

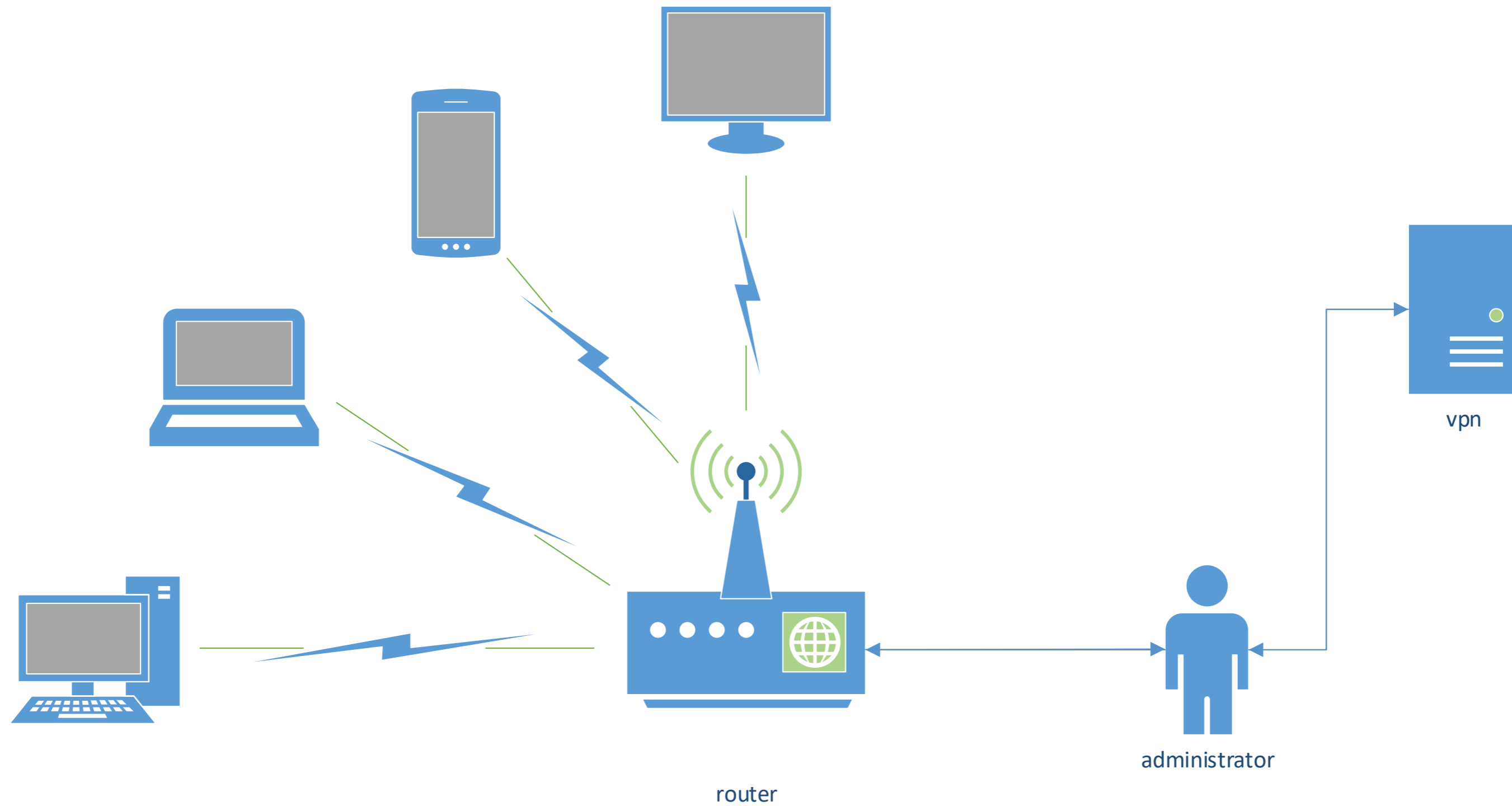


SibEDGE

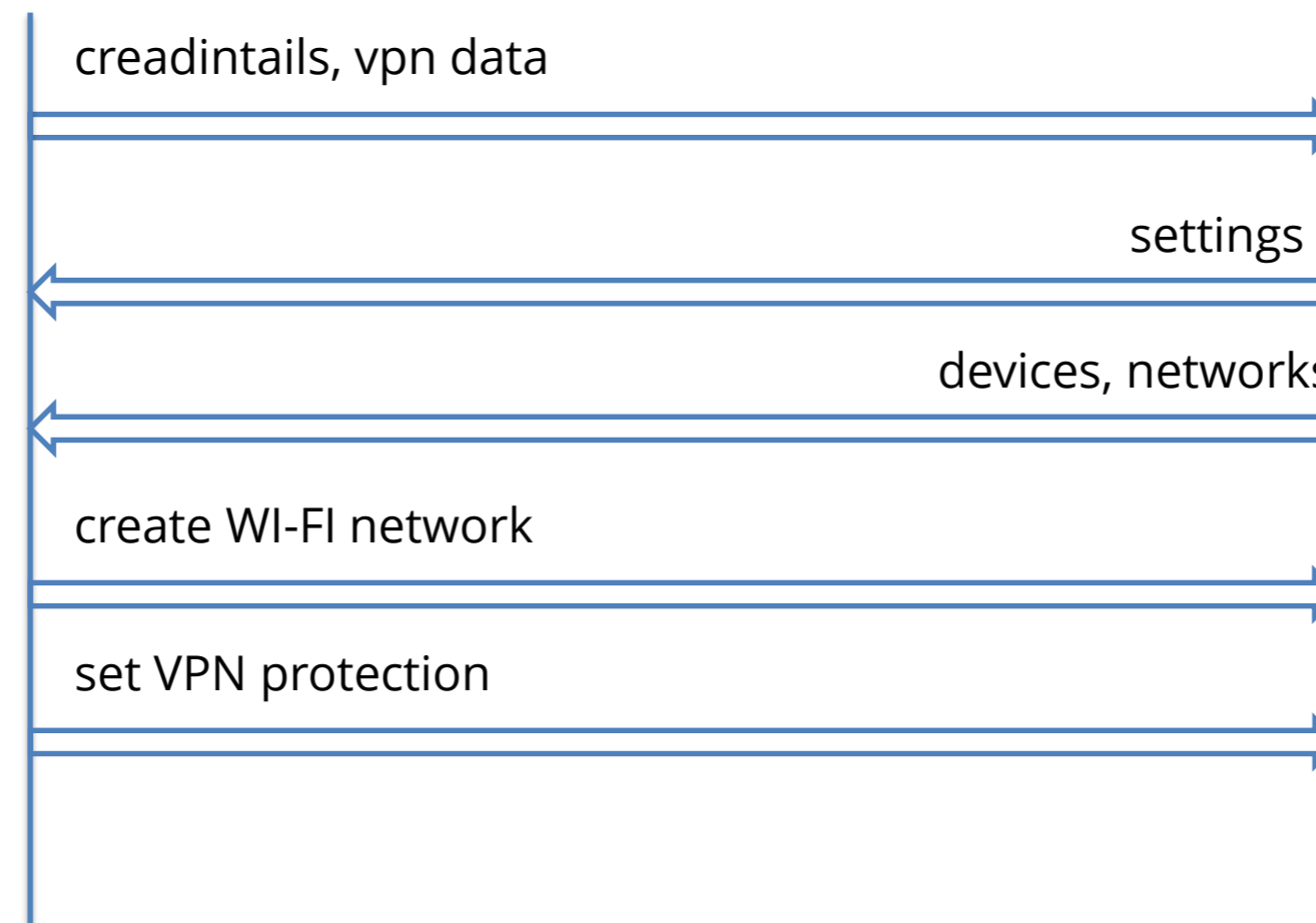
Слава Абрамов

Интеграция WebSocket в архитектуру React/Redux своими руками

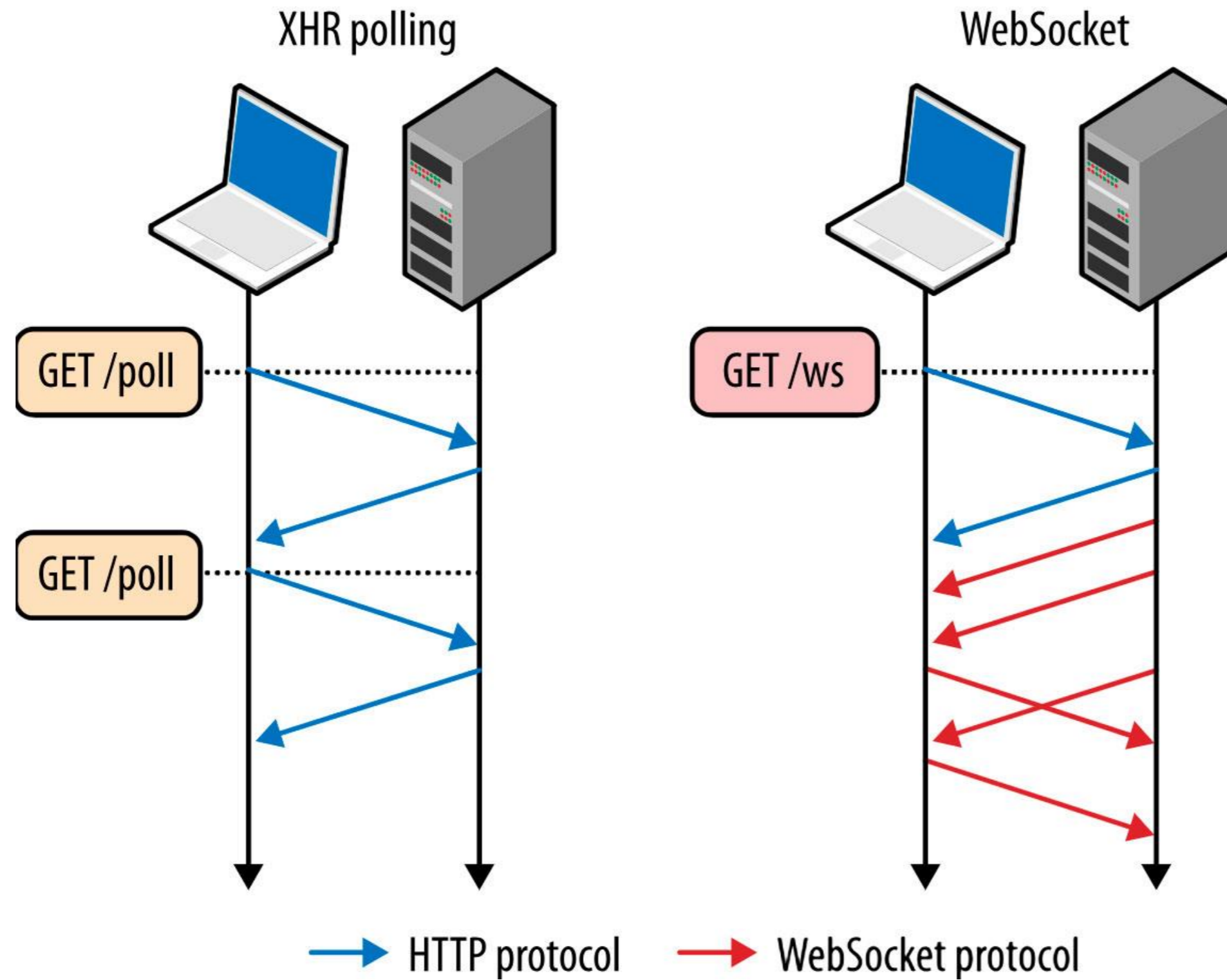
Application



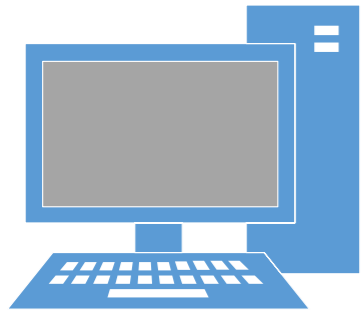
Application



WebSocket

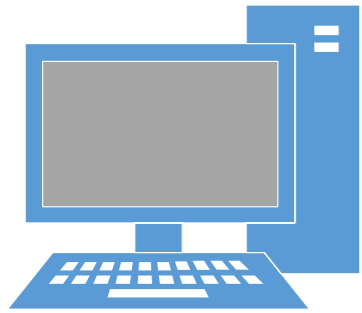


WebSocket



```
let socket = new WebSocket('ws://example.com', ... );
```

WebSocket



```
let socket = new WebSocket('ws://example.com');
```

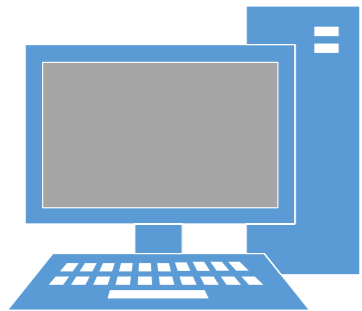
```
socket.onopen = (event) => { ... };
