Platform for Pushing and Pulling Device Oriented Information via Cloud

Nilima Sandbhor, Snehal Shinde, Varsha Hulawale

Department of Computer Engineering, JSPM's Bhivarabai Sawant Institute of Technology and Research, Wagholi, Pune

***Abstract* -** Our project is to provide a platform to share device oriented information between devices via local private cloud. The information like missed call history, massages, files, notes, to do's etc. are shared between devices. For sharing such information we build the local private cloud. For first time we use windows OS and android as platform on which client application runs. This Application can set various kinds of triggers to push information into the cloud. To get easy access to the stored data anytime, anywhere and from multiple devices we use 'software as service' (SaaS) offer through cloud computing.

Index Terms- Android, Cloud, Device oriented information, Platform, Software as service (SaaS), Windows OS

I. INTRODUCTION

In this paper, we are building a platform to share the device oriented information using private cloud. Our purpose is to create an information-sharing platform that enables easy sharing of device-oriented information. The cloud is build using SaaS platform. Whenever required, the private cloud enables access to shared resources via a network connection. Through the private cloud, information is shared among various devices. The use of mobile devices is increasing rapidly day by day and people have started to use many devices simultaneously so sharing information between these devices has become an important aspect. For example, suppose a person at work needs some file or document immediately which is present on his laptop and incase if he forgets his laptop at home then it can be retrieved if the same data is stored in the cloud. Also user can get notified of the missed call history, low battery level of a device not in immediate reach to him.

Our project also provides some important and interesting modules to make the system better. This includes providing security by the use of SHA-1 hash algorithm, platform independence (presently supporting windows OS, android OS etc. but can be extended to other platforms.)

Our application provides following functionalities:

- Device independence
- Providing security
- Remote accessibility
- Easy Expansion
- Expanded to various OS
- Virtual ability
- Setup new system
- Easy access of data anytime and anywhere
- Can me modified depending on future trend

Service providers need not worry about the resource management since they are managed by cloud itself. So user can access data and service anytime and anywhere which enables data sharing in a convenient way. Location information is also a part of application. Users are left with a choice to share location information with other people. A registered user can login from any device with this application and get access to the information in the cloud. Recovery of data is easy in case of device failure and hence providing device independence.

Our platform provides users an easy way to connect devices and cloud services. Devices with Android and Windows OS are client side components which are connected to the cloud services using the communication interface. Different triggers and conditions can be set to notify about missed calls, message alerts (if device is not available with user) and low battery level.

The System architecture shown in fig.1 shows how the system actually works or interacts. The main modules are the Android and Windows OS client and Cloud Server. The application will be installed on both the OS devices and they can share information with each other.

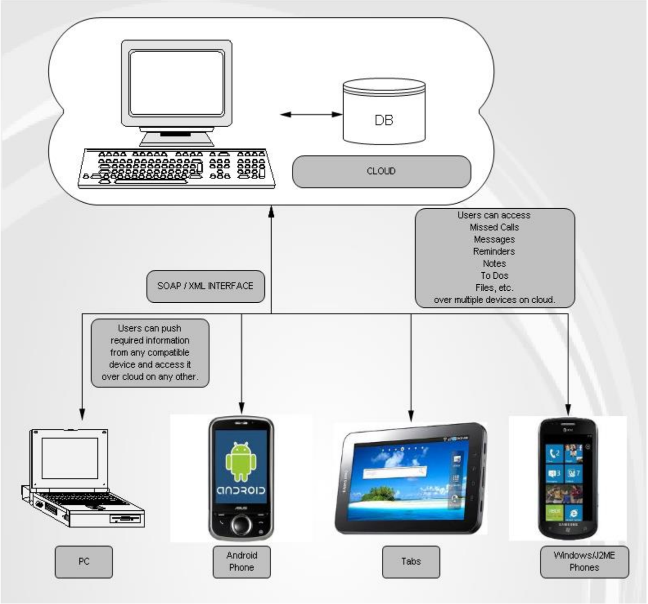


Fig 1.System Architecture

II. RELATED WORK

Some solutions of information sharing have been proposed.

a) Notitter

Notitter is a mobile bot that notifies you wherever you are. Even if you forget your phone at your home, office or car, this application will let you know different information. You can use this for emergency notification, life log, and so on. You can choose Twitter and Evernote for notifying address.

b) Context Watcher

This mobile application enables mobile phone users to easily share personal context data such as their location, heart rate, speed, or view, with their mutual consent.

