Enterprise Software Development #5

Flutter

Simon Vegelahn, Timo Schröder

Dienstag, 04.11.2025

Outline

1. What is Flutter?
2. Flutter vs. Mobile
3. Flutter vs. Desktop
4. Flutter vs. Web
5. Counter Example
6. Refactored Counter Example
7. Exercise
8. Where to go from here?
9. References

What is Flutter?

- Flutter is an open-source UI software development kit created by Google.
- It is used to develop cross platform applications for Android, iOS, Linux, MacOS, Windows, Google Fuchsia, and the web from a single codebase.
- Written in the Dart programming language.
- First released in May 2017.

[1]

Flutter vs. Mobile

Category	**Flutter** [1]	**Android** [2]	**iOS** [3]	**React Native** [4]	**Kotlin Multiplatform** [5]
Cross-Platform	Yes	No	No	Yes	Yes
Programming Language	Dart	Kotlin	Swift	JavaScript/ TypeScript	Kotlin + Swift
UI Development	Declarative UI with Widgets	Declarative UI with Jetpack Compose	Declarative UI with SwiftUI	Declarative UI with Components	Native UI with Platform APIs
Performance	Near-Native	Native	Native	Near-Native	Native
Hot Reload	Yes	Yes	Limited	Yes	Depends on Platform
Ecosystem Maturity	Growing	Mature	Mature	Mature	Emerging

Flutter vs. Desktop

Category	**Flutter** [1]	**Electron JS** [6]	**Tauri** [7]	**Kotlin Multiplatform** [5]	**Native**
Cross-Platform	Yes	Yes	Yes	Yes	No
Programming Language	Dart	JavaScript/ TypeScript	Rust + Web Technologies	Kotlin + Platform Languages	C/C++, Swift, Rust, etc.
Performance	Near-Native	Moderate	High	Native	Native
Hot Reload	Yes	Limited	Limited	Depends on Platform	No
Ecosystem Maturity	Growing	Mature	Emerging	Emerging	Mature

Flutter vs. Web

Category	Flutter Web [1]	Web Frameworks
Programming Language	Dart	JavaScript/TypeScript
Performance	Moderate	High
Hot Reload	Yes	Yes
Ecosystem Maturity	Growing	Mature

Notice!

In Flutter Web, text not highlightable and extensions do not work!

Counter Example

- Widgets
- Stateful vs. Stateless

Refactored Counter Example

- ServiceLocator Pattern
- ValueNotifier
- watch_it

Exercise

1. Location Selection
2. Error Handling
3. Conditional UI
4. Input Validation *(optional)*
5. Data Scaling *(optional)*

Please refer to the README.md in the exercise repository for detailed instructions.

Where to go from here?

- Official Documentation: https://flutter.dev/docs
- Flutter YouTube Channel: https://www.youtube.com/flutterdev
- Dart Programming Language: https://dart.dev/guides
- Flutter Packages: https://pub.dev/

References

[1] "Flutter - Build apps for any screen." Accessed: Oct. 22, 2025. [Online]. Available: https://flutter.dev/

[2] "Android Mobile App Developer Tools - Android Developers." Accessed: Oct. 22, 2025. [Online]. Available: https://developer.android.com/

[3] "iOS - Apple Developer." Accessed: Oct. 22, 2025. [Online]. Available: https://developer.apple.com/ios/

[4] "React Native · Learn once, write anywhere." Accessed: Oct. 22, 2025. [Online]. Available: https://reactnative.dev/

[5] “Kotlin Multiplatform | Build Cross-Platform Apps.” Accessed: Oct. 22, 2025. [Online]. Available: https://kotlinlang.org/lp/mobile/

[6] “Build cross-platform desktop apps with JavaScript, HTML, and CSS | Electron.” Accessed: Nov. 04, 2025. [Online]. Available: https://www.electronjs.org/

[7] “Tauri 2.0 | Tauri.” Accessed: Nov. 04, 2025. [Online]. Available: https://tauri.app/