redux createStore

@Migo F2E Duncan

2016 / 04 / 14

Data Flow

view

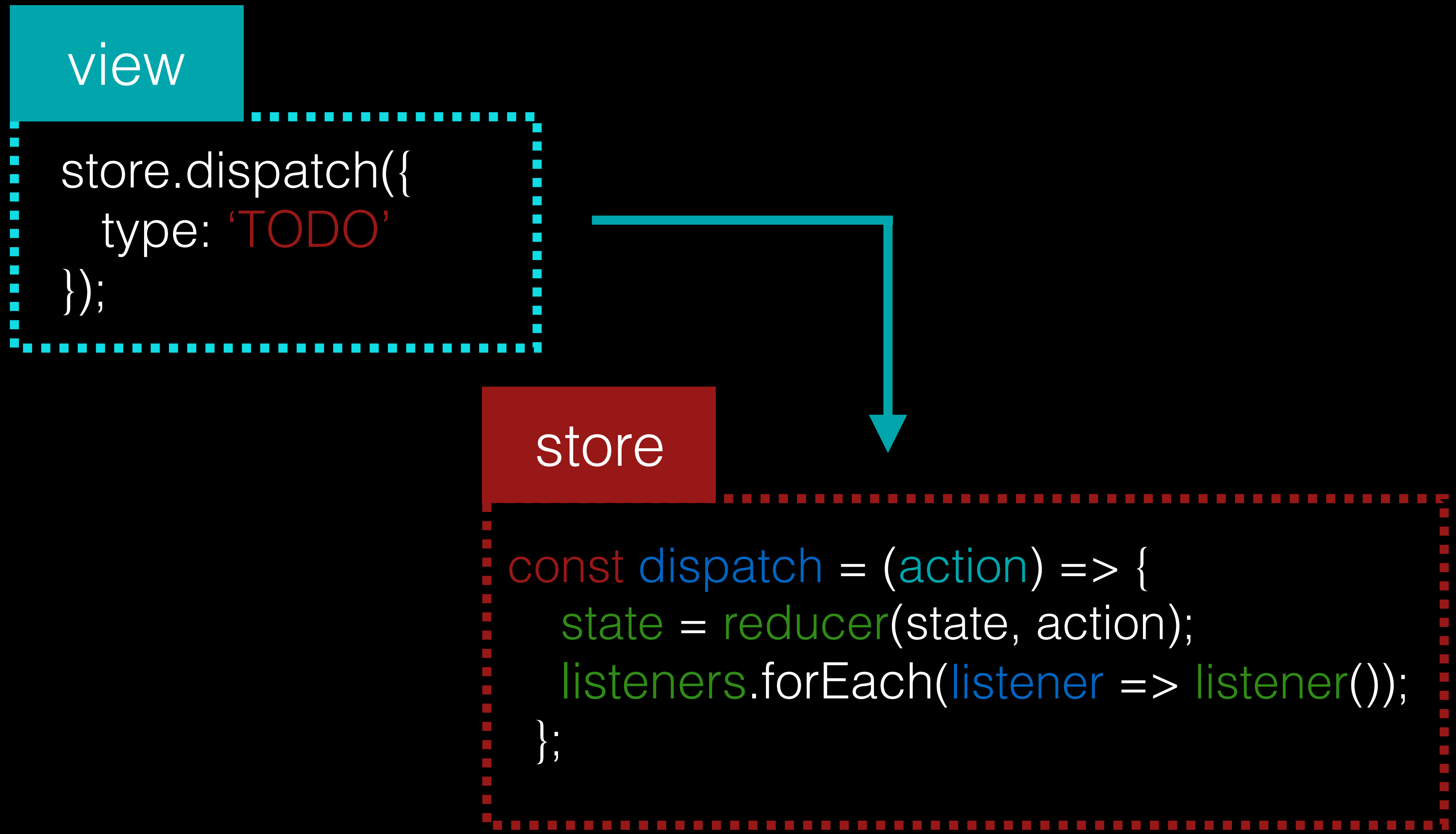
```
store.dispatch({
   type: ‘TODO’
});
```

store

```
const dispatch = (action) => {
    state = reducer(state, action);
    listeners.forEach(listener => listener());
  };
```

reducer

```
const reducer = (state = 0, action) => {
  switch (action.type) {
    case 'TODO':
      return state + 1;
  }
};
```

