22nd IMEKO TC4 International Symposium & 20th International Workshop on ADC Modelling and Testing
SUPPORTING WORLD DEVELOPMENT THROUGH ELECTRICAL&ELECTRONIC MEASUREMENTS
IASI, ROMANIA, September 14-15, 2017

Mobile Applications for Smart Notifications

Calin Rugina, Olga Plopa, Alexandra Iulia Vizitiu

"Gheorghe Asachi" Technical University of Iaşi, Iasi, Romania, calin@sqnp.net

***Abstract* – The central part of the mobile experience of the consumer is the notification system. Through mobile applications can be delivered rich and contextual content to any person who has installed a specific application. The smart notifications are the result of a complex, logical reaction in some specific situations, look and feel more like a note from your assistant or another human being. A smart notification received on a mobile device can trigger an action on device in background or use human interaction. Start from this fact the mobile device became from a receiver an intelligent tool.**

Keywords – mobile application, smart notification, push notification, text messaging, sensors, internet of things

I. INTRODUCTION

Today the app world as we know it is changing, mobile notifications are becoming a primary interface that drive the actions we take on our devices.

The central part of the mobile experience of the consumer is now the notification system. The Smart notifications can be sent via push notification, normal text messages, email or combinations between them.

A push notification is a message that pops up on a mobile device. App publishers can send them at any time (users don't have to be in the app or using their devices while they receive them).

Unlike pull notifications, in which the client must request information from a server, push notifications originate from a server (Fig. 1). Typically, the end user must opt-in to receive alerts; opt-in usually takes place during the install process and end users are provided with a way to manage alerts if they change their minds later on.

An important advantage of push notifications in mobile computing is that the technology doesn't require specific applications on a mobile device to be open in order for a message to be received. This allows a smartphone to receive and display social media or text message alerts even when the device's screen is locked and the social media application that is pushing the notification is closed.

Different devices and services rely on different methods to deliver push notifications. Apple developers, for example, can use the Apple Push Notification Service's Developers application programming interface (APIs) to have their apps deliver push notifications to iOS devices. Another approach is to use mobile backend as a service (mobile BaaS) cloud services to provide push notification functionality for a mobile app.

The Smart notifications are the results of a complex logical reaction in some specific situations. Smart notifications will feel more like a note from your assistant or another human being. Each mobile platform has a support for push notifications — iOS, Android, Fire OS, Windows and BlackBerry and all of them have their own services.

Notifications sent to consumers are processed on a server unit – following complex logic, and after that they are sent to mobile devices using each device's platform notification system.

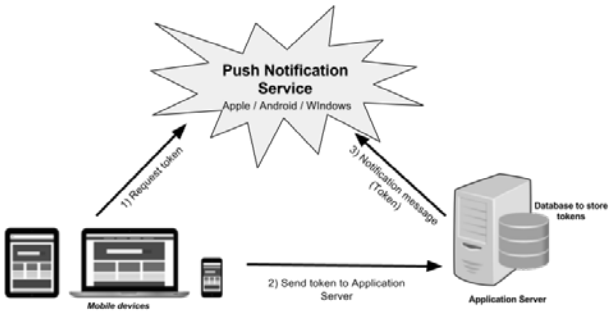

Fig. 1. Push notification flow

A little history. Apple was the first mobile platform to launch a push service - Apple Push Notification Service (APNs) – in June 2009.

In May 2010, Google released its own service, Google Cloud to Device Messaging (C2DM). Three years later, Google introduces "rich notifications" which can contain images, as well as action buttons. Action buttons let users take immediate action from a notification. For example, the user can play a song, open the app, or see more information.

The response from Apple came in September 2014, when Apple added interactive buttons which allow users to send a response right away to the app publisher. Shortly after, Apple extended push notifications to the Apple Watch.

Each mobile operation system has its own Push Notification Service (PNS). When an application is published in a store, application publisher is enabling PNS within application. When the user is opening the

application a token (a unique identifier generated by a specific code library – SDK – step 1) is registered with the OS push notification service, and is sent back to the app from the OS push notification service. At this step, the application is sending the unique token to Application server to be stored, using a set of methods to communicate through Internet – API (step 2). When the application publisher is composing a message, he defines the audience to whom the push notifications will be sent, choosing tokens (unique identifiers), which will receive the messages (step 3).

