

I'm in!



# Flutter

mobile guild  
#cododajnia

intive

Bartosz Kosarzycki  
@bkosarzycki

# What is Flutter?

- multi-platform - Android & iOS
- **high performance, low latency**
- DART as main language
- open-source / github
- “flutter” frame render / rapid variation of signal parameters
- not a monolith structure  
*access to, and control over, all layers of the system*
- custom ui-rendering engine



# Dart

# GitHub

# Google



# What's more?

- material design on iPhones
- drawer, FAB implementation on iPhones
- change iphone/android behaviour style dynamically
- iPhone navigation style on Android
- hot reload of classes
- <https://www.youtube.com/watch?v=iPIPk4>
- AOT compilation for iOS & Android



# Dart

# GitHub

# Google



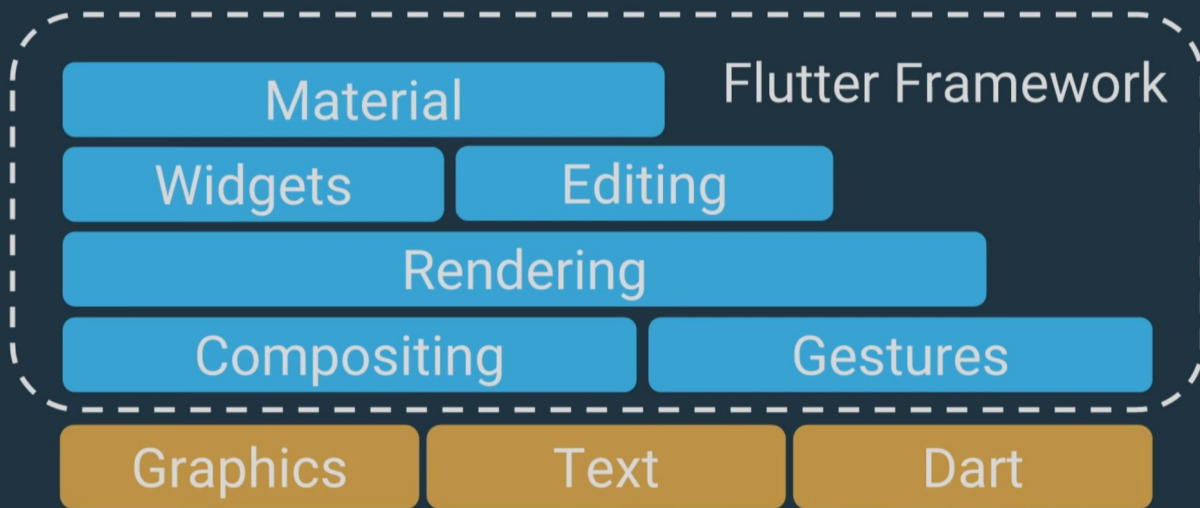
# Architecture

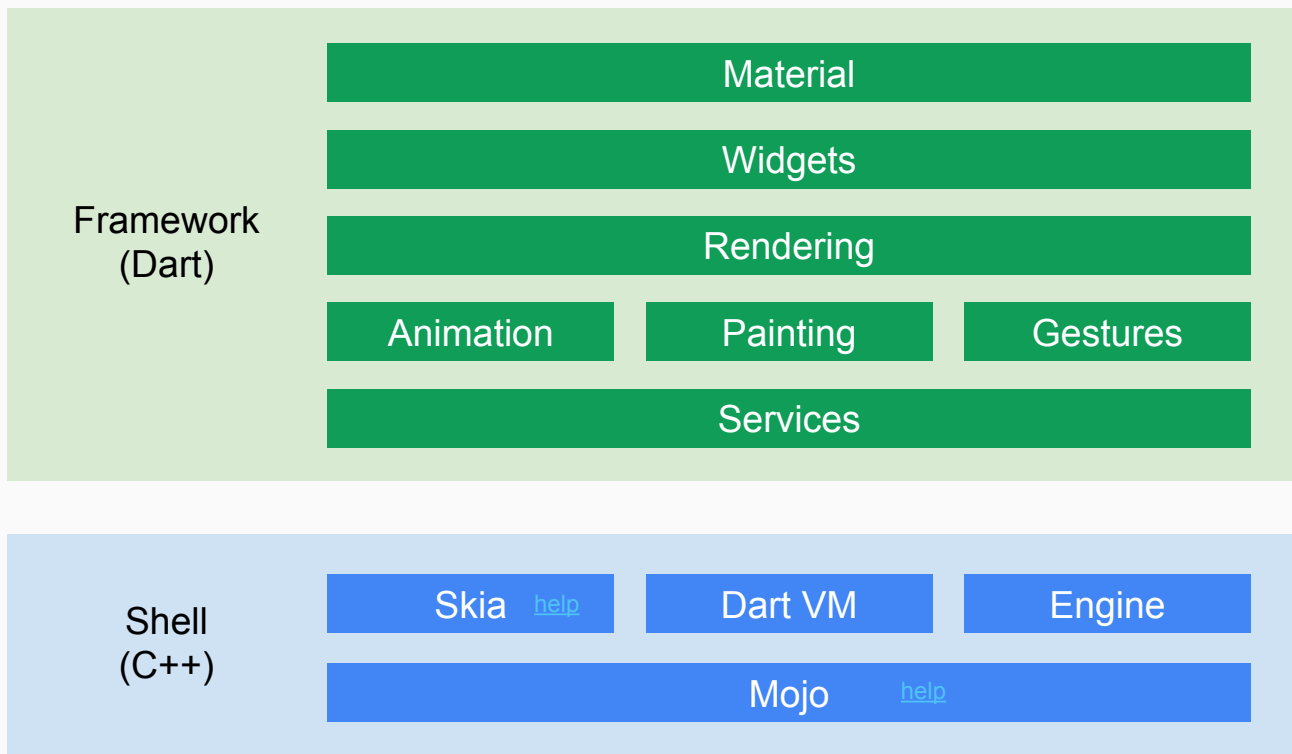
- a heavily optimized, mobile-first 2D **rendering engine** (with excellent support for text)
- a functional-**reactive** framework
- a set of Material Design **widgets** (which can be extended)
- command-line tools and plugins for [IntelliJ IDEA](#)
- highly productive and fast development experience



