

React Fundamentals

Jithin Zacharia



About Me

- Jithin Zacharia
- Full Stack Developer @ Infosys
- Website - <https://jithinqw.github.io/>

Agenda

- The Javascript elders
- What is React ?
- Why React ?
- Key Fundamentals - V-DOM, React Fiber
- React Components
- React Props & State
- Demo - Writing your React App.
- State Management - Introduction to Redux.
- Who is using React ?
- Writing Mobile Apps using React - React Native
- Testing

Why Javascript ?

PRIORITIES

WALLS ARE NO
LONGER THE PRIORITY.
THE CLIENT WANTS THE
PAINTING DONE
FIRST !



JavaScript Elders



- Write less and do more.
- AJAX was introduced.
- Complex manipulations ? I can do it in back off my head.

AngularJS



- Standing on the shoulders of backend developers
- Move your logic to FrontEnd for the first time.
- JQuery ? It' just 1800's stuff.
- Hey what about building and shipping apps ?
Do it old style!!! Copy and Paste.

Enter React !



How to get pit hair in 5 seconds

72K views · 5 years ago



What is React ?

- I am a JavaScript framework developed in 2013 @ Facebook.
- I am extremely good at developing visual interfaces.
- I split the visual space into components.

 facebook / react

 Used by ▾ 3.6m

 Watch ▾ 6.6k

 Unstar 148k

 Fork 28.7k

 Code

 Issues 485

 Pull requests 100

 Actions

 Projects 0

 Wiki

 Security 0

 Insights

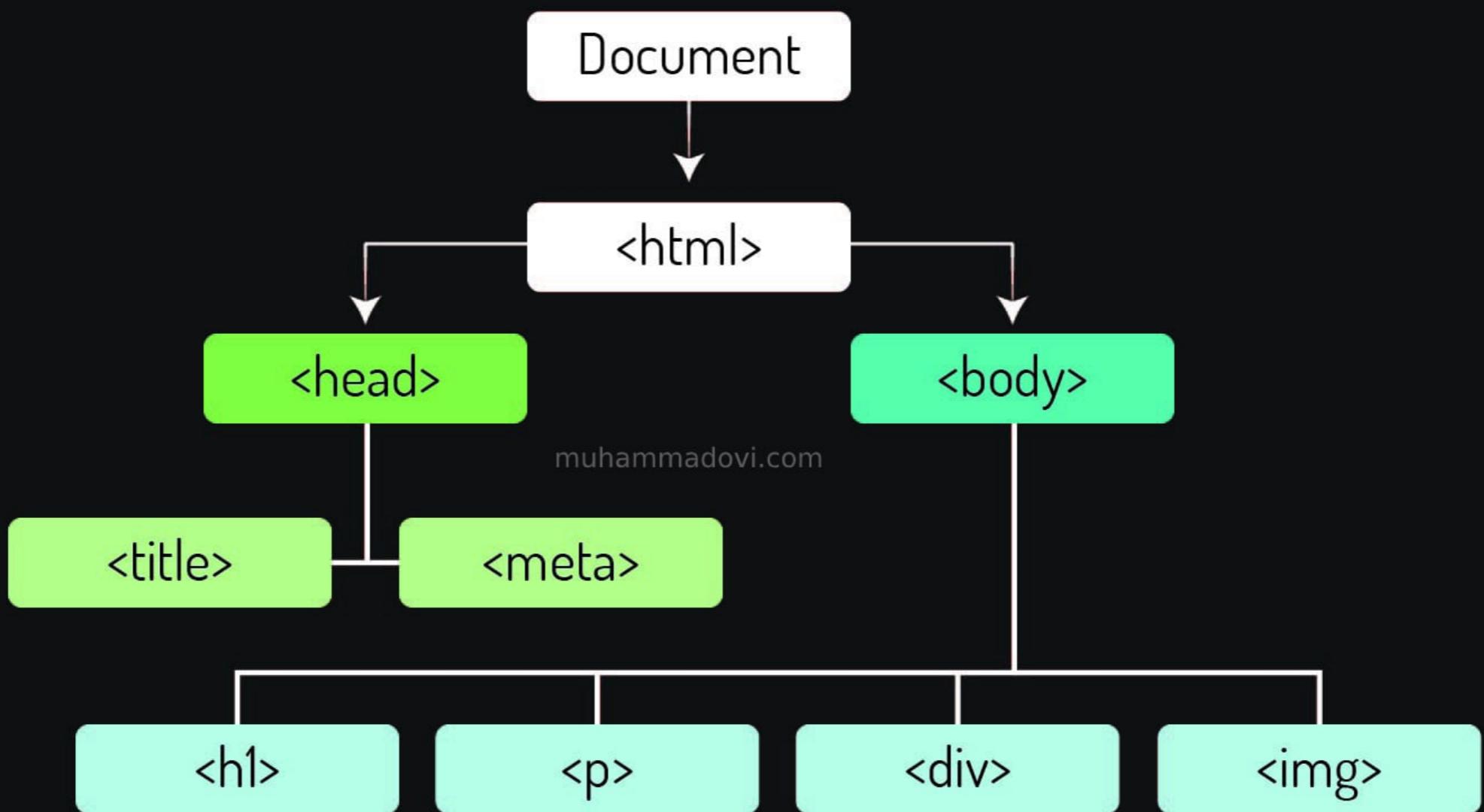
Why is React so popular ?

