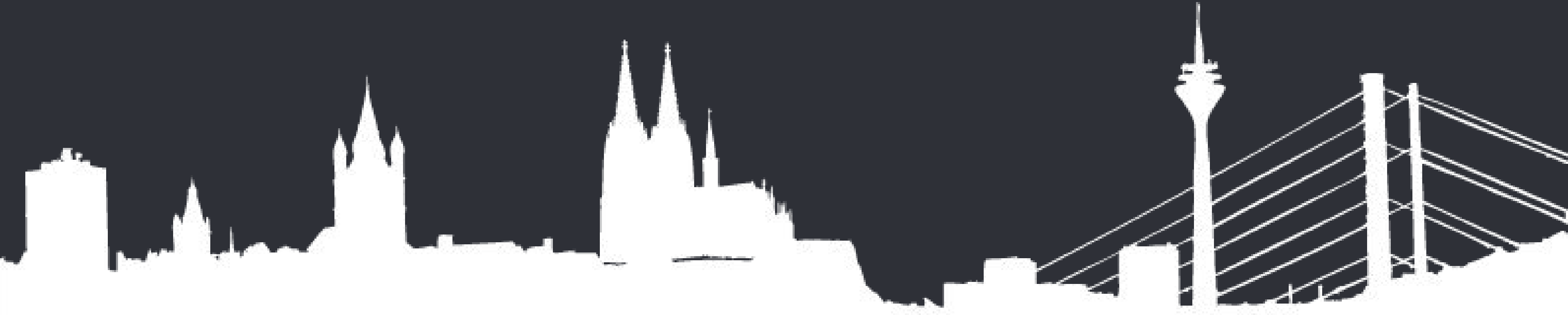
Flutter

60 FPS UI of the Future

grandcentrix.net
@grandcentrix

+Albrecht Noll
@UhrArt

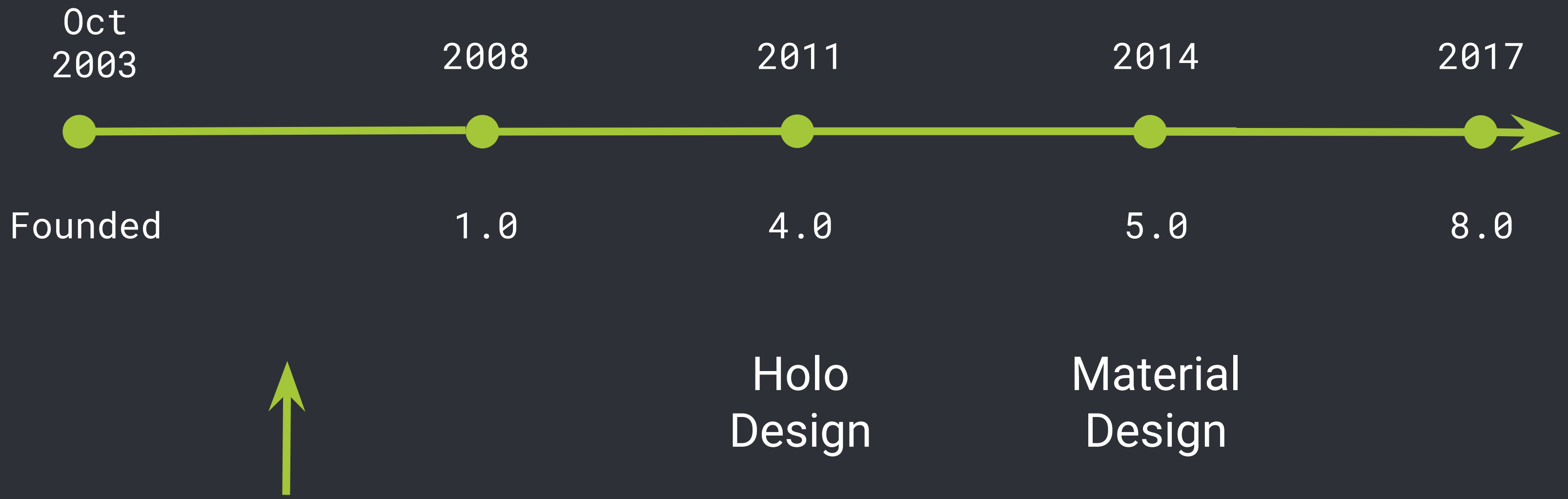
+Pascal Welsch
@passsy



Agenda

- Android (facts and opinions)
- Flutter (facts)
- Dart (code and opinions)
- Flutter (opinions)
- Fuchsia (speculations)

Android History



Android View
API Design
(View.java)

UI Bugfixes and Improvements

- Project Butter
- RecyclerView
- Design support library
- Instant Run
- Databinding in XML layouts
- Vector Drawables
- ...and thousands small fixes every release

My smartphone is lagging

- Every Android user ‘17

Android UI Framework

- >10 years old
- The Java API hasn't seen major changes
- No architectural changes, we are still using `android.view` to render our UIs

- Feels old
  - XML still "best practice"
  - No virtual dom

The entire UI architecture is wrong from the start.

Erik Hellman
@ErikHellman

Flutter

120 ~~60~~ FPS UI of the Future

What is Flutter?

flutter.io

- A mobile app SDK containing
  - complete UI Framework for mobile apps
  - huge Widget catalog
  - Tools
- Allows building beautiful mobile apps
- Platform independent, currently supporting Android, iOS and Fuchsia
- Uses Dart - Easy to learn language by Google

