





## 7.2 Network Carrier Statistics

This information will help developers know which network carrier is preferred by their application users. Carrier billing, wherein the in-application purchases are directly build to the network carrier, is a popular practice today. By knowing which carrier is most popular the developer can collaborate with them and come up with some schemes which will benefit both parties.

## 7.3 OS Version Statistics

OS stats will help the developers understand which android version is most popular among the users. If a significant number of users use an older version of an operating system, like Gingerbread, updating the application with features not supported by Gingerbread will lead to reduction in the number of users.

## 7.4 Mobile Model Statistics

Android is a fragmented market. A developer may find it difficult to analyze the popularity of his application based on the make of the phone. This feature will help the developer make sure the devices that use their applications the most, are always compatible with the updates.

## 7.5 Application Version Statistics

If many users use an older version of an application, then the developer cannot remove support for that version. To take call on which versions to support, the developer can use this feature.

## 8. Data Visualization for Web Interface

For an analytics tool, it is very important to show the data in an interactive and understandable format. Using RoloKnow's powerful data visualization tools, a developer will be able to compare different statistics using a donut chart, understand the regions where the application is working using an interactive world map and also use histograms for other analyses [5].

## 9. Architecture

RoloKnow works in four major steps:

- 1) Smartphones to MongoDB:** All the required user data that the developer needs is sent through the devices to MongoDB through a web server. Here the data is in its original form [6].
- 2) MongoDB to Hadoop:** Selective data from MongoDB is sent to Hadoop for processing [7]. Here, Pig is used to carry out the MapReduce tasks. Pig is a high-level platform for creating MapReduce programs used with Hadoop. The language for this platform is called Pig Latin [8].
- 3) Hadoop to MySQL:** The processed and meaningful data needs to be stored. For this RoloKnow uses MySQL. All the

churned data is stored here for further use.

**4) MySQL to Web Interface:** Numbers are difficult and sometimes monotonous to analyze. That is why, all the meaningful data is shown to the developer using RoloKnow's web interface with powerful data visualization capabilities [5]. This is the final stage of the process and the developer gets meaningful results. Figure 2 below shows the architecture of RoloKnow.

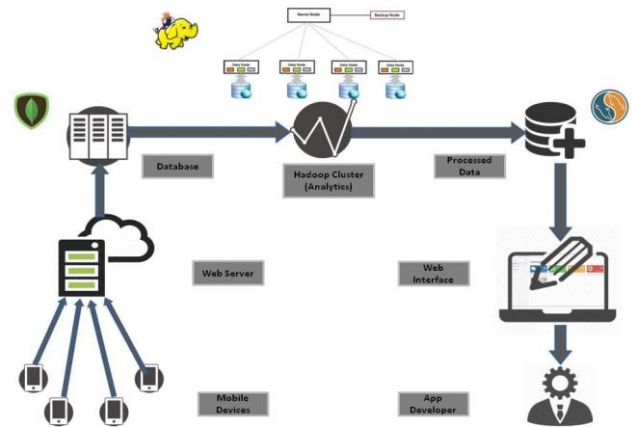


Figure 2: Architecture of RoloKnow

## 10. Implementation

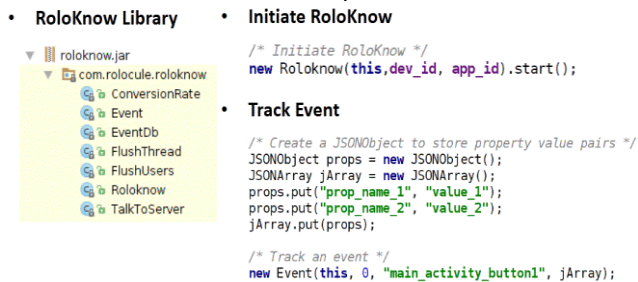
### 10.1 Client-side Library

Client Library- A library, 'roloknow.jar', will be provided to the developer, to deploy in the application along with the developer id and an application id. The task of this library is to track event data, store it locally and send to MongoDB. Flushing of data has been implemented using a thread. If the user is not connected to Wi-Fi, the thread is put to sleep for larger amounts of time to save user data [4].

