

NICCOLÒ BORGIOLI

FIGHT OF THE MOBILES

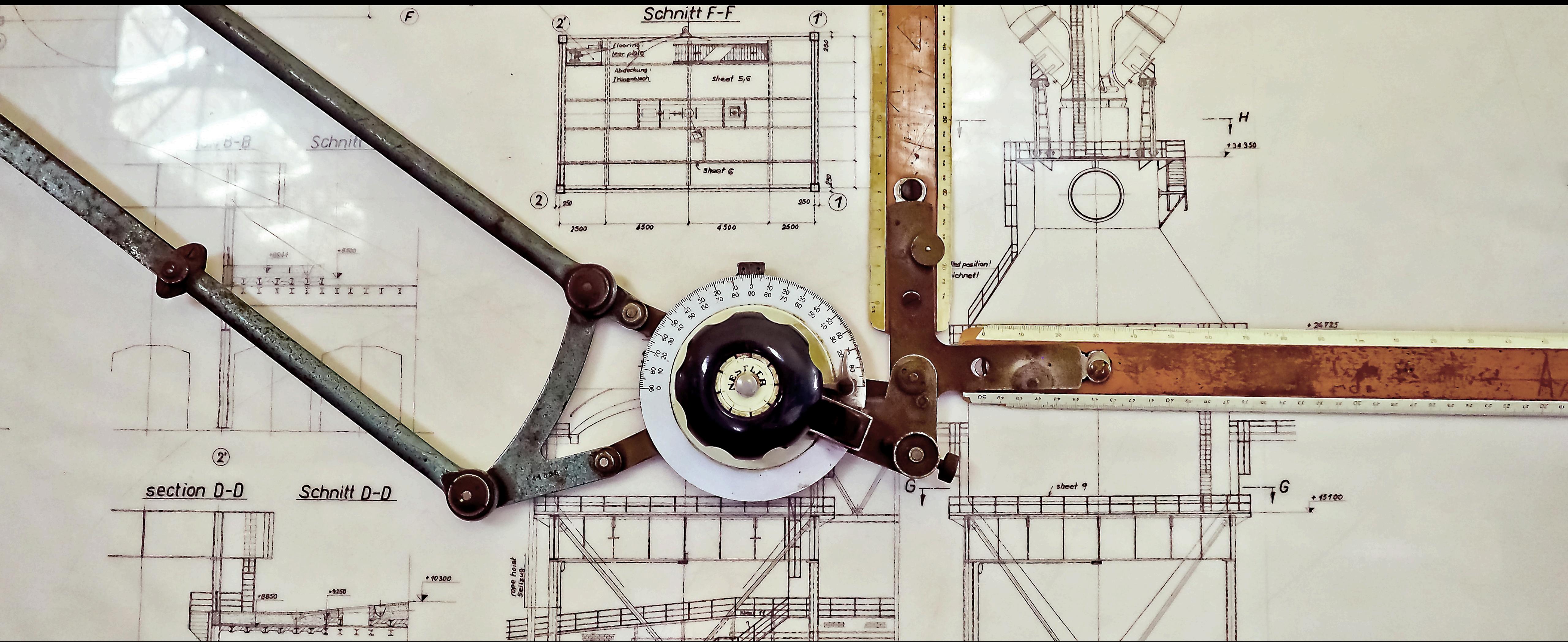
TABLE OF CONTENTS

- Intro
 - Problems
 - Solution candidates
 - Current state
- Solution
 - Test cases & design
 - Building
 - Conclusion



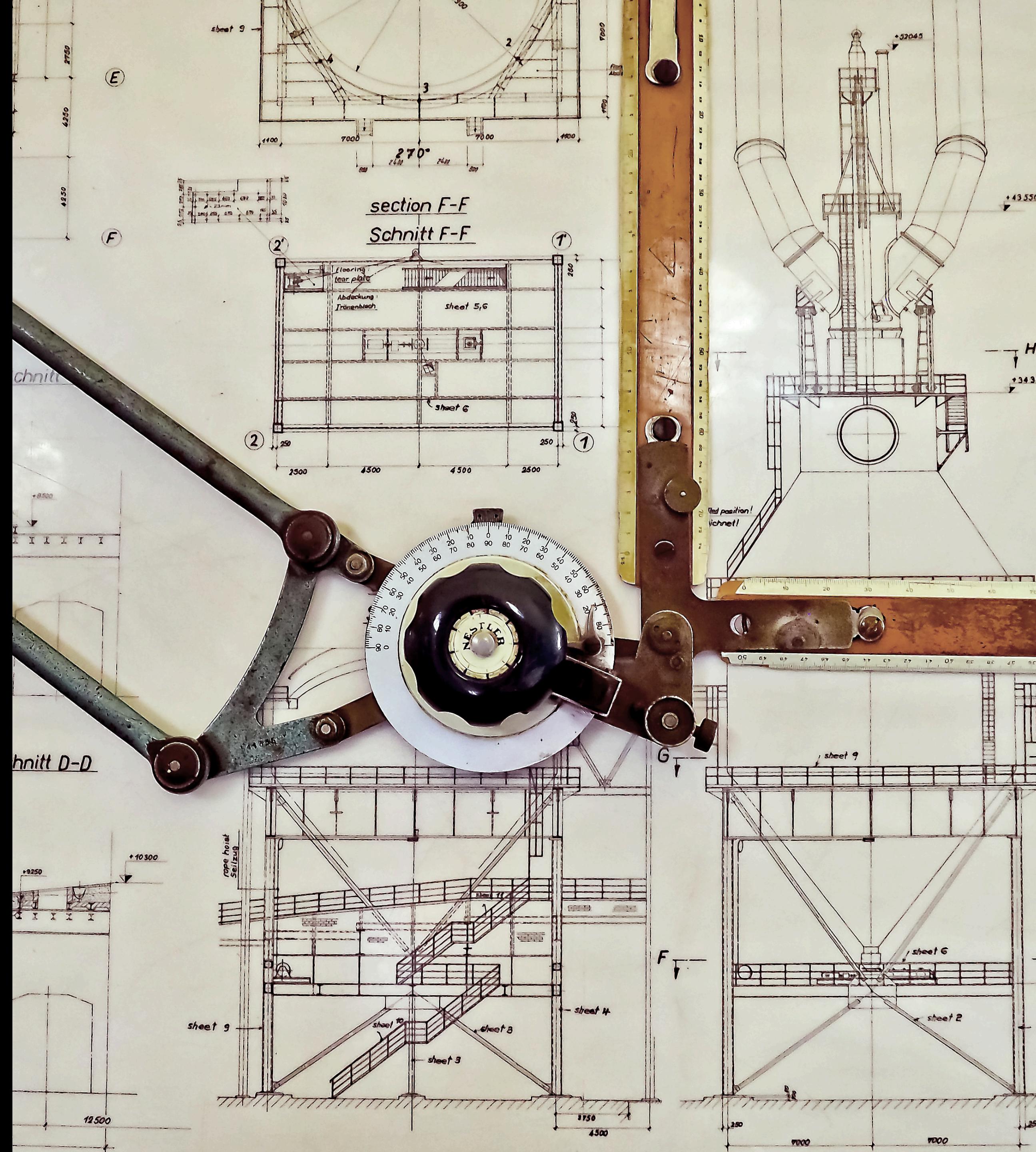
INTRO

BUILDING A MOBILE APP IN 2018



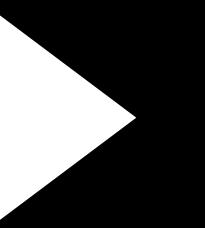
THE PROBLEM WITH NATIVE

- 2 or more codebases
- Code redundancy
- Challenging pushing simultaneous updates
- Different expertise of devs



