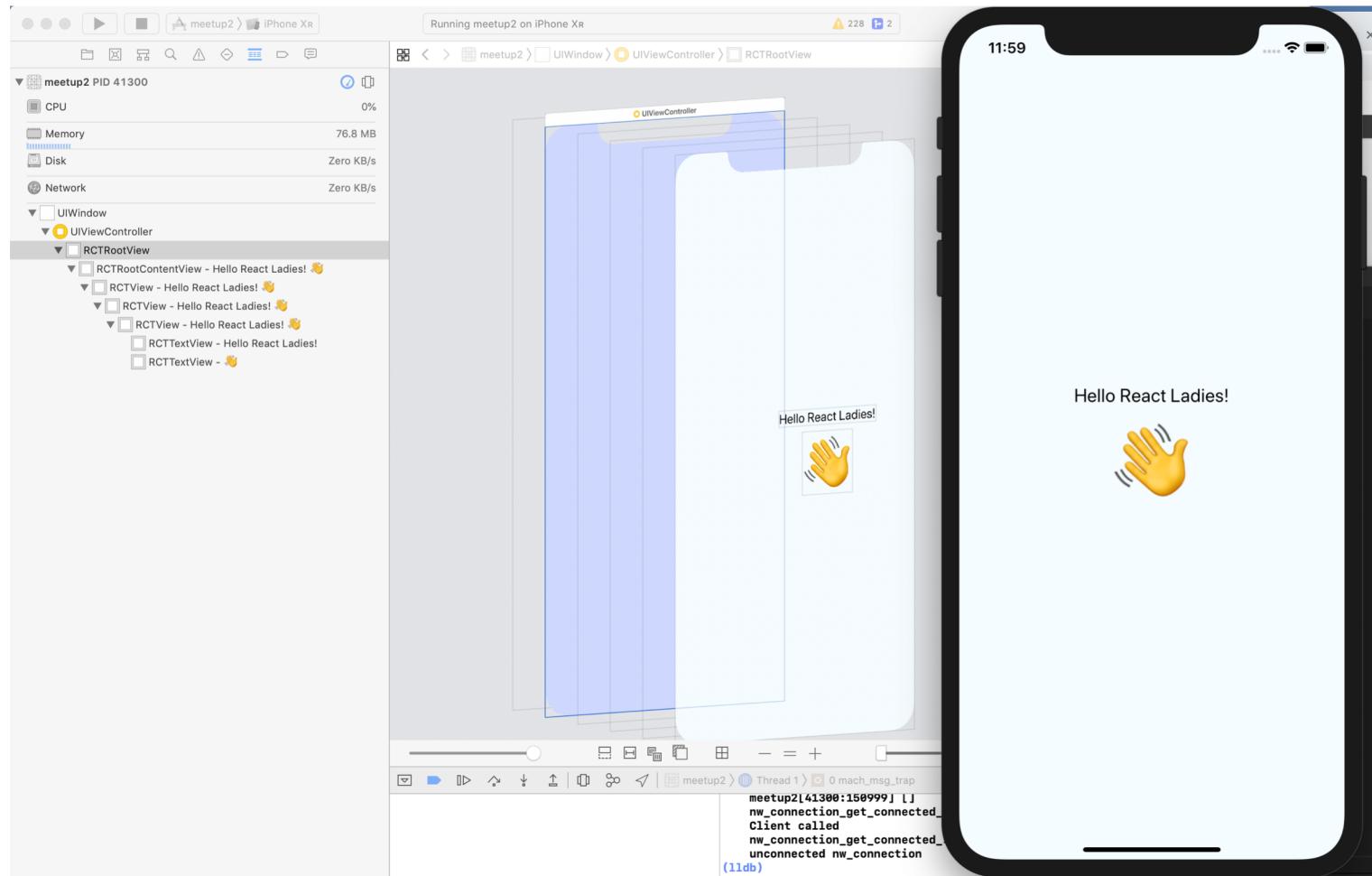
A large, semi-transparent watermark of the React Native logo (an atom-like symbol) is positioned in the upper left and lower left corners of the slide.

