The development of hybrid mobile applications with Apache Cordova

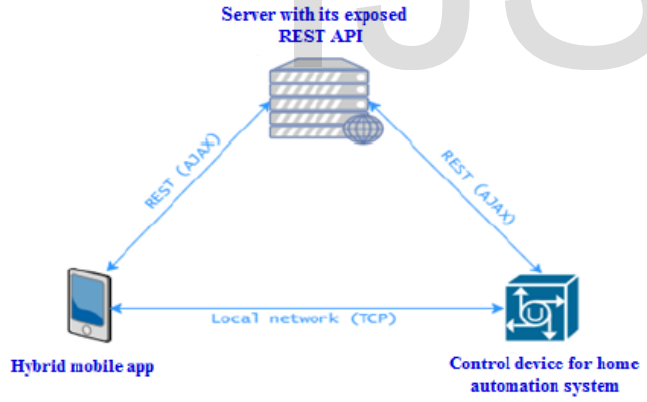
Shubham Sutar,Prof. Nirali Verma

Abstract— Native mobile applications are applications developed by using the SDK and programing language specific to the mobile platform. The key limitation of those mobile applications is that the inability to transfer applications to a different platform, without writing the applying from scratch. This situation has led to new requirements, like tools that enable generation of mobile applications with one codebase. This paper presents the simplest way of generating such applications using the Apache Cordova. It allows developers to implement applications on multiple platforms, employing a single codebase. Applications are executed in an exceedingly platform-specific native container.

Keywords -- Cordova, mobile application, hybrid, Ionic, Angular JS,Apache ,Methods.

1 Introduction

The rapid development of smart devices affects the development of many applications, which are used in today's lifestyle. Many companies are showing more interest in developing their own mobile applications, as a way to increase their business process productivity. These applications are platform-specific, thus requiring a separate application for each platform. This requires a lot of resources and represents the motive for hybrid application development. All devices uses single hybrid application, regardless of what type of



This work was partially supported by the Ministry of Education, Science
and Technological Development of the Republic of Serbia, under grant
number: TR32014.

device or platform it is. (Smartphone or tablet). Hybrid applications or cross-platform applications are a flexible solution [1]. In this report we will write about Apache Cordova tools that we used to develop cross-platform or hybrid applications. Benefit from this solution will have many companies which developing their own mobile applications. It can implemented in all the firms where native applications can be applied. In other words, they can replaced native applications. Cordova enables software developers to create mobile applications using CSS3, HTML5 and JavaScript code. CSS3 is used for user-interface and JavaScript for application logic.The results are hybrid applications that integrate with the platform-specific Cordova environment present on the device [2]. In this paper we describe the development and testing of a complex hybrid application.

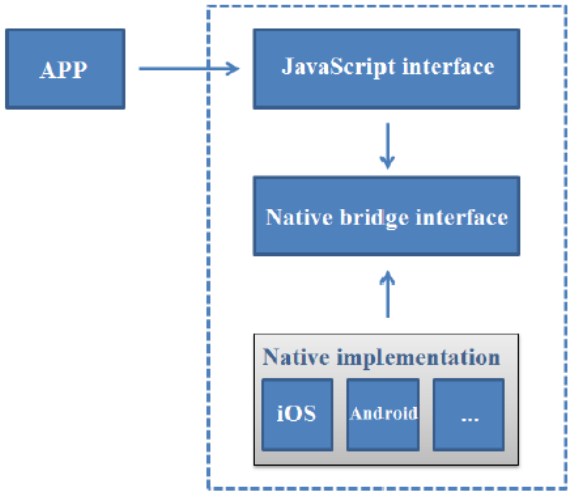
2. Methods:

The performance of applications, functionalities and user experience are the main challenges when developing a mobileapplication. user experience varies from platform to platform. Different platforms instill different habits to the users, which is why creating a universal user experience for all platforms is a challenge. We rely on Cordova and Ionic to adapt the user interface to the platform on which the hybrid application is executed. Application development is done in Visual Studio environment. Visual Studio includes tools for Apache Cordova. If Cordova is not installed on the device, tools for the Apache Cordova will install Cordova on device beforeinstalling an application. With the Cordova installed on thedevice, any Cordova-based hybrid application can be installedand run.

- *Shubham Sutar is currently pursuing masters degree program in Information Technology in Mumbai University, India, PH-919326176534.. E-mail: sutarshubham23@gmail.com*
- *Nirali Verma is currently a head of depatment in PHCASC, Rasayan, India, PH-7020802098. E-mail: Niraliverma@mes.ac.in*

Hybrid applications architecture and usage of Apache Cordova tools. Application life cycle, regarding cycle of its execution and usage, can be seen. Angular JS is used for application logic and Ionic for application look and feel. Cordova plugins are one of most important part of this application. They allowed native resources similar camera, contacts, accelerometer, file system, geolocation etc. Webview is native container for run application. native container allowed application execute on different mobile platforms.

