



OBJECT PARTNERS

Native-centric Approach Hybrid App Development



Ionic, React Native, NativeScript

Torey Lomenda, Chief Technologist

Exceptional results with Exceptional people

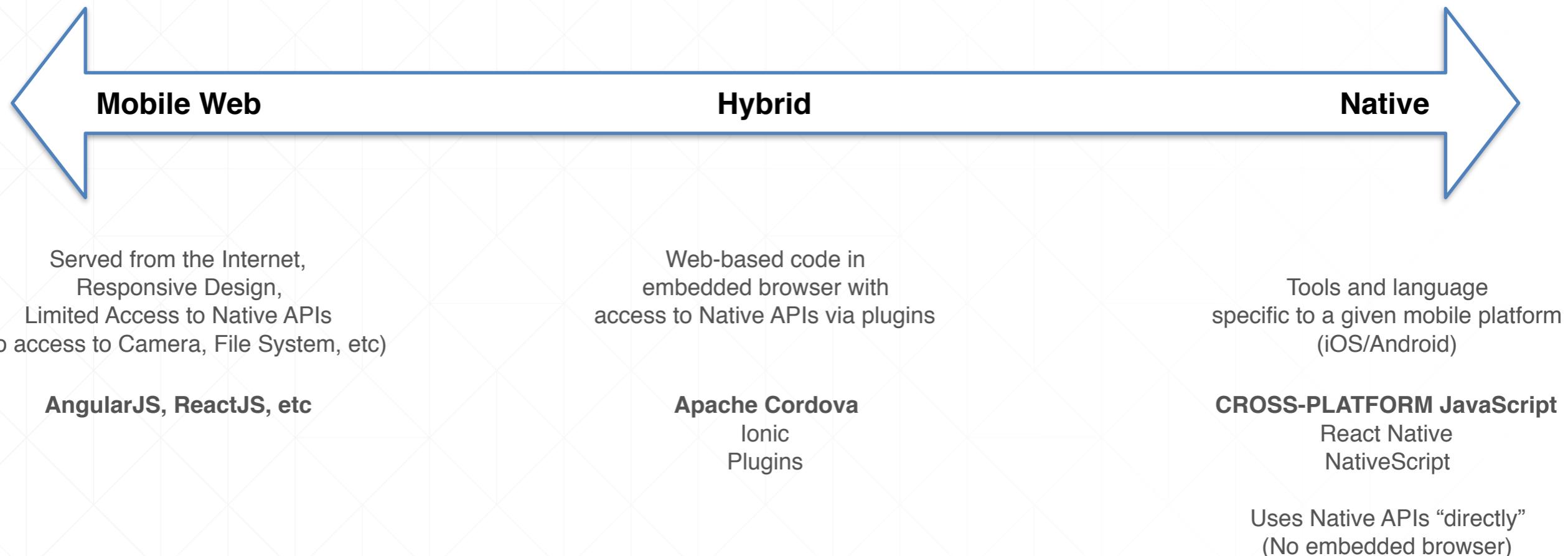


About Us

- Founded in 1996
- Software Technology Expertise
 - Web Front-end and Mobile Applications,
 - Microservices, Real-time Data
 - Continuous Delivery
- > 100 Employees
 - Twin Cities, Omaha, Chicago
 - Average Tenure over 5 years



HYBRID APP DEVELOPMENT WITH JAVASCRIPT (MIXING NATIVE AND WEB)





Exploring Hybrid...

Provide a *glimpse* into my journey exploring Hybrid and Cross-platform JavaScript and how well they ***mix*** with Native APIs and components.

Native-centric

Support mixing pure native components (iOS/Android) with JavaScript components enabling “best of both” while accepting that both are first class citizens



Why Native-centric?

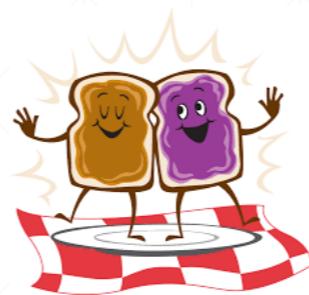
You have an existing native app and want to leverage JavaScript components or skills.

Native provides the fastest performance and UX.

What are advantages embedding hybrid/cross-platform JavaScript into a natively developed app?

One Codebase to Rule Them All....Leads to Many Hacks to Fight Them All

"I am not a fan of the ***all-in*** approach. I feel better knowing if/when I need to use a pure native component I can. I like technologies that compliment each other..."



Devices are Personal

Provide an intimate **UX** to get things done anytime, anywhere.



Native gives you the best chance for **UX** *but can come at a premium.*



Hybrid Web

Use web (HTML/JS/CSS) skills WITHOUT having to intimately know iOS and Android

“We can use skills we already have to build apps?”



Cross-platform JavaScript

Use front-end JavaScript skills to call into Native APIs.

As described in a React Native blog:

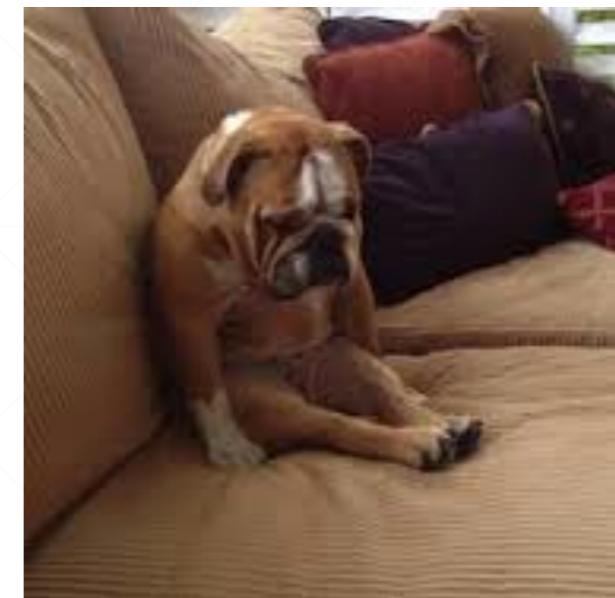
“...Have access to the power of the native environment, but also iterate quickly leveraging our existing JavaScript infrastructure.”



