



# SNS COLLEGE OF TECHNOLOGY



(An Autonomous Institution)

Re-accredited by NAAC with A+ grade, Accredited by NBA(CSE, IT, ECE, EEE & Mechanical)  
Approved by AICTE, New Delhi, Recognized by UGC, Affiliated to Anna University, Chennai



## Department of MCA

### Mobile Hybrid Architecture

**COURSE**

**19CAT701  
Mobile  
Application  
Development**

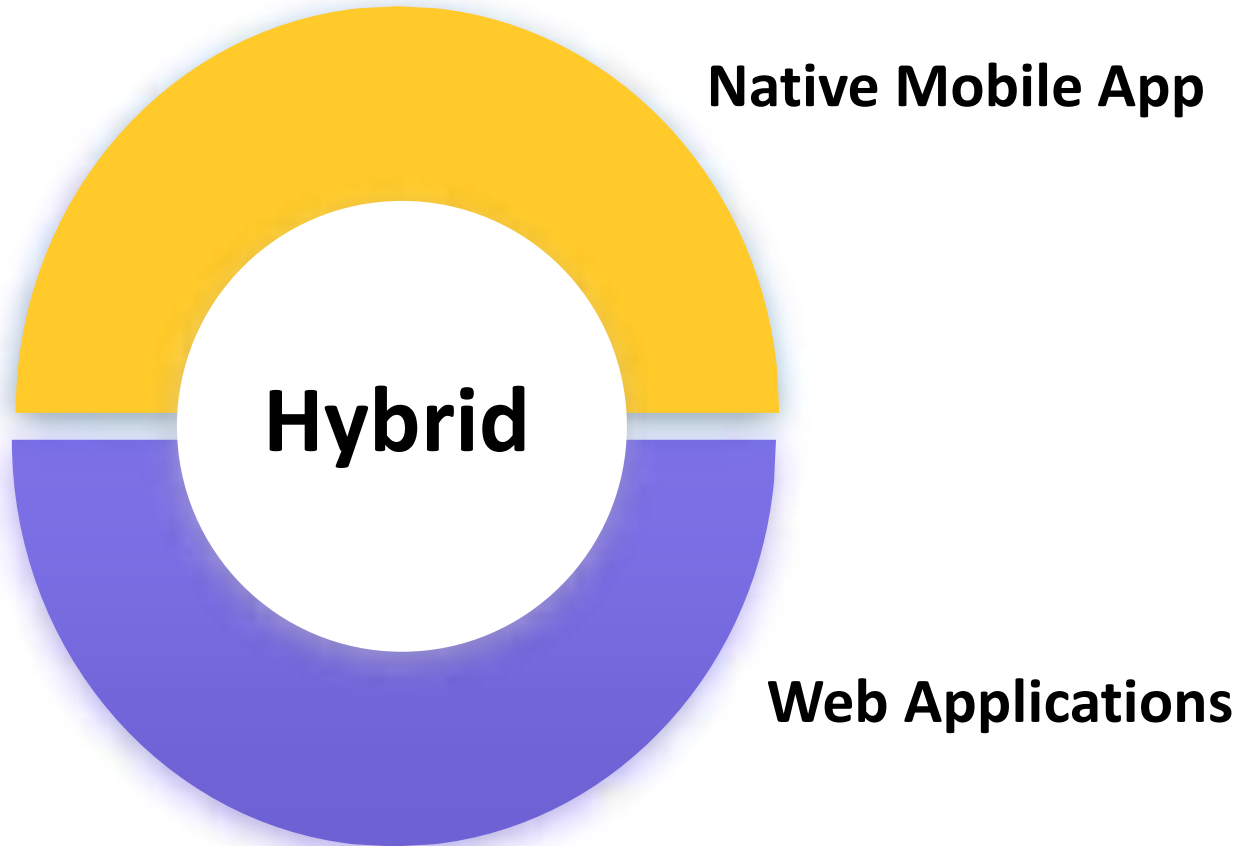
**UNIT V**

**IONIC Hybrid  
Framework**

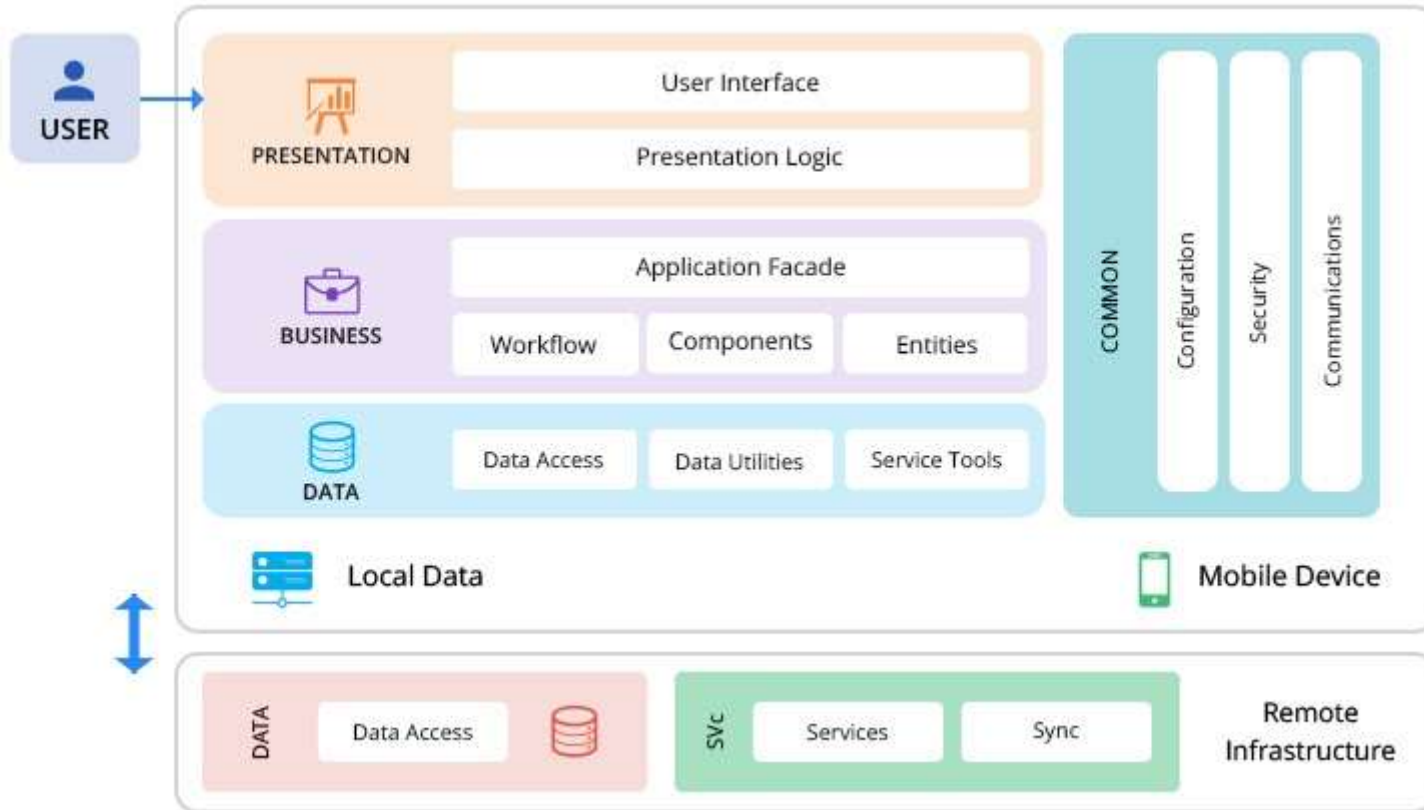
