



# Real-time Social Media Multi-Platforms Application based on Firebase in NFV Environment

Jinsul Kim<sup>1</sup>,  Akm Ashiquzzaman<sup>1</sup>, Van Quan Nguyen<sup>1</sup>, Sang Woo Kim<sup>1</sup>, Kwangki Kim<sup>2</sup>

<sup>1</sup>*School of Electronics and Computer Engineering, Chonnam National University, Yongbong-ro, Buk-gu, Gwangju 500-757, Korea*

<sup>2</sup>*School of IT Convergence Korea Nazarene University, Cheonan-si, Korea*

zamanashiq3@gmail.com,  quanap5@gmail.com, swmax88@gmail.com, k2kim@kornu.ac.kr, jsworld@chonnam.ac.kr

**Abstract**— In recent times, practicality of web applications has become more reliant upon big-data orientated materials such 4K videos, hi-def. resolution images, lossless audios and massive texts. Structured Query Languages (SQL) faces compatibility issues with large scale databases. Because of this data storage problem, NoSQL databases are used for storing big-data. NoSQL databases have been recently gaining traction with many options such MongoDB, CouchDB, Redis and Apache Cassandra. One of the major restrictions companies, enterprises and developers encounter during developing an application is multiplicative cost of building a native programing across different platforms. Besides, network Function Virtualization (NFV) plays a vital role for providing services for utilizing such applications in larger and more effective scale. Hence, in this paper, we discussed our main motivation behind selecting Ionic Framework, a hybrid system for rapid development real-time application based on Firebase in the NFV environment cooperating with Mobile Edge Computing (MEC). As a result, this approach provides comparatively flexible features.

**Keywords**—Multi-platform, Firebase, real-time application, Social Network, Hybrid Mobile Application, Ionic Framework, Big Data, NFV, MEC.

## I. INTRODUCTION

Hybrid Mobile Application (HMA) enables web technologies such as HTML5, CSS and JavaScript (JS) to execute in Native Mobile application platform. Native mobile applications are the most frequent form of medium used in modern web based technologies [1]. The main reason behind choosing the Native approach for rapid mobile application development and implementation is for their platform independence and adaptive compatibility. Usual approach for developing application for specific platforms are essentially building it based on native programming languages that the such platform accepts. For example, Swift and Objective-C is used exclusively for iOS apps and Java and Kotlin for native Android apps. Also, Native apps are developed and built exclusively using the individual Integrated Development Environment (IDE) for the given operation systems. Native applications are very fast and responsive performance since they are built for that specific platform. Aside the Native based approach, the Web platforms are similar to web application deployed on computer, but they are normally accessed through the mobile browser. The core technologies used in developing web based application includes HTML5, CSS and JavaScript [2]. A hybrid mobile application is seen as a

consequential derivative from heterogeneous sources consisting of Native and Web apps. It works across platform and behave like native apps. A hybrid app is essentially a combination of a native app and a web app. We can install it on devices as a native app but it is actually a web app. In broader tram, this type of apps is built with the same web apps based on JavaScript, HTML and CSS and run in Web view.

Database is an organized collection of data [6]. The fundamental idea behind creating database is to store data systematically and fetch easily and in relative time when querying. Every application like Android, iOS, or Web application has its own database. We have discussed many features of firebase that are not available in most of other database backend services. Even though most of the important features can be found in most platform, there is no API that holds all features in one. Firebase is a real-time database which stands above all the traditional database services like SQL, SQLite, shared preference etc. Firebase can be integrated to any app in short time and app using firebase requires least development time to deploy. Firebase requires step by step processing for the successful application run [3]. Firebase is now provided by Google, Inc. as a service. This ensures reliability and new features in future. In this paper, features of Firebase API as well as instructions for adding firebase to the hybrid android application using Ionic framework is described in brief details.

Network Function Virtualization (NFV) is a relatively new idea for telecommunication service provisioning. The idea of implementing all the functionality in virtual space gives service provider an opportunity to scale up and improve service [7]. With new modules of Mobile Edge Computing (MEC), this technology can have a potential to revolutionize the any hybrid mobile application performance [8].

