Current Trends in Software Engineering

Lesson 2: Introduction to Flutter

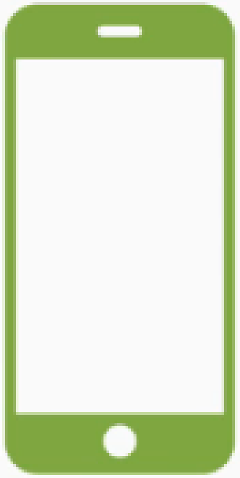
From previous Lesson

- History of AI
- Machine Learning, Deep Learning and Artificial Intelligence
- Types of ML
- ML Algorithms
- Azure machining learning Studio

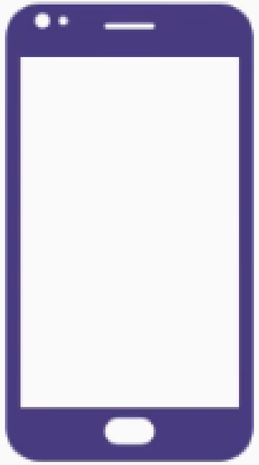
Content

- Mobile App Development Frameworks
- What is Flutter?
- Why Flutter?
- Flutter vs React Native
- Flutter Framework Architecture
- Drawbacks and Limitations in Flutter

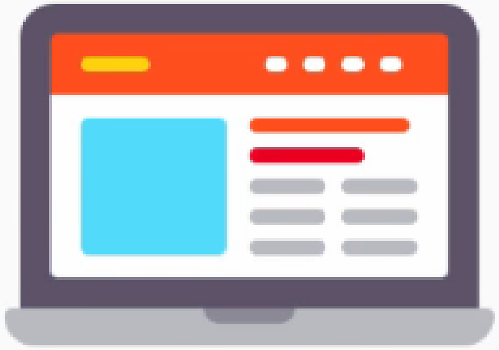
Ways to Create Mobile Apps



Native Apps



Hybrid Apps



Web Apps

Top Mobile App Development Frameworks/Libraries

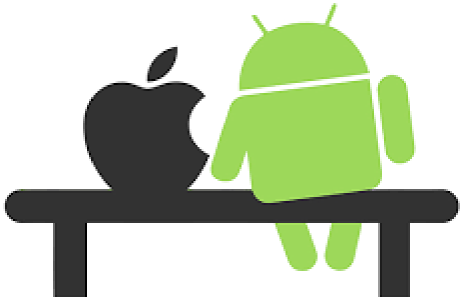
- React Native
- Ionic
- Xamarin
- Apache Cordova/PhoneGap
- Corona SDK
- JQuery Mobile
- Intel XDK
- Native Scripts
- Mobile Angular UI

Which is the most suitable framework for Mobile Development?

- The frameworks are the most important driving tools for building any mobile or web application.

- But for every project/app has a **different requirement** and so the choice of the framework must be made carefully and after ascertaining the other **essential factors**.

Cross Platform App development techniques



Write once, deploy everywhere

- Some time back developing and maintaining the same app for both **iOS** and **Android** is a tedious task.
- **Facebook** lead this revolution and back in **2015**, introduced **React native**.
- Cross-platform apps are comparatively **less performant** but are **cost-effective.**
- So startups and small businesses go for cross-platform app development much more easily than large enterprises.

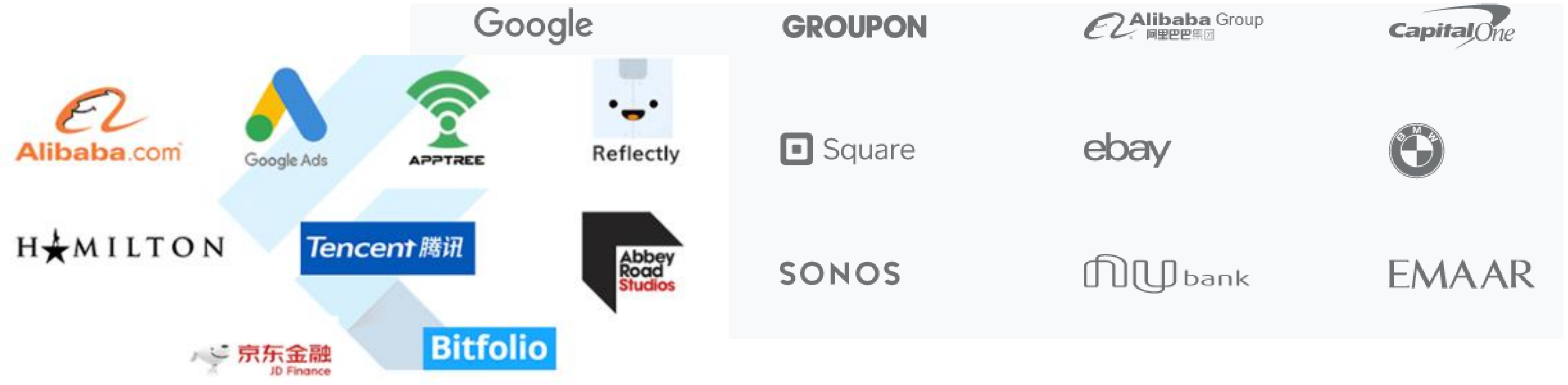
What is Flutter?