THE PROBLEM WITH NATIVE

- 2 or more codebases
- Code redundancy
- Challenging pushing simultaneous updates
- Different expertise of devs



THE SOLUTION: HYBRID / COMPILED APPS

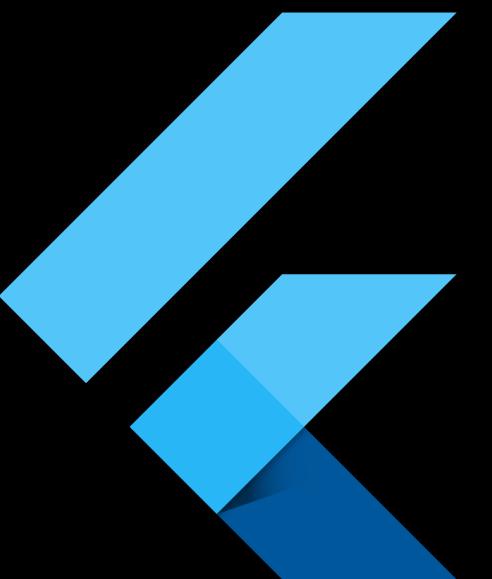
- Single codebase
- Code reusability
- Ensuring a uniform experience across multiple platforms
- Same dev can develop both platforms

SOLUTIONS

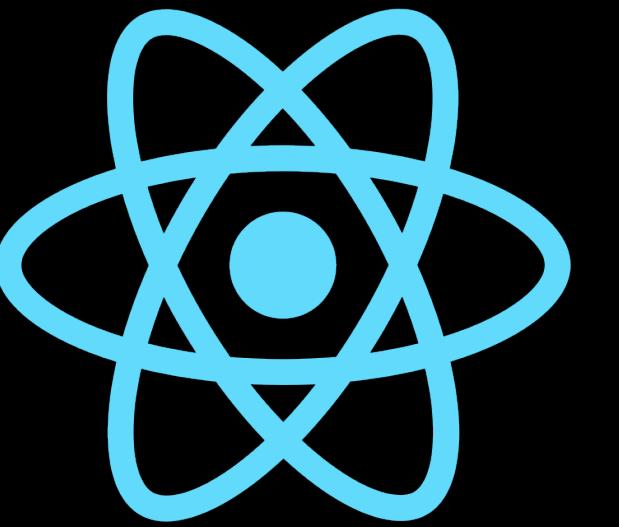


SOLUTIONS

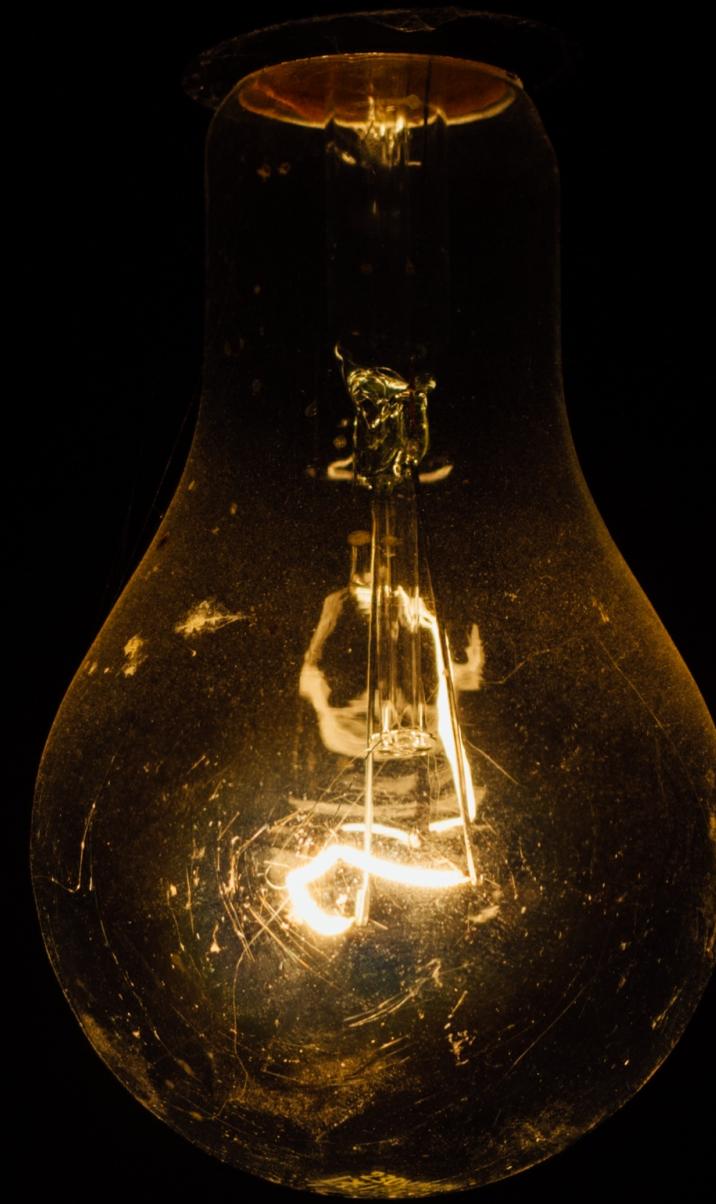
Flutter



React Native



Cordova



HISTORY



FLUTTER

HISTORY

Started by Google as "Sky", which could run under Android. Later rebranded into Flutter.

Unveiled in 2015, with the first preview release (v0.6.0) in July of 2018.

Primary method to write apps and UI for Fuchsia.

Preview Release in July 2018.

USERS

- Alibaba
- Google AdWords
- Greentee

FLUTTER

TECH BEHIND

Flutter does not use the OEM components. It has its own widgets and compiles everything (views and code) down to the phone.

It believes in 120 FPS animations on every platform. Uses Skia under the hood for rendering. Has bindings for GL or Vulkan.

MOTIVATION

- Fuchsia
- Android
- Hip Factor

REACT NATIVE

HISTORY

Inside Facebook, Jordan Walke found a way to generate iOS UI elements from a background JavaScript thread. They decided to organise an internal hackathon to perfect this prototype in order to be able to build native apps with this technology.

Unveiled at React.js Conference in 2015

USERS

- Facebook
- Messenger
- Instagram
- Uber
- Baidu
- Walmart
- OneDrive
- Skype

REACT NATIVE

TECH BEHIND

Code is Javascript that runs on a separated Thread.

