

Implementing Mobile Applications with the MIPAMS Content Management Platform

Xavier Maroñas, Silvia Llorente, Eva Rodríguez and Jaime Delgado

Distributed Multimedia Applications Group (DMAG), Departament d'Arquitectura de Computadors (DAC), Universitat Politècnica de Catalunya (UPC),
C/Jordi Girona, 1-3, 08034 Barcelona
{xmaronas, silviall, evar, jaime.delgado}@ac.upc.edu

Abstract. New mobile devices (pda's, tablets) permit the implementation of new business models as they are always connected and provide multimedia capabilities for capturing images, videos, music or even conversations. Together with an architecture for the secure management and distribution of multimedia content called MIPAMS, we propose a mobile business model with the implementation of a mobile application based on iOS (Apple operating system for mobile devices) for publishing added value content captured with a mobile device.

Keywords: Mobile Applications, MIPAMS, iOS, Android

1 Introduction

This paper describes how to use MIPAMS (Multimedia Information Protection And Management System) [1], a service-oriented DRM platform developed at the authors' research group, Distributed Multimedia Applications Group (DMAG) [2] to implement fully functional mobile applications for the management and distribution of multimedia content in a secure way, respecting intellectual property rights governing them.

The solution presented in this paper implements part of the ideas described in [3], where several business scenarios based on MIPAMS were presented. Specifically, we focus on the scenario designed for mobile devices, which applies to an electronic publishing scenario, but other applications could be implemented as we will describe in the future work section.

The paper is organized as follows. First, MIPAMS architecture and its modules are briefly described. Then, we show a use case where a mobile application connects to MIPAMS architecture for implementing an electronic publishing scenario for mobile devices. Afterwards, some mobile operating systems are presented, together with a description of how new applications can be developed over those platforms. Then, we describe how we can connect MIPAMS with applications developed for iOS [4]. Finally, some conclusions and future work are presented.

2 Implementing Mobile Applications for an Electronic Publishing Scenario

In [3] we presented a scenario where mobile devices and applications were especially relevant for electronic publishing. It represented an electronic publishing scenario where mobile devices were used to generate “publishable” content that could have some added value. In that paper, some other scenarios were defined, showing the different ways in which an architecture for the secure management and distribution of multimedia content, known as MIPAMS, could be used to implement those functionalities in a secure way. A brief overview of this architecture is given in subsection 2.1.

The scenario described was considering the principle that mobile devices are not the most suitable devices for capturing and distributing high quality images or videos. However, they are always available and, possibly the most important, always connected. So, the most important feature for our scenario is the “opportunity” of the multimedia content being captured; that is, it was taken at the right place at the right moment. This is particularly important for unforeseen events such as natural disasters, accidents or celebrities found in an unexpected or funny situation, as this kind of events cannot be later reproduced and official mass media are not present at the instant when the event happens. In such cases, the multimedia content could even have monetary value for the author, who may try to sell or at least ask for attribution of the images or videos taken.

Starting from that scenario, this paper details a specific use case using a mobile scenario with Apple devices integrated with the MIPAMS platform. Thus, we provide the tools for a business model to facilitate exploitation of content generated in those devices.

2.1 Use of a Back-End existing Architecture to support Electronic Publishing with Mobile Devices

This section describes MIPAMS (Multimedia Information Protection And Management System), a service-oriented content management platform, developed by the DMAG (Distributed Multimedia Applications Group) [2]. It is mainly intended for applications where management of rights over digital multimedia content is required. This architecture will act as a back-end for the mobile scenario described before.

The MIPAMS architecture is based on the flexible web services approach, as it consists of several modules and services that provide an important part of the functionality needed for governing and protecting multimedia content. The main advantage of having service-oriented DRM functionality relies on the possibility of decoupling it into different subsystems depending on the needs of the application that is going to be implemented, while being able to share the same common services between different applications with different requirements, thus reducing costs. This also permits its use in innovative scenarios very different from the ones originally intended, like Social Networks [5][6] or Electronic Health Record management and exchange [7]. Nowadays, users of both applications are really concerned about

privacy. MIPAMS and more specifically the License, Protection and Authorization services, can be the key elements to improve the protection of users' privacy in both, online social networks and personal health records systems.

