

Boris Dinkevich

# ReactJS & AngularJS

(not vs !)



Chuck Norris doesn't need a framework - he updates the DOM directly, in C.

# Boris Dinkevich

CEO @ 500TECH

Cats, good weather and Ruby

500TECH - AngularJS Consultancy

# What is ReactJS ?

- Component framework
- Virtual DOM

# Adding React to Angular

# The basics

```
<body>
  <sample-directive></sample-directive>
</body>
```

```
angular.module("sampleapp", [])
  .directive("sampleDirective", function() {
    return {
      template: "<h2>Angular directive</h2>"
    }
  });
```

```
var MyComponent = React.createClass({
  render: function() {
    return <h2>React Component</h2>;
  }
});
```

## Example code

# Directive to render a Component

```
<body>
  <sample-directive></sample-directive>
  <react-container></react-container>
</body>
```

```
var MyComponent = React.createClass({
  render: function() {
    return <h2>React Component</h2>;
  }
});
```

```
angular.module("sampleapp")
  .directive("reactContainer", function() {
    return {
      template: '<div></div>',
      link: function(scope, element, attrs) {
        return React.render(React.createElement(MyComponent, null), element[0]);
      }
    }
  });
```

# Directive to render a Component

```
angular.module("sampleapp")
  .directive("reactContainer", function() {
    return {
      template: '<div></div>',
      link: function(scope, element, attrs) {
        return React.render(
          React.createElement(MyComponent, null),
          element[0]);
      }
    }
  });
```

# ngReact

<https://github.com/davidchang/ngReact>



# Thank you, buh bye !



500TECH - AngularJS Consultancy

# Dependency Injection - I

```
angular.module("sampleapp")
  .factory("ReactComponent", function() {
    return React.createClass({
      render: function() {
        return React.createElement("h2", null, "React Component");
      }
    });
  });
```

Example code

# Dependency Injection - II

```
angular.module("sampleapp")
  .factory("ReactComponent", function(DataService) {
    return React.createClass({
      render: function() {
        return React.createElement("h2", null, "React Component: ", DataService.data());
      }
    });
  })
```

Example code

# Dependency Injection - III

```
angular.module("sampleapp")
  .directive("reactContainer", function(ReactComponent) {
    return {
      template: '<div></div>',
      link: function(scope, element, attrs) {
        return React.render(
          React.createElement(ReactComponent, null), element[0]
        );
      }
    }
  });
```

Example code

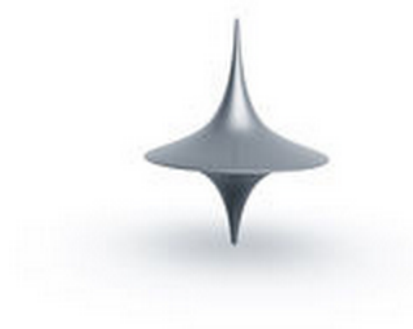
# Dependency Injection - IV

```
angular.module("sampleapp")
  .factory("HeaderComponent", function(DataService) {
    return React.createClass({
      render: function() {
        return React.createElement("h2", null, "React Component: ", DataService.data());
      }
    });
  })
```

```
angular.module("sampleapp")
  .factory("ComplexComponent", function(HeaderComponent, FooterComponent) {
    return React.createClass({
      render: function() {
        React.createElement("div", null,
          React.createElement(HeaderComponent, null),
          React.createElement("div", null, "Some nice graphics"),
          React.createElement(FooterComponent, null)
        )
      }
    });
  })
```

# Going crazy

Angular in React in Angular in React in Angular



Example code

# JSX

```
angular.module("sampleapp")
  .factory("ComplexComponent", function(HeaderComponent, FooterComponent) {
    return React.createClass({
      render: function() {
        return (
          <div>
            <HeaderComponent />
            <div>Some nice graphics</div>
            <FooterComponent />
          </div>
        );
      }
    });
  })
```

- Babel
- JSX-transpiler
- Works with Webpack/Gulp/Grunt/etc

# Directory structure

App

- directives
- services
- components

•  
•  
•



# But wait..

```
return React.createElement(  
  "h2",  
  null,  
  "React Component: ",  
  DataService.data()  
);
```

DataService.data() ?

