Introduction to Angular + Ionic + Capacitor

...

Luca Sciullo
luca.sciullo@unibo.it

Mobile Apps

“A mobile application, also referred to as a **mobile app or simply an app**, is a computer program or software application designed to run on a mobile device such as a phone, tablet, or watch”

Wikipedia

Native Apps

A native mobile app is intimately tied to the platform on which it is running.

- It has the ability to fully integrate with the capabilities of both the hardware and the OS on which it resides.
- It is completely analogous to most PC applications, which are downloaded to a desktop or laptop hard drive and completely executed within that machine.
- In order to accomplish this tight integration, the mobile app developer utilizes an SDK from the hardware manufacturer directly or via the mobile device's OS vendor.
- Combined with an IDE, a developer is able to code the mobile application logic and take advantage of any hardware or OS functionality that is available via the exposed APIs

Hybrid Apps

A hybrid mobile application is developed using both native libraries and web technologies in an attempt to get the best of both worlds.

- The interface between the separate components is an on-platform, embedded HTML rendering engine, which is either developed in-house or acquired from a 3rd-party.
- The native portion of the app can be written as a top to bottom native app, which communicates to a web-based server backend. This has the same porting issues as a purely native app.
- 3rd-party cross-platform development tools exist that use native library containers to achieve near-native performance. Such tools bring the benefits of cross-platform development in both the native and web-based portions of a hybrid mobile app

Mobile Web Applications - Progressive Web Applications (PWAs)

Web-based mobile apps are developed with the same tools used for mobile website development through the use of HTML, CSS style sheets and JavaScript.

- HTML5 provides the ability to create rich UI experiences with support for rich media, UI components, geolocation, and offline execution.
- Third-party suppliers of JavaScript toolkits can supply UI components that allow web-based apps to mimic native look and feel on the mobile device, such as Dojo or jQuery. However, their ability to provide precise native look and feel varies across toolkits.

Example: https://www.gardainformatica.it/blog/sviluppo-software/esempio-progessive-web-app-android-installa.webp?v=1614264237

Hybrid applications vs PWAs

Hybrid

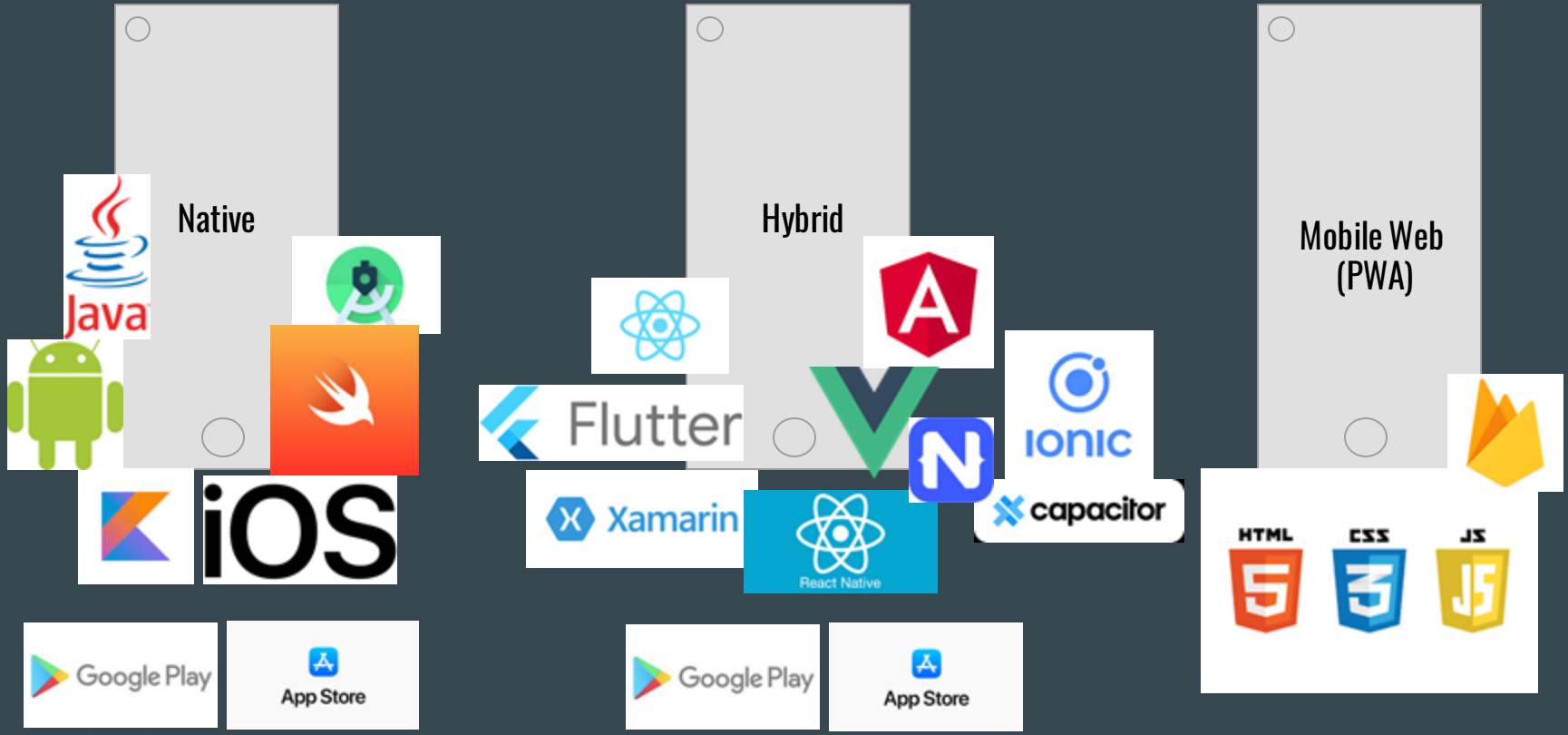
- Web technologies (HTML, CSS, Javascript)
- Run in a browser Web View
- Access device capabilities via plugins
- Wrapped in a native app shell
- Native app

