

HISTORIEN OM REACT

OG OPEN-SOURCE I
VERDEN S STØRSTE SELSKAP

ERIK WENDEL, BEKK CONSULTING, OSLO





ERIK WENDEL

UTVIKLER & KONSULENT

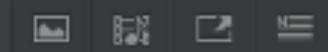
- » jobber i BEKK
- » engasjert hos SpareBank1
(nett- og mobilbank)
- » fagleder for
frontendarkitektur
- » ~2.5 års erfaring med React
- » ~1 års erfaring med Redux



Display

Top New

Filter



NSFW content

Hide Safe Show

Top submissions

Thomas Edison's Nightmare

Created by [The Oatmeal](#) and submitted 25 seconds ago by [Gabriele Cirulli](#) Admin

Autumn leaves

Submitted 36 seconds ago by [Gabriele Cirulli](#) Admin

Give feedback



**Are you an
entrepreneur?**

We can help you manage
your expenses!

[Learn More](#)

www.entrepreneursupport.co

Ads by Google

[AdChoices ▶](#) [◀](#) [▶](#)

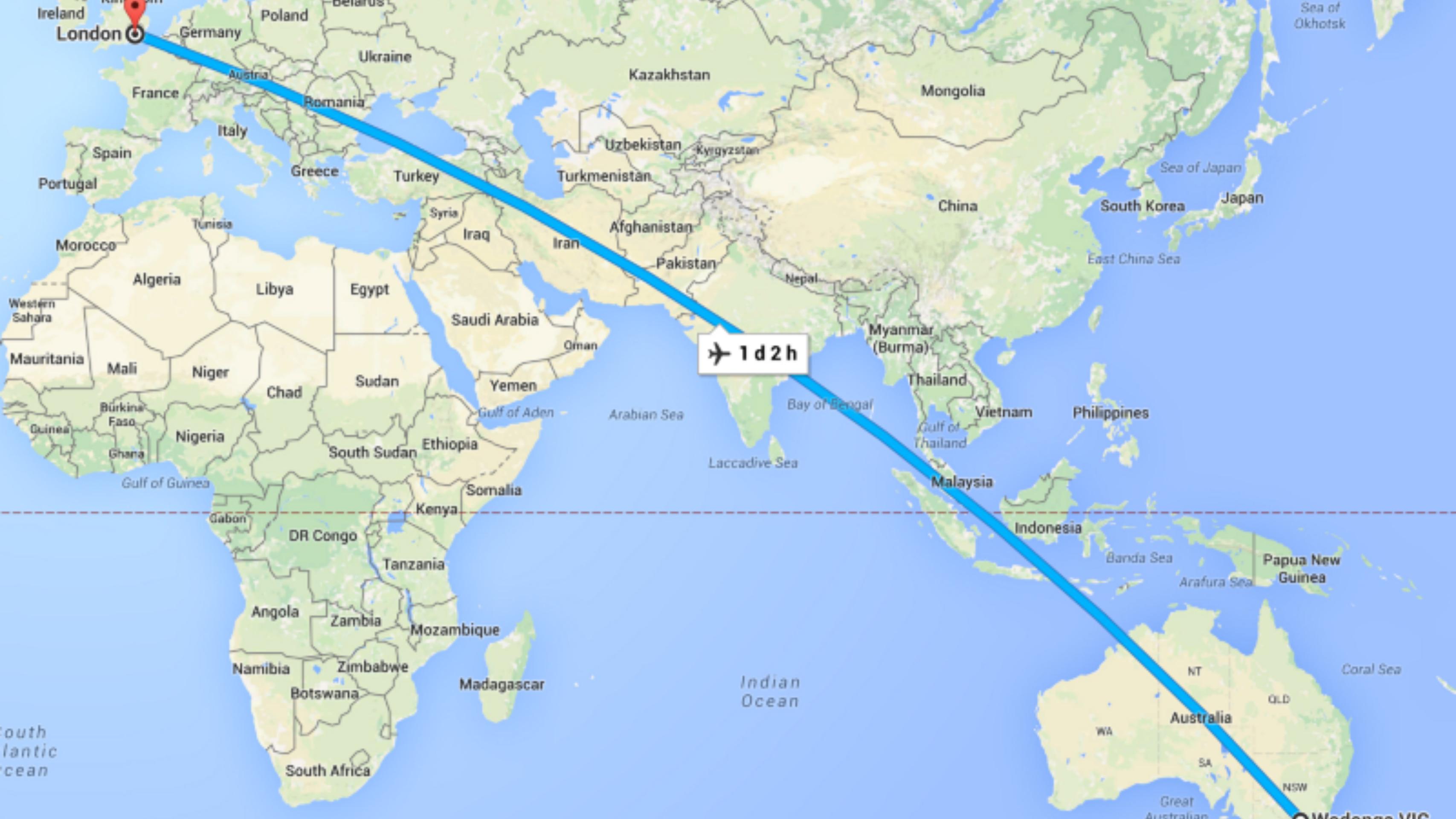
[Affordable Hotels](#)

Special rates until the
end of the month. No
booking fees.
www.example.com



CRIPPLING SOCIAL ANXIETY





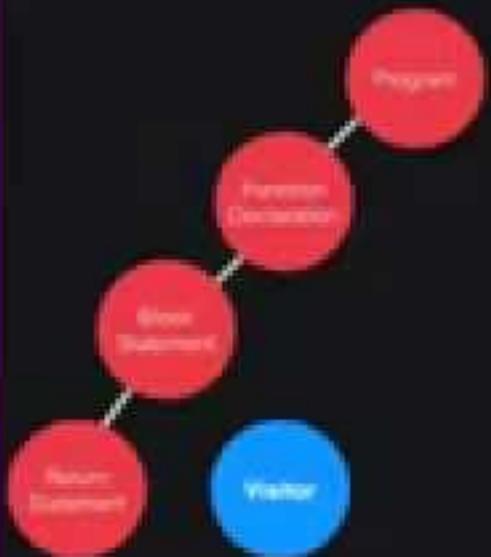
1 d 2 h

Weddington, VIC





Traversal



Babel is a JavaScript compiler.

Use next generation JavaScript, today.

Babel 6.16.0 (Happy Birthday Babel)

 Star 18,067

 Slack 89/4615

Babel transforms your JavaScript

You put JavaScript in

```
[1,2,3].map(n => n + 1);
```

And get JavaScript out

```
[1,2,3].map(function(n) {  
  return n + 1;  
});
```

Babel transforms your JavaScript

You put JavaScript in

```
[1,2,3].map(n => n + 1);
```

And get JavaScript out

```
[1,2,3].map(function(n) {  
    return n + 1;  
});
```

Try it Out

Start by installing the Babel CLI and a preset

```
$ npm install --save-dev babel-cli babel-preset-latest
```

See who's using Babel

Logos are added by company representatives. These companies may or may not be using Babel on their main web properties, but they're definitely using it somewhere in their organizations :)

facebook



NETFLIX

mozilla

YAHOO!

 **PayPal**

 **airbnb**

 **reddit**

 **Spotify®**

 **Atlassian**



 **bocoup**



 **zendesk**

vimeo



Sebastian McKenzie [Follow](#)

I'm a Software Engineer living in London working on JavaScript Engineer at Facebook. From Wodong...

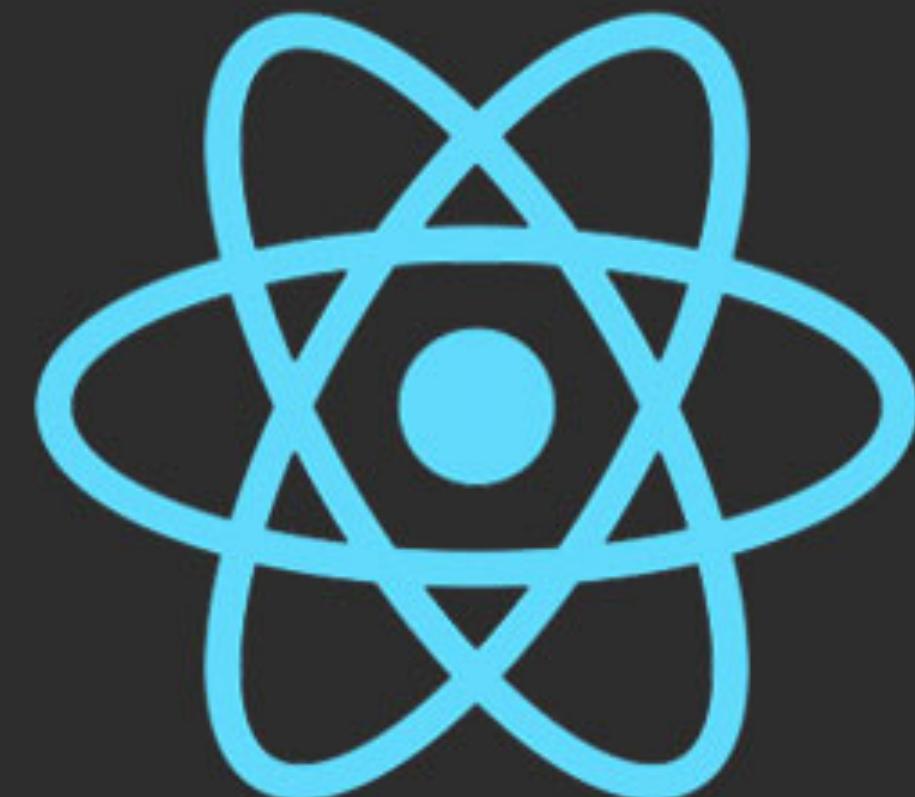
Jan 5 · 18 min read

~2015 in review

This started off as a generic year in review but as I was writing it became a lot more personal. This is my story of how I dealt with going from high school to moving to another country and working in big tech in a little over a year.

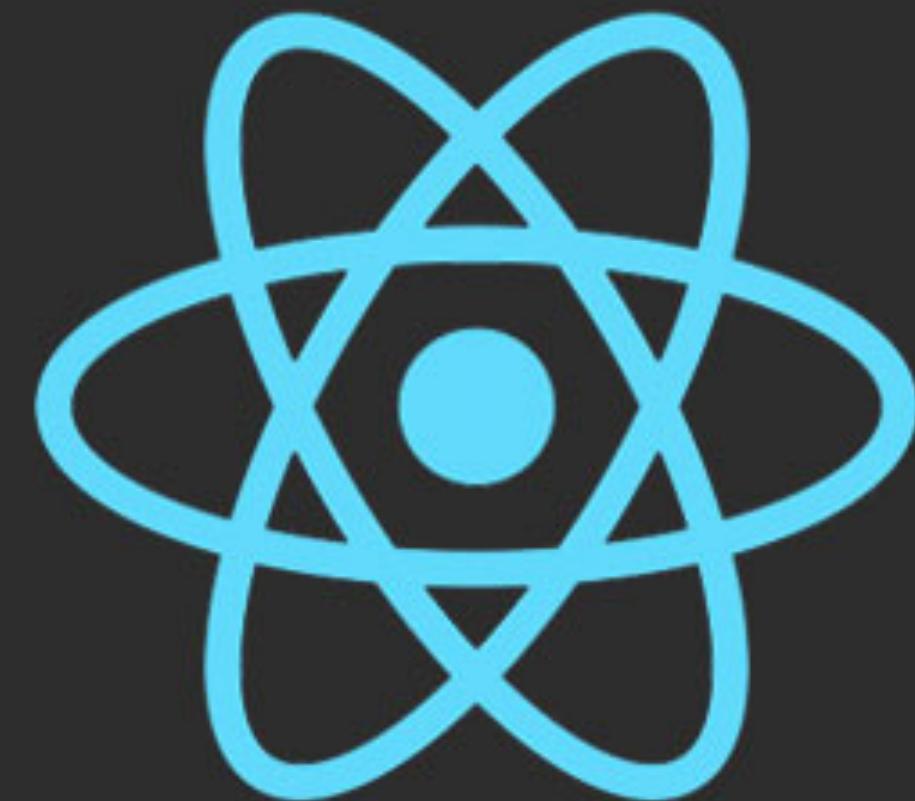
All throughout my childhood I've had an interest in computers and the web. When I was eight I published my [first website](#), when I was eleven I was [causing mischief](#) on a forum for a kids MMO, and at fifteen I developed a [Tumblr-like service](#) with a friend. It was no surprise to my family or school teachers when I said that I wanted to get a job in software development. I didn't know quite how I'd go about it but I certainly knew the direction that I wanted to pursue.

The screenshot shows the Amuzor homepage with the title 'Top submissions'. The first submission is titled 'Thomas Edison's Nightmare' and is created by 'The Oatmeal' and submitted by 'Gabriele Cirulli'. The image for this submission is a cartoon illustration of a man shouting into a megaphone while standing on a large, grinning, grey cat-like creature. Another smaller image shows a woman sitting cross-legged with her arms raised, working on a laptop. The interface includes a sidebar with 'Display' options (Top, New), 'Filter' options, and 'NSFW content' checkboxes. Navigation links include 'Browse', 'Submit', 'Admin', and a search bar.



React





React

SIMPLY

**DECLARATIVE
INTERACTIVE DATA
COMPONENTS**

**SIMPLE
DECLARATIVE
INTERACTIVE DATA
COMPONENTS**

A black and white photograph of a large, classical-style building with a prominent portico and a steeple visible in the background, set against a dark sky.

SIMPLE DECLARATIVE INTERACTIVE DATA COMPONENTS

A black and white photograph of a large, ornate building with a prominent clock tower and a pedimented entrance, set against a dark background.

A black and white photograph of a large, classical-style building with a prominent tower and columns, set against a dark sky.

SIMPLE DECLARATIVE INTERACTIVE DATA COMPONENTS

A COMPONENT'S RESPONSIBILITY:

DATA ➤ DOM



EKSEMPELKOMPONENT

```
function MyComponent() {  
    return (some HTML goes here)  
}
```

JSX - JAVASCRIPT EXTENSIONS

```
function MyComponent() {  
  return (  
    <div id="my-id">  
      <span>  
        Hello World!  
      </span>  
    </div>  
  );  
}
```

EKSEMPELKOMPONENT

```
function MyComponent() {  
  return React.createElement(  
    'div',  
    { id: 'my-id' }  
    [  
      React.createElement(  
        'span', null, 'Hello World!')  
    ]  
  );  
}
```

PROPS

```
function MyComponent(props) {  
  return (  
    <div id="my-id">  
      <span>  
        Hello { props.name }  
      </span>  
    </div>  
  );  
}
```

Babel is a JavaScript compiler.

Use next generation JavaScript, today.

