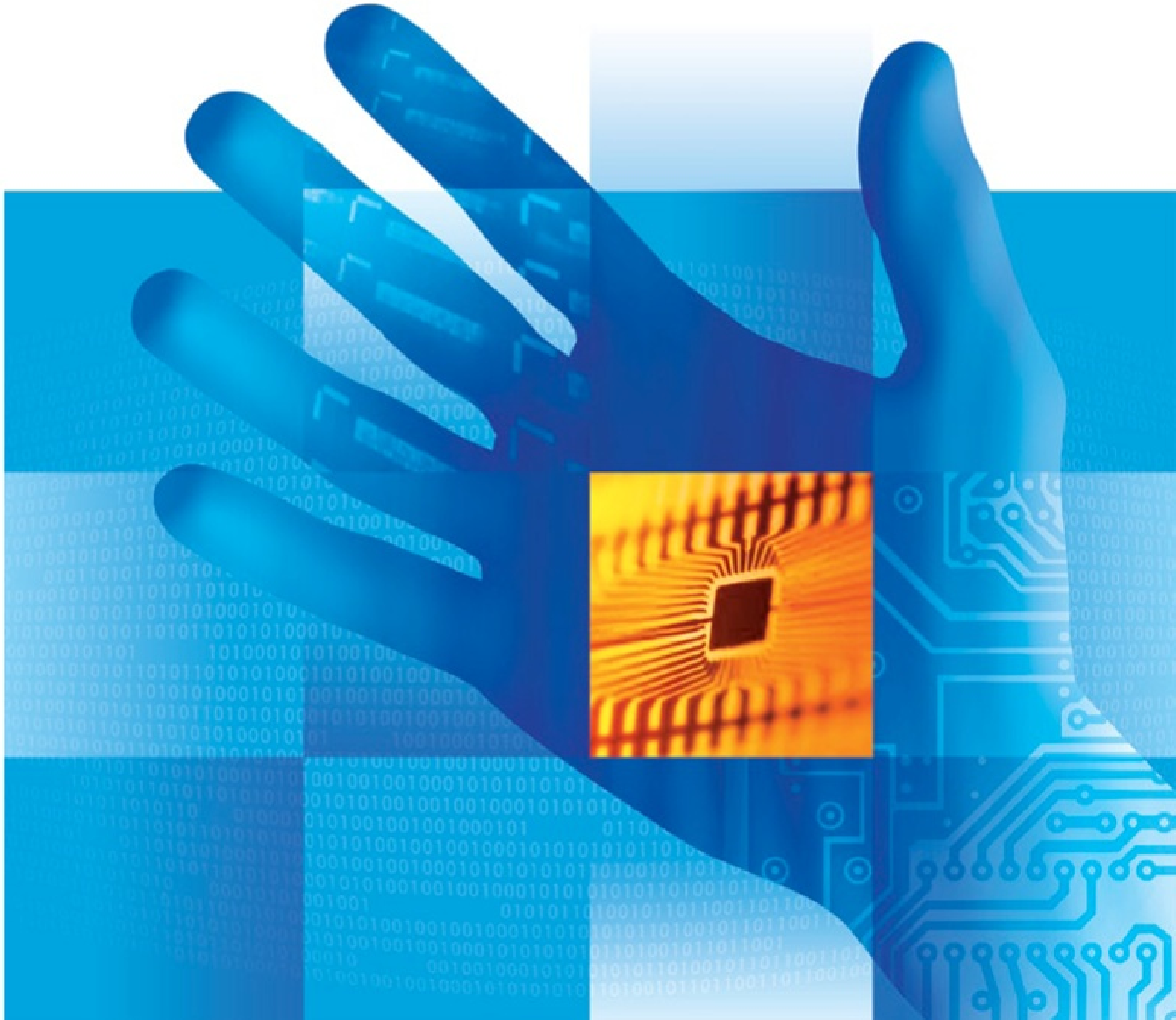
TAMZ I
(Design of Applications for Mobile Devices I)



