Department of Electrconics and Computer Engineering
Institute of Engineering, Pulchowk Campus, TU, Nepal
BE Computer Engineering Program

Paper
Presentation

Analysis of Cross-Platform Mobile App Development Tools

K. Shah, H. Sinha and P. Mishra

Student:

Bishal Chaudhary, Bishal Katuwal
Department of Electronics and Computer Engineering,
IOE Pulchowk Campus, TU
075BCT027, 075BCT028

28th Feb., 2023

Outline

- Introduction
- Statement of the problem
- Approach
- Analysis of Cross-platform and Native applications
- Evaluating Cross-Platform Alternatives
- Results
- Conclusion

Introduction

- Basic Introduction
  - Title: Analysis of Cross-Platform Mobile App Development Tools
  - Authors: K. Shah, H. Sinha, and P. Mishra
  - Conference: 2019 IEEE 5th International Conference for Convergence in Technology (I2CT)
  - Location: Bombay, India
  - Year: 2019
  - DOI: 10.1109/I2CT45611.2019.9033872

Introduction

- Cross platform apps
  - The process of developing mobile apps that can run on multiple platforms using a single codebase.
  - Aims to reduce development time and costs by enabling developers to write code once and deploy it on multiple platforms.
  - Different Approaches:
    - Native apps
    - Web apps
    - Hybrid apps
    - Interpreted apps
    - Widget-based apps.

Statement of the Problem

- Development of mobile applications for multiple platforms is a complex and challenging task.
- Different platforms have different user interface guidelines and capabilities
- To address this, several cross-platform mobile app development tools have emerged to provide a unified development environment.
- With several cross-platform mobile app development tools available, it can be challenging to determine which tool is the best fit for a particular use case
- There are multiple factors to be considered
  - Choice of SDK
  - User Experience
  - Stability of Framework
  - Ease of Updating
  - Cost of Development
  - Time to Market the App.

Which cross-platform approach is the best-fit for given purpose?

Approach

- Delves into technological know-how from an implementation standpoint, however doesn't directly discuss the techniques to build an app.
- Seeks to perform side-by-side comparison of cross-platform tools.
- Takes an example from each of the cross-platform approaches and compares them to a native development environment.
  - Web
  - Hybrid
  - Interpreted
  - Widget-based apps

Analysis of Cross-platform and Native applications

- Advantages of Cross-platform application over Native application
  - Not restricted to a single platform
  - Usually use well-known programming languages and syntaxes
    - Easy and quick means for development.
    - Doesn't need rewriting and customization of code for separate platforms.
  - Large community-driven and open-source platform which is open-source in nature.
    - Continuous addition and revision of features.
    - Not tied down to a select group of developers.
  - Effective cost cutting as development is simpler and a single developer is sufficient
  - Subsequent updates and bug fixes can be rolled out for every platform affecting all platforms at once.

Analysis of Cross-platform and Native applications

- Advantages of Native application over Cross-platform application
  - For target platform, minimal lag and faster CPU render time with improved performance compared to its cross-platform counterparts.
  - Deeper integration with the device
  - Richer user experience(Rendering of high-end graphics is only effectively possible with native app development.)
  - Deployment to the app store of the respective platform is usually frowned upon in the case of cross-platform approaches.
  - Support for all devices is still practically infeasible with cross-platform solutions.
  - Complex apps are more prone to app freezing and crashes

Evaluating Cross-Platform Alternatives

- Web applications:
  - Similar to websites : source code exists on a remote server and their content is rendered onto the browser of the device.
  - Follow a three-tiered architecture
  - Main Languages : HTML, CSS, and JavaScript being the for development.
    - Angular is an open-source TypeScript-based front-end web app framework.
  - Features:
    - Uses components and directives to control the flow of logic and the view of the DOM
    - Follows a modular and hierarchical architecture
      - Root module provides the bootstrap mechanism and components defining views and services.
    - Two-way data binding
    - Dependency injection
    - Powerful template engine.
  -