It is a context-aware application which enables end-user to automatically record, store, and use context information. For example, family members can stay better informed about each other and their friends, without synchronous interaction by using automatically derived context information. In the context management framework, different entities, context providers, are exposed to and interact via the internet. Using this application a mobile phone can be used for sharing context information in everyday life. Hence, context-awareness does not force people to adapt or change significantly their daily patterns in their normal environments, but accompanies them throughout everyday life.

c) SECE

SECE is a context-aware platform that connects isolated services, making it more useful and user-personalized, composite services. SECE converges, fixed and mobile services by integrating the Internet, cellular and sensor networks. SECE takes actions automatically on behalf of the users depending on the monitored information and triggered events. SECE enables end-users to create advanced services. Although users today can use several individual Internet services, there is currently no easy way to create new services which integrate diverse information, such as location, presence, IM, SMS, calls, Facebook, Twitter, sensors and actuators.

d) Growl

Growl is local application used to share information between devices, if they are configured. When user receives shared information on device(smart phone/PC) the growl's application which is activated on that device sends a notification to Growl, which shows a pop-up to notify user that the device(smart phone/PC) receive a shared information. Depending on users preferences Growl will show the notification using one of the displays plug-in, speak to the user, email it to user, or block it from going anywhere if you have Growl set up that way. Growl notifies application information to that user's devices which are inside LAN and can access the information.

These applications are restricted to share data only on particular devices or services. Our goal is to build an application that use cloud services that avail users to share the information anytime anywhere. Cloud services provide a scalable, cost effective, and secure environment for managing your data. They also reduce expenditure for hardware and software, providing an opportunity for cost containment.

III. PROPOSED SYSTEM

A. System Outline

Our goal is to provide platform to share device-oriented information and the user is free to use any computing device they own. The following problems are to be addressed:

a) Easy adding of services

A mechanism is needed to add information sharing services along with the shared information since services in a cloud are increasing rapidly. Similarly, a convenient approach is needed to add the device oriented information. Management interface between the controller and input/output is needed to solve below problems:

- Data shared differs with each service and user
- Information may be related to data from a sensor or user context data.

b) Relation of trigger and information

Triggers and information sharing may be interrelated to each other, and so we need to collect them. For example, the platform retrieves battery level and notifies the controller.

The controller then examines the level; if it is lower than a threshold, the controller sends the level to output. Easy development of the input and triggers should be done to enable simultaneous release of services.

c) Enriching Information

We can elaborate the additional information that is strongly related to the original information (add, delete or update data). With the development of web services, there are more ways to use information. For example, location information can now be used to find addresses or landmarks nearby. We need to make it easy for users to adapt to new services.

System security depends on information sharing services that users use. Therefore, to provide security we include SHA-1 algorithm which enables storing of password in an encrypted format making it attack free from external sources. When a user wants to share information with his or her friends, they can use services with community.

B. System implementation

In this study, we implemented a platform on Android and Windows OS; we named the platform 'RIGHT ARM'. Right Arm consists of three main parts: input, output and controller. It first needs to set some parameters of the parts as shown below:

a) Sharing Information (input/output)

User is provided with the facility to choose information to push to external services. Similarly, user can choose the information to pull from the external services. Right Arm currently supports following types of information to be shared:

- Location information from GPS
- Low battery level of the device
- Missed call and unread message history

- Notes, to-do's, files

b) Controller

Setting is done by the user with respect to which information to share and the conditional triggers. There are currently three trigger settings: critical battery level, missed call and unread messages.

