LOGIN INTO MY MOBILE

Mrs. P.S. Gaikwad[1], Ms. Shital Mandlik[2], Ms. Arti Patil[3], Ms. Snehal Thakur[4], Ms. Snehal Zurunge[5]
[1]Asst. Prof., [2,3,4,5]Student, at AISSMS'S IOIT Pune - 01

Abstract – Remote control systems are a very useful element to control and monitor devices quickly and easily. This paper proposes a new architecture for remote control of Android mobile devices, analyzing the different alternatives and seeking the optimal solution in each case. Our paper mainly aims to realize a system that remotely controls information inside the terminals (Mobile) in this case. The system is a software that will track our mobile event during offline i.e when the mobile is not with us and notify us all the event through Web access. We can simply log on to our web service and track all the event happening on the mobile on our desktop in the event we are not carrying the device with us.

Keywords: SNS (Social Network Service), Augmented Reality, GPS (Global Positioning System) ,Android, Remote Control, VNC, Java, Mo- bile devices, Security solutions for mobile devices, Remote visualization.

I. INTRODUCTION

The growing popularity and spread of smartphones has changed the design of computer systems as they were known in recent years. Technological developments have enabled the creation of mobile devices with technical features previously only conceived in PC architectures or similar devices. With this evolution comes the need to integrate these devices with others so they can take actions and monitor interaction on mobile devices. Government and public institutions have provided a variety of information by development of applications using Smartphone's. Also, corporations have used Smartphone's in terms of information processing such as work capability and customer relations. Furthermore, Smartphone's have influenced change of individual life through services such as Information Search, SNS (Social Network Service), Augmented Reality, GPS (Global Positioning System). Even with technology development, mobile terminals could be lost or stolen for several reasons. As expensive terminals with diverse functions are increased, terminal loss leads to loss of personal information stored on the terminals as well as economical loss. All the information in the terminal is vital and need access to it in emergency. The proposed system can take control of the mobile device remotely and can track the entire event occurring during the non-availability period.

This system is a part of the main system that can be divided greatly into four parts: Website/Desktop application offer part to play a role about the interface enabling the user to control the terminal remotely, Authorization part to confirm whether the user joins its own service, Access management part to transmit the access code to terminal for invoking the remote control program and setting the access, and Service running part to receive the service completion code from the terminal and transmit the remote control service code and setting value for remote control the terminal. Through the application the user can view the call logs and can know who has called him. Also the user can receive SMS and can also send remote SMS.

In addition to enable the execution of tasks remotely, it is intended that the architecture allows performing software management tasks on the device and forensics tasks in order to analyze the current state of the system and analyze the traces of previous executions to analyze the state of the device. To perform communication between mobile devices and control equipment wired connections (using USB communication) and wireless connections are utilized. Communication standards will be analyzed with the aim of establishing a stable, optimal and safe communication. An important consideration in designing the architecture of remote control of devices is the security within the system. Personal information of the user will be ex- changed by the device and operations on the mobile device will be carried out remotely, and it therefore must be ensured that no external element is able to both access the data exchanged, and to take control of external access.

II. SYSTEM MODEL

System Architecture

The architecture proposed in this paper consists of remote control architecture of mobile devices on the Android platform based on a client / server model oriented to services. The server layer is performing the services of mobile device management and accepts the connection from different clients. The client layer, available from a remote device, performs the interaction between the control equipment and the monitored device. As can be observed in above Figure, the architecture offers several types of connection to different clients in order to allow the remote control to all the users. Below, the features that will implement the architecture will be listed and will be determined the chosen solution will be determined.

File system management service

A common need in working with mobile devices is the exchange of files between both systems. The user could need a file generated by the mobile to process or use it in another system. Using this service it is possible to do it without any action from the device. Furthermore, the client could inject files into the device, for example, to upgrade the firmware, make some data available to users, etc. The server will offer the information of the remote file system and will allow requesting operations of getting, putting or removing files.

Event Monitoring Module

The event monitoring is a background process that keeps tracks of the event happening on the mobile device. It monitors for any incoming calls, incoming sms, any updates in contacts or any call being dialed or received etc.

