Mobile Programming and Multimedia

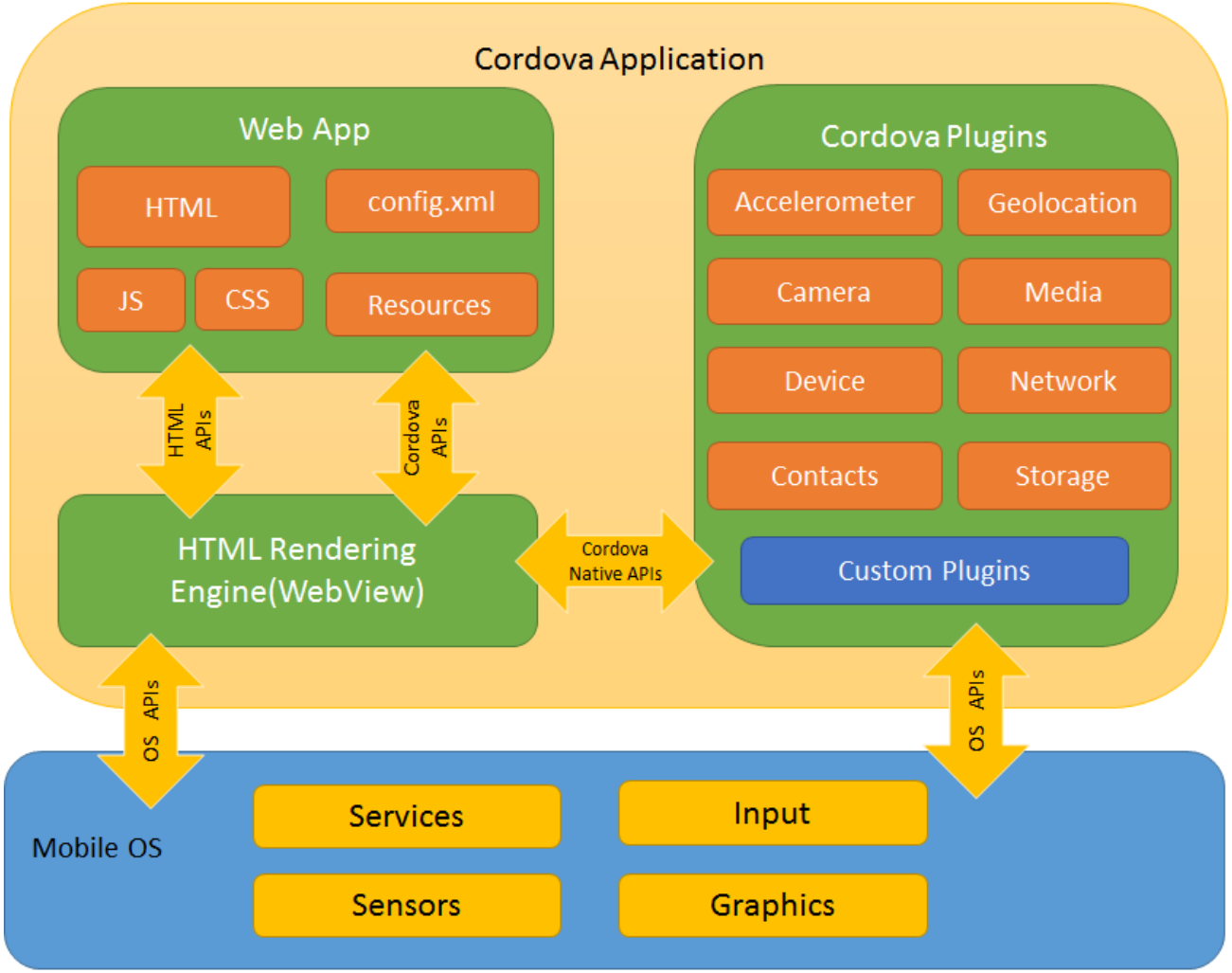
PhoneGap/Cordova

Prof. Ombretta Gaggi
University of Padua

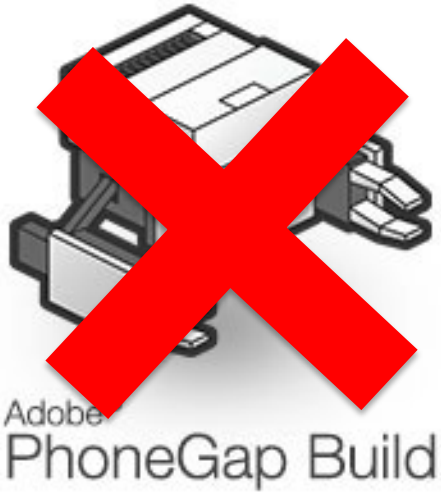
Apache Cordova and PhoneGap: what differences?

In 2011 PhoneGap code was offered to Apache to continue the development. Apache Cordova is the engine below PhoneGap, like WebKit is the engine of several browser

Architecture



Tools



It is an open-source project

From October 1st ,2020

A bit of history

The project started in 2008 trying to solve these problems:

- Development of mobile applications using web technologies
- Solve the problem of low support of mobile browsers to HTML5
- Allow access to different features of the device

Actual support to HTML5 of the mobile browsers and the HTML5 evolution have partially solved these problems

Progressive Web App (PWA) - 1

Progressive Web App (*PWA*) are web pages that behave like native applications. In particular:

- They are developed using web technologies, therefore HTML5, CSS3, Javascript
- It works independently from the browser, using *progressive enhancement* (the more features the browser provides, the more features provides the app)
- It works even offline, but with limited support
- Can be installed without using the store (but in this case, they are a sort of link)

Progressive Web App (PWA) - 2

Progressive Web App (*PWA*) are web pages that behave like native applications. In particular:

- Like every web page, these apps adapt themselves to device size (*responsive*)
- Secure (https) and indexed by search engines
- Easy to update
- Support push notifications (but…)
- No need for stores to publish the app, but there is no payments management, and there is no control of what is published

Not a new concept

Steve Jobs coined the term web app in 2007