Cordova is compatible with two frameworks: Angular and Ionic. Angular is more focused on organizing the applying logic and architecture, counting on the MVC pattern. It also provides modularity, through services and controllers. Ionic framework is targeted on the planning and interface interaction applications. Ionic is a wonderful fit with Cordova and Angular framework, and simplifies an oversized a part of the application's front-end. the total potential of Ionic is manifested with its usage within Angular [3].Hybrid applications aren't completely native application because they run in within the device`s Webview as an online application. they're not completely sand-boxed as they need access to native APIs of the device, through appropriate JavaScript APIs that are injected by Cordova. The Crosswalk plugin can improve the user experience on older version of Android that use an outdated Webview. It enables better application performance. It`s supported Google Chromium, thus improving the performance of HTML, CSS and JavaScript. JavaScript is executed and run within the Webview component [4]. Cordova provides extension of JavaScript capabilities with its plugins. This extension provides a universal API that abstracts away platform-specific APIs required to access the device's hardware. Everything that the standard API can`t provide, is resolved through the utilization of appropriate plugin. Plugins are an integrated a part of Cordova system. They provide an interface for Cordova and native components, in order to determine mutual communication and relationship with the standard API of the device. Plugins contain an API, that maps common functionality to the platform-specific calls [5].In essence, this hides the varied native code implementations behind a standard interface JavaScript. When creating Cordova project, plugins aren't included by default. Any plugin that we wish to use (core or custom) must be explicitly added. Apache Cordova supports a group of default plugins called core plugins. These plugins allow us to access device capabilities like the battery, camera, contacts, storage, etc. In addition to core plugins, there are independent plugins (developed by third party members, the Cordova community etc), which give additional features that aren't necessarily available on all platforms. Such plugins are called custom plugins. Cordova allows us to develop a custom plugin for communication between Cordova and other native components [6]. Fig. 3. Plugin architecture. JavaScript, native bridge interface and native implementation make structure of plugin architecture. Application use APIs in JavaScript interface to realize access to the plugin. Cordova allows the online application to access the device's capabilities outside of the browser sandbox. Plugin.xml contains information about the trail to the JavaScript and native source files for supported platforms. Depending on the parameters passed to the strategy from JavaScript interface, enables access to the acceptable native functionalities.. JavaScript interface is probably the foremost important a part of the plugin. For the structure of JavaScript interface there are no restrictions. Communication with native API requires a call to appropriate cordova.exec methods. Method cordova.exec acts as a proxy between applications and native interface, depending on the passed arguments calls appropriate native method. For android devices, the execute method is employed and for iOS devices, action method is used[7]. The plugin repository must feature a plugin.xml file. Plugin.xml file is an XML document that defines the plugin and contains instructions for adding plugin in config.xml file.



Plugin architecture. JavaScript, native bridge interface and native implementation make structure of plugin

architecture.

3.Conclusion

Hybrid or cross-platform application development that was described in this paper is meant to contribute the development and advancement of mobile applications. And above all to propose a solution the problem of developing applications for different platforms.After completing the development and testing of hybrid applications we can compare the advantages and disadvantages of these applications compared to native apps. The advantage is generating applications for multiple platforms from a single code base. But test results confirmed that optimization is the biggest problem in mobile applications of this type. For this reason, achieved inferior performance compared to the native application. Apache Cordova tools are suited for hybrid applications that are not very complex, or require less functionality.For complex applications, native applications have priority.But Cordova is a big step forward and an advancement in mobile technology.

4.References

[1] Hui. Ng Moon, Chieng. Liu Ban, Ting. Wen Yin, M. HasimahHj, H.
Arshad and M. Rafie, "Cross-platform mobile applications for android
andiOS", WMNC, pp. 1-4, 2013.
[2] Z. Qing, L. Ying, P. G. Yuan and L. Z. Sheng, "Music Player Based on
the Cordova Cross-Platform," Applied Computing and Information
Technology/2nd International Conference on Computational Science
and Intelligence (ACIT-CSI), 2015 3rd International Conference, pp.
451-453, July. 2015.
[3] ArvindRavulavaru, ”Learning Ionic,” July 2015.
[4] John M. Wargo, "Apache Cordova API Cookbook," July 2014.
[5] John M. Wargo, "Apache Cordova 4 Programming," April 2015.
[6] "Plugin Development Guide - Apache Cordova", [online], Available :
https://cordova.apache.org/docs/en/latest/guide/hybrid/plugins/index.htm
l
[7] “Explore the Apache Cordova Plugin Architecture | Code | Telerik
Platform”, [online], Available:
http://docs.telerik.com/platform/appbuilder/cordova/usingplugins/
working-with-plugins
[8] "Plugin Specification - Apache Cordova", [online], Available:
https://cordova.apache.org/docs/en/4.0.0/plugin_ref/spec.html
[1]