```

```
socket.onclose = (event) => { ... };
```

```
socket.onerror = (event) => { ... };
```

```
socket.onmessage = (event) => { ... };
```

WebSocket



```
let socket = new WebSocket('ws://example.com');
```

```
socket.onopen = (event) => { ... };
```

```
socket.onclose = (event) => { ... };
```

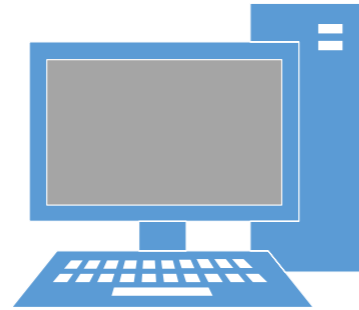
```
socket.onerror = (event) => { ... };
```

```
socket.onmessage = (event) => { ... };
```

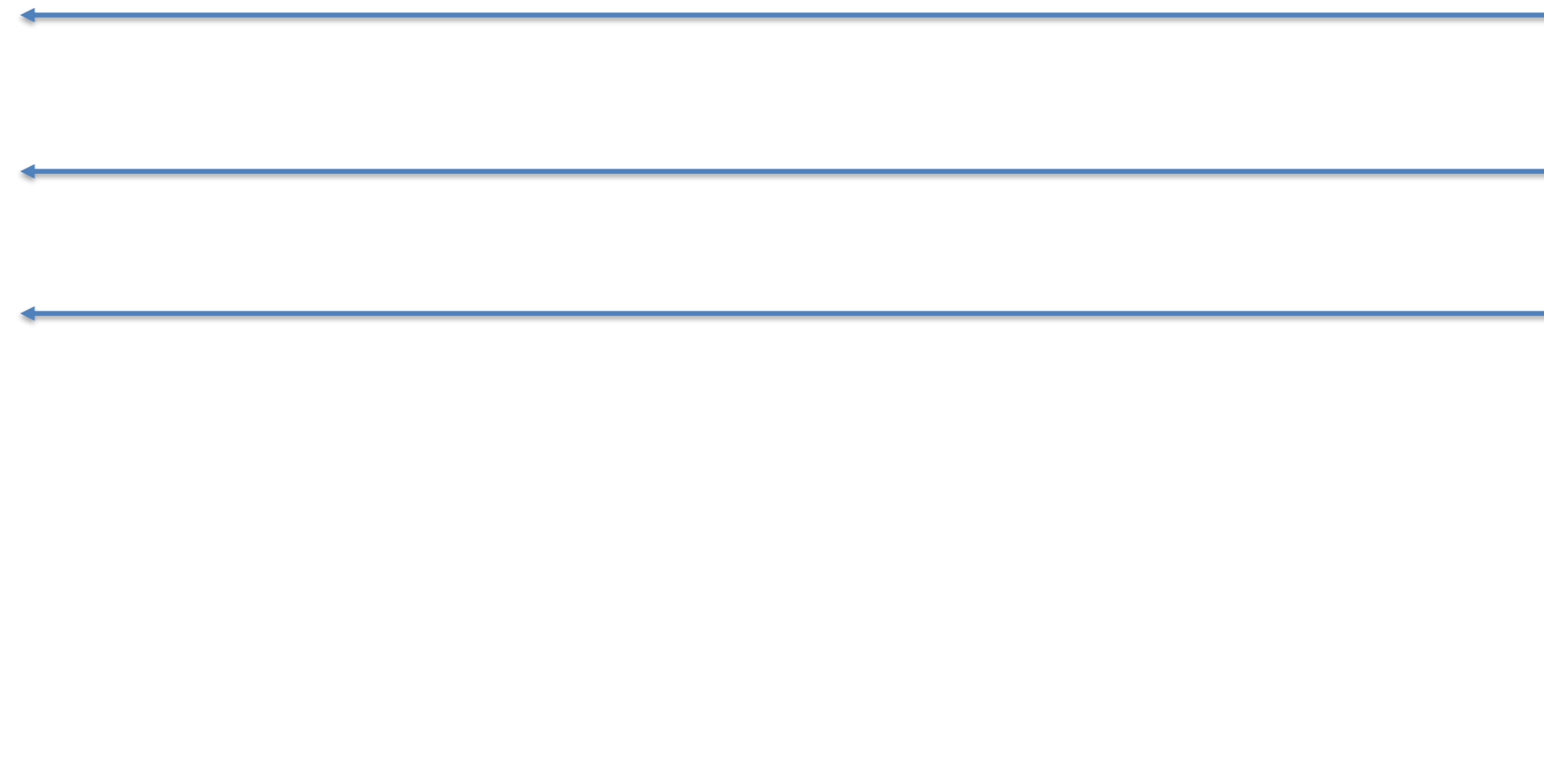
```
socket.send( ... );
```

```
socket.close( ... );
```