Evaluating Cross-Platform Alternatives

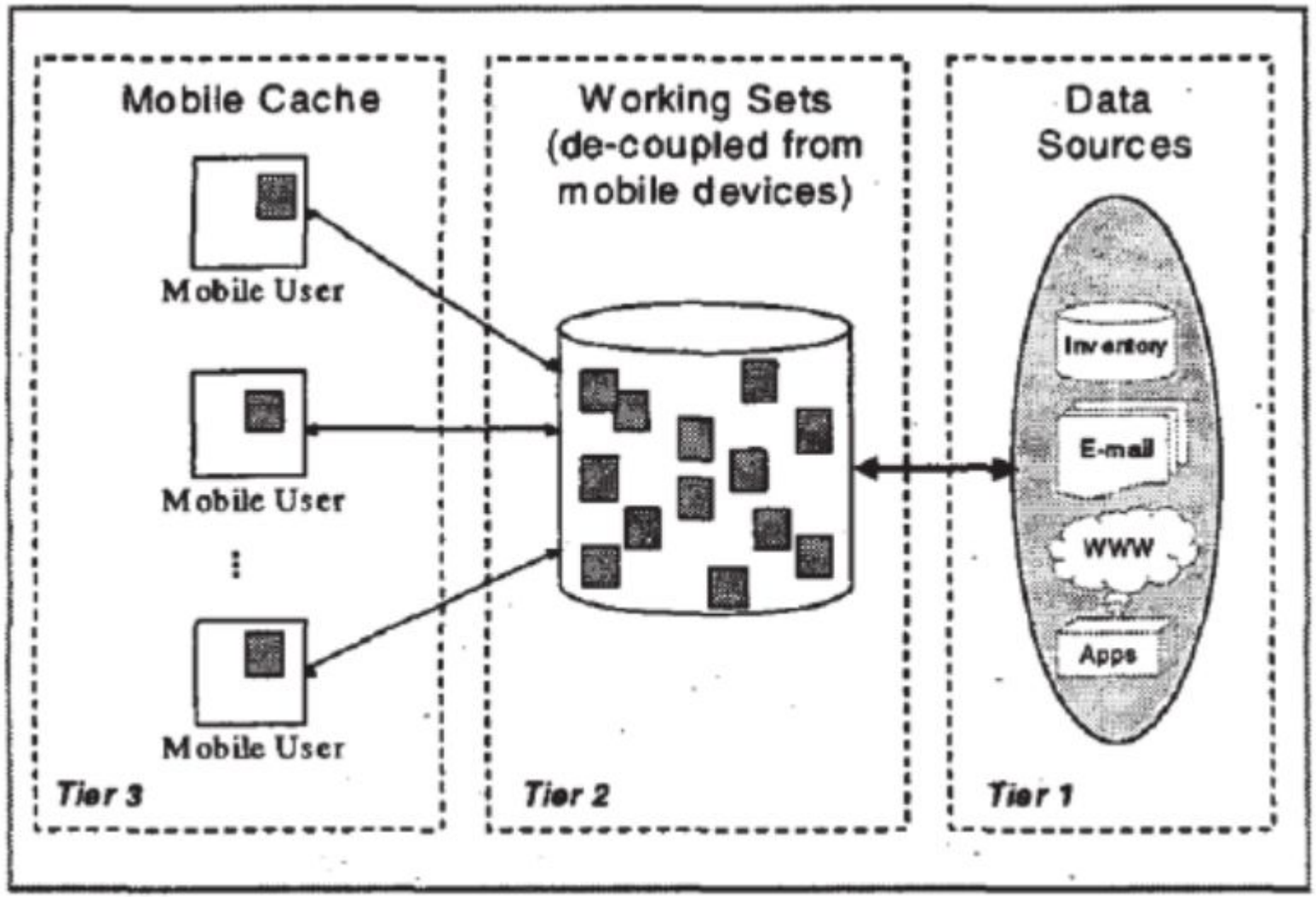