Babel 6.16.0 (Happy Birthday Babel)

ES6 (DESTRUCTURING)

Babel transforms your JavaScript

You put JavaScript in

```
[1,2,3].map(n => n + 1);
```

And get JavaScript out

```
[1,2,3].map(function(n) {  
  return n + 1;  
});
```

PROPS DESTRUCTURED

```
var props = {  
  name: 'World'  
}
```

```
function MyComponent(props) {  
  return (  
    <div id="my-id">  
      <span>  
        Hello { props.name }  
      </span>  
    </div>  
  );  
}
```

PROPS DESTRUCTURED

```
function MyComponent({ name }) {  
  return (  
    <div id="my-id">  
      <span>  
        Hello { name }  
      </span>  
    </div>  
  );  
}
```

RENDERING

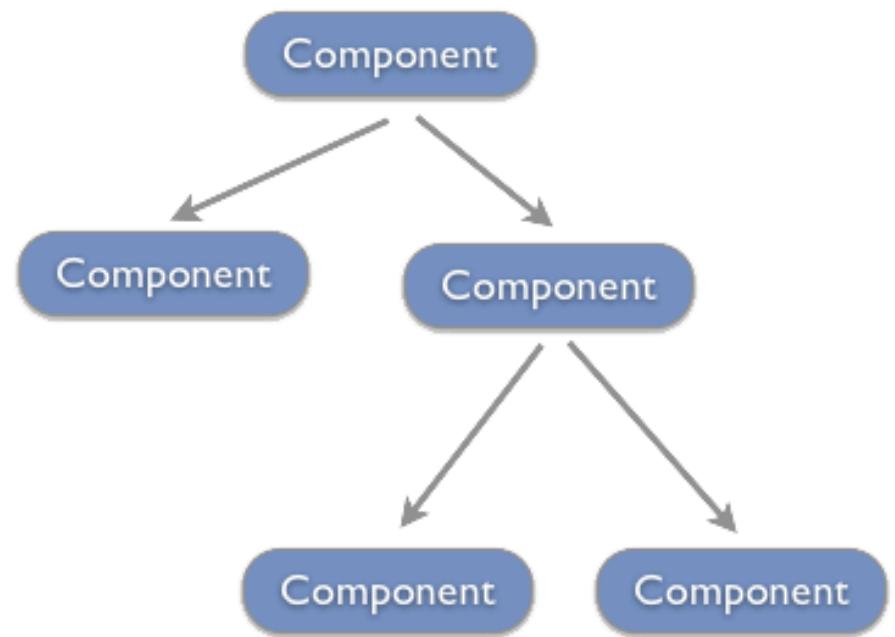
```
import { render } from 'react-dom';
import Hello from './hello.jsx';

render(<Hello name='World' />,
  document.getElementById('container'));

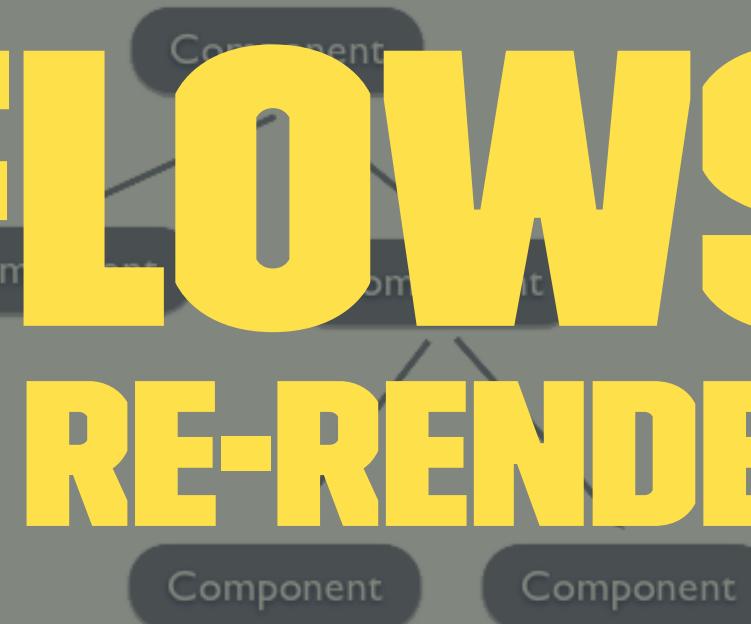
props = {
  name: 'World'
}
```

KOMPONENTTRE

```
function MyComponent() {  
  return (  
    <div>  
      <FancyHeader />  
      <Content />  
      <AwesomeFooter />  
    </div>  
  );  
}
```



**PROPS FLOWS DOWN
DATA CHANGE = RE-RENDER EVERYTHING**





BUT WAIT...

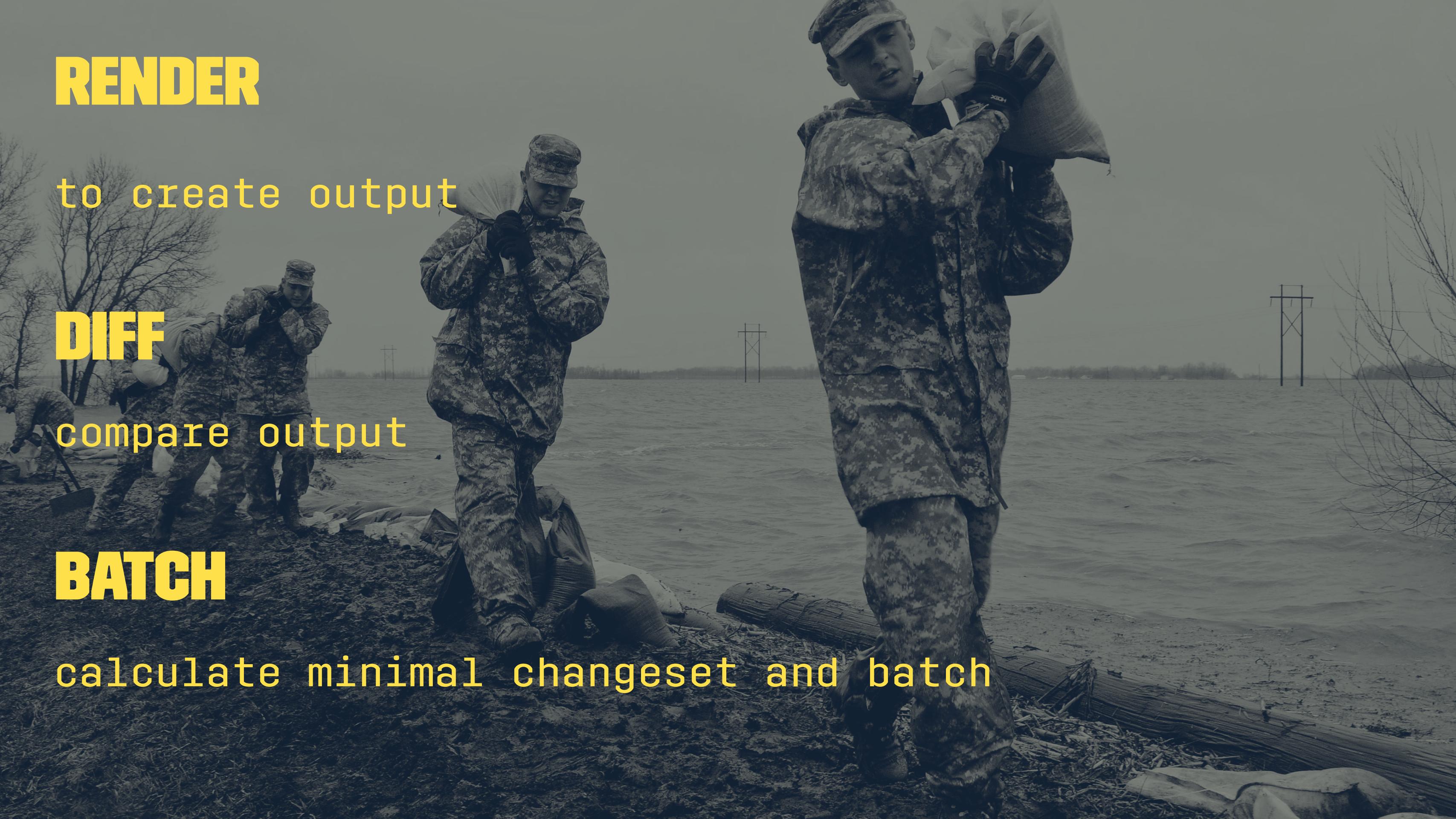
RE-BUILD THE DOM FOR EVERY RENDER, YOU SAY?
THIS IS MADNESS!



VIRTUAL DOM

VIRTUAL DOM

1. When something could have changed, re-render everything to a new DOM-representation
2. Diff new output with previous output
3. Update only what has changed



RENDER

to create output

DIFF

compare output

BATCH

calculate minimal changeset and batch

✖️ <top frame>

▼ Preserve log

```
> var li = document.querySelector('li');
li.ATTRIBUTE_NODE
```

```
isPrototypeOf
isSameNode
lang
lastChild
lastElementChild
localName
lookupNamespaceURI
lookupPrefix
matches
namespaceURI
nextElementSibling
nextSibling
nodeName
nodeType
nodeValue
normalize
offsetHeight
offsetLeft
offsetParent
offsetTop
offsetWidth
onabort
oncopy
oncontextmenu
oncut
ondrag
ondrop
onpaste
onbeforecut
onbeforepaste
onblur
oncancel
oncanplay
oncanplaythrough
```

HE AIN'T HEAVY, HE'S HTML

```
var ReactDOMElement = {  
  type : string,  
  props : { ... },  
  key : string | boolean | number | null,  
  ref : string | null  
};
```

ENZYME

```
it('renders a check mark when complete', () => {
  const item = { complete: true };
  const wrapper = shallow(<TodoItem item={item}>);
  expect(wrapper.find('.item-mark').text()).to.equal('✓');
});
```

REACT

SUMMARY:

- » GUI as declarative functions
- » JSX
- » Virtual DOM
- » High testability



Vise PIN

for XXXX XXXX XXXX 7046

-
- PIN-koden vil vises i 30 sekunder.
 - Hold PIN-koden hemmelig. Oppgi den aldri til andre, selv ikke banken eller politiet.
-

Kortets sikkerhetskode 

Vis PIN

[Avbryt](#)

[Prisliste](#)

Visa motiv (** 0227) BRUFSKONTO FRI ▾

```
import Dropdown from 'dropdown';

function MyComponent({ cards }) {
  return <div>
    ...
    <Dropdown
      label="Credit Cards"
      onChange={ () => ... }
    >
      { cards.map(toOption) }
    </Dropdown>
  }
}
```



Felles Frontend / ffe-dropdown-react

Source

[master](#) ... ffe-dropdown-react / README.md

Roger Andreassen committed 2948b9bef08 2 days ago

[Source view](#) [Diff to previous](#) [History](#)[Raw file](#)

ffe-dropdown-react

React implementation of the dropdown found in FFE.

Install

```
$ npm install --save ffe-dropdown-react
```

`ffe-dropdown-react` depends on `ffe-core`, `ffe-form`, `ffe-spinner` and `ffe-dropdown` being present and imported in your project. More specifically, the CSS classes related to dropdowns in those packages should be in your CSS Object Model when using this component.

Usage

```
import Dropdown from 'ffe-dropdown-react';

<Dropdown label={ labelTextString } onChange={ function }>
  { cards.map(c => <option value={ c.id }>{ c.name }</option>) }
</Dropdown>
```

The passed function will be called with the `SyntheticEvent` dispatched by `React`, allowing access to the `value` property of the selected option through `e.target.value`, where `e` is the argument passed.

```
<Dropdown label={ labelTextString } onChange={ function } onBlur={ anotherFunction }>
  { cards.map(c => <option value={ c.id }>{ c.name }</option>) }
</Dropdown>
```

The passed 'anotherFunction' will be called on blur event, ie when user tabs out of the select element. Same usage as the `onChange` property.

If you need to render a label tag outside of the component and pass an id for the select as a parameter you can pass an `id` attribute. The `labelAttribute` should then be omitted.

```
<label htmlFor={ uniqueIdString }>{ labelTextString }</label>
<Dropdown id={ uniqueIdString } onChange={ function }>
  { cards.map(c => <option value={ c.id }>{ c.name }</option>) }
</Dropdown>
```

If you want to initialize the dropdown with a non-empty value, you can supply a `defaultValue` prop.

To get an error message showing pass `errorMessage` to the `Dropdown`. Passing `invalid` to the `Dropdown` will give a red color on the element.

```
<Dropdown defaultValue="foo" invalid="true" errorMessage="test">
  <option value="make_foo" disabled="true">Not Bar</option>
  <option value="bar">Bar</option>
</Dropdown>
```



Repositories



Name

	eslint-config-ffe	PUBLIC
	eslint-config-ffe-base	PUBLIC
	ffe-accordion	PUBLIC
	ffe-accordion-react	PUBLIC
	ffe-buttons	PUBLIC
	ffe-buttons-react	PUBLIC
	ffe-cards	PUBLIC
	ffe-checkbox-react	PUBLIC
	ffe-core	PUBLIC
	ffe-datepicker	PUBLIC
	ffe-datepicker-react	PUBLIC
	ffe-details-list	PUBLIC
	ffe-dropdown	PUBLIC
	ffe-dropdown-react	PUBLIC
	ffe-expandable-react	PUBLIC
	ffe-form	PUBLIC
	ffe-formatters-js	PUBLIC

FFE PROBLEM 1:
Repositories

ENSURING PACKAGE CONSISTENCY

ffe-checkbox-react

PUBLIC

ffe-core

PUBLIC

ffe-datepicker

PUBLIC

ffe-datepicker-react

PUBLIC

ffe-details-list

PUBLIC

ffe-dropdown

PUBLIC

ffe-dropdown-react

PUBLIC

ffe-expandable-react

PUBLIC

ffe-form

PUBLIC

ffe-formatters-js

PUBLIC

Repositories

Name

 eslint-config-ffe	PUBLIC
 eslint-config-ffe-base	PUBLIC
 ffe-accordion	PUBLIC
 ffe-accordion-react	PUBLIC
 ffe-buttons	PUBLIC
 ffe-buttons-react	PUBLIC
 ffe-cards	PUBLIC
 ffe-checkbox-react	PUBLIC
 ffe-c	PUBLIC
 ffe-	PUBLIC
 r	PUBLIC
 s-list	PUBLIC
 opdown	PUBLIC
 opdown-r	PUBLIC
 e-expandable-react	PUBLIC
 ffe-form	PUBLIC
 ffe-formatters-js	PUBLIC

Repositories

Name

-  eslint-config-ffe PUBLIC
-  eslint-config-ffe-base PUBLIC
-  ffe-accordion PUBLIC
-  ffe-accordion-react PUBLIC
-  ffe-buttons PUBLIC
-  ffe-button-group PUBLIC
-  ffe-carousel PUBLIC
-  ffe-checkbox-react PUBLIC
-  ffe-core PUBLIC
-  ffe-datepicker PUBLIC
-  ffe-datepicker-react PUBLIC
-  ffe-details-list PUBLIC
-  ffe-dropdown PUBLIC
-  ffe-dropdown-react PUBLIC
-  ffe-expandable-react PUBLIC
-  ffe-form PUBLIC
- ffe-formatters-js PUBLIC

\$YO NEW-COMPONENT

FFE

PROBLEM 2:
Repositories

COMMUNICATING PACKAGE CONTENTS

ffe-core

PUBLIC

ffe-datepicker

PUBLIC

ffe-datepicker-react

PUBLIC

ffe-details-list

PUBLIC

ffe-dropdown

PUBLIC

ffe-dropdown-react

PUBLIC

ffe-expandable-react

PUBLIC

ffe-form

PUBLIC

ffe-formatters-js

PUBLIC

FFE

Repositories

Name

eslint-config-ffe

PUBLIC

eslint-config-ffe-base

PUBLIC

ffe-accordion

PUBLIC

ffe-accordion-react

PUBLIC

ffe-buttons

PUBLIC

ffe-buttons-react

PUBLIC

ffe-cards

PUBLIC

ffe-checkbox-react

PUBLIC

ffe-datepicker

PUBLIC

ffe-datepicker-react

PUBLIC

ffe-datetime

PUBLIC

ffe-datetime-react

PUBLIC

ffe-dropdown

PUBLIC

ffe-dropdown-react

PUBLIC

ffe-form

PUBLIC

EXAMPLES

RUNNING EXAMPLES IN EVERY PACKAGE



Repositories

Name

eslint-config-ffe

PUBLIC

eslint-config-ffe-base

PUBLIC

ffe-accordion

PUBLIC

ffe-accordion-react

PUBLIC

ffe-buttons

PUBLIC

ffe-buttons-react

PUBLIC

ffe-cards

PUBLIC

ffe-checkbox-react

PUBLIC

GEMINI

VISUAL REGRESSION TESTS

ffe-form

PUBLIC

ffe-formatters-js

PUBLIC



Roger Andreassen

master_2.0.5-ffe-input-label-styling → master

OPEN

Search for code or repositories...



