



Tutorial MOBILESoft 2016: Opportunities and pitfalls when using cross-platform tools for mobile app development.

Ruben Smeets
ES&S
KU Leuven, Belgium

Michiel Willocx
MSEC iMinds-Distrinet
KU Leuven, Belgium

About us

MSEC – Mobile and Secure

- Research line 1: designing secure mobile applications



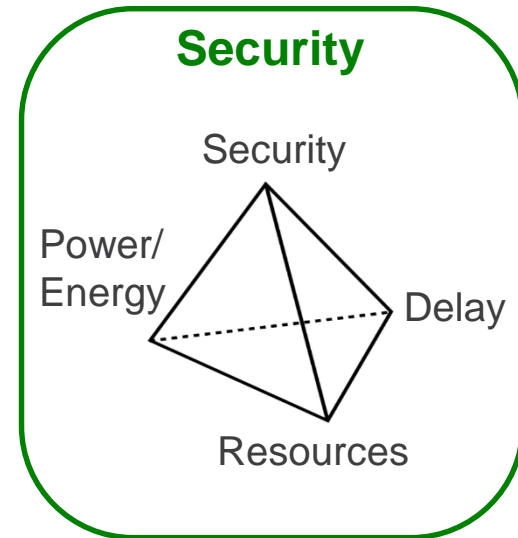
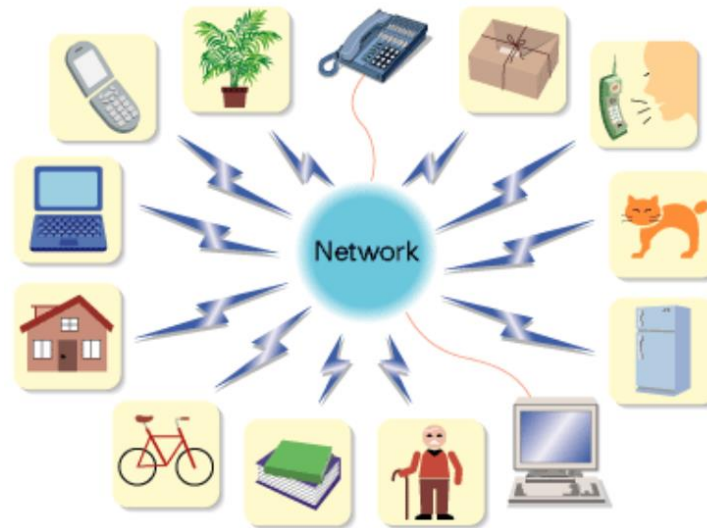
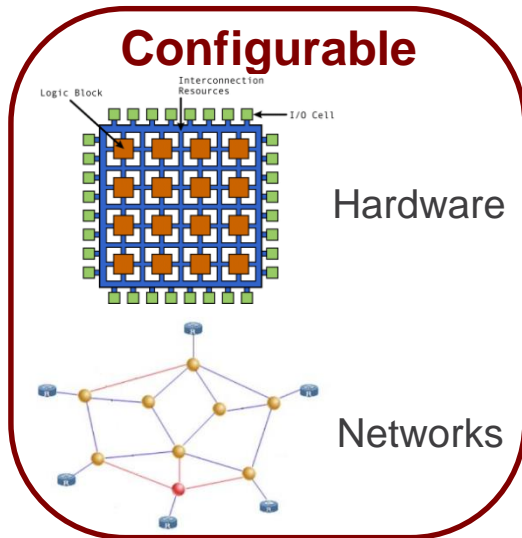
- Research line 2: inspecting system level security & privacy



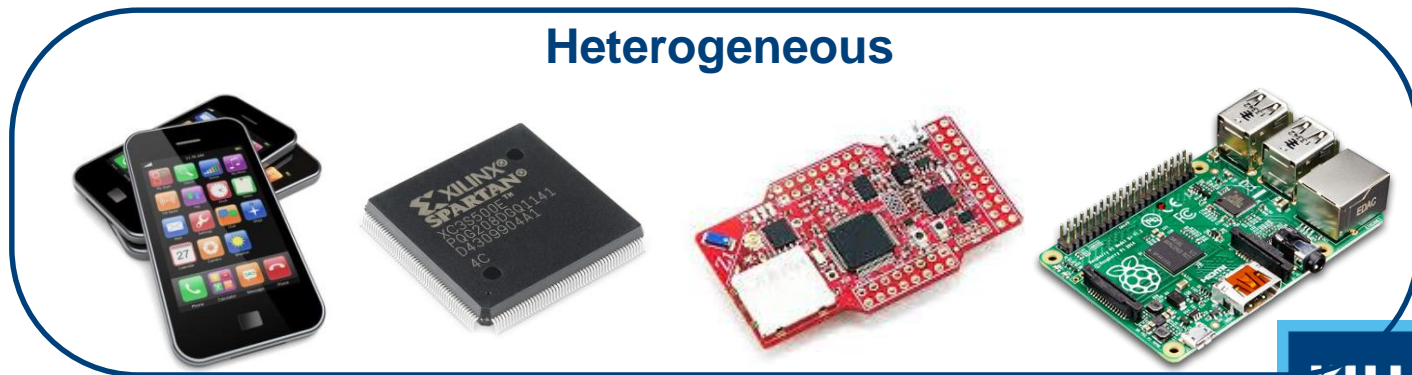
About us

ES&S – Embedded Systems & Security

Research focus: Internet Of Things

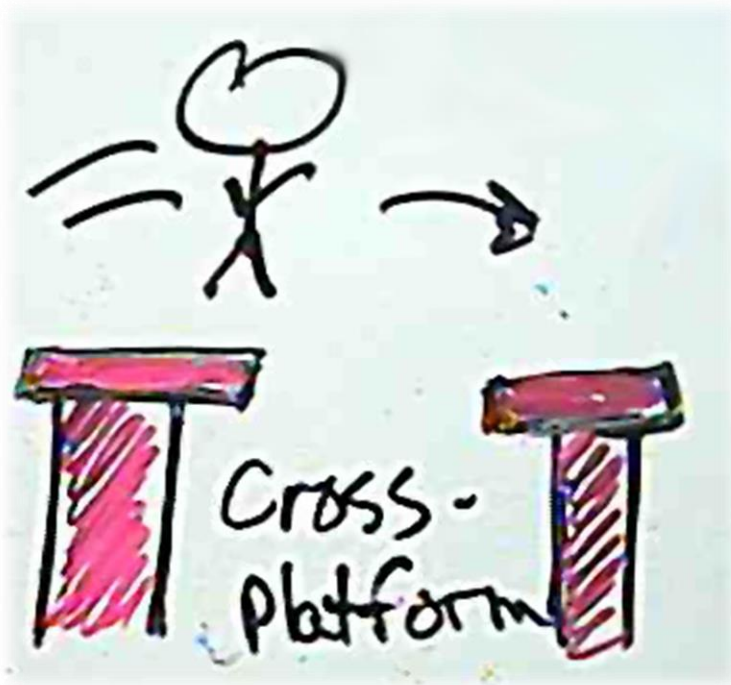


Heterogeneous



CrossMoS

*Cost-efficient development of advanced,
cross-platform mobile applications*



- IWT/VLAIO TETRA project



AGENTSCHAP
INNOVEREN &
ONDERNEMEN

- Project of 2 years
- In collaboration with app developers, small companies and SMEs
- Researchers:
 - Michiel Willocx & Ruben Smeets
- Project Website:
 - <https://www.msec.be/crossmos/>

Table of contents

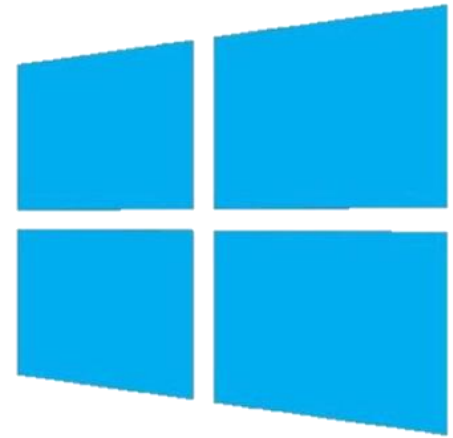
- TUTORIAL TODAY:
 - PART 1:
 - Introduction: What are cross-platform tools and why should I use them?
 - Classification of cross-platform tools
 - Cross-platform tool selection criteria
 - PART 2: The native JavaScript framework landscape (Ruben)
 - Why Native JavaScript?
 - Possible candidates
 - Comparison of three popular tools
 - PART 3: Discussion
- TOMORROW: Tutorial Session
 - Web-based Hybrid Mobile Apps: State of the Practice and Research Opportunities (Ivano Malavolta)

Introduction



Introduction

- Mobile platforms



Introductie

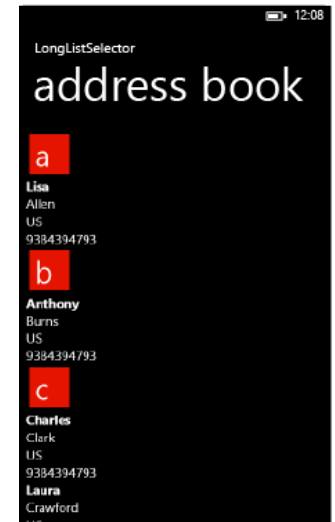
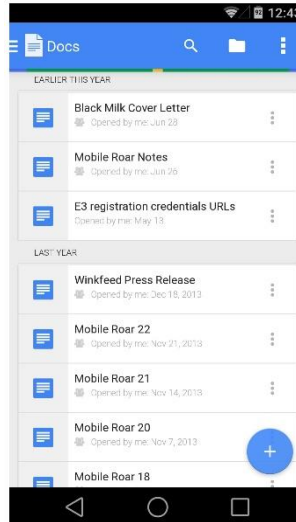
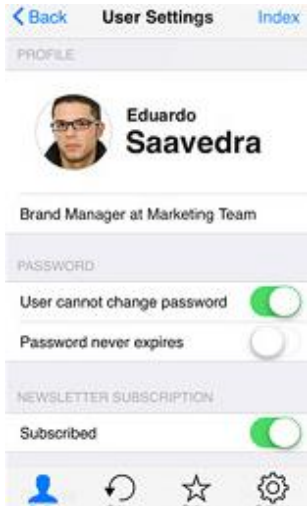
- Mobile platform sales

Worldwide Smartphone Sales to End Users by Operating System in 4Q15 (Thousands of Units)

Operating System	4Q15 Units	4Q15 Market Share (%)	4Q14 Units	4Q14 Market Share (%)
Android	325,394.4	80.7	279,057.5	76.0
iOS	71,525.9	17.7	74,831.7	20.4
Windows	4,395.0	1.1	10,424.5	2.8
Blackberry	906.9	0.2	1,733.9	0.5
Others	887.3	0.2	1,286.9	0.4
Total	403,109.4	100.0	367,334.4	100.0

Source: Gartner (February 2016)

Native Development



Native development: Android



Native development: Android

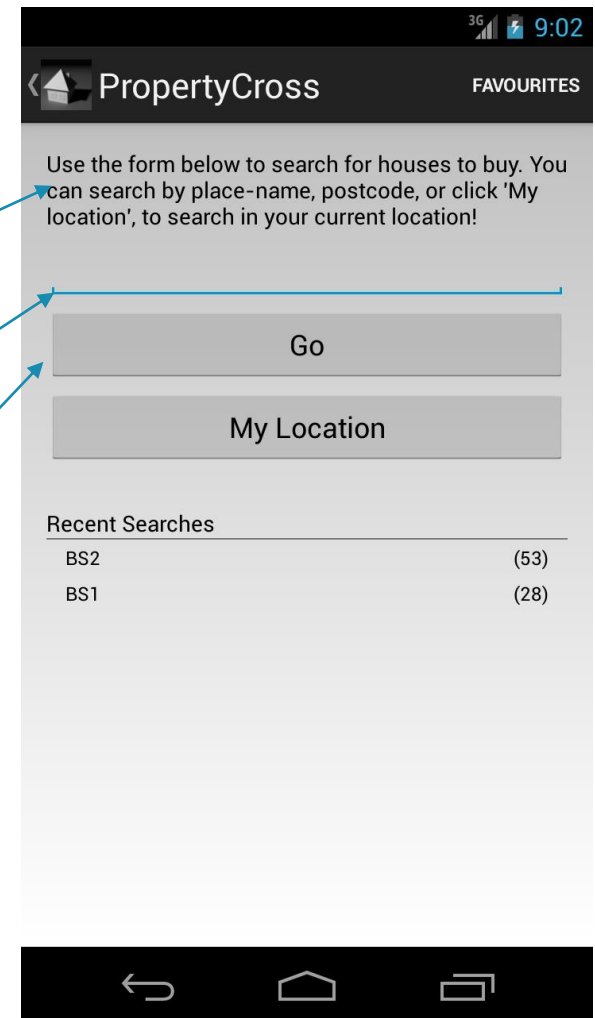
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/propview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_gravity="top"
    android:layout_margin="16dp"
    android:orientation="vertical" >

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="4dp"
        android:text="@string/search_description" />

    <EditText
        android:id="@+id/search"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="4dp"
        android:inputType="text"
        android:imeOptions="actionSearch" />

    <Button
        android:id="@+id/do_search"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="4dp"
        android:text="@string/do_search_text" />

</LinearLayout>
```

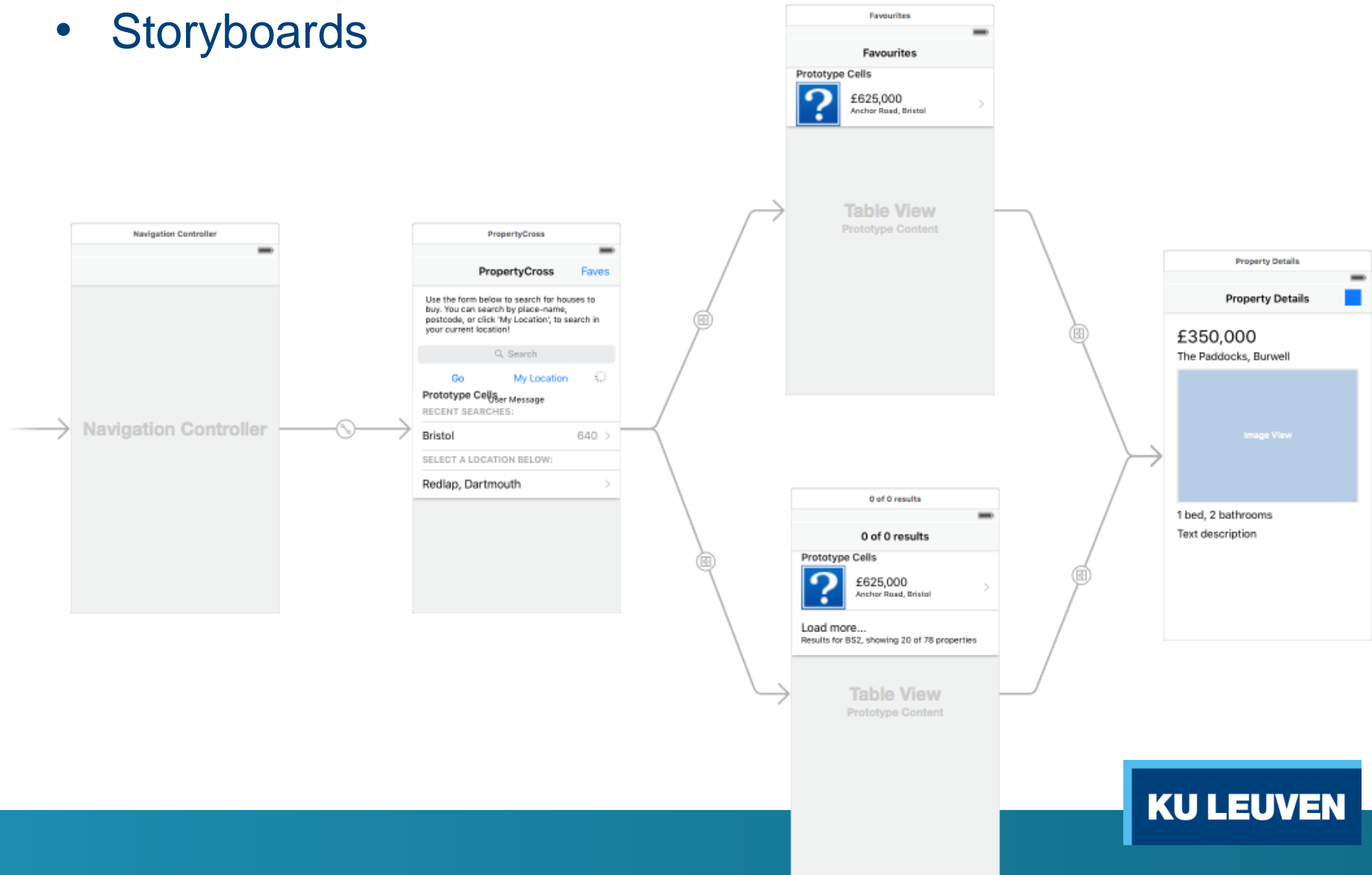


Native development: iOS

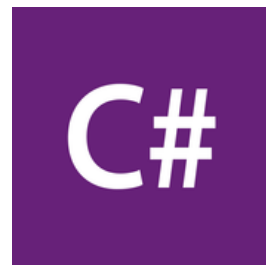


Native development: iOS

- Storyboards



Native development: Windows Phone



Native development: Windows Phone