Inflated Expectations



Everything is AWESOME



...UNTIL IT IS NOT :-(



OBJECT PARTNERS

Ionic & Cordova

Bridging Native and Web Views

objectpartners.com



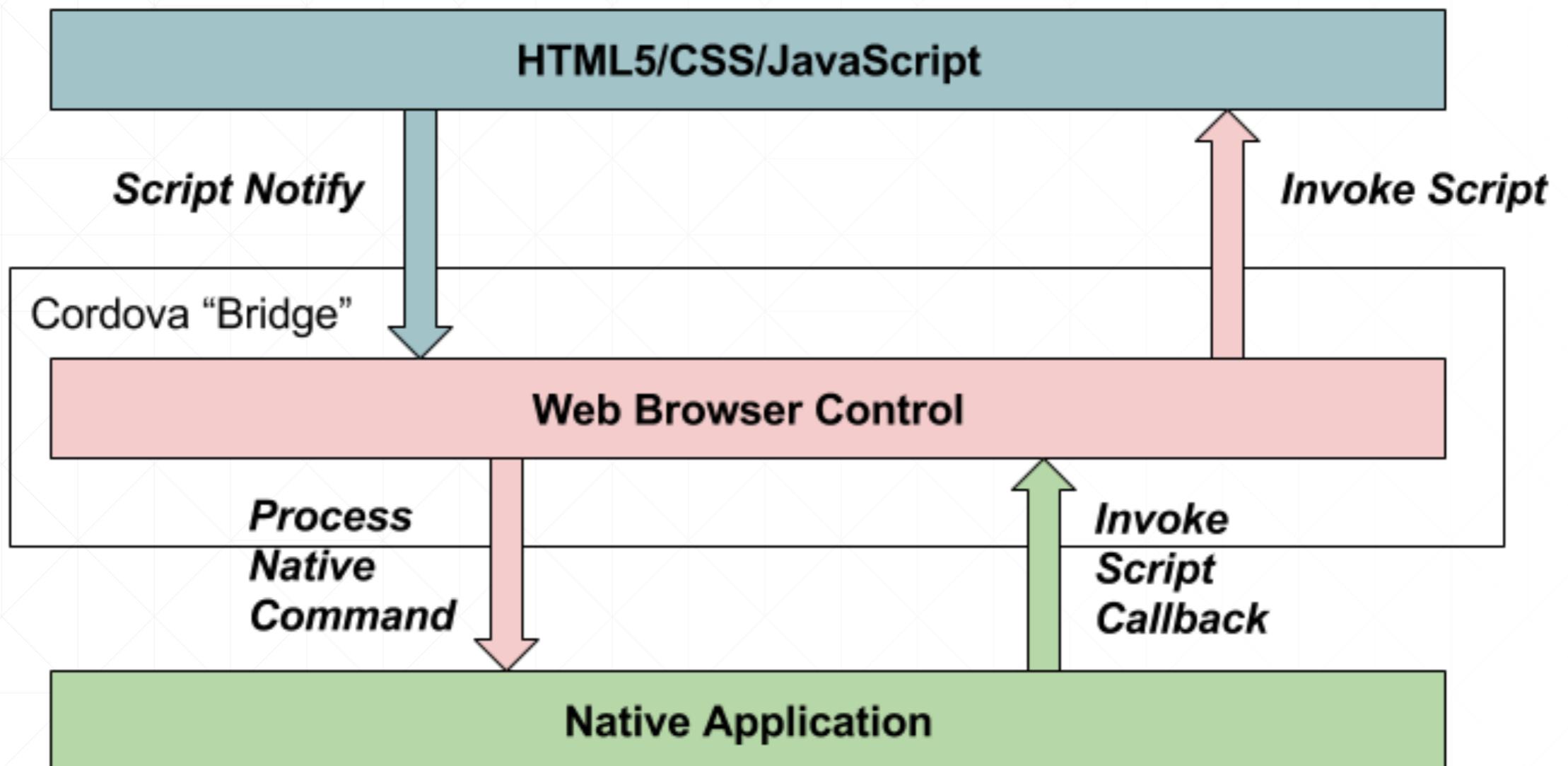
Ionic (Advanced HTML5 Framework)

- Eases the “pain” of hybrid mobile
- Great for Rapid Prototyping (Validating Ideas Quickly)
- Cost effective way to develop Minimum Viable Product (MVP)

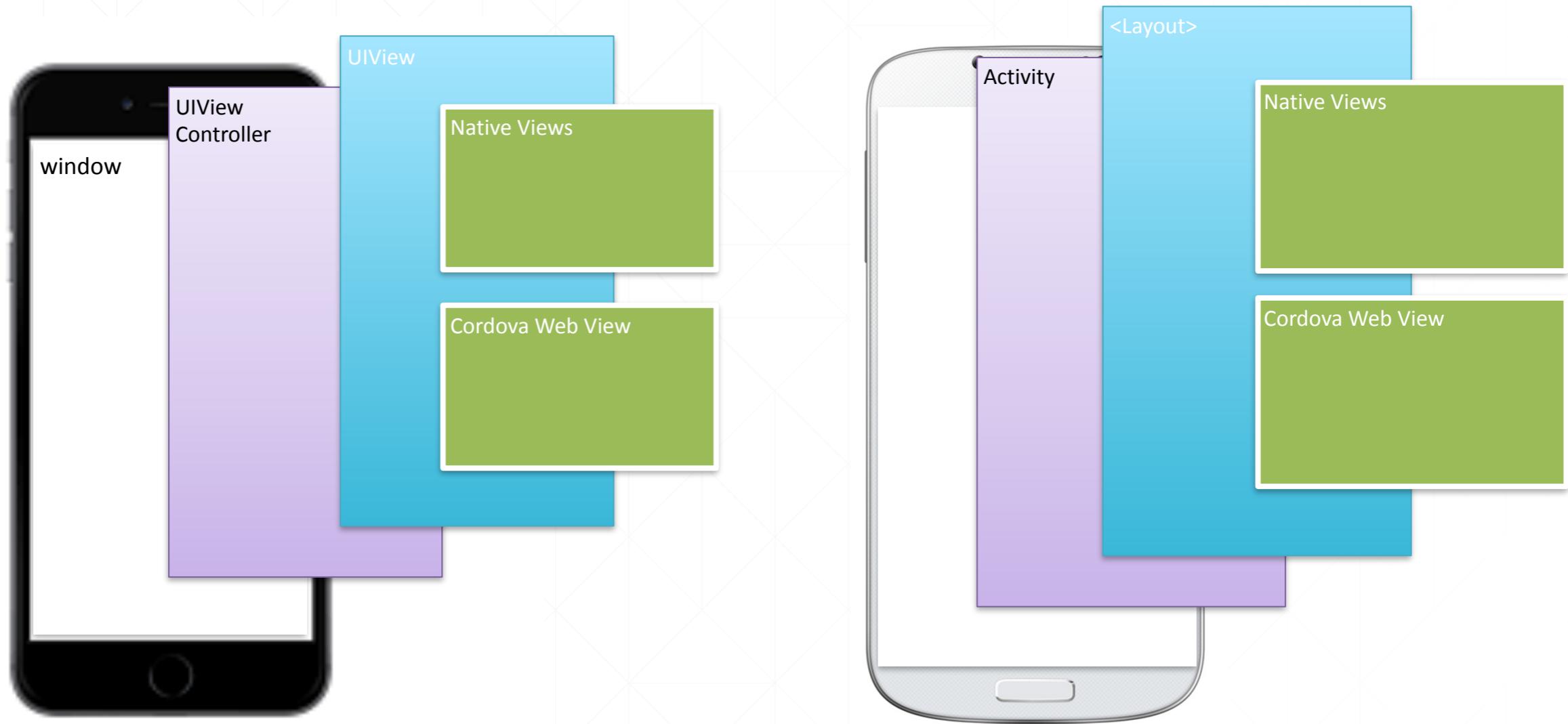
Opportunity for JavaScript code to be reused across web and mobile

