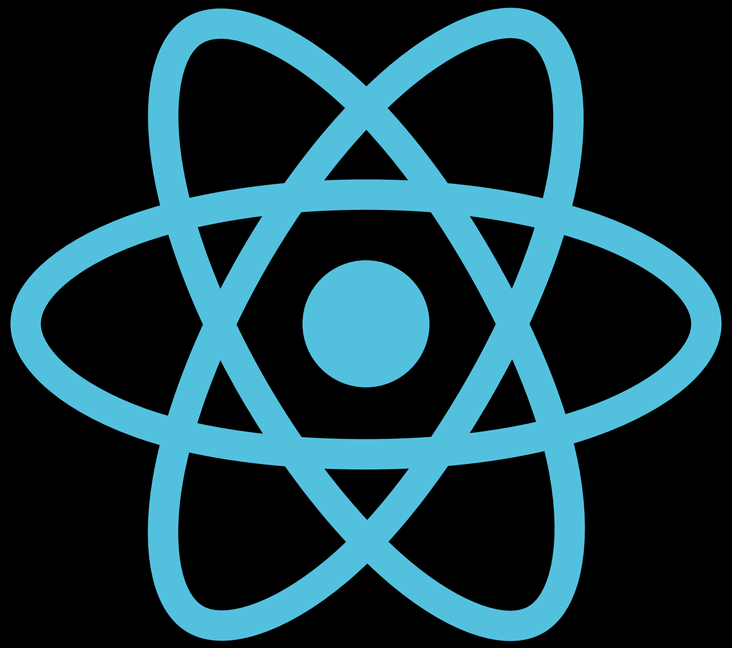


React for Beginners

Sytac sLab

2017 01 06

Tom Hastjarjanto

Software developer

Jorigis Snaras

Software developer

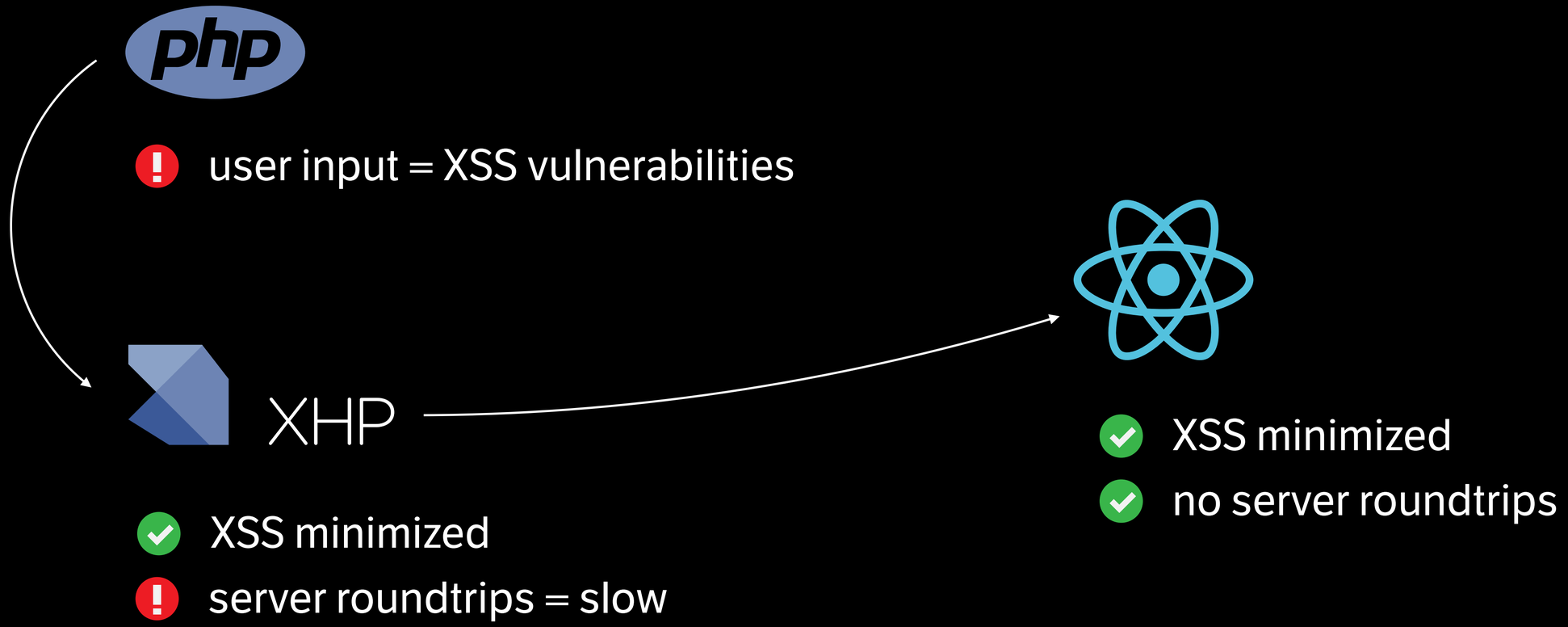
What is React?