Figure 1 depicts the MIPAMS architecture, for which we provide next a general overview of its components and the different services being offered.

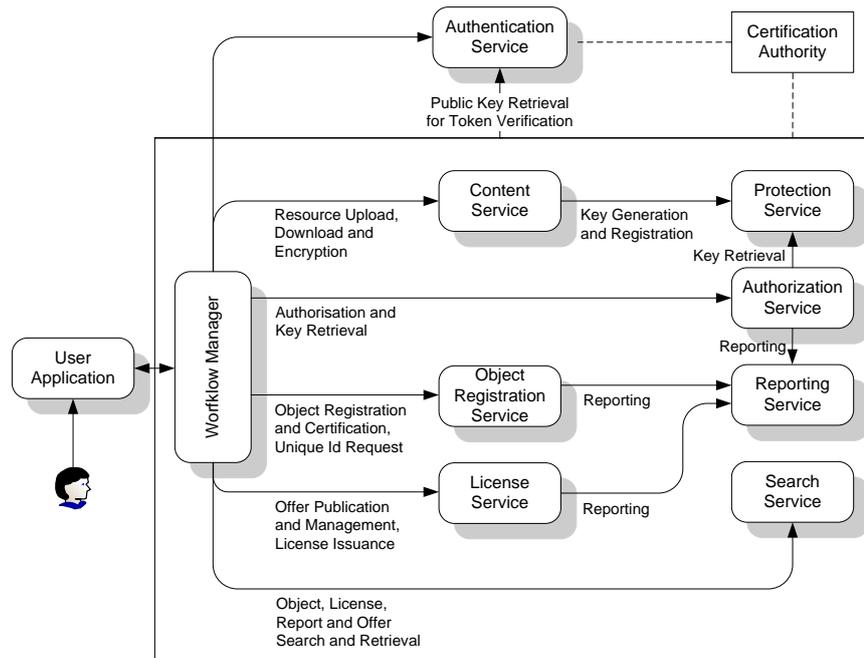


Fig. 1. MIPAMS Architecture overview.

Content Service (CS) enables applications to upload and download digital resources such as audio or video files, text documents, etc. Those resources can be optionally encrypted under request. If encryption is selected, the protection keys will be first requested to the Protection Service (PS) and then registered back into PS, once encryption is performed. Moreover, one can request unique identifiers to the Object Registration Service (ORS) in order to uniquely identify uploaded content in the system. Unique identifiers can be also obtained from external sources; it just depends on the business scenario implemented using MIPAMS.

Object Registration Service (ORS) permits the registration of digital representations (i.e. digital objects) of multimedia content (comprising content and metadata). This information is packaged using the MPEG-21 Digital Item XML-based format [8]. ORS registers and digitally signs objects so that they can be later checked for authenticity and integrity. It also provides unique identifier for uploading contents into CS, as already explained.

License Service (LS) deals with rights offers and license issuance over digital objects. Rights offers include the rights and conditions that can be acquired by users over some digital content. They are defined by contents' rights holders, which include

content creators. License issuance refers to the creation of a license according to a rights purchase or any other situation where a rights holder grants a set of rights to other user. Licenses are expressed using MPEG-21 Rights Expression Language [9].

Authorization Service (AS) checks whether a user owns any appropriate license that grants him the right to perform a requested action (e.g., play) over a digital object. The authorization is based on the license based authorization mechanism defined in [9]. This module shares license repository with LS. After positive authorization and if content is encrypted, AS requests corresponding encryption keys to PS and returns them to the requesting application. This is the only means for decrypting protected content.

Protection Service (PS), as introduced before, generates encryption keys upon request, registers encryption keys associated to uniquely identified content and provides the encryption keys for protected content to the AS.

The User Application (UA) is the player, edition tool, browser or any other means managed by the user to deal with the digital content, for instance registering or accessing it. It may have an internal trusted module to locally enforce DRM features.

Workflow Manager (WM) may be an integral part of the UA or otherwise be located in the server part (e.g. web portal, brokerage service) to reduce the UA complexity. It controls access to the rest of services inside MIPAMS architecture, like license issuance, authorization, content upload, etc.