***2015 Most Dreaded Technology:** 58% of Developers surveyed dread Cordova (Stack Overflow Survey)

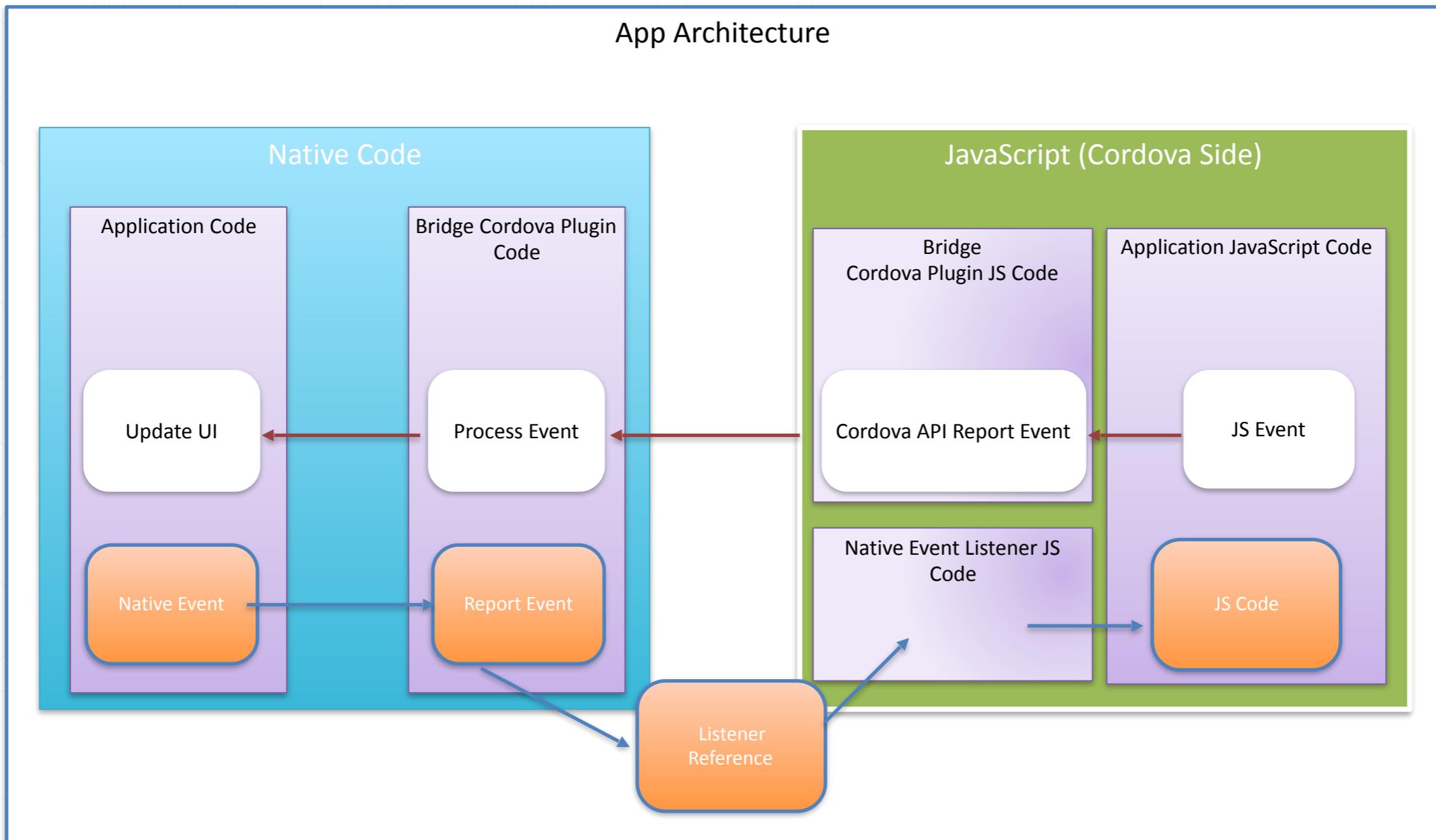
Cordova Architecture



Mixing Native & Cordova



Native & Cordova





NATIVE (ANDROID, IOS)



WWW (HTML/CSS/JS)

Native & Ionic/Cordova Demo

“Hello” Duet

Cordova CLI (web-centric)

```
$ cordova create HelloApp  
$ cordova platform add ios  
$ cordova platform add android  
$ cordova platform add browser  
$ cordova platform remove android
```

JS/CSS/HTML

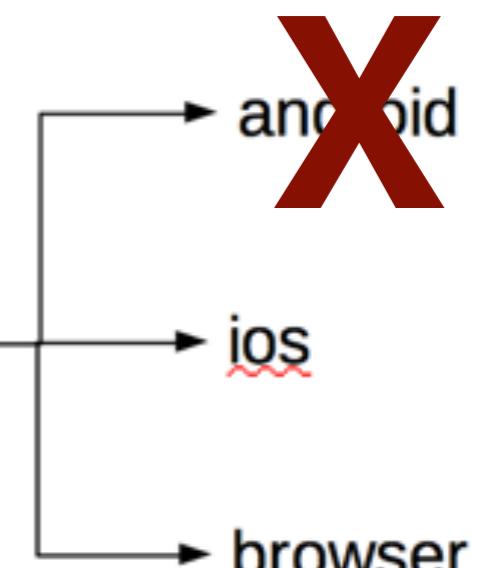


config.xml

www

plugins

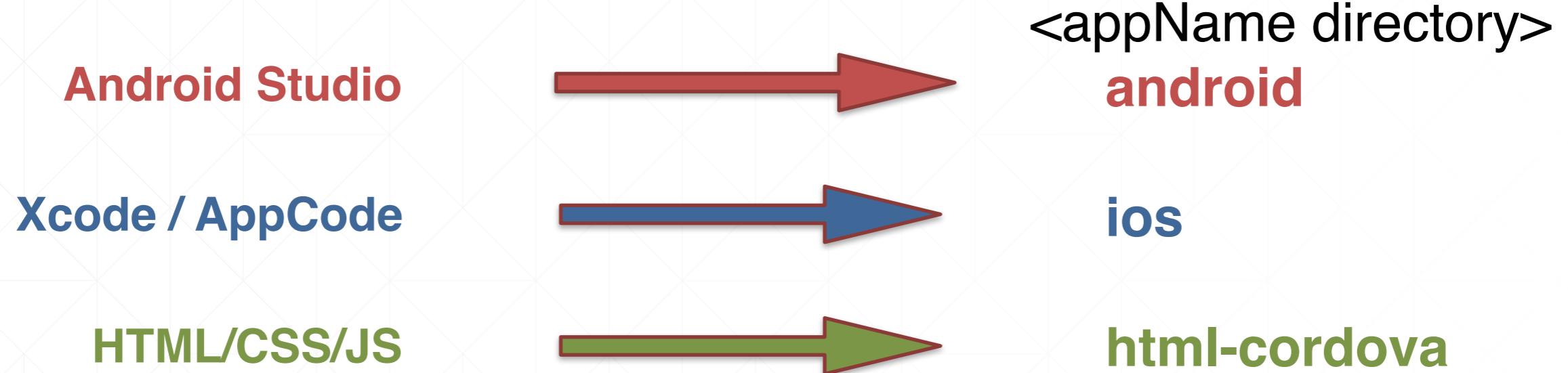
platforms



Embedding Cordova (Native-centric)