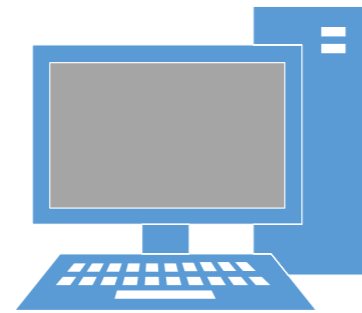
WebSocket



```
socket.onmessage = (event) => { ... };
```



WebSocket



```
socket.onmessage = (event) => { ... };
```



WebSocket

```
let socket = new WebSocket('ws://example.com');
```

```
socket.readyState
```

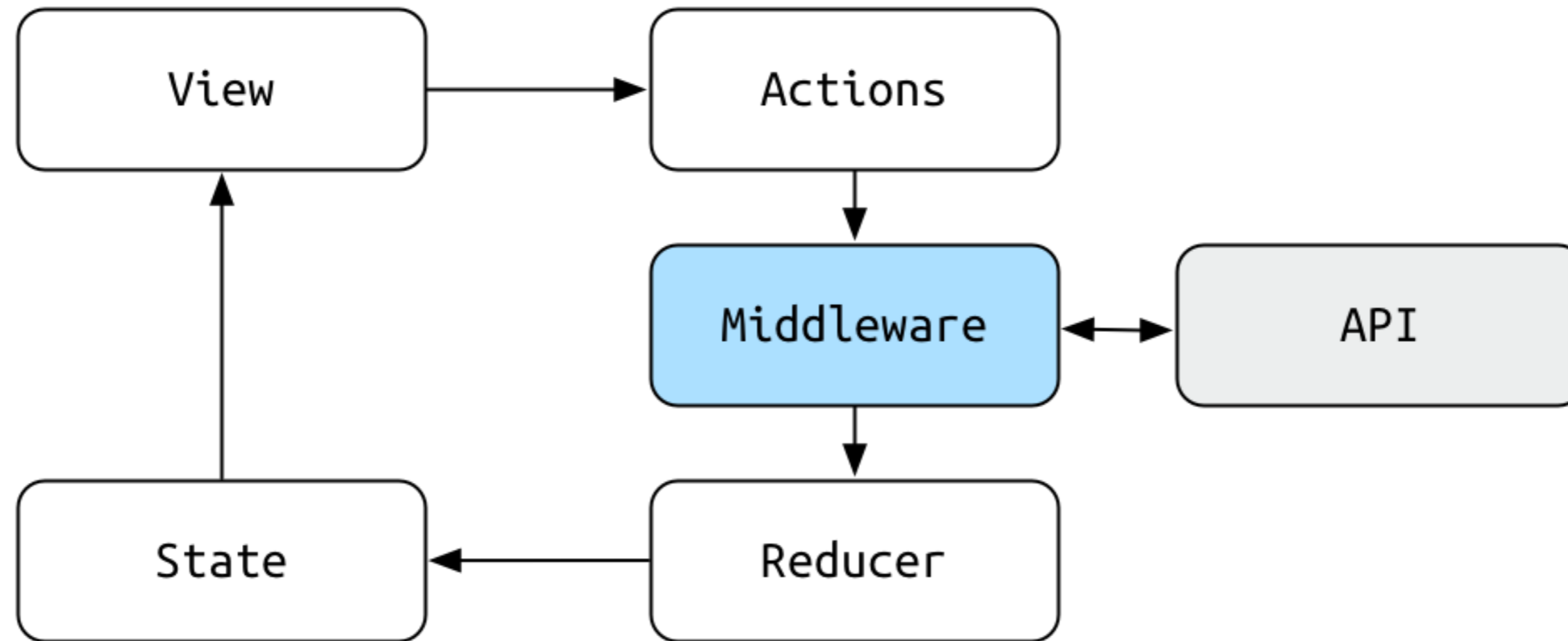
```
0    CONNECTING
```

```
1    OPEN
```

```
2    CLOSING
```

```
3    CLOSED
```