- Flutter is a SDK released by Google
  - **free and open source**
  - **Google mobile UI framework**

  that provides a **fast and expressive way** for developers to build native apps.

- It allows the developers to create apps for IOS & Android, using a single code base.

- Flutter was released in May 2017.
- On May 2019, Google announced the availability of the new stable build, Flutter 1.7.



Why Flutter?

- The biggest selling point of Flutter is that once you develop an app it runs on both Android and IOS and flutter is backed by **Google**.

- **Fast development**

  Flutter engineered for high development velocity. Stateful **hot reload** allows to change the code and see it come to life is less than a second without losing the state of the app. (JIT and AOT)

- **Hot Reload**

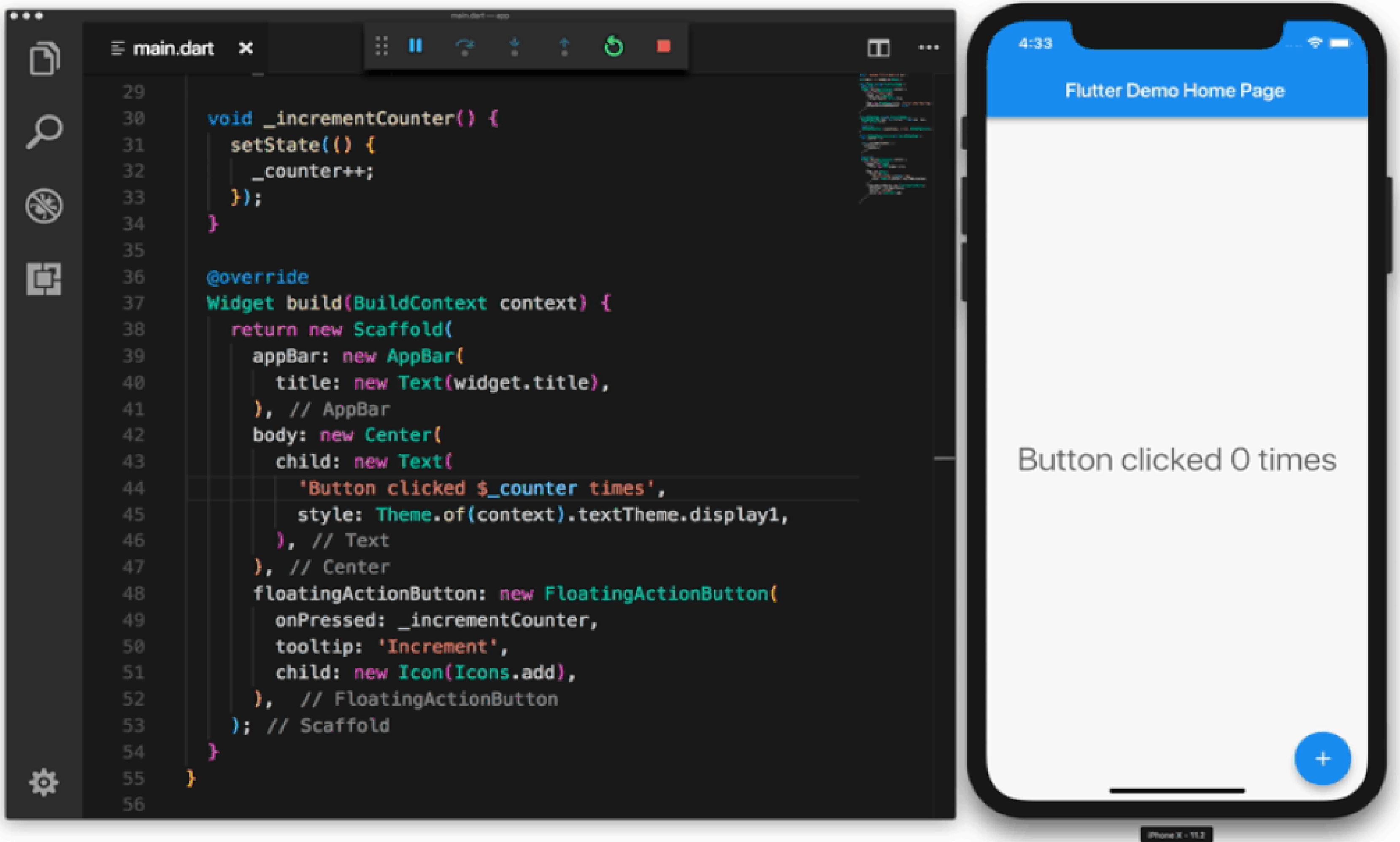
  Same as the web, just hit a refresh and your codes also refresh.