- Cordova Command-line composed of tools:
 - **plugman**
 - **cordova-android**
 - **cordova-ios**
- Tools can be used individually to “**break away**” from web-centric development
- What about a custom *native-centric* command-line interface (so native is first class citizen)?

```
$ hybrid-app cordova sample.HelloApp HelloApp
```



iOS

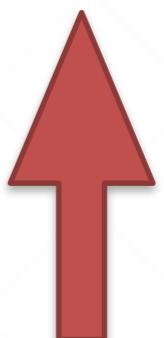
```
@objc(DuetHybridBridge) class DuetHybridBridge: CDVPlugin {  
    func JS_TO_NATIVE_CALL(command: CDVInvokedUrlCommand) {  
        self.commandDelegate!.runInBackground { () -> Void in  
            . . .  
        }  
    }  
  
    func ACTION_BIND_LISTENER(command: CDVInvokedUrlCommand) {  
        self.commandDelegate!.runInBackground { () -> Void in  
            self.listenerCallbackId = command.callbackId  
            . . .  
        }  
    }  
  
    func reportEvent(eventData: NSDictionary) {  
        self.commandDelegate!.runInBackground { () -> Void in  
            let pluginResult: CDVPluginResult =  
                CDVPluginResult(  
                    status: CDVCommandStatus_OK,  
                    messageAsDictionary: eventData as [NS0bject: Any0bject])  
  
            pluginResult.setKeepCallbackAsBool(true)  
            self.commandDelegate!.sendPluginResult(  
                pluginResult,  
                callbackId: self.listenerCallbackId)  
        }  
    }  
}
```

JavaScript

```
this.nativeCall = function (listener) {  
    cordova.exec(null, null,  
        PLUGIN_NAME,  
        "JS_TO_NATIVE_CALL", []);  
};
```

```
this.bindListener = function (listener) {  
    cordova.exec(listener, listener,  
        PLUGIN_NAME, "ACTION_BIND_LISTENER", []);  
};
```

```
//allows JS to respond to events sent by the native OS  
DuetHybridBridge.bindListener(  
    NativeEventsListener.onReceivedEvent);
```





OBJECT PARTNERS

React Native

Bridging Directly to Native APIs (Cross-platform)



Runtime leverages packaged JavaScript VM

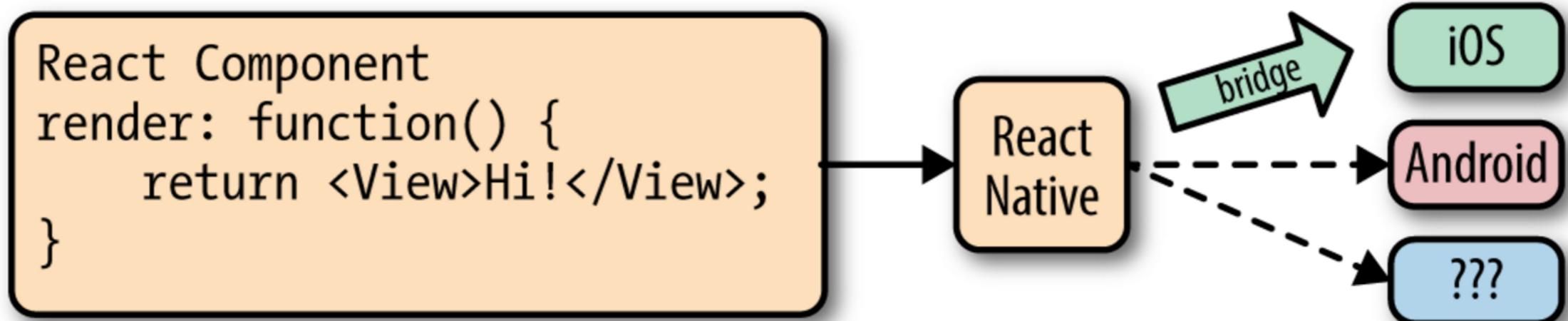
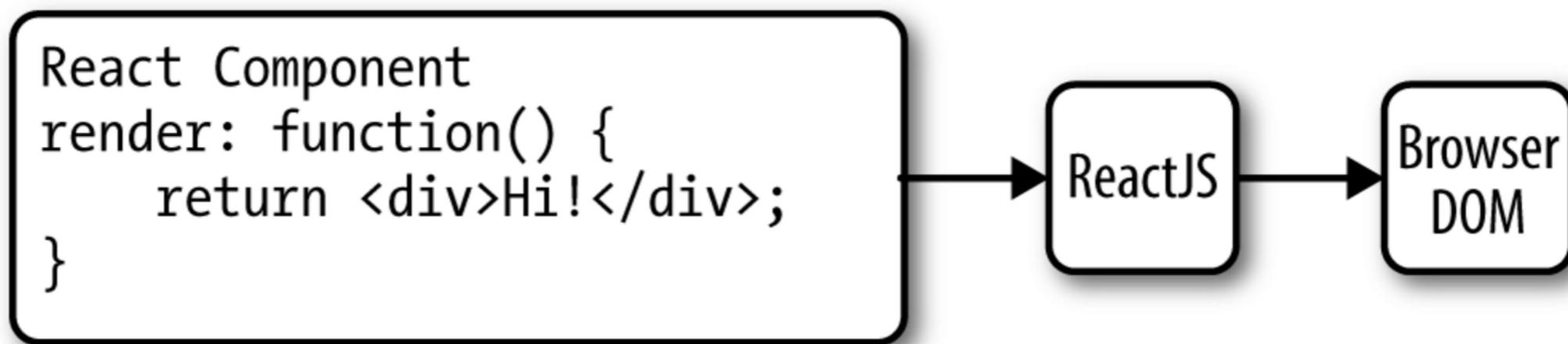
React Components

- JSX allows for markup and JS to “live” in same file
- Separation of concerns NOT technology (css, js, html)