III. PREVIOUS WORK

1)Design of Remote Control System for Data Protection and Backup in Mobile Devices Inwhee Joe, Yoonsang Lee Division of Computer Science and Engineering Hanyang University

Government and public institutions have provided a variety of information by development of applications using Smartphone's. Also, corporations have used Smartphone's in terms of information processing such as work capability and customer relations. Furthermore, Smartphone's have influenced change of individual life through services such as Information Search, SNS (Social Network Service), Augmented Reality, GPS (Global Positioning System).

Even with technology development, mobile terminals could be lost or stolen for several reasons. As expensive terminals with diverse functions are increased, terminal loss leads to loss of personal information stored on the terminals as well as economical loss.

Therefore, terminal users have used a lock function just in case of terminal loss. But, this method is cumbersome and inconvenient because password is required to enter whenever they access the terminals. If they forget the lock setting, users can be uneasy about terminal loss. In case a mobile terminal gets lost, we should keep strangers from using it. Thus, we propose our remote control system as one solution to that.

Remote Control of Mobile Devices in Android Platform IEEE TRANSACTIONS ON MOBILE COMPUTING

Remote control systems are a very useful element to control and monitor devices quickly and easily. This paper proposes a new architecture for remote control of Android mobile devices, analyzing the different alternatives and seeking the optimal solution in each case. Although the area of remote control, in case of mobile devices, has been little explored, it may provide important advantages for testing software and hardware developments in several real devices. It can also allow an efficient management of various devices of different types, perform forensic security tasks, etc ...

The main idea behind the proposed architecture was the design of a system to be used as a platform which provides the services needed to perform remote control of mobile devices.

As a result of this research, a proof of concept was implemented. An Android application running a group of server programs on the device, connected to the network or USB interface, depending on availability. This servers can be controlled through a small client written in Java and runnable both on desktop and web systems.

Storage Tradeoffs in a Collaborative Backup Service for Mobile Devices

Ludovic Courtès Marc-Olivier Killijian David Powell LAAS-CNRS 7 avenue du Colonel Roche 31077 Toulouse cedex 4 France

Embedded computers are becoming widely available, in various portable devices such as PDAs, digital cameras, music players and laptops. Most of these devices are now able to communicate using wireless network technologies such as IEEE 802.11, Bluetooth, or Zigbee. Users use such devices to capture more and more data and are becoming increasingly dependent on them. Backing up the data stored on these devices is often done in an ad hoc fashion: each protocol

and/or application has its own synchronization facilities that can be used when a sister device, usually a desktop computer, is reachable. However, newly created data may be held on the mobile device for a long time before it can be copied. This may be a serious issue since the contexts in which mobile devices are used increase the risks of them being lost, stolen or broken.

Our goal is to leverage the ubiquity of communicating mobile devices to implement a collaborative backup service. In such a system, devices participating in the service would be able to use other devices' storage to back up their own data. Of course, each device would have to contribute some of its own storage resources for others to be able to benefit from the service.

Internet-based peer-to-peer systems paved the way for such services. They showed that excess resources available at the peer hosts could be leveraged to support wide-scale resource sharing. Although the amount of resources available on a mobile device is significantly smaller than that of a desktop machine, we believe that this is not a barrier to the creation of mobile peer-to-peer services. They have also shown that wide-scale services could be created without relying on any infrastructure (other than the Internet itself), in a decentralized, self-ad- ministered way. From a fault-tolerance viewpoint, peer-to-peer systems provide a high diversity of nodes with independent failure modes [13]. In a mobile context, we believe there are additional reasons to use a collaborative service. For instance, access to a cell phone communication infrastructure (GPRS, UMTS, etc.) may be costly (especially for non-productive data transmission "just" for the sake of backup) while proximity communications are not (using 802.11, Bluetooth, etc.). Similarly, short-distance communication technologies are often more efficient than long-distance ones: they offer a higher throughput and often require less energy. In some scenarios, infrastructure-based networks are simply not available but neighboring devices might be accessible using single-hop communications, or by ad hoc routing. Our target service raises a number of interesting issues, in particular relating to trust management, resource accounting and cooperation incentives. It raises novel issues due to, for instance, mostly-disconnected operation and the consequent difficulty of resorting to centralized or on-line solutions. A preliminary analysis of these issues may be found in [6,14]. In this paper, the focus is on the mechanisms employed at the storage layer of such a service. We investigate the various design options at this layer and discuss potential trade-offs.