# Architecture

Devs tell us: no monolith please.





source: flutter.io

# Technology

- C, C++, Dart, Skia (a 2D rendering engine), Mojo IPC, and Blink's text rendering system

# Compatibility

- Android: Jelly Bean, v16, 4.1.x or newer,
- iPhone: iOS 8 or newer
- emulator / simulator

# Performance


- constant 60 fps





# Dart



BSD license 



slimmed down  
graphics stack  
from CHROME

**2 million lines of production  
code in use**

Actively in use  
at Google  




slimmed down  
graphics stack  
from ANDROID

***since 2011***

# JavaScript



Scala

Java

TypeScript

# DART



# Language features

- dynamically typed language with optional types
- **dart SDK** - includes libraries & command-line tools
- **dart VM** - online version - [dartpad](#)
- **dartanalyzer** - statically analyzes code, helping you catch errors early
- **pub** - package manager
- **dart2js** - Dart-to-JavaScript compiler
- **dartfmt** - Dart code formatter



# In short...

## Variables & constants:

```
var name = 'Voyager I';  
var year = 1977;  
final bar = const [];  
const baz = const [];
```

## Getters and setters

```
class Spacecraft {  
    DateTime launchDate ;  
    int get launchYear => launchDate?.year ;  
}
```

## Functions:

```
int fibonacci(int n) {  
    if (n == 0 || n == 1) return n;  
    return fibonacci(n - 1) + fibonacci(n - 2);  
}
```

## Lambda (fat-arrow) expressions:

```
flybyObjects.where((name) => name.contains('flower')).forEach(print);
```



# Language features

- dart **native** extensions enable dart to run C/C++ code - [check out](#)
- public package repository at <https://pub.dartlang.org/packages>  
*package imports:*  
`import 'package:vector_math/vector_math.dart';`
- “everything is an object”, even - numbers, functions, and nulls are objects
- You can create top-level functions and also create functions within functions
- reified generics support
- mixins



# In short...

## Async await:

```
String news = await gatherNewsReports();

Future gatherNewsReports() async {
  String path = 'https://www.dartlang.org/f/dailyNewsDigest.tx';
  return (await HttpRequest.getString(path));
}
```

## Mixins:

```
class Manned {
  int astronauts;
  void describeCrew() {
    print('no. $astronauts');
  }
}

class Orbiter extends Spacecraft with Manned {
}
```

## Future API:



```
var future = gatherNewsReports();
future.then((content) => print(content))
  .catchError((e) => handleError(e));
```

# Syntax of choice:

## 'Java' style:

```
String title = "Long title text";  
String title2 = title.substring(5);
```

## 'Scala' style:

```
var title = "Long title text";  
var title2 = title.substring(5);
```

## Lamdas & closures:

```
var addition = () => 2 + 3;  
print(addition());  
  
var errorType = "[ERROR]";  
var closure = (msg) => errorType + " "  
+ msg;  
print(closure("Some error"));
```

## Everything is an object:

```
double a = 3.2;  
var floor = a.floor();  
(floor + 2).abs();
```



# Language syntax:

## Anonymous functions:

```
var list = ['iPhone', 'Android'];  
list.forEach((i) {  
    print(list.indexOf(i).toString() + ': ' + i);  
});
```

## Type checking:

```
if (a is AndroidPhone) {  
    a.macAddress = '00-14-22-01-23-45';  
}
```





# Reified generics:

## Generics are kept at runtime

```
var names = new List<String>();  
names.addAll(['Seth', 'Kathy', 'Lars']);  
  
//check at runtime  
print(names is List<String>);
```

## Parameterized types with constructors

```
var names = new List<String>();  
var nameSet = new Set<String>.from(names);
```



# Reified generics:

## Generics methods

```
class Foo<E> {  
  E _item;  
  
  <T> T transform(T fn(E)) {  
    return fn(_item);  
  }  
}
```

## New instance

```
class B<C> {  
  C create() {  
    return new C();  
  }  
}
```

**STILL IN DEVELOPMENT**

(as of Nov, 2016)

