

CPD Flutter

Tristan Damm, Hauke Ingwersen

Roadmap

- Flutter allgemein
- Dart allgemein
- Laboraufgabe 1 (Dart)
- UI und State in Flutter
- Laboraufgabe 2 (Flutter)
- Weiteres
- Vor/Nachteile

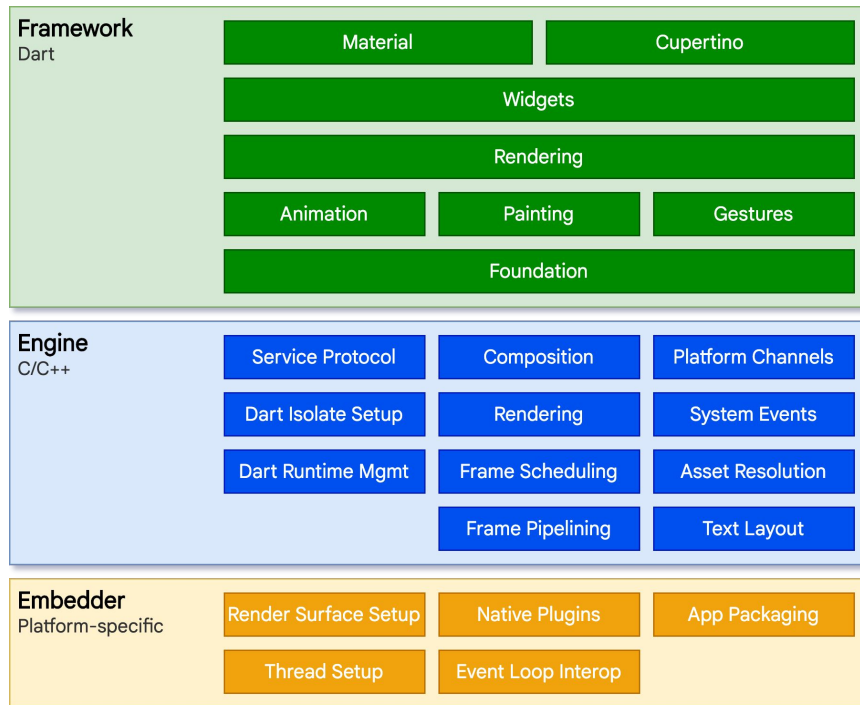
Flutter - Was ist das?

- Cross Platform Framework von Google
- Entwicklung von Applikationen für iOS und Android
- Bietet auch die Möglichkeit für Web, Linux und Windows zu entwickeln
- Open-Source: github.com/flutter/flutter
- Website: flutter.dev

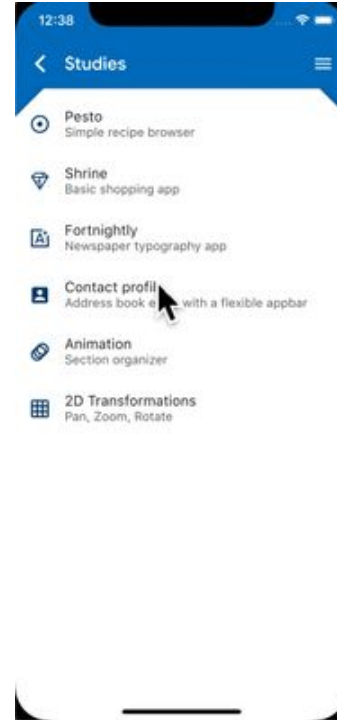
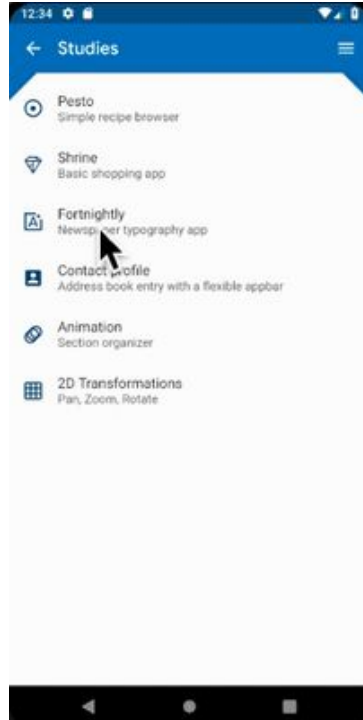
Flutter - Besonderheiten

- Dart als Programmiersprache
- Kompiliert zu nativen Maschinencode
- Alle UI-Elemente rendert Flutter selbst
- UI-Entwicklung mit “Widgets”

Flutter - Architektur



Flutter - Plattform Unterschiede



Flutter - Plattform Unterschiede

- Typography (Roboto für Android, San Francisco für iOS)
- Navigation
- Scrolling
- Gestures

