

Mobile Application Development

Mobile Platforms

Lecture Plan

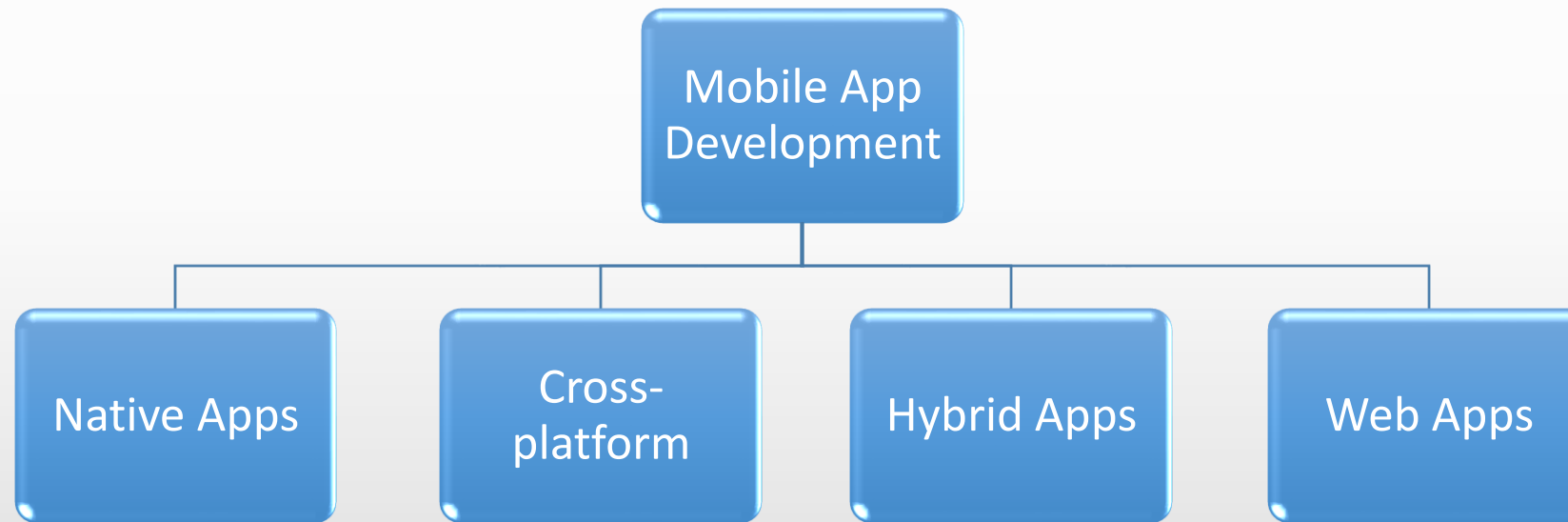
- Mobile Application Development fundamentals
- **Mobile Platforms**
- Introduction to Android Operating System
- Main Components of Android Application
- Android Interface Design Concepts
- Data handling in Mobile App Development
- Sensors and Media Handling in Android Applications
- Kotlin Language to develop Android Mobile Apps
- Android Application Testing and security aspects

Learning Outcomes of the Lecture

At the end of this Lecture students will be able to:

- Comprehend native mobile operating systems.
- Describe cross-platform mobile development.
- Describe Hybrid mobile development.

Mobile Application Development



Native Mobile Application

- A native mobile app is an application developed using platform-specific development tools.
- These apps are developed individually for each of the three popular mobile operating systems.





