The Magic of Flutter

What is Flutter?

A SDK that makes building high-performing, modern and beautiful apps easy

Works for both Android and iOS

An open-source toolkit, developed by Google*

100+ contributions from the open source community

* Currently in Alpha

Who is Flutter for?

Designers that care about a brand-driven experience on Android and iOS

Prototypers get a high-fidelity and fast way to build working prototypes.

Developers benefit from fantastic developer tools, an easy-to-use language, a rich set of widgets and great IDE support. Flutter frees up valuable time for working on features and delightful experiences.

1. Developer Experience
2. Performance

Design-oriented Development Flow

What do you see here?

Oeschinen Lake Campground

Kandersteg, Switzerland

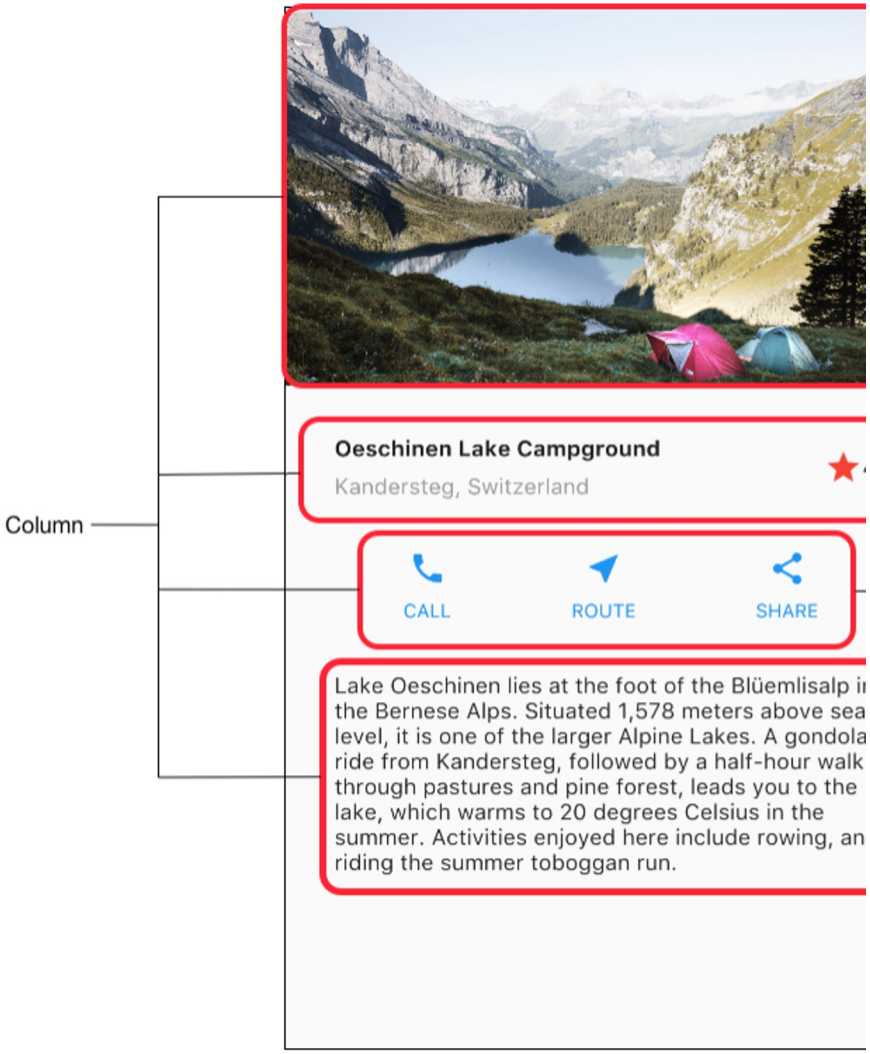
★ 41

CALL ROUTE SHARE

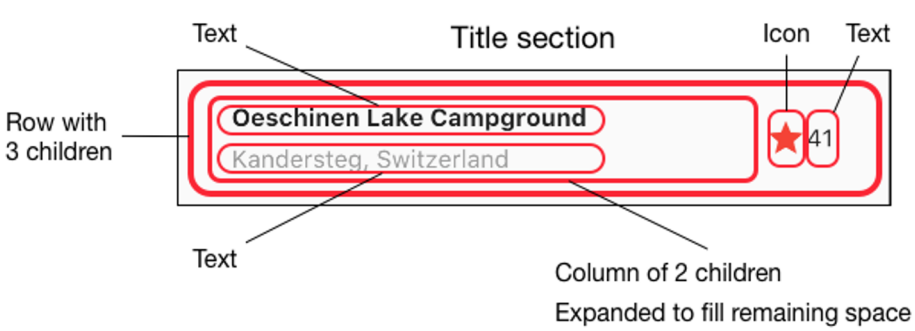
Lake Oeschinen lies at the foot of the Blüemlisalp in the Bernese Alps. Situated 1,578 meters above sea level, it is one of the larger Alpine Lakes. A gondola ride from Kandersteg, followed by a half-hour walk through pastures and pine forest, leads you to the lake, which warms to 20 degrees Celsius in the summer. Activities enjoyed here include rowing, and riding the summer toboggan run.