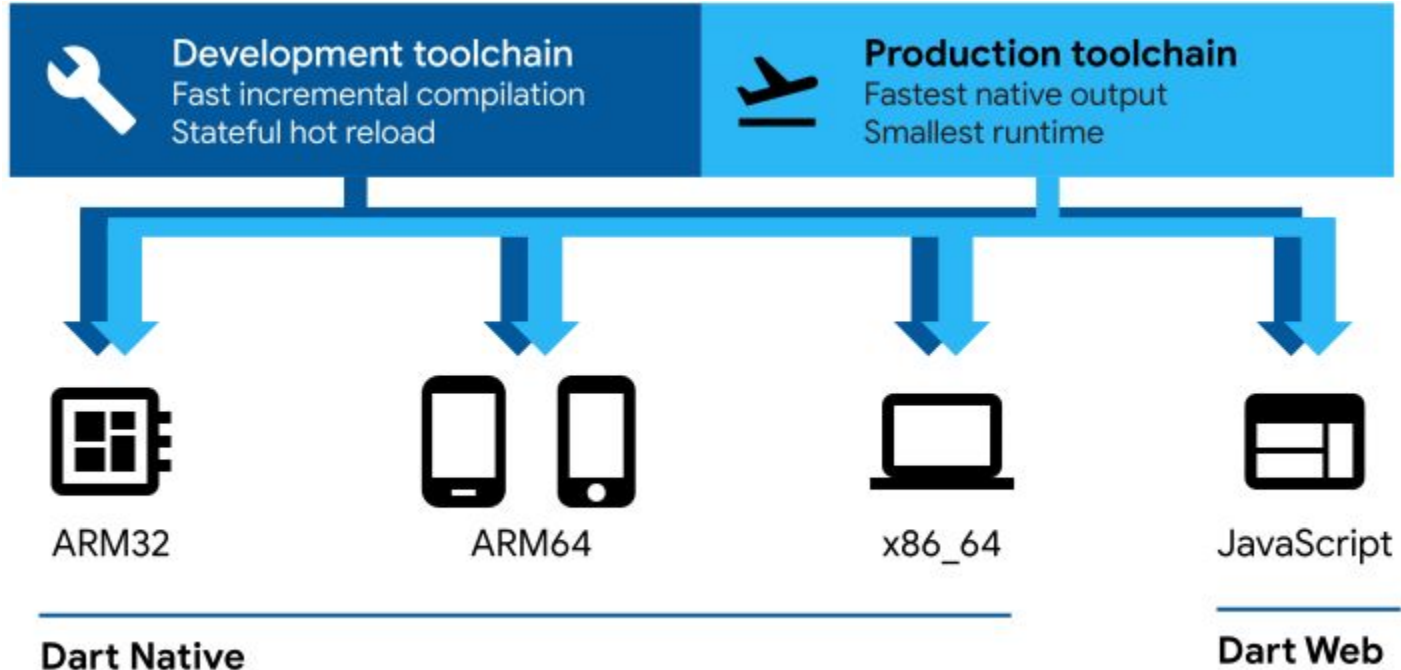
Flutter - Entwicklung

- Flutter-Plugin für Android Studio
- Flutter-Extension für Visual Studio Code
- **Playground über DartPad:**
 - `dartpad.dev`
 - Für Laboraufgaben relevant!

Dart - Was ist Dart?

- Static Types
- Flexible mit Dynamic Types
- Mix aus Javascript und Java
- Flexible execution runtime platform

Dart - Multiple Platforms



Dart - Woher kommt das?

- Erstellt von Google im Jahr 2011
- Übernommen von Flutter
- Dient als Basis für Flutter

Dart - Language Intro

Laboraufgabe 1

github.com/dekort-dev/CPD-Flutter

UI und State in Flutter

- UI wird aus Widgets gebaut
- Komposition aus mehreren Widgets möglich
- Auswahl von mehreren Komponenten aus dem Android Design (Material) und iOS Design (Cupertino)
- Stateless und Stateful Widgets

You have pushed the button this many times:

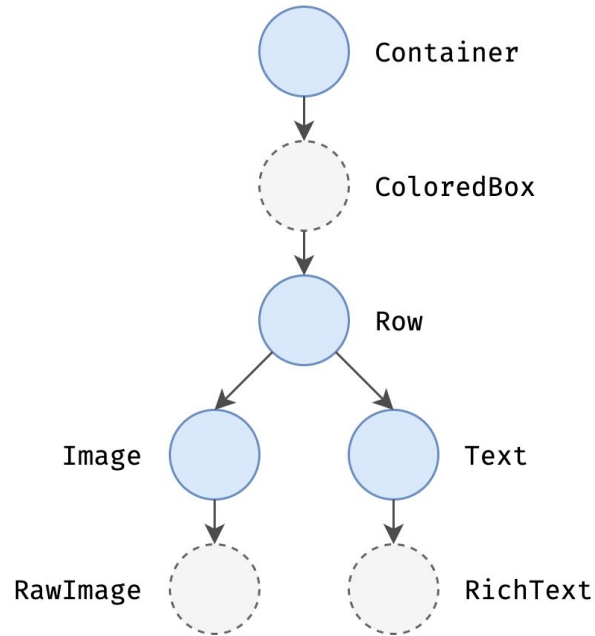
0



UI und State in Flutter - Alles ist ein Widget


- UI items wie z.B. AppBar
- Layoutelemente wie Center/Column/Row Widgets
- Widgets kann man Eigenschaften übergeben, wie Größe oder Farbe
- Flutter bietet einen großen Katalog an Widgets
 - docs.flutter.dev/development/ui/widgets

UI und State in Flutter - Widgetbaum






```
1  import 'package:flutter/material.dart';  
2  
3  void main() {  
4      runApp(MyApp());  
5  }  
6
```



```
1 class MyApp extends StatelessWidget {  
2   @override  
3   Widget build(BuildContext context) {  
4     return MaterialApp(  
5       title: 'Flutter Hello World',  
6       home: Scaffold(  
7         appBar: AppBar(  
8           title: Text('Flutter Hello World'),  
9         ),  
10        body: Center(  
11          child: Text('Hello World'),  
12        ),  
13      ),  
14    );  
15  }  
16 }
```

UI und State in Flutter - Stateless Widgets

- Stateless Widgets können ihren Zustand während der Laufzeit der Anwendung nicht ändern, was bedeutet, dass die Widgets nicht neu gerendert werden können, während die Anwendung aktiv ist.
- z.B. Icons oder statischer Text



```
1 class MyHomePage extends StatefulWidget {  
2   final String title;  
3  
4   const MyHomePage({  
5     Key? key,  
6     required this.title,  
7   }) : super(key: key);  
8  
9   @override  
10  _MyHomePageState createState() => _MyHomePageState();  
11 }
```

```
1 class _MyHomePageState extends State<MyHomePage> {
2   int _counter = 0;
3
4   void _incrementCounter() {
5     setState(() {
6       _counter++;
7     });
8   }
9
10  @override
11  Widget build(BuildContext context) {
12    return Scaffold(
13      appBar: AppBar(
14        title: Text(widget.title),
15      ),
16      body: Center(
17        child: Text(
18          '$_counter',
19          style: Theme.of(context).textTheme.headline4,
20        ),
21      ),
22      floatingActionButton: FloatingActionButton(
23        onPressed: _incrementCounter,
24        tooltip: 'Increment',
25        child: const Icon(Icons.add),
26      ),
27    );
28  }
29 }
```

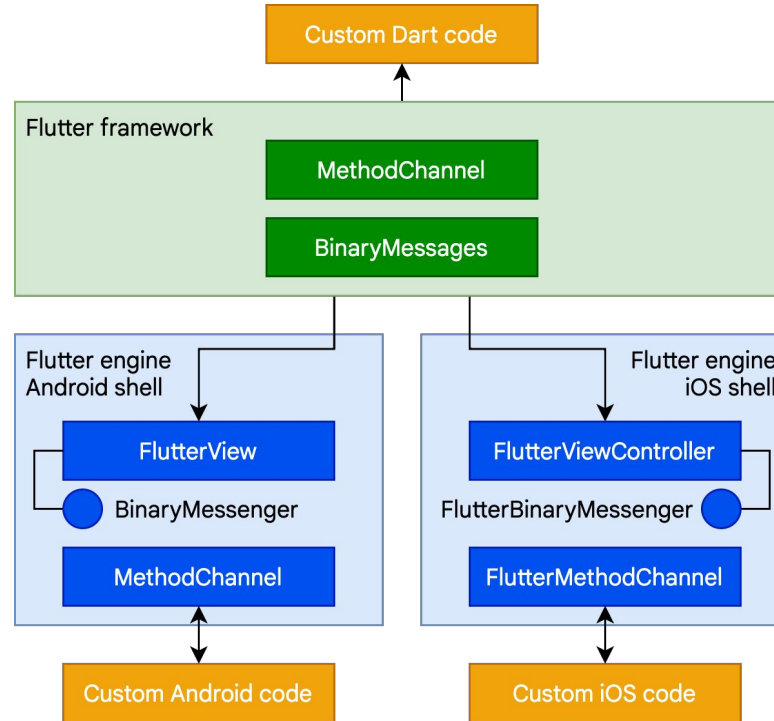
UI und State in Flutter - Stateful Widgets