Merge

...

DIG-11112 Styling nytt søknadsskjema kredittkort.

Overview Diff Commits

Find text in diff and context lines

visual-tests / baseline-screenshots / radioblock / plain / firefox-large.png MODIFIED

+3

...

- less
 - form-label.less
- visual-tests/baseline-screenshots
 - checkbox/plain
 - firefox-large.png
 - firefox-medium.png
 - firefox-small.png
 - dropdown/plain
 - firefox-large.png
 - firefox-medium.png
 - firefox-small.png
 - input/plain
 - firefox-large.png
 - firefox-medium.png
 - firefox-small.png
 - radioblock/plain
 - firefox-large.png
 - firefox-medium.png
 - firefox-small.png
 - radiobutton/plain
 - firefox-large.png
 - firefox-medium.png
 - firefox-small.png
 - radioswitch/plain
 - firefox-large.png
 - firefox-medium.png
 - firefox-small.png
 - tooltip/plain
 - firefox-large.png
 - firefox-medium.png
 - firefox-small.png

Two-up Blend Split Pixel diff

Old

Radio block

Alltid synlig kontekst

Hva ønsker du å gjøre?

Alternativ #1
Denne teksten er alltid synlig.

Alternativ #2
Denne teksten er alltid synlig.

Kontekst kun synlig for aktivt valg

Hva ønsker du å gjøre?

Block one
Denne teksten er synlig fordi valget er aktivt.

Block two

Radio block

Alltid synlig kontekst

Hva ønsker du å gjøre?

Alternativ #1
Denne teksten er alltid synlig.

Alternativ #2
Denne teksten er alltid synlig.

Kontekst kun synlig for aktivt valg

Hva ønsker du å gjøre?

Block one
Denne teksten er synlig fordi valget er aktivt.

Block two

FFE PROBLEM 3:
Repositories

ENSURING PACKAGE DISCOVERABILITY

ffe-cards

PUBLIC

ffe-checkbox-react

PUBLIC

ffe-core

PUBLIC

ffe-datepicker

PUBLIC

ffe-datepicker-react

PUBLIC

ffe-details-list

PUBLIC

ffe-dropdown

PUBLIC

ffe-dropdown-react

PUBLIC

ffe-expandable-react

PUBLIC

ffe-form

PUBLIC

ffe-formatters-js

PUBLIC

DESIGN SYSTEM

Bygget: 06.10.2016,
09:49:05
Core Versjon: 8.1.2

Oversikt

Oversikt
Typografi
Farger
Ikoner
Knapper
Tabeller
Lister
Skjema
Fremdrift
Media
Navigasjon
Meldinger
System-meldinger
Formatering
ffe-datepicker

Med SpareBank 1s Design System kan man lage digitale løsninger eller trykkmateriale som er samstemte med vår profil og merkevare.

IMPROVING THE
DESIGN GUIDE
DESIGN.TEST.DOMAIN.COM

[Vis kode ▾](#)

Description list (multicolumn)

RELATERTE PAKKER PÅ STASH: [LESS](#)

Dette er en liste der hvert punkt består av en betegnelse og en eller flere beskrivelser.

EKSEMPEL

Navn

Ola Nordmann

E-post

email@mail.no

Telefon

+47 555 555 55

+46 555 555 55

+45 555 555 55

Motto eller valgspråkOmnia mirari etiam tritissima. Omnia mirari etiam tritissima. Omnia
mirari etiam tritissima. Omnia mirari etiam tritissima.[Skjul kode ^](#)

```
<dl class="ffe-description-list ffe-description-list--multicolumn">
  <div class="ffe-description-list__avoid-break">
    <dt>Navn</dt>
    <dd>Ola Nordmann</dd>
  </div>
  <div class="ffe-description-list__avoid-break">
    <dt>Telefon</dt>
    <dd>+47 555 555 55<br/>+46 555 555 55<br/>+45 555 555 55</dd>
  </div>
  <div class="ffe-description-list__avoid-break">
    <dt>E-post</dt>
    <dd>email@mail.no</dd>
  </div>
  <div class="ffe-description-list__avoid-break">
    <dt>Motto eller valgspråk</dt>
    <dd>Omnia mirari etiam tritissima. Omnia mirari etiam tritissima. Omnia mirari etiam tritissima. Omnia mirari etiam tritissima.</dd>
  </div>
</dl>
```

[Vis kode ▾](#)

Description list (multicolumn)

RELATERTE PAKKER PÅ STASH: LESS

Dette er en liste der hvert punkt består av en betegnelse og en eller flere beskrivelser.

EKSEMPEL

Navn

Ola Nordmann

E-post

email@mail.no

Telefon

+47 555 555 55

+46 555 555 55

Motto eller valgspråkOmnia mirari etiam tritissima. Omnia mirari etiam tritissima. Omnia
mirari etiam tritissima. Omnia mirari etiam tritissima.

```
<div class="ffe-description-list ffe-description-list__multicolumn">
  <dt>Navn</dt>
  <dd>Ola Nordmann</dd>
</div>
<div class="ffe-description-list__avoid-break">
  <dt>Telefon</dt>
  <dd>+47 555 555 55<br/>+46 555 555 55<br/>+45 555 555 55</dd>
</div>
<div class="ffe-description-list__avoid-break">
  <dt>E-post</dt>
  <dd>email@mail.no</dd>
</div>
<div class="ffe-description-list__avoid-break">
  <dt>Motto eller valgspråk</dt>
  <dd>Omnia mirari etiam tritissima. Omnia mirari etiam tritissima. Omnia mirari etiam tritissima. Omnia mirari etiam tritissima.</dd>
</div>
</dl>
```

AUTOUPDATING DESIGN GUIDE

```
function main() {
    npm run build

    if should_publish then
        npm run has-published -s || npm publish
        bob ci job build --jobname design-system_build_deploy
    fi
}

function should_publish() {
    [[${GIT_BRANCH} =~ ^*(origin/)?master$ ]]
}
```

WHAT THIS ACHIEVES

- » easy to create new components
- » easy to discover existing components
- » easy to update existing components
- » improved dev-design GUI discussions
- » ensuring consistent design

(even with home-made macgyver-solution)

PAUSE

**SO REACT MAKES OUR VIEWS, BUT
WHAT ELSE
IS THERE?**



Live React

Hot Reloading with Time Travel

@dan_abramov

0:17 / 30:40

Up Next

Dan Abramov - The Redux Journey at react-europe 2016
ReactEurope
18,699 views
26:57

Diablo 2 Soundtrack (Full)
VGameOSTs
Recommended for you
1:24:05

Jorn - Dio (Full Album)
Hernan Achor
Recommended for you
1:06:24

Game of Thrones + Skyrim MASHUP
Jonathan Young
Recommended for you
3:51

React.js Conf 2016 - Jared Forsyth - Redux, Re-frame, Facebook Developers
14,545 views
21:32

Dan Abramov - Live React: Hot Reloading with Time Travel at react-europe 2015

ReactEurope

Subscribe 7,114

179,584 views

2,656 likes 9 dislikes

Add to Share More

Published on Jul 5, 2015

React's unique strength is bringing to JavaScript development some of the benefits previously exclusive to more radically functional languages such as Elm and ClojureScript, without forcing you to completely eschew local state or rewrite code with exclusively immutable data structures. In this talk, Dan will demonstrate how React can be used together with Webpack Hot Module Replacement to create a live editing environment

SHOW MORE

Филипп Андрейчев. Беседы о react и redux
Михаил Кривушин
2,795 views
1:14:09

Joe Satriani - Soloing
Jorge Romo
Recommended for you
7:16

▶ todos: 1 item

✓ What needs to be done?



Use Redux

1 item left

All

Active

Com



Dan Abramov
@dan_abramov



Following

Oh no am I really writing my own Flux library

RETWEETS

12

LIKES

41



8:41 PM - 29 May 2015



12



41



Reply to @dan_abramov



Juho Vepsäläinen @bebraw · 29 May 2015

@dan_abramov Writing a Flux library is the hello world of React developers?



5



10



Dan Abramov @dan_abramov · 29 May 2015

@bebraw I might be late to the party though



...



Pete Hunt @floydophone · 29 May 2015

@dan_abramov wait wait contribute to mine @bebraw



...



Dan Abramov @dan_abramov · 29 May 2015

@floydophone @bebraw You're kidding right? You don't have one. At least I didn't see it..



...

[View other replies](#)



Pete Hunt @floydophone · 29 May 2015

@dan_abramov not exactly flux :) [@bebraw](https://github.com/petehunt/minim...)



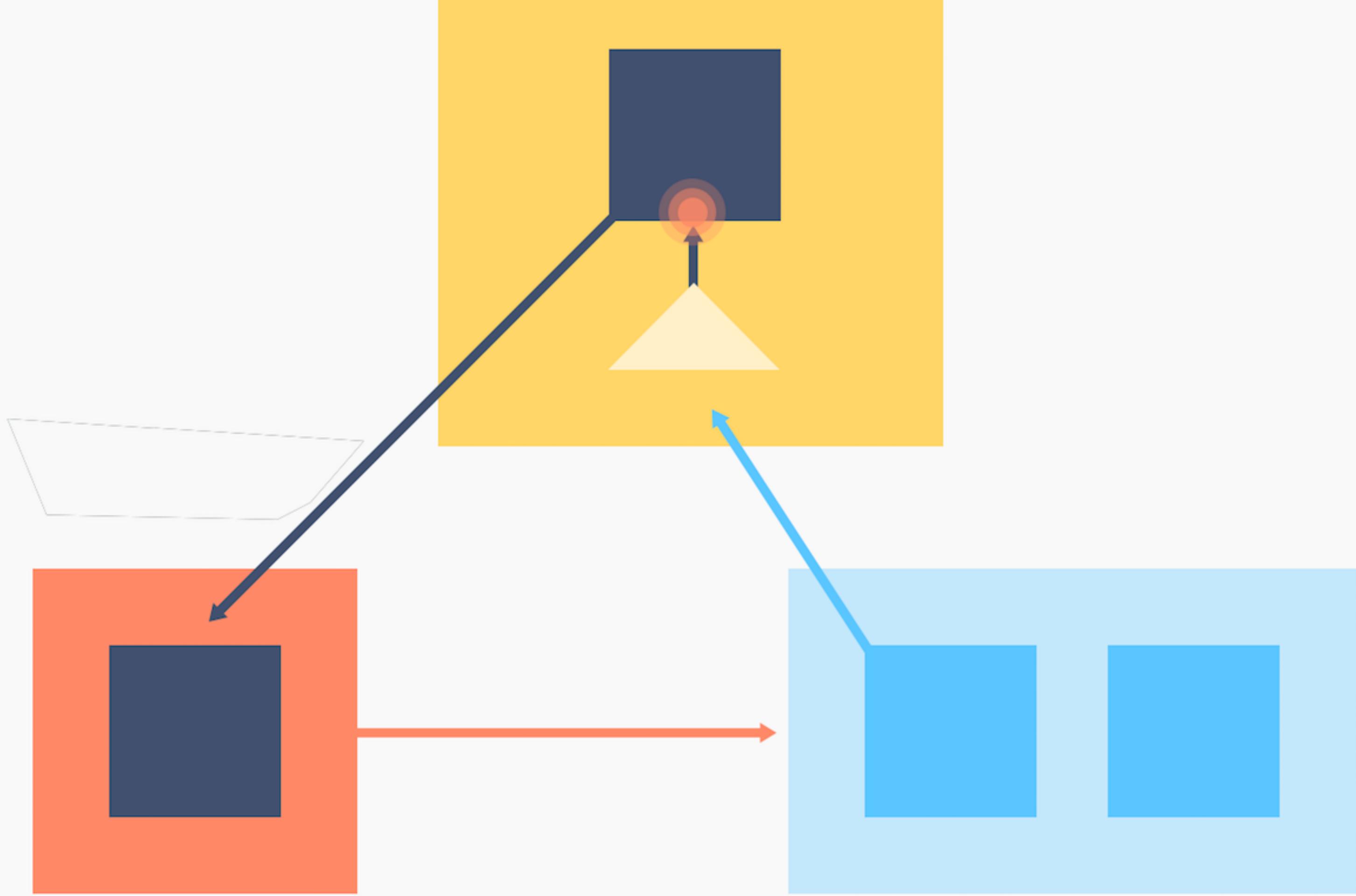
[Code](#)[Issues 40](#)[Pull requests 11](#)[Pulse](#)[Graphs](#)

Predictable state container for JavaScript apps <http://redux.js.org>

[2,035 commits](#)[9 branches](#)[49 releases](#)[331 contributors](#)Branch: [master](#) ▾[New pull request](#)[Create new file](#)[Upload files](#)[Find file](#)[Clone or download](#) ▾ timdorr 3.6.0

