Native vs. Cross-Platform Development

Native

Advantages and Disadvantages

Advantages

- Better performance, quality, and reliability
  - Optimal hardware and operating system utilization
    - Very fast and responsive
    - Better UI/UX
    - Better security, stability, and reliability
      - Fewer bugs, and vulnerabilities
    - Direct access to hardware capabilities such as,
      - Touchscreens
      - Buttons
      - Graphics
      - Cameras
      - GPS/GNSS
      - Accelerometers
      - Magnetometers
      - Bluetooth, and so on.
- Offline mode
  - Can keep track of all data changes in the device's local database
  - Begins automatic synchronization of its local data with the web server once connection is restored

Disadvantages

- Require a high level of specialized knowledge and expertise for each OS
- Slower development
  - Requires a significant amount of time and resources to develop and maintain an app.
- Difficult to deploy or update native apps across all platforms simultaneously
- No guarantee that all platforms will have the same features

Source:

https://clutch.co/app-developers/resources/pros-cons-native-apps

https://arateg.com/blog/native-vs-cross-platform-app-development#:~:text=A%20cross%2Dplatform%20mobile%20app,carried%20out%20the%20same%20way.

https://decemberlabs.com/blog/native-apps/

http://www.optimusinfo.com/downloads/white-paper/native-hybrid-or-mobile-web-applications.pdf

Android

Programming Languages:

- **Kotlin**
    - **Advantages:**
        - Google's preferred language for Android
        - Safe, fewer crashes
        - Concise, more readable, and maintainable
            - Speeds up app development time
        - Interoperable, fully compatible with all Java-based frameworks
    - **Disadvantages:**
        - Slow compilation speed
        - Relatively young
            - Fewer resources, community support, and native libraries
- **Java**
    - **Advantages:**
        - Slightly faster
        - Long-term support
        - Wide range of tested and well-maintained custom APIs and third-party frameworks
        - App sizes are smaller
    - **Disadvantages:**
        - Requires way more lines of code than Kotlin
            - Slows down development time
            - Prone to more bugs
        - Experiences some issues with Android API due to inherent limitations

Source:

https://developer.android.com/kotlin/first

https://www.moveoapps.com/blog/java-vs-kotlin/

https://krify.co/advantages-and-disadvantages-of-kotlin/

https://faun.pub/7-reasons-to-choose-kotlin-over-java-f0d1b93c7b06

https://www.javaassignmenthelp.com/blog/java-vs-kotlin/#cons-of-java

iOS

Programming Languages:

- **Swift**
    - **Advantages:**
        - Faster
        - More
        - Concise, easier to read, write, and maintain
            - Speeds up app development time
        - Safer, less prone to crashes or bugs
        - Interoperability with Objective-C
    - **Disadvantages:**

        - Still in its early stages, constantly changing and evolving
            - Fewer resources, tools, and libraries compared to Objective-C
        - Lack of support of earlier iOS versions
        - Lack of backward compatibility

- **Objective-C**
    - **Advantages:**
        - Been around for over 30 years
            - Stable
            - Better overall resources, tools, and libraries (SDK)
        - Wide range of tested and well-maintained custom APIs and third-party frameworks
            - Better choice if C/C++ frameworks is required
        - Support for earlier version of iOS
    - **Disadvantages:**
        - Limited functionality
            - Slower
        - Lack of new updates, outdated
            - Security issues, vulnerable to malicious attack
        - Verbose
            - Difficult syntax

Source:

https://www.gamedeveloper.com/programming/swift-vs-objective-c-which-ios-language-to-choose

https://www.devteam.space/blog/how-are-objective-c-and-swift-different/

https://mlsdev.com/blog/swift-vs-objective-c

https://nix-united.com/blog/swift-vs-objective-c-which-is-better-for-your-next-mobile-app/

https://www.ideamotive.co/blog/picking-the-best-language-for-ios-app-development

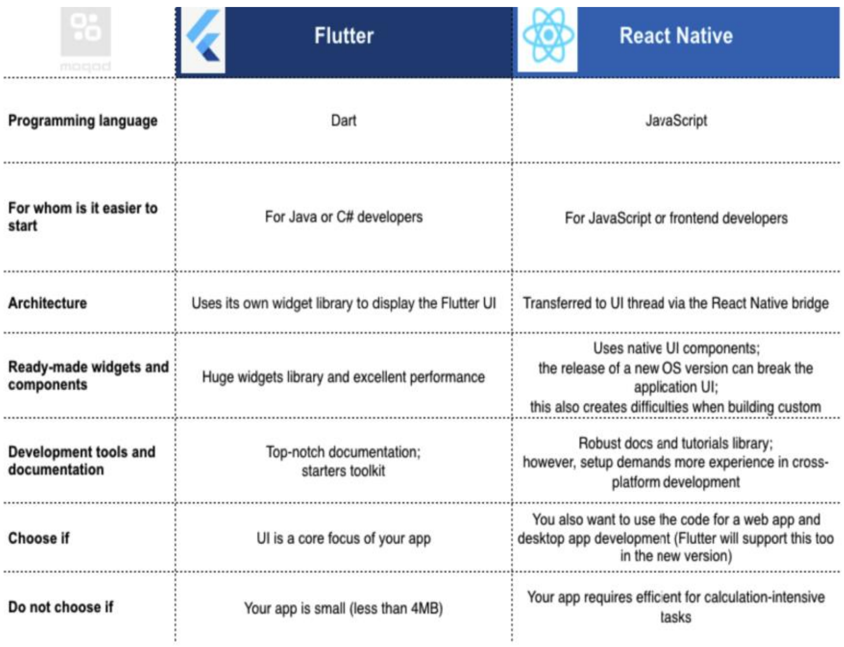
https://mdevelopers.com/blog/ios-app-development-which-technology-to-choose-