Flutters goals

- Beautiful fluid UIs
- Run same UI on multiple platforms, perfect for brand-first designs
- high-performance apps that feel natural on different platforms
- Be productive

flutter.io

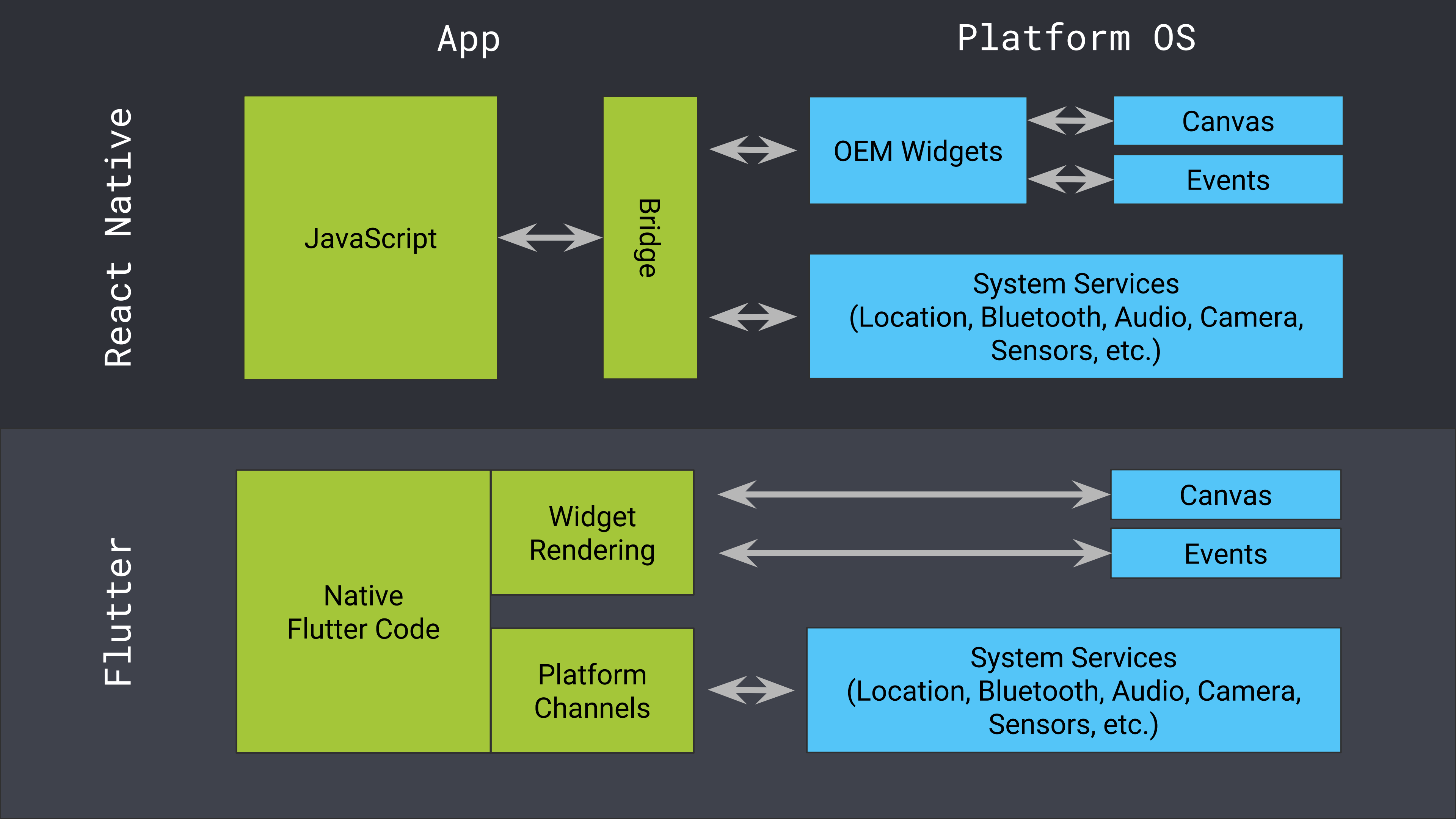
Flutter highlights

- Super performant, 120fps without optimizations
- Fast development - Hot Reload
- Modern, reactive framework like React

flutter.io

Flutter is not yet another Cross-Platform SDK

- Engine is shipped in apk (∽7.5 Mb)
- No bridge needed, direct drawing to platform canvas
- Doesn’t use OEM widgets
- Ships SDK with the app, no fragmentation or compatibility issues


App
Platform OS
React Native
JavaScript
Bridge
OEM Widgets
Canvas
Events
System Services
(Location, Bluetooth, Audio, Camera, Sensors, etc.)
Flutter
Native Flutter Code
Widget Rendering
Platform Channels
Canvas
Events
System Services
(Location, Bluetooth, Audio, Camera, Sensors, etc.)

What is Dart?

dartlang.org

- Java like language - easy to learn
- aimed to replace Javascript (2010)
- DartVM
- Javascript compiler (dart2js)

- Great language compared to Javascript and Java 6
- Missing syntactical sugar from Kotlin

Dart 2.0

dartlang.org

- 2.0 is coming soon™
- Will be sound (type safe)
- Will most likely get nullable types
- “new” could become optional
- Language discussions are available in the sdk repository
- https://github.com/dart-lang/sdk/tree/master/docs
  - Weekly newsletter (6 weeks in a row)
  - informal specifications

First Steps - 5 min Setup

- Clone repo and add to $PATH:

```
$ git clone -b alpha https://github.com/flutter/flutter.git
$ export PATH=`pwd`/flutter/bin:$PATH
```

