

BUILDING APPS WITH REACT NATIVE



maartenm@infosupport.com



@mthmulders



VARA

VARA

L

O

N

G

. W . . .
S W O R N
S W O R D
S W E D E

DEMO TIME!

Getting started

BUT THERE'S...

... Cordova/PhoneGap

- HTML, JavaScript, CSS
- web-app inside browser inside app

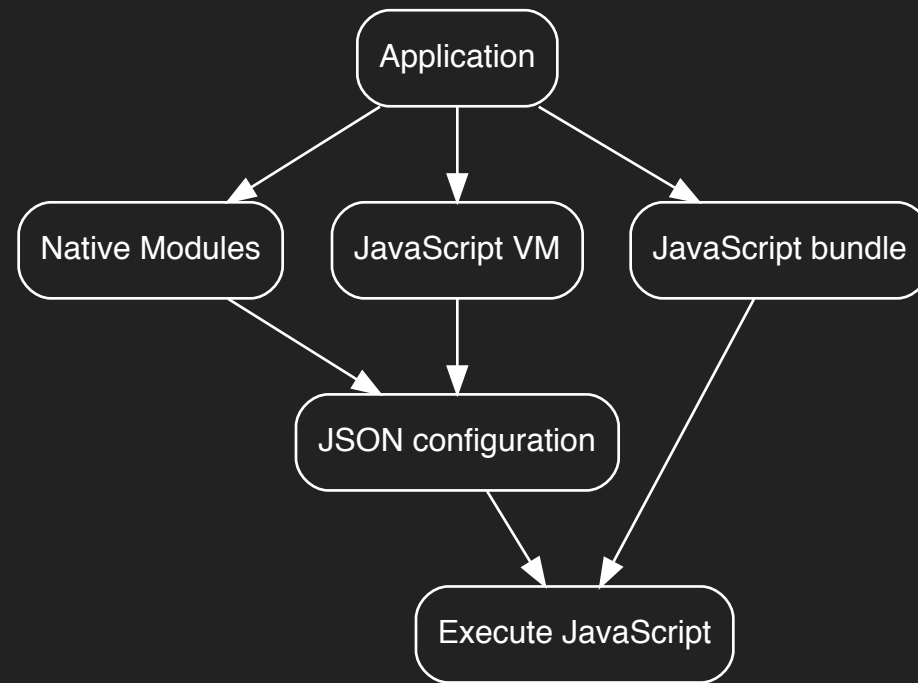
... Xamarin

- C#
- includes Mono runtime

INTERNALS

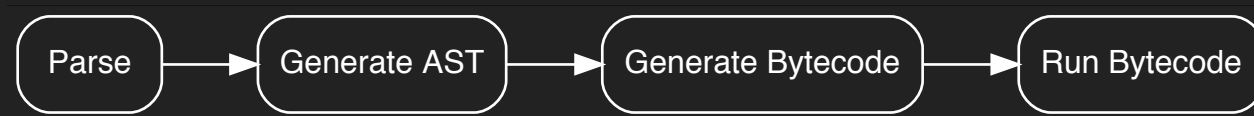
- Startup
- Execution
- Runtime

APPLICATION STARTUP

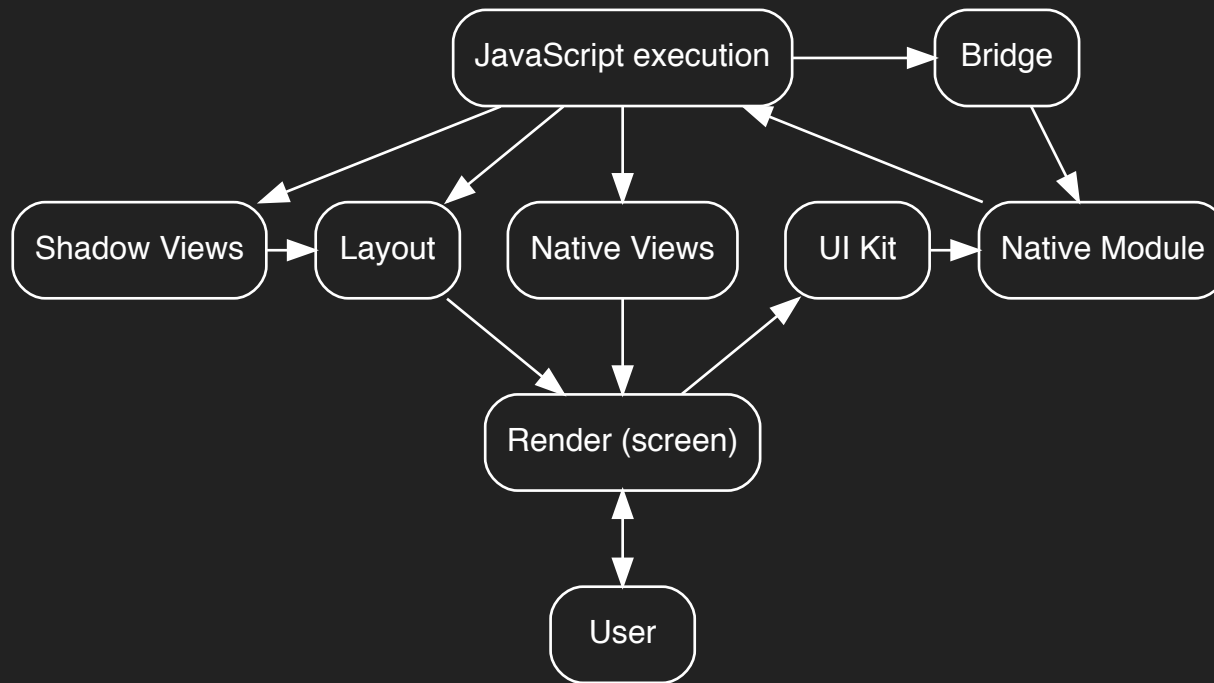


JAVASCRIPT EXECUTION

- JavaScriptCore from WebKit (Safari)
- Available on iOS, packaged on Android
- Except when debugging in Chrome → V8



PROGRAM FLOW



BUILDING YOUR APP

- React
- Styling
- State / Redux

REACT

- Declarative UI
- Component-based

COMPONENTS

```
class HelloMessage extends React.Component {
  render() {
    return <div>Hello { this.props.name }</div>;
  }
}

class App extends React.Component {
  render() {
    return <HelloMessage name={ 'Jfokus' } />;
  }
}

ReactDOM.render(<App />, document.getElementById("app"));
```

ES6 & JSX

```
class HelloMessage extends React.Component {  
  render() {  
    return <div>Hello { this.props.name }</div>;  
  }  
}
```

```
var HelloMessage = React.createClass({  
  render: function() {  
    return <div>Hello, { this.props.name }</div>;  
  }  
});
```

```
var HelloMessage = React.createClass({  
  render: function() {  
    return React.createElement('div', null,  
      'Hello ' + this.props.name);  
  }  
});
```

STYLING YOUR APP

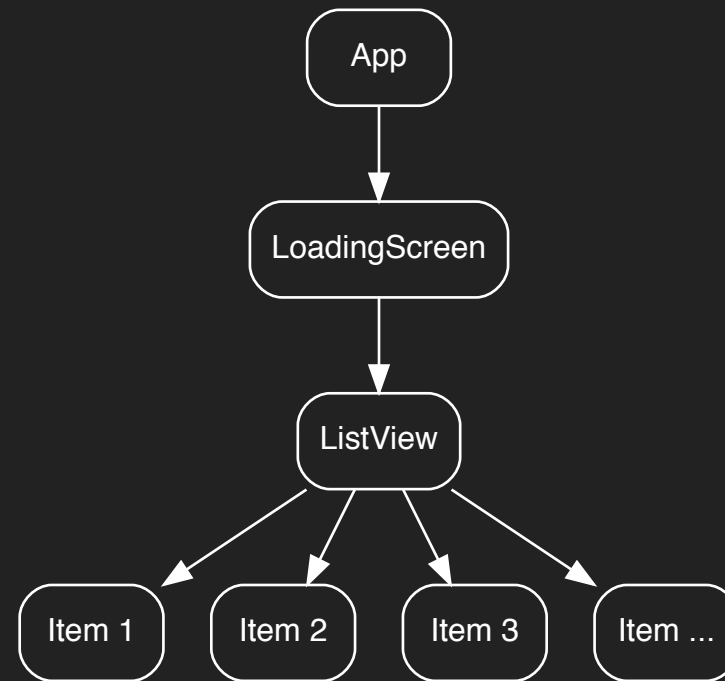
No browser, no CSS, so...?

```
import { StyleSheet, Text } from "react-native";

const styles = StyleSheet.create({
  red: {