- **High Performance**

  Flutter doesn't require a JavaScript bridge and the speed is much faster.

➢ Flutter is focusing on a single codebase

Flutter doesn’t use any of the tech stacks which is most popular on the internet. javascript, Swift, C++, etc.



```

  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      appBar: new AppBar(
        title: new Text(widget.title),
      ), // AppBar
      body: new Center(
        child: new Text(
          'Button clicked $_counter times',
          style: Theme.of(context).textTheme.display1,
        ), // Text
      ), // Center
      floatingActionButton: new FloatingActionButton(
        onPressed: _incrementCounter,
        tooltip: 'Increment',
        child: new Icon(Icons.add),
      ),  // FloatingActionButton
    ); // Scaffold
  }
}
```

Why Flutter? Cont…

- **A single language for layout and backend.**

  Because Android has separate layout and Java/Kotlin files, we need to get references for views and then alter them in the backend files. Flutter uses a single language(Dart) for design as well as backend and uses a reactive framework

- **Less testing**

  Because of the same app for 2 platforms. So the Quality Assurance process can be faster.

Why Flutter? Cont…

- **Expressive + Flexible UI Animations**

  Widget, rendering, animation and gestures

  Have the flexibility to build a custom design.

- **Easy-to-learn development language for existing mobile developers.**

  One code for 2 platforms

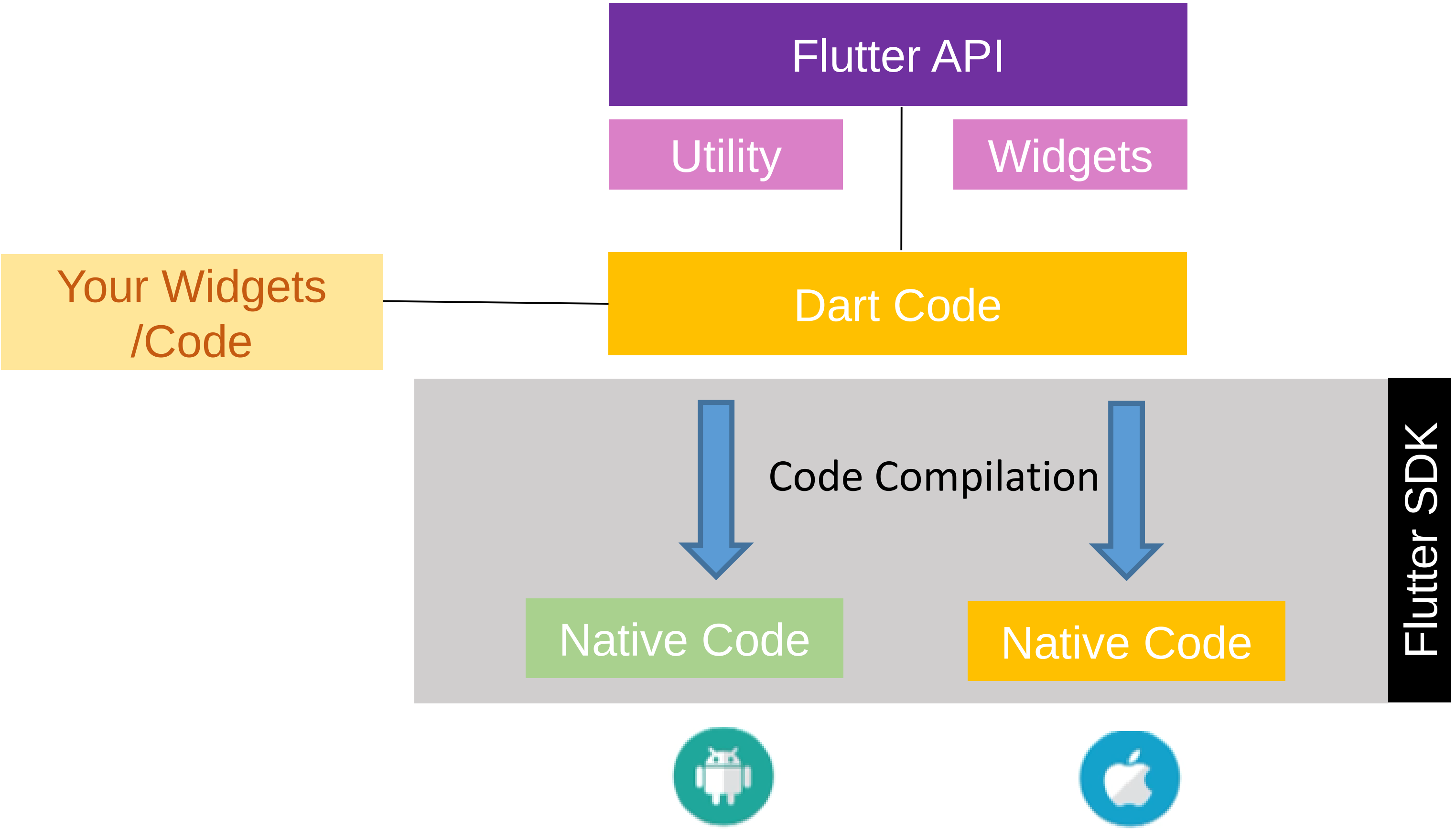
  Allows Android + iOS apps from the same codebase.