<https://github.com/dart-lang/sdk/issues/254>



# Optional types in dart

- dart language is dynamically typed
- it is possible to add type annotations to programs
- incomplete or plain wrong type annotations DO NOT prevent compilation
- variables with no specified type get a special type: **dynamic**

Type annotations give a couple of benefits, however:

- Documentation for humans, name completion and improved navigation in IDEs, static checker warns you about potential problems
- more info: [here](#)



# Loading libraries

- Importing only part of a library

```
import 'package:lib1/lib1.dart' show foo;
```

- Lazily loading a library

```
import 'package:deferred/hello.dart' deferred as hello;  
await hello.loadLibrary();
```



# PUB package manager

File name:

/pubspec.yaml

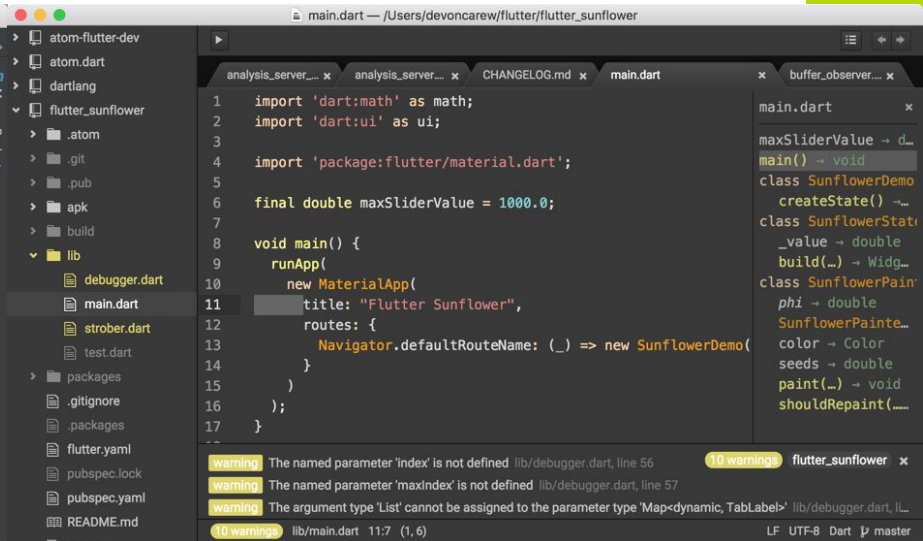
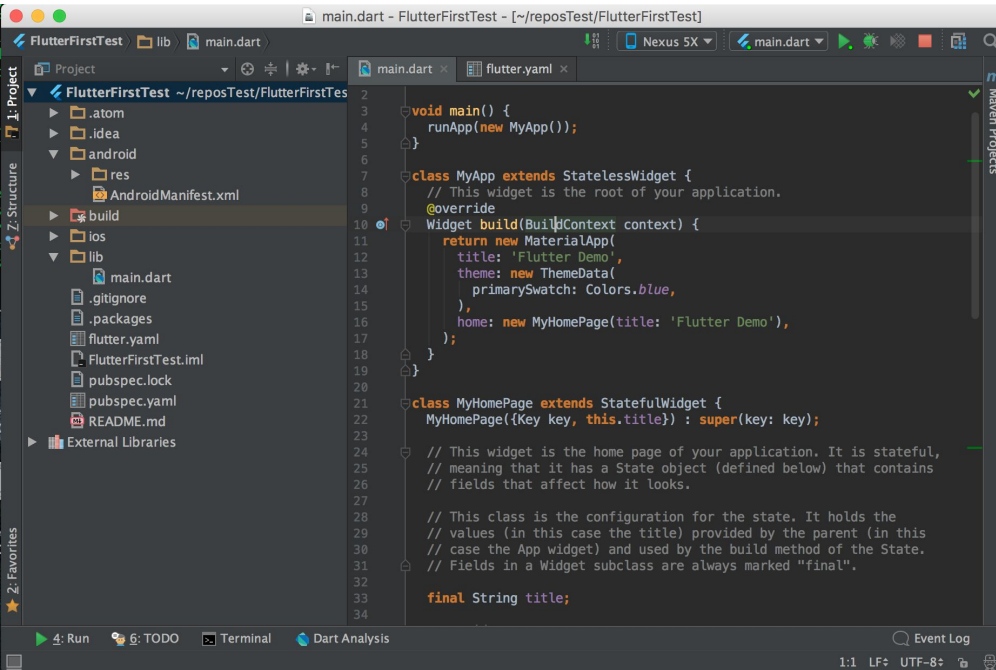
```
name: my_app
dependencies:
  js: ^0.3.0
  intl: ^0.12.4
```

```
$ cd /Users/bko/repos/abcrepo
$ pub get
```

```
name: angular2
version: 2.0.0
author: Dart Team <web@dartlang.org>
description: Framework for modern web applications
homepage: 'https://angulardart.org'
documentation: 'https://angular.io/docs/dart/latest/api/'
environment:
  sdk: '>=1.19.0 <2.0.0'
dependencies:
  analyzer: '>=0.27.0 <0.30.0'
  barback: ^0.15.2+2
  build: '>=0.3.0 <0.5.0'
  dart_style: '>=0.1.8 <0.3.0'
  glob: ^1.0.0
  html: '>=0.12.0 <0.14.0'
  intl: '>=0.13.0 <0.15.0'
  logging: '>=0.9.0 <0.12.0'
  observable: ^0.14.0+1
  path: ^1.0.0
  protobuf: ^0.5.1
  source_span: ^1.0.0
  stack_trace: ^1.6.5
  js: ^0.6.0
```