A **JavaScript** library for building **user interfaces**

Often referred to as the **V** in **MVC**

Created at Facebook

Why React?


php
user input = XSS vulnerabilities
XHP
XSS minimized
server roundtrips = slow
XSS minimized
no server roundtrips

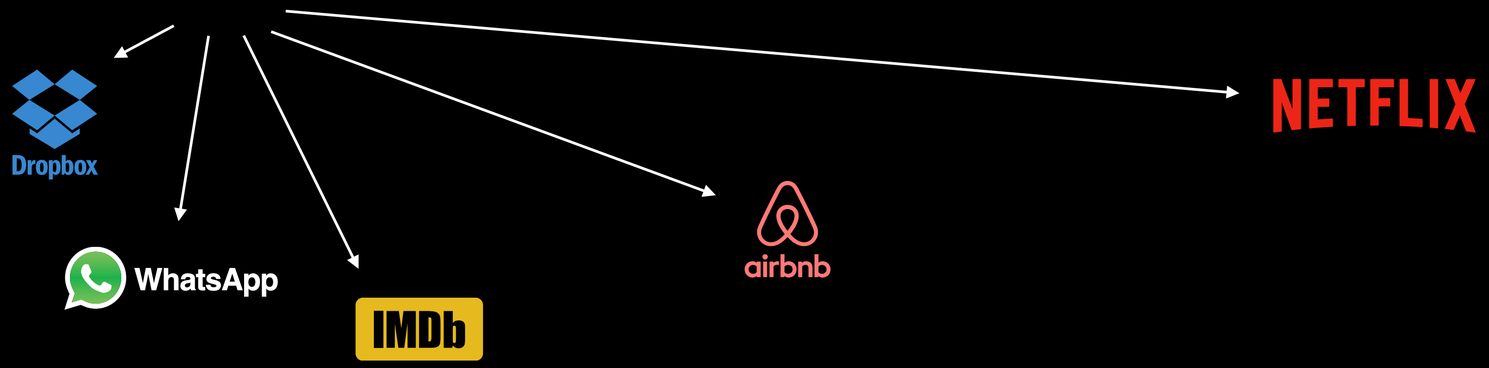
Timeline

2011 Created by Facebook engineer Jordan Walke

2011 First deployed on Facebook's newsfeed

2012 Deployed on Instagram.com

2013 Open-sourced at JSConf US



Core Concepts

Virtual DOM

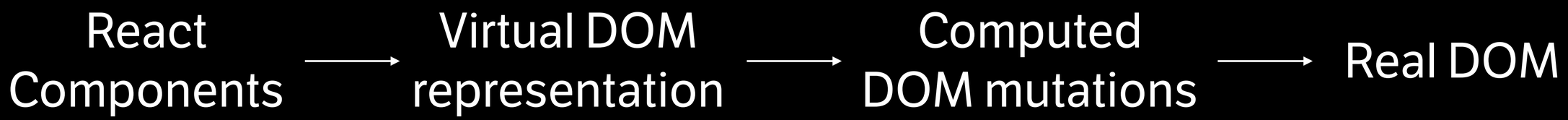
Components (and JSX)