Lecture 8
Orientation
Apache Cordova

HTML5 Orientation

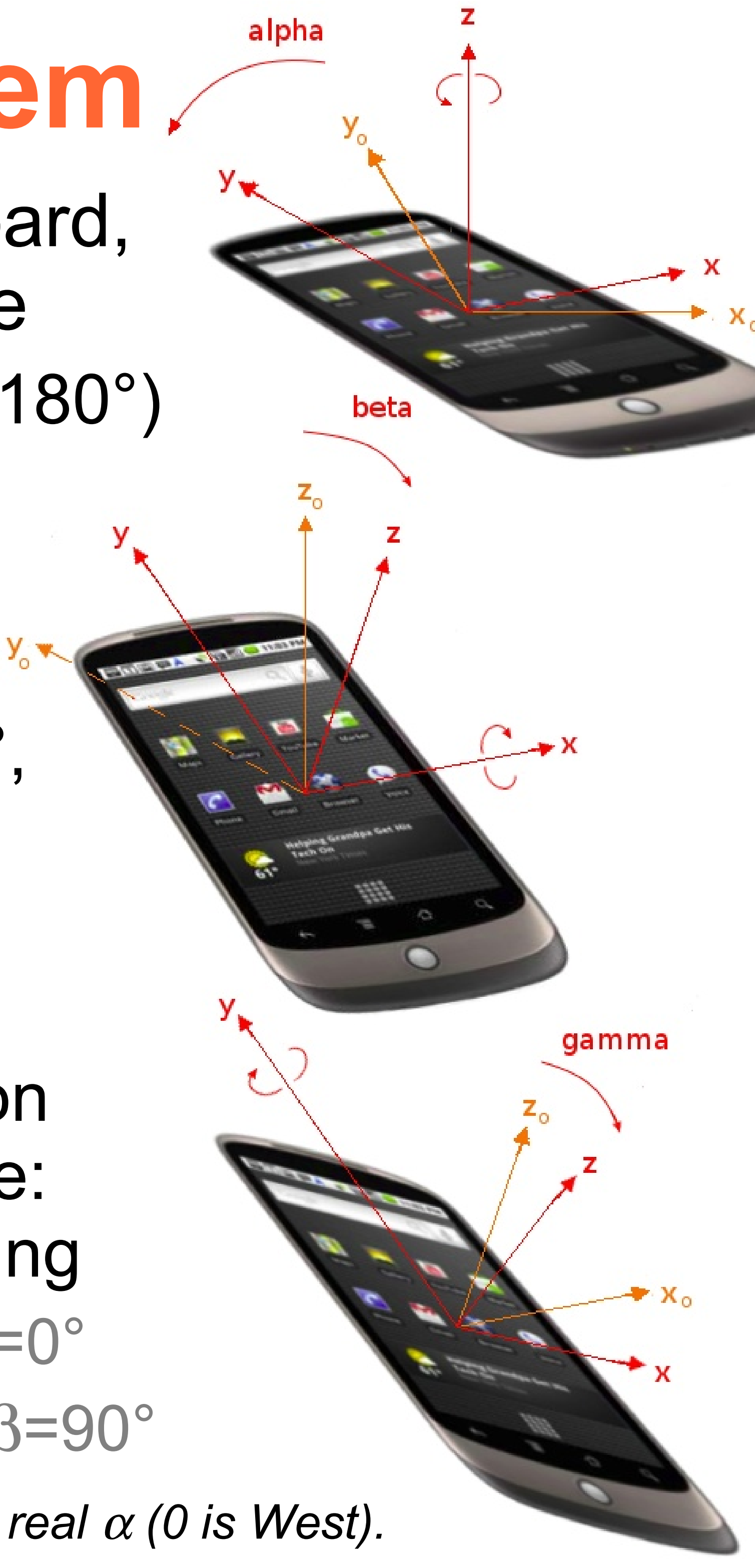
(Android 3.0+, iOS 4.2+)

See e.g.: http://sandbox.juurlink.org/accelerometer/
http://ariya.github.io/device/compass/
http://goo.gl/ulUFYO
http://www.peterfriese.de/how-to-use-the-gyroscope-of-your-iphone-in-a-mobile-web-app/

Specification: http://www.w3.org/TR/orientation-event/

Device coordinate system

- **x** – in the plane of screen or keyboard, positive towards the right hand side
  - **beta** (β) rotation around **x** [-180°, 180°)
- **y** – in the plane of the screen or keyboard, positive towards the top of them
  - **gamma** (γ) rotation around **y** [-90°, 90°)
- **z** – perpendicular to the screen or keyboard, positive out of them
  - **alpha** (α) counterclockwise rotation around **z** [0°, 360°), absolute angle: mostly opposite of compass heading
    - horizontal position: $360-\alpha$ for $\beta=0°$
    - vertical position: $360-(\alpha+\gamma)$ for $\beta=90°$



Images: http://www.w3.org/TR/orientation-event/

Note: Alpha angle in default Android browser is 90° less then real α (0 is West).

Obtaining device position

Based on the device support, we have following events:

- deviceorientation – event provides current orientation angles alpha, beta, gamma (α, β, γ), or null if unknown, absolute indication (true/false) & method initDeviceOrientationEvent
  - All angles use the right-hand convention (thumb – positive axis, remaining fingers – rotation)
  - Init method args: type, bubbles, cancellable, α, β, γ, absolute
- devicemotion – provides 3 separate information mentioned later, data time interval (if set) and initAccelerometerEvent
  - acceleration – the x, y & z acceleration of the device relative to the Earth frame (from gyroscope if available, without the impact of gravity) with axes defined in previous slide
  - accelerationIncludingGravity – as above, but with gravity cancelled (both in [m s^{-2}]), for devices which lack gyroscope
  - rotationRate – device rotation rate in [deg s^{-1}] in angles alpha, beta, gamma
  - Init method args: type, bubbles, canc., acc, accG, rotR, interval
- compassneedscalibration event is almost unsupported