- Run flutter doctor and do the suggested tasks

```
$ flutter doctor
```

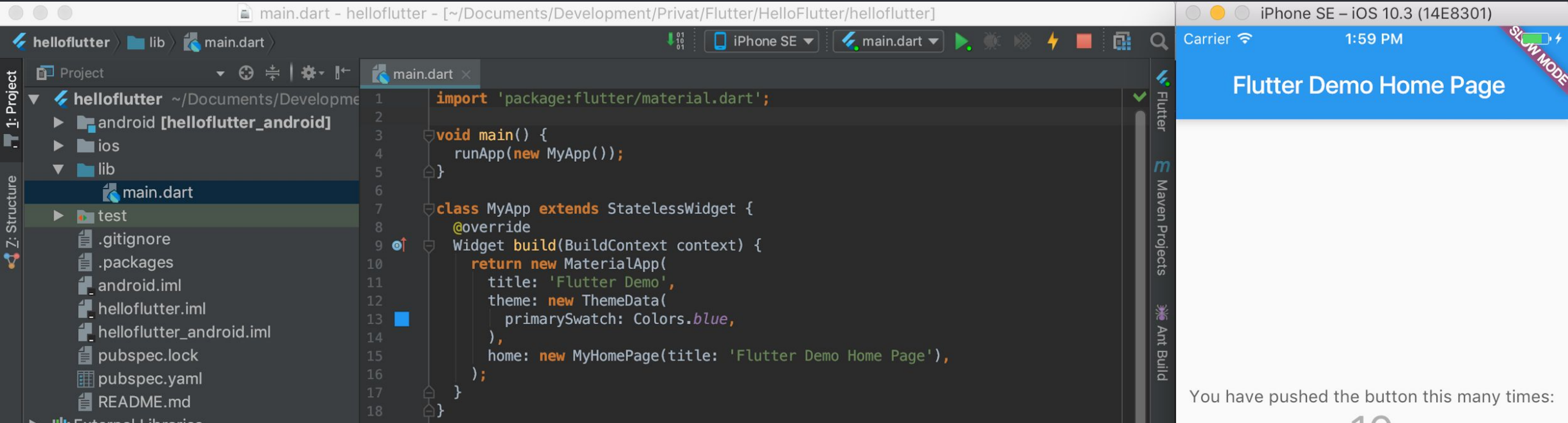
- Start developing

First Steps - Hello Flutter

- Create a new project

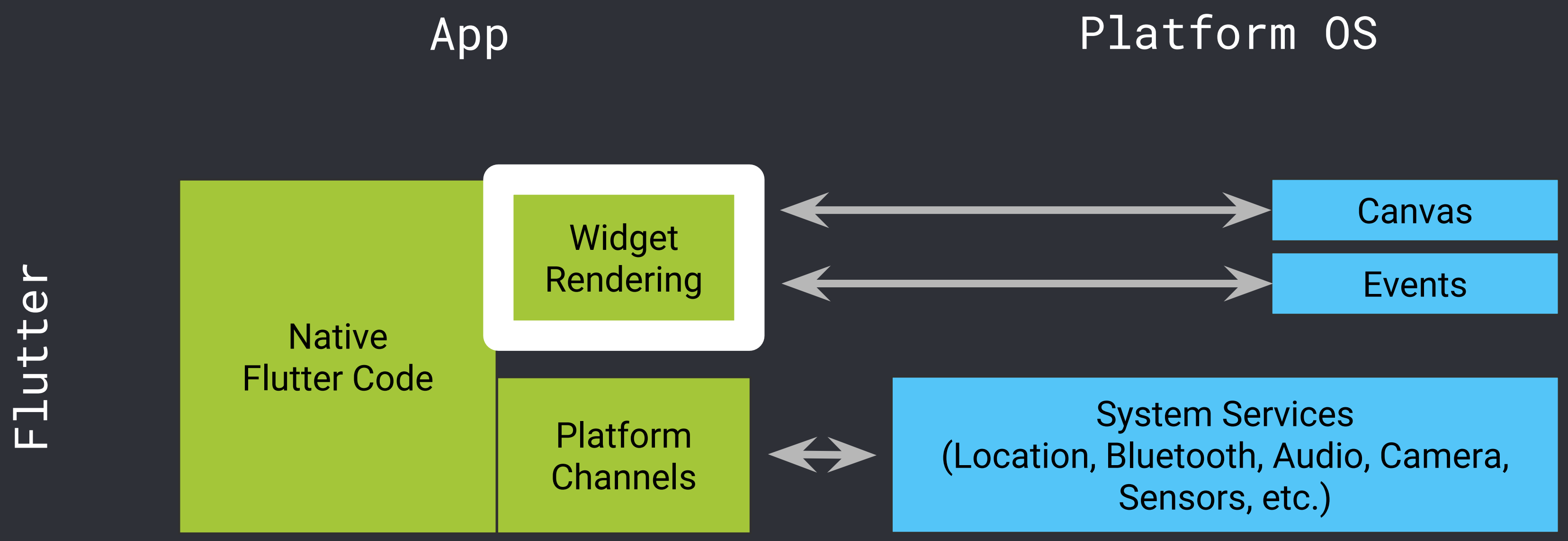
```
$ flutter create myapp
```

- Or use the Project Wizard in IntelliJ IDEA



```dart
import 'package:flutter/material.dart';

void main() {
  runApp(new MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new MaterialApp(
      title: 'Flutter Demo',
      theme: new ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: new MyHomePage(title: 'Flutter Demo Home Page'),
    );
  }
}
```

Widget Rendering

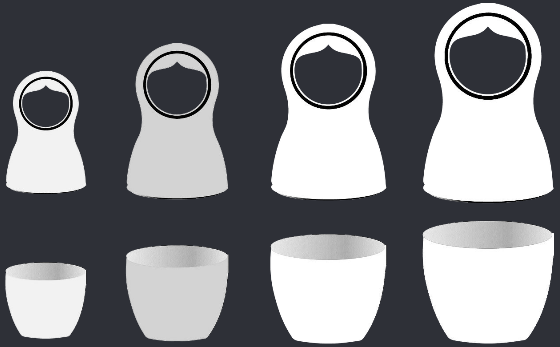


What are Widgets?

- Widgets are immutable declarations of parts of the UI
- Like a `<div/>`
- a structural element
  (e.g. button, menu)
- a stylistic element
  (themes, styles, fonts)
- an aspect of **layout**
  (padding, center)

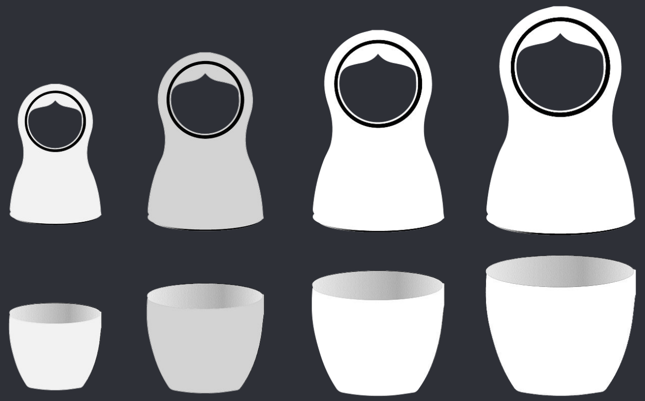
```dart
class PaddedText extends StatelessWidget {

 final String _data;

 PaddedText(this._data, {Key key})
      : super(key: key);

 @override
 Widget build(BuildContext context) {
   return new Padding(
         padding: const EdgeInsets.all(4.0),
         child: new Text(_data)
   );
 }
}
```

