Яндекс Поиск

Место Flutter в жизни Android-разработчика

Артур Василов, Android-разработчик

Дисклеймер

Все, сказанное здесь, является личным мнением и никак не может быть связано с позицией или продуктами компании «Яндекс»

Что такое Flutter?

- Фреймворк для кроссплатформенной разработки UI мобильных приложений от Google
- Dart в качестве языка
- Еще пока в бете

Flutter

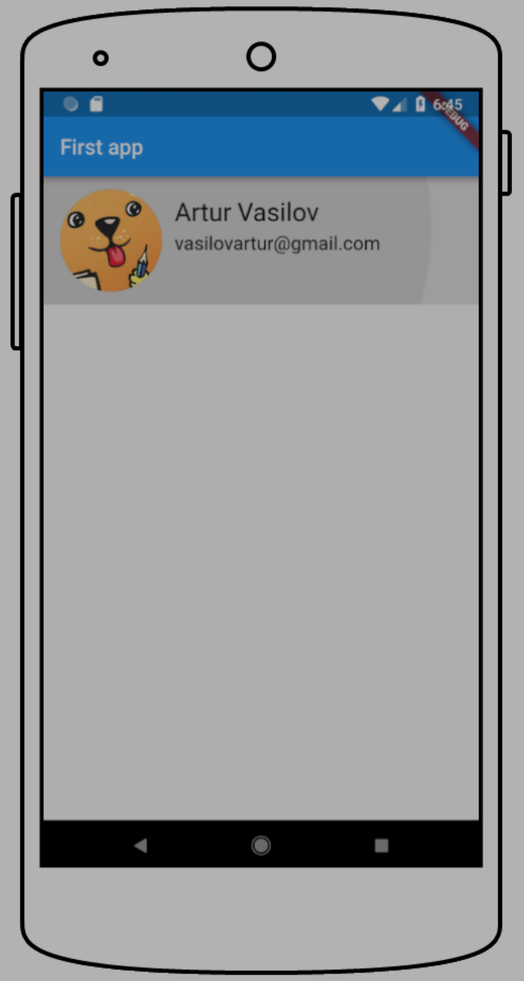
Что мы сегодня рассмотрим?

- Небольшое введение
- Почему Flutter может быть нам интересен?
- Почему Flutter не может быть нам интересен?
- Что в итоге?

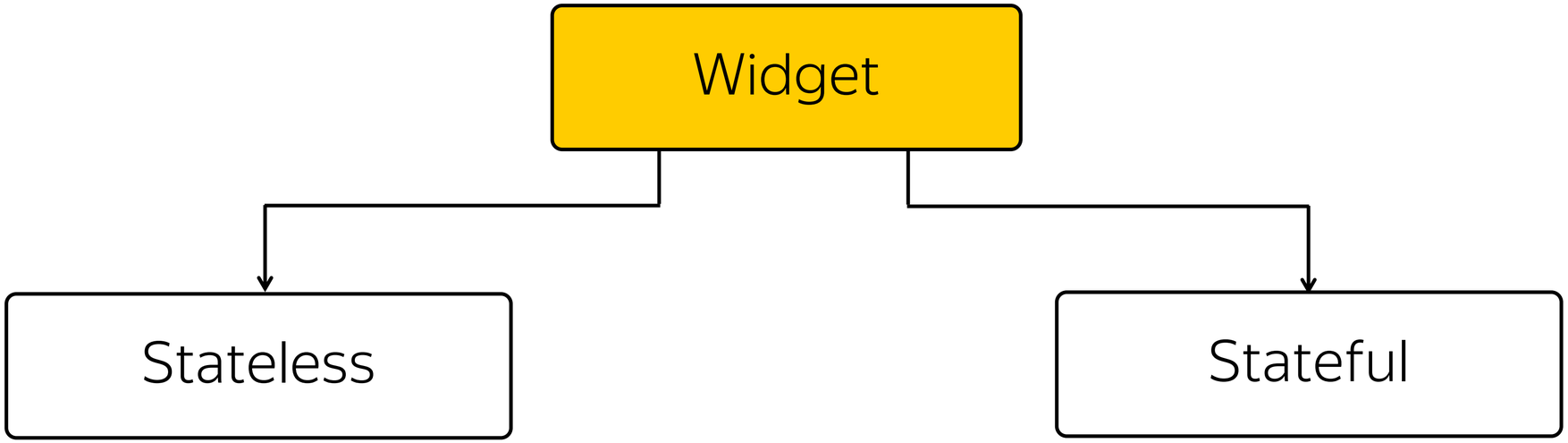
Попробуем сделать что-то простое

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

```
class MyApp extends StatelessWidget {


 @override
 Widget build(BuildContext context) {
    // TODO : implement
 }
}
```

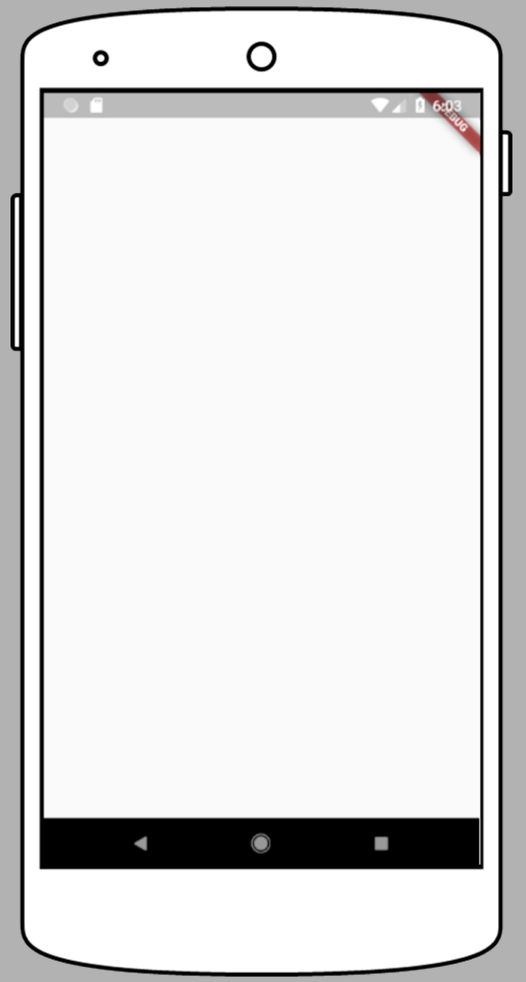


Введение



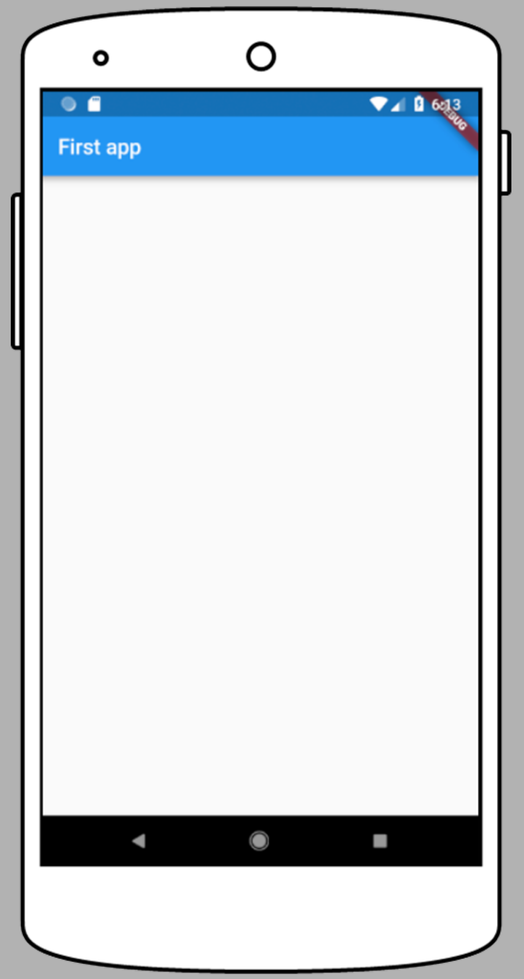
MaterialApp

```
class MyApp extends StatelessWidget {

 @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Hello, World',
      home: Scaffold(


      ),
    );
  }
}
```



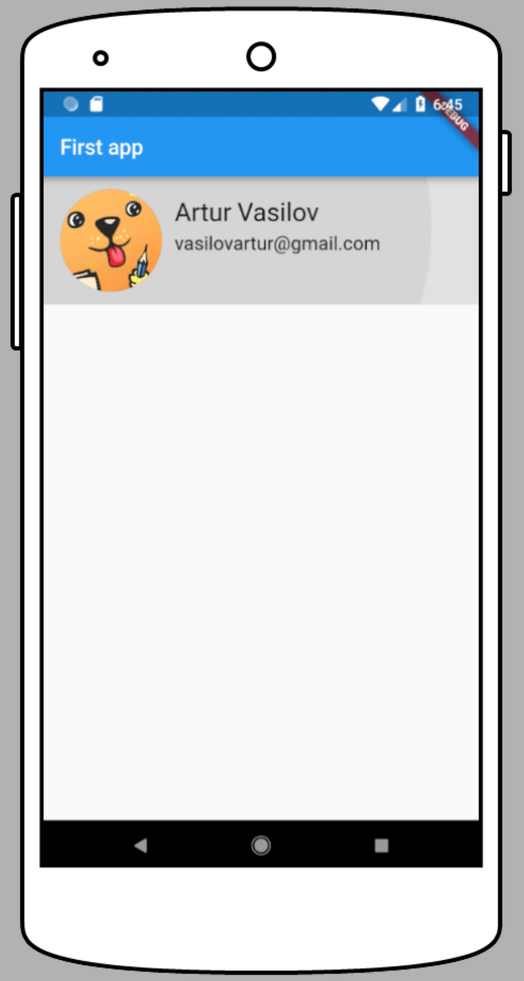
Добавляем AppBar

```
@override
Widget build(BuildContext context) {
 return MaterialApp(
   title: 'Hello, World',
   home: Scaffold(
     appBar: new AppBar(
       title: Text('First app'),
     ),
   ),
 );
}
```



Создаем отдельный виджет

```
return MaterialApp(
 title: 'Hello, World',
 home: Scaffold(
   appBar: new AppBar(title: Text('First app')),
   body: PersonWidget(
     Person(
       "Artur Vasilov",
       "vasilovartur@gmail.com",
       "https://website.com/avatar.jpg"),
   ),
 ),
);
```

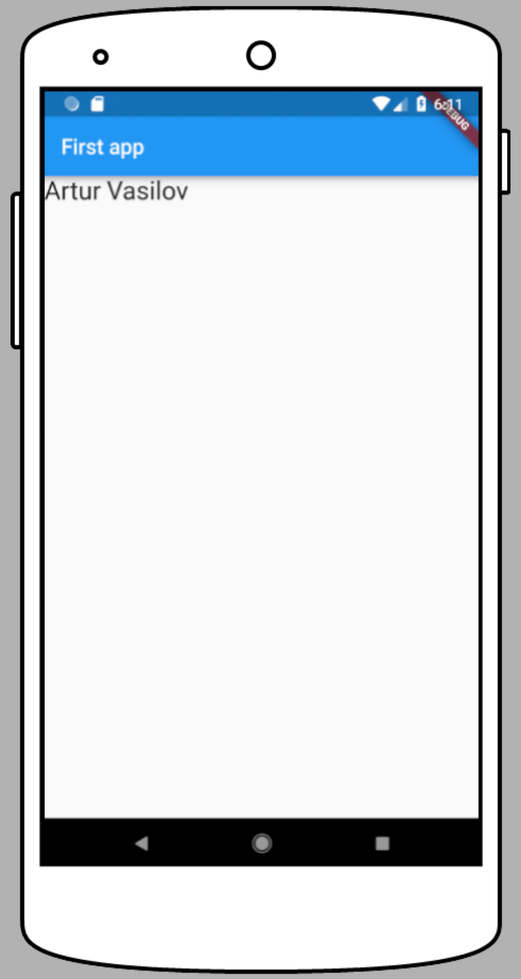


PersonWidget

```
class PersonWidget extends StatelessWidget {
 final Person _person;

 PersonWidget(this._person);

 @override
 Widget build(BuildContext context) {
   // TODO : implement
 }
}
```