Coordinate transformations

- To transform the orientation angles column vector [alpha, beta, gamma] ([α, β, γ]) to a vector in WGS-84 XYZ earth frame [R_x,R_y,R_z] vector, we use following transformation matrix:

$$R = \begin{bmatrix} -\cos(\alpha)\sin(\gamma) - \sin(\alpha)\sin(\beta)\cos(\gamma) \\ -\sin(\alpha)\sin(\gamma) + \cos(\alpha)\sin(\beta)\cos(\gamma) \\ -\cos(\beta)\cos(\gamma) \end{bmatrix}$$

- To calculate compass heading (absolute angles are required for correct result), we can calculate the angle by

$$\theta = \tan^{-1}\left(\frac{R_x}{R_y}\right) = \tan^{-1}\left(\frac{-\cos(\alpha)\sin(\gamma) - \sin(\alpha)\sin(\beta)\cos(\gamma)}{-\sin(\alpha)\sin(\gamma) + \cos(\alpha)\sin(\beta)\cos(\gamma)}\right)$$

Fixing compass heading

- Different compass rotation direction (some do not observe right-hand-rule) and zero angle for different devices:
- A good overview, shown in following table may be found at https://github.com/ajfisher/deviceapi-normaliser:

	Zero point	Right Hand Rule	Range
API Reference	North (0)	Y	[0, 360]
iOS Chome/Safari	East (90)	Y	[0, 360]
Blackberry	South(180)	N	[0, 360]
Chrome/Opera Android ICS/JB	North (0)	Y	[0, 360]
Stock browser Android ICS	West (270)	Y	[0, 360]
Firefox Android ICS/JB	North (0)	N	[0, 360]

- The page also contains normalizer script to fix behavior across the devices

HTML5 Widgets

(General specification, which is followed by Apache Cordova)

Specifications: http://www.w3.org/TR/widgets/
http://www.w3.org/TR/widgets-access/

Widgets / Packaged Web Apps

- Name comes from Window gadget
- Defined packaging, media type, extension requirements
  - ZIP archive (+permitted variant), application/widget, .wgt
- Reserved file names, package structure definition
  - Reserved: **config.xml**, icon.{png|gif|jpg|ico|svg}, locales
    - Default start files: *index*.{html|htm}, *index*.{svg|xhtml|xht}
  - The files with reserved filenames located in root directory are the widget **configuration** document, default icon, *start page* and locale folder (default icon and start page may be located in root of locale folder, custom defined in the config.xml instead)
- Localization (internationalization) may be done with folder structure or xml:lang attribute.
  - Standard country codes are used, e.g. /locales/cs /locales/cs-cz or xml:lang="en-us"
- Configuration file contains basic settings, also origin:
  - <access origin="http://example.org" subdomains="true"/>
  - <access origin="*"/>

Widget configuration file example

```
<?xml version="1.0" encoding="UTF-8"?>
<widget xmlns="http://www.w3.org/ns/widgets"
id="http://example.org/exampleWidget" version="0.1" height="200"
width="200" viewmodes="fullscreen">
    <name short="Example 0.1">The example Widget!</name>
    <name short="Ukázka 0.1" xml:lang="cs">Ukázkový Widget!</name>
    <feature name="http://example.com/camera" required = "true">
        <param name="autofocus" value="true"/>
    </feature>
    <preference name="apikey" value="deadbeef0075417e" readonly="true"/>
    <description>Adapted example from documentation</description>
    <description xml:lang="cs">Jen ukázka konfigurace</description>
    <author href="http://foo-bar.example.org/" email="foo-bar@example.org">
        Foo Bar Corp
    </author>
    <icon src="icons/example.png"/>
    <icon src="icons/boo.png"/>
    <content src="myWidget.html"/>
    <license>Example license (based on MIT License) …</license>
</widget>
```

Widget configuration file items

- Only the widget element is compulsory
  - All widget attributes (id, version, …) are optional
  - Viewmodes: fullscreen, windowed, floating, maximized, minimized
  - Change of default locale ("") by e.g. defaultlocale="en-us"
- You can specify 0-* children elements
  - name, description, license only once per language
  - author, icon, feature, preference may appear multiple times
- License tag may contain href attribute directing it to the full text of the license and the license description between tags
- Icon and content tags are used to specify custom icons and start page instead of the default files.
  - <content src="lbg-maps.swf" type="application/x-shockwave-flash" />
  - <content src="olddocument.htm" encoding="iso-8859-2" />
  - <icon src="xxx.png" width="48" height="48"/>
- Widget will not work without required features
- Span element may be used in config (changes xml:lang, dir)

