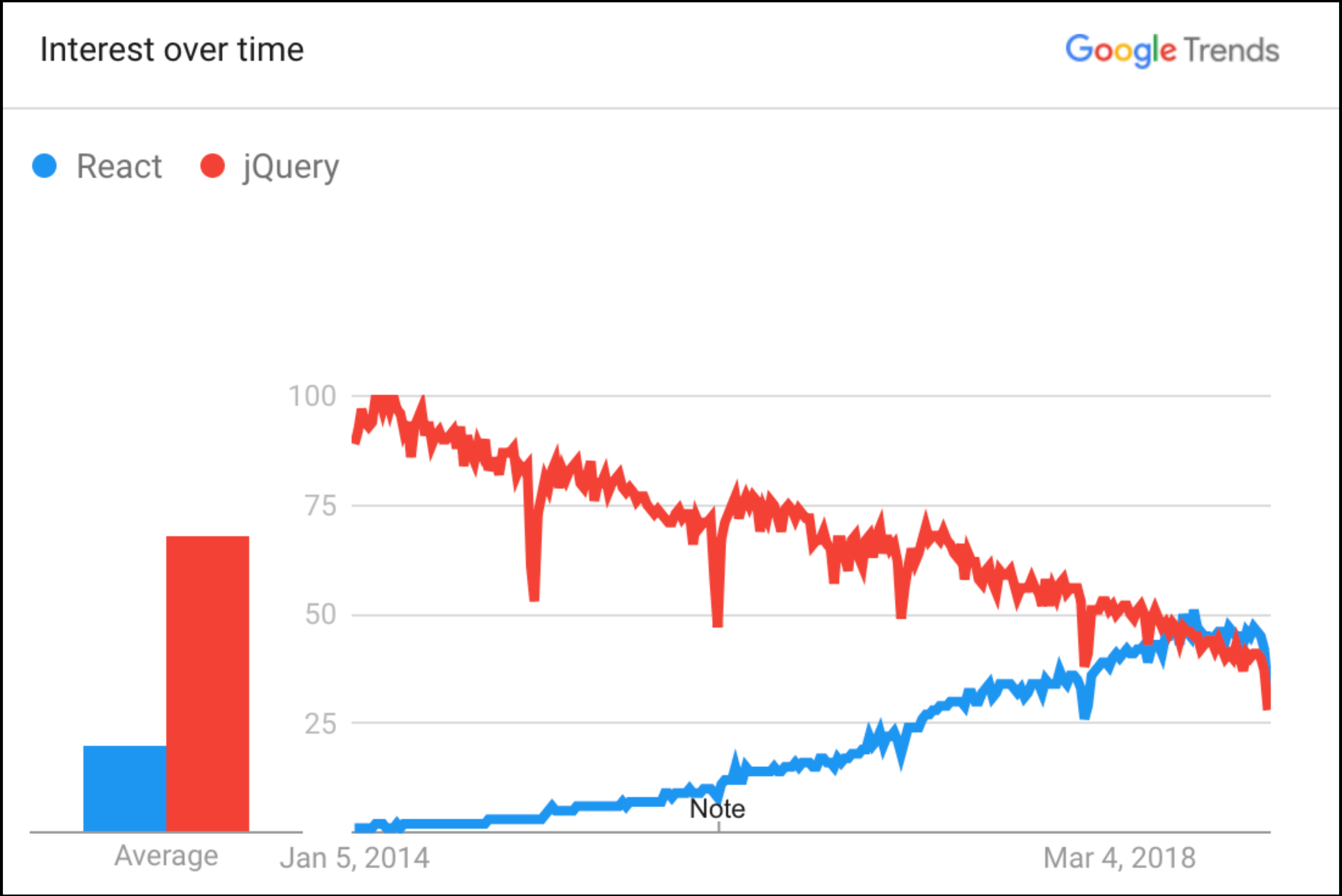
React 4 years later, React vs Vue

Tor Arne Kvaløy
FINN småjobber TDE


Interest over time
Google Trends
● React
● jQuery
100
75
50
25
Note
Average
Jan 5, 2014
Mar 4, 2018