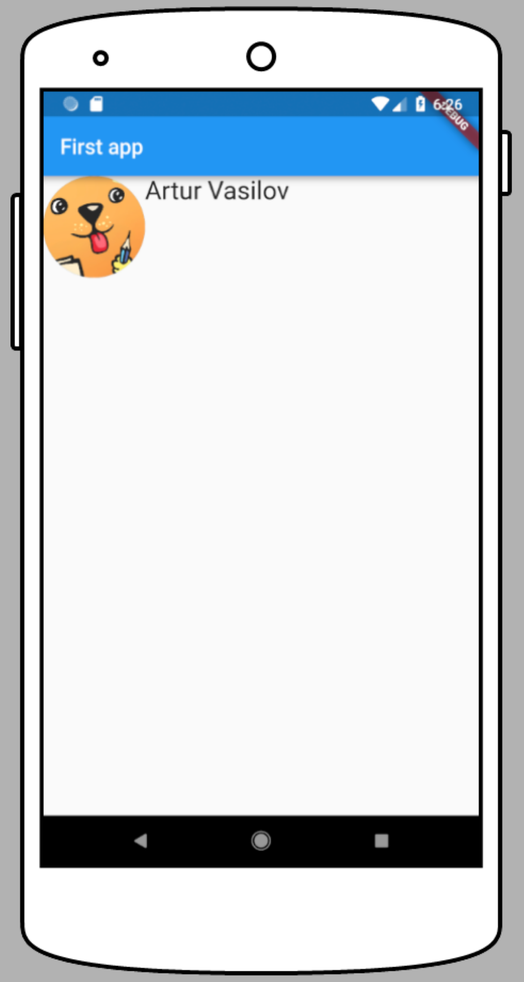
Имя

```
class PersonWidget extends StatelessWidget {
 final Person _person;

 PersonWidget(this._person);

 @override
 Widget build(BuildContext context) {
   return Text(
     _person.name,
     style: TextStyle(fontSize: 24.0),
   );
 }
}
```



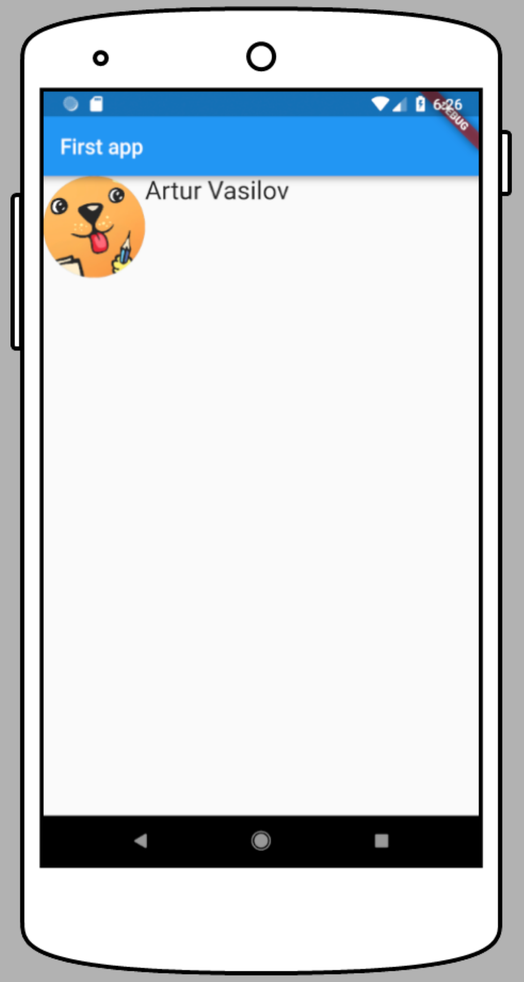
Картинка

```
return Row(
 crossAxisAlignment: CrossAxisAlignment.start,
 children: <Widget>[
   CircleAvatar(
     radius: 48.0,
     backgroundImage: NetworkImage(_person.avatarUrl),
   ),
   Text(
     _person.name,
     style: TextStyle(fontSize: 24.0),
   )
 ],
);
```



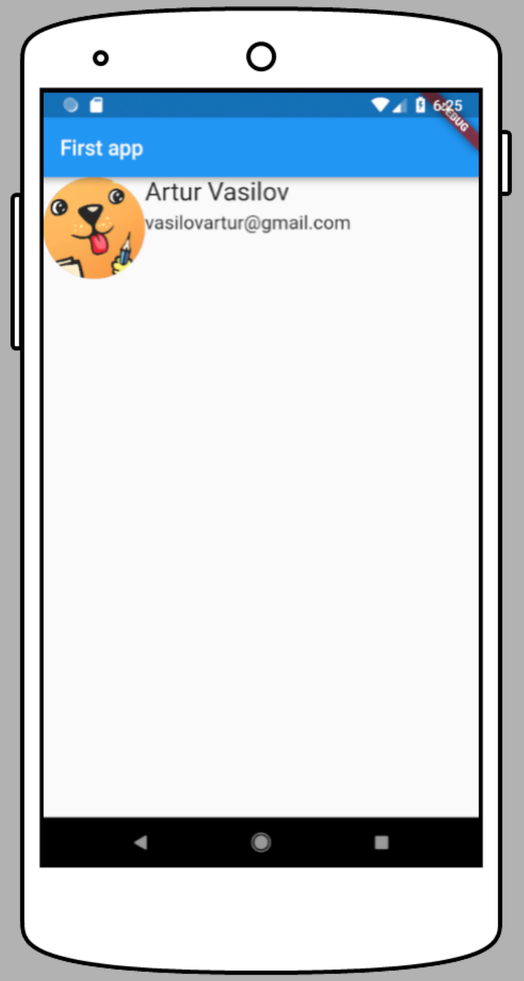
Картинка

```
return Row(
 crossAxisAlignment: CrossAxisAlignment.start,
 children: <Widget>[
   CircleAvatar(
     radius: 48.0,
     backgroundImage: NetworkImage(_person.avatarUrl),
   ),
   Text(
     _person.name,
     style: TextStyle(fontSize: 24.0),
   )
 ],
);
```



Почта

```
Column(
 crossAxisAlignment: CrossAxisAlignment.start,
 children: <Widget>[
   Text(
     _person.name,
     style: TextStyle(fontSize: 24.0),
   ),
   Container(height: 4.0),
   Text(
     _person.email,
     style: TextStyle(fontSize: 18.0),
   )
 ],
),
```

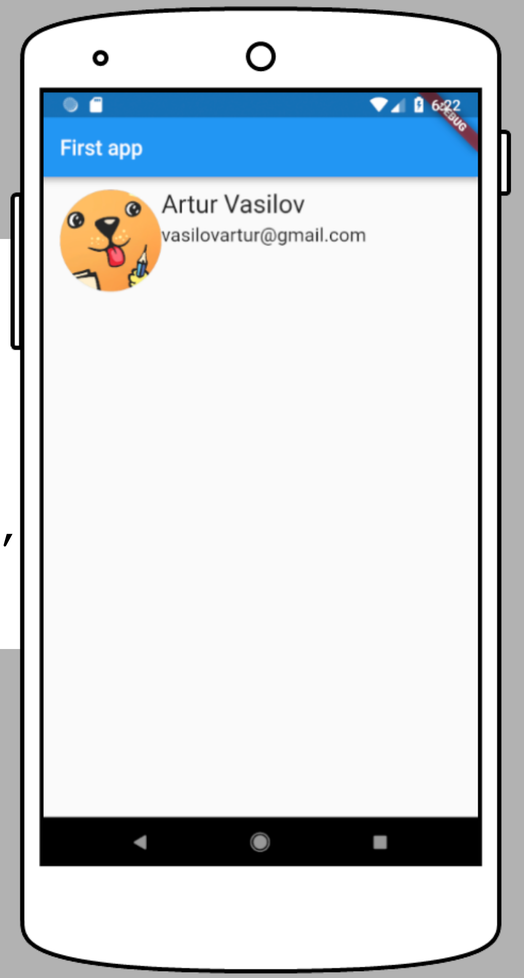


Почта

```
Column(
 crossAxisAlignment: CrossAxisAlignment.start,
 children: <Widget>[
   Text(
     _person.name,
     style: TextStyle(fontSize: 24.0),
   ),
   Container(height: 4.0),
   Text(
     _person.email,
     style: TextStyle(fontSize: 18.0),
   )
 ],
),
```

Отступы

```
@override
Widget build(BuildContext context) {
  return Container(
    height: 120.0,
    padding: EdgeInsets.only(left: 16.0, top: 12.0,
                             right: 16.0, bottom: 12.0),
    child: Row(
      // ...
    ),
  );
}
```

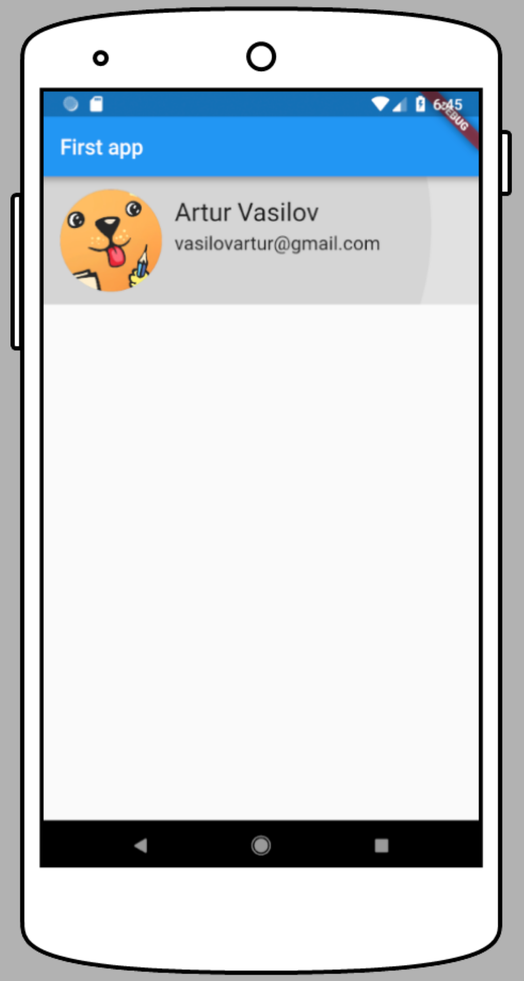


Отступы

```
Container(
 padding: EdgeInsets.only(left: 12.0, top: 8.0),
 child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: <Widget>[
      Text(...),
      Container(...),
      Text(...)
    ],
 ),
),
```

Ripple effect

```
@override
Widget build(BuildContext context) {
  return InkWell(
    onTap: () => {},
    child: Container(
      // ...
    ),
  );
}
```



Что можно заметить

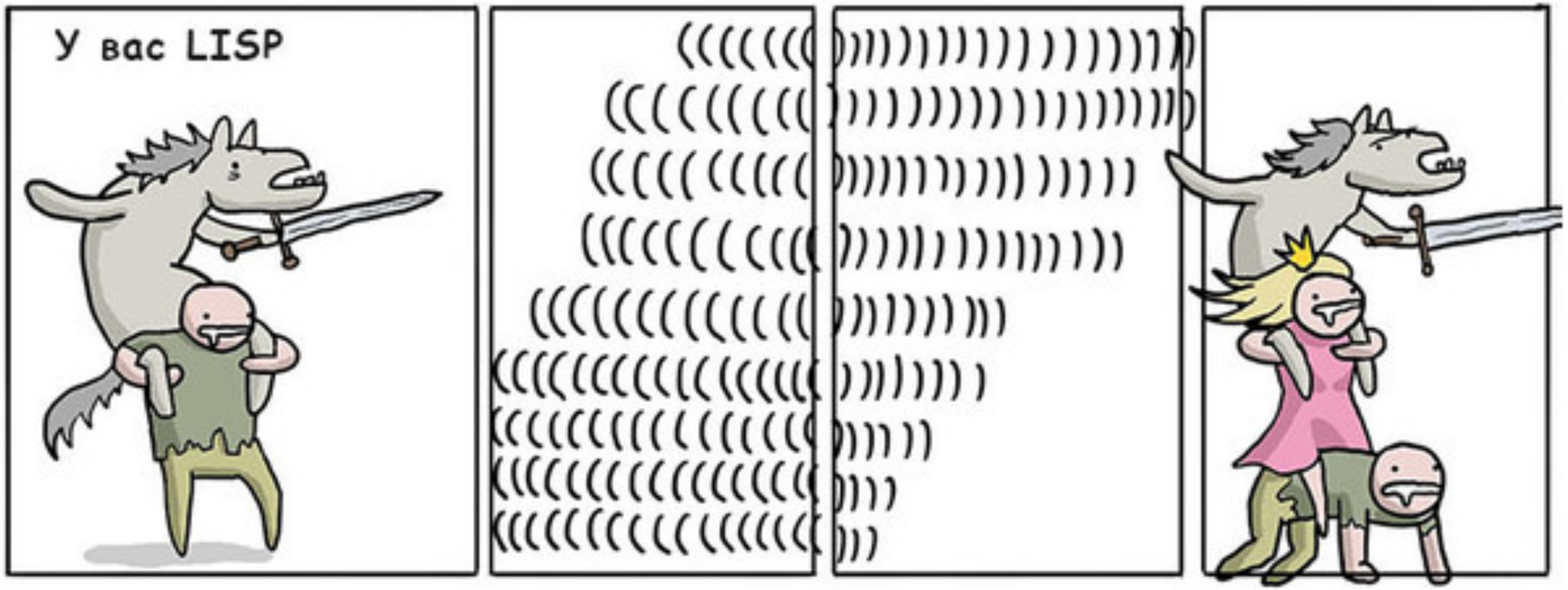
Хорошо подходит, чтобы быстро что-то сделать (прототип):

- Нет всяких Activity / Fragment / xml / ...
- Работа с сетью тоже намного короче
- Hot reload
- Кроссплатформенный UI (заметить было нельзя, но да)
- Архитектура?