```
64 }
65
66 {
67     <TextBlock Text="PropertyCross"
68         FontSize="{StaticResource PhoneFontSizeExtraLarge}"/>
69
70     <TextBlock Text="Use the form below to search for houses to buy. You can search by place-name, p
71         in your current location!"
72         Grid.Row="1"
73         TextWrapping="Wrap"
74         Margin="0,30,0,0"/>
75
76     <!-- search form -->
77     <StackPanel Orientation="Horizontal"
78         Grid.Row="2"
79         Margin="0,10,0,0">
80         <TextBox Text="{Binding SearchText, Mode=TwoWay}"
81             KeyDown="TextBox_KeyDown"
82             Width="200"
83             IsEnabled="{Binding IsLoading, Converter={StaticResource BooleanNotConverter}}"/>
84         <Button Content="Go"
85             IsEnabled="{Binding IsLoading, Converter={StaticResource BooleanNotConverter}}"
86             Command="{Binding SearchCommand}"/>
87         <Button IsEnabled="{Binding IsLoading, Converter={StaticResource BooleanNotConverter}}"
88             Command="{Binding SearchMyLocationCommand}"
89             Content="My location"/>
90     </StackPanel>
91
92     <!-- loading indicator -->
93     <Grid Height="8" Grid.Row="3">
94         <ProgressBar IsIndeterminate="True"
95             Visibility="{Binding IsLoading, Converter={StaticResource BoolToVisibilityConve
```

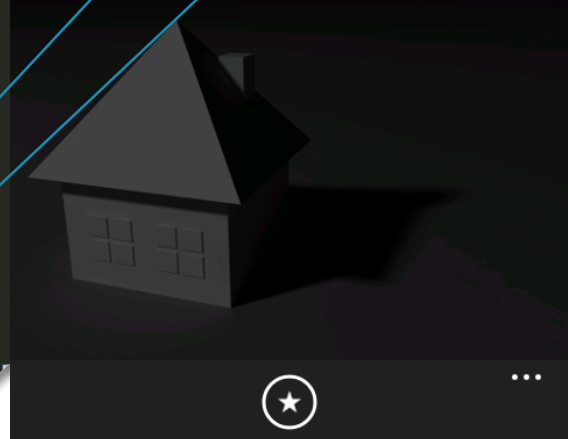
PropertyCross

Use the form below to search for houses to buy. You can search by place-name, postcode, or click 'My location', to search in your current location!

Bristol Go My Location

Recent Searches

Bristol	314
BS1	33



Overview native development

			
Programming Language	Objective – C Swift	Java	.NET
User Interface	Story boards	XML-files	.XAML-files
IDE	Xcode	Android Studio	Visual Studio

Problems native development

			
Programming Language	Objective – C Swift	Java	.NET
User Interface	Story boards	XML-files	.XAML-files
IDE	Xcode	Android Studio	Visual Studio

+different development techniques
+different application lifecycles



Problems native development

Development time



Time to release updates and to fix bugs



Development cost



Necessary programming skills



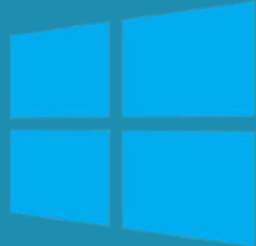


Solution?

Cross-Platform Tools



Cross-Platform Tools (CPTs)

			
Programming Language	Objective – C Swift	Java	.NET
User Interface	Story boards	XML-files	.XAML-files
IDE	Xcode	Android Studio	Visual Studio

➔ support all platforms with one (partially) shared code base

Examples



Xamarin

snōwkit



OpenFL



appcelerator®

Corona SDK

PhoneGap



AppGyver®



Factr



ENYO



ADOBE AIR™



ionic



unity



Sencha



ICENIUM™



jQuery
mobile

KU LEUVEN

Classification of Cross-Platform-Tools



Classification of CPT's

- Web Apps (JavaScript Frameworks)
- Web-to-native Wrappers
- Runtimes
- Source code translators
- App Factories

Classification of CPT's

- Web Apps (JavaScript Frameworks)
- Web-to-native Wrappers
- Runtimes
- Source code translators
- **App Factories**

App Factories



- Drag and drop app design
- Automatic code generation
- Little to no self-written code
- Used for writing simple applications (e.g. RSS feed reader)

App Factories



- No programming skills required
- Often ability to develop in cloud



- Limited UI capabilities
- Limited overall possibilities

App Factories: Examples



The screenshot displays the appmakr web interface for creating an app. It is divided into three main sections: a left sidebar, a central preview area, and a right sidebar.

Left Sidebar:

- Step 1 Step 2 Step 3** (Progress indicators)
- Available in-app functions**
 - Common Functions** (with a blue arrow): Websites, MyBlog, News, Photos, Videos, Contacts, Directions, Calendar, HTML Page, Forms, Docs, LiveChat.
 - Social Feeds**
 - News & Blogs**
 - Photos & Videos**
- Appearance**
 - Backgrounds**: A blue folder icon.
 - Header**: The letters 'ABC' in a stylized font.
 - Icons**: Social media icons for Facebook, Twitter, RSS, and a plus sign for more.

Central Preview Area:

- My Very First App** (Title)
- MyBlog** (Icon)
- infinite monkeys** (Text)
- Live Preview** (Button)
- App Saved 32 secs ago** (Text)

Right Sidebar:

- BACK** **NEXT** **HELP** **EXIT** (Navigation buttons)
- Video Help** (Section header) **OFF ON** (Toggle switch)
 - MyBlog** (Section header)
 - Include RSS feed from your favourite blog, or just about any RSS feed. You will need to find the URL (address) for the feed itself, not just the homepage. Look for this symbol or the word 'RSS' or 'feed' on the source site.
 - [Click here to watch the Help Video](#)
- Specs / Info** (Section header)
 - icon title:** **MyBlog**
 - Icon** (Image of a speech bubble)
 - Background** (Image of a monkey)
 - Feed Uri:** **RSS feed link**
 - SAVE** (Button)

App Factories: Examples



See All Fields

Edit Field
Copy Field
Paste Field
Delete Field
Label
Text Field
Text Area
Image
Button
Image Button
Switch
CheckBox
Table
Picker
Web View
GPS
Slider
Camera
Audio
Alert
Hidden Field
Bring To Front
Send To Back

Amount

Item

Amount (\$)

Tip %

Tax %

Total (\$)

Next

Download App Design as PDF File

Native App Design

Current App TapNCharge

App Properties Select ->

Select This App To Test On Device

Validate Field References

App Page amount_page

Save Undo Edits After Last Save

Go To Web App Design

amount_page

- amount_title
- description
- item_label
- amount
- amount_label
- get_tip
- tip
- percent_tip
- tip_label
- get_percent_tax
- tax
- tax_label
- percent_tax
- sum
- total

Classification of CPT's

- **Web Apps (JavaScript Frameworks)**
- Web-to-native Wrappers
- Runtimes
- Source code translators
- App Factories

Web Apps



→ Mobile Websites

- Accessed in standard mobile browser (Chrome, Safari, ...)
- Optimized for mobile device screen sizes

Web Apps: JavaScript Frameworks

JS

UI Components

- Layout
 - Optimization, scaling and formatting for mobile screen sizes
 - Native-looking skins (not always available)
- Optimization for touch functionality



Web Apps: JavaScript Frameworks

JS

Other Components

Assistance in:

- DOM manipulation
- Utility functions (e.g. array manipulations, access Web resource)
- Event handling (e.g. on click, gestures)



Web Apps: JavaScript Frameworks

JS



Use of design patterns

- MVC (e.g. AngularJS)
- MVVM (e.g. KnockoutJS)
- ...

Web Apps



- No platform-specific code
- Easy to develop
- Easy to update
- Easy to distribute (URL)
- Lots of support and different frameworks available



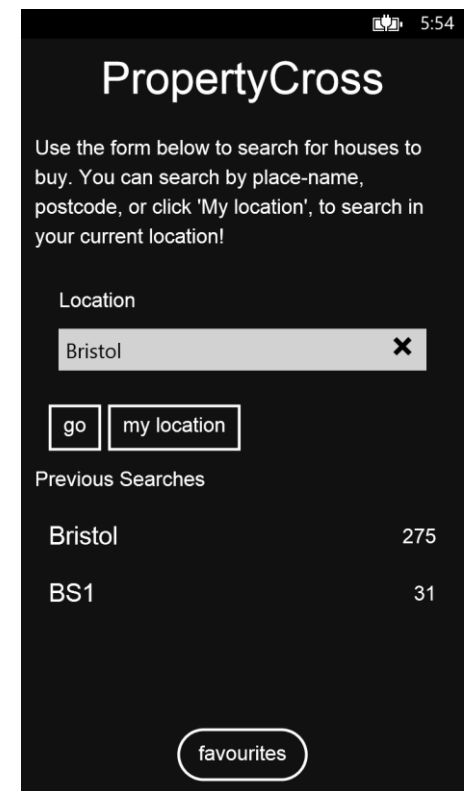
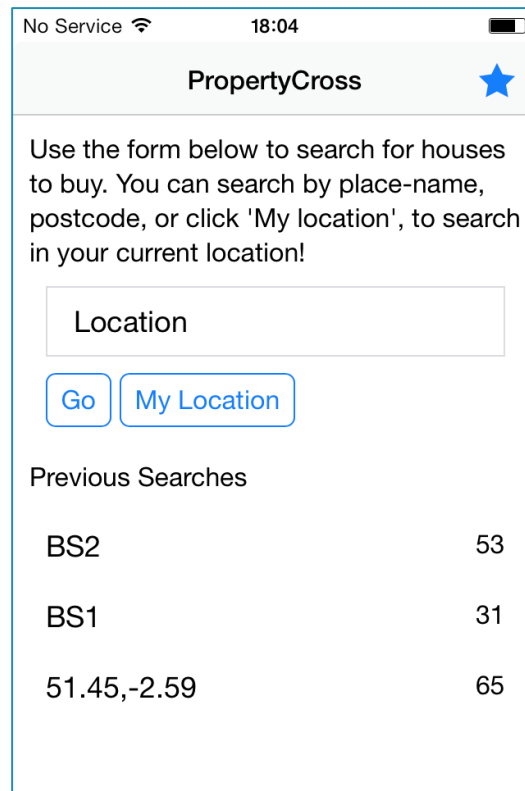
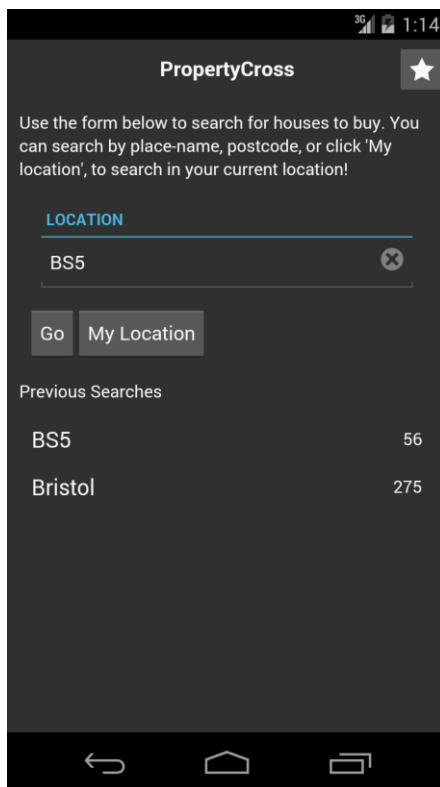
- Internet access always needed
- Responsiveness (partly) depends on Internet connection
- Not a real, stand-alone application
- Limited access to device features
- Often no native look and feel
- Depends on browser capabilities

JavaScript Frameworks: Examples



Web Apps: Sencha Touch

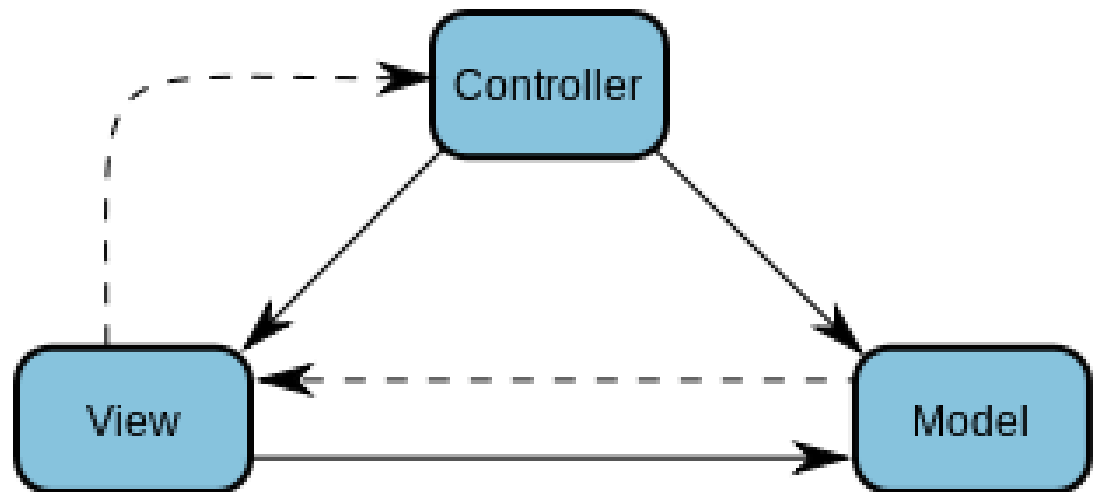
- Ability to use native skins



Web Apps: Ionic

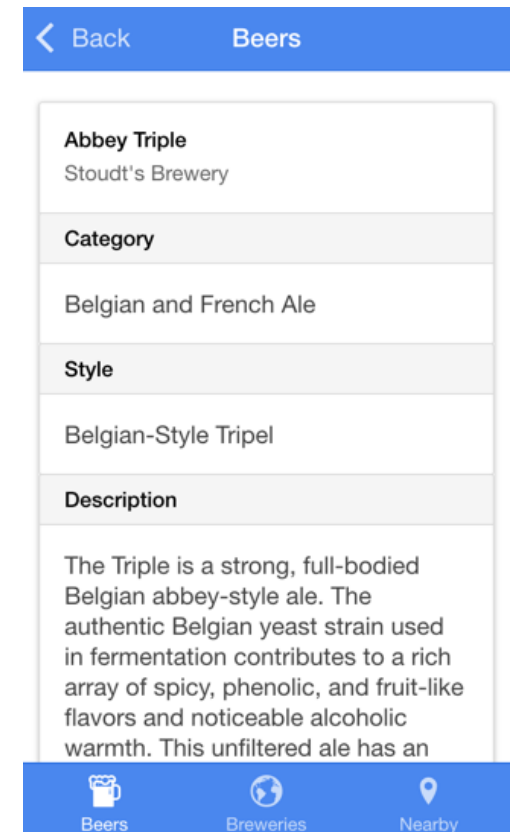
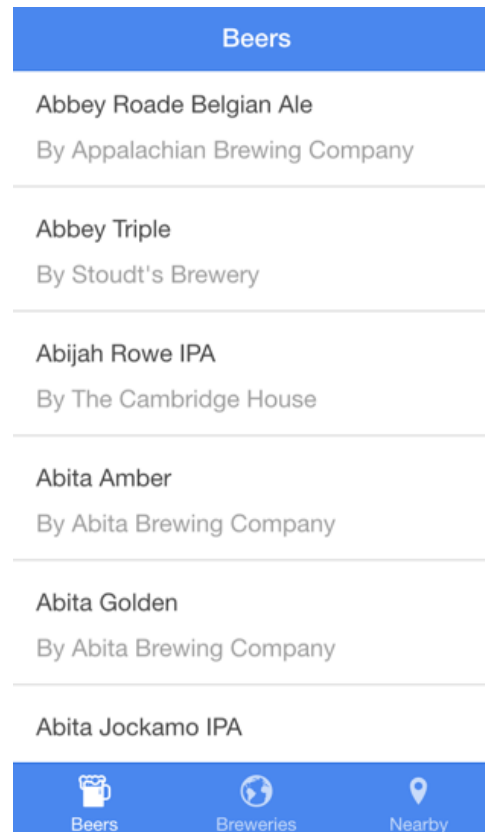
- Based on  **ANGULARJS**
by Google

→ MVC Design Pattern



Web Apps: Ionic

- Focusses on look & feel and UI interaction



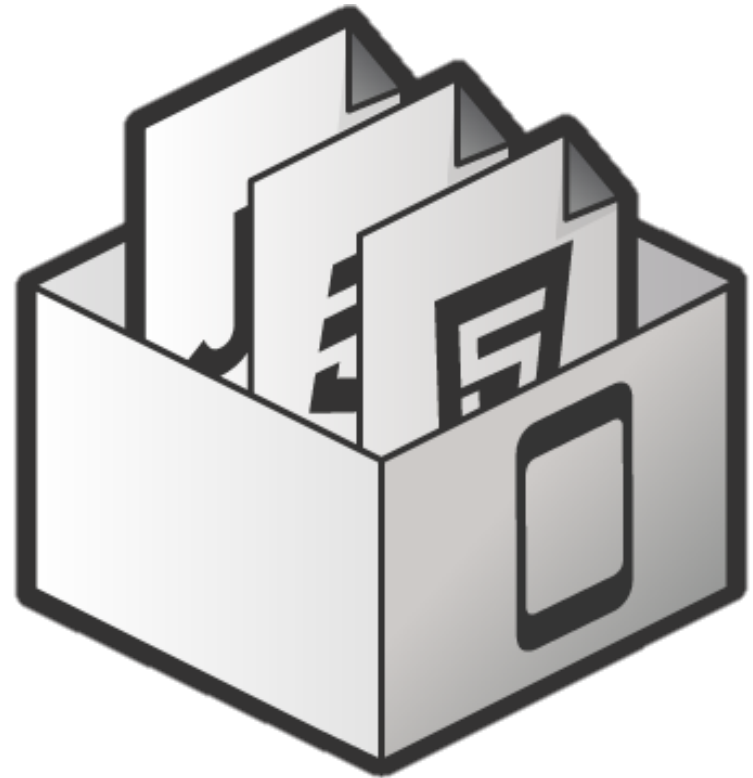
(Recently also native skins)

Classification of CPT's

- Web Apps (JavaScript Frameworks)
- **Web-to-native Wrappers**
- Runtimes
- Source code translators
- App Factories

Web-To-Native Wrappers

- Web Apps, packaged as a native, stand-alone application
- Web code is displayed in a chromeless webview
- Wider range of native API calls compared to normal Web browser



Web-To-Native Wrappers



- Allow Web developers to make mobile applications
- Convert existing Web services to mobile applications
- Stand-alone application
- More available device features than Web apps



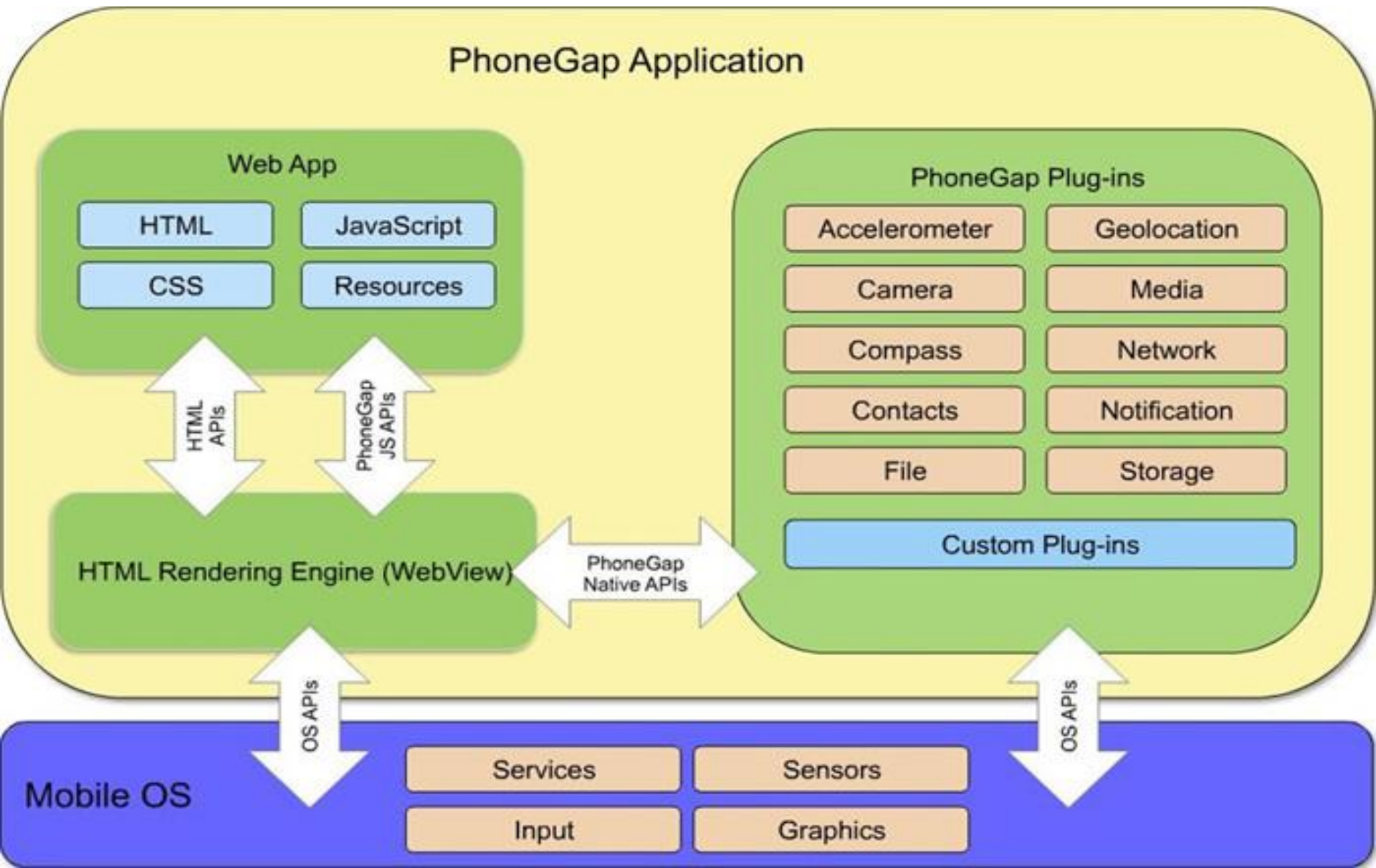
- Poorer UX compared to native
- Often no native look and feel
- Performance overhead

Web-To-Native Wrapper: Phonegap

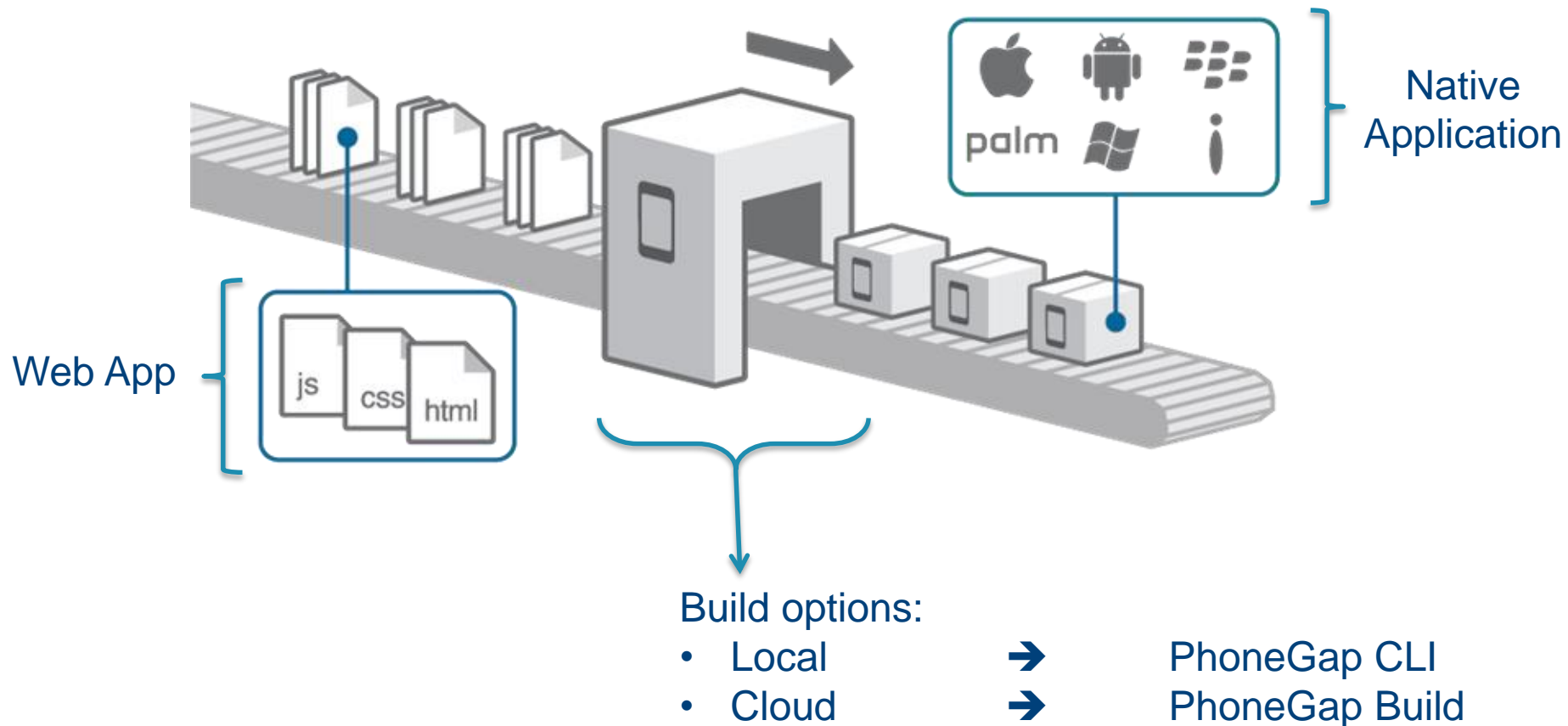


- Acquired by Adobe in 2011
- Supported OS:
 - Android
 - iOS
 - Windows Phone
 - BlackBerry
 - ...
- Alternatives for PhoneGap as web-to-native wrapper?
 - ➔ discontinued, never used, bankrupt,...

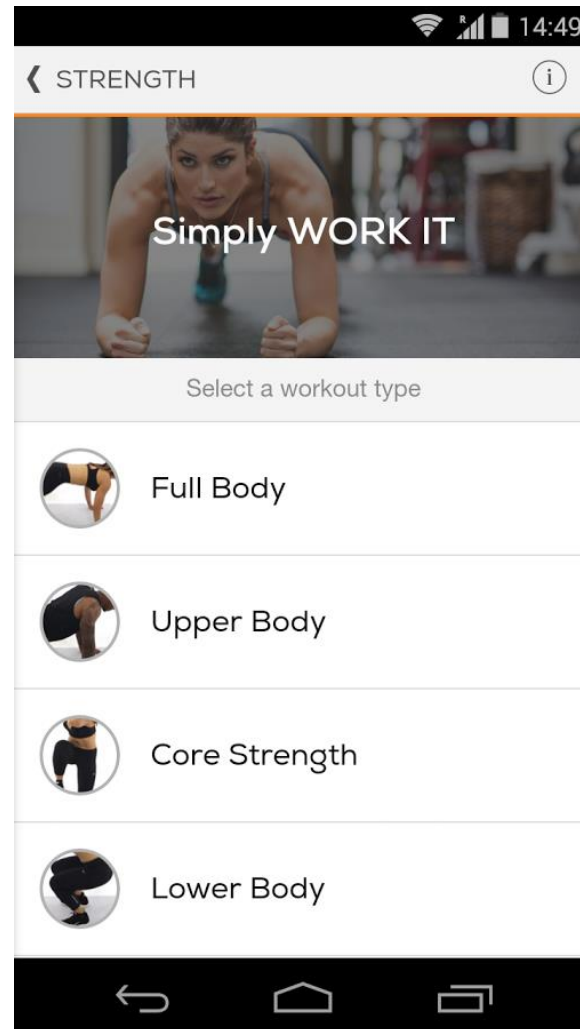
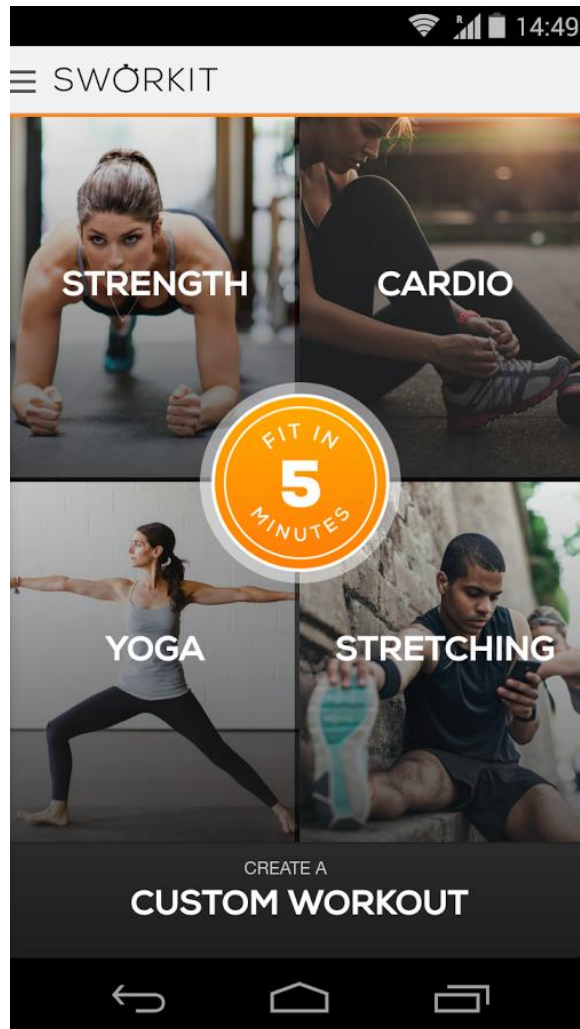
PhoneGap Explained



PhoneGap Explained: Packaging



Examples of PhoneGap applications



Examples of PhoneGap applications



Wikipedia

by Wikimedia Foundation | added 04 Feb 2012

Official Wikipedia App for Android, iOS and Playbook. Wikipedia is the free encyclopedia containing more than 20 million articles in 280 languages, and is the most comprehensive and widely used reference work humans have ever compiled.