The rest of the this paper is organized as follows: Section II briefly discusses hybrid mobile application based on Ionic framework and Firebase, one NoSQL solution. The step by step description for installation and configuration is content of Section III. Then, the demonstration and performance are presented in the Section IV. Finally, Section V contains discussion and conclusion.

## II. LITERATURE REVIEW

### A. Hybrid Mobile Applications

There are three approaches [4] to develop mobile applications, these are called Native approach, Web and Hybrid mobile approaches shown as Fig. 1. Each approach has its features as well as including both advantages and disadvantages.

We experienced emergence of various hybrid mobile application using various technologies so far. This has the potential for making the native method obsolete [1]. The hybrid approach based on Ionic framework [5] is discussed below.

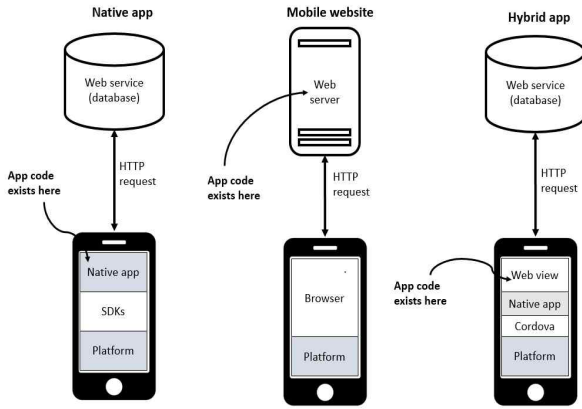


Fig. 1. Three Approaches for Developing Mobile Application

### B. Firebase

This paper focuses on the application of Firebase [3] with Hybrid mobile application. In the previous part, we discussed how Hybrid mobile application can be developed using Ionic framework technologies. In this section, Firebase solution from Google will be main objective. These following services are available in Firebase:

**Real-time Database:** The Firebase Real-time Database is a cloud-host NoSQL database that allows application data to be synchronized across clients and stored on Firebase in real-time. Real-time syncing makes it easier for users to access their data from any devices, even in web or mobile. Real-time database also helps user collaborate with one another. When user goes to offline, the Real-time Database SDK use local cache on the device to serve and store changes. When the devices come back online, the local data is automatically synchronized. The Real-time Database can also integrate with Firebase Authentication to provide a simple and intuitive authentication process.

**Firebase Authentication:** Firebase Authentication support social login provider like Facebook, Google, Github and Twitter. It provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate user to your app. Firebase Authentication makes building secure authentication systems easier, while also improving the sign-in and onboarding experience for end users.

**Firebase Cloud Messaging (FCM):** Firebase provides a reliable and battery-efficient connection between your server and devices that allows you to deliver and receive

messages and notifications on iOS, Android, and the Web application, that is also formerly known as Google Clouds Messaging (GCM).

**Firebase Storage:** It facilitate easy and secure file transfer regardless of network quality for the Firebase apps. It is backed by Google Cloud Storage which is cost-effective object storage service. The developer can use it to store images, audio, video, or other user-generated content.

**Firebase Test Lab:** It is a cloud-based app-testing infrastructure. Firebase Test lab provides automatic and customized testing of your app on virtual and physical devices hosted by Google. With one operation only, you can test your Android or iOS app across a wide variety of devices and devices configurations, and see the results like logs, videos, and screenshots-in the Firebase console as well as to find hidden problems throughout the development life.

**Firebase Crash Reporting:** plays an indispensable role for diagnosing and fixing problems in your mobile app. It creates detailed report of the bugs in your app triaged by severity of the impact on your user. The errors are grouped into clusters of similar stack traces.

## III. DEVELOPPING HYBRID MOBILE APPLICATION ON FIREBASE

### A. Proposed System Overview

The main idea for implementing NVF into HMA to reduce stress and overhead from the service. Specifically, the proposed NVF system is shown in the Fig. 2. In the NVF environment, we can divide the media platform into three separated services such as the firebase for database handling, the control server acts as the main service head and streaming server can be classified as the transmission service. As all this services are being implemented virtually, the main Mobile Edge Computing components can be scaled by with services in need. So the rollout of the HMA can be scaled up by increasing and decreasing as needed. The whole idea of operational flow for the idea application in shown in Fig. 2.

