

Flutter a new hope for single code
base Apps

2019 kickoff with mobile.lx

Bruno Oliveira

Mobile developer @Unbabel

Flutter a new hope for single code base Apps

- Problem

Flutter a new hope for single code base Apps

- Problem
- Other Frameworks

Flutter a new hope for single code base Apps

- Problem
- Other Frameworks
- Flutter analysis

Flutter a new hope for single code base Apps

- Problem
- Other Frameworks
- Flutter analysis
- Code examples

Flutter a new hope for single code base Apps

- Problem
- Other Frameworks
- Flutter analysis
- Code examples
- Useful information

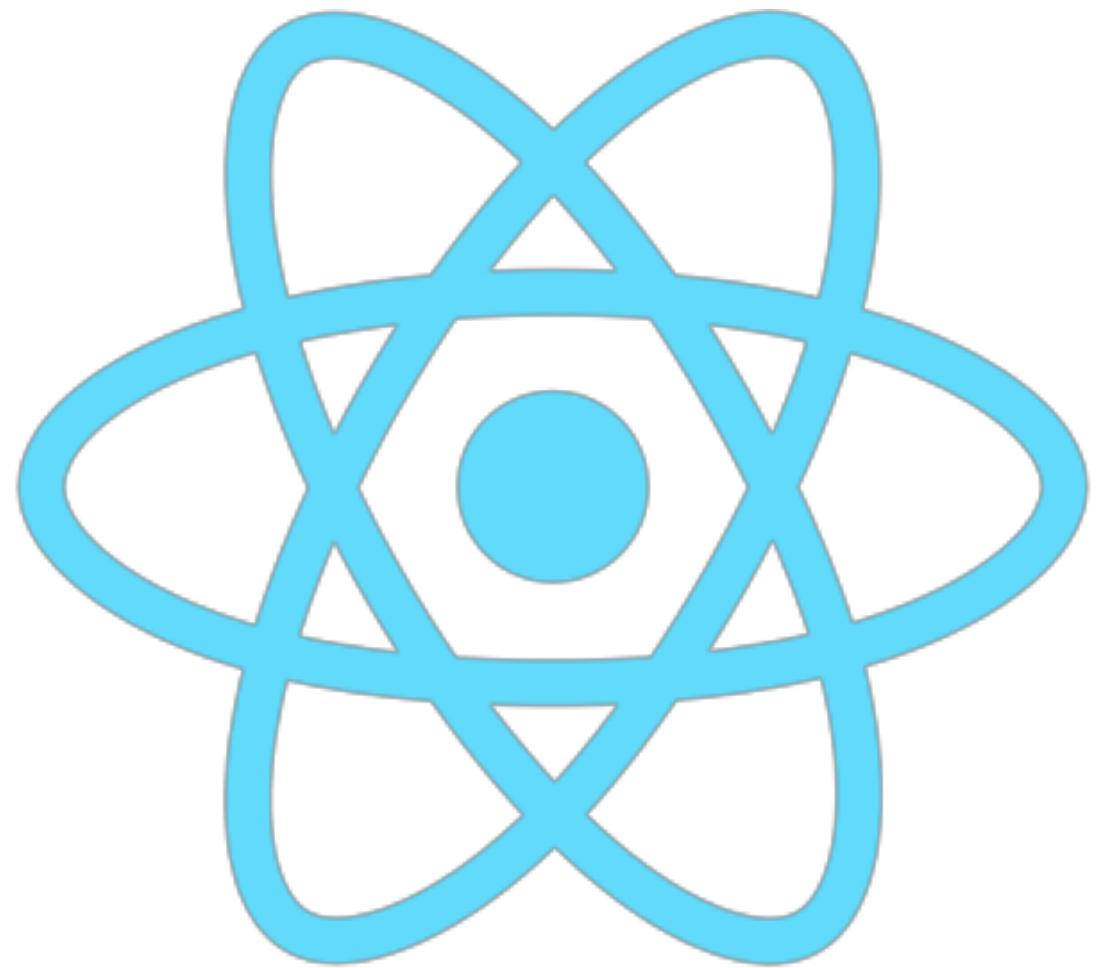
Problem

- Android
- iOS / MacOS
- Windows

Problem

- Android
 - Android 4.0, 4.4, 5 ...
 - Custom Roms
- iOS / MacOS
- Windows

Existing alternatives



React Native

Pros

- Native components
- Javascript
- Development app reload time
- High percentage of reused code