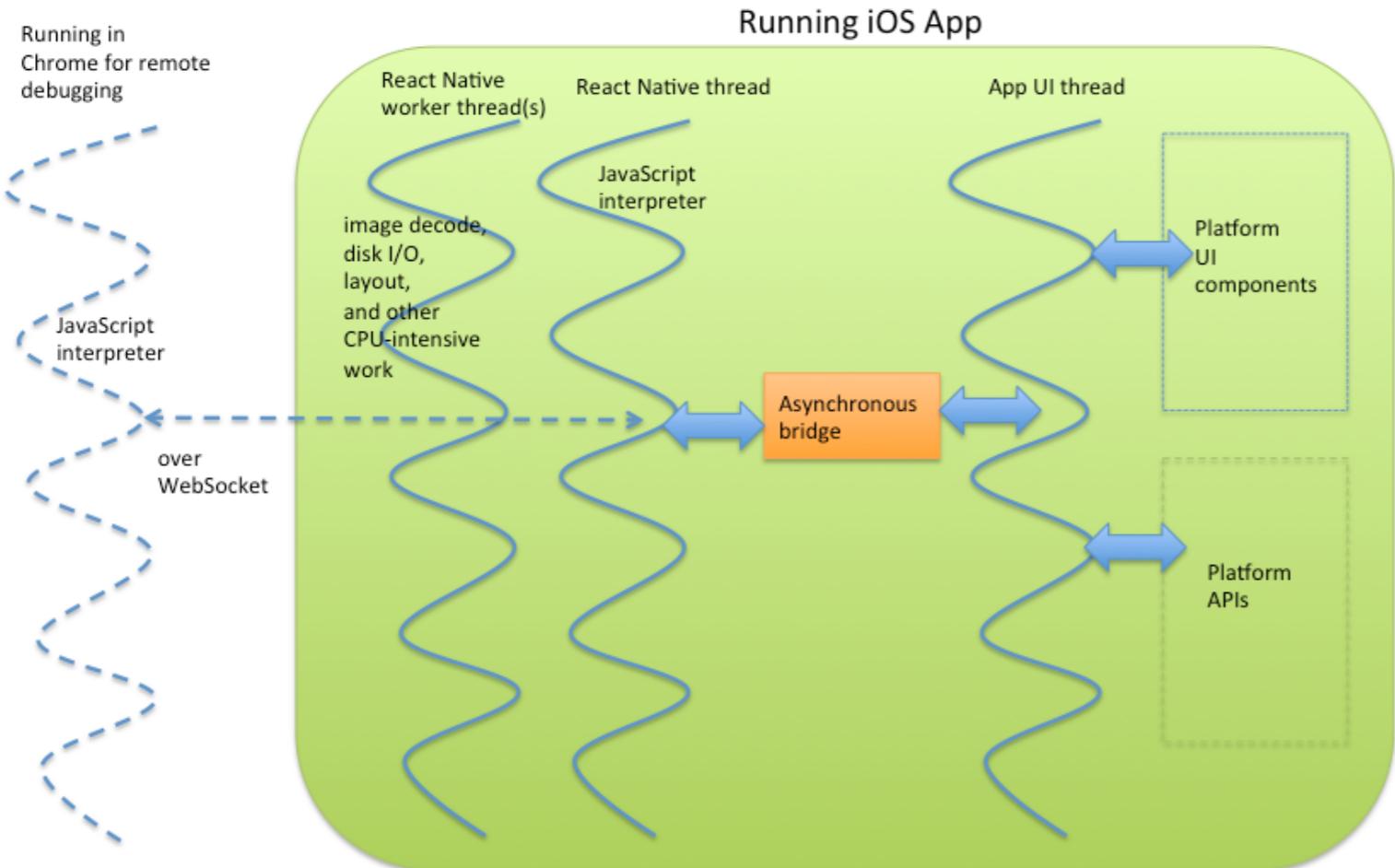
CSS Flexbox for Layout

React Native also has the potential to positively impact your product release cycle. For instance, Apple permits JavaScript-based changes to an app's behavior to be loaded over the air with no additional review cycle necessary.

The “virtual DOM” Abstraction



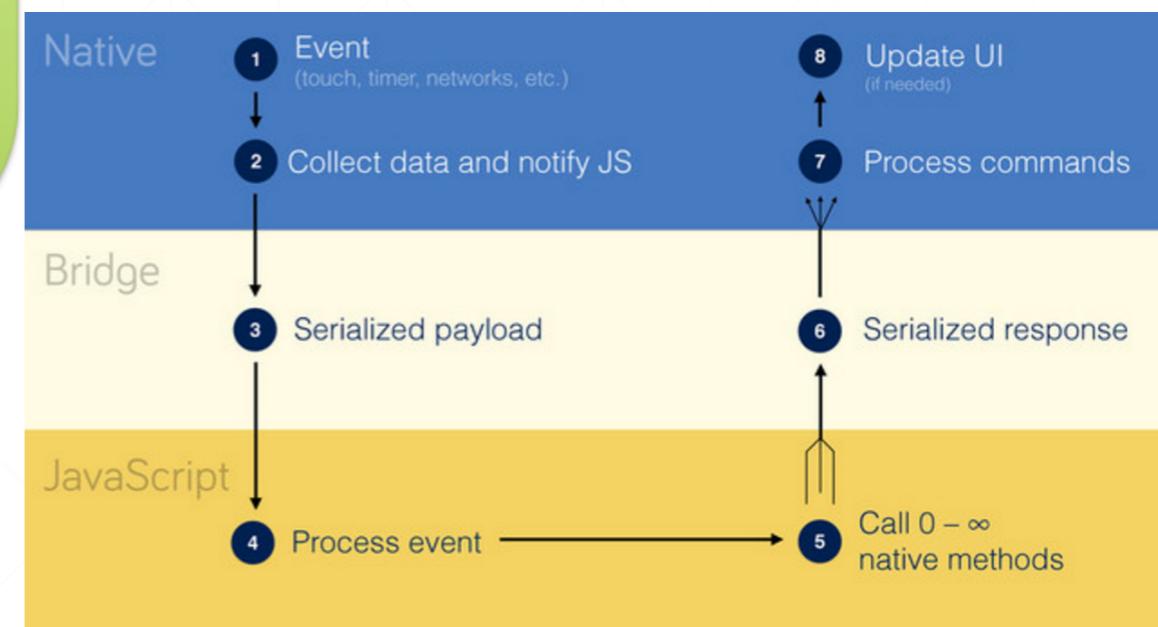
React Native Architecture



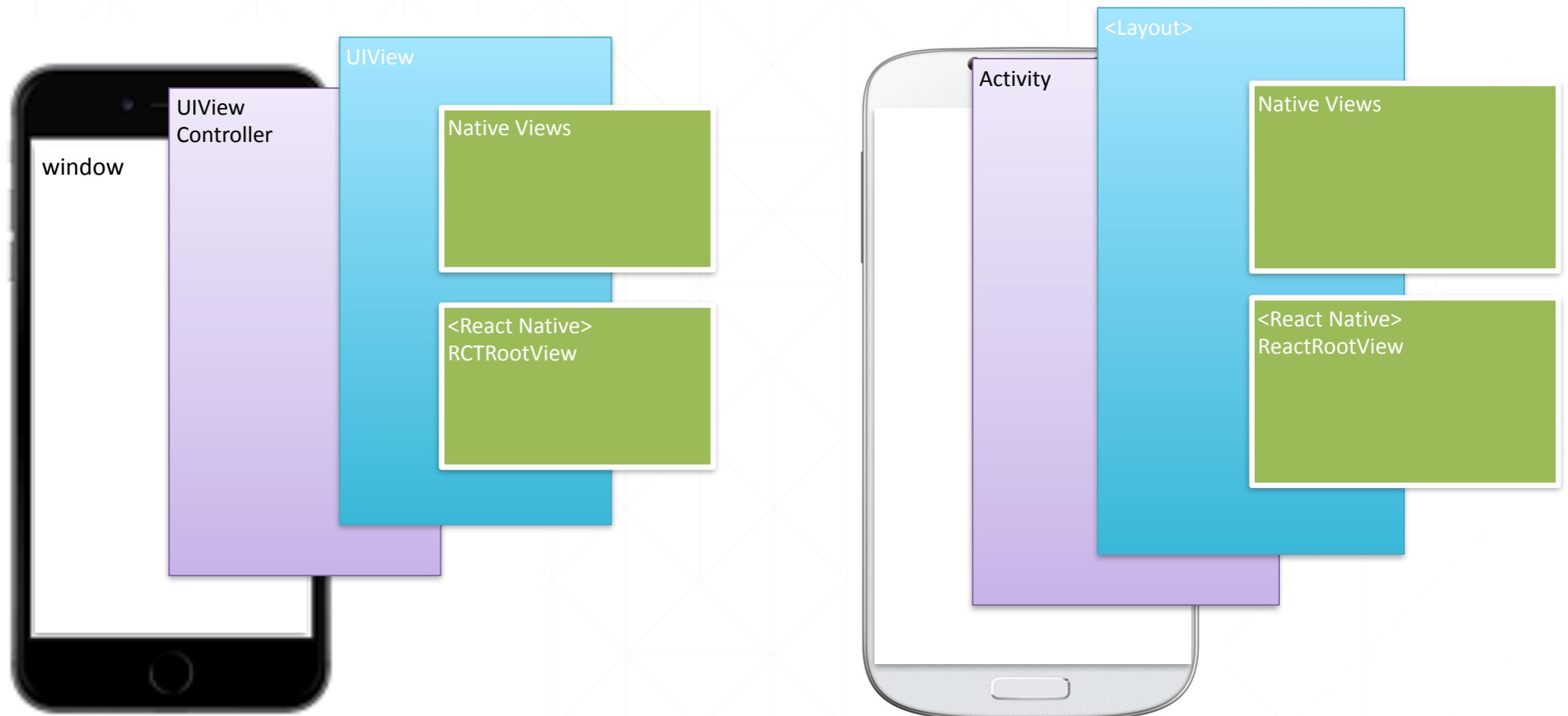
Asynchronous (avoid waiting)