# IDEs



Also:



# Material design

## ▼ material

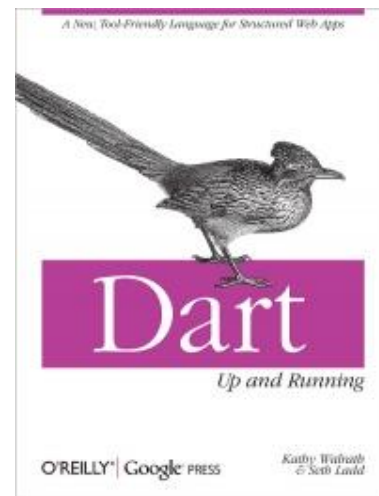
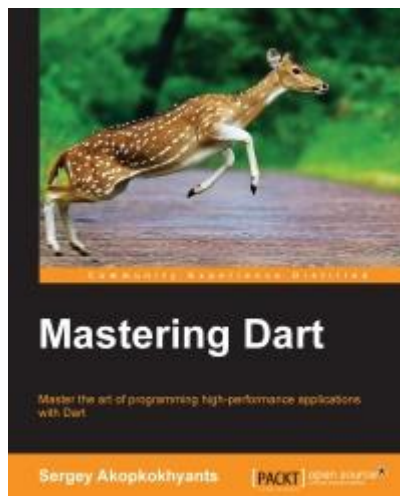
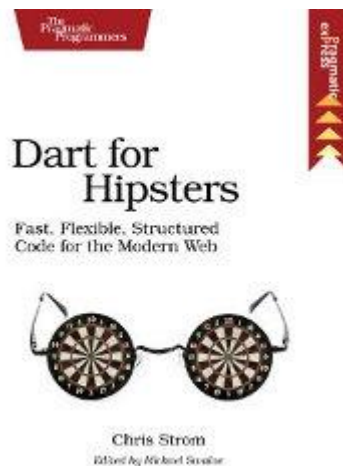
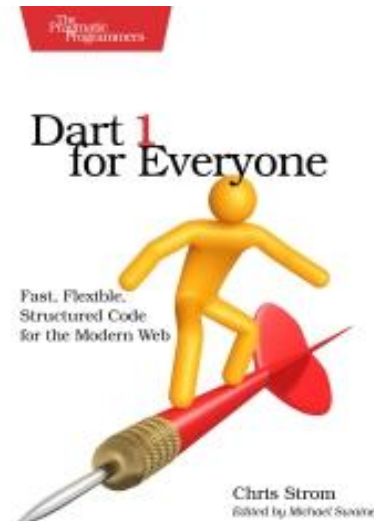
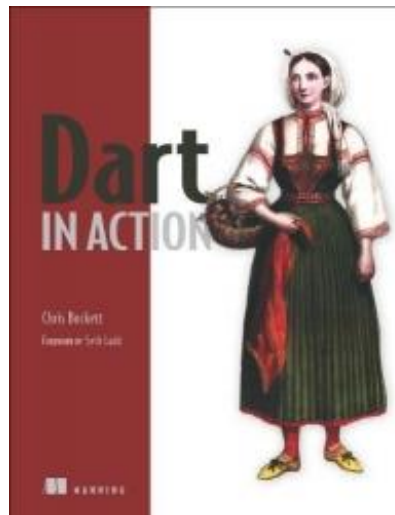
- about.dart
- app.dart
- app\_bar.dart
- arc.dart
- bottom\_navigation\_bar.dart
- bottom\_sheet.dart
- button.dart
- button\_bar.dart
- card.dart
- checkbox.dart
- chip.dart
- circle\_avatar.dart
- colors.dart
- constants.dart
- data\_table.dart
- data\_table\_source.dart
- date\_picker.dart
- debug.dart
- dialog.dart
- divider.dart
- drawer.dart
- drawer\_header.dart

- drawer\_item.dart
- drop\_down.dart
- expand\_icon.dart
- expansion\_panel.dart
- flat\_button.dart
- flexible\_space\_bar.dart
- floating\_action\_button.dart
- flutter\_logo.dart
- grid\_tile.dart
- grid\_tile\_bar.dart
- icon.dart
- icon\_button.dart
- icon\_theme.dart
- icon\_theme\_data.dart
- icons.dart
- image\_icon.dart
- ink\_well.dart
- input.dart
- list.dart
- list\_item.dart
- material.dart
- mergeable\_material.dart
- overscroll\_indicator.dart
- page.dart
- paginated\_data\_table.dart