How React Native Works

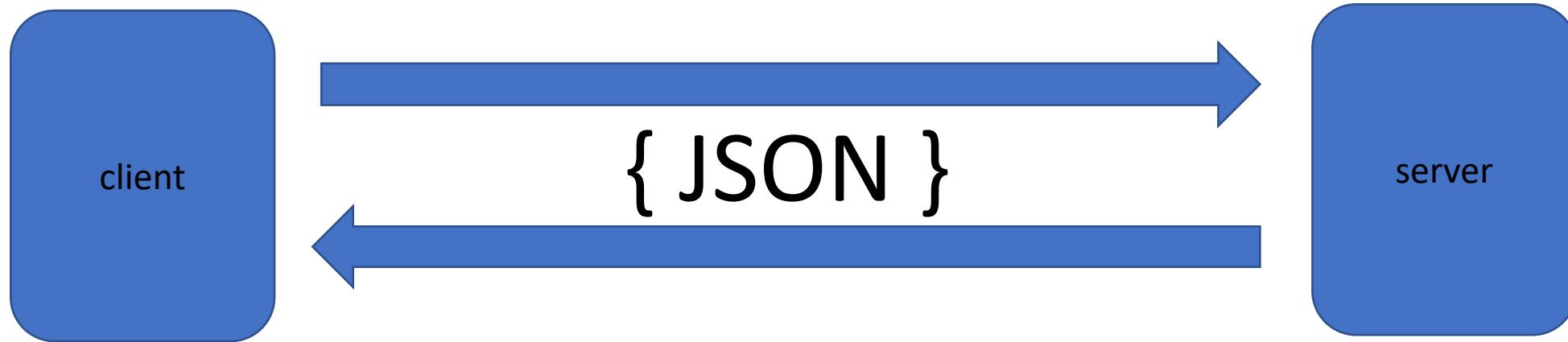
a look at the current and future architecture

