



#IntroFlutter

da beginner a beginner

Damiano “dag7” Gualandri

Developer

dag7.it

@dag7_

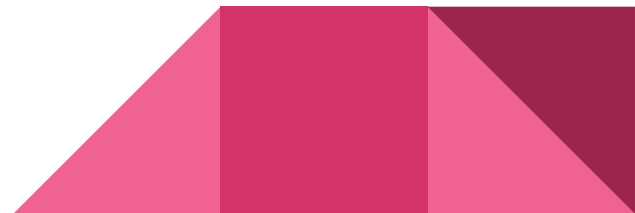
Cos'è Flutter?

Un SDK con il quale è possibile costruire applicazioni in “material design” in maniera molto semplice

Funziona per Android, per iOS e per Web.

È open-source, sviluppata da Google

Un unico codice per tutti!



A chi è indirizzato?

- **Designers** che vogliono un'esperienza personalizzata su Android e iOS
- **Collaudatori (tester)** che vogliono il proprio prototipo già funzionante
- **Sviluppatori** che beneficiano di strumenti per gli sviluppatori fantastici, un linguaggio semplice da utilizzare, un set vasto di widget e un grande supporto dell'IDE. Flutter aiuta a risparmiare tempo sull'implementazione di design e funzionalità.



- 
1. Lato sviluppatore
 2. Performance



Sviluppo orientato al design

Cosa viene mostrato in questa finestra qui a destra?



Oeschinen Lake Campground

Kandersteg, Switzerland

★ 41



CALL



ROUTE



SHARE

Lake Oeschinen lies at the foot of the Blüemlisalp in the Bernese Alps. Situated 1,578 meters above sea level, it is one of the larger Alpine Lakes. A gondola ride from Kandersteg, followed by a half-hour walk through pastures and pine forest, leads you to the lake, which warms to 20 degrees Celsius in the summer. Activities enjoyed here include rowing, and riding the summer toboggan run.

Cosa c'è? - Parte 1

- Guardiamo righe e colonne
- C'è una griglia?
- Elementi che si sovrappongono?
- Abbiamo bisogno delle tabs?
- Abbiamo bisogno di margini o padding speciali?

Column



Oeschinen Lake Campground

Kandersteg, Switzerland

★ 41



CALL



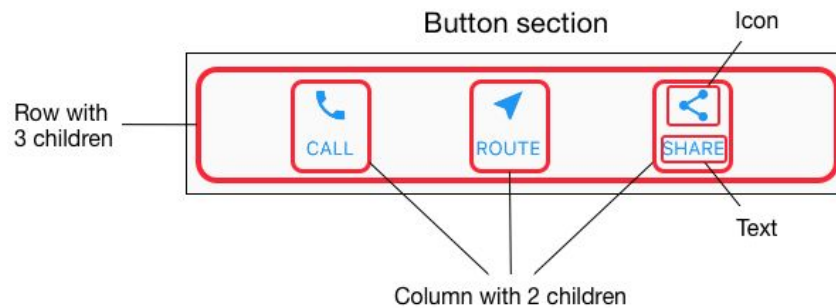
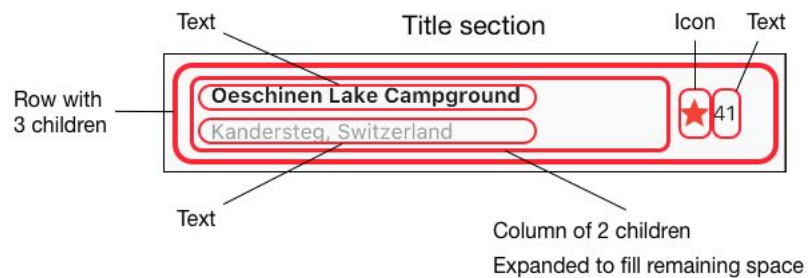
ROUTE



SHARE

Lake Oeschinen lies at the foot of the Blüemlisalp in the Bernese Alps. Situated 1,578 meters above sea level, it is one of the larger Alpine Lakes. A gondola ride from Kandersteg, followed by a half-hour walk through pastures and pine forest, leads you to the lake, which warms to 20 degrees Celsius in the summer. Activities enjoyed here include rowing, and riding the summer toboggan run.