- paginated\_data\_table.dart
- popup\_menu.dart
- progress\_indicator.dart
- radio.dart
- raised\_button.dart
- refresh\_indicator.dart
- scaffold.dart
- scrollbar.dart
- shadows.dart
- slider.dart
- snack\_bar.dart
- stepper.dart
- switch.dart
- tabs.dart
- text\_selection.dart
- theme.dart
- theme\_data.dart
- time\_picker.dart
- toggable.dart
- tooltip.dart
- two\_level\_list.dart
- typography.dart
- user\_accounts\_drawer\_header.dart



# Books





# Tools

- gitignore file - [.gitignore](#)
- environment variables

```
$ nano ~/.bash_profile
```

.gitignore:

```
.DS_Store  
.atom/  
.idea  
.packages  
.pub/  
build/  
packages/  
pubspec.lock
```

GNU nano 2.0.6

File: /Users/bko/.bash\_profile

```
export PATH="/Users/bko/flutter/flutter/bin":"/Users/bko/flutter/flutter/bin/cache/dart-sdk/bin"
```

```
flutter pub dart dart2js dartanalyzer etc...
```



# Project checkout / get dependencies

- pub get

```
Bartoszs-MacBook-Pro:spacewar bko$ cat pubspec.yaml
name: spacewar
dependencies:
  sky: any
  flutter: any
  sky_tools: any
  vector_math: any
dependency_overrides:
  flutter:
    path: /Users/bko/flutter/flutter/packages/flutter
```

```
Bartoszs-MacBook-Pro:spacewar bko$ pub get
Resolving dependencies... (3.3s)
+ async 1.11.3
+ charcode 1.1.0
+ collection 1.9.1 (1.11.0 available)
+ flutter_services 0.0.1 from path ../../flutter/flutter/bin/cache/pkg/flutter_services
+ http_parser 3.0.3
+ intl 0.14.0
+ meta 1.0.4
+ mojo 0.4.27 (0.4.28 available)
+ mojo_sdk 0.2.31 (0.2.32 available)
+ mojo_services 0.4.34 (0.4.35 available)
+ path 1.4.0
+ sky 0.0.56
+ sky_engine 0.0.99 from path ../../flutter/flutter/bin/cache/pkg/sky_engine
+ sky_services 0.0.99 from path ../../flutter/flutter/bin/cache/pkg/sky_services
+ sky_tools 0.0.38
+ source_span 1.3.0
+ string_scanner 1.0.1
+ vector_math 2.0.3
Warning: You are using these overridden dependencies:
! flutter 0.0.21 from path /Users/bko/flutter/flutter/packages/flutter
Downloading string_scanner 1.0.1...
Downloading source_span 1.3.0...
Downloading sky_tools 0.0.38...
Downloading sky 0.0.56...
Changed 19 dependencies!
Precompiling dependencies...
Loading source assets...
Precompiled intl.
Precompiling executables...
Loading source assets...
Precompiled vector_math:mesh_generator.
Bartoszs-MacBook-Pro:spacewar bko$ pub get
Resolving dependencies...
```



# Run the application

- open iPhone simulator / Android emulator

```
open -a Simulator.app
```

```
emulator -avd Nexus_5X_API_24
```

- check connected devices

```
Bartoszs-MacBook-Pro:spacewar bko$ flutter devices  
1 connected device:
```

```
iPhone SE • 4DC7D2BF-B884-41F4-9F1C-A68648D2A503 • ios • iOS 10.0 (simulator)
```

```
Bartoszs-MacBook-Pro:spacewar bko$ flutter devices  
1 connected device:
```

```
Android SDK built for x86 64 • emulator-5554 • android-x64 • Android 7.0 (API 24) (emulator)
```



# Run the application

- run in debug mode

```
flutter run
```

- install the app

```
flutter install
```

- run in 'release' mode (not supported for emulators)

```
flutter run --release
```

- run on a specific device

```
flutter run -d emulator-5554
```

```
flutter run -d 5B5BB06C-8368-41CE-A79E-296880EE3A24
```