Cons

- Javascript
- The bridge
- Performance
- iOS/Android



Xamarin

Pros

- Compiled into bytecode
- Also supports some MacOS components
- Basic MVPs can have up to 100% codeshare with .Forms

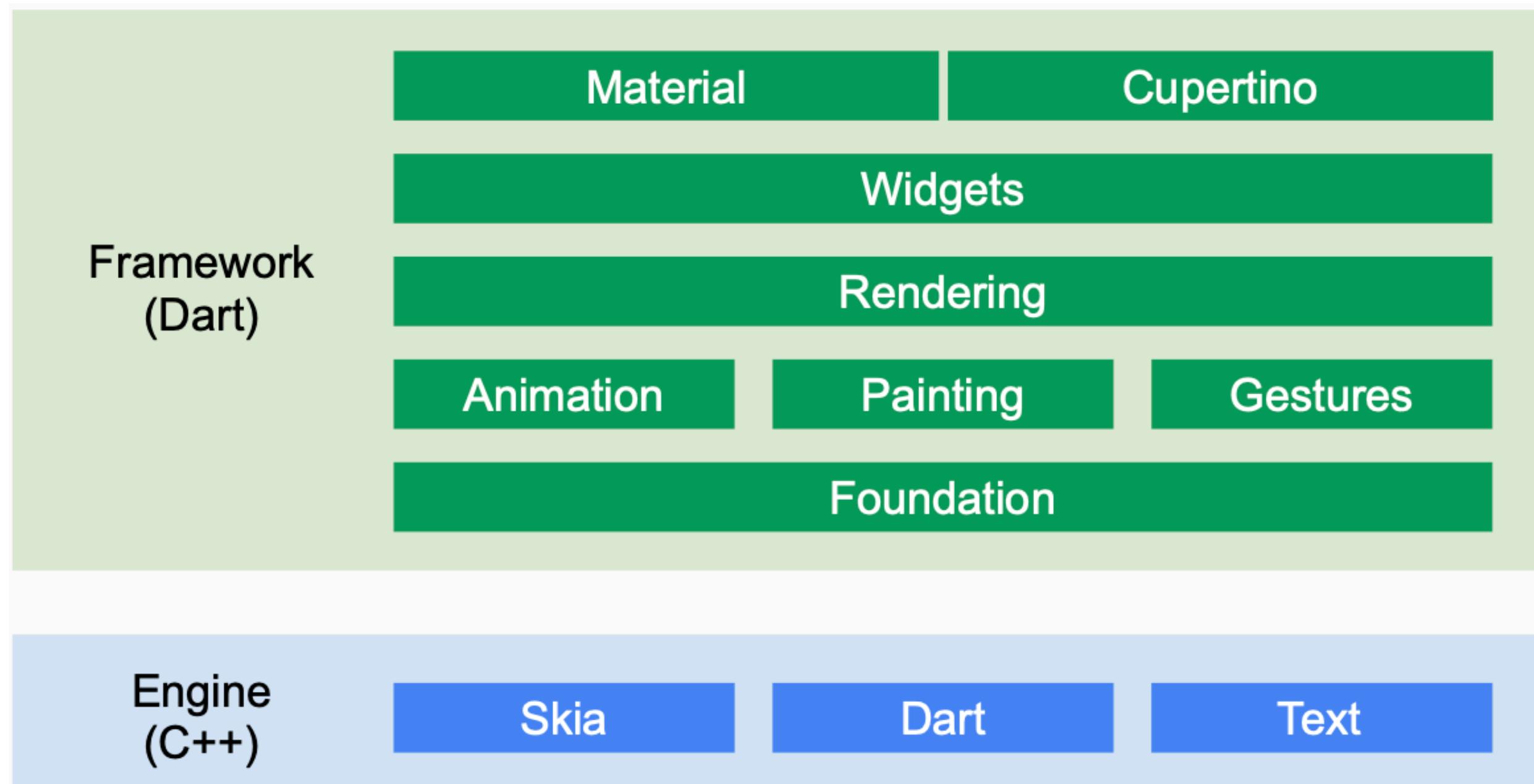
Cons

- Performance on heavy UI apps
- Size of the community
- Larger app size
- Delayed support for new OS Platform updates