Search Service (SS) enables applications to perform accurate searches amongst metadata in the MIPAMS system. It can be used for searching content, licenses, offers or reports or a combination of them.

Reporting Service (RS) collects usage reports regarding object registration, license issuance and positive authorizations. Those reports may be used for computing statistics as well as for billing or tracking purposes. From the information stored it is possible to generate standards-based representations like MPEG-21 Event Reports [10].

Authentication Service (ATS) is needed to authenticate the identity of users. It generates SAML (Security Assertion Markup Language)-based tokens [11] that identify MIPAMS users. Any service in the MIPAMS architecture will require a token argument to be provided in order to authenticate users. Tokens are digitally signed by the ATS, so that they can be checked for authenticity and integrity by the receiving service. Moreover, the ATS deals with user registration and management.

Finally, there is a need for having a recognized Certification Authority (CA), which issues credentials for the different Components and Actors in the system, as X.509 [12] certificates and private keys for the different architectural components.

As far as we are concerned, there is no other modular and standards based architecture that provides all functionality offered by MIPAMS. The most similar initiatives are Chillout [13], developed by the Digital Media Project (DMP) [14], and Convergence European project [15]. There is also a new MPEG standard under development, Multimedia service platform technologies (MSPT), also known as MPEG-M. The aim of MPEG-M Part 4 [16] is to define elementary services that can be combined to provide complex services. MPEG-M follows the same principles as MIPAMS where each module could be seen as an elementary service, although MIPAMS is already implemented and in use by several research projects. To provide

a complex service, MIPAMS implements a workflow manager which performs the appropriate calls to each elementary service, that is, a MIPAMS module.

3 Use Case

The use case presented in this section defines how MIPAMS modules interact with the electronic publishing scenario described in section 2. In such cases, the multimedia content captured by the mobile device may even have economical relevance, as they could be published in online newspapers, the gossip news or even broadcasted on television. Therefore, the author may register the content for different purposes: to try to get some revenues or just for later attribution.

Figure 2 depicts MIPAMS modules involved in the electronic publishing scenario and their interaction with the Mobile User (the person using the mobile device) and the Mobile Application (a specific application for registering content). Figure 2 shows the registration of the content, image or video into the registration portal, including how the author can create some offers over the content to get some revenues. It is worth noting that one could create an offer that provides the content for free, but the authorship and some limiting conditions may still apply.