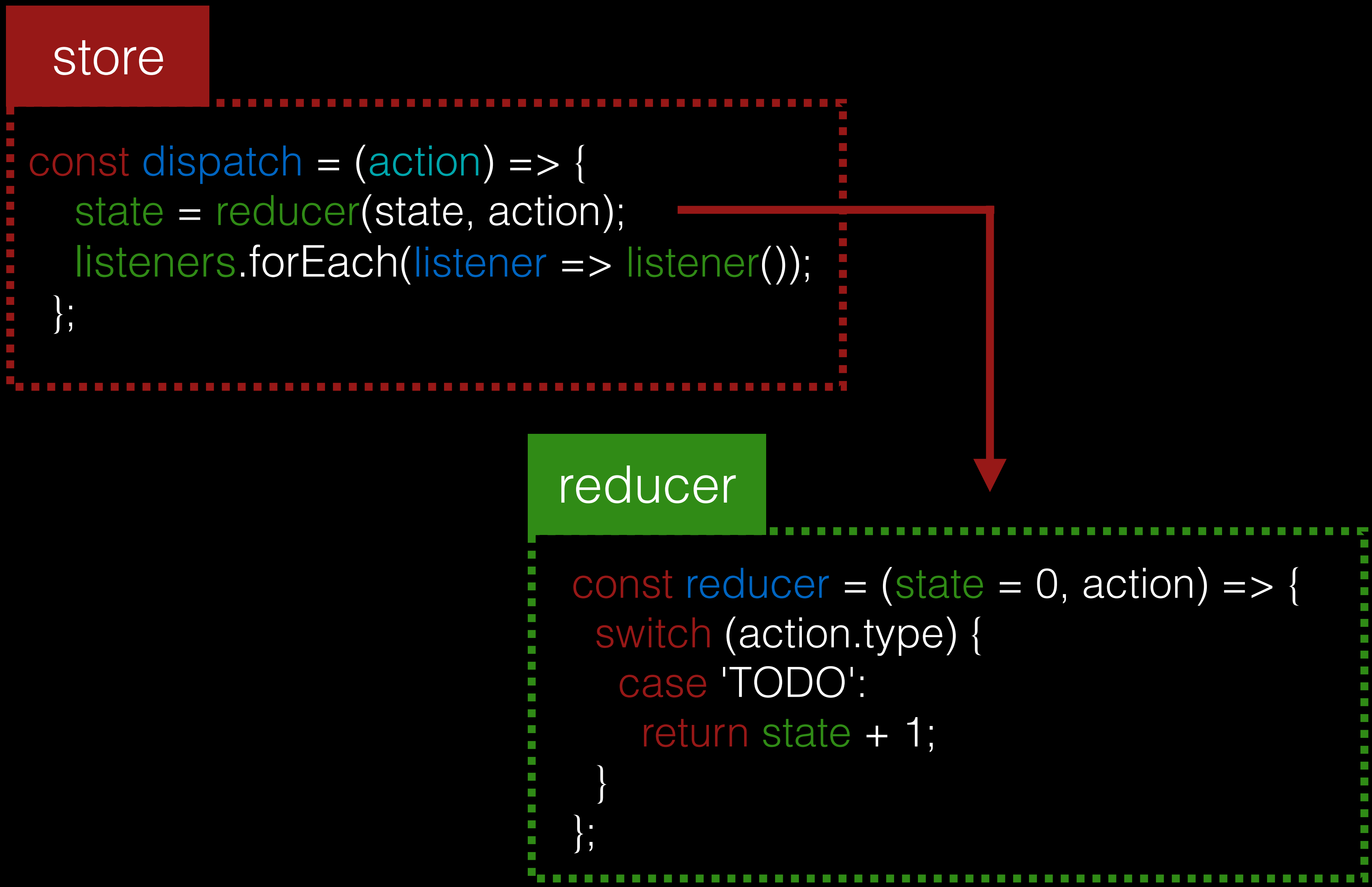


view

```
store.dispatch({
  type: ‘TODO’
});
```

store

```
const dispatch = (action) => {
    state = reducer(state, action);
    listeners.forEach(listener => listener());
  };
```

