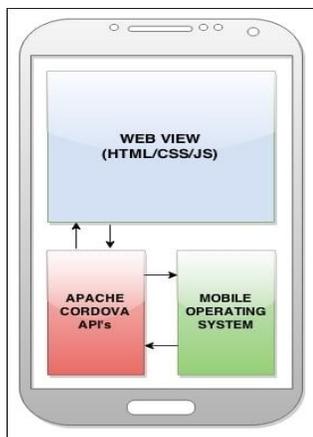


UNIT V

Mobile Hybrid Architecture

It is a software application that combines elements of both native apps and web applications. Every Mobile OS/Platform has APIs to develop apps. API consists of a component named Web View. It is typically a browser that runs inside the scope of a mobile application. This browser runs the HTML, CSS, and JS codes. We build a web page using the preceding technologies and then execute it inside your app.

Apache Cordova is a platform for building native mobile applications using HTML, CSS and JavaScript. Cordova stitches the web application and the native application together. It provides a set of APIs written in JavaScript to interact with the native features of the device access camera, take a picture, and send it in an e-mail

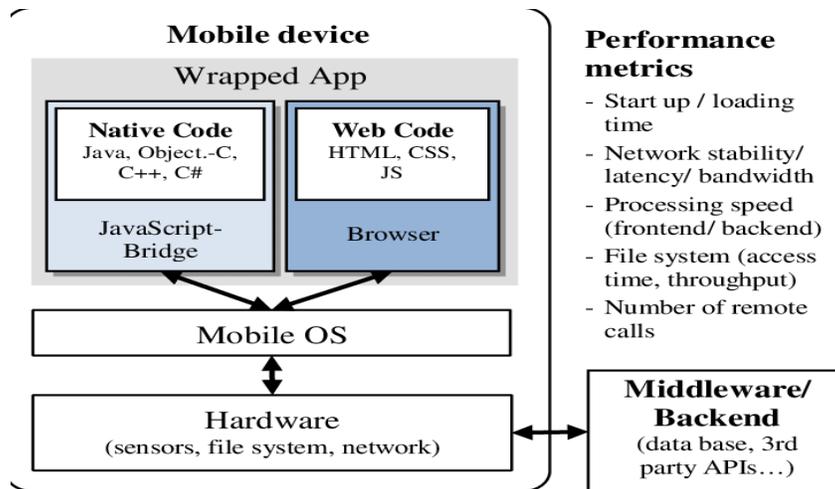


Hybrid apps work similar to Web apps but like native apps, are downloaded to the device. Similar to Web apps, hybrid apps are typically written in HTML5, CSS and JavaScript. Hybrid apps run code inside a container. The device's browser engine is used to render HTML and JavaScript and native APIs to access device-specific hardware.

Mobile App Development involves the following four layers:

1. Building **Mobile UI** using layouts, adding workflow with the help of events supporting navigation, transitions, and gestures. Mobile specific widgets allow you to take advantage of the device features
2. **Data integration** via REST APIs, variables, and data binding

3. The **Native Support** enhances the app functionality by enabling native look and feel, access to device-specific features, offline data support, and responsiveness
4. **Installer** helps build platform-specific installers for deployment



Features

- Ability to function whether or not the device is connected.
- Integration with the mobile device's file system
- Integration with Web-based services
- An embedded browser to improve access to dynamic online content

Limitations

- Appearance of an application may vary from platform to platform
- Need to test the application on a range of devices to ensure proper operation
- User experience (UX) may fall

Feature	Native	Web-only	Hybrid
Device Access	Full	Limited	Full (with plugins)
Performance	High	Medium to High	Medium to High
Development Language	Platform Specific	HTML, CSS, Javascript	HTML, CSS, Javascript
Cross-Platform Support	No	Yes	Yes
User Experience	High	Medium to High	Medium to High
Code Reuse	No	Yes	Yes

IONIC Framework

Open-source UI framework for building high quality **mobile apps**, desktop apps, and progressive **web apps** using web technologies such as HTML, CSS, and JavaScript.