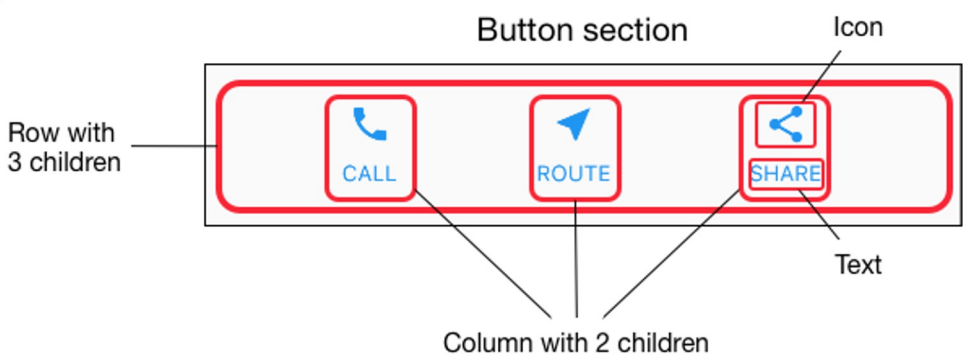
Diagram the Layout

- Look for rows and columns
- Is there a grid?
- Any overlapping elements?
- Do we need tabs?
- Padding, alignment or borders needed?



Designing bottom up





HTML/CSS Analogs in Flutter

```
<div class="greybox">
    Lorem ipsum
</div>

.greybox {
      background-color: #e0e0e0; /* grey 300 */
      width: 320px;
      height: 240px;
      font: 900 24px Georgia;
    }
```

```
var container = new Container( // grey box
  child: new Text(
    "Lorem ipsum",
    style: new TextStyle(
      fontSize: 24.0
      fontWeight: FontWeight.w900,
      fontFamily: "Georgia",
    ),
  ),
  width: 320.0,
  height: 240.0,
  color: Colors.grey[300],
);
```

Efficient Tooling

```
$ flutter doctor
```

Checks your environment and displays a report to the terminal window

```
$ flutter upgrade
```

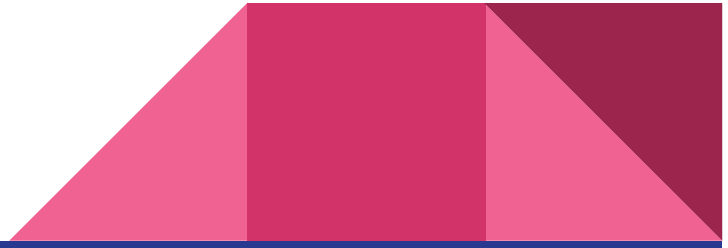
Updates both the Flutter SDK and your packages

pubspec.yaml

```
name: flutter_project

description: An amazing Flutter project using Firebase Auth

dependencies:
  flutter:
    sdk: flutter
  firebase_auth: "^0.1.2"
```



pubspec.yaml

```
name: flutter_project

description: An amazing Flutter project using Firebase Auth

dependencies:

  flutter:

    sdk: flutter

  firebase_auth: ">=0.1.2 <0.2.0"
```

```
$ flutter packages get
```

Checks your environment and displays a report to the terminal window

```
$ flutter packages upgrade
```

Will retrieve the highest available version of the package

```
$ flutter format
```

Automatically formats your code according to the Flutter-style

```
$ flutter analyze
```

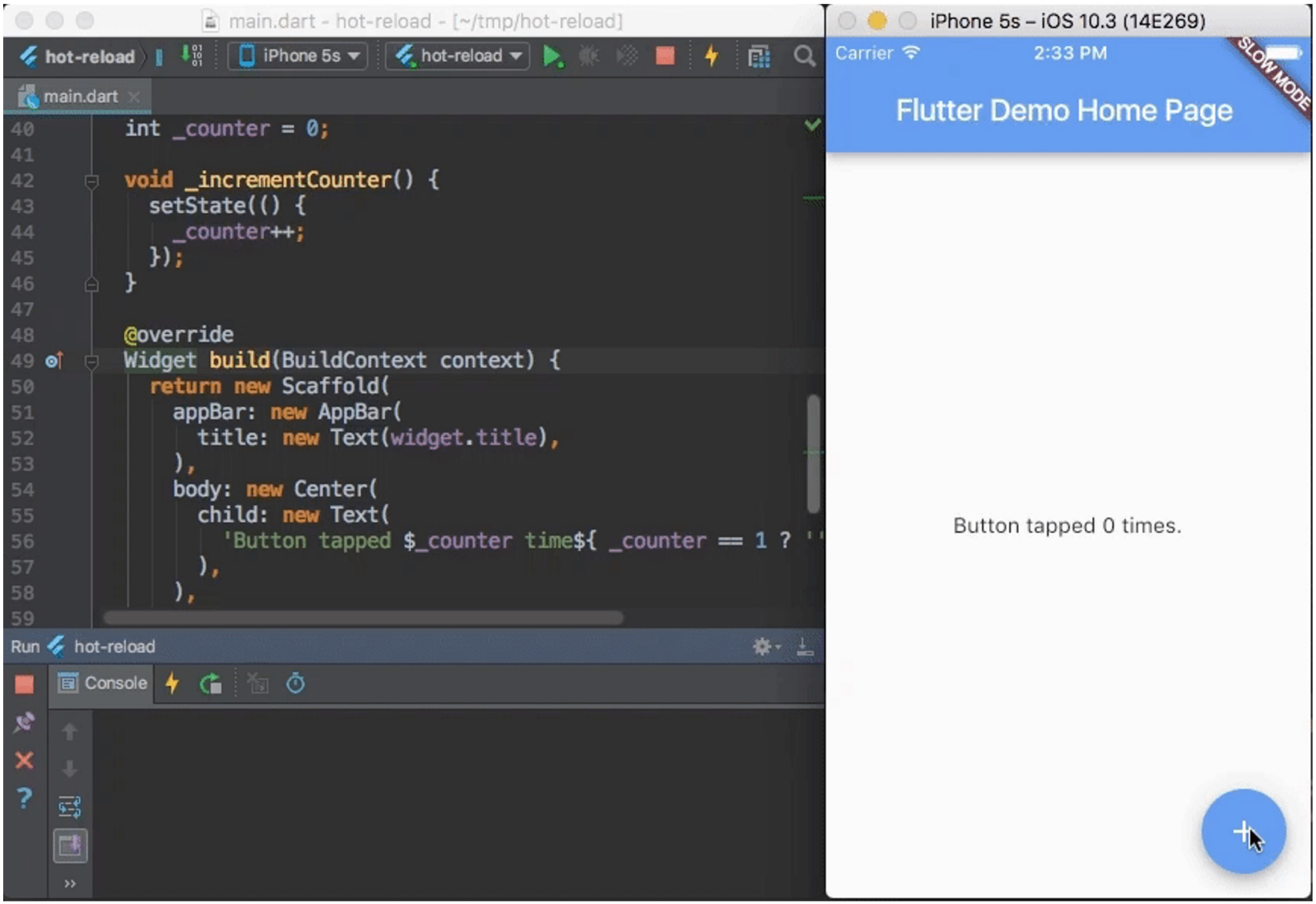
Analyzes your code and help you find possible mistakes