- **The same app UI on older devices**

  The app will look the same, even on old versions of Android and iOS systems.

  There are no additional costs for supporting older devices. Flutter runs on Android Jelly Bean or newer, as well as iOS 8 or newer.

How is Flutter/Dart “transformed” to a Native App?



Flutter API

Utility

Widgets

Your Widgets /Code

Dart Code

Code Compilation

Native Code

Native Code

Flutter SDK

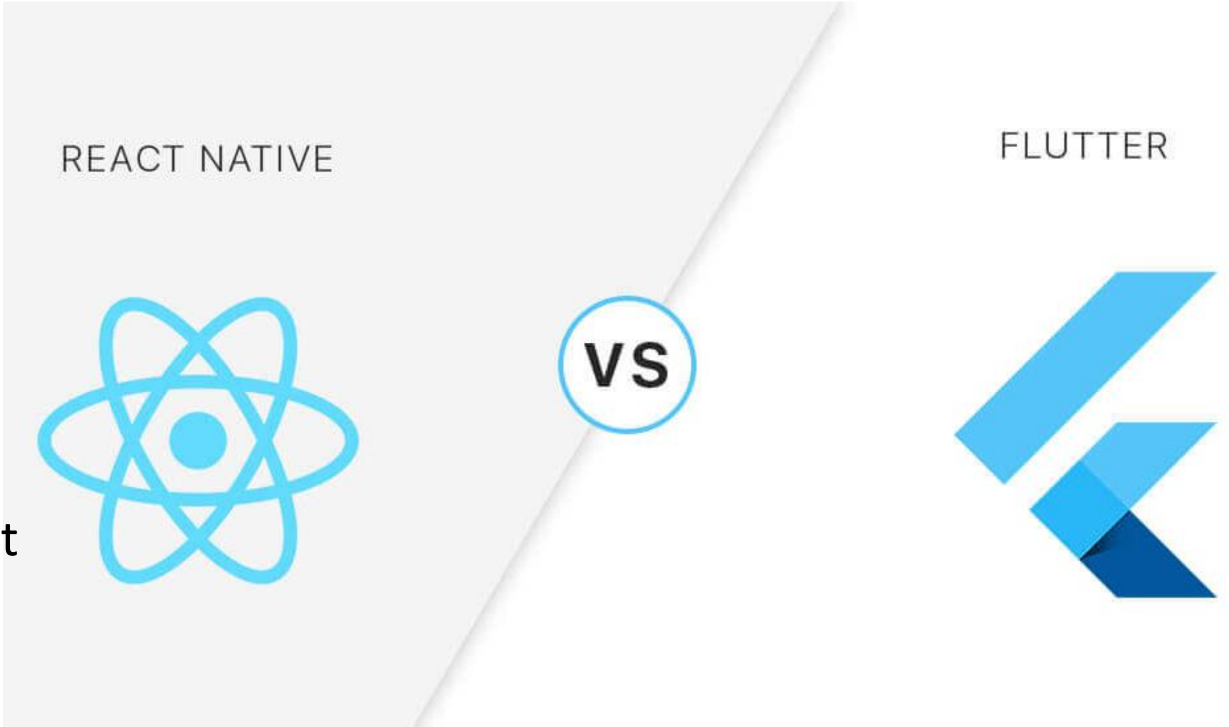
Why Flutter different from React Native?

- **React Native** is a JavaScript Library, designed for building genuinely **native apps** for platforms like iOS and Android. It's based on a **JavaScript library** created by Facebook called **React**, and thus brings its power to native mobile app development.

- But Flutter **doesn't use any of the tech stacks** which is most popular on the internet.