Features:

- Save article to read later or offline
- Search articles nearby
- Share articles using Android "Share" function
- Read article in a different language
- Full screen search

"Wikipedia has to be everywhere, and Adobe PhoneGap helps us get it there," says Tomasz Finc, director of mobile and special projects at Wikimedia. "Within its first three weeks of release, the app became the number one search result for Wikipedia in the Android marketplace, and now has more than 3.6 million total user installs and 5.3 million active device installs (as of Sept 2012). As we push our products out to multiple mobile platforms, the PhoneGap development process becomes easier and easier. If we're spending less and less time on each platform then we're doing something right."

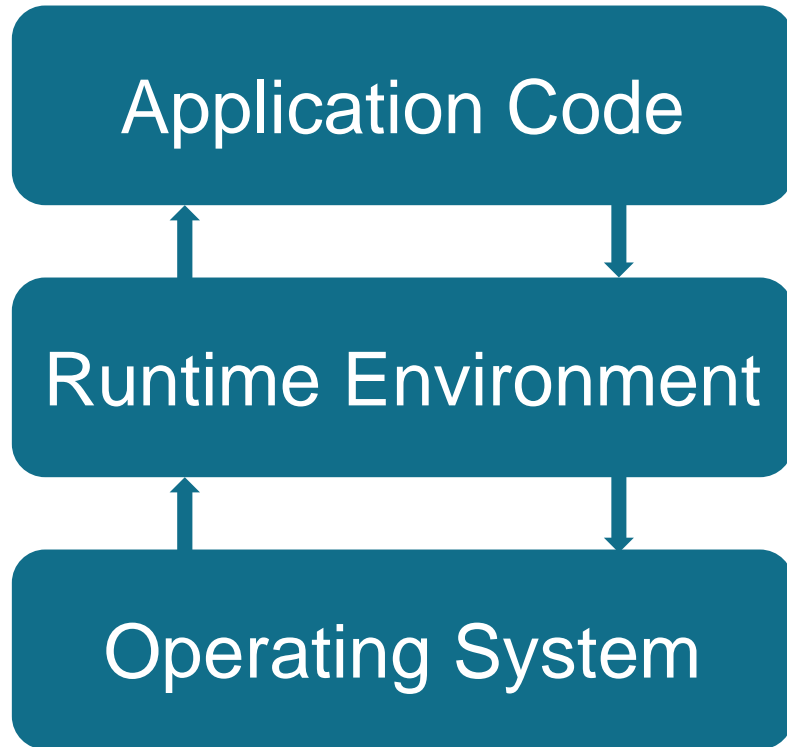
For more information on this topic...



Classification of CPT's

- Web Apps (JavaScript Frameworks)
- Web-to-native Wrappers
- **Runtimes**
- Source code translators
- App Factories

Runtimes



- Cross-platform compatibility layer
- Shields app from underlying differences between platforms
- Different strategies:
 - Interpreted at runtime
 - Compiled in advance (*source code translators*)

Runtimes



- Good overall user experience
- Less reliant on native webview component/JavaScript engine
- Application developers can choose Runtime based on programming language

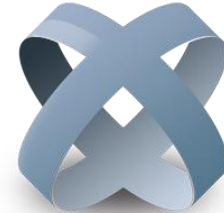


- Often, platform specific code is needed
- Runtimes introduce significant overhead
- Learning curve is often quite steep

Runtimes: examples



ADOBE® AIR™



titanium™



Xamarin



unity

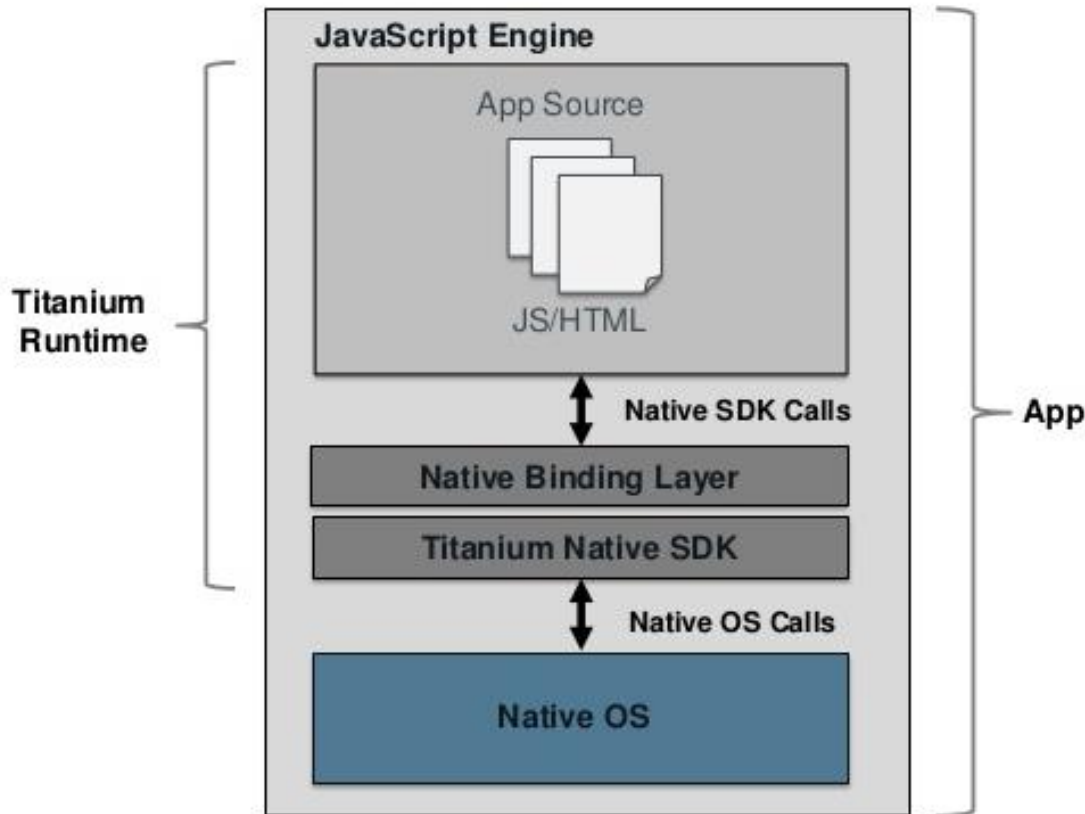


Corona

Titanium explained



titanium™

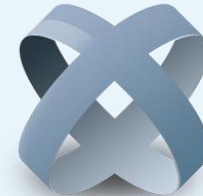


- Written in JavaScript
- No cross-compilation
- JavaScript code evaluated at runtime
- Titanium API maps JavaScript code on Native API (1:1)

Difference between PhoneGap and Titanium



Phone**Gap**



titanium™

Uses JavaScript

Uses JavaScript

WebView

Runtime

Renders HTML pages in
chrome-less browser

Interprets JavaScript code
and maps on Native API

Developer writes Web app

Developer writes “native”
application using JavaScript

Classification of CPT's

- Web Apps (JavaScript Frameworks)
- Web-to-native Wrappers
- Runtimes
- **Source code translators**
- App Factories

Source Code Translator



- Different strategies:
 - Translate to native source
 - Translate to intermediary language
 - Translate to low level machine code
- Often used in combination with Runtime

Source Code Translator



- Good user experience and performance
- Application developers can choose tool based on programming language
- Generate 100% native applications



- Often, platform specific code is needed
- Learning curve is often quite steep
- High complexity, supporting new APIs is time consuming, extending the framework is not trivial

Source Code Translator: examples

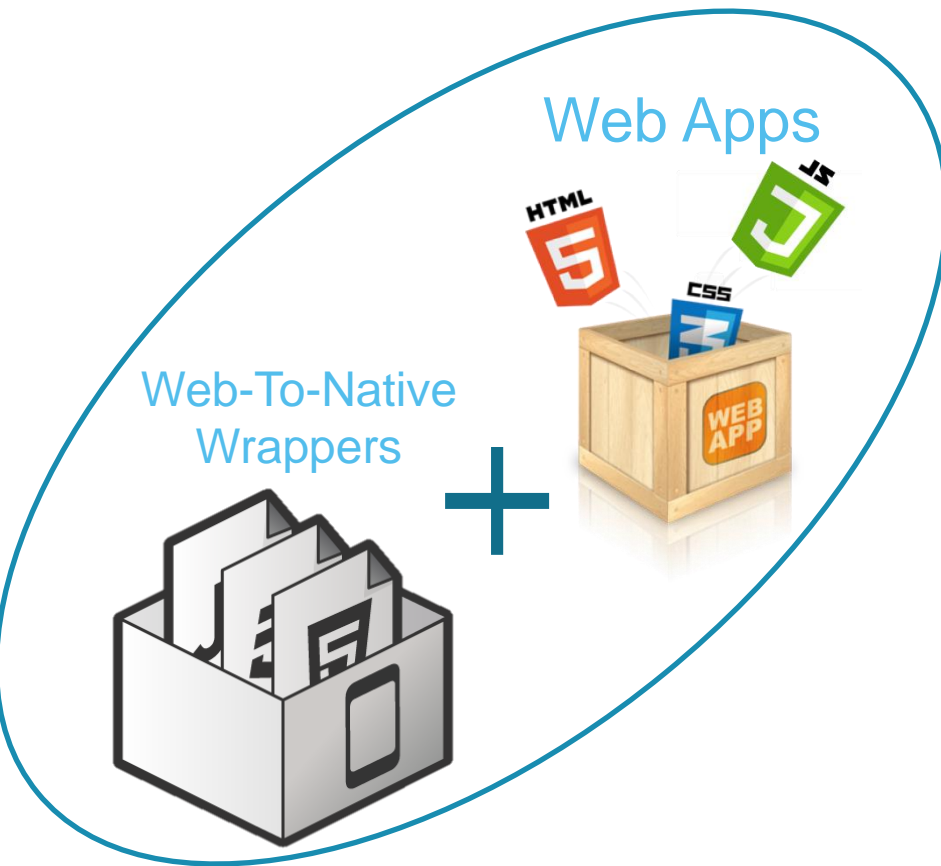


Source Code Translator: NEOMAD

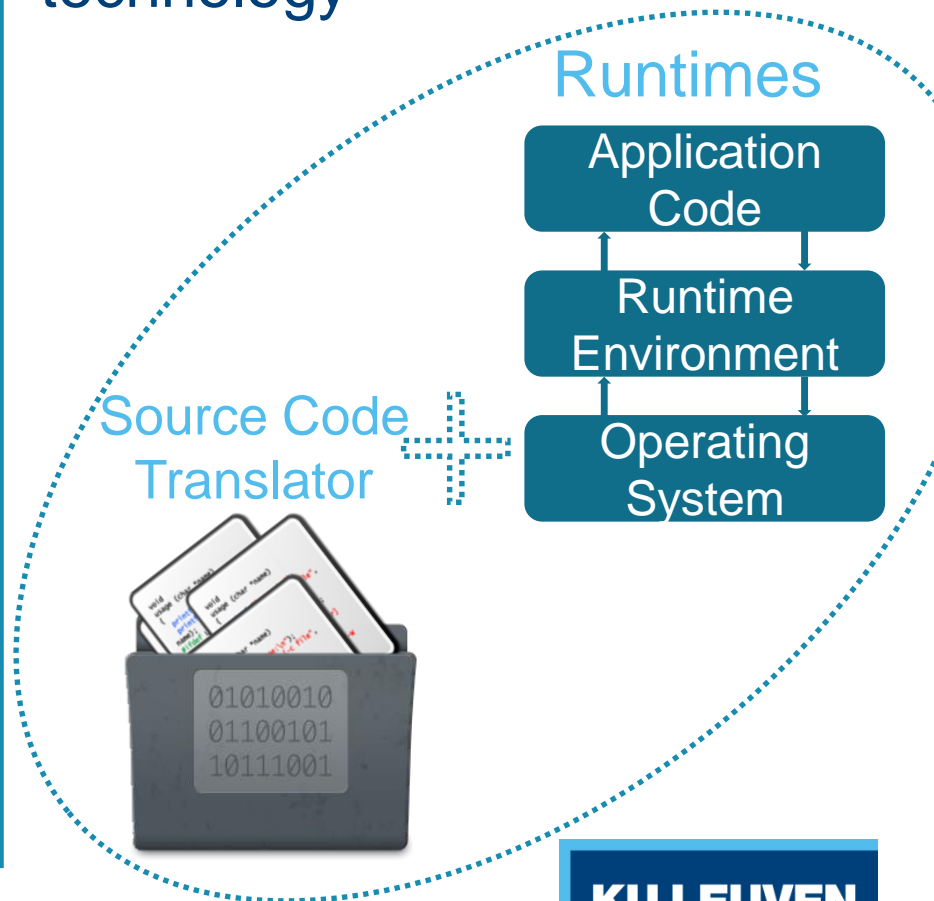


Cross-platform technology

1) Based on web technology



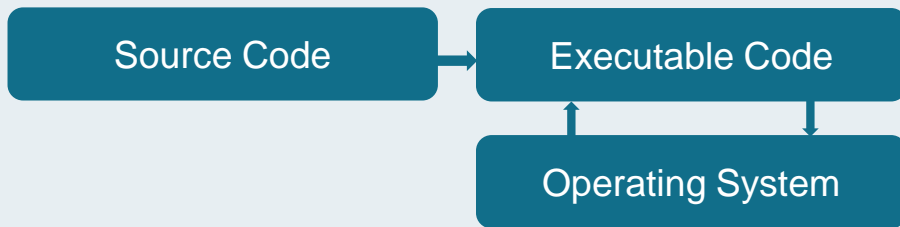
2) Not based on Web technology



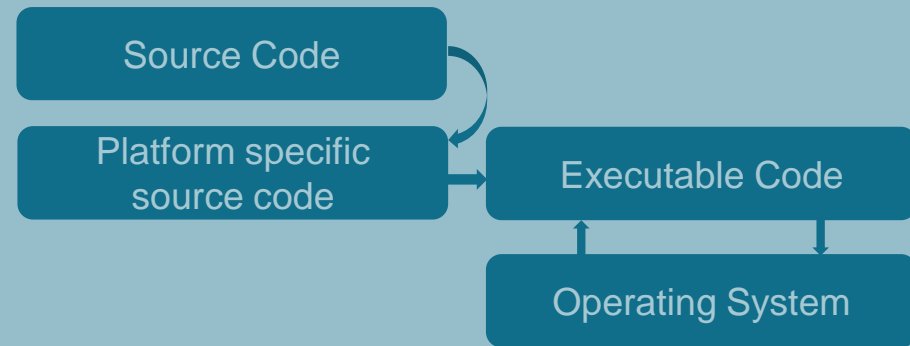
Runtimes & Source code translators

4 possibilities

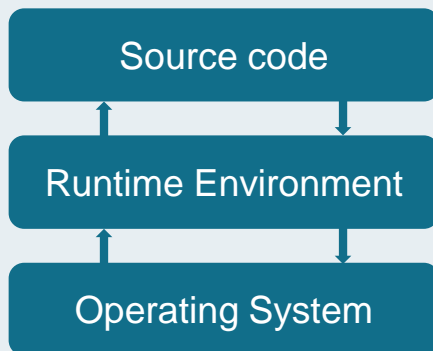
1)



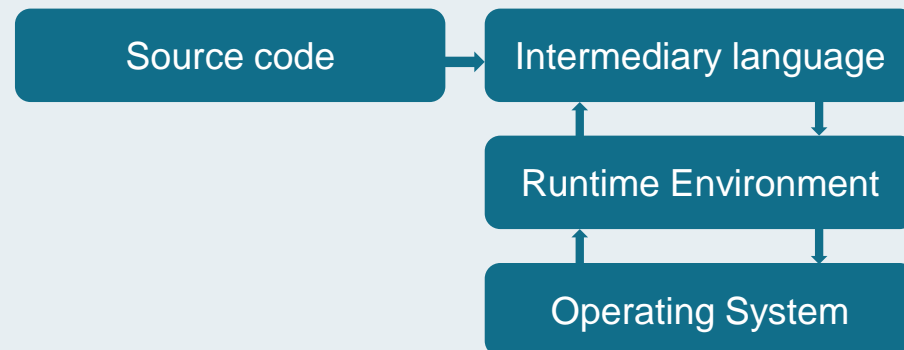
2)



3)



4)

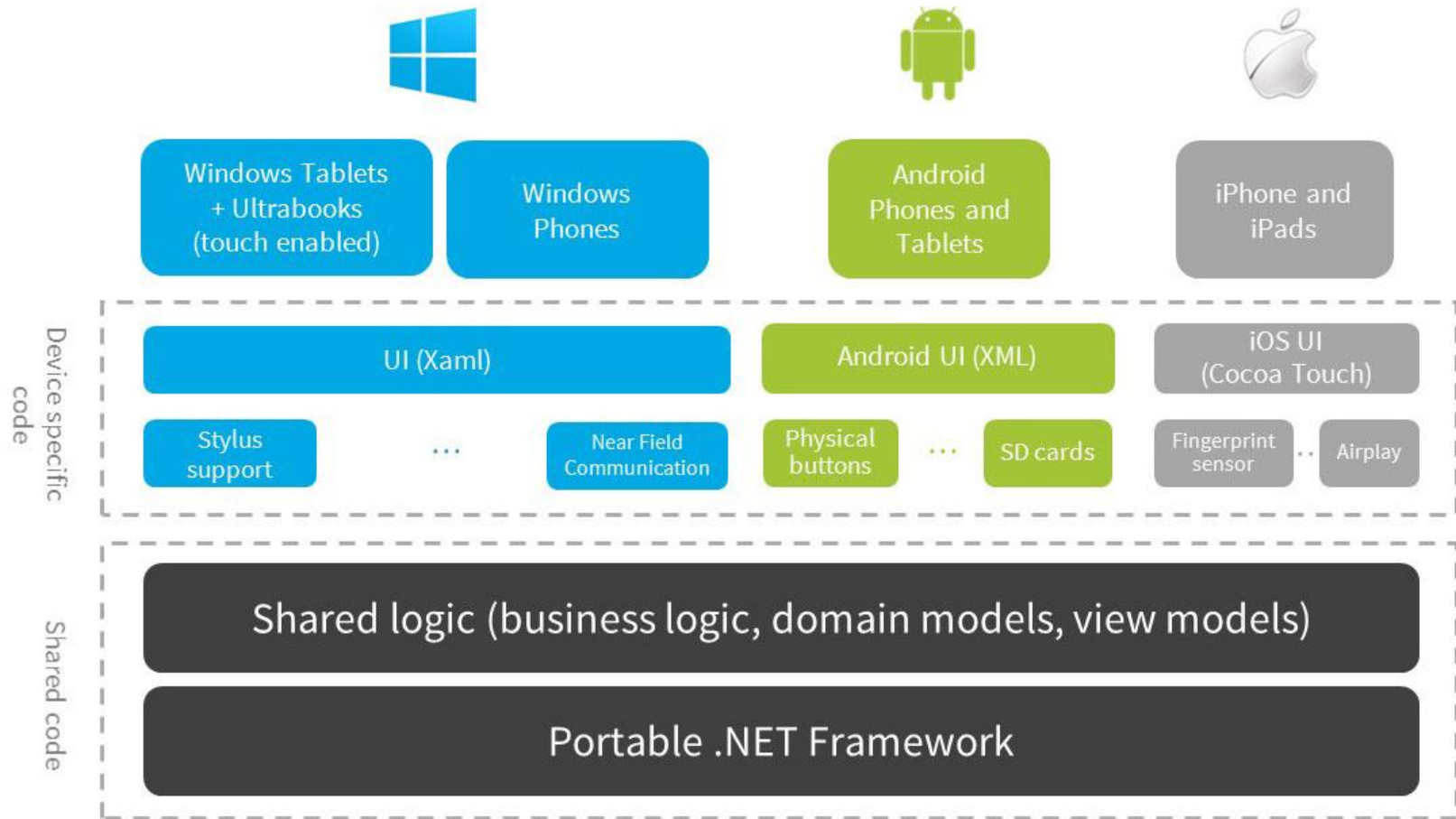


Combination Source Code Translator and Runtime: Xamarin

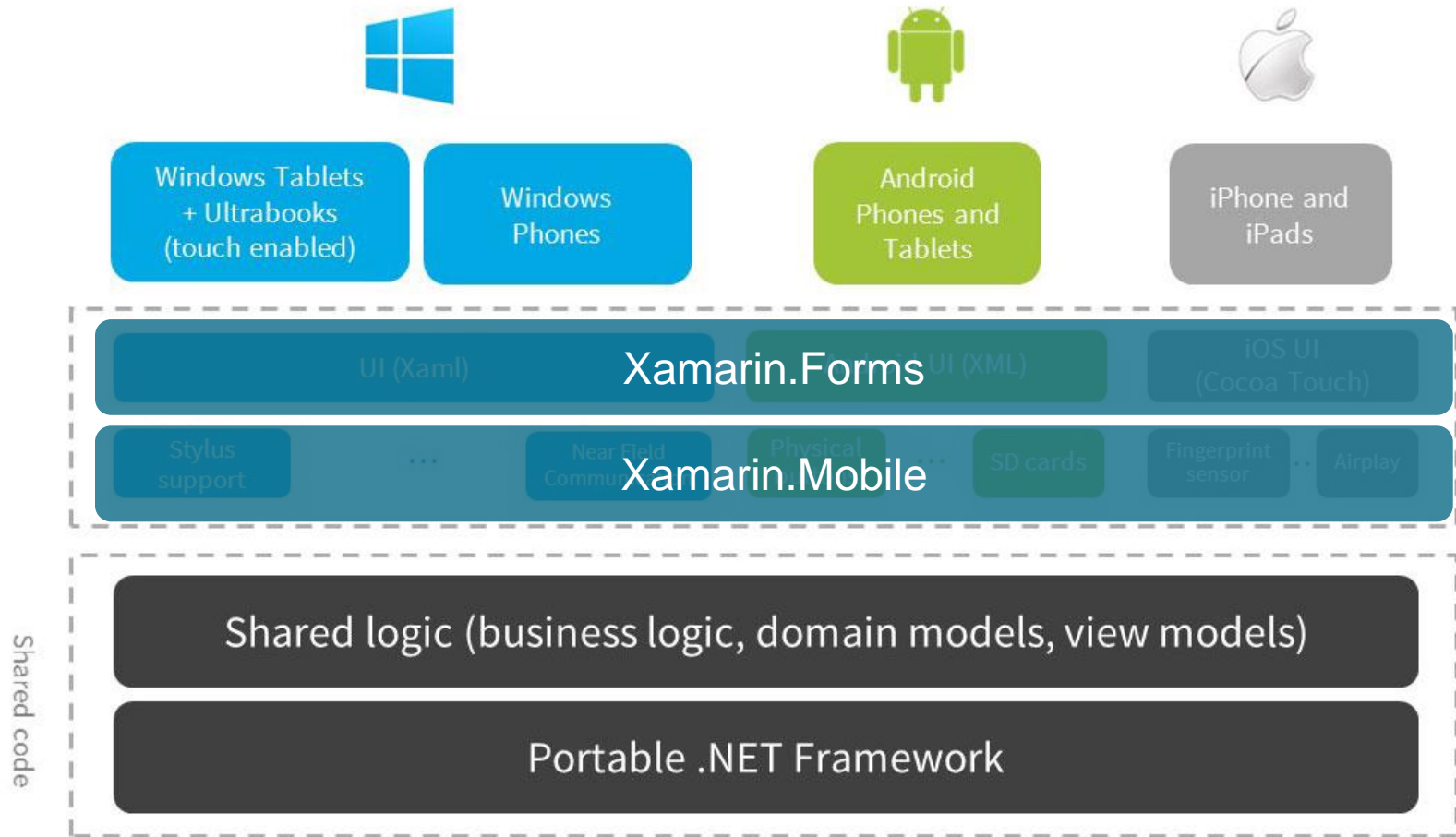
- Uses Runtime
- Code written in C#
- Supported platforms:
 - Android
 - iOS
 - Windows Phone
 - (OS X)
 - ...
- Recently acquired by Microsoft



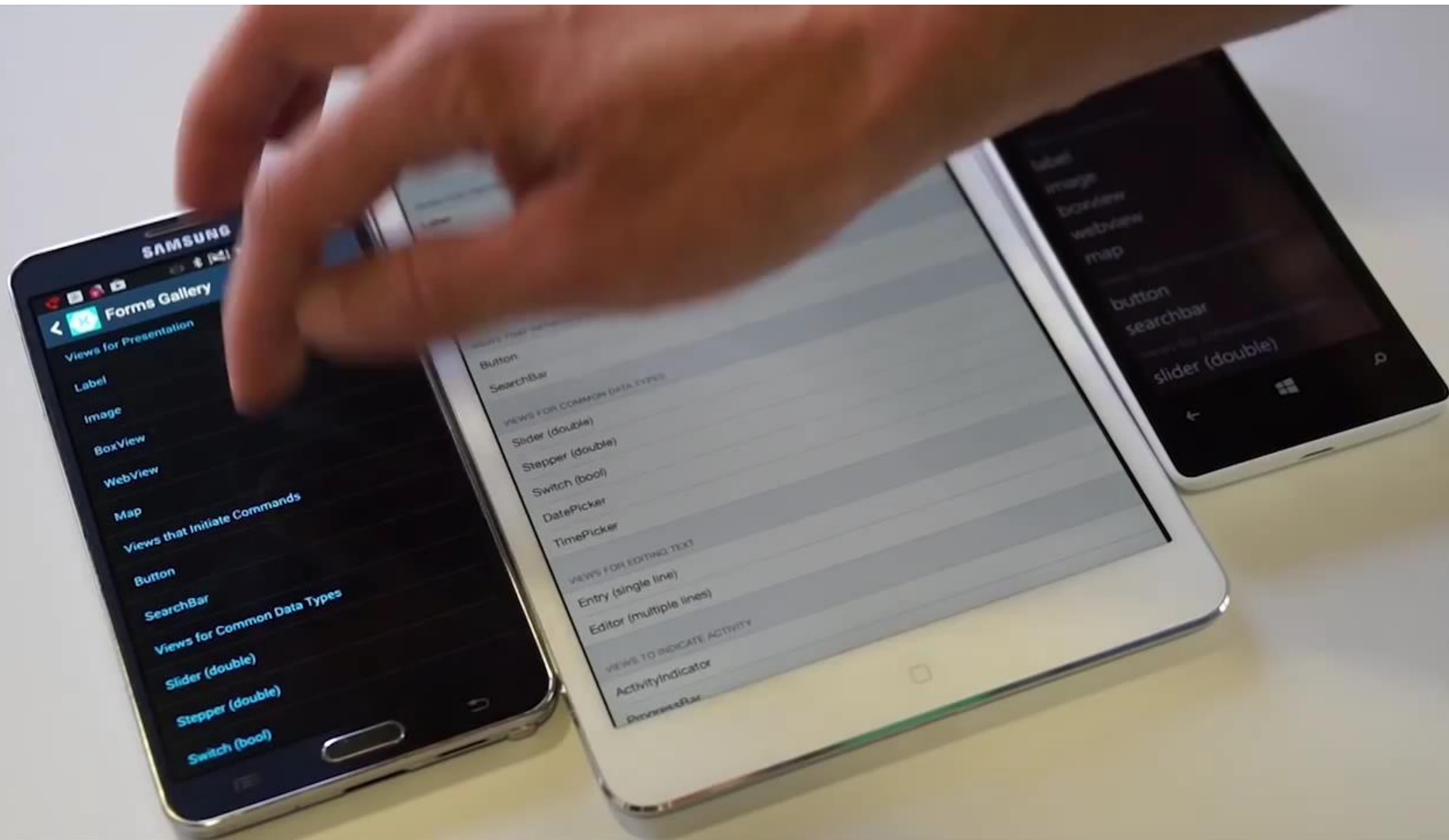
Xamarin explained: development



Xamarin explained: Xamarin.Forms



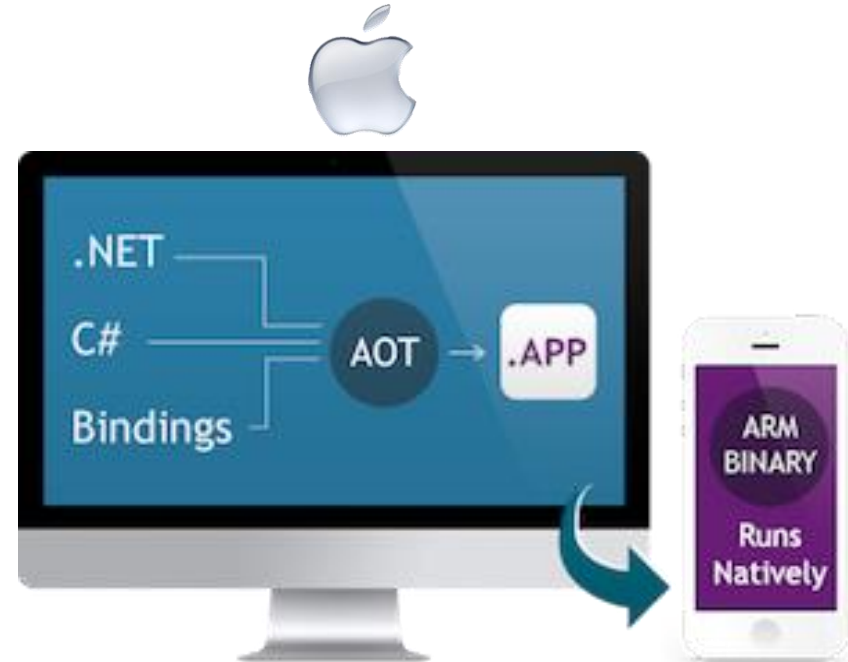
Xamarin.Forms



Xamarin: Android vs iOS



- Source translated to Intermediary Language (IL)
- Just-In-Time (JIT) compilation

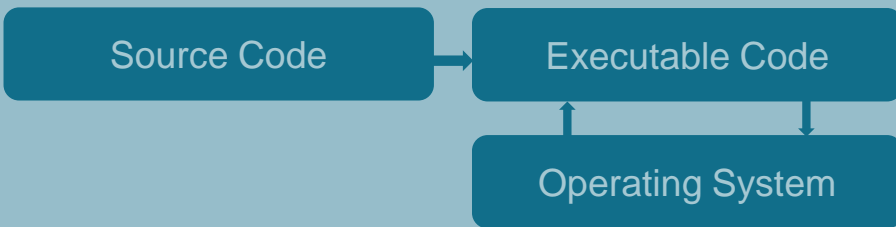


- Source translated to executable binary code
- Ahead-Of-Time (AOT) compilation

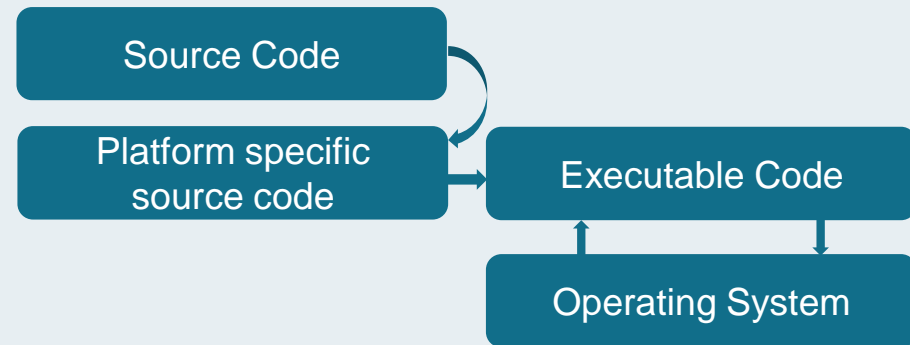
Runtimes & Source code translators

4 possibilities

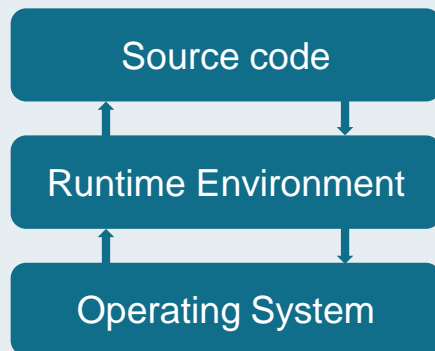
1)



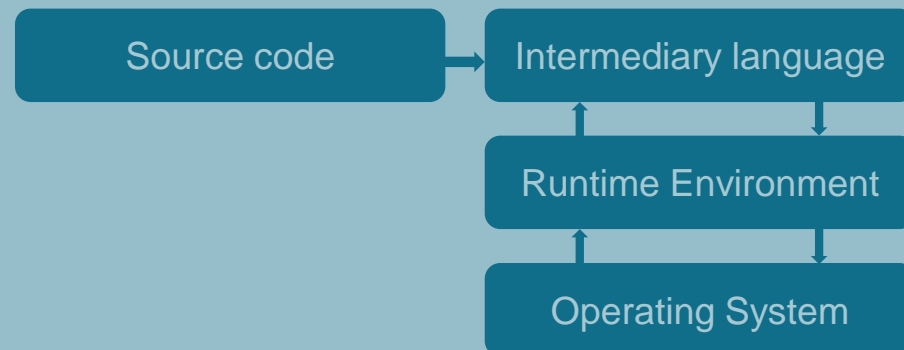
2)



3)



4)



Unity



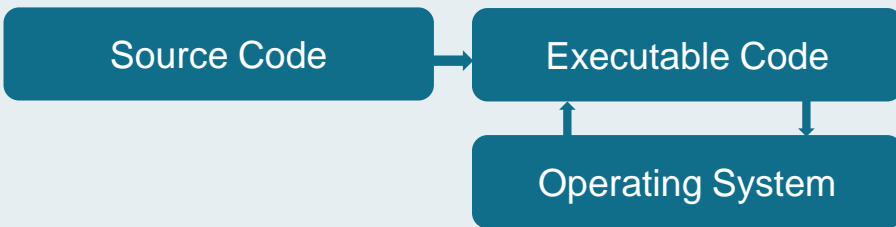
- Used in many gaming applications
- Specialised in rendering 3D and animated images
- Runtime (with translation to an intermediary language)



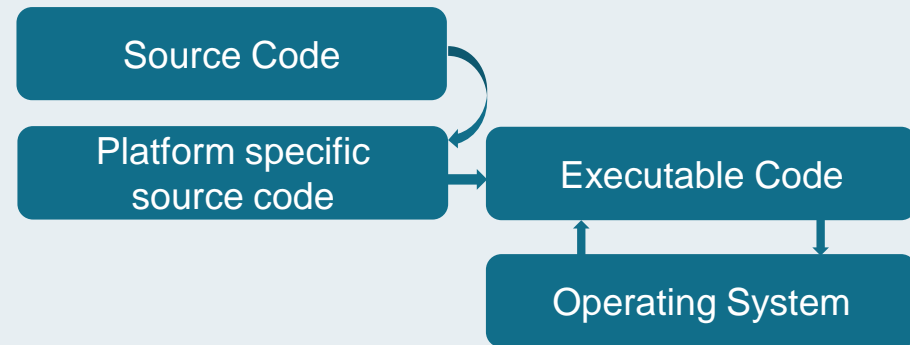
Runtimes & Source code translators

4 possibilities

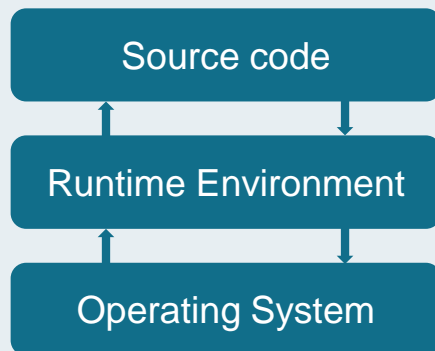
1)



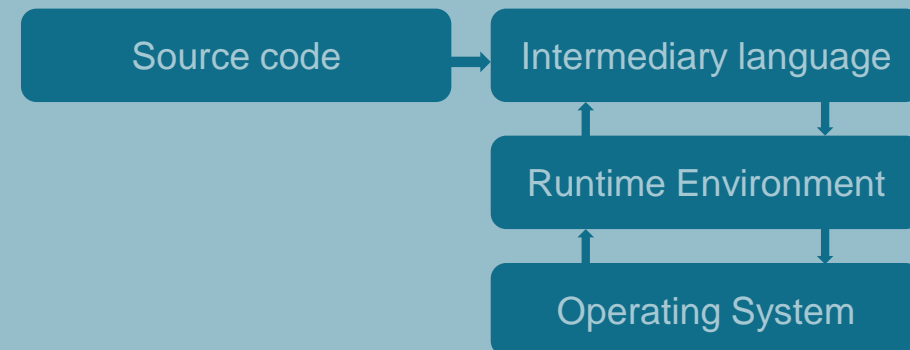
2)



3)



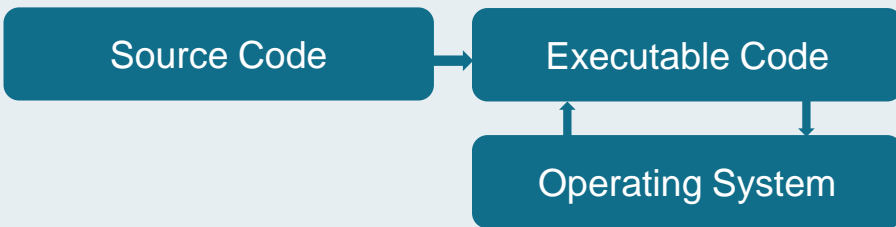
4)



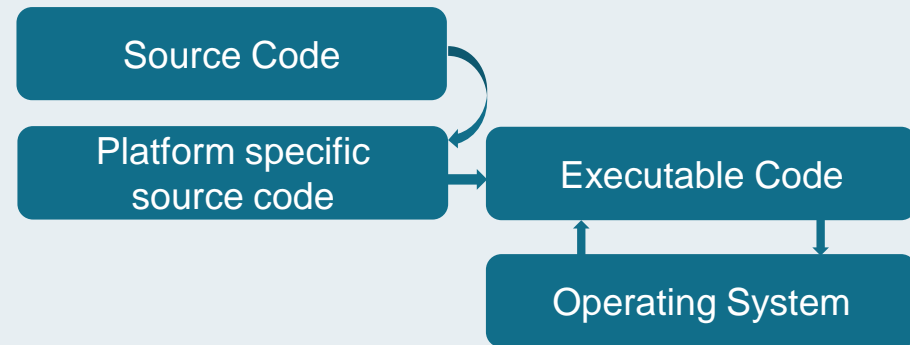
Runtimes & Source code translators

4 possibilities

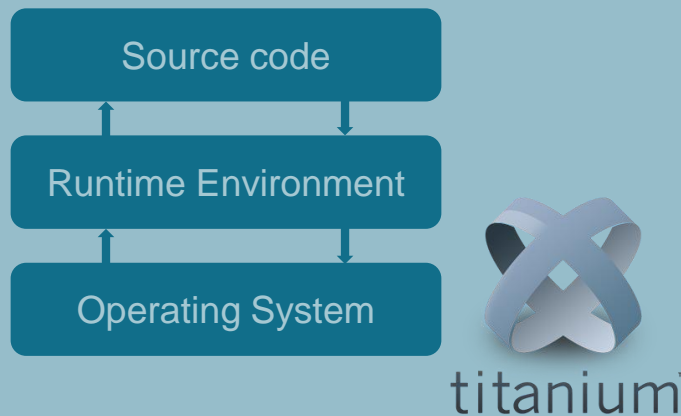
1)



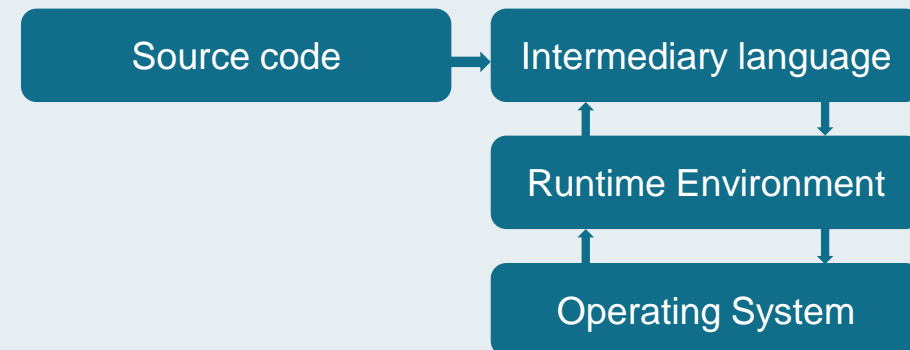
2)



3)



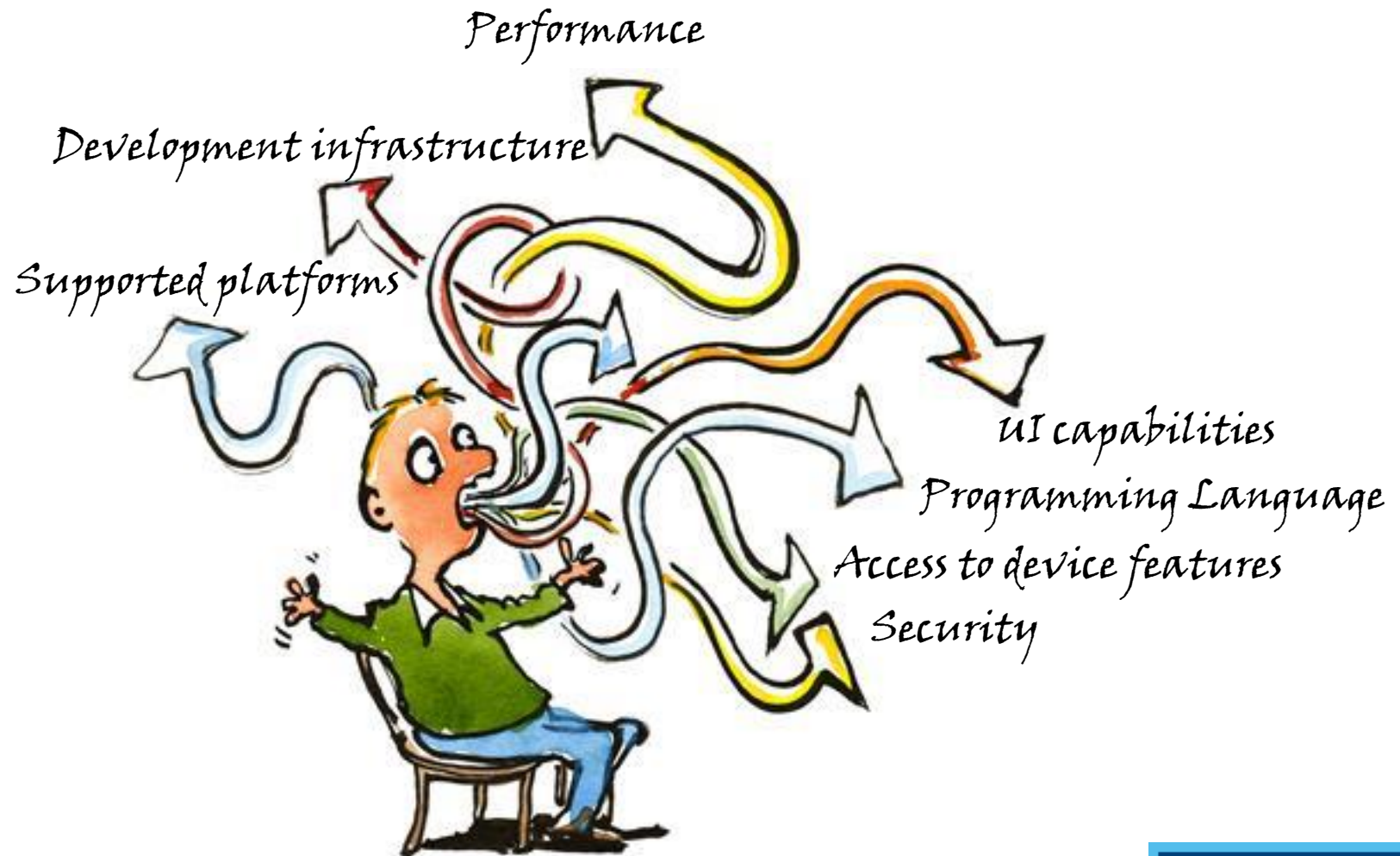
4)



Cross-Platform Tool Selection Criteria



Cross-Platform Tool Selection Criteria



Supported platforms

Technology	Tool	Android	iOS	WP
JavaScript Framework + PhoneGap	ALL JavaScript Frameworks			
Runtime	Titanium			
	NativeScript			ALPHA
	ReactNative			ALPHA
Source code Translator	Eqela			
	NeoMAD			
Source Code Translator + Runtime	Xamarin			
	Qt			
	Adobe Air			
	Unity			
App Factory	AppMakr			
	ViziApps			

Development infrastructure: Programming languages

Technology	Tool	Programming language
JavaScript Framework + PhoneGap	ALL JavaScript Frameworks	JavaScript, HTML, CSS
Runtime	Titanium	JavaScript
	NativeScript	JavaScript
	ReactNative	JavaScript
Source code Translator	NeoMAD	Java
Source Code Translator + Runtime	Xamarin	C#
	Qt	C++/QML
	Adobe Air	ActionScript
	Unity	C#, UnityScript (JavaScript)
App Factory	AppMakr	Drag & Drop
	ViziApps	Drag & Drop

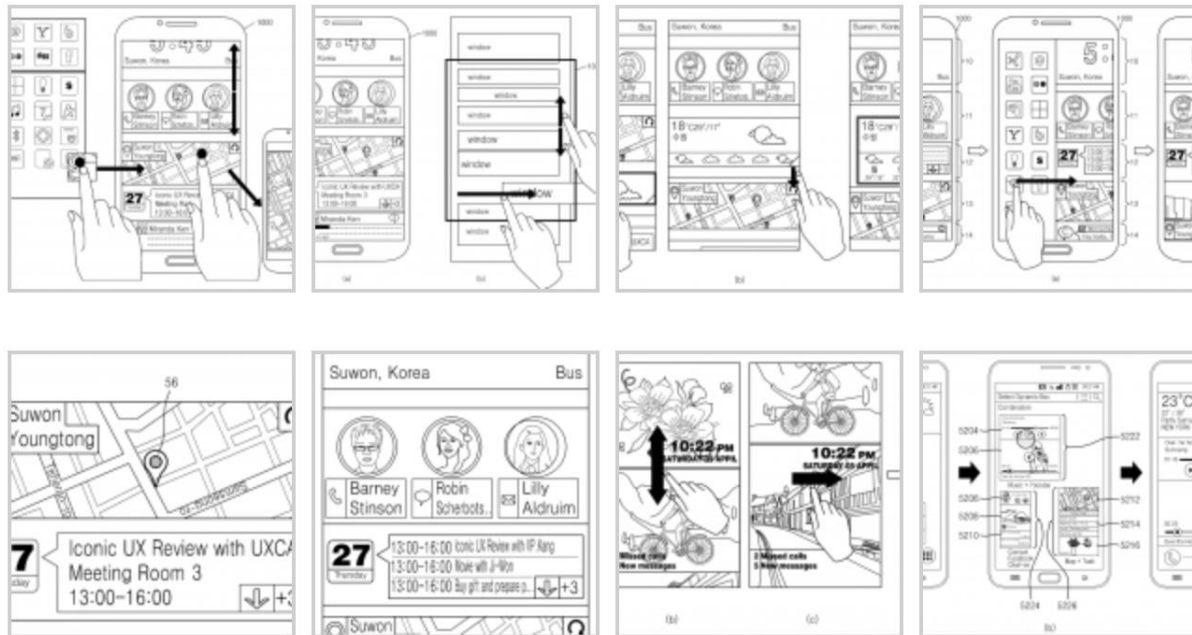
Development infrastructure: Programming environment

Technology	Tool	Programming environment
JavaScript Framework + PhoneGap	ALL JavaScript Frameworks	Any text editor / web IDE
Runtime	Titanium	Titanium IDE
	NativeScript	Appbuilder
	ReactNative	Text editor, Nuclide, Deco
Source code Translator	NeoMAD	NeoMAD IDE (based on eclipse)
Source Code Translator + Runtime	Xamarin	Xamarin Studio
	Qt	QT creator
	Adobe Air	Adobe Flash Builder
	Unity	Visual Studio (+plugin)
App Factory	AppMakr	Cloud development tool
	ViziApps	Cloud development tool

Development infrastructure: License cost

Technology	Tool	Programming environment
JavaScript Framework + PhoneGap	ALL JavaScript Frameworks	PhoneGap: always free FREE: Ionic, jQuery Mobile, ... PAID: Sencha Touch \$4475+/year (5 devs), ...
Runtime	Titanium	\$39/month (1 dev)
	NativeScript	Free
	ReactNative	Free
Source code Translator	NeoMAD	\$999/year (1 dev)
Source Code Translator + Runtime	Xamarin	Free community license
	Qt	\$3540/year (1dev)
	Adobe Air	Free
	Unity	\$75/month (1dev)
App Factory	AppMakr	\$1/month
	ViziApps	\$33/month

UI Capabilities

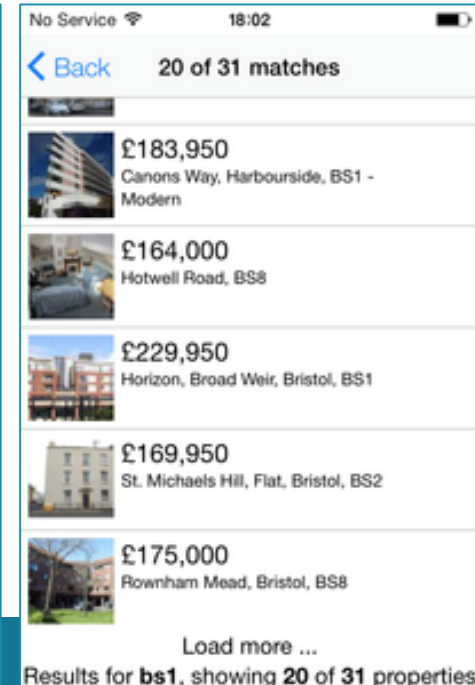
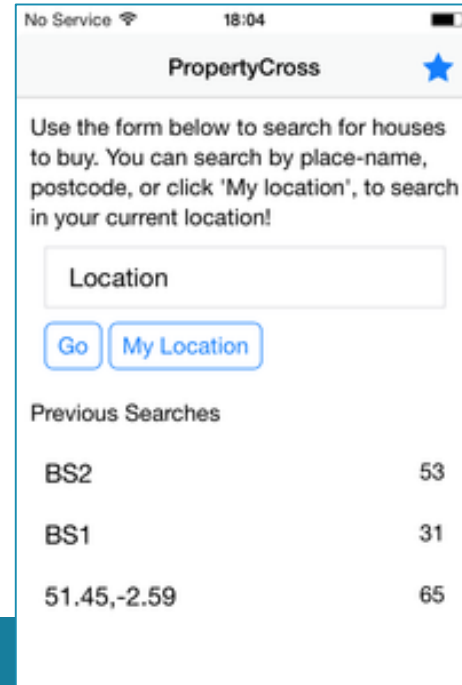
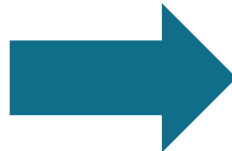
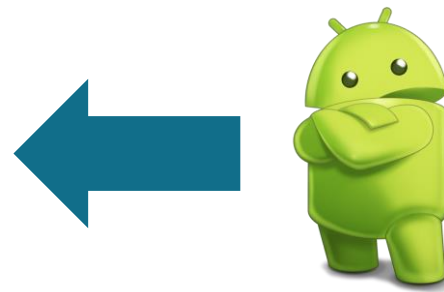
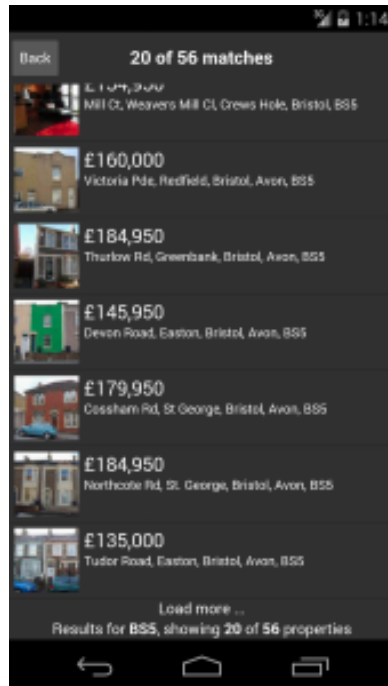
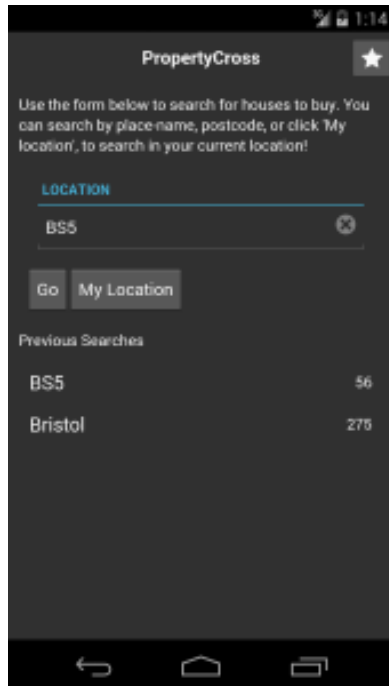


UI Capabilities:

Web Apps and Web-To-Native Wrappers

- Easy UI development
 - CSS templates
 - JavaScript Frameworks (e.g. Ionic)
- Tons of CSS and JavaScript Frameworks freely available
- Good looking applications with little to no effort
- Some JavaScript Frameworks offer native *skins* (e.g. Sencha Touch)

Example Sencha Touch



UI Capabilities:

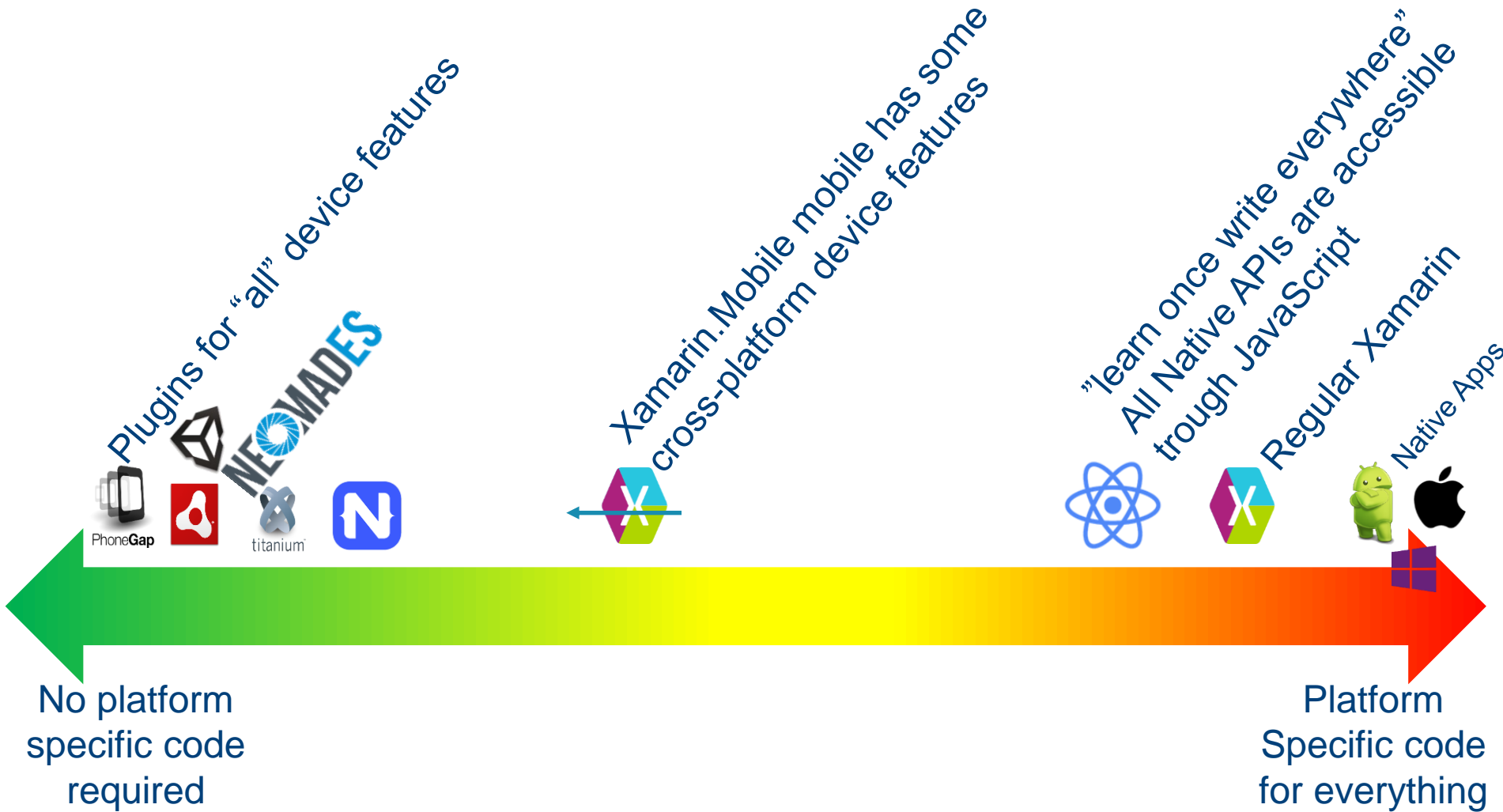
Runtimes and Source Code Translators

- Sometimes, platform specific code is needed for the UI (e.g. Xamarin)
- Often access to native UI *components* (e.g. Xamarin, native javascript frameworks)
- Some provide advanced graphical support (e.g. Unity, Qt)
 - Game Engines
 - 2D and 3D acceleration
 - ...

Example Unity



Device feature access & platform specific code



Performance

- See presentation: *Comparing performance parameters of mobile app development strategies*



Performance

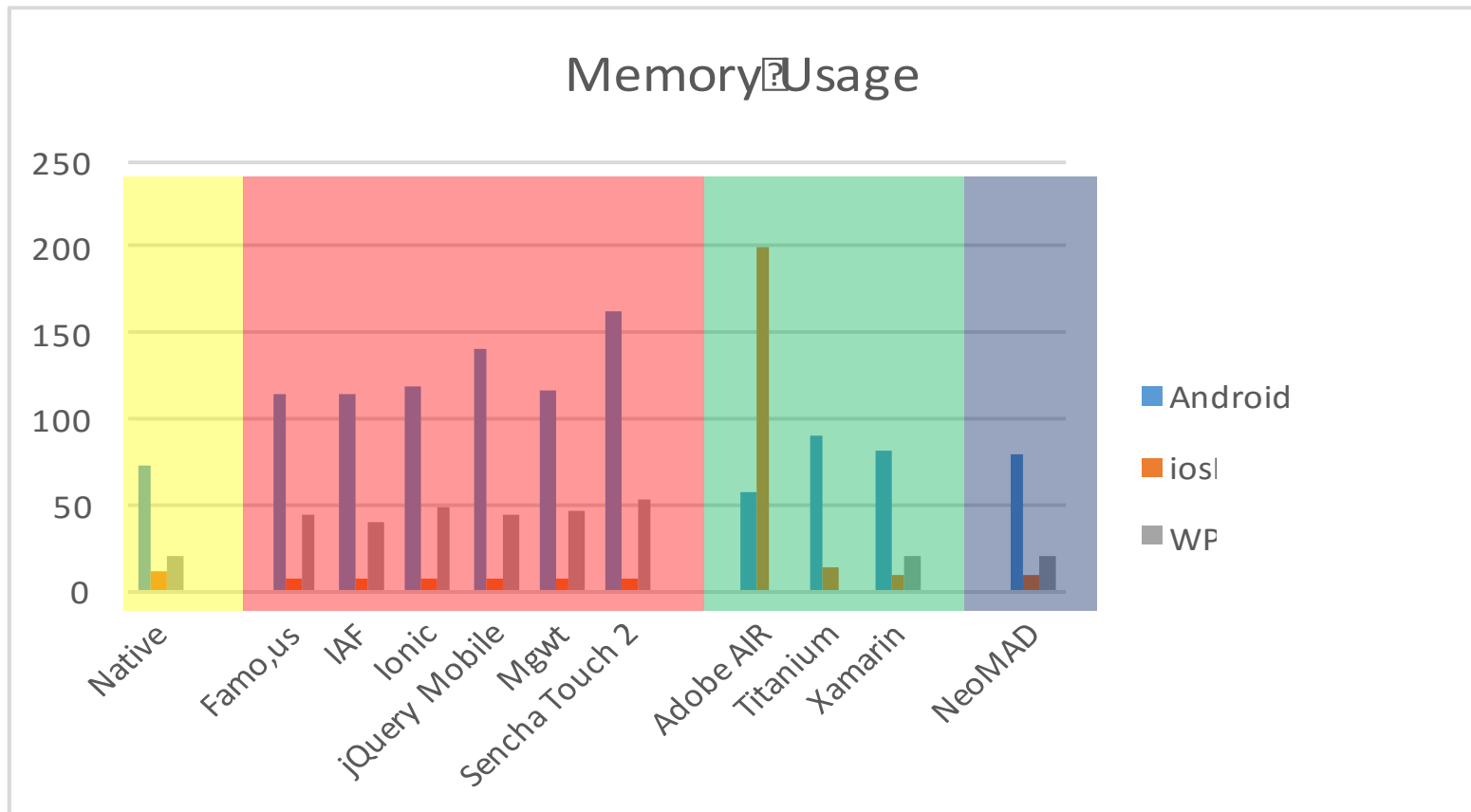
Cross-platform tools of the same category show similar behavior

The performance penalty resulting from the use of cross-platform tools is generally acceptable

Page rendering: JavaScript frameworks vs Runtimes, speed vs Native UI components

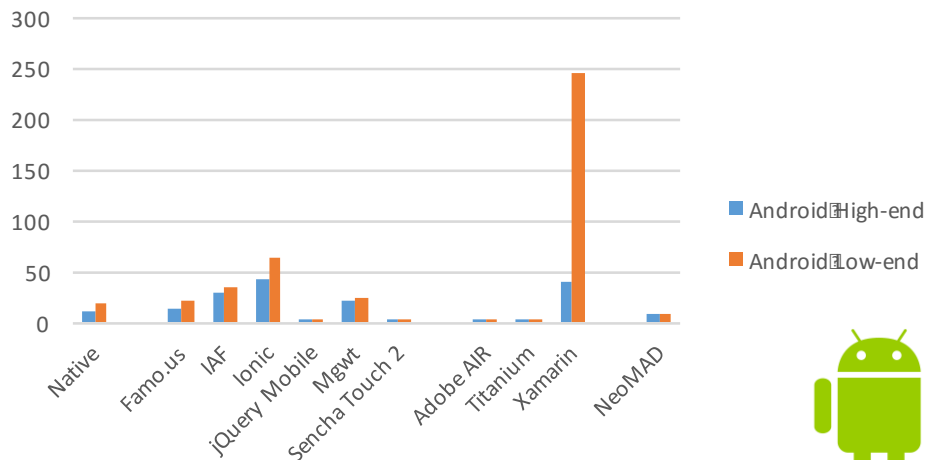
The performance of a cross-platform application strongly depends on the targeted platform

Cross-platform tools of the same category show similar behavior

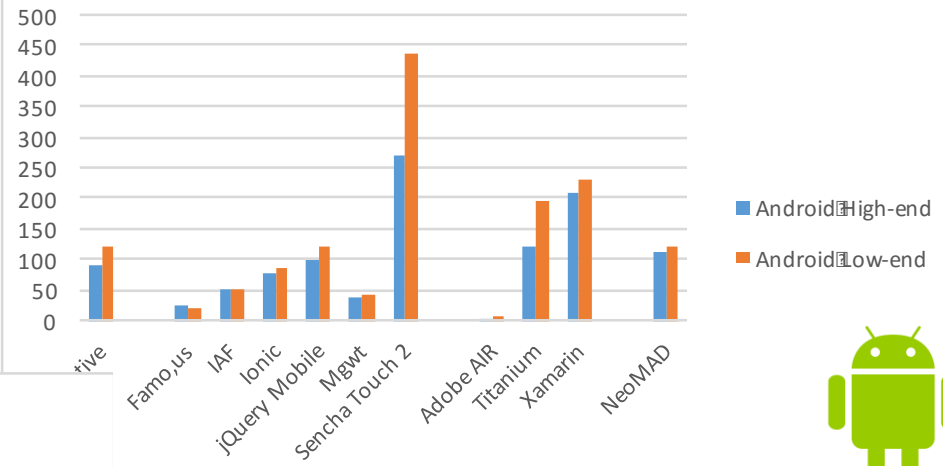