React Native is Native



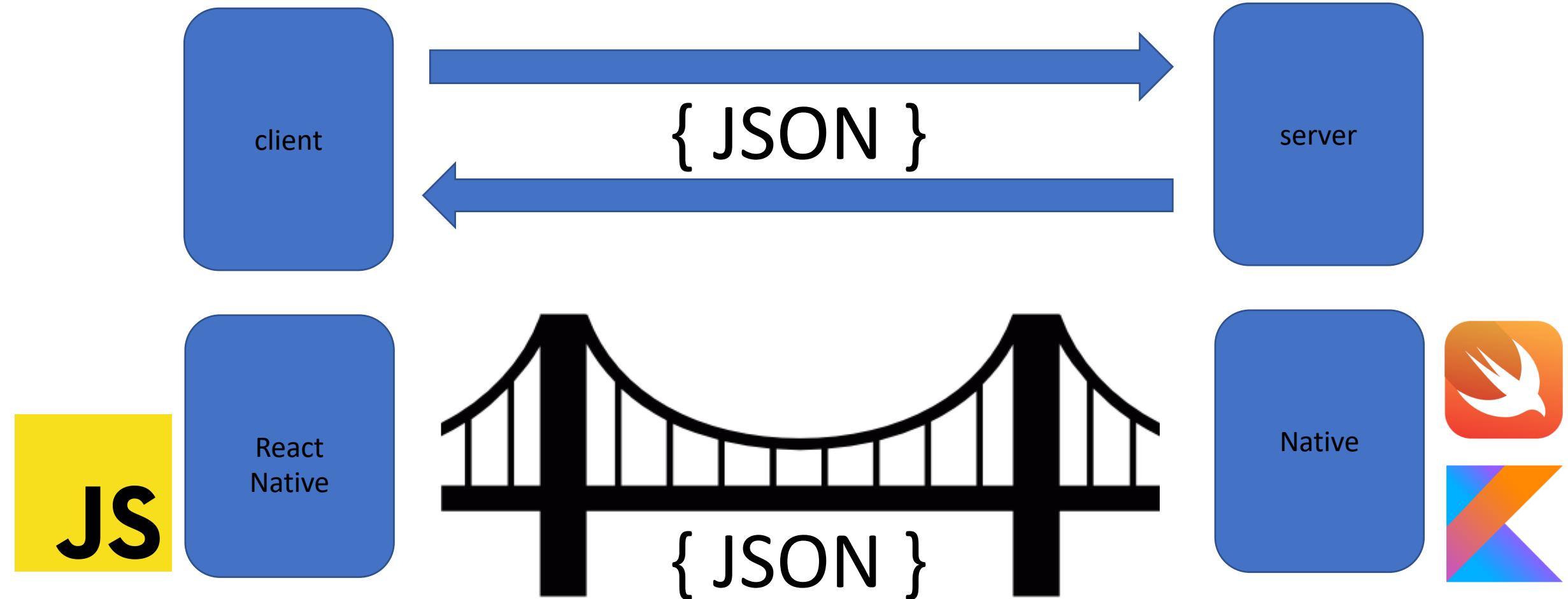
Language interoperability

- different languages interact as part of the same system



Language interoperability

- different languages interact as part of the same system



Native Modules

```
@interface MyNativeComponent : NSObject  
<RCTBridgeModule> @end
```

```
import {NativeModules} from 'react-native';  
  
let MyNativeComponent =  
  NativeModules.MyNativeComponent;
```

Report abuse

 React Native Community

Quality code from the React Native Community

World <https://github.com/react-native-community/react-native-community>

Repositories 64 People 26 Projects 0

Find a repository... Type: All Language: All

Top languages

- JavaScript
- Java
- Objective-C
- TypeScript
- Shell

Most used topics

- react-native
- android
- ios
- react
- polyfill

People 26 >

react-native-tab-view
A cross-platform Tab View component for React Native

react-native tabs swipe tabbar swipeview pager-component

TypeScript ★ 2,957 630 MIT 2 issues need help Updated 6 minutes ago

react-native-webview
React Native Cross-Platform WebView

Objective-C ★ 1,146 377 MIT 30 issues need help Updated an hour ago

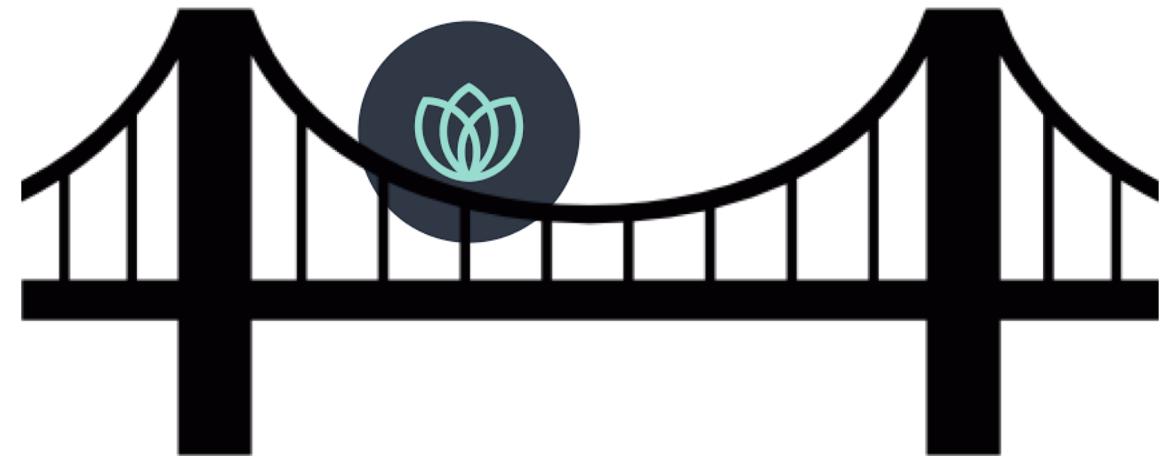
react-native-audio-toolkit
Cross-platform audio library for React Native