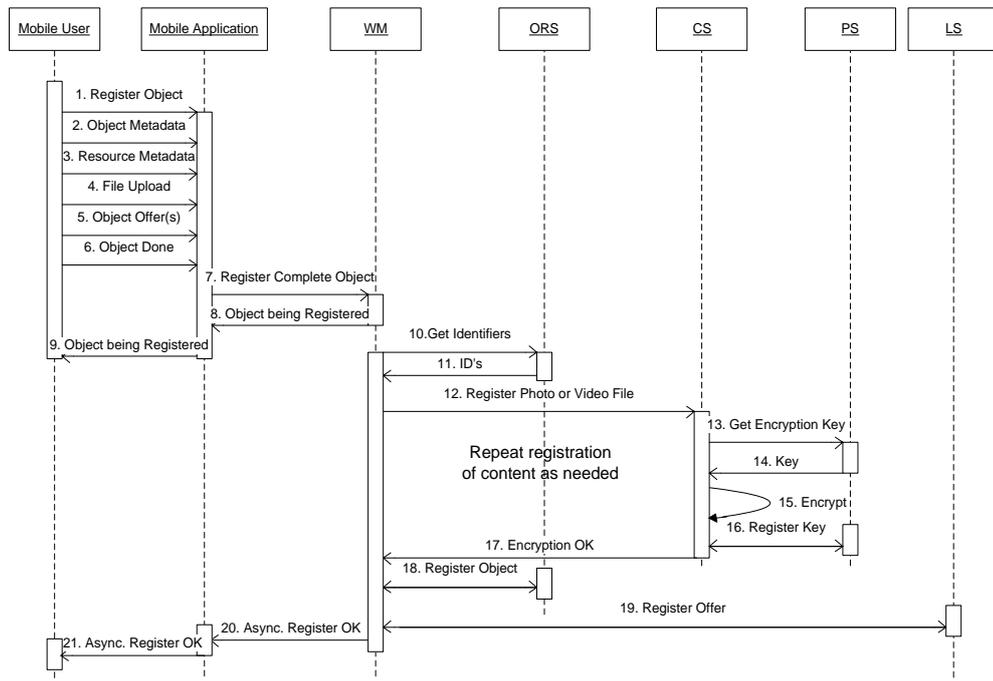


Fig. 2. Registration of image or video captured with the mobile device, including offers.

Figure 2 illustrates the content registration process, including also offer creation, to facilitate the registration in a one-step-process from a Mobile Application. The specific steps involved in this process are the following:

1. User starts registration of content locally in an application for mobile devices (Mobile Application, MA). Some information is predefined to facilitate the registration process.
2. User fills a form with all metadata associated to the complete digital object. Some fields, like author, can be automatically filled by the MA.
3. User fills several forms (one for each image or video, that is, each resource) with metadata associated to each resource.
4. User selects the file containing each resource. This file is accessible by the mobile device (inside any local storage).
5. User defines the offers applying to the object. User has to insert the different sale conditions they offer for the registered object. These conditions include what can be done with the content (play, print, etc.) together with some conditions (territory, number of times one can perform the action or payment conditions). Again, license templates or predefined forms can be used to facilitate the task.
6. User indicates that all object information has been inserted and the registration information has to be sent to the server.
7. MA sends all information to the Workflow Manager (WM) module.
8. WM sends an immediate response to the MA. Later on, if the registration is successful, the Mobile User will be informed through the MA.
9. MA shows an indication that registration is in progress to the Mobile User.
10. WM requests identifiers to the Object Registration Service (ORS).
11. ORS sends the identifiers requested to the WM, one for the object and one for each resource file (even if they have not been already uploaded).
12. WM sends resource to the Content Service (CS).
13. Since the user has requested encryption of the resource, CS asks for encryption keys to the Protection Service (PS).
14. PS returns the keys for encryption algorithm and key length specified by CS.
15. CS encrypts and stores the file with the given key.
16. CS registers the encryption key in the PS for permitting later decryption.
17. CS sends WM notification of correct content storage and encryption. Steps 10 to 17 are repeated for each resource uploaded by the user. If no resources are uploaded, these steps can be done later.
18. When all resources are properly uploaded and encrypted, WM requests ORS the registration of the complete object, which is digitally signed to guarantee digital object integrity. The format used for storing the object is the MPEG-21 Digital Item.
19. WM registers the offer provided by the Mobile User. If there is more than one offer, this step will be repeated as many times as needed.
20. WM sends asynchronous notification of object registration to the MA.
21. MA informs the user that the object has been properly registered. In any case, Mobile User always can search her registered objects to check if everything is

correct. This is especially useful when the mobile device connection is lost for some reason.

Once there are offers over a registered object, other users are able to purchase it. At that moment, a license is created based on the selected offer and the purchasing user can access the purchased content. For the moment, content download is only possible from an application installed in a laptop or PC, but with the major introduction of tablets between users, other possible applications and business scenarios could apply. Tablets are devices between 7 and 10 inches of screen that are not as powerful as PCs but they are more than a simple mobile phone. So, for tablets one could use the mobile application for registering content due to its quickness but also content download and consumption could be done combining both functionalities, the ones for mobile phones and the ones for PC.

Nevertheless, there are some other tasks that could be done from the mobile device. After registering and creating offers over some content, the author may look for her registered objects or check if any of her objects has been purchased, requesting the event reports associated to them. Figure 3 illustrates the interaction between the Mobile User and the different MIPAMS modules. In fact, there is one service dedicated to searches inside MIPAMS, which is the Search Service. This service accesses the necessary databases (Objects, Reports, etc.) to get the requested information.