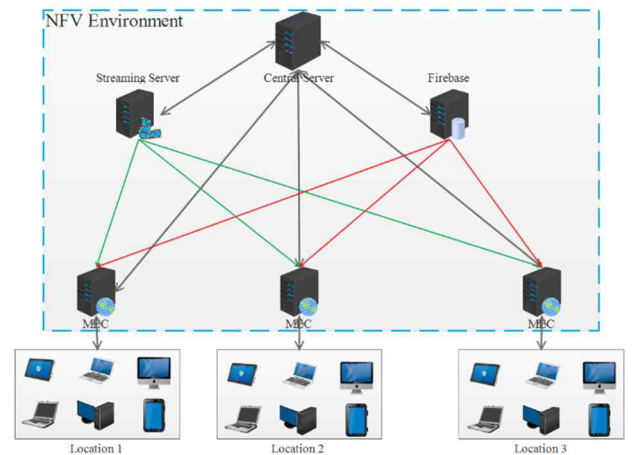


Fig. 2. Concept of NVF incorporated in Hybrid Mobile Application

An experimental setup for such application to run in smaller scale and using the utilities to later maintain and

rollout in larger option is greatly investigated in this paper. The idea is then simulated for an experimental application and various aspects for this application is explored in later sections.

### B. Application Design Flow

This section, we aim at providing details of the research process to develop a hybrid mobile application. The application flow is depicted as Fig. 3. There are four main processes including process of design, building, testing and running the system.

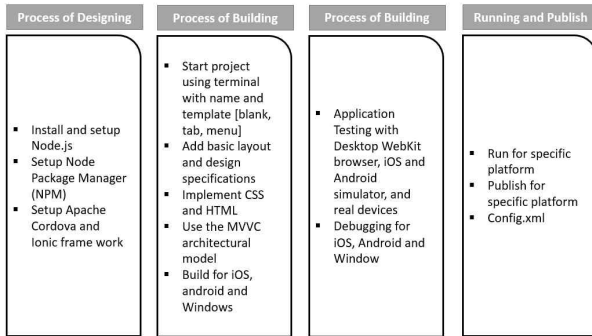


Fig. 3. Methodology for Developing Hybrid Mobile Application

### Installation of Components

The steps needed to carry installation of the components are considerably easier than other mobile approaches. Node.js was installed from official website. The .msi and .pkgfile were installed on windows and macOS respectively. We check to see if Node.js was installed correctly, this was done on terminal and command prompt utility of both OS by typing 'node-v'. This would print the version number. Node Package Manager (NPM) is package manager or code library that extends the functionality of Node.js by adding useful features. These commands were implemented in order to carry on the process of installation and building. We followed the general process as follows:

#### Installing Ionic

- `install node`
- `npm install -g cordova`
- `npm install -g ionic`

#### Create mobile App project

- `Ionic start`
- `Ionic start "app-name" "template"`

#### Run the project

- `run ionic serve`
- `ionic serve -l` (l denotes ionic lab to display android, window and iOS on browser)

#### Add and remove mobile platform –iOS or Android

- `ionic platform add "platform_name"`
- `ionic platform rm "platform_name"`

#### Build platform

- `ionic build "platform_name"`

#### Run mobile platforms

- `ionic run "platform"`

#### Add and remove plugin

- `ionic plugin add "plugin package name"`
- `cordova plugin add "plugin package name"`
- `ionic plugin rm "package name"`

#### To run existing project

- `npm install`
- `bower install, gulp install` (built-in tool that automates build process)

### C. Implementing Dating Application

#### Software specification:

The application was build and test on multi-platform for Android and iOS device.

#### Code design and implementation:

Our target is developing a dating application on which there are social users with their registered information could make friends, messages and communicate to each other's. User could publish their photo via upload avatar or gallery and update their profile including name, birthday, level of education, hometown. Even they can require other information about characteristic, favorite food, type of blood, religion or weight and height. All data will be used for recommendation algorithm or matching algorithm that help user to find the matched friend.

#### Using Firebase features in Hybrid mobile application:

##### 1.1 Authentication

We build a hybrid mobile application look like dating application with require accounts that is registered use email or provider like Facebook account as shown in Fig. 4.