<span xml:lang="en" dir="ltr">English text</span>

Native applications with Apache Cordova (PhoneGap)

(Android, iOS, WP, …)

See e.g.: http://cordova.apache.org/docs/en/3.4.0/
http://cordova.apache.org/docs/en/3.4.0/guide_platforms_index.md.html
http://netbeans.org/kb/docs/webclient/cordova-gettingstarted.html
Web site: https://cordova.apache.org/

Apache Cordova basics

- Web-oriented (HTML5+JS+CSS3)
- But it's used for building embedded applications (with installation packages instead of mobile web)
- Large scope of platforms (Android, IOS, WP, Blackberry, Tizen, Ubuntu, Firefox OS, …)
- Common JavaScript API over different platforms, frameworks programming languages & IDEs
- Uses features from HTML5
- But you have to prepare the basic project for deployment to individual platforms
  - Instead of building locally, it is possible to use cloud-based build (may require publically-available sources on Github)

How to use Apache Cordova

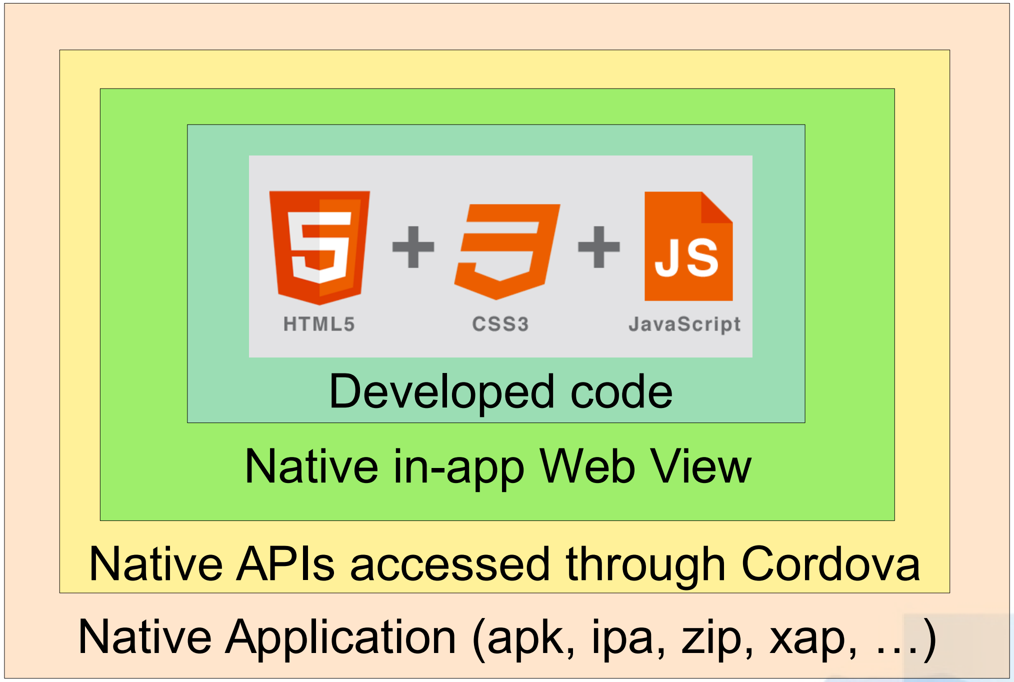
- You add Cordova libraries into your project:
  - the common platform-independent core
  - the platform-dependent API adaptation code
- You use the native IDE for given platform
  - Some IDEs, like NetBeans, offer support for seamless use of Apache Cordova
- You can sell the application using platform-dependent store (App Store, Google Play, Windows Marketplace, …)
- It is possible to use (3rd-party) plugins
  - Plugin may be available only for some platforms, e.g. Android & iOS
  - Examples: Calendar plugin, SMS plugin, …

Libraries for individual platfroms

List for Cordova 3.6.0 (2014-10-28):

- Android
- iOS, Mac OS X
- Bada, Tizen
- Blackberry
- Firefox OS
- Ubuntu, QT
- WebOS
- Windows Phone 7, 8
- Amazon Fire OS

- Desktop Windows


HTML5
CSS3
JS
JavaScript
Developed code
Native in-app Web View
Native APIs accessed through Cordova
Native Application (apk, ipa, zip, xap, …)

Apache Cordova disadvantages

- Applications are slower than native ones
  - especially graphics-intensive applications such as games (unless we write a lot of native code)
- Not fully integrated into the platform
  - may lack the platform-specific look&feel (but you may develop different CSS-based skins for different platforms)
  - lack of pre-built UI widgets, transitions, standard controls – but you can use something else (e.g. jQM) for this task
- For Android development you may use AppLaud plugin, for debugging Weinre or debug.phonegap.com
  - However, pure-HTML parts may be debugged as usual