Shared Backup & Restore

Save, recover and share personal information into closed groups of smartphones

Vittorio Ottaviani, Alessandro Lentini, Antonio Grillo, Silvia Di Cesare and Giuseppe F. Italiano

Department of Computer Science, Systems and Production University of Rome "Tor Vergata" Rome, Italy

Backup is a crucial task, since hardware faults and software or human errors can lead to the loss of important information. In addition to faults, backups are even more important for devices such as laptops and smartphones, since they are more prone to loss or to theft. Currently, smartphones are used more as handheld computers than as mobile phones, and consequently a lot of data is stored in those devices. This makes more critical the need to keep data stored on those devices safe from losses. In addition, the rapid technological evolution in mobile devices makes it more difficult to restore data saved from old devices to new ones. Thus, mobile devices pose new challenges for the backup and restore problem.

Making backups on external memory devices, such as on Secure Digital (SD) cards or on laptop disks, suffers from the same risks of failure or loss. As smartphones tend to be always connected to the Internet, it seems natural to move the information online and to provide backup and restore services based on the cloud computing paradigm, which is considered to be more reliable and less expensive by end users. This approach reduces also the risk of data loss and decouples the data from a specific device. Once information about backups moves online, it can be used in shared applications. In an enterprise scenario, for example, it can be useful for users to share business or personal data contained in their mobile's backups, such as calendar or business cards, with some selected contacts of their choice. In such scenario, it is easy to imagine a community of people willing to share some of their data within their mobile network.

A backup that allows data sharing, however, can suffer the same security and privacy issues present in social networks ; such limitations can be approached in different ways depending on the environment where the system is used. In an enterprise scenario, data sharing can be monitored by administrators which can enforce the company privacy policies. In a general purpose environment, like a mobile social network, ownership of data must be verified and sharing

must be allowed only for the data owner. Security can be ensured by deploying secure connections and data encryption.

The main goal of this work is a backup system for smartphones that allows users to share part of their personal data in the backup with a selected set of contacts. In order to be platform independent, our approach is based on a novel management of data, and hinges on a data model which abstracts from the underlying platform and focuses on the data type. The same backup and restore method can be applied both on mobile and on desktop platforms. With such system, users can manage different devices, under different operating systems, and keep data synchronized across different platforms.

In order to assess the feasibility and impact of our approach in a real scenario, we realized two prototypes of our backup and restore system for the Android and the Symbian OS, and tested them on actual mobile devices.

IV. PROPOSED METHODOLOGY

Call Logs

Android can access data through various built-in Content Providers such as contacts, browser, call logs etc. Content providers manage access to a structured set of data. One such Content Provider is android.provider.CallLog.Calls which provides access to the call logs data. Call Logs contain information about outgoing, incoming and missed calls.

Algorithm Step to Read Call Logs

1. Get the URI of the call log content provider, android, provider, CallLog, Calls is a static class that exposes a CONTENTfiURI field for the call log content provider (database).
2. Query the content provider for specific information. The android, content, Content Resolver query() method is used to query the content provider.

Querying the given URI returns a Cursor over the result set. Parameters of the query(…) method are:
uri - will be android.provider.CallLog.Calls (Where to get data from ?)
projection - will be the columns to be return. Passing null will return all columns of the content provider (What columns to return?)
sortOrder - the sorting order of the rows (How to sort the rows?.)

The following table presents the android.provider.CallLog.Calls columns constants that can be used for projections:

CACHED_NAME (String)	The cached name associated with the phone number, if it exists.
CACHED_NUMBER_LABEL (String)	The cached number label for a custom number type associated with the phone number, if it exists.
CACHED_NUMBER_TYPE (String)	The cached number type (Home, Work, etc) associated with the phone number, if it exists.
DATE (String)	The date the call occurred in milliseconds since an epoch
DURATION (String)	The duration of the call in seconds
IS_READ (String)	Whether an item is read or consumed by the user.
NEW (String)	Whether or not the call has been acknowledged
NUMBER (String)	The phone number as the user entered it.
TYPE (String)	The type of the call (incoming, outgoing or missed).

