REDUX

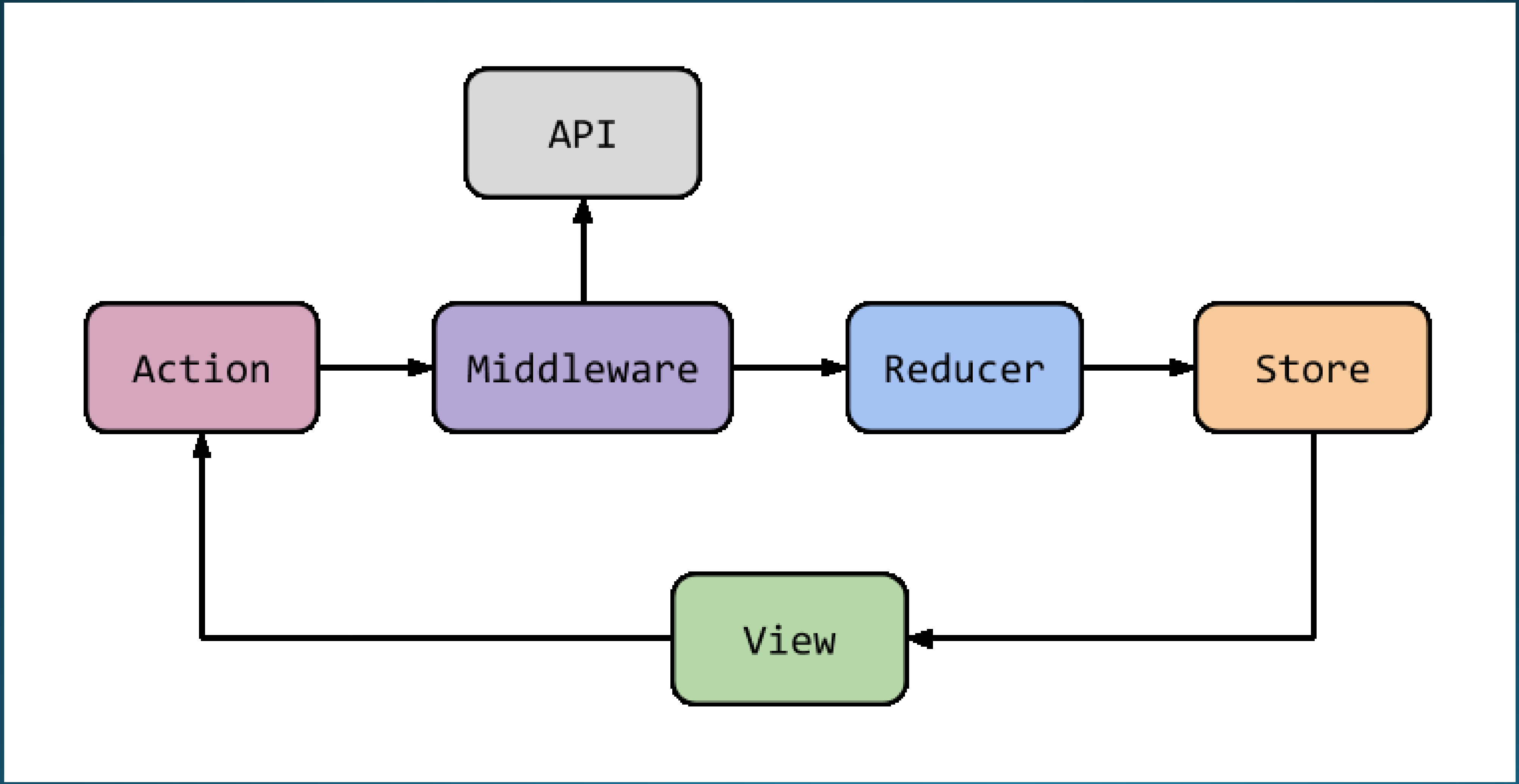
Three Principles Of Redux

- Single source of truth.
- The state is read-only.
- Changes are made with pure functions.