Что можно заметить

- Хорошо подходит, чтобы быстро что-то сделать (прототип):
  - Нет всяких Activity / Fragment / xml / ...
  - Работа с сетью тоже намного короче
  - Hot reload
  - Кроссплатформенный UI (заметить было нельзя, но да)
- Декларативный UI-фреймворк
- Скобочки))))))))))

Спасти принцессу



))])))])) } – разбиваем код

```
Widget buildPersonInfo() {
 return Column(
   crossAxisAlignment: CrossAxisAlignment.start,
   children: <Widget>[
     Text(...),
     Container(...),
     Text(...)
   ],
 );
}
```

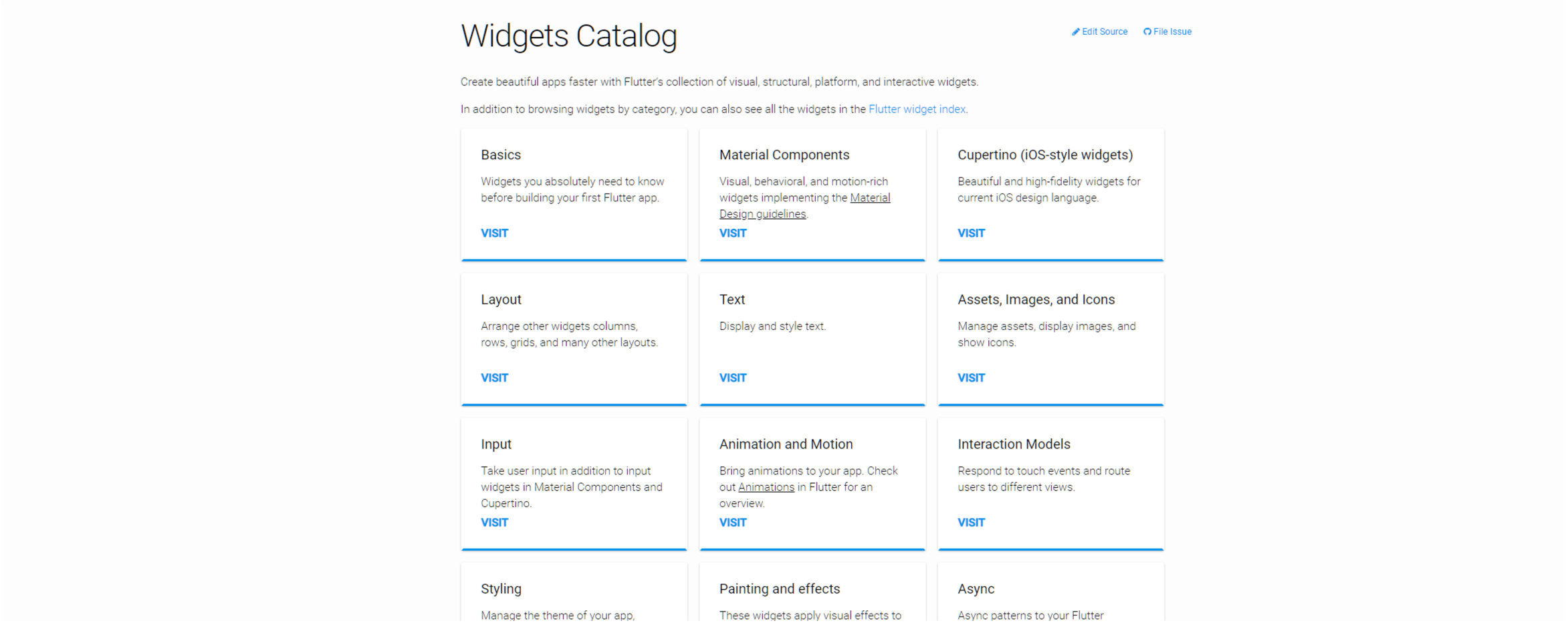
))]))))]))}– разбиваем код

```
child: Row(
 crossAxisAlignment: CrossAxisAlignment.start,
 children: <Widget>[
   CircleAvatar(...),
   Container(
     padding: EdgeInsets.only(left: 12.0, top: 8.0),
     child: buildPersonInfo(),
   ),
 ],
),
```

Что еще крутого во Flutter

Большое количество виджетов, в том числе и хорошая интеграция с Material Design

Widgets



flutter.io/widgets/

Что еще крутого во Flutter

Большое количество виджетов, в том числе и хорошая интеграция с Material Design

Высокая производительность (похоже, что так, но это не 100%)

- Нативный рендеринг
- Изменение только того, что меняется

Stateful Widget

- Widget строится не на основе констант, а на основе параметров, которые могут меняться
- Все то же самое, только при изменении параметров надо вызывать setState

Stateful Widget

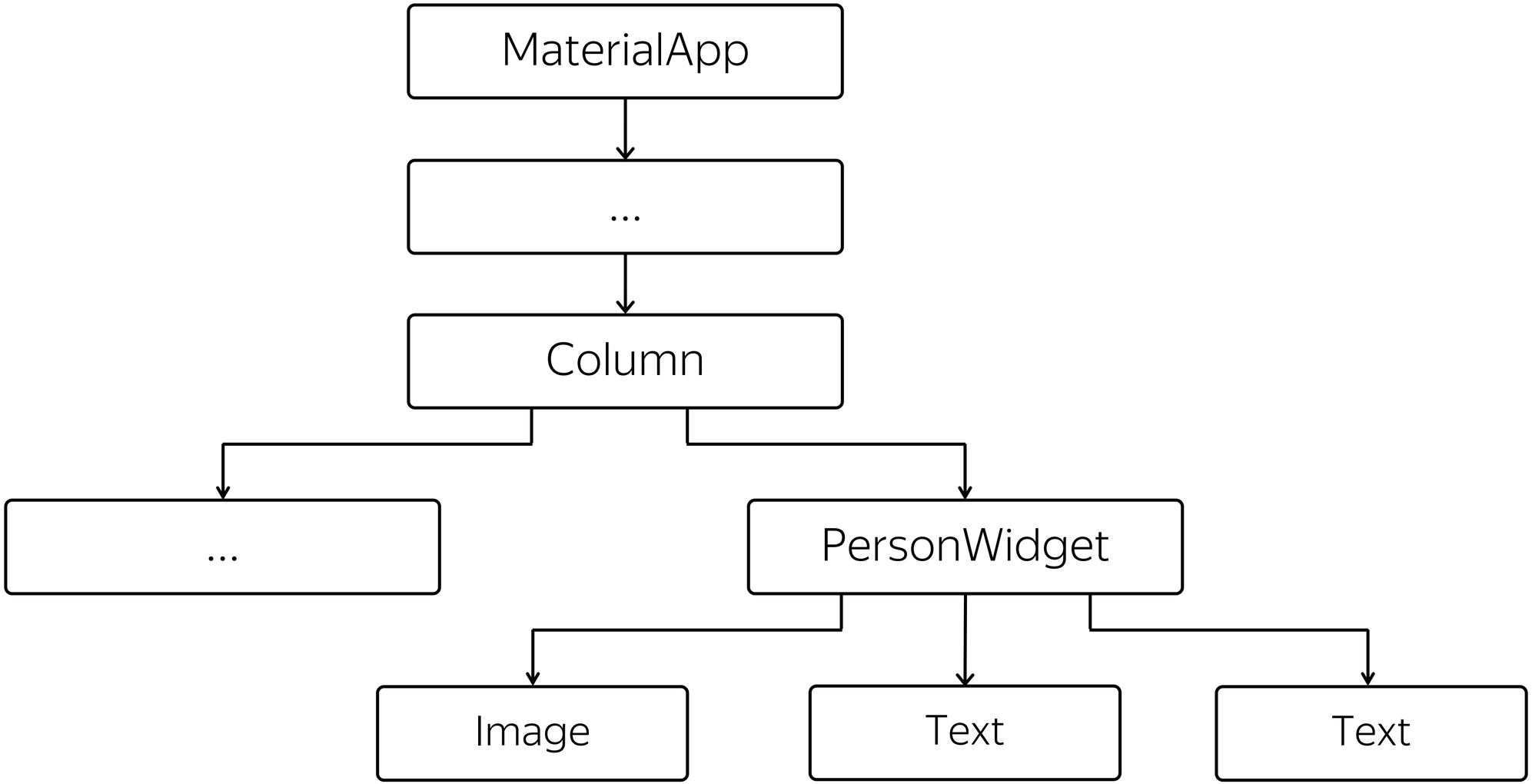
```
class NetworkPersonWidget extends StatefulWidget {

 @override
 State createState() => NetworkPersonWidgetState();
}

class NetworkPersonWidgetState extends State<NetworkPersonWidget> {

 @override
 Widget build(BuildContext context) {
   // TODO : implement
 }
}
```