Hot Reload

Injecting updated source code files into the running Dart VM

Stateful: App state is retained after a reload.

Quickly iterate on a screen deeply nested in your app


main.dart - hot-reload - [~/tmp/hot-reload]
hot-reload
iPhone 5s
hot-reload
main.dart
int _counter = 0;
void _incrementCounter() {
setState(() {
_counter++;
});
}
@override
Widget build(BuildContext context) {
return new Scaffold(
appBar: new AppBar(
title: new Text(widget.title),
),
body: new Center(
child: new Text(
'Button tapped $_counter time${ _counter == 1 ?
),
),
Run
hot-reload
Console
iPhone 5s – iOS 10.3 (14E269)
Carrier
2:33 PM
SLOW MODE
Flutter Demo Home Page
Button tapped 0 times.

Dart Observatory

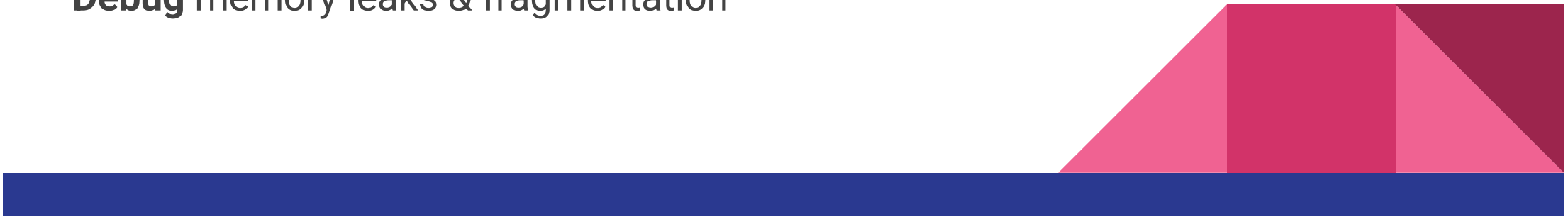
Statement-level single-stepping debugger and profiler

