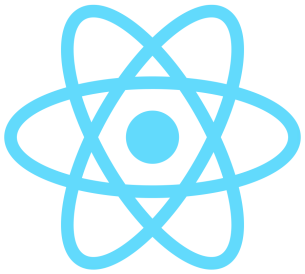
EDGE ONE SOLUTIONS

Wstęp do React

Dawid Studziński



Główne punkty:

- Czym jest (a czym nie jest) React?
- Zalety/wady Reacta.
- Jak działa? Porównanie z np AngularJS
- Keys
- JSX
- Rozpoczęcie nowego projektu, Webpack

Czym jest (a czym nie jest) React?

- React to tylko biblioteka,
- React to nie framework (jak np Angular),
- React służy jedynie do tworzenia komponentów web
- React to ‘dyrektywy’ z Angulara,
- sam React (bez dodatków) nie wystarczy do zbudowania SPA

zawód i rozczarowanie

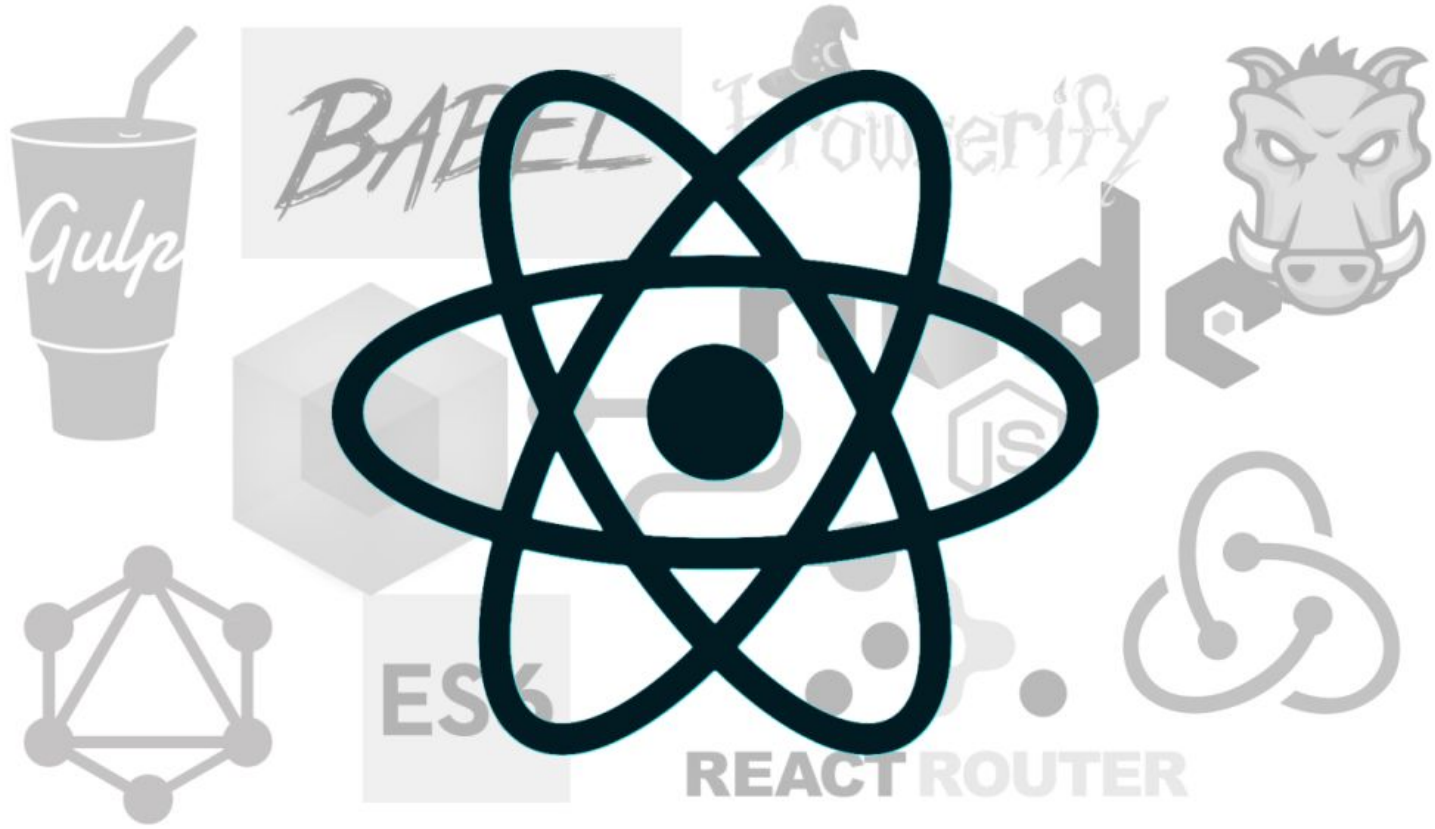
Zalety Reacta

- Robi jedną rzecz,
- Można go łączyć z innymi puzzlami aby stworzyć własną architekturę,
- Nie narzuca żadnej konkretnej architektury do np zarządzania stanem aplikacji (Redux, Mobix, Relay…), pobieraniem danych (fetch API, axios…)
