



LECTURE 12

REACT NATIVE IMPLEMENTATION

DANIEL RYS
JAN VÁCLAVÍK

OVERVIEW

- Component implementation
 - Movies
 - Movie List
 - Movie Detail
 - Movie Search
 - Back Button
 - Loading Overlay

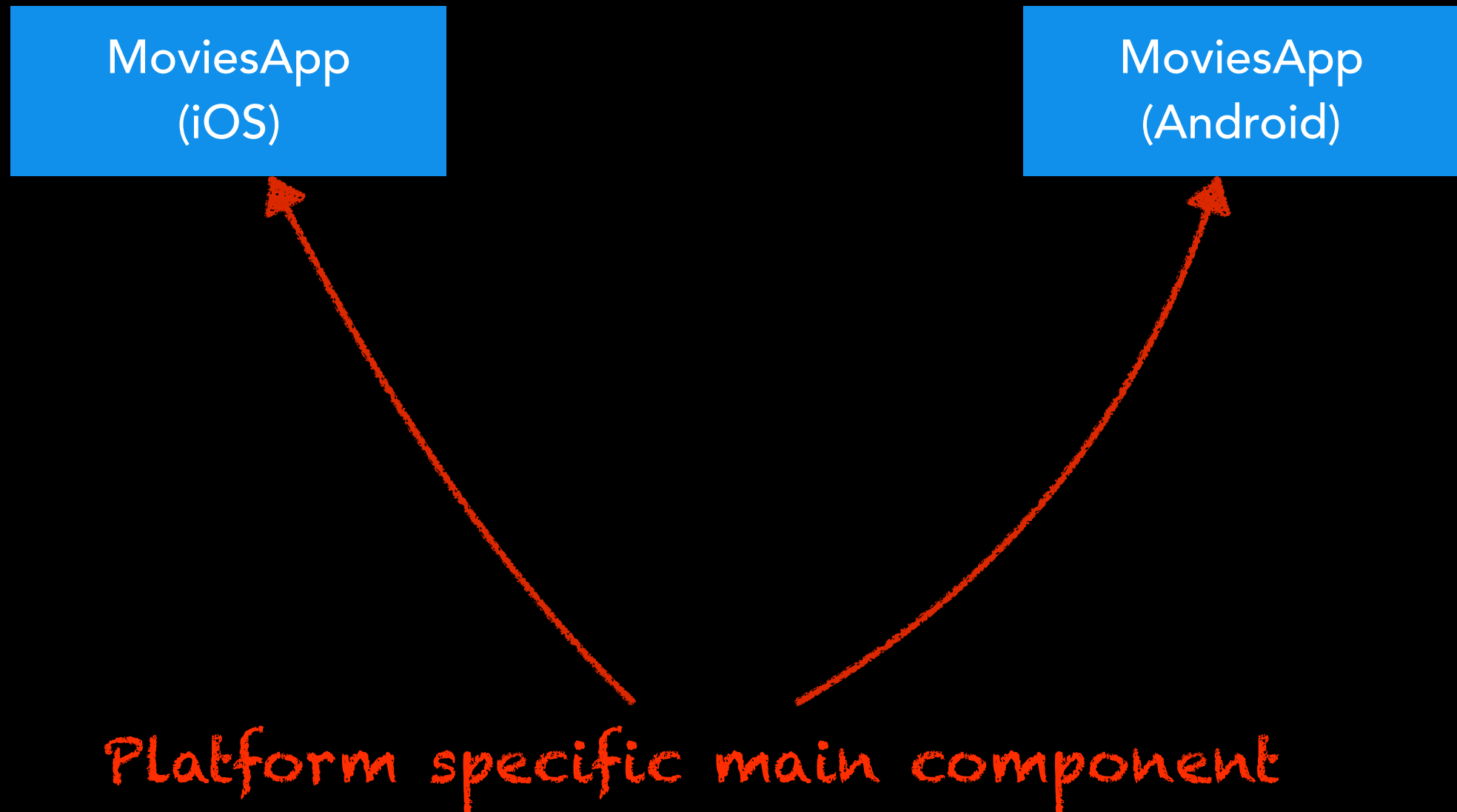
COMPONENT TREE

COMPONENT TREE

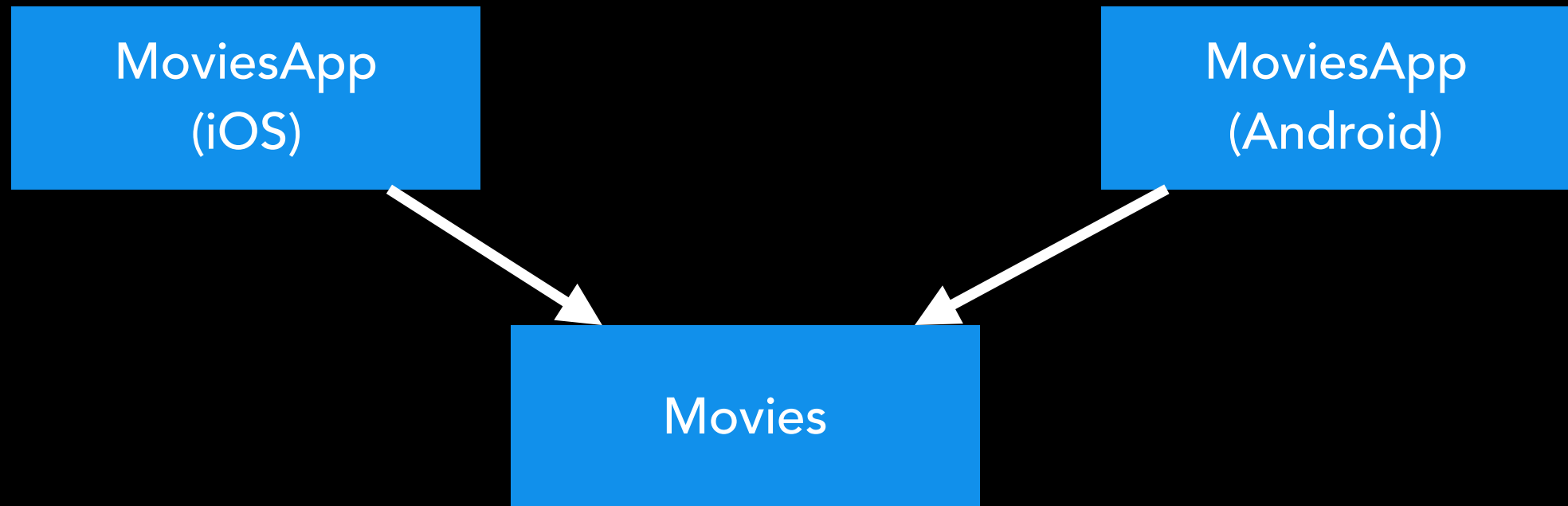
MoviesApp
(iOS)