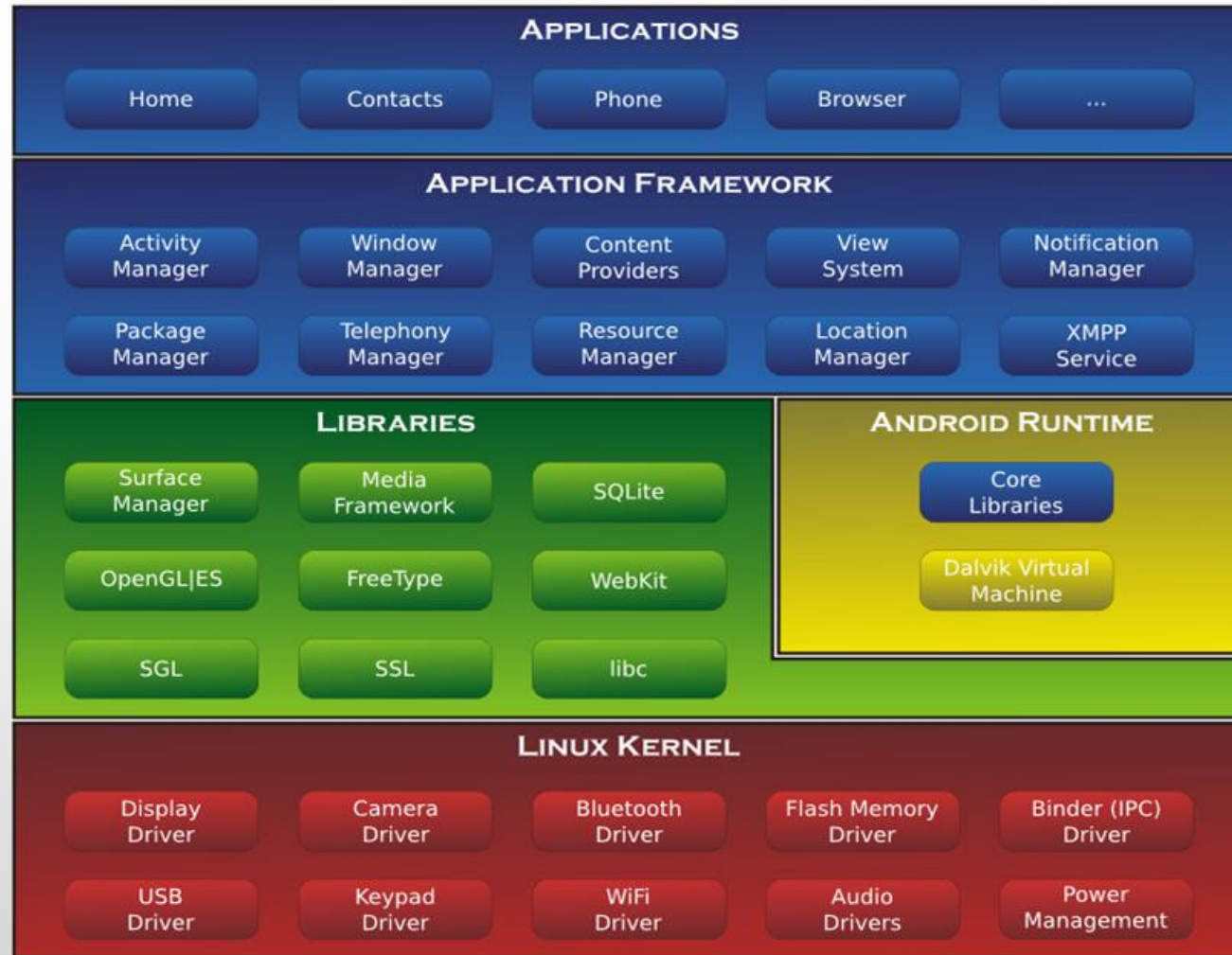
Android

- Android is a mobile operating system developed by Google, based on a modified version of the Linux kernel and other open source software. It is primarily designed for touchscreen mobile devices such as smartphones and tablets.
- Android is the most popular mobile operating system at present.
- Founders of android were Rich Miner, Nick Sears, Chris White, and Andy Rubin.





Android Architecture



Android versions





Android Devices

Devices using android operating system

Smartphones

- Samsung
- Sony
- HTC
- Google
- LG
- Lenovo
- Oppo
- Huawei





Android Devices

Tablets

- Samsung Galaxy Tab
- Asus ZenPad
- Huawei MediaPad
- Lenovo Yoga Tab
- Amazon Fire HD
- Sony Xperia Z4 Tablet
- Nvidia Shield Tablet K1





Android Devices

TV

- Sony Bravia Smart TV
- Sharp Smart TV
- Philips Smart TV



Smartwatch

- Ticwatch
- LG Watch Style
- Misfit Vapor
- Asus ZenWatch
- Fossil Q Venture





Android Devices

Development Environments

- Android Studio
- Eclipse
- Apache Cordova
- App Inventor for Android
- C++ Builder
- Blue J
- FlashDevelop
- Titanium