Cross Platform

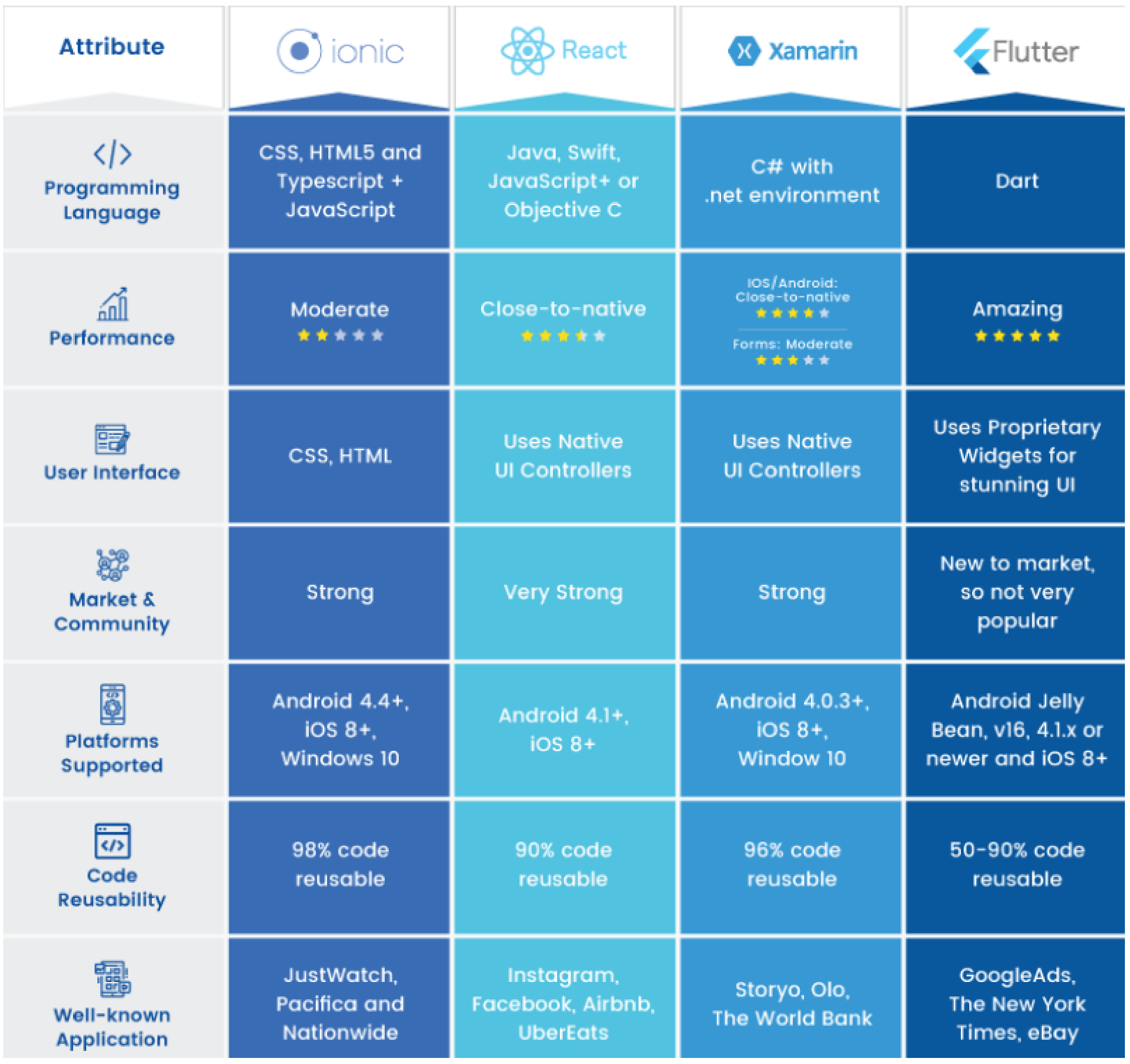
Frameworks:

- **Ionic:**

  - **Advantages:**
    - It’s flexible.
    - One codebase, multiple apps.
    - Tools with native compatibility. (Plugins that connect to GPS)
    - Frontend agnostic
  - **Disadvantages:**
    - If performance is a huge concern, going fully native is a better choice.
    - Ionic uses live reloading (refreshes the whole application to activate changes.)
- **Flutter:**
  - **Advantages:**
    - Flutter promotes portable GPU, which renders UI power, allowing it to work on the latest interfaces.
    - Eliminates additional processing steps that decrease performance making it noticeably faster
  - **Disadvantages:**
    - The size of Flutter applications may be problematic and lead the developer to choose a different language
    - Third-party libraries still have fewer resources than those available for other development tools.
    - To use Flutter, you must know Google’s Dart programming language
    - Flutter’s functionality may be better on Android than iOS
- **React Native:**
  - **Advantages:**
    - React Native is highly compatible with third-party plugins, such as Google Maps.
    - React Native environment eliminates the time taken in loading and delivers a smooth interface to the applications.
  - **Disadvantages:**
    - Very Hard to debug
    - Hard to determine user interfaces
    - Tougher to build a cross-platform team



	Flutter	React Native
Programming language	Dart	JavaScript
For whom is it easier to start	For Java or C# developers	For JavaScript or frontend developers
Architecture	Uses its own widget library to display the Flutter UI	Transferred to UI thread via the React Native bridge
Ready-made widgets and components	Huge widgets library and excellent performance	Uses native UI components; the release of a new OS version can break the application UI; this also creates difficulties when building custom
Development tools and documentation	Top-notch documentation; starters toolkit	Robust docs and tutorials library; however, setup demands more experience in cross-platform development
Choose if	UI is a core focus of your app	You also want to use the code for a web app and desktop app development (Flutter will support this too in the new version)
Do not choose if	Your app is small (less than 4MB)	Your app requires efficient for calculation-intensive tasks



Attribute	ionic	React	Xamarin	Flutter
Programming Language	CSS, HTML5 and Typescript + JavaScript	Java, Swift, JavaScript+ or Objective C	C# with .net environment	Dart
Performance	Moderate ★★☆☆☆	Close-to-native ★★★½☆	IOS/Android: Close-to-native ★★★★☆ Forms: Moderate ★★★☆☆	Amazing ★★★★★
User Interface	CSS, HTML	Uses Native UI Controllers	Uses Native UI Controllers	Uses Proprietary Widgets for stunning UI
Market & Community	Strong	Very Strong	Strong	New to market, so not very popular
Platforms Supported	Android 4.4+, iOS 8+, Windows 10	Android 4.1+, iOS 8+	Android 4.0.3+, iOS 8+, Window 10	Android Jelly Bean, v16, 4.1.x or newer and iOS 8+
Code Reusability	98% code reusable	90% code reusable	96% code reusable	50-90% code reusable
Well-known Application	JustWatch, Pacifica and Nationwide	Instagram, Facebook, Airbnb, UberEats	Storyo, Olo, The World Bank	GoogleAds, The New York Times, eBay

Advantages and Disadvantages

Advantages

- Develop once, deploy everywhere
- Faster and easier development
- Maintenance and updates can be done simultaneously without requiring individual changes on each platform

Disadvantages

- Difficult to support a "one-size-fits-all" approach without sacrificing performance, quality, and reliability.
    - UI/UX inconsistency
    - Incompatibility with the OS and/or hardware
    - Limited to using the lowest common subset of features available on all platforms to minimize bugs or unpredictable behaviour
        - Higher resource usage
- Have delayed access to the latest android or iOS updates and features, which makes it not the best choice for building apps focused on the latest platform-specific technologies.
- An app built with cross-platform technology consumes at least double CPU power than an identical app built with native technology.

Source:

https://surf.dev/advantages-and-disadvantages-of-cross-platform-mobile-development/#:~:text=Also%2C%20cross%2Dplatform%20frameworks%20usually,latest%20and%20platform%2Dspecific%20technologies.

https://appinventiv.com/blog/cross-platform-app-frameworks/#:~:text=Cross%2Dplatform%20app%20frameworks%20are,in%20the%20development%20of%20course.

https://stackoverflow.blog/2022/02/21/why-flutter-is-the-most-popular-cross-platform-mobile-sdk/#:~:text=Compared%20to%20other%20cross%2Dplatform,performance%20making%20it%20noticeably%20faster.

https://softjourn.com/insights/ionic-app-development-advantages-and-disadvantages

https://moqod-software.medium.com/flutter-vs-react-native-for-cross-platform-development-821b44138b4a#:~:text=Therefore%2C%20the%20developer%20always%20needs,close%20to%20native%20as%20possible.

https://www.thirdrocktechkno.com/blog/pros-and-cons-of-react-native-development-in-2021/#:~:text=Disadvantages%20of%20React%20Native%20App%20Development&text=Mobile%20app%20development%20done%20using,Native%20language%20of%20the%20platform.