

Ситник Роман Сергійович

*НТУ “Дніпровська політехніка”, кафедра програмного забезпечення
комп’ютерних систем, Україна*

Дослідження Flutter як сучасного рішення для розробки мобільних застосунків

Сьогодні на ринку домінує дві операційні системи для мобільних пристроїв - Android та iOS. Обидві вони мають різні інструменти, мови та підходи для розробки застосунків: для програмування під Android використовуються мови Kotlin та Java, для програмування під iOS - Swift та Objective-C.

З цього виходить, що для розробки функціонально однакового застосунку під обидві платформи потребується дві окремі команди розробників. Для невеликого бізнесу, який має обмежений бюджет, дві окремі команди для створення власного застосунку під мобільні пристрої може бути проблемою з фінансової точки зору.

Для розв’язання цієї задачі існують кросплатформні фреймворки для розробки мобільних застосунків для Android та iOS одночасно. Серед таких найбільш популярними є Cordova та React Native, які дозволяють писати мобільні застосунки за допомогою мови JavaScript. Але головними недоліками цих рішень є продуктивність роботи вихідних застосунків, погана підтримка фреймворків, та зростаюча складність розробки застосунку з його подальшим розвитком. Так, наприклад, компанії Airbnb та Udacity у 2018 році відмовилась від використання React Native, та повернулася до нативної розробки застосунків окремо для кожної платформи.

Cordova та Ionic використовують вбудований WebView (рушій рендерінгу з браузера), що дуже погано позначається на безпеці, швидкодії та продуктивності вихідного застосунку. React Native, фреймворк від компанії Facebook, використовує трохи інший підхід — фреймворк рендерить нативні UI-елементи інтерфейсу для кожної платформи, як справжній нативний застосунок для Android або iOS, без використання браузерного рендерінгу. Це

дозволяє зробити кросплатформенний застосунок швидшим, у порівнянні з WebView-фреймворками, та візуально подібним до застосунку, який був би розроблений спеціально для цієї платформи окремою командою програмістів.

Але під час роботи такого застосунку, завжди працює другий потік, який виконує JavaScript-код та використовує спеціальний міст для комунікації з потоком, який є відповідальним за рендеринг UI. Це негативно вказується на продуктивності та швидкодії застосунку.

Також такий підхід все одно доволі часто вимагає дублювання коду для обох платформ для налаштування вигляду та стилів UI-елементів у застосунку, в залежності від платформи, що створює додаткові проблеми при розробці.

Тому в якості свого рішення для кросплатформної розробки, у 2018 році компанія Google випустила стабільну версію свого нового фреймворку - Flutter.

Цей фреймворк сильно відрізняється від попередніх рішень на ринку: Flutter не використовує рушій WebView для рендерингу, і не використовує JavaScript та мостів для зв'язку з UI-потокком застосунку.

Замість цього, Flutter використовує свій власний рушій для рендерингу UI-інтерфейсу на основі Skia Graphics Engine, компактної графічної бібліотеки з відкритим сирцевим кодом, написаної на мові C++.

Таким чином, Flutter є відповідальним за рендеринг кожного пікселя на екрані користувача. Це вигідно відрізняє цей фреймворк від інших, і навіть дає змогу малювати UI зі швидкістю 60-120 кадрів на секунду, без використання додаткових ресурсів девайсу.

В якості основної мови програмування застосунків використовується мова Dart. Це мова з C-подібним синтаксисом, яку також розробила Google у 2011, як альтернатива для JavaScript. Сам Dart може компілюватися у нативний код та у JavaScript.

Flutter складається з 3 основних шарів:

1. Шар Flutter Framework, написаний мовою Dart, який відповідає за анімації, жести, малювання елементів, рендеринг та віджети. Це

верхній шар фреймворку, з яким і працює розробник під час створення застосунку.

2. Шар Engine, написаний мовами C/C++. Ця частина - це двигун фреймворку, який відповідає за рендеринг за допомогою Skia, працю Dart Runtime та інші низькорівневі задачі.
3. Шар Embedder, який є platform-specific, та відрізняється на різних платформах, на яких запускається Flutter. Цей шар відповідальний за усі нативні плагіни для Android та iOS, працю з залізом девайсів та все інше.