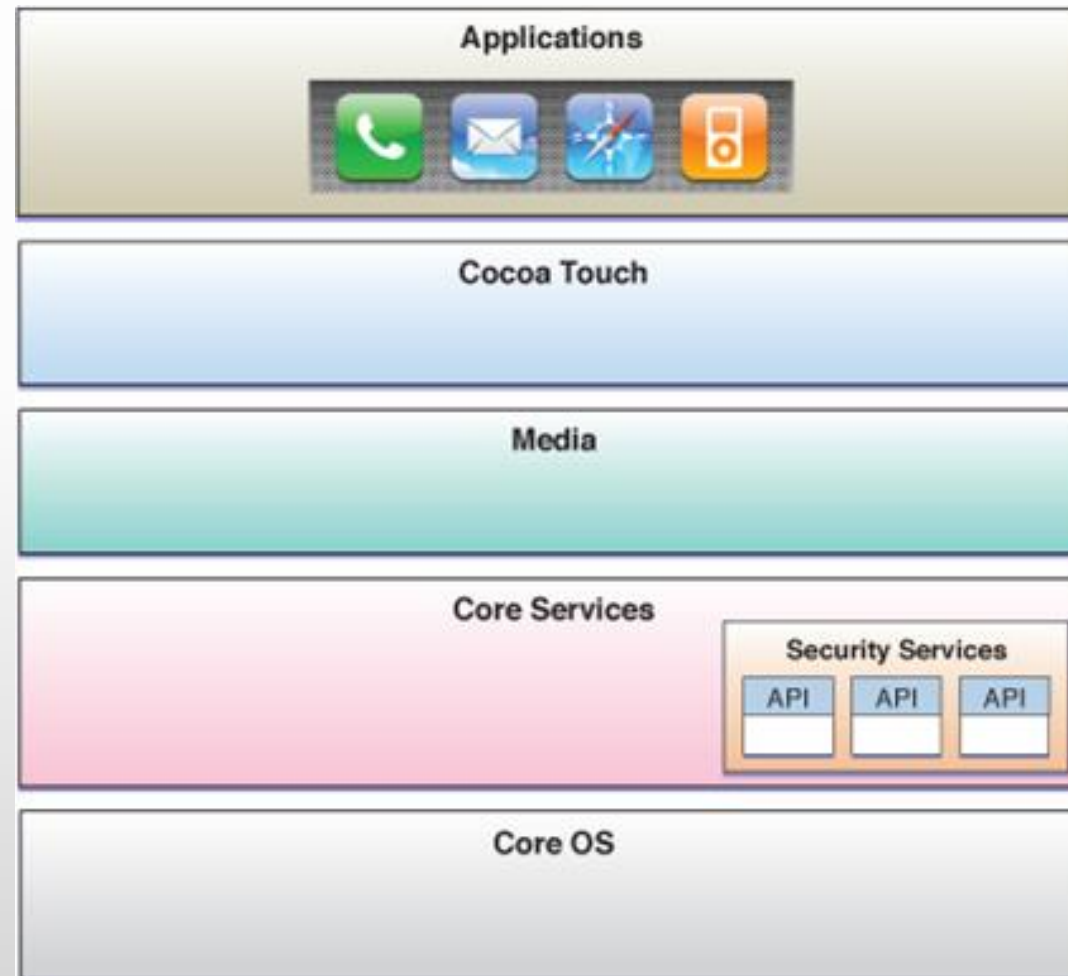


- iOS is a mobile operating system created and developed by Apple Inc.
- It is exclusively designed for Apple hardware.
- It is the second most popular mobile operating system globally after Android.
- Founders of iOS/Apple were Steve Jobs, Steve Wozniak, and Ronald Wayne





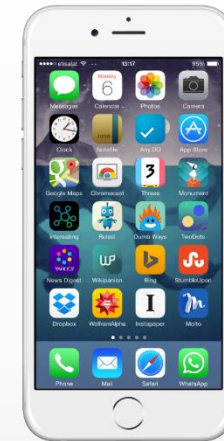
iOS Architecture



iOS Devices

Devices using iOS operating system

- iPhone
- iPod Touch
- iPad
- iPad Mini
- iPad Pro
- Apple TV
- Apple Watch





Development Environments

- Xcode
- AppCode
- Apache Cordova





Windows Mobile

- Windows Mobile is a discontinued family of mobile operating systems developed by Microsoft.
- Its origin dated back to Windows CE in 1996, though Windows Mobile itself first appeared in 2000 as PocketPC 2000.
- It was renamed "Windows Mobile" in 2003, at which point it came in several versions and was aimed at business and enterprise consumers



Windows Mobile

Devices using windows mobile operating system

- Dopod 515
- Krome Intellect iQ200
- Mitac Mio 8390 and 8860
- Motorola MPx200
- O2 Xphone
- Orange SPV E200 and e100
- QTEK 7070 and 8080
- Sagem myS-7