Everything is a Widget



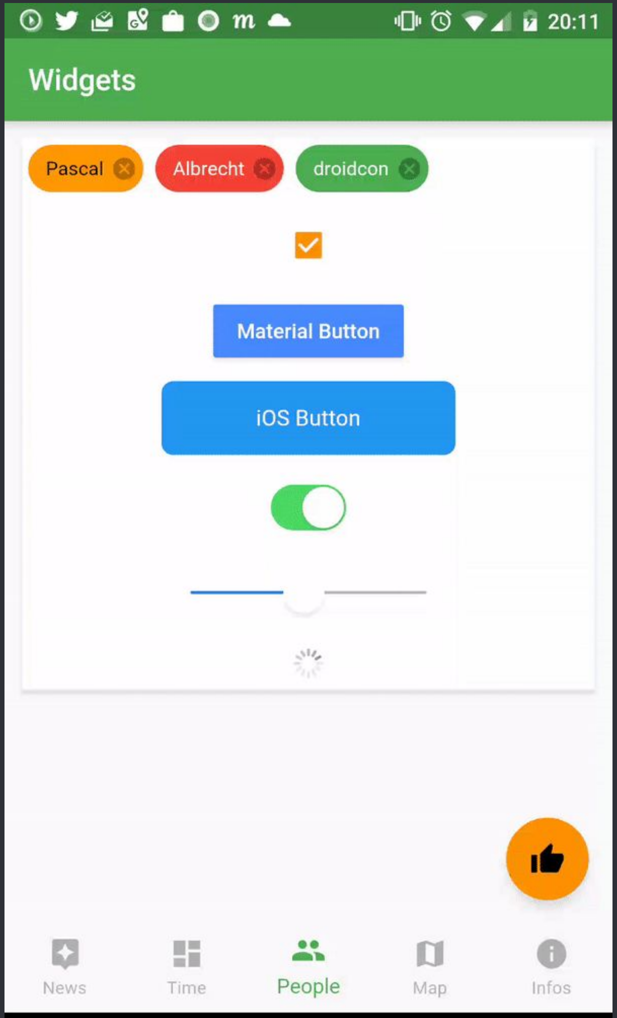
Everything is a Widget

- Application itself is a widget
- Hierarchically stacked
- inherit parent properties
- **Composition > inheritance**