Cosa c'è? - Parte 2



Analogie HTML/CSS in Flutter

```
<div class="greybox">
  Lorem ipsum
</div>
```

```
.greybox {
  background-color: #e0e0e0; /* grey 300 */
  width: 320px;
  height: 240px;
  font: 900 24px Georgia;
}
```

```
var container = new Container( // grey box
  child: new Text(
    "Lorem ipsum",
    style: new TextStyle(
      fontSize: 24.0
      fontWeight: FontWeight.w900,
      fontFamily: "Georgia",
    ),
  ),
  width: 320.0,
  height: 240.0,
  color: Colors.grey[300],
);
```

Strumenti di sviluppo

```
$ flutter doctor
```

Controlla il tuo ambiente di sviluppo e produce un report sul terminale


```
$ flutter upgrade
```

Aggiorna sia l'SDK che i tuoi pacchetti




pubspec.yaml

```
name: flutter_project
description: Un progetto fighissimo che usa firebase
dependencies:
  flutter:
    sdk: flutter
  firebase_auth: " ^0.2.5"
```



pubspec.yaml

```
name: flutter_project
description: Un progetto fighissimo che usa firebase
dependencies:
  flutter:
    sdk: flutter
  firebase_auth: ">=0.1.2 <0.2.6"
```



```
$ flutter packages get
```

Controlla il tuo ambiente di sviluppo e produce un report sul terminale

```
$ flutter packages upgrade
```

Aggiorna i pacchetti all'ultimissima versione disponibile



```
$ flutter format
```

Auto-indenta il codice in modo tale che sia più semplice da leggere

```
$ flutter analyze
```

Analizza il codice e prova a rilevare possibili errori



Hot Reload

Inserisce il codice sorgente aggiornato nella Dart VM.

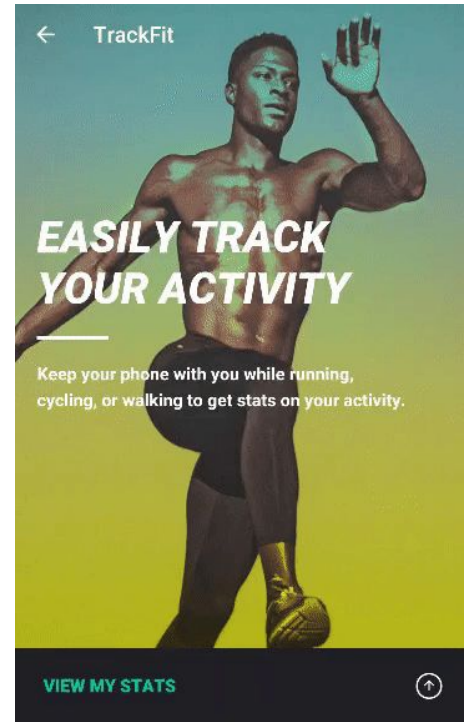
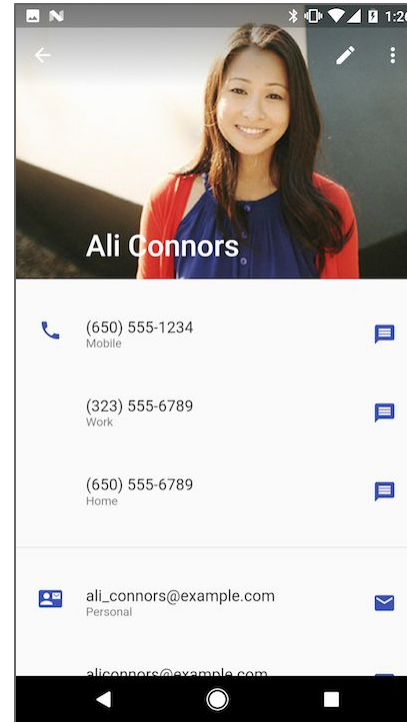
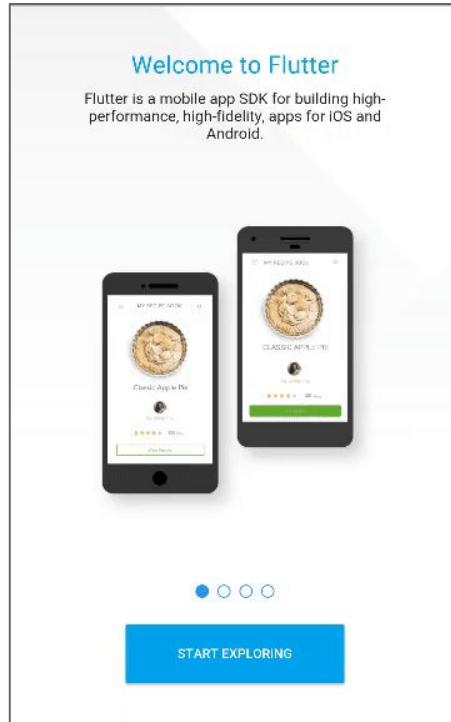
Stateful: Lo stato dell'app è mantenuto dopo un hot reload.