Flutter

Flutter how it works





Dart

Pros

- Syntax similar to Java
- Interpreted and compiled
- Hot reload
- Asynchronous programming

Cons

- Not well established
- Syntax sugar features



Flutter

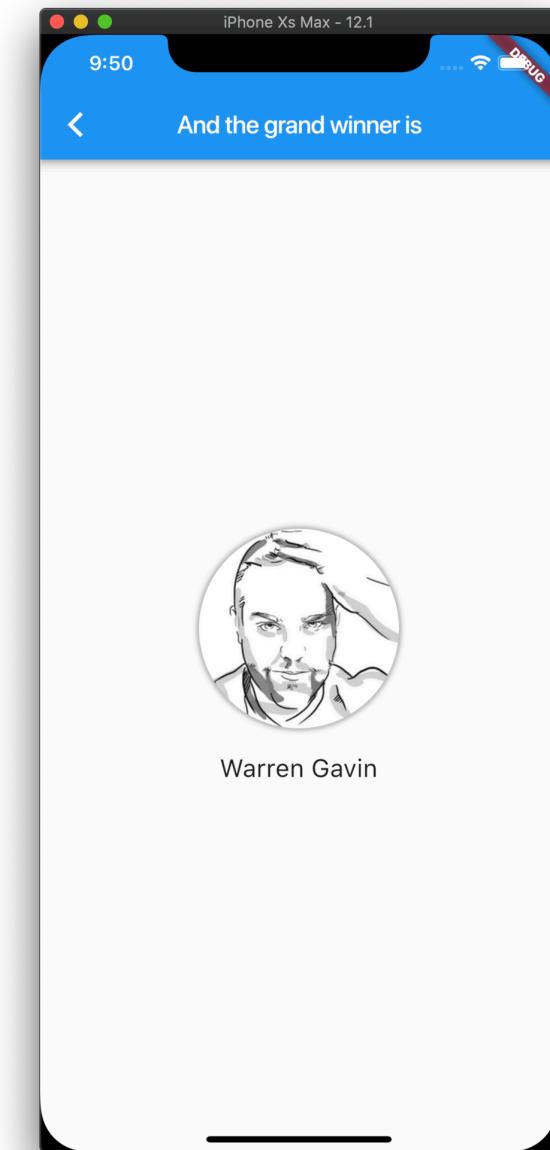
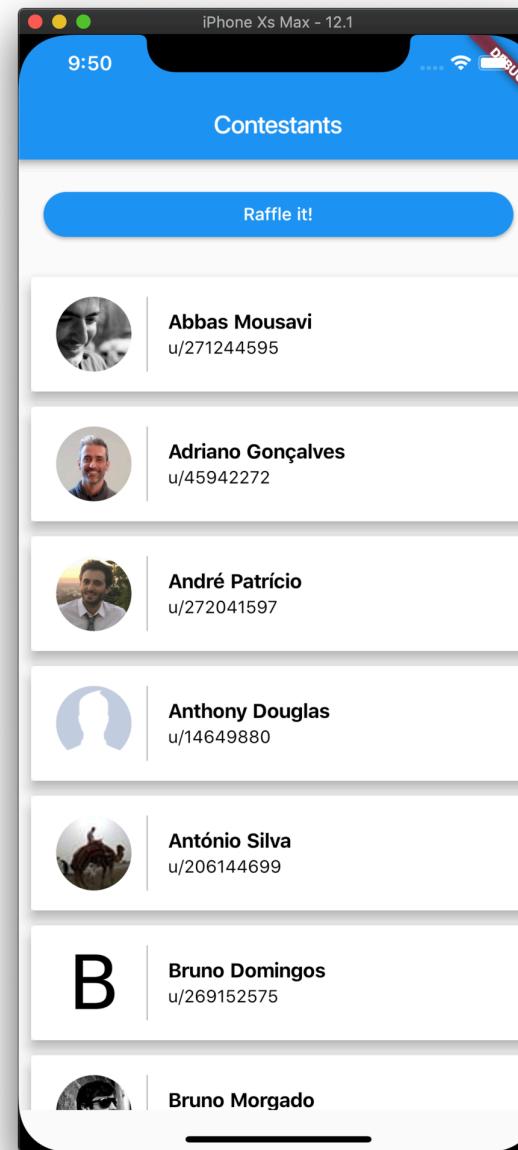
Pros