It was originally built on top of AngularJS but now supports all the top front-end frameworks. Ionic is used to create hybrid apps, which is essentially a web app that's been wrapped up in a native shell. The device installs the app container locally (like a native app), but the app uses an embedded browser to connect to any mobile platform capabilities. The embedded browser and any plugins are invisible to the user. The embedded browser separates the source code from the device and acts as a translator between the behavior outlined in the code and the specifics of the device .

The pros of using Ionic are:

- Platform-specific tuning
- Backend agnostic
- Extensive built-in UI options and elements
- Supported by top mobile and web platforms
- Easy to pick up for those experienced in Sass, CSS, or HTML

The cons are:

- Slow performance for Ionic Native apps (especially for graphics-heavy apps)
- Plugin dependency, apps may fully break if a plugin is missing
- No hot-reloading support, must restart the app to apply changes

Features

- Cross-platform framework
- It uses the AngularJS MVC architecture to develop app
- It allows user to create a simple, clean, and functional mobile application
- The development based on the standardized web technologies **HTML, CSS, JavaScript**
- It is free and open-source project,
- Ionic Command Line Interface
- Ionic components expand the JavaScript capabilities
- Cordova plugin provides the API to use the native device functions with JavaScript

Cordova

Ionic apps become hybrid apps with the inclusion of mobile app development framework **Apache Cordova**. These programs create the embedded browser layer that allows Ionic's web content to work in a native app. Once wrapped up with the embedded browser, your app functions as a native app and can be deployed to app stores on iOS, Android, Electron, or to the web as a PWA. Cordova acts as a translator to allow your hybrid app to access the same resources as a native app.

How Angular JS powers IONIC

Ionic combines HTML5+CSS, AngularJS and use Cordova to work on native device functions. Angular is the underlying framework that powers Ionic. It is responsible for the component API that is the building block of Ionic.

A Single Application for Multiple Platforms

Ionic has its library of optimized JS, HTML and CSS tools as Ionic platform is developed upon AngularJS along with Sass. A tedious coding and developing processes have been discarded through Ionic. Push a single code, and your web application will perform over different platforms.

Reusable App Logic:

Fusing AngularJS and Ionic enables the developers to separate user interfaces, system logic, and data models. This means that the developers in parallel can reuse the logic and also customize the UI for different gadgets and different platforms

Performance of Ionic Apps

Ionic has optimized UI components with Crosswalk in the CLI to ensure the smooth performance across various platforms

Create Custom Components

Creating innovative icons and splash screens is a challenging task, especially for hybrid app development. But Ionic required to build only two files – .ai, .psd or .png within the resources directory for creating custom components.

Angular's Three-way Data-Binding

Angular allows three-way data binding which means it enables simultaneous updating of the data model, UI as well as backend data. It means that in Angular, it is easy to take the users' inputs, process the information, save it on the server, and hence keeping the user interface synched with the backend is much easier.

Testability

AngularJS eases out the testing process to a certain extent due to the Dependency Injection (DI) feature. This feature lets the developers fetch the pieces of the code from anywhere and duplicates the code for testing purposes.

AngularJS framework is known for delivering robust performance and captivating front-end for the apps. On top of that, when it is clubbed with Ionic, it unlocks new possibilities for developing captivating apps. Ionic has attractive themes and the availability of several components that help in the creation of an outstanding UI.

SCSS

SCSS (Syntactically Cascading Style Sheet) is the superset of CSS. Due to its advanced features it is often termed as Sassy CSS. SCSS have file extension of .scss. It provides the CSS friendly syntax to closing the gap between Sass and CSS. It is fully CSS compatible. You can rename a CSS file as .scss extension and it will also work.

SCSS helps you to customize Bootstrap. This language is a preprocessor one, and we need to compile or interrupt it into the CSS language.

SASS (Syntactically awesome style sheets) is the older version, one of the major differences in writing these two is the **use of semicolons and brackets**