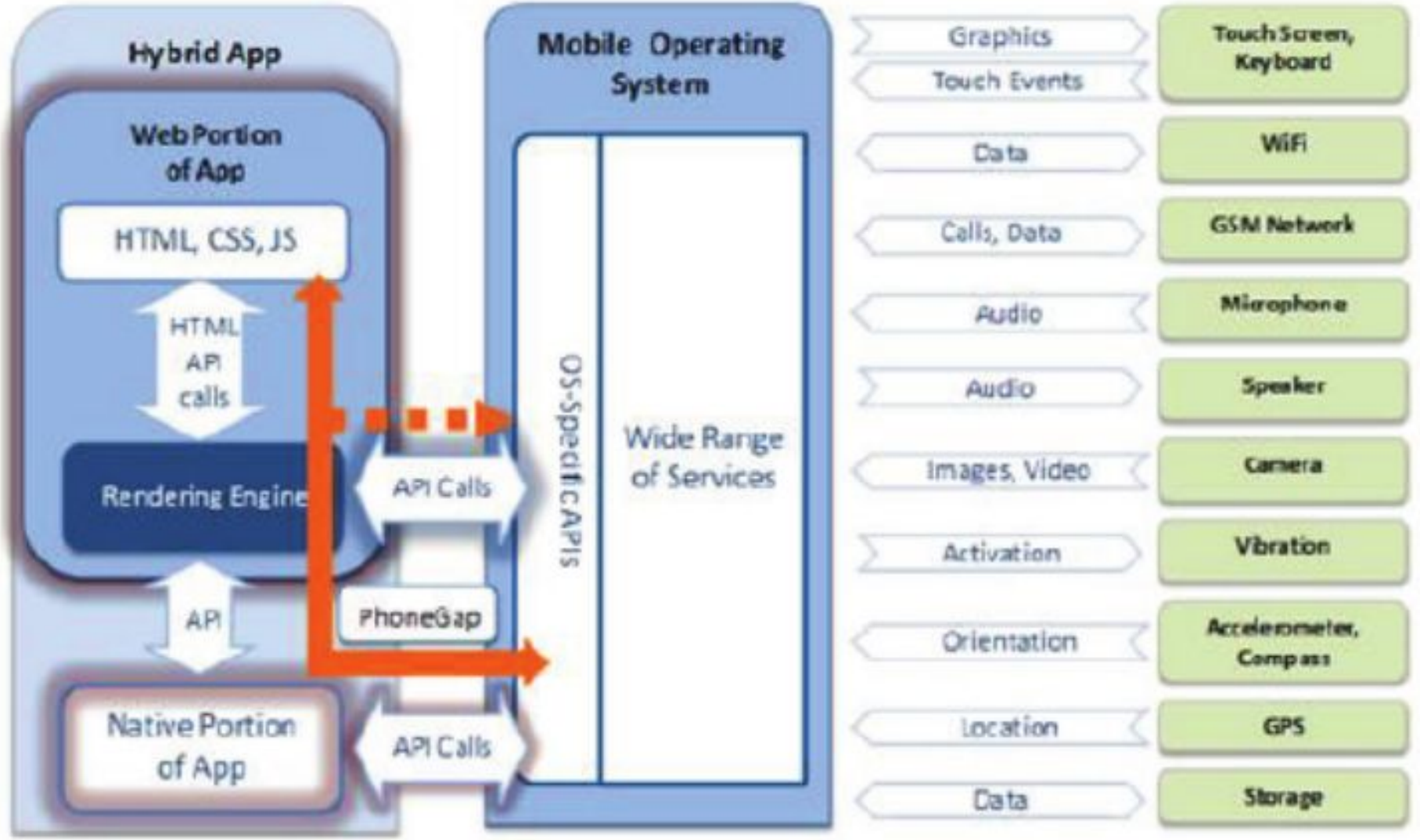