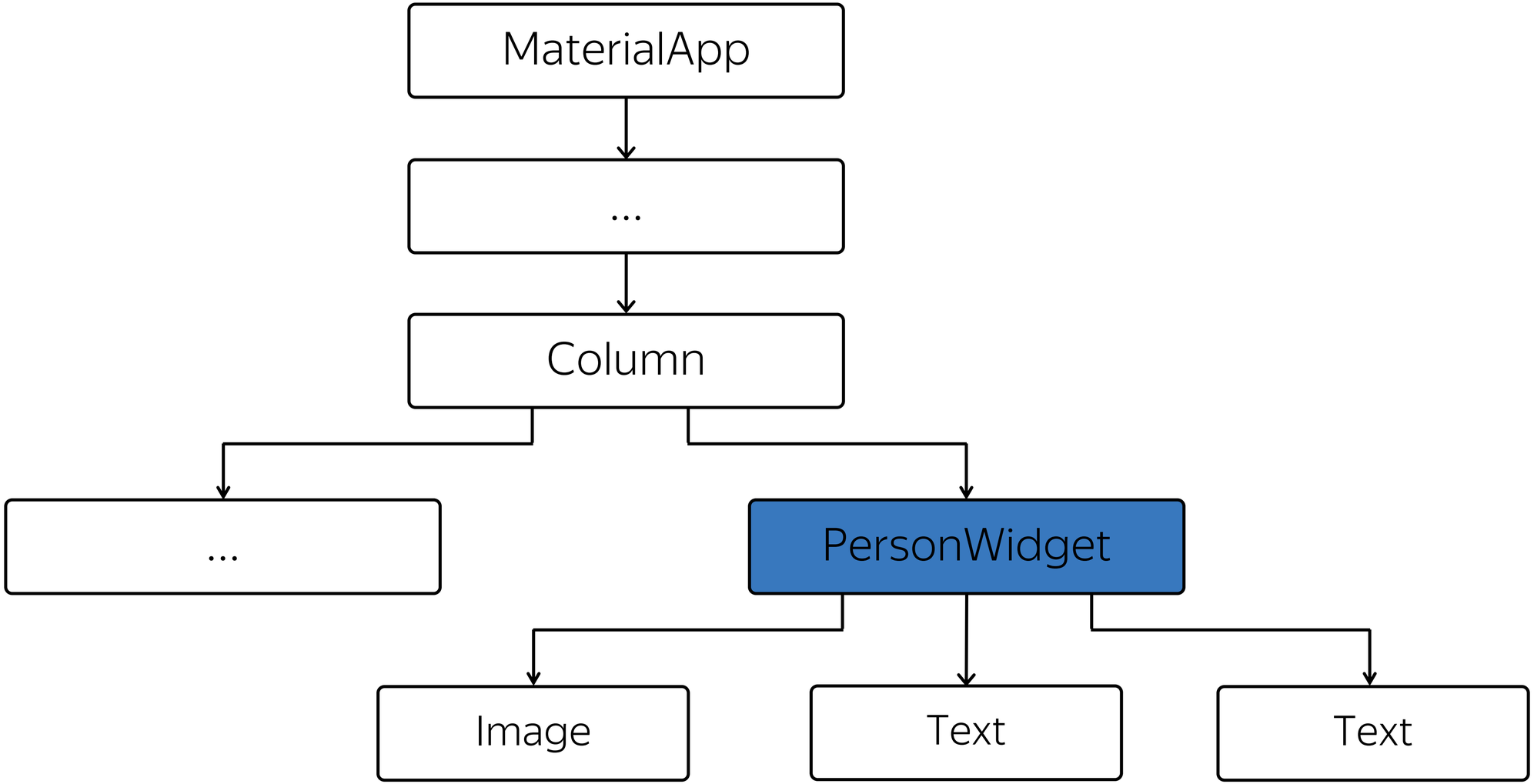
Получаем данные с сервера

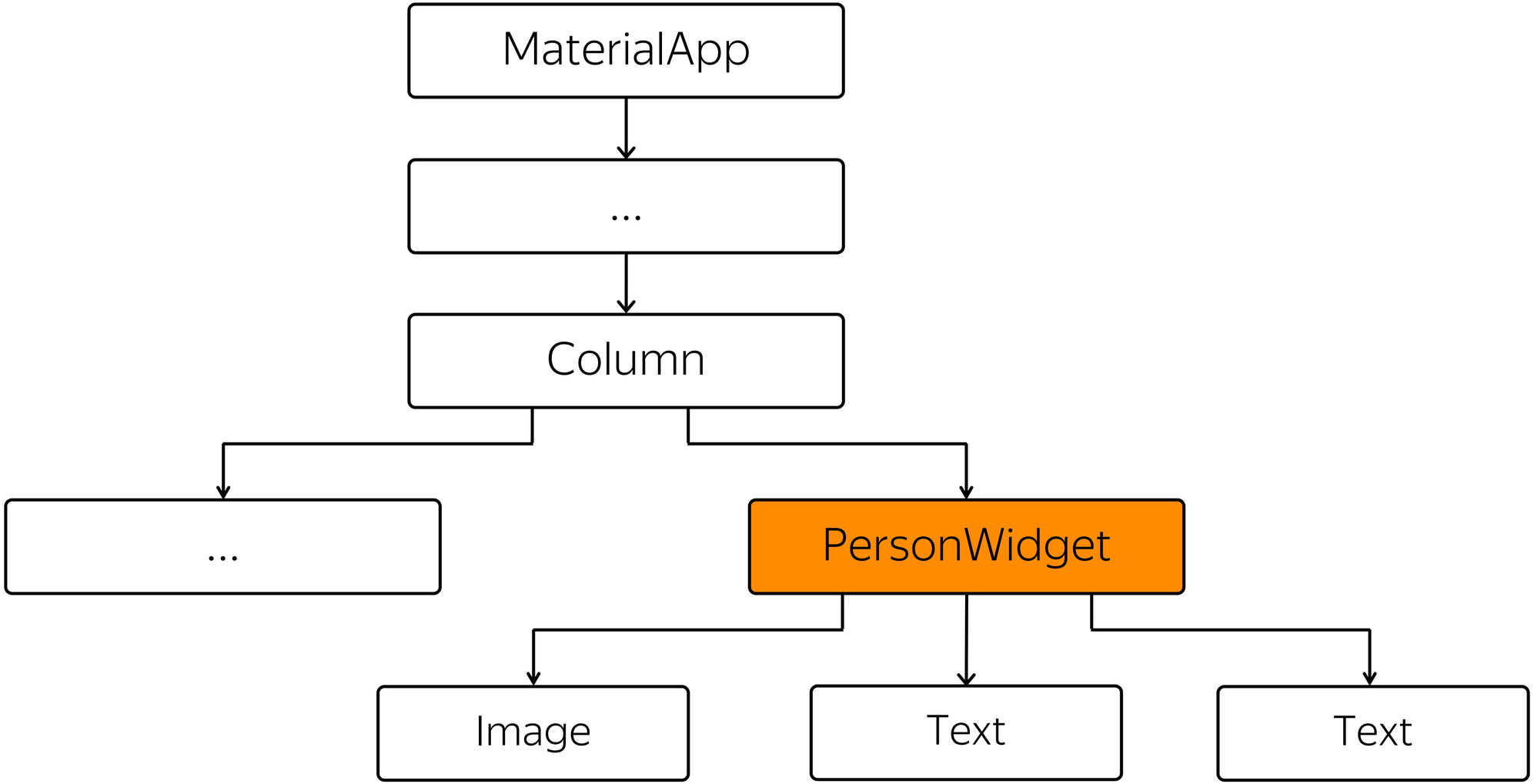
```
Future<Person> fetchPerson() async {
 final response = await http.get('https://myserver.com/person');

 if (response.statusCode == 200) {
   // If server returns an OK response, parse the JSON
   return Person.fromJson(json.decode(response.body));
 } else {
   // If that response was not OK, throw an error.
   throw Exception('Failed to load post');
 }
}
```

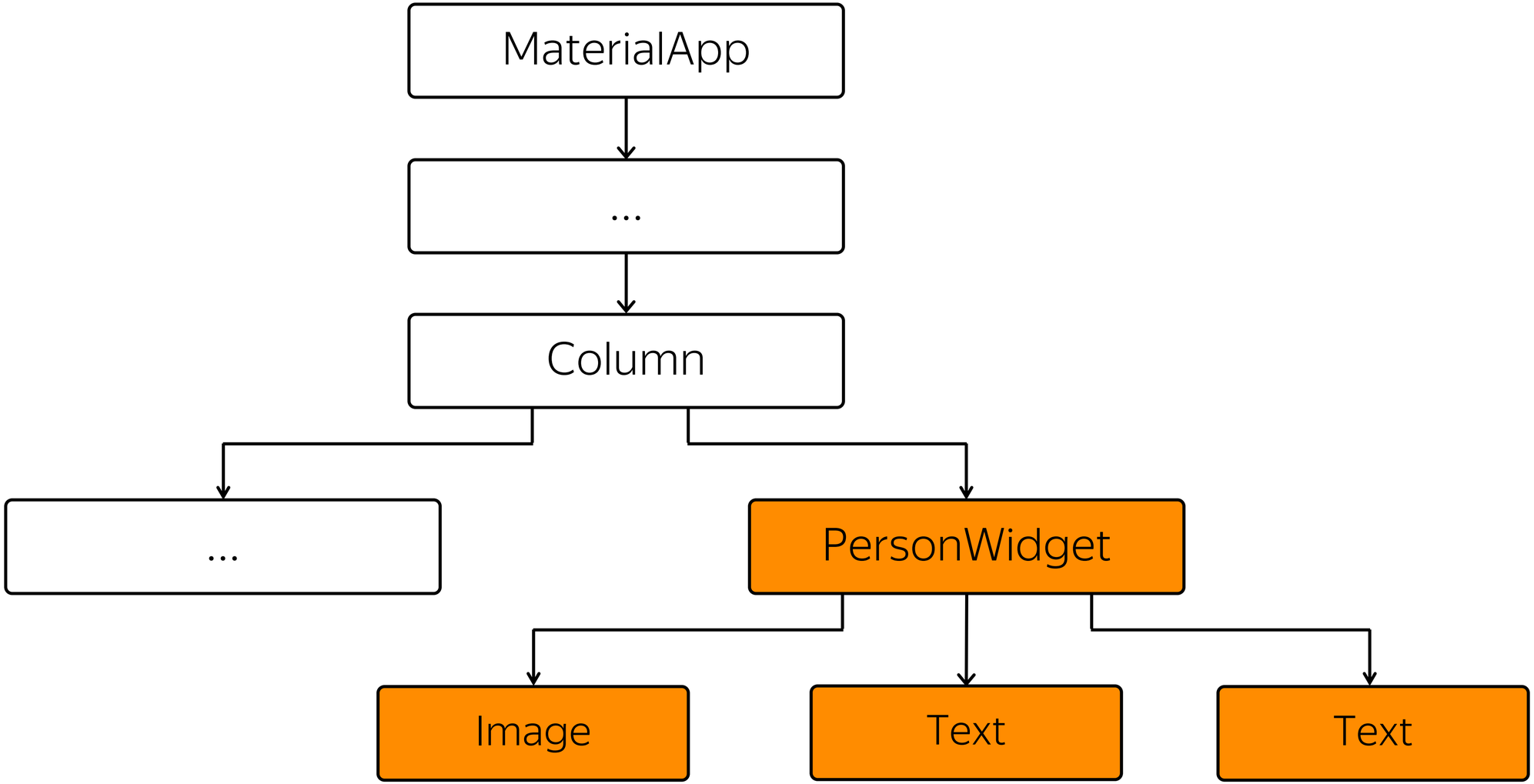
Меняем стейт

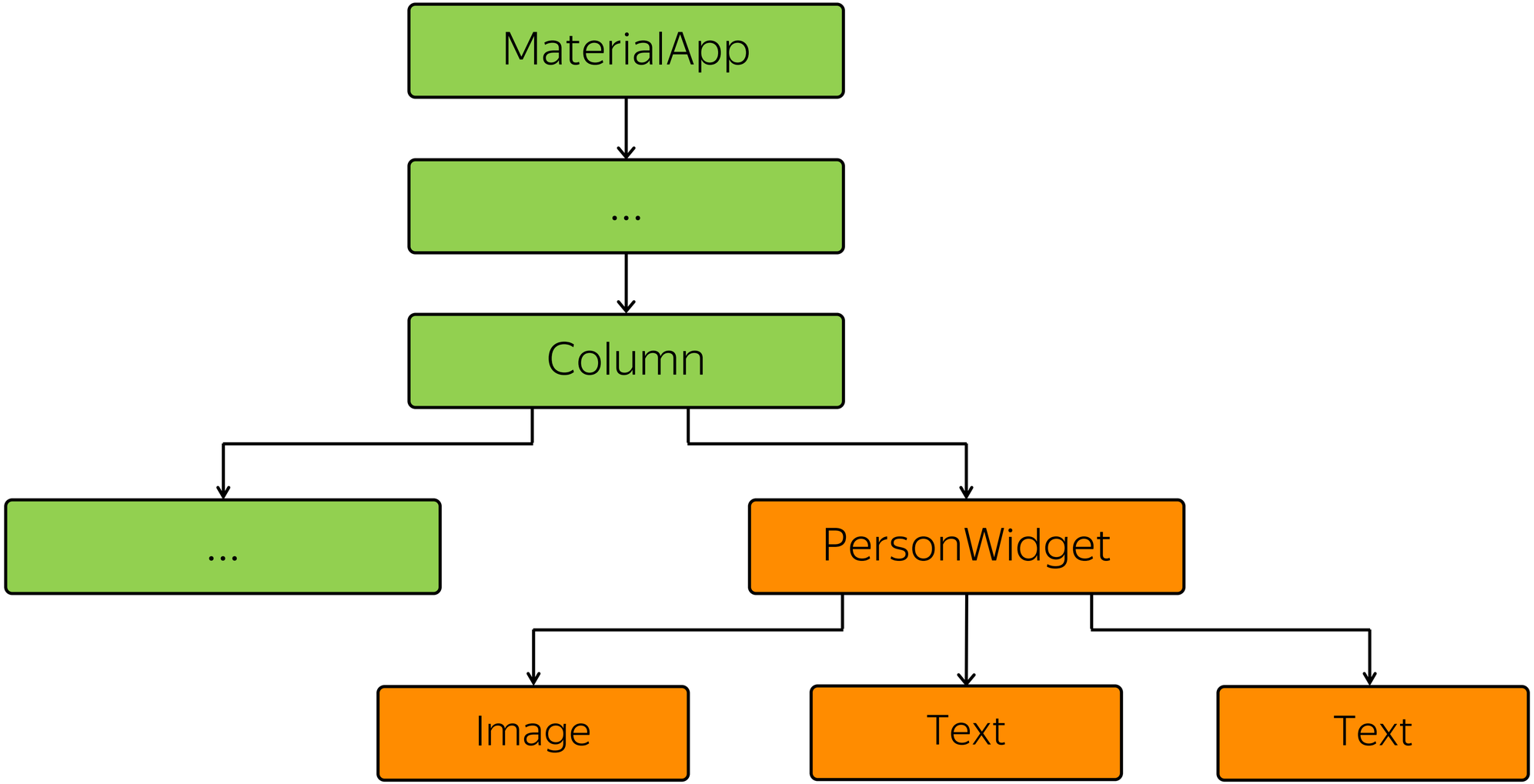
```
@override
void initState() {
 super.initState();

 Future<Person> personFuture = fetchPerson();
 personFuture.then((person) {
   setState(() {
     _person = person;
   });
 }, onError: (e) {
   handleError(e);
 });
}
```