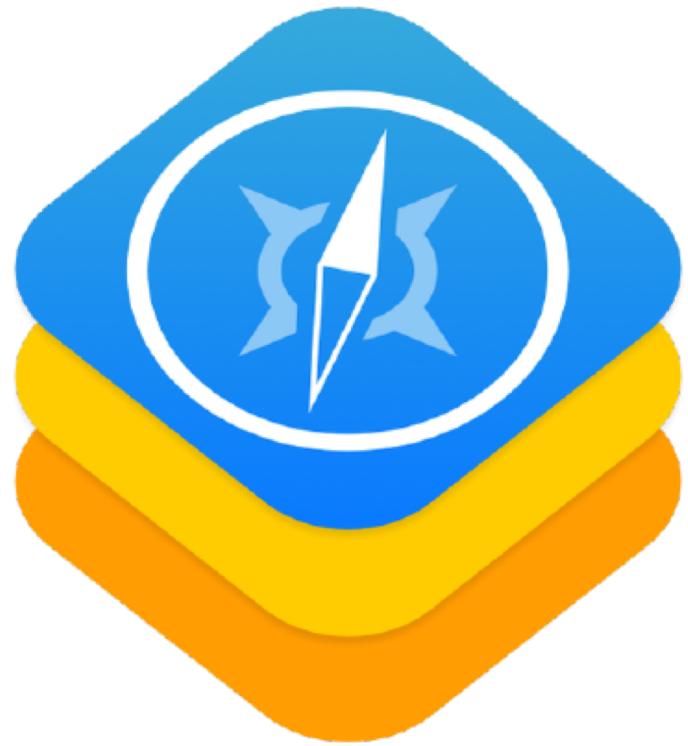
cross-platform reactnative audio-library

