SuusMDM: Information Resource-Manager for Mobile Devices

Roberto Leal Jacinto - N° 56883
roberto.jacinto@ist.utl.pt

Universidade Técnica de Lisboa
Instituto Superior Técnico
Department of Computer Science And Engineering

Abstract. The massification of mobile devices make them important work tools in today business world. As such, they must be managed like any other computer infrastructure equipment. Issues of connectivity not being permanently guaranteed or highly heterogeneous systems with no common basis, however, make this task a difficult one.
The objective of this work is the creation of a framework for mobile device management that facilitates the integration and maintenance of these devices remotely and in a centralized fashion, requiring minimal user intervention. Moreover, to demonstrate the possible impact of this new computing paradigm, a prototype was made to assign profiles consisting of several applications, to users, making it possible to share devices, keeping at the same time the user applications on them.

Keywords: Mobile Devices, Smartphones, Tablets, Asset Management, Distribution, Personal Context.

1 Introduction

The massification of mobile devices make them important work tools in today business world.[17,11,8] As such, they must be managed like any other computer infrastructure equipment. The identification of their characteristics, control the installed software and update it when necessary, are some of the management activities expected.[15,12] However, due to the nature of these devices, there are some distinct points in this management activity. Issues such as connectivity not being permanently guaranteed or highly heterogeneous systems with no common basis, require a different approach not provided by current solutions, typically targeted at personal computers.[4] The objective of this work is the creation of a framework for mobile device management that facilitates the integration and maintenance of these devices remotely and in a centralized fashion, requiring minimal user intervention.

To demonstrate the possible impact of this new computing paradigm, a prototype was made to assign profiles, consisting of several applications, to users. It is then possible, in any terminal associated with the system, to log in and applications will become automatically available to the user. When the session is terminated, those applications are removed, allowing the reuse of devices by several users.

2 Related Work

There are, today, multiple MDM - Mobile Device Management - solutions that offer inventory of both software and hardware, with more or less complex mechanisms that offer the remote deploy of applications. This solutions have, however, some drawbacks.

We can divide this solutions in two major groups: those who are offered by manufacturers and those who aren't. In reality, one can not make this division without some caution, as even those solutions not attached to some manufacture profit from their devices and use them with the mechanisms provided. The great disadvantage in a manufacture oriented solution is, for start, a logistic one. Mobile devices are heterogeneous both in hardware device and in operating system. If a solutions is build on top of a brand there are no guaranties that it supports other devices from other manufactures which may lead to significant money investments to redesign and change the mobile management solutions. As the mobile devices evolve both in characteristics and in working tasks, this is a probably scenario.

As for the solutions that don't have a direct manufacture association, they share some disadvantages with them. There are no possibility for remotely register a new mobile device in a management system from the device itself for example. Also, they are all centered in device/owner view. This means that a device is viewed as an extent of a user in a *"one device one use"* sort of way.

We have reviewed the major manufactured oriented solutions - e.g: BlackBerry® Enterprise Server, Apple® Enterprise Features - and the independent ones - e.g: Open Mobile Alliance Device Management. We have even gone further and analyzed some other solutions that are not exactly MDM systems but share a common permitting to manage the software on a mobile device, without the need to specifically registering the device in the system - the Android® Market Webstore is a great example of how a mobile can be managed, to some extend, by a system without the need of a complex registering making the user the focus point and the terminal just a work tool.

3 Architecture

The SuusMDM system is capable of an inventory capability of both software and hardware of mobile devices, users and application profiles. Profiles are associated to one or more users, in a many-to-many relationship. After register the terminal in the system, and log in it's possible to deploy all the applications present in the chosen profile. Those applications and their associated data will be erased from the terminal once the user logs off.