Latest commit 3e114f8 2 days ago

 [.github](#) Update ISSUE_TEMPLATE.md 3 months ago [build](#) Update Gitbook version 27 days ago [docs](#) Add link to UsageWithReactRouter 2 days ago [examples](#) Port examples to use Create React App (#1883) 4 days ago [logo](#) Switch out non-ASCII quotes for ASCII versions. (#1874) a month ago [src](#) Revert "throw if getState, subscribe, or unsubscribe called while dis... a month ago [test](#) Revert "throw if getState, subscribe, or unsubscribe called while dis... a month ago [.babelrc](#) Close #1687, Replace es3ify with Babel ES3 transforms (#1688) 4 months ago [.editorconfig](#) editorconfig: do not trim trailing whitespaces in Markdown files 7 months ago [.eslintignore](#) Don't lint examples (#1928) 3 days ago [.eslintrc](#) Bump eslint version 11 months ago [.gitignore](#) Add ES Modules build 7 months ago [.travis.yml](#) Also test node v6 on travis 4 months ago [CHANGELOG.md](#) ractk -> reactjs (see #1392) 7 months ago



THE GIST

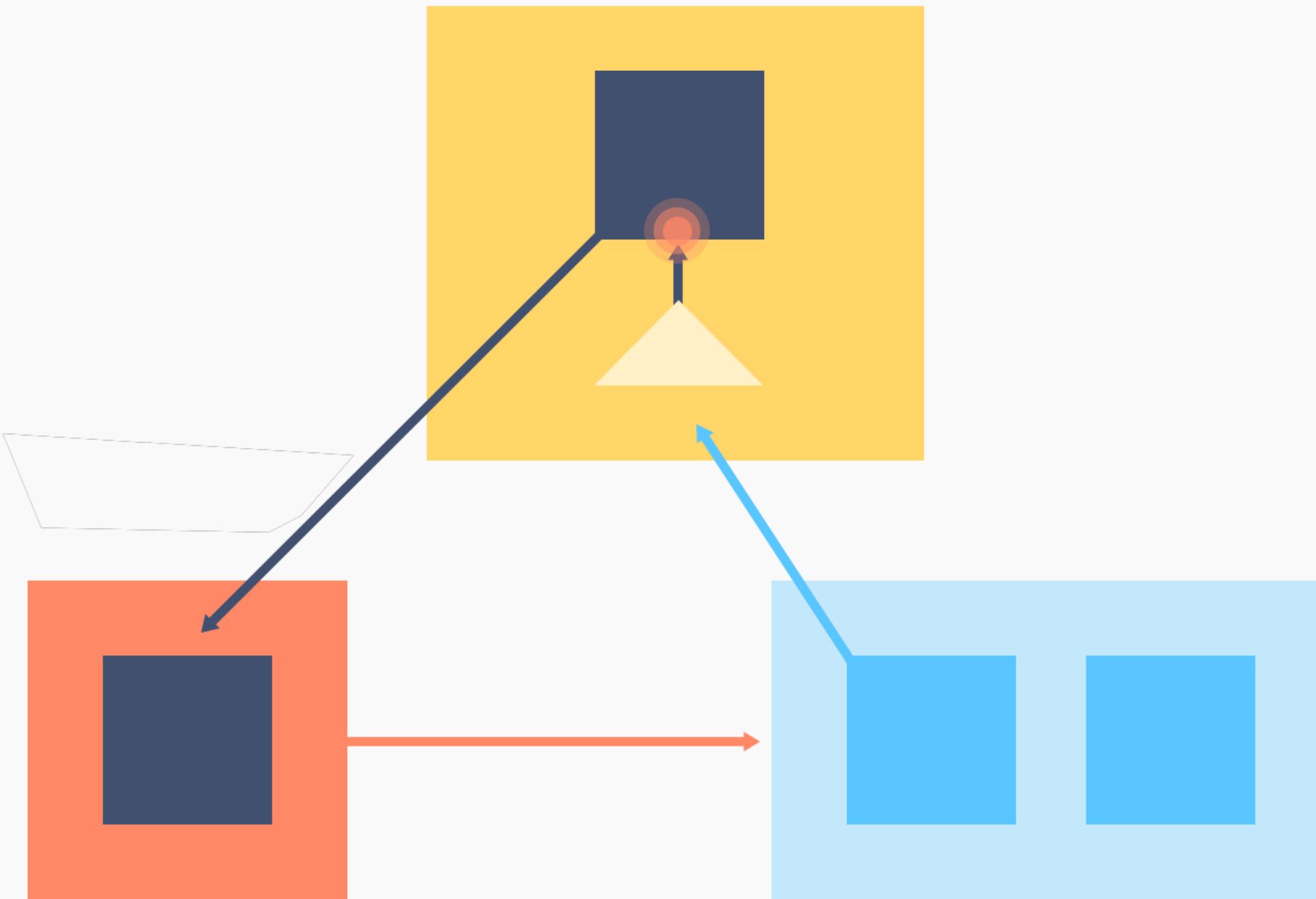
- » The whole state of your app is stored in an object tree inside a single store.
- » The only way to change the state tree is to emit an action, an object describing what happened.
- » To specify how the actions transform the state tree, you write pure reducers.

THE GIST

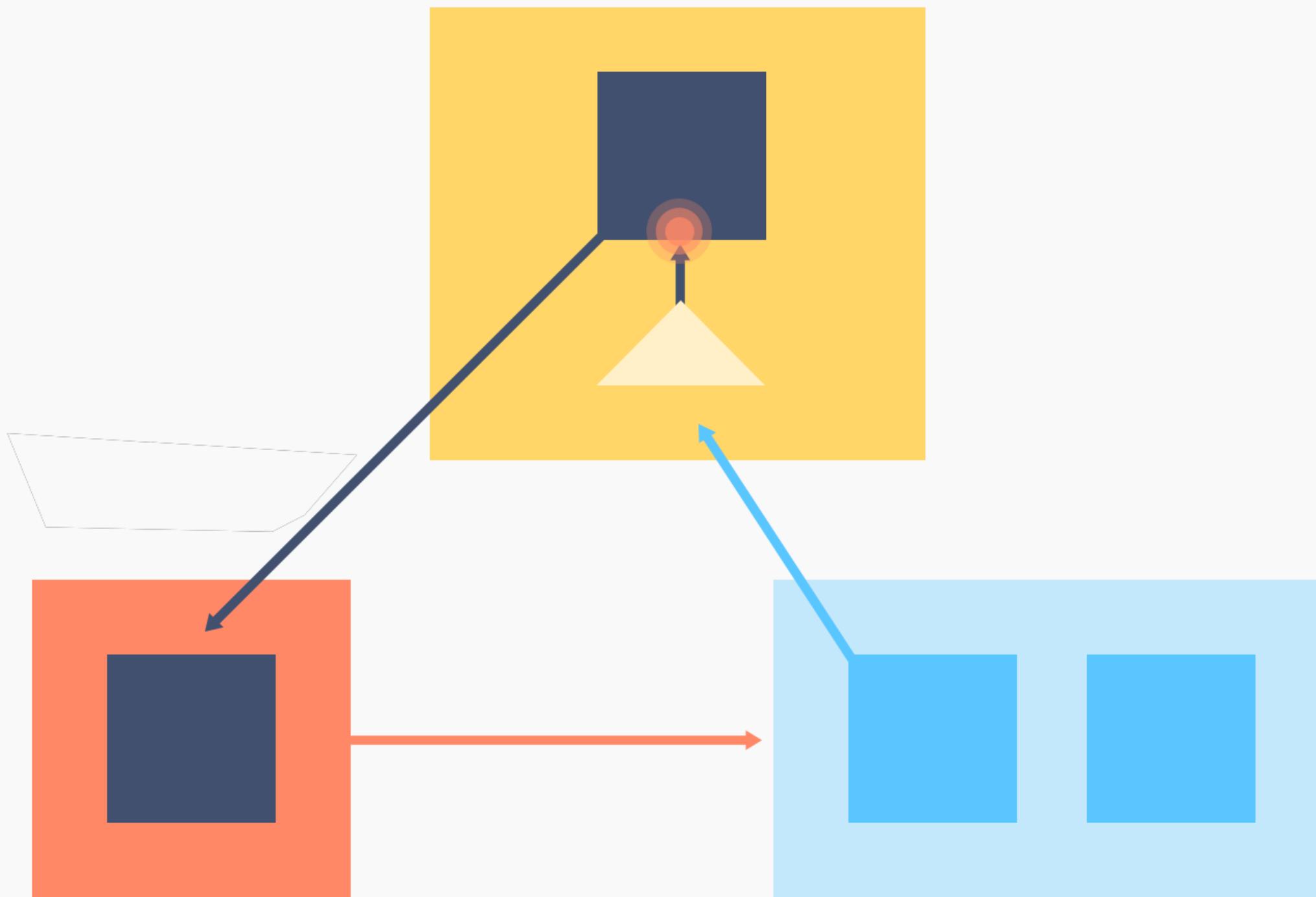
- » The whole state of your app is stored in an object tree inside a single store.
- » The only way to change the state tree is to emit an action, an object describing what happened.
- » To specify how the actions transform the state tree, you write pure reducers.

THE GIST

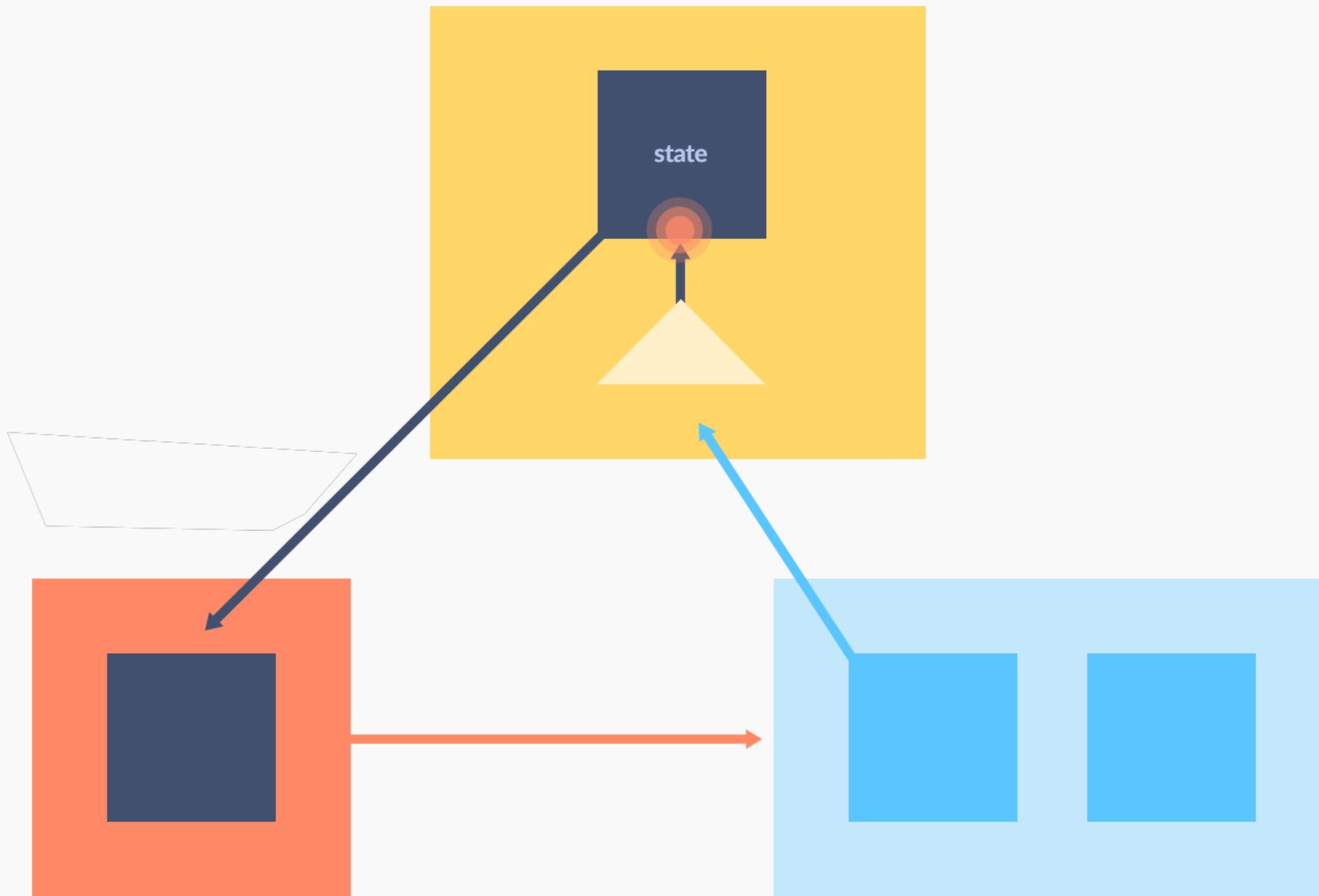
- » The whole state of your app is stored in an object tree inside a single store.
- » The only way to change the state tree is to emit an action, an object describing what happened.
- » To specify how the actions transform the state tree, you write pure reducers.