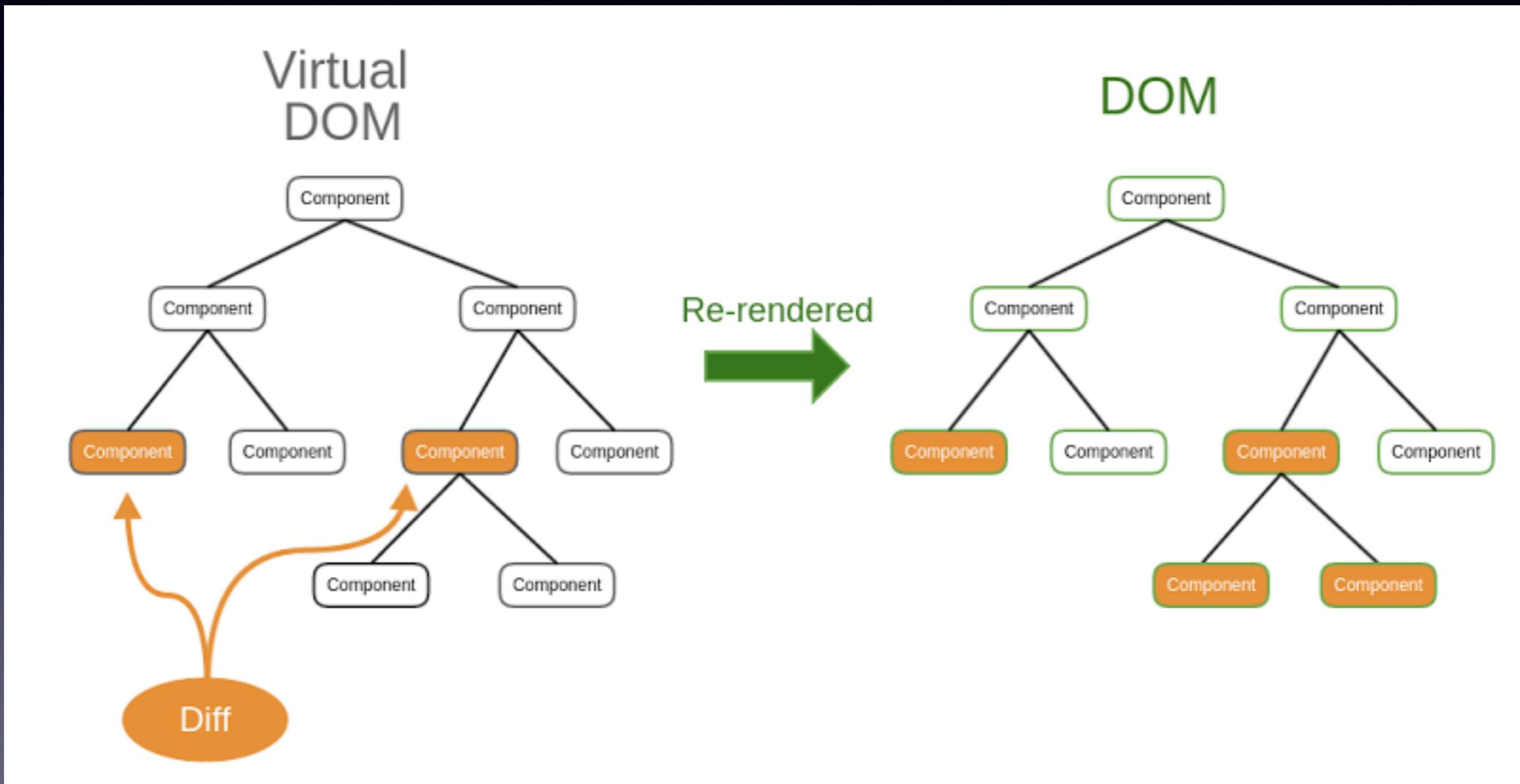
- No backward compatibility between Angular 1.x and 2.x
- Everyone was working to achieve MVC - I didn't
- Less complex than other frameworks.
- Can work anywhere, even SSR!
- Less configuration, more speed.

