

# Flutter и миллион строк нативного кода

Mikhail Zinov  
Software Engineer, Grab

**Приложение**

**Почему Flutter**

**Интеграция**

**Влияние на метрики**

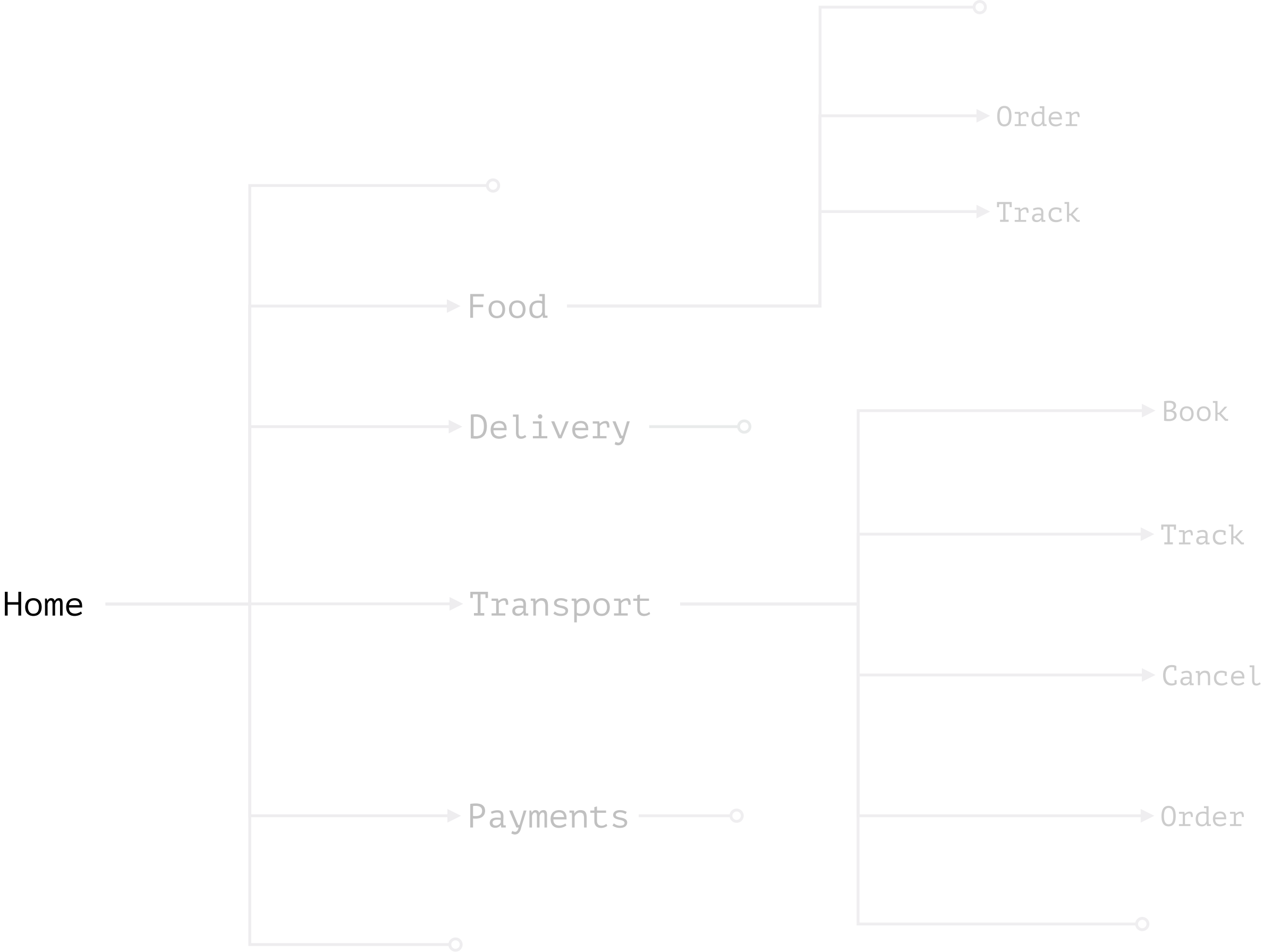
**Приложение**

Почему Flutter

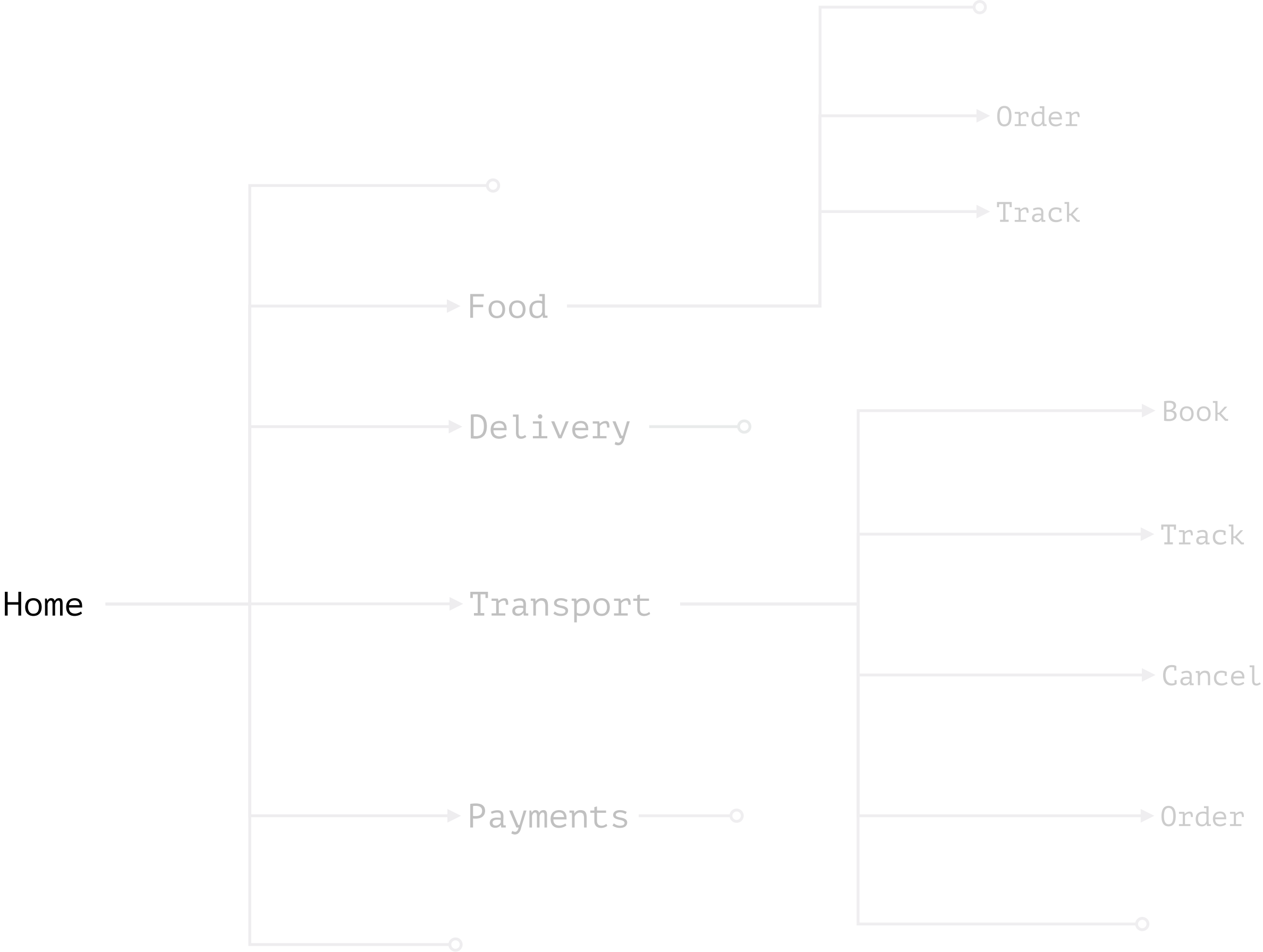
Интеграция

Влияние на метрики

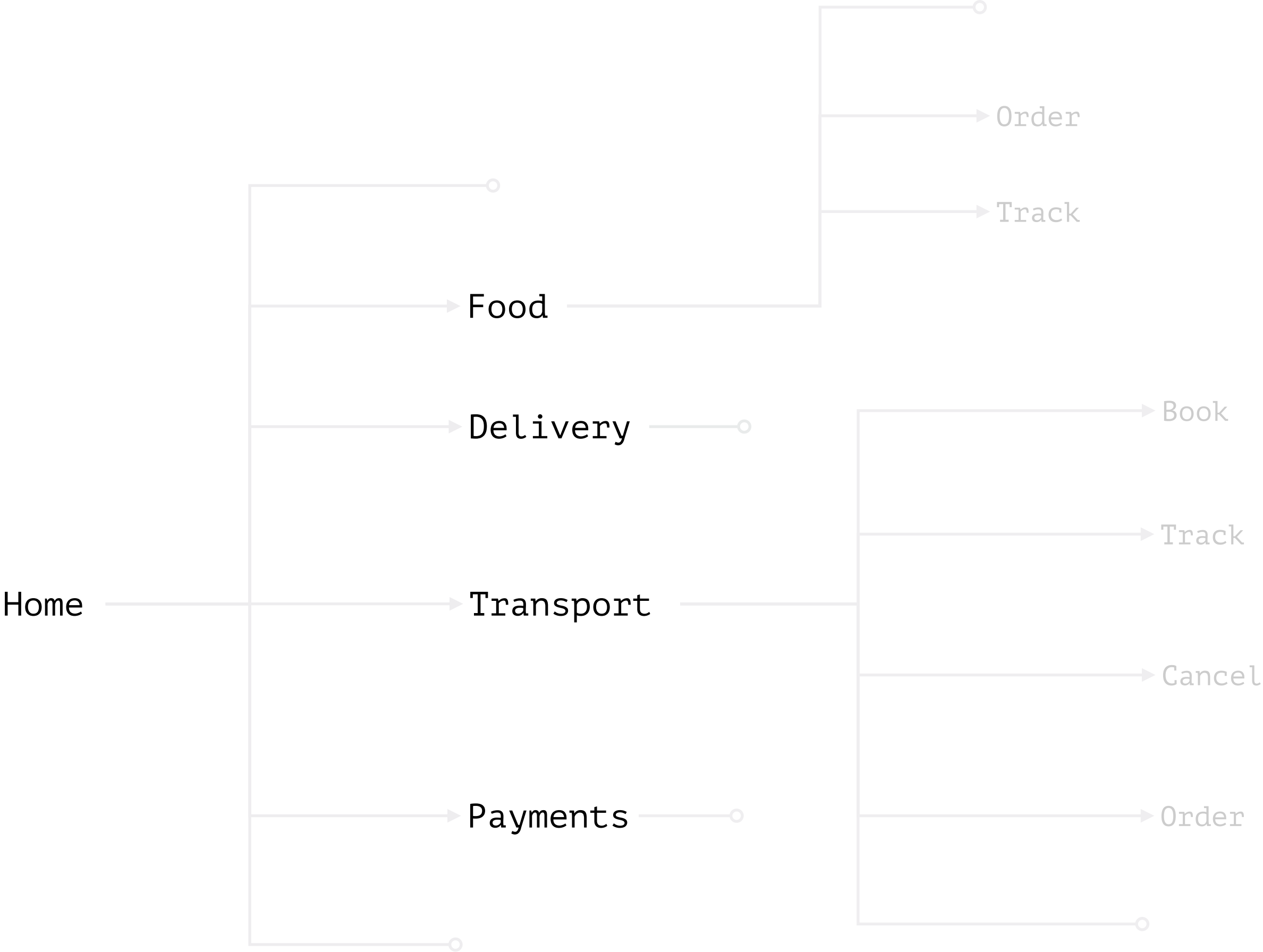
# Grab Super App



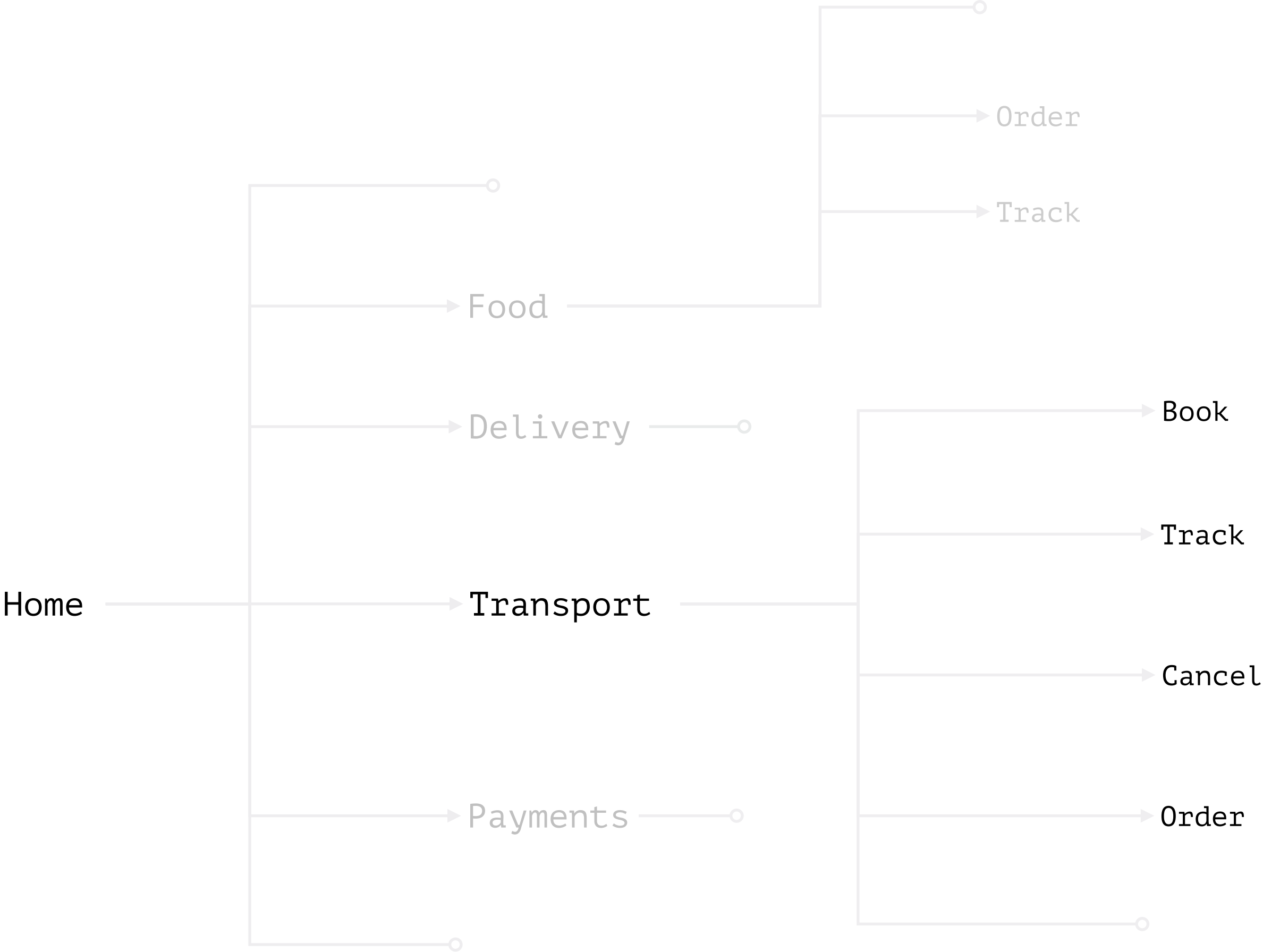
# Grab Super App



# Grab Super App



# Grab Super App



Приложение

**Почему Flutter**

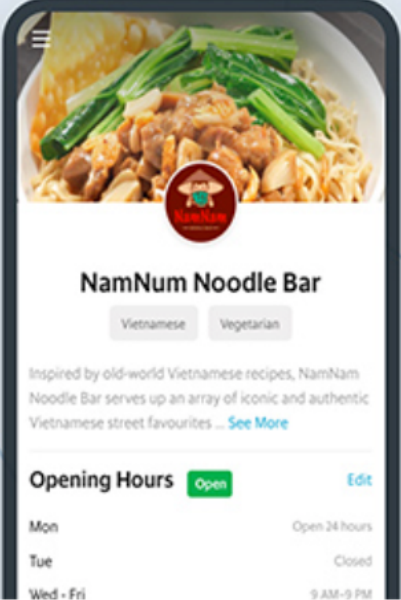
Интеграция

Влияние на метрики



# Grab Food Merchant App

Позитивный фидбэк от разработчиков и бизнеса



The screenshot shows the merchant app interface for 'NamNum Noodle Bar'. At the top is a large image of a bowl of noodles with chicken and vegetables. Below the image, the restaurant name 'NamNum Noodle Bar' is displayed, followed by 'Vietnamese' and 'Vegetarian' tags. A short description follows: 'Inspired by old-world Vietnamese recipes, NamNum Noodle Bar serves up an array of iconic and authentic Vietnamese street favourites ... See More'. Below this is an 'Opening Hours' section with a green 'Open' button and an 'Edit' link. The hours are listed as: Mon - Open 24 hours, Tue - Closed, and Wed - Fri - 9 AM - 9 PM.

**Grab**

Flutter helped Grab build the merchant app for its fast-growing food delivery business.