The performance penalty resulting from the use of cross-platform tools is generally acceptable

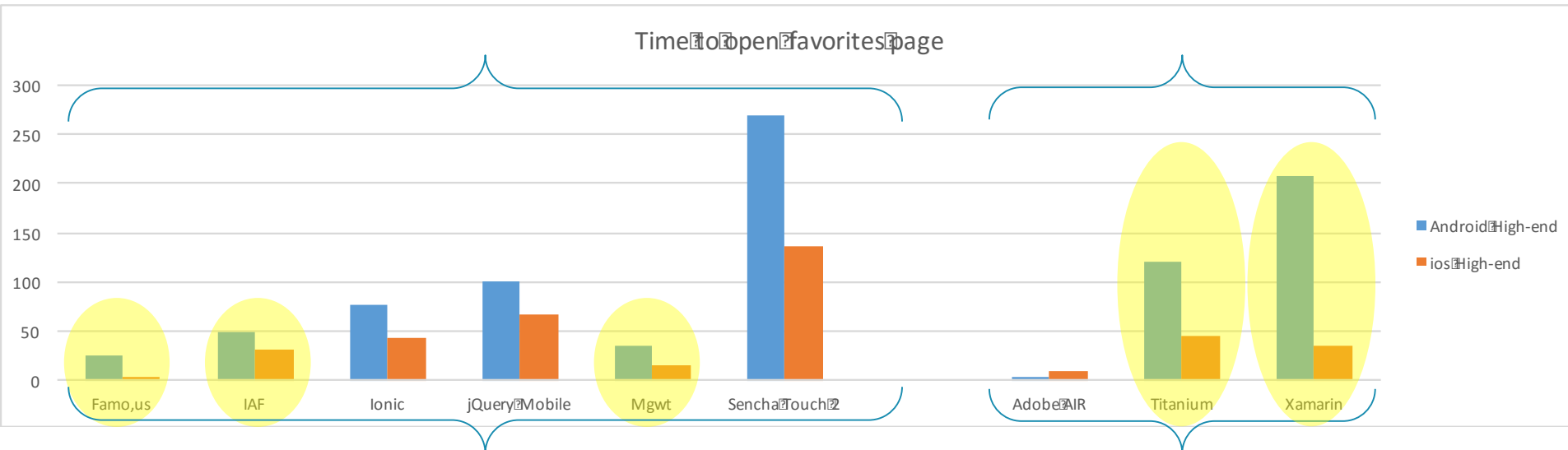
Time to return to previous page of the app



Time to open a favorite page of the application



Page rendering: JavaScript frameworks vs Runtimes, speed vs Native UI components



JavaScript Frameworks

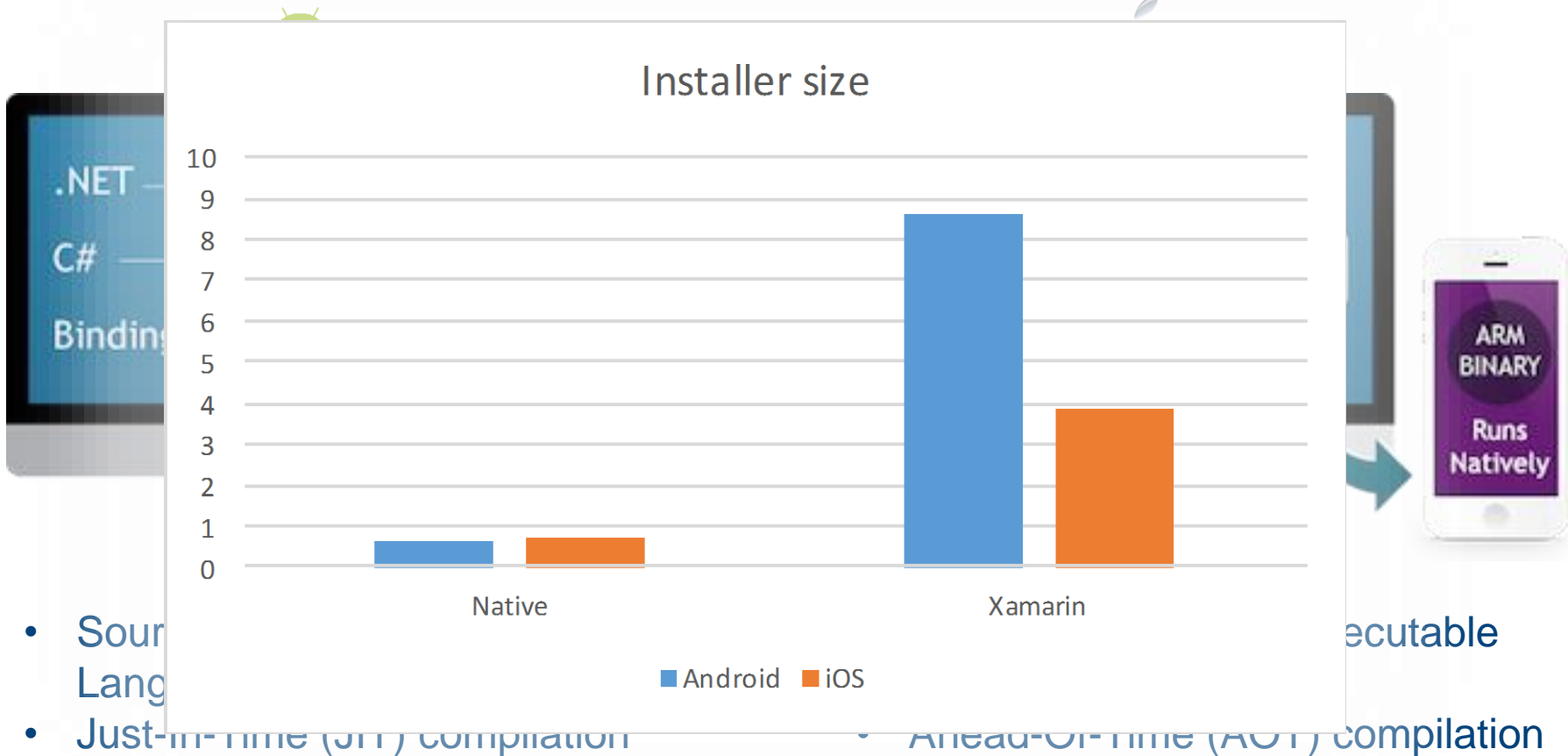
- Webview renders HTML pages
- Some JavaScript frameworks have faster response times than native apps
- Sometimes native skins
No real, native UI components

Runtimes

- Creates UI View elements
- Makes use of Native UI components
- Additional overhead introduced

The performance of a cross-platform application strongly depends on the targeted platform

Xamarin: Same tool, different strategy



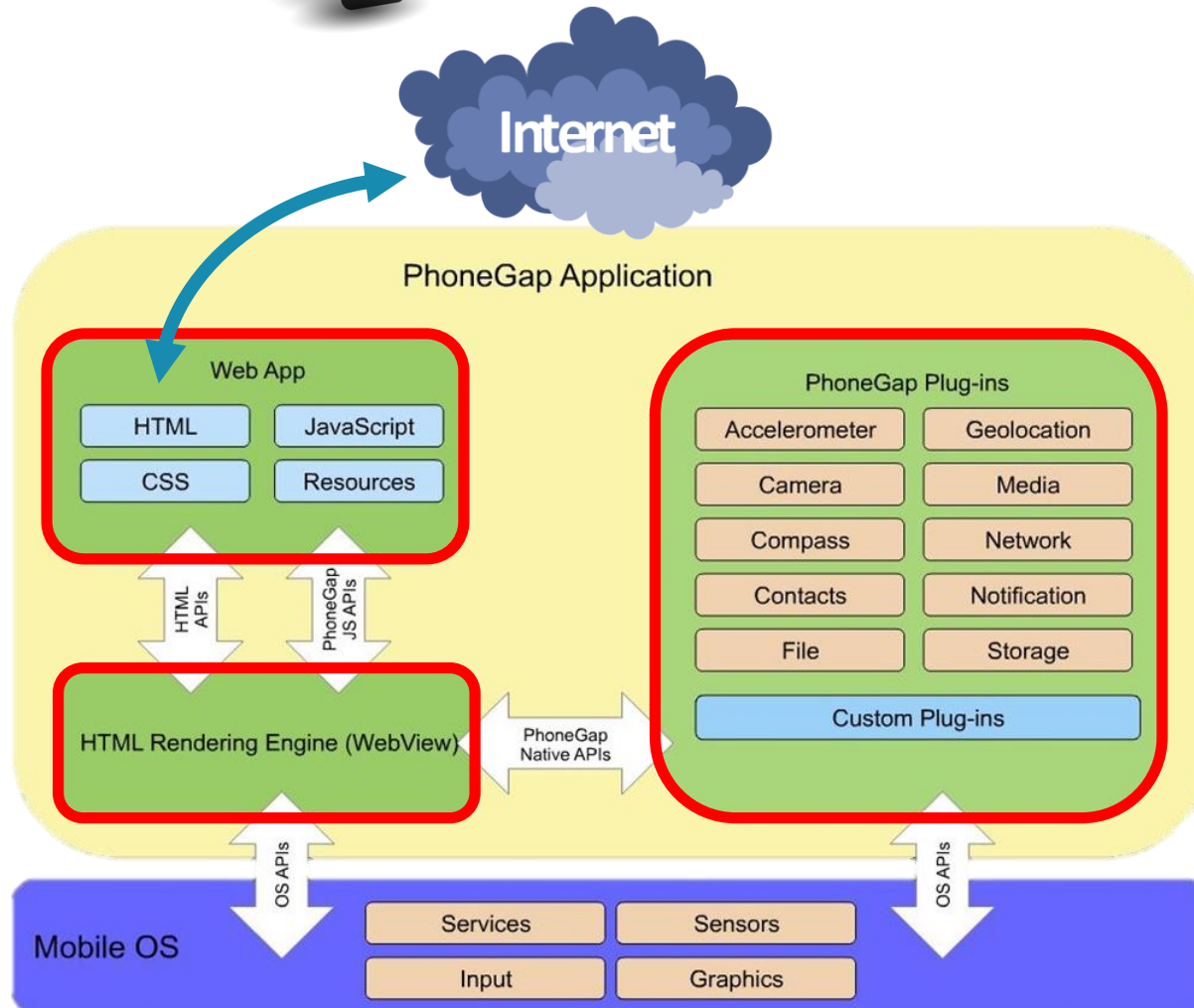
Security



Security concerns in cross-platform apps

- **Possible introduced software vulnerabilities by**
 - Runtimes
 - Translation of code
 - Mapping of code on native APIs
 - Extra software layers
- **Significant part of the code base in the application becomes third party**
 - Developer has little to no control over this.

Case Study: PhoneGap












Plugins and Security



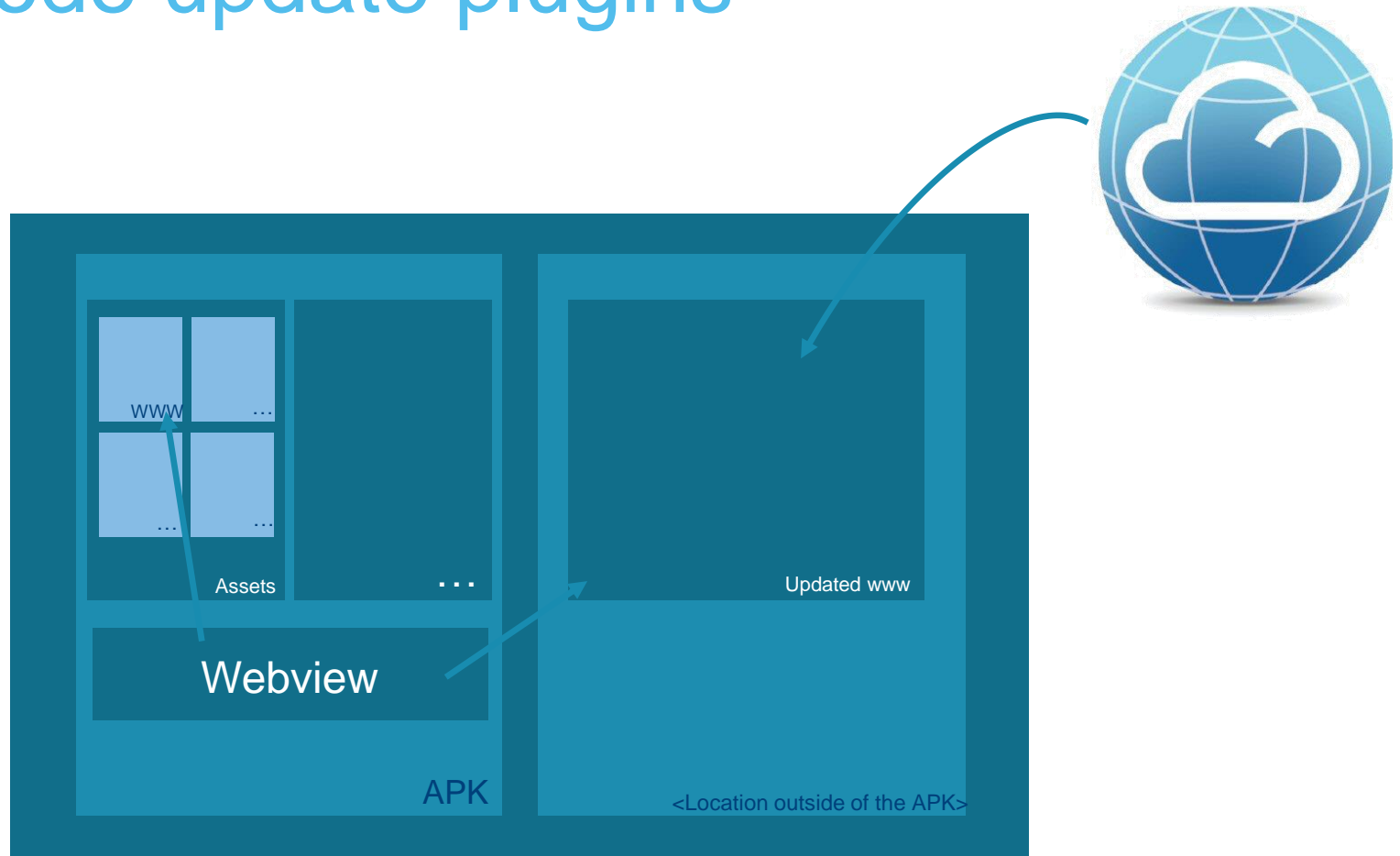
Hot code update plugins

- Allow PhoneGap applications to be updated without the app

cordova-hot-code-push-plugin Cordova plugin to perform code updates on the fly  	v1.2.5 by nikdem	MIT 509 downloads last month Last updated 47 days ago	
cordova-plugin-code-push CodePush Plugin for Apache Cordova  	v1.5.1-beta by vsmobile	Licensed 393 downloads last month Last updated 19 days ago	
meteor-cordova-update-plugin CordovaUpdate is a plugin to serve files from file-system	v0.0.2 by luisherranz	MIT 80 downloads last month Last updated 255 days ago	
cordova-plugin-dynamic-update Cordova Plugin to dynamically update the app www directory. 	v0.2.1 by leecrossley	MIT 38 downloads last month Last updated 199 days ago	



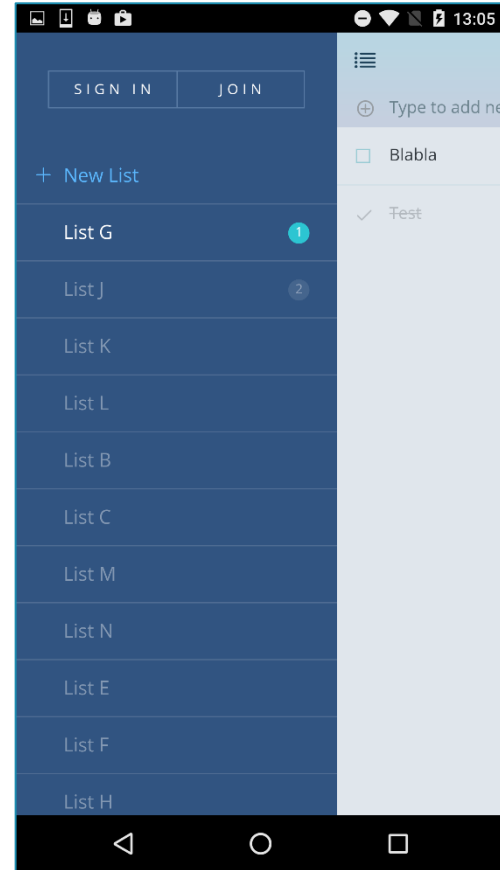
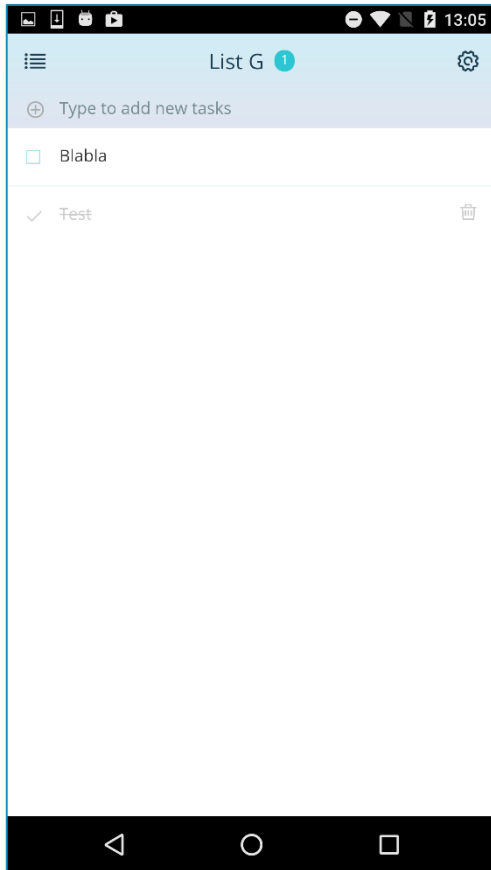
Hot code update plugins



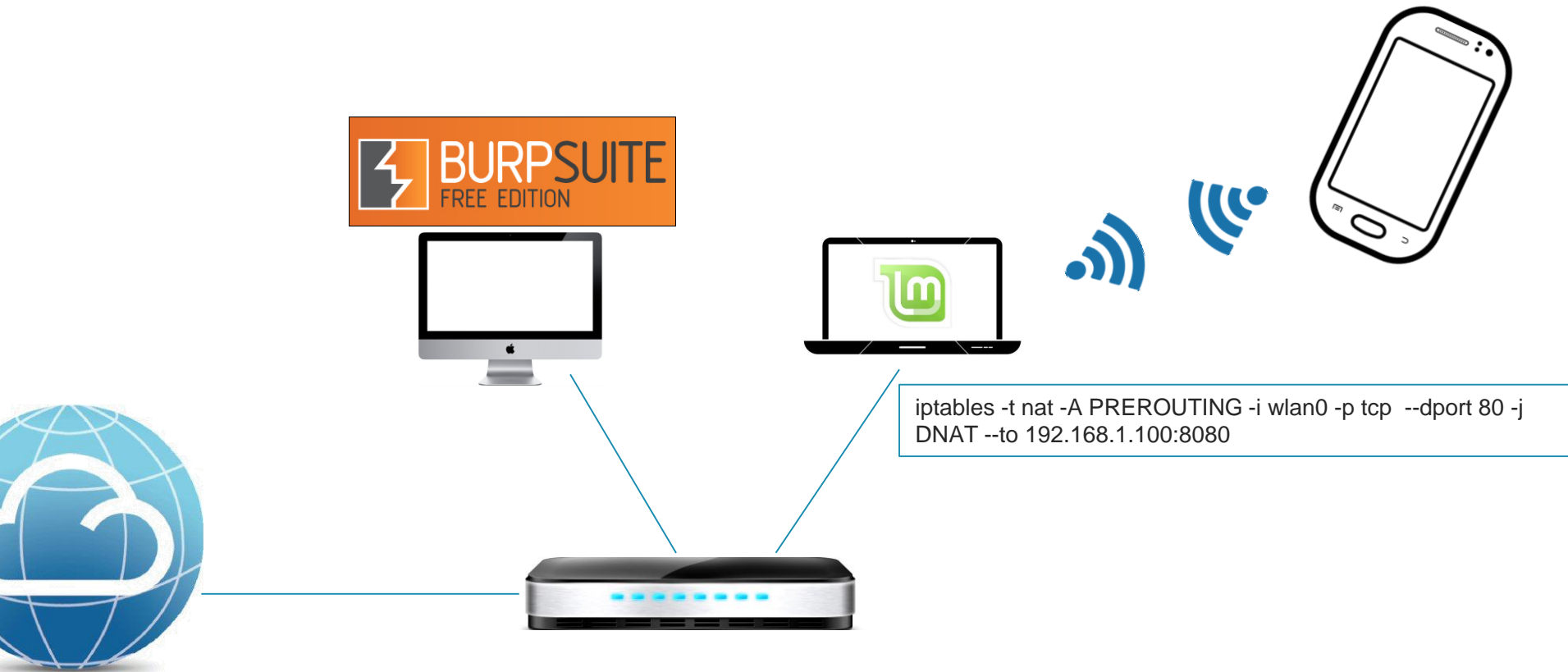
Meteor-cordova-update-plugin



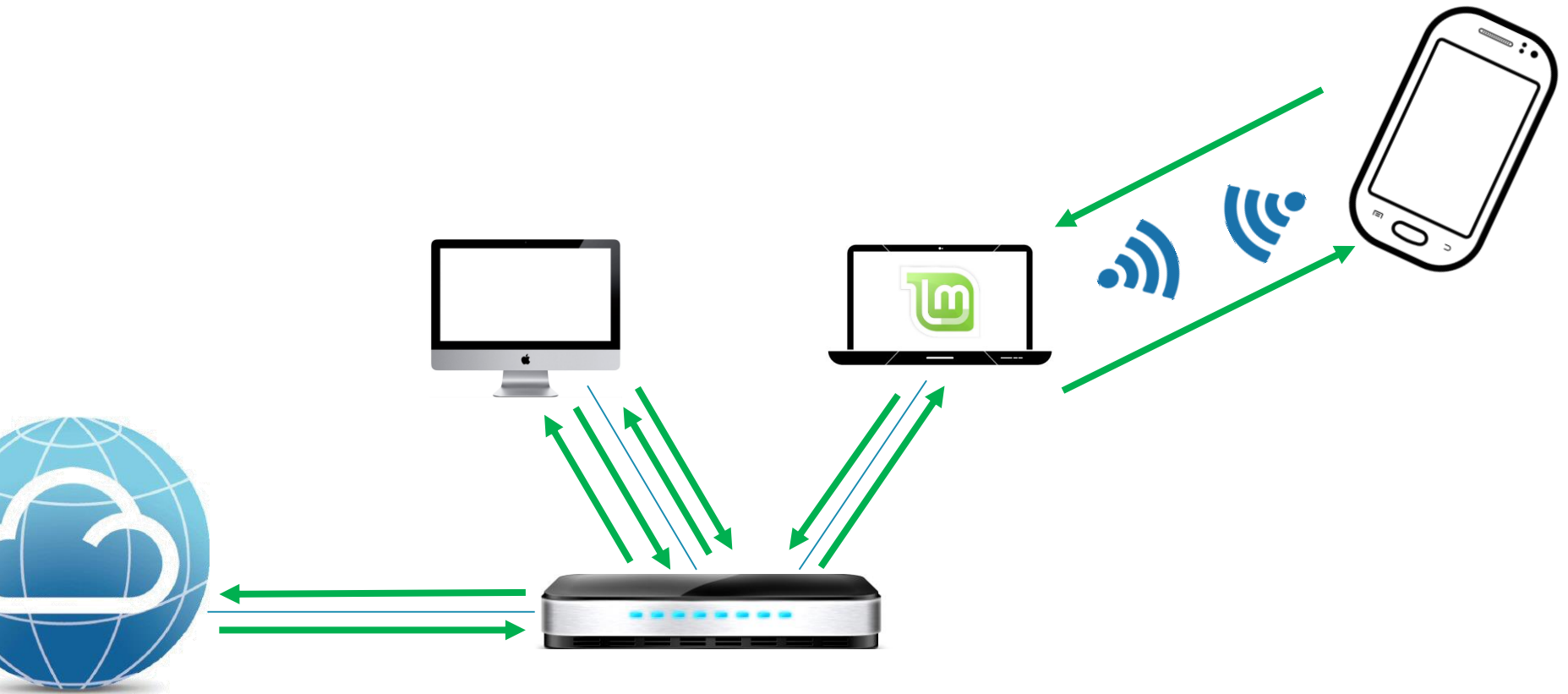
Meteor Todo App



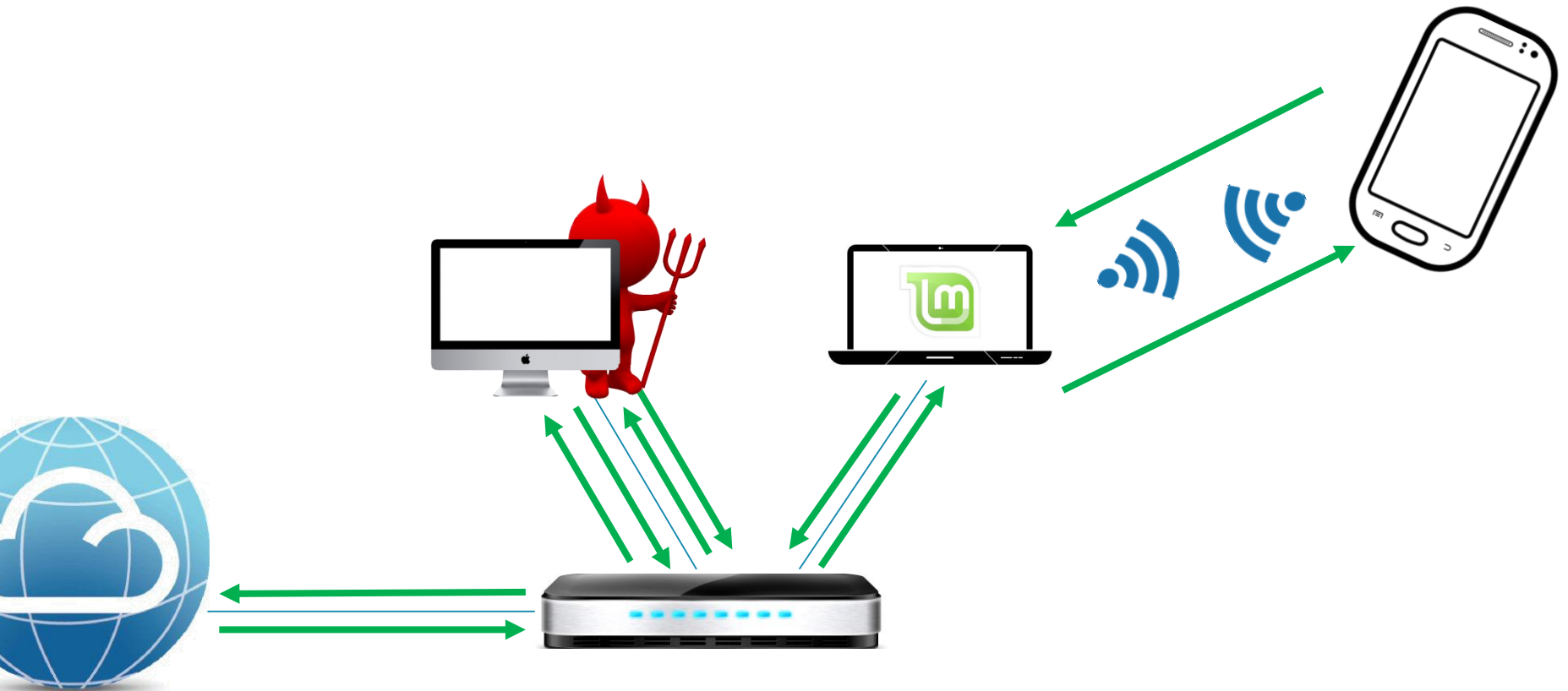
Man-In-The-Middle Attack



Man-In-The-Middle Attack



Man-In-The-Middle Attack



ⓧ ⓪ Ⓜ Ⓛ
Burp Suite Free Edition v1.6.32

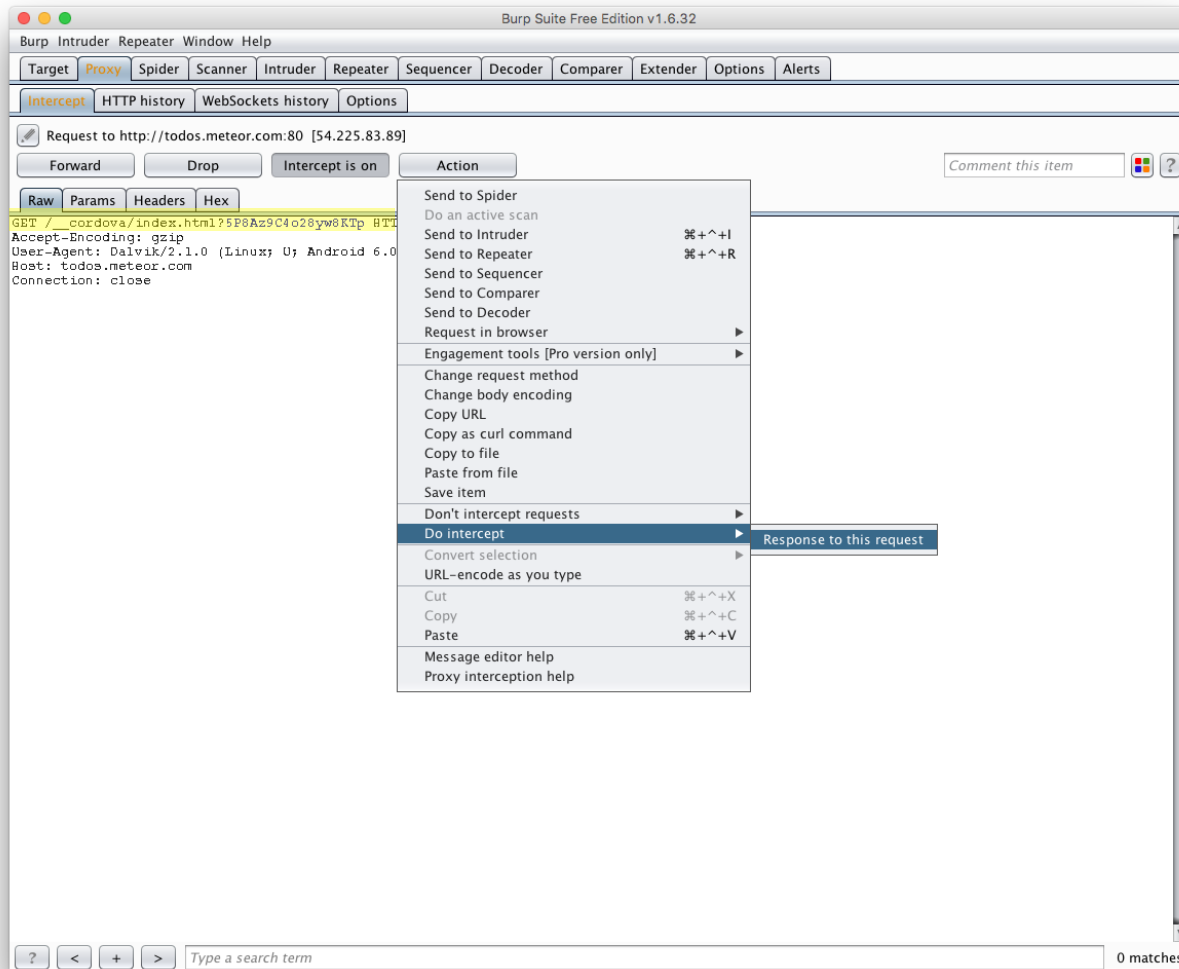
Burp Intruder Repeater Window Help

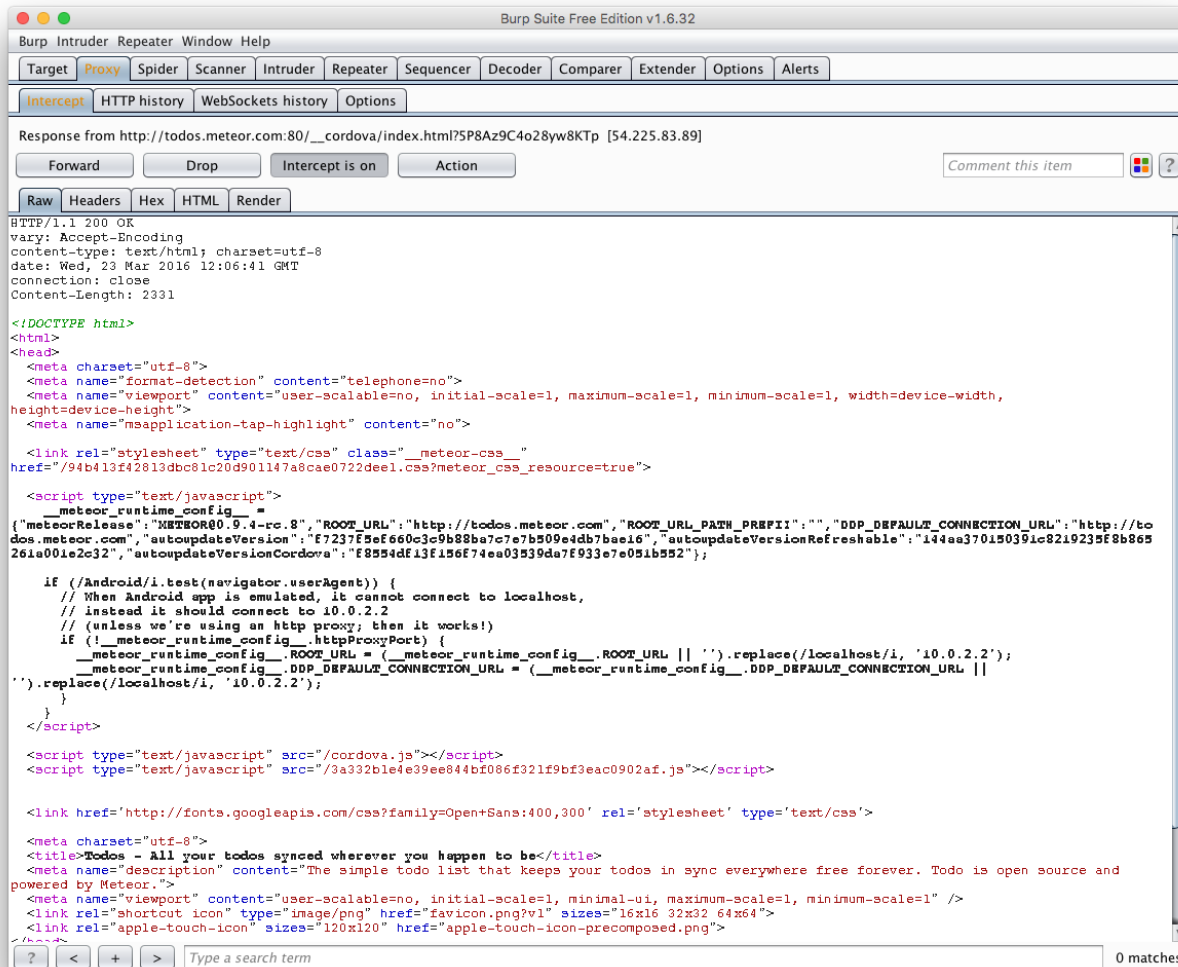
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

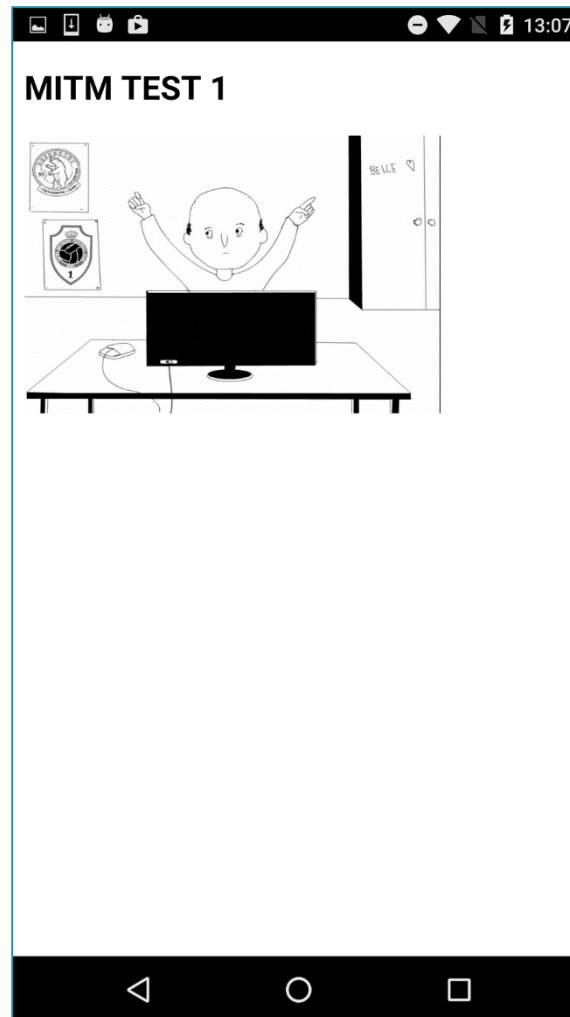
#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title
11	http://todos.meteor.com	POST	/sockjs/439/734kvo6o/xhr_send	<input checked="" type="checkbox"/>	<input type="checkbox"/>	204	291	text		
12	http://todos.meteor.com	POST	/sockjs/439/734kvo6o/xhr	<input type="checkbox"/>	<input type="checkbox"/>	200	21045	JSON		
13	http://todos.meteor.com	GET	/__cordova/manifest.json	<input type="checkbox"/>	<input type="checkbox"/>	200	2036	JSON	json	
14	http://todos.meteor.com	POST	/sockjs/439/734kvo6o/xhr	<input type="checkbox"/>	<input type="checkbox"/>	200	520	JSON		
15	http://todos.meteor.com	POST	/sockjs/439/734kvo6o/xhr_send	<input checked="" type="checkbox"/>	<input type="checkbox"/>	204	291	text		
16	http://todos.meteor.com	GET	/__cordova/img/logo-todos.svg?...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200	4349	XML	svg	
19	http://todos.meteor.com	GET	/__cordova/icon/todos.svg?e2Mu...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200	17696	XML	svg	
22	http://todos.meteor.com	GET	/__cordova/3a332b1e4e39ee844b...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200	427520	script	js	
23	http://todos.meteor.com	GET	/__cordova/index.html?qfHyTzkJa...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200	2491	HTML	html	Todos - All your todo...
26	http://todos.meteor.com	POST	/sockjs/439/734kvo6o/xhr	<input type="checkbox"/>	<input type="checkbox"/>	200	964	JSON		
27	http://todos.meteor.com	POST	/sockjs/439/734kvo6o/xhr	<input type="checkbox"/>	<input type="checkbox"/>	200	339	JSON		
29	http://todos.meteor.com	GET	/sockjs/info?cb=5mbk88fiul	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200	387	JSON		
30	http://fonts.gstatic.com	GET	/s/opensans/v13/cJZKeOuBrn4kE...	<input type="checkbox"/>	<input type="checkbox"/>	200	10752			



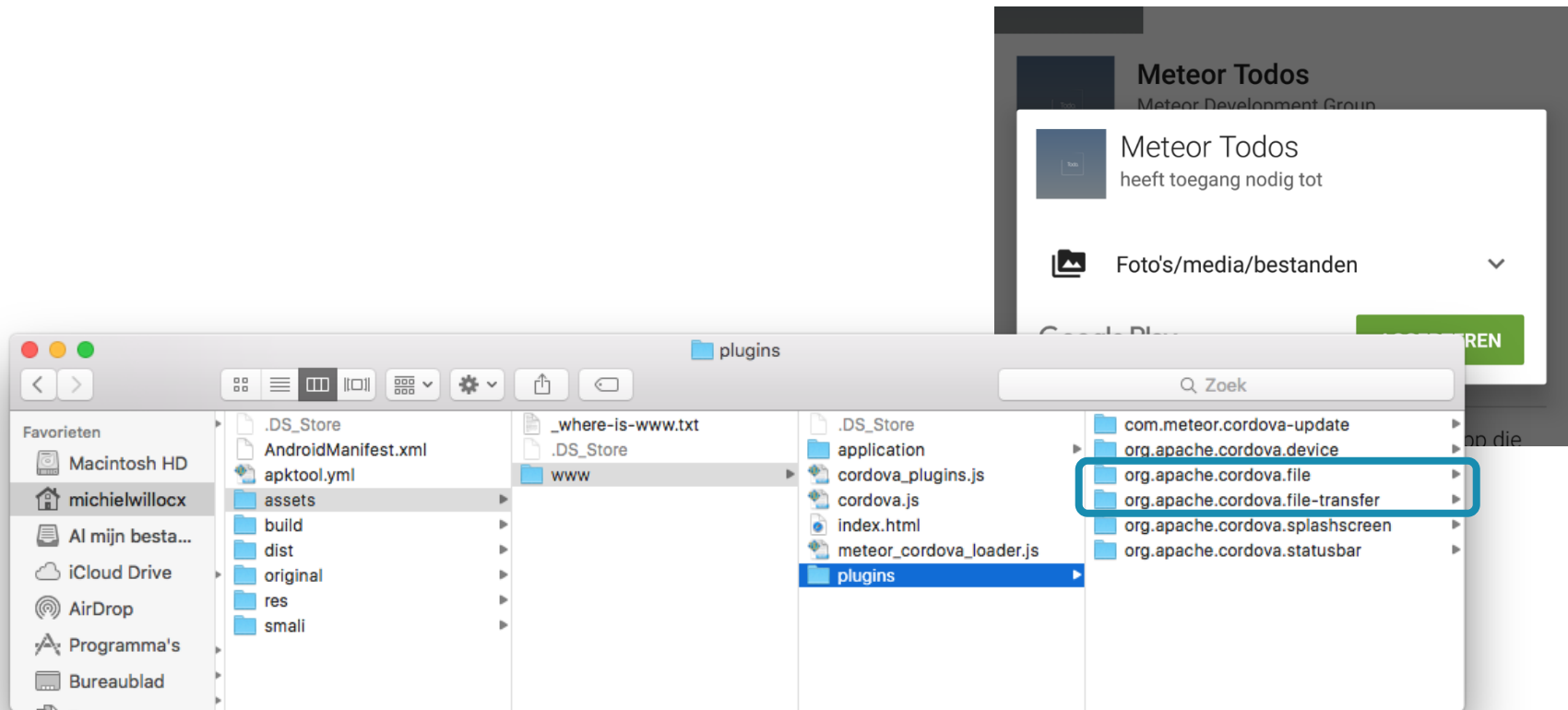


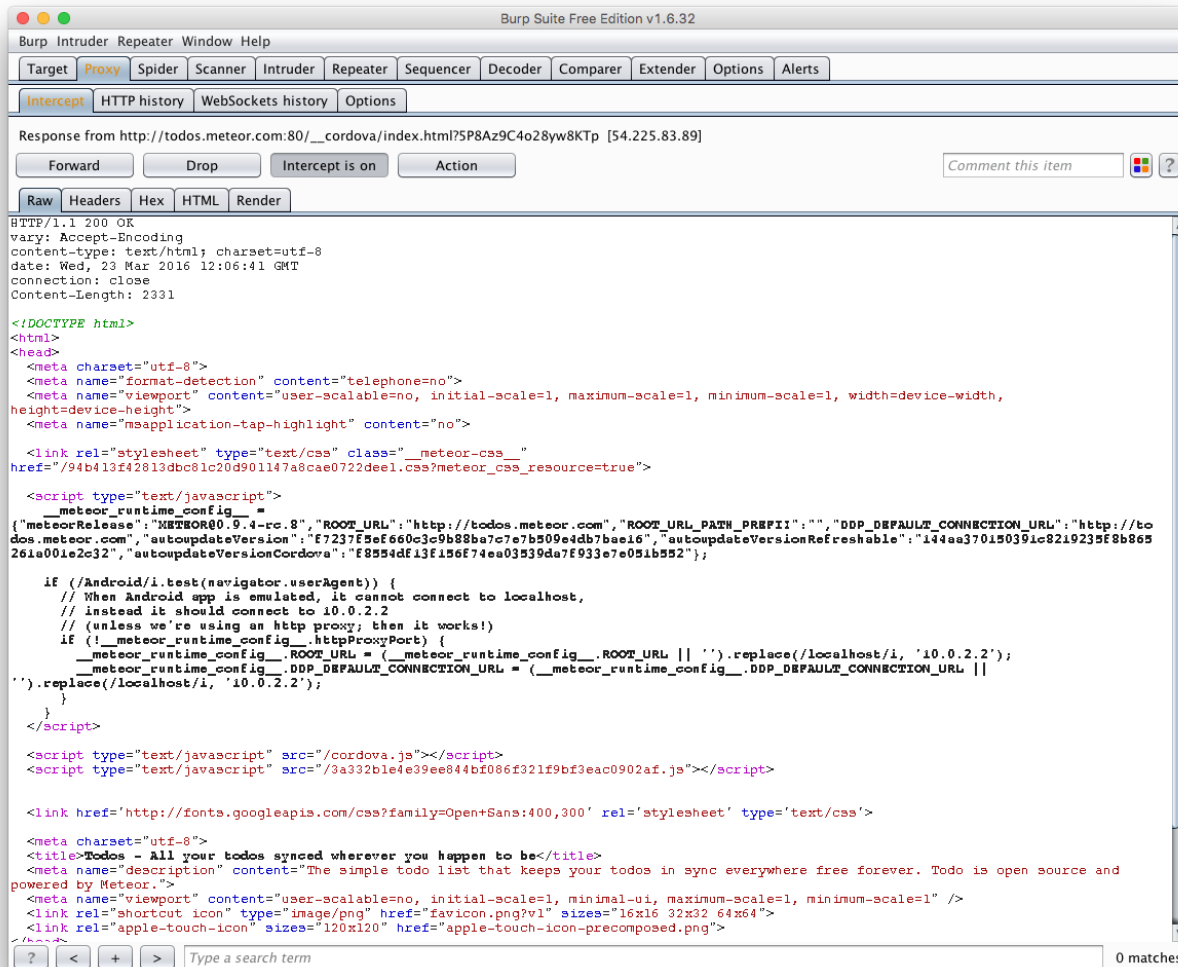
Actual content,
loaded into the
application

Result



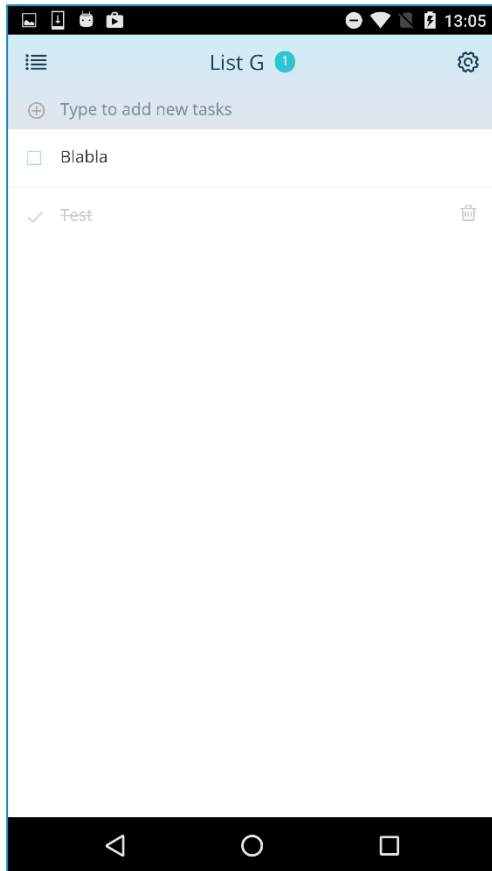
Okay, but what is the point?



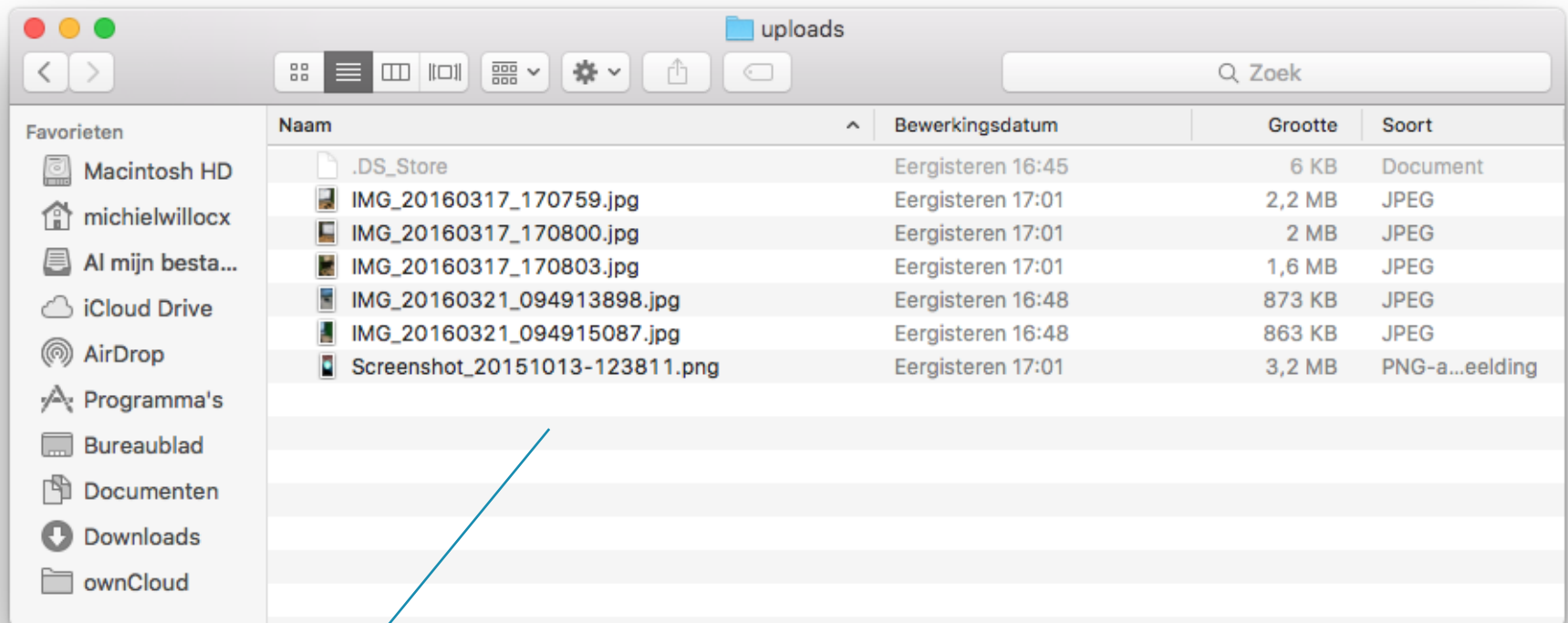


Actual content,
loaded into the
application

Result



App launches as always, nothing special?

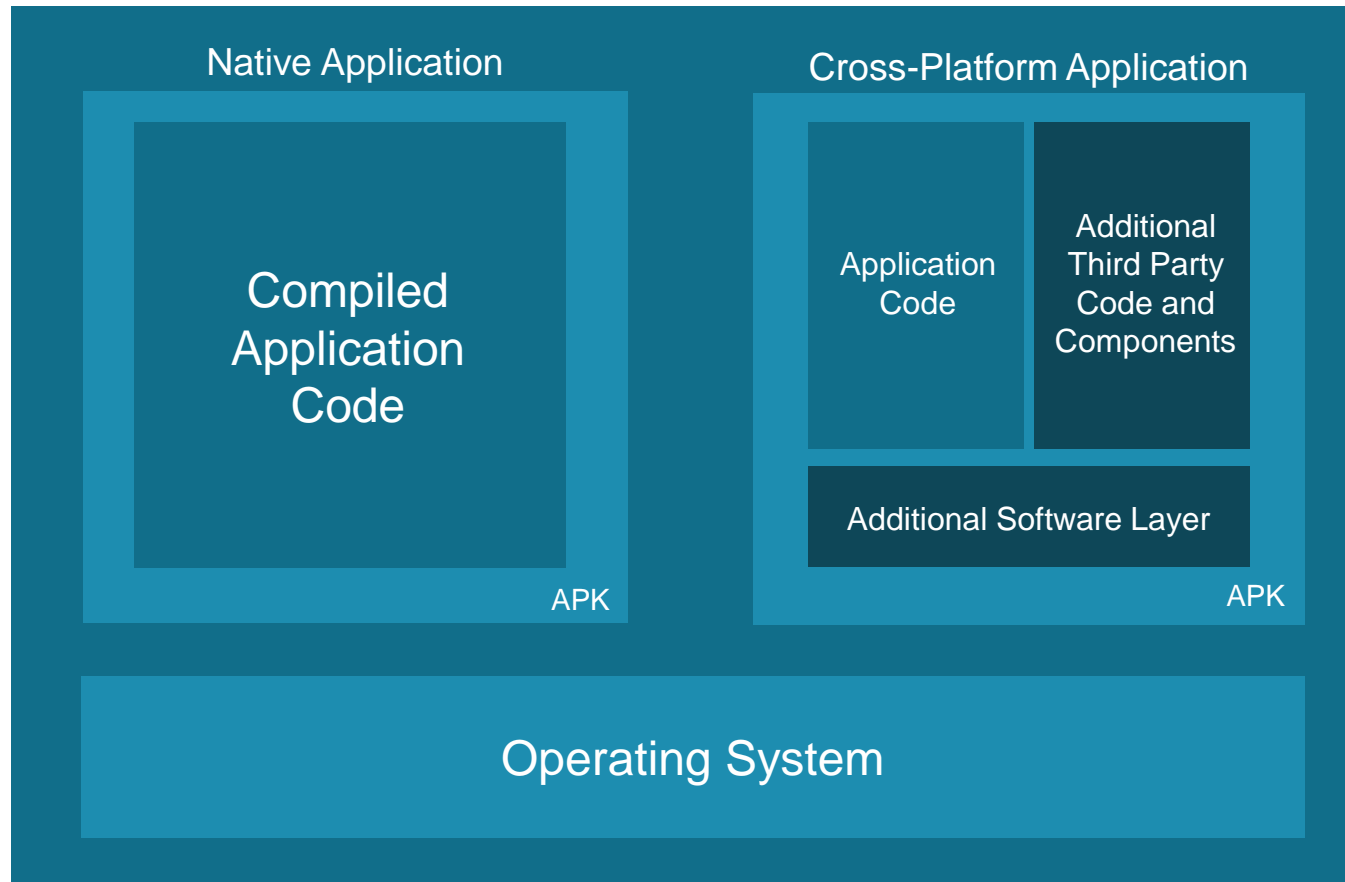


Pictures from the device

How to protect against this?



The impact of bugs, patches and updates



End of part 1

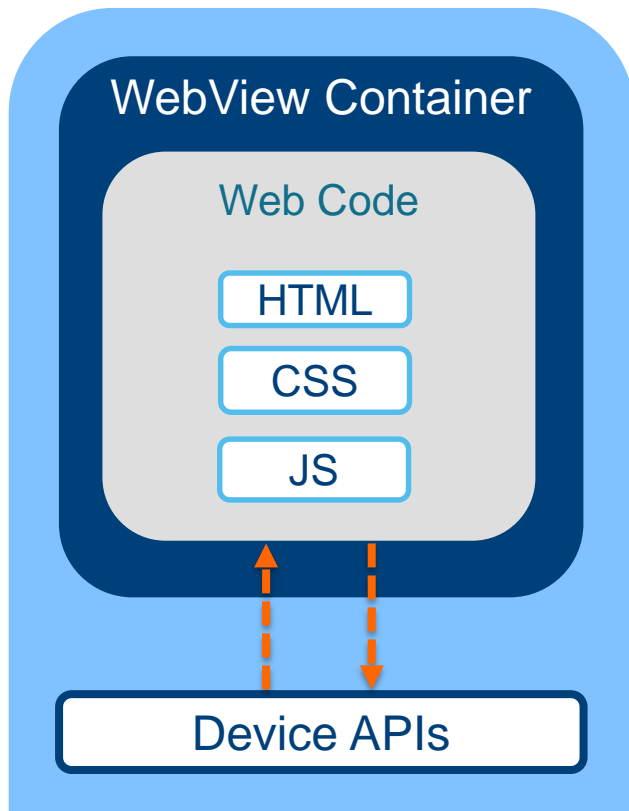


Agenda

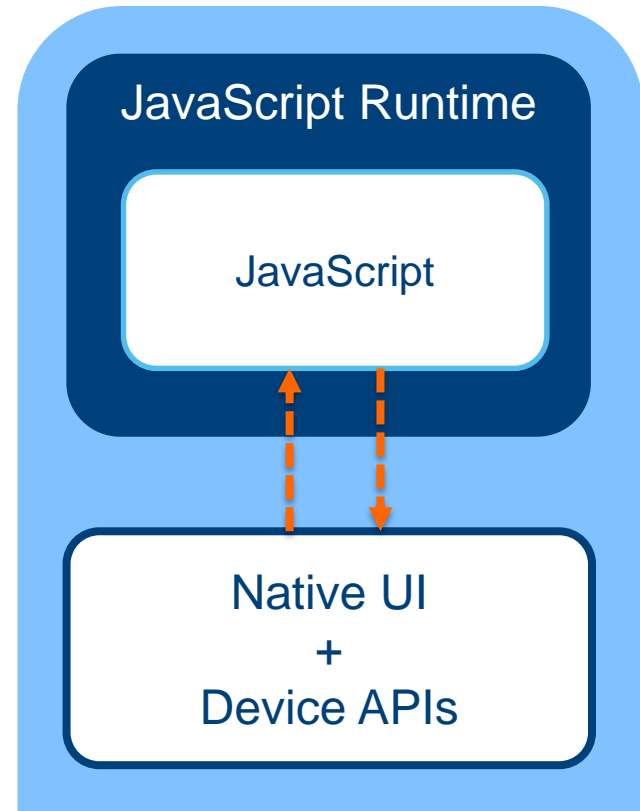
- Hybrid Approaches
- Runtime Based CPT Candidates
- Philosophy of Titanium / React Native / NativeScript
- Why choose what when? And how to use?
 - Architecture
 - Developer Experience
 - Future goals
- Conclusions

Hybrid Approaches

WebView Based

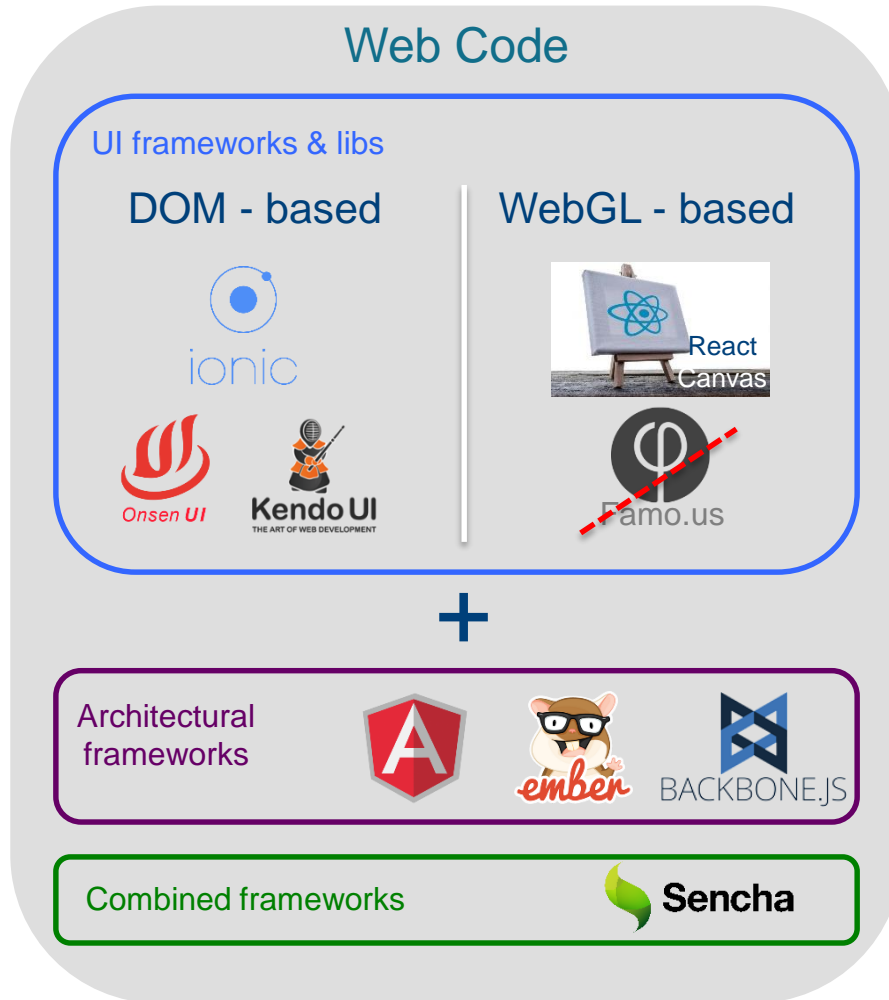


Runtime Based



Hybrid Approaches

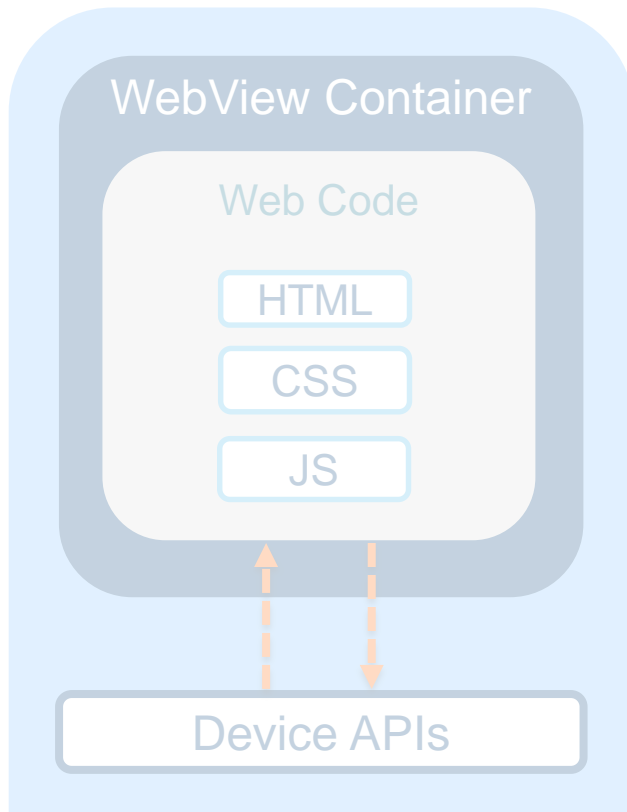
WebView Based



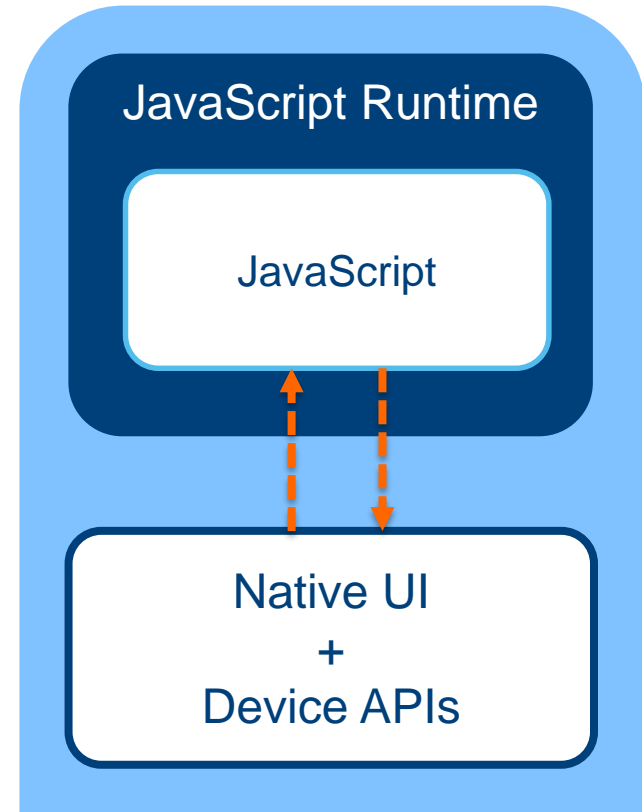
Hybrid Approaches

Runtime Based

WebView Based



Runtime Based



Hybrid Approaches

Runtime Based – The “WHY”...



- Consistent with platform
 - Fast and responsive
 - Complex gestures and smooth animations
-
- No knowledge & code sharing
 - Different technology stacks
 - Slow iteration speeds*
 - Hard to scale

*Android Instant Run

Hybrid Approaches

Runtime Based – The “WHY”...



- HTML / CSS / JavaScript
 - Same code and technologies
 - Frameworks provide scaling
 - F5 / ⌘+R
-
- Very hard to provide smooth experiences
 - Not designed for complex interactions
 - Feel out of place with the platform
 - WebView fragmentation
 - Performance

Hybrid Approaches

Runtime Based – The Solution??



Web advantages

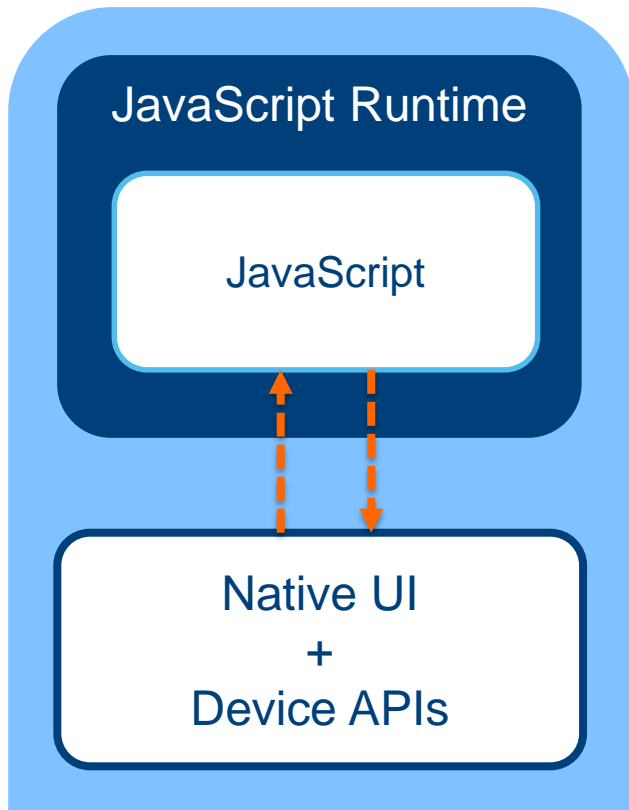
- JSX / XML / CSS / JavaScript
- Same code and technologies
- Frameworks provide scaling
- Live-reload

Native advantages

- Consistent with platform
- Fast and responsive UI
- Complex gestures and smooth 60fps animations

Hybrid Approaches

Runtime Based – The “HOW”




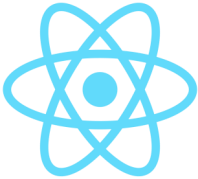




























Composition

- Native side
- JavaScript side
- JS-to-native bridge

Key concepts

- Proxy objects
- Asynchronous calls

Runtime Based CPT Candidates

	 Titanium	 React Native	 Nativescript	 Fusetools	 TabrisJS	 Smartface
Announced	2008	2015	2014	2015	2014	2011
Version	V5.2.2	V0.25.1	V2.0	V0.12.4	V1.7	v4.5.0
Platforms	Android 4.0.x – 6.0.x iOS 7.1.x – 9.2.x WP8.1-UWP	Android 4.1.x – 6.0.x iOS 7.0.x – 9.2.x UWP (alpha)	Android 4.2.x – 6.0.x iOS 7.1.x – 9.2.x UWP (alpha)	Android 4.2.x – 6.0.x iOS 7.1.x – 9.2.x	Android 3.7.x - 5.x iOS 6.x - 8.x	Android 4.2.x – 6.0.x iOS 7.1.x – 9.2.x
Popularity	 11658  23564  2063  2303	 556  72340  30206  3439	 2300  7142  6393  259	 5871  16028  /  1119	 634  252  384  2	 63  1105  /  283

Popularity numbers are checked on 04/2016

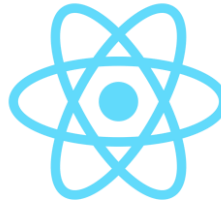
Agenda

- Hybrid Approaches
- Runtime Based CPT Candidates
- **Philosophy of Titanium / React Native / NativeScript**
- Why choose what when? And how to use?
 - Architecture
 - Developer Experience
 - Future goals
- Conclusions

Philosophy of Titanium/React Native/NativeScript



*“Write once,
adapt everywhere”*



*“Learn once,
write anywhere”*



*“Write once,
run anywhere”*

UI \neq shared
Logic = shared



Up to 70,80,90%
Code share



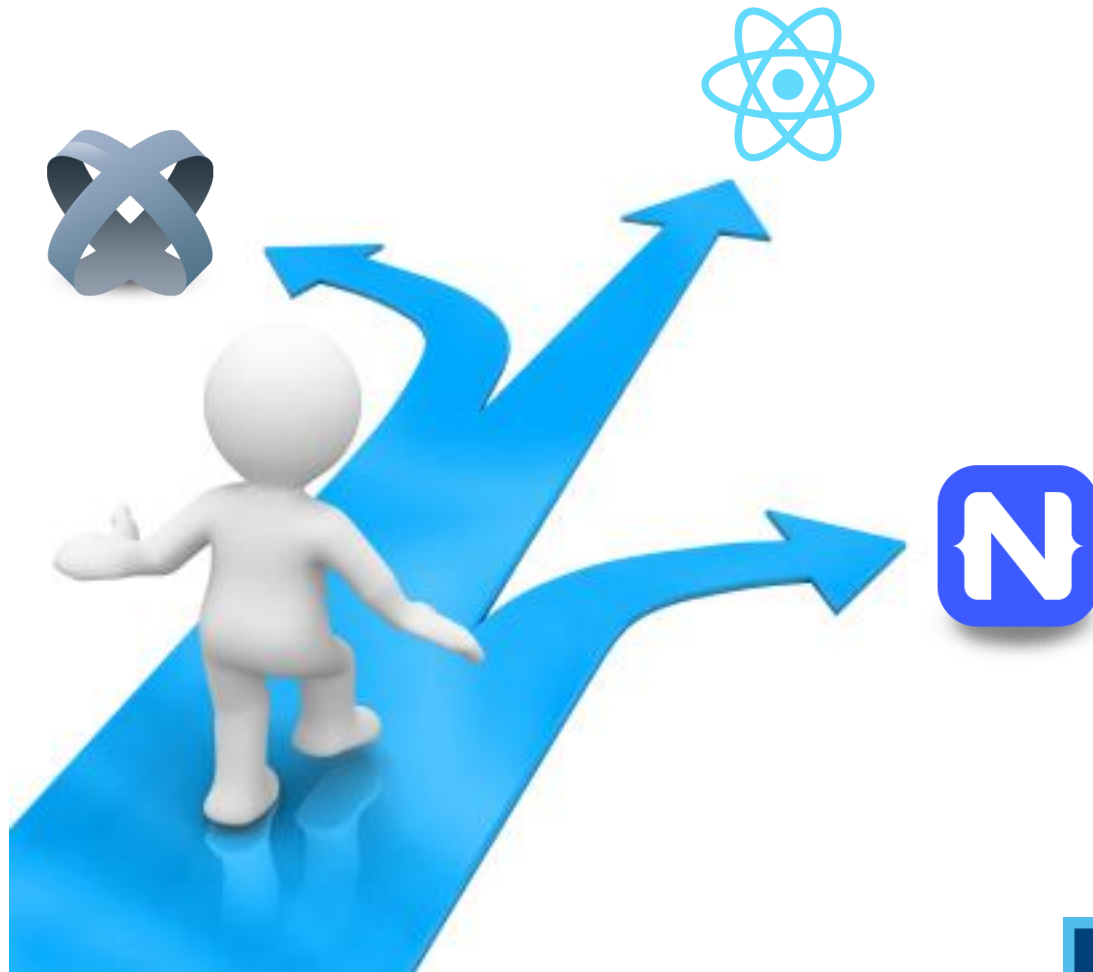
Up to 85% Code share
Facebook’s Ads manager

UI = shared
Logic = shared



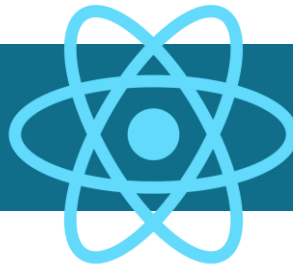
Possibility of writing
platform specific UI

Why choose what when? And how to use?





VS



VS



Application structure

Used technologies

Runtime Architecture

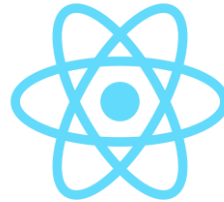
Modularity



Architecture

Comparing Ti vs RN vs {N}

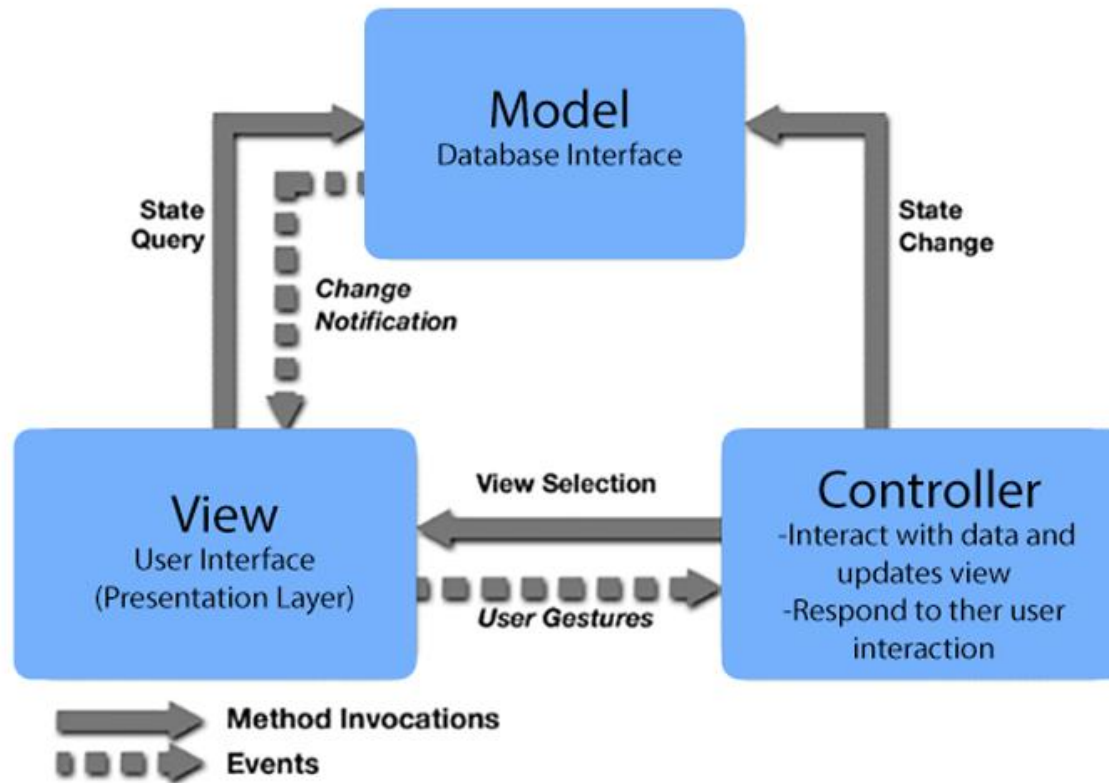
Architecture – Application Structure



Design Pattern		
<ul style="list-style-type: none">MVC (using Alloy)		
Additional concepts		

Comparing **Ti** vs RN vs {N}

Architecture – MVC pattern



Model:
*uses BackboneJS
models and collections*

View:
*represent a page in
your application*

Controller:
*each view can have
a controller*

Comparing Ti vs RN vs {N}

Architecture – Application Structure



+

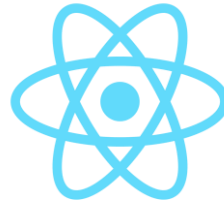