PWA

It is just a regular website that runs in a browser with some enhancements and gives app-like experience to users by using modern web capabilities.

- Installation on a mobile home screen
- Offline usage
- Camera, push notifications
- Background synchronization

Technologies



Hybrid-Native vs Hybrid-Web

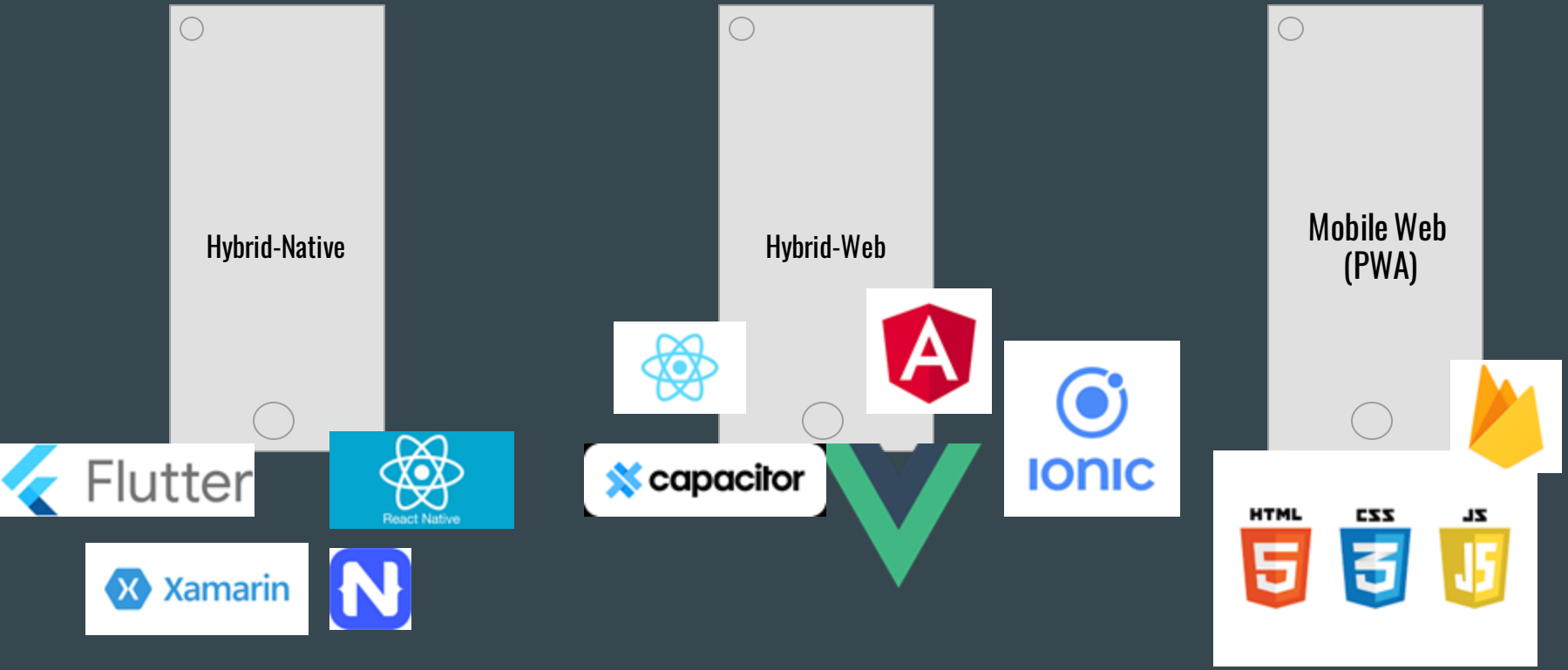
Hybrid-Native: Native UI, Shared Code

JavaScript code runs and orchestrates native UI controls under the hood, so that your app UI is running (almost but not completely) natively. It means that the underlying native UI component that you would like to use or customize must be supported by the framework in order to use that component or change it.

Hybrid-Web: Web UI, Shared Components

The UI is built with HTML/CSS/JS, with native functionality being accessed through portable APIs (or “plugins”) that abstract the underlying platform dependencies. Instead of your entire UI depending on the native platform, only certain native device features, like the Camera, are platform-dependent.

Technologies: Hybrid-Native vs Hybrid-Web vs PWAs



Native, Hybrid or PWA ?

Native, Hybrid or PWA ?

Well, as usual, it depends...

Native, Hybrid or PWA ? - Pros

Native

- Direct access to all the mobile device's features
- Highest performance possible
- Direct access to UI components of specific platforms

Hybrid

- Easy access to mobile device's features
- UX is nearly identical to native apps
- High performances
- Single codebase, and lower development costs

PWA

- No fee nor approval procedures to publish the application
- No installation required
- Lowest development costs
- Single codebase

Native, Hybrid or Mobile Web ? - Cons

Native