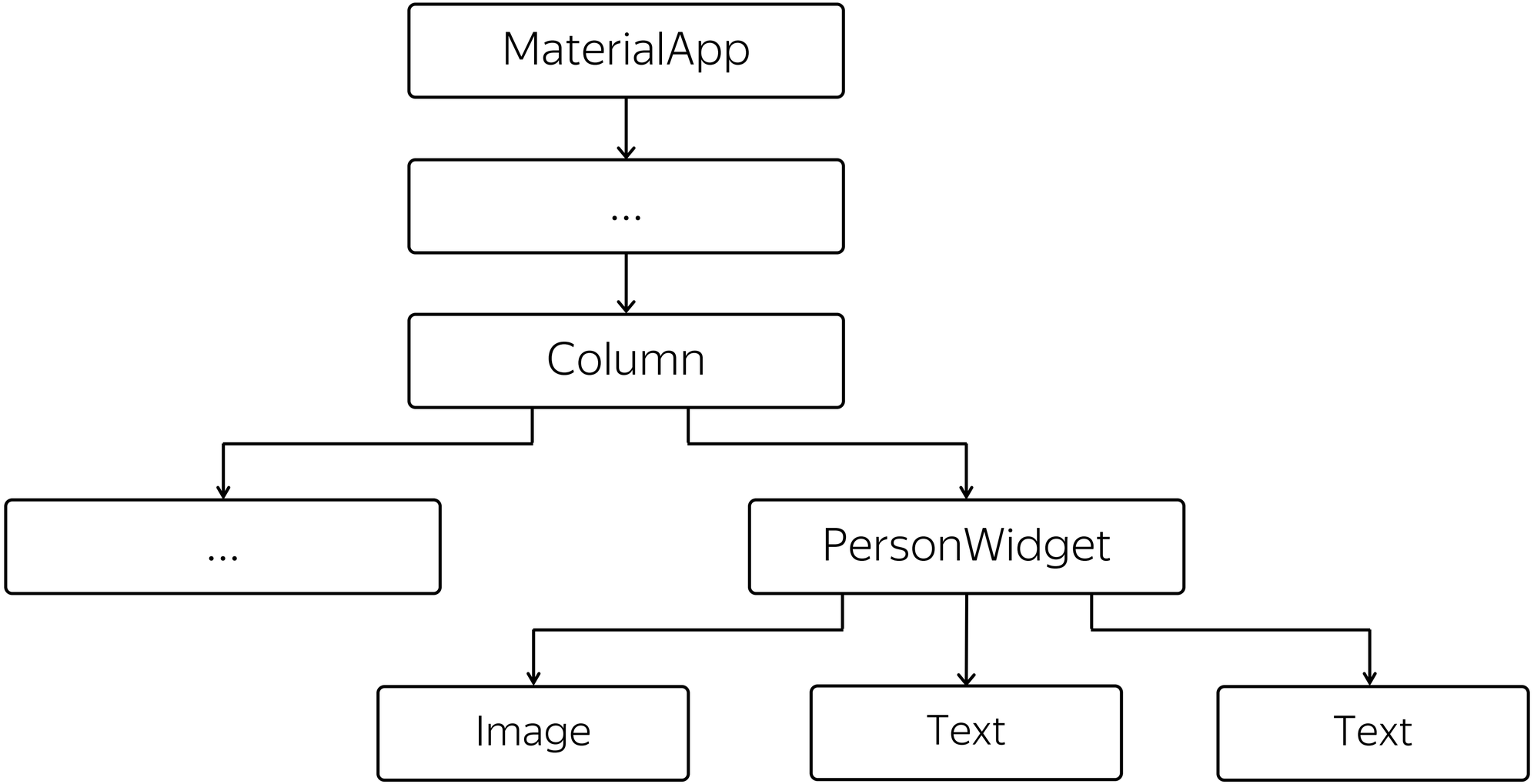

MaterialApp
...
Column
...
PersonWidget
Image
Text
Text

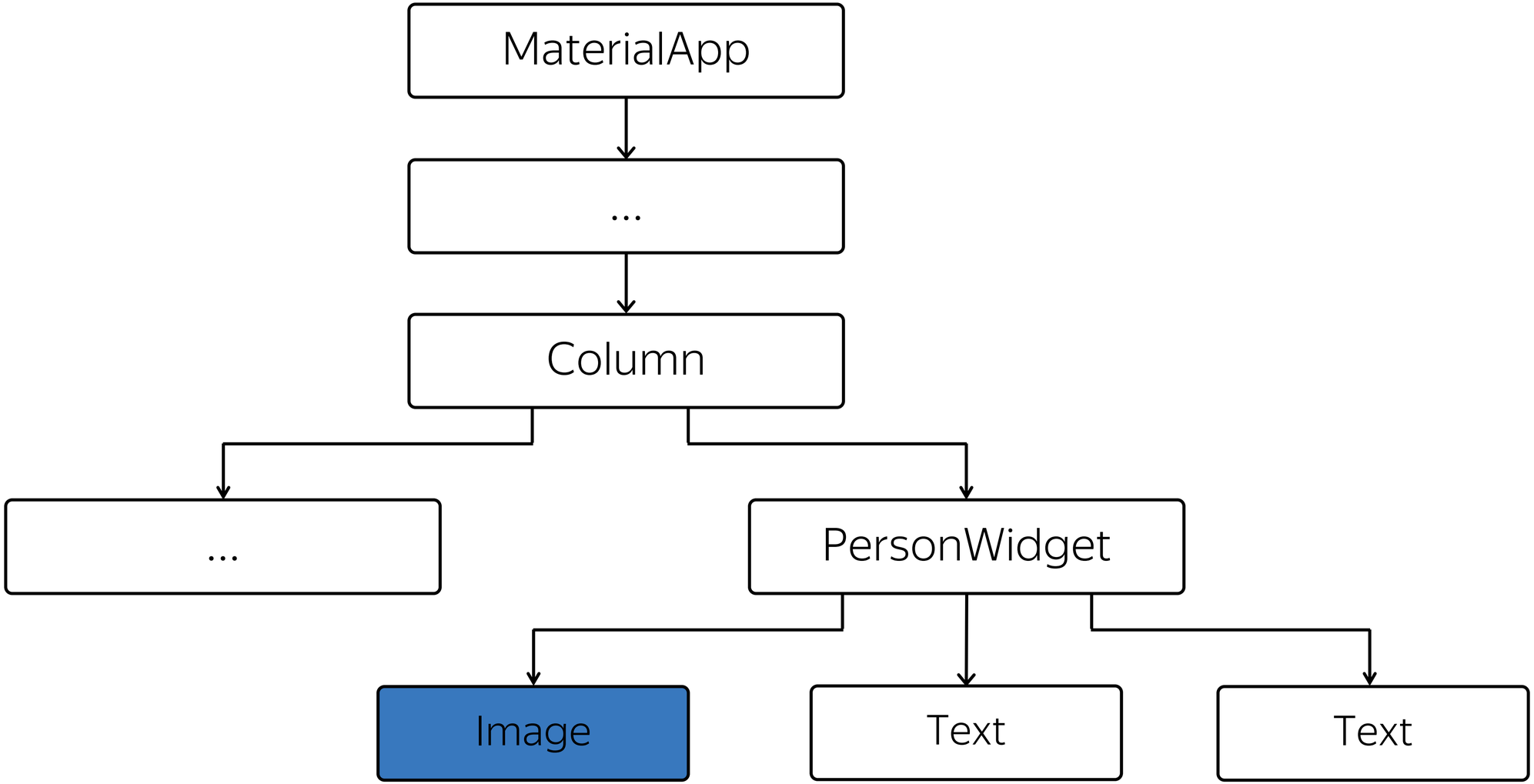

MaterialApp
...
Column
...
PersonWidget
Image
Text
Text

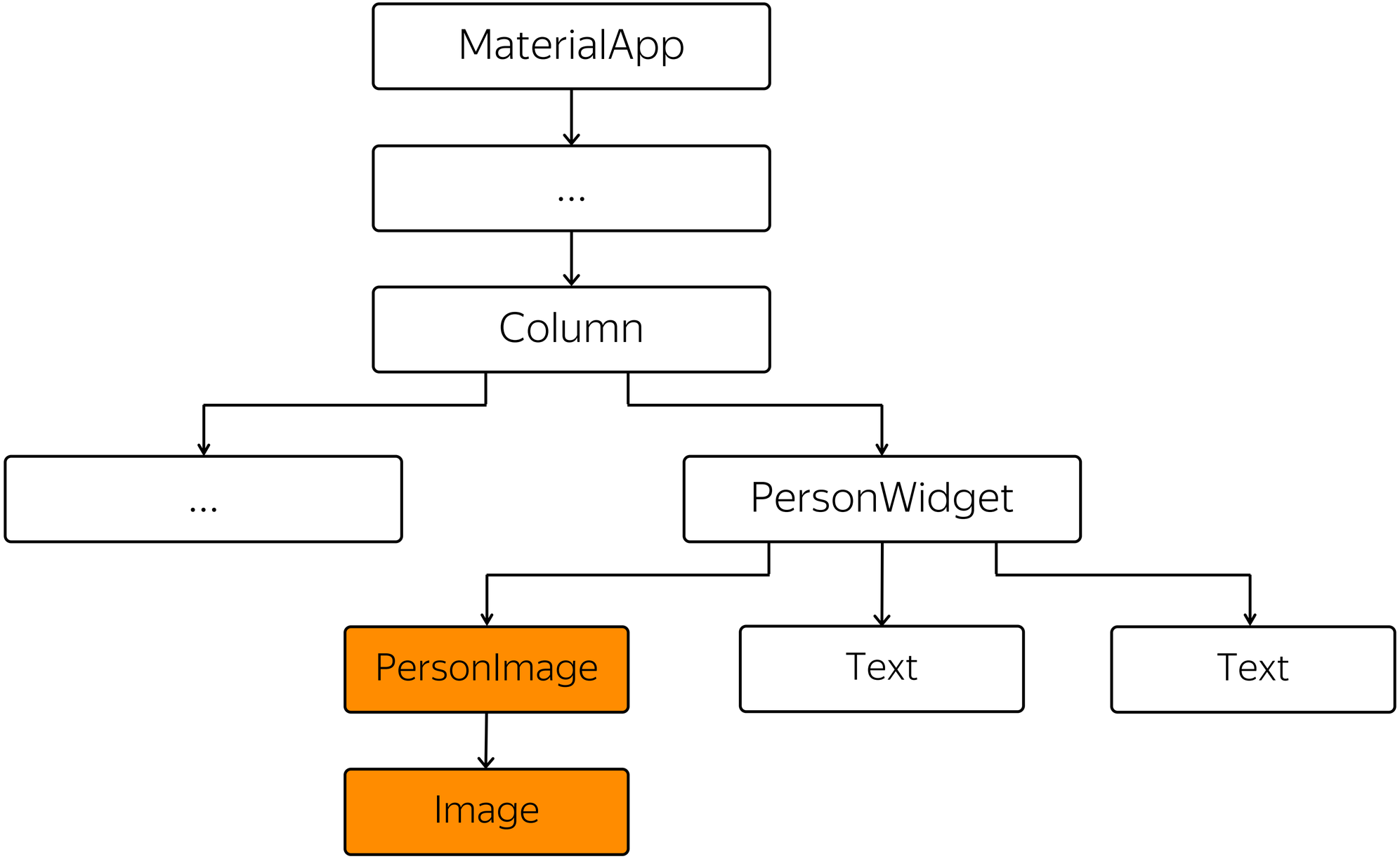

MaterialApp
...
Column
...
PersonWidget
Image
Text
Text

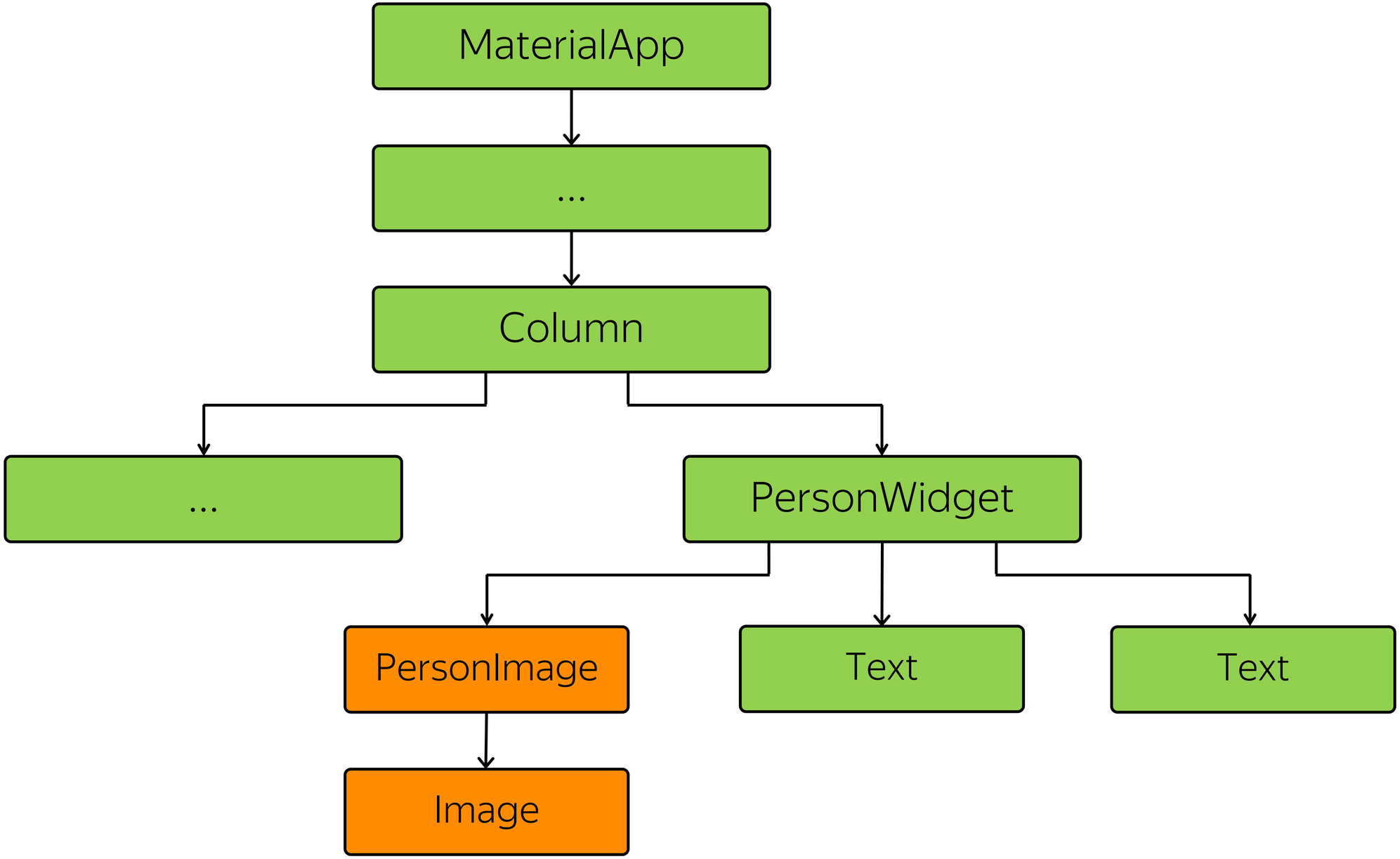

MaterialApp
...
Column
...
PersonWidget
Image
Text
Text


MaterialApp
...
Column
...
PersonWidget
Image
Text
Text


MaterialApp
...
Column
...
PersonWidget
Image
Text
Text


MaterialApp
...
Column
...
PersonWidget
Image
Text
Text


MaterialApp
...
Column
...
PersonWidget
PersonImage
Text
Text
Image