One-way Data Flow

Virtual DOM

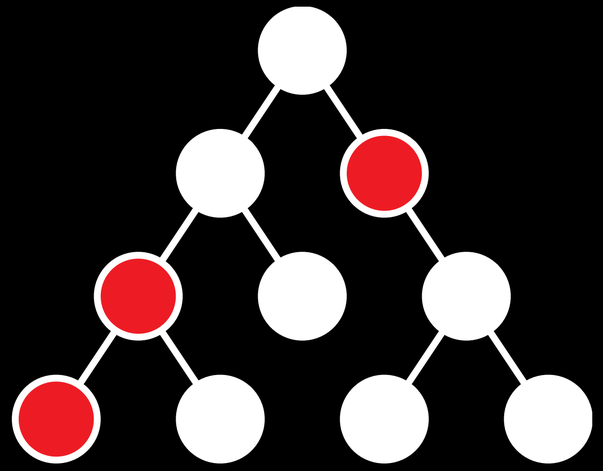
“Re-render everything!”

Virtual DOM


React Components
Virtual DOM representation
Computed DOM mutations
Real DOM

Virtual DOM

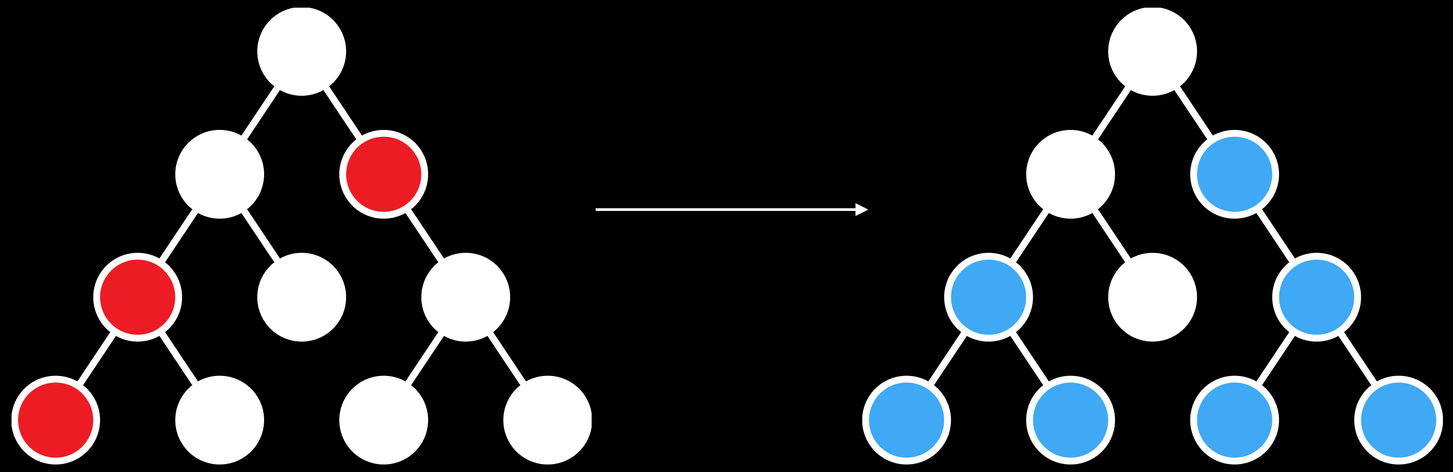
Dirty



Virtual DOM

Dirty

Re-rendered



Components

“Components let you split the UI into independent, reusable pieces, and think about each piece in isolation”

https://facebook.github.io/react/docs

Components

A React component is:

Declarative description

of a cohesive

and composable

unit of UI

Components

```
1   import React from 'react';
2   import ReactDOM from 'react-dom';
3
4   class MyComponent extends React.Component {
5     render() {
6       return React.DOM.div({ className: 'myDiv' }, 'my text');
7     }
8   }
9
10  ReactDOM.render(
11    React.createElement(MyComponent),
12    document.body
13  );
```