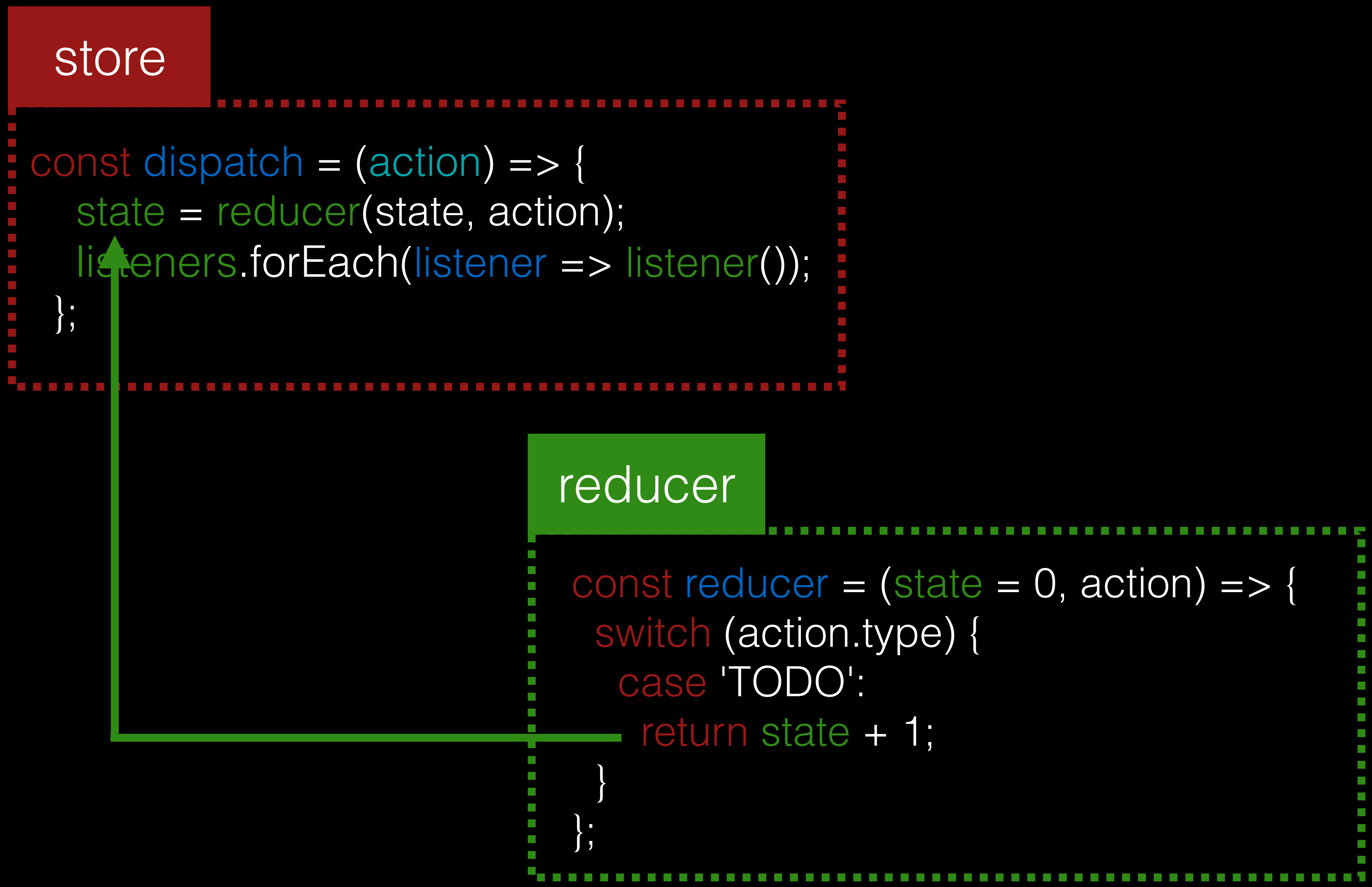


store

```
const dispatch = (action) => {
    state = reducer(state, action);
    listeners.forEach(listener => listener());
  };
```

reducer

```
const reducer = (state = 0, action) => {
  switch (action.type) {
    case 'TODO':
      return state + 1;
  }
};
```