Views get compiled to Native views. Instead of the normal VirtualDOM in the browser it manipulates native Views in Android and iOS.

MOTIVATION

- Hip Status
- Internal usage

CORDOVA

HISTORY

Development started at an iPhoneDevCamp
under approval of Apple.

Acquired by Adobe in 2011 and rebranded
under PhoneGap.

A subset got open sourced later as Apache
Cordova.

USERS

?

CORDOVA

TECH BEHIND

Provides a WebView with bindings for native sensors and functions. Much like Electron.

MOTIVATION

- PhoneGap

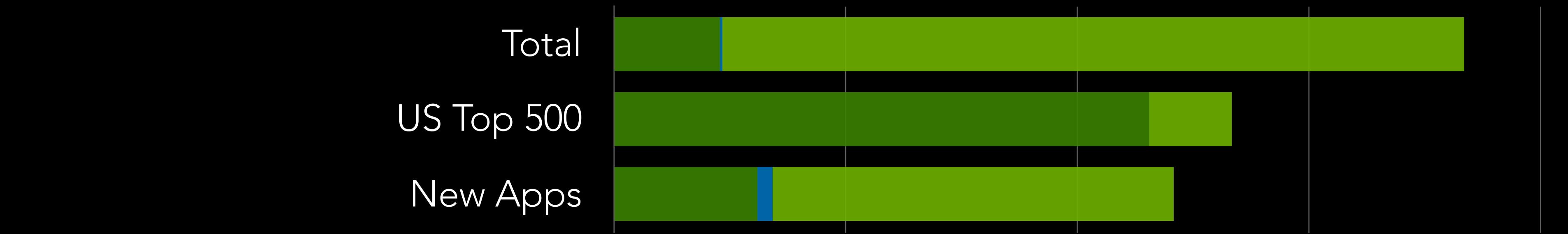
INTRO

CURRENT STATE



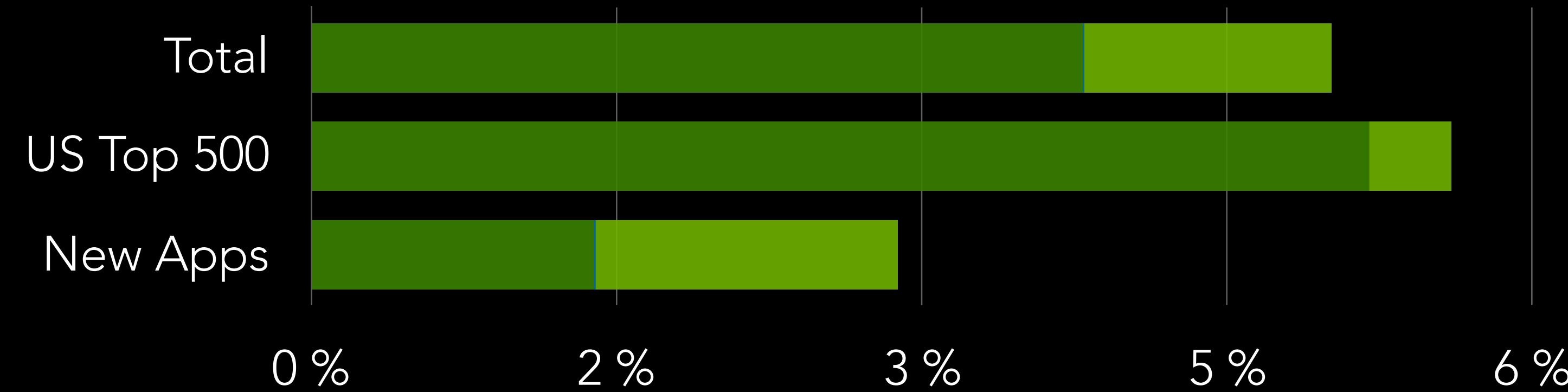
CURRENT STATE

STORE



RN
Flutter
Cordova

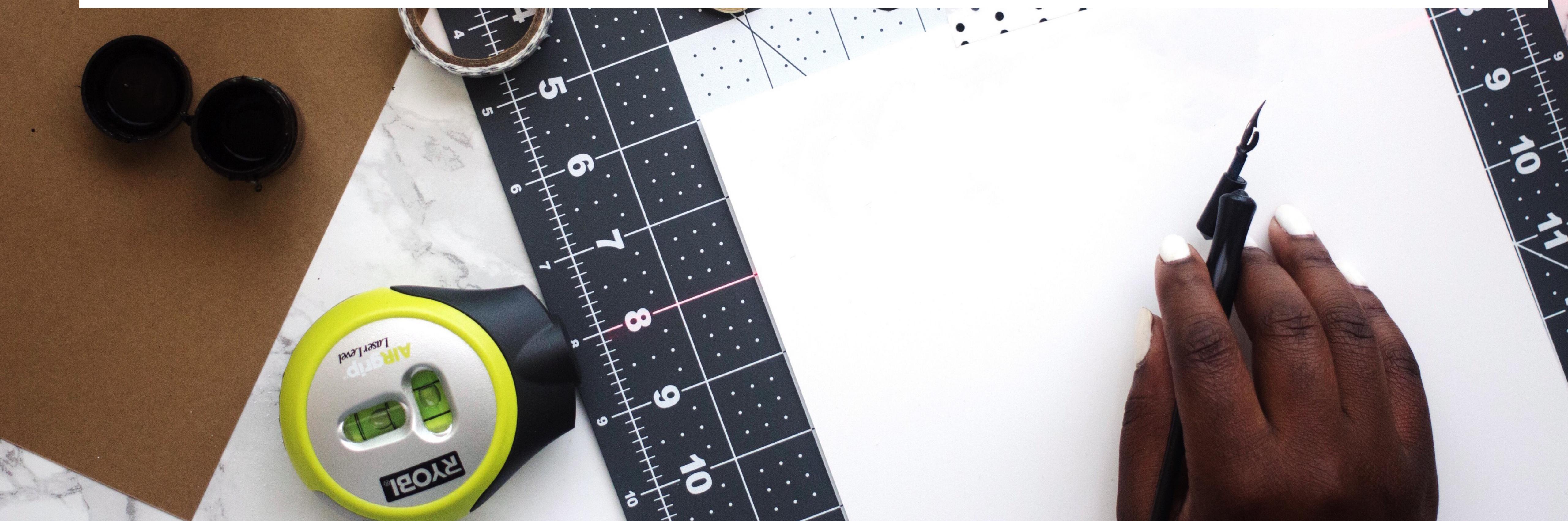
INSTALLS





HOW TO CHOOSE?

BUILD ALL THE FRAMEWORKS



WHAT DO WE WANT TO INCLUDE?

- App Icon
- Navigation
- Animations Simple loading animation
- Camera Scan a QR Code
- Notifications
- Styling Text, fonts, components, gradients
- Reusability Reuse components between screens

DESIGN



Screenshot of Sketch application interface showing a design workspace and various tools.

Sketch Application Bar:

- Insert, Colors, Create Symbol, Symbols, Text Styles, 48%, Edit, Transform, Rotate, Flatten, Mask, Scale, Union, Subtract, Intersect, Difference, Forward, Backward, View, Export.