Automatically running when you start your app using `flutter run`

See which lines of code have executed

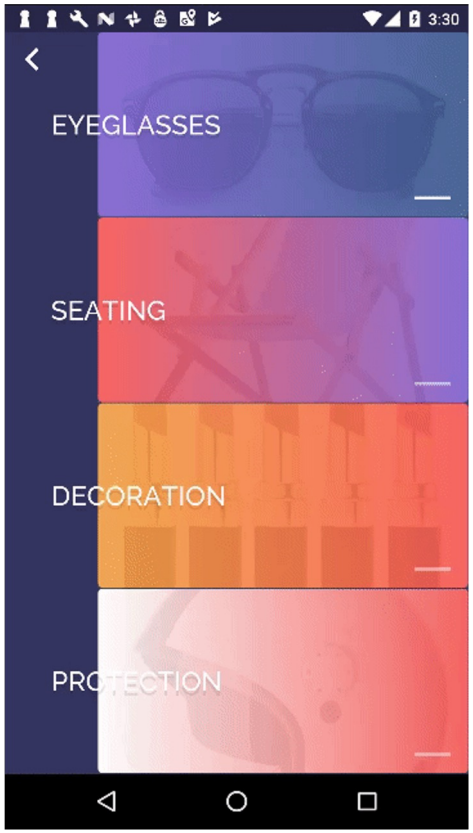
Check out memory allocations

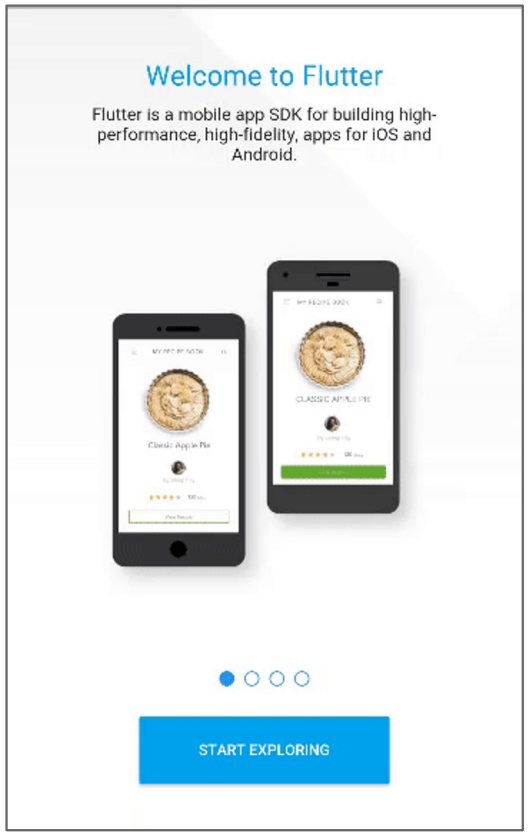
Debug memory leaks & fragmentation

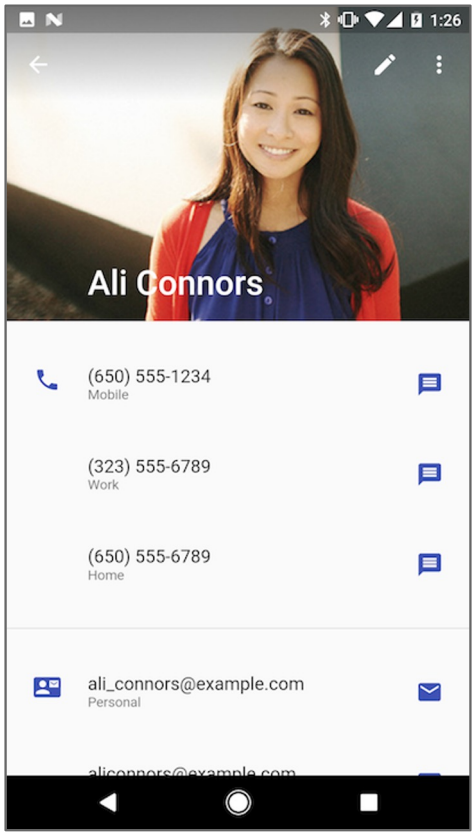


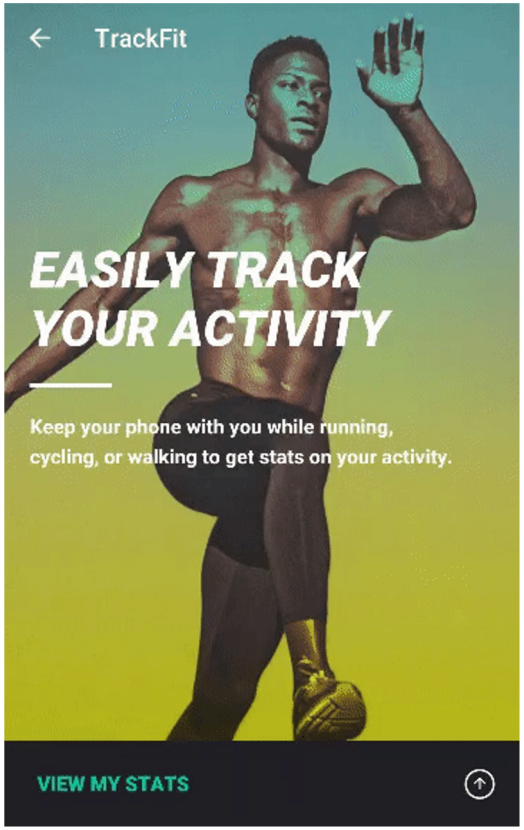
The Power of Widgets