- Można stosować funkcje JS (map, reduce…) do tworzenia widoku,
- Można łatwo pisać w ES6, używać nowości JS (dekoratory),
- JSX to tak naprawdę JS. Komponenty można podstawiać pod zmienne, przekazywać, modyfikować
- Zwiększa wiedzę o samym JS

Wady Reacta

- Robi tylko jedną rzecz,
- Trzeba go łączyć z innymi puzzlami aby stworzyć własną architekturę,
- Nie narzuca żadnej konkretnej architektury, każdy może pisać jak chce i używać ulubionych dodatków
- Logika wymieszana z widokiem ‘PHP style’
- Każdy przykład wygląda inaczej. Brak styleguide (jak John Papa dla Angulara),
- Fragmenty widoku w zmiennych JS. Nie wiadomo o co chodzi.
- Trzeba zwiększyć wiedzę o samym JS

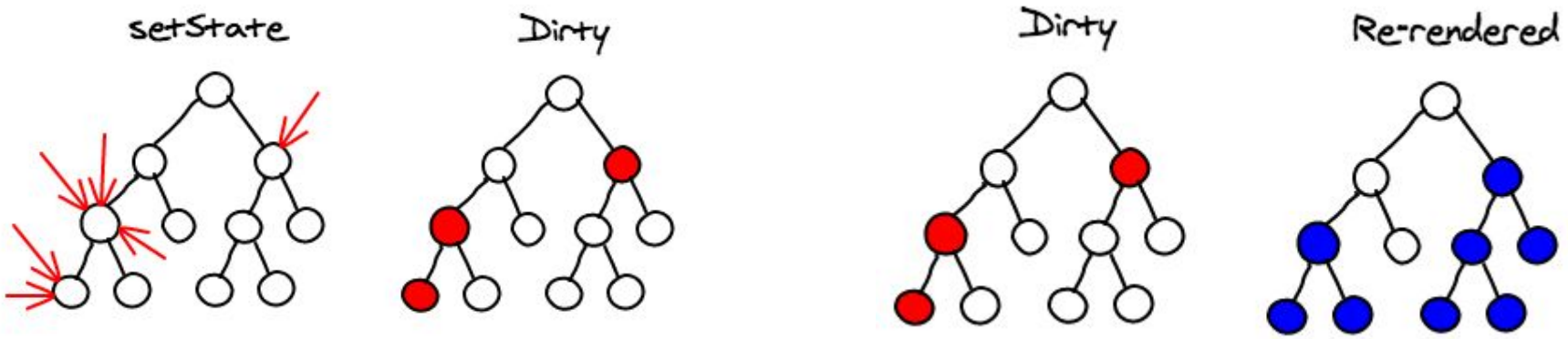
So you want to learn React.js?