```
1  app/
2  | - assets/      (images, etc.)
3  | - controllers/ (controllers for the views)
4  |   - android/
5  |     -index.js
6  |     index.js
7  | - models/      (models)
8  | - views/       (views)
9  |   - ios/
10 |     -index.xml
11 |     index.xml
12 | - styles/      (view styling)
13 | - themes/      (customizes assets and styles)
14 | - widgets/     (app-like directory structure for each widget)
15 | - migrations/  (database migration)
16 | - lib/         (JavaScript libraries)
17 | - specs/       (same as lib/ but for development only)
18 | - i18n/        (Language strings)
19 | - platform/    (platform resources)
20 | - alloy.jmk     (build config)
21 | - alloy.js      (init file for preconfig)
22 | - config.json   (project config)
23 Resources/
24 i18n/
```

1. Definitive application structure
2. Platform separation on folder level

Comparing Ti vs RN vs {N}

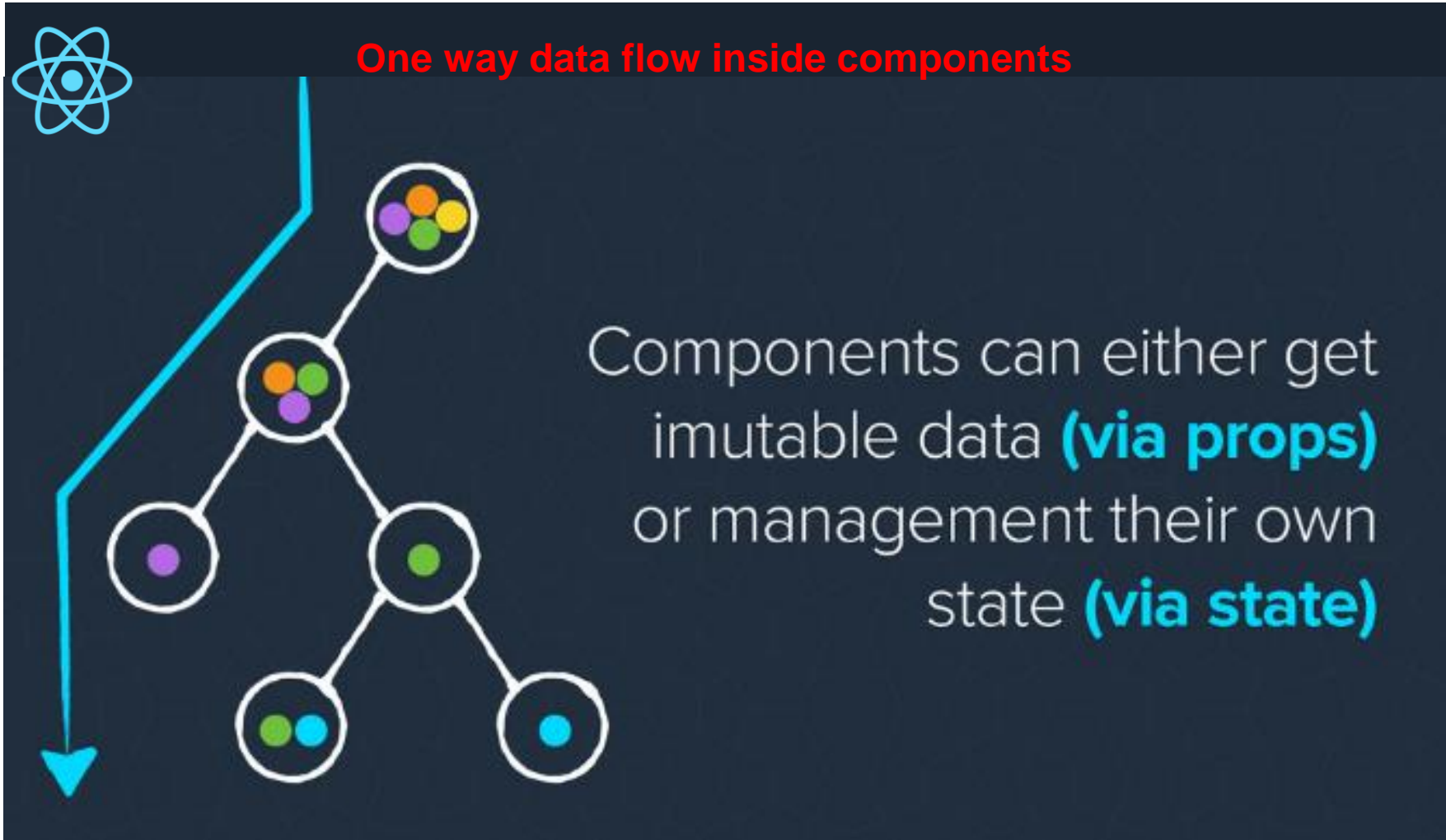
Architecture – Application Structure



Design Pattern		
<ul style="list-style-type: none">MVC (using Alloy)	<ul style="list-style-type: none">Flux → Redux (unidirectional data flow)	
Additional concepts		
	<ul style="list-style-type: none">React Components	

Comparing Ti vs **RN** vs {N}

Architecture – React Components

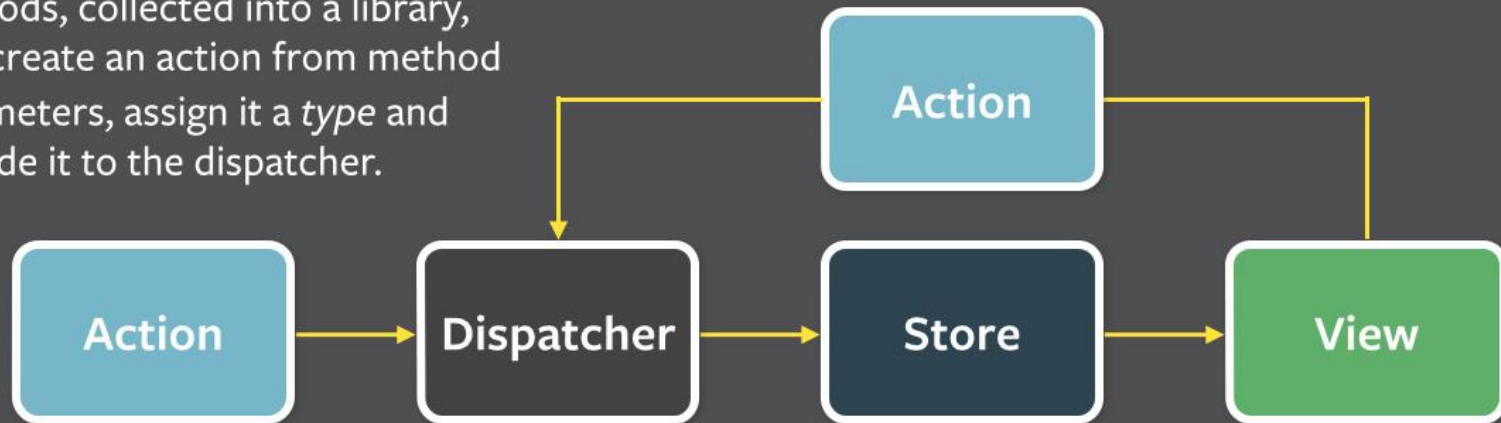


Comparing Ti vs RN vs {N} Architecture – Flux pattern



One way data flow inside the application

Action creators are helper methods, collected into a library, that create an action from method parameters, assign it a *type* and provide it to the dispatcher.



Every action is sent to all stores via the *callbacks* the stores register with the dispatcher.

After stores update themselves in response to an action, they emit a *change* event.

Special views called *controller-views*, listen for *change* events, retrieve the new data from the stores and provide the new data to the entire tree of their child views.

Comparing Ti vs RN vs {N}

Architecture – Application Structure



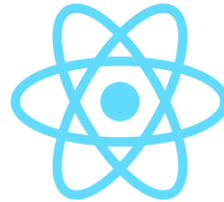
1	App/	
2	- actions/	(flux actions)
3	- components/	(react components)
4	- BigButton.ios.js	
5	- BigButton.android.js	
6	- dispatchers/	(flux dispatchers)
7	- stores/	(flux stores)
8	- utils/	(utilities)
9	- constants/	(global constants)
10	- mixins/	(share common functionality inside components)
11	ios/	(ios resources, builds, config, etc...)
12	- ...	
13	android/	(android resources, builds, config, etc...)
14	- ...	
15	node_modules/	(external libraries)
16	index.ios.js	(root component ios)
17	index.android.js	(root component android)
18	package.json	(project configuration)

1. User defined application structure (Scaffolding available)

2. Platform separation on file level

Comparing Ti vs RN vs {N}

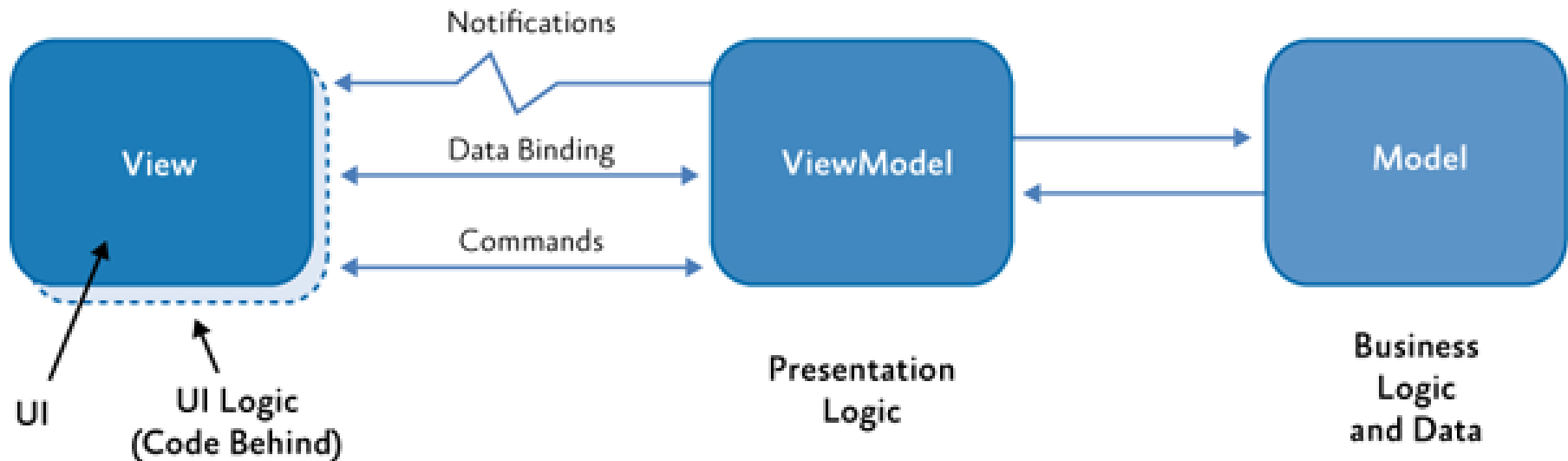
Architecture – Application Structure



Design Pattern		
<ul style="list-style-type: none">MVC (using Alloy)	<ul style="list-style-type: none">Flux → Redux (unidirectional data flow)	<ul style="list-style-type: none">MVC or MVVM
Additional concepts		
	<ul style="list-style-type: none">React Components	

Comparing Ti vs RN vs {N}

Architecture – MVVM pattern



- Two-way data binding
 - Observables

Comparing Ti vs RN vs {N}

Architecture – Application Structure

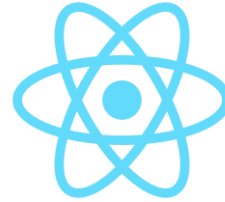