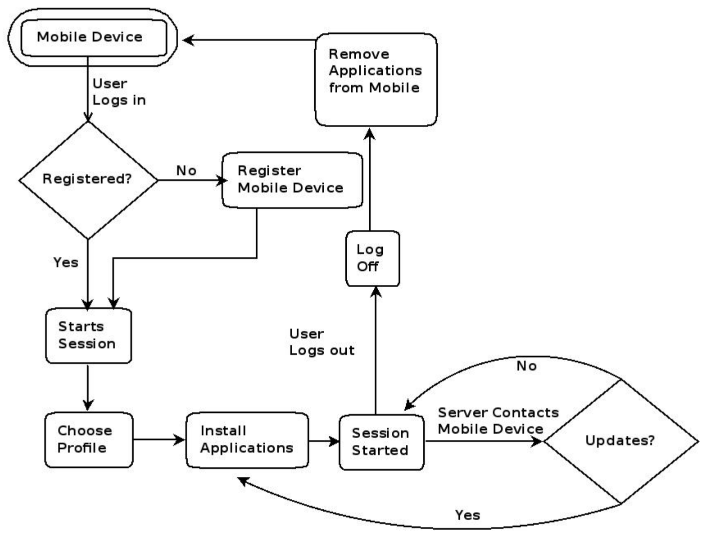


Fig. 1. SuusMDM Life Cycle

In figure 1 we can graphically see the typically life cycle of the SuusMDM system. From there we don't expect a great deal of processing demand. Also, being directed to a corporate environment, there are numerous concerns about information control. As such we have chosen, for the management system, a centralized architecture with a single entity with powers to create/remove/update information.

The small processing requirements mean that a distributed architecture is not essential for the management system (server node of SuusMDM). That's not the same as saying that there is no need for a large storage space or that it doesn't need to increase as the number of terminals also increase. Thus, a centralized architecture with a load balance that can connect several information replicas is desirable.[2,3,14] Moreover, there will be several mobile terminals to be managed. These can only access the information - e.g.: applications - having no ability to directly change the system. This architecture fits a single-writer multiple-readers model. Note that there might be competition problems if a reader were to access information that is being changed an that instant. However, being the management system - single writer - the one who contacts the terminals, sending them information, such scenario can not happen.

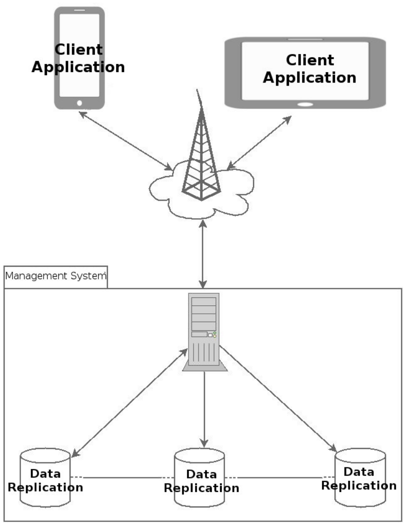


Fig. 2. SuusMDM Basic Architecture

Figure 2 shows how a number of mobile devices always access the same node in the management system, being the process made in centralized fashion and the information accessed in a replicated manner when needed. Today's technologies, such as HSPA+ / HSUPA, also make it manageable to execute a system like SuusMDM even when the device is far away from the management system.[9,16]

Also on figure 2 we can see the systems architecture from a technical execution point of view. As refereed, we have a *single-write multiple-readers* where the mobile devices managed - the readers - contact the the management system, the only one with permissions to modify and deploy information - single writer. In the mobile devices runs an application that manages all the business logic from the device point of view. On the management system side, we have a server responsible for the distribution of information to the mobile devices, as well as manage the inventory of hardware, software, profiles and users.

One of the greatest challenges about a solution as the one presented, is how to communicate directly with the mobile device, without overuse important resources, such as processor, memory, battery and network traffic. Also, there is also some challenges in identifying a single terminal, making all the communication to it secure. For the first, we used new models to aggregate the *push* technologies with a small foot-print.[7,10,13] We also search new, more light weighted, cryptographic algorithms such as the elliptic curve cryptography that provides the same level of security as today systems, but with a much minor resource hungry algorithm.[5,6] As for the second challenge, mobile device identification, we figured that there was really not a single *tag* we could use. We took a less technologies approach and used a more human communication one were random information is queried in a random, as need, basis. The verification of this information is compared with the one given in the register process which is associated to a user. That way, we can always pin-point who did what.