Pages: Screens iPhone 8, Logo, Symbols (selected).

Symbols:

- Misc/Notification
 - Notification
 - Header
 - Notification Icon
 - App Name
 - Timestamp
 - Title
 - Body
 - Background
 - Button
 - Rectangle
 - Rectangle 2
- Overrides/Status Bar/Signal - ...
 - Group
- Bars/Status/Black
 - Signal
 - Time
 - Pin Right
 - Bluetooth

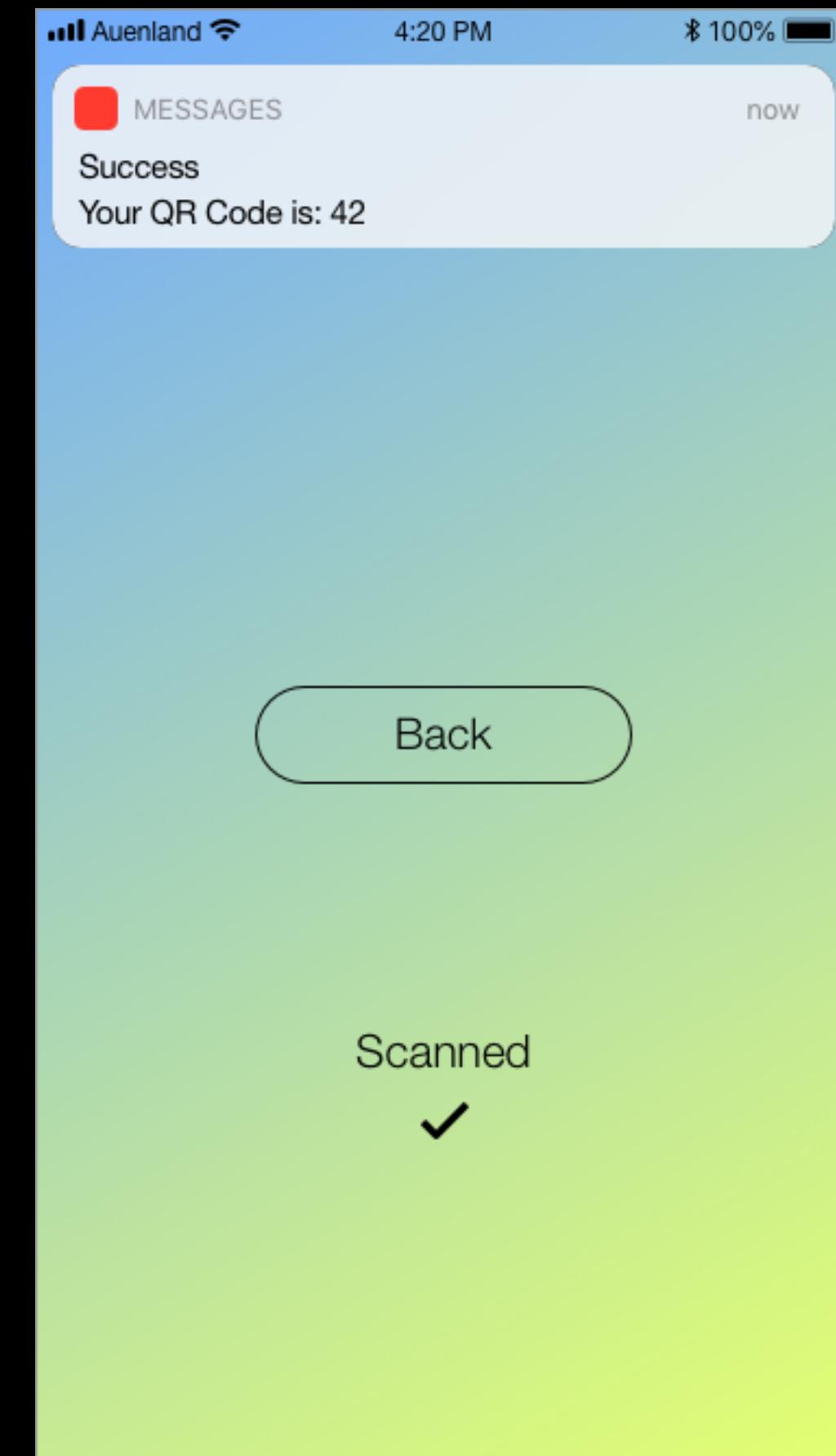
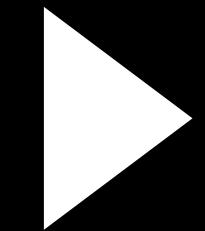
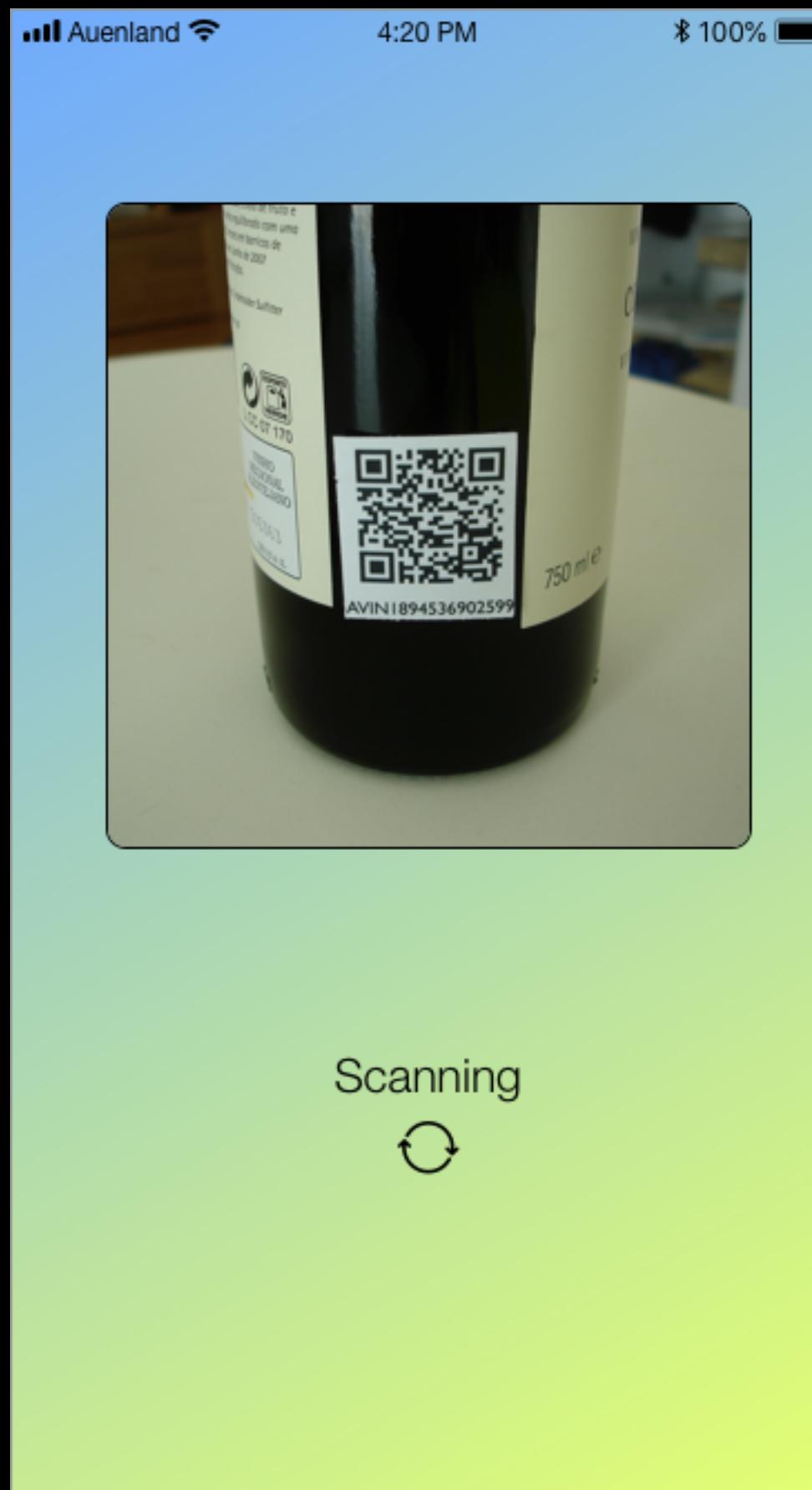
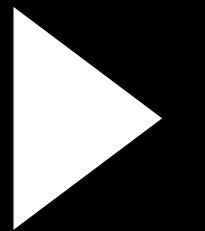
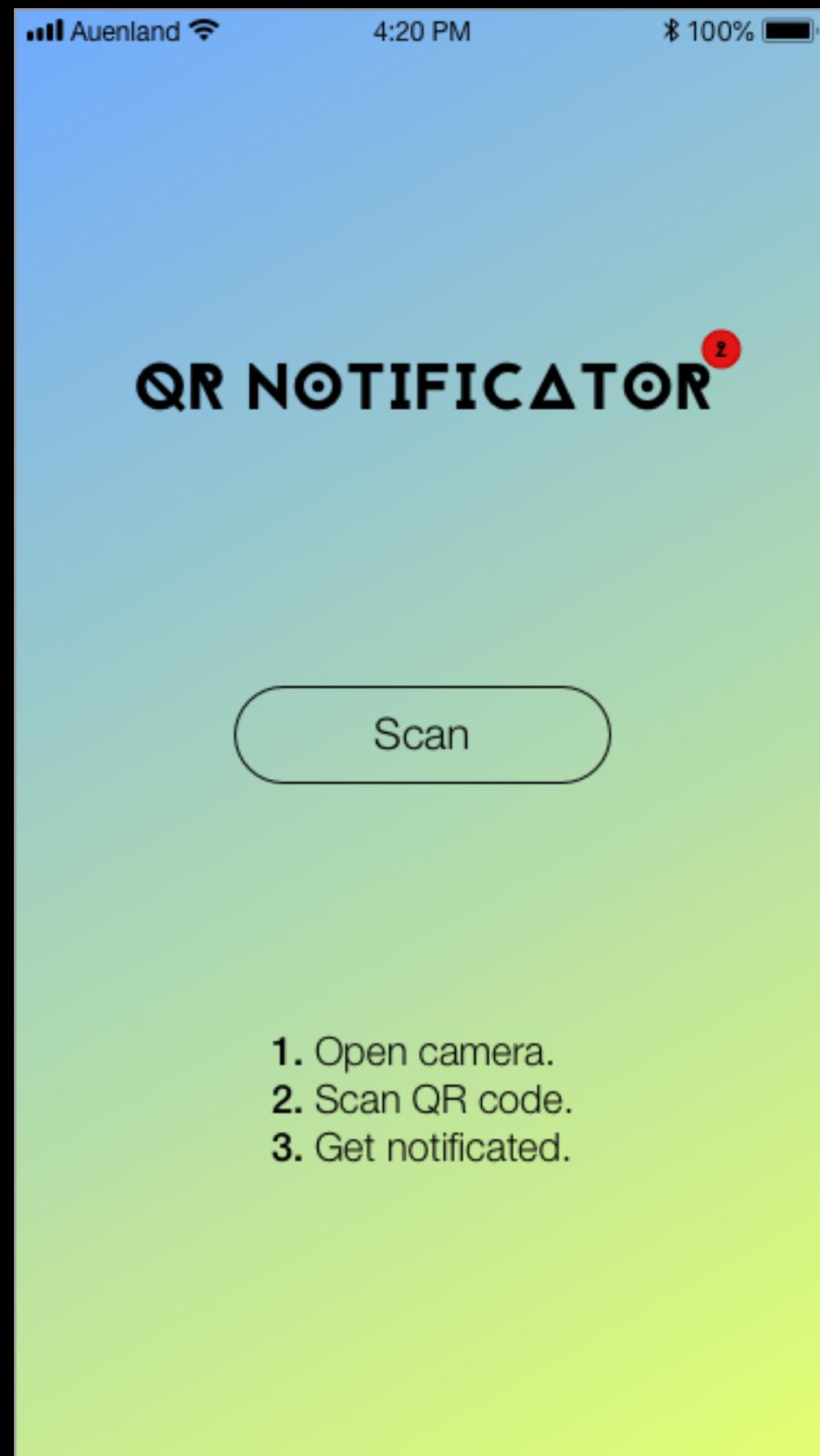
Inspector:

- Position: X, Y
- Size: Width, Height
- Transform: Rotate, Flip
- Fills
- Borders
- Shadows
- Inner Shadows
- Gaussian Blur

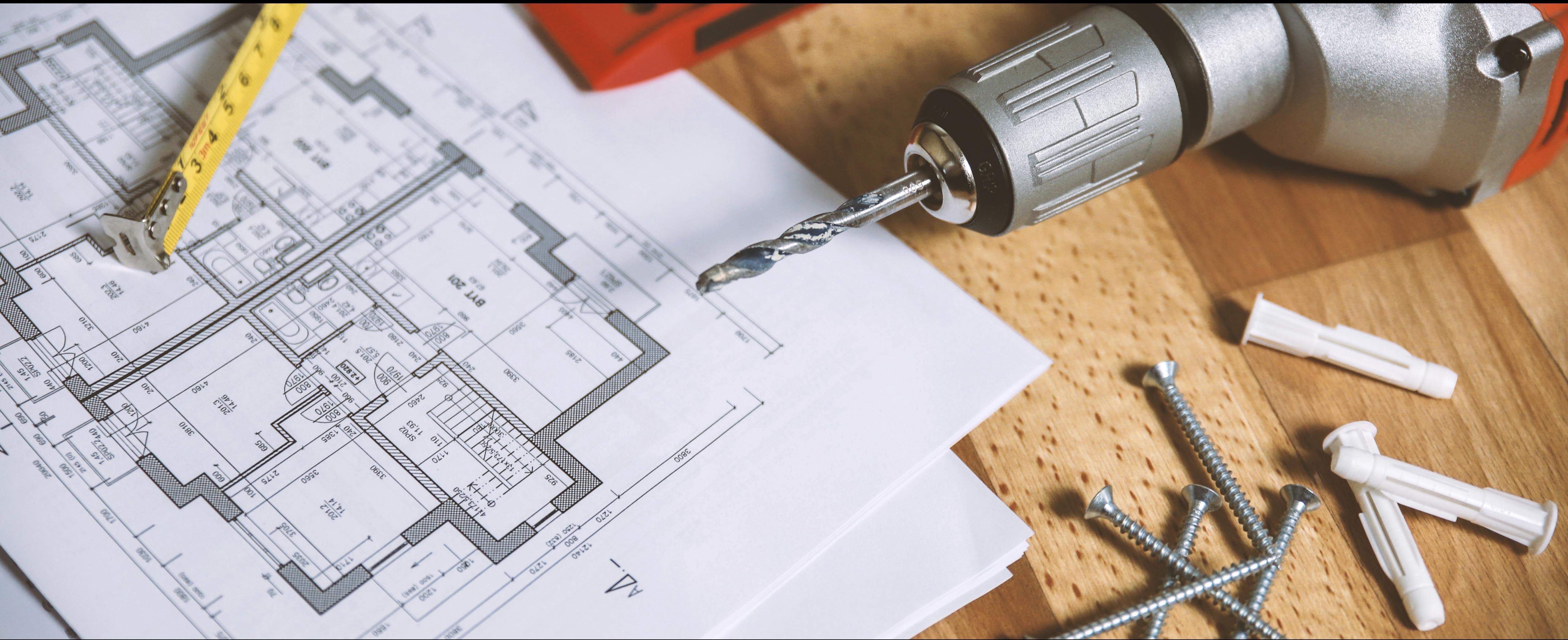
Canvas:

- Background (BG) with a blue-to-yellow gradient.
- Over...Black (status bar placeholder).
- Button (button placeholder).
- Misc/Notification (notification placeholder with message: "MESSAGES Success Your QR Code is: 42 now").
- Bars/Status/Black (status bar placeholder).

DESIGN



TIME TO BUILD



FLUTTER

- Dart: a mix between Typescript and Java
- Styling is much more "Native"
- Lot of nesting

```
9   import 'package:flutter/material.dart';
8
7   class Button extends StatelessWidget {
6   |   Button({@required this.text, @required this.callback});
5
4   |   final String text;
3   |   final Function callback;
2
1   |   @override
10  |   Widget build(BuildContext context) {
1   |   |   return GestureDetector(
1   |   |   |   onTap: this.callback,
1   |   |   |   child: Container(
1   |   |   |   |   width: 180.0,
1   |   |   |   |   height: 45.0,
1   |   |   |   |   decoration: BoxDecoration(
1   |   |   |   |   |   borderRadius: BorderRadius.circular(100.0),
1   |   |   |   |   |   border: Border.all(width: 1.0, color: const Color(0xFF000000)),
1   |   |   |   |   |   child: Center(
1   |   |   |   |   |   |   child: Text(
1   |   |   |   |   |   |   |   text,
1   |   |   |   |   |   |   |   style: TextStyle(fontSize: 20.0),
1   |   |   |   |   |   |   )))); // Text // Center // Container // GestureDetector
14  |   |
15  | }
16 }
```

FLUTTER

POSITIVE ❤️

- "Batteries included"
- Docs

NEGATIVE 🚫

- Dart
- Ecosystem

REACT NATIVE

- It's basically React
- Instead of HTML Tags we use predefined native components.

```
3 import React, { Component } from 'React'  8.1K (gzipped: 3.3K)
2 import { StyleSheet, TouchableOpacity, Text } from 'react-native'
1
4 export default class extends Component {
1   render() {
2     return <TouchableOpacity
3       style={styles.button}
4       onPress={this.props.callback}>
5       <Text style={styles.label}>{this.props.text}</Text>
6     </TouchableOpacity>
7   }
8 }
9
10 const styles = StyleSheet.create({
11   label: {
12     fontSize: 20
13   },
14   button: {
15     width: 180,
16     height: 45,
17     borderRadius: 100,
18     borderColor: '#000000',
19     borderWidth: 1,
20     alignItems: 'center',
21     justifyContent: 'center',
22   },
23 })
```

REACT NATIVE

POSITIVE ❤️

- Currently the biggest community
- A LOT of 3rd party packages
- For react devs very easy to get started

NEGATIVE 🚫

- Feels Hacky
- Nothing out of the box.
- Docs

CORDOVA

- In this case React.
- Works with every Website, since it packs into an "enhanced" browser.

```
2 import React from 'react'  8.1K (gzipped: 3.3K)
1
3 export default class Logo extends React.Component {
1   render() {
2     return <div className="Logo">
3       <div className="title">QR NOTIFIER</div>
4       <div className="badge">
5         <div>2</div>
6       </div>
7     </div>
8   }
9 }
```

```
.Logo {
  position: relative;
  font-family: 'Jaapokki';
}
.title {
  font-size: 30px;
}
.badge {
  .center;
  position: absolute;
  right: -20px;
  top: -5px;
  width: 20px;
  height: 20px;
  background-color: #f00;
  border-radius: 20px;
  div {
    font-size: 10px;
  }
}
```