II. MOBILE APPLICATIONS

The Mobile applications have become an essential part of our life. Through mobile apps, it can be delivered rich and contextual content to any person who has installed a specific app.

Using a mobile device with a specific application, the user can receive notifications in different situations. The smart notifications are capable to start actions on user's device using user interaction or running in background.

A very good example is 'notifica.ro', a web and mobile suite applications, which is capable to initiate a normal SMS text, which will be sent from user's mobile device (using mobile contract for sending SMS) to a specific person or to a list of persons, all of this in background without human interaction, action fired by a smart notification sent to the mobile application.

According to bestappsguru.com, among the smartest notification apps for Android, in 2017, are iNoty (it brings the IOS notification window to your Android device. You can configure this application to show IOS style signal strength, battery status, WiFI, time in the notification bar), AcDisplay (one of the smartest notification app on Google Play Store. The app has the option to prevent showing full notifications which may have sensitive information such as OTP, bank transaction details, personal message on WhatsApp, etc.) or Heads-up Notifications (HUN offers a settings window, with which you can turn off notification for any app, enable or disable don't disturb feature, set notification priority, show song metadata in case the notification is about the current track you're listening to, etc).

A. Healthcare application example

More and more applications are health oriented, and they are looking for ways to build an authentic relation between patients and healthcare professionals, or interpreting the health records in real time. Mobile applications, which are recording and process data regarding some health parameters can offer and announce in real time statuses of crises and announce persons or other systems about status or locations of user.

B. Smart house monitor application example

The Mobile applications transform your smart device in a tool for monitoring proprieties - a central place for alerts but in same time a sender of data for different events to a server (a receptor) or to other person.

To develop smart notification in a mobile application, it needs a complex judgment system (logical filters), which is organized to deliver notifications in some specific situations or trigger actions for some other situations.

The smart notifications can triggered actions based on logical filters in multiple cases: data used is stored on the device (local) or using data stored remotely and accessible to the device by web services.

III. SMART NOTIFICATIONS

Smart notifications can be spited in two big categories: *pull notifications* – have origin on client's device and can trigger a request from a remote server or an action on device within application, and *push-notifications* which originate from a remote server. In both ways the notifications are results of a complex filters or actions and can trigger complex actions.

Smart push notifications can be reminders or messages appearing on the device screen and help users to keep track of several activities, places to go or events to attend.

Push notifications, having origin on a remote server, can be broadcasted to a group of users, or sent individually to a single user, to give personalized information. At a minimum way they would be helpful, personal, time-sensitive and relevant.

A. Smart timing

Notifications at the wrong time are useless.

Following our 2 mobile application examples:

- *Healthcare application* - Smart notifications with confirmation of upcoming appointments or procedures will improve back-office workflow, from health surgery points of view, and in same time will be critical for a crises situation from patient point of view.
- *Smart house monitor application* - When there is an intrusion in your house, or a motion is detected – the user can be announced about situation and in same time a notification can be sent to a security agency.

The specific time when you need to know what is happening, everything announced in real time can be considered smart timing notifications.

B. Smart location

Why to announce somebody who is not in the right place? It is useless.

Following our 2 mobile application we can consider very useful information:

- *Healthcare application* - When patient left the house in case of a disability a notification is sent to a

central server to announce and react

- *Smart house monitor application* - Why to start automatically the light in garden, or start heating in a smart house, if there are no persons (mobile devices are acting as a person presence sensor) in a specific area?

C. Smart grouping

Why to be notified about an event more times? Can be annoyed.

- *Healthcare application* – With the medical application, the user or surgery may set up some extreme limits for the patient, in this case send announcements. The notifications will be grouped and depending on other criteria may escalade and trigger other actions.
- *Smart house monitor application* - Why to receive lots of notifications during motion detection which can take minutes? The notifications can be grouped and announced only when an action starts and ends.

These can be considered smart grouping notifications.

D. Smart reactions

Every user is unique. The app cannot be build for everyone, so compromises have to be made. Notifications that react intelligently rather than having default values can help to provide a layer of product personalization. How does the person normally react to notifications?