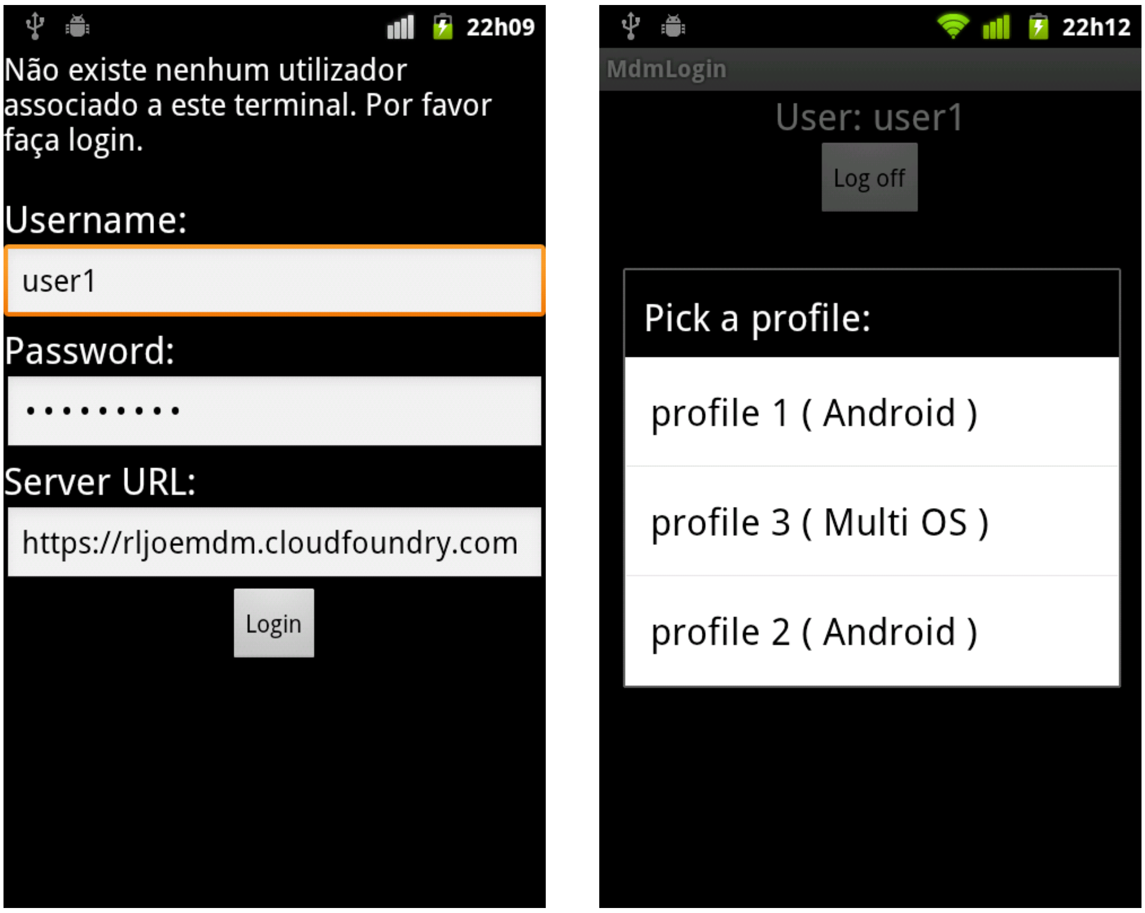
Another challenge faced during the SuusMDM development is the case were the mobile device is noncommunicable. Maybe it lost network coverage, maybe it's out of battery. Even so, wen he gets back he should be presented with all the new changes. Another scenario is that we may loose communication ability just after receive the information that there are new updates. As we discovered, the information about connection failures e much more predictable from the mobile device point of view. There for, after being informed, it's him who should chose the better time to make those update. In SuusMDM, we designed it so that a small sized information is sent to the device in order to make him aware of pending updates. After that point, it's the mobile device responsibility to get that information from the remote server. The information was also designed in away that minimizes what the device needs to know in order to fetch it. Even tho, security issues were kept in mind. We also made it in a way that the updates might be done in numerous steps, not in a direct and interrupt sequence, without loosing context. The terminal might want to fetch smaller sized information first if a disconnect is imminent, leaving the largest one to after a successful future reconnect.