In 2015 Chrome developers coined the term *Progressive Web App* to describe those apps using new functionalities like service workers and web app manifest

Other frameworks with Cordova

Other frameworks/tools allow app development using Cordova:

- Monaca
- Framework7
- NativeScript
- Ionic Capacitor
- Progressive Web Apps

Cordova usually is not used stand-alone, but as a support framework for other frameworks

Development languages

Apache Cordova framework is a hybrid framework

- Applications development works with HTML, CSS and JSS, tools already known by all web developers
- It uses plugins to access hardware components of the smartphone (camera, GPS, etc.)

It provides tools for testing (emulators) and deployment of the final application

App structure

UNIVERSITÀ DEGLI STUDI DI PADOVA

A phonegap application is essentially made up of:

- config.xml
- index.html (in a www folder)
- CSS for layout definition
- JS files with app logic
- WebView is a container like a browser for app rendering

Cordova vs PhoneGap

Before, there were two options available:

- PhoneGap Desktop App: provides a drag&drop interface easy to use
- CLI (Command Line Interface): an interface with a command line, with additional features

Now there is only the CLI

Creation of a new app

UNIVERSITÀ DEGLI STUDI DI PADOVA

- The commands for the command line are:
  1. cordova create **hello** com.example.hello **HelloWorld** → **HelloWorld** is the name of the app and **hello** is the name of the folder with the source code
  2. From hello folder: cordova platform add **platform** → **platform** is the name of the platform (ios, android)
- This operation creates 4 folders
  - Node_modules
  - platforms
  - plugins
  - www: in this folder index.html is the initial page of the app