Batched (avoid overhead)

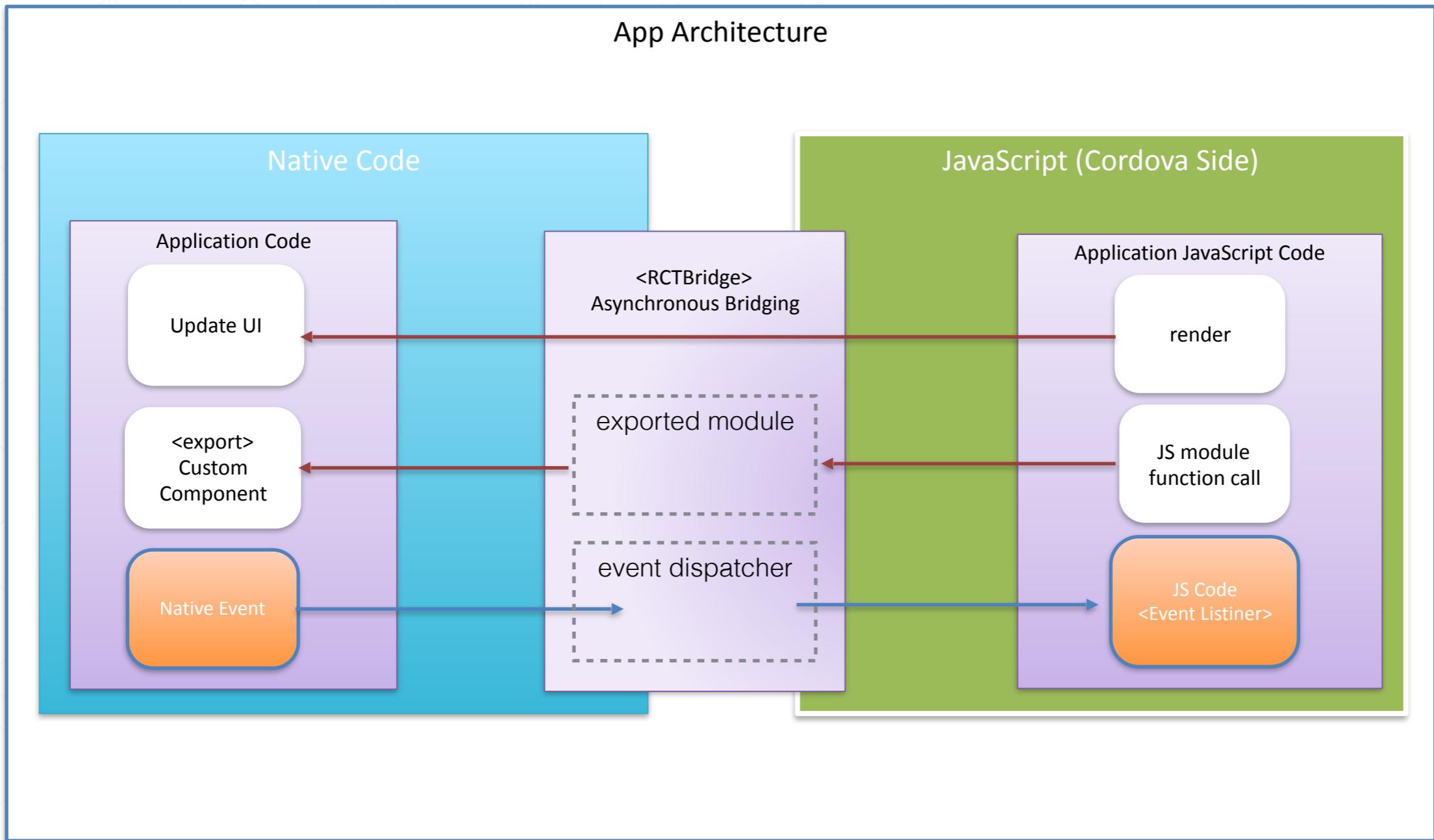
Serializable messages
(avoid shared mutable data
between JavaScript and Native)



Mixing Native & React Native



Native & React





NATIVE (ANDROID, IOS)



WWW (HTML/CSS/JS)

Native & React Native Demo

“Hello” Duet

React Native CLI (Native-centric)

```
$ react-native init HelloApp
```

Android Studio



<appName directory>
android

Xcode / AppCode



ios

HTML/CSS/JS



index.ios.js
index.android.js

iOS

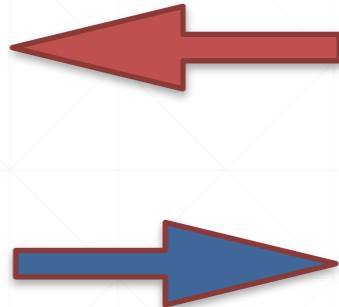
```
@implementation MyModule  
  
@synthesize bridge = _bridge;  
@synthesize methodQueue = _methodQueue;  
  
RCT_EXPORT_MODULE();  
  
RCT_EXPORT_METHOD(jsToNativeCall) {  
    [self someMethod];  
}  
  
- (void) sendEvent {  
    [eventBridge.eventDispatcher  
        sendAppEventWithName:@"EventName"  
        body:@{@"key": value}];  
}  
@end
```

Android

```
public class MyModule extends ReactContextBaseJavaModule {  
  
    @ReactMethod  
    public void jsToNativeCall() {  
        this.someMethod();  
    }  
  
    void sendEvent() {  
        ReactContext reactContext = getReactApplicationContext();  
        WritableMap params = new WritableNativeMap();  
  
        params.putString("key", value);  
        reactContext  
            .getJSMODULE(RCTNativeAppEventEmitter.class)  
            .emit("EventName", params);  
    }  
}
```