[Download](#) ▾

# Проблемы с React Native

Падения в проде

Затратное по времени обновление

Процесс разработки

# Почему не отказаться от cross-platform

Скорость cross-platform разработки

Большое время сборки нативных проектов

Консистентность UI и функционала

Приложение

Почему Flutter

**Интеграция**

Влияние на метрики

**Что предлагает Flutter?**

```
flutter create --template=module mobius_flutter
```

## **mobius\_flutter/**

```
|— .android/  
|   |— build.gradle  
|   |— include_flutter.groovy  
|   └─ ...  
|— .ios/  
|   |— Runner.xcworkspace  
|   └─ Flutter/podhelper.rb  
|— lib/  
|   └─ main.dart  
|— test/  
└─ pubspec.yaml
```

## **mobius\_flutter/**

```
|— .android/  
|   |— build.gradle  
|   |— include_flutter.groovy  
|   └─ ...  
|— .ios/  
|   |— Runner.xcworkspace  
|   └─ Flutter/podhelper.rb  
|— lib/  
|   └─ main.dart  
|— test/  
└─ pubspec.yaml
```



```
mobius_flutter/  
├── .android/  
│   ├── build.gradle  
│   ├── include_flutter.groovy  
│   └── ...  
├── .ios/  
│   ├── Runner.xcworkspace  
│   └── Flutter/podhelper.rb  
├── lib/  
│   └── main.dart  
├── test/  
└── pubspec.yaml
```

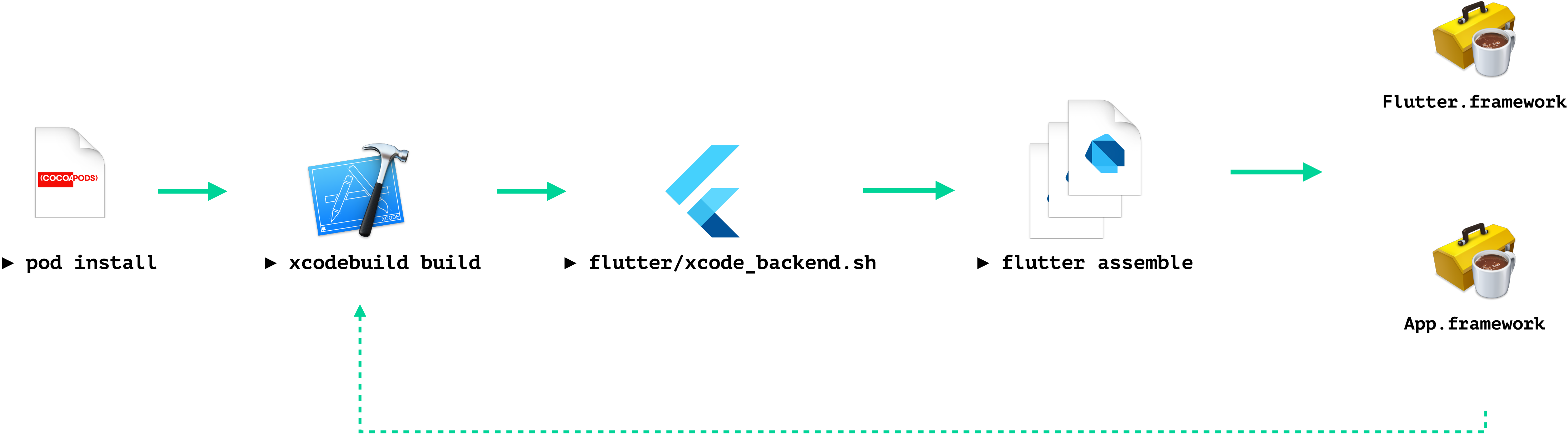
## **mobius\_flutter/**

```
|— .android/  
|   |— build.gradle  
|   |— include_flutter.groovy  
|   └─ ...  
|— .ios/  
|   |— Runner.xcworkspace  
|   └─ Flutter/podhelper.rb  
|— lib/  
|   └─ main.dart  
|— test/  
└─ pubspec.yaml
```

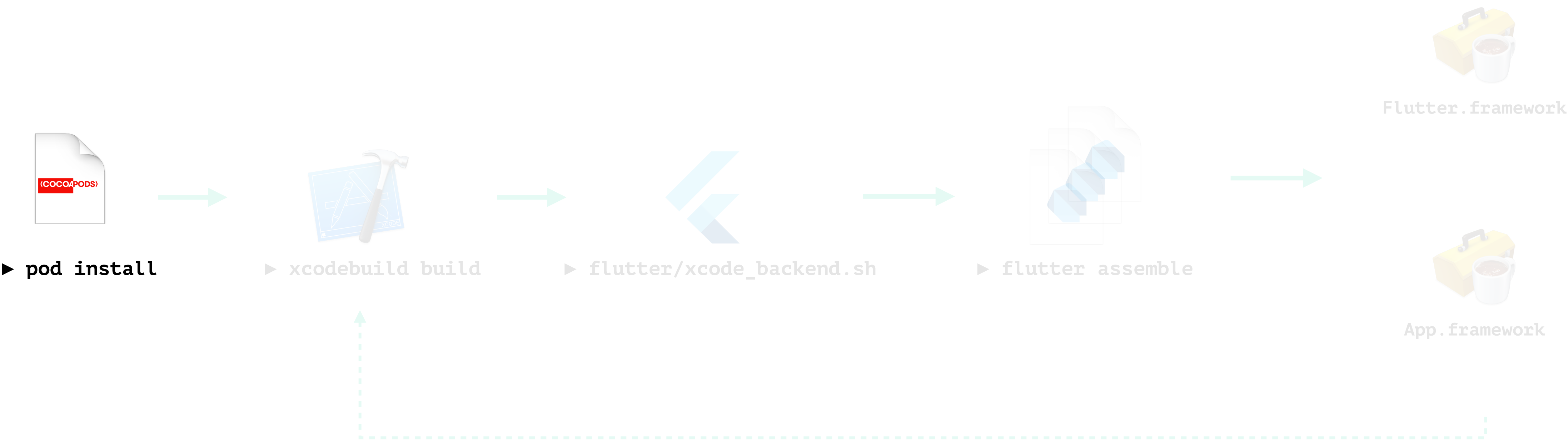
**mobius\_flutter/**

```
|— .android/  
|   |— build.gradle  
|   |— include_flutter.groovy  
|   └─ ...  
|— .ios/  
|   |— Runner.xcworkspace  
|   |— Flutter/podhelper.rb  
|— lib/  
|   └─ main.dart  
|— test/  
└─ pubspec.yaml
```

# Flutter build ios



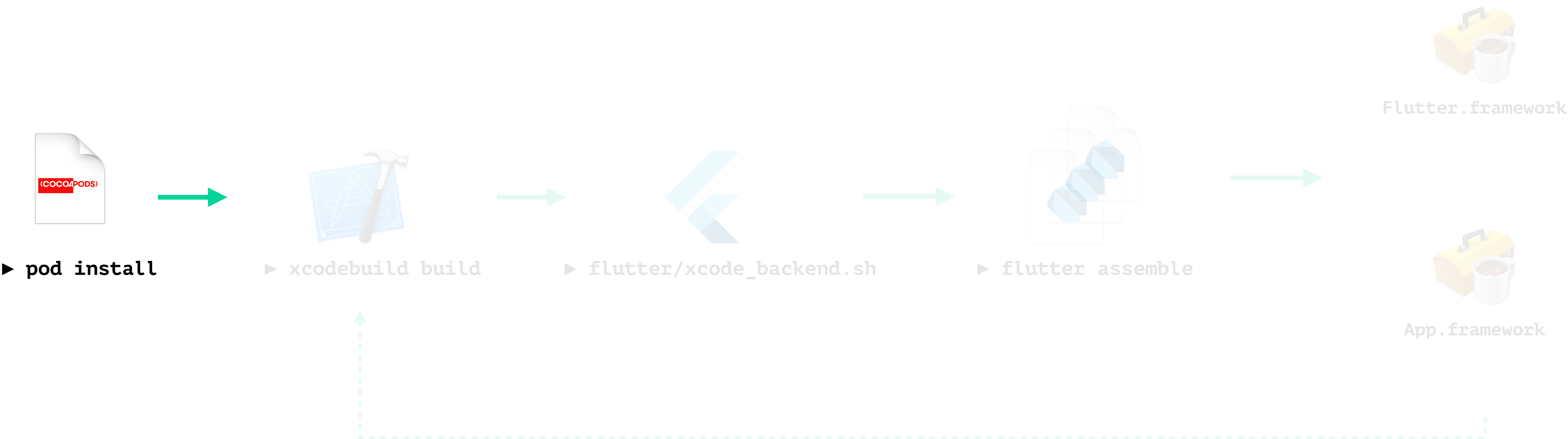
# Flutter build ios





```
1 #Podfile
2
3 load File('podhelper.rb')
4
5 target 'Mobius_App' do
6   install_flutter_engine_pod
7   install_flutter_application_pod(flutter_app_path)
8 end
```

# Flutter build ios





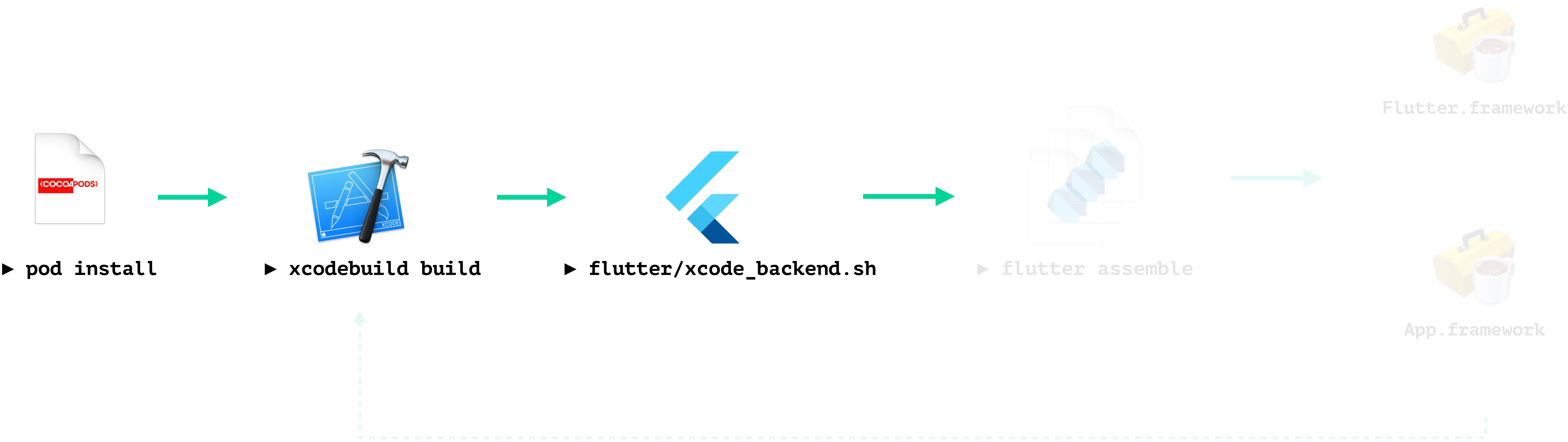
```
1 #podhelper.rb
2
3 def install_flutter_engine_pod
4   # Копируем Flutter engine из Flutter SDK
5   cp '$FLUTTER_ROOT/.../Flutter.framework' './engine'
6   cp '$FLUTTER_ROOT/.../Flutter.podspec' './engine'
7
8   pod 'Flutter', :path => './engine'
9 end
```





```
11 #podhelper.rb
12
13 def install_flutter_application_pod(path)
14   pod 'mobius_flutter', :path => './'
15
16   # Добавляем вызов xcode_backend.sh в билд фазы Xcode
17   script_phase
18     :name => 'Run Flutter Build Script',
19     :script => '$FLUTTER_ROOT/.../xcode_backend.sh build',
20 end
```

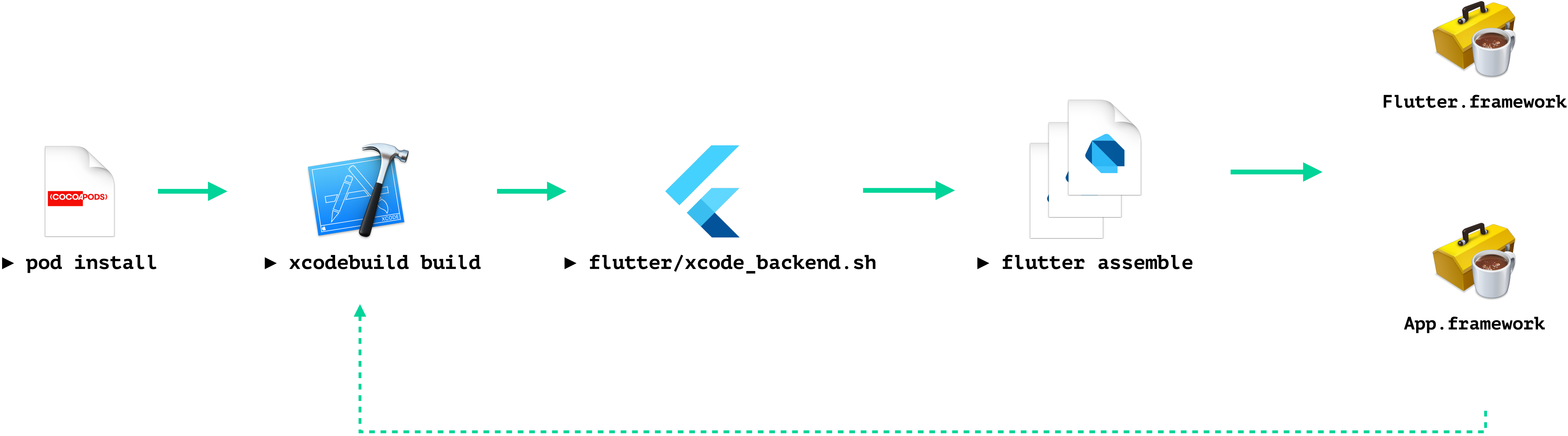
# Flutter build ios



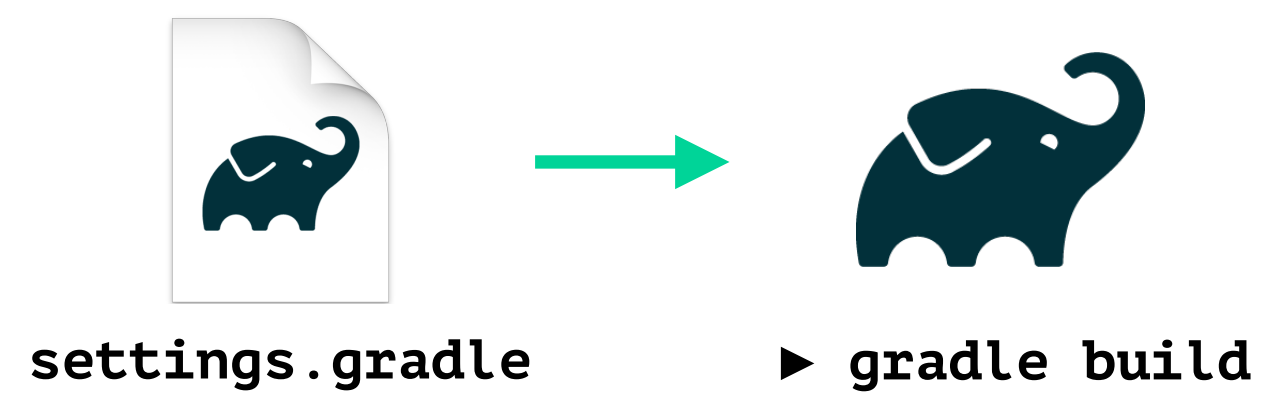


```
11 #xcode_backend.sh
12
13 build() {
14     ...
15     "${FLUTTER_ROOT}/bin/flutter" \
16     assemble \
17     --output="${derived_dir}/" \
18     -dTargetPlatform=ios \
19     -dTargetFile="lib/main.dart" \
20     ...
21 }
```