# There is no \$watch

If you want data binding, you can either use Angular's or roll out your own (flux).

Example code



*“ \$broadcast makes more sense there*

# The Angular way

We have Dependency Injection...

We have \$rootScope or container's Isolated Scope

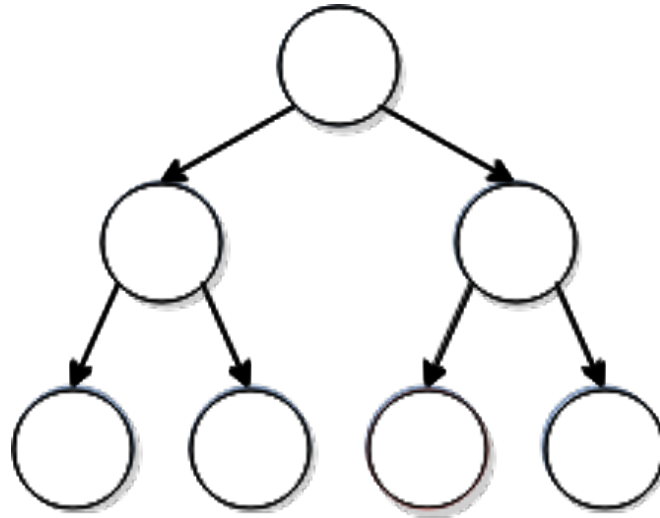
We have \$on & \$watch

We can use digest !

*“ Uh oh..*

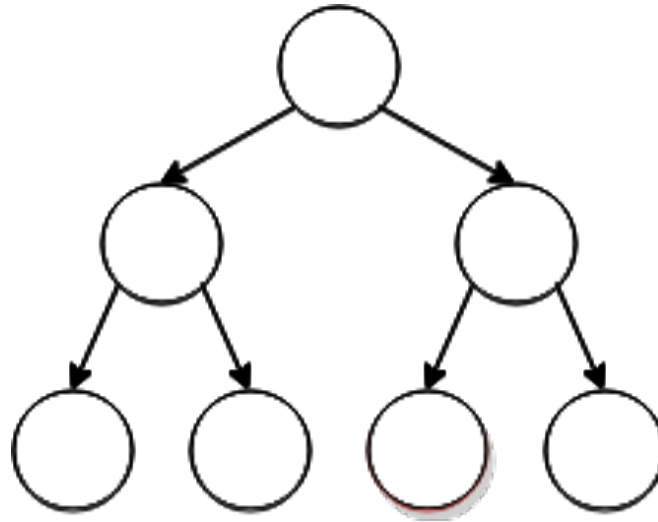
# How does Angular render

Digest cycle

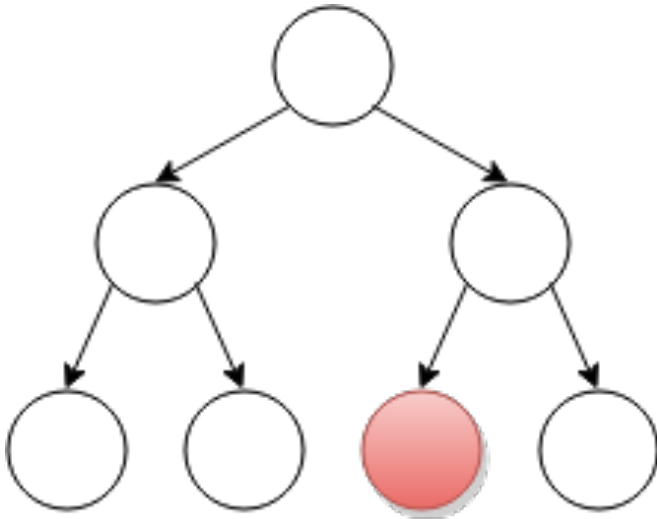


# How does React render

Data modeling - NOT Virtual DOM



# So what is this Virtual DOM ?



```
-- ProfileComponent

<div class="row">
  <div class="col-sm-6">
    <div class="profile-pic">
      
    </div>
  </div>
  <div class="col-sm-6">
    <div class="data-container">
      <span>
        <i class="icon-name"></i>
        Boris Dinkevich
      </span>
    </div>
    <div class="data-container">
      <span>
        <i class="icon-age"></i>
        33
      </span>
    </div>
  </div>
</div>
```

