Platform for Pushing and Pulling device Oriented Information via Cloud

Hozefa Bata
Department of Information Technology, M.H.Saboo Siddik College of Engineering, Mumbai, India

Hussain Champeli
Department of Information Technology Vishwatmak Om Gurudev College of Engineering, Shahpur, India

Farhan Pasha
Department of Information Technology, M.H. Saboo Siddik College of Engineering, Mumbai,India

ABSTRACT

Mobile devices are spreading at a remarkable rate, and more people are starting to use several devices simultaneously as their own. For example, a user may carry around many devices; in this case, sharing information between devices becomes important. Our Objective is to build 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. Through the private cloud, information such as Call history, Call Logs & GPS location is shared among various devices. Sharing such information between devices would be of great help to users and at the same time cloud computing is becoming popular. The cloud is build using SaaS platform. Whenever required, the private cloud enables access to shared resources via a network connection.

Keywords

Android, Cloud, Device oriented information, Windows OS, Platform, Software as service (SaaS).

1. INTRODUCTION

Mobile devices are spreading at a remarkable rate, and more people are starting to use several devices simultaneously as their own. For example, a user may carry around three devices: a cell phone, Android tablet, and laptop computer. In this case, sharing information between devices becomes important. Information such as mail or the schedule is related to the user and not to devices. Sharing such information between devices would be of great help to users. At the same time, cloud computing is also becoming popular. Through the development of cloud computing, service providers no longer need to worry about resource management[1]. Resources are managed by cloud providers, and service providers can use resources depending on their demands. In addition, users can access data and services anytime and anywhere. This lets users share data more easily than before. Users can access the same data in the same way from any device. However, some information such as missed calls and sensor information, referred to as device-oriented information, is still isolated in a particular device. However, since such information is related to a user, it should be shared between the user's devices. When such sharing is realized, a user can be notified of a phone call to his or her home telephone or an SMS to his or her cell phone from any device the user has immediate contact with, and communication becomes smoother. Sharing sensor information between devices is also helpful. For example, PCs are usually not equipped with GPS sensors, but mobile phones usually are. If the location information of the mobile phone's GPS sensor can be used at the user's PC, this would help in the use of location-oriented services. We assumed that we can solve this situation by sharing this information through some cloud services. 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 in case 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.

2. PROPOSED SYSTEM

2.1 Literature Survey

2.1.1 Android

Google usually refers to the Android OS as a software stack. Each layer of the stack groups together several programs that support specific operating system functions. These layers are illustrated in Figure 2.1. The base of the stack is the kernel. Google used the Linux version 2.6 OS to build Android's kernel, which includes Android's memory management programs, security settings, power management software and several hardware drivers. The next level of software includes Android's libraries. Libraries are a set of instructions that tell the device how to handle different kinds of data. Android runtime layer includes a set of core Java libraries -- Android application programmers build their apps using the Java programming language. It also includes the Dalvik Virtual Machine. The next layer is the application framework. This includes the programs that manage the phone's basic functions like resource allocation, telephone applications, switching between processes or programs and keeping track of the phone's physical location. Application developers have full access to Android's application framework[2].

Application Framework is used to write applications for Android. Unlike other embedded mobile environments, Android applications are all equal, for instance, applications which come with the phone are no different than those that any developer writes. The framework is supported by numerous open source libraries such as openssl, sqlite and libc. It is also supported by the Android core libraries. From the point of security, the framework is based on UNIX file system permissions that assure applications have only those abilities that mobile phone owner gave them at install time.

Dalvik virtual machine is extremely low-memory based virtual machine, which was designed especially for Android to run on embedded systems and work well in low power situations. It is also tuned to the CPU attributes. The Dalvik VM creates a special file format (. DEX) that is created through build time post processing. Conversion between Java classes and. DEX format is done by included “dx” tool.

Integrated browser, Web Kit is chosen as an open source web browser. Google added a two pass layout and frame flattening. Two pass layout loads a page without waiting for blocking elements, such as external CSS or external JavaScript and after a while renders again with all resources downloaded to the device. Frame flattening converts founded frames into single one and loads into the browser. These features increase speed and usability browsing the internet via mobile phone.