Components

```
1   import React from 'react';
2   import ReactDOM from 'react-dom';
3
4   class MyOtherComponent extends React.Component {
5     render() {
6       return React.DOM.h1(null, 'Hi there!');
7     }
8   }
9
10  class MyComponent extends React.Component {
11    render() {
12      return React.DOM.div({ className: 'myDiv' }, [
13        React.createElement(MyOtherComponent),
14        React.DOM.p(null, 'How are things with you?')
15      ]);
16    }
17  }
18
19  ReactDOM.render(
20    React.createElement(MyComponent),
21    document.body
22  );
```

Components

```
1   import React from 'react';
2   import ReactDOM from 'react-dom';
3
4   class MyOtherComponent extends React.Component {
5     render() {
6       return <h1>Hi there!</h1>
7     }
8   }
9
10  class MyComponent extends React.Component {
11    render() {
12      return (
13        <div className="myDiv">
14          <MyOtherComponent />
15          <p>How are things with you?</p>
16        </div>
17      )
18    }
19  }
20
21  ReactDOM.render(<MyComponent />, document.body);
```

Components

```
1   import React from 'react';
2   import ReactDOM from 'react-dom';
3
4   class MyComponent extends React.Component {
5     render() {
6       return (
7         <p> ??? </p>
8       )
9     }
10  }
11
12  ReactDOM.render(<MyComponent />, document.body);
```

"Hello there!"

Components

```
1   import React from 'react';
2   import ReactDOM from 'react-dom';
3
4   class MyComponent extends React.Component {
5     render() {
6       return (
7         <p>{this.props.greeting}</p>
8       )
9     }
10  }
11
12  ReactDOM.render(
13    <MyComponent greeting="Hello there!"/>,
14    document.body
15  );
```

Components

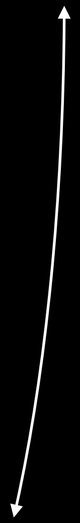
Props are to components

what

arguments are to functions

Components

```
1   import React from 'react';
2   import ReactDOM from 'react-dom';
3
4   class MyComponent extends React.Component {
5     render() {
6       return (
7         <p>
8           {this.props.greeting}
9           <button onClick={this.props.onButtonClick}>
10            Click me
11          </button>
12        </p>
13      )
14    }
15  }
16
17  function myHandler(event) {
18    console.log('Button clicked');
19  }
20
21  ReactDOM.render(
22    <MyComponent greeting="Hello there!"
23                 onButtonClick={myHandler}/>,
24    document.body
25  );
```

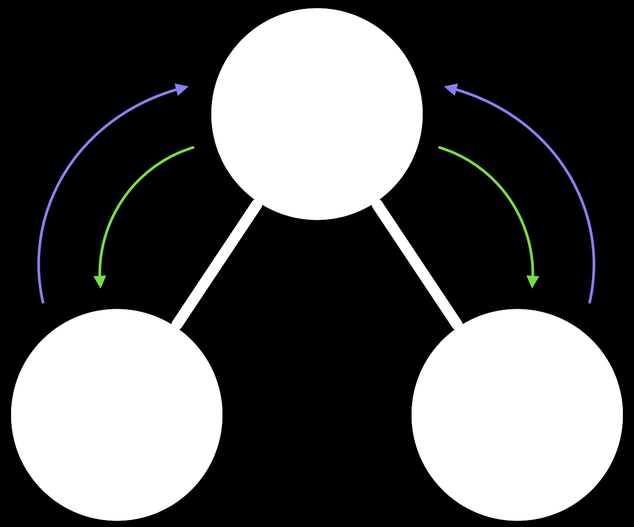


One-way Data Flow

“Properties flow down;

actions flow up”