```
1 Hello-World
2   - app/
3     - App_Resources/      (Assets, manifests, plist, ...)
4       - Android/
5       - ios/
6     - fonts/              (icon fonts)
7     - shared/             (services, utility, ...)
8     - ...
9     - views/              (views)
10    - main/
11      - main-page.css      (view related styling)
12      - main-page.js       (compiled from typescript)
13      - main-page.ts       (code behind view)
14      - main-page.xml      (page mark-up)
15      - main-page.ios.xml  (platform specific mark-up)
16      - ...
17    - widgets/             (custom widgets)
18    - ...
19    - app.css               (global styling, theming)
20    - app.js                (starting point, app lifecycle)
21    - package.json          (NativeScript configuration)
22  - node_modules/           (linked libs)
23  - platforms/              (platform specific files, build, ...)
24  - package.json            (app's configuration)
```

1. User defined application structure (Scaffolding available)
2. Platform separation on file level

Comparing Ti vs RN vs {N}

Architecture – Application Structure Rating

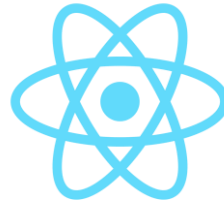


	MVC	Flux → Redux	MVC or MVVM
Entry Level			
Scaling			
Testability			

Rating

Comparing Ti vs RN vs {N}

Architecture – Used Technologies



Programming		
<ul style="list-style-type: none">• JavaScript (ES5)• XML mark-up• TSS styling (CSS-like styling)		
Layout system		
<ul style="list-style-type: none">• Relative to parent view		

Comparing Ti vs RN vs {N}

Architecture – Used Technologies Examples



views/index.xml

