

ANGULAR & REDUX

with angular-redux

ABOUT:ME

Jean-Christophe Puja (JC)

Alsace 

@bluesky

8+ years of fullstack

2 projects in prod with Angular + Redux

WHAT WE'LL SEE

Why Redux?

Redux building blocks

Redux + Angular

Managing side-effects with Epics

WHY REDUX?

Your app has state.

Fahrzeug Aussenansicht Innenansicht Autopilot Bezahlung +



Farbe wählen

Red Multi-Coat CHF 2'100

Felgen wählen

19-Zoll Sport-Felgen CHF 1'600

Winterreifen hinzufügen CHF 2'150

Anhängerkupplung

Mit der Anhängerkupplung aus hochfestem Stahl mit abnehmbarem Kugelkopf können Sie bis zu 910 kg Anhängelast ziehen. Ein wichtiges Merkmal dieses Pakets ist die spezielle Software, die Schlingern des Gespanns durch Ansteuern einzelner Radbremsen unterdrückt.

Option wählen CHF 1'080

BAR ▾

CHF 63'590 CHF 52'790

Vor Einsparungen Nach Einsparungen

Einzelheiten anzeigen

Voraussichtliche Auslieferung: Mai

WEITER

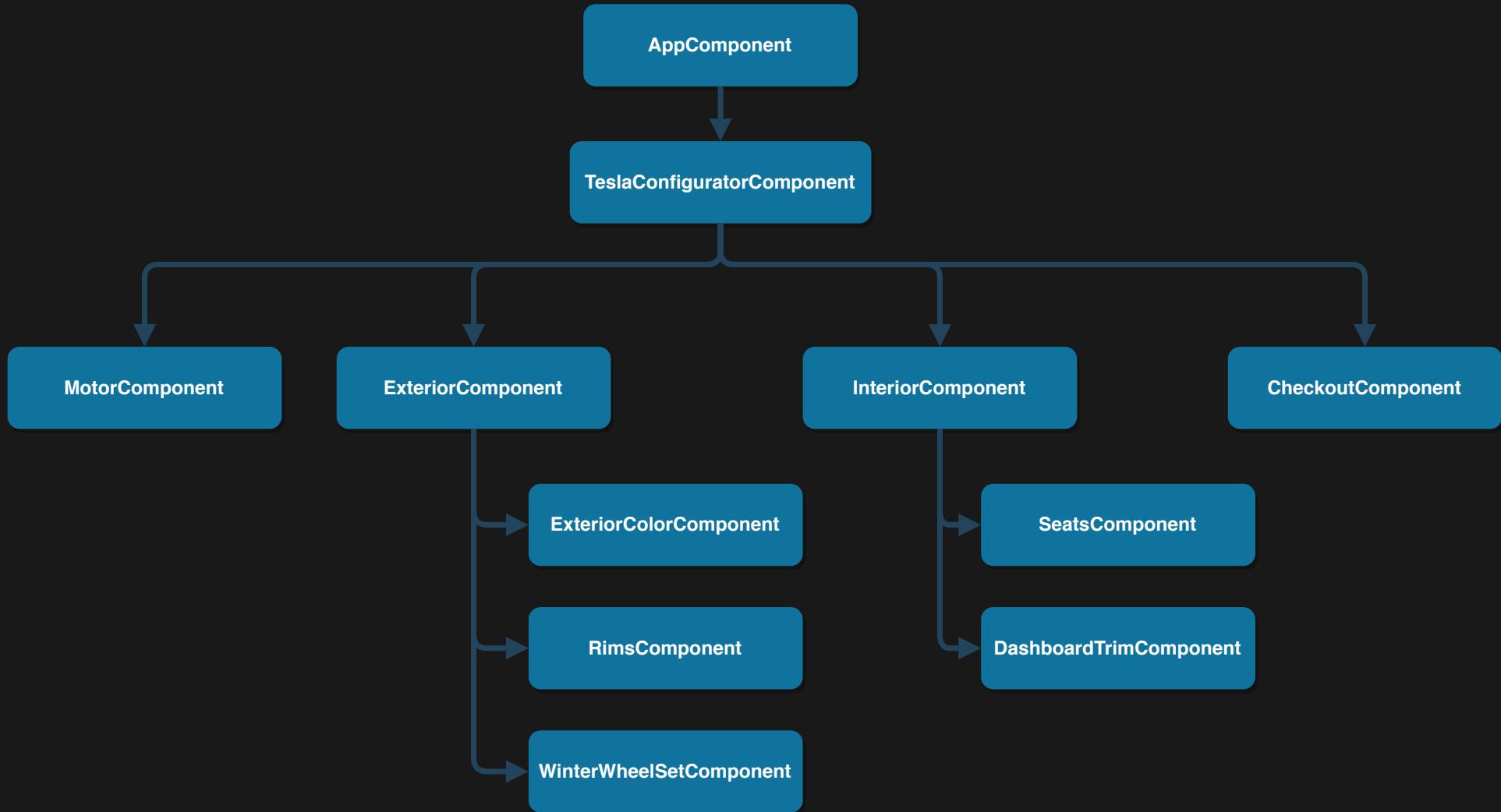
State is usually shared...