https://edgecoders.com/so-you-want-to-learn-react-js-a78801d3cd4d

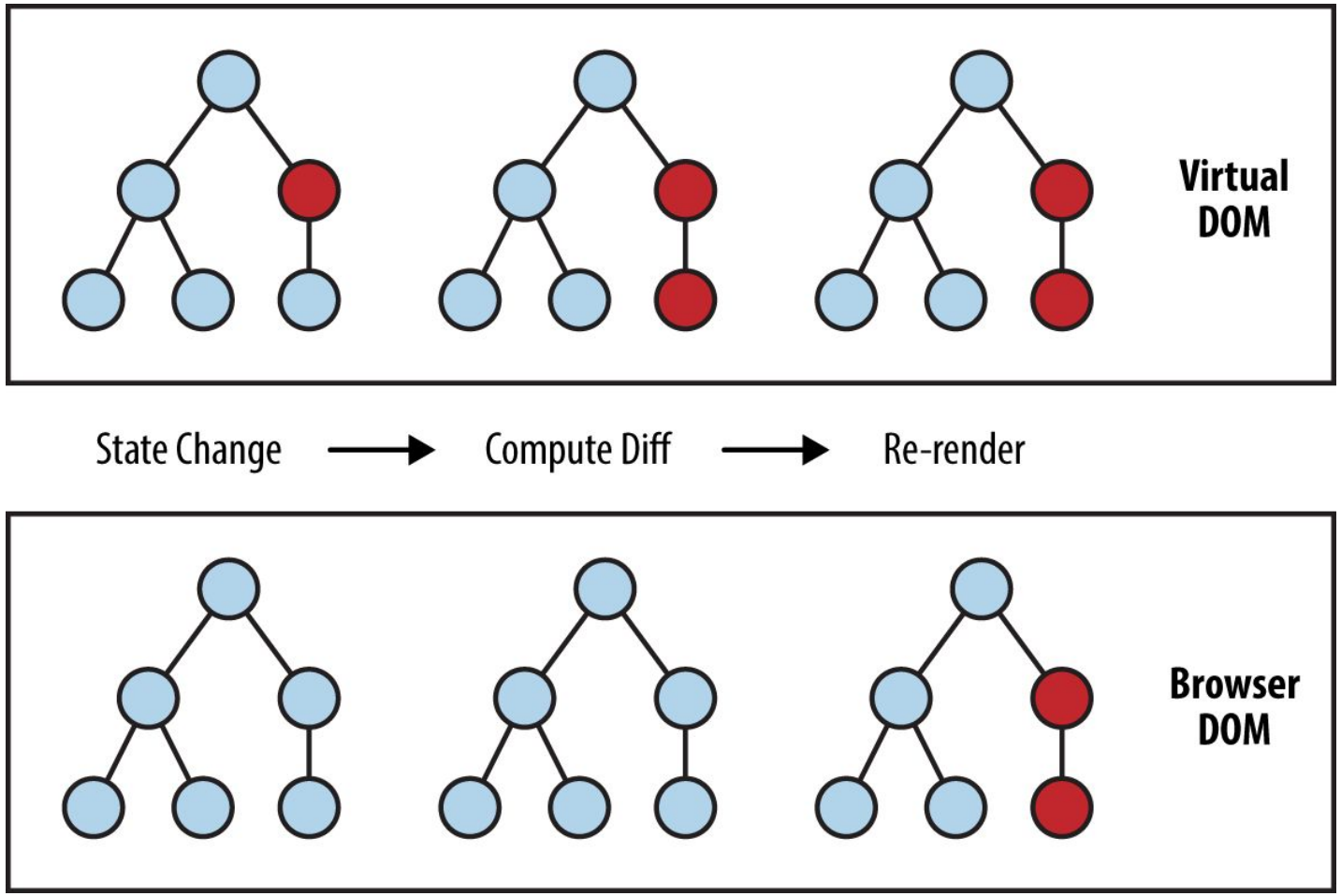
Jak działa React?

- Nie operuje bezpośrednio na DOM (VirtualDOM),
- Szuka zmian do przerenderowania w procesie zwanym reconciliation przy założeniach (standardowo $O(n^3)$):
  - dwa różne elementy tworzą różne drzewa,
  - obiekty o takim samym key są takie same



https://calendar.perfplanet.com/2013/diff/

Jak działa React?



http://slidedeck.io/DonaldWhyte/isomorphic-react-workshop

Jak działa React?

Przykład 1:

```
<div>
  <span>Node1</span>
  <Header/>
</div>
```

```
<div>
  <Header/>
</div>
```

Przykład 2:

```
<div>
  <span>Node1</span>
  <Header/>
</div>
```

```
<div>
  {null}
  <Header/>
</div>
```

Przykład 3:

```
<div>
  <span>Node1</span>
  <Header key="header"/>
</div>
```

```
<div>
  <Header key="header"/>
</div>
```

Jak działa React?

```
if (showNode) {
 return (
   <div>
     <span>Node1</span>
     <Header/>
   </div>
 )
}

return (
 <div>
   <Header/>
 </div>
)
```

```
return (
 <div>
   {showNode ? <span>Node1</span> : null}
   <Header/>
 </div>
)
```

http://jsbin.com/rumedikipa/1/edit?js,output

Kilka słów o Key

Nie używać indeksu w tablicy jako key:

```
<div>
 {items.map((item, idx) => <Item key={idx}/>)}
</div>
```

http://output.jsbin.com/wohima#

Nie używać Math.random() jako key :)

```
<div>
 <input key={Math.random()} type='text' value=""/>
</div>
```

http://output.jsbin.com/zorijiyewi/1/

https://medium.com/@robinpokorny/index-as-a-key-is-an-anti-pattern-e0349aece318

JSX

```jsx
const Header = <Header title="My header"/>;

const Todos = todos.map(todo => <Todo key={todo.id} todo={todo}/>);

const content = <div className="content">{Todos}</div>;
```

```
4:10  error  'React' must be in scope when using JSX  react/react-in-jsx-scope
```

https://babeljs.io/repl/