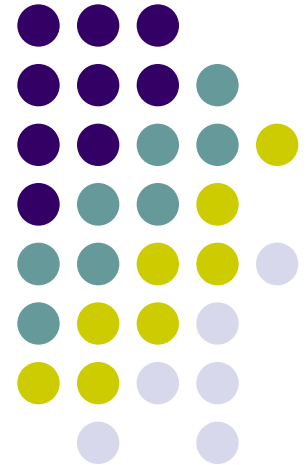


CS 528

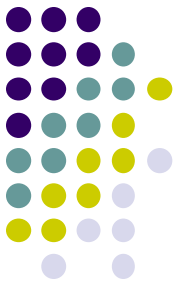
***Ubiquitous and Mobile Computing
Tech Talk: Ionic***



*Yijie Yan, Qinlun Luan,
Wei Xiong, Zinan Yue
Computer Science Dept.
WPI*



Contents



Background

Specific Problems

Use Case

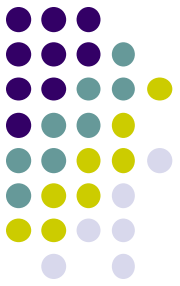
Real World Examples

How it Works

Code Snippet

ionic

Background



Native

Hybrid

Application developed **specifically** for a mobile operating system.

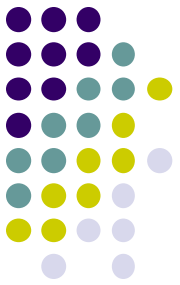


VS

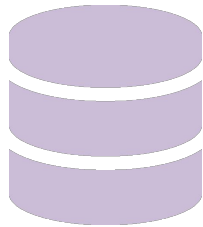


Hybrid applications are, at core, **websites** packaged into a native wrapper.

Background



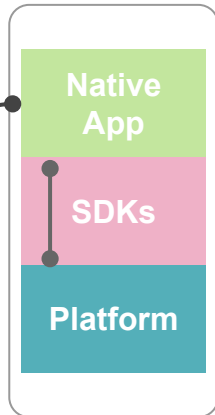
Native app



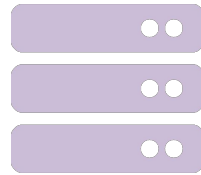
Web service
(database)

HTTP
request

This is
where the
app code
exists.

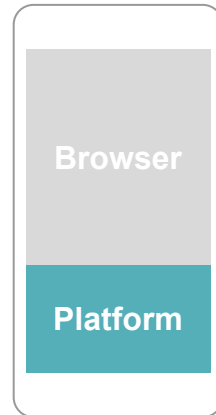


Mobile website



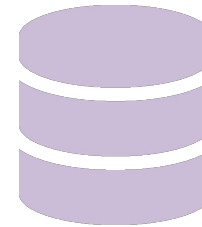
Web server

HTTP
request



This is
where the
app code
exists.

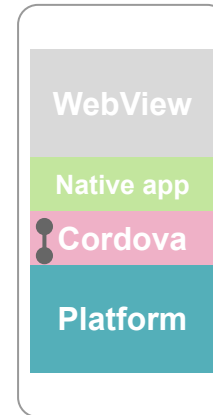
Hybrid app

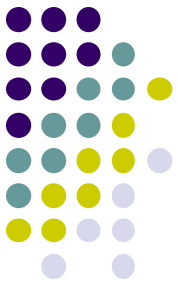


Web service
(database)

HTTP
request

This is
where the
app code
exists.





Background



Hybrid

is an **APP** development platform

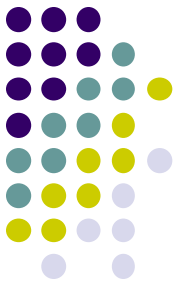
Help developers build and deploy cross-platform apps. From open source to premium services, Ionic makes app creation lightning fast.

An **open source UI toolkit** for building mobile and desktop apps using web technologies (HTML, CSS, and JavaScript).

Focused on **frontend** user experience, or UI interaction of an app (controls, interactions, gestures, animations).

Has official **integrations** with **Angular** and **React**, and support for **Vue** is in development.

Specific Problems



Cross-platform

One code base

Build and deploy apps across **multiple platforms**, such as native iOS, Android, desktop, and the web as a Progressive Web App

Web Standards-based

Web tech

Built on top of reliable, standardized **web technologies**: HTML, CSS, and JavaScript

Beautiful Design

Base theme

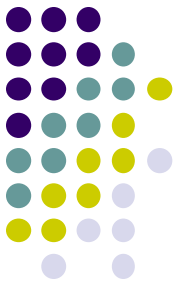
Designed to work and display cross all platforms. Start with pre-designed components, typography, interactive paradigms, and a base theme.

Simplicity to use

Easy develop

Creating Ionic apps is easy to learn, and accessible to anyone with **web development** skills.

