18:00 🍕 Pizza, beer, drinks

18:30 📊 Adam Vesecký: Android vs Flutter - which is better from which aspects?
18:55 🖥 Johannes Milke: Develop desktop applications with Flutter
19:20 📱 Vince Varga: Interaction between Flutter and native iOS or Android

19:45 💬 Open discussion

The event's location, food, and drinks are provided by codecentric AG.

Interactions between Flutter and native hosts

github.com/vargavince91/flutter-munich-april

Vince Varga • 4 April 2019, Munich

Overview

github.com/vargavince91/flutter-munich-april

- Platform channels
- Event vs Method Channels
- Create and Publish a Flutter plugin
- Start an activity/view controller
- Platform views

Why?

You might face issues that you cannot solve only in Flutter

- Access platform APIs
- Leverage third party SDKs
- Performance
- Flutter is great for UI, Dart is a good language, but community is not that big
- New features coming out on platforms (e.g. Android Q)

Packages and plugins

Packages enable the creation of modular code that can be shared easily.

Dart packages

- General packages written in Dart
- Private repo, local packages, etc… supported
- Private pub server???

Plugin packages

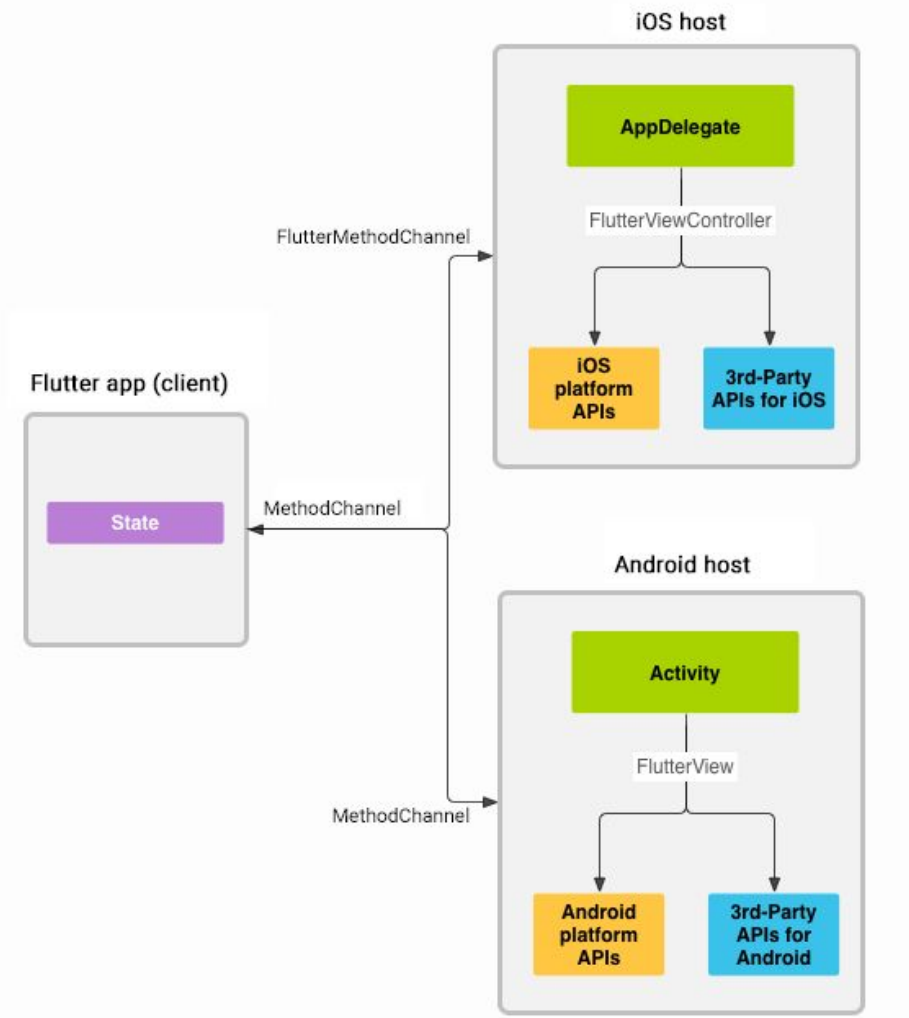
- A specialized Dart package which contains API written in Dart code combined with a platform-specific implementation for Android (using Java or Kotlin) and/or for iOS (using Objective-C or Swift)

Platform channels

Messages are passed between the client (UI) and host (platform) using platform channels

The Flutter portion of your app sends messages to its host, the iOS or Android portion of your app, over a platform channel.