- *Healthcare application* - Having the medical application, the patient receives a notification of a high limit of his health parameter based on data collected. In this case the user is redirected automatically to some Step-by-step instructions with some indications about what he has to do. In the same time the central unit can be announced about it.
- *Smart house monitor application* - In case of an intrusion or malfunction a smart notification can trigger automatically a call or start record the event.

These can be considered smart reaction notifications.

E. Smart targeting

Send targeted messages to people who are best positioned to answer.

A notification is useful when it is sent to the right person, based on the right filter of data.

IV. ALGOTITHMS FOR SENDING NOTIFIOCATIFICATIONS

In our examples we consider that we have a table of users (***USERS***), a table of filters (***FILTERS***), and a table where we store device values (***DATA***).

Once the user on the device, some personal details of the user along with device specific install application information (device token, system operator information and Geo location) will be sent to the server in order to register user in the system. User has option to define filters for specific values with conditions (value is less, greater or equal) or subscribe to predefined filters. Once they define or subscribe to a filter, system will add a record in FILTERS_USERS table for each filter.

We will have a background script, which will gather values and will store them in DATA table. Another background script will check filter conditions (at time intervals stored in check-time field of FILTERS table). Once condition defined in filters is met, system will get all users that required notifications to be sent for that specific filter and will send a push notification to their devices.

Database structure looks like this:

USERS - Table that stores all user related information

- *id* = unique number assigned to current record
- *user-personal-details* = personal data of user (first name, last name)
- *device-token* = unique identifier based on mobile device and application id
- *ap-type* = ios | google | MS
- *ap-enabled* = true / false (if user has PN enabled)
- *last-location* = GEO IP (lat, long)
- *last-announced* = date of last PN sent

FILTERS - Filters to check data values against

- *id* = unique number assigned to current record
- *name* = Filter name
- *condition-type* = read value is greater, less or equal to critical-value
- *critical-value* = value of a critical data
- *last-check-date* = when did we check for critical-value last time
- *check-time* = time interval to check this filter

FILTERS_USERS - Tells which users must be alerted when filters conditions are met

- *id* = unique number assigned to current record
- *user-id* = which user to alter to provided filter
- *filter-id* = id of filter

DATA - Values gathered by background script

- *id* = unique number assigned to current record
- *value* = value read
- *value-date* = date when value was read

To achieve required functionalities we will use following procedures:

ALG1. Convenient procedure which takes as parameter a list of users and a message to send push notifications to a list of user devices:

```
1  SEND_MULTIPLE_PN( users[], MESSAGE )
2  {
3      foreach user in USERS
4      {
5          SEND_SINGLE_PN( user, MESSAGE )
6      }
7  }
```

ALG2. Procedure to send a push notification to a provided user object with specified message:

```
SEND_SINGLE_PN( user, MESSAGE )
{
    if NOT user.ap-enabled
        return

    switch( user.ap-type )
    {
        case 'ios':
            POST /push/message/simple HTTP/1.1
            Host: pn.apple.com
            Authorization: Basic
                iosQWxashZGRpbjpvcGVuIHNlc2FtZQ==
        case 'google':
            POST /push/message/simple HTTP/1.1
            Host: pn.google.com
            Authorization: Basic
                googleQWxashZGRpbjpvcGVuIHNlc2FtZQ==
        case 'MS':
            POST /push/message/simple HTTP/1.1
            Host: pn.microsoft.com
            Authorization: Basic
                msQWxashZGRpbjpvcGVuIHNlc2FtZQ==
    }
    Content-Type: application/json
    Accept: application/json
    {
        "from": "APPLICATION",
        "to": {
            "UserId": user.device-token
        },
        "text": MESSAGE
    }

    UPDATE user SET user.last-announced = NOW()
}
```

ALG3. Once a condition of a filter is met we call this procedure to extract all users that required notification for this filter and we alert them for values provided in *data_records* parameter:

```
ALERT_USERS_FOR_FILTER( filter, data_records )
{
    SELECT * FROM FILTERS_USERS WHERE FILTERS_USERS.
        filter-id = filter.id INTO USERS

    foreach data in data_records
    {
        message = "Filter " + filter.name + "
            triggered for value " + data.value + "
            at " + data.value-date + " for critical
            value " + filter.critical-value + "."

        SEND_MULTIPLE_PN( USERS, message )
    }
}
```