Use Case



Device

Device loads
Cordova app
wrapper



APACHE
CORDOVA

Cordova
APP Wrapper

WebView with
JavaScript API

Cordova is a hybrid mobile app platform for building mobile apps that can run using HTML, CSS, and JavaScript inside of a native app . It's a utility for creating a bridge between the platform and the application.



Web app

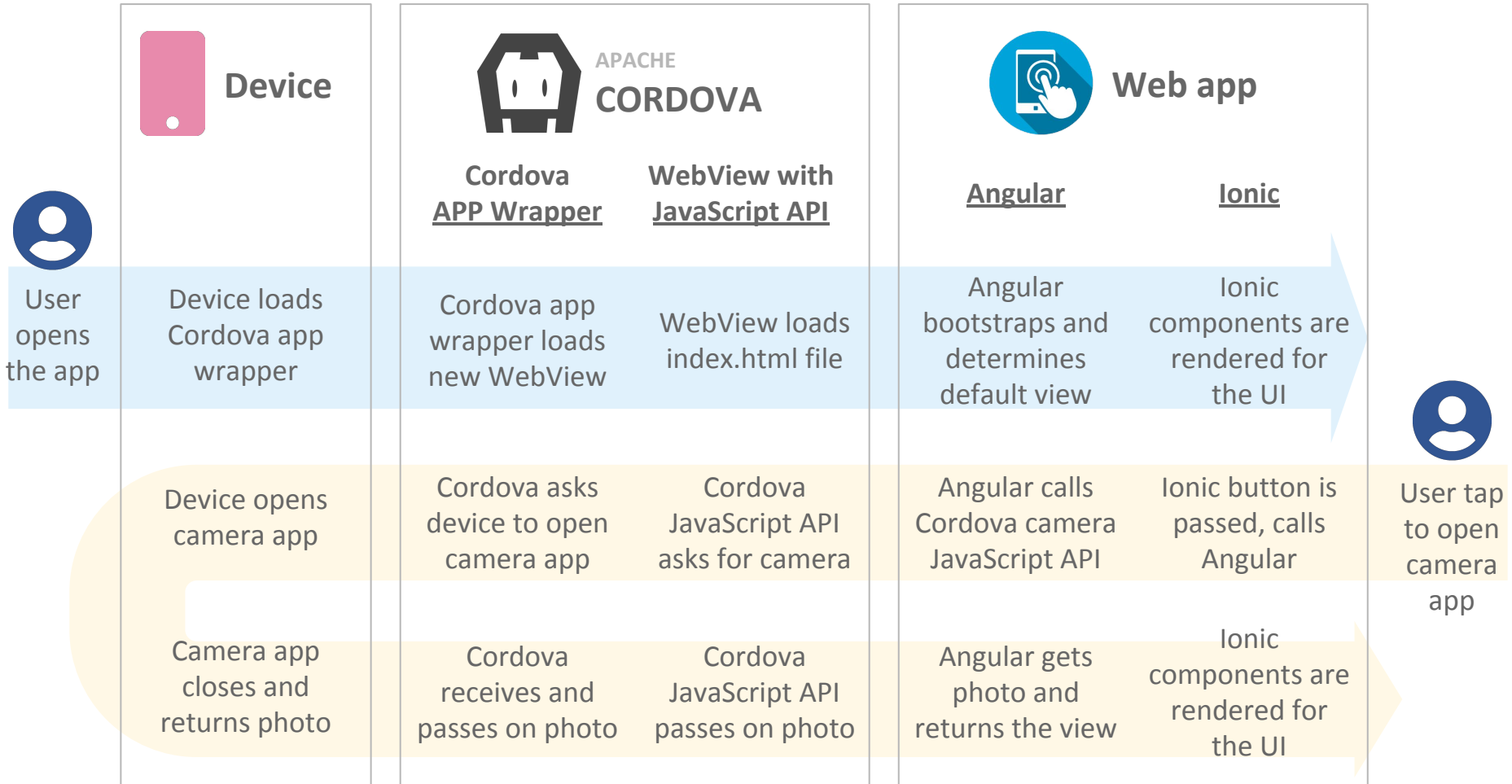
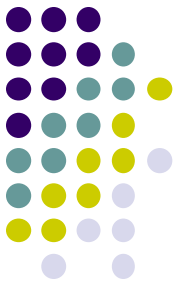
Angular

Ionic

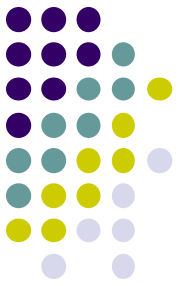
Angular is a web app that controls the app routing and function

Ionic provides the user interface components rendered in the app

How it Works



Real World Examples



Amtrak did a complete overhaul of their existing app and converted it to Ionic.



GE Transportation uses Ionic to build mission critical apps ensuring timely deliveries.