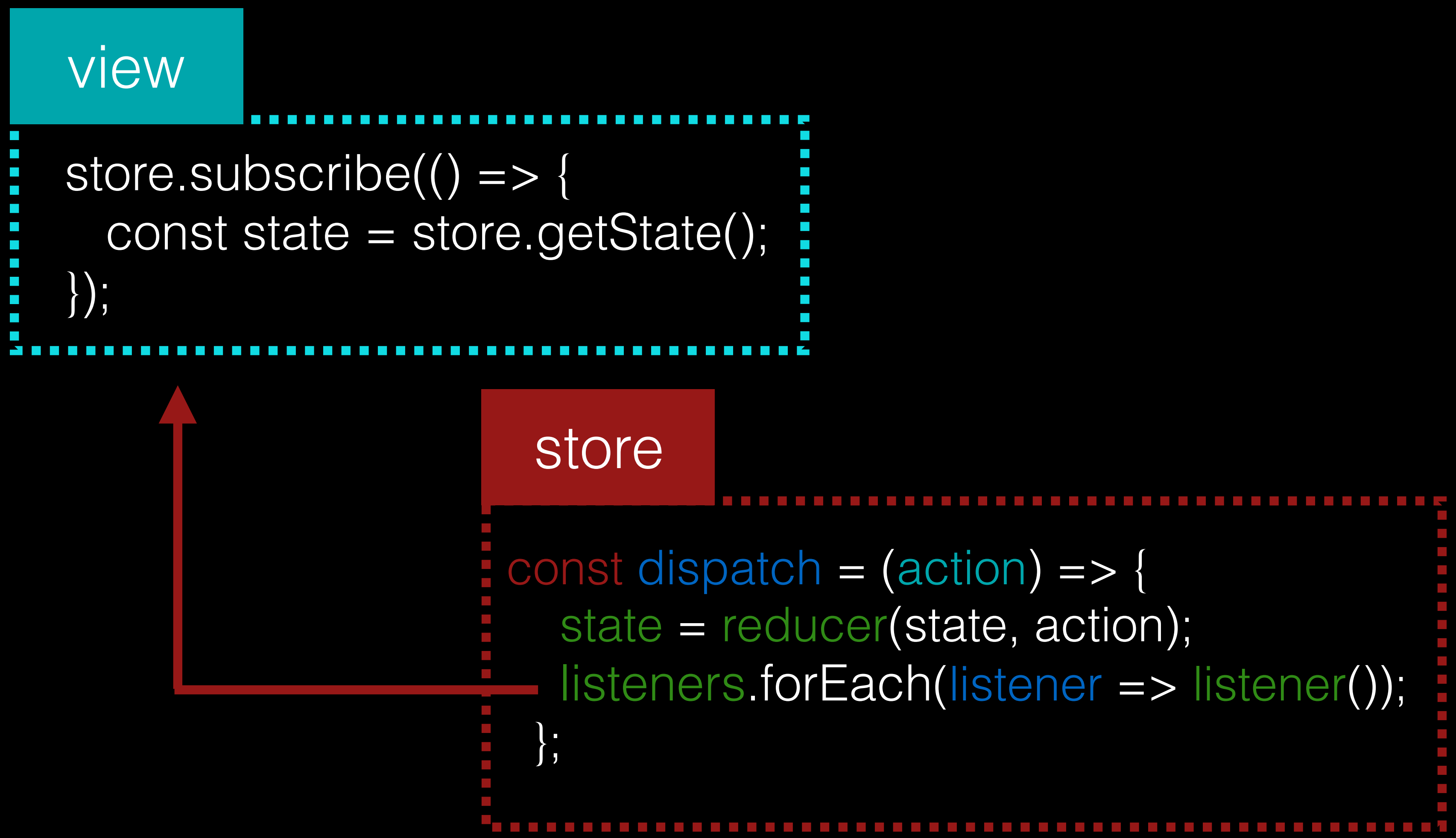


store

```
const dispatch = (action) => {
   state = reducer(state, action);
   listeners.forEach(listener => listener());
 };
```

reducer

```
const reducer = (state = 0, action) => {
 switch (action.type) {
   case 'TODO':
     return state + 1;
 }
};
```