**Semester**

**III Semester /  
II MCA**

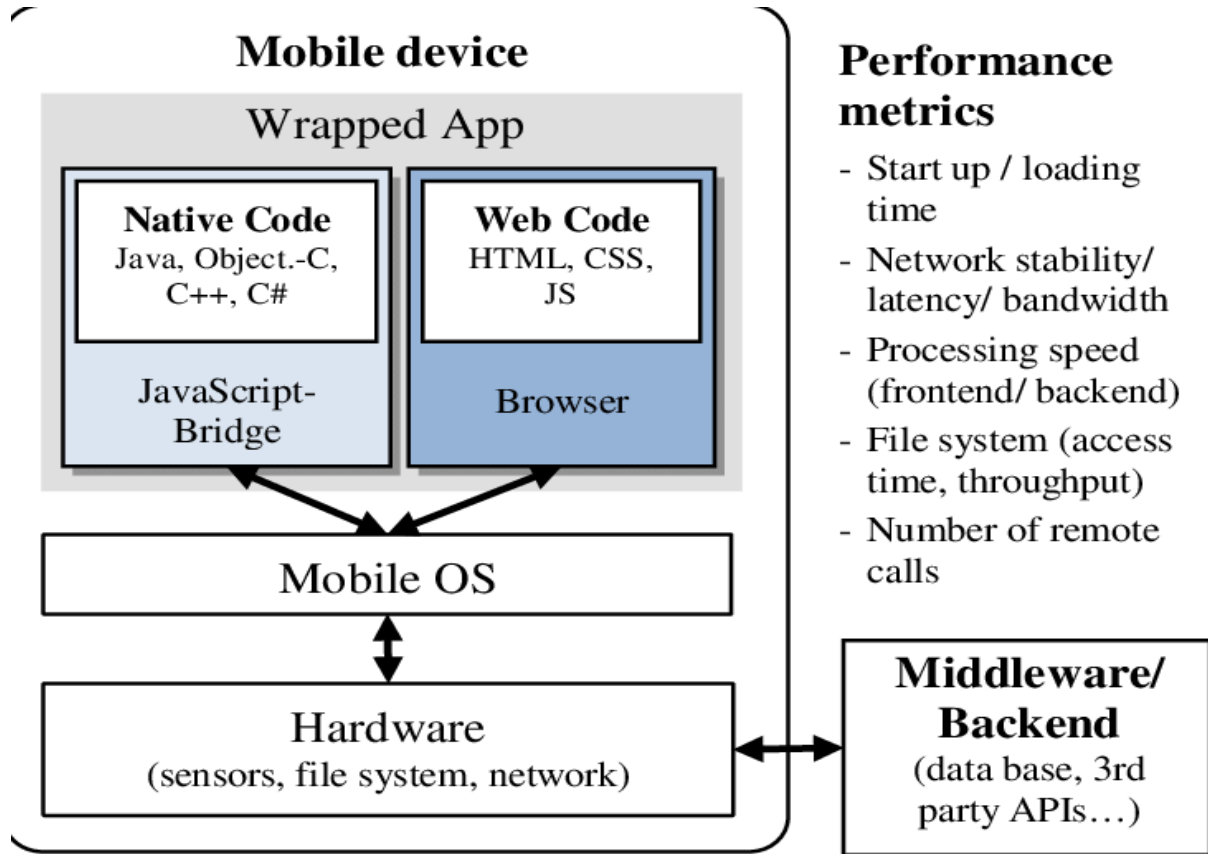
# Mobile Hybrid Architecture



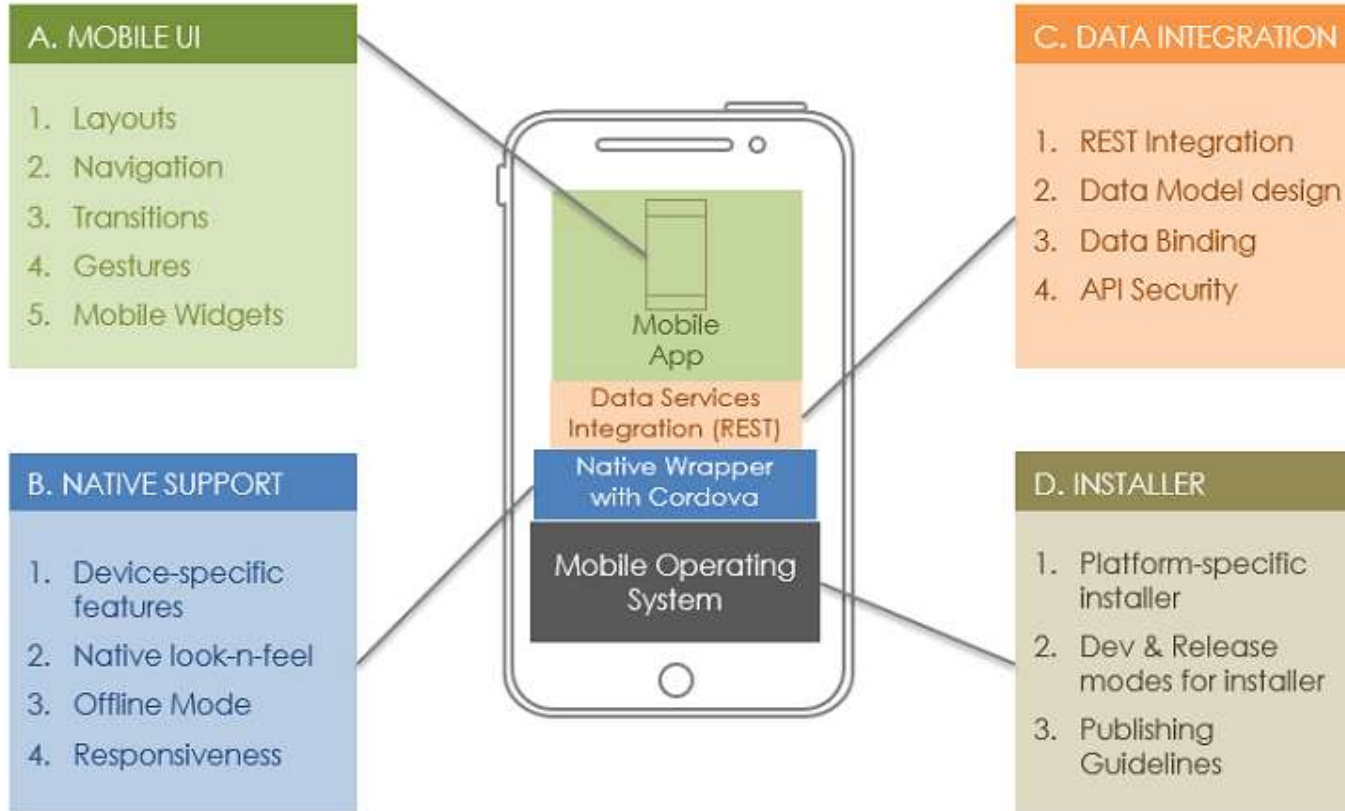
# Elements of App Architecture



# Hybrid Architecture



# Hybrid App Development



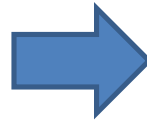
# Mobile Hybrid Architecture

- ❑ 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

## Native Apps



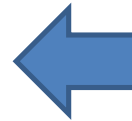
## Pros



- Bare Metal
- Best Performance
- Standardized UI Elements
- Full API Access

- Separate code base
- Completely different platforms
- Expensive
- More time

## Cons



# Comparison

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



# Features of Hybrid

- ❑ 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
- ❑ Hybrid apps run code inside a container
- ❑ Device's browser engine is used to render HTML and JavaScript and native [APIs](#) to access device-specific hardware