2.2 Cloud computing

Cloud computing, or the cloud, is a colloquial expression used to describe a variety of different types of computing concepts that involve a large number of computers connected through a real-time communication network such as the Internet. Cloud computing is a term without a commonly accepted unequivocal scientific or technical definition. In science, cloud computing is a synonym for distributed computing over a network and means the ability to run a program on many connected computers at the same time.

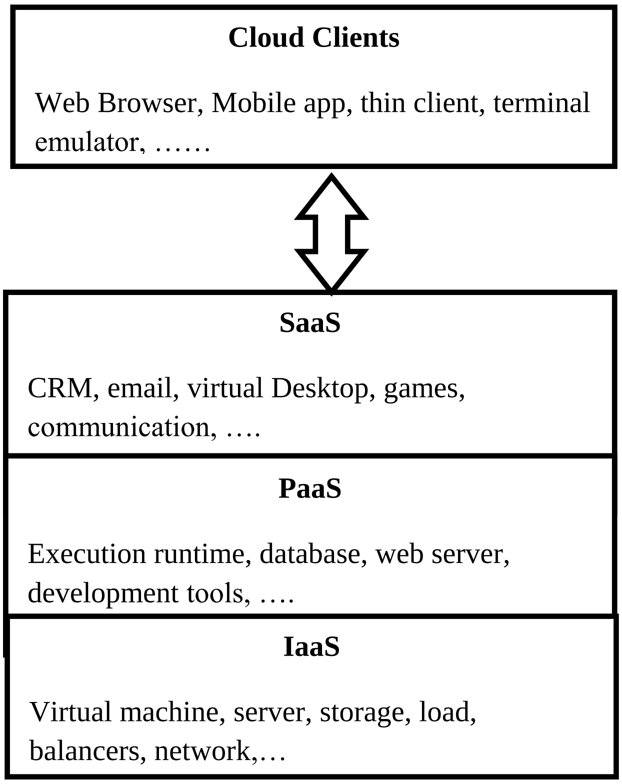
Cloud Clients
Web Browser, Mobile app, thin client, terminal emulator, ……
SaaS
CRM, email, virtual Desktop, games, communication, ….
PaaS
Execution runtime, database, web server, development tools, ….
IaaS
Virtual machine, server, storage, load, balancers, network,…

Figure 2.1 Service Models

2.3 Characteristics of Cloud computing

Cloud computing exhibits the following key characteristics:
Agility improves with users' ability to re-provision technological infrastructure resources.

Application Programming Interface (API) accessibility to software that enables machines to interact with cloud software in the same way the user interface facilitates interaction between humans and computers. Cloud computing systems typically use REST-based APIs[3].

Cost is claimed to be reduced and in a public cloud delivery model capital expenditure is converted to operational expenditure. This is purported to lower barriers to entry, as infrastructure is typically provided by a third-party and does not need to be purchased for one-time or infrequent intensive computing tasks. Pricing on a utility computing basis is fine-grained with usage-based options and fewer IT skills are required for implementation (in-house).

Device and location independence enable users to access systems using a web browser regardless of their location or what device they are using (e.g., PC, mobile phone). As infrastructure is off-site (typically provided by a third-party) and accessed via the Internet, users can connect from anywhere. Multi-tenancy enables sharing of resources and costs across a large pool of users thus allowing for: Centralization of infrastructure in locations with lower costs (such as real estate, electricity, etc.)

Peak-load capacity increases (users need not engineer for highest possible load-levels)

Utilization and efficiency improvements for systems that are often only 10–20% utilized.

Reliability is improved if multiple redundant sites are used, which makes well-designed cloud computing suitable for business continuity and disaster recovery.

Scalability via dynamic ("on-demand") provisioning of resources on a fine-grained, self-service basis near real-time, without users having to engineer for peak loads.

Performance is monitored and consistent and loosely coupled architectures are constructed using web services as the system interface. Security could improve due to centralization of data, increased security-focused resources, etc., but concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels. Security is often as good as or better than under traditional systems, in part because providers are able to devote resources to solving security issues that many customers cannot afford. However, the complexity of security is greatly increased when data is distributed over a wider area or greater number of devices and in multi-tenant systems that are being shared by unrelated users. In addition, user access to security audit logs may be difficult or impossible. Private cloud installations are in part motivated by users' desire to retain control over the infrastructure and avoid losing control of information security.

Maintenance of cloud computing applications is easier, because they do not need to be installed on each user's computer. They are easier to support and to improve, as the changes reach the clients instantly.