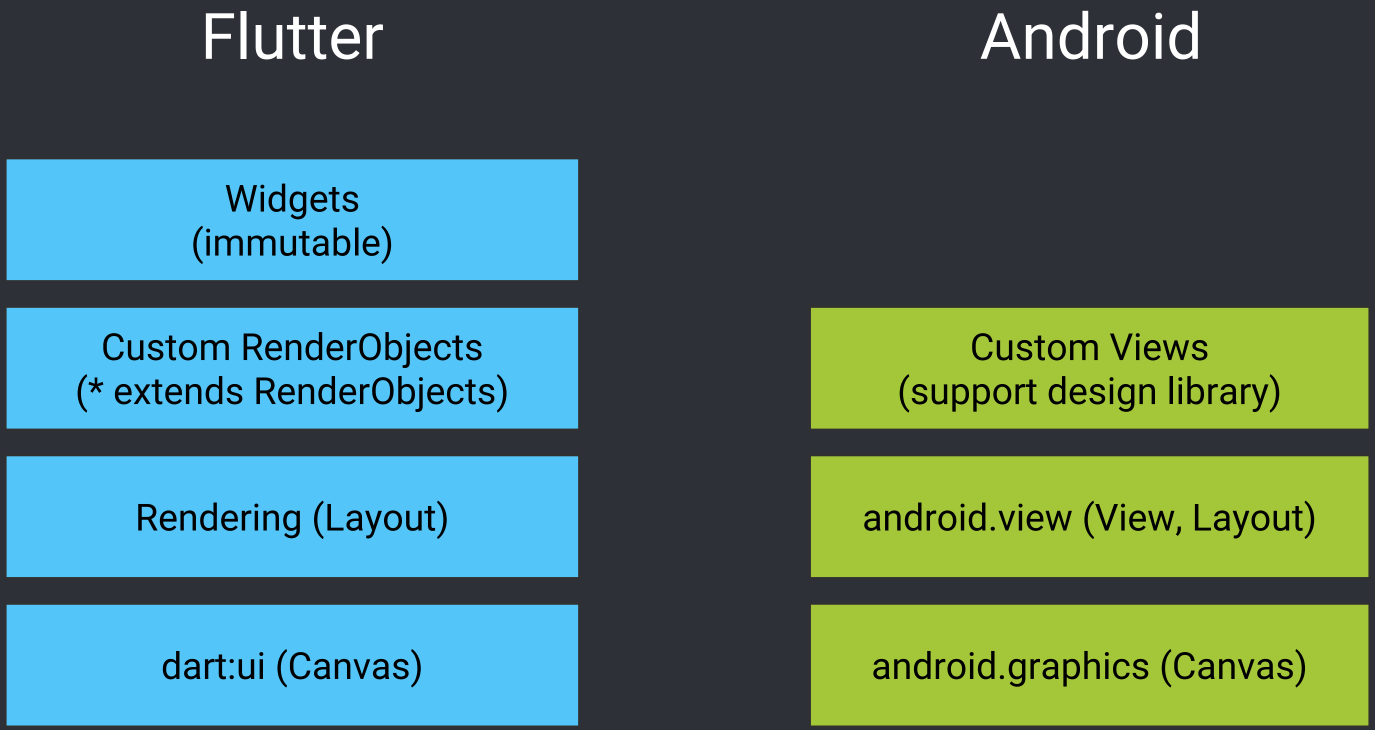


Existing Widgets

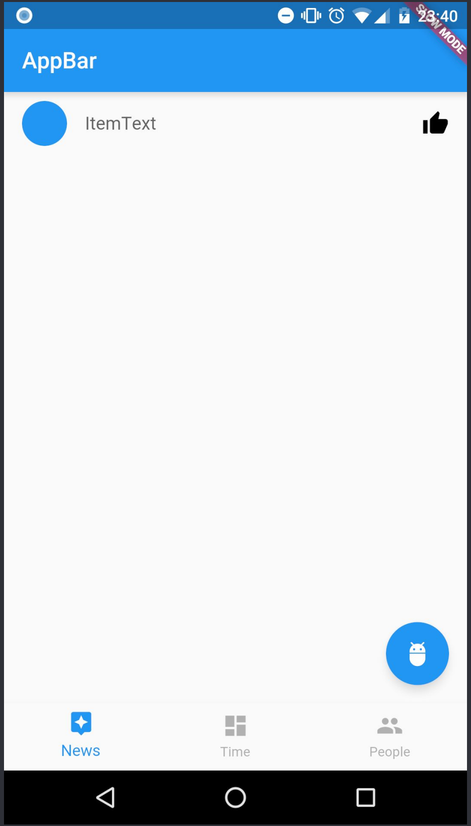
- **Material Guidelines** fully covered by Material Package
- **Human Interface Guidelines** iOS covered by Cupertino Package
- Premium Flutter Documentation



Flutter layered UI Architecture


Flutter
Android
Widgets (immutable)
Custom RenderObjects (* extends RenderObjects)
Rendering (Layout)
dart:ui (Canvas)
Custom Views (support design library)
android.view (View, Layout)
android.graphics (Canvas)

Important Material Widgets



```
new Scaffold(
 appBar: new AppBar(title: new Text('AppBar')),
 body: new ListView(
   children: <Widget>[
     new ListTile(
       leading: new CircleAvatar(),
       title: new Text('ItemText'),
       trailing: new Icon(Icons.thumb_up),
     )
   ],
 ),
 floatingActionButton: new FloatingActionButton(
     child: new Icon(Icons.adb),
     onPressed: () { /* do nothing */ }
 ),
 bottomNavigationBar: new BottomNavigationBar(
   items: [
     new BottomNavigationBarItem(
         icon: new Icon(Icons.assistant),
         title: new Text("News")),
     ...
],),);
```

Why do we want immutable Widgets?

Mixed responsibilities (Android)

1. Declare View in XML with initial attributes

```
<TextView
   android:id="@+id/myText"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:text="Hello Droidcon"
   android:textSize="14sp" />
```

! The `TextView`, responsible for drawing text, is now also responsible for the state of the text

2. Mutate View with updated data

```
override fun onCreate(savedInstanceState: Bundle?) {
   super.onCreate(savedInstanceState)

   val myText = findViewById<TextView>(R.id.myText)

   api.getData()
           .subscribe({ data ->
               // mutate view, append text
               myText.text += data
           }, { e ->
               handleError(e)
           })
}
```

Widgets on Flutter are immutable

```
// boilerplate
class DroidconWidget extends StatefulWidget {
 @override
 State<StatefulWidget> createState() {
   return new DroidconState();
 }
}
```

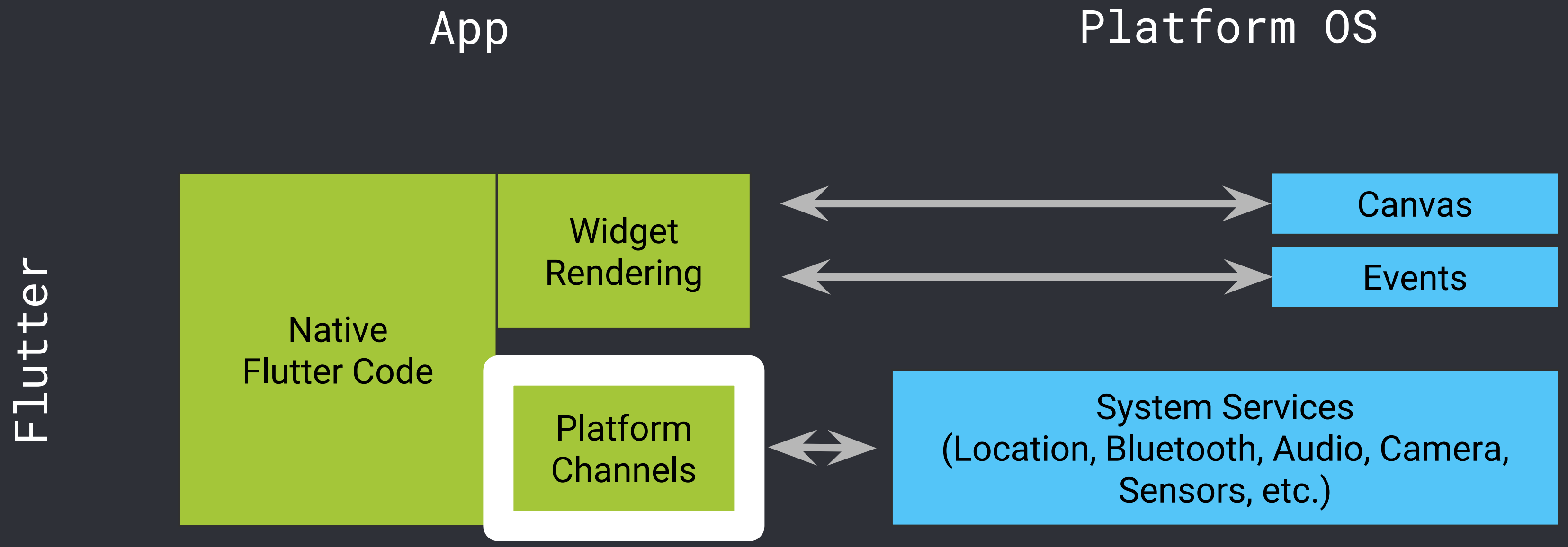
- You can’t change the text of a Widget, a new Widget instance is required
- build is a one way function binding data to immutable Widgets
- setState schedules rebuild of the widget

```
class DroidconState extends State<DroidconWidget> {

 var _data = "Hello Droidcon";

 @override
 void initState() {
   api.getData().then((data) {
     // append text, trigger rebuild
     setState(() {
       _data += data;
     });
   });
 }

 @override
 Widget build(BuildContext context) {
   return new Text(_data);
 }
}
```