# IONIC Framework

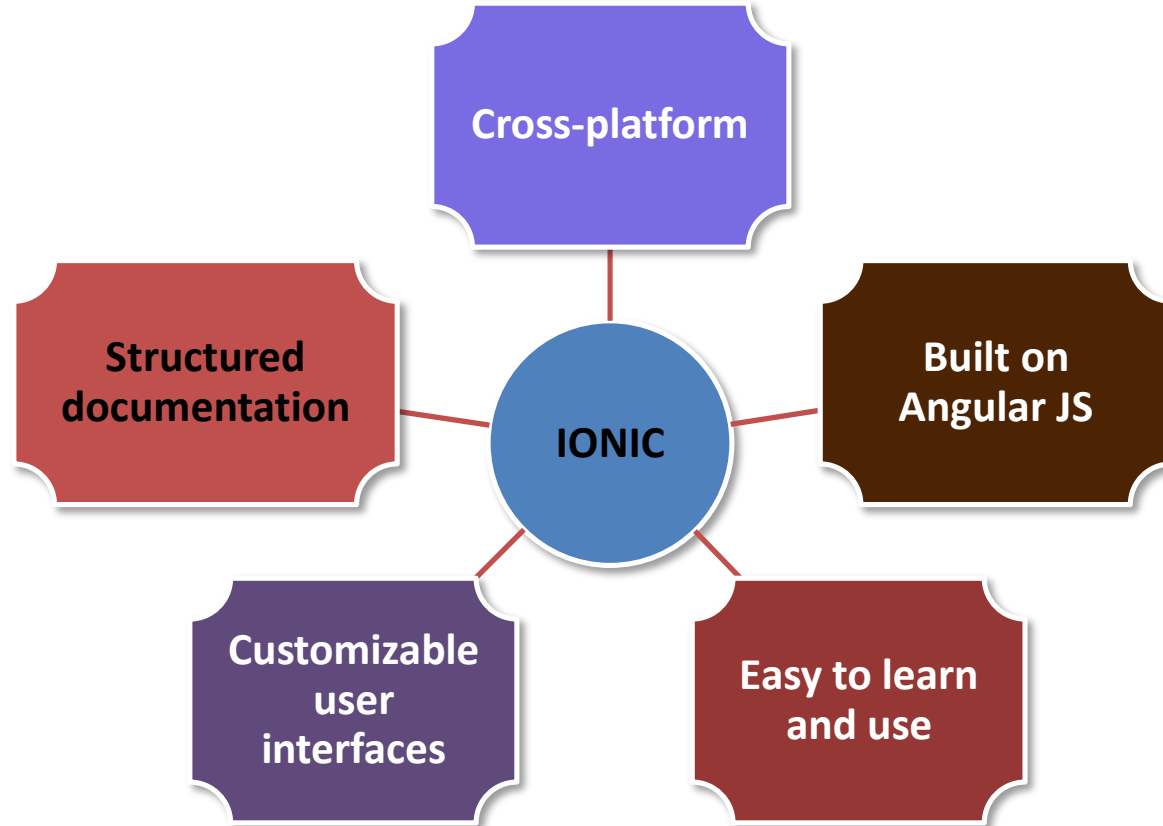


Open-source UI framework for building high quality **mobile apps**, desktop apps, and progressive **web apps** using web technologies such as HTML, CSS, and JavaScript

# PROGRESSIVE WEB APPS

- ❑ Normal web app which looks and feel like native mobile apps
- ❑ It uses modern web capabilities to deliver an app-like experience to the user
- ❑ It can easily deploy to servers, accessible through URLs, and indexed by search engines