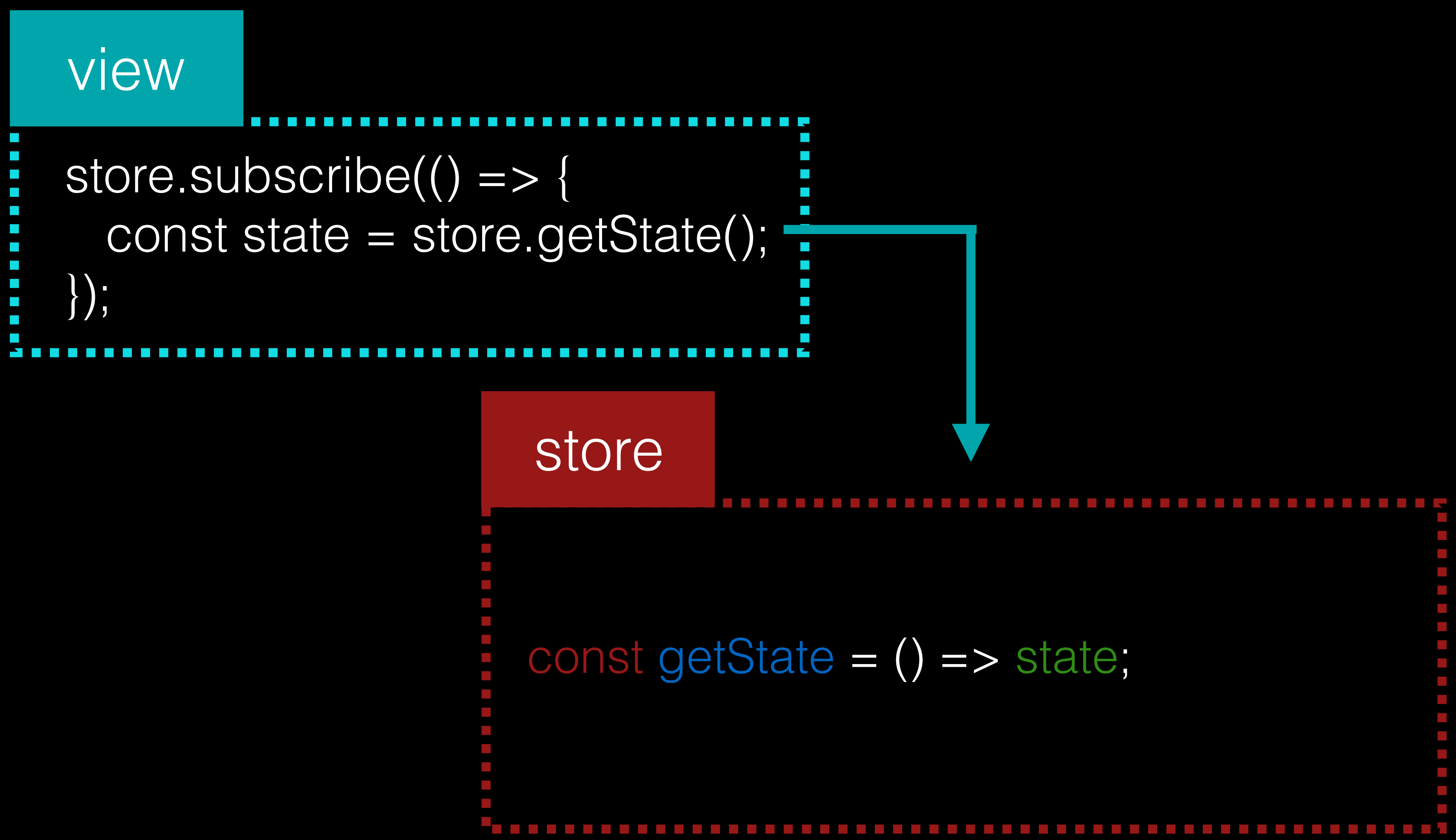


view

```
store.subscribe(() => {
   const state = store.getState();
});
```

store

```
const dispatch = (action) => {
   state = reducer(state, action);
   listeners.forEach(listener => listener());
 };
```