Comparison between Flutter & React Native

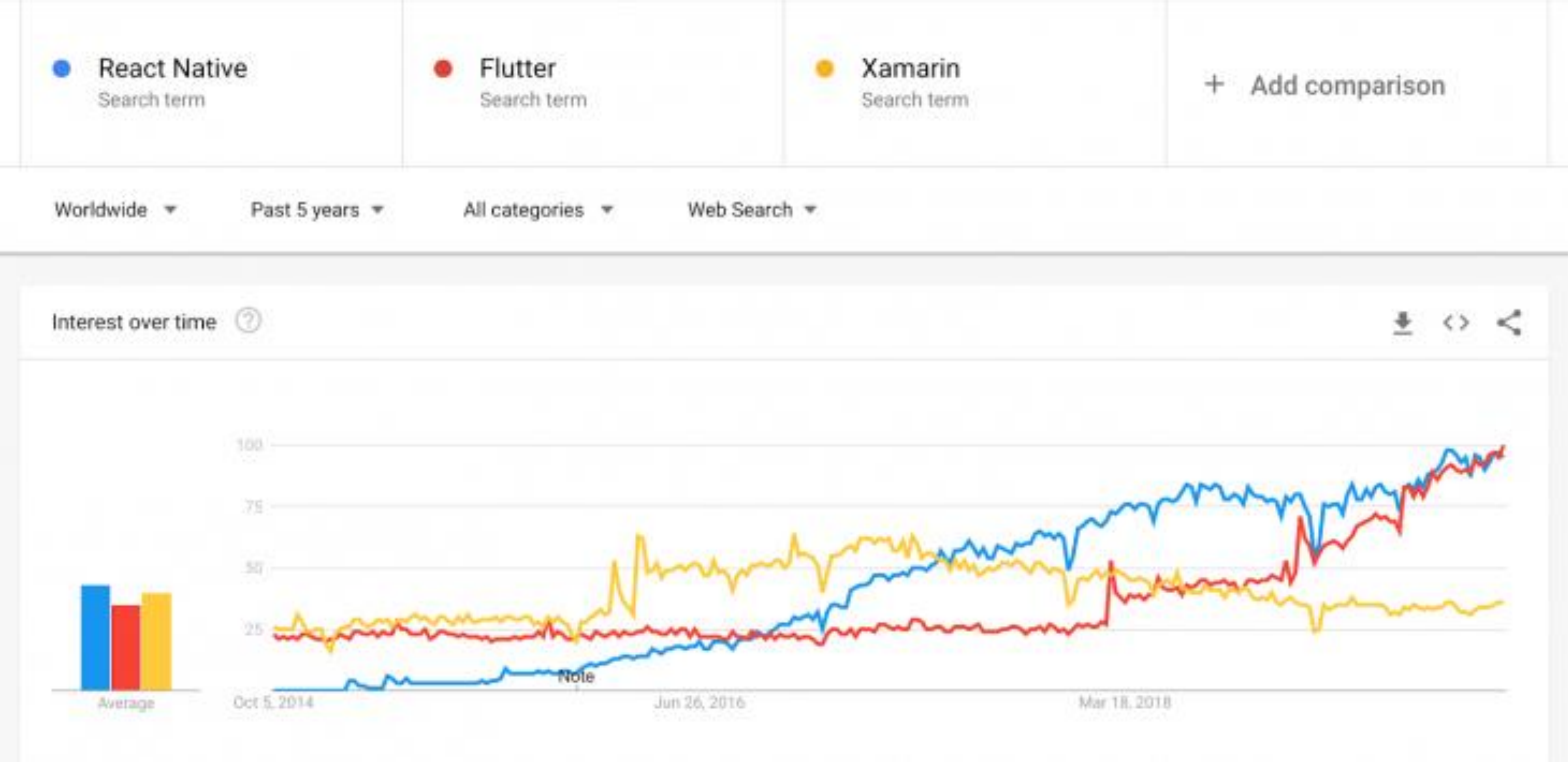
- Programming Languages
- Performance
- Architecture
- IDE and Ease of Coding
- Code Structure
- Code reusability
- Popularity
- Support and ecosystem/community support



Technology	React Native	Flutter
First Release	Jan, 2015	May 2017
Built By	Facebook	Google
Coding Languages	JavaScript	Dart
Github Stars (in 2019)	77,827	66,183
Learning curve	Easy to learn for React or JavaScript developers	“Hard for novices. But easy to learn for experienced C++ and JAVA experts
Main architecture	Flux & Redux	bLoC
Components	Some are adaptive automatically	Nonadaptive. Need to be configured manually
Development Speed	make an app with a very less turnaround time	Not much than React native

UI components	OS native components	Proprietary widgets
Documentation	Smooth documentation	Disorganized documentation
Hot Reload	Supported	Supported
Time to market	Slower than Flutter	Faster
Native Performance	Good	Best
Popular users	Facebook, Instagram, Pinterest, Tesla, Uber, Walmart, Wix.com	Alibaba, Google Greentea, GoogleAds, App tree, Tencent