```
1 <Alloy>
2   <Window class="container">
3     <Label id="label" onClick="doClick">
4       Hello, World
5     </Label>
6   </Window>
7 </Alloy>
```

controllers/index.js

```
1 function doClick(e) {
2   alert($.label.text);
3 }
4
5 $.index.open();
```

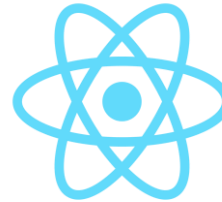
styles/index.tss

```
1 ".container": {
2   backgroundColor:"white"
3 }
4
5 "Label": {
6   width: Ti.UI.SIZE,
7   height: Ti.UI.SIZE,
8   color: "#000"
9 }
```

- Per view styling
- Global styling in “themes”

Comparing Ti vs RN vs {N}

Architecture – Used Technologies



Programming		
<ul style="list-style-type: none">JavaScript (ES5)XML mark-upTSS styling (CSS-like styling)	<ul style="list-style-type: none">JavaScript (ES6+ES7 using Babel transpiler)JSX (XML like mark-up language)JavaScript “inline” stylesFlow (static type checker for JavaScript)	
Layout system		
<ul style="list-style-type: none">Relative to parent view	<ul style="list-style-type: none">Flexbox	

Comparing Ti vs **RN** vs {N}

Architecture – Used Technologies Examples



index.ios.js

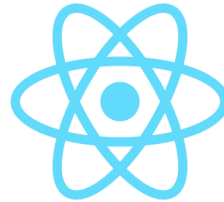
```
8  class HelloWorldApp extends React.Component {
9    renderText: function() {
10      return (
11        <Text style={styles.baseText}>
12          Hello World
13        </Text>
14      );
15    },
16  }
```

```
1  var styles = React.StyleSheet.create({
2    baseText: {
3      color: 'black',
4      backgroundColor: 'white',
5    }
6  });
```

- Everything is a React component
- Mixing XML-like mark-up inside JavaScript
- Inline styles → referenced inside the same file

Comparing Ti vs RN vs {N}

Architecture – Used Technologies



Programming		
<ul style="list-style-type: none">JavaScript (ES5)XML mark-upTSS styling (CSS-like styling)	<ul style="list-style-type: none">JavaScript (ES6+ES7 using Babel transpiler)JSX (XML like mark-up language)JavaScript “inline” stylesFlow (static type checker for JavaScript)	<ul style="list-style-type: none">JavaScript (ES5) or Typescript (ES6+ES7)XML mark-upCSS styling
Layout system		
<ul style="list-style-type: none">Relative to parent view	<ul style="list-style-type: none">Flexbox	<ul style="list-style-type: none">Native layout sys.

Comparing Ti vs RN vs {N}

Architecture – Used Technologies Examples



View/home/home.xml

```
1 <Page navigatingTo="onNavigatingTo">
2   <StackLayout>
3     <Label text="Tap the button" class="title"/>
4     <Button text="TAP" tap="{{ onTap }}" />
5     <Label text="{{ message }}" class="message"/>
6   </StackLayout>
7 </Page>
```

View/home/home.js

```
1 var viewModel = require("../main-view-model");
2
3 function onNavigatingTo(args) {
4   var page = args.object;
5   page.bindingContext = viewModel.createViewModel();
6 }
7 exports.onNavigatingTo = onNavigatingTo;
```

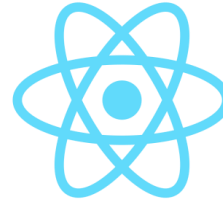
View/home/home.css

```
1 .title {
2   font-size: 30;
3   horizontal-align: center;
4   margin: 20;
5 }
6
7 button {
8   font-size: 42;
9   horizontal-align: center;
10  color: red
11 }
```

- Per view styling
- Global styling in “app.css”

Comparing Ti vs RN vs {N}

Architecture – Used Technologies Rating



ES5, XML, TSS

ES6/7, JSX, JS-styles

TypeScript, XML, CSS

Web compliance



**Strong typed
entry level**



Layout

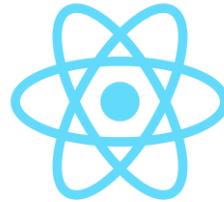


Rating



Comparing Ti vs RN vs {N}

Architecture – Runtime Architecture

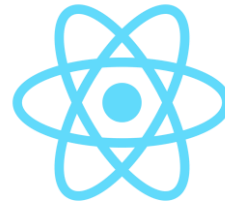















Properties		
<ul style="list-style-type: none">• 2 Threads<ul style="list-style-type: none">• Main UI thread• JS thread• Asynchronous• Serializable (iOS)• Spawn native threads for work offloading	<ul style="list-style-type: none">• 3 Threads<ul style="list-style-type: none">• Shadow thread• Main UI thread• JS thread• Asynchronous• Batched native calls• Serializable	<ul style="list-style-type: none">• 1 Thread<ul style="list-style-type: none">• Main UI thread• Asynchronous• Direct native API access

Note: every bridge has a type conversion mechanism

Comparing Ti vs RN vs {N}

Architecture – Runtime Architecture Rating

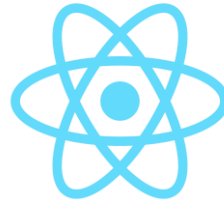


	Titanium Runtime	React Native Runtime	NativeScript Runtime
Bridge performance			
Size overhead			
Flexibility			
Used runtime iOS , Android	  		 

Rating     

Comparing Ti vs RN vs {N}

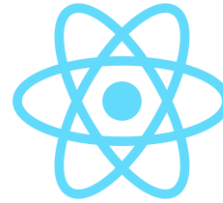
Architecture – Modularity












Options		
<ul style="list-style-type: none"> • Titanium modules <ul style="list-style-type: none"> • Built-in (100+) • gitTrio community modules and widgets (1351) + (60) • NPM JS libraries through titaniumifier (no DOM reliance) <div data-bbox="131 786 629 982" style="border: 2px solid red; padding: 5px; transform: rotate(-5deg); color: red; font-weight: bold;">OWN ECOSYSTEM</div>	<ul style="list-style-type: none"> • React Native modules <ul style="list-style-type: none"> • Built-in (60+) • JS Coach (1000+) • Cordova plugin integration • CocoaPods/JAR integration • NPM JS libraries (no DOM reliance) <div data-bbox="981 772 1572 982" style="border: 2px solid red; padding: 5px; transform: rotate(-5deg); color: red; font-weight: bold;">INTEGRATION WITH ECOSYSTEMS</div>	<ul style="list-style-type: none"> • NativeScript modules <ul style="list-style-type: none"> • Built-in (54) • Community marketplace (198) • Marketplace (13) • Telerik UI for NativeScript (4) • CocoaPods/ Java JAR integration • NPM JS libraries (no DOM reliance)

Comparing Ti vs RN vs {N}

Architecture – Modularity Rating



	Ti Modules	React Native components, etc...	NS modules, etc...
Module utilization ease			
Module Development complexity			
3 rd party library integration			

Rating     

Comparing Ti vs RN vs {N}

Architecture – Modularity



```
var fileModule = require( "file-system" );  
new fileModule.File( path );
```

At runtime executed on the JS engine



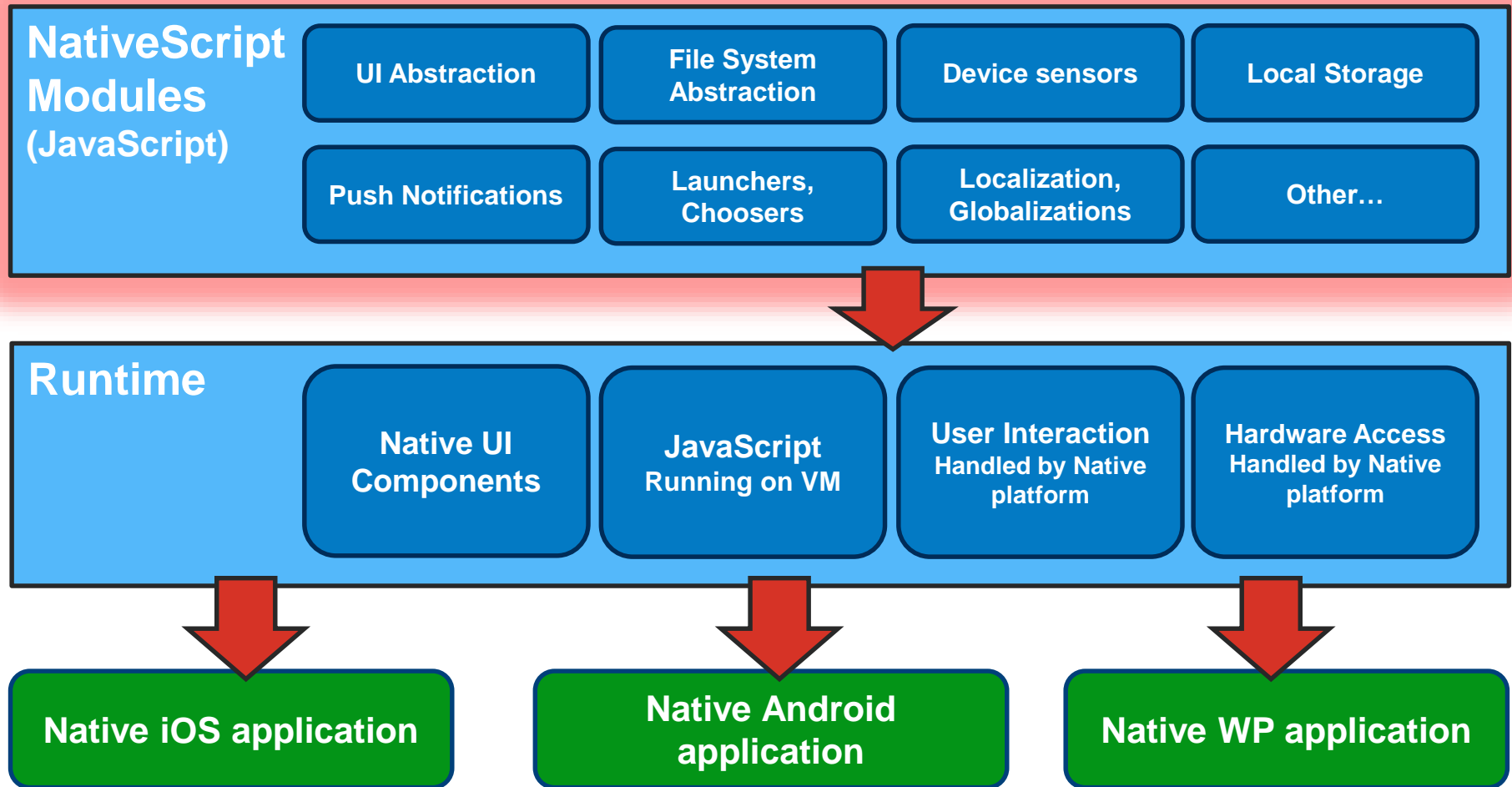
```
new java.io.File( path );
```