Search by email address, phone number or user UID			
Identifier	Providers	Created	Signed In ↓
quanap5@gmail.com		26 Aug 2018	26 Aug 2018
nobles3v2@gmail.com		26 Aug 2018	26 Aug 2018
nobles2v2@gmail.com		26 Aug 2018	26 Aug 2018
nobles1v2@gmail.com		26 Aug 2018	26 Aug 2018

Fig. 4. Authentication on Firebase

##### 1.2 Database

Fig. 5 is a descriptive JSON of the Firebase data structure of hybrid mobile application. From root of structure, there are eight child including *buddychats*, *friends*, *interests*, *news*, *payments*, *report*, *requests*, *users*. They are containing information about message (id sender, id receiver, content of message, timestamp of messages), friend relationship between user in network of hybrid application, list of user who viewed current user, list of user who give a like to current user, list of new member who has just registered to join in network, information about payment (token, amount, method of payment, status of payment and other information of transaction), detailed report about number of users on male, female and matched

couple, profile of active user (name, birthday, age, gender, education, ...) respectively.



Fig. 5. Firebase Data Structure

### 1.3 Searching algorithm

Since target of hybrid application is find the matching user who is the most similar to current user for recommendation system. These similar users will be appeared on the home page first. For our project, we implement two searching mode by manual and grouping by k-mean clustering algorithm.

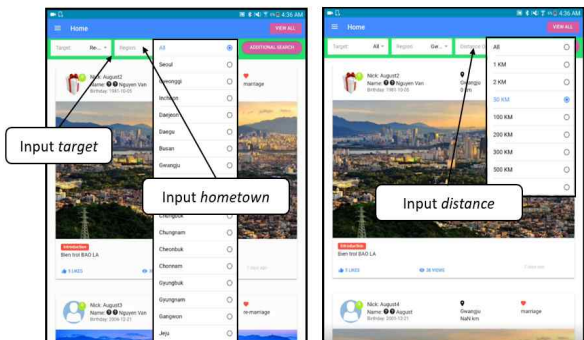


Fig. 6. Searching GUI On Hybrid Mobile Application (Input Value Are Selected Via Option Selection)

Searching algorithm refer to information of hometown and target (marriage or remarriage).

## IV. EXPERIMENTS

The main idea for this experiment is to see the NfV environment and how social media application acts in it. Moreover, HMA deployment in the NfV environment with the MEC functionality is also observed. In the main experiment, the application is rolled out partially to observe the behavior of network operation. As mentioned before, Hybrid approach based on Ionic framework is used for this project. This approach helps us in developing the hybrid

mobile application for multi-platform in the short time. Fig. 5 is the simulated results on the browser using commands “ionic serve -l” and local machine on <http://localhost:8100/ionic-lab>. Fig. 6 and Fig. 7 illustrates various instances of the application.

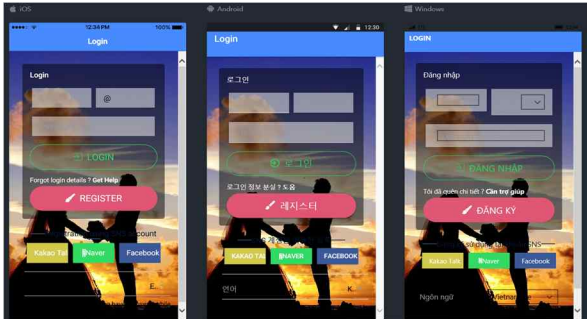


Fig. 6. Multi-platform for Login page with English, Korea and Vietnamese language respectively