MoviesApp
(Android)

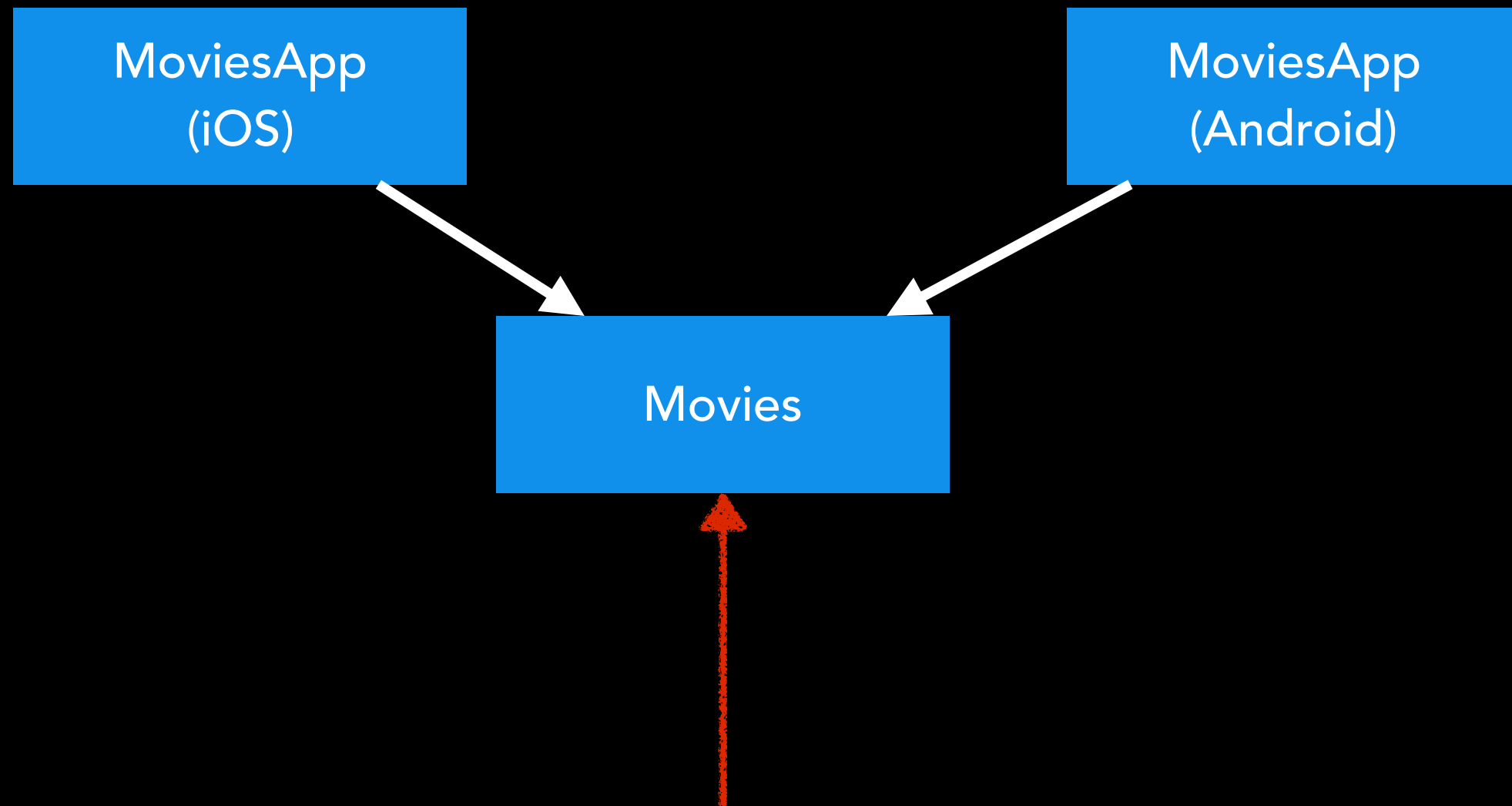
COMPONENT TREE



COMPONENT TREE

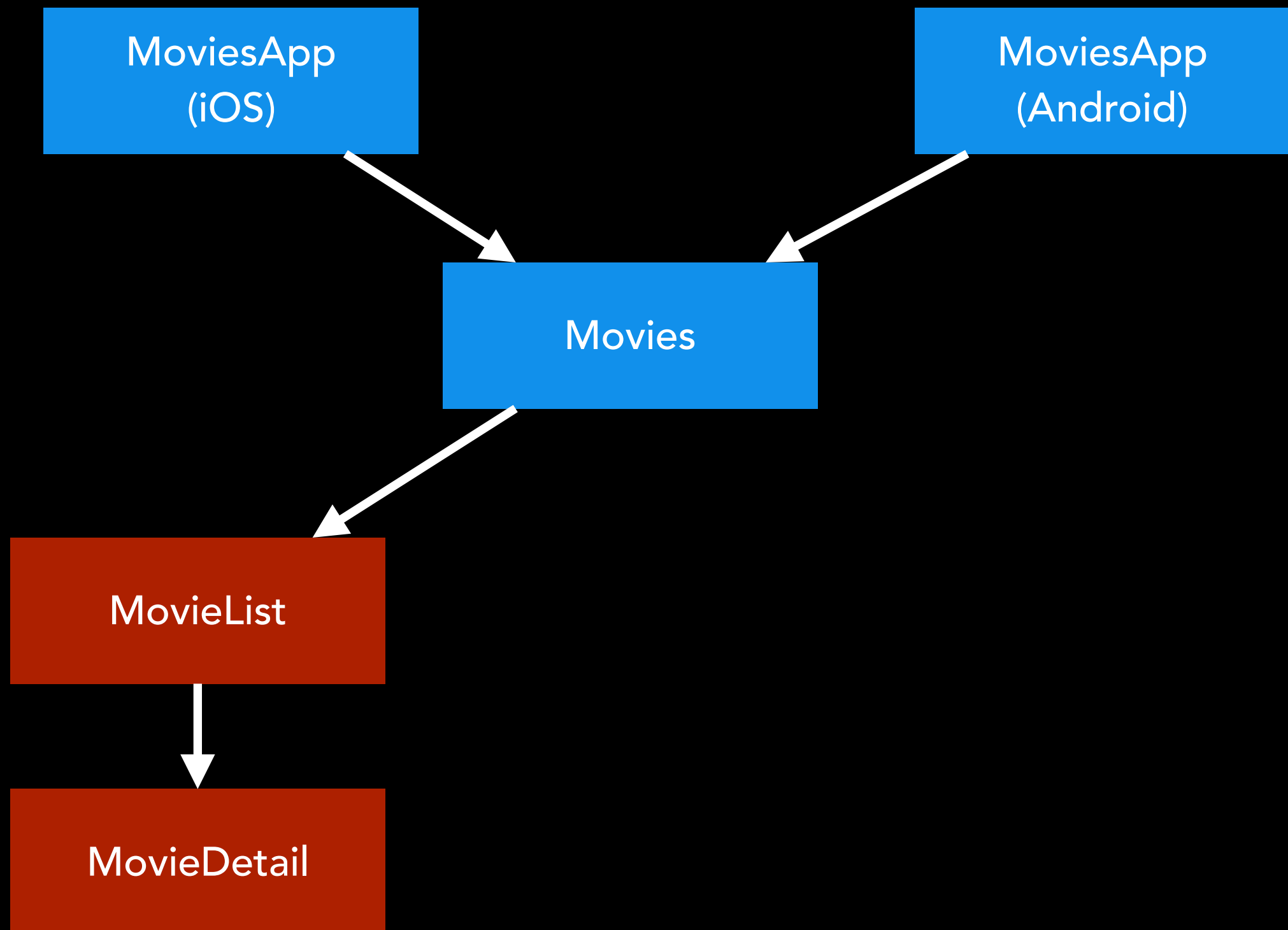


COMPONENT TREE

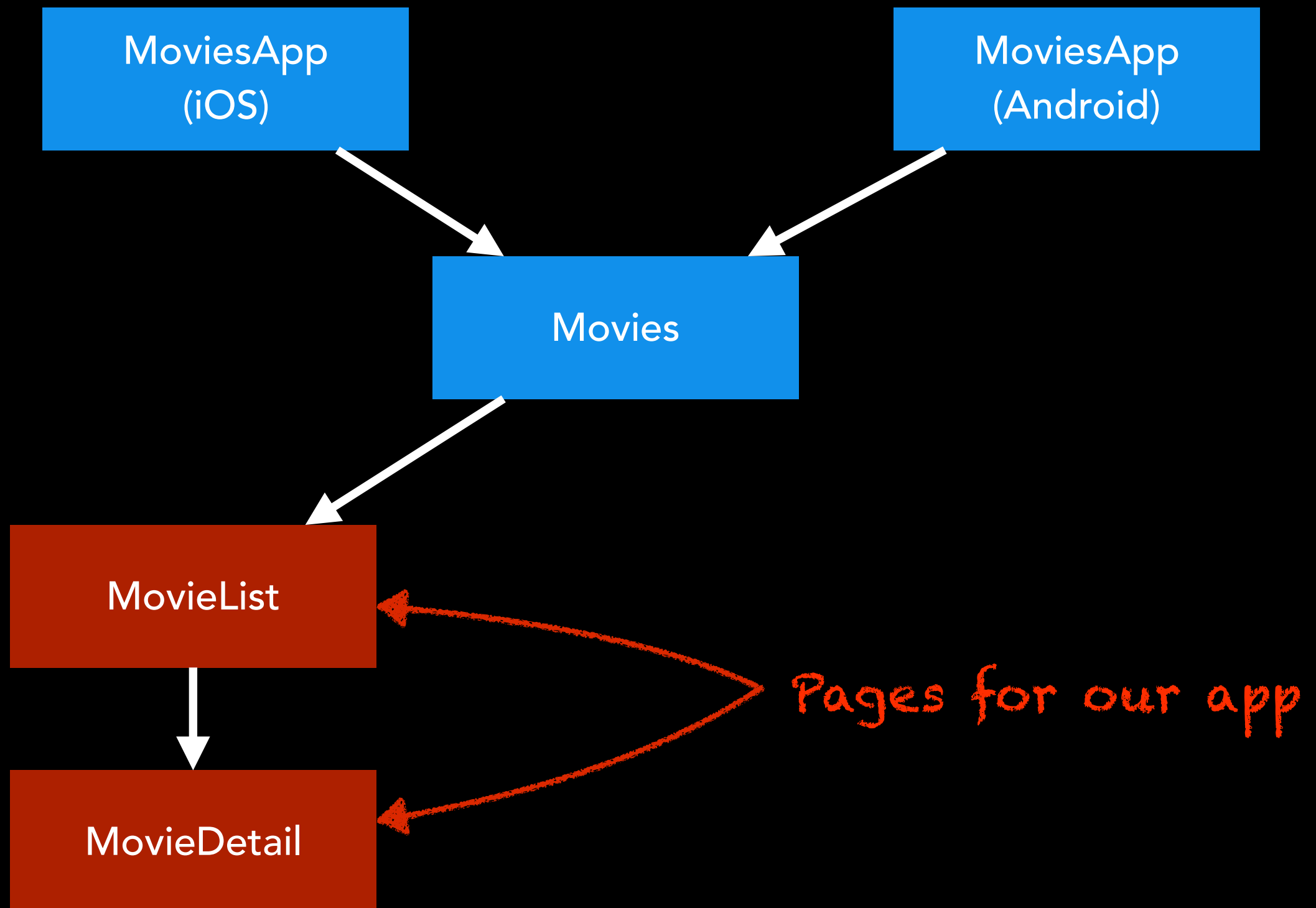