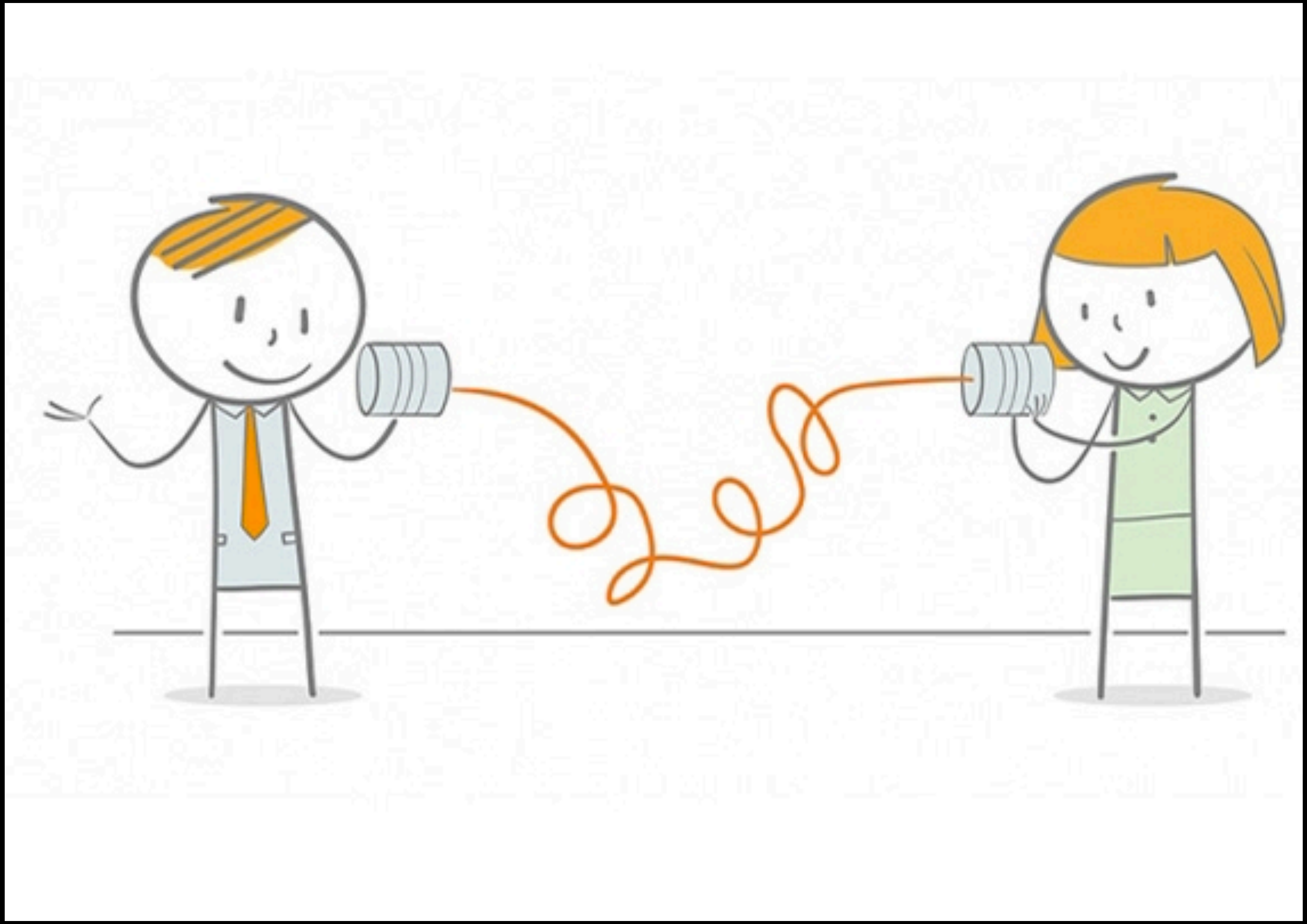
Imperative (jQuery)

```
const div = $("div")
if(show) {
  div.append($("span").text("Hello"))
}
```