JavaScript

```
var MyModule = NativeModules.MyModule;  
  
class HelloApp extends Component {  
    constructor(props) {  
        super(props);  
  
        this.state = {key: 'value'}  
  
        this._onNextLyric = this._onEvent.bind(this);  
  
        this.subscribeToEvent =  
            NativeAppEventEmitter.addListener(  
                'EventName', this._onEvent  
            );  
    }  
  
    _callMyModule() {  
        MyModule.jsToNativeCall();  
    }  
  
    // Event Listeners  
    _onEvent(eventData) {  
        console.log(eventData);  
        this.setState(eventData);  
    }  
}
```





NativeScript

Cross-platform JavaScript
No Built-In Support for Embedding,
Opinionated Framework



Runtime leverages packaged JavaScript VM

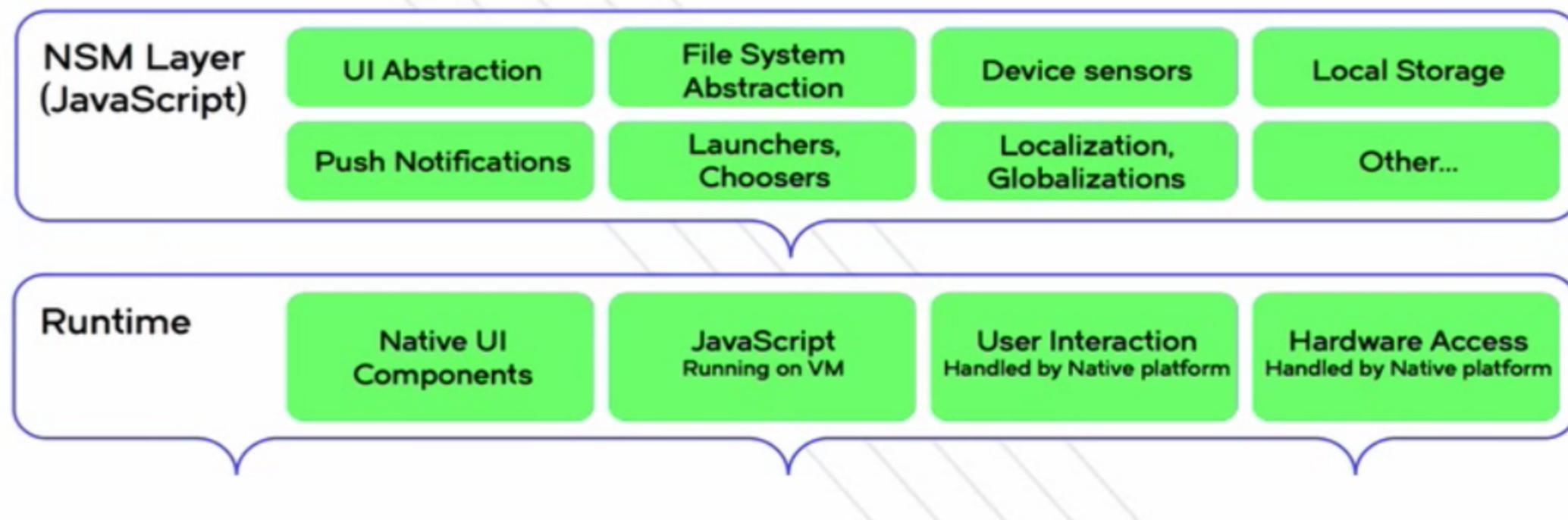
NativeScript Components

- Angular 2 and TypeScript
- Separate HTML, CSS, JavaScript parts

Provides *layout containers* including ***absolute, dock, grid, stack, wrap***

***No Built-in support to update JavaScript/Web changes over the air**

NativeScript Architecture

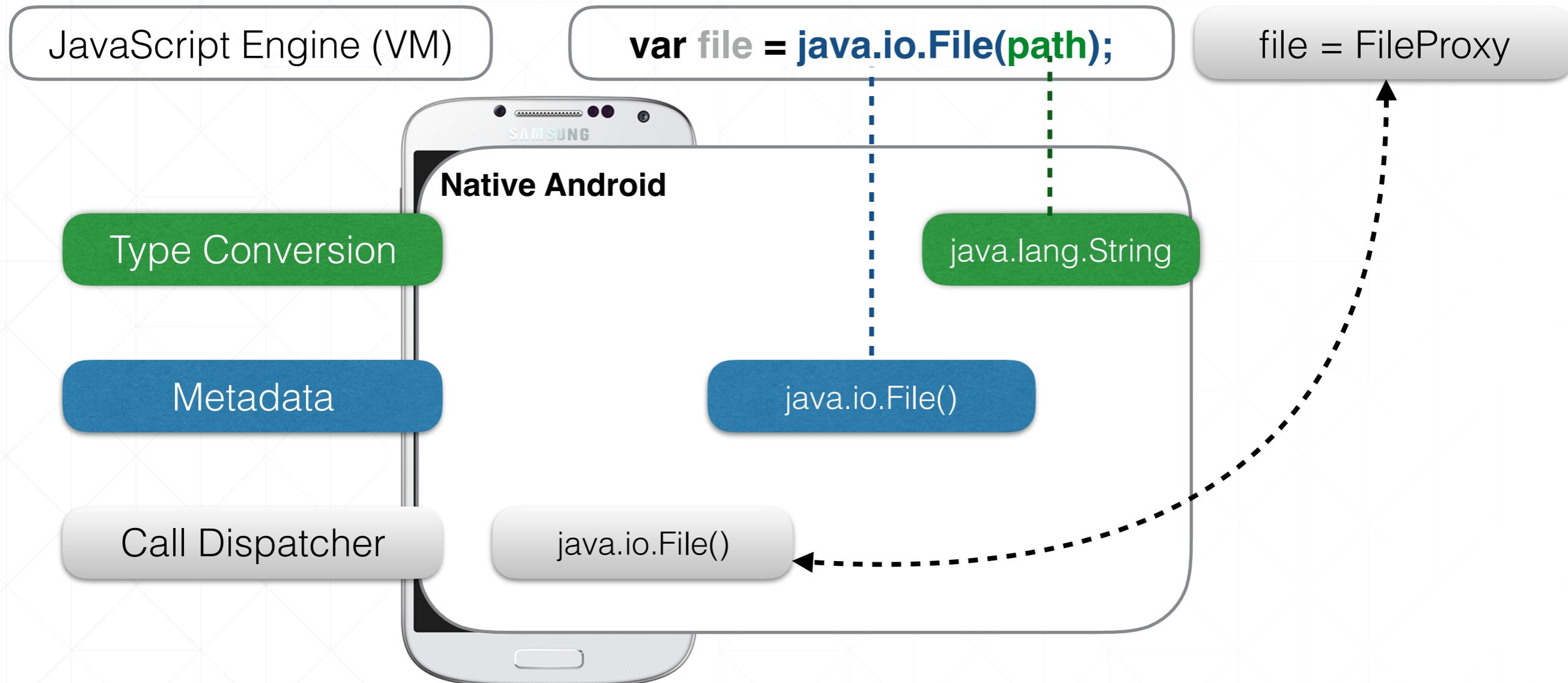


Runtime uses JavaScript VM

Google V8 for Android

WebKit's JavaScriptCore for iOS and Windows

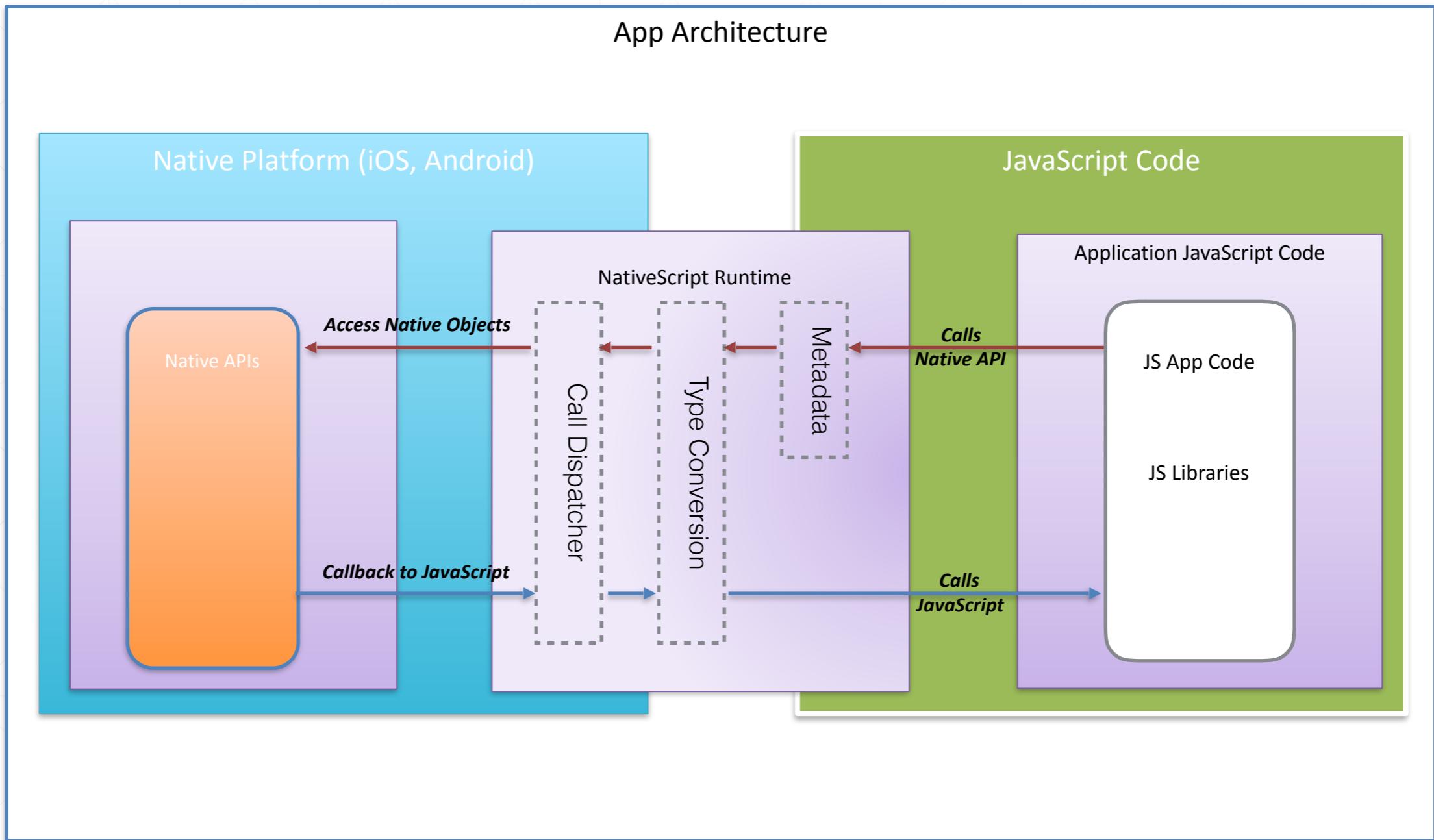
Access any Type and Method on Native Platform



***10% performance overhead**



NativeScript In Action





NativeScript

Using Angular 2 to call Native APIs
(Android, iOS)

NativeScript Demo

“Hello” Duet



@Component JavaScript App Code

```
import {Component} from "@angular/core";
import {View} from "ui/core/view";
import {TextField} from "ui/text-field";
import {LyricMgrService} from "../../shared/lyricsManager/lyric-mgr.service";

@Component({
    selector: "hello-duet-page",
    providers: [LyricMgrService],
    templateUrl: "pages/helloDuet/helloDuet.html",
    styleUrls: ["pages/helloDuet/helloDuet-common.css", "pages/helloDuet/helloDuet.css"]
})

export class HelloDuetPage {
    lyricForLionel: string;
    lyricForAdele: string;

    constructor(private _lyricsMgr: LyricMgrService) {
        this.lyricForLionel = "<Start Singing>";