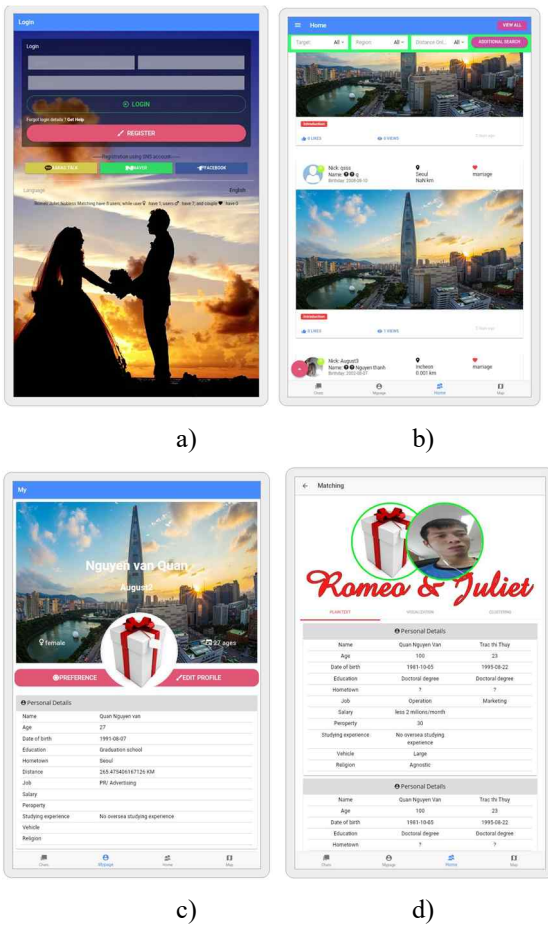


Fig. 7. GUI: a) Login Page, b) Home Page, c) My Profile Page, d) Matching Page

## V. CONCLUSION

In this paper, we have explored the main fundamentals of network function virtualization and mainly the implementation of this module for the mobile applications. Introduction of mobile edge computing gave the mobile application to flourish high performance in deploying hybrid mobile applications. We proposed a new framework for NfV environment to deploy rapid mobile application for making. We have presented the fundamental

principle behind developing mobile application. Later the proposed model was partially implemented developing a mobile application. This social media application has the mobile edge computing (MEC) adaptability for future improvements. Specially, we have focused on hybrid mobile application approach combining with Firebase solution to deploy a real-time social network mobile application. The hybrid approach allows a single code base work for multi-platforms such as iOS, Android and Windows. As a result, the cost and developing time will be reduced significantly. This is especially suitable for start-up companies for publishing their products to customer base in minimal time.

#### ACKNOWLEDGMENT

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program(IITP-2018-2016-0-00314) supervised by the IITP (Institute for Information & communications Technology Promotion). Besides, this research was supported by the IT R&D program of MSIT (Ministry of Science and ICT), Korea / NIPA (National IT Industry Promotion Agency) 12221-14-1001, Next Generation Network Computing Platform Testbed. Furthermore, this research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (MEST) (Grant No. NRF-2017R1D1A1B03034429).

#### REFERENCES

- [1] Dunka Bakwa, Emmanuel Edim & Oyerinde Dantala. "Hybrid mobile application based on ionic framework technologies". International Journal of Recent Advances in Multidisciplinary Research. 04. 3121-3130. 2017.
- [2] Gawley, Rachel & Barr, Jonathan & Barr, Michael. "Native to HTML5: A Real-World Mobile Application Case Study." 188-206. 10.1007/978-3-642-32320-1\_13.
- [3] Firebase. (2015). *Firestore*, Google Inc.. [online] Available at: <http://firebase.com> [Accessed 4 Sep. 2018].
- [4] Anibal A., A (March 20, 2017). Hybrid Mobile Development. [Web log Post]. Work with The Best. Retrieved June 19, 2017 from <https://workwiththebest.intraway.com/blogpost/mobile-hybrid-development>. [Accessed 4 Sep. 2018].
- [5] Ionic Framework. (2018). *Ionic Framework*. [online] Available at: <http://ionicframework.com> [Accessed 4 Sep. 2018]
- [6] Bernstein, Philip A., Vassos Hadzilacos, and Nathan Goodman. "Concurrency control and recovery in database systems." (1987).
- [7] Mijumbi, R., Serrat, J., Gorricho, J. L., Bouten, N., De Turck, F., & Boutaba, R. "Network function virtualization: State-of-the-art and research challenges". IEEE Communications Surveys & Tutorials, 18(1), 236-262, 2016.
- [8] Beck, Michael Till, Martin Werner, Sebastian Feld, and S. Schimper. "Mobile edge computing: A taxonomy." In *Proc. of the Sixth International Conference on Advances in Future Internet*, pp. 48-55. Citeseer, 2014.