React/Redux

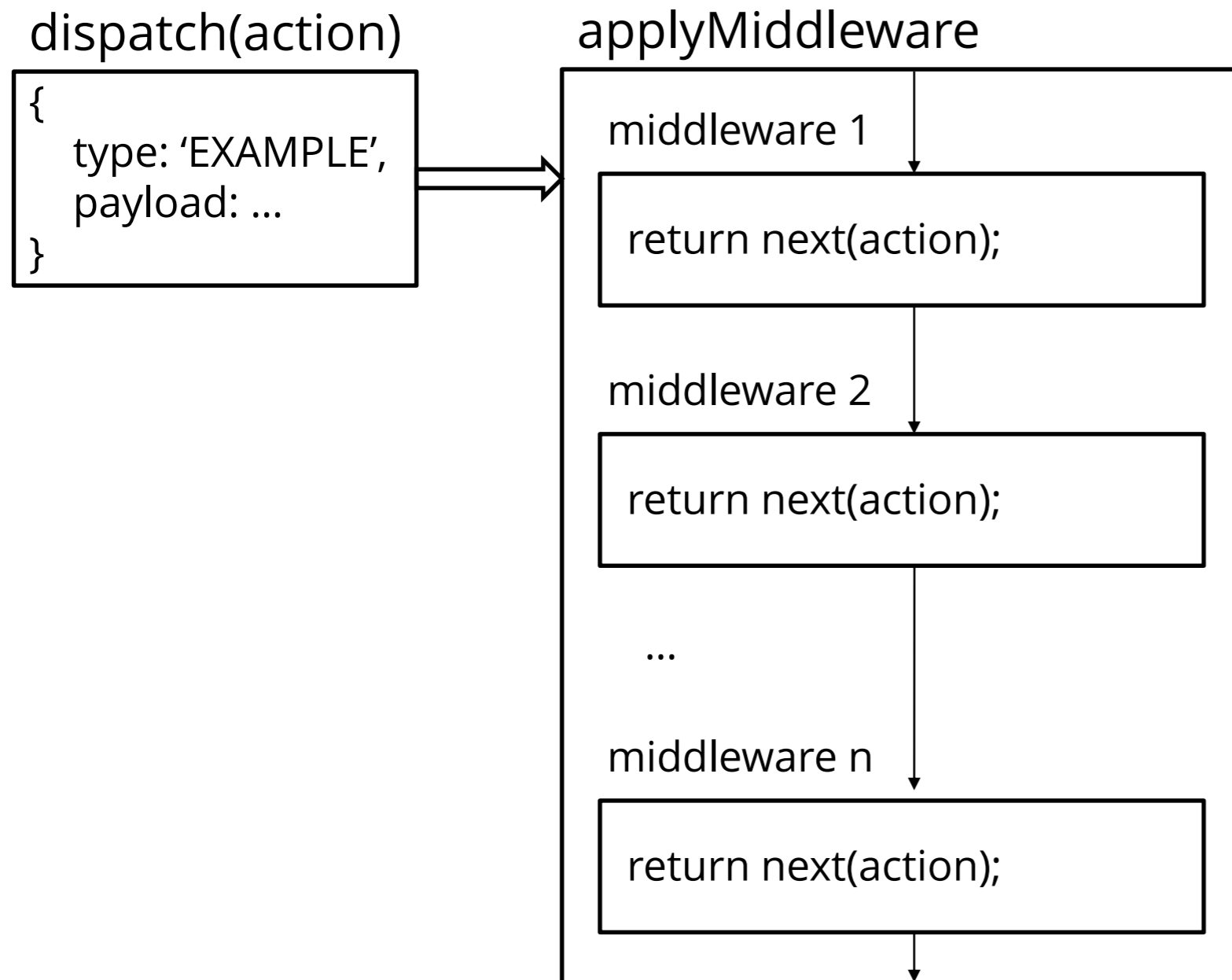


React/Redux

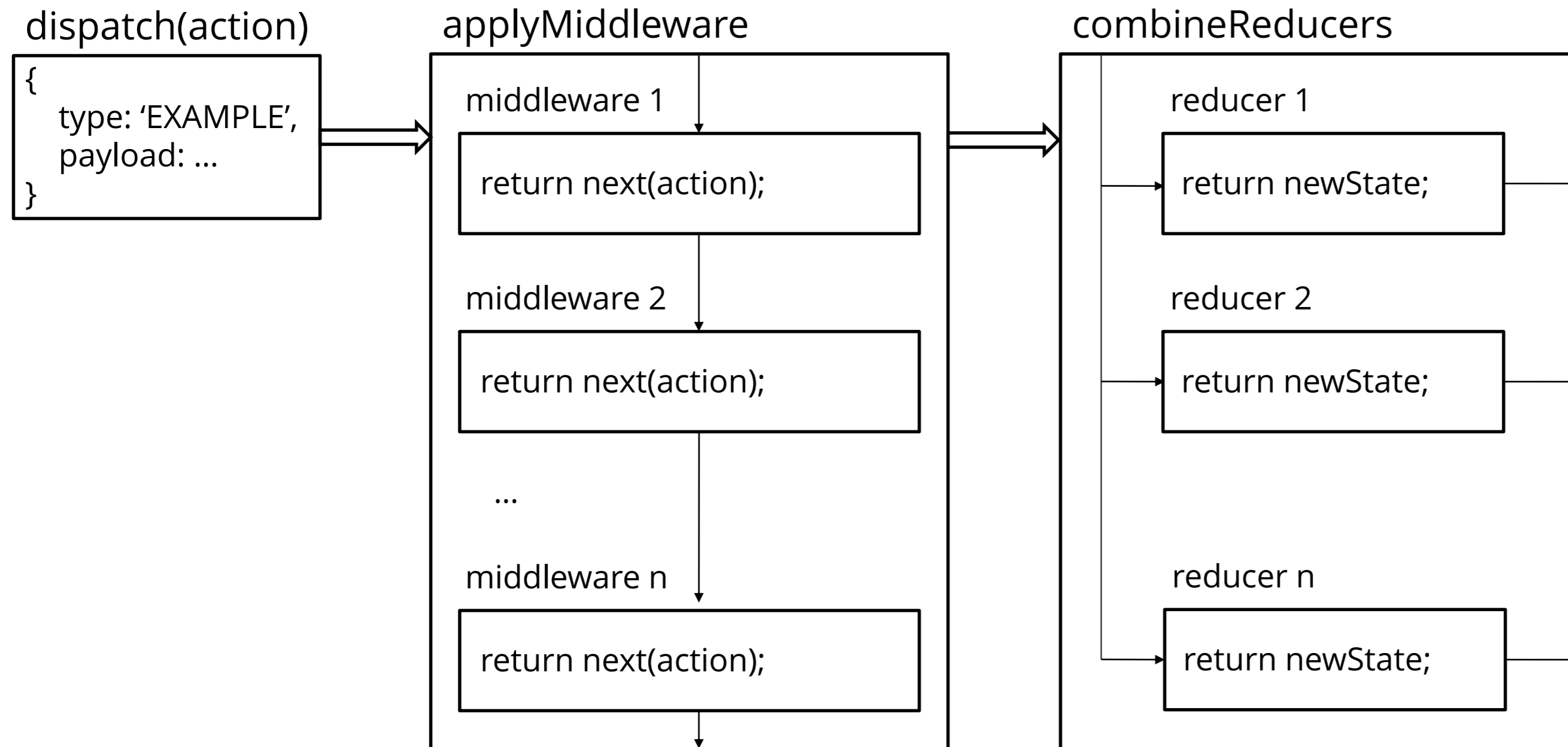
dispatch(action)

```
{  
  type: 'EXAMPLE',  
  payload: ...  
}
```