Build function

- For smooth animations it may be called for every frame (remember: 120FPS!)
- Flutter diffs the result with the previous build result to minimize updates
- You don’t have to nest it very deep,
  - extract static parts
  - Split it in multiple build functions

Integration with the OS



Communication between Android and Flutter

- FlutterView (extends SurfaceView) is placed fullscreen in your Activity.
- Plugins can be initialized which register a MethodChannel on the FlutterView.
- These MethodChannel are invoked by the plugins Dart API

SharedPrefs Plugin example

Dart part of plugin

```dart
static const MethodChannel methodChannel =
   const MethodChannel('samples.flutter.io/battery');

String batteryLevel;
try {
     final int result =
              await methodChannel.invokeMethod('getBatteryLevel');
     batteryLevel = 'Battery level: $result%.';
} on PlatformException {
     batteryLevel = "Failed to get battery level.";
}
```

SharedPrefs Plugin example

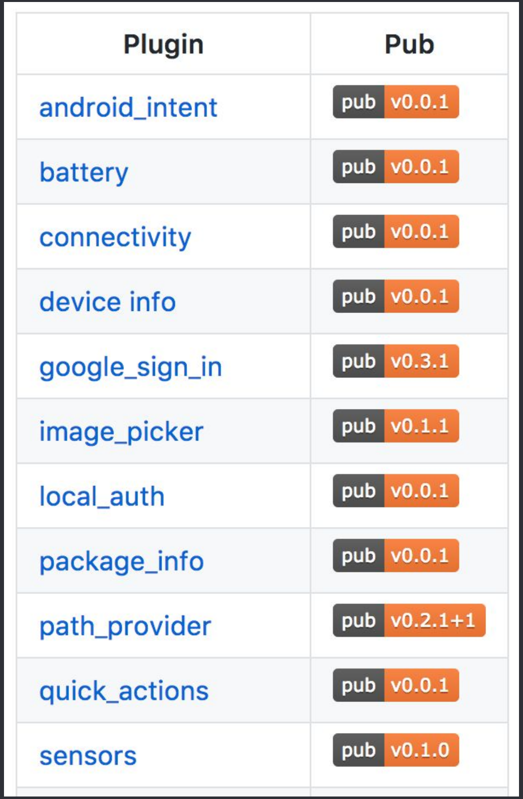
Android Kotlin part of plugin

```
val msgHandler: MethodCallHandler = MethodCallHandler { call, result ->
   if (call.method == "getBatteryLevel") {
       val level: Int = getBatteryLevel()

       if (level != -1) {
           result.success(level)
       } else {
           result.error("UNAVAILABLE", "Battery level not available.", null)
       }
   } else {
       result.notImplemented()
   }
}


MethodChannel(flutterView, "samples.flutter.io/battery").setMethodCallHandler(msgHandler)
```

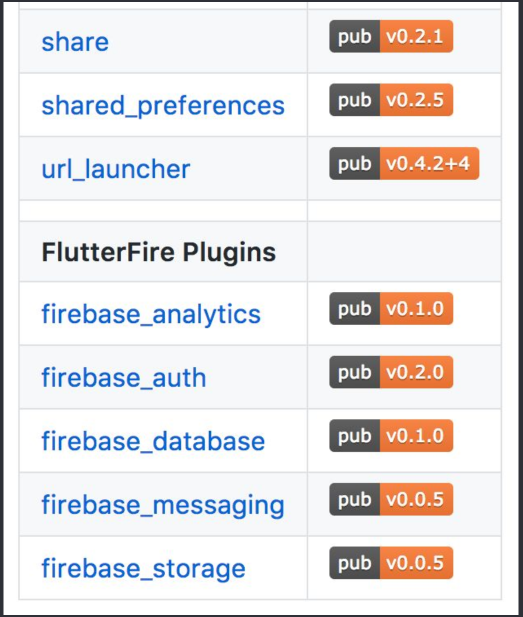
Plugins

- Communication is contract based, can’t be type safe
  - Method name is `String`
  - Method args are named and dynamic (`Map<String, dynamic>`)
- `MethodChannel` work in both directions

Official Plugins



Plugin	Pub
android_intent	pub v0.0.1
battery	pub v0.0.1
connectivity	pub v0.0.1
device info	pub v0.0.1
google_sign_in	pub v0.3.1
image_picker	pub v0.1.1
local_auth	pub v0.0.1
package_info	pub v0.0.1
path_provider	pub v0.2.1+1
quick_actions	pub v0.0.1
sensors	pub v0.1.0



share	pub v0.2.1
shared_preferences	pub v0.2.5
url_launcher	pub v0.4.2+4
FlutterFire Plugins	
firebase_analytics	pub v0.1.0
firebase_auth	pub v0.2.0
firebase_database	pub v0.1.0
firebase_messaging	pub v0.0.5
firebase_storage	pub v0.0.5

github.com/flutter/plugins

Shared code with Dart

- `FlutterView` is required to run dart code. You always need an Activity.
- You can’t run Dart code in a background service
- You can’t reuse network or parsing logic in your JobScheduler
- Unclear if this will ever work

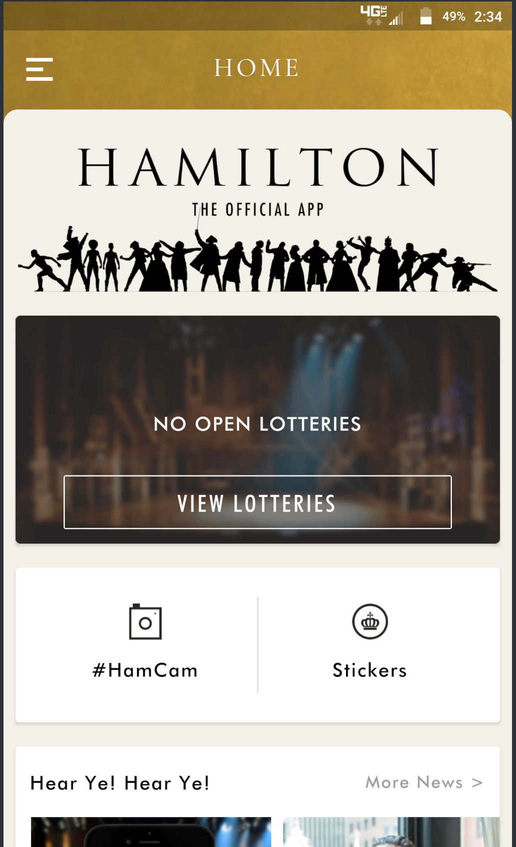
Is flutter production ready?