- Highest Development & Maintenance costs
- Multiple codebases
- High Development Time
- Limited customization

Hybrid

- High Development & Maintenance costs
- The efficiency and the quality of the application strongly depends on the technology used
- Worse performances

PWA

- Limited set of functionalities
- Worse UX
- The efficiency and the quality of the application strongly depends on the technology used
- Worst performances

Angular 2+

Angular is an application design framework and development platform for creating efficient and sophisticated single-page apps.

Angular 2+ is the evolution of AngularJS. Angular.x, with x > 1, is called Angular 2+.

The current version is the 15.2.1. Angular has a major release every 6 months.

Angular 2+

- It is a Javascript framework used to build client-side applications with a rich UI
- Angular code is written in Typescript, that is then compiled into Javascript
- Angular extends HTML tag through *directives* and provide *data binding* and *dependency injection* for reducing the amount of code
- Angular is inspired by the MVC/MVVC Architecture, but it is said to be *Component-based*
    - there are no Controllers or ViewModels in Angular. Instead, there are components, which are made up of a Template (like a view), Classes and MetaData (Decorators).

https://stackoverflow.com/questions/36950582/angular-design-pattern-mvc-mvvm-or-mv

Angular 2+: a lot of stuff

- Property binding Events binding
- Directives - Structural directives
- Modules
- Services
- Components
- Pipes
- Router
- Forms
- Animations

AND SO ON!

https://angular.io/docs

Ionic

Ionic is an open source mobile UI toolkit for building high quality, cross-platform native and web app experiences.