# So what is this Virtual DOM ?

```
-- ProfileComponent

<div class="row">
  <div class="col-sm-6">
    <div class="profile-pic">
      
    </div>
  </div>
  <div class="col-sm-6">
    <div class="data-container">
      <span>
        <i class="icon-name"></i>
        Boris Dinkevich
      </span>
    </div>
    <div class="data-container">
      <span>
        <i class="icon-age"></i>
        33
      </span>
    </div>
  </div>
</div>
```



```
-- ProfileComponent

<div class="row">
  <div class="col-sm-6">
    <div class="profile-pic">
      
    </div>
  </div>
  <div class="col-sm-6">
    <div class="data-container">
      <span>
        <i class="icon-name"></i>
        Boris Dinkevich
      </span>
    </div>
    <div class="data-container">
      <span>
        <i class="icon-age"></i>
        34
      </span>
    </div>
  </div>
</div>
```

# Wow right ?

ReactJS saved DOM manipulations by calculating the diff itself.

- [React.js Conf 2015 - Hype!](#)
- [Angular + React = Speed Dave Smith](#)

.. in JavaScript ..



**Dude...wait....**

**What???**

# "Speed" comparison

If you look carefully, most samples of AngularJS vs ReactJS either:

Unoptimized Angular side

- [React.js Conf 2015 - Hype!](#)
- [Fixed github demo PR \(Angular is 2x faster than React\)](#)

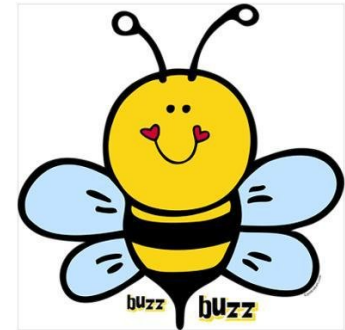
Compares data binding (\$digest), not rendering.

- [Angular + React = Speed Dave Smith](#)
- [Fixed github demo PR \(Angular and React same speed\)](#)

“ *"In theory, there ought to be no difference between theory and practice. In practice, there is."*

# So why React ?

- Components life cycle
- Clear state management
- Break away from \$digest
- Server side rendering !
- Buzz



# So why React ?

Total results: {{ ? }}

← Isolated scope digest

Oct 1	Oct 2	Oct 3	Oct 4	Oct 5
2 results	3 results	3 results	3 results	2 resul
3 results	4 results	3 results	3 results	1 resul
4 results	3 results	2 results	1 results	4 resul
4 results	4 results	2 results	0 results	0 resul
1	4	4	4	1

← Isolated scope digest

Available ?