CORDOVA

POSITIVE ❤️

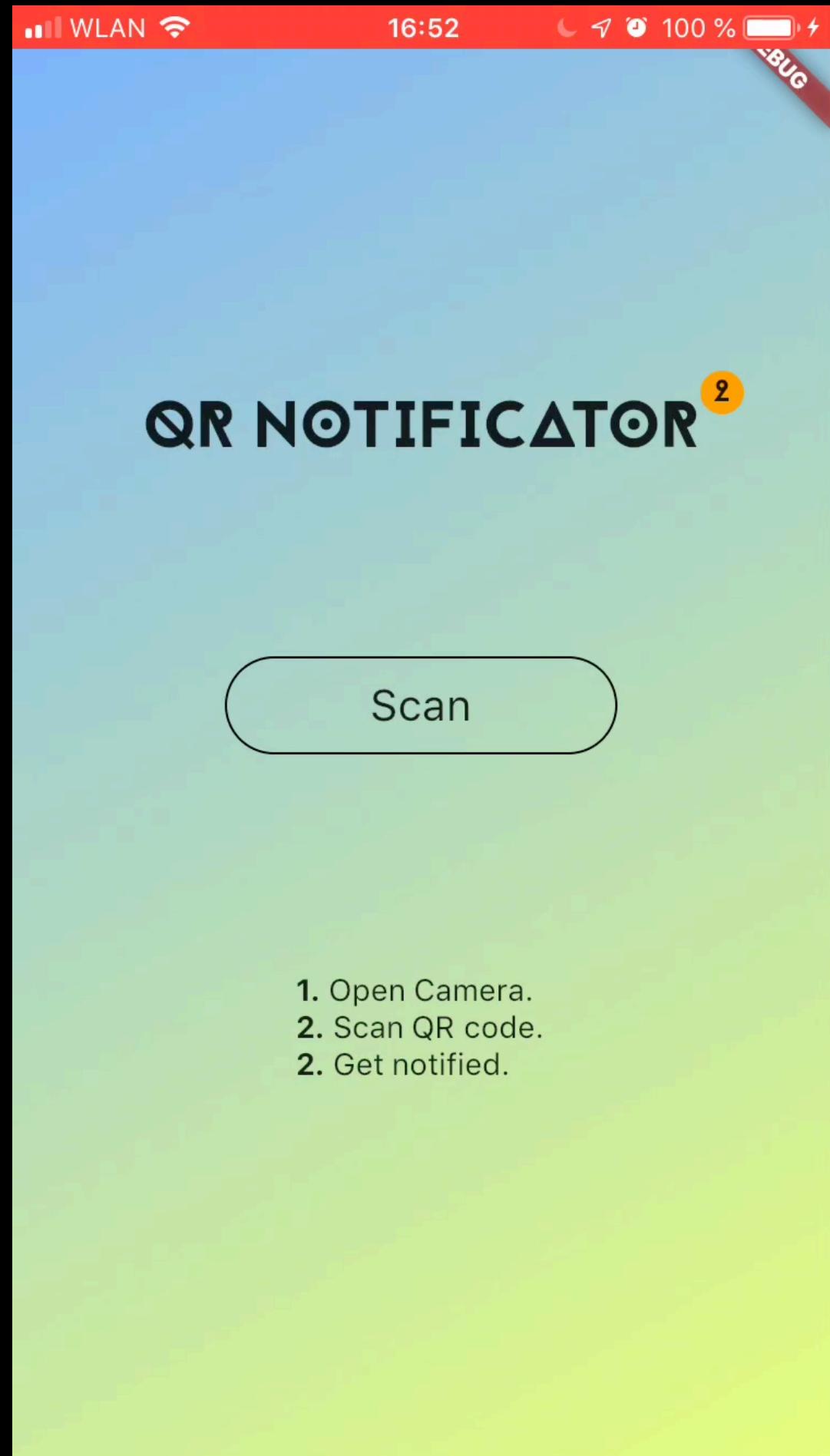
- Known technologies
- Reusable codebase between app and website!
- Huge ecosystem