Right Arm works as described in the steps below:

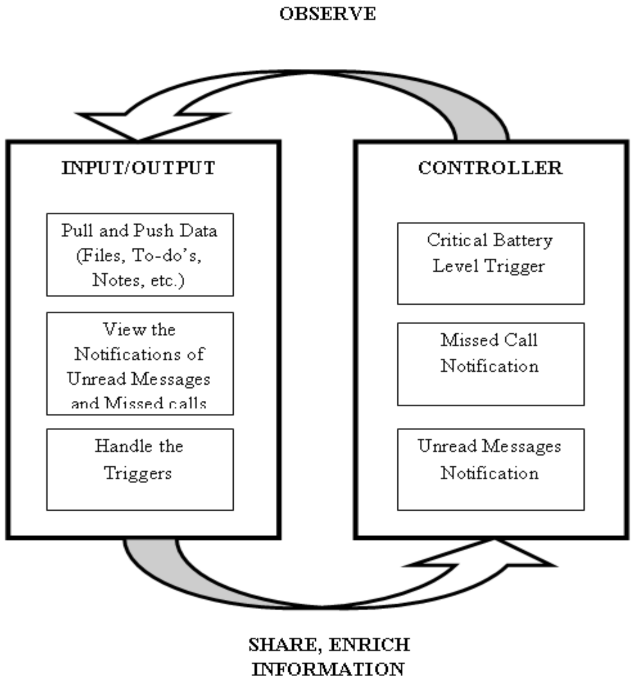

OBSERVE
INPUT/OUTPUT
Pull and Push Data (Files, To-do's, Notes, etc.)
View the Notifications of Unread Messages and Missed calls
Handle the Triggers
CONTROLLER
Critical Battery Level Trigger
Missed Call Notification
Unread Messages Notification
SHARE, ENRICH INFORMATION

Fig 2. System overview

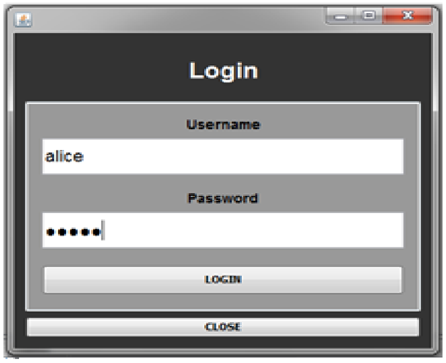
1) Settings are made as per user's preference.
2) The user starts Right Arm which works in the background.
3) The input data is constantly checked at regular intervals by Right Arm.
4) Right Arm share data with information sharing services if a trigger event is detected.

In case the user wants to check related information, Right Arm will add information by using web services.

Screen shots of Right Arm are shown in below. On the main page, the user has to login in his registered account. During registration the user enters all his details along with the user name and password. Here the password is saved in the database in the encrypted format using the SHA-1 algorithm. This will provide user complete security as the password cannot be tracked. The respective information that can be shared on Windows and Android OS client modules is as below:

a) Window Client API:

After the user has logged in the main menu is displayed showing the various types of options: missed calls, messages, files, to-do, notes etc. On clicking the respective buttons related information is shown. For example, files can be downloaded and viewed. Similarly, the missed calls and unread messages from the cell phone can be viewed on the click of missed call and messages button. Also a trigger message appears notifying about the critical battery level of the mobile phone.