Apache Cordova project workflows

- Native platform (the original one)
  - We have common platform-independent sources in HTML, JS & CSS, but the project is created for specific platform
  - Needs plugman for each platform
  - Hard to make multi-platform projects
  - Easier to change the lower-level platform details
  - Suitable, when the CordovaWebView is a small part in a larger native application
- Web project (available since Cordova 3.0)
  - A set of wrapper scripts, CLI (esp. cordova command)
  - Multi-platform, easy to add/remove platforms
    - As many mobile operating systems as possible
  - As little platform-specific development work as possible

Installing & using Apache Cordova

- Installation requires Node.js framework
  - Node.js Installation files are at http://nodejs.org/download/ or in your distribution (external) repositories (Linux)
- Cordova installation uses Node.js package manager:
  - npm install -g cordova #(or: sudo npm install -g cordova)
  - Use update instead of install to upgrade the installation
- We can also install plugman – Plugin Manager
  - npm install -g plugman #(or: sudo npm install -g plugman)
- Cordova is integrated in Netbeans, but you can also use Cordova CLI (first creates project dir, use the rest in it):
  - cordova create <project_PATH> [ID [NAME [CONFIG]]]
    - e.g. cordova create example cz.vsb.mor03.Ex Example
  - cordova platform [{add|remove|update} <PLATFORM> | ls]
    - e.g. cordova platform add android
  - cordova plugin [{add|remove} <PATH|URI> | ls | search kw]
    - e.g. cordova plugin add org.apache.cordova.device
  - cordova {build|run|emulate|serve} [PLATFORM]

Apache Cordova config.xml

- The configuration file is placed in the main web project directory or in a subdirectory based on given platform
- Uses Widget specification mentioned earlier
  - extra namespace xmlns:cdv="http://cordova.apache.org/ns/1.0" for widget tag
- The supplied configuration preference may also
  - add fullscreen setting:
    `<preference name="fullscreen" value="true"/>`
  - enforce screen orientation:
    `<preference name="Orientation" value="landscape" />`
    - Values: **default** (both orientations), landscape, portrait
  - disable rubber-band scrolling bounce:
    `<preference name="webviewbounce" value="true"/>`
  - be set to prevent overscrolling on iOS & Android:
    `<preference name="DisallowOverscroll" value="true"/>`
- Some platform-specific preferences may be also included

Executing native code

Apache Cordova offers a system of Native Plugins

- There is a set of basic (core) plugins which should be available for all/most platforms.
- You can create a wrapper JavaScript class to provide missing/additional API.
- The native calls are executed by cordova.exec()
  - e.g. cordova.exec(function(winParam) {}, function(error) {}, "service", "action", ["firstArg", "secondArg", 42, false]);
- Plugin repository has a plugin.xml file in top directory, which describes the plugin
  - platforms, their source files and configuration, …
- The native code differs platform-to-platform, e.g. Android
  - extends CordovaPlugin
  - overrides method: public boolean execute(String action, JSONArray args, CallbackContext cCont) throws JSONException
- You can publish your plugins for others via plugman
  - You have to create an account first.

Apache Cordova core plugins

- Battery Status
- Camera
- Console
- Contacts
- Device
- Device Motion
- Device Orientation
- Dialogs
- *File (System)*
- *File Transfer*
- *Geolocation*
- Globalization
- In-App Browser
- Media
- *Media Capture*
- Network Information
- Splashscreen
- Vibration

Core plugins (A-F)

- *Battery status* – support for battery state
- *Camera* – get a picture from Camera
- *Contacts* – access contacts in device database (search and adding)
- *Device* – provides information about device HW, SW, UUID, manufacturer, …
- *Device Motion* (*Accelerometer*) – returns x,y,z acceleration
- *Device Orientation* (*Compass*) – current compass heading or watching the heading
- *Dialogs* – visual and audible (sound) device notifications.
- *File (System)* – provides W3C File API to read, write and navigate file system hierarchies.
- *File Transfer* – file download and upload

Core plugins (G-Z)

- *Geolocation* – access to the device's location provider (GPS sensor).
- *Globalization* – information & operations specific to the user's locale and timezone.
- *InAppBrowser* – web-browser after the use of window.open
- *Media* – recording and playback of audio files
- *Media Capture* – capture audio, video or image with external application
- *Network Information* – access to the device's cellular and WiFi connection information.
- *Splashscreen* – show&hide the application's splash screen.
- *Vibration* – haptic (tactile) feedback.