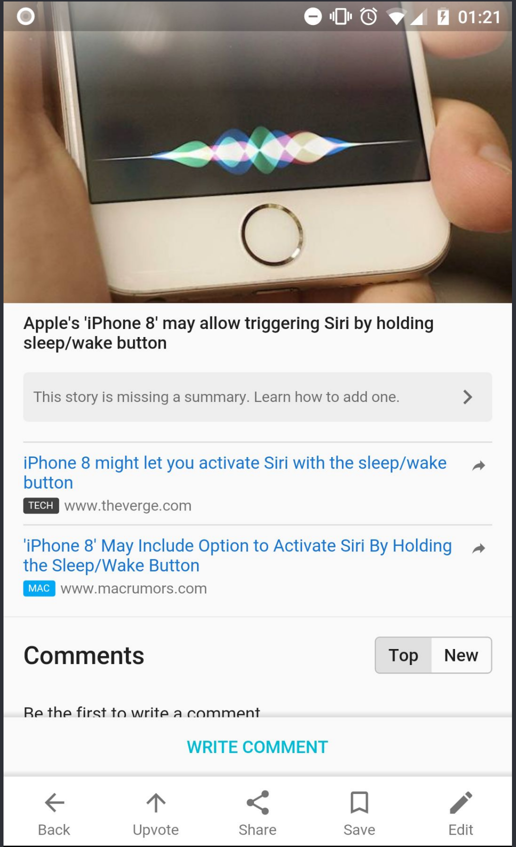
No, but...

...the Flutter team is very aware of it and working hard to make it production ready.

Flutter in Production

Hamilton





Apple's 'iPhone 8' may allow triggering Siri by holding sleep/wake button

This story is missing a summary. Learn how to add one.

iPhone 8 might let you activate Siri with the sleep/wake button
TECH www.theverge.com

'iPhone 8' May Include Option to Activate Siri By Holding the Sleep/Wake Button
MAC www.macrumors.com

Comments

Be the first to write a comment

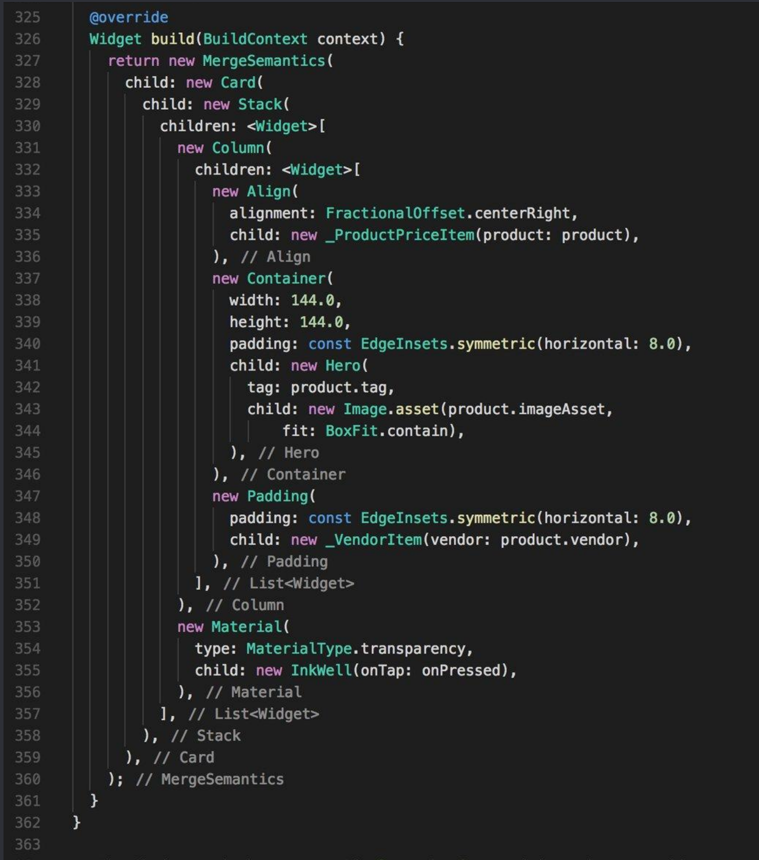
Newsvoice

What's missing

- Retrofit/OkHttp and a persistent cache
- Google Maps
- Push Notifications (iOS) sometimes give no callback
- No “headless flutter”

Room for improvement

- brackets hell, no DSL
  - workaround 'closing labels' in VS Code in IntelliJ maybe?!
  - Flatten with variables, extract methods
- One missing comma, breaks code completion



```
  @override
  Widget build(BuildContext context) {
    return new MergeSemantics(
      child: new Card(
        child: new Stack(
          children: <Widget>[
            new Column(
              children: <Widget>[
                new Align(
                  alignment: FractionalOffset.centerRight,
                  child: new _ProductPriceItem(product: product),
                ), // Align
                new Container(
                  width: 144.0,
                  height: 144.0,
                  padding: const EdgeInsets.symmetric(horizontal: 8.0),
                  child: new Hero(
                    tag: product.tag,
                    child: new Image.asset(product.imageAsset,
                        fit: BoxFit.contain),
                  ), // Hero
                ), // Container
                new Padding(
                  padding: const EdgeInsets.symmetric(horizontal: 8.0),
                  child: new _VendorItem(vendor: product.vendor),
                ), // Padding
              ], // List<Widget>
            ), // Column
            new Material(
              type: MaterialType.transparency,
              child: new InkWell(onTap: onPressed),
            ), // Material
          ], // List<Widget>
        ), // Stack
      ), // Card
    ); // MergeSemantics
  }
}
```