Store



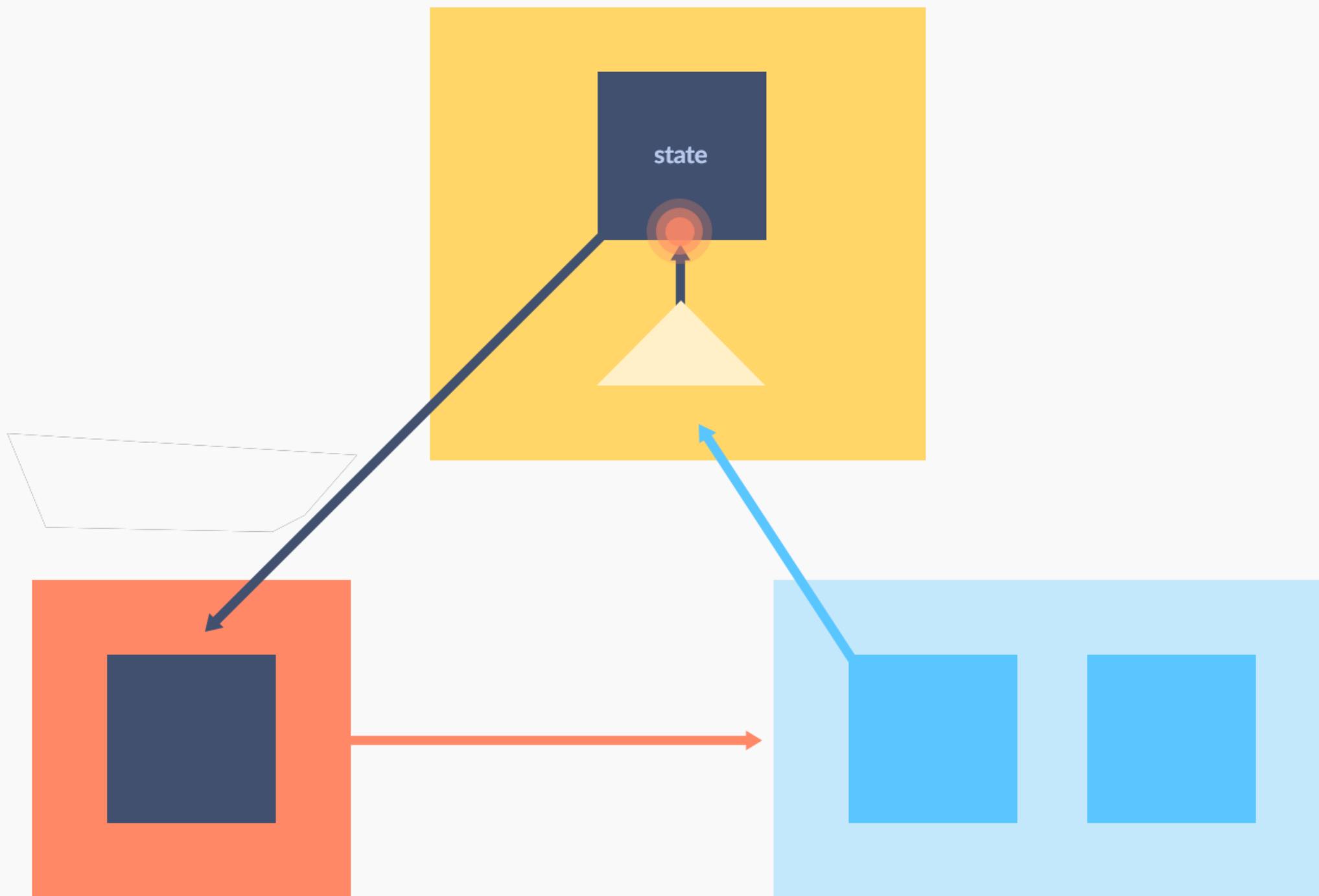
Store



STATE

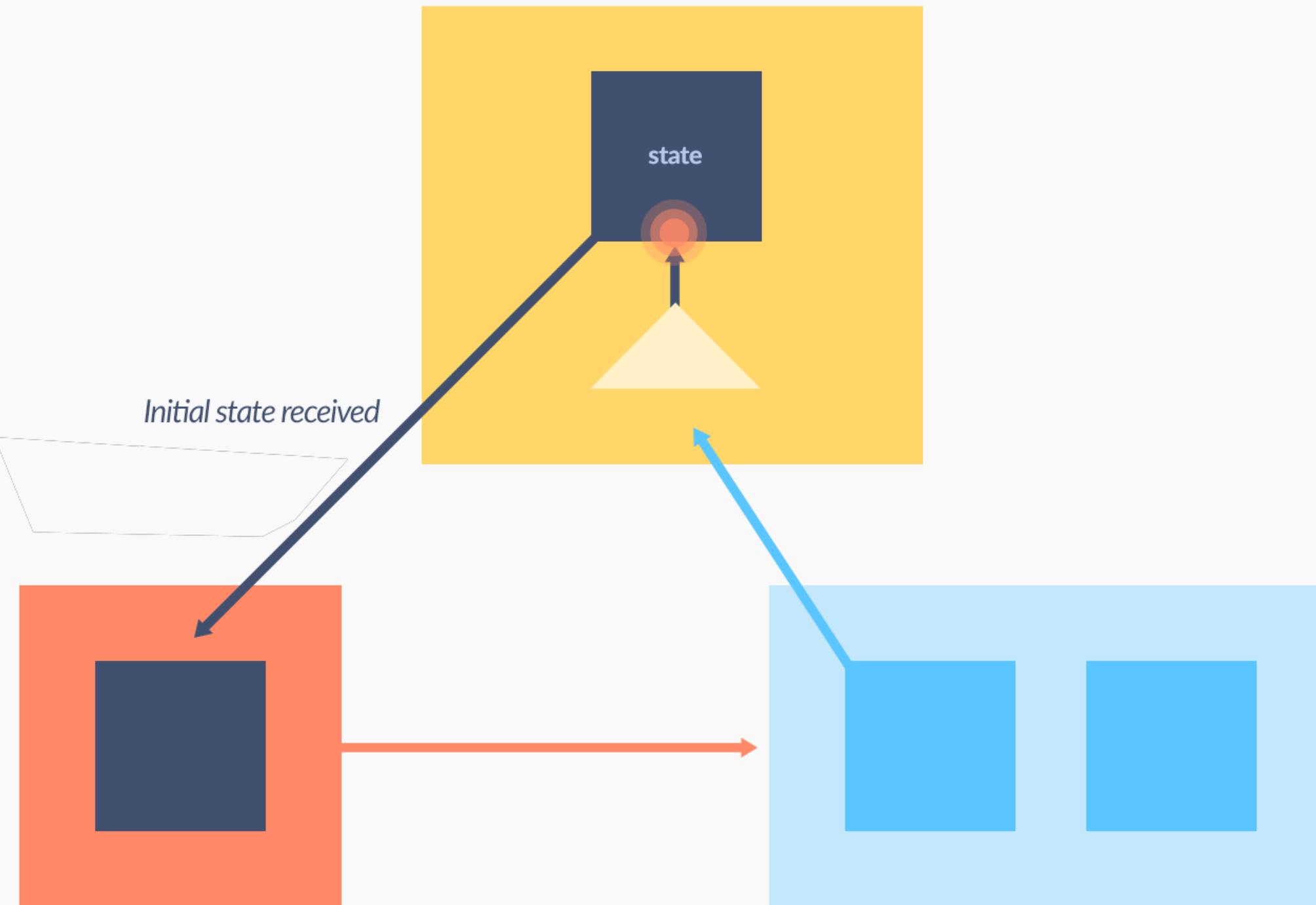
```
{  
  todos: [],  
  activeFilter: 'completed',  
}
```

Store



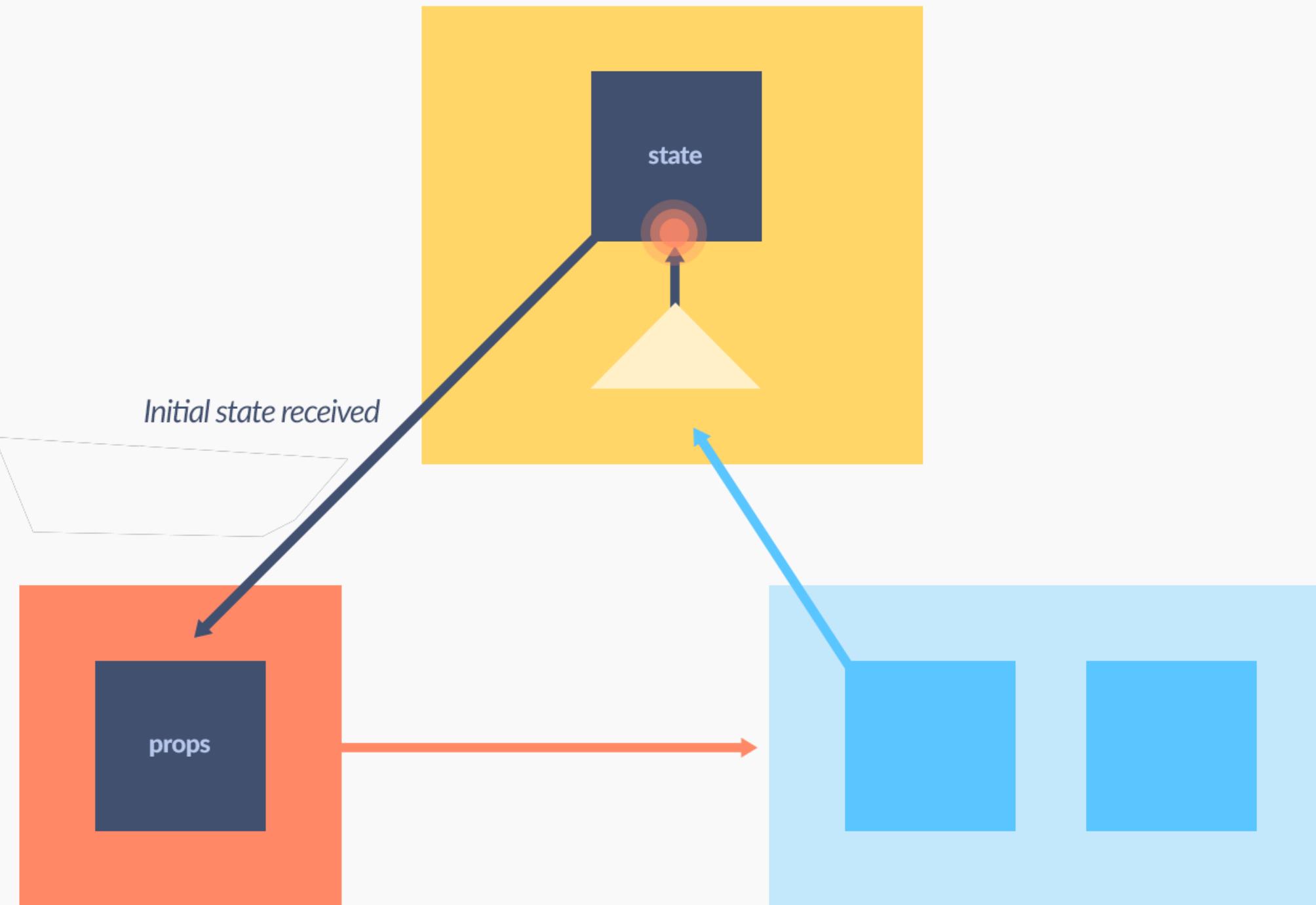
Component

Store



Component

Store



Component

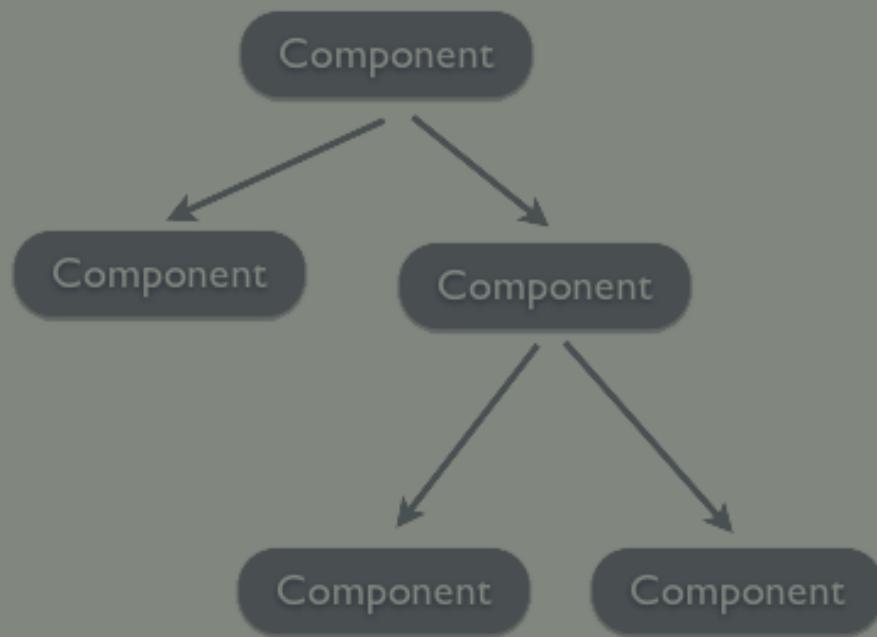
```
{  
  todos: [],  
  activeFilter: 'completed',  
}  
  
//
```

```
function mapStateToProps(state) {  
  return {  
    todoCount: state.todos.length  
  }  
}
```

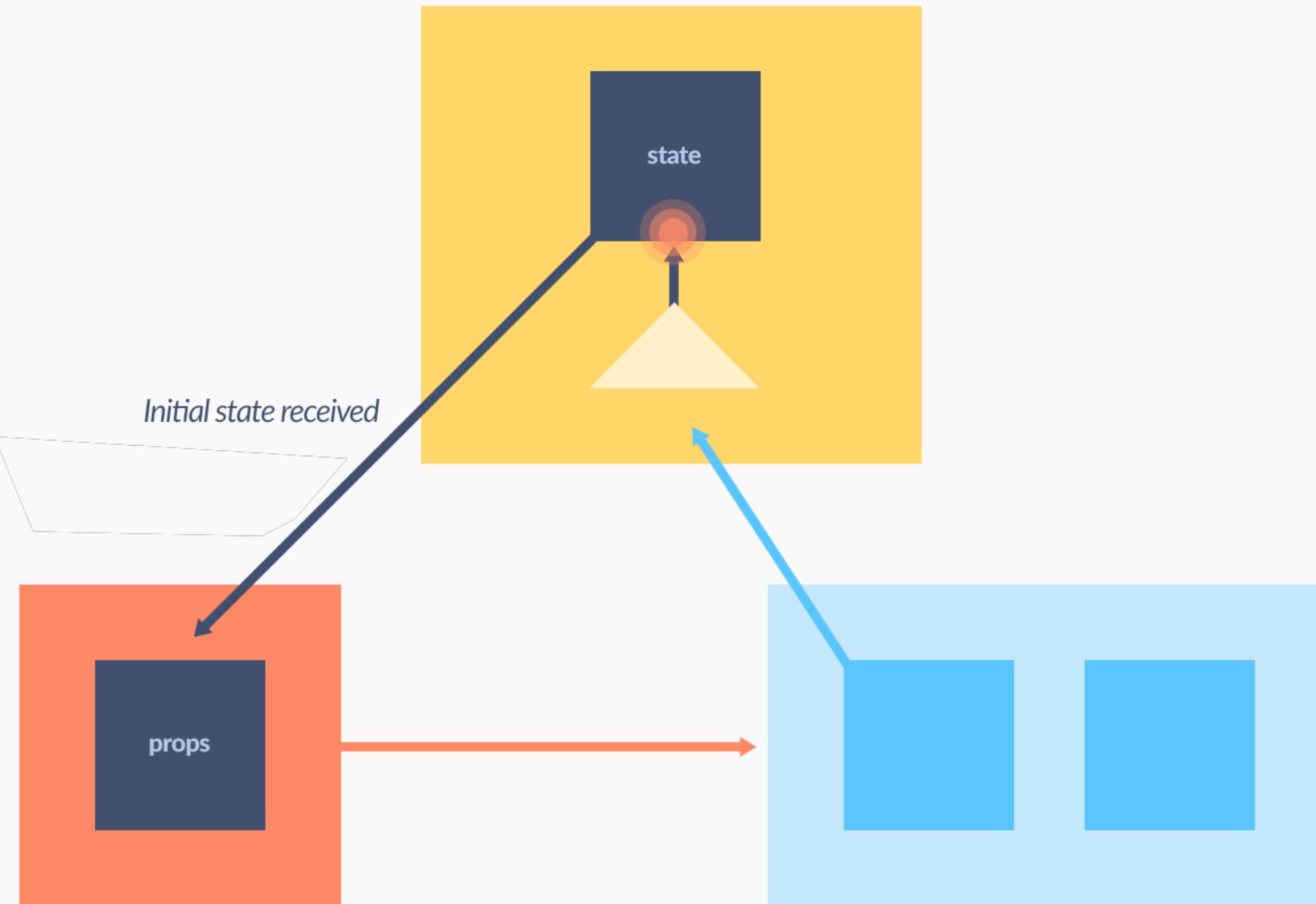
```
function MyComponent() { ... }

export default connect(mapStateToProps)(MyComponent);
```

