Comparison of multiplatform technologies for mobile application development

Michał Ankiersztajn

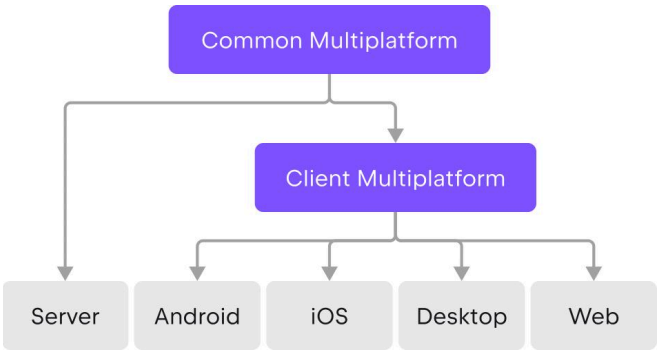
Wydział Elektryczny
Politechnika Warszawska

Student IT/EE Workshop 2025
Warsaw, Poland, April 24

Introduction

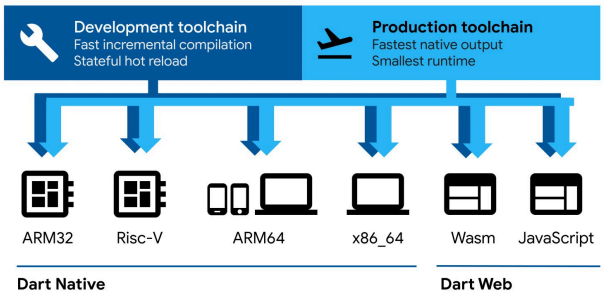
The mobile environment is highly fragmented due to multiple devices being physically different depending on the manufacturer, having different system versions depending on the vendor and each system having different functionality depending on the system version. Therefore, developing an application for a single mobile platform severely limits the application's potential userbase, business perspectives and usefulness. The dedicated application can be developed separately but is time-consuming and expensive [5]. This paper explores differences between multiplatform technologies namely KMP and Flutter due to lack of existing studies and the gain of popularity of the former [6].

KMP



KMP [1] is a technology built on top of Kotlin programming language, which is multiplatform by design and can be used to write shared business logic, data storage, networking and ViewModels. It also consists of Compose Multiplatform, a framework designed to render UI on multiple platforms, with Android and iOS being among the target platforms. When writing this paper, Compose Multiplatform for iOS is still in Beta.

Flutter



Flutter [3] - built on top of Dart [2] programming language, which is free and open source, currently supported by Google. Dart is optimised to create approachable and portable client applications on any platform.

Dart is optimised to create approachable and portable client applications on any platform. The standard approach is always to use Dart and Flutter together and to share the UI code. However, it is possible only to share the logic and keep the UI separate, yet no research or reliable sources were found.

Experimental app

An app [4] featuring a single screen loads and displays a list of cat photos from an API. Additionally, each photo can be clicked to show a zoomed view of the cat. Measurements and comparisons of the differences between the Flutter, KMP and Native approaches were made.

Application size [4]

Technology	size
Native	1.463 MB
KMP	1.463 MB
Flutter	6.828 MB

Android

Technology	size
Native	1.7 MB
KMP	24.8 MB
Flutter	17.9 MB

iOS

Startup performance [4]

Technology	min	median	max
Native	408.7 ms	413.1 ms	423.1 ms
KMP	403.6 ms	425.3 ms	466.4 ms
Flutter	600.5 ms	634.2 ms	649.8 ms

Android

Android benchmarks: Launch app 5 times on Pixel 4a device
iOS benchmarks: Launch app once on iPhone 12 Mini

Technology	duration
Native	1441 ms
KMP	1618 ms
Flutter	1608 ms

iOS

Lines of code [4]

Technology	lines of code
Native	2500 (1739 Android + 761 iOS)
KMP	2409
Flutter	2170

Summary

This analysis concludes that KMP is a better option for mobile application development because it can adjust to platform-specific needs, interoperability, app size and performance.

KMP's capabilities to migrate existing codebases from Android to Multiplatform make it a much more desirable technology than Flutter, which requires re-writing the application.

Whenever application size matters for iOS platform only sharing logic between the platforms is the desired choice.

Bibliography

[1] Kotlin Documentation, [Online]. Available: https://kotlinlang.org/
[2] Dart Documentation, [Online]. Available: https://dart.dev/
[3] Flutter Documentation, [Online]. Available: https://flutter.dev/multi-platform
[4] Jacob Ras, "flutter-vs-native-vs-kmp", [Online]. Available: https://github.com/jacobras/flutter-vs-native-vs-kmp
[5] Sanna Ottka, "Comparison of mobile application development tools for multi-platform industrial applications", March 25, 2015
[6] 2024 Stack Overflow Developer Survey, [Online]. Available: https://survey.stackoverflow.co/2024/technology/