# Flutter build ios



# Flutter build android

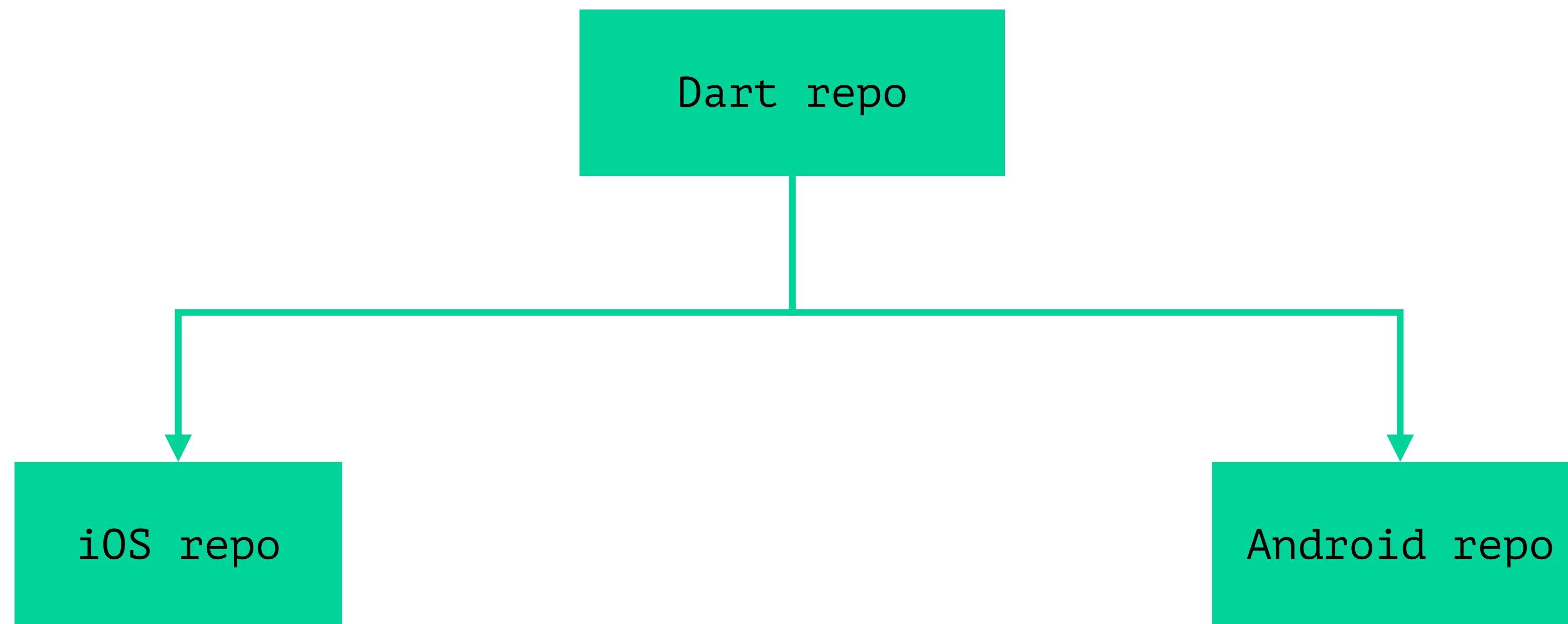


# Dart репозиторий



Dart repo

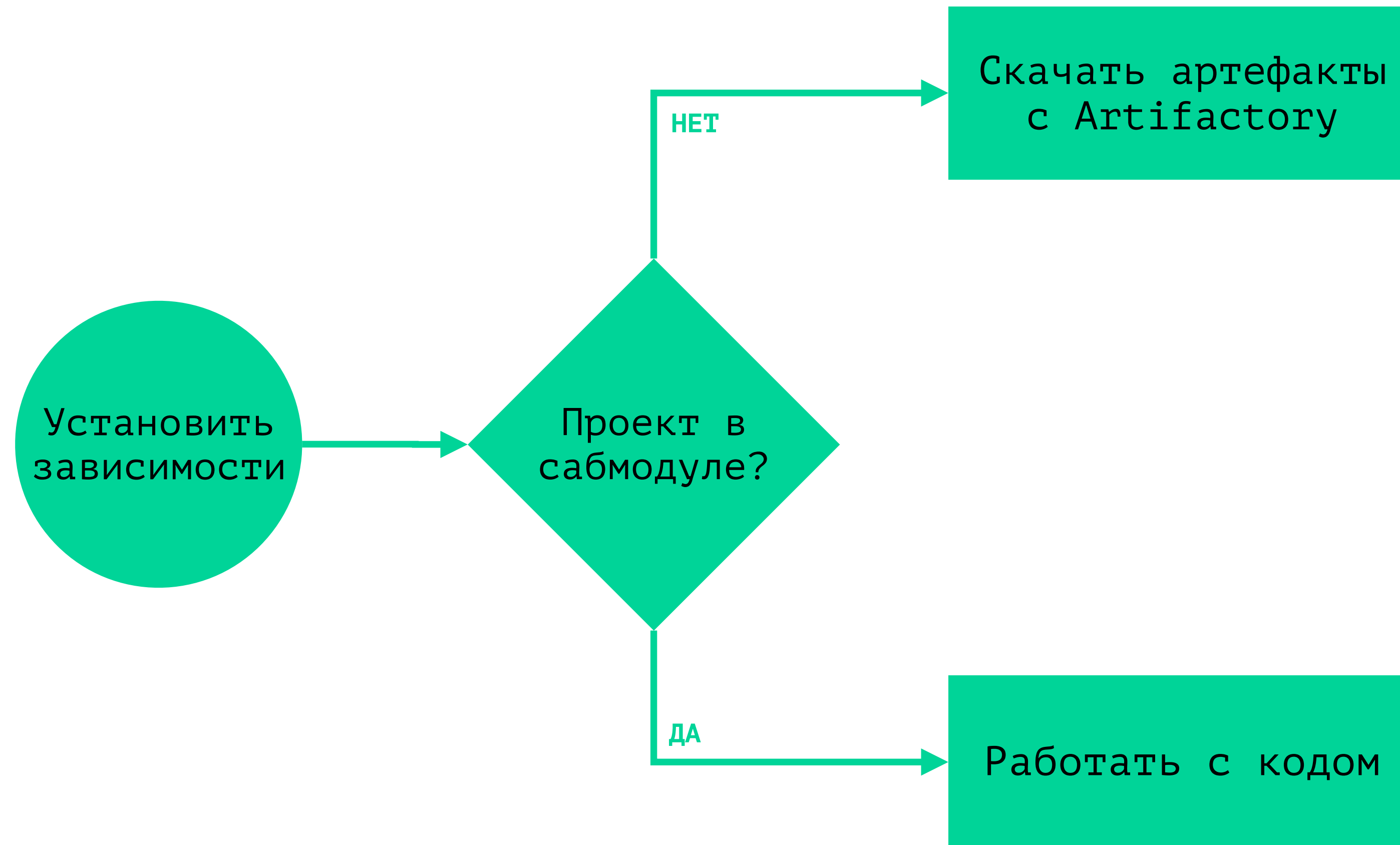
# Dart репозиторий



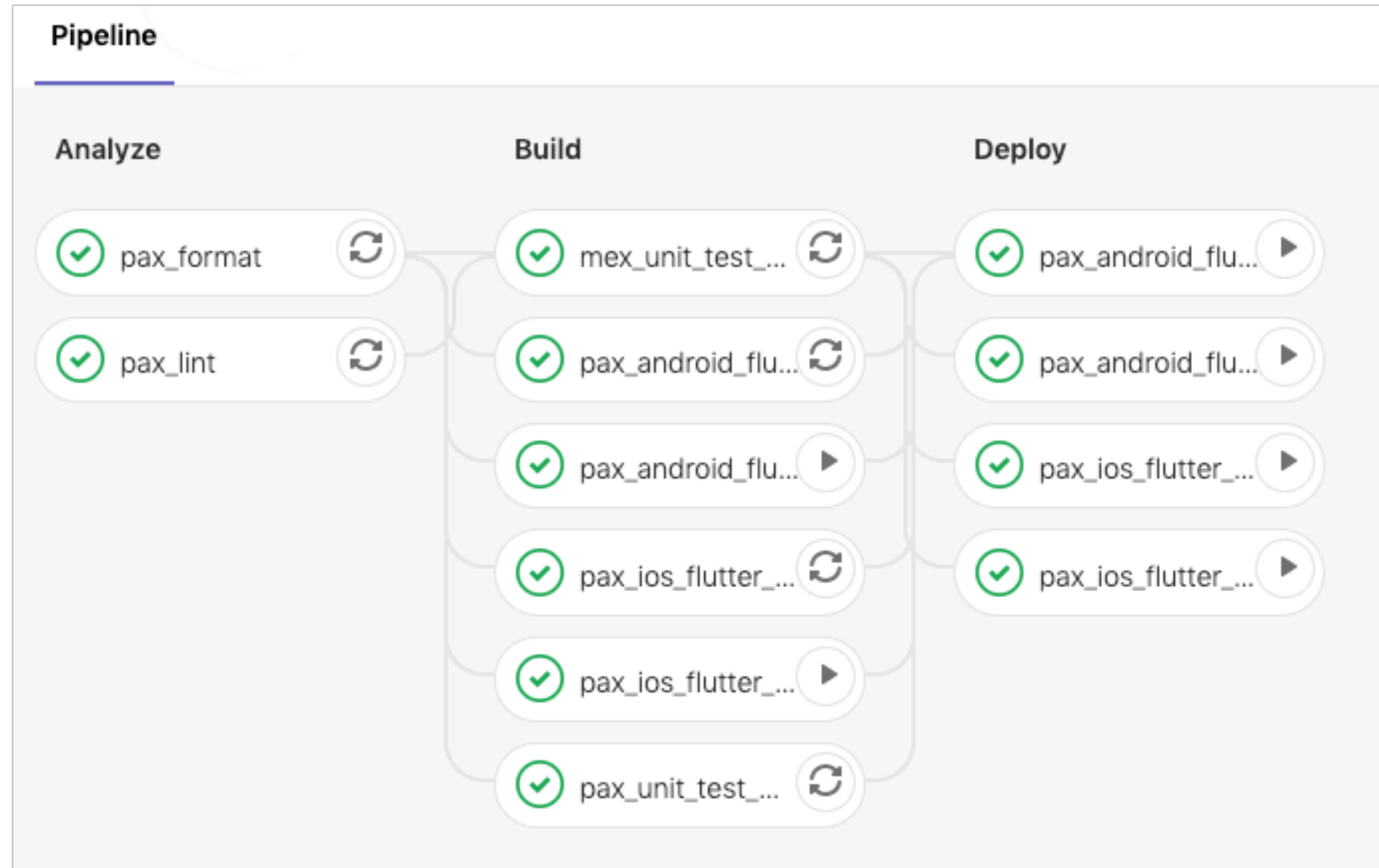
Нативные проекты - git сабмодули

**Всем мобильным разработчикам нужно  
устанавливать Flutter?**

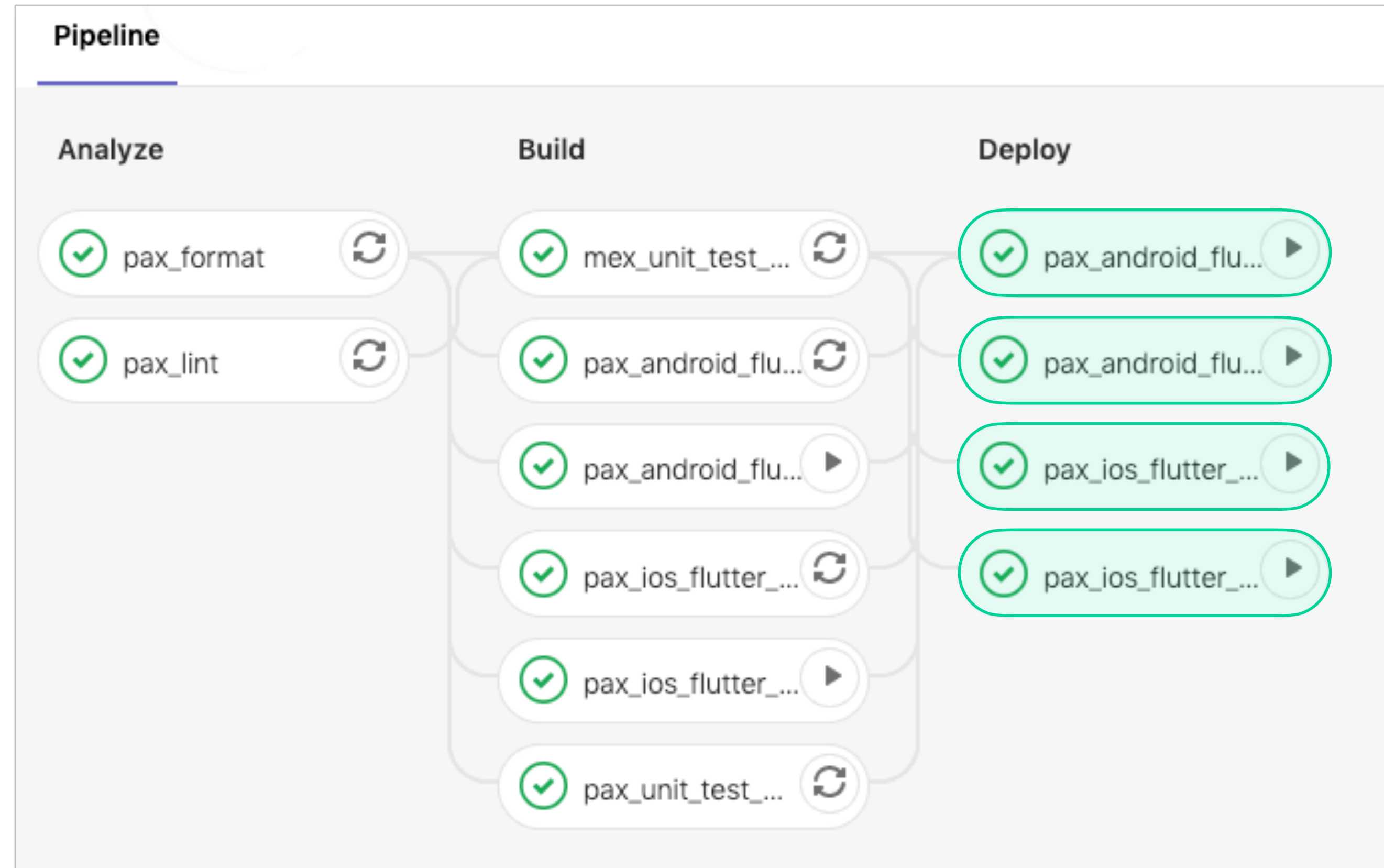




# CI

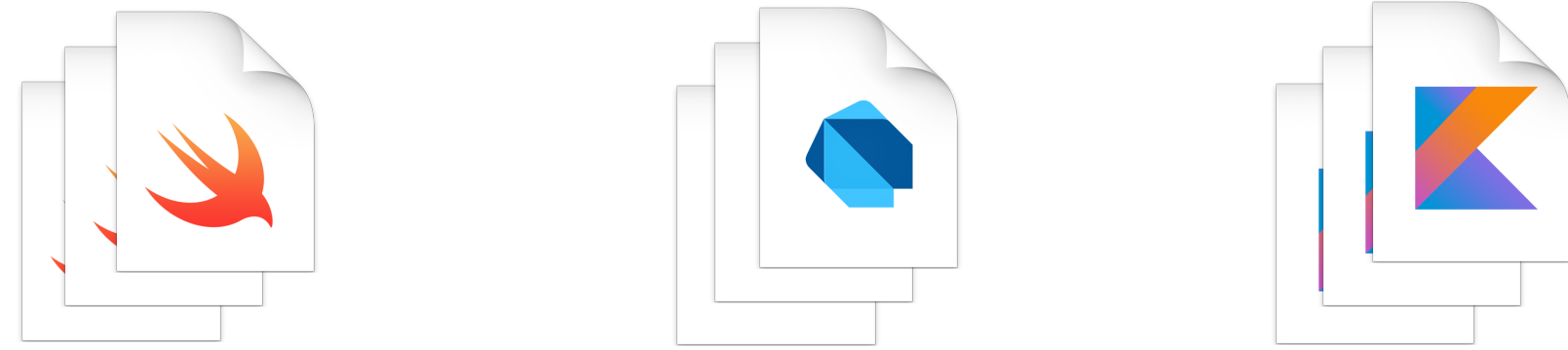


# CI

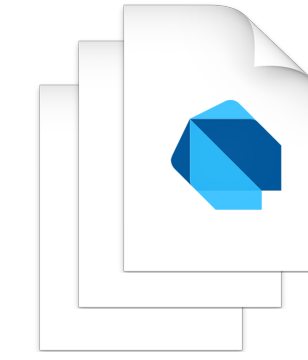


**Как переиспользовать нативный код?**

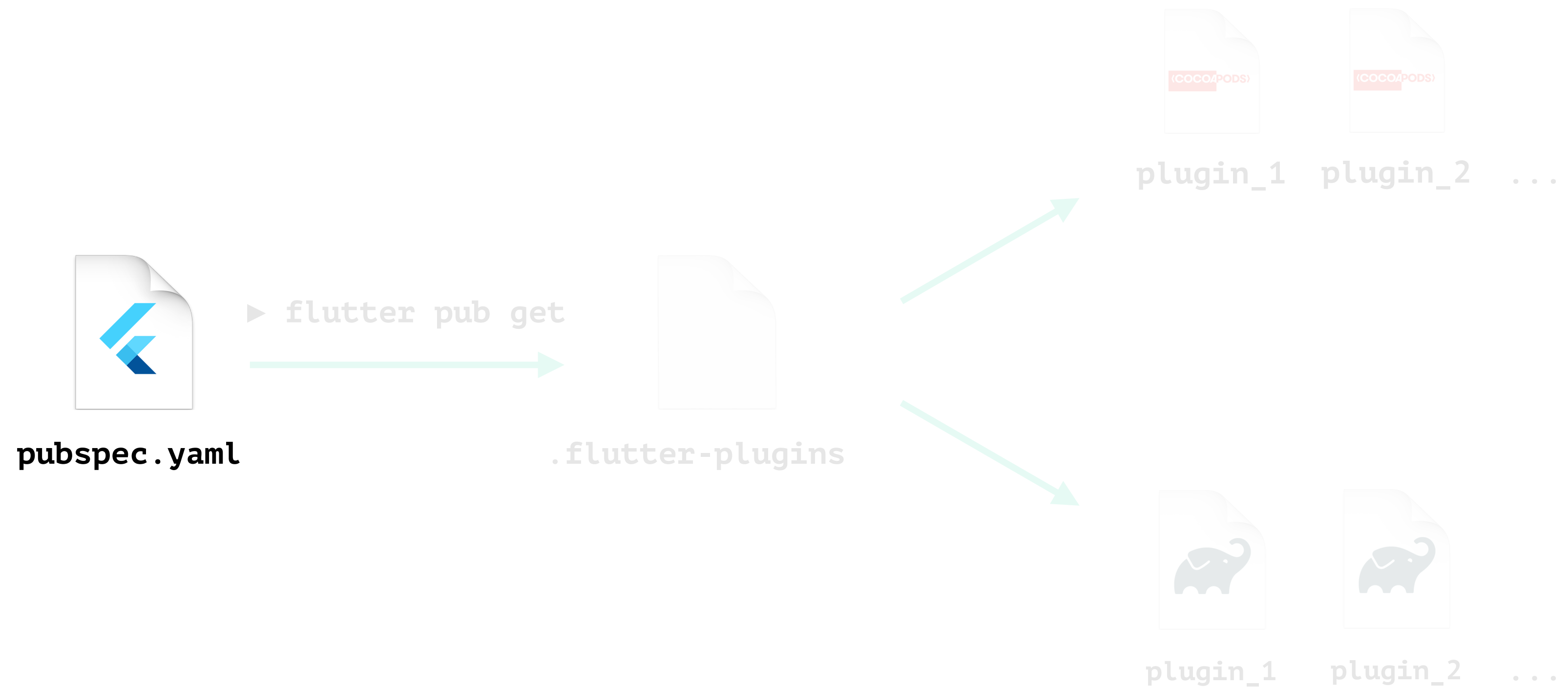
# Flutter plugin



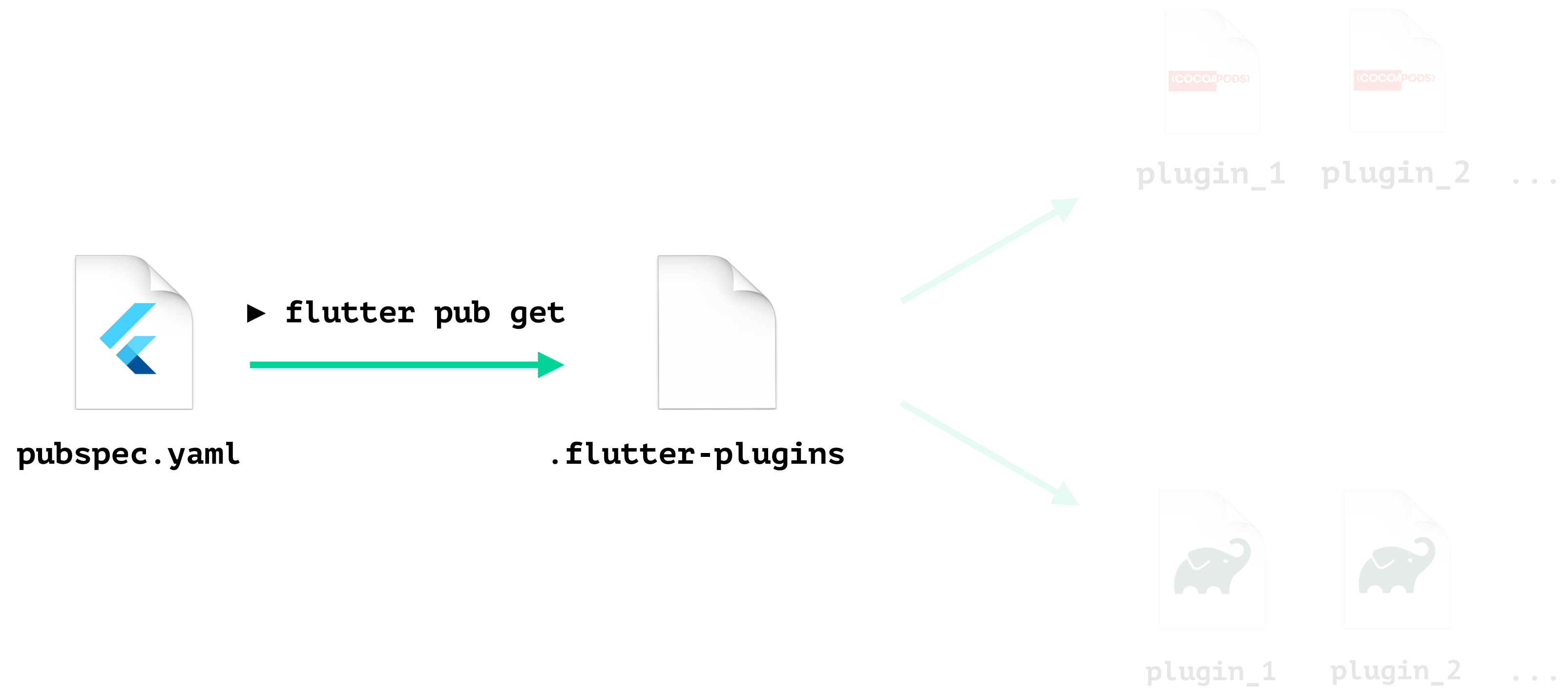
# Dart package



# Flutter Plugins



# Flutter Plugins

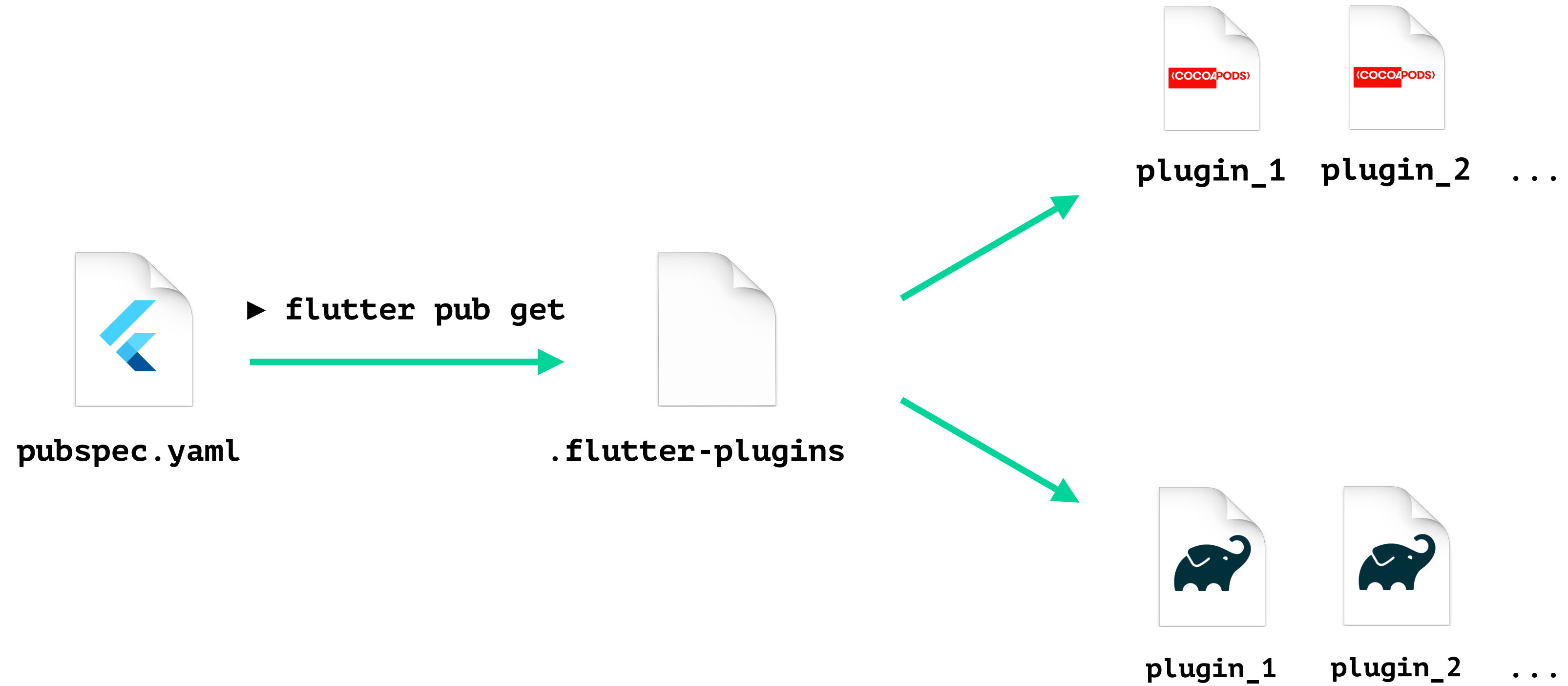


# .flutter-plugins

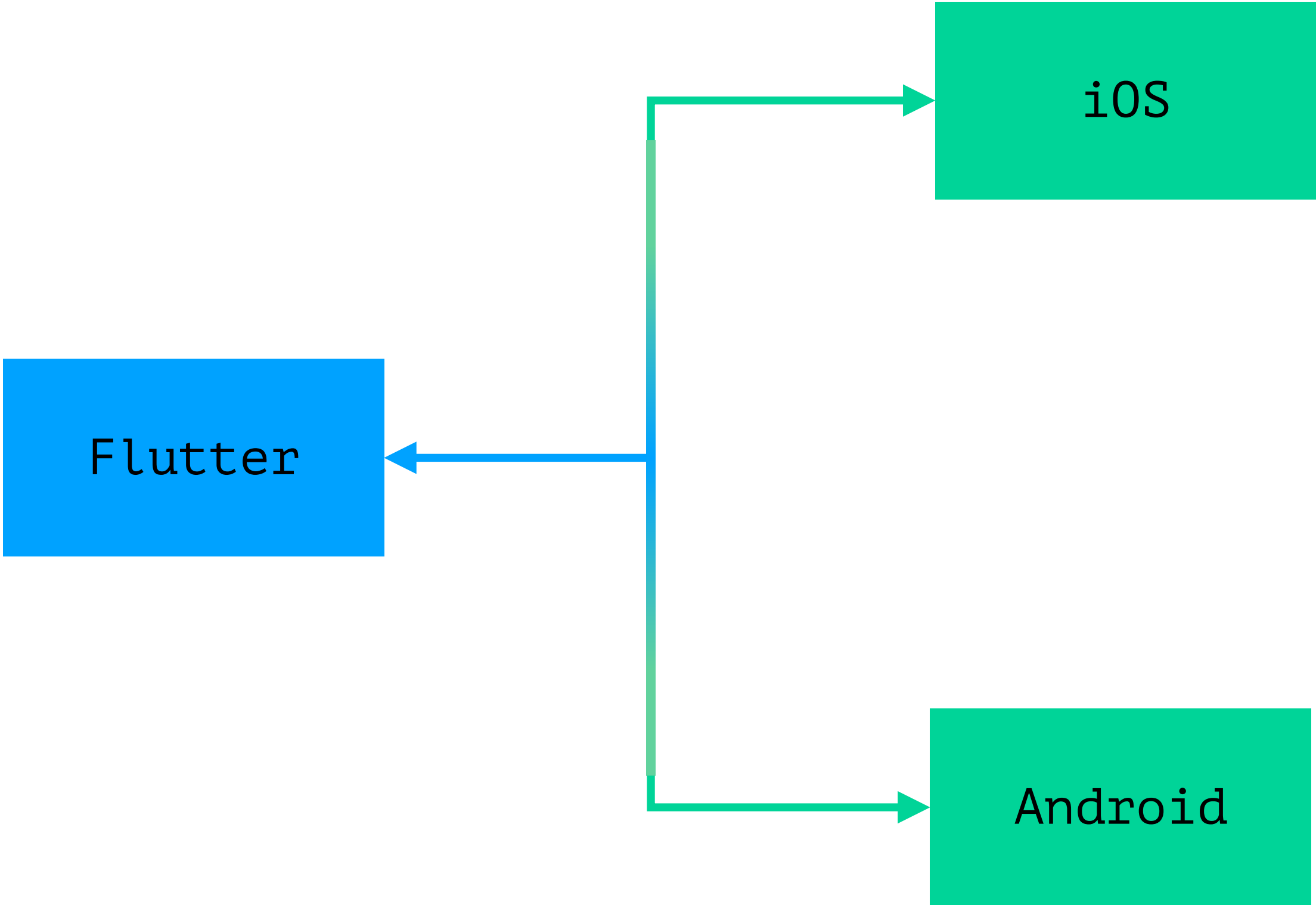
```