SMART VS DUMB COMPONENTS

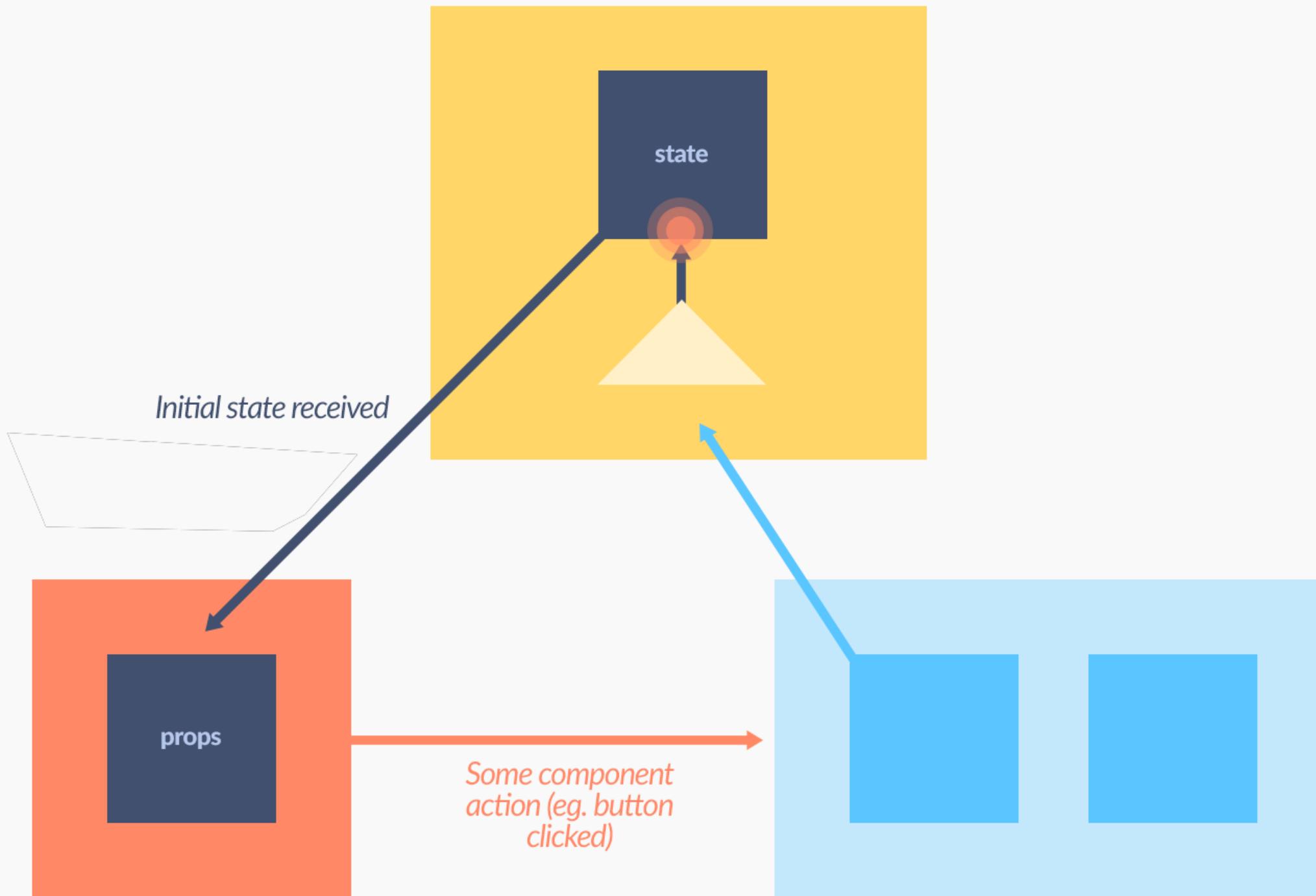


Store

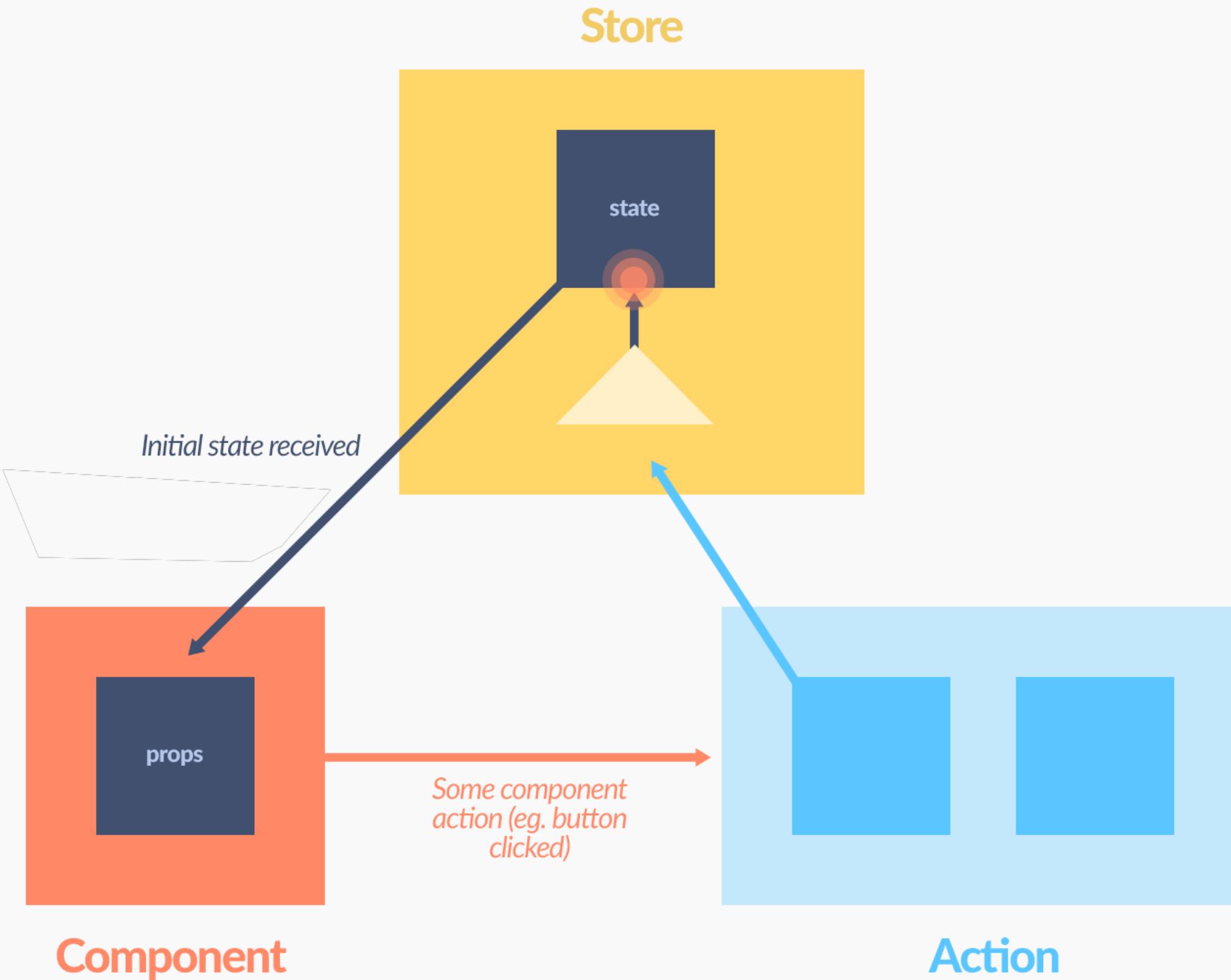


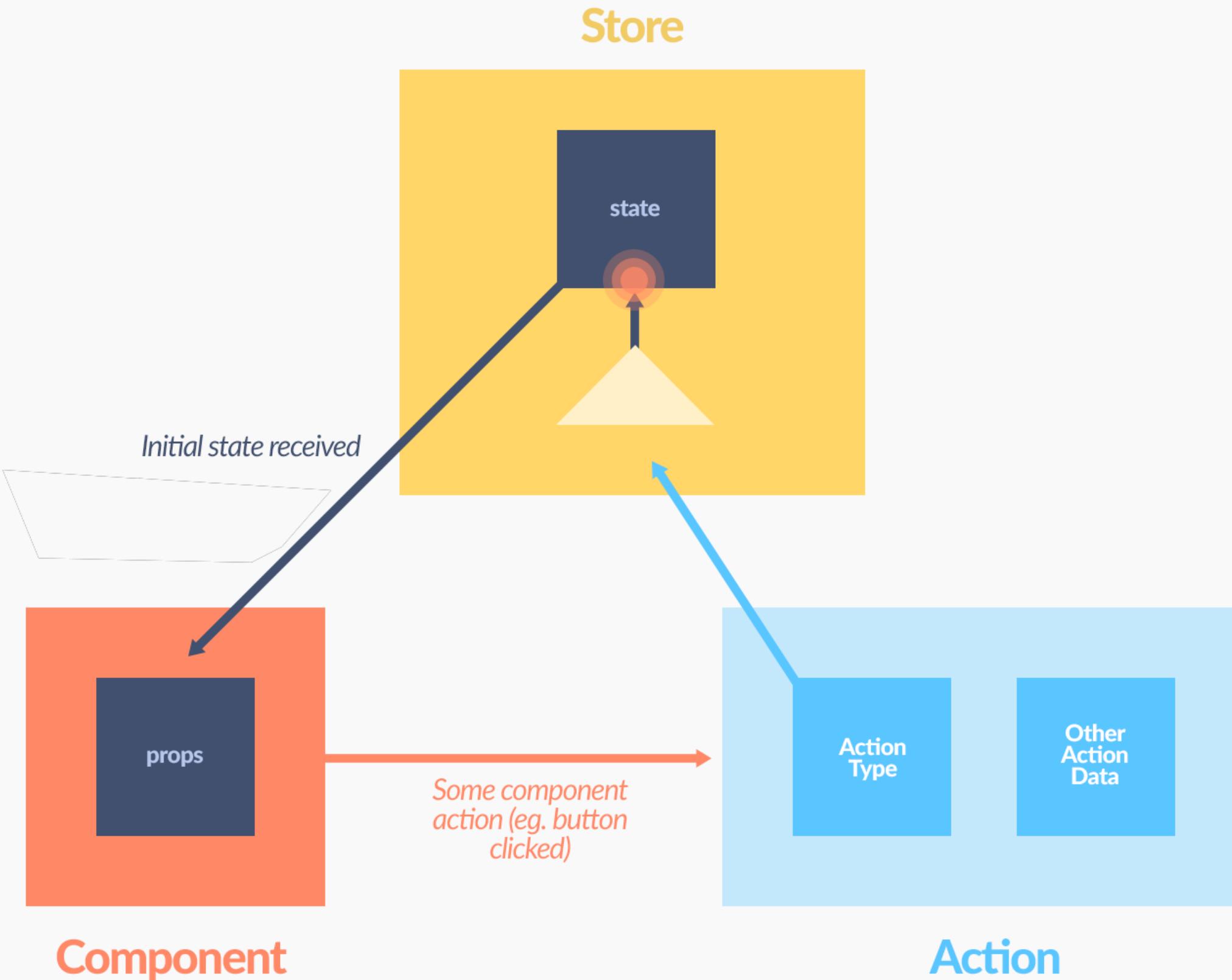
Component

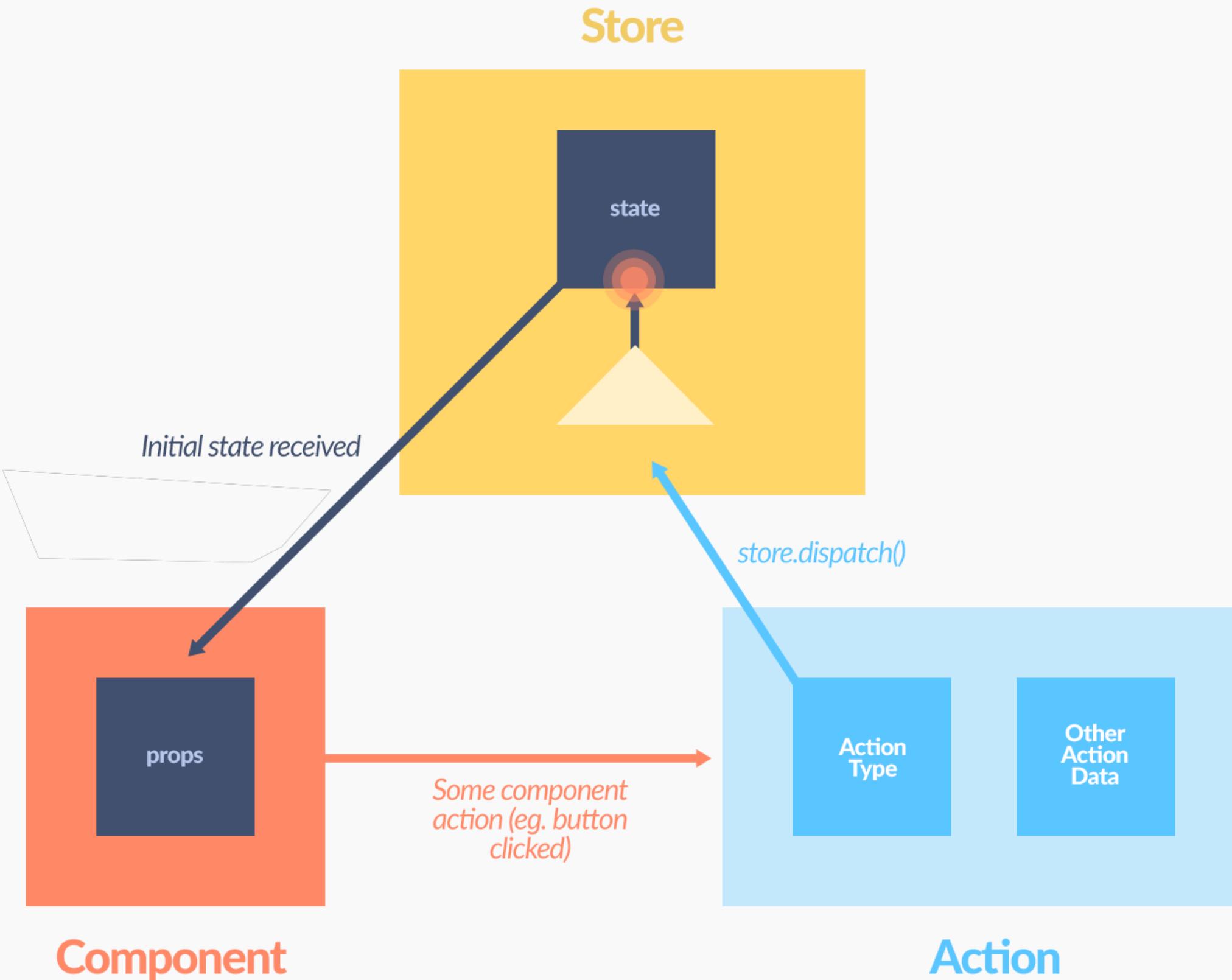
Store

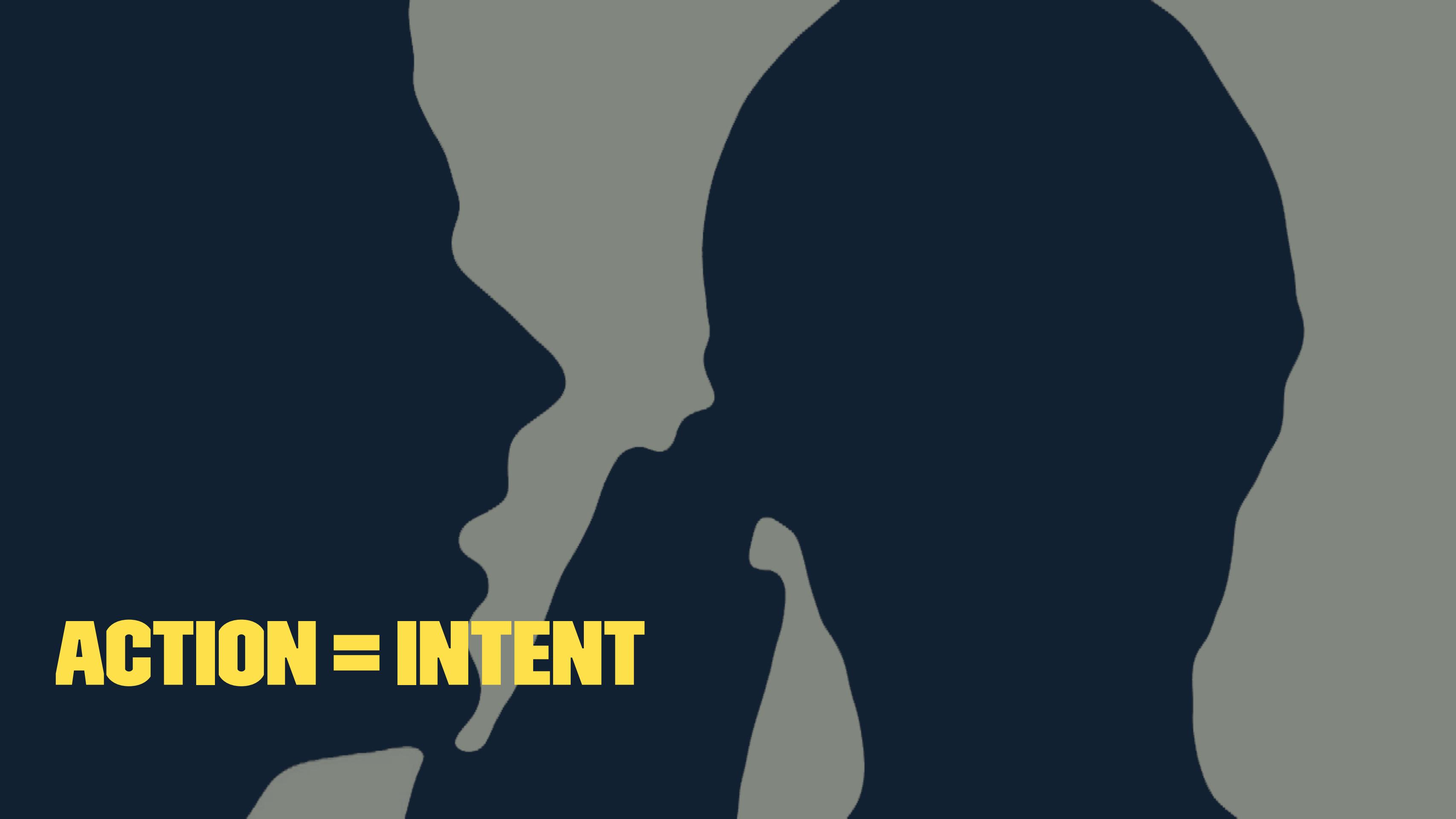