MaterialApp
...
Column
...
PersonWidget
PersonImage
Text
Text
Image

Widgets

- Это основное, что нужно знать о виджетах, чтобы строить UI
- Дальнейшие рассмотрения – это уже изучение фреймворка

Что еще крутого во Flutter

Большое количество виджетов, в том числе и хорошая интеграция с Material Design

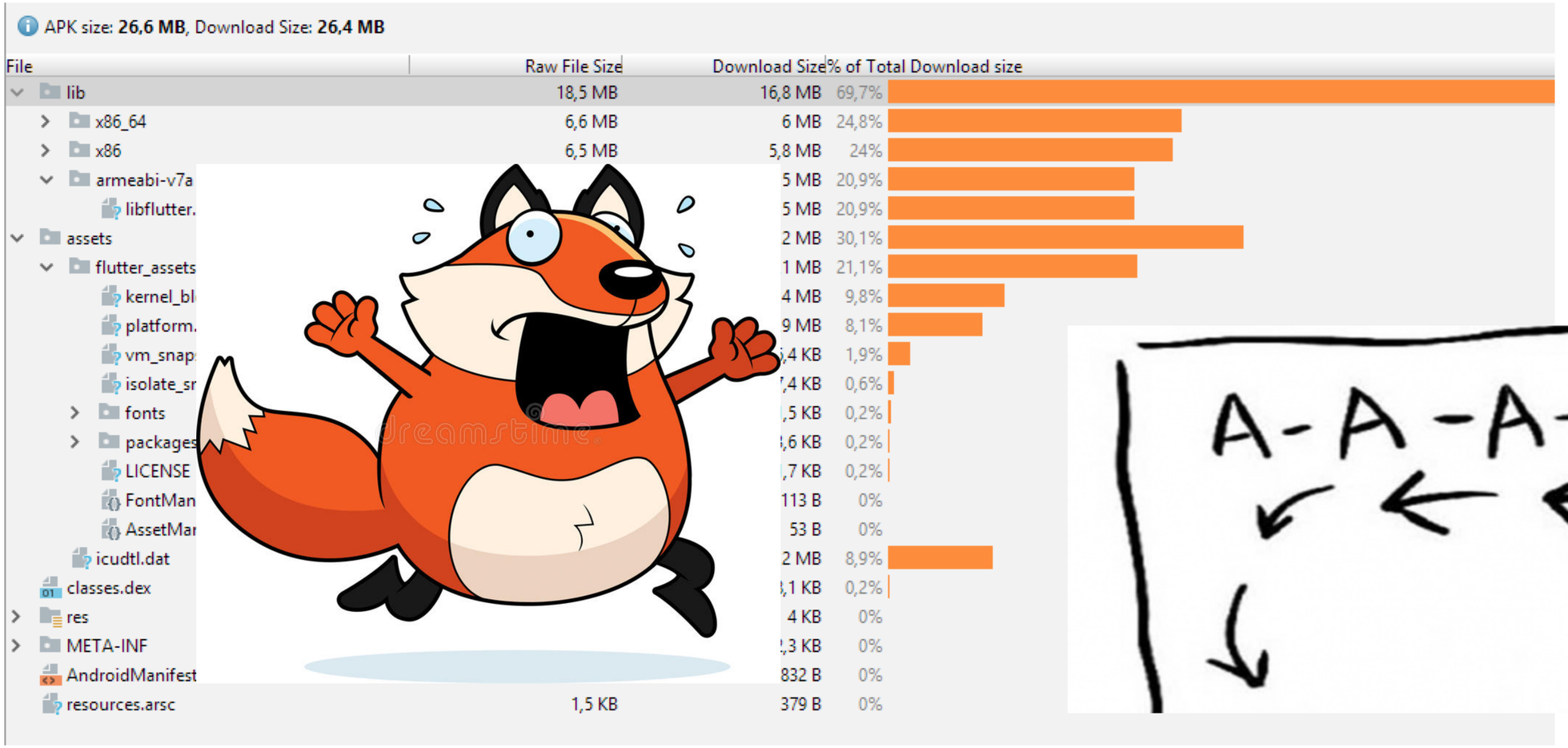
Высокая производительность (похоже, что так, но это не 100%)

- Нативный рендеринг
- Изменение только того, что меняется
- Measure всегда в один проход
- Грамотное использование слоев рендеринга

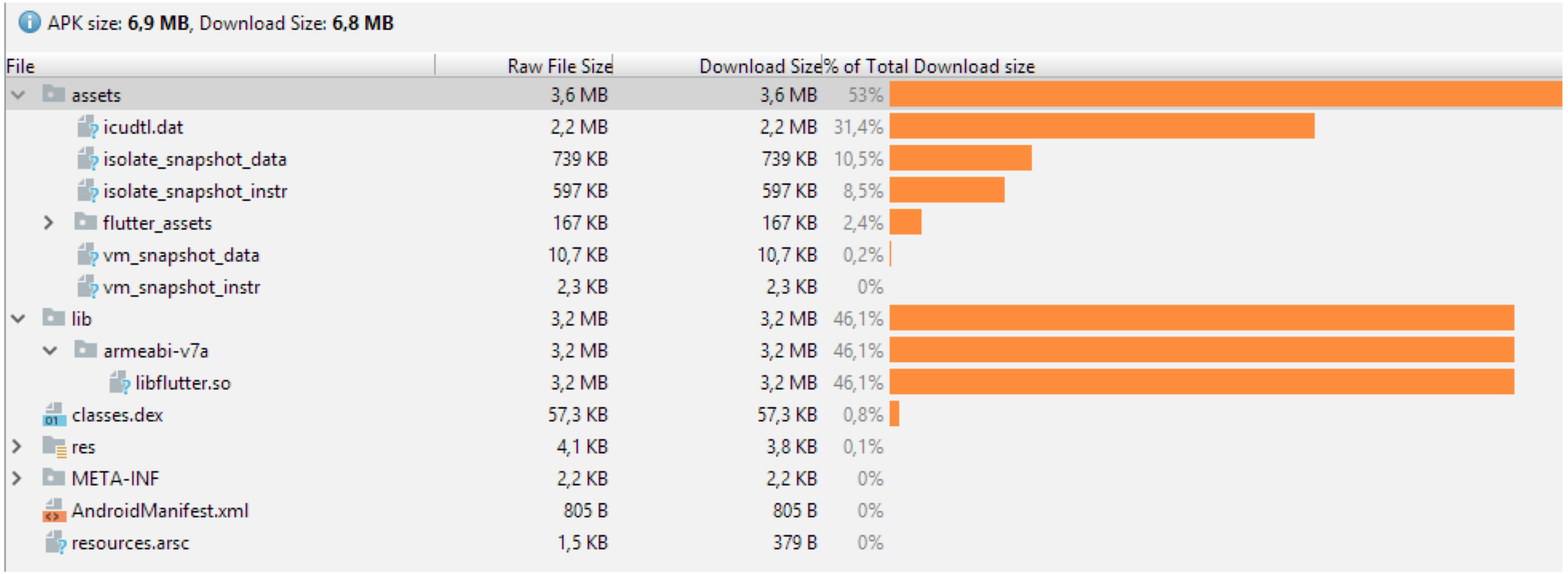
youtube.com/watch?v=UUfXWzp0-DU

А вот теперь
о недостатках...