Puoi testare tantissime cose sulla tua applicazione senza dover ricompilare tutto, ad esempio il design dei componenti.

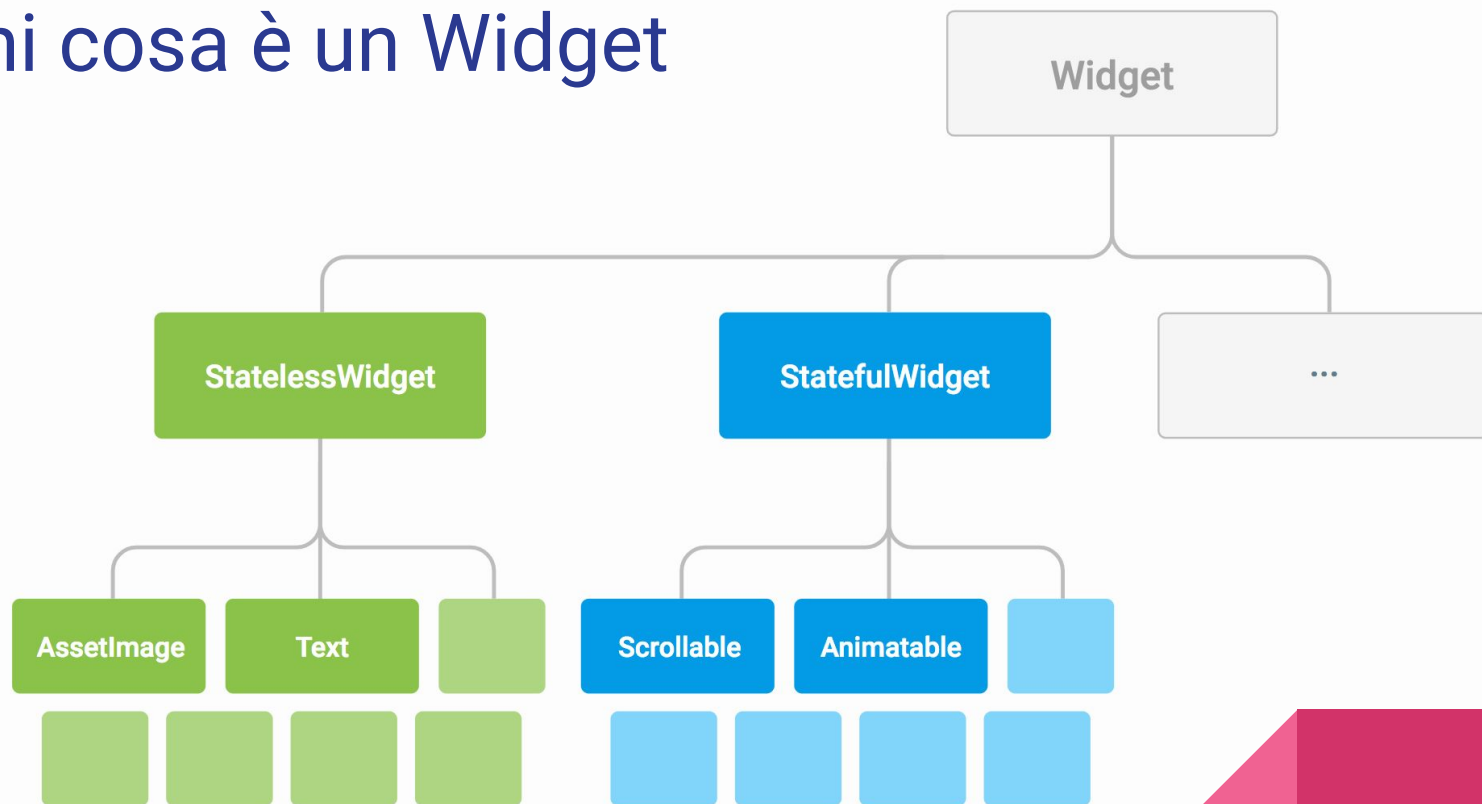


Il potere dei Widget

Alcuni esempi



Ogni cosa è un Widget



Addio, global layout system

```
new Center(  
  child: new Text('Centered Text', style: textStyle),  
)
```