Login

Username: alice

Password: •••••

Fig 3.Login Window

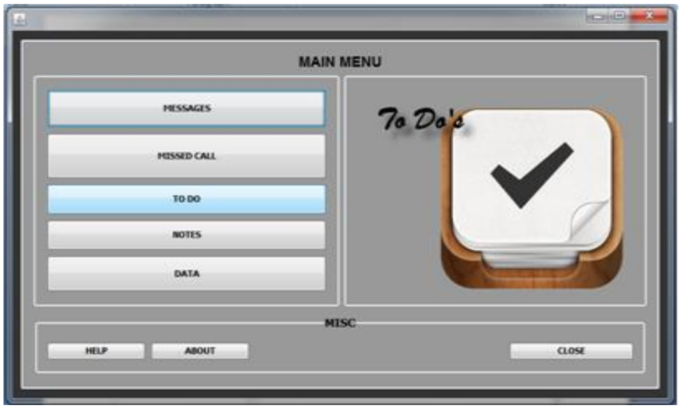
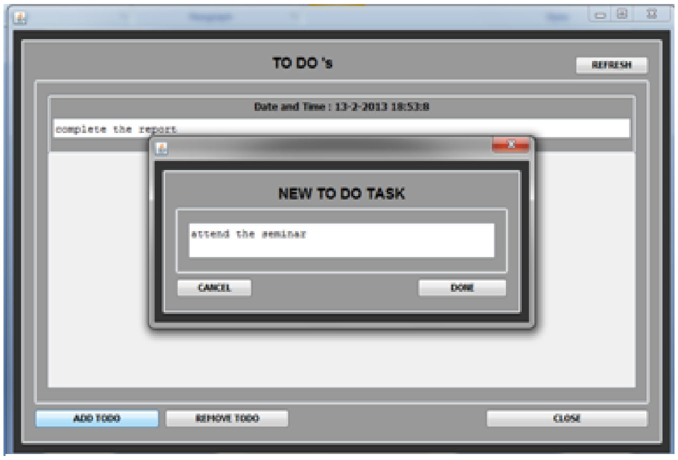


Fig 4.Main Menu Window


TO DO 's

Date and Time : 13-2-2013 18:53:8

complete the report

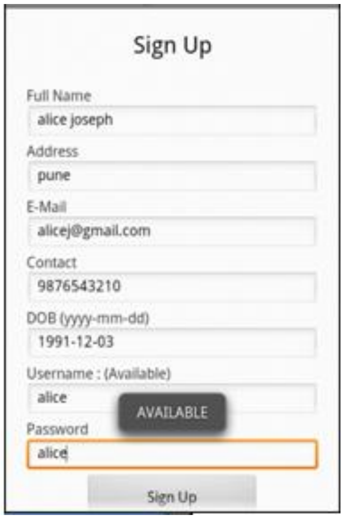
NEW TO DO TASK

attend the seminar

Fig 5.To Do Window

b) Android Client API:

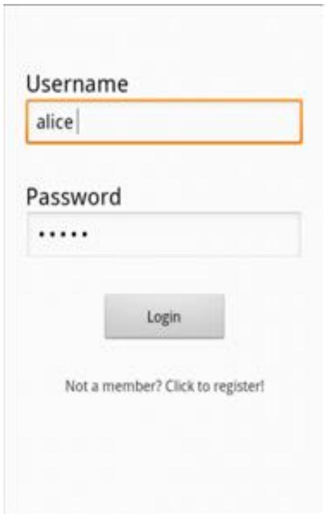
The main menu for android client is displayed with the various options: notes, to-do, files, battery level, start and stop monitor, GPS. On the click of the respective buttons the information related to it can be seen. For example, notes can be added, deleted or updated on the click of notes buttons. Similarly the start and stop monitor will notify the windows side client of the unread messages and missed calls.