Что нам стоит Flutter взять ...



Давайте все же соберем релиз



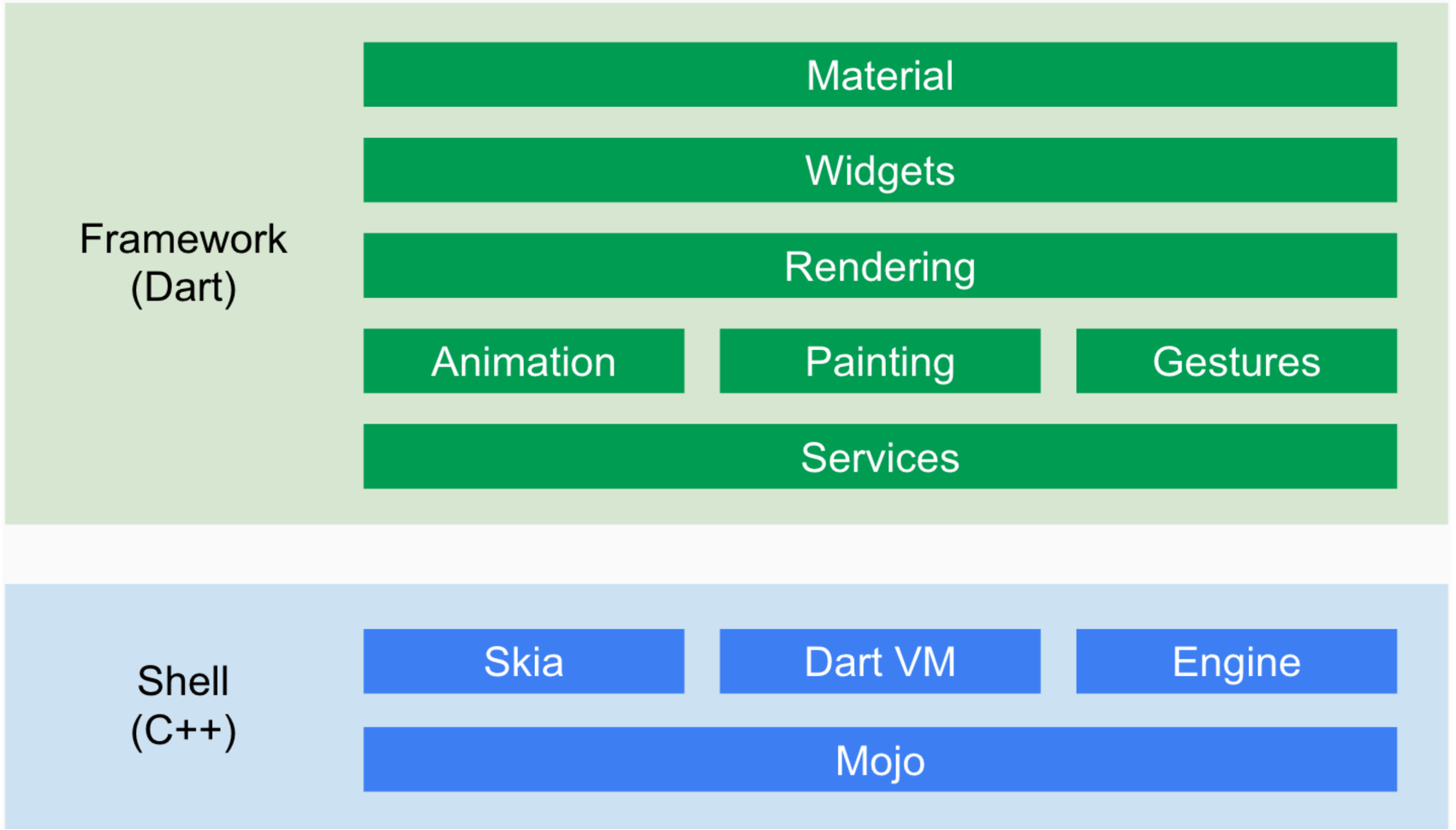
File	Raw File Size	Download Size	% of Total Download size
assets	3,6 MB	3,6 MB	53%
icudtl.dat	2,2 MB	2,2 MB	31,4%
isolate_snapshot_data	739 KB	739 KB	10,5%
isolate_snapshot_instr	597 KB	597 KB	8,5%
flutter_assets	167 KB	167 KB	2,4%
vm_snapshot_data	10,7 KB	10,7 KB	0,2%
vm_snapshot_instr	2,3 KB	2,3 KB	0%
lib	3,2 MB	3,2 MB	46,1%
armeabi-v7a	3,2 MB	3,2 MB	46,1%
libflutter.so	3,2 MB	3,2 MB	46,1%
classes.dex	57,3 KB	57,3 KB	0,8%
res	4,1 KB	3,8 KB	0,1%
META-INF	2,2 KB	2,2 KB	0%
AndroidManifest.xml	805 B	805 B	0%
resources.arsc	1,5 KB	379 B	0%

Главный вопрос жизни, вселенной и всего такого

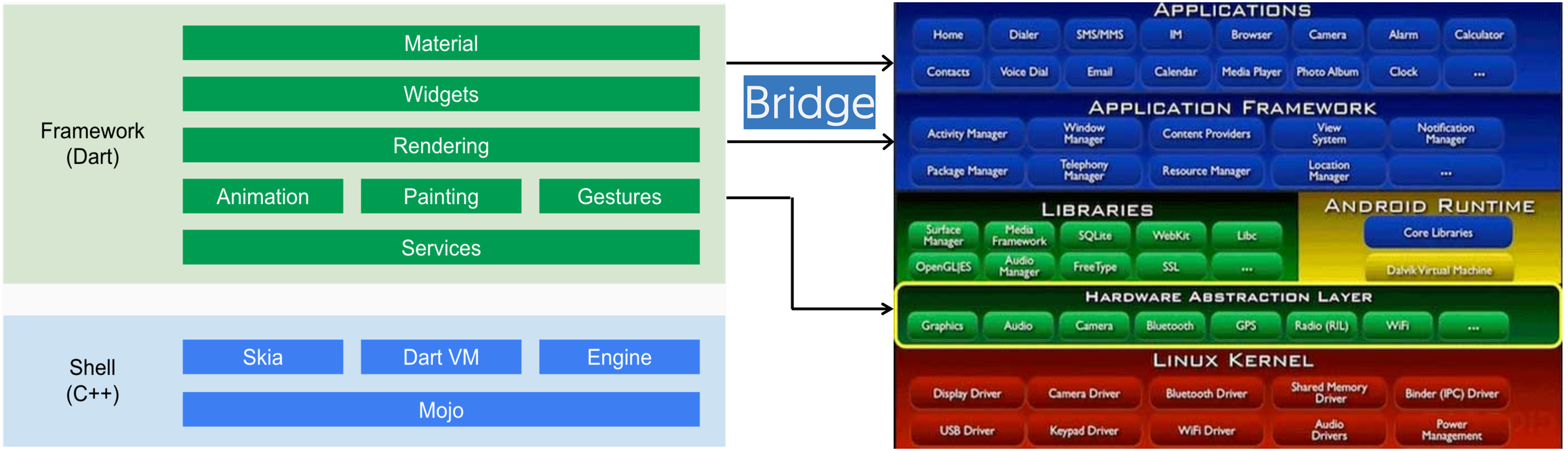
Вопрос

Как мне достучаться до Activity Lifecycle, пермишенов, камеры, WebView, Firebase, Google Maps, других приложений и всего такого?

Flutter – устройство



Flutter – устройство



Method channel – Dart

```dart
static const channel = const MethodChannel('ru.artur_vasilov.channel');
```

Method channel – Dart

```
void browseUrl() async {
 bool result;
 try {
   result =
       await channel.invokeMethod('browse_url', {'url': 'https://flutter.io'});
 } on PlatformException catch (e) {
   result = false;
 }

 setState(() {
   _urlOpened = result;
 });
}
```

Method channel – Native

```
new MethodChannel(getFlutterView(), CHANNEL_NAME)
        .setMethodCallHandler((methodCall, result) -> {
            if (BROWSE_URL_METHOD.equals(methodCall.method)) {
                String url = methodCall.argument("url");
                boolean shown = browseUrl(url);
                result.success(shown);
            } else {
                result.notImplemented();
            }
        });
```

Method channel – Native

```
new MethodChannel(getFlutterView(), CHANNEL_NAME)
        .setMethodCallHandler((methodCall, result) -> {
            if (BROWSE_URL_METHOD.equals(methodCall.method)) {
                String url = methodCall.argument("url");
                boolean shown = browseUrl(url);
                result.success(shown);
            } else {
                result.notImplemented();
            }
        });
```

Стандартные Method Channel

```
public class FlutterView extends SurfaceView {
    private final MethodChannel mFlutterLocalizationChannel;
    private final MethodChannel mFlutterNavigationChannel;
    private final BasicMessageChannel<Object> mFlutterKeyEventChannel;
    private final BasicMessageChannel<String> mFlutterLifecycleChannel;
    private final BasicMessageChannel<Object> mFlutterSystemChannel;
    private final BasicMessageChannel<Object> mFlutterSettingsChannel;

    // ...
}
```

Flutter plugins

shared_preferences 100
Flutter plugin for reading and writing simple key-value pairs. Wraps NSUserDefaults on iOS and SharedPreferences on Android.
v 0.4.2 • Updated: Jun 1, 2018 FLUTTER

path_provider 100
Flutter plugin for getting commonly used locations on the Android & iOS file systems, such as the temp and app data directories.
v 0.4.1 • Updated: Jun 1, 2018 FLUTTER

url_launcher 100
Flutter plugin for launching a URL on Android and iOS. Supports web, phone, SMS, and email schemes.
v 3.0.3 • Updated: Jul 10, 2018 FLUTTER

firebase_auth 100
Flutter plugin for Firebase Auth, enabling Android and iOS authentication using passwords, phone numbers and identity providers like Google, Facebook and Twitter.
v 0.5.18 • Updated: Aug 10, 2018 FLUTTER

cloud_firestore 100
Flutter plugin for Cloud Firestore, a cloud-hosted, noSQL database with live synchronization and offline support on Android and iOS.
v 0.7.4 • Updated: Jul 23, 2018 FLUTTER

google_sign_in 100
Flutter plugin for Google Sign-In, a secure authentication system for signing in with a Google account on Android and iOS.
v 3.0.4 • Updated: Jun 1, 2018 FLUTTER