Windows Mobile

Development Environments

- Visual Studio
- Apache Cordova



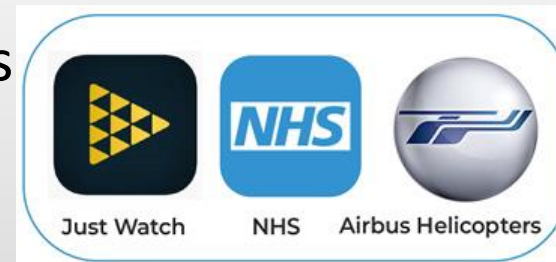
Hybrid App Development

- Less time for development.
- Allows for code sharing.
- Blend web elements with mobile ones.
- Create codebase using standard web technologies (HTML, CSS, JavaScript)

Tools:

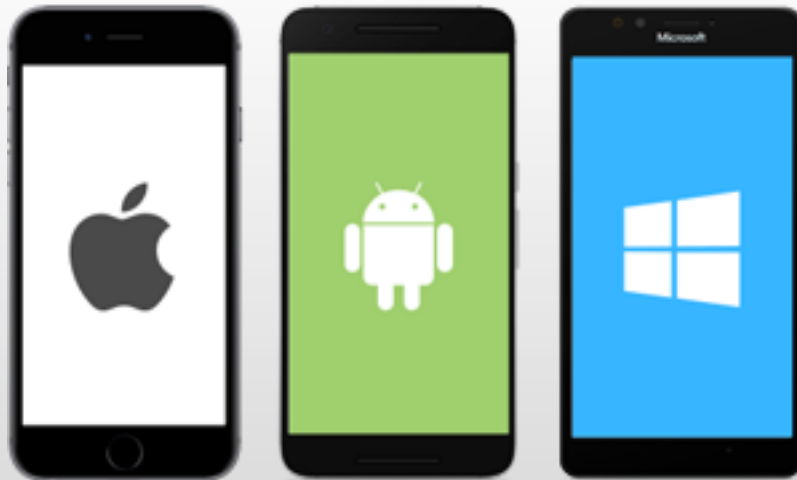


Examples



Cross-platform mobile application development

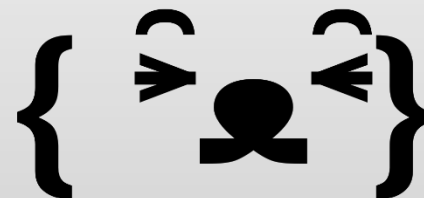
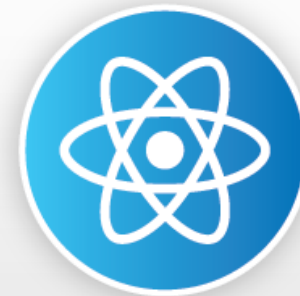
Cross-platform mobile application development refers to the development of mobile apps that can be used on multiple mobile platforms.



Cross-platform mobile application development

Development Environments

- Apache Cordova
- PhoneGap
- Xamarin
- Ionic
- Framework 7
- React Native
- Jasonette



Cross-platform mobile application development

Advantages

- Codes can be reused
- Controls Cost
- Quicker development time
- Easier Implementation
- Sameness and Uniformity