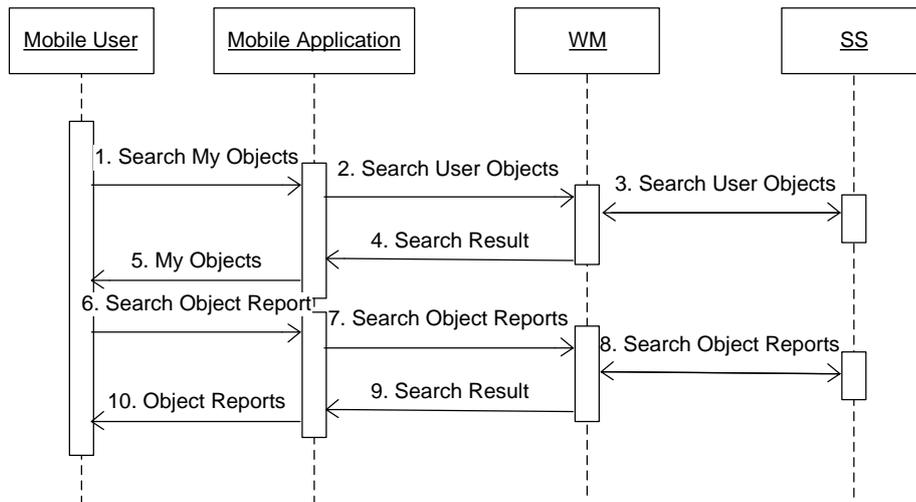


Fig. 3. Search of object reports.

Figure 3 illustrates how a user can look for her registered objects and request the activity reports (mainly purchases) over them. The following steps are involved:

1. Mobile User searches her objects through the Mobile Application.
2. MA sends the request to the WM.
3. WM contacts the SS, which searches the objects registered by the specified author. SS sends the search result to the WM.

4. WM returns results to the MA.
5. MA presents objects information to the Mobile User.
6. The Mobile User selects an object from the ones returned by the previous search in the MA and requests its reports.
7. MA sends a report request to WM indicating which user's object wants to query.
8. WM sends a report request to SS indicating which user's object wants to query. SS returns Event Reports accomplishing the criteria to the WM.
9. WM returns Event Reports to the MA.
10. MA shows results to the Mobile User.

There are other possibilities in the use cases presented. For instance, the registration process may be implemented as a simple one-step process using predefined offers, where the right and some basic conditions are already defined and the user only has to select them. Once content is registered, users may always create other offers from a web portal. For instance, a user gives play permission to a low quality version of the content and later on offers a high quality version for broadcasting at a higher price.

Another improvement for mobile devices may be the minimization of metadata associated to the different resources. This will facilitate registration as the messages sent will be smaller and thus the connection time will be less, too.

Apart from the mobile application, users will always be able to access content from a PC, and update offers and metadata as required.

Finally, a major concern for the mobile user may be the connection bandwidth and speed. So, if she does not have a quick connection, uploading a photo or a video may not be possible. In that case, she may just register basic metadata and offers and upload the digital resource later, by means of WiFi connection, providing better connection capabilities and usually supported by current mobile devices.

4 Implementing Electronic Publishing Scenario with Different Mobile Platforms

The implementation of the electronic publishing scenario presented in sections 2 and 3 could be done on different Mobile Platforms, including iOS [4], Android [17], BlackBerry OS [18] or Windows Phone [19] as most of them provide direct connectivity over HTTP/HTTPS. With this basic connection support, it is possible to connect with web services implementing the SOAP protocol [20].

For each platform, there are several possibilities for establishing the connection between the mobile device and a remote web service, as presented in the used case defined in section 3.

The first and possibly the most expensive one in terms of development time is the implementation of a dedicated SOAP client working over the HTTP/HTTPS support provided by the mobile platform.

The second one is the use of an external library that makes the implementation of a SOAP client easier, as provides the basic functionality for creating a SOAP request

message and receiving a SOAP response message. Almost every mentioned mobile operating system provides a library with this kind of protocol support.

Finally, there is the possibility of creating another Front End using REST [21] services, as there is a major support of REST services on mobile devices as they have less communication overhead than SOAP.

In the following subsections an overview of Android and iOS systems is provided before the description of a mobile application running on iOS.