Great looking and fast Widgets

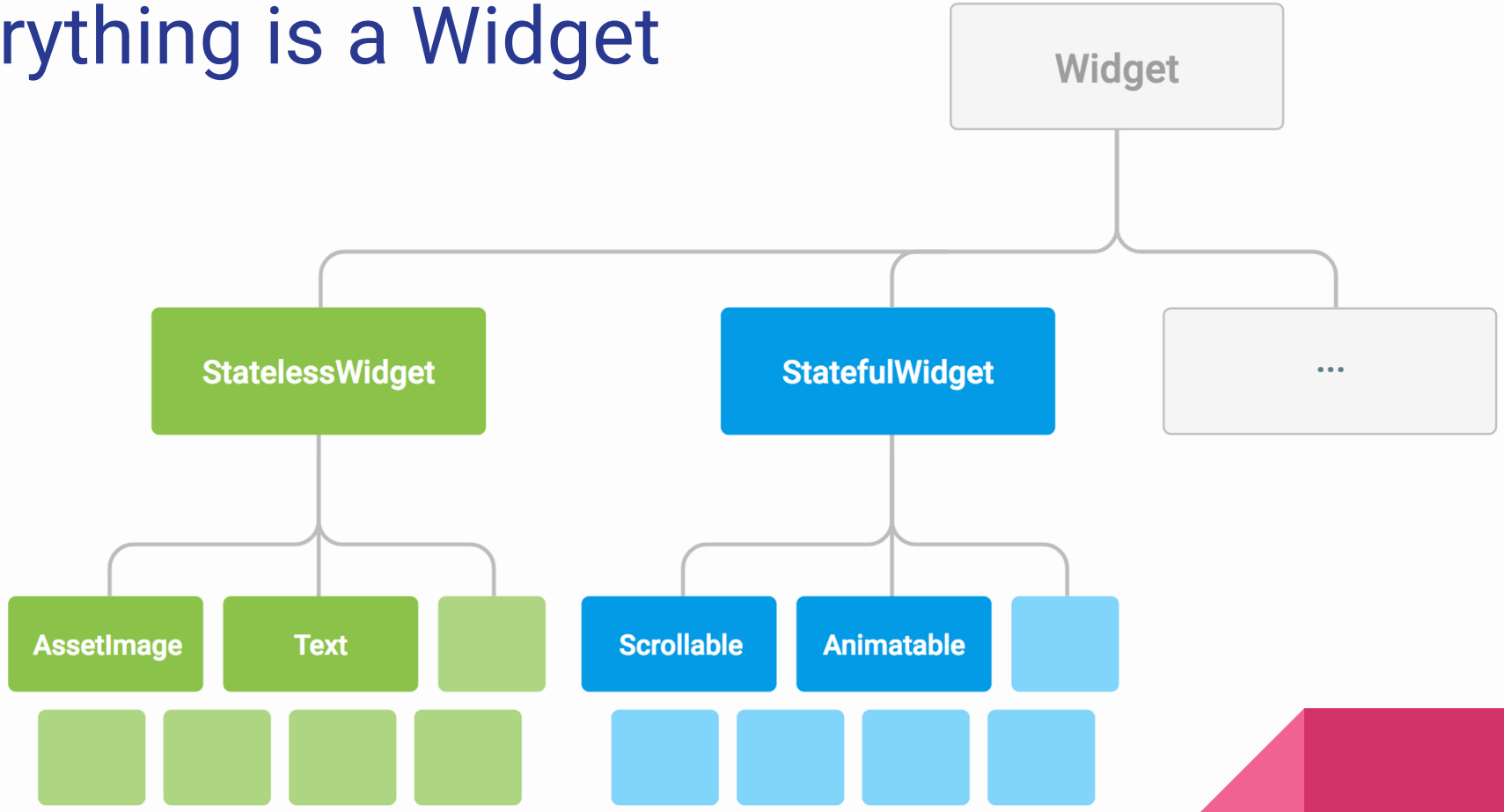








Everything is a Widget



Goodbye, global layout system

```
new Center(
    child: new Text('Centered Text', style: textStyle),
)
```

Local layouts: Every Widget defines it’s **own** layout. No need to tell the parent that it’s children are supposed to be centered.