Cross-platform mobile application development

Disadvantages

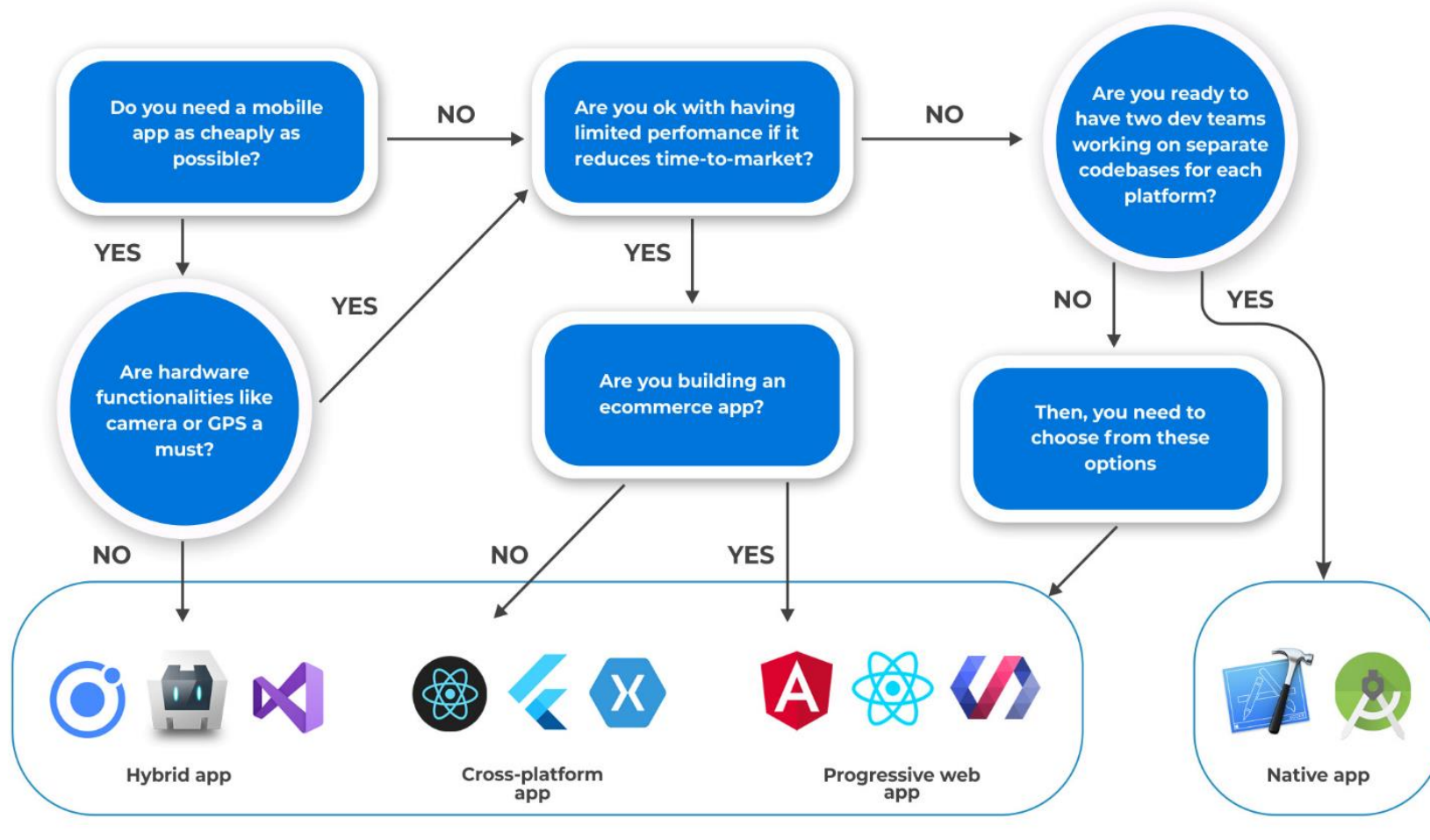
- Loss of Flexibility
- Problems in platform Integration
- Diversity in user Interaction
- Poor user experience
- Difficulty in satisfying all users

App Type	Native	Hybrid	Cross-platform
Tools	<ul style="list-style-type: none">• XCode• AppCode• Android Studio	<ul style="list-style-type: none">• Ionic• Apache Cordova• Visual Studio	<ul style="list-style-type: none">• React Native• Xamarin• Flutter
Rendering Engine	Native	Browser	Native
Libraries	Not much dependency on open-source libraries and platforms	Highly dependent on different libraries and frameworks	Highly dependent on different libraries and frameworks
Debugging	Native debugging tools	Native + web development debugging tools	Depends on the framework
Codebase	Separate codebase – one per platform	Single codebase with potential platform-specific abilities	Single codebase with potential platform-specific abilities

App Type	Native	Hybrid	Cross-platform
Pros	<ul style="list-style-type: none">• Full access to device's/ OS's features• Powerful performance• Native UI (updating along with the OS)• Efficient App Running• High-quality functionality and UX• Access to all native APIs and the platform-specific functionality	<ul style="list-style-type: none">• Lower development cost• Different OS support• Code reuse• Cost effective development• Big customization capabilities	<ul style="list-style-type: none">• Different OS support• UI performance is almost as fast as native• Code reuse• Cost-effective development
Cons	<ul style="list-style-type: none">• No multi-platform support• High dev cost if different OS support is needed• No code reuse	<ul style="list-style-type: none">• Slower performance• Limited access to OS features• No interaction with other native apps	<ul style="list-style-type: none">• *Slower performance• Limited access to OS features• Poor interaction with other native apps

Choose a Development approach for your Mobile App

CHOOSE A DEV APPROACH FOR YOUR MOBILE APP



Thank You