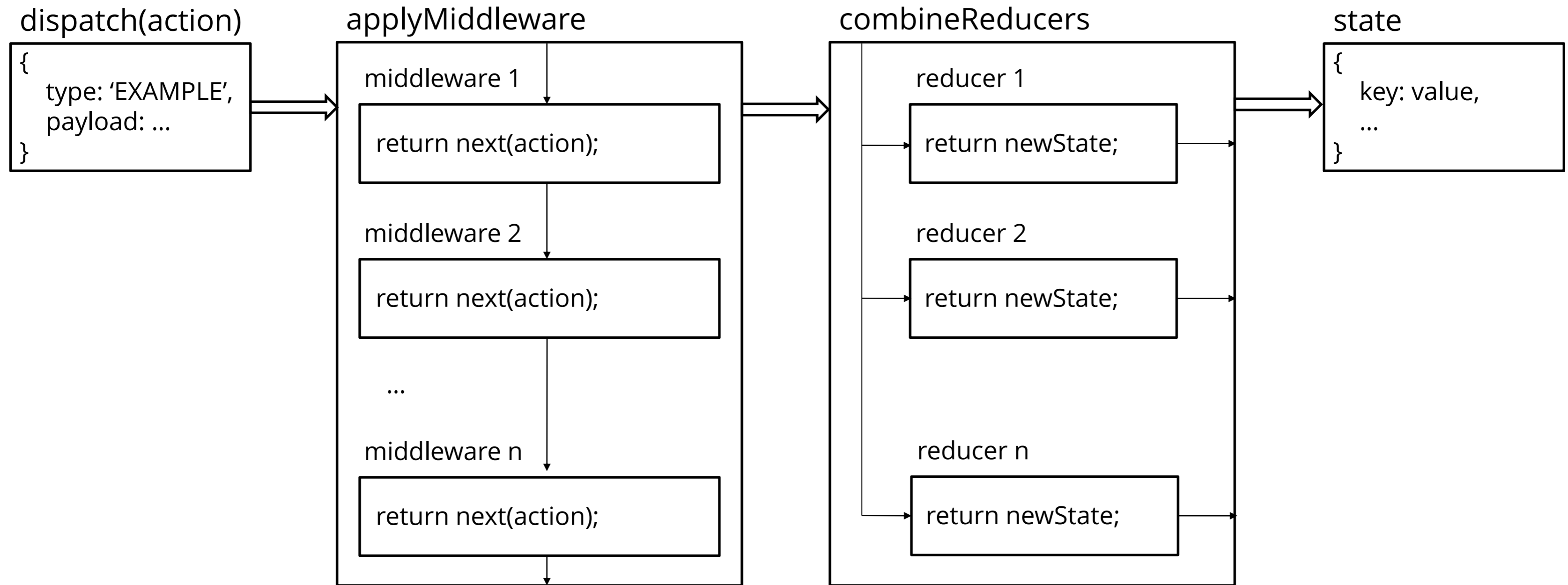
React/Redux



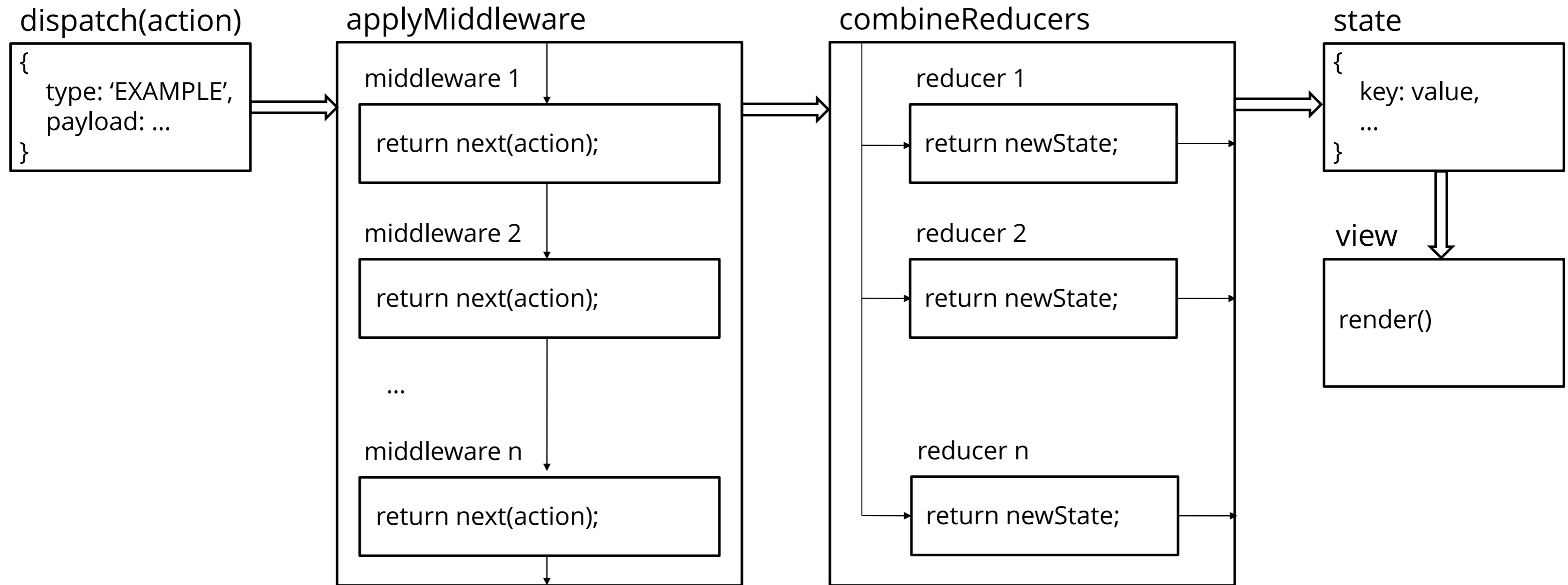
React/Redux



React/Redux



React/Redux



WebSocket + React/Redux

initialState

socket: false,
auth: false,
...