# Run the application

- run lint analysis

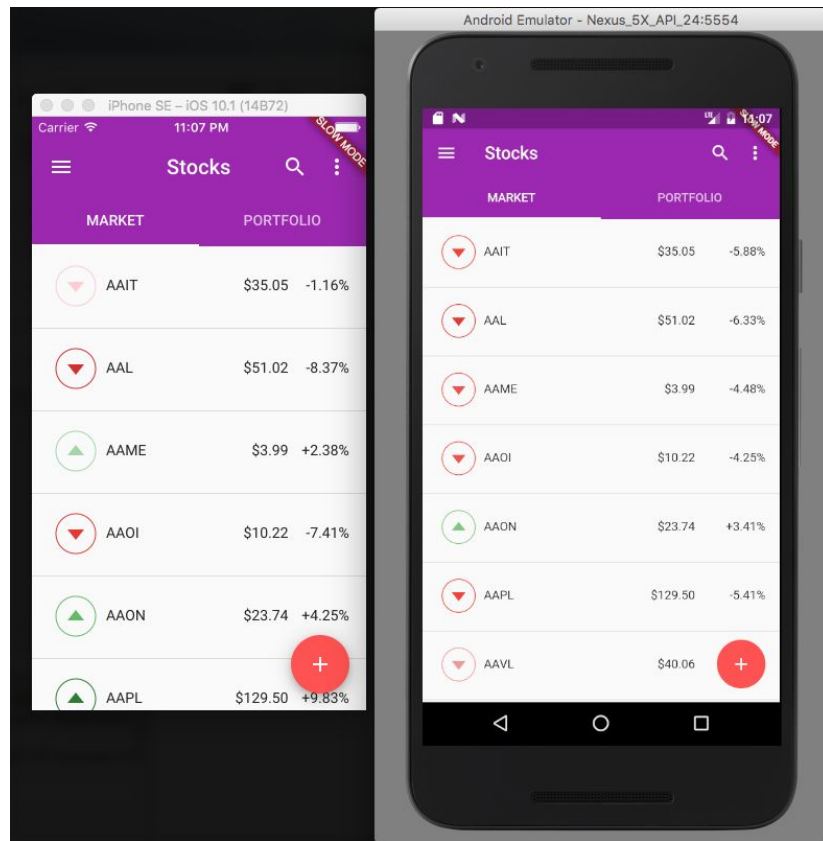
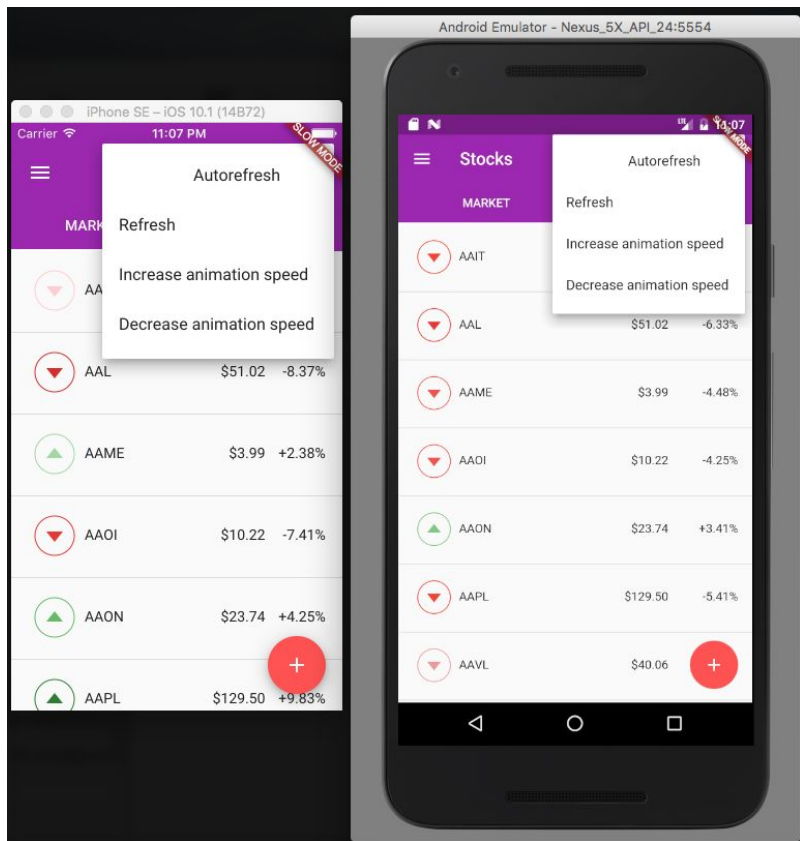
```
Bartoszs-MacBook-Pro:stocks bko$ flutter analyze
Analyzing 17 files...
[lint] Don't import implementation files from another package. (lib/i18n/stock_messages_all.dart, line 9, col 8)
[warning] 'lib' is not a method (lib/i18n/stock_messages_all.dart, line 34, col 54)
(Ran in 7.5s)
```

- run tests

```
Bartoszs-MacBook-Pro:stocks bko$ flutter test
00:01 +2: All tests passed!
```



# Platforms comparison



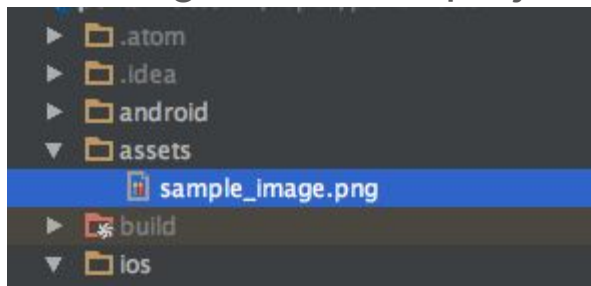
# Using resources

- to add an image edit **flutter.yaml** file

```
Document 1/1  name:
1  name: peka_flutter
2  uses-material-design: true
3
4  assets:
5    - assets/sample_image.png
```