Today smart phones are not exclusive property of early adopters or IT professionals. Global Smart phone shipments grew a relatively healthy 43 per cent year-over-year to reach 600 million units in Q2 2010. According to com Score’s report, 234 million Americans subscribed to mobile phone plans I n January 2010. Of these 42.7 million owned internet accessible smart phones, this represented an 18 per cent increase over the three months ended in October.

The problem of mobile device theft has become sufficiently severe that legislators have decided to file bills discussing it. In theory, this is supposed to make it more difficult for stolen devices to be reused and thus less appealing. It is not always advisable for users to physically handle mobile devices to update or remove the data. The level of police involvement after the theft of a mobile device depends on the type of loss (burglary, theft robbery) and the amount of evidence on hand. Without forensic or physical evidence, there is very little police can do to attempt a device recovery. Many criminal reports do not have enough solvability factors to be assigned to an investigator (even with GPS tracking data, which is often insufficient for search warrants because of its inaccuracy). Instead, they are reviewed and occasionally the victim is contacted or the property entered into the stolen property database.

With the increasing rate of mobile theft and loss cases in recent time, force us to concentrate over the concept of Remote Mobile Monitoring and Data access. This model would enhance the current communication and monitoring schemes within mobile devices by-passing the data over cloud server.

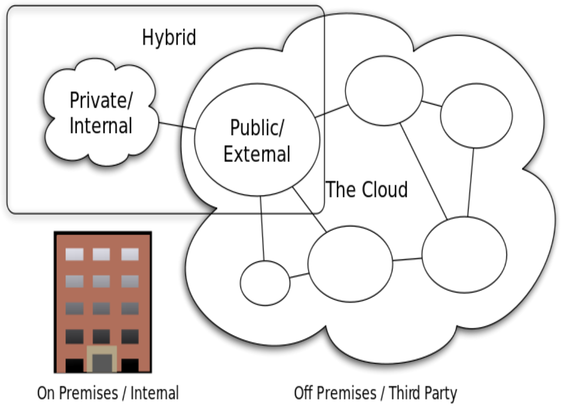

Fig 2.2 Cloud Computing Types

3. PROPOSED SYSTEM

We propose a platform to share device-oriented information between devices. Our application considers missed call history, location and battery information as typical device-oriented information. A platform and released a client application that works on Android. Our platform makes it easy to support new cloud services as output, and our Android application can set various kinds of trigger to push the information into the cloud. Our goal is to provide platform to share device-oriented information and the user is free to use any computing device they own. For that 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.

We will be developing an android based application which would be power full enough to reply for the request triggered from sever machine. This application will have functions for answering device oriented request as call logs, messages, to do, etc. it would be code as per sever request format such as once the request is being received it will first check for the type of request (e.g.: call logs, messages, battery info etc.) depending on the request type it will command device admin for response.

4. SYSTEM DESIGN

The OMG's Unified Modelling Language (UML) helps specify, visualize, and document models of software systems, including their structure and design, in a way that meets all of these requirements. Using any one of the large number of UML-based tools on the market, one can analyze future application's requirements and design a solution that meets them, representing the results using UML's twelve standard diagram types. One can model just about any type of application, running on any type and combination of hardware, operating system, programming language, and network, in UML. To model our system we have used the following diagrams:

1. Use case diagram

2. Sequence diagram

3. Component

5. REQUIERMENT ANALYSIS

5.1 Software Interfaces

The product featured, requires a database to be maintained at both the ends - client and server. Hence we will be deploying two different databases on both of them which will be in used c on demand. The client DB will be implemented using MY SQL. The UI on client will be a typical android user interface using the default options available at the time of development. UI on server will be implemented using PHP, HTML AND CSS and My SQL as a back end

5.1.1 Communications Interfaces

This product basically works on the client-server architecture. Hence a communication mechanism between server and client has to be developed. This communication will be established over https from mobile and web interface. Internal Request can be used to send/receive reports, log files and mobile details.

5.1.2 Design and Implementation Constraints

This is a product primarily being developed for users of a specific mobile platform i.e. Android. Hence users of mobile phones apart from android cannot use the features of this product. Also it may be a challenge to create sharing of data between an android device and a device running on some other platform. Hence these constraints cannot be overlooked. Fortunately, this product will be designed in a way that it will be extensible for integrating sharing feature for other platforms of mobile devices.