The host listens on the platform channel, and receives the message. It then calls into any number of platform-specific APIs—using the native programming language—and sends back a response to the client, the Flutter portion of your app.



Platform channels

- Easy to establish these channels on Flutter projects.
- You can send numbers, strings, booleans, lists, maps, byte arrays
- The serialization and deserialization of these values to and from messages happens automatically when you send and receive values.
- Setup is slightly different in plugins and in apps, but principles are the same

Platform channels: Method and event channels

Method Channels when

- You have to call a function once and you receive the result once
- When you can repeat the first item (focus on: Dart knows when to fetch the results again)

Event channels when

- Need to pass results continuously from platform to Flutter
- Only the host knows when new events are triggered:
  - Sensor information
  - Anything with listeners
  - Esptouch example

Code example: Method channels

```dart
class AndroidWifiInfo {
  static const _channelName = 'eng.smaho.com/android_wifi_info';
  static const MethodChannel _channel = const MethodChannel(_channelName);

  static Future<bool> get ssidHidden {
    return _channel.invokeMethod('ssidHidden');
  }

  static Future<int> get linkSpeed {
    return _channel.invokeMethod('linkSpeed');
  }

  static Future<String> get ssid {
    return _channel.invokeMethod('ssid');
  }
}
```

```
public class AndroidWifiInfoPlugin implements MethodCallHandler {
  public static void registerWith(Registrar registrar) {
    final MethodChannel channel =
      new MethodChannel(registrar.messenger(), channelName);
    channel.setMethodCallHandler(new AndroidWifiInfoPlugin(registrar));
  }

  private AndroidWifiInfoPlugin(Registrar registrar) {
    this.registrar = registrar;
  }

  @Override
  public void onMethodCall(MethodCall call, Result result) {
    switch (call.method) { // and many more cases
      case "ssidHidden": ssidHidden(call, result); return;
      default: result.notImplemented(); break;
    }
  }

  private void ssidHidden(MethodCall call, Result result) {
    result.success(wifiInfo.ssidHidden());
  }
}
```

Before you write platform channels...

Do you have to write platform channels?

- Check existing plugins. New plugins supported features on different platforms are released, so what's not supported today, could be supported in a month.
- Maybe you don't even need to go to the platform

- Think about what you need to pass between the platforms
    - Instead of bytestreams of images, can you just send a source and destination file name?

When working with plugins and writing platform channels

- Run/ write example app
  - MQTT, QR readers...
- Assume existing plugins are not always complete
  - Connectivity BSSID
- Use IDEs for each platform (at least in the beginning)
  - Xcode for Objective-C, Swift
  - Android Studio for Java, Kotlin
  - Dart