Entry point
The same app for
each platform in our case

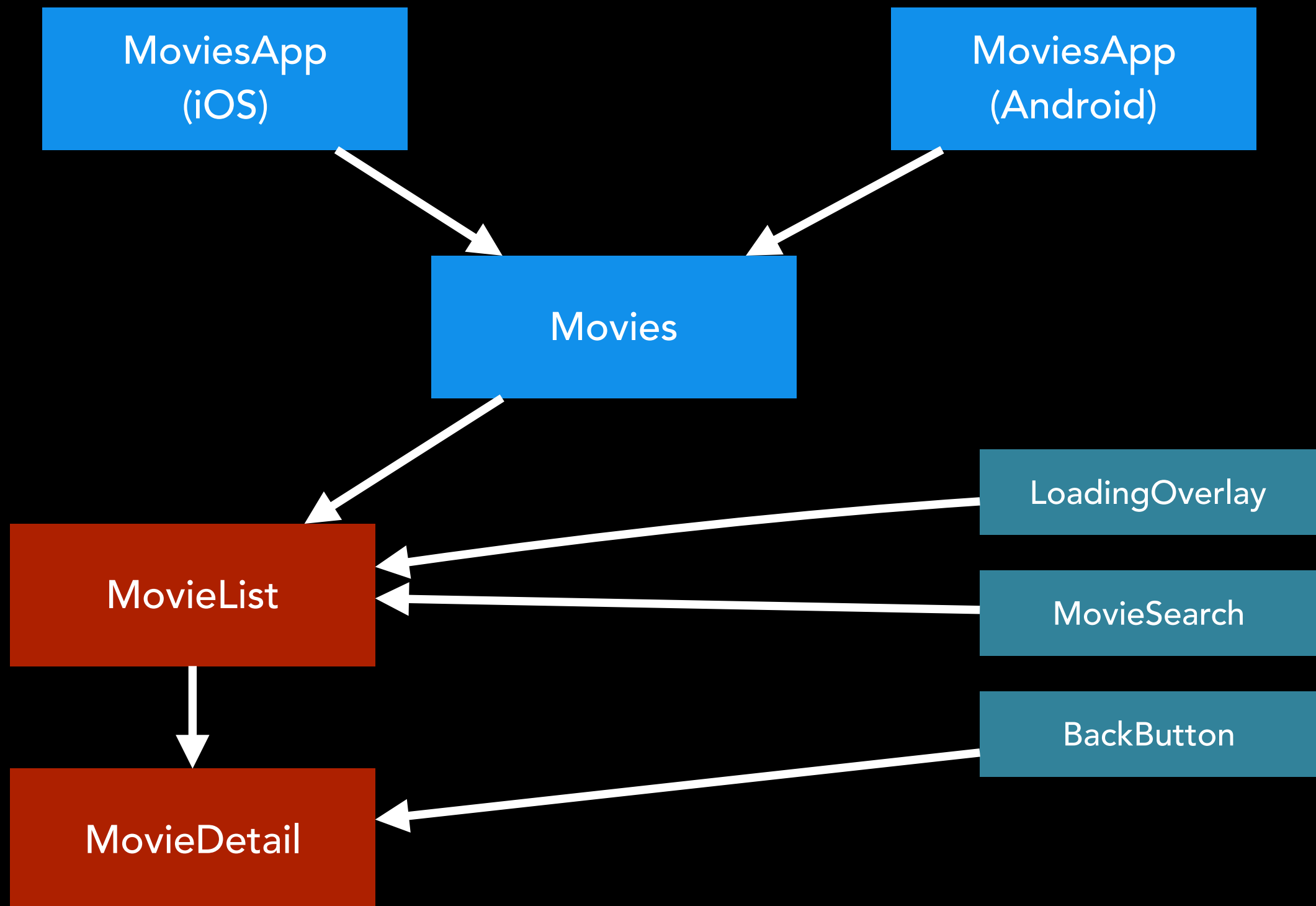
COMPONENT TREE



COMPONENT TREE

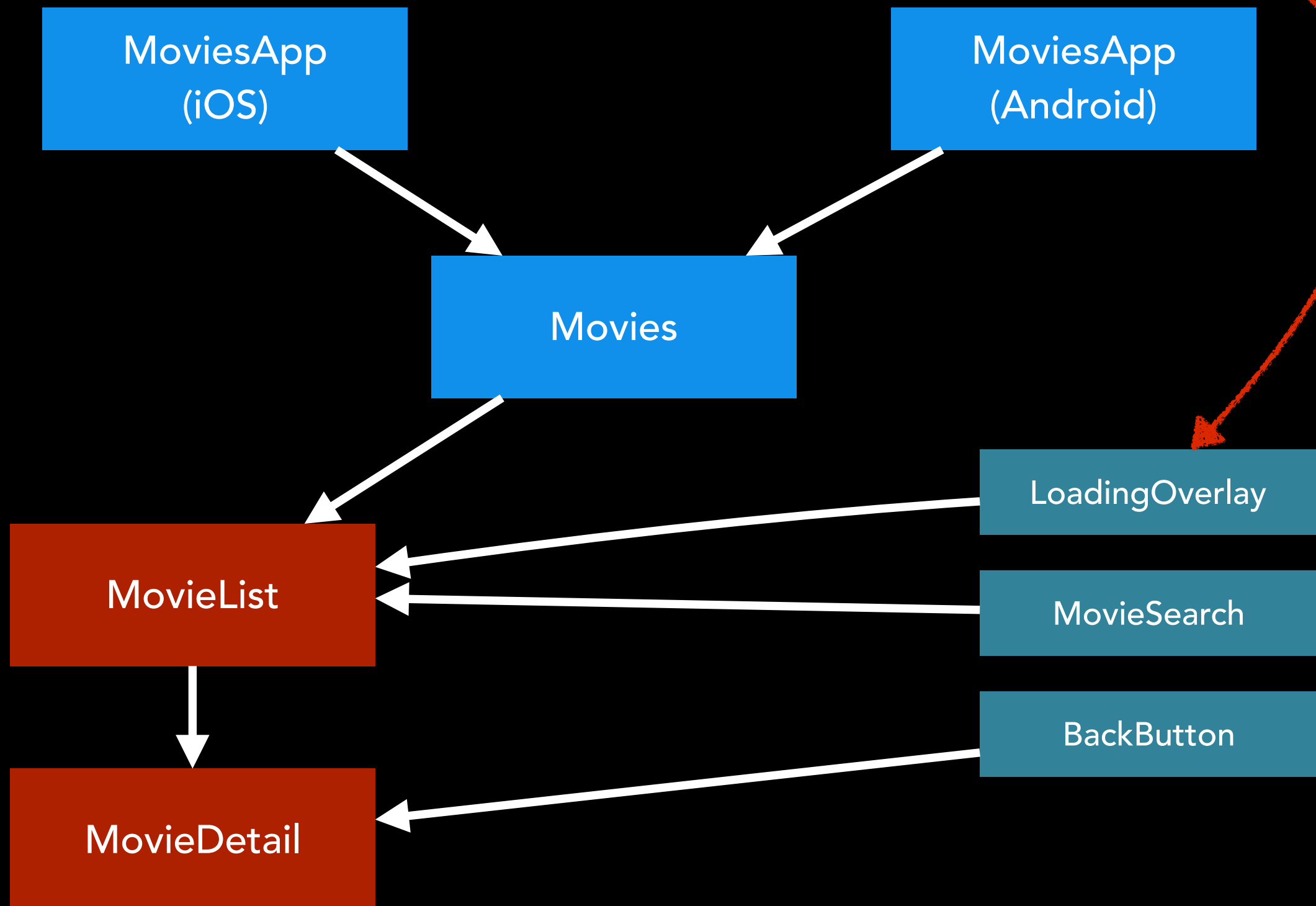