Java ★ 452 196 MIT 2 issues need help Updated 2 hours ago

Yoga

cross-platform
layout engine that
implements
Flexbox





BABEL



8



BABEL



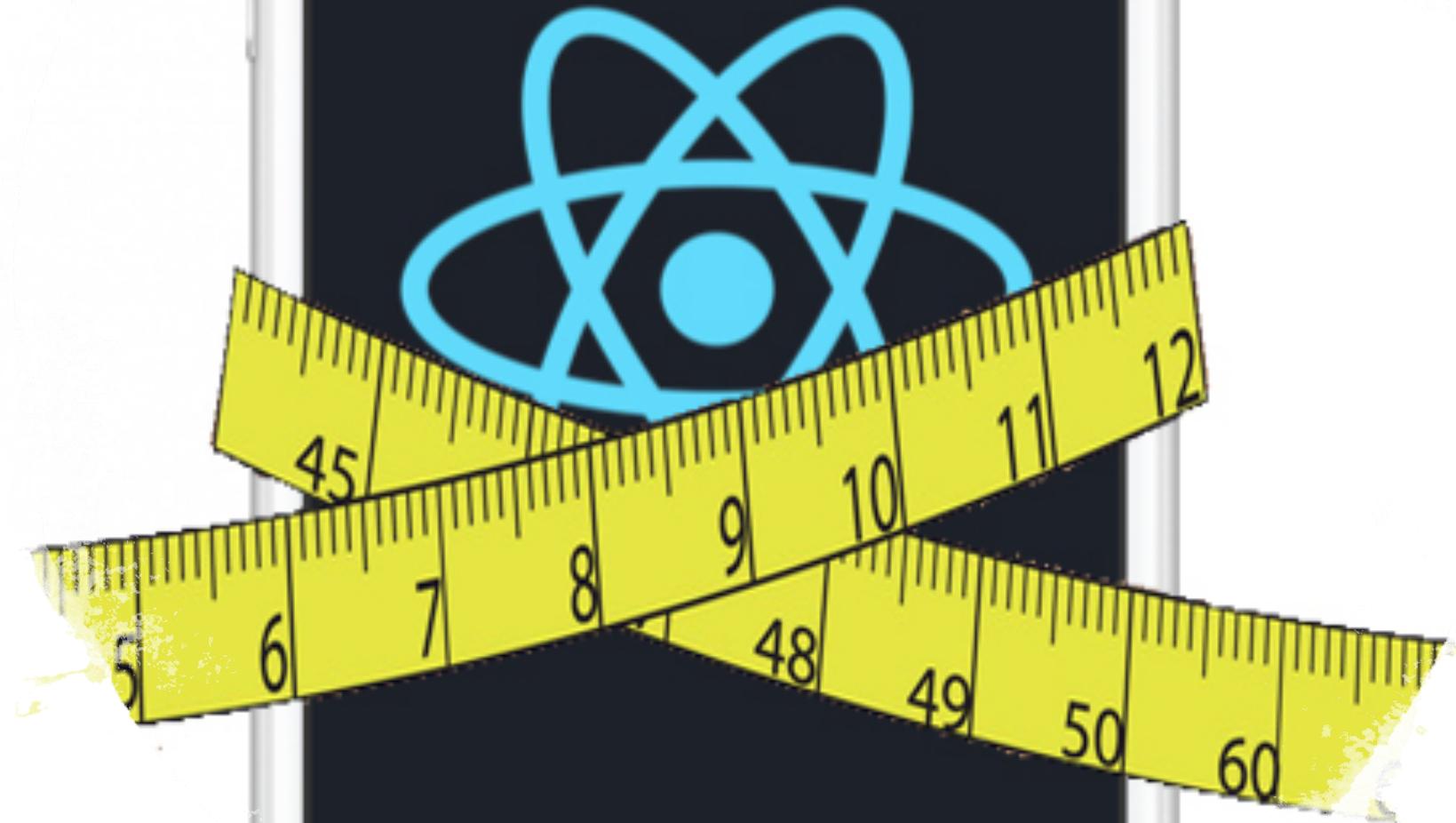
v8



{ JSON }

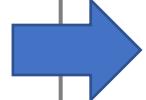


React Native Fabric



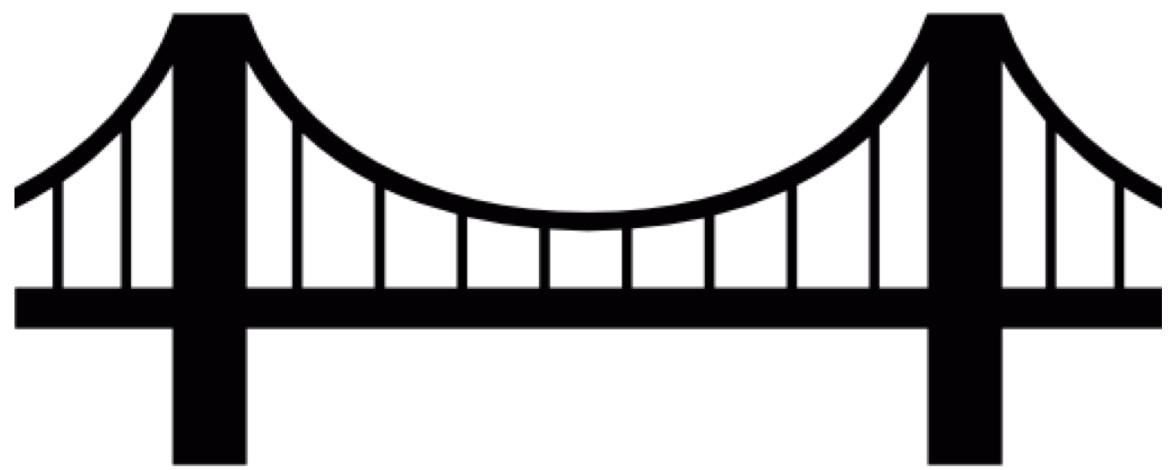
JSI (JavaScript Interface)

swappable



ChakraCore
SpiderMonkey
Nashorn
V8
JavaScriptCore
InScript
Rhino
JerryScript
Etc.....





```
class CustomTextInput extends React.Component {
  constructor(props) {
    super(props);
    // create a ref to store the textInput DOM element
    this.textInput = React.createRef();
    this.focusTextInput = this.focusTextInput.bind(this);
  }

  focusTextInput() {
    // Explicitly focus the text input using the raw DOM API
    // Note: we're accessing "current" to get the DOM node
    this.textInput.current.focus();
  }

  render() {
    // tell React that we want to associate the <input> ref
    // with the `textInput` that we created in the constructor
    return (
      <div>
        <input
          type="text"
          ref={this.textInput} />

        <input
          type="button"
          value="Focus the text input"
          onClick={this.focusTextInput}
        />
      </div>
    );
  }
}
```

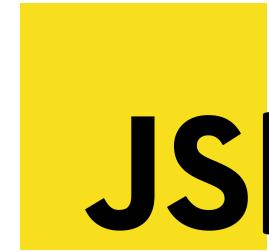
LOADING

Please feel free to wait forever.