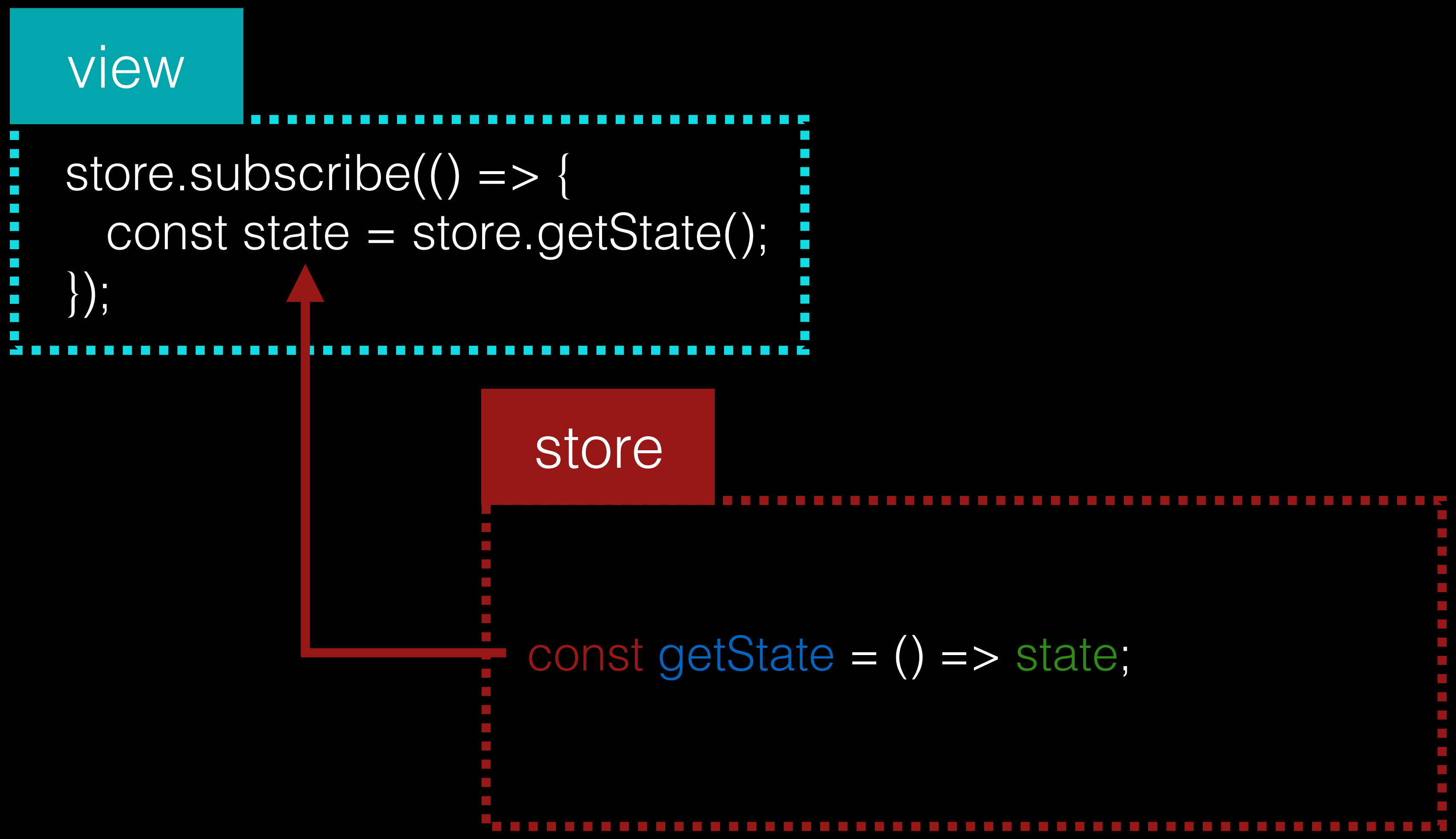


view

```
store.subscribe(() => {
  const state = store.getState();
});
```

store

```
const getState = () => state;
```



view

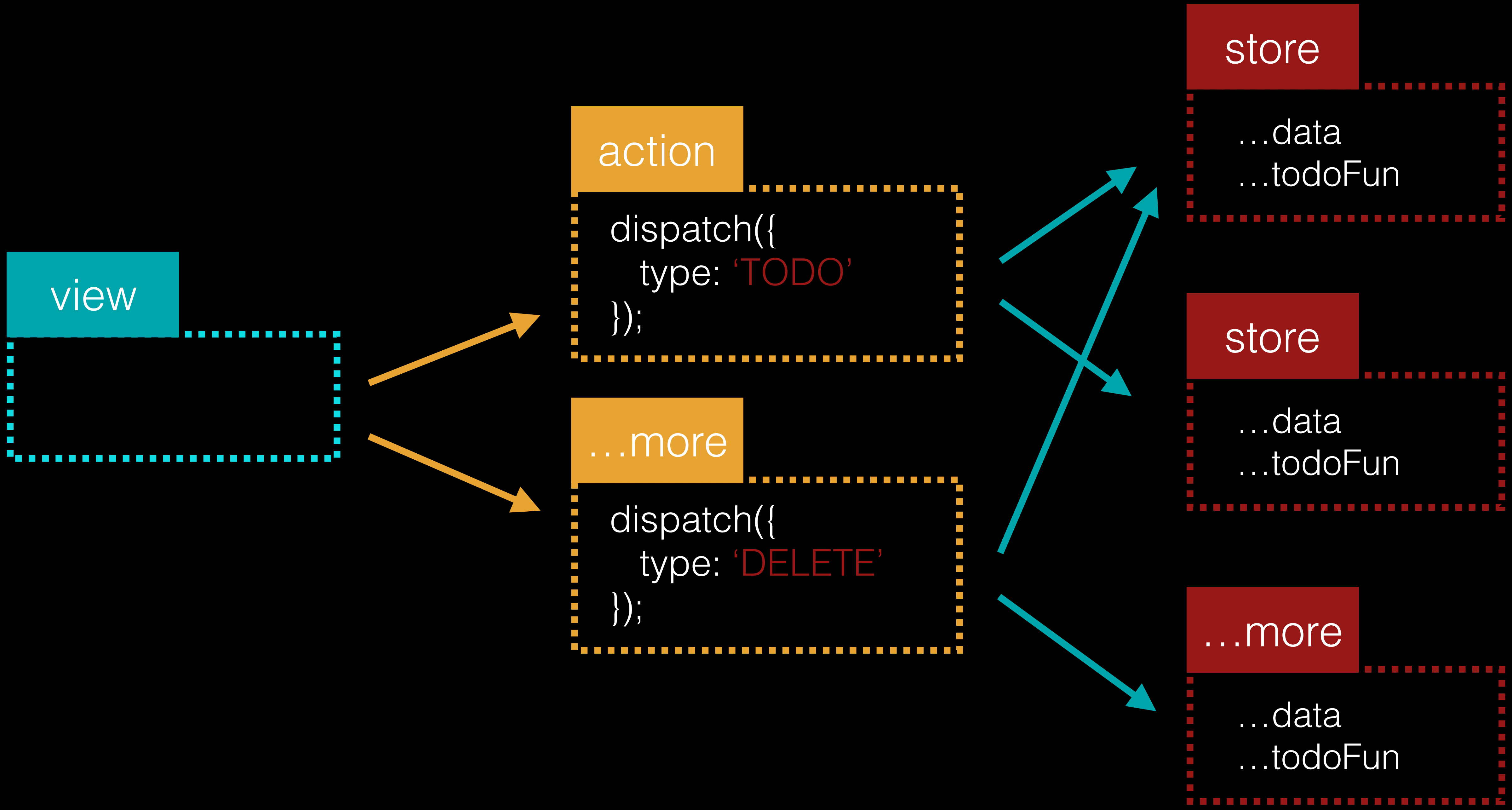
```
store.subscribe(() => {
  const state = store.getState();
});
```