Openness of Dart/Flutter/Fuchsia

- Everything is open source
- Bug trackers are public and used by Googlers
- Dartlang newsletter inside sdk repository with detailed language decisions for Dart 2.0

- Get help in Gitter gitter.im/flutter/flutter

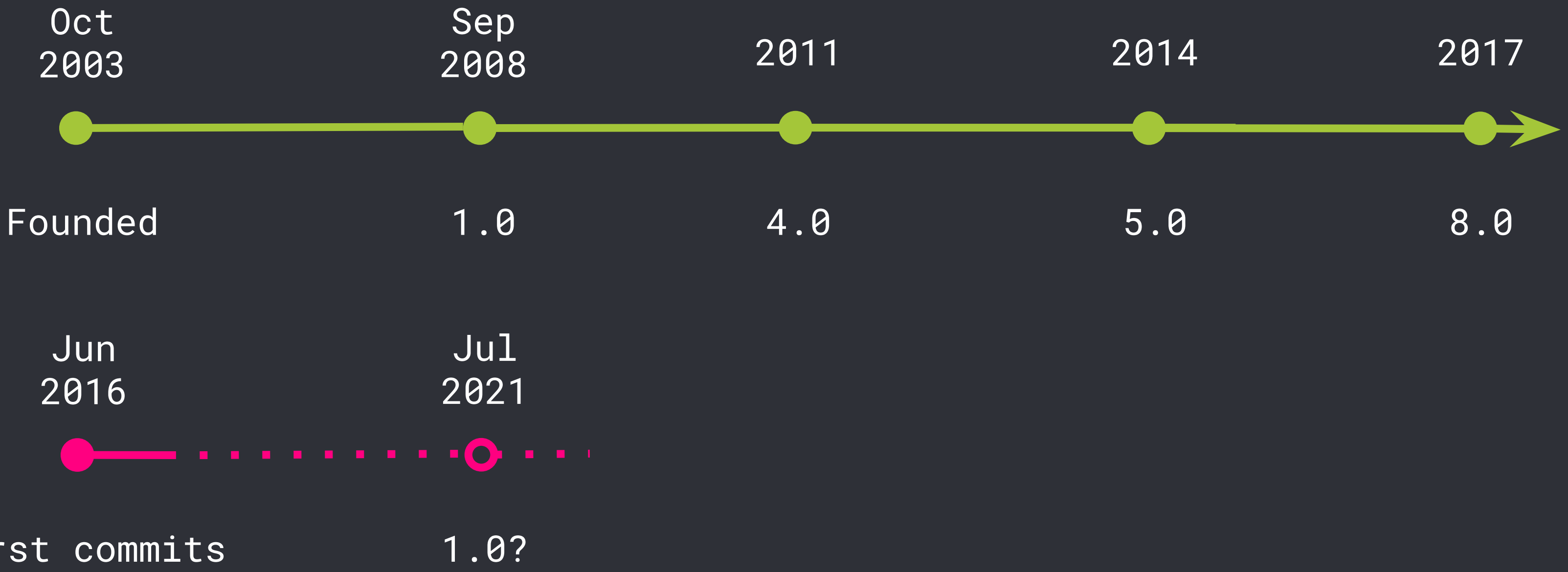
What is Fuchsia?

- Open-source OS by Google
- /ˈfjuːʃə/
- No Linux kernel - Google Kernel called Magenta
- Sky Engine with Vulkan
- Languages:
  - Dart, C++, Go, C, Python
  - No Java
- **Flutter Apps are native apps**

fuchsia.googlesource.com

Fuchsia Roadmap

Android

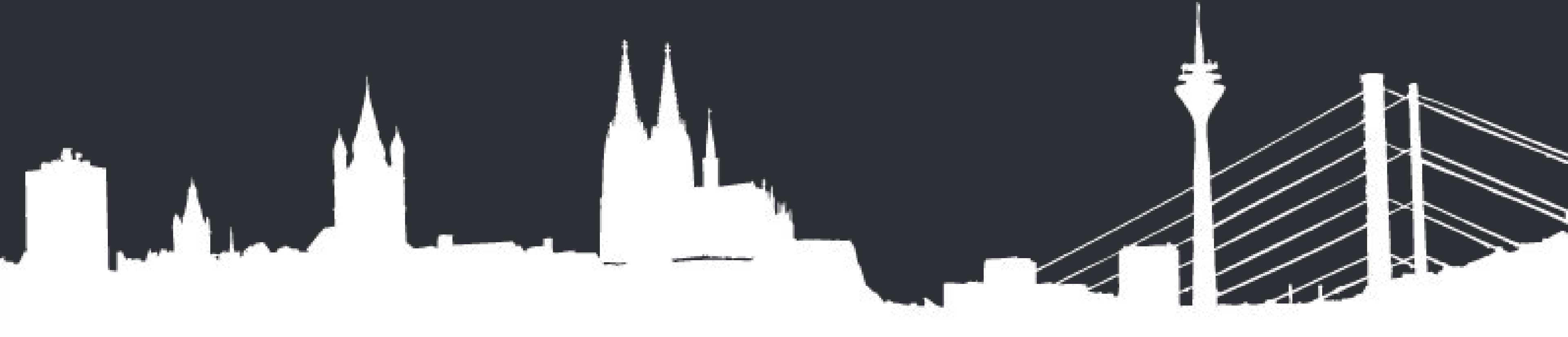


We are hiring!

grandcentrix.jobs
@grandcentrix

+Albrecht Noll
@UhrArt

+Pascal Welsch
@passsy



Learning Resources

- Official Page: https://flutter.io
- Dart Bootstrap: https://www.dartlang.org/guides/language/language-tour
- Widget catalog: https://flutter.io/widgets
- Ui Codelab: https://codelabs.developers.google.com/codelabs/flutter/
- Firebase Codelab: https://codelabs.developers.google.com/codelabs/flutter-firebase
- **Valuable Flutter Links**: https://github.com/Solido/awesome-flutter
- Flutter Examples: https://github.com/nisrulz/flutter-examples