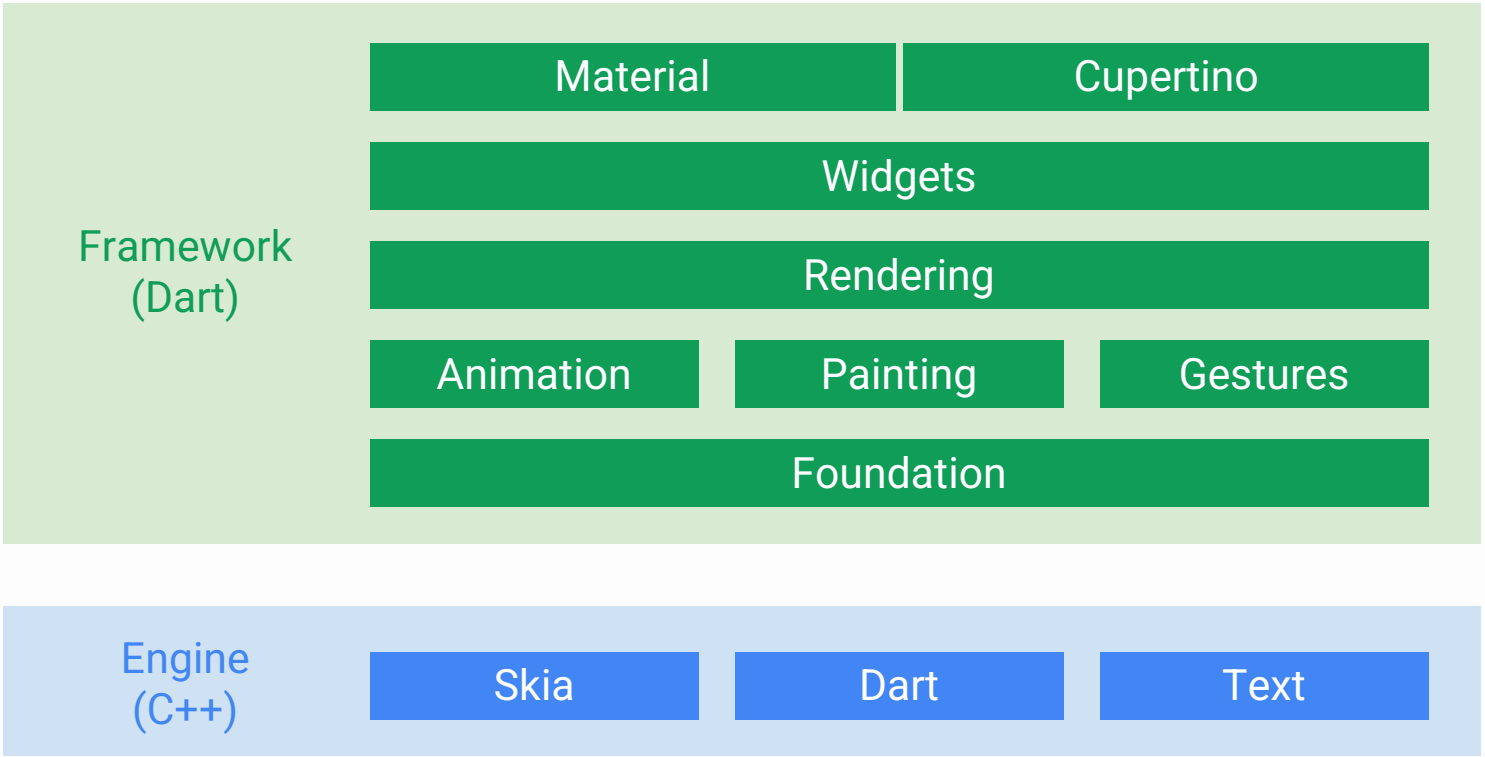
StatefulWidget
vs.
StatelessWidget

Customizing and extending Widgets

Flutter’s Widget system was designed to be easily customizable

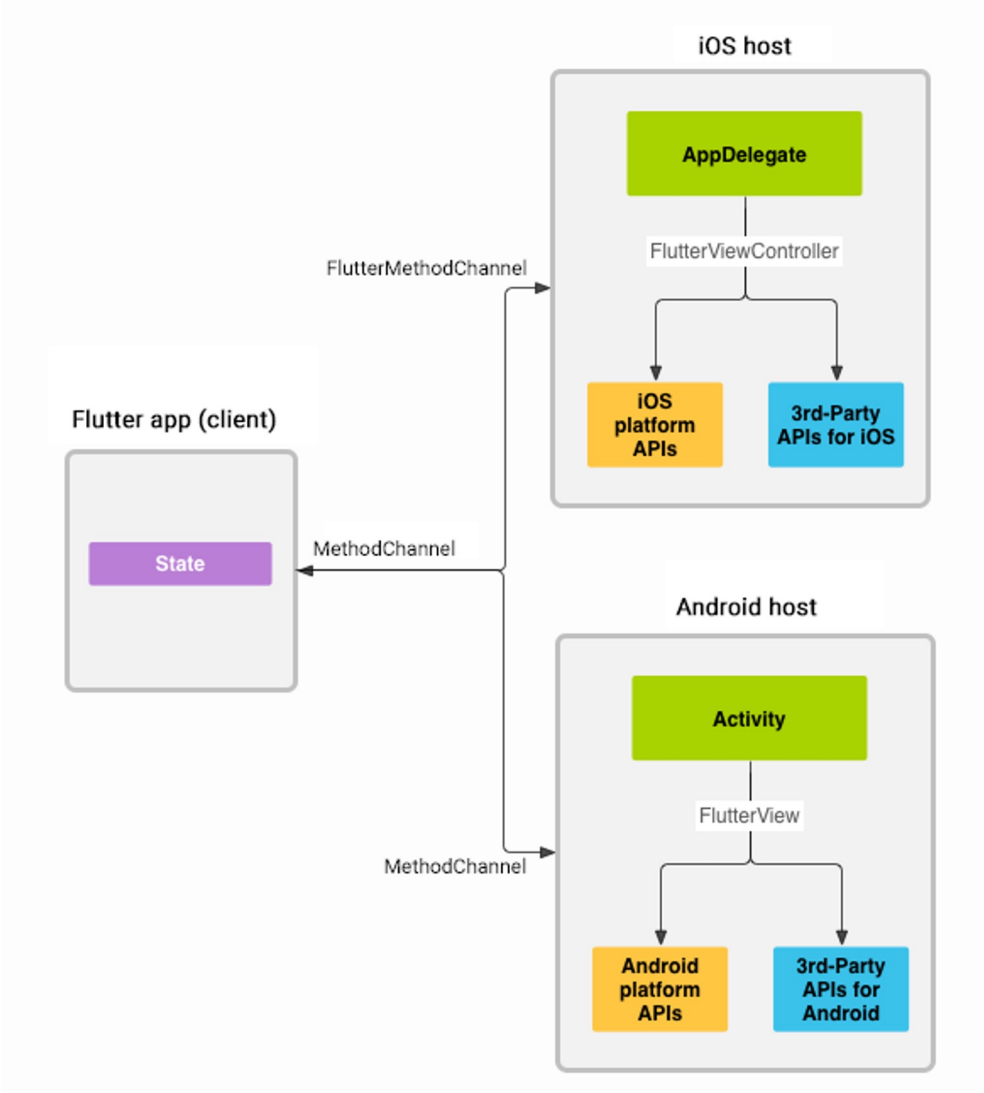
Composition: Widgets are built out of smaller widgets that you can reuse and combine in novel ways to make custom widgets

```
class RaisedButton extends StatelessWidget {
    ...
}
```



Each layer builds upon the previous layer

Platform Channels



Using platform channels allows for receiving method calls and sending back results

Example: Retrieving the battery level*

```
class MainActivity() : FlutterActivity() {
  private val CHANNEL = "samples.flutter.io/battery"

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    GeneratedPluginRegistrant.registerWith(this)

    MethodChannel(flutterView, CHANNEL).setMethodCallHandler { call, result ->
      // TODO
    }
  }
}
```

* Example written in Kotlin for Android

Working with the response argument*

```
MethodChannel(flutterView, CHANNEL).setMethodCallHandler { call, result ->
  if (call.method == "getBatteryLevel") {
    val batteryLevel = getBatteryLevel()
      if (batteryLevel != -1) {
        result.success(batteryLevel)
      } else {
        result.error("UNAVAILABLE", "Battery level not available.", null)
      }
  } else {
    result.notImplemented()
  }
}
```