Sign Up

Full Name: alice joseph
Address: pune
E-Mail: alicej@gmail.com
Contact: 9876543210
DOB (yyyy-mm-dd): 1991-12-03
Username : (Available): alice
Password: alice

Fig 6.Sign Up Window



Username: alice
Password: •••••

Fig 7.Login Window

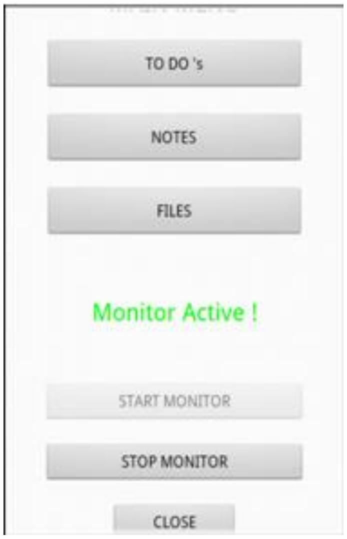
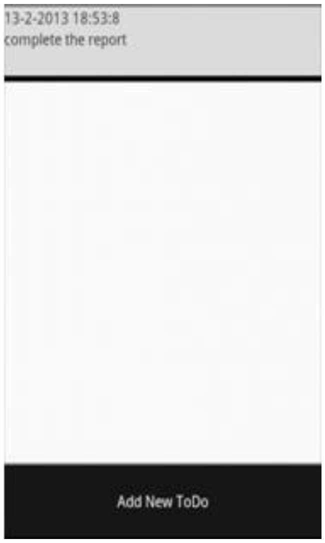


Fig 8.Main Menu Window



13-2-2013 18:53:8
complete the report

Fig 9.To Do Window

In the implemented system, user can choose the setting for information sharing, setting of cloud services, and activation of the system. The setting of each item is a layered structure. For example, the input data and trigger level of the battery is set when the battery is chosen. Corresponding APIs are used for settings of cloud services.

We present an example scenario of using Right Arm. The user first sets the settings of Right Arm on the smart phone.

Here, the setting is done by choosing the 'start monitor' option. In case the mobile is unavailable with the user the unattended call and messages is pushed into the cloud. The windows side client will pull this information. This missed call history shows number of missed calls, the person who had called, their numbers, date and time, so that the user can make a contact to that person with available device. Similar is the case with messages the entire message can be read, the senders name, time and date is also viewed.

While working with the laptop the user is not paying attention to the phone. In this situation, if the phone battery goes down to the critical level the user will be notified. This notification is given in the form of trigger message popping out of the windows API. This helps the user charge the phone's battery.

IV. CONCLUSION

Thus, Platform for pulling and pushing device oriented information via cloud is an application which aims at accessing data from cloud via registered users instantly. It let the user automatically share device-oriented information so that users can observe the same information on different devices. By using cloud services, users can now access information anytime and anywhere regardless of the situation of information source device.

V. REFERENCES

[1] Platform for Pushing the Device-Oriented Information into a Cloud Kenji Morita, Department of Advanced Information Technology, Graduate School of Information Science and Electrical Engineering Kyushu University 744 Motooka Nishi-ku, Fukuoka 819-0395, Japan.

[2] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D.Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia, Above the clouds: A Berkeley view of cloud computing, Technical Report UCB/EECS-2009-28, EECS Department, U.C. Berkeley,2009.

[3] http://growl.info/

[4] Johan Koolwaaij, Anthony Tarlano, Marko Luther, Petteri Nurmi, Bernd Mrohs, Agathe Battestini, Raju Vaidya, Context Watcher - Sharing Context Information in Everyday Life, IASTED International Conference on Web Technologies, Applications, and Services, Calgary, Canada, pp.12–21, 2006

AUTHORS

First Author – Nilima Sandbhor, Bachelor of Computer Engineering (Pursuing), JSPM's Bhivarabai Sawant Institute of Technology and Research, Wagholi, Pune, nilimasandbhor@gmail.com

Second Author – Snehal Shinde, Bachelor of Computer Engineering (Pursuing), JSPM's Bhivarabai Sawant Institute of Technology and Research, Wagholi, Pune, s.snehal121@gmail.com

Third Author – Varsha Hulawale, Bachelor of Computer Engineering (Pursuing), JSPM's Bhivarabai Sawant Institute of Technology and Research, Wagholi, Pune, hvarsha156@gmail.com

Correspondence Author – Prof. Mrs. Vrunda Bhusari, vrundabhusari82@gmail.com, +91 9970069364