Popularity



● React Native
Search term

● Flutter
Search term

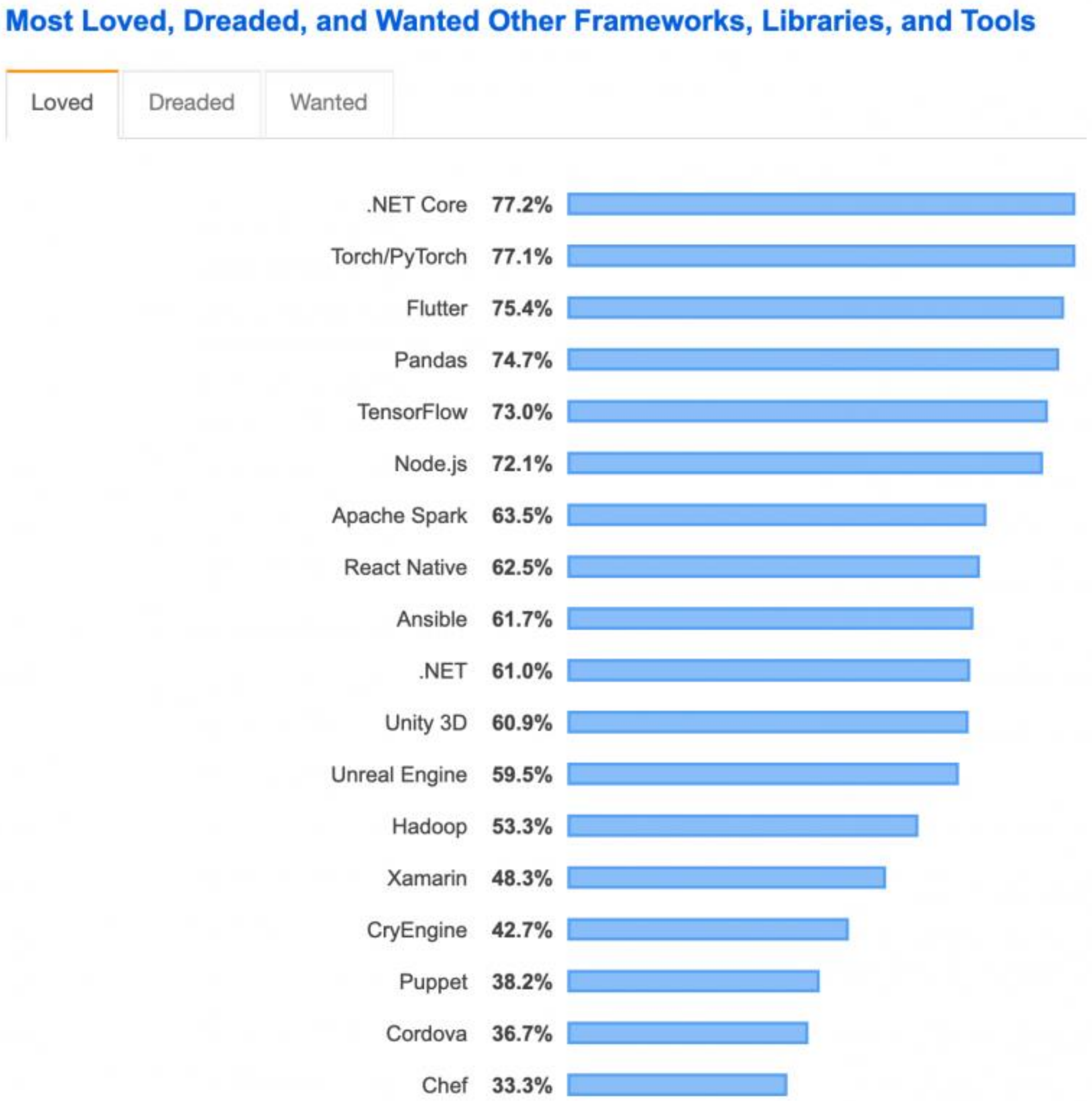
● Xamarin
Search term

+ Add comparison

Worldwide ▾ Past 5 years ▾ All categories ▾ Web Search ▾

Interest over time

According to **the Stack Overflow** Developer Survey **2019**, Flutter ranked as the most loved framework out of the three, with 75.4 percent of users expressing interest in continuing to develop with Flutter.



Most Loved, Dreaded, and Wanted Other Frameworks, Libraries, and Tools

Performance

- Flutter does not have a JavaScript bridge like React Native to interact with native components.
- This improves the performance in comparison to React Native which uses the JavaScript bridge.

Language

- React Native is written entirely in JavaScript using React.
- Flutter is written in a language called Dart.
- Although, Dart is relatively easy to pick up and is a good programming language, it does not have the popularity that JavaScript has.

Components

- React Native offers some pre-built and **partly adaptive** components like buttons and text inputs.
- Flutter provides a more extensive library of components, which are called widgets but they are **non-adaptive**.