← Isolated scope digest

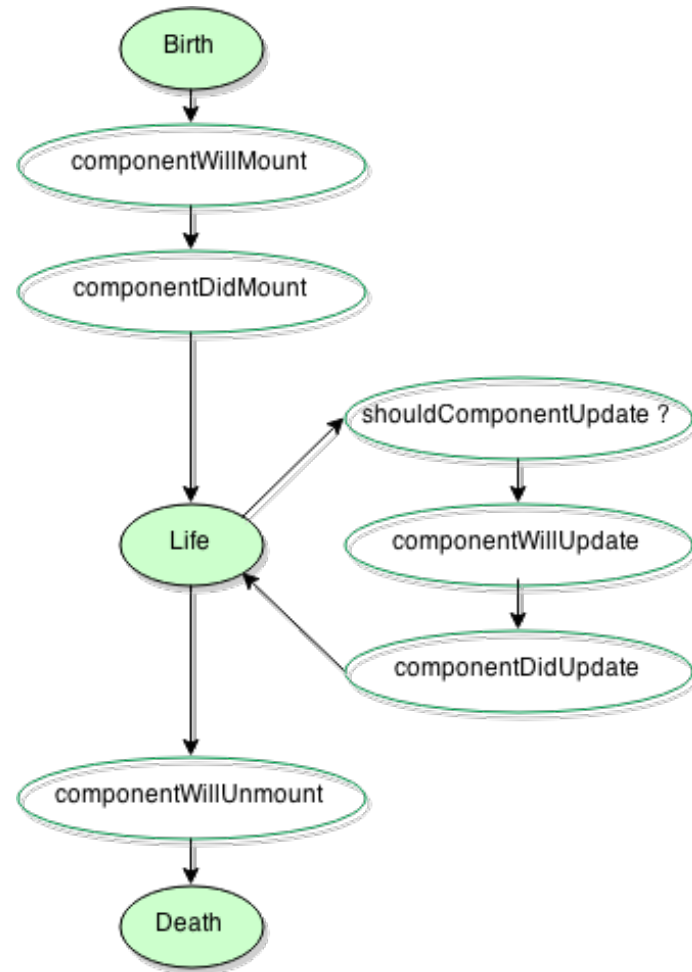
# Server Rendering

- SEO
- Fast page load (bind later)
- Embed in page before Angular even runs

`["hip","hip"]`

# Life cycle

*“ You can simulate everything in AngularJS, but then its not AngularJS.*



# JSX ?

```
React.createElement("div", null,  
  React.createElement(HeaderComponent, null),  
  React.createElement("div", null, "Some nice graphics"),  
  React.createElement(FooterComponent, null)  
)}
```

# Directive Template

```
<div class="row">
  <div class="controls" ng-class={ visible: ctrl.canAdd}>
    <button ng-click="ctrl.addRow">New row</button>
    <a class="back-button" href="">Go back</a>
  </div>
  <div ng-repeat="row in ctrl.rows" class="item">{{ row.name }}</div>
  <HelpMessage type="addRow" ng-if="ctrl.canAdd"></HelpMessage>
</div>
```



# JSX

```
render: function() {
  var className = cx({
    'visible': this.state.canAdd,
    'controls': true
  });

  var renderRow = function(row) {
    return (<div className="item">{ row.name }</div>);
  }

  return (
    <div className="row">
      <div className={ className }>
        <button onClick={ this.addRow }>New row</button>
        <a className="back-button" href="">Go back</a>
      </div>
      { this.state.rows.map(renderRow) }
      { this.state.canAdd ? <HelpMessage type="addRow" /> : null }
    </div>
  );
}
```