Fig. 1. 3-tiered architecture for web apps

Evaluating Cross-Platform Alternatives

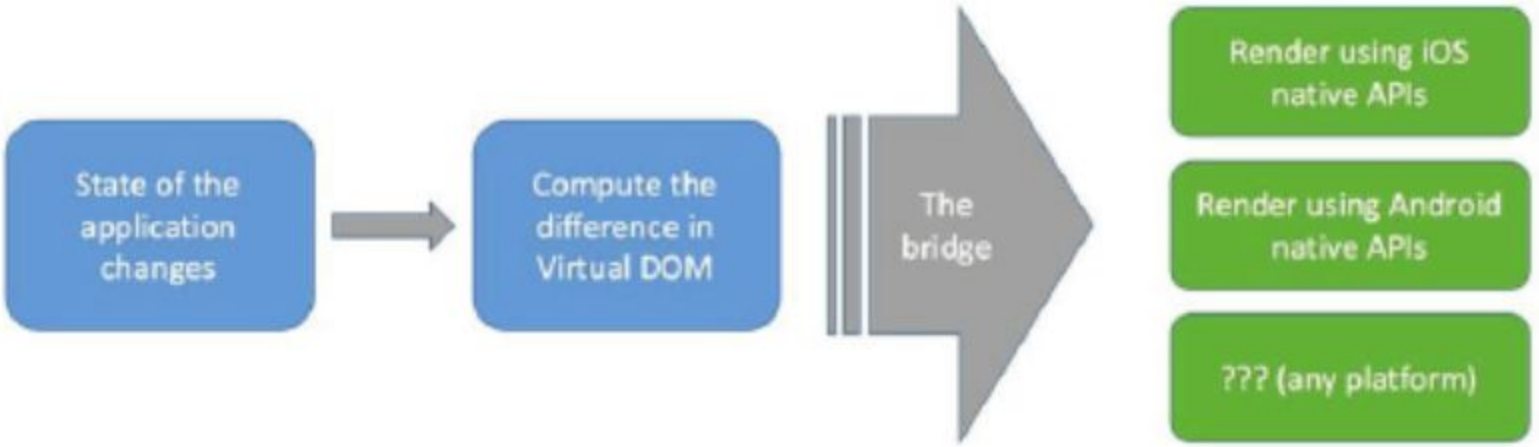
- Hybrid applications:
  - Combines web technologies such as HTML, CSS, and JavaScript with native code
  - Can be developed using frameworks such as PhoneGap
  - The tools which follow this approach usually follow a component-based structure which injects HTML code into native containers based on event-driven actions.
  - Closer integration with the device as native APIs calls are managed by the tool's plugins.
  - Compromise between native apps and web apps, most useful for creating interactive but simple user-friendly apps.
  - Apache Cordova(PhoneGap)
    - Open-source framework
    - Provides a unified development environment to write code once and deploy it on multiple platforms.
    - Runs as a web page using web technologies.
    - Is a wrapper that provides a way to run code inside a native wrapper, with a WebView and Plugins(closely tied to native device features such as data storage, cameras, and the accelerometer)
    - Compatible with the MVC pattern.
    - Bind device APIs to the application, providing a universal API for each feature.