COMPONENT TREE

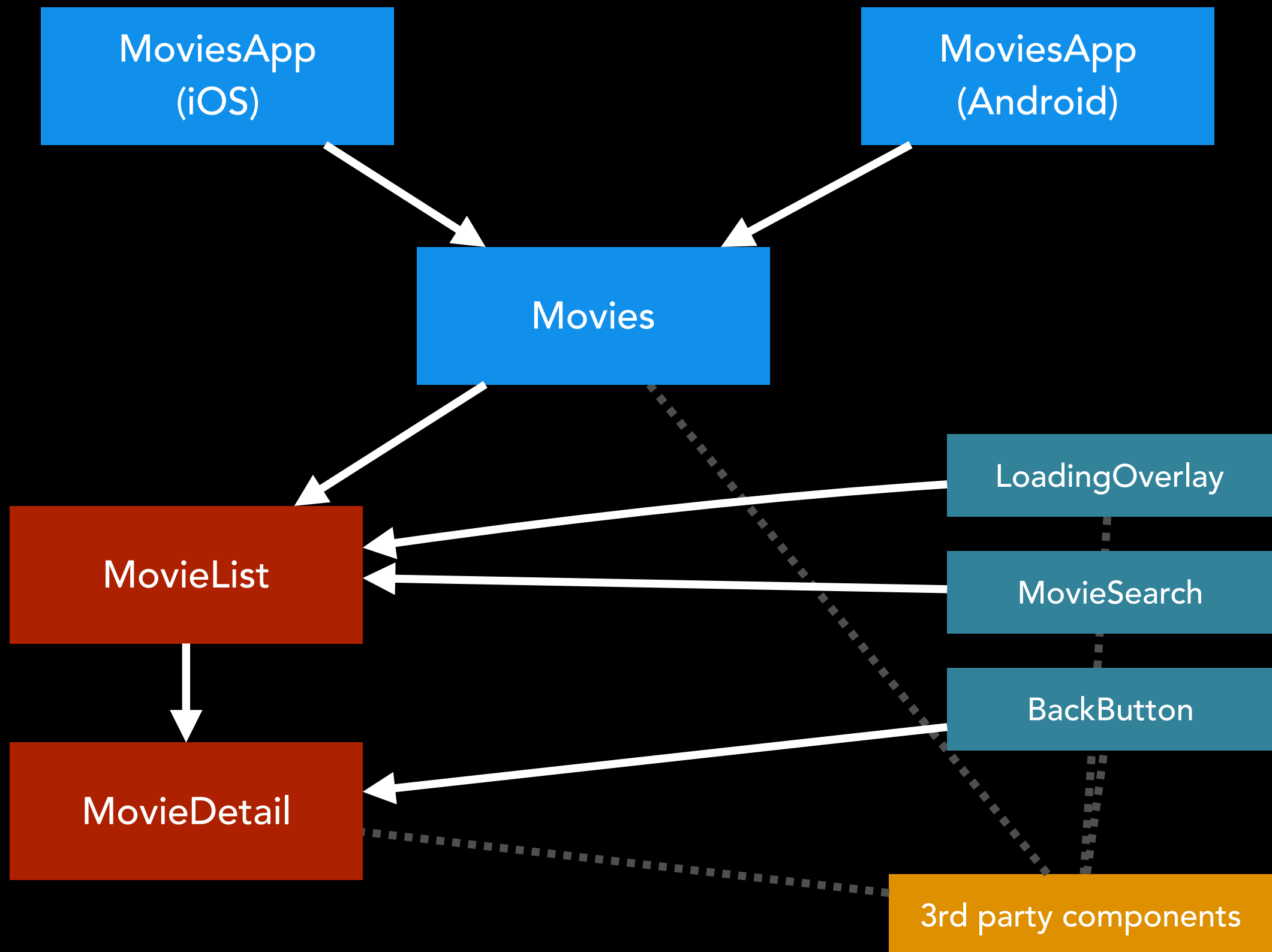


COMPONENT TREE

Smaller components



COMPONENT TREE



INIT PROJECT

```
$ react-native init MoviesApp
```

CREATE COMPONENTS

- Create directory **app/components**
- **Create entry point** (app/Movies.js)
- Create components
 - **MovieList** (app/components/MovieList.js)
 - **MovieDetail** (app/components/MovieDetail.js)