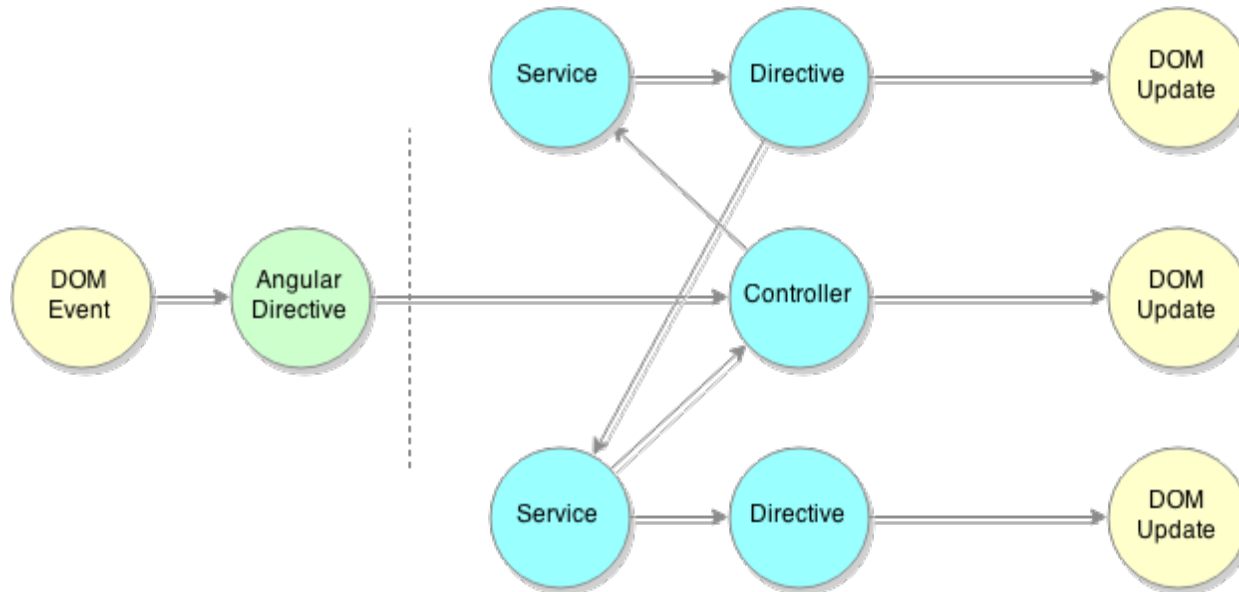
# JSX

- Interesting solution from Wix ([react-templates](#))
- Work directly with final result
- Splitting to very large of mini components

# Data Modeling

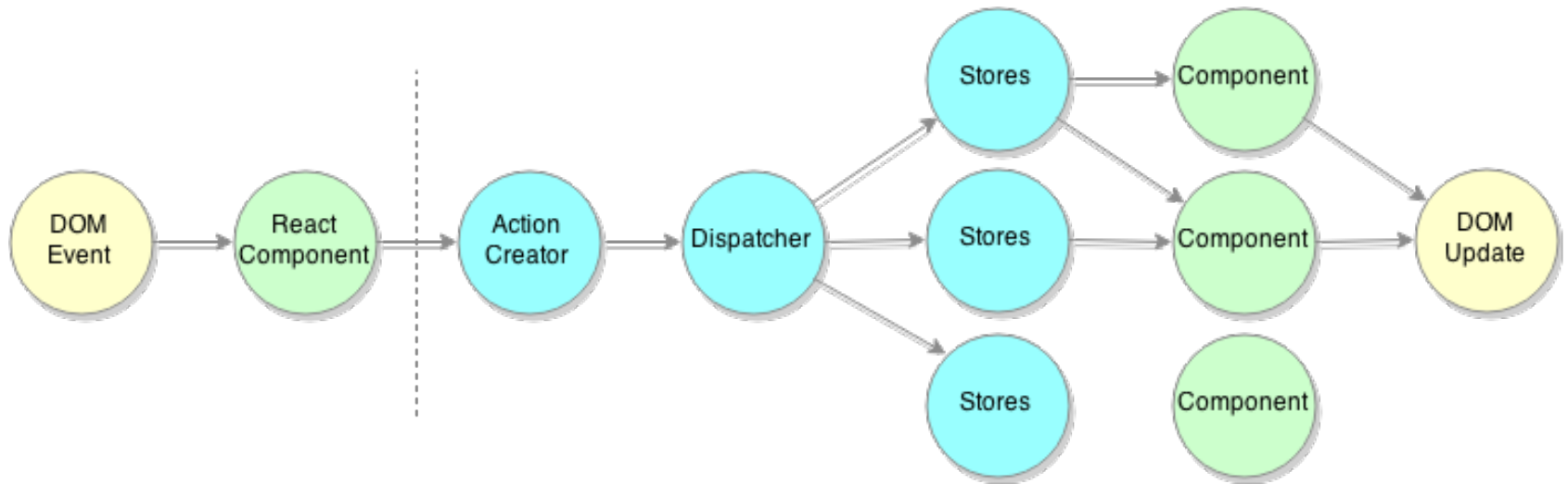
# Angular / MVVM

## Data flow



# Flux

## A Pattern - Not a library



*“ All arrows go one way*

# Flux libraries

- Facebook Flux
- Fluxible by Yahoo
- Reflux
- Alt
- Flummox
- Marty
- McFly
- Lux
- Material Flux
- Delorean



# Flux in Angular

- Regular services
- EventEmitter / \$broadcast
- ReactJS Flux libraries

# Angular 2.0



# Questions ?

Amazing AngularJS / NodeJS / ReactJS Dev ?

Want to hone your skills on projects of all size and shape ?

Work in Israel & NYC ?

- Talk to me at the break

Thank you !

Boris@500Tech.com

Our next AngularJS Course  
starts this Wednesday