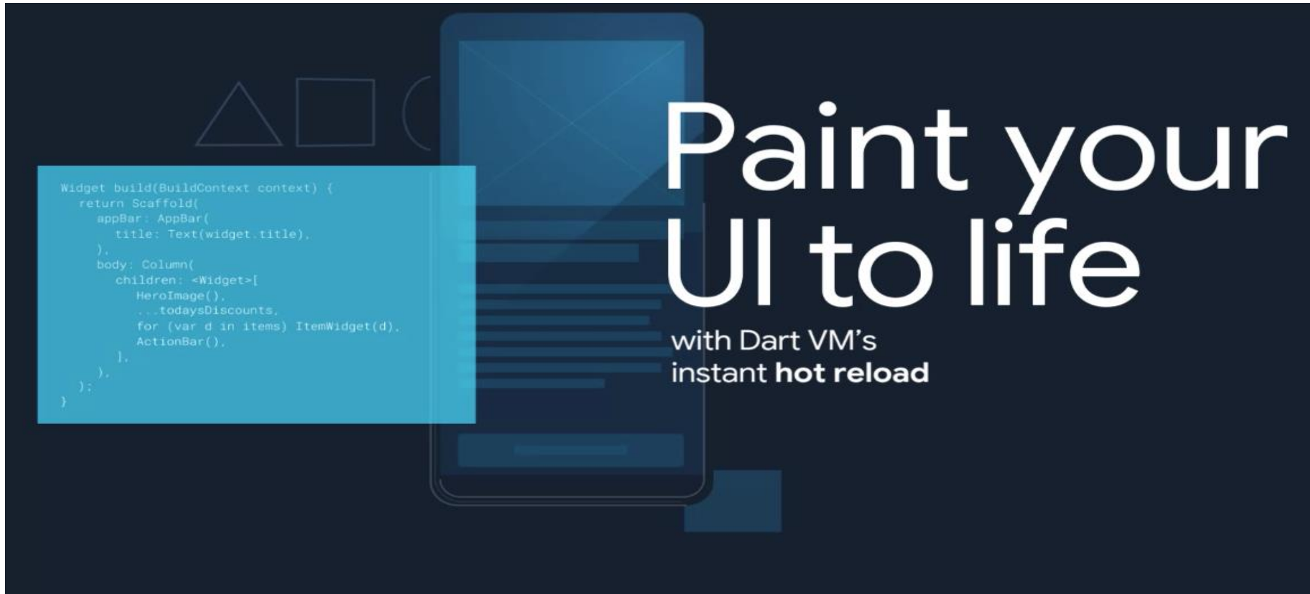
Code reuse

- React Native allows you to write the code once and ship anywhere, but it also embraces platform differences.
- Flutter’s codebase is **more reusable** as it allows you to define one UI widget tree and reuse the defined logic so you don’t have to do a lot of differentiation, which you can also still do if you need to.

Framework architecture

- **Dart platform**

- Dart is a language uniquely optimized for client-side development for web and mobile.
- Dart is AOT (Ahead Of Time) compiled to fast, predictable, native code, which allows almost all of Flutter to be written in Dart.
- This not only makes Flutter fast, virtually everything (including all the widgets) can be customized.



```
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(widget.title),
    ),
    body: Column(
      children: <Widget>[
        HeroImage(),
        ...todaysDiscounts,
        for (var d in items) ItemWidget(d),
        ActionBar(),
      ],
    ),
  );
}
```

Framework architecture Cont…

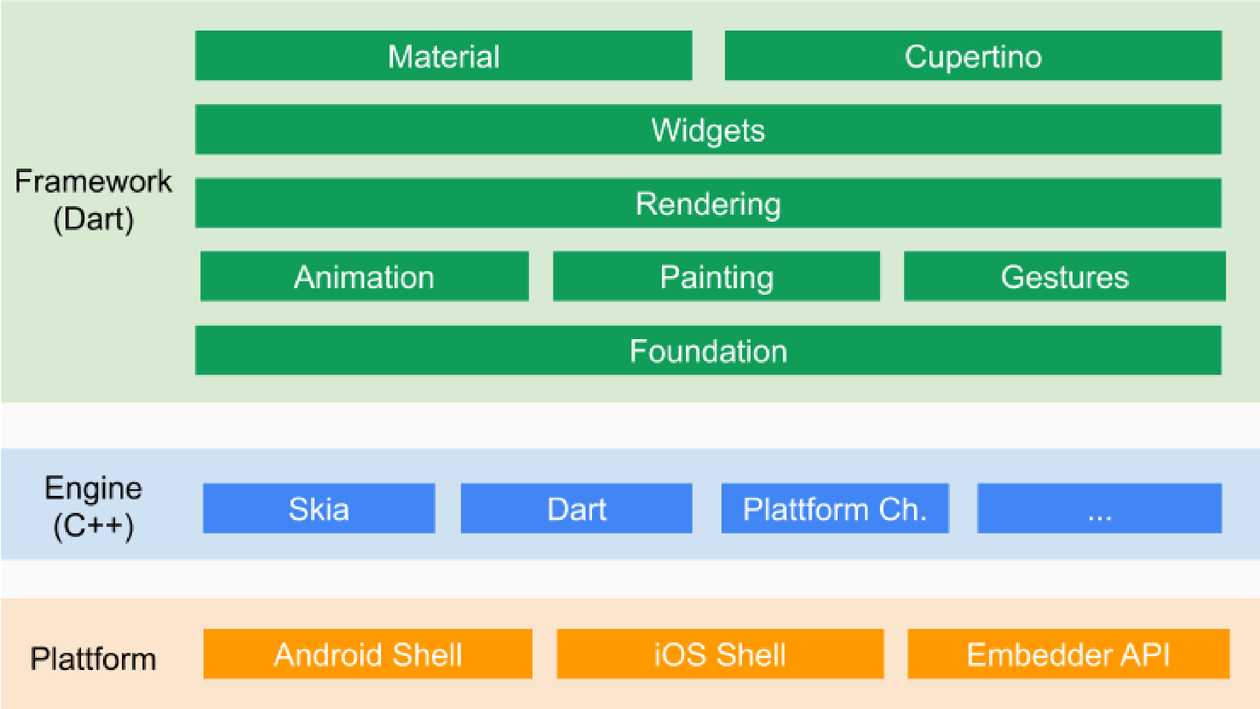
- **Flutter engine**

Flutter's engine takes core technologies,

Skia — a 2D graphics rendering library

Dart — a VM for a garbage-collected object-oriented language

Additionally, it interfaces with platform-specific SDKs such as those provided by Android and iOS