But taking into consideration the huge number of mobile phone manufacturers, adding this feature may take a considerable amount of time. These additions can be added keeping in view the popularity of the platform this product is being ported into.

5.1.3 Assumptions and Dependencies

While developing this product it is being assumed that the android platform version used would be v2.1 [and above]. The usage of the same on a lower version may give unexpected results. But the product will be developed to keep it as cross-platform compatible as possible

5.2 User Side Requirements

1. User should have an android mobile device running OS version 2.1 or above.
2. He should have Android Application installed within the mobile.

3. User must be able to connect device using Wi-Fi.
4. User must have internet plan enabled on his device or use Wi-Fi connection for transfer
5. Interface must be user friendly and attractive.

5.3 System Side Requirements

1. System must have Android based drivers for USB and Wi-Fi adapters.
2. The server side application must be installed on the server.
3. System must be connected to the internet.
4. Interface must be user friendly and attractive.

5.4 Hardware and Software Requirements

1) Android Based Mobile with minimum of:

a. 128KB non-volatile memory to run Mobile Information Device (MID).

b. 8KB of non-volatile memory for storage of persistent application data.

c. 2KB of volatile memory to run JVM

2) Device should have Wi-Fi connectivity.

3) 600mhz processor

4) 150 MB of RAM

5) Computer Device with available Internet.

5.5 External Interface Requirements

5.5.1 Hardware Interfaces

Typical hardware interfaces that would be required would be the standard USB data cable provided with the device. No other additional hardware interfacing would be needed to be bought by the user. Also Internet communication channel is supposed to be present with in Mobile as well as web interfaces. Communication protocol would be developer implemented and the user need not worry about communicating using this protocol. Hence this product will be extremely user-friendly.

5.5.2 Software Interfaces

The product featured, requires a database to be maintained at both the ends - client and server. Hence we will be deploying two different databases on both of them which will be in used c on demand. The client DB will be implemented using MY SQL. The UI on client will be a typical android user interface using the default options available at the time of development. UI on server will be implemented using PHP , HTML AND CSS and My SQL as a back end

5.5.3 Communications Interfaces

This product basically works on the client-server architecture. Hence a communication mechanism between server and client has to be developed. This communication will be established over https from mobile and web interface. Internal Request can be used to send/receive reports, log files and mobile details.

5.6 Security Requirements

In our product, security is of out-most importance. Hence we need to implement communication over HTTPS. For that it is required that the user should be in an environment where the HTTPS port is not blocked. In case these ports are not enabled, an http connection will have to be used. This means using http instead of https would lead to vulnerability of the user’s data to be sniffed by third-party. Hence this should be considered as a security requirement of the product.

6. EXISTING SYSTEM

There are many cloud service providers which allow a specific amount of free storage to the users. Some of such service providers are Google Drive, Apple iCloud and Dropbox. These service providers have specific applications for different platforms such as iOS and Android. Google’s Drive and Dropbox are most famous now days. Many people use this service to store the data and these applications are similar to our project. But these applications don’t allow user to backup each and every type of data user wants to back. They have some restrictions on how these applications could be used by the users. For example they don’t allow user to store the device oriented data such as call logs, contacts, Messages etc

Our Project – MyCloud will remove such restrictions and will allow users to store these types of data which are generally restricted by the cloud service providers. Using our application user can store his/her messages, contacts, call logs and even locations. iCloud provides these type of services but it only allows a backup of 5Gb maximum per user and is only available for apple device users or iOS supported devices.

Our project will try to make this service available for Android users as there are huge amount of android users now a days. Thus accomplishment of this project would result in making many people’s life easy with the help of simple cloud backup.

Some of the leading cloud services providers are:

6.1 Copy

Copy is one of the less heard-of veterans of cloud storage on Android. It's extremely simply to use, and aside from its obvious benefit of backing all your stuff up to the cloud, it has a few welcome extra features. These include Photo Copy, which lets you upload any photo you take into the cloud automatically and folder-sharing too. One of the more interesting recent features added to Copy is Chromecast support, so by simply tapping the Chromecast icon in copy, you can stream music, photos and videos from your device straight onto your TV. For most people this is now a standard feature on Android, of course, but if you don't have it yet then this is a great alternative.