- Stateful Widgets können ihren Zustand ändern
- **setState()** benachrichtigt das Framework, dass sich der interne Zustand dieses Objekts geändert hat.
- Daraufhin folgt ein re rendering des User Interfaces
- Widgets die als **const** markiert sind werden vom Compiler optimiert, da sie sich nicht ändern zur Laufzeit

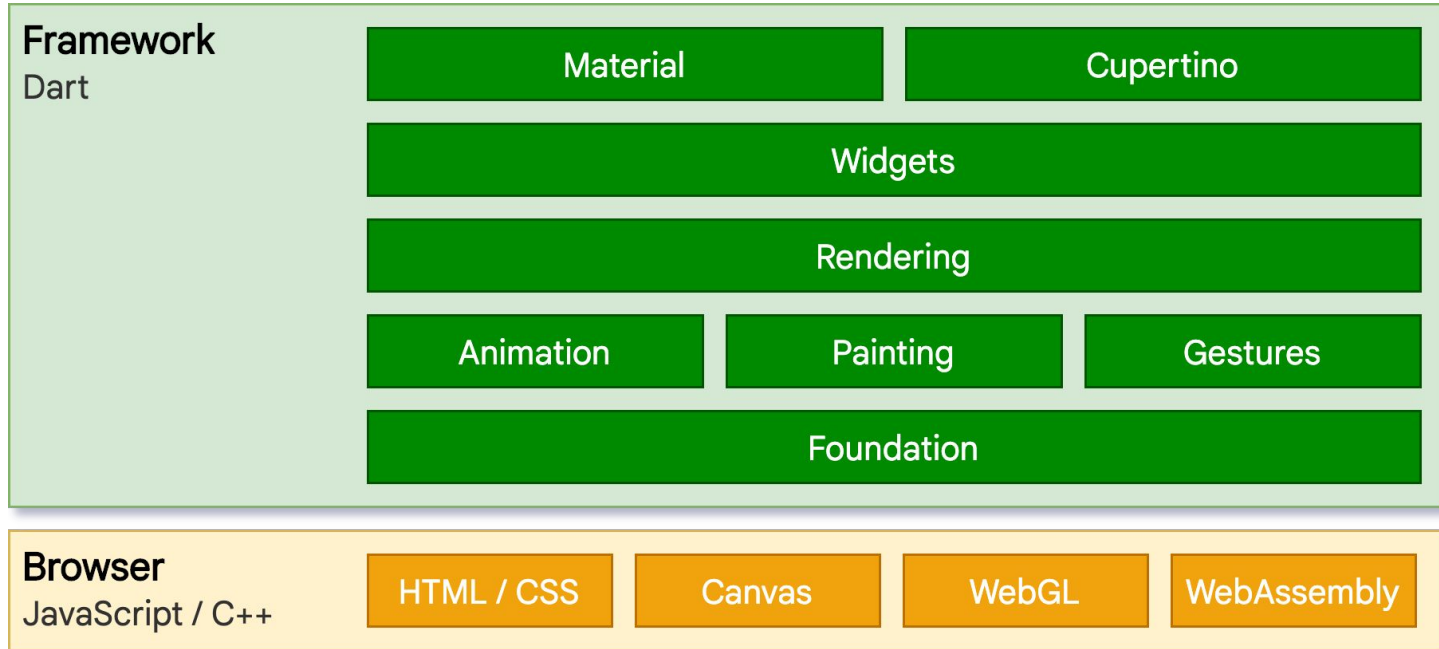
Laboraufgabe 2

github.com/dekort-dev/CPD-Flutter

Weiteres - Platform channels



Weiteres - Flutter Web-Unterstützung

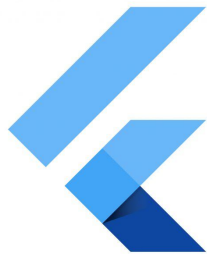


Vorteile

- Eine gemeinsame Codebasis für mehrere Plattformen
- Spart Zeit und damit auch Entwicklungskosten
- Gute Performance der Applikationen
- Große Community (140.000 Sterne auf GitHub)

Nachteile

- Für plattformspezifische Features muss man meist auf native Funktionen zugreifen
- Die Apps sind meist größer
- Drittanbieter auswahl an Libraries ist noch nicht so gut wie bei anderen Frameworks
- Verschachtelung mehrerer Widgets kann schnell unübersichtlich werden



Noch Fragen?