* Example written in Kotlin for Android

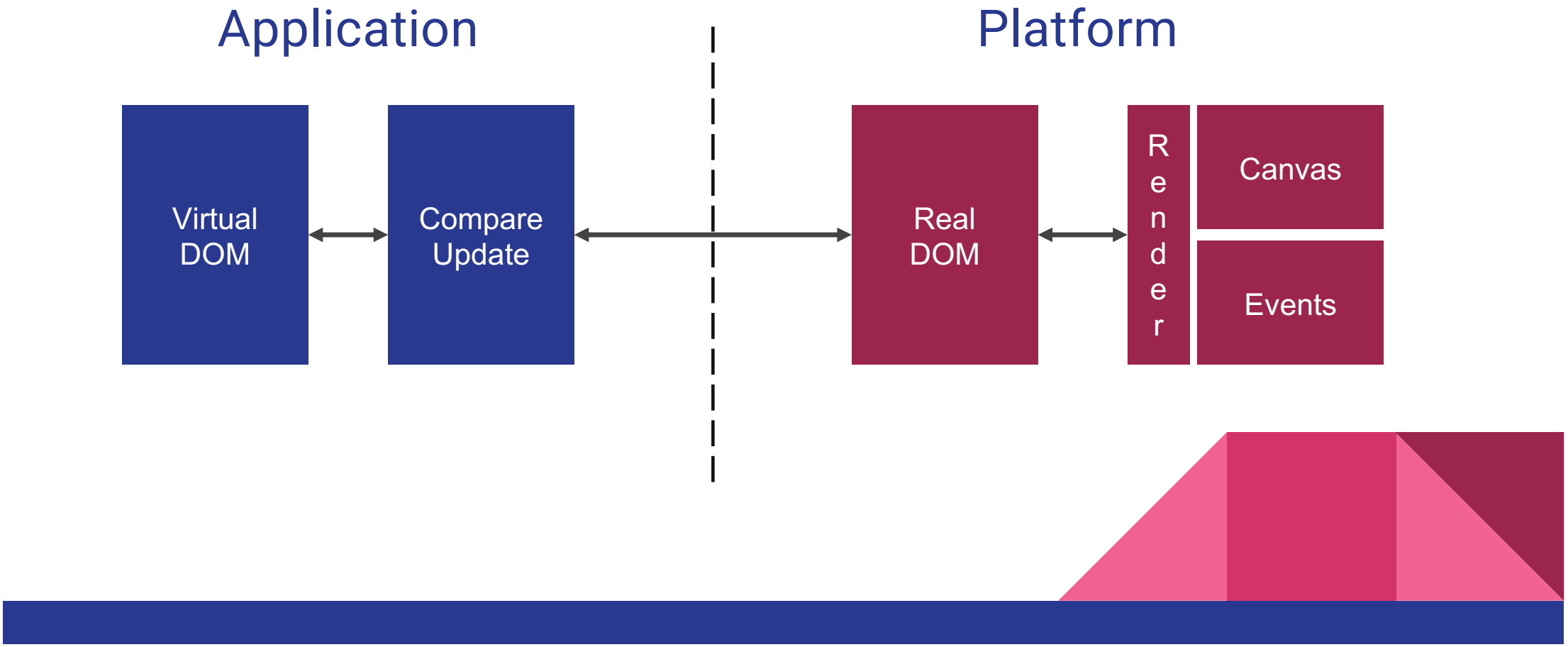
Flutter-side invocation of platform methods

```
String _batteryLevel = 'Unknown battery level.';
Future<Null> _getBatteryLevel() async {
  String batteryLevel;
  try {
    final int result = await platform.invokeMethod('getBatteryLevel');
    batteryLevel = 'Battery level at $result % .';
  } on PlatformException catch (e) {
    batteryLevel = "Failed to get battery level: '${e.message}'.";
  }
  setState(() {
    _batteryLevel = batteryLevel;
  });
}
```