The screenshot illustrates a user interface for configuring a Tesla Model 3. The top navigation bar includes tabs for Fahrzeug, Aussenansicht (selected), Innenansicht, Autopilot, Bezahlung, and a plus icon. The main area shows a red Tesla Model 3 from a side-front perspective, enclosed in a red oval. To the right, configuration options and their costs are listed:

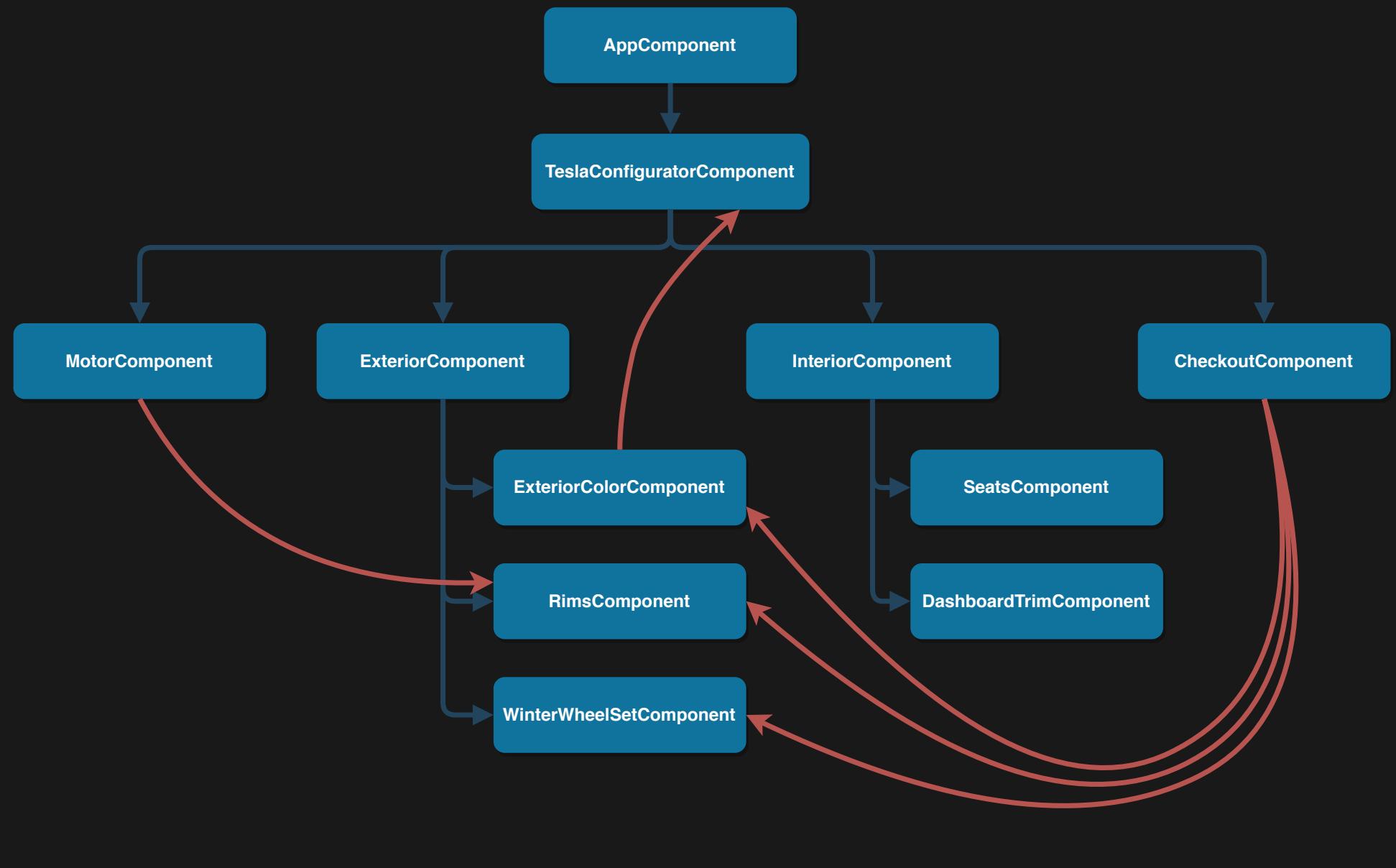
- Farbe wählen**: Red Multi-Coat (selected, highlighted with a blue border) costs CHF 2'100.
- Felgen wählen**: 19-Zoll Sport-Felgen (selected) costs CHF 1'600. An option to add winter tires for CHF 2'150 is available.
- Anhängerkupplung**: Described as a package with a steel hitch ball and software for up to 910 kg towing capacity.
- Option wählen**: An option costing CHF 1'080.