- Fully reactive mindset
- Easy learning curve
- Low boilerplate code
- Performance on low-end devices
- Multiple IDE support

Cons

- "official" architecture still not defined
- App size
- Some questions about iOS support for OS updates

Flutter in action - Screens



Flutter in action - Screens



Contestants

Raffle it!

	Abbas Mousavi u/271244595
	Adriano Gonçalves u/45942272
	André Patrício u/272041597
	Anthony Douglas u/14649880
	António Silva u/206144699

← And the grand winner is

Pedro Okawa

Lucky One Screen

```
Scaffold(  
    appBar: AppBar(  
        title: Text('And the grand winner is'),  
    ),  
    body: Center(  
        child: Column(  
            mainAxisAlignment: MainAxisAlignment.center,  
            children: <Widget>[  
                _photoContainer(),  
                _nameContainer()  
            ],  
        ),  
    ),  
)
```

Navigator

```
Navigator.push(  
    context,  
    MaterialPageRoute(builder: (context) => RaffleWinnerScreen(winner))  
);
```

Rich Text

```
Directionality(  
    textDirection: TextDirection rtl,  
    child: RichText(  
        textAlign: TextAlign center,  
        text: TextSpan(  
            style: TextStyle(fontSize: 40, fontWeight: FontWeight bold),  
            text: _text)),  
);
```

Photo Container

```
Container(  
    child: Hero(  
        tag: winner.photoURL,  
        child: Container(  
            decoration: BoxDecoration(  
                boxShadow: [BoxShadow(spreadRadius: 2, blurRadius: 2, color: Colors.black26 )],  
                shape: BoxShape.circle,  
                image: DecorationImage(  
                    fit: BoxFit.cover,  
                    image: NetworkImage(winner.photoURL)  
                )  
            )),  
        ),  
    ),
```


HTTP Requests

```
import 'package:http/http.dart' as http;  
  
static Future<http.Response> _fetchContestants() {  
  return http.get(MeetupURL);  
}
```

Response parsing

```
import 'dart:convert';

static Future<List<Contestant>> fetchContestants() async {
    final response = await _fetchContestants();

    if (response.statusCode == 200) {
        final values = json.decode(response.body);
        return values.map((value) => Contestant.fromJson(value)).cast<Contestant>().toList();
    } else {
        throw Exception('Failed to load post');
    }
}
```

Detailed parsing

```
factory Contestant.fromJson(Map<String, dynamic> json) {  
    return Contestant(  
        identifier: json["member"]["id"],  
        name: json["member"]["name"],  
        photoURL: json['member']['photo']['photo_link'],  
        isOrganizer: json['member']['event_context']['host'],  
        hasConfirmed: (json['response'] as String).toLowerCase() == "yes");  
}
```

Conclusions

- Performance

Conclusions

- Performance
- Fast development cycles

Conclusions

- Performance
- Fast development cycles
- High percentage of reused code

Conclusions

- Performance
- Fast development cycles
- High percentage of reused code
- Perfect for companies with custom experiences

Useful links

- <https://github.com/bmolineira/mobile-lx.raffle>
- <https://flutter.io>
- <https://github.com/Solido/awesome-flutter>
- <https://github.com/2d-inc/HistoryOfEverything>

Questions?