# Why IONIC?



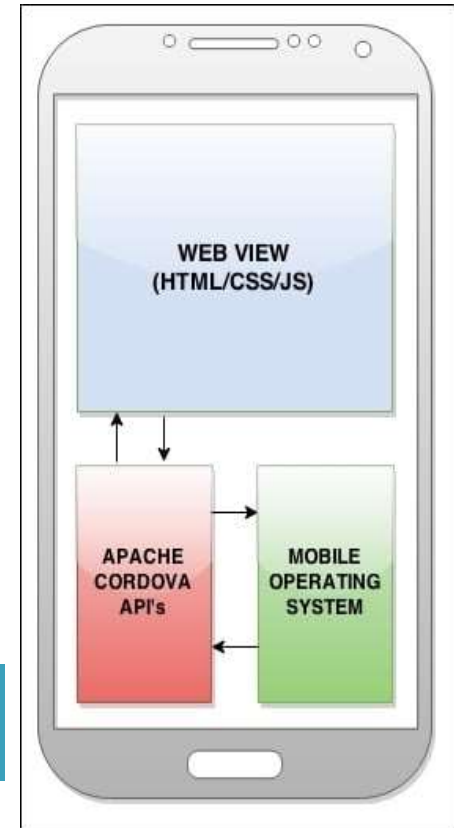
# IONIC Framework



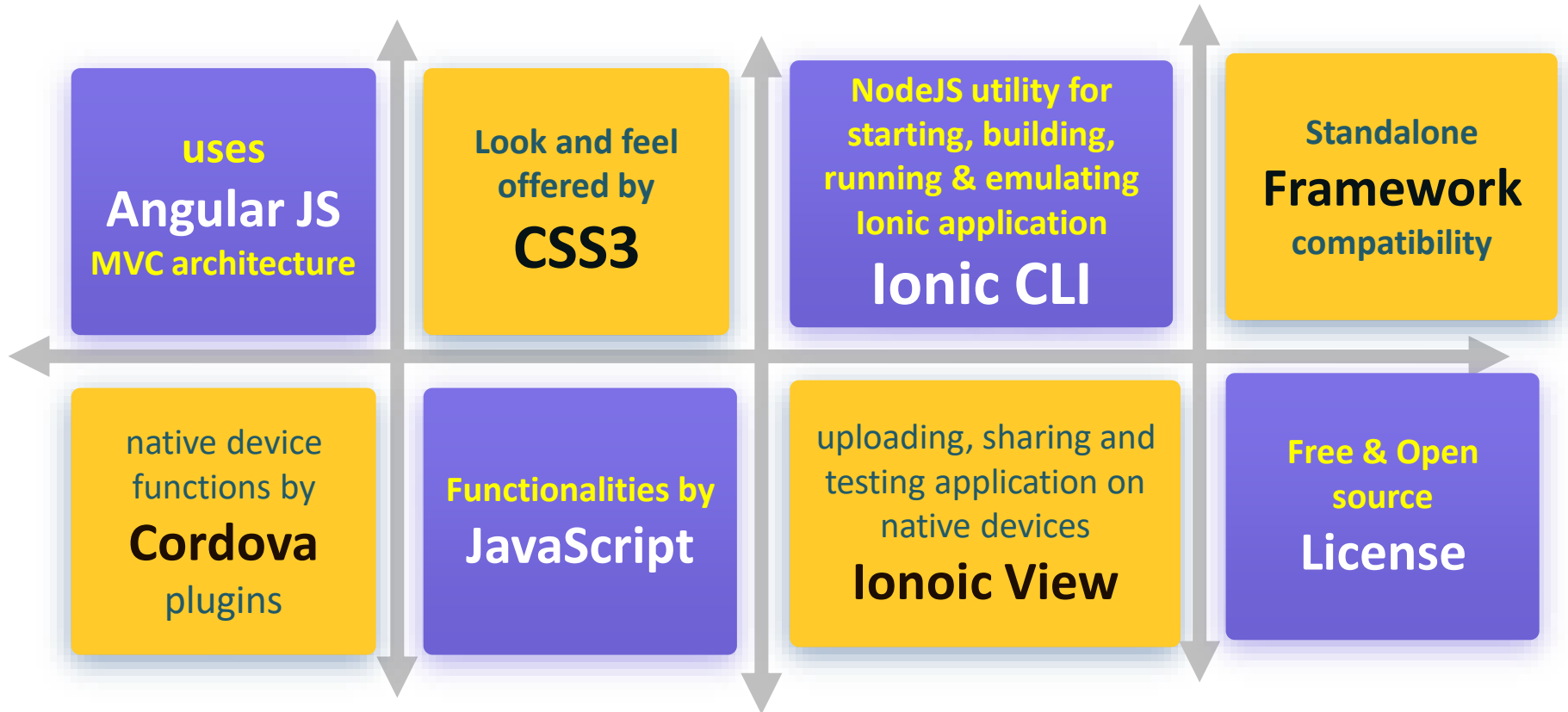
# Mobile Hybrid Architecture

- ❑ 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

Cordova has a bunch of APIs that interface with the WebView using JavaScript and then talk to the device in its native language



# Features of IONIC?



# Project Structure

- ❑ **src:** folder where app source code will be placed here
- ❑ **hooks:** contains scripts to perform Cordova task
- ❑ **Resources:** consists of various versions of the application icon and splash screen
- ❑ **www:** consists of the build Ionic code, written inside the src folder
- ❑ **config.xml:** file consists of all the meta information needed by Cordova while converting our Ionic app to a platform-specific installer
- ❑ **ionic.config.js:** This file consists of the configuration that is needed for the build task
- ❑ **package.json:** File consists of the project-level node dependencies
- ❑ **tsconfig.json:** File consists of the TypeScript configuration
- ❑ **tslint.json:** File consists of TS lint rules

```
├── config.xml
├── hooks
├── ionic.config.js
├── node_modules
├── package.json
├── platforms
├── plugins
├── resources
├── src
├── tsconfig.json
├── tslint.json
└── www
```



# IONIC?

- ❑ Ionic has three main/go-to templates for developing apps
  - **Blank:** This is a blank Ionic project with one page
  - **Tabs:** This is a sample app that is built using Ionic tabs
  - **Side menu:** sample app that is built to consume side menu driven navigation
- ❑ To experiment, create a project folder and tupe the following command inside the folder

**ionic start -a "Example 1" -i app.example.one example1 blank --v2**

- -a "Example 1": This is the human-readable name of the app
- -i app.example.one: This is the app ID/reverse domain name
- example1: This is the name of the folder
- blank: This is the name of the template
- --v2: This flag indicates that the project will be scaffolded with the latest version of Ionic. This may be removed in the future

# Getting started IONIC

- Install Node/NPM
- Install Ruby/Sass
- Learn the basics of Angular and Sass
- Install Gulp (optional)
- Install XCode, Android Studio or Visual Studio for distribution (optional)

## Steps to create Project

- ❑ Install Cordova and IONIC

- ❑ Clone a starter project by

```
ionic start mySideMenu sidemenu
```

```
$ cd mySideMenu
```

- ❑ x