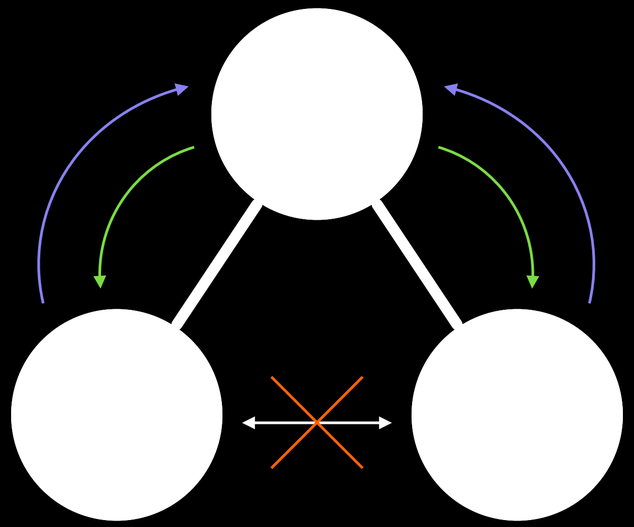
One-way Data Flow



props

callbacks

One-way Data Flow



props

callbacks

State

Besides props, components can store and manipulate local state

State

```
constructor() {
  this.state = { someValue: 1 };
}

someHandler() {
  this.setState({ someValue: 2 });
}

render() {
  return <p>{this.state.someValue}</p>
}
```

State

```
1   import React from 'react';
2
3   class MyComponent extends React.Component {
4     constructor() {
5       this.state = { enabled: false };
6
7       this.handlePress = this.handlePress.bind(this);
8     }
9
10    handlePress() {
11      this.setState({ enabled: !this.state.enabled });
12    }
13
14    render() {
15      return (
16        <div>
17          <span>Is enabled: {this.state.enabled}</span>
18          <button onClick={this.handlePress}>Toggle</button>
19        </div>
20      );
21    }
22  }
```

State

```
1   import React from 'react';
2
3   class MyComponent extends React.Component {
4     constructor() {
5       this.state = {
6         currentMessage: '',
7         messages: []
8       };
9     }
10
11    handleInputChange(event) {
12      this.setState({ currentMessage: event.target.value });
13    }
14
15    handleAddClick() {
16      const { currentMessage, messages } = this.state;
17
18      this.setState({
19        currentMessage: '',
20        messages: [ ...messages, currentMessage ]
21      });
22    }
23
24    render() {}
25  }
```

State

```
1   import React from 'react';
2
3   class MyComponent extends React.Component {
4     constructor() {...}
10
11    handleInputChange(event) {...}
14
15    handleAddClick() {...}
23
24    render() {
25      return (
26        <div>
27          <p>
28            <input type="text" value={this.state.currentMessage}
29                   onChange={this.handleInputChange}/>
30          </p>
31
32          {this.state.messages.map((message, index) => <p key={index}>{message}</p>)}
33
34          <button onClick={this.handleAddClick}>
35            Add message
36          </button>
37        </div>
38      );
39    }
40  }
```

Lifecycle

“Each component has several ‘lifecycle methods’ that you can override to run code at particular times in the process”

https://facebook.github.io/react/docs

Lifecycle

Mounting

constructor()

componentWillMount()

render()

componentDidMount()

Updating

componentWillReceiveProps()

shouldComponentUpdate()

componentWillUpdate()

componentDidUpdate()

Unmounting

componentWillUnmount()

Lifecycle

Mounting

constructor()

componentWillMount()

render()

componentDidMount()

Updating

componentWillReceiveProps()

shouldComponentUpdate()

componentWillUpdate()

componentDidUpdate()

Unmounting

componentWillUnmount()

Lifecycle

```
1   import React from 'react';
2
3   class MyComponent extends React.Component {
4     constructor() {
5       this.state = { ip: null };
6     }
7
8     componentDidMount() {
9       fetch('http://ip.jsontest.com/')
10        .then(response => response.json())
11        .then(result => this.setState({ ip: result.ip }));
12    }
13
14    render() {
15      return (
16        <p>
17          My IP is: {this.state.ip ? this.state.ip : 'unknown'}
18        </p>
19      );
20    }
21  }
```