```
NSFileManager defaultManager();  
fileManager.createFileAtPathContentsAttributes( path );
```

Comparing Ti vs RN vs {N}

Architecture – Modularity

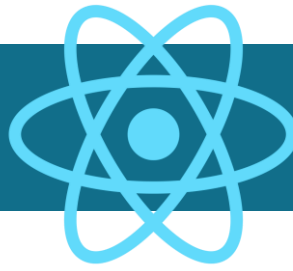


Agenda

- Hybrid Approaches
- Runtime Based CPT Candidates
- Philosophy of Titanium / React Native / NativeScript
- Why choose what when? And how to use?
 - Architecture
 - **Developer Experience**
 - Future goals
- Conclusions



VS



VS



Available tools

Support

Documentation

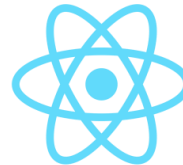
Contribution ease



Developer Experience

Comparing Ti vs RN vs {N}

Developer Experience – Tools

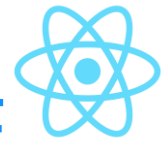


IDE		
<ul style="list-style-type: none"> Appcelerator Studio (based on Eclipse), (paid) 	<ul style="list-style-type: none"> Any IDE* Nuclide Atom (free) Deco IDE (paid?) Visual studio code extension (free) 	<ul style="list-style-type: none"> Any IDE Visual studio code extension (free) Telerik Platform (paid)
Build/Deploy/Debug		
<ul style="list-style-type: none"> CLI tool (paid) <ul style="list-style-type: none"> LiveView Unit test Etc.. 	<ul style="list-style-type: none"> CLI tool <ul style="list-style-type: none"> Live reload Performance debug Etc.. 	<ul style="list-style-type: none"> CLI tool <ul style="list-style-type: none"> Live sync Node-inspector debug Etc..
OverTheAir JS updates		
<ul style="list-style-type: none"> Possibility 	<ul style="list-style-type: none"> Siphon, Codepush,... 	<ul style="list-style-type: none"> Possibility

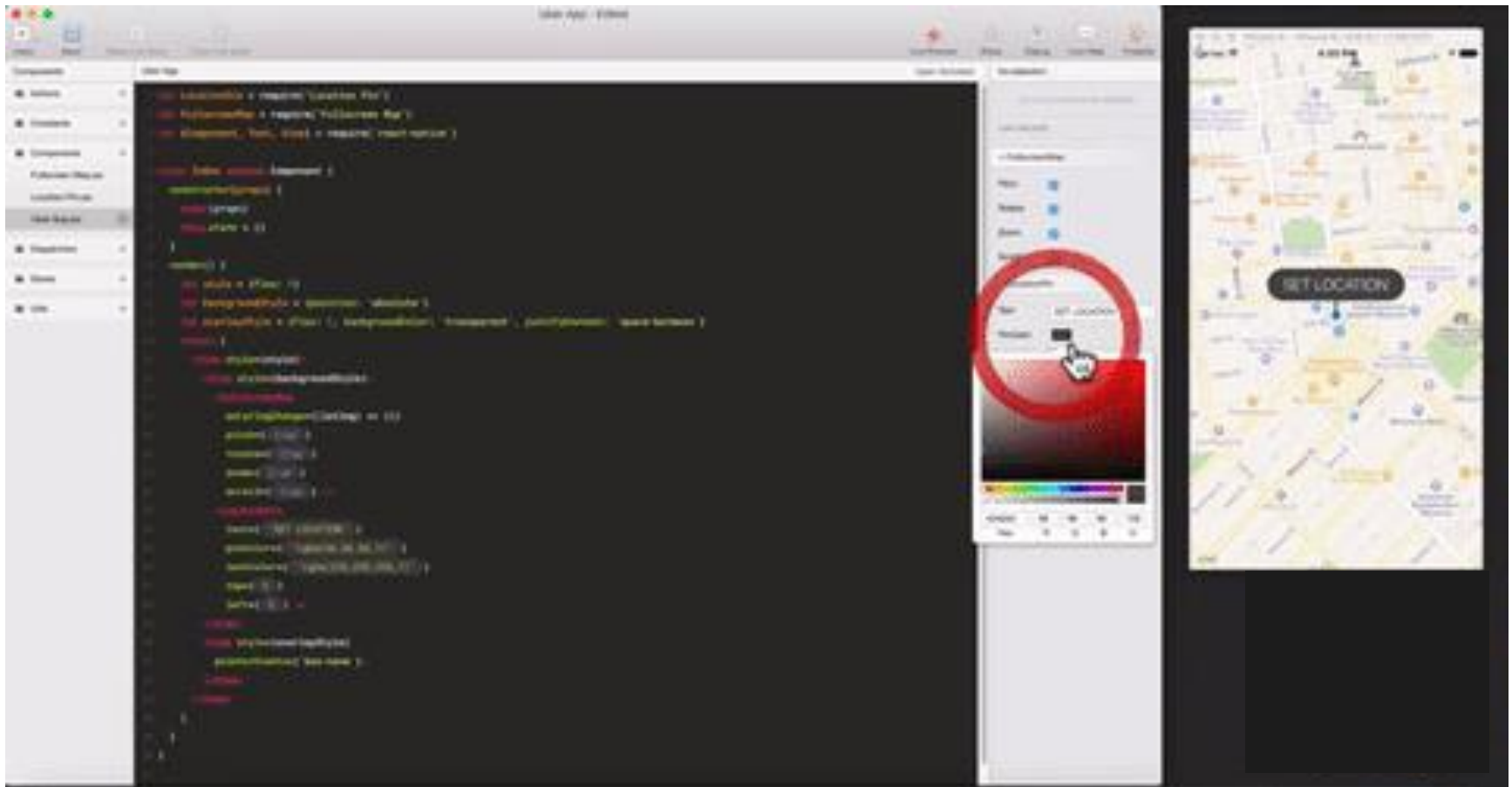
* With JSX support

Comparing Ti vs **RN** vs {N}

Developer Experience – Hot-reload IDE

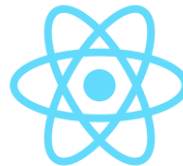


Deco IDE



Comparing Ti vs RN vs {N}

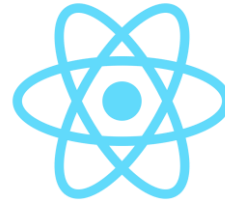
Developer Experience – Community



Support		
<ul style="list-style-type: none">• Official Docs• Slack channel• JIRA Tickets• Stack Overflow• Developer Blogs• Paid support	<ul style="list-style-type: none">• Official Docs• Reactiflux (channel)• Product Pains• Stack Overflow• Developer Blogs	<ul style="list-style-type: none">• Official Docs• Slack channel• Github issues• Stack Overflow• Developer Blogs
News		
<ul style="list-style-type: none">• Official Blog• Twitter	<ul style="list-style-type: none">• Official Blog• Twitter• React News letter	<ul style="list-style-type: none">• Official Blogs• Twitter
Contribution		
<ul style="list-style-type: none">• Github (160)• JIRA Tickets	<ul style="list-style-type: none">• Github (704)• Product Pains	<ul style="list-style-type: none">• Github (51)• NativeScript Ideas

Comparing Ti vs RN vs {N}

Developer Experience – Tools/Support Rating

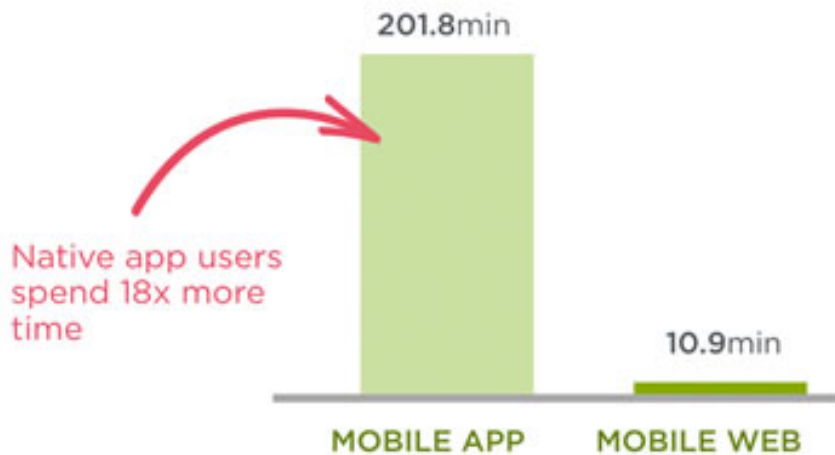


			
Available Tools			
Live Cycle Support			
Community			
Support			
Maturity			

Rating     

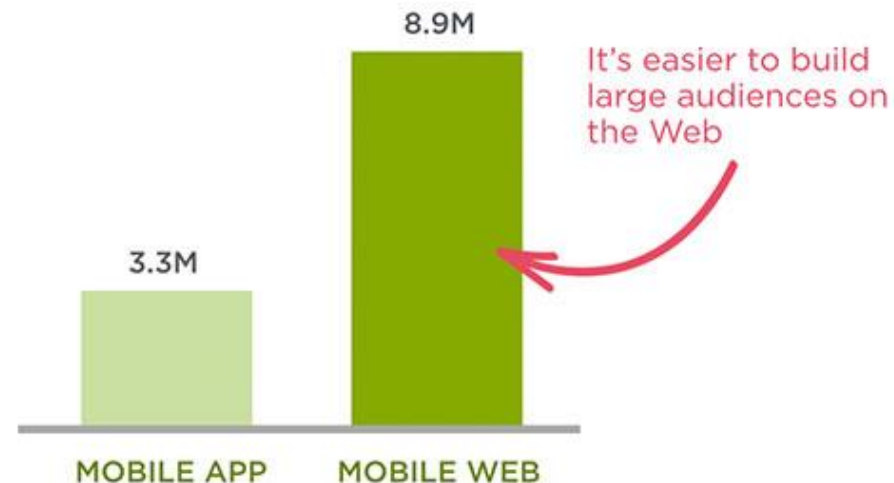
Future Goals

AVG. MONTHLY MINUTES PER VISITOR



comScore Mobile Metrix, U.S., Age 18+, June 2015

MONTHLY UNIQUE VISITORS

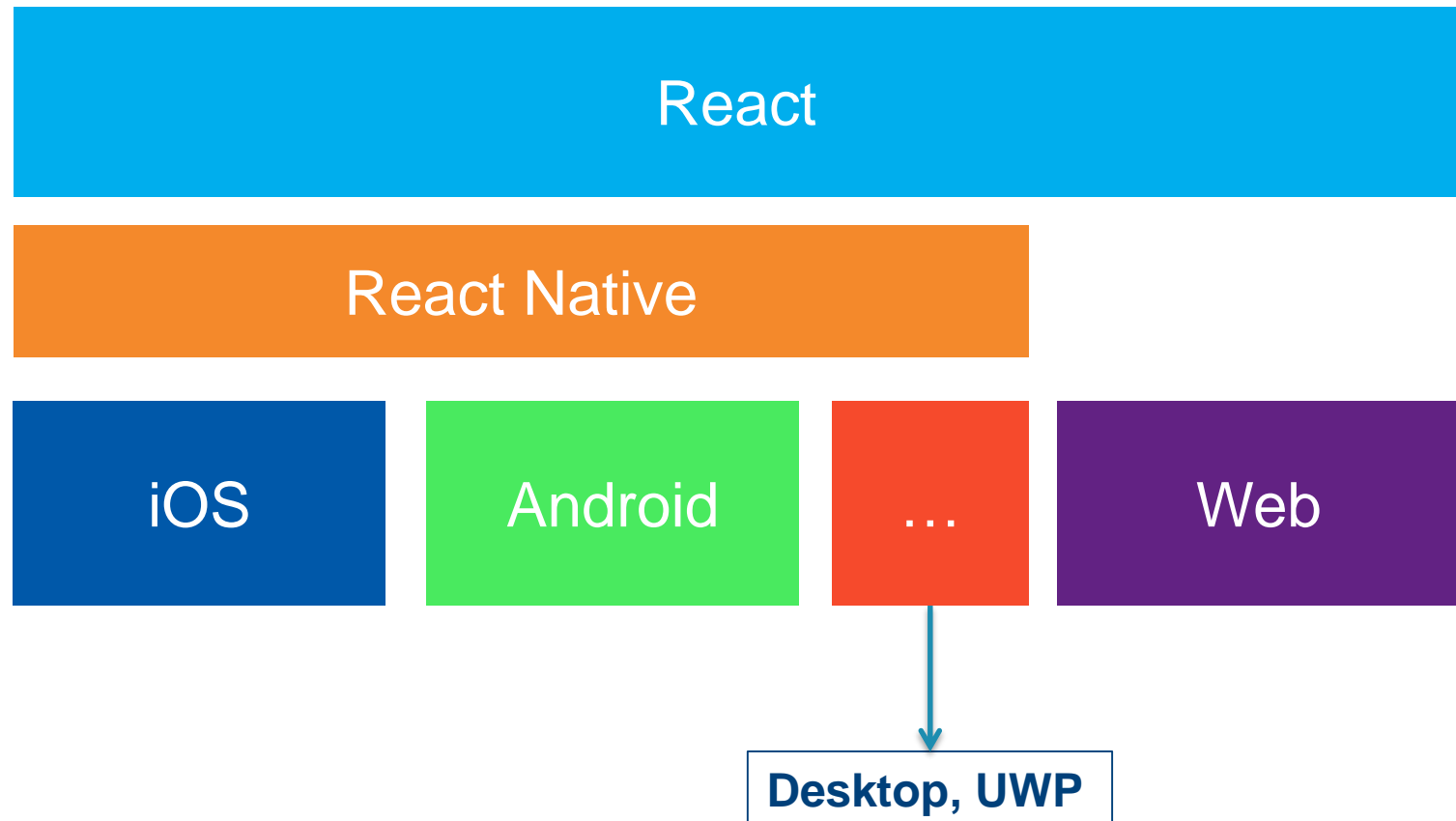
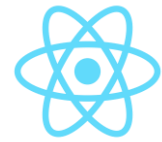


comScore Mobile Metrix, U.S., Age 18+, June 2015

“The Web is for audience reach and native apps are for rich experiences. Both are strategic. Both are valuable. So when it comes to mobile, it’s not Web vs. Native. It’s both.”

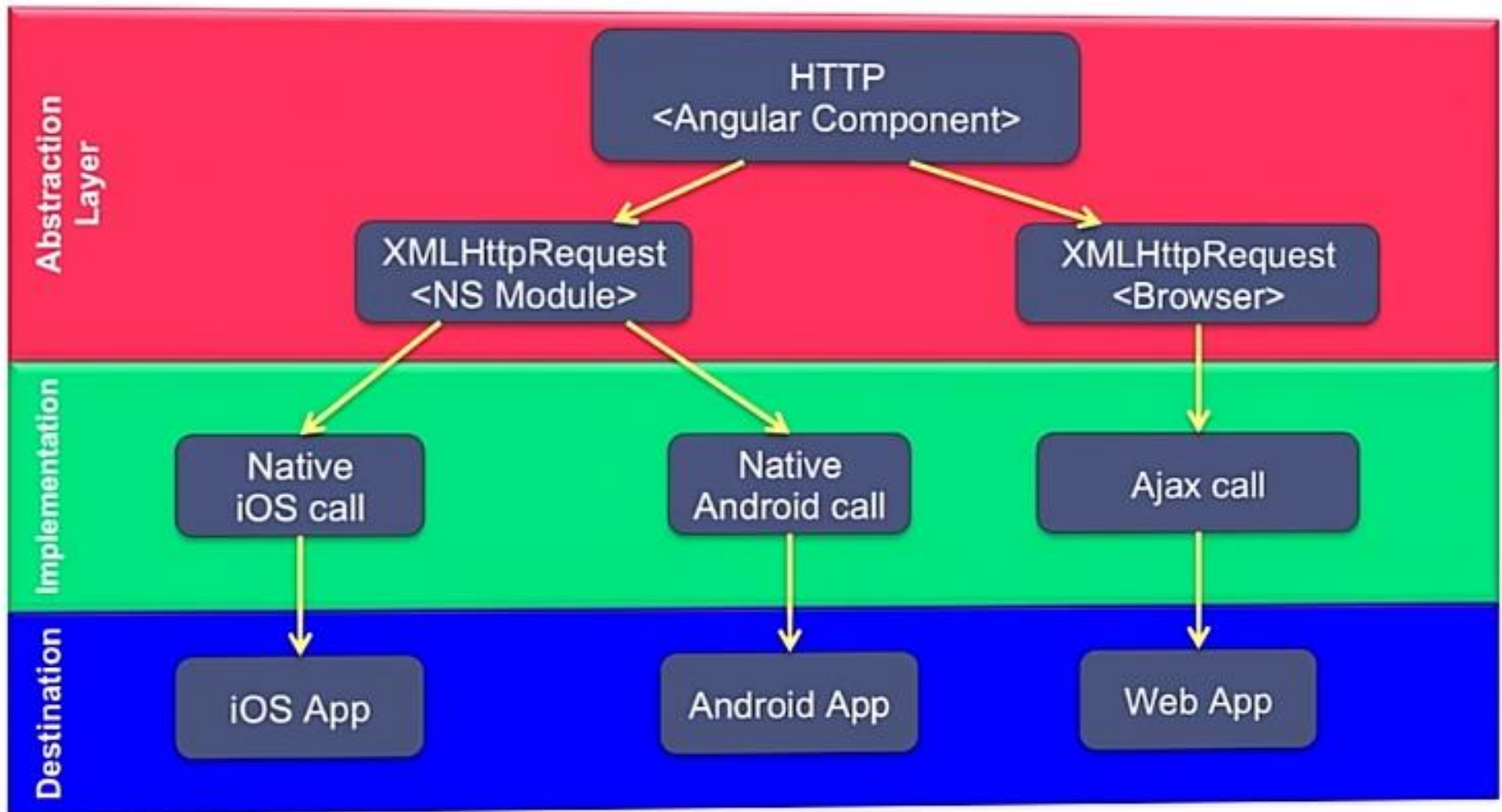
Comparing Ti vs **RN** vs {N}

Future Goals – The Horizontal Platform



Comparing Ti vs RN vs {N}

Future Goals – NativeScript + Angular 2



Comparing Ti vs RN vs {N}

Future Goals – NativeScript + Angular 2

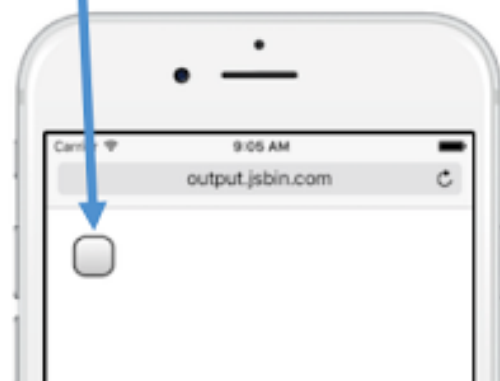
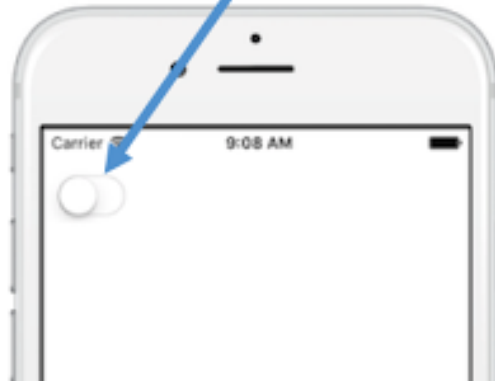


```
@Component({  
  selector: "checkbox"  
  templateUrl: "checkbox.html"  
});
```



<switch>

<input type="checkbox">



Conclusion

React Native brings React further than the browser

Titanium Appcelerator is still a viable and stable runtime based solution

NativeScript is easiest to start with and has 0-day support for new features

End of part 2



Discussion



Appendix



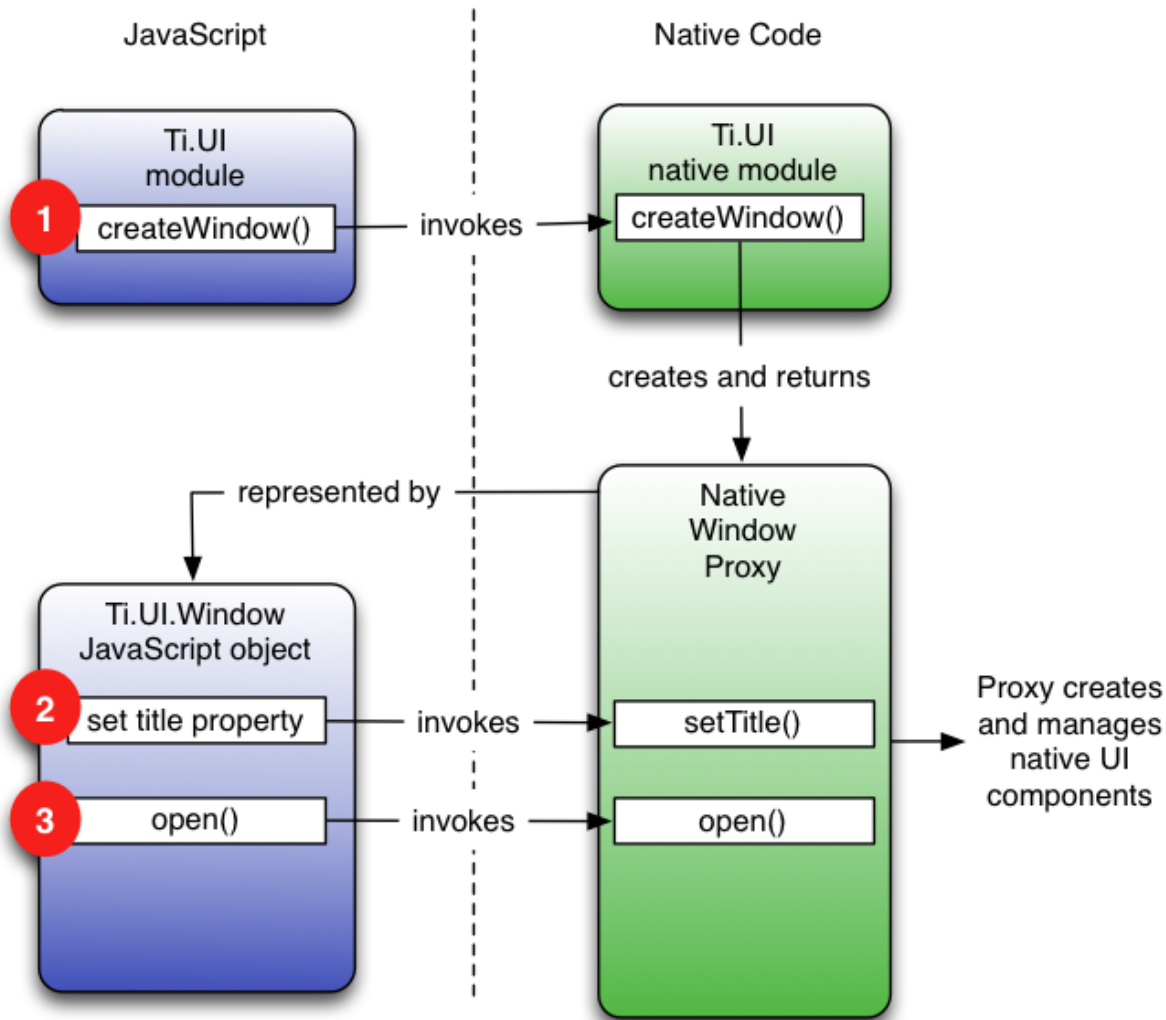
Comparing Ti vs RN vs {N}

Developer Experience – Over-the-air Updates

3.3.2 An Application may not download or install executable code. Interpreted code may only be used in an Application if all scripts, code and interpreters are packaged in the Application and not downloaded. The only exception to the foregoing is scripts and code downloaded and run by Apple's built-in WebKit framework, provided that such scripts and code do not change the primary purpose of the Application by providing features or functionality that are inconsistent with the intended and advertised purpose of the Application as submitted to the App Store.

Comparing Ti vs RN vs {N}

Architecture – Runtime Architecture

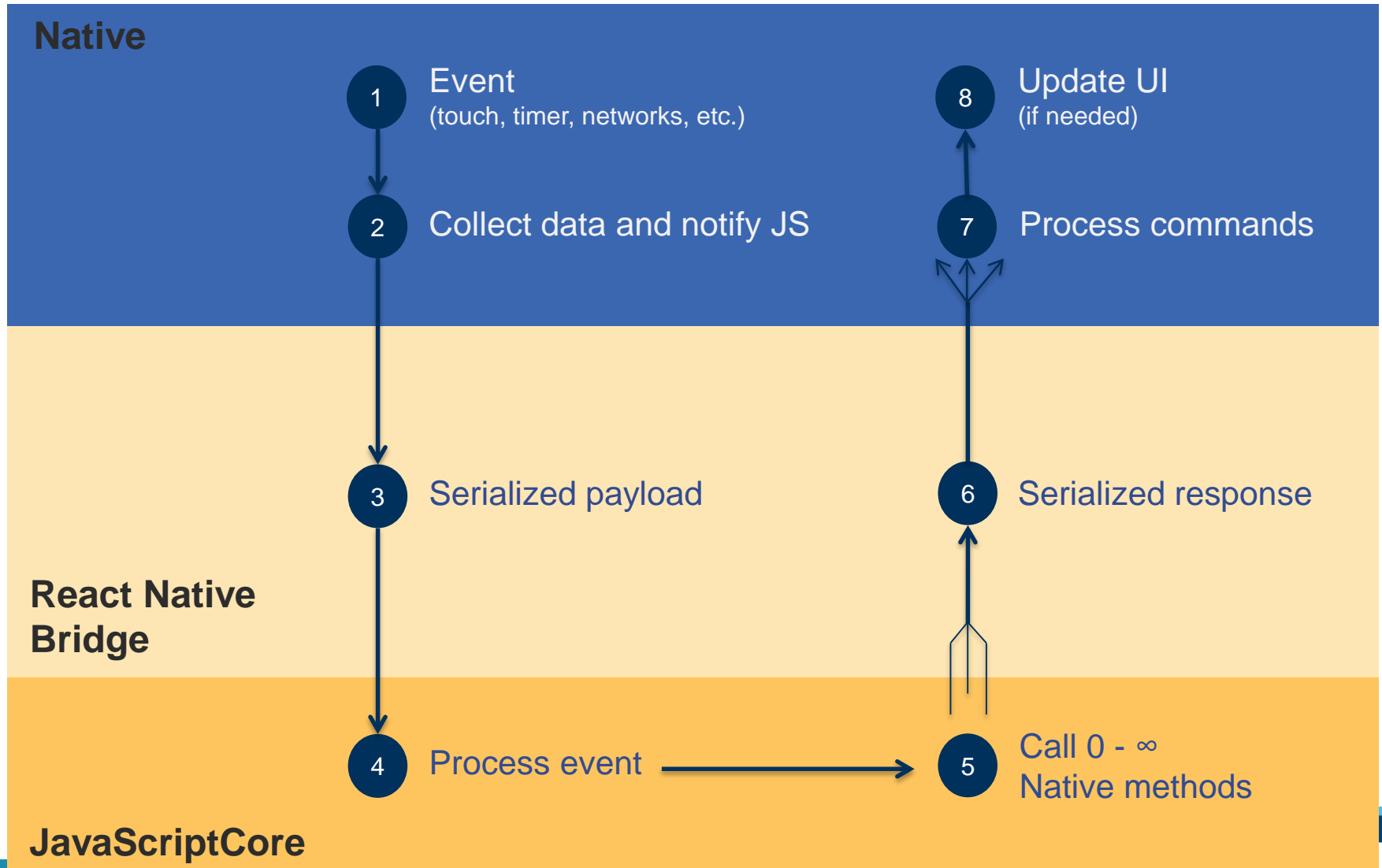
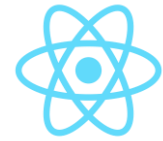


Various Proxies:

- Proxy
- Module
- ViewProxy
- View

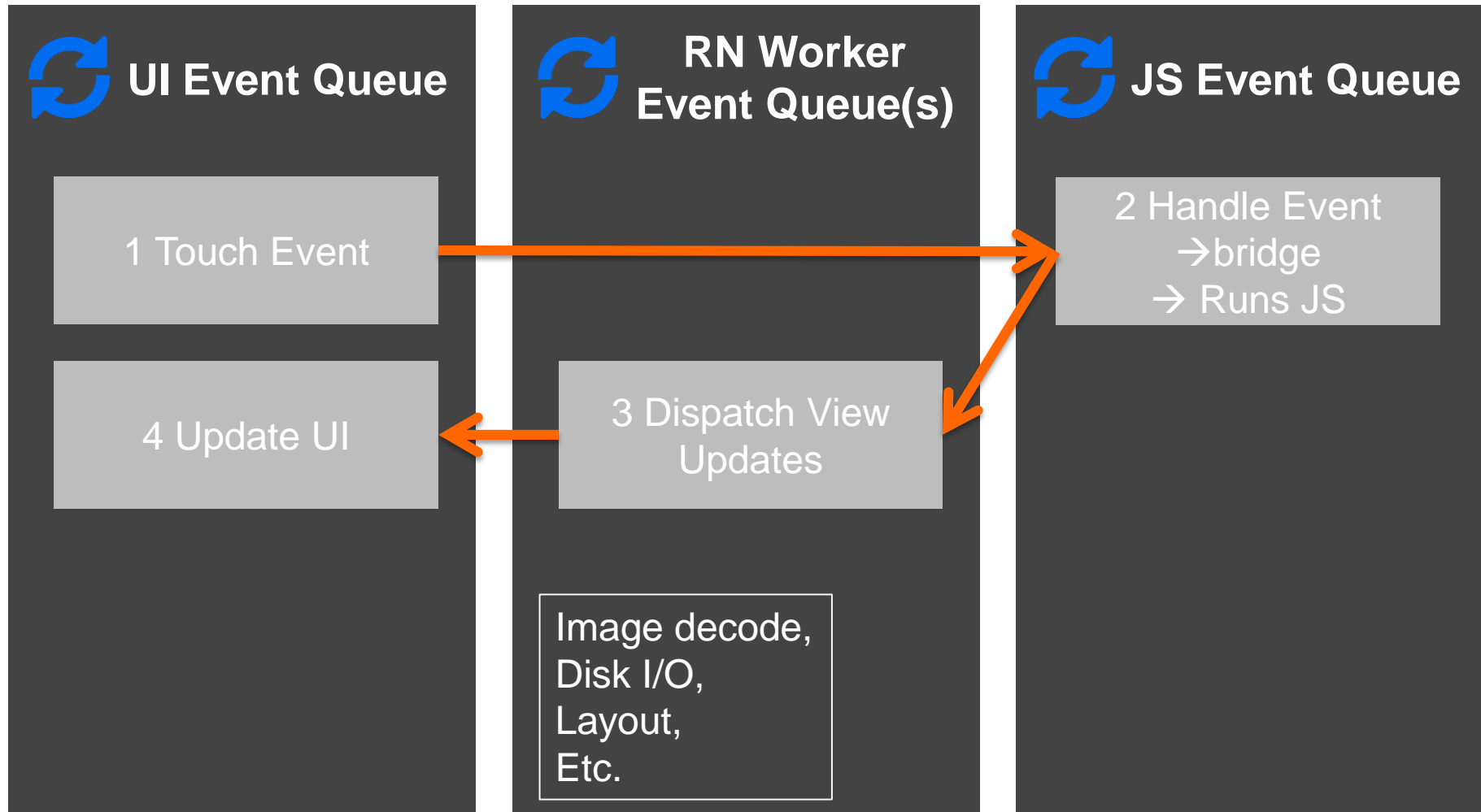
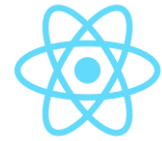
Comparing Ti vs **RN** vs {N}

Architecture – Runtime Architecture



Comparing Ti vs **RN** vs {N}

Architecture – Runtime Architecture Ex.



Appendix Developer Experience – Tools LiveSync



LiveSync both ios and Android

EXPLORER

- WORKING FILES
 - app.js app
 - main-view-model.js app
 - main-page.js app
 - app.css app
- HELLOWORLD
 - .vscode
 - launch.json
 - app
 - App_Resources
 - app.css
 - app.js
 - main-page.js
 - main-page.xml
 - main-view-model.js
 - package.json
 - references.d.ts
 - node_modules
 - platforms
 - package.json

app.css app

```
1 .title {
2   font-size: 30;
3   horizontal-align: center;
4   margin: 20;
5   color: red;
6 }
7
8 button {
9   font-size: 42;
10  horizontal-align: center;
11  color: blue;
12 }
13
14 .message {
15   font-size: 20;
16   color: red;
17   horizontal-align: center;
18   margin: 0 20;
19   text-align: center;
20 }
21
```

DEBUG CONSOLE

iPhone 6 - iPhone 6 / iOS 9.2 (...)

Carrier 7:05 PM

Tap the button

TAP

42 taps left

5554:reactNative

helloworld

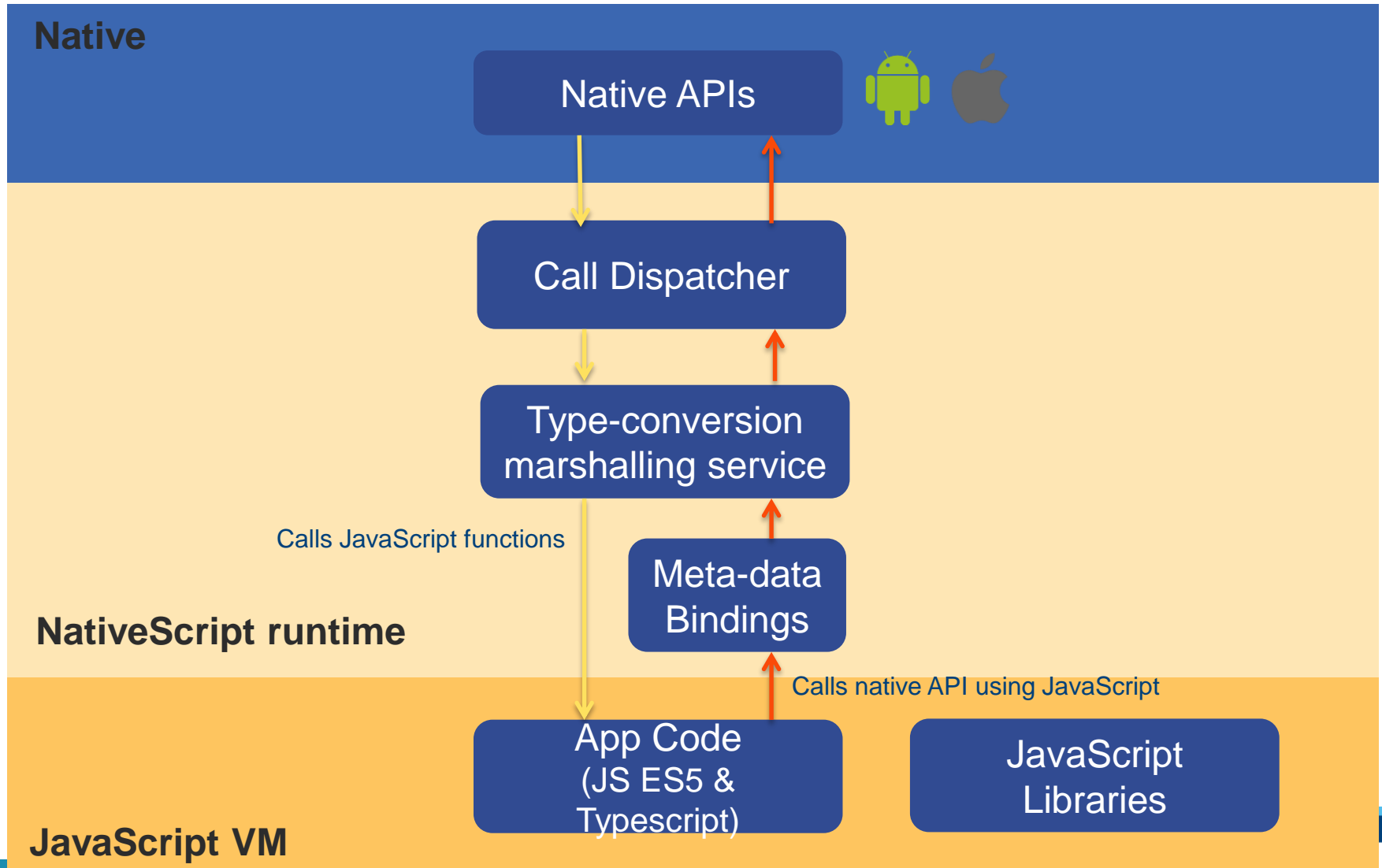
Tap the button

TAP

42 taps left

Comparing Ti vs RN vs {N}

Architecture – Runtime Architecture



Comparing Ti vs RN vs {N}

Architecture – Runtime Architecture Ex.