A horizontal progress bar consisting of two rectangular bars. The first bar is white and represents 20% completion. The second bar is black and represents the remaining 80%.

20%



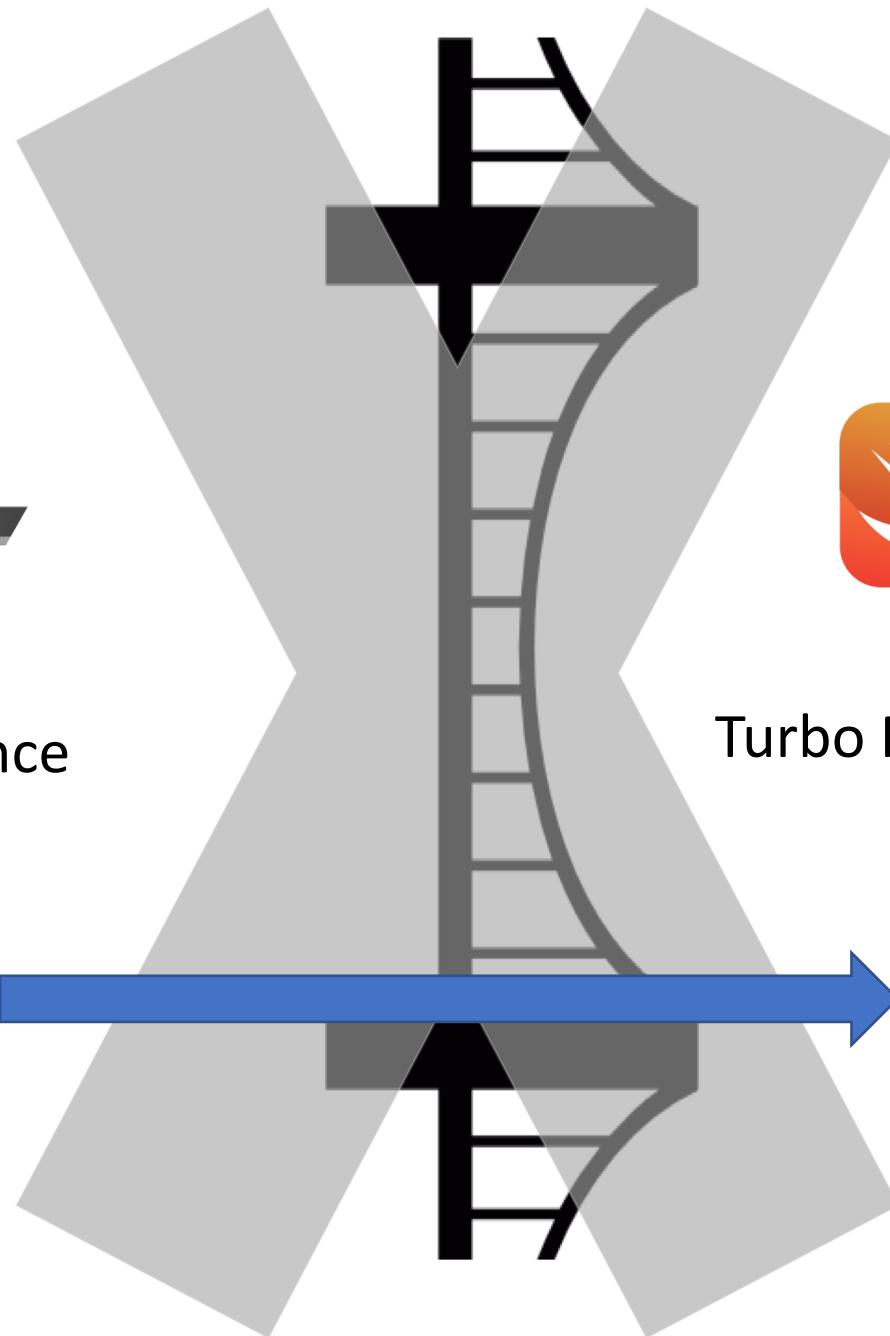
BABEL



Object Reference



CodeGen



Turbo Modules C++ Object

[react-native-community / discussions-and-proposals](#)