Layout locali: ogni Widget ha il **proprio** layout. Non c'è bisogno di specificare al “widget genitore” che quel widget “figlio” deve essere centrato



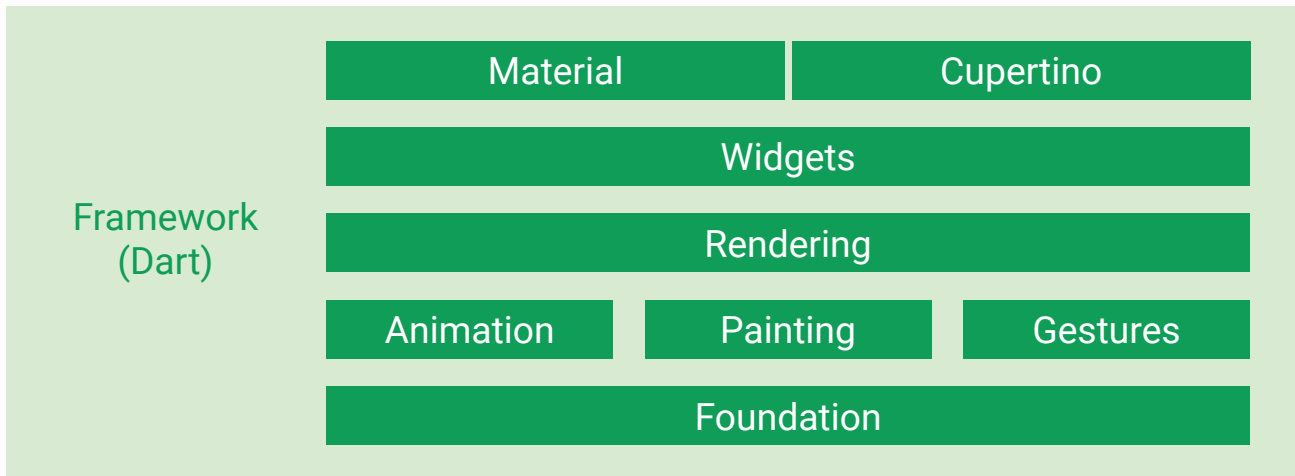
Personalizzare ed estendere i Widget

Il sistema dei Widget è stato progettato per essere facilmente personalizzabile.

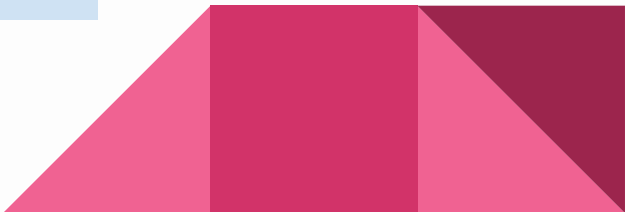
I Widget sono fatti da widget più piccoli che si possono riutilizzare e riassemblare per costruire dei widget personalizzati (“custom”).

```
class RaisedButton extends StatelessWidget {  
    ...  
}
```