4 Implementation

A prototype of SuusMDM was developed, targeting Android mobile devices. The prototype is divided in two components: a client and a server.

4.1 Client

The client component of the SuusMDM system is an application targeted to Android mobile devices and is the only component that a mobile device needs in order to use the presented solution.


Não existe nenhum utilizador associado a este terminal. Por favor faça login.

Username: user1

Password: •••••••••

Server URL: https://rljoemdm.cloudfoundry.com

User: user1

Pick a profile:

profile 1 (Android)

profile 3 (Multi OS)

profile 2 (Android)

Fig. 3. Main screen (left) and profile choosing after log in (right).

Function wise, it enables a user to log in with is credentials, chose a associated profile (as shown in figure 3), and deploy the applications of the profile in the terminal. Wile logged in, every time an update occurs in the management system of the SuusMDM (server side), if the active profile is the one modified, changes are propagated directly to the device, without the user intervention. Also, when the user logs off, all the applications installed via SuusMDM are removed. This way we can effectively share a mobile device by many users.

In order to make the updates work in an almost real-time fashion, saving network communication as well as other mobile resources such as processor, memory and battery, a push methodology was used. In it, the server connects to the mobile device every time, and only when, it has some new information to transmit. This way, there is no need for resource expensive pull requests from the mobile device every other time, even when there is nothing new to share. One major problem with push connections, especially in mobile devices, is the way we maintain them. Typically, there is a *keep-alive* model connection that is persistent. This approach has the obvious disadvantage that not only consumes the mobile device resources in a much faster way, but also that it is not appropriated for scenarios where the connection can be canceled because of network flaws. We chose to implement a typically server scenario where the device itself has an open door, much like a web server, and parses the information when someone sends it. The use of new methodologies such as the Java NIO have been used to minimize the resources used. The results were very much satisfactory and can be seen in the Evaluation section of this paper.

Moreover, this prototype successfully implemented the automatic registering of the device by itself. This means, if the device is not registered in the SuusMDM system, there is no need to manually register it. All the human intervention required is that the user is added to the system. After that, every user can register one or more mobile devices in the system.

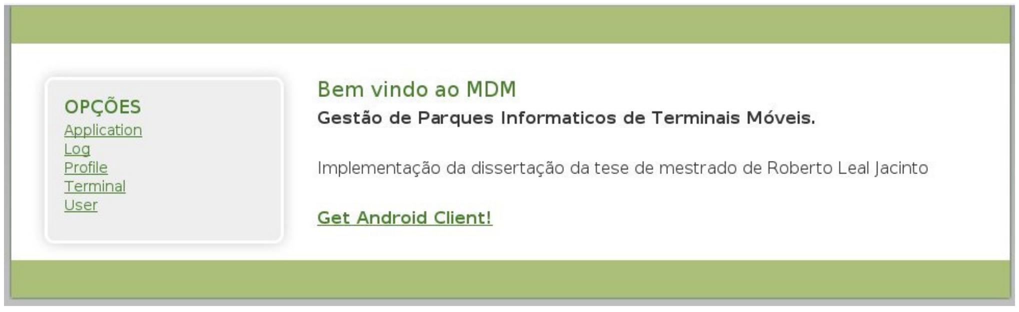
This prototype, as far as the mobile device characteristics has only two requisites:

– **Android API 8 or up -** Due to some Android specific call-backs, only Android devices with the Froyo or newer system can use our prototype. At the momment of writing, this account for 96.1% of all Android based mobile devices.[1]

– **Android root device -** In order to make silent install of applications in Android, we must have root access in the system. This can be accomplished in three ways: our application is digitally signed by Google, the application is pre-installed on the main system or the device allows for root access. Although the last solution is not officially supported, it is by far the simplest and the only that does not have monetary costs associated. As such, it is the one we followed, making it a requisite to use our prototype.