At the bottom, a summary shows the total price: **CHF 63'590** (Vor Einsparungen) and **CHF 52'790** (Nach Einsparungen). A link to "Einzelheiten anzeigen" is provided. The delivery date is listed as "Voraussichtliche Auslieferung: Mai". A "WEITER" button is at the bottom right. The bottom left shows a dropdown menu for payment method.

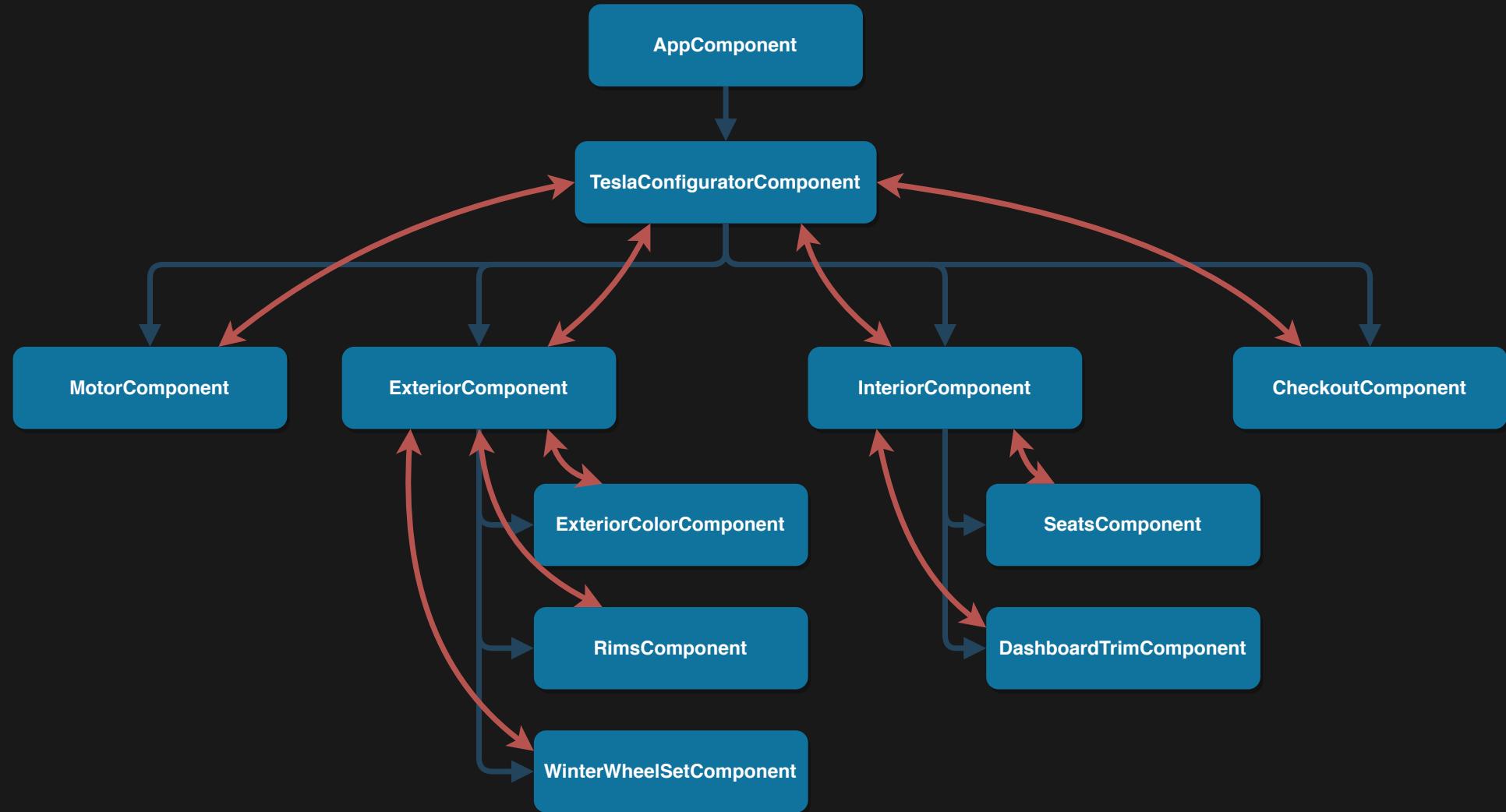
...across the app.