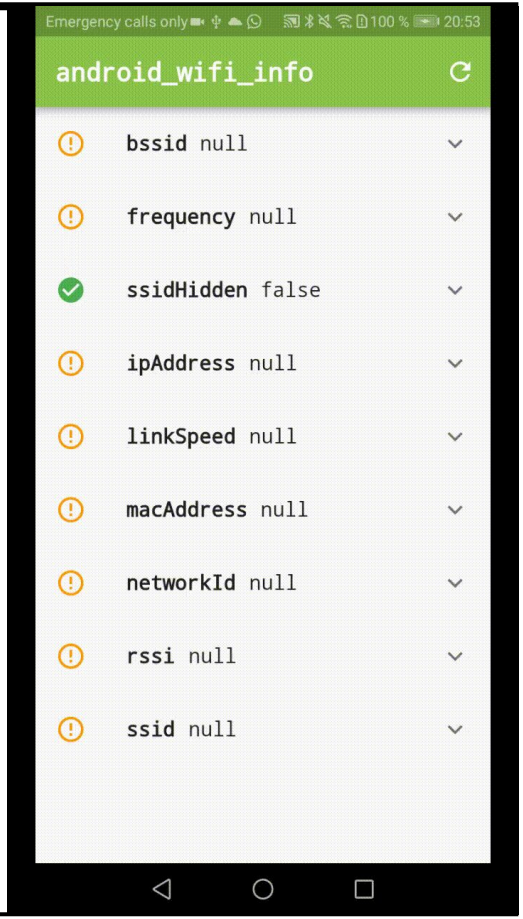
Publishing on Dart Pub

Steps

1. Create plugin using flutter create
    - Need to specify that you create a plugin
    - Set Project name and organization
    - Set iOS and Android language
2. Implement minimum useful features
3. Clean up: changelog, unit tests, README
4. pub publish --dry-run
5. Add author to pubspec.yaml
6. pub lish (or pub publish)
7. DONE!

Plugins for only one platforms

- Provide a nice interface from Dart
- Solve the problem once
  - Creating a plugin requires some boilerplate
  - No real type safety while passing parameters and invoking methods
- You can combine two plugins to make a cross-platform plugin



Plugins for only one platforms

```
demoDartDocs() {
  |
}

@override
Widget build(BuildContext context) {
  return MaterialApp(
    debugShowCheckedModeBanner: false,
    home: Scaffold(
        appBar: AppBar(
          actions: <Widget>[
            IconButton(
              icon: Icon(Icons.refresh),
              onPressed: fireAllFutures,
            ) // IconButton
          ], // <Widget>[]
          title: Text(
```