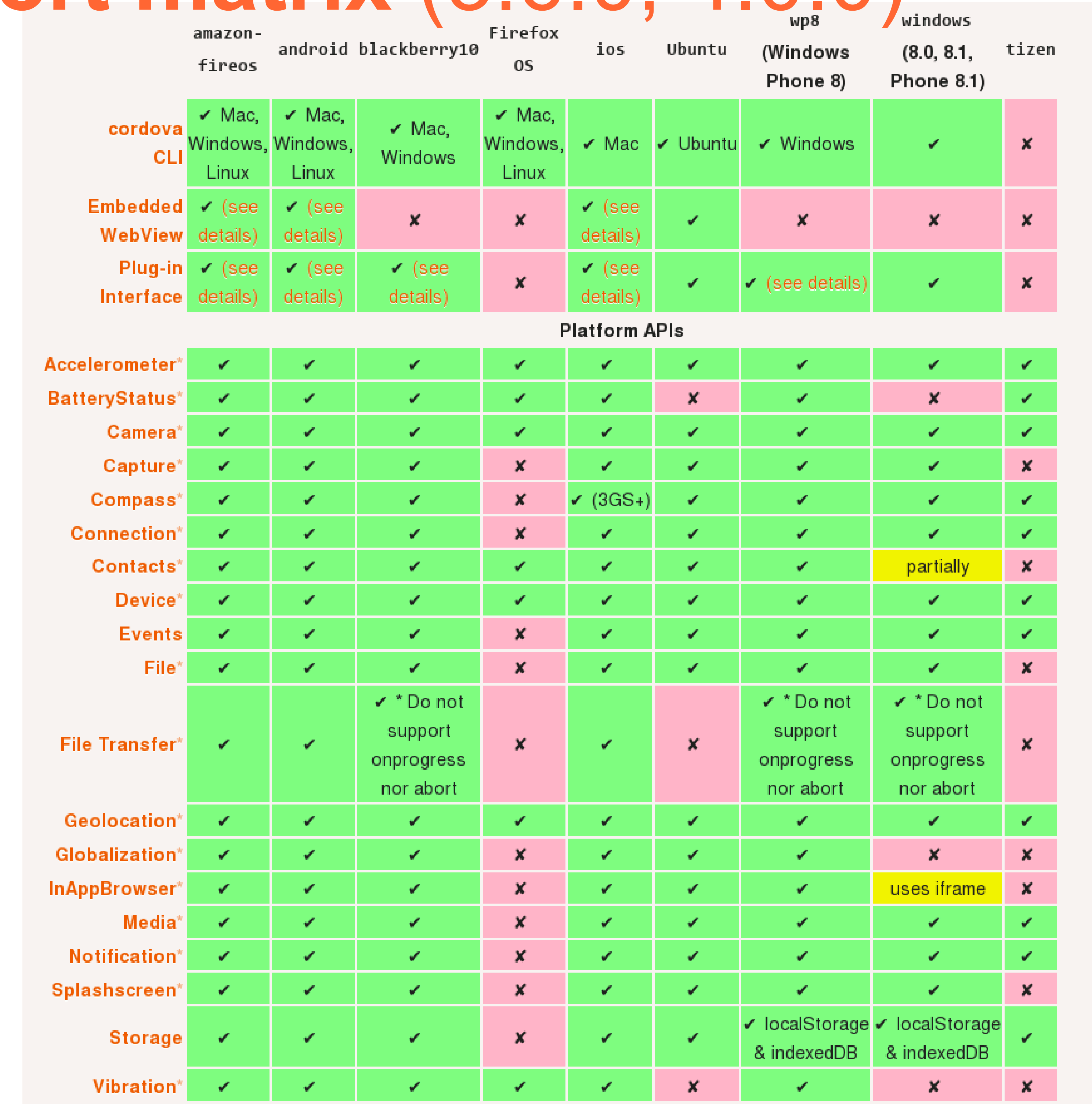
Other plugins & built-in features

Built-in features:

- *Console* – logging to device's console (log)
- *Storage* – local storage and SQL database W3C "Web SQL Database", Indexed DB & Web Store API Specification
- *Lifecycle events* – HW buttons and changes in state
  - deviceready, pause, resume
  - backbutton, menubutton, searchbutton, startcallbutton, endcallbutton
  - volumedownbutton, volumeupbutton
- …

Additional plugins can be found at:
http://plugins.cordova.io/

Support matrix (3.6.0, 4.0.0)