1 # This is a generated file; do not edit or check into version control.  
2  
3 plugin_1=/Users/mikhail.zinov/.pub-cache/hosted/pub.dartlang.org/plugin_1/  
4 plugin_2=/Users/mikhail.zinov/.pub-cache/hosted/pub.dartlang.org/plugin_2.1/  
5 plugin_3=/Users/mikhail.zinov/.pub-cache/hosted/pub.dartlang.org/plugin_3.3/
```



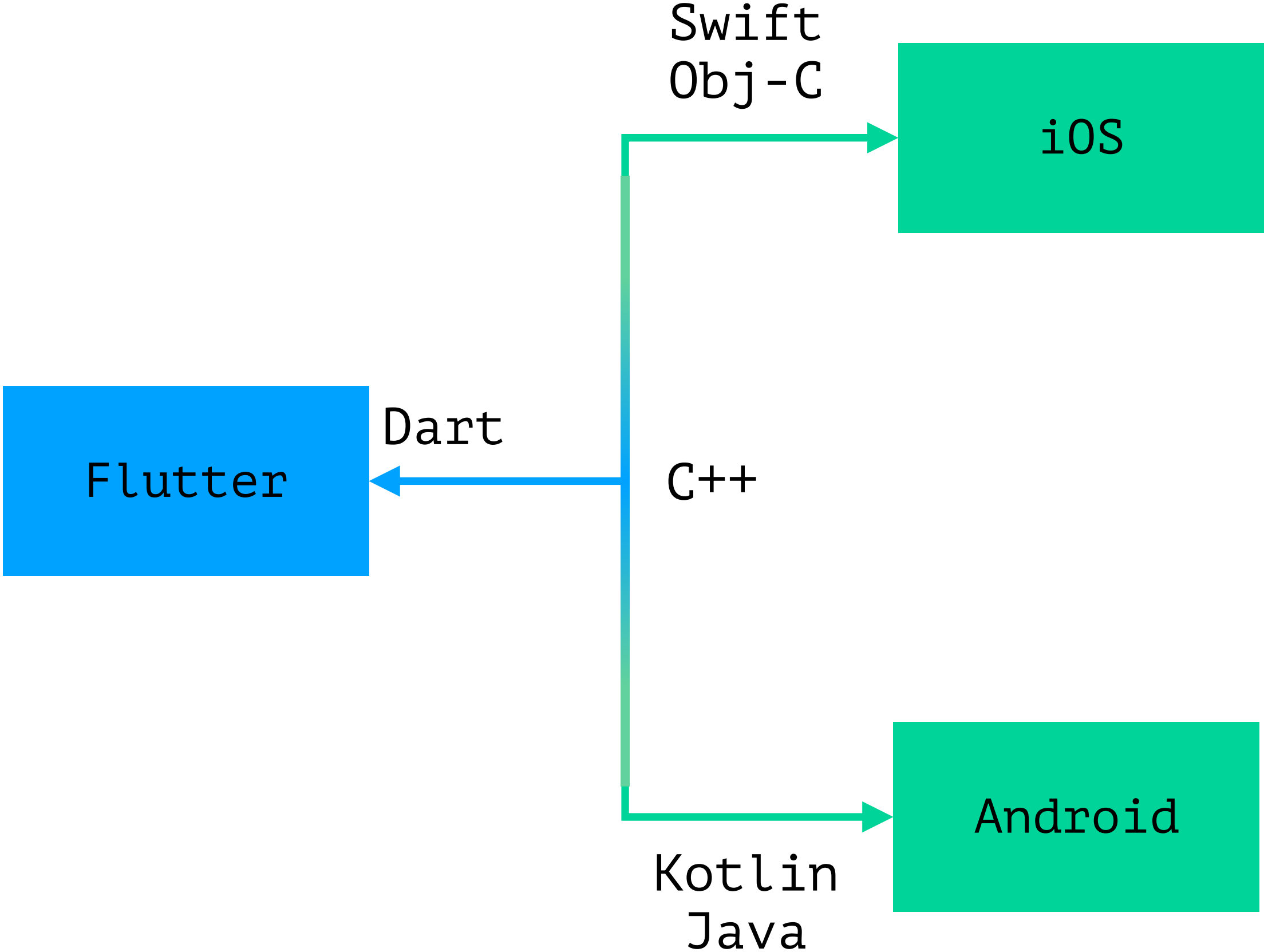
# Flutter Plugins



# Flutter + Native



# Flutter + Native



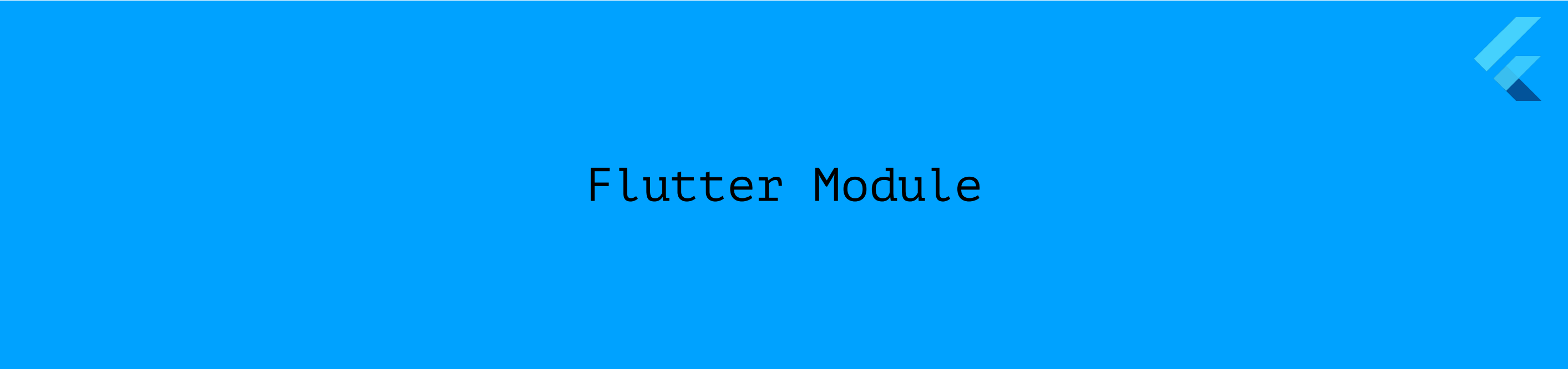
# Flutter Method Channel - Native

```
1 // engine/FlutterChannels.mm
2
3 @implementation FlutterMethodChannel {
4     NSObject<FlutterBinaryMessenger>* _messenger;
5     NSString* _name;
6     NSObject<FlutterMethodCodec>* _codec;
7
8     - (void)sendMessage:(id)message {...}
9     - (void)setMessageHandler:(FlutterMessageHandler)handler {...}
10 }
```