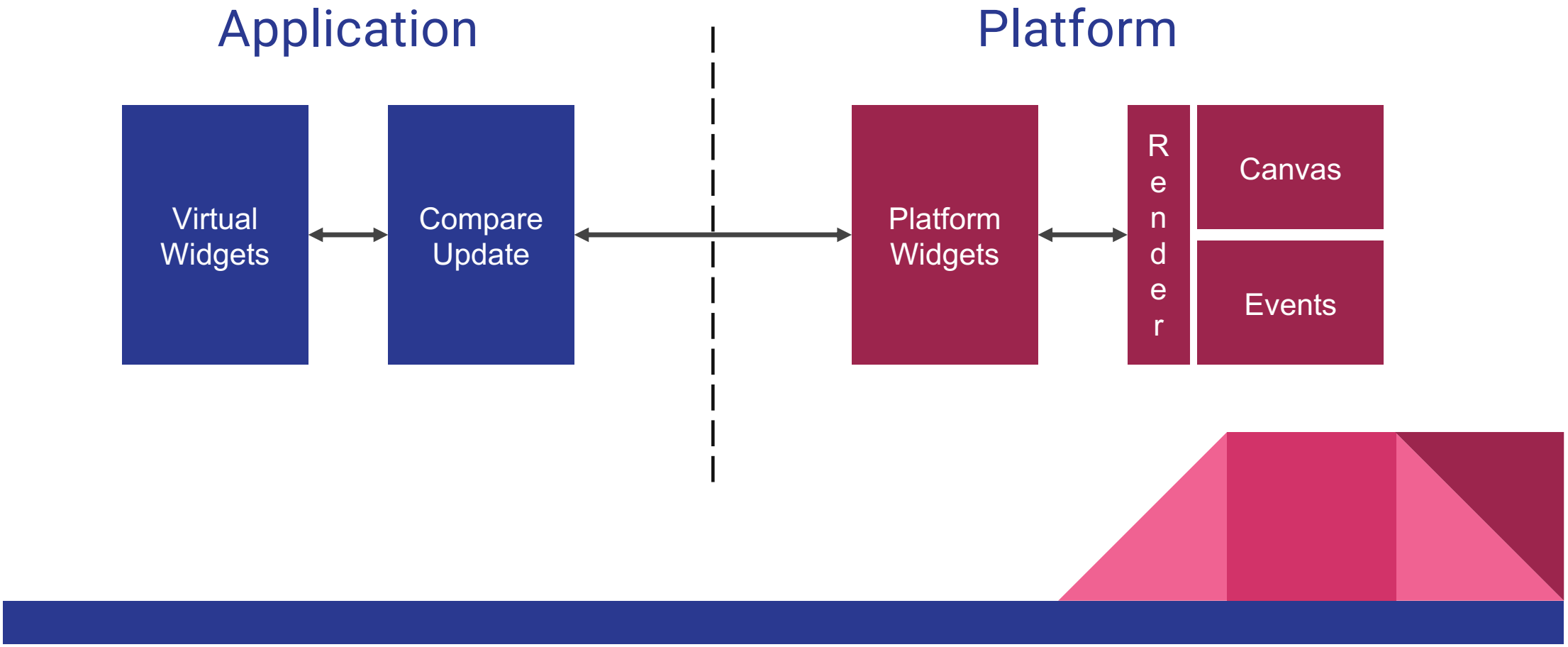
Optimized for Performance

- Compiles to Native Code
- No reliance on OEM widgets
- No bridge needed
- Structural Repainting

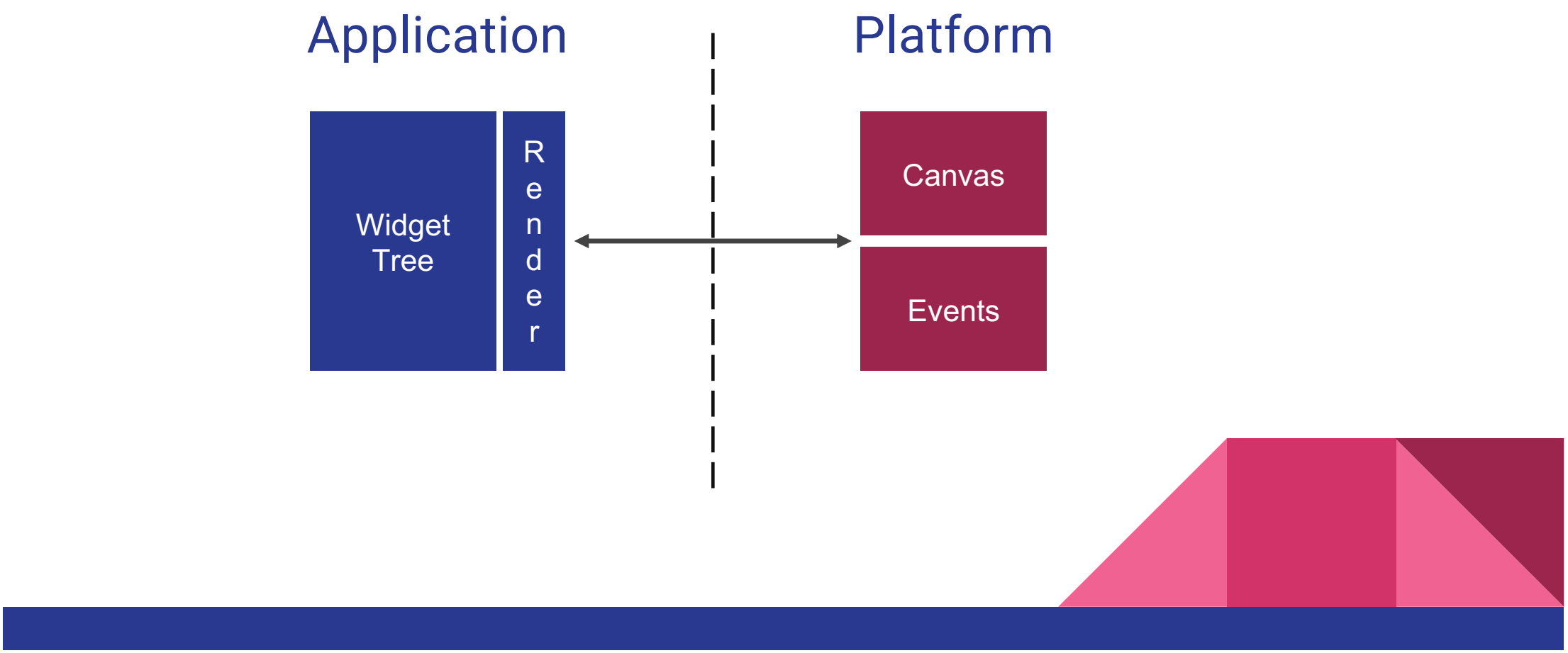
Reactive Frameworks on the Web



Reactive Frameworks on Mobile



Using Flutter



Superpowered by Dart

- A sound type system
- Tree Shaking
- Rich core libraries
- Multi-gen, lockless GC

- A single codebase for Android and iOS
- Rapid development cycles
- Great tooling

“Running at 60 fps, user interfaces created with Flutter perform far better than those created with other cross-platform development frameworks.”

code.tutsplus.com/tutorials/developing-an-android-app-with-flutter--cms-28270

“Coding with Dart and Flutter rekindled the joy I had when I started with mobile dev way back when … No B.S.”

traversoft.com/blog/2017/08/08/conference-app-flutter

"The UI is butter smooth (when building a release version), I have never seen such a smooth Android app"

Pascal Welsch, Speaker at Droidcon Berlin

Additional resources

Blog What's Revolutionary about Flutter by Wm Leler: goo.gl/bZcFR9

Video Flutter's Rendering Pipeline by Adam Barth: youtu.be/UUfXWzp0-DU

Video The Mahogany Staircase by Ian Hickson: youtu.be/dkyY9WCGMi0

And of course: github.com/flutter & flutter.io