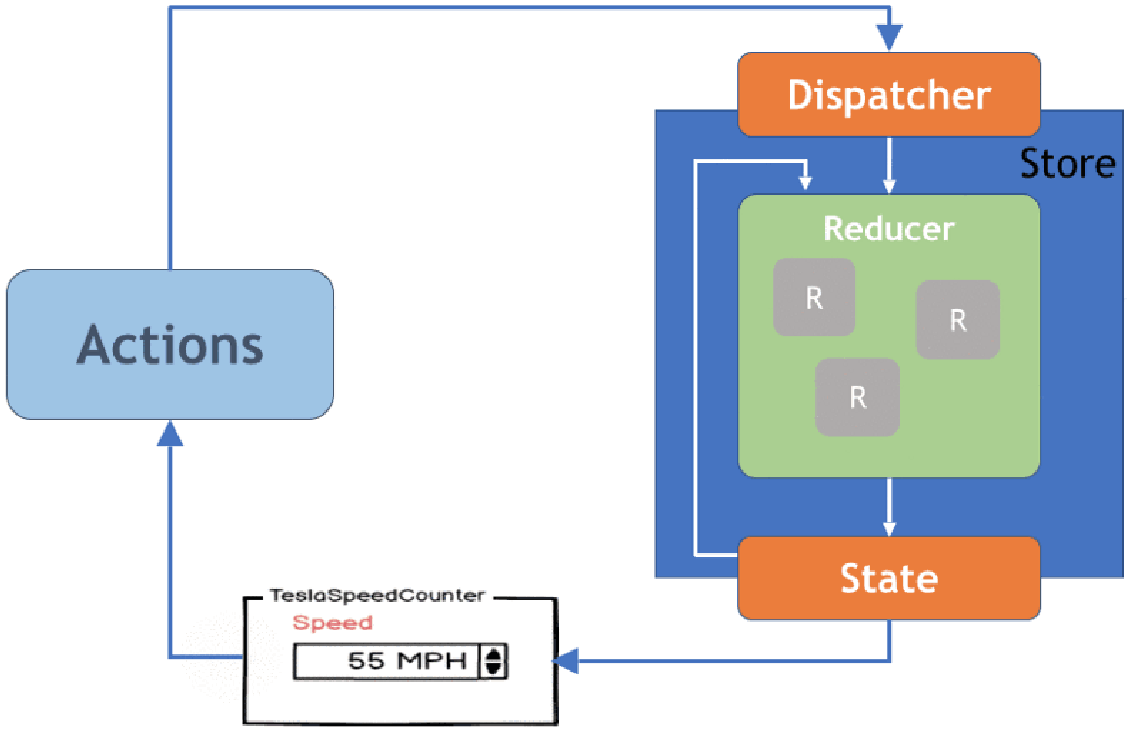
2.13 How they all work together

Finally, all the puzzle pieces were completed. Now let’s look at the following animation as an example of when all the puzzle pieces are tied together and the user triggers the **speed up** event.



Now run **npm start** and it will be compiled normally and the application will be started.

But there are still a few things to do.

- First, copy all the contents of /**containers/TeslaBattery.css** that you created in part 1 and add them to **App.css**.
- Next, open /**components/TeslaCounter/TeslaCounter.js** and modify the **onClick** event handler as follows: This is because part 2 no longer handles event handling in **TeslaBattery.js**.

```
onClick={(e) => props.increment(e, props.initValues.title)}
-->
```

```
onClick={(e) => {
  e.preventDefault();
  props.increment(props.currentValue)}}

onClick={(e) => props.decrement(e, props.initValues.title)}
-->
onClick={(e) => {
  e.preventDefault();
  props.decrement(props.currentValue)}}
```

- Next, let’s not use props repeatedly by using ES6 Object destructuring.

```
const TeslaCounter = (props) => (
  <p className="tesla-counter__title">{props.initValues.title}</p>
  ...
-->
const TeslaCounter = ({ initValues, currentValue, increment, decrement }) => (
  <p className="tesla-counter__title">{initValues.title}</p>
  ...
```

Finally, our **Redux version of Tesla Battery Range Calculator app** is complete!

```
import { getModelData } from '../services/BatteryService';

const initialState = {
  carstats: [
    { miles: 246, model: "60" },
    { miles: 250, model: "60D" },
    { miles: 297, model: "75" },
    { miles: 306, model: "75D" },
    { miles: 336, model: "90D" },
    { miles: 376, model: "P100D" }
  ],
  config: {
    speed: 55,
    temperature: 20,
    climate: true,
    wheels: 19
  }
}

function updateStats(state, newState) {
  return {
    ...state,
    config: newState.config,
    carstats: calculateStats(newState)
  }
}

function calculateStats(state) {
  const models = ['60', '60D', '75', '75D', '90D', 'P100D'];
  const dataModels = getModelData();
  return models.map(model => {
    const { speed, temperature, climate, wheels } = state.config;
    const miles = dataModels[model][wheels][climate ? 'on' : 'off'].speed[speed][temperature];
    return {
      model,
      miles
    };
  });
}

function appReducer(state = initialState, action) {
  switch (action.type) {
    case 'CHANGE_CLIMATE': {
      const newState = {
        ...state,
        config: {
          climate: !state.config.climate,
          speed: state.config.speed,
          temperature: state.config.temperature,
          wheels: state.config.wheels
        }
      };
      return updateStats(state, newState);
    }
    case 'SPEED_UP': {
      const newState = {
          ...state,
          config: {
            climate:state.config.climate,
            speed:action.value + action.step,
            temperature:state.config.temperature,
            wheels:state.config.wheels
          }
      };
      return updateStats(state, newState);
    }
    case 'SPEED_DOWN': {
      const newState = {
          ...state,
          config: {
            climate:state.config.climate,
            speed:action.value - action.step,
            temperature:state.config.temperature,
            wheels:state.config.wheels
          }
      };
      return updateStats(state, newState);
    }
    case 'TEMPERATURE_UP': {
      const newState = {
          ...state,
          config: {
            climate:state.config.climate,
            speed:state.config.speed,
            temperature:action.value + action.step,
            wheels:state.config.wheels
          }
      };
      return updateStats(state, newState);
    }
    case 'TEMPERATURE_DOWN': {
      const newState = {
          ...state,
          config: {
            climate:state.config.climate,
            speed:state.config.speed,
            temperature:action.value - action.step,
            wheels:state.config.wheels
          }
          }
      };
      return updateStats(state, newState);
    }
    case 'CHANGE_WHEEL': {
      const newState = {
          ...state,
          config: {
            climate:state.config.climate,
            speed:state.config.speed,
            temperature:state.config.temperature,
            wheels:action.value
          }
      };
      return updateStats(state, newState);
    }
    default:
      return state
  }
}

export default appReducer;
```