6.2 Google Drive

Google Drive is just so hard to overlook because it offers 15 GB of no-questions-asked free storage and never fails. Google Drive shows a nice history of your changes too, you can allow others to view and edit your work and it offers support for many different file types. It's an obvious solution for anybody with a Google account because it's so seamless and requires very little time or management from the user.

As a side note, the Google+ social network app provides a pretty excellent cloud storage service, allowing you to store an infinite amount of pictures (at less than 2048 pixels) and videos (under 15 minutes long and no more than 1080p resolution). Not too shabby.

6.3 Dropbox

Dropbox has been a firm favourite on Android for many years now. It has a relatively small storage capacity - a paltry 2 GB to begin with - but this can be expanded up to 16 GB in various (and fairly simple) ways. Dropbox is fast, intuitive and free, with some unobtrusive automatic backup features perfect for those who want to take quick snaps and then view them on their desktop computer in a flash. It's a great overall package if you can invest a little bit of time expanding the initial storage cap.

6.4 Box

Box is available free in the Google Play Store and is very easy to use, offering 10 GB of free space with a 250 MB upload limit. Beyond this point you are asked to pay 10 USD per year for 25 GB of cloud storage.

Box is perhaps the most basic application of our list because it has no super special features: Box is there simply to upload, download and share files, although it is possible to edit and comment on files in the cloud. Box also has a widget that notifies you of any changes to shared documents[4].

6.5 SugarSync

SugarSync offers 5 GB of free data as a 90 day trial, and after that the cheapest package is 9.99 USD per month for 100 GB. But it is really simple and fast! This cloud application differs from others because it offers more payment methods, more storage space in the cloud, and it allows you to invite friends to use your files for example, and is also connected to social networks. Personally, this is the most intuitive online storage app I've ever tested, especially for inexperienced cloud service users. Instead of a series of folders, we are shown a clear menu with quick options such as Photos, Shared Documents, Private Documents, etc. The app's simplicity and clarity certainly helps to better locate each file.

6.6 MediaFire

With this tool, you can really say that you are using cloud storage[5]. MediaFire serves it up on a platter, because this application offers up to 50 GB of free available space, which is perfect for saving or sharing music or videos. But all that glitters is not gold: there is obviously critical points too. First of all, you only get 12 GB to start and have to earn the rest (or you can pay $2.50 US/month for 100 GB) and although its interface is clean the menu is a bit confusing. It has all the basic functions to download images, download and share archives etc., including those off your camera, which saves a lot of time. But another disadvantage is that the size limit for each file is 200 MB, at least in the free version. Paying more of course will earn you access to more privileges[6].

Mega is an application that provides 50 GB of storage to users upon arrival, free of charge. Everything you upload is encrypted and the encryption keys stay with the user, so there's no need for privacy concerns here. You can manage your Mega account, download files and even synchronize your camera directly to your account to automatically obtain a copy of the images and videos when you connect[7].

7. CONCLUSION

Android operating system is still in its infancy despite of some tremendous progress. This System mainly implements the cloud computing with the easiness of mobile applications to make the communication among the devices easy and more efficient. Such applications are useful not only for high end users but also for common man to make sharing information easy.

The proposed system focuses on some important and fruitful technologies which may assist human race in easier interaction with the digital world. This system has a sample scope in academic institutions where presentations are a mode of sharing knowledge. It incorporates easy access to information which is to be projected on the screen by eliminating use of stationary devices. Furthermore, the future work could draw in research on similar operations using Java-powered devices.

8. REFERENCES

[1] Platform for Pushing the Device-Oriented Information into a Cloud Kenji Morita.

[2] Department of Advanced Information Technology, Graduate School of Information Science and Electrical Engineering Kyushu University 744 Motooka Nishi-ku, Fukuoka 819-0395, Japan.

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

[4] http://growl.info/

[5] B. Kantarci, H. T. Mouftah, Sensing Services in Cloud-Centric Internet of Things: A Survey, taxonomy and challenges, in: Proceedings of IEEE ICC, pp. 1865–1870, 2015

[6] What is a cloud platform? www.adrianotto.com/2011/02/cloud-platform/. Accessed on August 13, 2016.

[7] Johan Koolwaaij, Anthony Tarlano, Marko Luther, Petteri Nurmi, Bernd Mrohs, AgatheBattestini, 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