    color: "red" // or "#ff0000", or "rgb(255, 0, 0)"
  }
});

class HelloMessage extends React.Component {
  render() {
    return <Text style={ styles.red }>...</Text>;
  }
}
```

APPLICATION STATE



Should I pass my state down through all components?!

MEET REDUX!

Core ideas:

1. State is single source of truth
 - no copies
2. State is read-only
 - state is mutated by emitting **actions**
3. Changes are made with pure functions
 - **reducers** take current state and action (+ params)
 - **always** return **new** objects

BTW, pure functions are easy to test :)

DEMO

```
const initialState = { todos: [] };
const reducer = (state = initialState, action) => {
  switch (action.type) {
    case 'ADD_TODO':
      const id = state.todos.length + 1;
      const todo = { title: action.title, complete: false, id };
    }
  };

const store = createStore(reducer);
store.dispatch({ type: 'ADD_TODO', title: 'Speak at Jfokus' });
console.log(store.getState());
```

REACT-REDUX: REACT <3 REDUX

Provide Redux' store to your app

```
import { Provider } from 'react-redux';  
const store = createStore(reducer);  
  
<Provider store={ store }>  
  <LoadingScreen />  
</Provider>
```

REACT-REDUX: REACT <3 REDUX

Connect components to Redux' store

```
import { Text } from 'react-native';
import { connect } from 'react-redux';

export const Counter = (props) => {
  return (<Text>There are { props.count } tasks</Text>);
}

const mapStateToProps = (state) => {
  return { count: state.todos.length };
}
const mapDispatchToProps = (dispatch) => {
  return { };
}

export default connect(mapStateToProps, mapDispatchToProps)
  (Counter);
```

DEMO TIME!



<https://git.io/vMEuW>

Q & A