ALG4. Convenient function which updated last-check-date field of the filter. This way we will check if condition is met for this filter only for specific intervals of time (e.g. if user wants to be alerted only once one hour for example):

```
UPDATE_FILTER_LAST_CHECK_DATE( filter )
{
    UPDATE filter SET filter.last-check-date = NOW()
        + filter.check-time
}
```

ALG5. Given a check value, condition to be checked (less, greater, equal) and a date to check from (till present), this function will take all values from DATA table that meets filter condition:

```
CHECK_DATA_VALUES( critival_value, condition,
    from_date )
{
    SELECT * FROM DATA WHERE value-date >= from_date
        AND value condition(<,>,=) critical_value
        INTO DATA_RECORDS

    return DATA_RECORDS
}
```

ALG6. This is the main function which is called from the background script that checks filters for which check interval allows notifications and also values from DATA table meets condition from filter:

```
CHECK_FILTERS()
{
    foreach filter in FILTERS
    {
        // if it's not yet time to check the filter
        if( filter.last-check-date + filter.check-
            time > NOW()
            continue

        UPDATE_FILTER_LAST_CHECK_DATE( filter )

        // or it is time, but we don't find any data
        values matching filter condition
        data_records = CHECK_DATA_VALUES( filter.
            critical-value, filter.condition-type,
            filter.last-check-date )
        if( data_records.count() == 0 )
            continue

        // Trigger alerts for current filter for
        found records in DATA table
        ALERT_USERS_FOR_FILTER( filter, data_records
            )
    }
}
```

V. BENEFITS OF SMART NOTIFICATIONS FROM MOBILE APPLICATION

An important advantage of smart notifications in mobile application is that the technology doesn't require new or external features, which are not already on all mobile devices.

The utility and value of smart notifications consists in making the best decisions at the best time. These decisions are based on complex and casuistic filters using elements of artificial intelligence, all implemented within mobile application.

In order to be successful, a good smart notification app has to have a good push notification strategy.

According to a recent research (clevertap.com) which analyzed the users' behavior of over 100 million app launches and a few million uninstalls, it is very important for the developer to have in mind some key factors when building the app:

- *Your Audience* – Who is your target audience and what information would you like to determine about them? Utilize rich user profile analytics to view demographic information and many other key insights.
- *Content* – Keep users engaged with the app by sending new and relevant content to view, like new blog articles, video updates, and trending articles. The user will stay engaged with the app if the notifications sent are relevant and interesting.
- *Timely* – Push notifications must be sent to a user at the appropriate time. Notifications about an offer near a movie theater where tickets have been purchased should be sent at the appropriate time, for example, not after the movie has ended.
- *Real-Time Notifications* – Sending a triggered notification can help drive conversions, especially for e-commerce apps. For example, a user could add an item to their cart, but not checkout. Set up a notification to trigger 15 minutes post the cart abandonment with an offer deep-linked directly back in to the cart, and you have just created a strong conversion driver!
- *Segment Users* – Marketers should efficiently segment users and filter notifications sent to them accordingly.
- *Inbox the Notifications* – If a user misses a push notification it should still be available to him elsewhere. Set up an in-app message inbox that readers can read at their convenience when they have the time to act upon your messages.
- *Metrics Management* – Analyze the success of your mobile marketing campaigns. For e.g. if you have sent a push notification about a newly launched product, you must consider measuring the number of users who opened the app – viewed the product – read the reviews or specifications – used the promo code – added the product to cart – purchased it and which user stopped where in the product life-cycle. Such in-depth mobile analytics would fetch more insights to improve the app marketing strategy and conduct better campaigns.

Smart notifications are helpful, personal, time sensitive and relevant.

VI. ACKNOWLEDGMENT

This work has been done within the ERA-NET Cofund Smart Cities and Communities Programme for Research of the EC project SMART URBAN ISLE. Authors acknowledge the financial support of the UEFISCDI in the framework of this project.

REFERENCES

[1] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press, 2014.

[2] Tilo Westermann, *User Acceptance of Mobile Notifications* (T-Labs Series in Telecommunication Services), ISBN-13: 978-9811038501.

[3] Google Cloud Messaging.

[4] Apple Push Notification Service.