	amazon-fireos	android	blackberry10	Firefox OS	ios	Ubuntu	wp8 (Windows Phone 8)	windows (8.0, 8.1, Phone 8.1)	tizen
cordova CLI	✔ Mac, Windows, Linux	✔ Mac, Windows, Linux	✔ Mac, Windows	✔ Mac, Windows, Linux	✔ Mac	✔ Ubuntu	✔ Windows	✔	✘
Embedded WebView	✔ (see details)	✔ (see details)	✘	✘	✔ (see details)	✔	✘	✘	✘
Plug-in Interface	✔ (see details)	✔ (see details)	✔ (see details)	✘	✔ (see details)	✔	✔ (see details)	✔	✘
Platform APIs									
Accelerometer*	✔	✔	✔	✔	✔	✔	✔	✔	✔
BatteryStatus*	✔	✔	✔	✔	✔	✘	✔	✘	✔
Camera*	✔	✔	✔	✔	✔	✔	✔	✔	✔
Capture*	✔	✔	✔	✘	✔	✔	✔	✔	✘
Compass*	✔	✔	✔	✘	✔ (3GS+)	✔	✔	✔	✔
Connection*	✔	✔	✔	✘	✔	✔	✔	✔	✔
Contacts*	✔	✔	✔	✔	✔	✔	✔	partially	✘
Device*	✔	✔	✔	✔	✔	✔	✔	✔	✔
Events	✔	✔	✔	✘	✔	✔	✔	✔	✔
File*	✔	✔	✔	✘	✔	✔	✔	✔	✘
File Transfer*	✔	✔	✔ * Do not support onprogress nor abort	✘	✔	✘	✔ * Do not support onprogress nor abort	✔ * Do not support onprogress nor abort	✘
Geolocation*	✔	✔	✔	✔	✔	✔	✔	✔	✔
Globalization*	✔	✔	✔	✘	✔	✔	✔	✘	✘
InAppBrowser*	✔	✔	✔	✘	✔	✔	✔	uses iframe	✘
Media*	✔	✔	✔	✘	✔	✔	✔	✔	✔
Notification*	✔	✔	✔	✘	✔	✔	✔	✔	✔
Splashscreen*	✔	✔	✔	✘	✔	✔	✔	✔	✘
Storage	✔	✔	✔	✘	✔	✔	✔ localStorage & indexedDB	✔ localStorage & indexedDB	✔
Vibration*	✔	✔	✔	✔	✔	✘	✔	✘	✘

Information plugins

- Battery Status Events (http://www.w3.org/TR/2011/WD-battery-status-20110915/)
  - Defines 3 events on window: batterystatus, batterycritical, batterylow. The event for each event object contains:
    - level (0-100) and isPlugged (true/false) – is charging?
- Network information (http://www.w3.org/TR/2011/WD-netinfo-api-20110607/)
  - Adds navigator.connection object with network type

    connection.type: Connection.UNKNOWN, .ETHERNET, .WIFI, .CELL_2G, .CELL_3G, .CELL_4G, .CELL, .NONE
- Device – adds a custom device object
  - available after deviceready event
  - properties: device.cordova (Cordova version), .model (e.g. 9800, iPhone 5,1, …), .platform (e.g. Android), .uuid (in most cases NOT IMEI), .version (of device OS, e.g. 4.3, 3.2)
- Dialogs – adds navigator.notification object (native dialogs)
  - notification.alert, notification.confirm, notification.prompt
  - navigator.notification.beep(times) – sound feedback

Contacts Plugin – classes

Following classes are defined in contacts plugin:

- Contact – contains instances of ContactName – name, ContactField[], ContactOrganization[] – organizations, ContactAddress[] – addresses, and other properties:
  - id (globally unique), displayName (end user-friendly name), nickname (casual name), birthday, note
    - ContactField[] based: phoneNumbers, emails, ims (IM addresses), photos, categories, urls
- ContactField – contains type, value & preferred (true/false)
- ContactName – formatted, familyName, givenName (Jim), middleName, honorificPrefix (Dr.), honorificSuffix (Esq.)
- ContactAddress – pref (true/false), type (e.g. home, work), formatted, streetAddress, locality (e.g. Ostrava), region (e.g. Alabama), postalCode, country
- ContactOrganization – pref, type (e.g. home), name, department, title (of contact in organization)
- ContactFindOptions – filter and multiple (returns all finds)
- ContactError – code value is one of the pre-set error codes

Contacts Plugin – methods

- A new contacts property is added to navigator object (navigator.contacts)
  - It has two methods – contacts.create & contacts.find
    - var contact = navigator.contacts.create(Contact);
      - A new local contact object, not yet saved on device
    - navigator.contacts.find(contactFields, contactSuccess, contactError, contactFindOptions) – asynchronous search:
      - var options = new ContactFindOptions();
        options.filter = "John"; options.multiple = true;
        var fields = ["name", "displayName"];
        navigator.contacts.find(fields, onSucc, onErr, options);
  - Contact object has following methods:
    - clone – returns a new, copied, Contact object (deep copy, id is set to null)
    - remove – removes the contact from the device contacts database, error callback if the contact is not present
    - save – inserts a new contact or updates an existing contact in device database

Platform quirks & details: https://github.com/apache/cordova-plugin-contacts/blob/dev/doc/index.md

Accelerometer & Compass plugins

In both cases, the API copies geolocation API approach:

- Acceleration – adds navigator.accelerometer
  - Methods: getCurrentAcceleration(accSuccess, accError), id = watchAcceleration(accSuccess, accError[, opts]), clearWatch(id)
  - Success: we get acceleration object with x, y, z including the effect of gravity (e.g. 0, 0, 9.81) and timestamp field
  - Options: we can specify freq. in [ms], e.g. { frequency: 3000 }
- Compass – compass heading through navigator.compass
  - Methods: getCurrentHeading(compassSuccess, compError), id=watchHeading(compassSuccess, compassError[, opts]), clearWatch(id)
  - Success: we get heading object with magneticHeading, trueHeading (geographic to the North Pole, negative - can't be determined), headingAccuracy (deviation in [°] between reported and the true heading) and timestamp fields
  - Error: we get error object with code being
    - CompassError.COMPASS_INTERNAL_ERR
    - CompassError.COMPASS_NOT_SUPPORTED
  - Options: we can specify frequency (see above) and filter (iOS)

Camera plugin

Camera API – get still image via navigator.camera.getPicture (cameraSuccess, cameraError, [cameraOptions]) as a string

- Options passed in in cameraOptions object:
  - quality (0-100, 100 – best, e.g. JPEG quality)
  - destinationType (0: base-64 string, 1: file://, 2: native URI)
  - sourceType (0: photo library, 1: camera, 2: saved photo)
  - allowEdit (simple image editing, e.g. cropping if true)
  - encodingType (0: JPEG, 1: PNG)
  - targetWidth/targetHeight (scale to output, keep aspect r.)
  - mediaType (for sources from photo library and saved photos, 0:picture, 1: video, 2: all media)
  - correctOrientation (auto-rotate image according to device position when captured?)
  - saveToPhotoAlbum (store captured image),
  - cameraDirection (0: back, 1: front in iOS)
- Success: Based on destination type, the resulting imageData is a string with file URL (file:// or native like capture://) or base-64 string with image data of encoding Type
- Error: error object with message property is returned

Media Capture plugin

- Media Capture API – adds capture object to record media through external file(s), stored on file system
  - Properties: arrays of ConfigurationData: .supportedAudioModes, .supportedImageModes, .supportedVideoModes
  - Methods: capture.captureAudio/.captureImage/.captureVideo (successCB, errorCB [, options])
  - Success: we get a MediaFile instance, which provides:
    - name (without path), fullPath (incl. name), type (mime type), lastModifiedDate, size (in bytes) and getFormatData()
  - Error: we get error object with a pre-defined code:
    - CaptureError.CAPTURE_INTERNAL_ERR
    - CaptureError.CAPTURE_APPLICATION_BUSY
    - CaptureError.CAPTURE_INVALID_ARGUMENT
    - CaptureError.CAPTURE_NO_MEDIA_FILES
    - CaptureError.CAPTURE_NOT_SUPPORTED
  - Options: limit – the number of captured items of given type, audio+video: duration – maximum length of a clip in [s]
- Media plugin – uses W3C media & media capture described in last lecture. Does not adhere fully for capture, will be reworked

Some frequently used utility plugins

- Vibration – adds navigator.notification.vibrate([time_ms])
- SplashScreen – adds splashscreen object
  - Methods: show(), hide()
  - config.xml
    - <preference name="splashscreen" value="file_name" />
    - AutoHideSplashScreen preference (true/false)
- InAppBrowser – a browser window created by window.open(), which is part of Cordova application, but does not have access to Apache Cordova
  - e.g. ref=window.open('http://apache.org', '_blank', 'location=yes');
- Console – changes the way provided console.log() works
- File – implements HTML5 FileSystem API (http://www.w3.org/TR/file-system-api/)
  - Addressed in next lecture

Interesting 3rd party plugins for Apache Cordova 3.x

- SMS plugin – https://github.com/aharris88/phonegap-sms-plugin
  - Allows to send SMS directly without using sms:// URL

- Calendar Plugin (Android & iOS only)
  https://github.com/EddyVerbruggen/Calendar-PhoneGap-Plugin

- Google Maps Plugin (Android & iOS with GM SDK use only)
  https://github.com/wf9a5m75/phonegap-googlemaps-plugin

- Barcode Scanner (Android & iOS only)
  http://plugins.cordova.io/#/package/com.mirasense.scanditsdk.plugin
  - WP8 alternative
    http://plugins.cordova.io/#/package/org.bloxlab.barcodescanner
  - Full JS alternative (no Cordova required, slower)
    https://github.com/LazarSoft/jsqrcode

Interesting 3rd party communication plugins for Apache Cordova 3.x

- Bluetoooth SPP plugin – https://github.com/don/BluetoothSerial
  - Allows to communicate through Bluetooth with standard BT devices emulating serial port through serial port profile

- Bluetooth Low Energy (BLE) Plugins
  - Android 4.3+, iOS 7+ - https://github.com/don/cordova-plugin-ble-central
  - Mainly for HRM, not fully tested. Android 4.3+, iOS 7+, W. Phone 8.1- https://github.com/randdusing/BluetoothLE

- NFC Plugin - https://github.com/chariotsolutions/phonegap-nfc
  - Support for basic NFC tags and communication on Android, Windows Phone and Blackberry together with NDEF messages (encrypted tags are not supported)