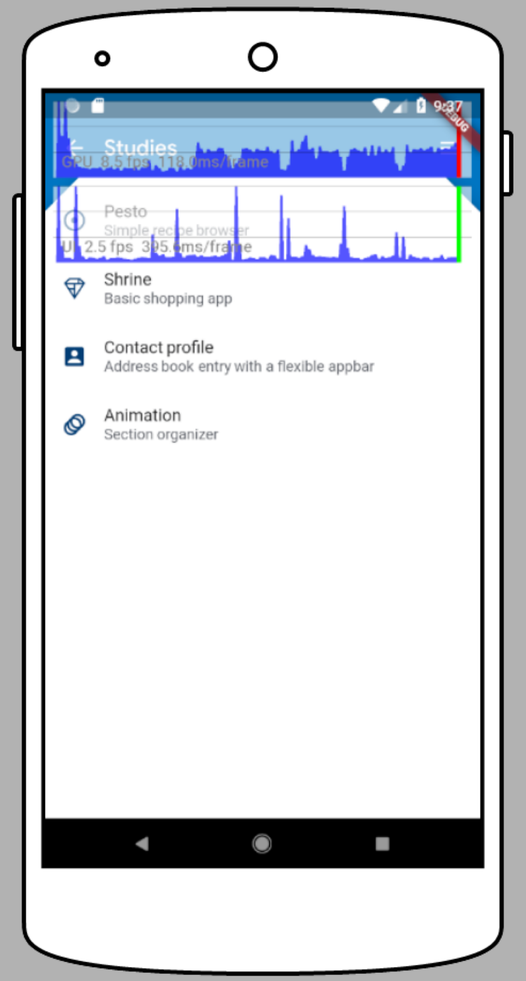
image_picker 100
Flutter plugin for selecting images from the Android and iOS image library, and taking new pictures with the camera.
v 0.4.6 • Updated: Jul 26, 2018 FLUTTER

pub.dartlang.org/flutter/packages

Как померять производительность и вообще все вот это

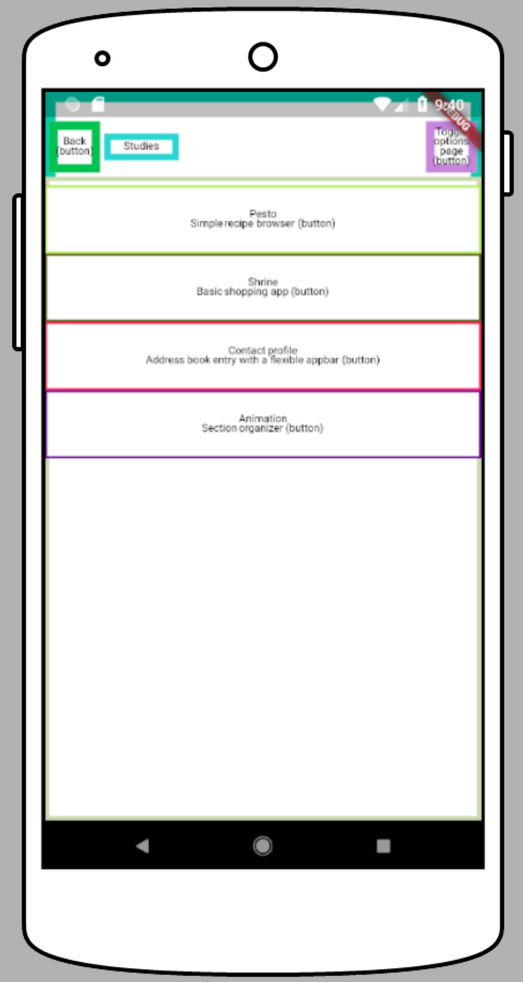
Следим за фреймами

```
@override
Widget build(BuildContext context) {
 return MaterialApp(
   debugShowCheckedModeBanner: true,
   showPerformanceOverlay: true,
   showSemanticsDebugger: true,
   // ..
   title: 'Hello, World',
   home: Scaffold(
   ),
 );
}
```

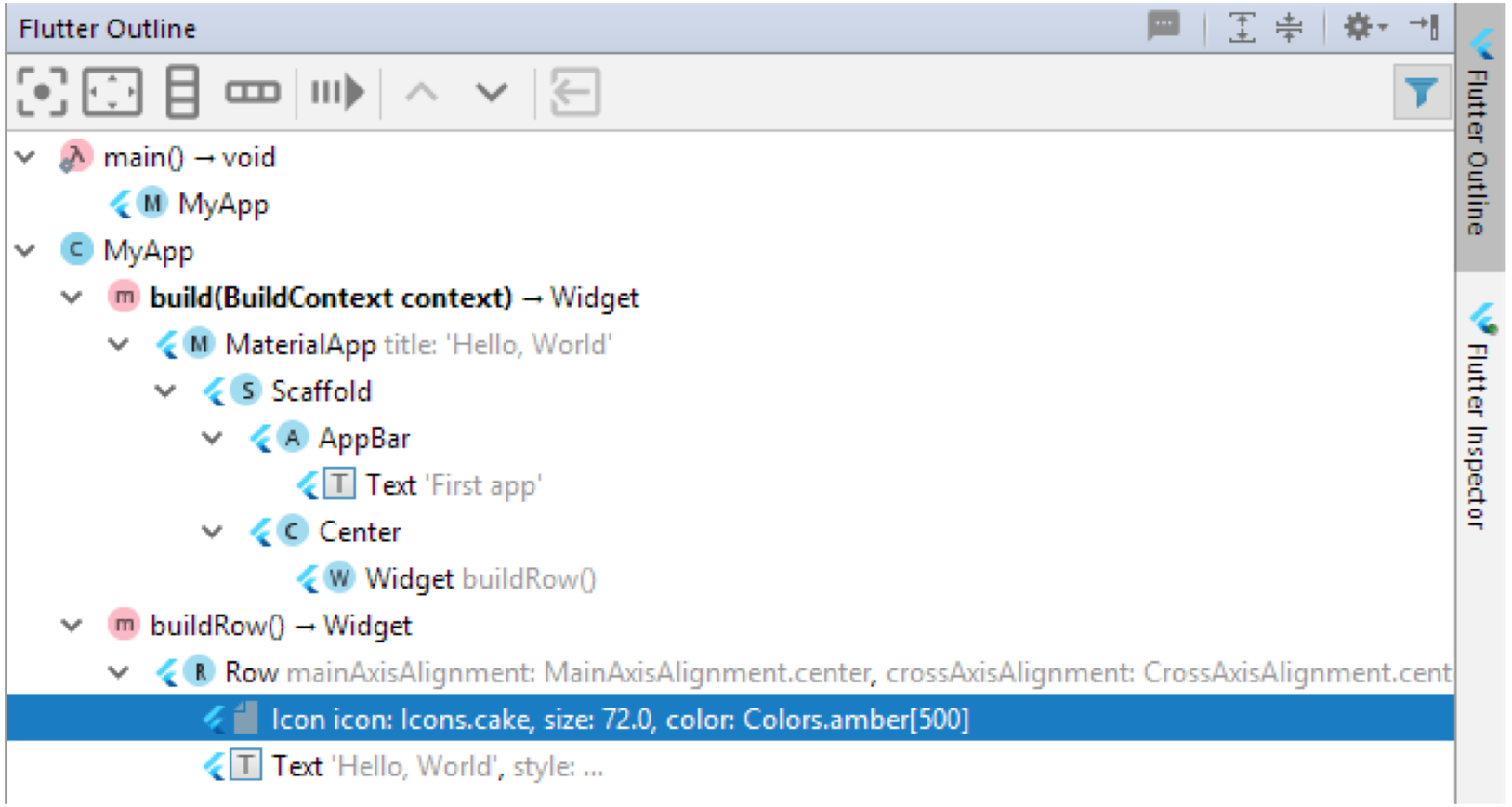


Странный отладчик

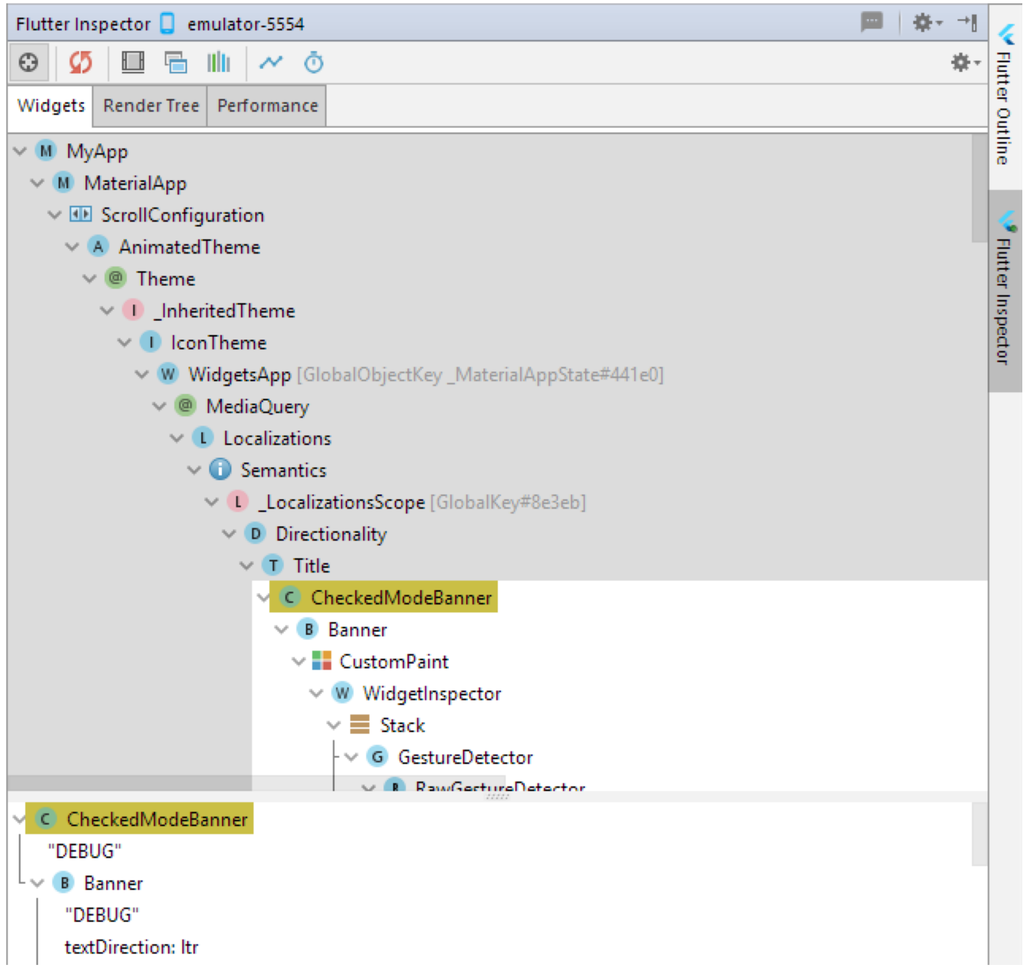
```
@override
Widget build(BuildContext context) {
 return MaterialApp(
   debugShowCheckedModeBanner: true,
   showPerformanceOverlay: true,
   showSemanticsDebugger: true,
   // ..
   title: 'Hello, World',
   home: Scaffold(
   ),
 );
}
```

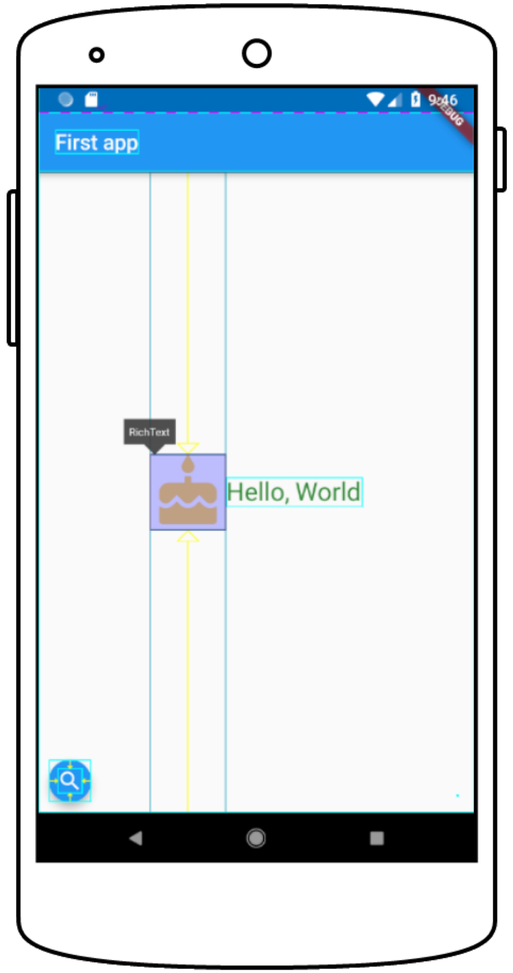


Flutter Outline



Flutter Inspector





Недостатки

- Вся платформенная интеграция через дополнительные нативные вызовы и плагины
- Дополнительный + к размеру приложения
- Недостаточно мощный инструментарий
- Необходимость учить новый язык и подход (?)

Итого

- Это интересно и стоит поиграться
- Пока плохо подходит для больших проектов
- Не стоит слишком верить (вспоминаем добрый словом RN)
- Следим за развитием

Ссылки

- flutter.io/
- github.com/flutter/flutter
- udacity.com/course/build-native-mobile-apps-with-flutter--ud905
- speakerdeck.com/arturvasilov/miesto-flutter-v-zhizni-android-razrabotchika
- ...

Место Flutter в жизни Android-разработчика

Артур Василов

Android-разработчик

 avasilov@yandex-team.ru

 VasilovArtur

 ArturVasilov

 ArturVasilov