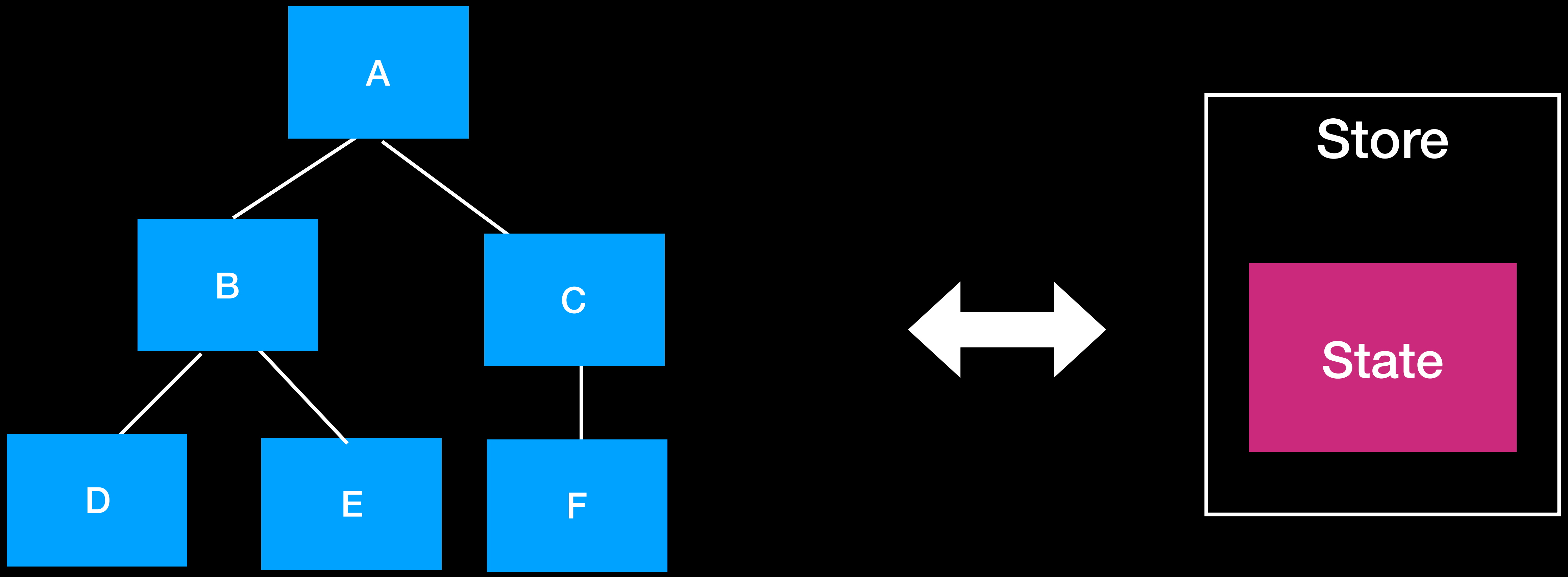
Declarative (React)

```
<div>
  {show &&
   <span>Hello</span>
  }
</div>
```