Steps to use the library in android studio-

- 1) Copy 'roloknow.jar' to 'app/libs' directory of your application.
- 2) Right Click and select 'Add to Library'.
- 3) Open build.gradle file and check if 'compile('app/libs/roloknow.jar')' is added under dependencies.
- 4) Perform a clean build.
- 5) Now, this library can be used with a few basic function calls. RoloKnow (this, dev\_id, app\_id).start() will initialize the library with the developer id and the application id. It will also start a thread to carry out flushing.
- 6) Event can be tracked using the function call given below in Figure 3. To add any additional data in the form of name value pairs, a json array can be passed as an argument to this function call.

## Android Client Library



**Figure 3:** Steps to include RoloKnow library in an application

$$DAU = \sum_{i=0}^{23} N \quad \dots(3)$$

where, **N** is the new user count for an hour,  
**H** is a list of devices active in a particular hour,  
**D** is a list of devices that have previously accessed the application on a particular day under observation,  
**U** is the count of hourly active devices,

(1) gives the hourly active user count, (2) gives the count of new users for a particular hour in a day and (3) gives the daily active user count for a day (24 hours).

## 10.2 Hadoop Analytics

RoloKnow uses Pig in Hadoop to perform analytics on event data. It calculates hourly, daily, weekly and monthly active users. In addition, it performs analytics to find correlation between events and also finds property statistics as mentioned above.

### 10.2.1 Algorithm for DAU and HAU calculation

- 1) Load the latest event details from MongoDB into P;
- 1) Group P by (app\_id, developer\_id, device\_id) and calculate latest timestamp for each group and store into Q;
- 2) Extract the data for the current hour from Q and store into 'hour\_data';
- 3) From 'hour\_data' generate unique count of 'device\_id's for every (app\_id, developer\_id) and store into 'unique\_count';
- 4) Load 'last\_activity', which is a data structure maintaining most recent timestamps for unique (app\_id, developer\_id, device\_id) into R;
- 5) Extract the data from R for the current day and store it in 'day\_data';
- 6) Perform a natural join on 'hour\_data' and 'day\_data' and group the result by (app\_id, developer\_id). Store the result into 'common\_data';
- 7) For each (app\_id, developer\_id) in 'common\_data' generate count of 'device\_id's and store into 'common\_count';
- 8) Subtract count of 'device\_id's in common\_count from unique\_count to get the new user count for the hour and store the result into 'FINAL'

'FINAL' will have developer\_id, app\_id, date, hour, unique\_user\_count, new\_user\_count. 'unique\_user\_count' is the count of unique users for a particular hour. Sum of new\_user\_count through the hours of the day is the daily active user (DAU) count. This data will be stored in MySQL to be used by the web interface.

### 10.2.2 Mathematical Representation

The above algorithm can be mathematically represented as,

$$U = [H] \quad \dots(1)$$

$$N = U - [(H \bowtie D)] \quad \dots(2)$$

## 11. Conclusion

RoloKnow is a product which can cater to the entire developer base and would become an integral part of application development. With the gap between the users and developers bridged with RoloKnow, the cycle of application development will surely become easier and faster. It is platform independent; therefore, any application developer would feel the need to use RoloKnow. There are going to be millions of users and their billions of events. RoloKnow will be a one-step solution for this.

## References

- [1] Oskar Wirén, "Data Driven Development for Mobile Applications", Teknisk- Naturventenskaplig, pp. 43-45, August 2013.
- [2] Website: <https://www.mixpanel.com/retention>
- [3] Website: <https://www.mixpanel.com/funnels>
- [4] Website: <https://developer.android.com>
- [5] Website: <http://docs.amcharts.com/3/javascriptcharts>
- [6] Website: <http://docs.mongodb.org/manual/>
- [7] Website: <https://hadoop.apache.org/docs/stable/>
- [8] Website: <https://pig.apache.org>