Hybrid App
Web Portion of App
HTML, CSS, JS
HTML API calls
Rendering Engine
API
Native Portion of App
PhoneGap
API Calls
API Calls
Mobile Operating System
OS-Specific APIs
Wide Range of Services
Graphics
Touch Events
Data
Calls, Data
Audio
Audio
Images, Video
Activation
Orientation
Location
Data
Touch Screen, Keyboard
WiFi
GSM Network
Microphone
Speaker
Camera
Vibration
Accelerometer, Compass
GPS
Storage

Evaluating Cross-Platform Alternatives

- Interpreted applications:
  - Mobile apps that are created using JavaScript or V8 open-source engines.
  - Better performance and more interaction with the device hardware
  - Most common example being React Native
  - Features of React Native :
    - Built upon the ReactJS, a JavaScript framework
    - Uses Virtual DOM to store a copy of the document object model in memory.
    - Invokes platform specific APIs needed for rendering components of the native view controller.
    - Uses JavaScriptCore for iOS and V8 for Android as an interpreter

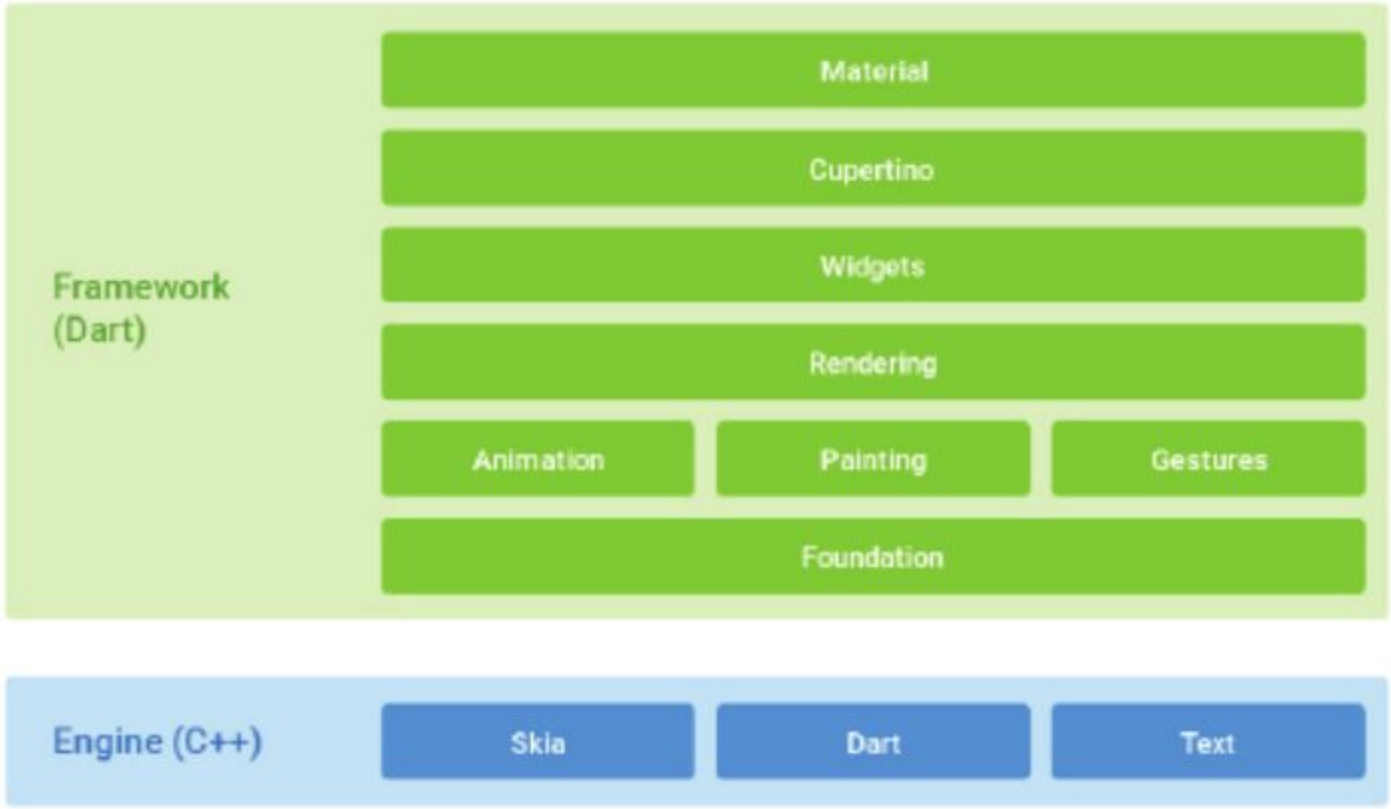
Evaluating Cross-Platform Alternatives



Evaluating Cross-Platform Alternatives

- Widget-based applications:
  - Widgets: Anything that can define a structural element
    - Stateless widget : Immutable and cannot be overwritten
    - Stateful widget : Mutable and can change its composition dynamically while responding to the user's actions.
  - Flutter: An open source UI framework and mobile (SDK) created by Google
  - Written in Dart
  - Treats every component as a widget; following a unified object model.
  - Creates its own components rather than depending on the device's native components by converting the application code (e.g. JavaScript or Dart) into native Code.

Evaluating Cross-Platform Alternatives



Findings

- Bases for Comparison:
  - UI/UX
  - Potential Users
  - Development Cost
  - App Security
  - Ease of Update
  - Implementation Complexity
  - Access to Native APIs
  - Development Environment
  - License
  - Language
  - Publishing to Marketplace

Findings

Decision Parameters	Native	Web apps	Hybrid apps	Interpreted apps	Widget Based (Flutter)
UI / UX	Excellent	Moderate	Moderate	Fairly Good	Very Good
Potential Users	Limited	Maximum	Large	Large	Limited
Develop ment Cost	High	Low	Low	Moderate	Moderate
App Security	Very High	Very Low	Low	Moderate	High