Component







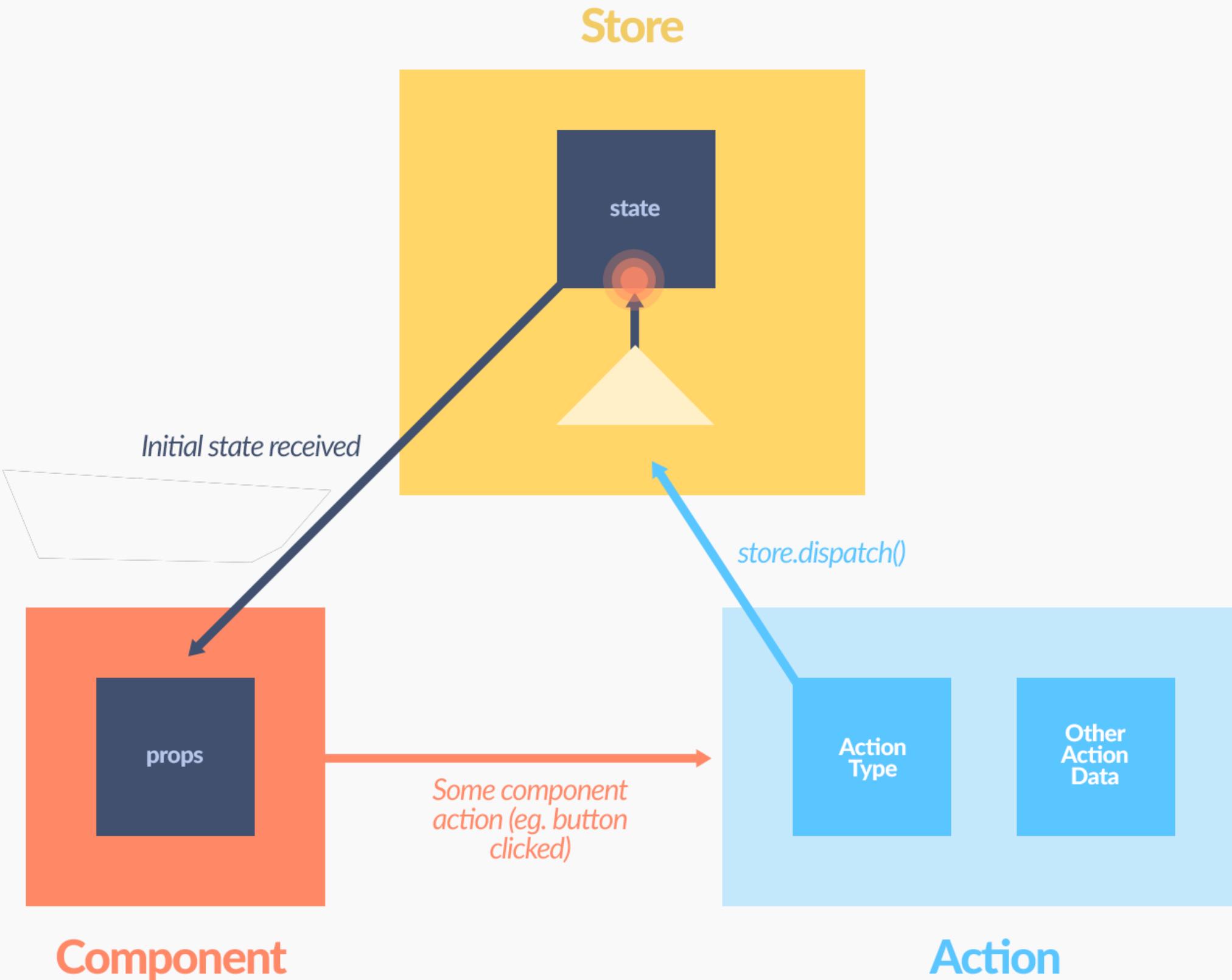


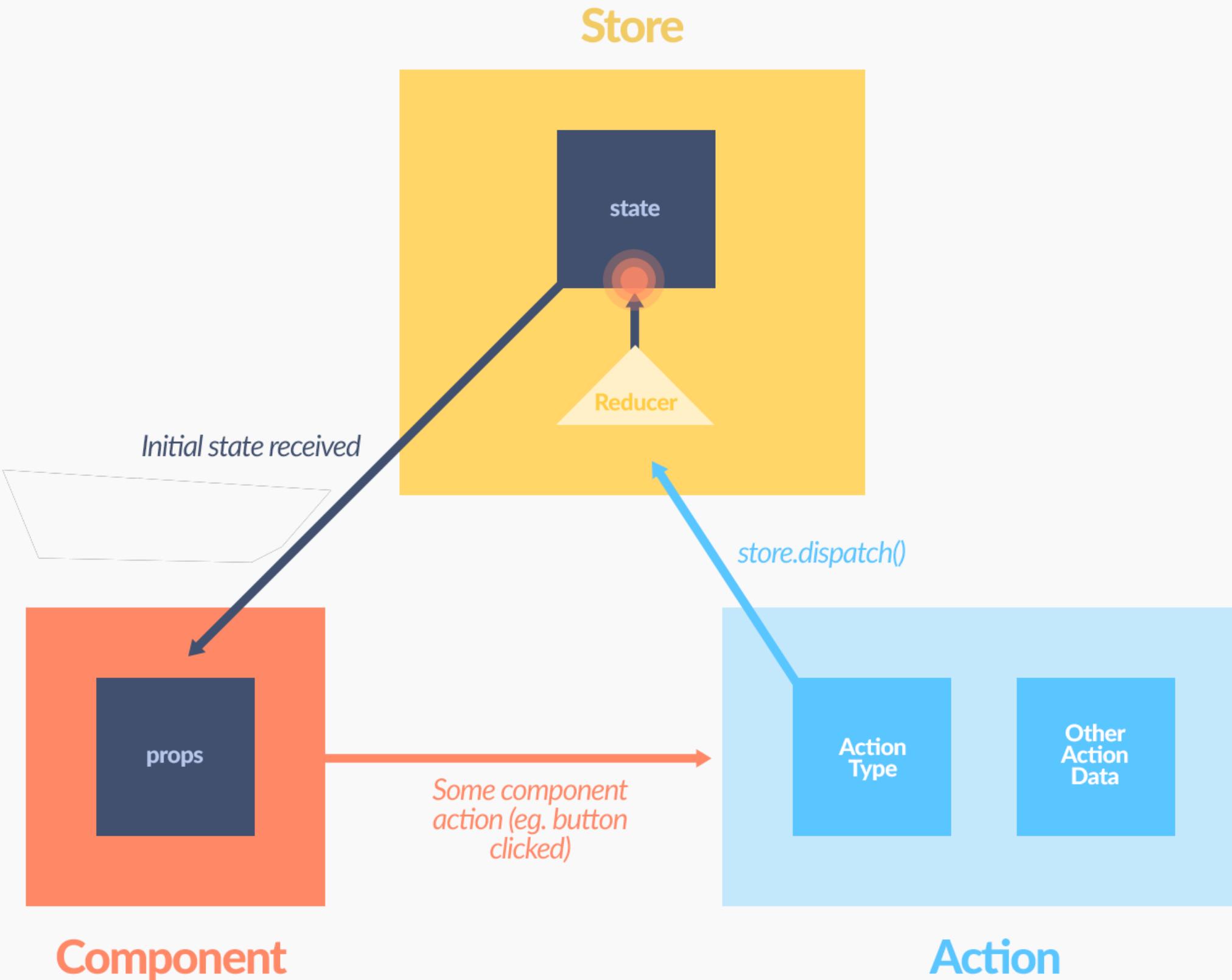
ACTION = INTENT

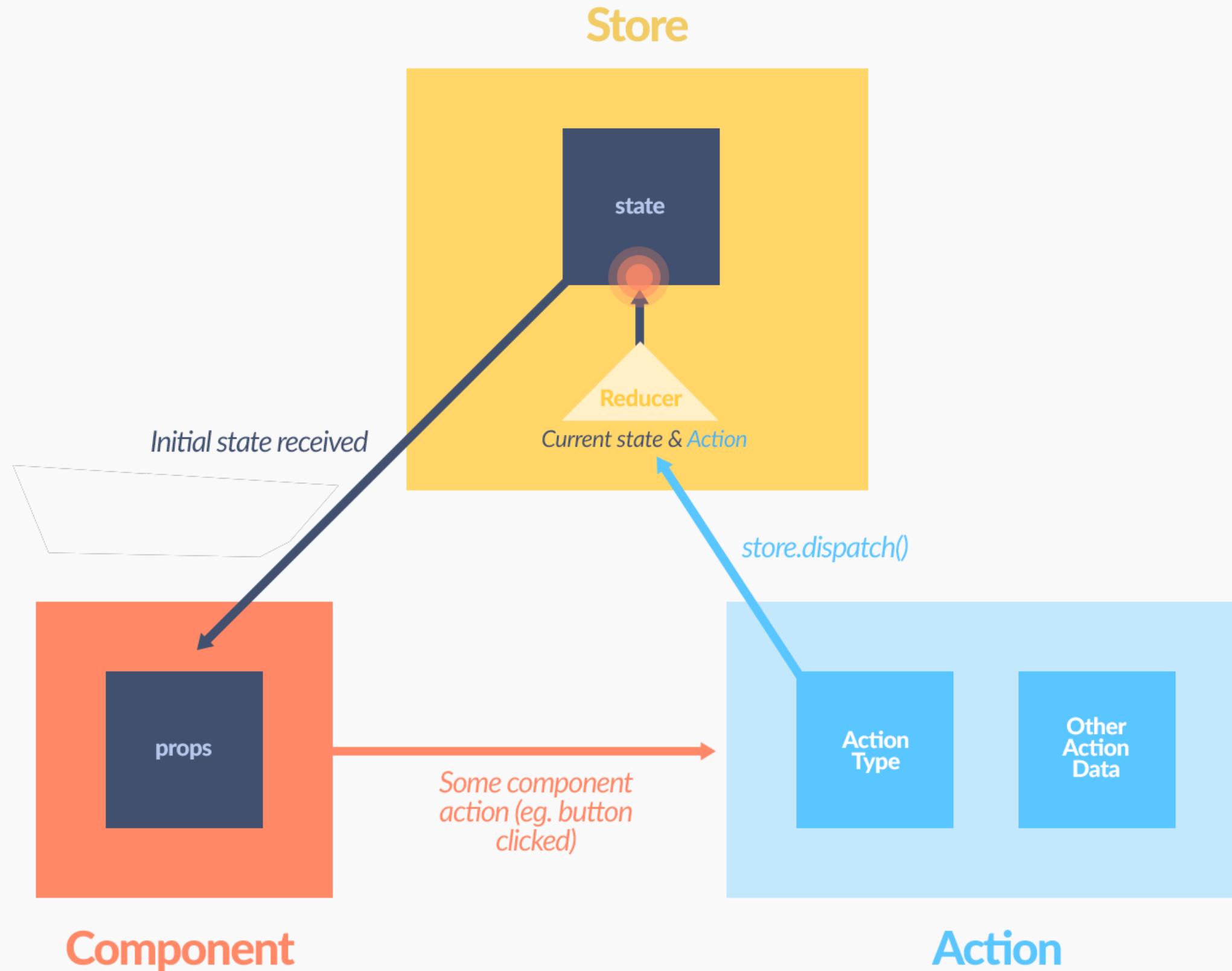
```
{  
  // action type  
  type: 'USER_CLICKED_BUY_BUTTON',  
  // action data  
  productId: 1  
}
```