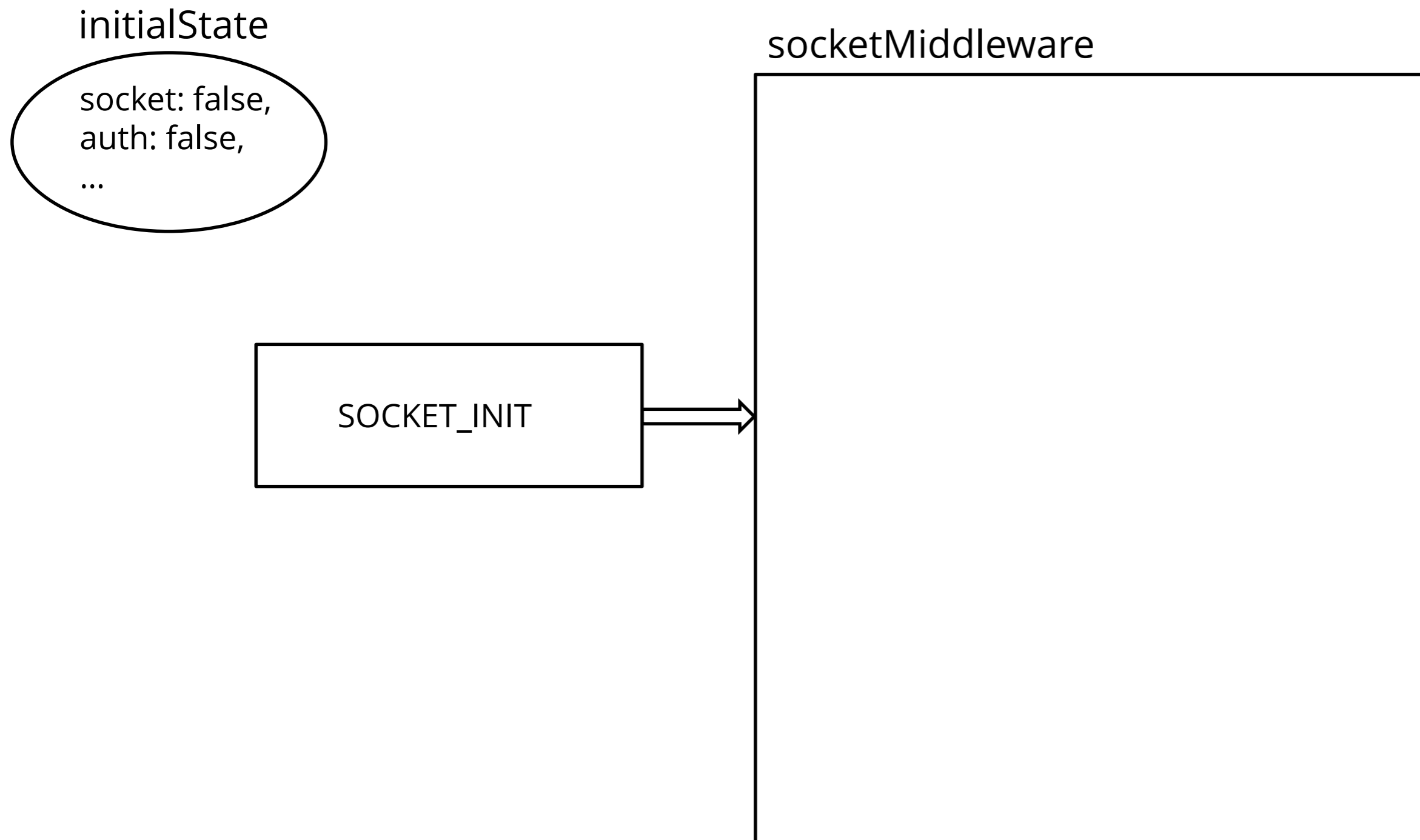
WebSocket + React/Redux

initialState

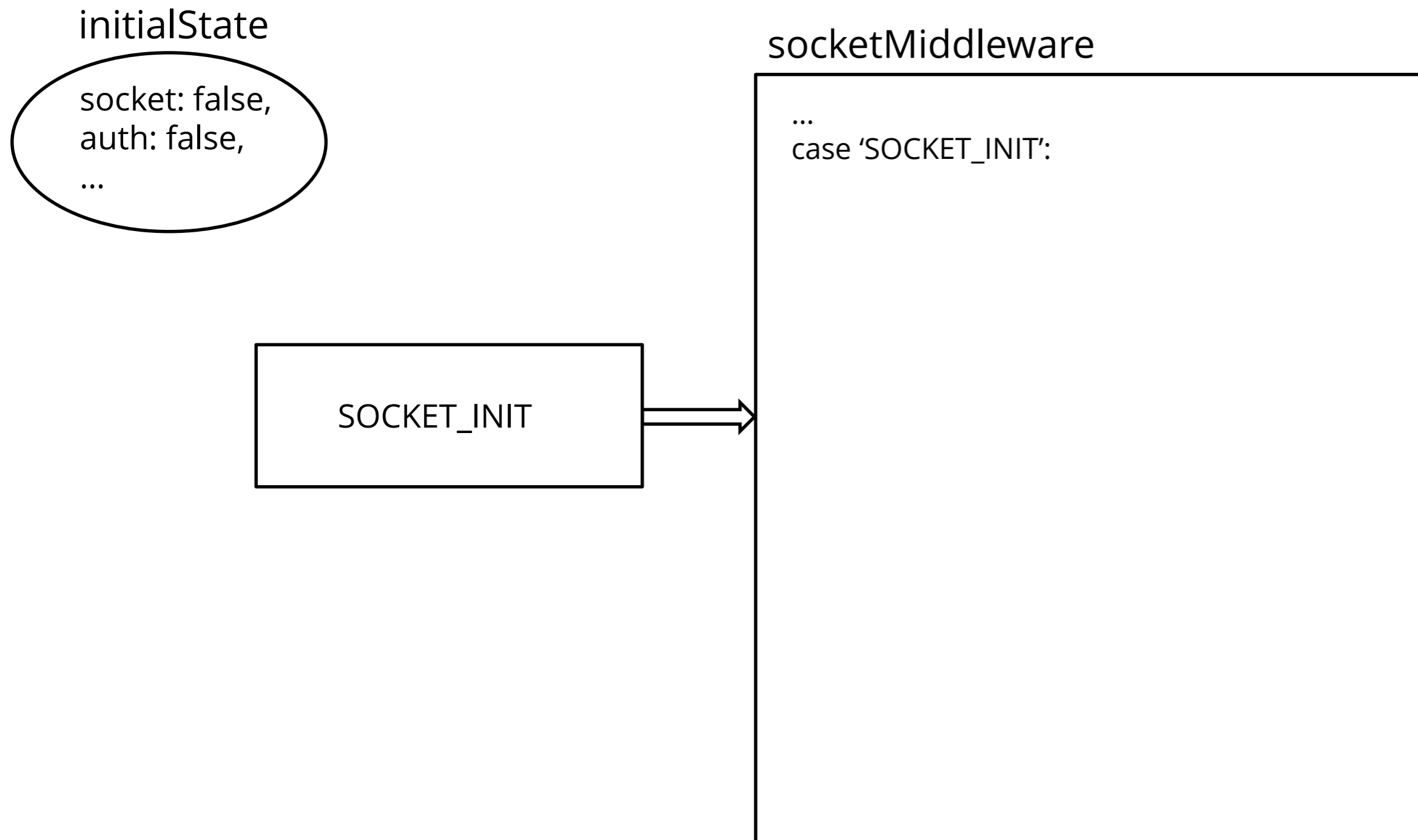
socket: false,
auth: false,
...

SOCKET_INIT

WebSocket + React/Redux



WebSocket + React/Redux



WebSocket + React/Redux

initialState

socket: false,
auth: false,
...

SOCKET_INIT

socketMiddleware

```
...  
case 'SOCKET_INIT':  
  window.MY_SOCKET = new WebSocket( ... );
```

WebSocket + React/Redux

initialState

socket: false,
auth: false,
...

SOCKET_INIT

socketMiddleware

```
...  
case 'SOCKET_INIT':  
  window.MY SOCKET = new WebSocket( ... );  
  window.MY SOCKET.onopen = (event) => {  
    ...  
    return next(action);  
  }  
}
```

WebSocket + React/Redux

initialState

socket: false,
auth: false,
...

SOCKET_INIT

socketMiddleware

```
...  
case 'SOCKET_INIT':  
  window.MY SOCKET = new WebSocket( ... );  
  window.MY SOCKET.onopen = (event) => {  
    ...  
    return next(action);  
  }  
  window.MY SOCKET.onmessage = (event) => {  
    if (...) { ... }  
    return next(action);  
  }  
  ...
```

WebSocket + React/Redux

initialState

socket: false,
auth: false,
...

SOCKET_INIT

socketMiddleware

```
...  
case 'SOCKET_INIT':  
  window.MY SOCKET = new WebSocket( ... );  
  window.MY SOCKET.onopen = (event) => {  
    ...  
    return next(action);  
  }  
  window.MY SOCKET.onmessage = (event) => {  
    if (...) { ... }  
    return next(action);  
  }  
  window.MY SOCKET.onclose...  
  window.MY SOCKET.onerror...  
  ...
```


WebSocket + React/Redux

initialState

socket: false,
auth: false,
...

SOCKET_INIT

socketMiddleware

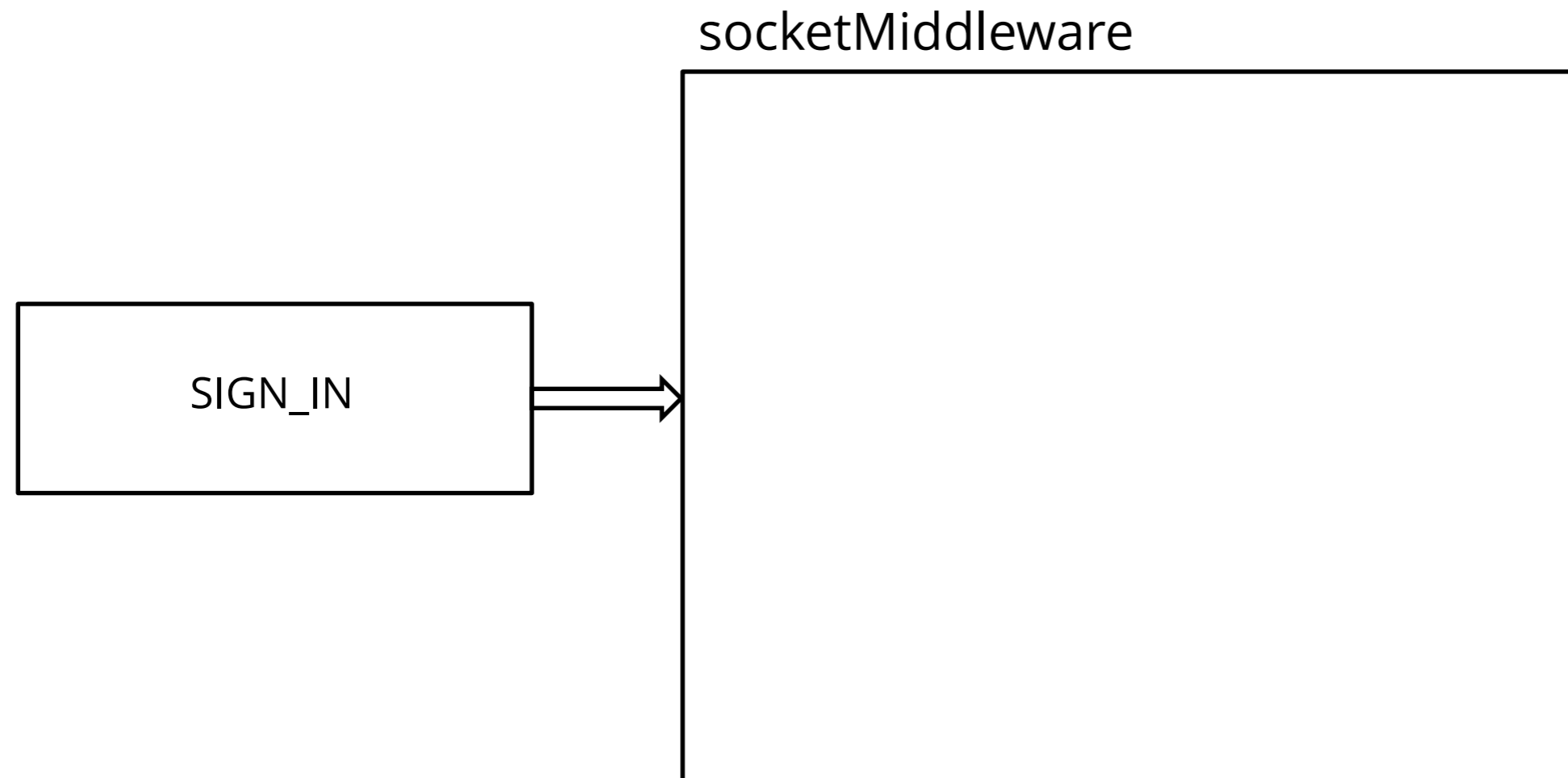
```
...  
case 'SOCKET_INIT':  
  window.MY SOCKET = new WebSocket( ... );  
  window.MY SOCKET.onopen = (event) => {  
    ...  
    return next(action);  
  }  
  window.MY SOCKET.onmessage = (event) => {  
    if (...) { ... }  
    return next(action);  
  }  
  window.MY SOCKET.onclose...  
  window.MY SOCKET.onerror...  
  ...
```

return newState;