store

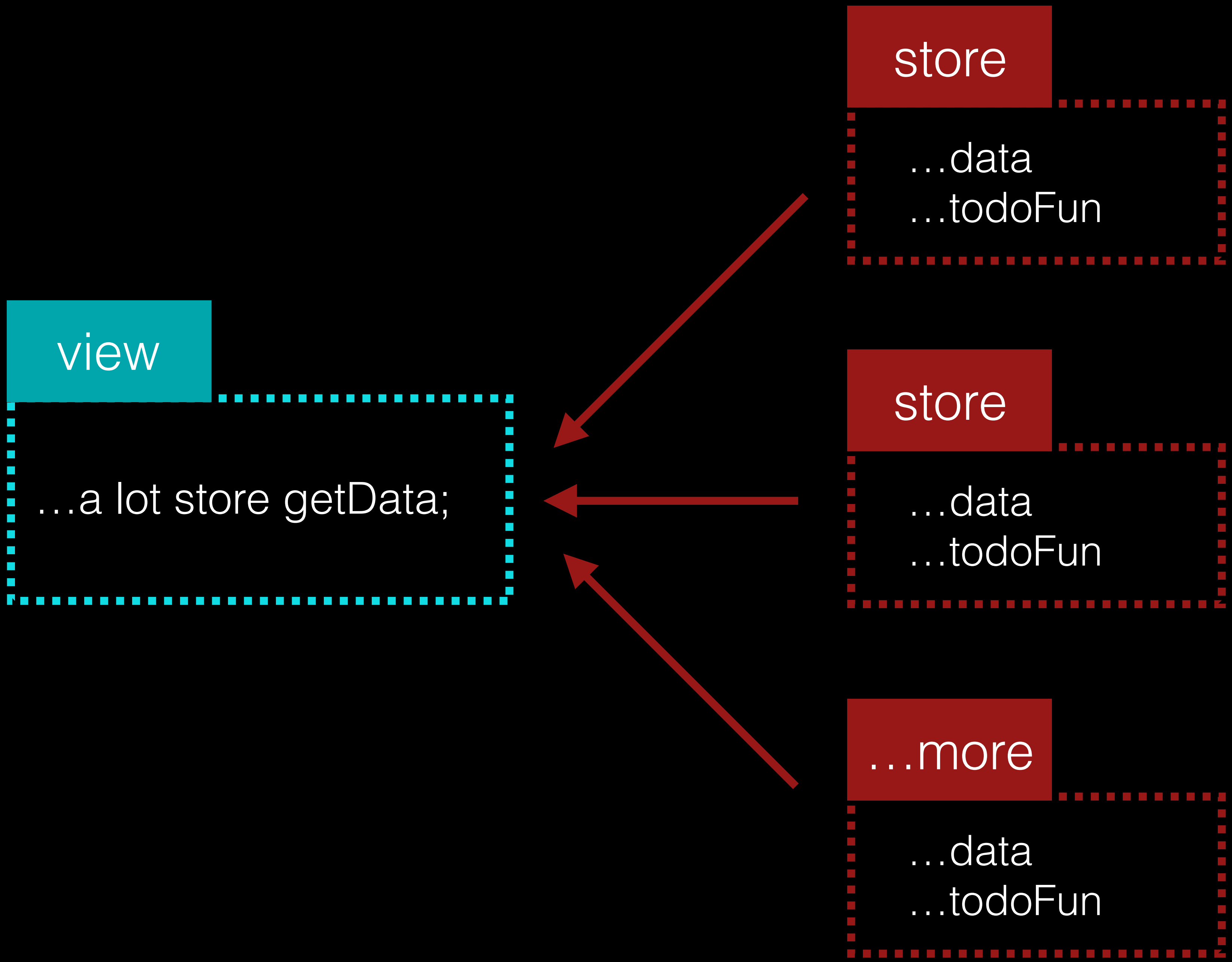
```
const getState = () => state;
```

Single Source of Truth

Flux - dispatch



Flux - getData



view

…a lot store getData;