React Fundamentals

Virtual DOM

- Converts HTML DOM to a JavaScript Object.
- Only needs updates when state changes.
- This approach is been used by most of the frameworks.





React Fiber

- Diffing Algorithm
- Finding out which component needs rendering ?
- Updating DOM is costly and it's the soul of React.

React Components

- Building block of a React Application.
- Each division in React can be a component.
- Each component represents a sub-tree in the React Virtual DOM.
- Each component is controlled by Props and State.

A blog posts listing page

Blog post

Blog post

Blog post with image preview

About

Links

State

- State represents data inside a specific component.
- Can mutate state using setState function.
- State is always uni-directional.
- Changing state in one component never changes its parent or its siblings .

```
class BlogPostExcerpt extends Component {
  constructor(props) {
    super(props)
    this.state = { clicked: false }
  }

  render() {
    return (
      <div>
        <h1>Title</h1>
        <p>Description</p>
      </div>
    )
  }
}
```

Props

- Default properties provided to every component.
- Passing props down to child components is a better way to pass data to child components.

```
import React, { Component } from 'react'

class BlogPostExcerpt extends Component {
  render() {
    return (
      <div>
        <h1>{this.props.title}</h1>
        <p>{this.props.description}</p>
      </div>
    )
  }
}
```

```
const desc = 'A description'  
//...  
<BlogPostExcerpt title="A blog post" description={desc} />
```

PropTypes

- How to make sure that each component is receiving props which are required ?
- PropTypes make a good linting tool for React Application.
- Great tool when comes to unit - testing.

```
import PropTypes from 'prop-types'
import React from 'react'

class BlogPostExcerpt extends Component {
  render() {
    return (
      <div>
        <h1>{this.props.title}</h1>
        <p>{this.props.description}</p>
      </div>
    )
  }
}

BlogPostExcerpt.propTypes = {
  title: PropTypes.string,
  description: PropTypes.string
}

export default BlogPostExcerpt
```