Platforms support

Platform:	Android	iOS	OS X	Windows 8.1, Phone 8.1, 10	Electron
CLI shorthand:	android	ios	osx	windows	electron
	Cordova CLI Development Platform				
Mac	✓	✓	✓	✗	✓
Windows	✓	✗	✗	✓	✓
Linux	✓	✗	✗	✗	✓

config.xml - 1


```
<?xml version='1.0' encoding='utf-8'?>
<widget id="io.cordova.hellocordova" version="1.0.0"
      xmlns="http://www.w3.org/ns/widgets"
      xmlns:cdv="http://cordova.apache.org/ns/1.0">

  <name>HelloCordova</name>
   <description>
      A sample Apache Cordova application that responds to the
      deviceready event.
   </description>
   <author email="dev@cordova.apache.org" href="http://cordova.io">
      Apache Cordova Team
   </author>
   <content src="index.html" />
   <access origin="*" />
```

config.xml - 2

UNIVERSITÀ DEGLI STUDI DI PADOVA

```
    <allow-intent href="http://*/*" />
    <allow-intent href="https://*/*" />
    <platform name="android">
        <allow-intent href="market:*" />
    </platform>
    <platform name="ios">
        <allow-intent href="itms:*" />
        <allow-intent href="itms-apps:*" />
    </platform>
    <plugin name="cordova-plugin-whitelist" spec="^1.3.2" />
    <plugin name="cordova-plugin-camera" spec="^2.4.1" />
    <plugin name="cordova-plugin-contacts" spec="^2.3.1" />
    <plugin name="cordova-plugin-geolocation" spec="^2.4.3" />
</widget>
```

HelloWorld – head in detail

UNIVERSITÀ DEGLI STUDI DI PADOVA

```
<html>
<head>
   <meta charset="utf-8" />
   <meta name="viewport"
         content="user-scalable=no, initial-scale=1,
            maximum-scale=1, minimum-scale=1,
            width=device-width" />
   <meta name="format-detection" content="telephone=no" />
   <meta name="msapplication-tap-highlight" content="no" />
   <link rel="stylesheet" type="text/css" href="css/index.css" />
   <title>Hello World</title>
</head>
```

viewport: used to indicate how much space of the screen is used by an application and how to scale. In this case it is fixed at screen size

Disable Microsoft functionality that colors in grey something where a tap is made

Disable Apple functionality that allows to call phone numbers, but frequently do not correctly recognize the numbers

HelloWorld – body in details

UNIVERSITÀ DEGLI STUDI DI PADOVA

```
<body>
   <div class="app">
      <h1>PhoneGap</h1>
      <div id="deviceready" class="blink">
         <p class="event listening">Connecting to Device</p>
         <p class="event received">Device is Ready</p>
      </div>
   </div>
   <script type="text/javascript" src="cordova.js"></script>
   <script type="text/javascript" src="js/index.js"></script>
   <script type="text/javascript">
      app.initialize();
   </script>
</body></html>
```

PhoneGap library,
The correct one will be added depending on the device

Specific library for the app that can be modified to add app logic

index.js in depth

This file allows adding the logic of the page

It manages

- Events binding
- Create functions for event handling

The most common events are:

- load
- *deviceready* is an event provided by Cordova API fired when the Cordova APIs are ready to be used. Not using it can create problems if APIs are not completely loaded
- offline
- online

Example: a calculator - 1


```
<form class="calcolatrice">
    <div class="bloc">
        <p>Calcolatrice PhoneGap</p>
        <input type="text"name="ans" id="ans" class="display-calc"/>
    </div>
    <div class="bloc">
        <div class="bloc">
            <input type="button" name="num1" value="1" class="calc" />
            <input type="button" name="num2" value="2" class="calc" />
            <input type="button" name="num3" value="3" class="calc" />
            <input type="button" name="divi" value="/" class="calc" />
        </div>
    </div>
    righe successive
```

Example: a calculator - 2

UNIVERSITÀ DEGLI STUDI DI PADOVA

```
<form class="calcolatrice">
    ....
    <div class="bloc">
        <input type="reset" name="canc" value="C" class="calc" />
        <input type="button" name="num0" value="0" class="calc" />
        <input type="button" name="virg" value="." class="calc" />
        <input type="button" name="addi" value="+" class="calc" />
    </div>
    <div class="bloc">
        <input type="button" id="soluzione" value="=" class="calcola calc" />
    </div>
</form>
```