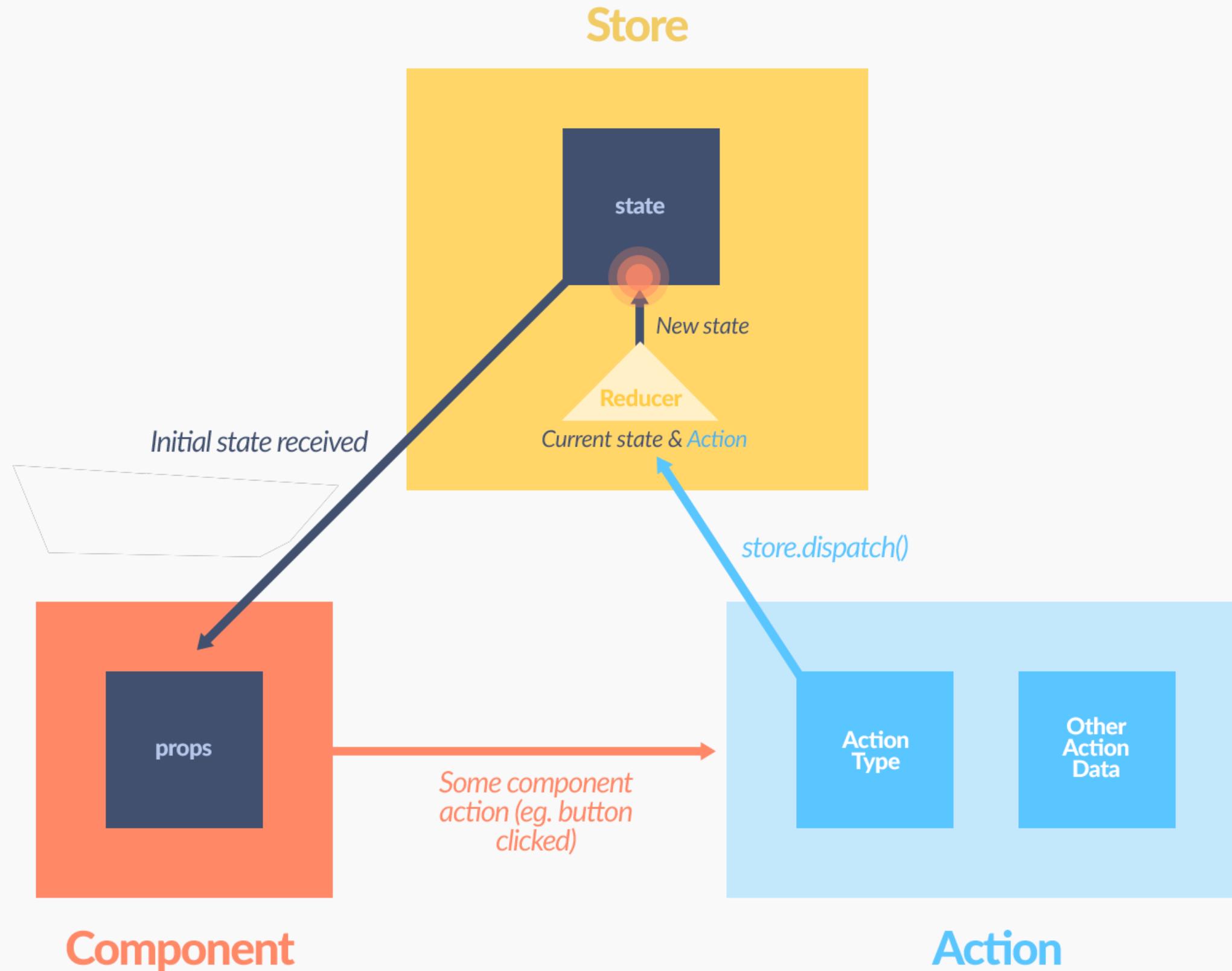
```
store.dispatch({  
  type: 'USER_CLICKED_BUY_BUTTON' ,  
  productId: 1  
})
```

```
function userClickedBuyButton(productId) {  
  const action = {  
    type: 'USER_CLICKED_BUY_BUTTON',  
    productId  
  };  
  
  return action;  
}  
  
store.dispatch(userClickedBuyButton(1));
```









REDUCER

STATE, ACTION → STATE

REDUCER

```
function reducer(state, action) {  
  ...  
  return newState;  
}
```

REDUCER

```
function reducer(state=0, action){  
  if (action.type === 'INCREMENT_CLICKED') {  
    return state + 1;  
  }  
  return state;  
}
```

REDUCER & IMMUTABILITY

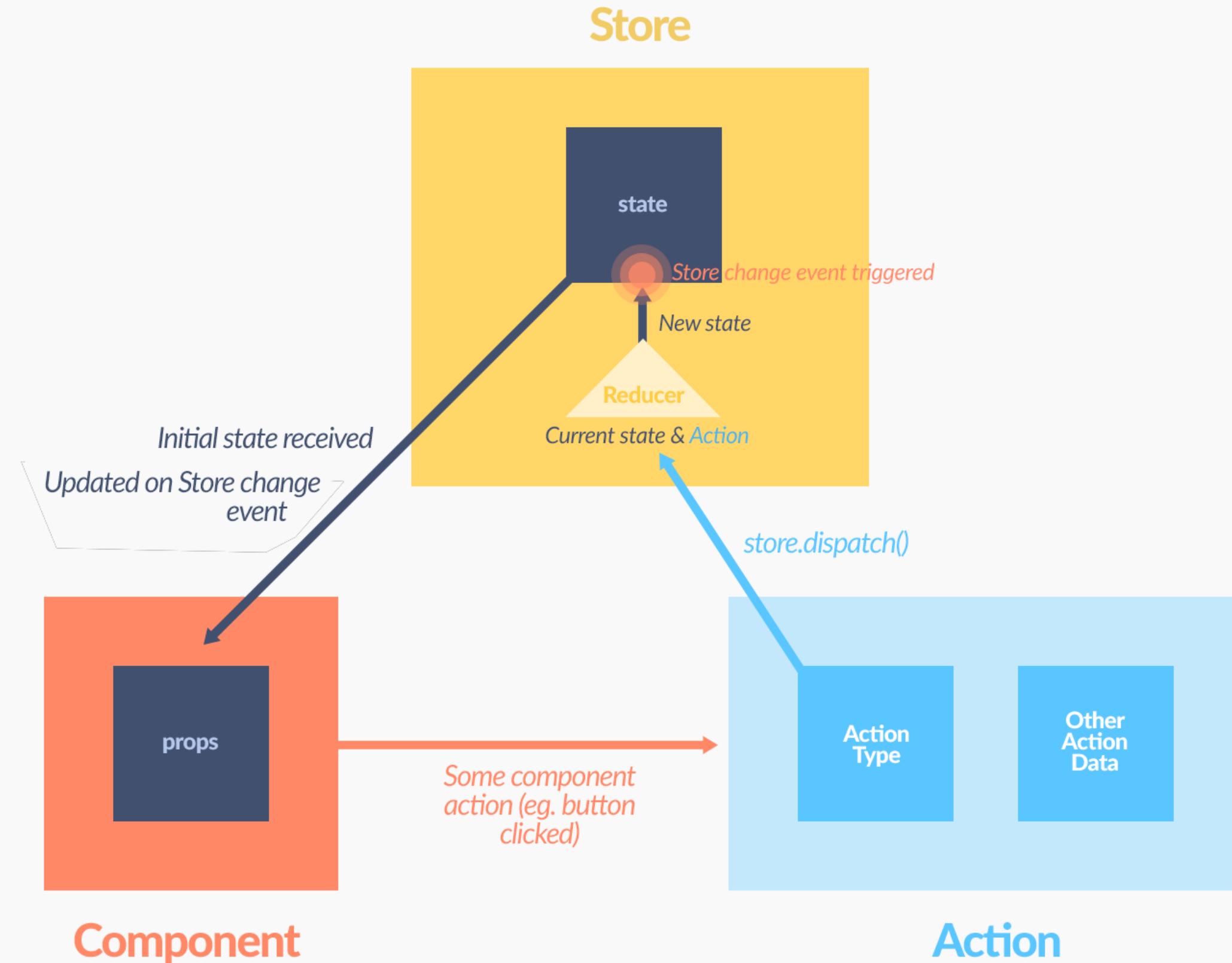
```
function reducer(state={ count: 0 }, action){  
  if (action.type === 'INCREMENT_CLICKED') {  
    // galt  
    state.count = state.count + 1;  
    return state;  
    // riktig  
    return { count: state.count + 1 }  
  }  
  return state;  
}
```

REDUCER & IMMUTABILITY

```
function reducer(state=[], action){  
  if (action.type === 'ADD_TODO_CLICKED') {  
    //galt  
    return state.push(action.todo);  
    // riktig  
    return [ ...state, action.todo];  
  }  
  return state;  
}
```

COMBINEREDUCERS

```
const rootReducer = combineReducers( {  
    todos: todoReducer,  
    filters: filterReducer,  
    user: userReducer  
} );
```



**TIME FOR A
BREAK**

STORE

```
{  
  todos: [ . . . ] ,  
  activeFilter: . . . ,  
  user: { . . . } ,  
  . . .  
  counter: 0  
}
```

COMPONENT

```
function CounterComponent({ counter }) {
  return (
    <div>
      Button was pressed { counter } times.
      <button>
        Increase
      </button>
    </div>
  );
}
```

CONNECTING THE COMPONENT

```
const mapStateToProps = state => {
  return {
    counter: state.counter;
  }
}
```

```
export default connect(mapStateToProps)(CounterComponent);
```

USING THE CONNECTED COMPONENT

```
function MyApplication() {  
  return <div>  
    <CounterComponent />  
  </div>;  
}
```

SIMPLIFYING WITH ES6

```
const mapStateToProps = state => ({  
  counter: state.counter;  
})
```

```
export default connect(mapStateToProps)(CounterComponent);
```

HANDLING CHANGES IN THE SYSTEM

```
const mapStateToProps = state => ({  
  counter: state.counter;  
});
```

```
const mapDispatchToProps = dispatch => ({  
  handleIncrease: () => dispatch({ type: 'INCREASE_CLICKED' })  
});
```

```
export default connect(mapStateToProps)(CounterComponent);
```

PASSING EVENT HANDLER TO COMP.

```
function CounterComponent({ counter, handleIncrease }) {  
  return (  
    <div>  
      Button was pressed { counter } times.  
      <button onClick={handleIncrease}>  
        Increase  
      </button>  
    </div>  
  );  
}
```

HANDLING THE ACTION

```
function reducer(state=0, action){  
  if (action.type === 'INCREMENT_CLICKED') {  
    return state + 1;  
  }  
  return state;  
}
```

ADDING OUR NEW REDUCER

```
const rootReducer = combineReducers({  
  todos: todoReducer,  
  filters: filterReducer,  
  user: userReducer,  
  counter: counterReducer  
});
```

REDOUX

SUMMARY:

- » all data in store
- » presentational logic in views
- » trigger changes in system with actions
- » control state change in pure reducers
- » sagas control asynchronosity and side-effects

REDUX

SUMMARY:

Store - views - actions - reducers - sagas

- » incredible testability (simple functions)
- » sideeffects are handled
- » explicit state and state changes
- » tidy architecture

**"NESTE ÅR KOMMER DET BARE
ENDA ET NYTT RAMMEVERK"**

DESIRED QUALITIES

- » easily understandable code
- » easily testable code
- » easily modularizable code
- » good developer experience code
- » code that can withstand time

DESIRED QUALITIES

- » easily understandable code ✓
- » easily testable code ✓
- » easily modularizable code ✓
- » good developer experience code ✓
- » code that can withstand time ✓

RESUME?

"EVERYONE" IS USING REACT/REDUX



Type Inference

Flow uses type inference to find bugs even without type annotations. It precisely tracks the types of variables as they flow through your program.

Idiomatic JS

Flow is designed for JavaScript programmers. It understands common JavaScript idioms and very dynamic code.

Realtime Feedback

Flow incrementally rechecks your changes as you work, preserving the fast feedback cycle of developing plain JavaScript.

Flow can catch common bugs in JavaScript programs before they run, including:

- silent type conversions,
- `null` dereferences,
- and the dreaded `undefined` is not a function.

```
1 // @flow
2 function foo(x) {
3     return x * 10;
4 }
5 foo('Hello, world!');
```

[show Flow output](#)

Flow also lets you **gradually** add type assertions to your code:

```
1 // @flow
2 function bar(x): string {
3     return x.length;
4 }
5 bar('Hello, world!');
```

[show Flow output](#)

FAST, RELIABLE, AND SECURE DEPENDENCY MANAGEMENT.

[GET STARTED](#)[INSTALL YARN](#)[!\[\]\(a8e730b080af54f42475b2e896db0828_img.jpg\) Star 13,200](#)

Ultra Fast.

Yarn caches every package it downloads so it never needs to again. It also parallelizes operations to maximize resource utilization so install times are faster than ever.





THANKS!

TWITTER: @EWNDL