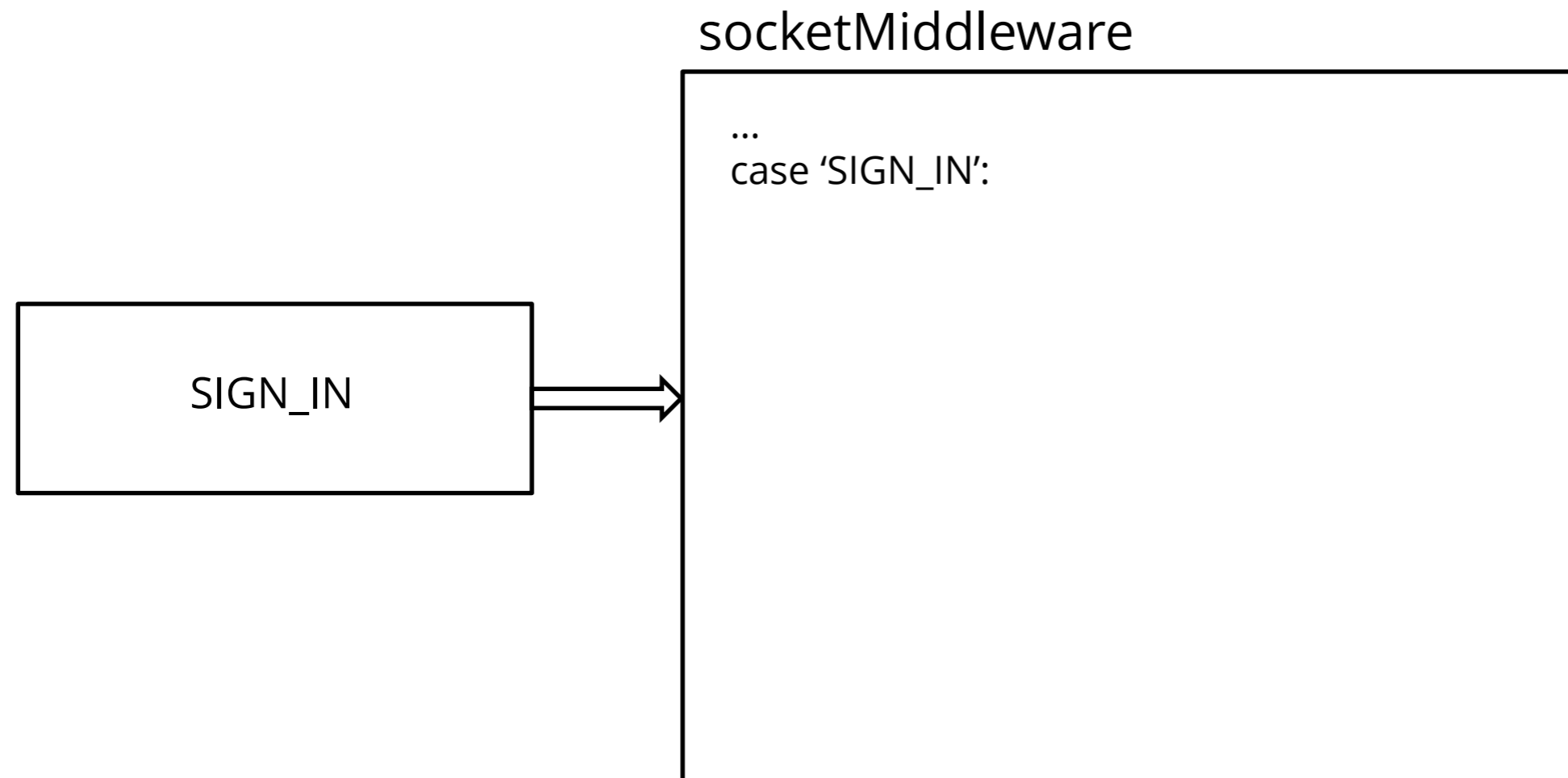
WebSocket + React/Redux

SIGN_IN

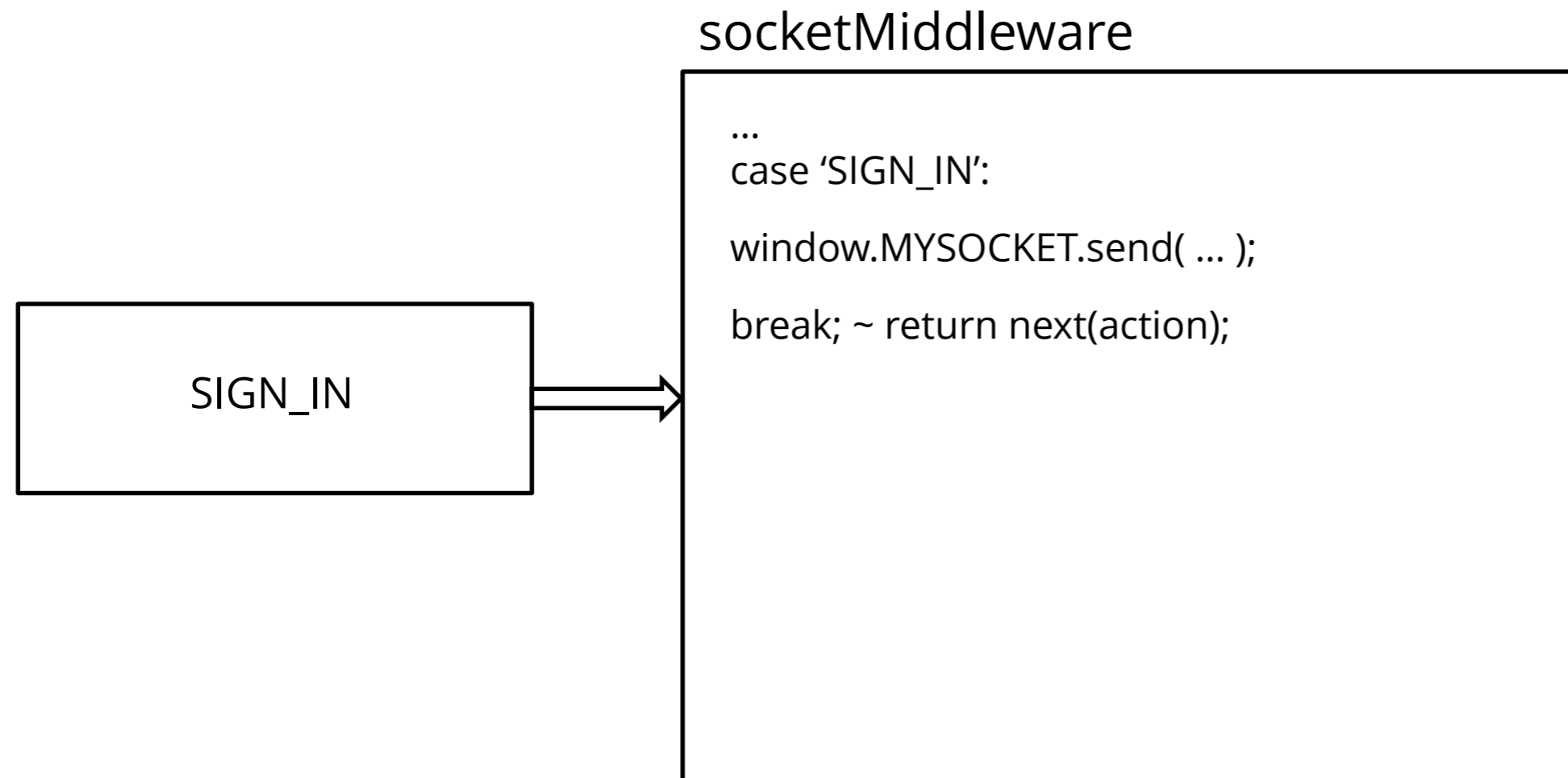
WebSocket + React/Redux



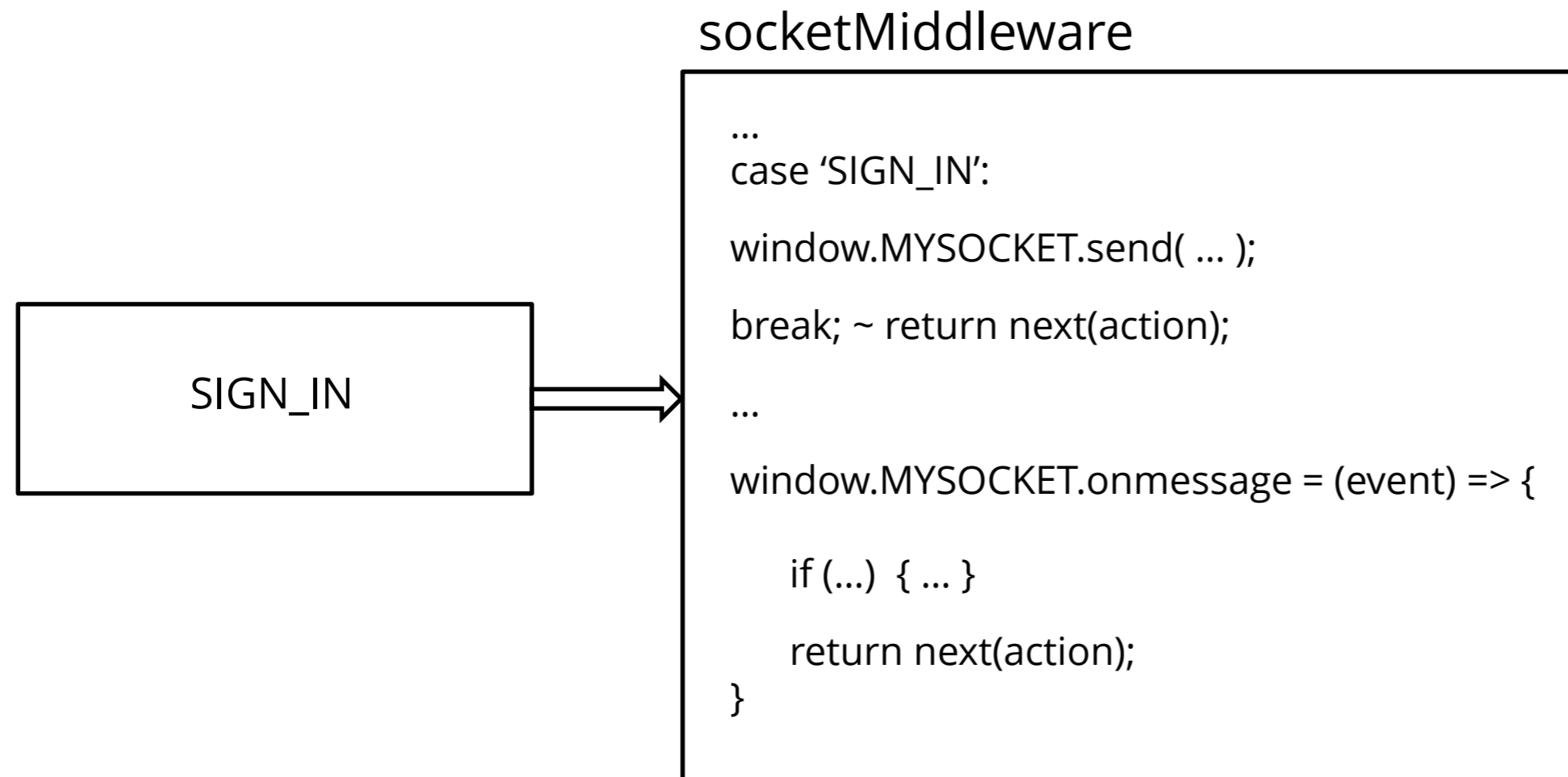
WebSocket + React/Redux



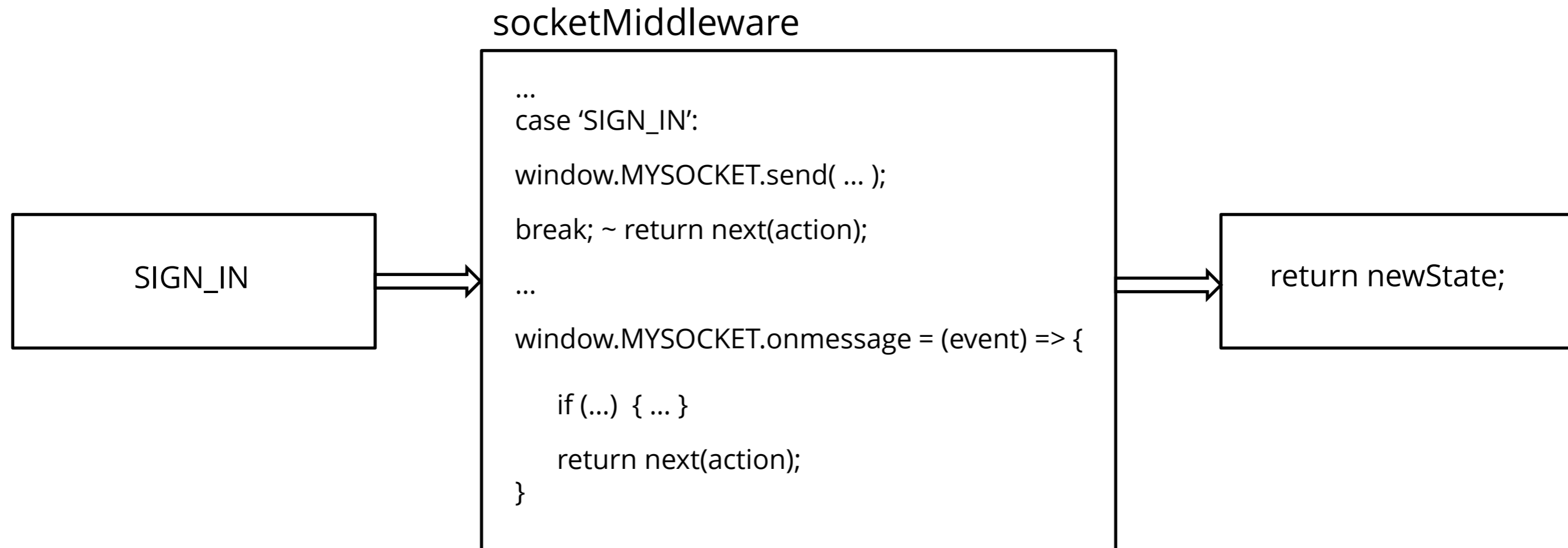
WebSocket + React/Redux



WebSocket + React/Redux



WebSocket + React/Redux



WebSocket + React/Redux

`new WebSocket`

`onopen`

`onclose`

`onerror`

`onmessage`

объявляются 1 раз

`send`

вызывается при каждом обращении к WebSocket

`onmessage`

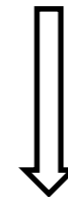
принимает данные от WebSocket и должен вызывать `return next(action)`

KeepAlive

```
let socket = new WebSocket('ws://example.com');
```



```
socket.readyState === 3 (CLOSED)
```



```
socket.onclose = (event) => { ... }