4.1 Android Overview

Android [22] is an open-source software stack for mobile devices (phones and other devices like tablets) that includes an operating system, middleware and key applications. The Android Open Source Project (AOSP) [17], led by Google, is tasked with the maintenance and further development of Android. The Android SDK [23] provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language. Many device manufacturers have brought to market devices running Android, and they are readily available around the world.

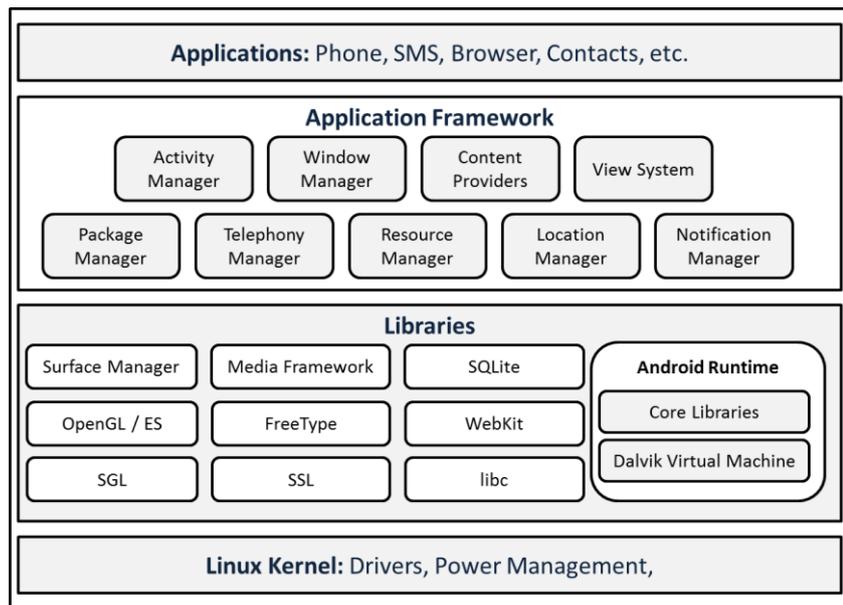


Fig. 4. Android Architecture overview.

Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack. Figure 4 shows an overview of the Android architecture, which is

organized in different layers. See [24] for a detailed description on how to implement a mobile application with Android.

Android latest version is 3.1, made public in May 2011 [25]. It introduces many changes for both, users and developers, like new user interface, new 2D and 3D graphics, etc. It is specifically optimized for devices with larger screen sizes, particularly tablets that currently are the new trend in smart always connected devices.

4.2 iOS Overview

iOS [4] is the official mobile platform for Apple [26] mobile devices, including the iPhone, iPad and iPod Touch. This platform includes both the operating system and the frameworks and libraries needed for developing iOS compatible applications. Also, the XCode developer tools package [27] provides the IDE, instruments, compilers and simulators with which build applications. All these stuff is only available inside the iOS developer program [28] in which you have to be registered to access the information, forums, tools, SDK, etc.

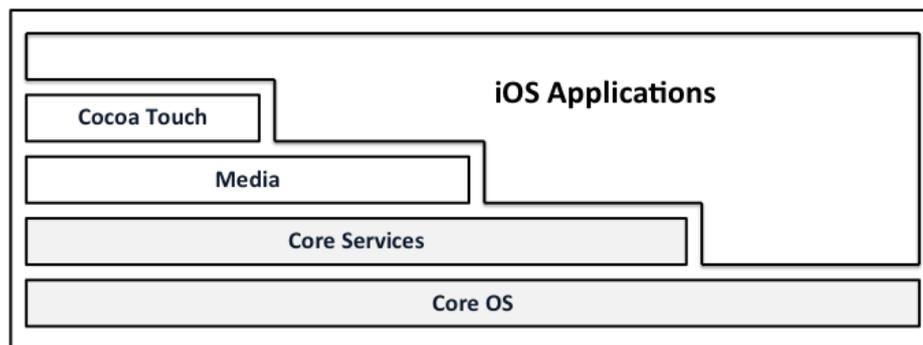


Fig. 5. iOS Architecture overview.

iOS platform is based on the Objective-C programming language [29], used by Apple both in Mac OS X and iOS development. iOS can be divided in 4 basic technology layers as described in Figure 5. Applications are built not only on top of these layers but using any of them depending on the requirements. However, most of things can be done using the “Cocoa Touch” level and other existing high-level frameworks.

iOS 4 is the latest major version, and introduces more than 100 changes from the previous one including the possibility of grouping applications using folders, change the wallpaper image, compatibility with external keyboards using Bluetooth connections, etc. But, one of the most significant improvements on the system was the introduction of the “multitasking” feature, which makes possible for the developers to keep the application alive even when users close it.