Framework architecture Cont…

- **Foundation library**

The Foundation library, written in Dart, provides basic classes and functions which are used to construct applications using Flutter, such as APIs to communicate with the engine

Framework architecture Cont…

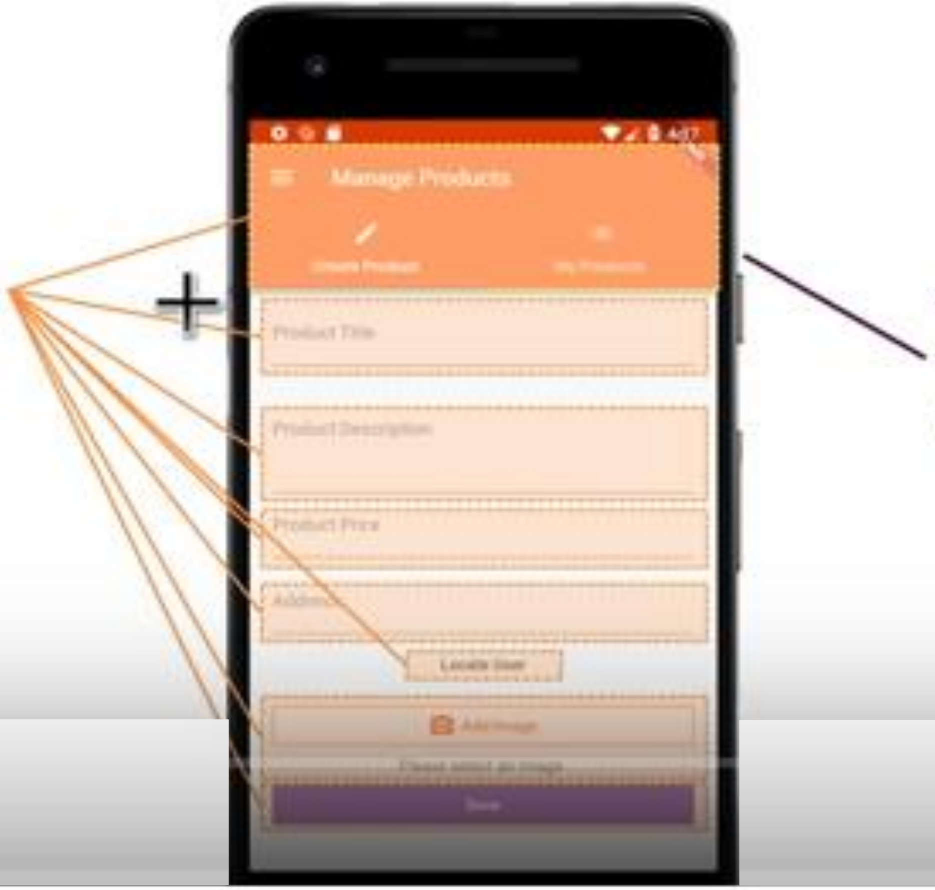
➢ **Widgets**

- In Flutter, the rough equivalent to a View is a Widget and they are fast, pretty and customizable.
- Flutter includes the Material Components library.
- These are widgets that implement the Material Design guidelines. Material Design is a flexible design system optimized for all platforms, including iOS.

❖ Widget catalog → https://flutter.dev/docs/development/ui/widgets

Everything's a Widget!

Widgets!



Actually, the whole page is a Widget!

Drawbacks and Limitations in Flutter?

- **Libraries & support**
  - Google support for Flutter is impressive, and there are many helpful libraries. But Flutter is still new and not every functionality which need can be found in these libraries.
  - So the developers would need to build them by themselves, which can be very time-consuming.

- **Continuous Integration support**
  - For now, Flutter it is not widely supported by CI platforms like Travis or Jenkins.
  - But There is a new CI/CD system for Flutter applications called **Codemagic** Codemagic https://docs.codemagic.io/

Drawbacks and Limitations in Flutter?

- **Large app size**
  - Flutter apps can be quite large compared to other frameworks
  - So developers may need to reduce the number of libraries and packages used, compress images, and even steer away from using animation altogether in favor of reducing their app’s size.

- **UI and logic is intermixed**
  - Modern framework tries to separate logic and UI as much as possible but, in Flutter, user interface and logic is intermixed. We can overcome this using smart coding and using high level module to separate user interface and logic.

Thank You