Explain the concept of Hybrid Mobile App Development

Native Apps are specific to a given mobile platform (iOS or Android) using the development tools and language that the respective platform supports. Native apps look and perform the best

HTML5 Apps use standard web technologies, like JavaScript and CSS to create cross platform mobile applications. Limitations from this strategy is, session management, no secure offline storage, no access to native device functionality (camera, calendar etc.)

Hybrid Apps make it possible to embed HTML5 apps inside a thin native container, combining the best (and worst) elements of Native and HTML5 apps.

Hybrid mobile applications are built in a similar manner as websites. Both use a combination of technologies like HTML, CSS, and JavaScript. However, instead of targeting a mobile browser, hybrid applications target a WebView hosted inside a native container. This enables them to do things like access hardware capabilities of the mobile device.

Today, most hybrid mobile applications leverage Apache Cordova, a platform that provides a consistent set of JavaScript APIs to access device capabilities through plug-ins, which are built with native code. As a side note, Apache Cordova originally started as a project named PhoneGap. These days, PhoneGap exists as a distribution of Apache Cordova that includes additional tools.

These plug-ins include APIs for accessing the device's accelerometer, contacts, camera, and more. There is also a number of plug-ins that are built and maintained by the developer community at-large. These can be found in the Apache Cordova Plugins Registry. A curated subset of these plug-ins that have been throughly tested, documented, and extended can also be found at the Telerik Verified Plugins Marketplace.

Explain the Pros & Cons of using Hybrid Mobile App Development compared to Native App Development

Features	Native Apps	HTML5 Apps	Hybrid Apps
Navigation	Fast	Slow	Slow
Look and Feel	Native	Emulated	Emulated
Graphics	Native APIs	HTML5, Canvas, SVG	HTML5, Canvas, SVG
Camera	Yes	No	Yes
Notifications	Yes	No	Yes
Contacts, Calendar	Yes	No	Yes
Geo-location	Yes	Yes	Yes
Swipe	Yes	Yes	Yes
Pinch, Spread	Yes	No	Yes
Connectivity	Online and offline	online	Online and offline
Performance	Fast	Slow	Slow
Development Skills	Objective-C, Java, C#	HTML5, CSS, JavaScript	HTML5, CSS, JavaScript
Distribution	App store	Web	App store

Hybrid / PhoneGap:

- For budgetary limitation, hybrid app may be a better choice.
- If there is a need to quickly develop the app, hybrid app may be a better choice.
- If the app is simple, does not have large animations, does not have lots of clicks and does not require lots of native user interaction, hybrid will be a better choice.

Native:

- If the requirement is to create the best user experience, native development is a better choice.
- For companies with sufficient allocated budget that are planning to build and maintain large app projects and do not want to worry about limitations for implementing new technologies, and support, native is a better choice.

Item	Native has Advantage	Hybrid has Advantage
Cost		✓
Platform Independent		✓
Quicker development		✓
User Experience	✓	
No Limitations	✓	
Support	✓	
Tools and Debugging	✓	

Explain how and why it is possible for a Hybrid Application to access native phone devices like location, calendar etc.

Cordova formerly called as Phone Gap is a platform to build Native Mobile Applicatons using HTML5, CSS and Java Script.

The core of Apache Cordova applications use CSS3 and HTML5 for their rendering and JavaScript for their logic. HTML5 provides access to underlying hardware such as the accelerometer, camera, and GPS. However, browsers' support for HTML5-based device access is not consistent across mobile browsers, particularly older versions of Android. To overcome these limitations, Apache Cordova embeds the HTML5 code inside a native WebView on the device, using a foreign function interface to access the native resources of it.

Apache Cordova can be extended with native plug-ins, allowing developers to add more functionalities that can be called from JavaScript, making it communicate directly between the native layer and the HTML5 page. These plugins allow access to the device's accelerometer, camera, compass, file system, microphone, and more.

Explain using an example how you Hybrid Application communicates with a backend and how CORS problems were solved (if any)

The easiest way to handle the CORS problem is to ultimately ask your API provider to allow all hosts. However, this isn't always an option.

Using the Angular constant and the replace module will give us a happy medium, in which we can work around CORS