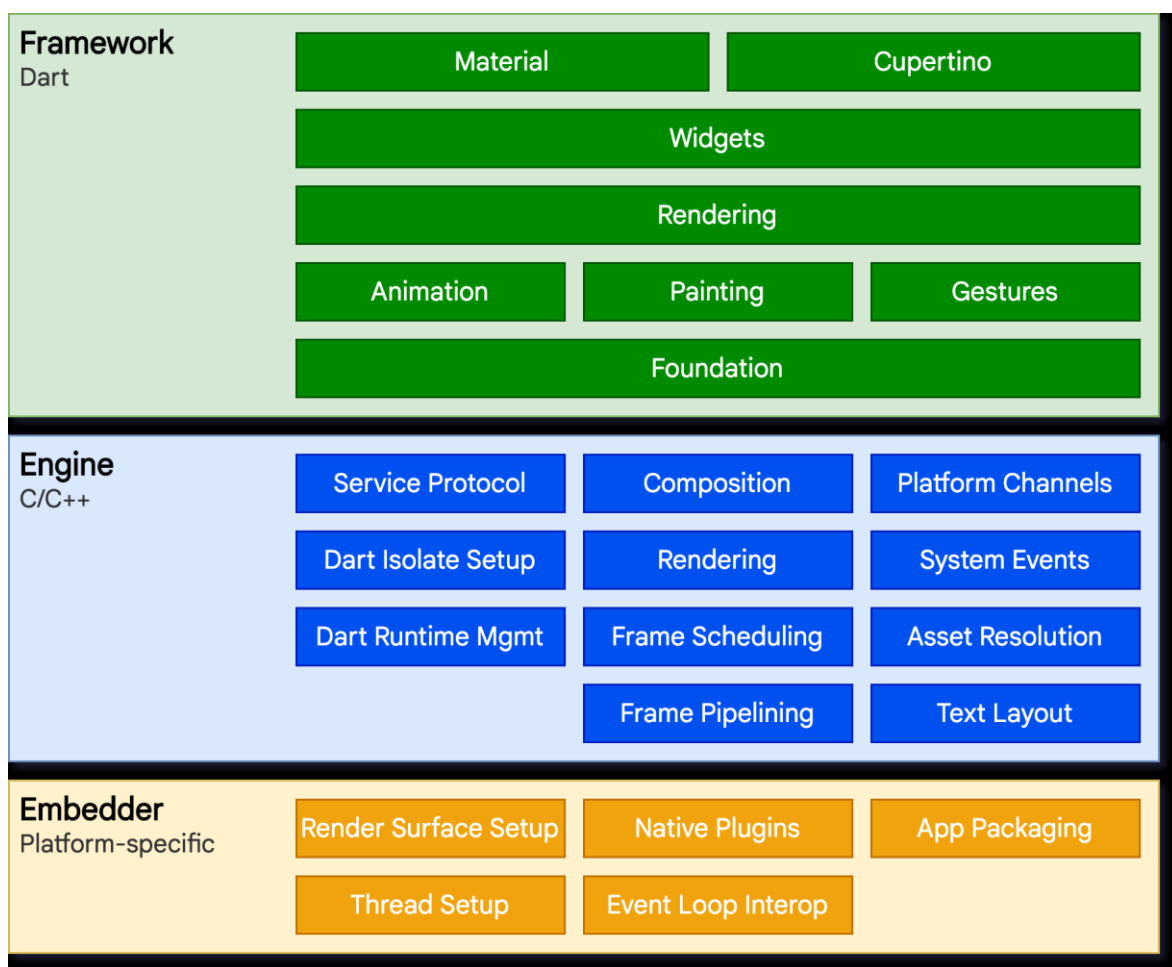


Рисунок 1. Архітектурні складові Flutter

Філософія Flutter - “все є віджет”. Побудова UI-екранів базується на компонуванні віджетів. Прикладом віджету може бути будь-який UI-елемент, такий як кнопка, список або зображення. Також віджетом є будь-який контейнер, який вміщує в себе інші віджети (наприклад, віджет Row, який

розташовує віджети всередині себе в один рядок). За допомогою віджетів також здійснюється збереження стану об'єктів, інших віджетів, їхня комунікація між собою (наприклад, для роутінгу між екранами або реалізації контролера з шаблону MVC).

В складі фреймворку Flutter присутні бібліотеки Material і Cupertino, які пропонують велику колекцію готових віджетів, виконаних у Material Design (для Android) та Cupertino (дизайн продуктів Apple, в тому числі iOS).

Висновки:

Фреймворк Flutter - це молодий, але потужний інструмент для розробки кросплатформених застосунків для Android та iOS одночасно. В майбутньому з його допомогою також можна буде будувати застосунки для Web та Desktop. Він вигідно відрізняється від своїх аналогів тим, що фреймворк сам є відповідальним за рендеринг кожного пікселя на екрані, а швидкість може досягати 60-120 кадрів на секунду без використання додаткових ресурсів девайсів.

Література:

1. Sunsetting React Native // medium.com [Електронний ресурс]. URL: <https://medium.com/airbnb-engineering/sunsetting-react-native-1868ba28e30a>
2. Flutter architectural overview // flutter.dev [Електронний ресурс]. URL: <https://flutter.dev/docs/resources/architectural-overview>
3. Eric Windmill, "Flutter in Action" (2019). Chapter 1, section 1.7 The Immediate Benefits of Flutter.