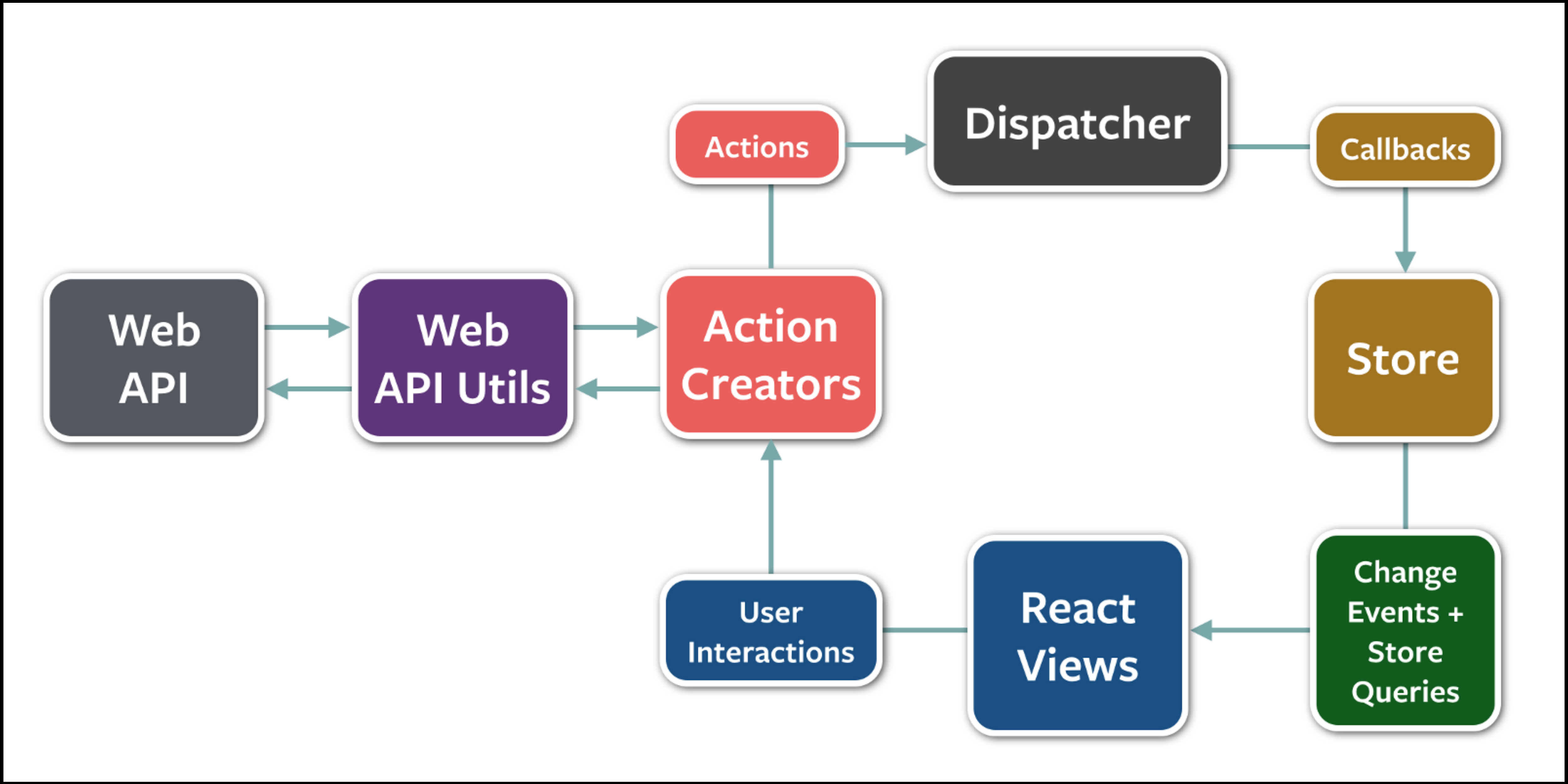
The problem



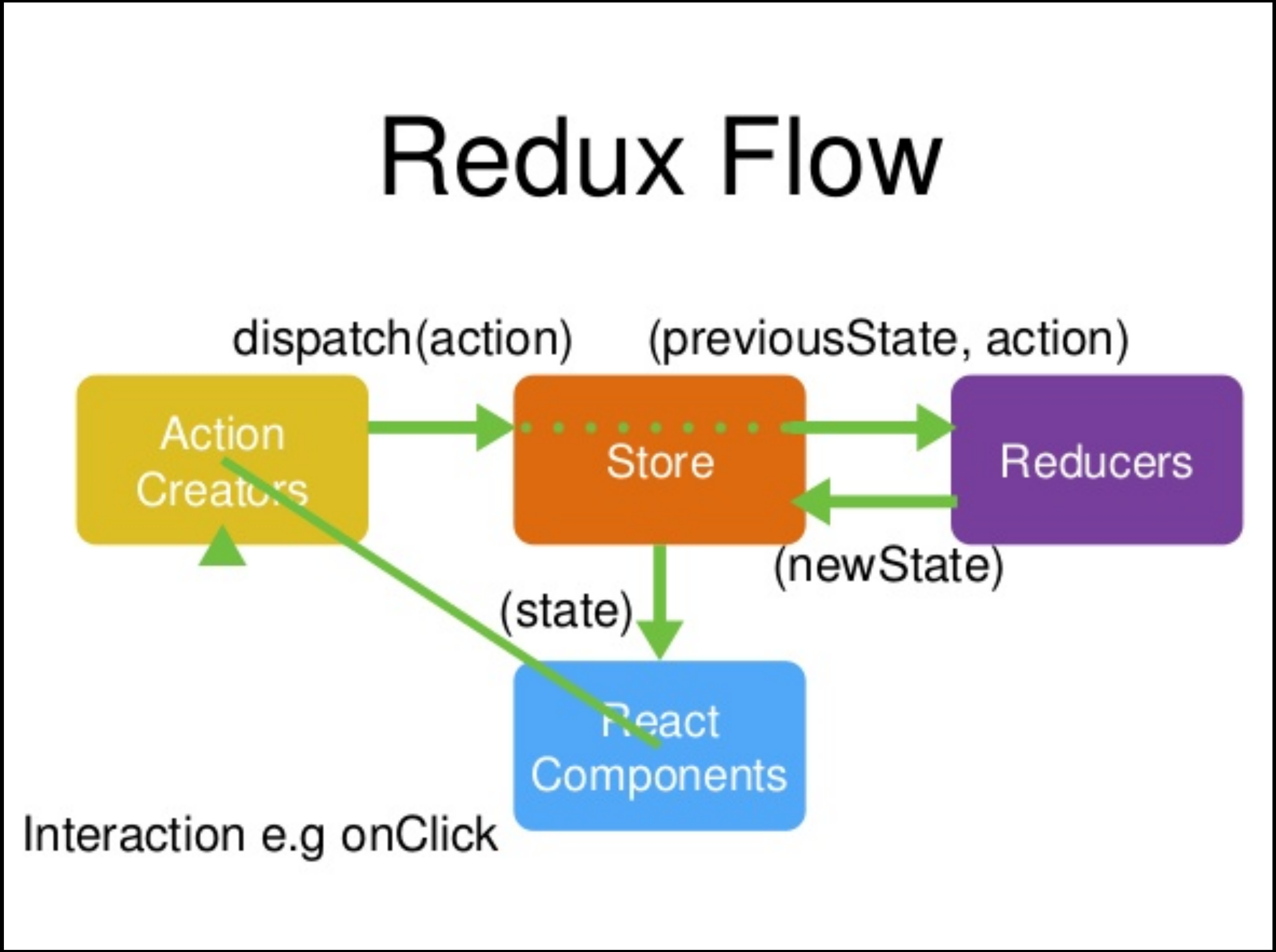
Communication

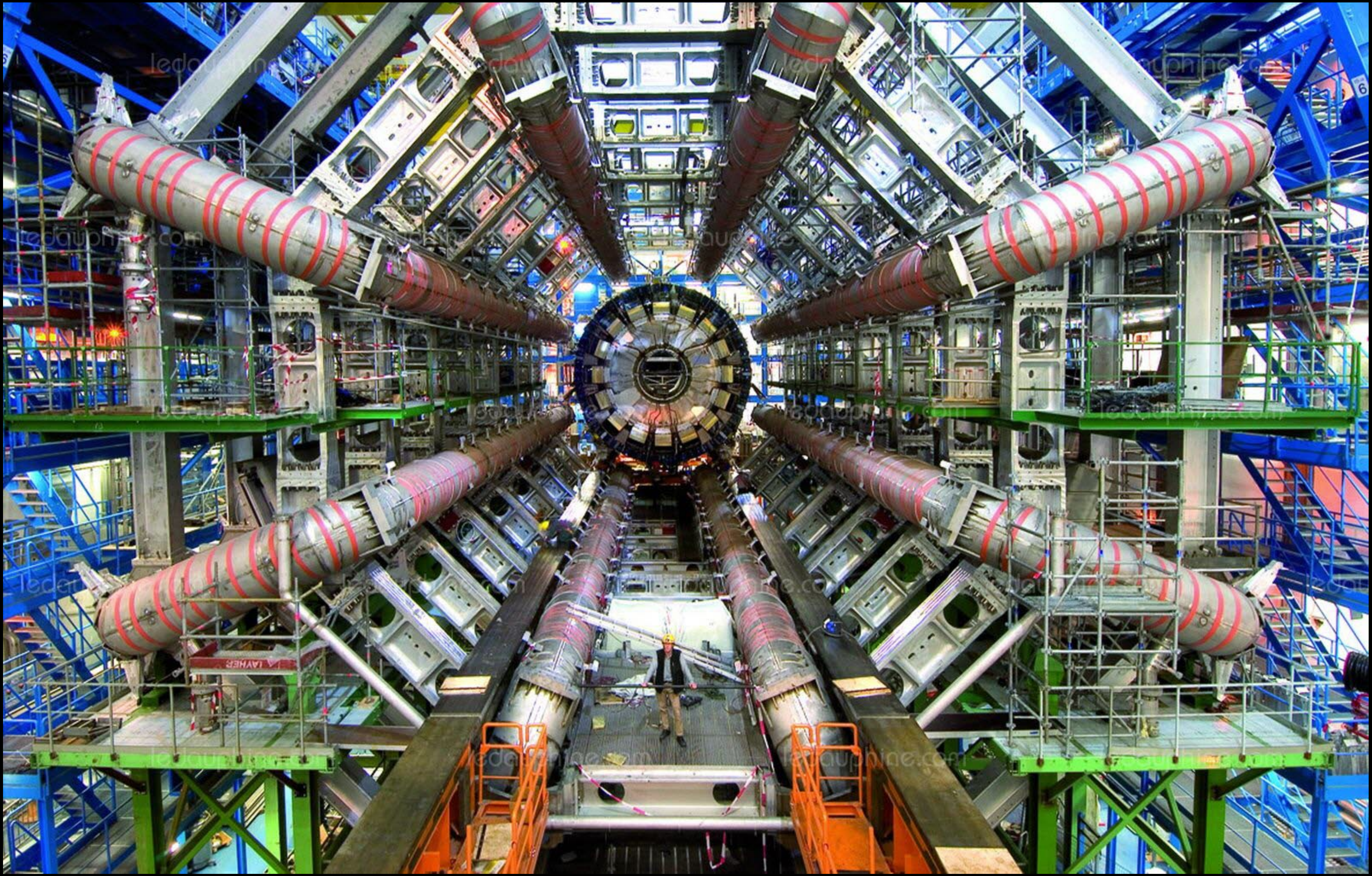


Flux

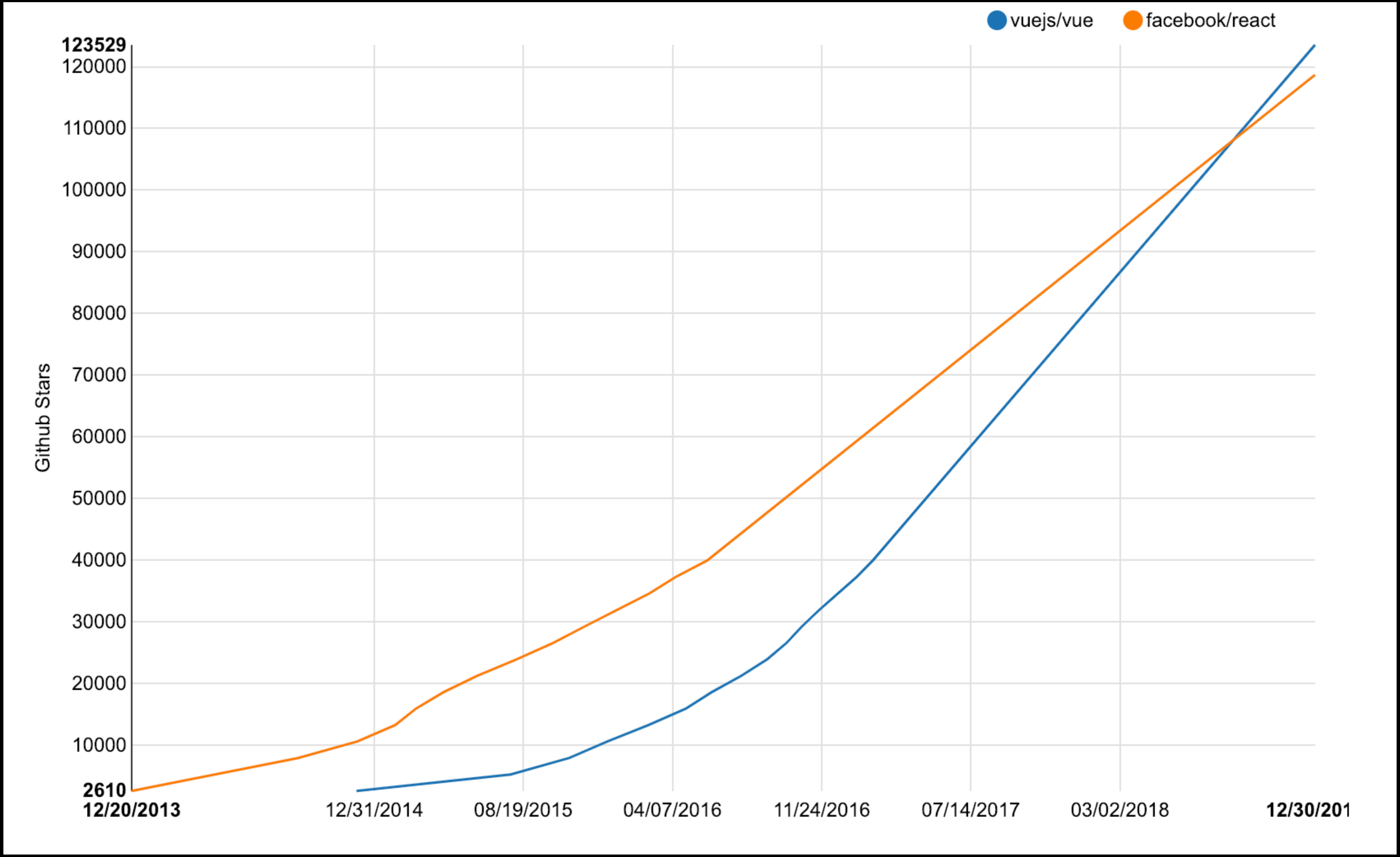

Actions
Dispatcher
Callbacks
Web API
Web API Utils
Action Creators
Store
User Interactions
React Views
Change Events + Store Queries

Redux


Redux Flow
dispatch(action)
(previousState, action)
Action Creators
Store
Reducers
(newState)
(state)
React Components
Interaction e.g onClick



github stars



Vue

- Templating language
- Patterns of best practise
- Solution for communication

Templating language

React (JSX)

```
function Hello({loading, name, age}) {
  return (
    loading ? (