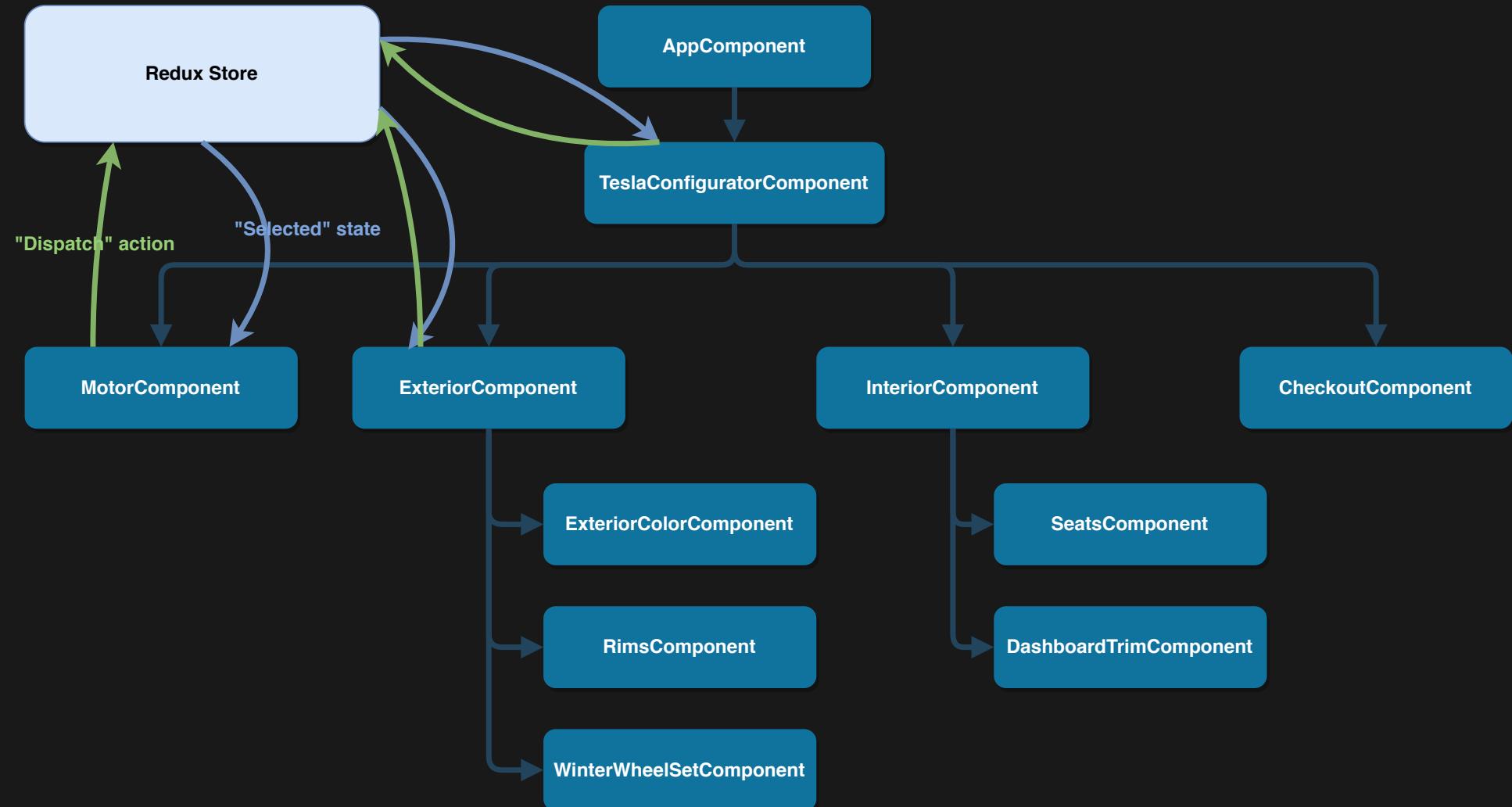
Ad-hoc?



Bucket brigade?



The Redux idea



REDUX BUILDING BLOCKS

State

Actions

"Reducer" functions

State

```
const state = {
  colors: {
    'red': {
      price: 2100.00,
      name: 'Red Multi-Coat'
    },
    'blue': {
      price: 1500.00,
      name: 'Midnight Blue'
    }
  },
  selectedColor: 'red',
  winterWheelsSelected: false,
  trailerHitchSelected: false
};
```

Accessed via the Redux store