# Flutter Method Channel - Dart

```
1 // flutter/platform_channel.dart
2 class MethodChannel {
3
4   final String name;
5   final MethodCodec codec;
6   final BinaryMessenger _binaryMessenger;
7
8   Future<T> _invokeMethod<T>(String method, { dynamic arguments }) async { ... }
9   void setMethodCallHandler(Future<dynamic> handler(MethodCall call)) { ... }
10 }
```

**Flutter**



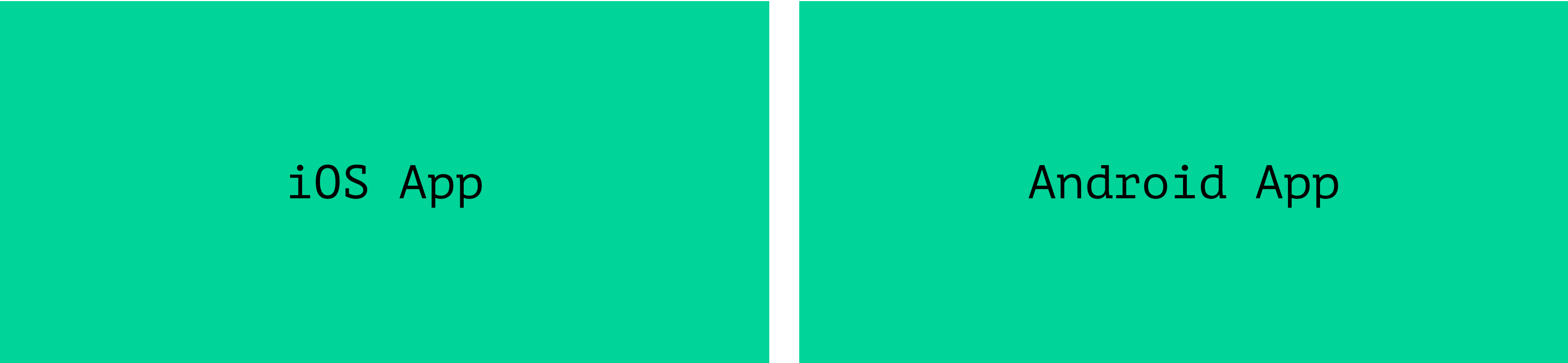
**Services Kits**



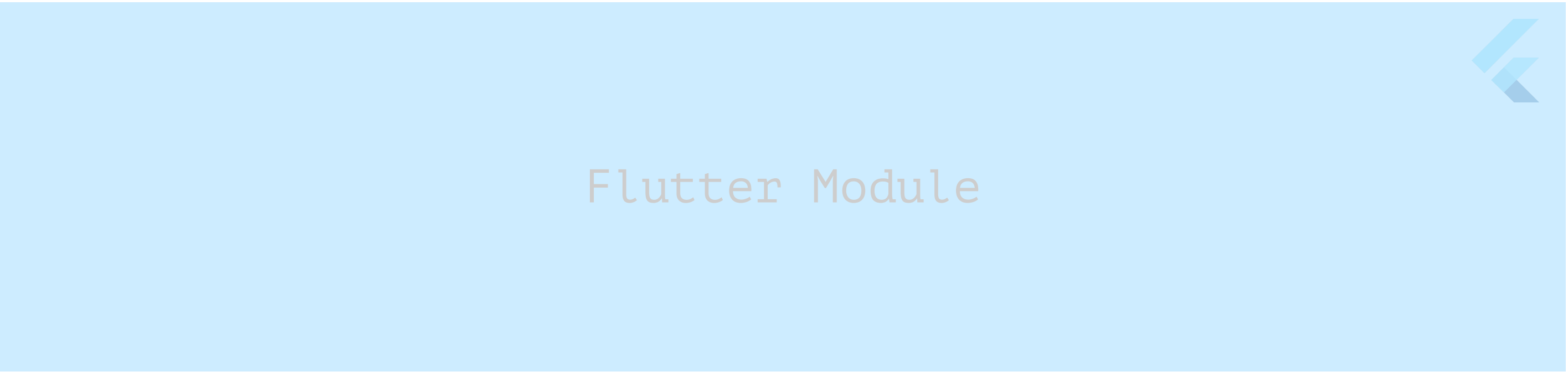
**Foundation Kits**



**Native Apps**



**Flutter**



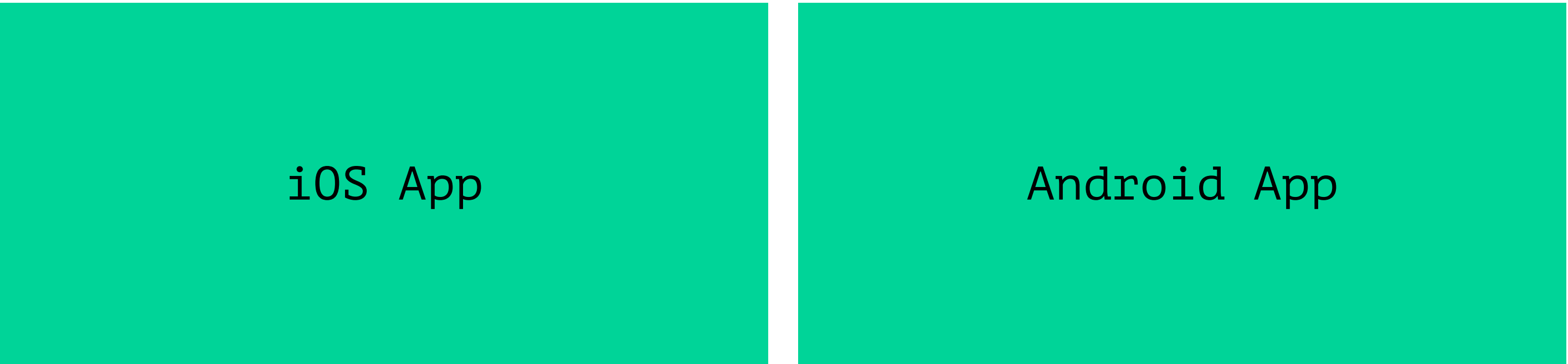
**Services Kits**



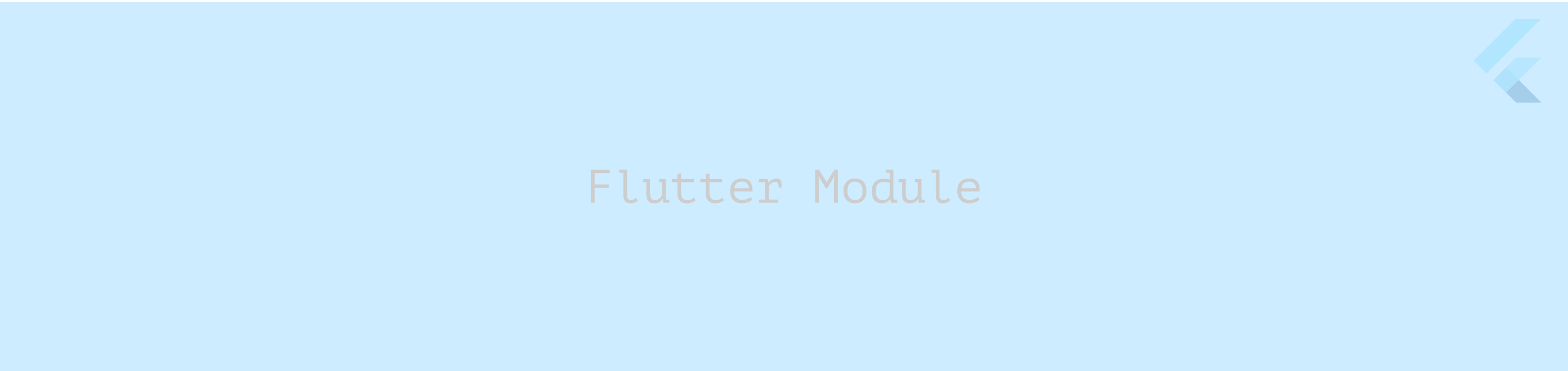
**Foundation Kits**



**Native Apps**



**Flutter**



Flutter Module



**Services Kits**



Payments

Rewards

Enterprise

...

**Foundation Kits**



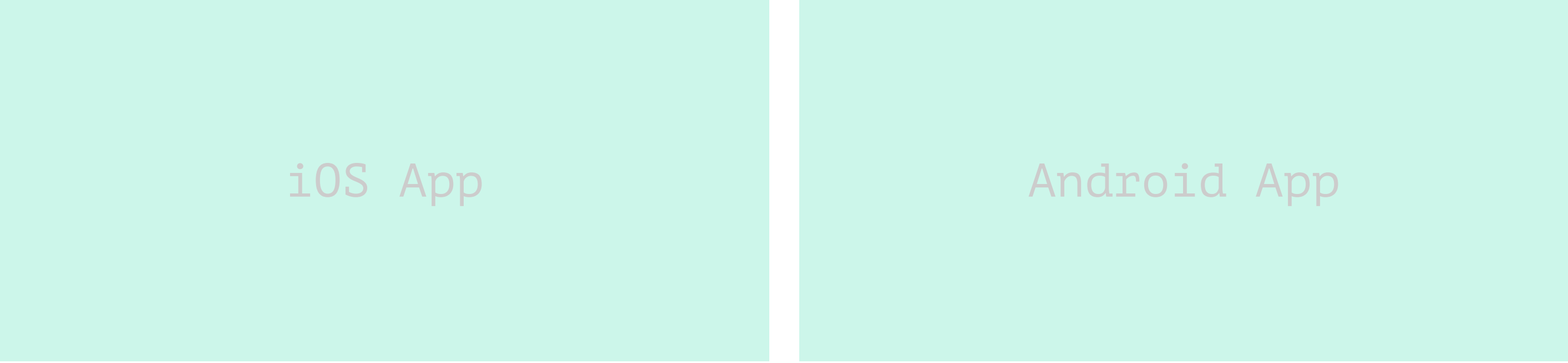
Networking

Storage

Experiments

...

**Native Apps**

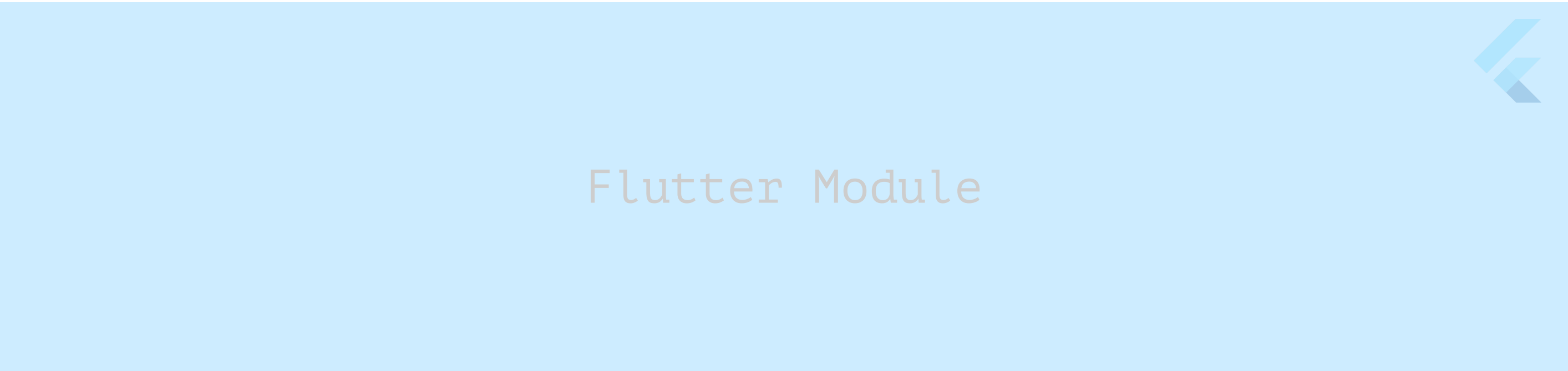


iOS App

Android App



**Flutter**



Flutter Module



**Services Kits**



Payments

Rewards

Enterprise

...

**Foundation Kits**



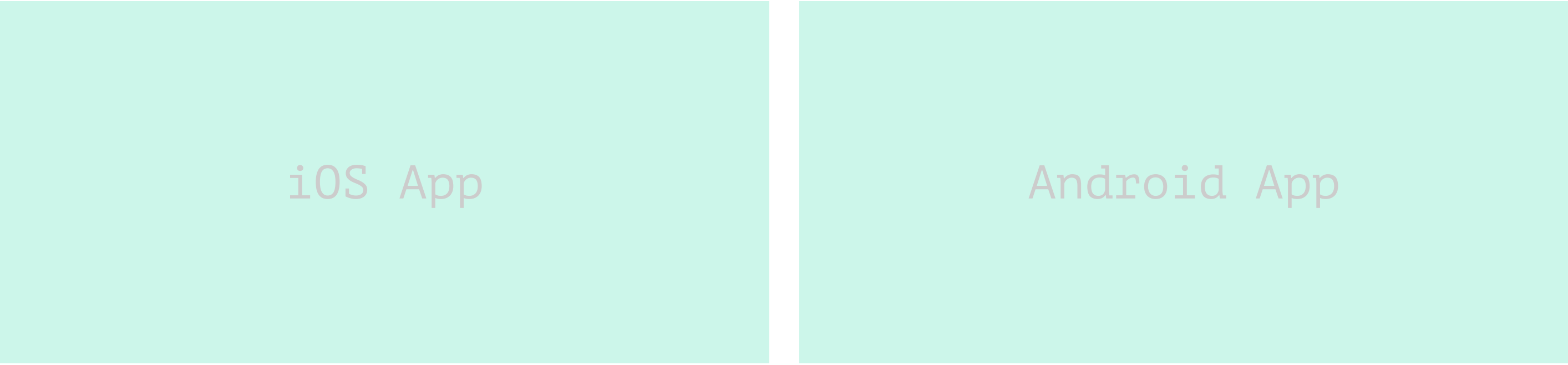
Networking

Storage

Experiments

...

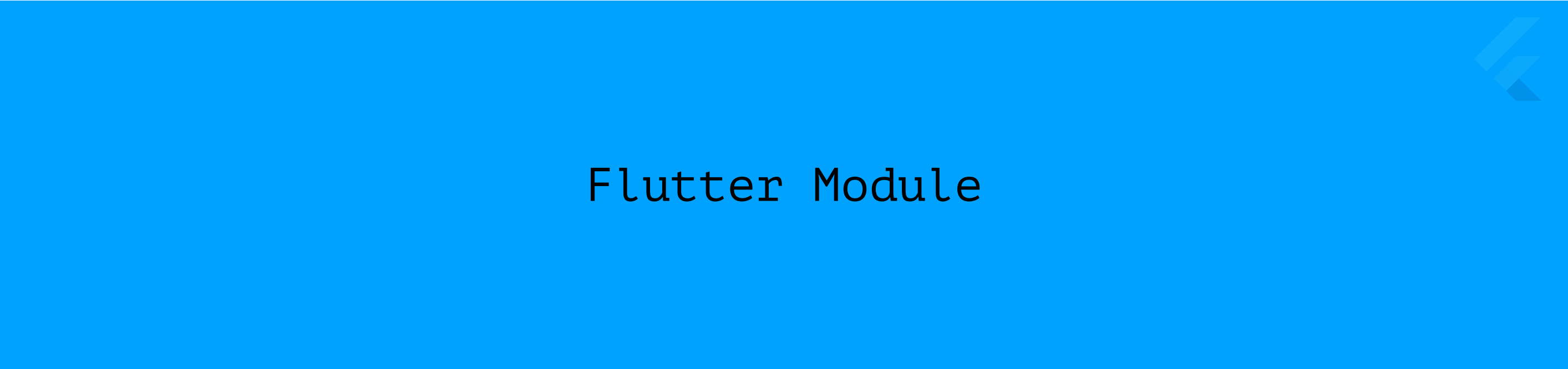
**Native Apps**



iOS App

Android App

**Flutter**



Flutter Module

**Services Kits**



Payments

Rewards

Enterprise

...