store

…data
…todoFun

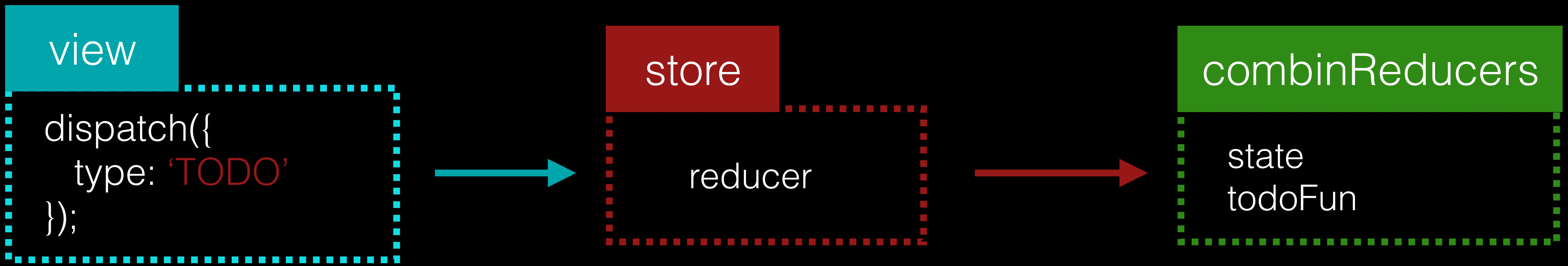
store

…data
…todoFun

…more

…data
…todoFun

redux- dispatch



view

```
dispatch({
   type: ‘TODO’
});
```

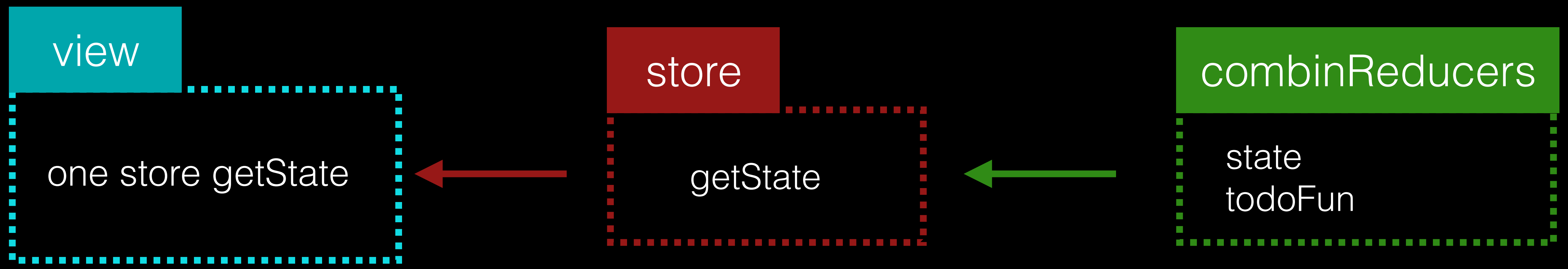
store

reducer

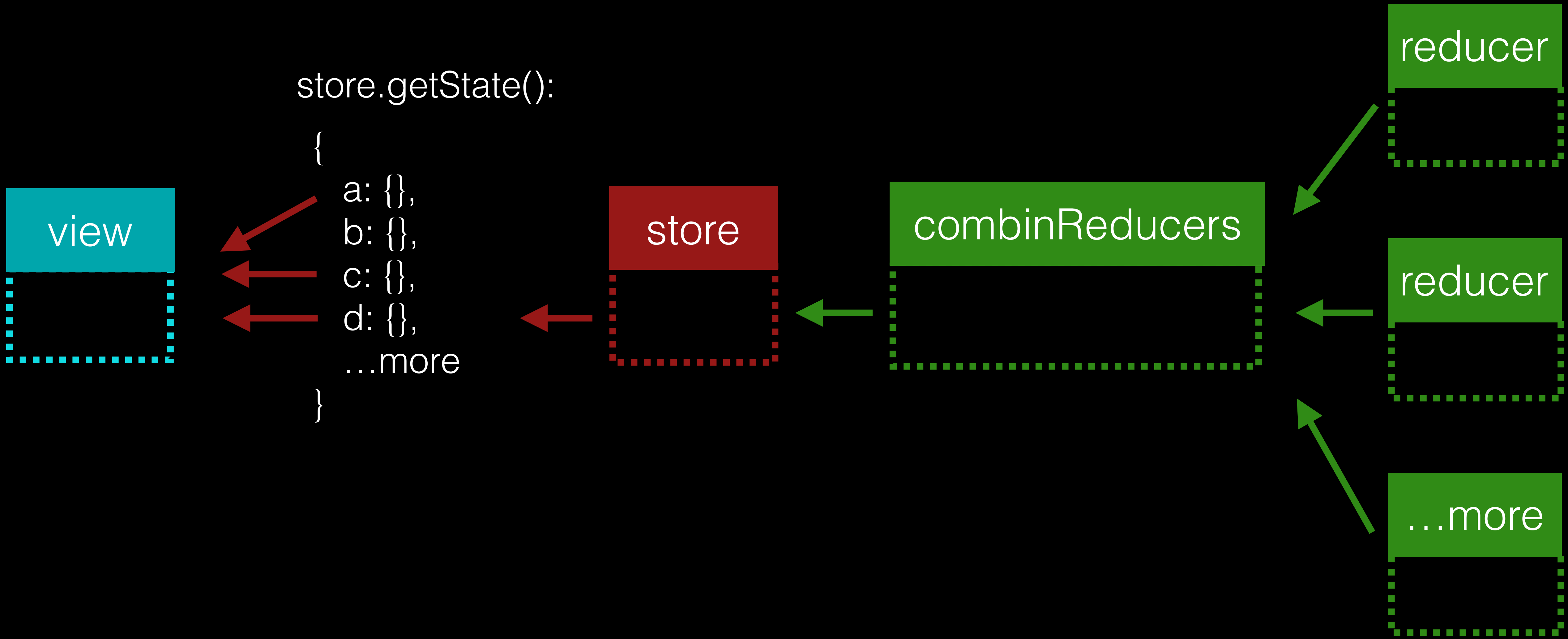
combinReducers

state
todoFun

redux- getState



redux- store real world



store.getState():

```
{
   a: {},
   b: {},
   c: {},
   d: {},
   …more
}
```