[Code](#) [Issues 22](#) [Pull requests 4](#) [Insights](#)

React Native Fabric (UI-Layer Re-architecture) #4

[Open](#) kelset opened this issue on Jul 31, 2018 · 62 comments

 kelset commented on Jul 31, 2018 • edited

[Member](#) +  [...](#)

Intro

With this issue I'd like to try and create a "one stop" for all the information available around the future re-architecture of the UI-Layer of React Native, codenamed "Fabric".

Terminology

- JSI: JavaScript Interface, it's a unified lightweight general purpose API for (theoretically) any JavaScript virtual machine. It enables every other piece of the rearchitecture. ([dedicated issue](#))
- TurboModules: re-architecture of NativeModules, also using JSI. ([dedicated issue](#))
- CodeGen: a tool to "automate" the compatibility between JS and native side. ([dedicated issue](#))

TL;DR

From [@axe-fb's blogpost](#), here's a temporary description of Fabric (*please consider that this is not yet finalized, it may change in the future*)

In the current architecture, all UI operations (like creating native views, managing children, etc.) are handled by a native module called UIManagerModule. The React Reconciler sends UI commands over the bridge, which are eventually handled by this module and delegated to UIImplementation. This in turn creates shadow nodes that represent the layout tree and are passed to Yoga to determine the relative co-ordinates based on the Flex box styles that are passed in from JS.

In the new system, the UI operations are directly exposed to JavaScript as functions using the JSI

Assignees
No one assigned

Labels
[Transparency](#)

Projects
None yet

Milestone
No milestone

Notifications
[Subscribe](#) 
You're not receiving notifications from this thread.

31 participants


[react-native-community / discussions-and-proposals](#)

[Code](#) [Issues 22](#) [Pull requests 4](#) [Insights](#)

TurboModules (NativeModules Re-architecture) #40

[Open](#) fkgozali opened this issue on Oct 12, 2018 · 27 comments

 fkgozali commented on Oct 12, 2018 • edited by kelset

[...](#)

Introduction

This is a place for discussions around the upcoming "TurboModule" feature.

- Full proposal here: <https://github.com/react-native-community/discussions-and-proposals/blob/master/proposals/0002-Turbo-Modules.md>
- Previous discussions: #11

Terminology

- Fabric: just the UI layer re-architecture, to take full advantage of concurrent React architecture. ([dedicated issue](#))
- JSI: JavaScript Interface, it's a unified lightweight general purpose API for (theoretically) any JavaScript virtual machine. It enables every other piece of the rearchitecture. ([dedicated issue](#))
- CodeGen: a tool to "automate" the compatibility between JS and native side. ([dedicated issue](#))

TL;DR

From [@axe-fb's blogpost](#), here's a temporary description of TurboModules (*please consider that this is not yet finalized, it may change in the future*)

The JSI system can also be used to call leverage device capabilities like bluetooth or other sensors by exposing functions that JS can call. This is similar to how browsers expose functions like navigator.geolocation.getCurrentPosition that, when invoked in JavaScript, trigger the respective

Assignees
No one assigned

Labels
[Transparency](#)

Projects
None yet

Milestone
No milestone

Notifications
[Subscribe](#) 
You're not receiving notifications from this thread.