Immutable & normalized

Actions

```
const colorSelectedAction = {  
  type: 'COLOR_SELECTED',  
  payload: {  
    color: 'blue'  
  }  
};
```

Type, maybe Payload

Dispatched to the Redux store

Reducer functions

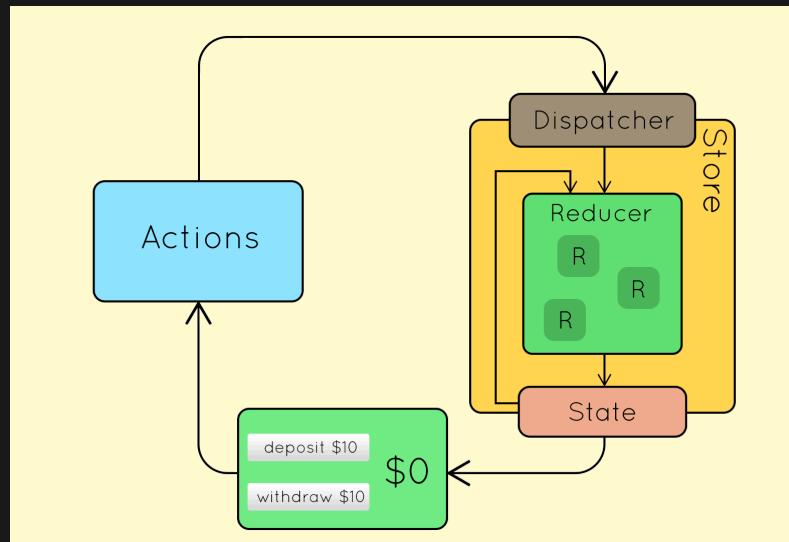
```
function reducer(state, action) {
  switch (action.type) {
    case 'COLOR_SELECTED':
      return {
        ...state,
        selectedColor: action.payload.color
      };

    // Other actions...
  }
}
```

