Comparison of Native, Non-native, and Multiplatform Applications

Jesus Manuel Armenta Telles
February 3, 2024

I. Introduction

Mobile applications have become ubiquitous, offering users convenient access to various services and information. Developers have multiple options when it comes to building mobile apps, including native development, non-native development, and multiplatform development frameworks. Each approach has its pros and cons, influencing factors such as development time, performance, and user experience. This paper aims to explore and compare these approaches to provide insights for developers and stakeholders.

II. Native Applications

Native applications are developed using platform-specific programming languages and tools. For example, iOS apps are typically written in Swift or Objective-C, while Android apps are written in Java or Kotlin. These apps have direct access to the device's hardware and operating system features, offering high performance and seamless integration with the platform's ecosystem. Native apps provide the best user experience, as they follow platform-specific design guidelines and behavior patterns.

A. Advantages of Native Applications

- High performance: Native apps are optimized for the target platform, resulting in faster performance and smoother user experience.
- Access to platform features: Developers can leverage the full capabilities of the platform, including sensors, cameras, and notification systems.
- Platform-specific design: Native apps can provide a consistent user interface and user experience, following the design guidelines of the respective platforms (e.g., Material Design for Android, Human Interface Guidelines for iOS).

B. Disadvantages of Native Applications

- Development time: Building separate apps for different platforms requires more time and resources compared to cross-platform development.
- Maintenance overhead: Managing multiple codebases for different platforms can be challenging, especially when it comes to updates and bug fixes.
- Limited audience reach: Targeting a single platform may limit the potential audience of the app, especially if the target platform has a smaller market share.

III. Non-native Applications

Non-native applications, also known as hybrid or web-based apps, are developed using web technologies such as HTML, CSS, and JavaScript. These apps run inside a native container, which allows them to be distributed through app stores and access device features using plugins or APIs. Non-native apps offer a compromise between development effort and platform reach, making them a popular choice for cross-platform development.

A. Advantages of Non-native Applications

- Cross-platform compatibility: Non-native apps can run on multiple platforms with minimal modifications, allowing developers to reach a broader audience.
- Faster development: Since non-native apps use web technologies, developers with web development skills can quickly build and deploy apps across different platforms.
- Cost-effective: Building a single codebase for multiple platforms can significantly reduce development costs compared to native development.

B. Disadvantages of Non-native Applications

- Performance limitations: Non-native apps may not perform as well as native apps, especially for graphics-intensive or CPU-heavy tasks.
- Limited access to platform features: While non-native frameworks provide access to certain device features through plugins, they may not offer the same level of integration and performance as native APIs.
- User experience inconsistencies: Non-native apps may not fully adhere to platform-specific design guidelines, leading to inconsistencies in user interface and user experience.

IV. Multiplatform Applications

Multiplatform applications aim to combine the benefits of native and non-native approaches by allowing developers to write code once and deploy it across multiple platforms. These apps typically use a cross-platform development framework that translates code into platform-specific binaries or uses a common runtime environment.

A. *Advantages of Multiplatform Applications*

- Code reusability: Multiplatform frameworks enable developers to write code once and deploy it across multiple platforms, reducing development time and effort.
- Native-like performance: Some multiplatform frameworks use ahead-of-time compilation or native bridging to achieve performance levels close to native apps.
- Broad platform support: Multiplatform frameworks support targeting multiple platforms, including iOS, Android, and sometimes even web browsers or desktop platforms.

B. *Disadvantages of Multiplatform Applications*

- Framework limitations: Multiplatform frameworks may not support all platform features or APIs, requiring developers to use platform-specific code or plugins in some cases.
- Learning curve: Developers may need to learn new frameworks and development patterns, especially if they have prior experience with native or non-native development.
- Performance trade-offs: While multiplatform frameworks aim to provide native-like performance, they may still incur overhead compared to fully native apps, especially for complex applications.

V. Conclusion

In conclusion, native, non-native, and multiplatform applications offer different trade-offs in terms of development effort, performance, and user experience. Native apps provide the best performance and user experience but require separate development efforts for each platform. Non-native apps offer cross-platform compatibility and faster development but may suffer from performance limitations and inconsistent user experience. Multiplatform apps aim to strike a balance between development effort and platform reach, allowing developers to write code once and deploy it across multiple platforms. Ultimately, the choice between these approaches depends on factors such as project requirements, target audience, and developer expertise.

References

Offiah, M. C., & Borschbach, M. (2016). Non-native mobile porting and multi-platform benchmarking of blind source separation algorithms. *International Journal of Computing and Digital Systems*, *5*(01).

Poku-Marboah, O. (2021). Mobile application development methods: Comparing native and non-native applications.