Read data through the use of Cursor

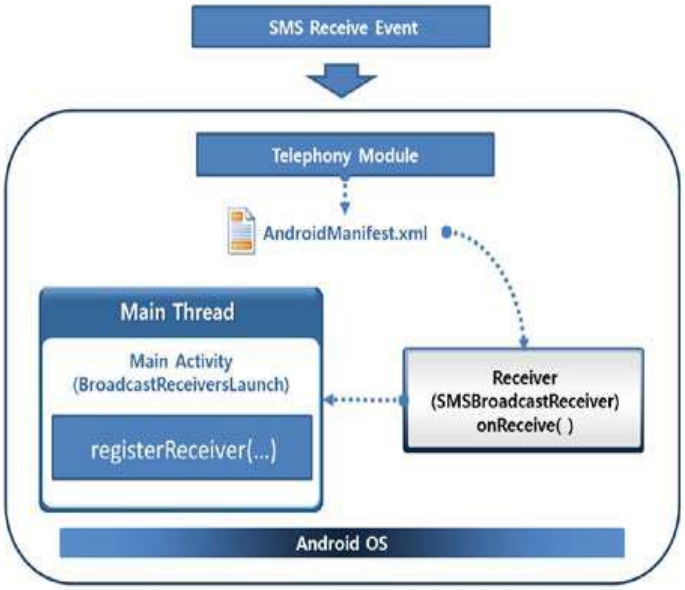
Cursor provides read access to the result set returned by the query() method. To read the data loop through the result set. Here call log information such as name of the contact, telephone number, date and time of call is accessed.

```
if (callLogCursor != null) {
        /*Looping through the results*/
        while (callLogCursor.moveToNext())
        {
                /*Contact Name*/
                String name = callLogCursor.getString(callLogCursor.getColumnIndex(CallL
og.Calls.CACHED_NAME));

                String cachedNumber = callLogCursor.getString(callLogCursor.getColumnInde
x(CallLog.Calls.CACHED_NUMBER_LABEL));

                /*Contact Number*/
                String number = callLogCursor.getString(CallLogCursor.getColumnIndex(Cal
lLog.Calls.NUMBER));

                /*Date*/
                long dateTimeMillis = callLogCursor.getLong(callLogCursor.getColumnIndex
(CallLog.Calls.DATE));
```



SMS Event

When an incoming SMS or text message arrives, a system event with the action android. provider. Telephony. SMS fi RECEIVED is broadcast. The approach, then, would be to simply create a BroadcastReceiver that catches this event and perform the tasks that follow.

Step 1: Change the AndroidManifest.xml as shown below

```
In AndroidManifest.xml, register the BroadcastReceiver and the permission to read the incoming SMS message.

<manifest ... >
    <uses-sdk ... />
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
    <application ... >
        ...
        <receiver android:name=".IncomingSmsListener">
            <intent-filter>
                <action android:name="android.provider.Telephony.SMS_RECEIVED" />
            </intent-filter>
        </receiver>
    </application>
</manifest>
```

Step 2: Create a New Class that extends BroadCastReceiver

Step 3: Overide the OnReceive Event

```
public void onReceive(Context context, Intent intent) {
    Bundle extras = intent.getExtras();
    Object[] pdus = (Object[]) extras.get("pdus");
    TelephonyManager tm = (TelephonyManager) context.getSystemService(Context.TELEPHONY_SERVICE);
    SmsMessage sms;

    for (Object pdu : pdus) {
        sms = SmsMessage.createFromPdu((byte[]) pdu);
        Log.d("Test", "originating number: " + sms.getOriginatingAddress());
        Log.d("Test", "time received: " + System.currentTimeMillis());
        Log.d("Test", "number of characters: " + sms.getMessageBody().length());
        Log.d("Test", "roaming: " + tm.isNetworkRoaming());
    }
```

File Sharing

File Sharing between Android and Server is done using FTP protocol. The FTP protocol is explained below:

The objectives of FTP are

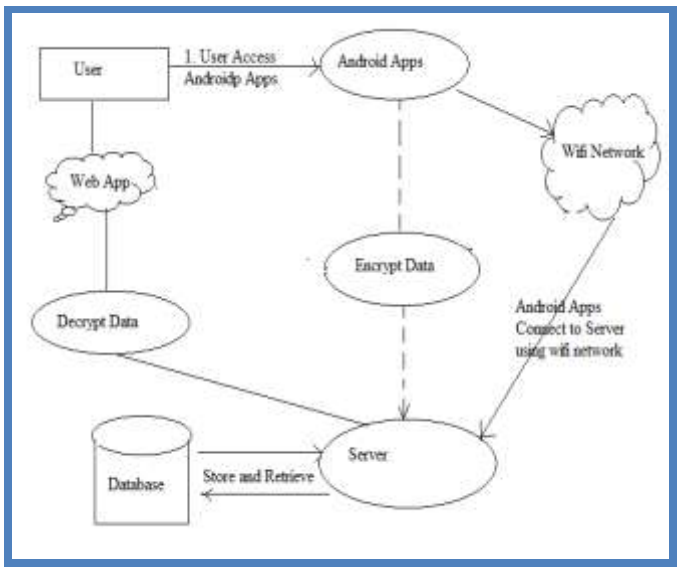
1) To promote sharing of files (computer programs and/or data),

2) To encourage indirect or implicit (via programs) use of remote computers,

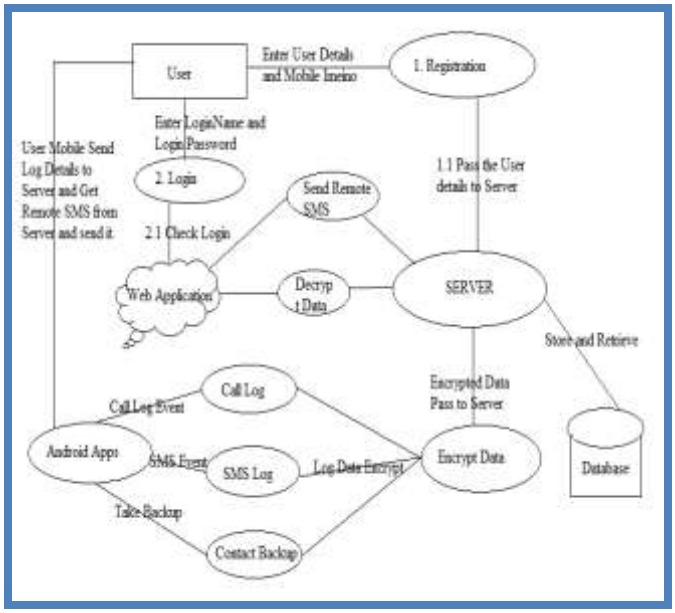
3) To shield a user from variations in file storage systems among hosts, and

4) To transfer data reliably and efficiently.

FTP, though usable directly by a user at a terminal, is designed mainly for use by programs.



ZERO LEVEL DFD



FIRST LEVEL DFD

V. CONCLUSION

Architecture has been introduced to perform the remote control of Android devices. In particular, different alternatives have been analyzed to perform the most relevant aspects, determining their strengths and weakness. As a result of the research process, the main features have been identified, and the architecture should offer the most optimal procedures to carry out the exchange of data.

To perform the connection two methods are available, USB interface or Socket networking. The connection via Networking will take advantage of the USB features as the mobile can be track remotely from a far away distance. Lastly, the system must implement a security system based in some security mechanism like encryption, password, certificates, etc. in order to ensure the data exchange and to avoid intrusions. Thus, in future works, implementation of the proposed architecture will be continued and will be made the research to integrate the challenges defined. The multi-touch events and the sensor data injection represent an important part of the remote control and should be included.

VI. FUTURE SCOPES

This system retrives sms events,call logs,messages etc.In future we can extend this proposed system to make call or video call remotely from our pc.

REFERENCES

[1] Inwhee Joe, Yoonsang Lee Division of Computer Science and Engineering Hanyang University, "Design of Remote Control System for Data Protection and Backup in Mobile Devices" , year- 2011.

[2] Vittorio Ottaviani, Alessandro Lentini, Antonio Grillo, Silvia Di Cesare and Giuseppe F. Italiano, "Shared Backup & Restore Save, recover and share personal information into closed groups of smartphones", year-2012.

[3] Angel Gonzalez Villan, Student Member, IEEE and Josep Jorba Esteve, Member, IEEE, "Remote Control of Mobile Devices in Android Platform", year-2013.

[4] David Rasch and Randal Burns Department of Computer Science Johns Hopkins University, "In-Place Rsync:File Synchronization for Mobileand Wireless Devices", {rasch,randal}@cs.jhu.edu, year-2004.