Terminology

- **STORE :** In charge for orchestrating all the interactions in Redux
- **REDUCER:** JavaScript function taking two parameters: state and action.
- **ACTION :** plain JavaScript objects with a property named type
- **MIDDLEWARE :** abstracting away business logic

Architecture



Functions

- **getState**: accessing the current state of the application

```
store.getState()
```

- **Dispatch :** dispatching an action

```
 store.dispatch(
         addBook({ title: ‘Book1', id: 1 })
                  )
```

- **Subscribe** : listening on state changes

```
store.subscribe(() => console.log(‘Hello World'))
```

React-Redux

- **Provider**

  Makes store available to components wrapped in connect function.
- **Connect**

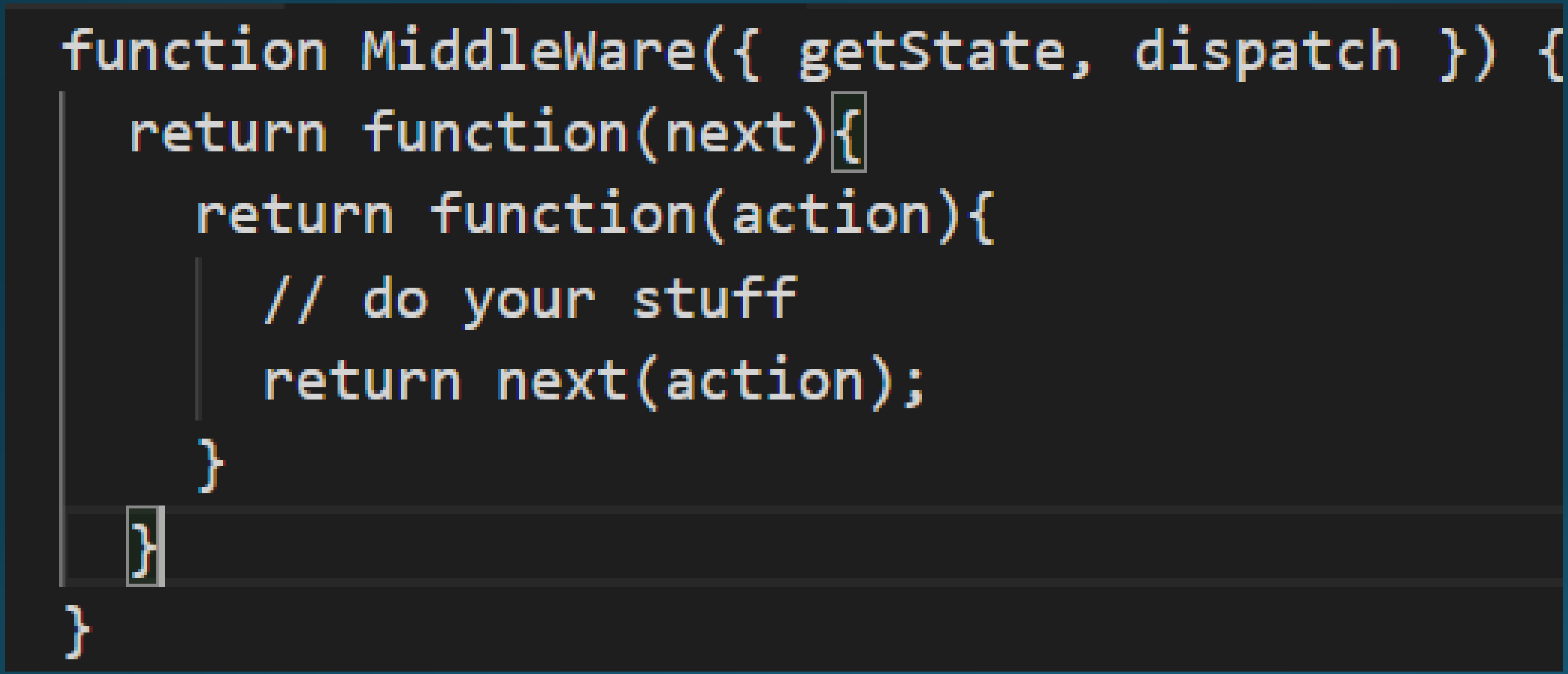
  Connects a React component to a Redux store.
- **mapStateToProps**

  Connects a part of the Redux state to the props of a React component
- **mapDispatchToProps**

  Connects Redux actions to React props.
- **combineReduers(reducers)**

  Combine reducers that are separated into functions, each managing independent parts of the state.

Middleware



```
function MiddleWare({ getState, dispatch }) {
  return function(next){
    return function(action){
      // do your stuff
      return next(action);
    }
  }
}
```