**Foundation Kits**



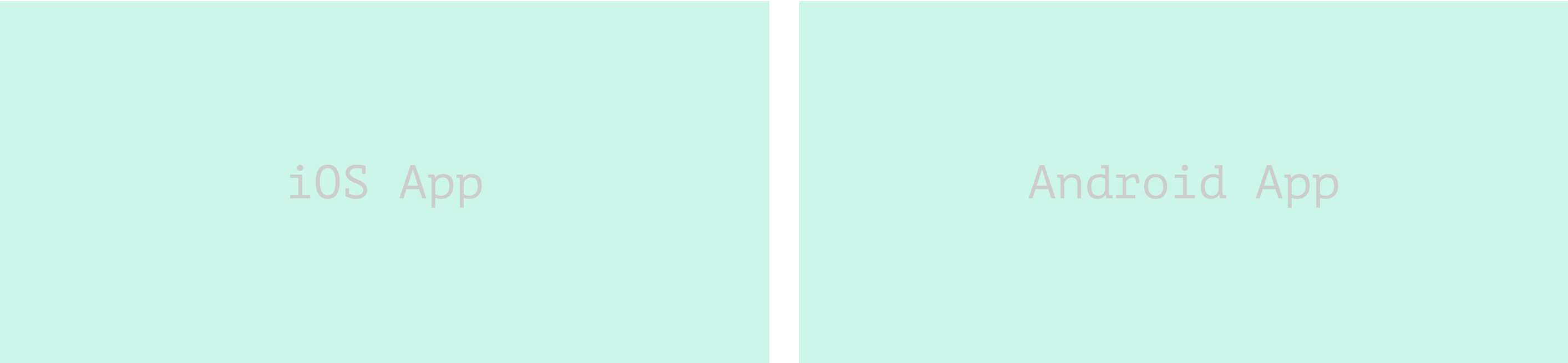
Networking

Storage

Experiments

...