CREATE COMPONENTS

index.ios.js

```
import React, { Component } from 'react';
import {
  AppRegistry,
  Navigator
} from 'react-native';
var Movies = require("./app/Movies");

class MoviesApp extends Component {
  renderScene(route, navigator) {
    return <route.component route={route} navigator={navigator} />;
  }

  render() {
    const initialRoute = {
      component: Movies
    };
    return (
      <View style={{ flex: 1 }}>
        <Navigator
          initialRoute={initialRoute}
          renderScene={this.renderScene}/>
      </View>
    );
  }
}
```

```
AppRegistry.registerComponent('MoviesApp', () => MoviesApp);
```

INSTALL NAVBAR

```
$ npm install react-native-navbar
```


MOVIES COMPONENT

MOVIES COMPONENT

- **This component should do:**
 - Get data from API
 - Show Navbar
 - Show MovieList component in ScrollView

MOVIES CLASS

app/Movies.js

```
import React, { Component } from 'react';
var NavigationBar = require('react-native-navbar');
var MovieList = require('./components/MovieList');

import {
  View,
  Navigator,
  ScrollView
} from 'react-native';

var Movies = React.createClass({
  getInitialState: function () {
    this.getDataFromApi();
    return {
      data: [],
    }
  },

  getDataFromApi: function () {
    // Get some data from API
  },

  render: function () {
    // TODO: Show Navbar, ScrollView and MovieList
  }
});
module.exports = Movies;
```

GETTING DATA FROM API

app/Movies.js

```
getDataFromApi: function () {  
    fetch("http://api.themoviedb.org/3/movie/popular?  
api_key=4aa883f95999ec813b8bfaf319f3972b")  
    .then(response => response.json())  
    .then(json => {  
        this.setState({  
            data: json.results  
        });  
        console.log(json.results);  
    })  
    .catch((error) => {  
        console.warn(error);  
    });  
},
```

JSX

app/Movies.js

```
render: function () {  
  return (  
    <View>  
      <NavigationBar  
        tint_color="#08b1ff"  
        title={{ title: 'Movie list', tint_color: "#ffffff" }}/>  
      <ScrollView>  
        <View>  
          <MovieList data={this.state.data}  
            navigator={this.props.navigator}/>  
        </View>  
      </ScrollView>  
    </View>  
  );  
}
```

MOVIE LIST COMPONENT

MOVIE LIST COMPONENT

- **This component should do:**
 - Show retrieved data as ListView items
 - After click on movie, MovieDetail should be shown

JSX

app/components/MovieList.js

```
import React, {Component} from 'react';
import {View, Text, ListView} from 'react-native';

var MovieList = React.createClass({
  getInitialState: function () {
    var ds = new ListView.DataSource({rowHasChanged: (r1, r2) => r1 !== r2});
    return {
      dataSource: ds.cloneWithRows(this.props.data),
    };
  },
  renderRow: function (movie) {
    // TODO
  },
  componentWillReceiveProps: function (nextProps) {
    // TODO
  },
  render: function () {
    return (
      <View>
        <ListView
          dataSource={this.state.dataSource}
          renderRow={(rowData) => this.renderRow(rowData)} />
      </View>
    );
  }
});

module.exports = MovieList;
```


JSX

app/components/MovieList.js

```
renderRow: function (movie) {  
  return (  
    <View>  
      <Text>{movie.title}</Text>  
    </View>  
  );  
}
```

JSX

app/components/MovieList.js

```
componentWillReceiveProps: function (nextProps) {  
  this.setState({  
    dataSource: this.state.dataSource.cloneWithRows(nextProps.data)  
  });  
},
```

JSX

app/components/MovieList.js

```
renderRow: function (movie) {
  return (
    <View>
      <TouchableHighlight onPress={() => this.showMovieDetail(movie)}
underlayColor='#dddddd'>
        <View style={styles.item}>
          <Image style={styles.thumbnail} source={{uri: "http://
image.tmdb.org/t/p/w500"+movie.poster_path}}/>
          <View style={styles.rightContainer}>
            <Text style={styles.title}>{movie.title}</Text>
            <Text>{movie.vote_average} / 10</Text>
          </View>
        </View>
      </TouchableHighlight>
      <View style={styles.separator}/>
    </View>
  );
},
```

SHOW MOVIE DETAIL CLICK

app/components/MovieList.js

```
showMovieDetail: function (movie) {  
  this.props.navigator.push({  
    title: movie.title,  
    component: MovieDetail,  
    passProps: {movie}  
  });  
},
```

STYLES

app/components/MovieList.js

```
var styles = StyleSheet.create({
  item: {
    flex: 1,
    flexDirection: "row",
    justifyContent: 'center',
    alignItems: 'center',
    padding: 10,
  },
  rightContainer: {
    flex: 1,
    justifyContent: 'center',
    paddingLeft: 10,
  },
  loading: {
    justifyContent: 'center',
    alignItems: 'center'
  },
  thumbnail: {
    width: 50,
    height: 50,
    borderRadius: 25,
  },
  separator: {
    height: 1,
    backgroundColor: '#dddddd'
  }
});
```

MOVIE DETAIL COMPONENT

MOVIE DETAIL COMPONENT

- **This component should do:**
 - Show data about specific movie
 - Show navigation bar with back button

MOVIE DETAIL JSX

app/components/MovieDetail.js

```
import React, { Component } from 'react';
import {
  View,
  StyleSheet,
  ScrollView,
  Text,
  Image,
} from 'react-native';
var NavigationBar = require('react-native-navbar');

var MovieDetail = React.createClass({
  render: function () {
    // TODO
  },
});
module.exports = MovieDetail;
```


MOVIE DETAIL JSX

app/components/MovieDetail.js

```
render: function () {
  var movie = this.props.route.passProps.movie;
  return (
    <View style={styles.container}>
      <NavigationBar style={styles.navbar} tintColor="#08b1ff"
        title={{ title: movie.title, tintColor: "#ffffff" }}
        leftButton={{title: "Back", tintColor: "#ffffff" }}
      />
      <ScrollView style={styles.scroll}>
        <Image
          style={styles.image}
          source={{uri: "http://image.tmdb.org/t/p/
w500/"+movie.backdrop_path}}/>
        <Text style={styles.overview}>{movie.overview}</Text>
      </ScrollView>
    </View>
  );
},
```