Ease of Update	Low	High	Varying	Varying	Moderate
Implementation Comp lexity	High	Low	Moderate	Low to Moderate	Moderate
Access to Native APIs	Yes	Yes, but only with HTML5	Yes, through plugins	Yes	Yes
Preferred Development environments	Android Studio, XCode, Momentics	Visual Studio Code, PhpStorm, WebStorm	Eclipse Hybrid Mobile Tools (THyM), Intel XDK,	Nuclide using Atom IDE, Appcelerator Studio	Android Studio, IntelliJ
License	Android: Open-source, iOS: Closed-source	Open-source	Generally open-source	Open-source	Open-source
Language	Java/Kotlin, Swift/Objective-C, etc.	HTML, CSS, JavaScript, TypeScript, etc.	HTML, CSS, JavaScript, Node.js etc.	JavaScript, JSX, UX Markup, etc.	Dart
Publishing to Marketplace	Yes	No	Yes	Yes	Yes

Conclusion

- Native apps
  - Better choice when it comes to performance-related parameters like access to native device APIs, ease of update, rendering UI and providing the better user experience.
- Web apps
  - Mobile-friendly versions of websites which are more interactive than simple websites
  - Handy approach useful for when the app itself is not too demanding of resources or complexity but requires some element of user interaction.
  - Better choice if ease of implementation and the development time is the developer's primary concern
  - Ccannot get published in the app marketplace.
- Hybrid apps
  - Publishable
  - Can access native device APIs
  - Not as easy to develop.

Conclusion

- Interpreted apps
  - Have better security
  - Have better access to device hardware as they are native apps generated from cross-platform tools.
  - Often use web development technologies and thus easie ti develop as compared to purely native apps
- Widget based app
  - Almost complete access to the underlying device hardware
  - Dart is easy to implement close to although not exactly as simple as web apps
  - Better than interpreted apps as it generates its own widgets compared to interpreted app tools like React Native that are dependent on the device original equipment manufacturer (OEM) widgets
  - Developed by Google that ensures high-security standards and maintenance
-

References

Shah, Kewal & Sinha, Harsh & Mishra, Payal. (2019). Analysis of Cross-Platform Mobile App Development Tools. 1-7. 10.1109/I2CT45611.2019.9033872.

Queries/Comments/Feedback?

Thank YOU!!