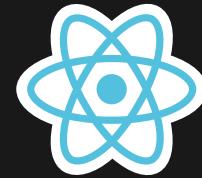
Pure functions (no side-effects)

Executed by the Redux Store

Recap



REDUX + ANGULAR



Created in the [React.js](#) world

Framework agnostic

Angular connector: [angular-redux](#)



Angular implementation: [ngrx/store](#)

👎 Not Redux, own implementation

👍 Fairly popular

Give it a try!

DEMO



MANAGING SIDE-EFFECTS WITH EPICS

Async actions

```
function asyncAction() {  
    ngRedux.dispatch(fetchStart());  
    http.get('http://url/foobar')  
        .subscribe(  
            result => ngRedux.dispatch(fetchSuccess(result)),  
            error => ngRedux.dispatch(fetchError(error))  
        );  
}
```

Hard to test

Don't scale well

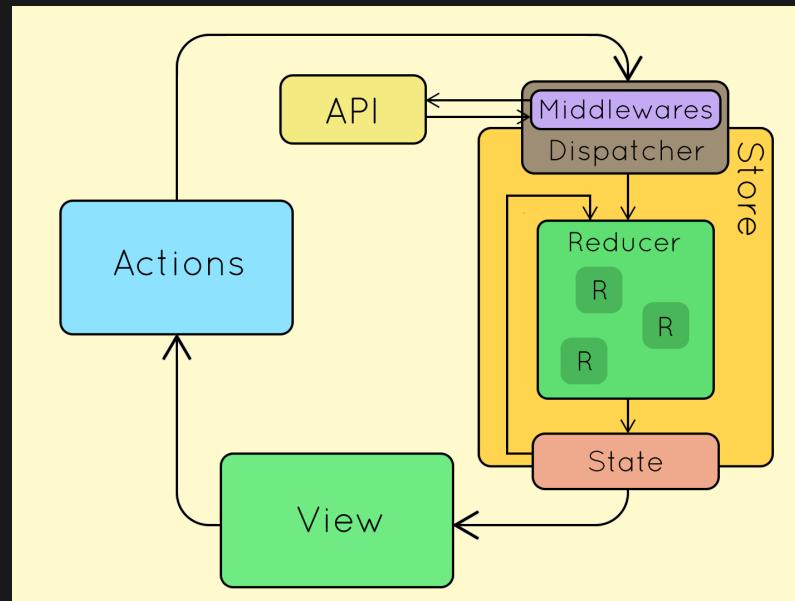
Hard to combine

"Epics": Actions in, actions out.

```
const pingEpic = action$ => action$.pipe(  
  filter(action => action.type === 'PING'),  
  delay(1000), // Simulation of async work  
  mapTo({ type: 'PONG' })  
);
```

Declarative
Isolate side-effects
Use the power of Observables

Data flow with Epics



DEMO (AGAIN)



REFERENCES

Analysis of state management in Angular: <https://blog.oasisdigital.com/2016/managing-state-angular-2/>

Redux docs & tutorial: <https://redux.js.org/docs/introduction/>

Flux Standard Actions, a good convention for Action structure: <https://github.com/redux-utilities/flux-standard-action>

angular-redux tutorial: <https://github.com/angular-redux/store/blob/master/articles/intro-tutorial.md>

Redux-observable: <https://redux-observable.js.org/>

Data flow diagrams: <https://github.com/reactjs/redux/issues/653>