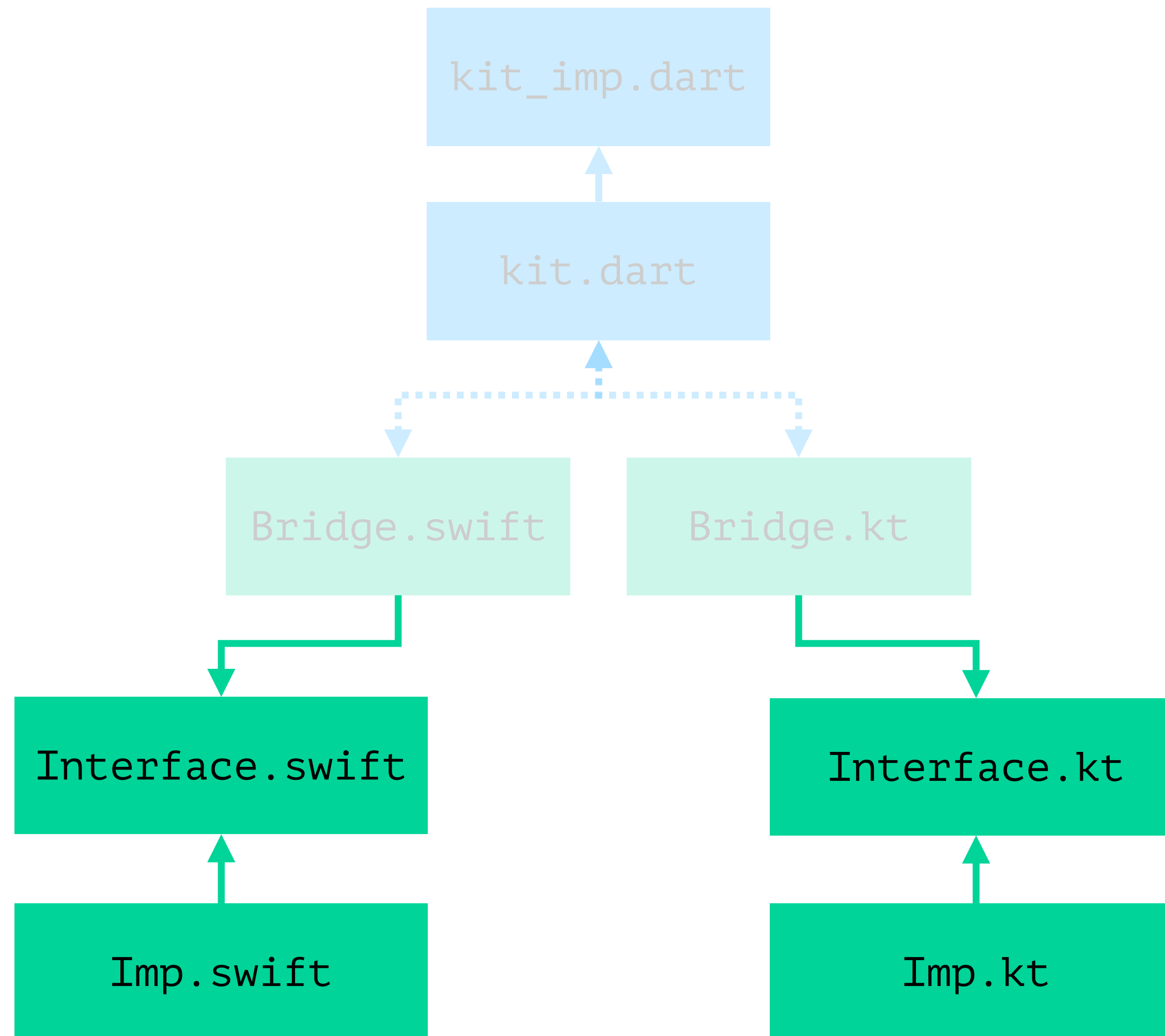
**Native Apps**



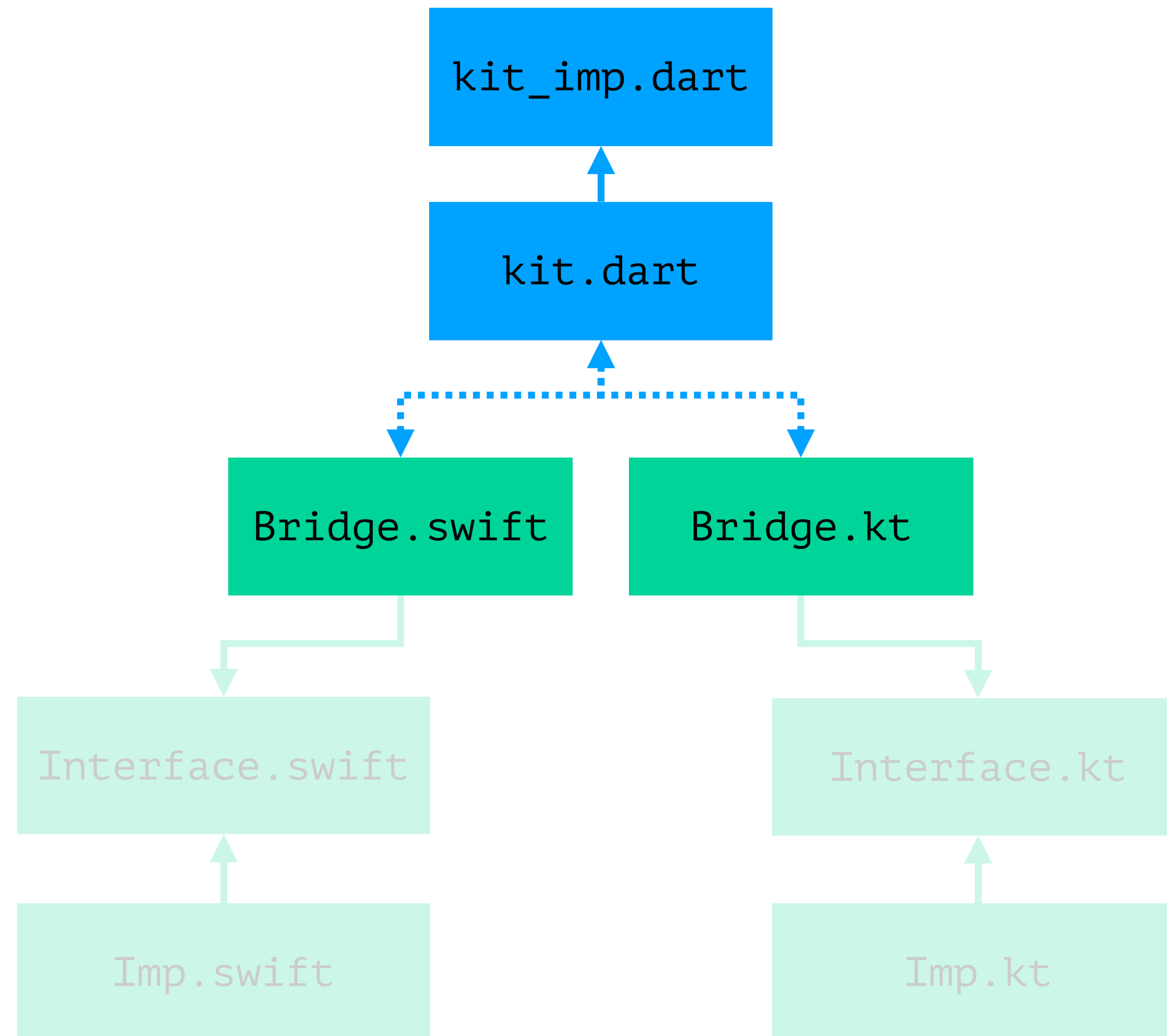
iOS App

Android App

# Kit

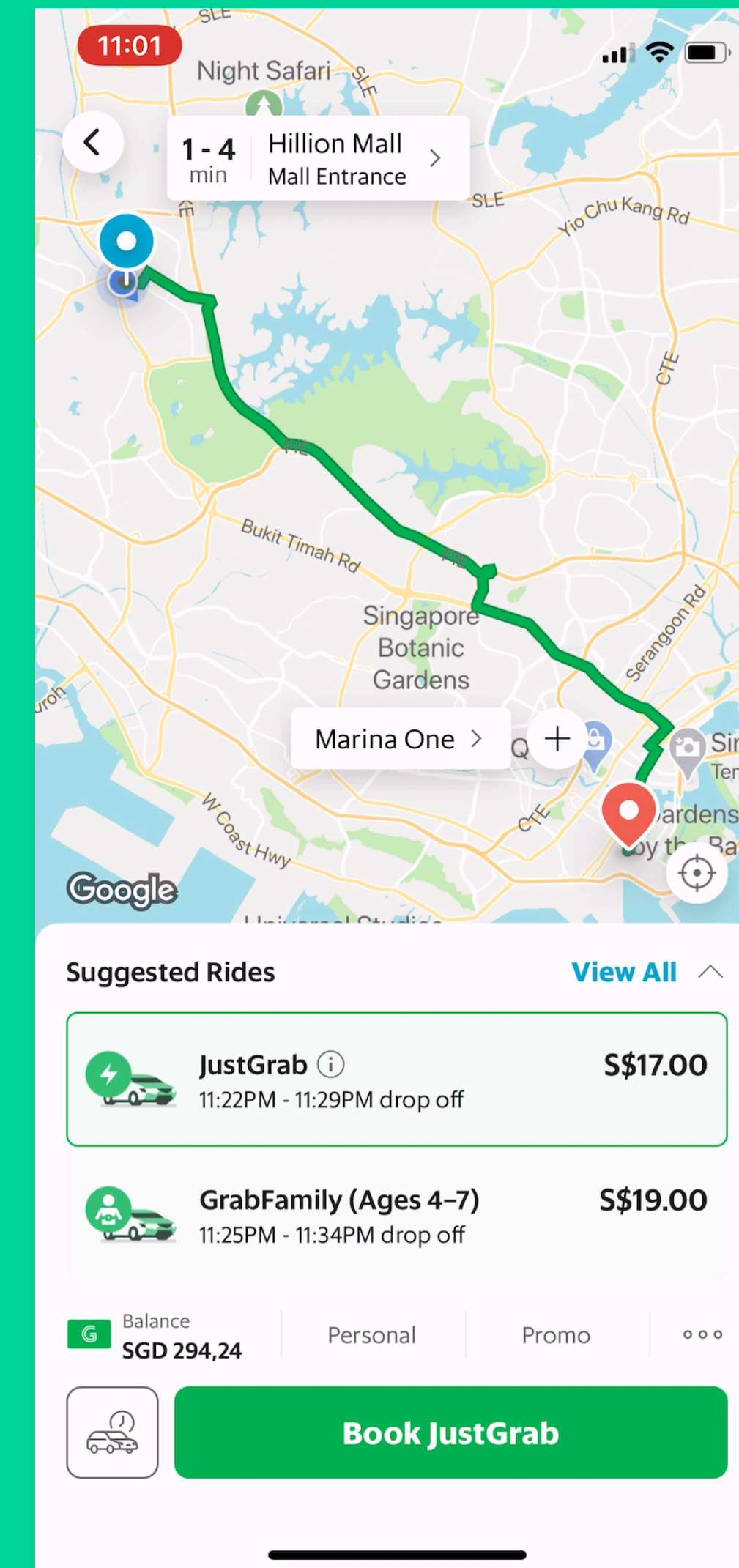


# Kit



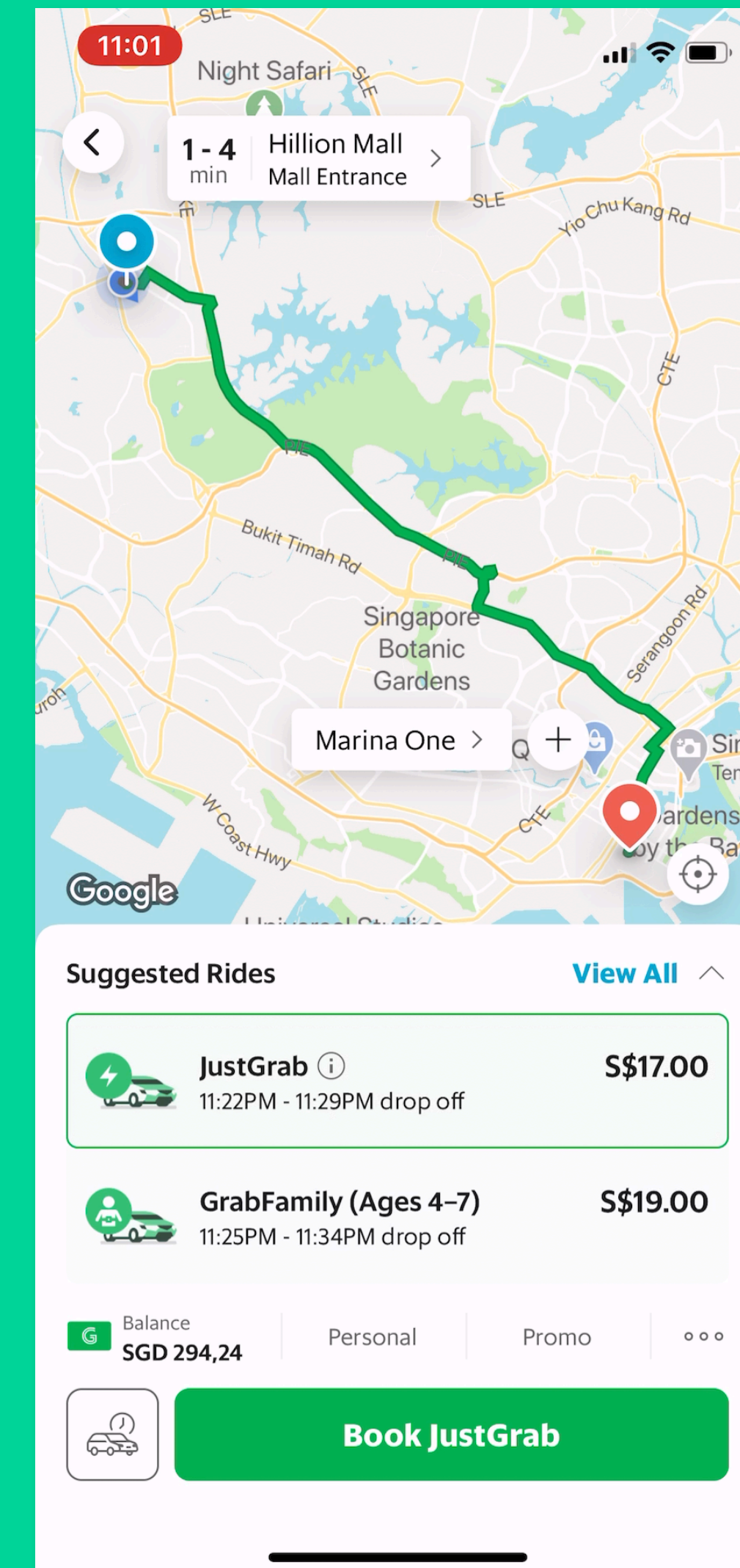
# Payments Kit

Переиспользуемые нативный функционал

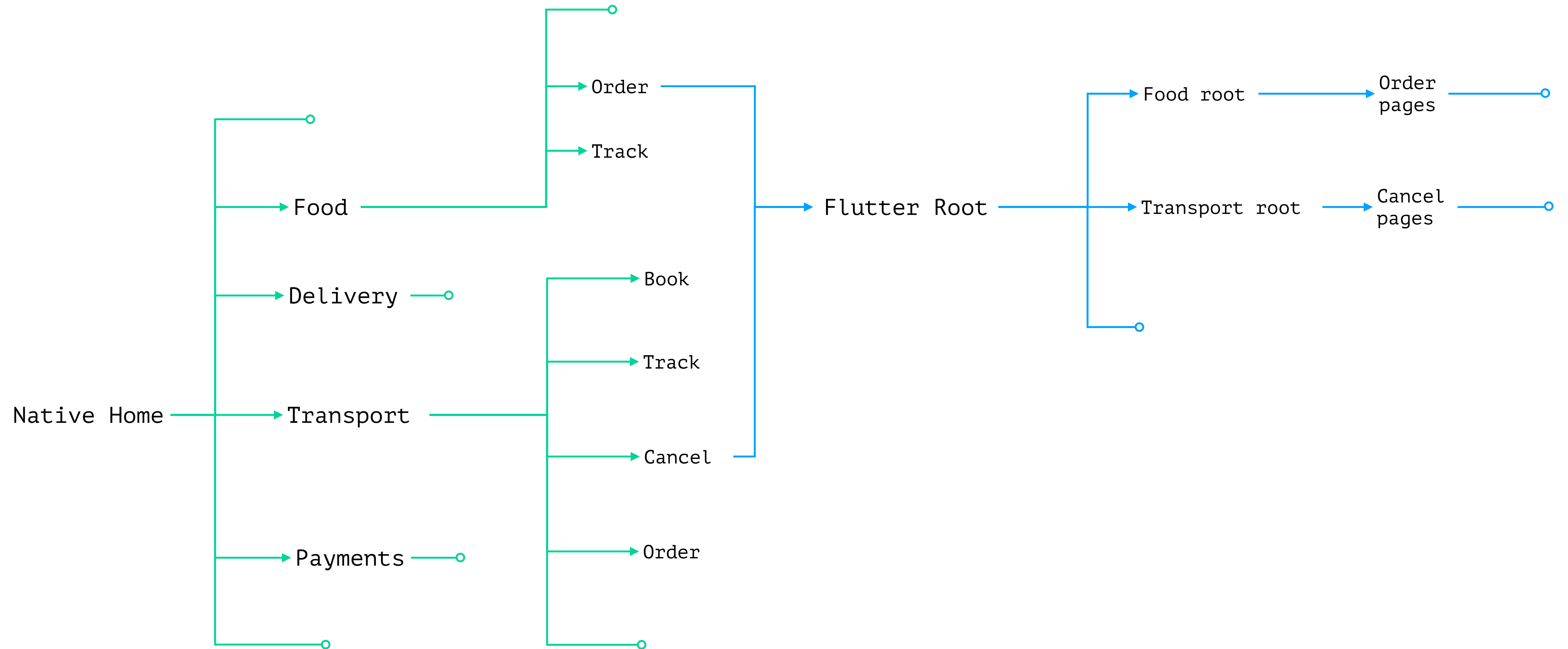


# Payments Kit

Переиспользуемые нативный функционал



**А что с навигацией?**





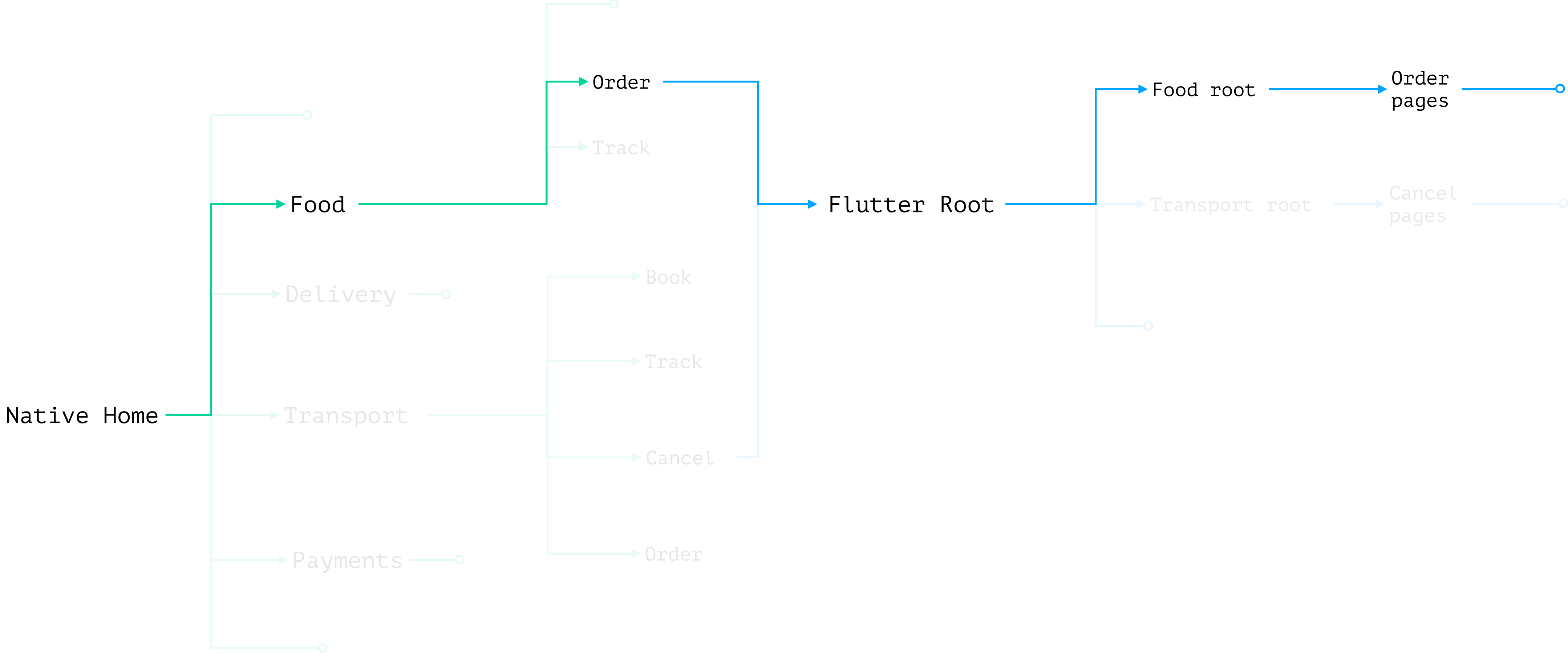
**grab://open?screenType=<screen type>&parameter1=<value 1>&...**

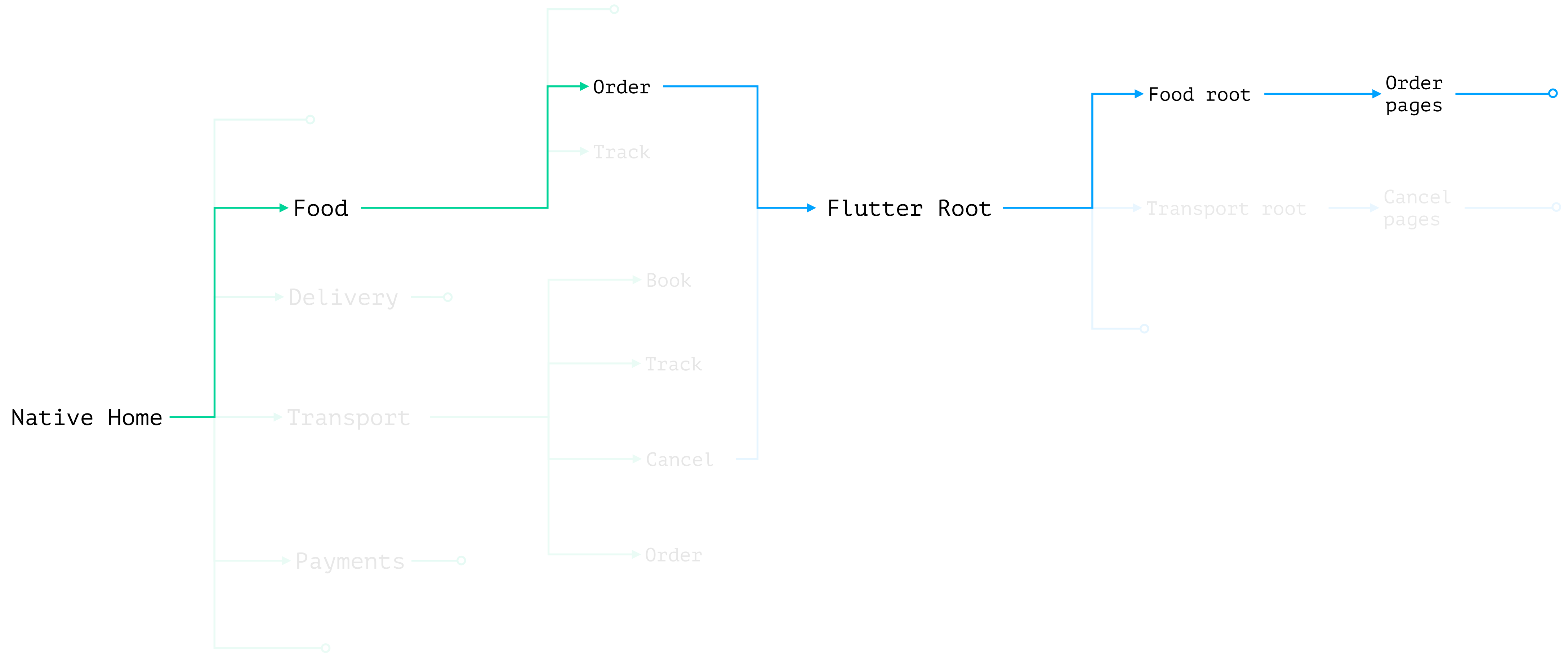
**grab://open?**screenType=<screen type>&parameter1=<value 1>&...

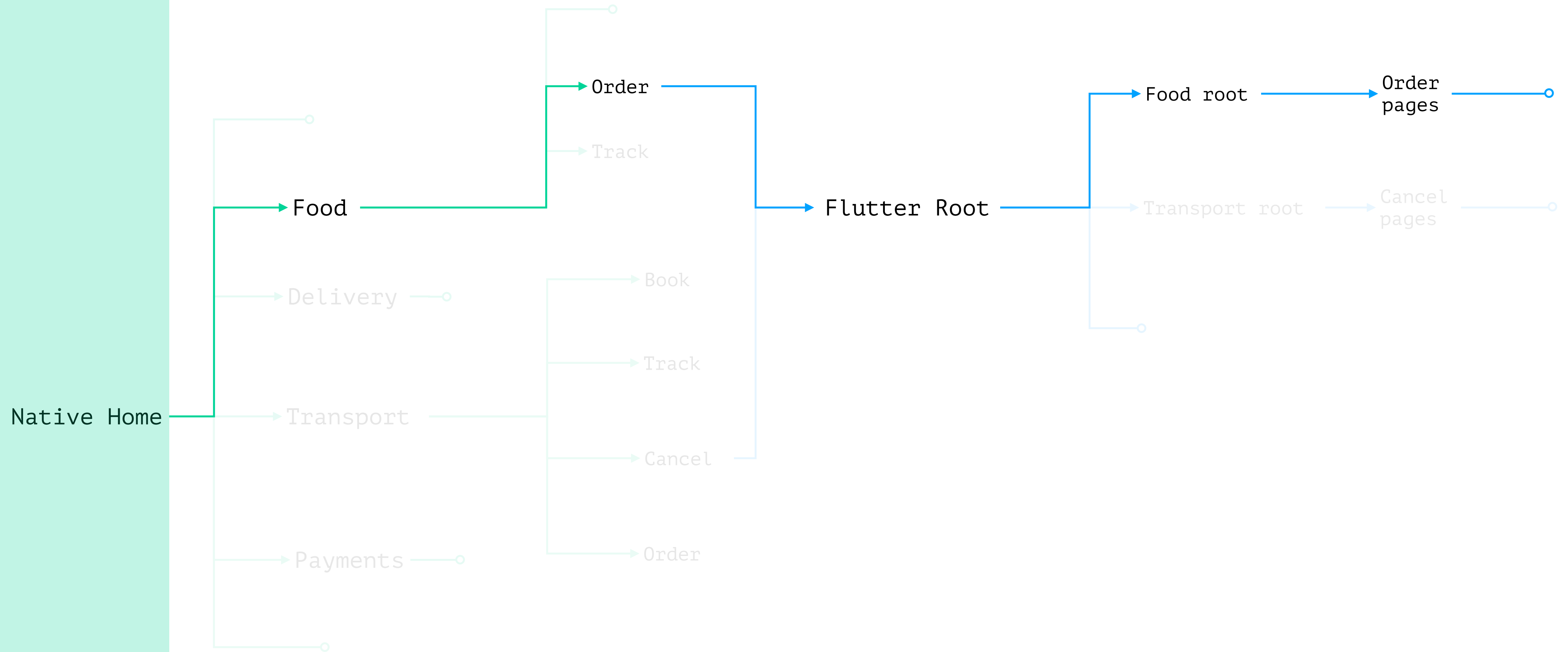
`grab://open?screenType=<screen type>&parameter1=<value 1>&...`

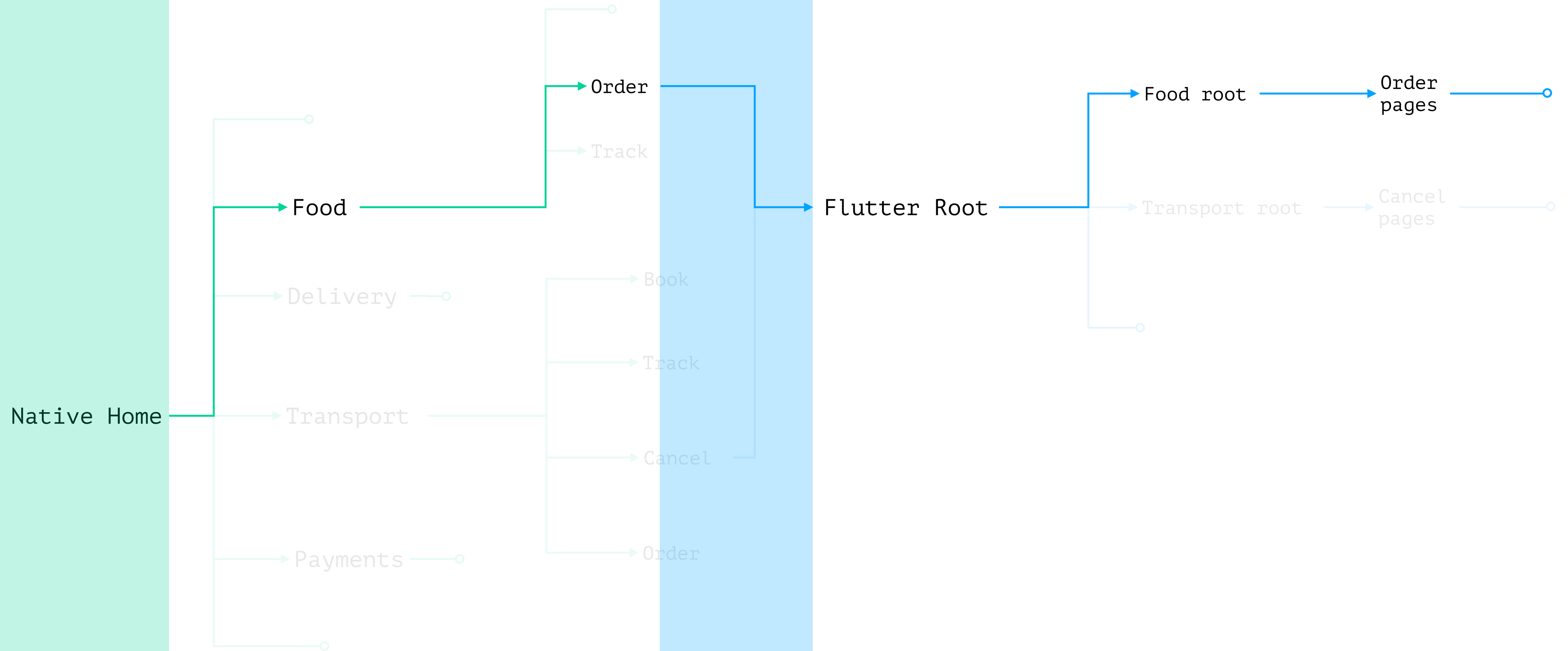
grab://open?screenType=<screen type>&parameter1=<value 1>&...