STYLES

app/components/MovieDetail.js

```
var styles = StyleSheet.create({
  container: {
    backgroundColor: "white",
    flex: 1,
  },
  image: {
    flex: 1,
    flexDirection: "row",
    height: 200,
    padding: 10
  },
  description: {
    padding: 10,
    fontSize: 15,
    color: '#656565'
  },
  scroll: {
    flex: 1,
    backgroundColor: "#ffffff",
  },
  navbar: {
    backgroundColor: "#08b1ff",
    flex: 1
  },
  overview: {
    padding: 10
  }
});
```

MOVIE SEARCH COMPONENT

MOVIE SEARCH COMPONENT

- **This component should do:**
 - Add search bar in Movies component
 - After change text field get data from API

MOVIE SEARCH COMPONENT

app/Movies.js

```
var MovieSearch = require('./components/MovieSearch');

...

setSearchText: function (searchString) {
  this.setState({searchString: searchString})
  if(searchString.length > 1)
    setTimeout(() =>{
      this.getDataFromApi(searchString);
    }, 500);
},

...

render: function () {
  return (
    ...
    <MovieSearch onChange={this.setSearchText} />
    ...
  );
}
```

MOVIE SEARCH COMPONENT

app/components/MovieSearch.js

```
var React = require('react-native');
var {View, Text, TextInput, StyleSheet} = React;

var MovieSearch = React.createClass({
  getInitialState() {
    this.setState({searchString: ""});
    return {};
  },

  render: function () {
    return (
      <View style={styles.searchBar}>
        <TextInput
          style={styles.input}
          value={this.state.searchString}
          onChangeText={this.props.onChange}
          ref="searchString"
          placeholder="Search"
          placeholderTextColor="#eee" />
        </View>
      );
  }
});

module.exports = MovieSearch;
```

MOVIE SEARCH COMPONENT

app/components/MovieSearch.js

```
var styles = StyleSheet.create({
  searchBar: {
    padding: 5,
    backgroundColor: "#08b1ff",

  },
  input: {
    paddingLeft: 10,
    textAlign: "center",
    fontSize: 14,
    height: 28,
    flex: .1,
    borderWidth: 1,
    borderColor: "#eee",
    backgroundColor: "#08b1ff",
    borderRadius: 5,
    color: "#fff"
  },
});
```

MOVIE SEARCH COMPONENT

app/components/BackButton.js

```
getDataFromApi: function(searchString) {  
  var url = "http://api.themoviedb.org/3/" + ((searchString) ?  
    'search/movie?api_key=4aa883f95999ec813b8bfaf319f3972b&query=' + searchString  
    :  
    'movie/popular?api_key=4aa883f95999ec813b8bfaf319f3972b');  
  
  fetch(url)  
    .then(response => response.json())  
    .then(json => {  
      this.setState({  
        loading: false,  
        data: json.results  
      });  
      console.log(json.results);  
    })  
    .catch((error) => {  
      console.warn(error);  
    });  
},
```


BACK BUTTON COMPONENT

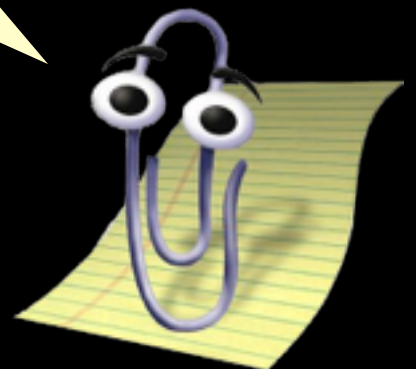
BACK BUTTON COMPONENT

- **This component should do:**
 - Show back button with icon

BACK BUTTON COMPONENT

- **This component should do:**
 - Show back button with icon

That's the easy one, right?



BACK BUTTON COMPONENT

app/components/MovieDetail.js

```
var BackButton = require('./BackButton');
```

...

```
render: function () {
  var movie = this.props.route.passProps.movie;
  return (
    <View style={styles.container}>
      <NavigationBar style={styles.navbar} tintColor="#08b1ff"
        title={{ title: movie.title, tintColor: "#ffffff" }}
        leftButton={
          <BackButton onPress={() => this.props.navigator.pop()}
            title="Back"
            tintColor="#fff" />
        } />
      <ScrollView style={styles.scroll}>
        <Image
          style={styles.image}
          source={{uri: "http://image.tmdb.org/t/p/
w500/"+movie.backdrop_path}} />
        <Text style={styles.overview}>{movie.overview}</Text>
      </ScrollView>
    </View>
  );
},
```

BACK BUTTON COMPONENT

app/components/BackButton.js

```
import React, { Component } from 'react';
import {Image, Text, View, TouchableHighlight, StyleSheet} from 'react-native';

var BackButton = React.createClass({
  render: function () {
    const icon = 'http://i.stack.imgur.com/4ByUK.png';
    return (
      <View>
        <TouchableHighlight onPress={this.props.onPress}
underlayColor="transparent">
          <View style={styles.container}>
            <Image
              source={{ uri: icon }}
              style={styles.icon}/>
            <Text style={{ paddingTop: 3, color: this.props.tintColor }}>
              >{this.props.title}</Text>
            </View>
          </TouchableHighlight>
        </View>
      );
    }
});

module.exports = BackButton;
```

BACK BUTTON COMPONENT

app/components/BackButton.js

```
var styles = StyleSheet.create({
  container: {
    flexDirection: 'row',
    paddingTop: 10,
  },
  icon: {
    width: 20,
    height: 20,
    marginRight: 5,
  }
});
```

LOADING OVERLAY COMPONENT

LOADING OVERLAY COMPONENT

- **This component should do:**
 - Show spinner while loading content from server

INSTALL DEPENDENCIES

```
$ npm install react-native-overlay  
$ npm install react-native-blur
```

- **For adding to XCode see**
<https://github.com/brentvatne/react-native-overlay>