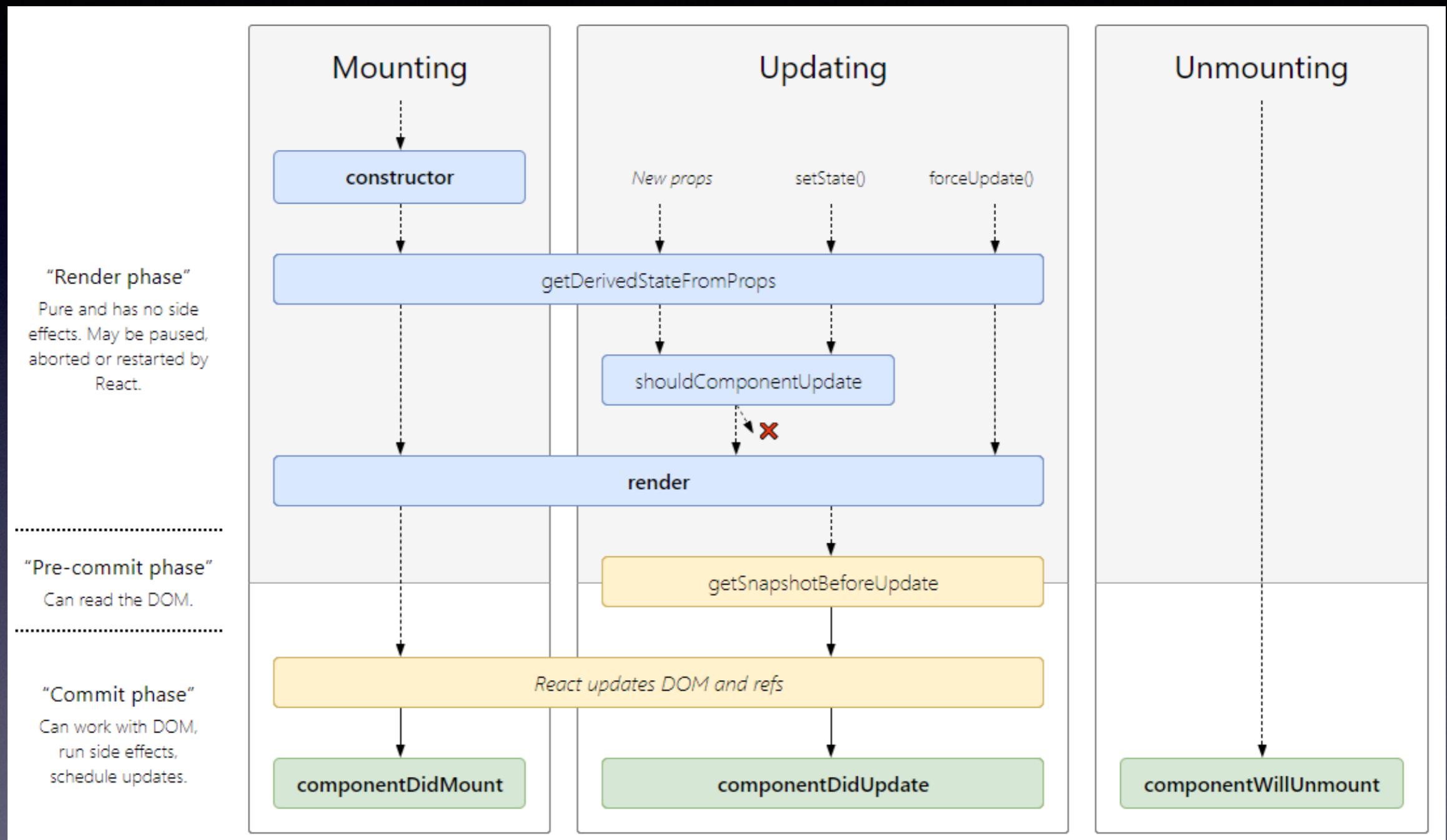
React Events

- Since DOM has events, React has to invent its own events.
- React events are single queue which listens to user activity .

```
const CurrencySwitcher = props => {
  return (
    <button onClick={props.handleChangeCurrency}>
      Current currency is {props.currency}. Change it!
    </button>
  )
}
```

Life-Cycle Events

- Maintaining data update on different stages of Component life-cycle.
- During these events user can layout management, getting data from a server etc...

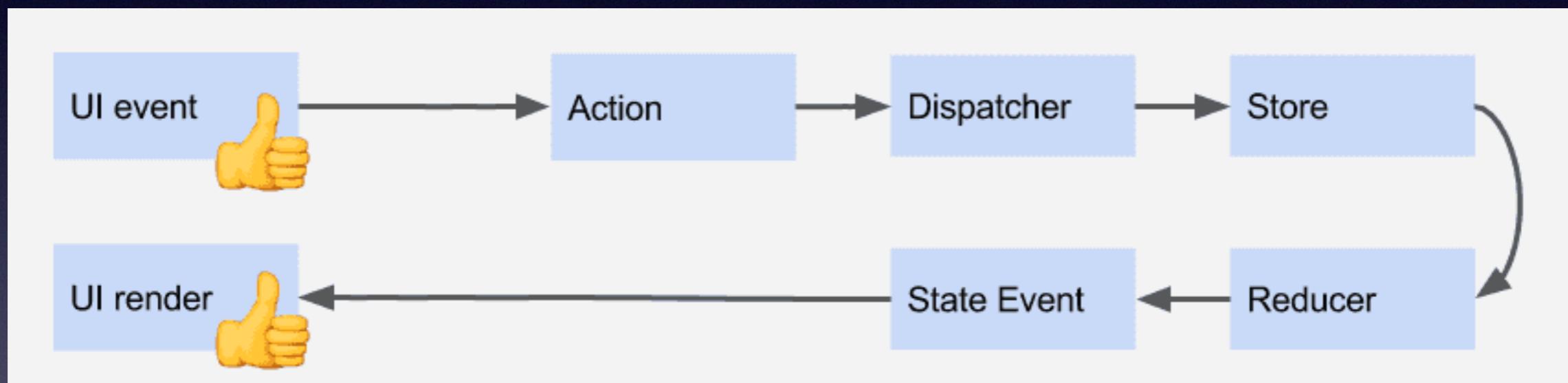


Demo

Managing - State

- Why do we need to mange state ?
- When to use state management tools ?

ReduxJS



Popular Libraries

- ReduxJS
- Flow
- Redux Saga
- RxJS

Demo

React Native



ENGINEERING BLOG CULTURE SPONSORSHIP



Careers

Infrastructure

Product Development

Developer Tooling

Mobile

Web Development

Data Science & Engineering

Trust



Farhan Thawar 01/29/2020

React Native is the Future of Mobile at Shopify



Share



Tweet



Share



Share

Testing using Jest and Enzyme

What's next ?

- GraphQL
- Apollo
- NextJS
- Svelte
- Angular - Ivy and beyond
- Creating Native apps using React.
- Deno is here !