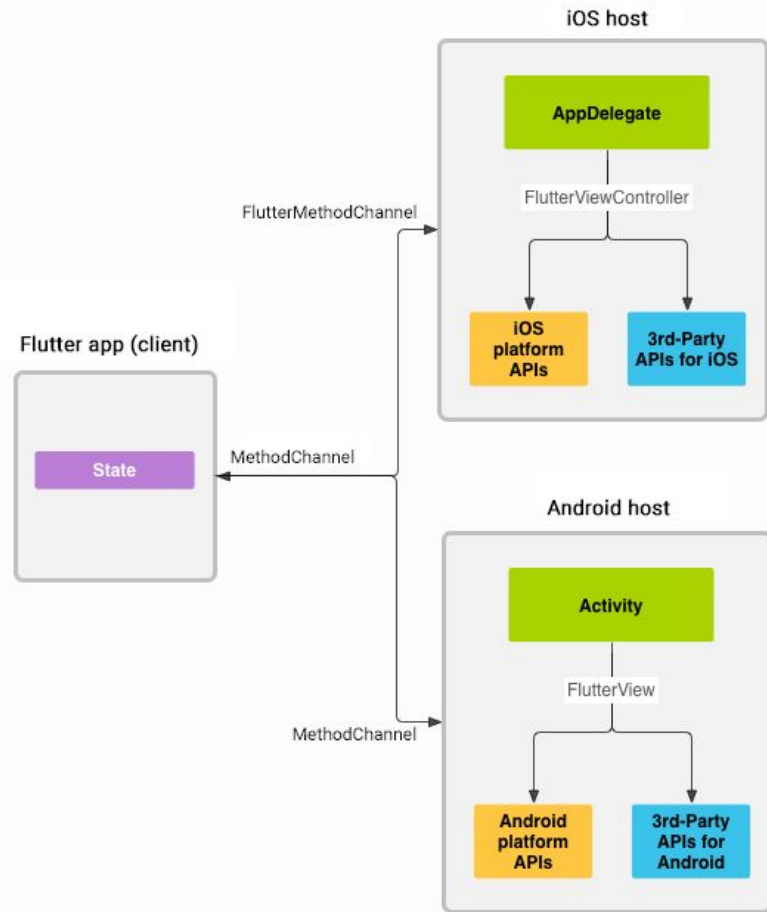


Ogni layer viene costruito sul layer precedente





Scrivere codice specifico per una
piattaforma anziché un'altra



Questo modello di architettura permette di ricevere “chiamate” e mandare indietro i risultati. (callback)

Esempio: mostrare il livello della batteria*

```
class MainActivity() : FlutterActivity() {  
    private val CHANNEL = "samples.flutter.io/battery"  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        GeneratedPluginRegistrant.registerWith(this)  
  
        MethodChannel(flutterView, CHANNEL).setMethodCallHandler { call, result ->  
            // TODO  
        }  
    }  
}
```

* Esempio scritto in Kotlin for Android



Lavorare con il “response”*


```
MethodChannel(flutterView, CHANNEL).setMethodCallHandler { call, result ->
    if (call.method == "getBatteryLevel") {
        val batteryLevel = getBatteryLevel()
        if (batteryLevel != -1) {
            result.success(batteryLevel)
        } else {
            result.error("UNAVAILABLE", "Livello batteria non disponibile.", null)
        }
    } else {
        result.notImplemented()
    }
}
```

* Esempio scritto in Kotlin perAndroid



Tutto ciò in Flutter si traduce come...

```
String _batteryLevel = 'Unknown battery level.';
Future<Null> _getBatteryLevel() async {
  String batteryLevel;
  try {
    final int result = await platform.invokeMethod('getBatteryLevel');
    batteryLevel = 'Battery level at $result % .';
  } on PlatformException catch (e) {
    batteryLevel = "Failed to get battery level: '${e.message}'.";
  }
  setState(() {
    _batteryLevel = batteryLevel;
  });
}
```