11 participants


[react-native-community / discussions-and-proposals](#)

[Code](#) [Issues 22](#) [Pull requests 4](#) [Insights](#)

JSI (JavaScript Interface) & JSC (JavaScript Core) Discussion #91

[Open](#) kelset opened this issue on Jan 24 · 7 comments

 kelset commented on Jan 24 • edited

[Member](#) +  [...](#)

Assignees
No one assigned

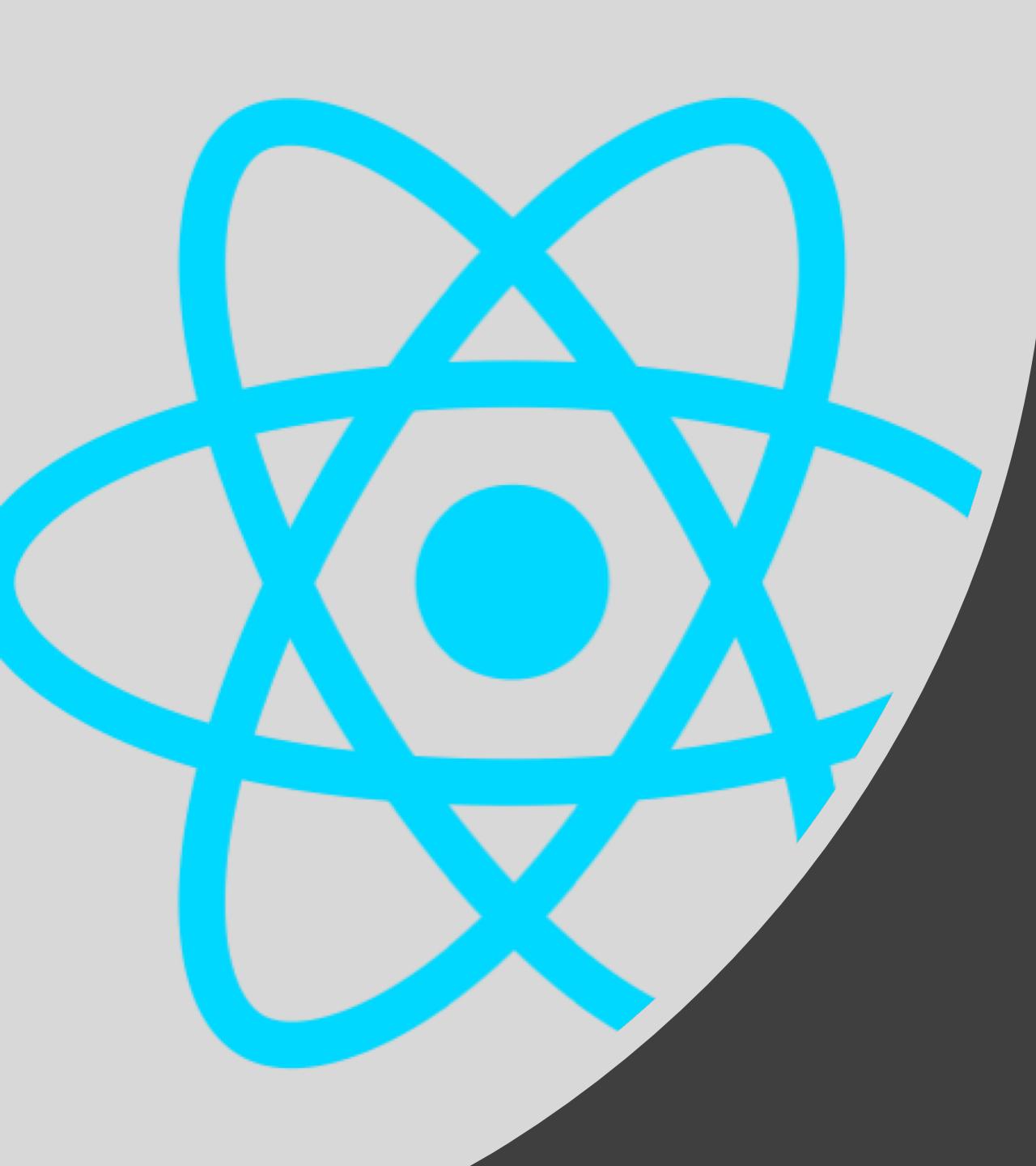
Labels
[Transparency](#)

Projects
None yet

Milestone
No milestone

Notifications
[Subscribe](#) 
You're not receiving notifications from this thread.

6 participants

Thanks!

~ Anna Garcia

@turnipdabeets