      <div>Loading</div>
    ) : (
      <div>
        {name && <div>{name}</div>}
        {age && <div>{age}</div>}
      </div>
    )
  )
}
```

Vue

```
<template>
  <div v-if="loading">Loading</div>
  <div v-else>
    <div v-if="name">{{name}}</div>
    <div v-if="age">{{age}}</div>
  </div>
</template>
```

loop

React

```
<ul>
  {names.map(name =>
    <li>{name}</li>
  )}
</ul>
```

Vue

```
<ul v-for="name in names">
  <li>{{name}}</li>
</ul>
```

CSS class-toggling

React

```
<div className={`foo ${enable ? 'bar' : ''}`}/>
```

Vue

```
<div class="foo" :class="{'bar': enable}"/>
```

Two-way binding

React

```
render() {
  return (
    <input type="text"
           value={this.state.name}
           onChange={this.setName}/>
  )
}

setName(event) {
  this.setState({name: event.target.value});
}
```

Vue

```
<input type="text"
       v-model="name"/>
```

Patterns for best practise

```
<template>
  <button @click="increaseCounter">
    Click me: {{ doubleCounter }}
  </button>
</template>

<script>
  export default {
    data: () => ({
      counter: 0,
    }),
    methods: {
      increaseCounter() {
        this.counter++;
      }
    },
    computed: {
      doubleCounter() {
        return this.counter * 2;
      }
    }
  }
</script>
```

State management

React

```
state = {
  counter: 0
};

increaseCounter() {
  this.setState({
    counter: this.state.counter + 1
  });
}
```

Vue

```
const state = {
  counter: 0
}

export default {
  data: () => ({
    state,
  }),
  method: {
    increaseCounter() {
      this.state.counter++;
    }
  },
}
```

```
import store from './store';

export default {
  data() {
    return {
      privateState: {a: 1},
      globalState: store.state,
    }
  },
  method: {
    increaseCounter() {
      store.increaseCounter();
    }
  },
  computed: {
    counter() {
      return this.globalState.counter;
    }
  }
}
```

```
export default new class Store {
  state = {
    counter: 0
  }

  increaseCounter() {
    this.state.counter++;
  }
}
```

Summary

- The problem with React
- Compared React and Due
- The solution for communication