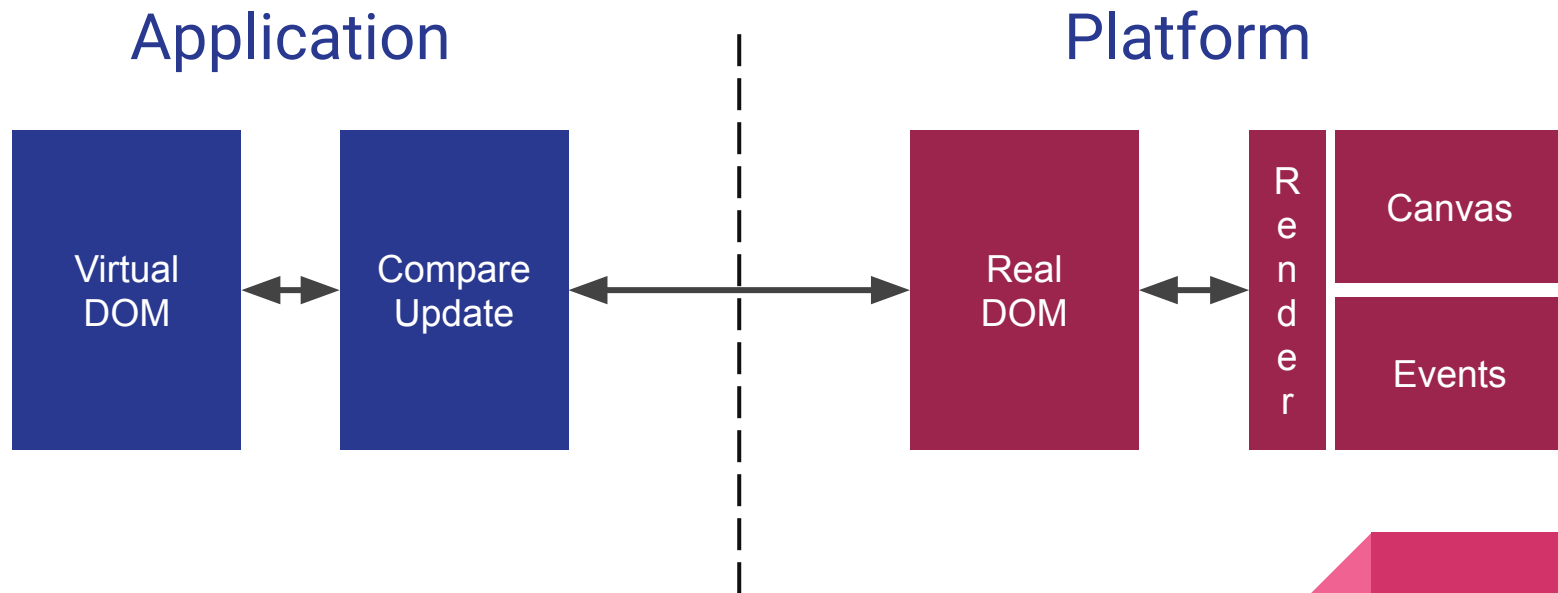




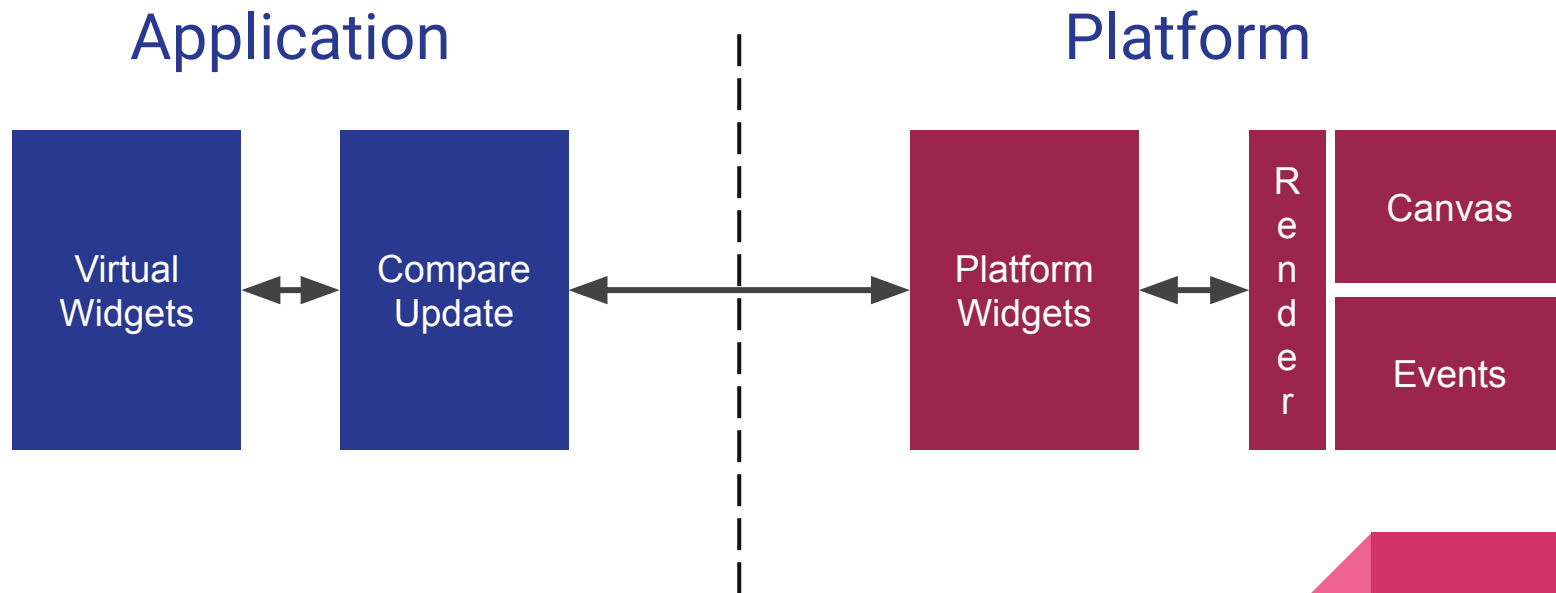
Ottimizzato per le performance

- Compila in codice nativo
- Nessun bisogno di usare widget proprietari
- Repainting

Framework sul web



Framework su dispositivi mobili

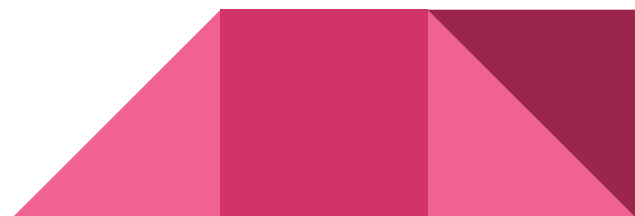
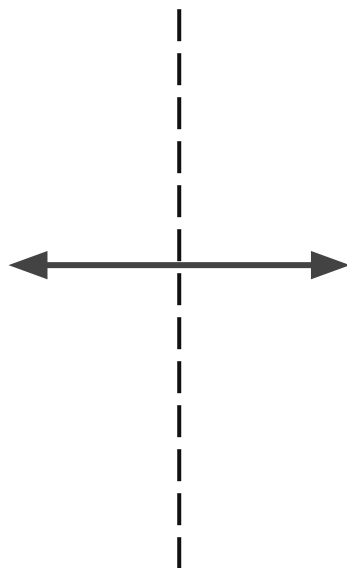
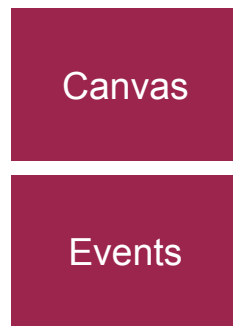


Flutter

Application



Platform





Supportato da Dart

- Sound type system:
- Tree Shaking
- Rich core libraries
- Multi-gen, garbage Collector


- Codice unico per Android e iOS
- Sviluppo rapido
- Strumenti

“Running at 60 fps, user interfaces created with Flutter perform far better than those created with other cross-platform development frameworks.”

code.tutsplus.com/tutorials/developing-an-android-app-with-flutter--cms-28270

“Coding with Dart and Flutter rekindled the joy I had when I started with mobile dev way back when ... No B.S.”

traversoft.com/blog/2017/08/08/conference-app-flutter

The top right corner of the slide features a decorative graphic consisting of several overlapping triangles in various shades of blue, ranging from dark navy to light sky blue.

"The UI is butter smooth (when building a release version), I have never seen such a smooth
Android app"

Pascal Welsch, Speaker at Droidcon Berlin

Link interessanti (in inglese)

Blog: Cosa c'è di rivoluzionario in Flutter di Wm Leler: goo.gl/bZcFR9

Video: Flutter's Rendering Pipeline di Adam Barth: youtu.be/UUfXWzp0-DU

Video: The Mahogany Staircase di Ian Hickson: youtu.be/dkyY9WCGMi0

Tutorial: Angela Ye - Bootcamp Udemy

<https://www.udemy.com/course/flutter-bootcamp-with-dart/>

Ovviamente: github.com/flutter - flutter.io - flutteritalia.dev





Grazie a tutti!

Damiano “dag7” Gualandri

Developer

dag7.it

[@dag7_](https://twitter.com/dag7_)

Nota importante

Queste slide sono state ispirate a quelle di Tim Messerschmidt in inglese:

https://docs.google.com/presentation/d/1B3p0kP6NV_XMOimRV09Ms75ymljU5gr6GGIX740m_DE/edit