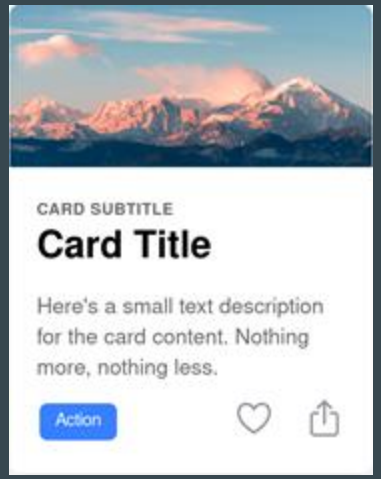
Ionic offers a library of mobile-optimized UI components, gestures, and tools for building fast, highly interactive apps.

Ionic is engineered to integrate seamlessly with all best frontend frameworks, including Angular, React, Vue, or even no framework at all with vanilla JavaScript

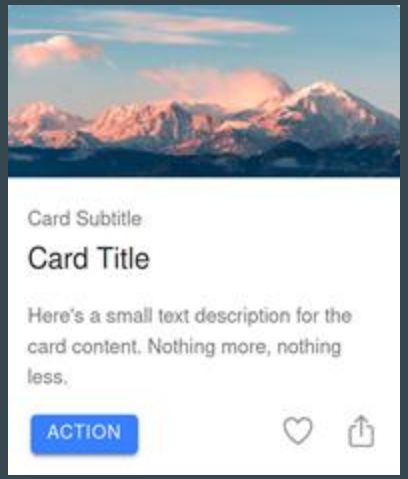
Ionic

Ionic's components are written in HTML, CSS, and JavaScript, making it easy to build modern, high quality UIs that perform great everywhere.

```
<ion-card>
  <ion-img src="/assets/myImg.png"></ion-img>
  <ion-card-content>
    <ion-card-header>
      <ion-card-subtitle>Card Subtitle</ion-card-subtitle>
      <ion-card-title>Card Title</ion-card-title>
    </ion-card-header>
    <p>Here's a small text description for the card component.
       Nothing more, nothing less.
    </p>
    <ion-item>
      <ion-button fill="solid">Action</ion-button>
      <ion-icon name="heart" slot="end"></ion-icon>
      <ion-icon name="share" slot="end"></ion-icon>
    </ion-item>
  </ion-card-content>
</ion-card>
```



iOS



Android

https://ionicframework.com/docs/components

Capacitor

Capacitor is an open source native runtime for building Web Native apps. It creates cross-platform iOS, Android, and Progressive Web Apps with JavaScript, HTML, and CSS

Capacitor's native plugin APIs make it extremely easy to access and invoke common device functionality across multiple platforms.

Capacitor is a spiritual successor to Apache Cordova and Adobe PhoneGap, with inspiration from other popular cross-platform tools like React Native and Turbolinks, but focused entirely on enabling modern web apps to run on all major platforms with ease. Capacitor is backward-compatible with many existing Cordova plugins.

Capacitor Plugins

Camera

Capture images, save photos, and configure hardware parameters like saturation and color balance.

File System

Save and read documents, assets, and other content your users need to access via native file systems.

Geolocation

Gather critical information about a user's device location, such as latitude and longitude.

Accelerometer

Access the device accelerometer sensors to measures changes in velocity of a device motion.

Notifications

Schedule local notifications on the device or handle push notifications sent from a server.

Network

Monitor for network connectivity and capability changes to build resilient offline apps.