4.2 Server

The server side component of SuusMDM is a web application that can be deployed in any Java Web Server. It was made in Grails, a web framework for the Groovy language. This framework follows de "coding by convention" paradigm which made de development of the prototype faster.


OPÇÕES
Application
Log
Profile
Terminal
User

Bem vindo ao MDM
Gestão de Parques Informaticos de Terminais Móveis.

Implementação da dissertação da tese de mestrado de Roberto Leal Jacinto

Get Android Client!

Fig. 4. Main screen of the SuusMDM server component

In figure 4 we can see the main screen of SuusMDM server side component. In the left we can see the options that permit to manage terminals, applications, users and profiles. Those can be created, modified

[1] http://developer.android.com/about/dashboards/index.html

and deleted with the actions propagated directly to the mobile devices.

For the prototype, we also used a memory-based database. As a prototype, there were no real use for a full blown database management system even thou one can be utilised by our prototype with little to no effort as Grails create a transparent layer for domain usage in their architecture.

5 Evaluation

The SuusMDM solution was evaluated both qualitative and quantitative. For the first, we conducted user test and made a small inquire about the perception of the users about the solution and the prototype presented. The results shown that our solution presents a new paradigm of how we can use mobile devices such as smartphones as working tools, in a way that changes the actual "bring your own device" culture.[1]

The users that were tested had no problem using the system, and we managed to test it in some users devices that had root permissions. Granted those users were the most tech-savvy, some even working for IT companies in the mobile device world.

The quantitative tests had a more technical meaning. We analyzed the typical metrics of a client/server system as the network usage. The processor, memory usage and, in the mobile devices, the battery consumption were also analyzed with very satisfying results.

We have concluded that the SuusMDM solution is very performant in mobile devices and does not compromise the server side component. The feedback we get from users who testes the system showed us that mobile devices are really getting in the fabric of every day life, being used as working tools in almost every realm of today's world. However, people have never contacted with a solution like ours, meaning there is still a large gap for innovation.

6 Conclusion

Mobile devices have changed the way we keep connected, sharing information. Nowadays they are viewed as essential work tools making their management an important issue.[12]

Common tasks of this management activity include, but are not limited to, inventory of the devices and their software and the deploy and actualization of applications remotely. This actions are accomplished using solutions commonly described as Mobile Device Management Solutions (MDM).

In this article we presented our solutions, SuusMDM, that is able to make the most common tasks of inventory in both software and hardware assets, also providing the capability of installing software present in profiles associated to users, in a non intrusive way. Furthermore, the possibility of remotely register a device in the MDM system without human intervention as been researched.

A prototype, oriented to Android devices, has been implemented in both the client and server side, providing the features discussed in the article. The client is an application ready to install in all Android devices and the server side of the solution may be deployed in any Java supporting web server.

The prototype SuusMDM was evaluated both in a qualitative point of view, were users were invited to test the system and answer some questions, and in a quantitative point of view, were the common metric of a client-server solutions were measured and analyzed as some specific mobile oriented metrics such as battery life. The results show that our solution is not only well adapted to a mobile environment, giving great performance both in the mobile device and in the server system, but also provides a new way at looking to mobile devices, changing the strong connection device/owner, to a more "everyone's work tool". This approach was well received by the testing users and brings a new perspective to the use of mobile devices by enterprises, for example, where the actual *Bring Your Own Device* politic is disseminated.