        this.lyricForAdele = "<Start Singing>";
    }

    playHelloByLionel() {
        this.lyricForAdele = "...Listening";

        this.lyricForLionel = this._lyricsMgr.playLyricForLionel();
    }

    playHelloByAdele() {
        this.lyricForLionel = "...Listening";

        this.lyricForAdele = this._lyricsMgr.playLyricForAdele();
    }
}
```



OBJECT PARTNERS

Discoveries?

objectpartners.com



Things can get tricky mixing UIs (especially if there is a high degree of interaction between them)

Consider

Native for app flow, transitions, animations, complex interaction

HTML/CSS/JS for content rich UI components, services & business logic



React Native is a MUCH friendlier option for mixing Pure Native and Cross-platform JavaScript

NativeScript and **Angular 2** will be interesting technologies to watch in this space

Not sure the added complexity of mixing **Cordova** and Pure Native is worth it. **nice to know it can be done though...*

** React Native compliments Pure Native like peanut butter and chocolate*

** Ionic/Cordova complements Pure Native like peanut butter and jelly*

Where to Look

- These Slides and Source Code (<https://github.com/tlomenda/nativecentric-hybrid>)
- Getting Started with NativeScript and Angular 2 (<http://docs.nativescript.org/angular/tutorial/ng-chapter-0>)
- React Native (<https://facebook.github.io/react-native/docs/getting-started.html#content>)
- Learning Reactive Native (<https://www.safaribooksonline.com/library/view/learning-react-native/9781491929049/>)
- React Native Blog Post (<https://code.facebook.com/posts/1014532261909640/react-native-bringing-modern-web-techniques-to-mobile/>)



Where to Look

- Mixing Cordova with Native (<http://devgirl.org/2014/07/22/mixing-cordova-phonegap-components-with-native/>)
- Hybrid Screens Demo (<https://github.com/AntonAleksandrov/HybridScreensDemoProject>)
- Apache Cordova (<https://cordova.apache.org/>)
- Ionic Framework (<http://ionicframework.com/>)

Contact:

Torey Lomenda

torey.lomenda@objectpartners.com

twitter: @tlomenda

GitHub: <https://github.com/tlomenda>