# Навигация через DeepLink-и



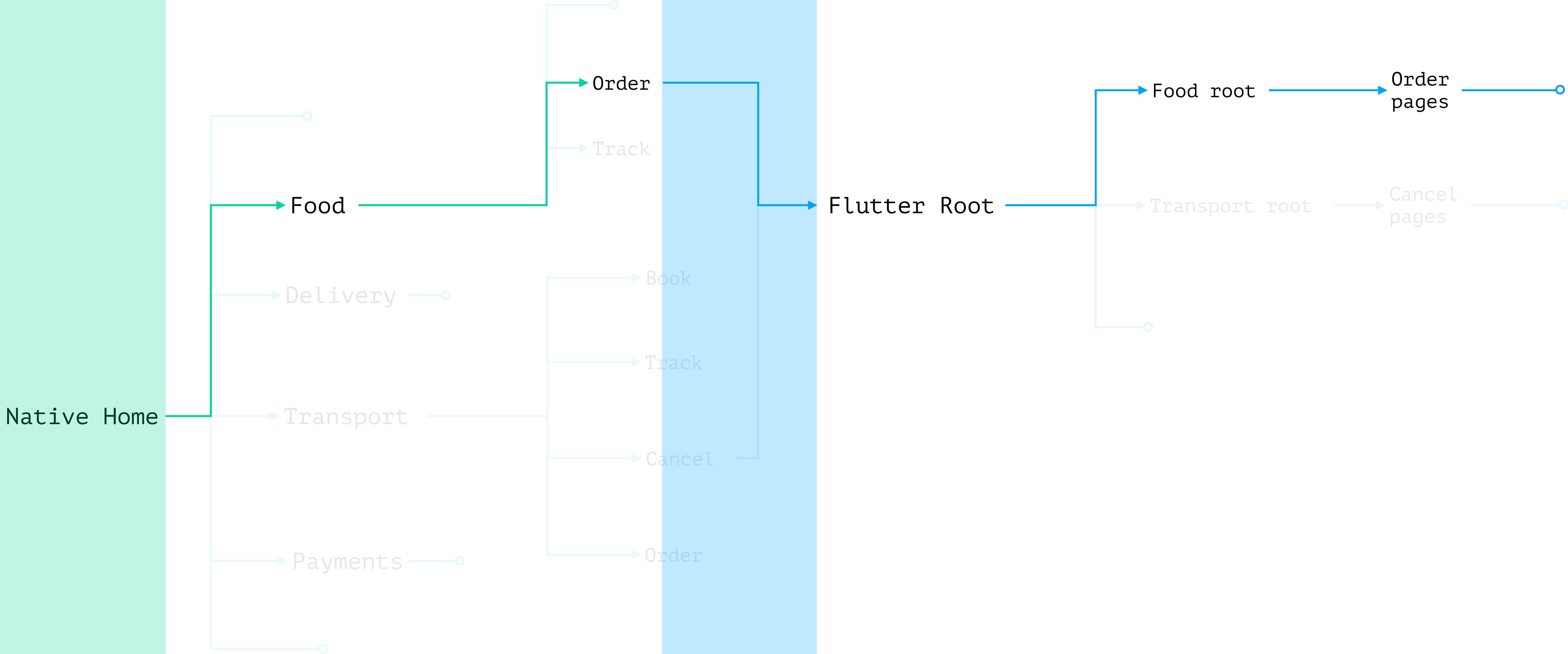








# Запуск Flutter Engine



# Запуск Flutter Engine

	<b>Без прогрева</b> миллисекунды	<b>С прогревом</b> миллисекунды
<b>iOS</b>	—	—
<b>Android</b>	600 ~ 1100	400 ~ 900

# Библиотека UI компонентов

Общий код для iOS и Android

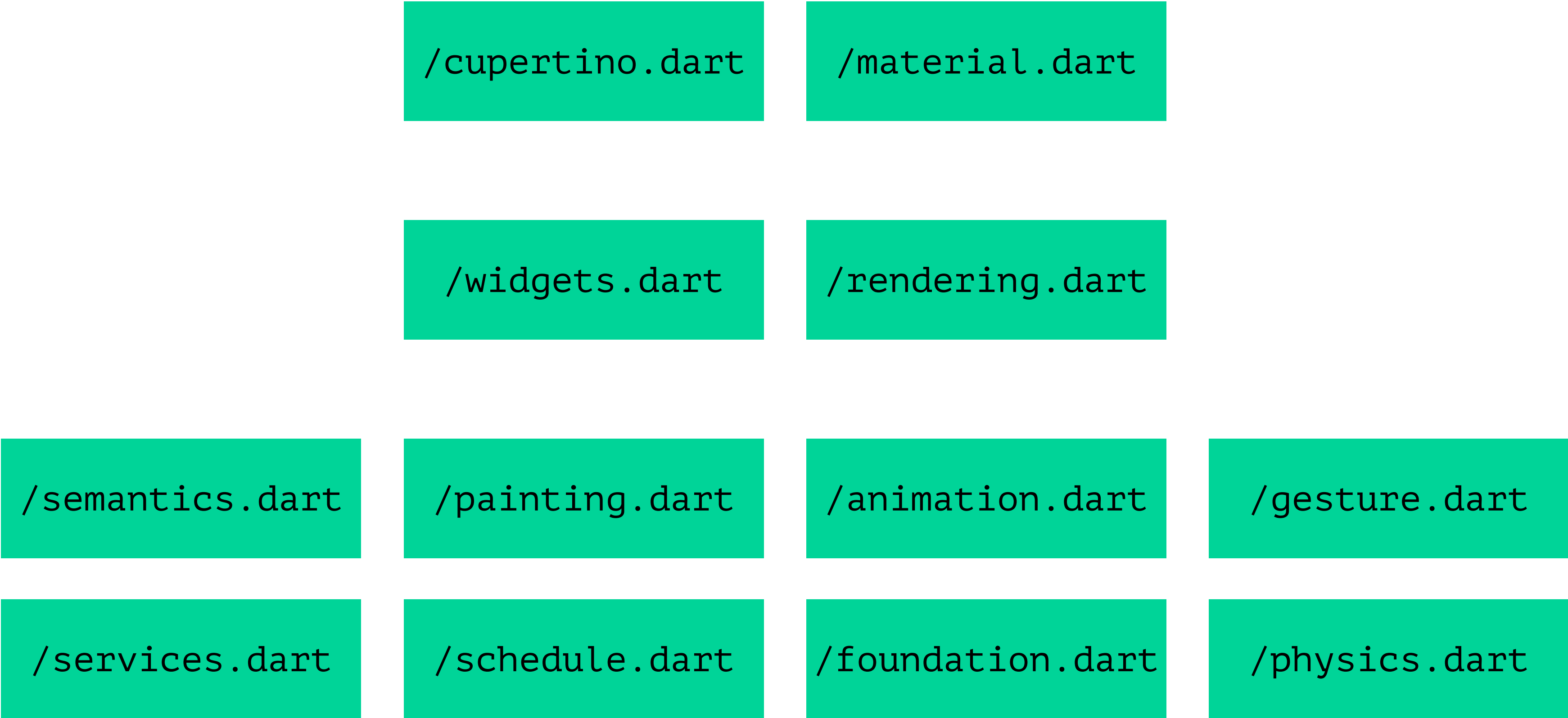
Многослойная архитектура Flutter

# **Библиотека UI компонентов**

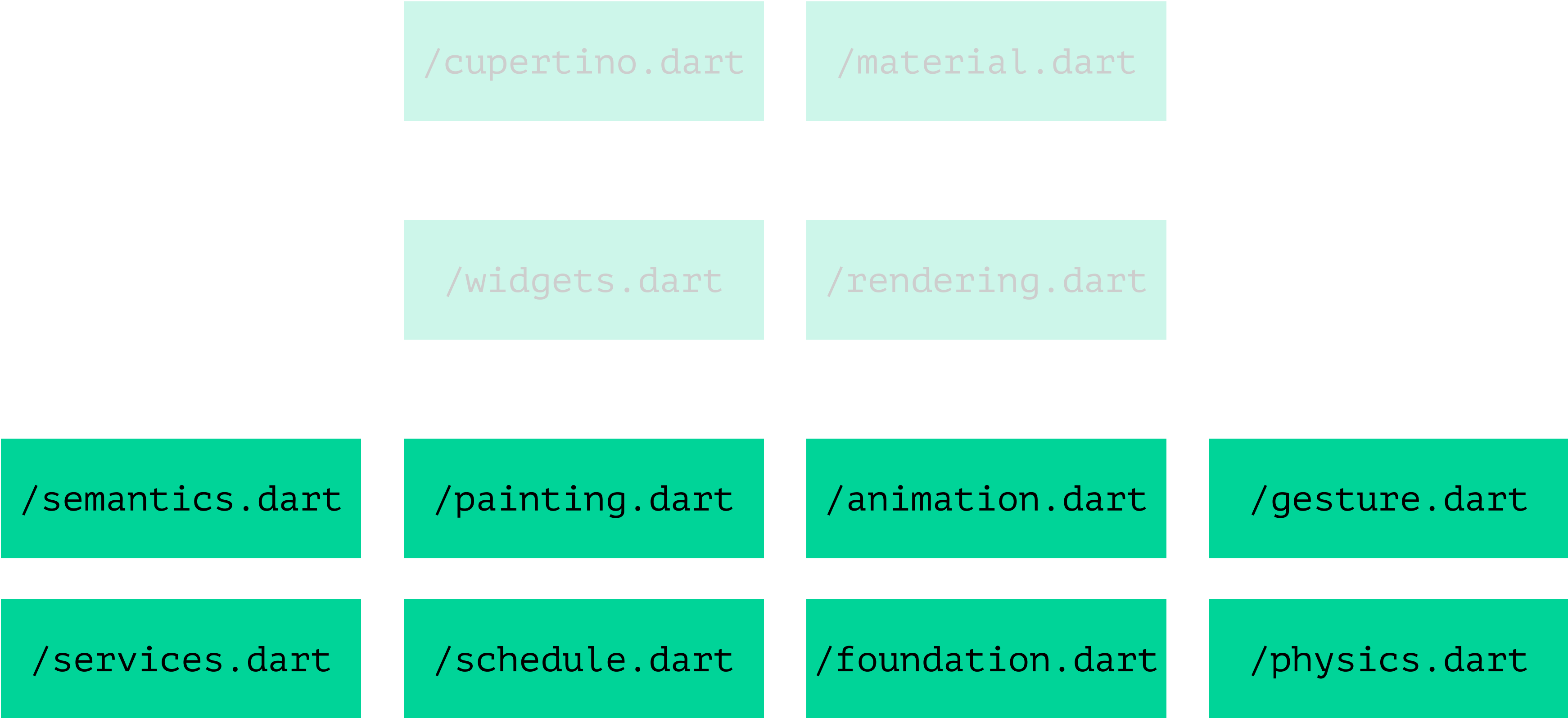
**Общий код для iOS и Android**

**Многослойная архитектура Flutter**

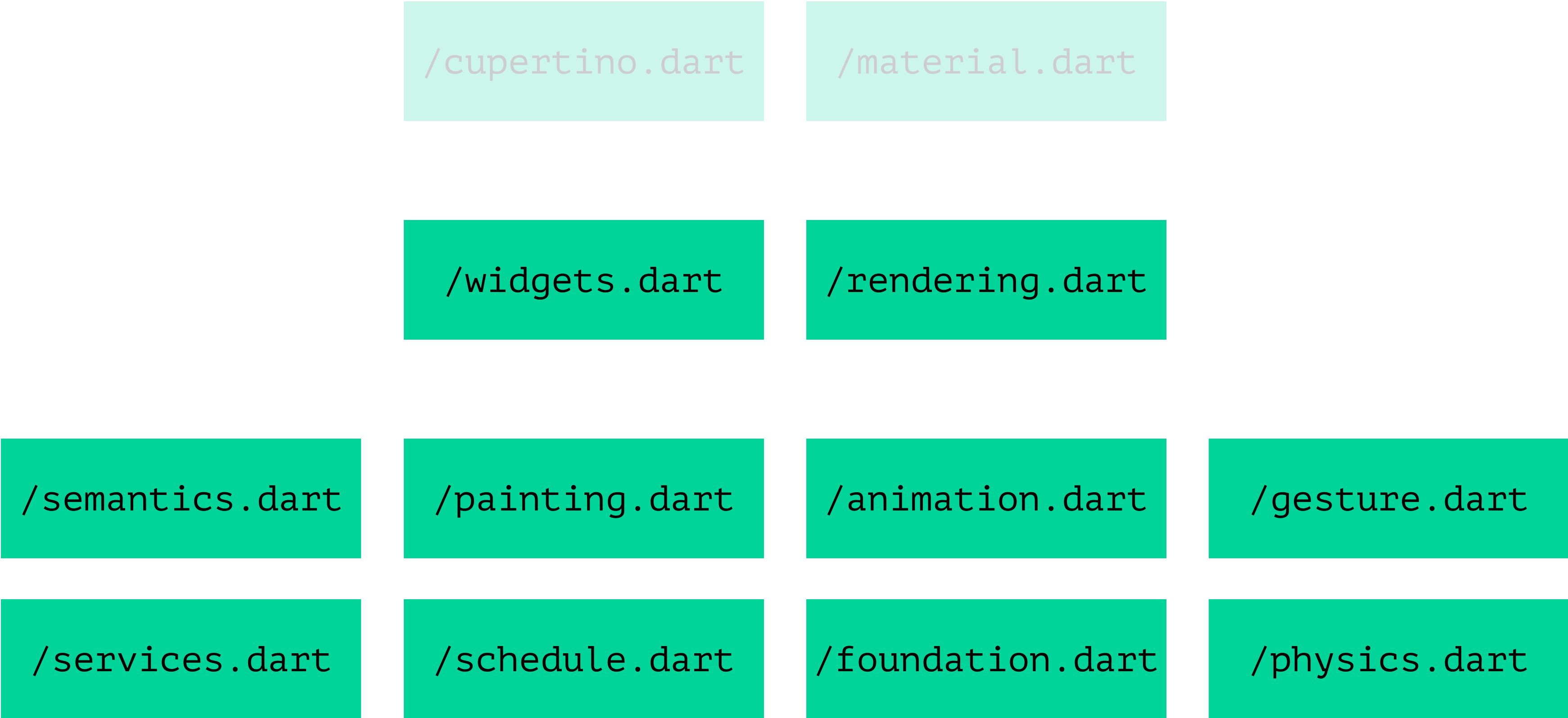
# Многослойная архитектура Flutter



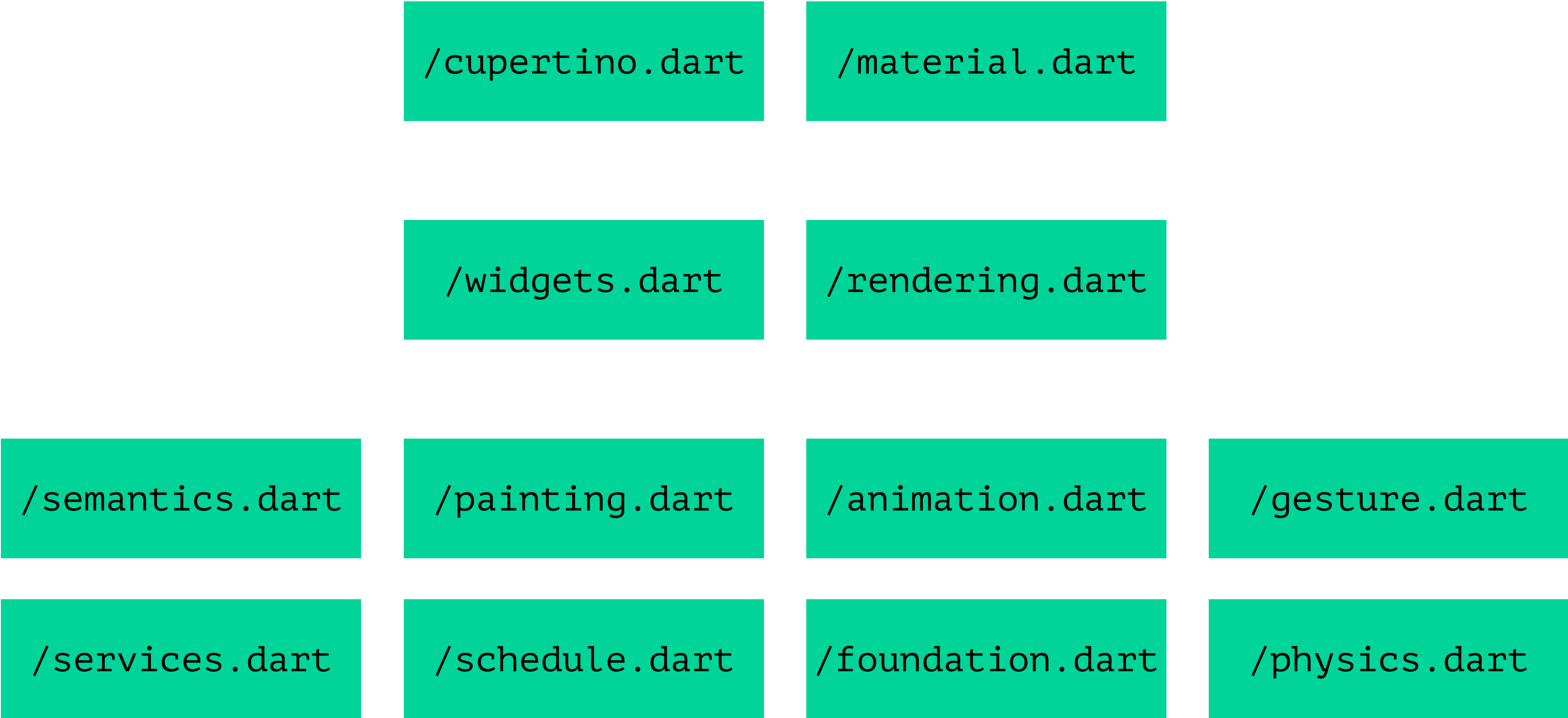
# Многослойная архитектура Flutter



# Многослойная архитектура Flutter



# Многослойная архитектура Flutter





# Тестирование

Unit tests

Widget tests

Integration Native

~~Golden test~~

~~Integration from Native to Flutter~~

Приложение

Почему Flutter

Интеграция

**Влияние на метрики**

	<b>Время старта приложения</b> миллисекунды
<b>iOS</b>	<b>60 ~ 150</b>
<b>Android</b>	<b>40 ~ 80</b>

	<b>Изменения в размере приложения</b> мегабайты
<b>iOS</b>	<b>~ 20.0</b>
<b>Android</b>	<b>~ 9.0</b>

# Как уменьшить размер

**Переиспользовать нативный код**

Не дублировать `assets` (шрифты, иконки, аудиофайлы)

Удалить неиспользуемые SDK виджеты

# Как уменьшить размер

**Переиспользовать нативный код**

**Не дублировать assets (шрифты, иконки, аудиофайлы)**

Удалить неиспользуемые SDK виджеты

# Как уменьшить размер

Переиспользовать нативный код

Не дублировать `assets` (шрифты, иконки, аудиофайлы)

Удалить неиспользуемые SDK виджеты

Приложение

Почему Flutter

Интеграция

Влияние на метрики

**Результаты**



# Результаты

В проде 6 месяцев

Низкий `crash rate`

Скорость разработки  $\sim 2.5$  быстрее

# Выводы

Не все работает из коробки

Обновлять Flutter придется часто

Нужно учитывать влияние Flutter на размер и время запуска приложения

Рефакторить нативный код

В команде нужны нативные разработчики под iOS и Android

**Mikhail Zinov**

[zinov.tech](http://zinov.tech)