```

KeepAlive

```
let socketClose;
```

KeepAlive

```
let socketClose;
```

```
socket.onopen = () => {
```

```
  ...
```

```
  socket.send( ... );
```

```
  socketClose = setTimeout(() => {  
    socket.close();
```

```
  }, 5000);
```

```
  ...
```

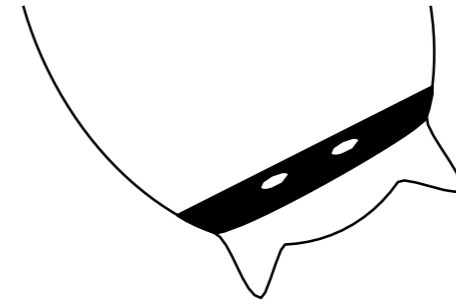
```
};
```

KeepAlive

```
let socketClose;
```

```
socket.onopen = () => {  
  ...  
  socket.send( ... );  
  
  socketClose = setTimeout(() => {  
    socket.close();  
  }, 5000);  
  ...  
};
```

```
socket.onmessage = () => {  
  ...  
  clearTimeout(socketClose);  
  
  setInterval(() => {  
    socket.send( ... );  
  
    socketClose = setTimeout(() => {  
      socket.close();  
    }, 5000);  
  
  }, 2000);  
  ...  
};
```



Спасибо за внимание



SibEDGE