6.1 Future Work

Here we present modifications and improvements that may be done in the future:

– **Automatically profile chooser based on ambiance metrics.** As it is, SuusMDM deploys a profile based on the log in action by the user. If he as another profile with other applications that it needs to use because he has changed context, the user needs to manually log off and log in,manually choosing a new profile. It would be interesting that the mobile device itself could realize that the user context has changed and adapt to it by auto choosing a more adequate profile.

– **Heterogeneous support of mobile devices** With the on groin number of smartphones and tablets in the market, the number of systems tens to climb and there is no winner in the horizon. From an enterprise point of view, the important is to have the tool for the job and the less money spent the best. As such, it is important to support a growing number of systems, the majority of them with no common development ground. The use of multi-platform technologies such as HTML5 / CSS / JavaScript may be a solution for the client side of the solution. We will always need some specific code for each platform - in order to install the applications or gather the system information - but we can reduce the platform dependent code and make it easier, faster and cheaper to support new devices.

References

1. It best practices: To support or not support consumer-owned smartphones. Technical report, Osterman Research, March 2009.
2. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, and M. Zaharia. Above the clouds: A berkeley view of cloud computing. Technical report, 2009.
3. J. Bacon and T. L. Harris. *Operating Systems: Concurrent and Distributed Software Design.* Addison-Wesley, 2003.
4. M. Chae and J. Kim. What's so different about the mobile internet? *Commun. ACM*, 46:240–247, December 2003.
5. W. Chou. Elliptic curve cryptography and its applications to mobile devices. Technical report, University of Maryland, College Park, Department of Mathematics.
6. D. S. A. Elminaam, H. M. A. Kader, and M. M. Hadhoud. Performance evaluation of symmetric encryption algorithms on power consumption for wireless devices. In *International Journal of Computer Theory and Engineering*, volume 1, pages 1793 – 8201 vol.1 N°4, oct. 2009.
7. M. Hauswirth and M. Jazayeri. A component and communication model for push systems. In *Proceedings of the 7th European software engineering conference held jointly with the 7th ACM SIGSOFT international symposium on Foundations of software engineering*, ESEC/FSE-7, pages 20–38, London, UK, 1999. Springer-Verlag.
8. K. Huberty, M. Lipacis, and A. e. a. Holt. Tablet demand and disruption, mobile users come of age. Morgan stanley blue papper, Morgan Stanley, June 2010.
9. K. Johansson, J. Bergman, D. Gerstenberger, M. Blomgren, and A. Wallen. Multi-carrier hspa evolution. In *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, pages 1 –5, april 2009.
10. J. Martin-Flatin. Push vs. pull in web-based network management. In *Integrated Network Management, 1999. Distributed Management for the Networked Millennium. Proceedings of the Sixth IFIP/IEEE International Symposium on*, pages 3 –18, 1999.
11. M. Meeker, S. Devitt, and L. Wu. Internet trends. Technical report, Morgan Stanley, June 2010.
12. I. Png, B. Tan, and K.-L. Wee. Dimensions of national culture and corporate adoption of it infrastructure. *Engineering Management, IEEE Transactions on*, 48(1):36 –45, feb 2001.
13. I. Podnar, M. Hauswirth, and M. Jazayeri. Mobile push: Delivering content to mobile users. In *in Proceedings of the International Workshop on Distributed Event-Based Systems (ICDCS/DEBS'02*, pages 563–570. IEEE Computer Society, 2002.
14. Y. Saito and M. Shapiro. Optimistic replication. *ACM Comput. Surv.*, 37:42–81, 2005.
15. S. Sirkemaa. It infrastructure management and standards. In *Information Technology: Coding and Computing, 2002. Proceedings. International Conference on*, pages 201 – 206, april 2002.
16. S. Tenorio, K. Exadaktylos, B. McWilliams, and Y. Le Pezennec. Mobile broadband field network performance with hspa+;. In *Wireless Conference (EW), 2010 European*, pages 269 –273, april 2010.
17. C. Zeite, M. Visitacion, E. Daley, and K. Q. Dowling. The mobile enterprise: Defining your strategy. Technical report, Forrester, March 2005.