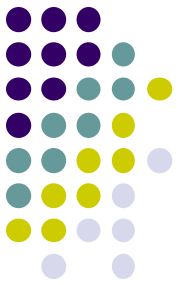
The National Health Service, one of the UK's top health providers went all in on Ionic.



The Marketwatch team built their flagship app in record time leveraging Ionic.



Code Snippet

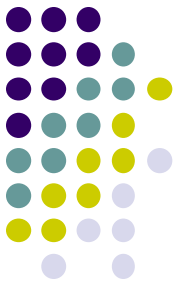


/src/index.html

```
<!-- Ionic's root component and where  
the app will load -->  
<ion-app></ion-app>  
  
<!-- The polyfills js is generated  
during the build process -->  
<script  
src="build/polyfills.js"></script>  
  
<!-- The vendor js is generated during  
the build process  
It contains all of the  
dependencies in node_modules -->  
<script  
src="build/vendor.js"></script>  
  
<!-- The main bundle js is generated  
during the build process -->  
<script src="build/main.js"></script>
```

`src/index.html` is the main entry point for the app, though its purpose is to set up scripts, CSS includes, and bootstrap, or start running our app

Ionic looks for the `<ion-app>` tag in your HTML

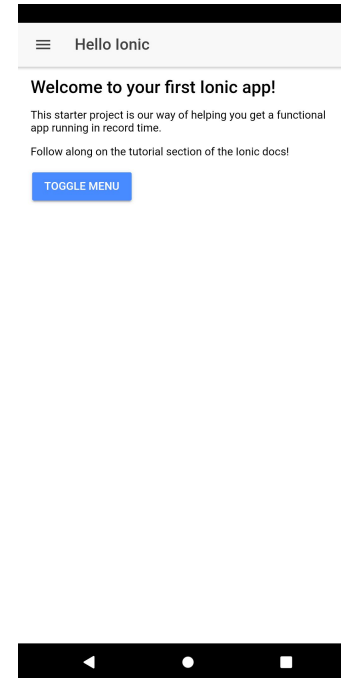


Code Snippet

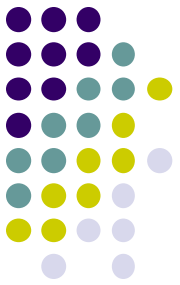
[/src/app/app.module.ts](#)

```
@NgModule({
  declarations: [MyApp,
HelloIonicPage, ItemDetailsPage,
ListPage],
  imports: [BrowserModule,
IonicModule.forRoot(MyApp)],
  bootstrap: [IonicApp],
  entryComponents: [MyApp,
HelloIonicPage, ItemDetailsPage,
ListPage],
  providers: [StatusBar, SplashScreen,
{provide: ErrorHandler, useClass:
IonicErrorHandler}]
})
export class AppModule {}
```

[src/app/app.module.ts](#) is the entry point for our app.



Code Snippet



/src/app/app.html

```
<ion-menu [content]="content">

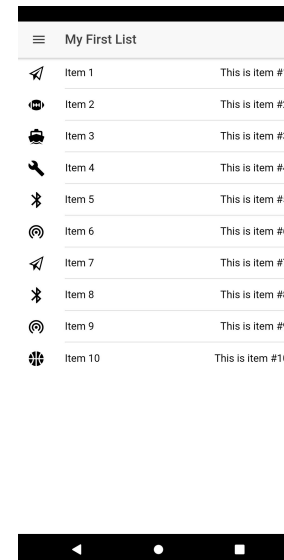
  <ion-header>
    <ion-toolbar>
      <ion-title>Pages</ion-title>
    </ion-toolbar>
  </ion-header>

  <ion-content>
    <ion-list>
      <button ion-item *ngFor="let p
of pages" (click)="openPage(p)">
        {{p.title}}
      </button>
    </ion-list>
  </ion-content>

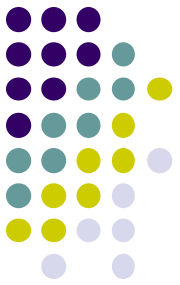
</ion-menu>

<ion-nav [root]="rootPage" #content
swipeBackEnabled="false"></ion-nav>
```

we set up an `ion-menu` to function as a side menu, and then an `ion-nav` component to act as the main content area. The `ion-menu`'s `[content]` property is bound to the local variable `content` from our `ion-nav`, so it knows where it should animate around



Code Snippet



/src/pages/item-details/item-details.html

```
<ion-header>
  <ion-navbar>
    <button menuToggle *ngIf="!selectedItem">
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>Item Details</ion-title>
  </ion-navbar>
</ion-header>
<ion-content>
  <h3 text-center *ngIf="selectedItem">
    {{selectedItem.title}}
    <ion-icon
[name]="selectedItem.icon"></ion-icon>
  </h3>
  <h4 text-center *ngIf="selectedItem">
    You navigated here from
  <b>{{selectedItem.title}}</b>
  </h4>
</ion-content>
```