Haptics

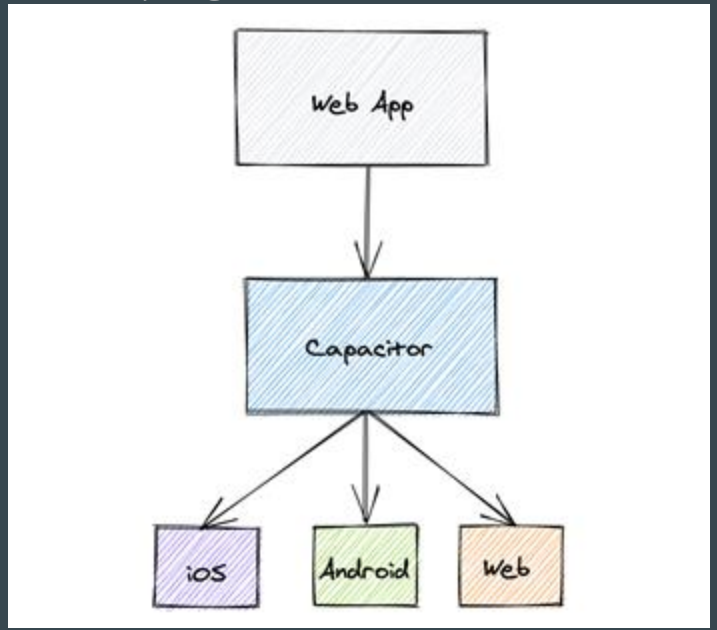
Add physical feedback through haptic features available on modern devices.

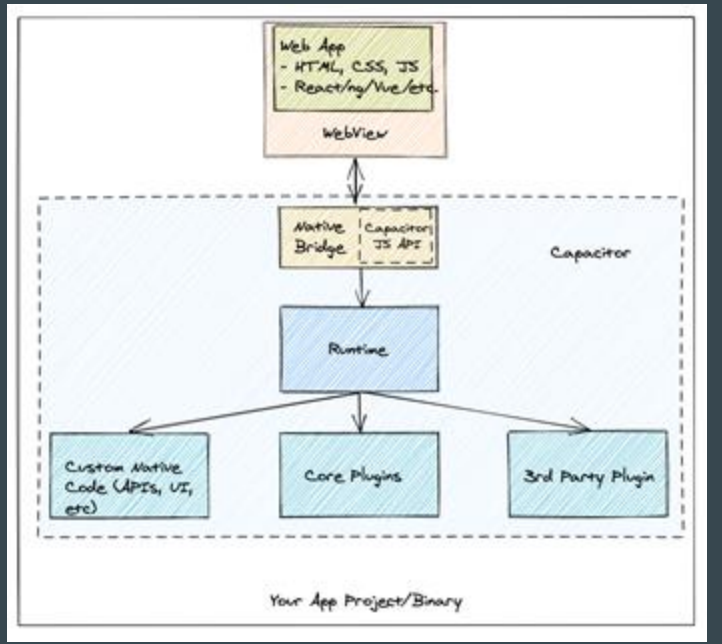
Your Own Plugin

Write your own custom plugins to access specialty features and easily integrate any 3rd-party SDK.

How it works

Capacitor acts as the runtime facilitating communication between the web app and the underlying OS.





https://capacitorjs.com/blog/how-capacitor-works

Demo time !

Angular: getting started

Install the Angular CLI:

$ npm install -g @angular/cli

Create a new workspace and initial starter app:

$ ng new my-app

Run the application

$ cd my-app
$ ng serve --open

Default port is 4200 -> http://localhost:4200/

https://angular.io/guide/setup-local

Demo app: fake login

The app shows a form with an email and a password field.

- If the values inserted match a predefined combination, then the app shows a list of elements.
- Otherwise it shows an error message.

This is a simplified example. For a real login, we would need to use Angular *guards, reactive forms, routing, http service, etc..*

Ionic: getting started

Install the Ionic CLI:

$ npm install -g @ionic/cli
Create a new workspace and initial starter app:

$ ionic start demo tabs --type=angular --capacitor

Run the application

$ cd demo
$ ionic serve

Default port is 8100 -> http://localhost:8100/

https://ionicframework.com/docs/intro/cli

Demo app: fake login

The app shows a form with an email and a password field.

- If the values inserted match a predefined combination, then the app shows a list of elements. From the same page user can open the camera through a button.
- Otherwise it shows an error message.
- See the differences between iOS and Android rendering in the browser
- Run it on Android emulator