NEGATIVE 🚫

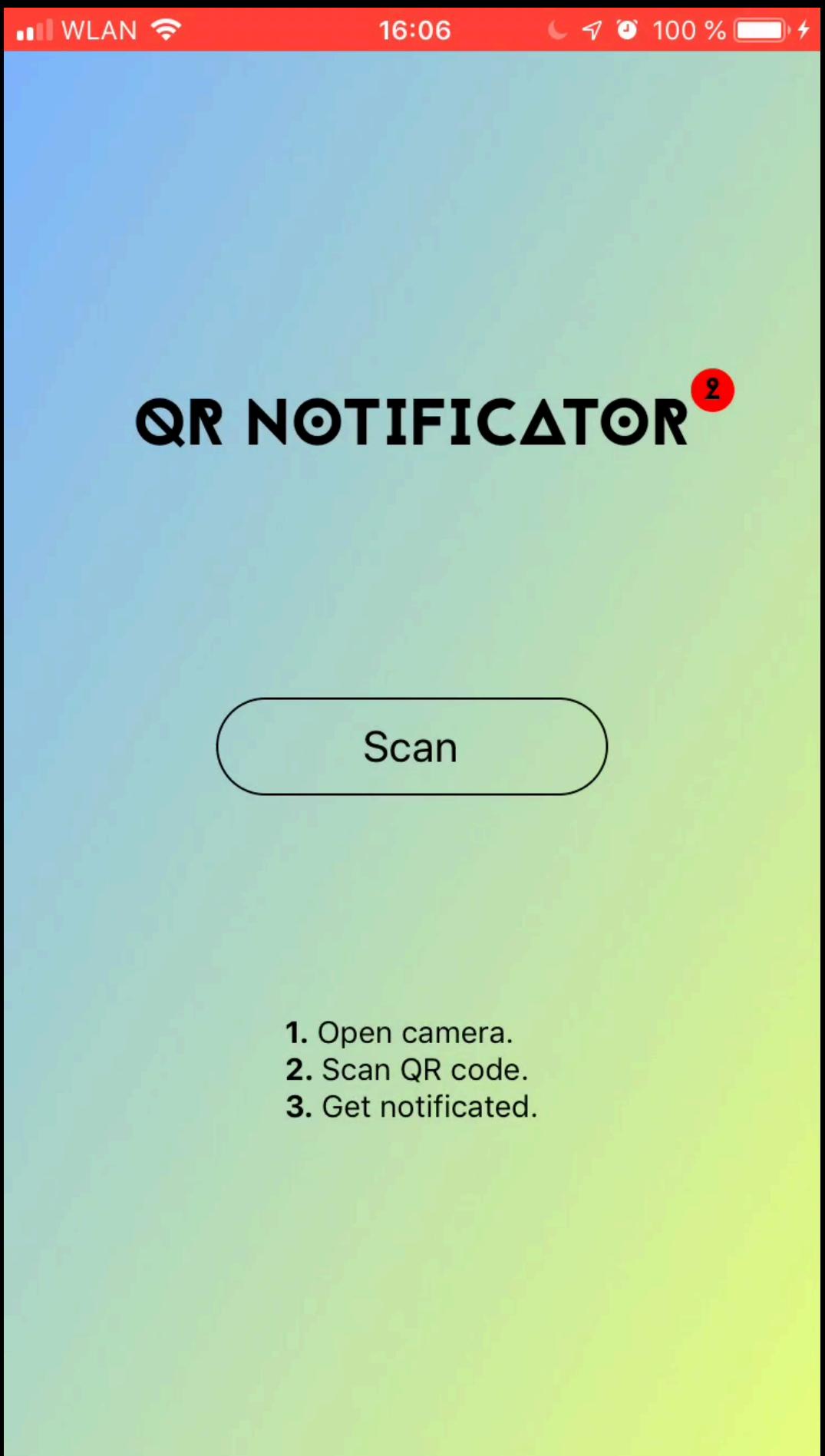
- Performance

RESULTS

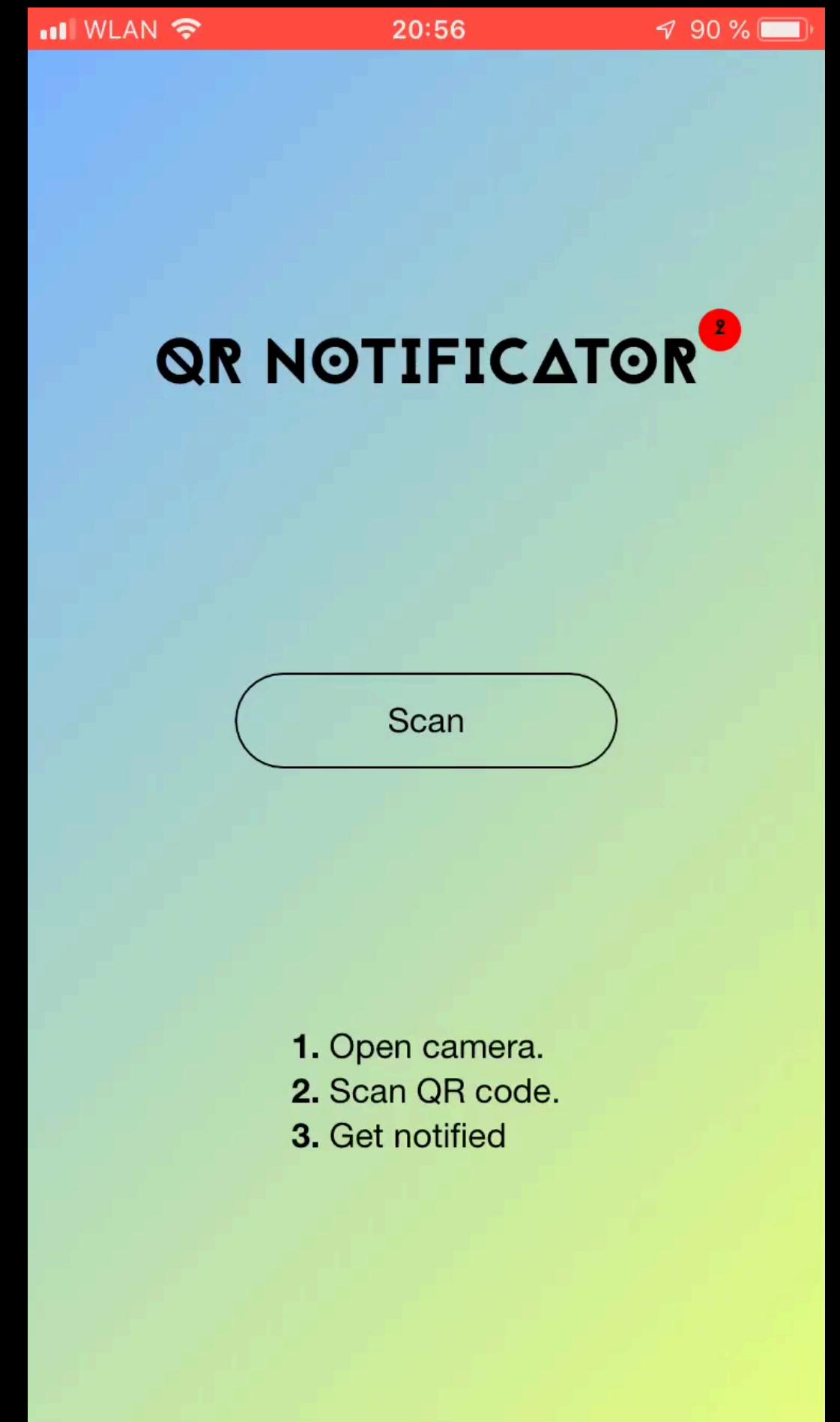
FLUTTER



REACT NATIVE



CORDOVA

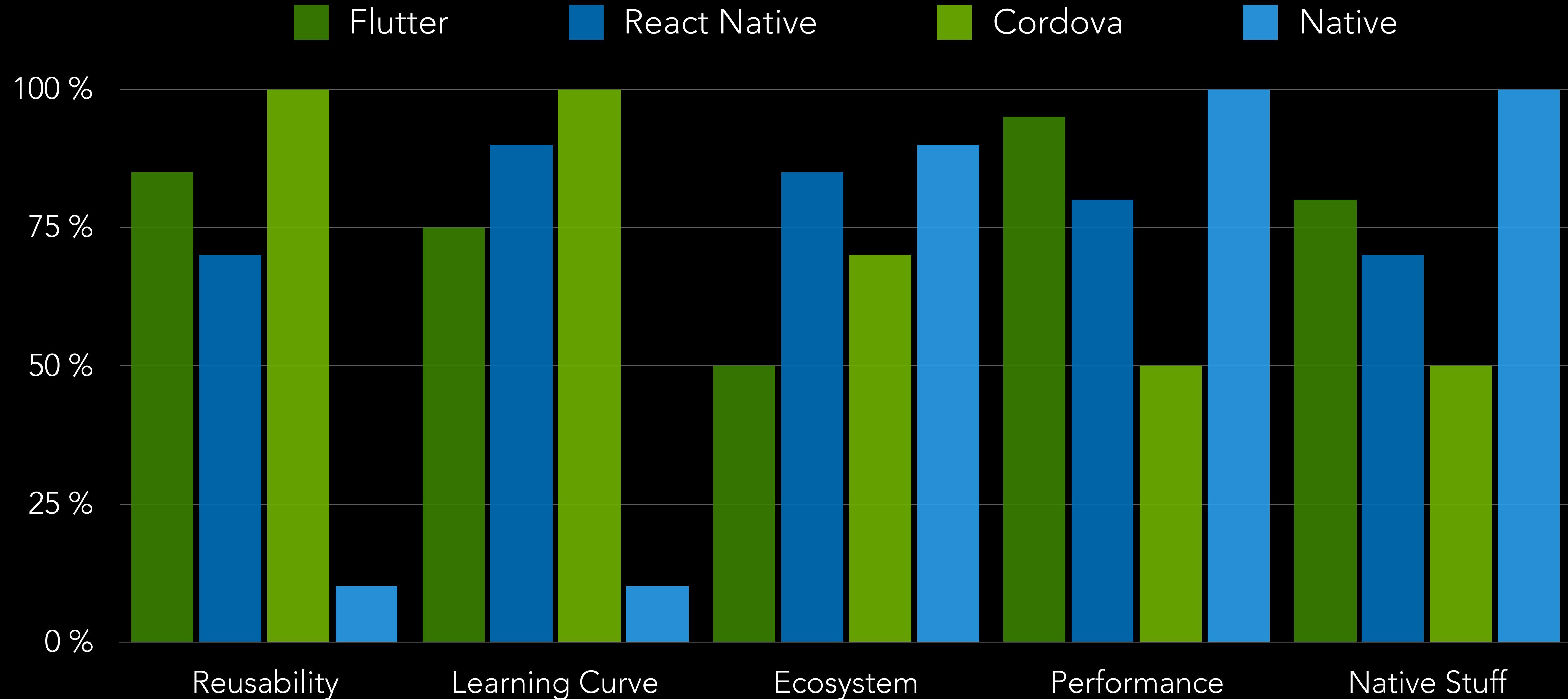


CONCLUSION

COMPARISON



COMPARISON



CONCLUSION

WHICH ONE IS THE RIGHT ONE ?



CONCLUSION

FLUTTER

- Right now a bit too "new" but in the future this could be become the way to go.

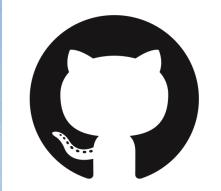
REACT NATIVE

- Currently the most viable and flexible option due to popularity and therefore community.

CORDOVA

- Only if the same codebase between mobile and web is strictly necessary.

END / QA



github.com/CupCakeArmy/fight-of-the-mobiles