LOADING OVERLAY COMPONENT

app/components/LoadingOverlay.js

```
var React = require('react-native');
var Overlay = require('react-native-overlay');
var BlurView = require('react-native-blur').BlurView;
var {View, ActivityIndicatorIOS, StyleSheet} = React;

var LoadingOverlay = React.createClass({
  getDefaultProps: function () {
    return {
      isVisible: false
    }
  },
  render: function () {
    return (
      <Overlay isVisible={this.props.isVisible}>
        <BlurView style={styles.background} blurType="dark">
          <ActivityIndicatorIOS
            size="large"
            animating={true}
            style={styles.spinner}/>
        </BlurView>
      </Overlay>
    );
  }
});

module.exports = LoadingOverlay;
```

LOADING OVERLAY COMPONENT

`app/components/LoadingOverlay.js`

```
var styles = StyleSheet.create({  
  background: {  
    flex: 1,  
    alignItems: "stretch",  
    justifyContent: 'center',  
  }  
});
```

UPDATE STATE BEFORE GETTING DATA

app/Movies.js

```
import React, { Component } from 'react';
var NavigationBar = require('react-native-navbar');
var MovieList = require('./components/MovieList');
var LoadingOverlay = require('./components/LoadingOverlay');
import {
  View,
  Navigator,
  ScrollView
} from 'react-native';

var Movies = React.createClass({
  getInitialState: function () {
    this.getDataFromApi();
    return {
      data: [],
      loading: true,
    }
  },

  getDataFromApi: function () {
    // Get some data from API
  },

  render: function () {
    // TODO: Show Navbar, ScrollView and MovieList
  }
});
module.exports = Movies;
```

UPDATE STATE BEFORE GETTING DATA

app/Movies.js

```
import React, { Component } from 'react';
var NavigationBar = require('react-native-navbar');
var MovieList = require('./components/MovieList');
var LoadingOverlay = require('./components/LoadingOverlay');
import {
  View,
  Navigator,
  ScrollView
} from 'react-native';
```

```
var Movies = React.createClass({
  getInitialState: function () {
    this.getDataFromApi();
    return {
      data: [],
      loading: true,
    }
  },
```

```
  getDataFromApi: function () {
    // Get some data from API
  },
```

```
  render: function () {
    // TODO: Show Navbar, ScrollView and MovieList
  }
```

```
});
module.exports = Movies;
```

Set loading on by default



UPDATE STATE AFTER GETTING DATA

app/Movies.js

```
getDataFromApi: function () {  
    fetch("http://api.themoviedb.org/3/movie/popular?  
api_key=4aa883f95999ec813b8bfaf319f3972b")  
    .then(response => response.json())  
    .then(json => {  
        this.setState({  
            data: json.results,  
            loading: false  
        });  
        console.log(json.results);  
    })  
    .catch((error) => {  
        console.warn(error);  
    });  
},
```

UPDATE STATE AFTER GETTING DATA

app/Movies.js

```
getDataFromApi: function () {  
    fetch("http://api.themoviedb.org/3/movie/popular?  
api_key=4aa883f95999ec813b8bfaf319f3972b")  
    .then(response => response.json())  
    .then(json => {  
        this.setState({  
            data: json.results,  
            loading: false  
        });  
        console.log(json.results);  
    })  
    .catch((error) => {  
        console.warn(error);  
    });  
},
```

Set loading off after getting data

SHOW LOADING COMPONENT

app/Movies.js

```
render: function () {
  return (
    <View>
      <LoadingOverlay isVisible={this.state.loading}/>
      <NavigationBar
        tintColors="#08b1ff"
        title={{ title: 'Movie list', tintColors: "#ffffff" }}/>
      <ScrollView>
        <View>
          <MovieList data={this.state.data}
            navigator={this.props.navigator}/>
        </View>
      </ScrollView>
    </View>
  );
}
```


SHOW LOADING COMPONENT

app/Movies.js

```
render: function () {  
  return (  
    <View>  
      <LoadingOverlay isVisible={this.state.loading}/>  
      <NavigationBar  
        tint="#08b1ff"  
        title={{ title: 'Movie list', tint: "#ffffff" }}/>  
      <ScrollView>  
        <View>  
          <MovieList data={this.state.data}  
            navigator={this.props.navigator}/>  
        </View>  
      </ScrollView>  
    </View>  
  );  
}
```

Add Loading component

THAT'S ALL!

Now you know everything about
Ionic, Ionic2 and React Native

REPOSITORY

<https://github.com/jvaclavik/react-native-movies-app>

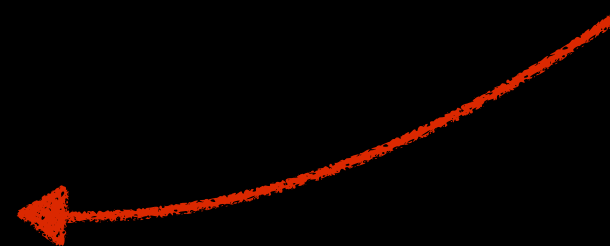
PREPARE ON SATURDAY!

- **Finish your projects**
- Prepare your presentations
- **Study for test** (seriously)
- Ask us if you need something

PREPARE ON SATURDAY!

- **Finish your projects**
- Prepare your presentations
- **Study for test** (seriously)
- Ask us if you need something

It's not so easy...





THANK YOU

WWW.USERTECHNOLOGIES.COM

STAY IN TOUCH



[@janvaclavik](https://twitter.com/janvaclavik)



linkedin.com/in/janvaclavik



facebook.com/johny89



janvaclavik.cz



[@danielrys](https://twitter.com/danielrys)



linkedin.com/in/rysdaniel



facebook.com/daniel.rys.56

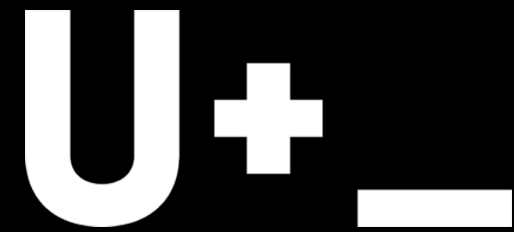
FOLLOW CZECH IONIC COMMUNITY



@ionicCzech

facebook.com/groups/ionicCzech

QUESTIONS?



JAN VÁCLAVÍK

@janvaclavik

DANIEL RYS

@danielrys

WWW.USERTECHNOLOGIES.COM