Logic - calculus

UNIVERSITÀ DEGLI STUDI DI PADOVA

```
function pulsante(numero){
      document.getElementById('ans').value+=numero.value;
}
function jsBottoni(){
      var bottoni = document.getElementsByTagName("input");
      for (var i=0; i<bottoni.length; i++){
            if(bottoni[i].getAttribute("type") == "button"){
                  bottoni[i].onclick = function(){pulsante(this);       }
            }
      }
      document.getElementById('soluzione').onclick=
                  function (){
                        document.getElementById('ans').value=eval(
                                    document.getElementById('ans').value)
                  };
}
jsBottoni();
```

Logic – constructor -1

UNIVERSITÀ DEGLI STUDI DI PADOVA

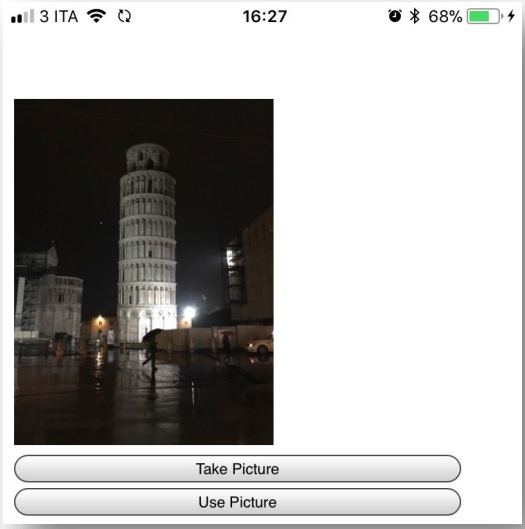
```
var app = {
    initialize: function() {
        this.bindEvents();
    },    // Bind Event Listeners
    bindEvents: function() {
                document.addEventListener('deviceready’,
                                          this.onDeviceReady, false);
    }, // deviceready Event Handler
    onDeviceReady: function() {
        app.receivedEvent('deviceready');
     },
```

Logic – constructor - 2

UNIVERSITÀ DEGLI STUDI DI PADOVA

```
// Update DOM on a Received Event
receivedEvent: function(id) {
    var parentElement = document.getElementById(id);
    var listeningElement = parentElement
                                .querySelector('.listening');
    var calc = document.getElementById("calcolatrice");
    listeningElement.setAttribute('style', 'display:none;');
    calc.setAttribute('style', 'display:block;');
    console.log('Received Event: ' + id);
}
```

Pictures from Camera - 1




```
//event binding
document.addEventListener("deviceready", init, false);
function init() {
    function onSuccess(imageData) { … }
    function onFail(message) { … }
    function takePhoto() { … }
    function takeFile(){ … }
    //Use from Camera
    document.querySelector("#takePicture").addEventListener("touchend",
                                                            takePhoto);
    //Use from Library
    document.querySelector("#usePicture").addEventListener("touchend",
                                                            takeFile);
}
```

Pictures from Camera - 2


```
function onSuccess(imageData) {
    console.log('success');
    var image = document.getElementById('myImage');
    image.src = imageData;
}
function onFail(message) {
    console.log('Failed because: ' + message);
}
```

In index:

```
<img id="myImage">
```

Pictures from Camera - 3


```
function takePhoto() {
    navigator.camera.getPicture(onSuccess, onFail, {
        quality: 50,
        sourceType: Camera.PictureSourceType.CAMERA,
        destinationType: Camera.DestinationType.FILE_URI
    });
}
function takeFile(){
    navigator.camera.getPicture(onSuccess, onFail, {
        quality: 50,
        sourceType: Camera.PictureSourceType.PHOTOLIBRARY,
        destinationType: Camera.DestinationType.FILE_URI
    });
}
```

Bibliography

- Documentazione Cordova
  - https://cordova.apache.org/docs/en/latest/
- Documentazione su config.xml
  - http://cordova.apache.org/docs/en/latest/config_ref/index.html
- Tutorial
  - http://ccoenraets.github.io/cordova-tutorial/
- Esempi
  - https://github.com/cfjedimaster/Cordova-Examples