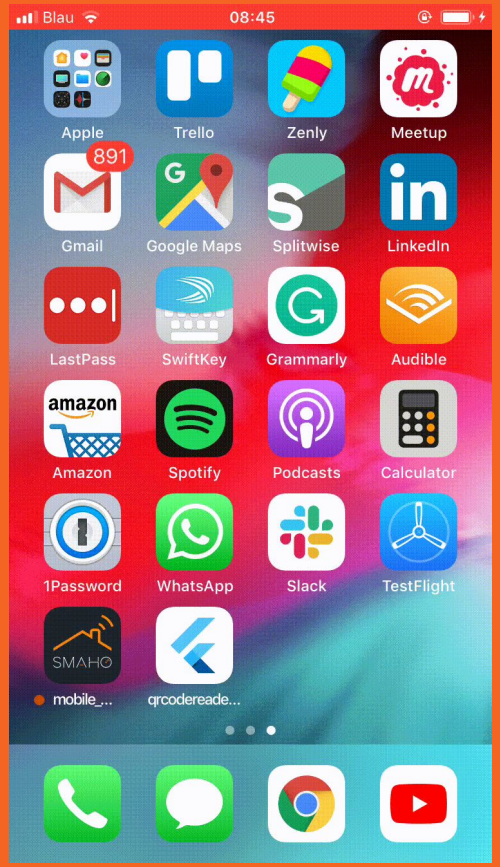
Combine plugins

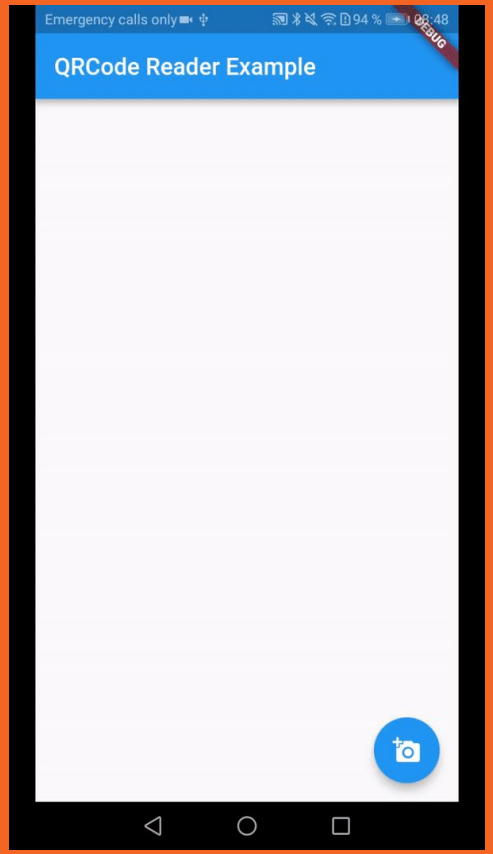
```dart
import 'dart:io';

import 'package:android_wifi_info/android_wifi_info.dart';
import 'package:ios_network_info/ios_network_info.dart';

get bssid {
  if (Platform.isAndroid) {
    return AndroidWifiInfo.bssid;
  } else if (Platform.isIOS) {
    return IosNetworkInfo.bssid;
  }
  throw Exception('WiFi BSSID is not supported on this platform');
}
```

Just start an activity...

- You need to do something that you know how to do in Java/Kotlin/Objective-C/Swift.
- We know we can pass information from the Dart code to the platform-specific code and receive some result using method channels.
- **You can call an activity or view controller.**

QR code reader





Start an activity

From Dart, you just invoke a method

```
static const _channelName = 'com.example/something'
static const MethodChannel _methodChannel =
    MethodChannel(_channelName);

final int result =
    await _methodChannel.invokeMethod('doSomething', input);
```

Start an activity

Get arguments. Start activity for result. Send result Back

```
// In method call handler
int value = methodCall.arguments();
if (methodCall.method.equals("doSomething"))
  launchActivity(value);

// In your class
private void launchActivity(int value) {
  Intent fullScreenIntent = new Intent(this, ValueActivity.class);
  fullScreenIntent.putExtra(ValueActivity.VALUE, value);
  startActivityForResult(fullScreenIntent, VALUE_REQUEST);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
  // check request and result code, then...
  result.success(data.getIntExtra(ValueActivity.VALUE, 0));
}
```

Platform Views: AndroidView and UIKitView

Embed an Android or iPhone platform control in a Flutter app. Platform view widgets are just widgets: you can compose them as any other widgets.

Last resort: shouldn't use it unless you have to.

I didn't have to work with it yet, but want to mention so that you know what to look up…

Google Maps and WebView

Thank you all for coming!

Upcoming events

1. Presentations for June?
2. Coding night in July
3. Flutter Stories in August

We need speakers!

THU, JUN 6, 6:00 PM

Flutter talks in June

Needs a location

The date and time are not final (we might have to cancel if we don't have enough speakers). More information is coming after the April 4th event.

3 attendees Manage Going

THU, JUL 4, 6:00 PM

Flutter Independence Coding Night

SinnerSchrader Deutschland GmbH

The concept is simple: we meet and we build beautiful apps together. 👷We are going to prepare a list of awesome tutorials, so you can create something on your own, all the while we are there to help you in case you get stuck. 🚀You can bring your own project or work on open-source projects. 🚵I'm working on a new Flutter tutorial site which will debut at this event, so if you feel adventurous, you can go through one of them and give me feedback. ❓ You can come with questions, we'll do our best to help you with any issue you bring. 🎤At the end of the event, you'll have the opportunity to show to the rest of us what you worked on (presenting is optional, but encouraged).

12 attendees Manage Going

Hands-on coding event

Flutter Stories

TUE, AUG 6, 6:00 PM

Built with Flutter - Share your Flutter story

Needs a location

Have you built something with Flutter? This is the time to show it to the world! This event is focussing on Flutter stories. You can share what you created with Flutter and how your experience was. This event will not focus heavily on coding, so feel free to invite your product owner, designer or entrepreneur friends. We have an attendee limit of 150 people.

6 attendees Manage Going