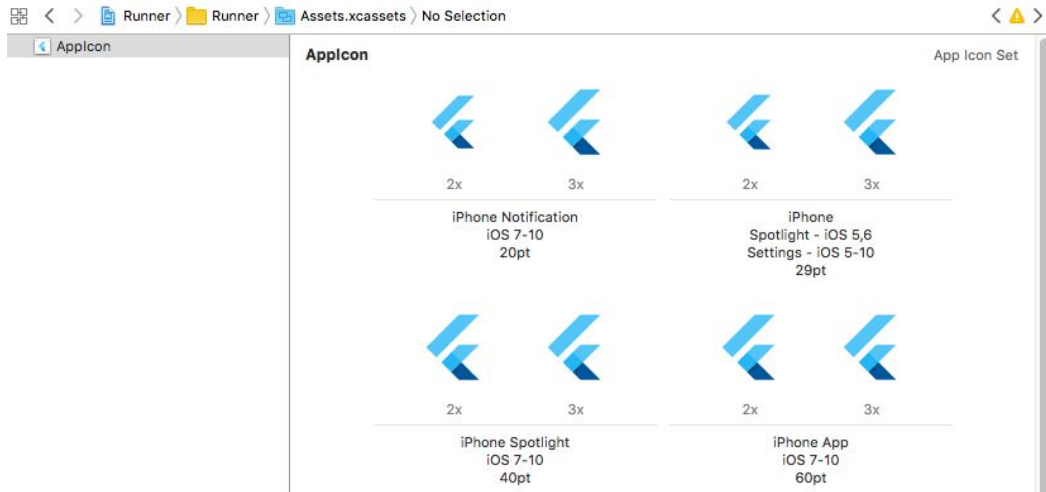
```
final Map<_Page, List<_CardData>> _allPages = <_Page, List<_CardData>>{
  new _Page(label: 'LEFT'): <_CardData>[
    const _CardData(
      title: 'Vintage car',
      imageAsset: 'assets/sample_image.png'
    )
  ]
};
```

- add image file to the project:

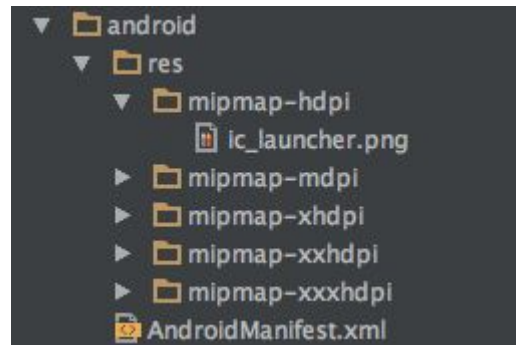


# App icon

## iPhone

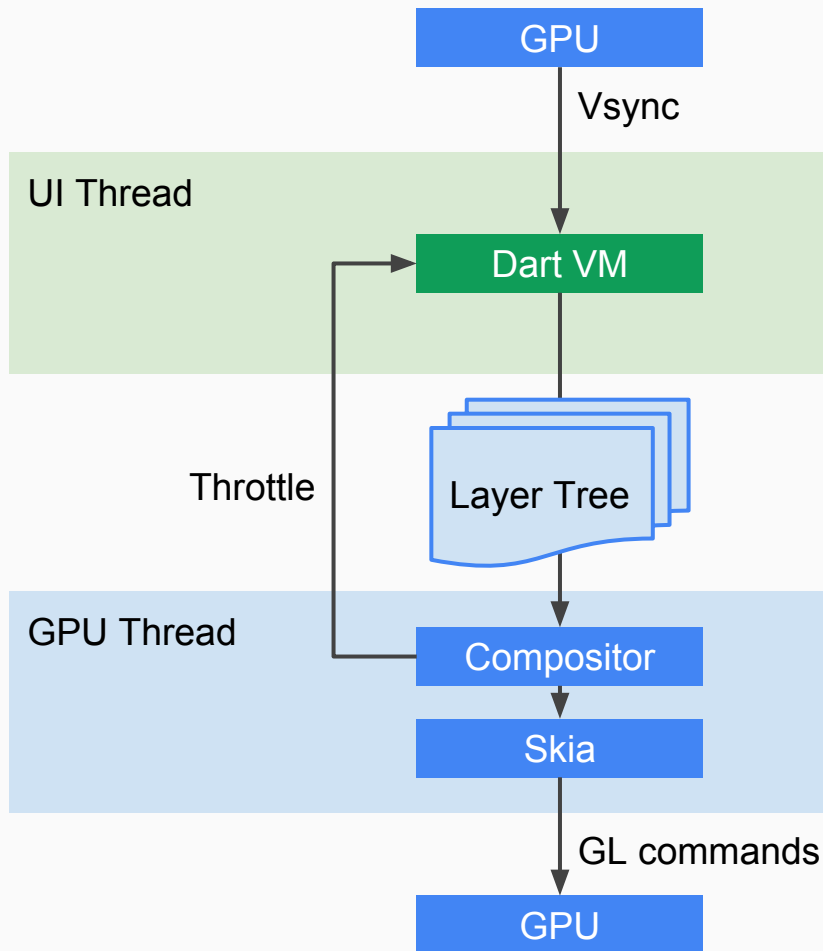


## Android

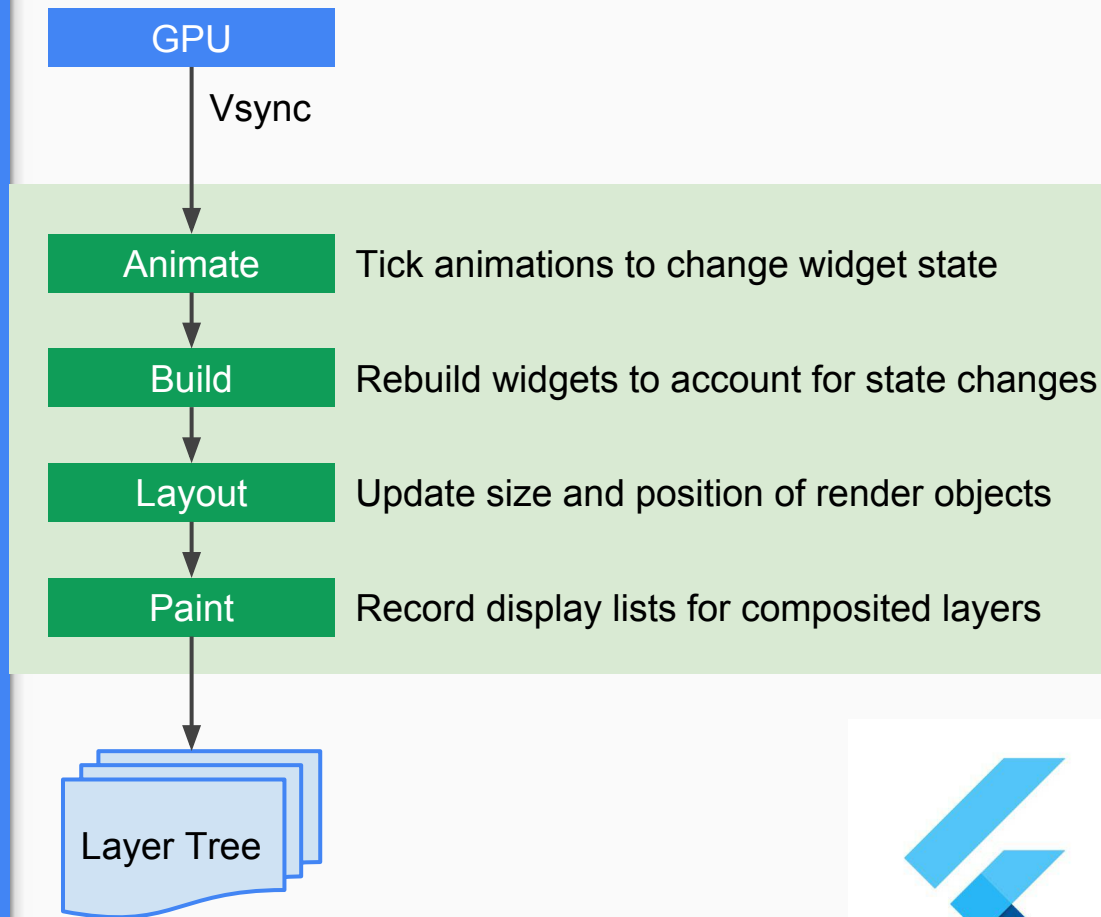


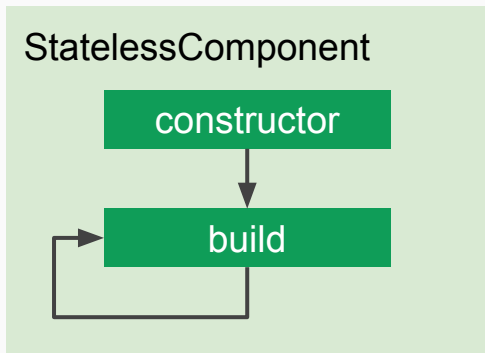


# Graphics Pipeline

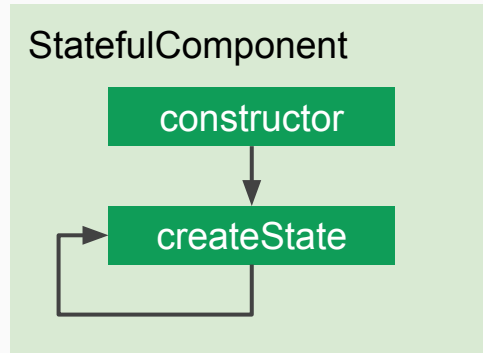


# Rendering Pipeline





A single `StatelessComponent` can build in many different `BuildContext`s



A `StatefulWidget` creates a new `State` instance for each `BuildContext`

# Links

- Dart language articles: <https://www.dartlang.org/articles/language>
- Online Dartpad: <https://dartpad.dartlang.org/>
- [Style](#), [Usage](#), [Design](#)
- Dart academy: <https://dart.academy/>
- Beginners: [part1](#), [part2](#)
- Dart news: <http://news.dartlang.org/>
- Flutter layered design: <https://www.youtube.com/watch?v=dkyY9WCGMi0>
- Sky (previous version): <https://www.youtube.com/watch?v=PnIWl33YMwA>



Thank you!



QUESTIONS

Bartosz Kosarzycki  
@bkosarzycki