4.3 iPhone Implementation

This section describes the details of the development and the distribution of a MIPAMS client in a mobile environment. In the development subsection there are also some screenshots of a preliminary interface design of the application. In [30][31] there are described previous mobile applications implemented by the authors, that give an idea of the evolution of the mobile devices in short time.

Development. Something to be considered when implementing a mobile application are the target device capabilities. Although there are many different devices that use the same OS, all of them have some common features: the reduced size of the screen and the limited computational capacity. Even the way in which user interacts with the device is very different from a PC environment. For all these reasons the application has to be as simple as possible in terms of screen interfaces and steps. We propose a one-step content registration and the use of predefined offers to make the registration process easier and faster. We can achieve those tasks by sending just one compacted message, reducing both the time and the amount of data sent.

Although both the content metadata and license introduced by the user are restricted, the current implementation of MIPAMS includes a web portal, which can be accessed from any web browser. Using this portal, a user may change object's information afterwards. That makes sense in a scenario where the user wants to upload the minimum object information to make it accessible as quickly as possible.

Another way to reduce the time waiting for the registration response from the server is the possibility of uploading the digital object separated from the metadata. This would reduce the global upload time and also the amount of data sent which is something to be taken into account when the user's connection is slow or inexistent. In addition, the registration process could be done asynchronously, letting the user interact with the other options of the application or even close it just a moment after the registration message is sent. In this case, the user would receive a message from the server with the registration result when the process is finished.

We are developing a first version of the described mobile application using iOS technologies. The first conceptual interface screenshots are the ones shown in Figure 6. In this version, only the content registration (metadata and licenses) and content upload is available. The search of other's objects and offers and the option of searching reports will be implemented later. As it is a mobile application, it would not include all the features that the MIPAMS portal has, but there are some other operations that could be considered like the possibility of including a simple license editor in the mobile device.

All these points make sense when talking about devices like phones or PDAs, but there are some other mobile devices that have to be taken in account. Both iOS and Android have a version of the operating system designed for tablets. It is not clear if it has to be a native application for this kind of devices or it is better to use an adapted web application.

Image 1 in Figure 6 shows the different options provided to the user, content registration and search and reporting. Image 2 refers to the location of the content to be uploaded, followed by Image 3 where some basic metadata can be filled. Images 4 and 5 refer to the offers part of the application. Image 4 shows the offers menu, where

some predefined offers can be selected. Image 5 shows the information the user has to fill in case she decides to create a new offer. Finally, Image 6 shows the object registration summary, showing the basic information that will be sent to the server for registering the object, which includes object information together with offers created.

At the time of writing, we have already developed some of the steps mentioned above. We have implemented some of the Workflow Manager (WM) functions in a servlet. Doing so, we reduce to the minimum the information sent and received by the mobile device, and the time waiting for the response.

The first service we have connected is the Authentication one. As this application is supposed to work in a trusted and secure environment, the user will not be able to use it without being authenticated in the MIPAMS system.

This first service has been connected using an OpenSource library called, ASIHTTP [32]. This framework lets us establish an HTTP connection in a very simple way. The application just has to know the destination URL and the parameters to send; then the library will manage the connection, the request and the response, including the possible errors. Also, it can work both synchronous and asynchronously, returning control to the mobile application while it is sending a request and waiting for the response. The library implements some protocol methods to alert the application whether it receives the response or identifies connection errors.

After this connection, the application is ready to continue with the registration process. To do so, we maintain the connection architecture used for login. A new method has been developed in the WM servlet, responsible of the Object Creation process (both the metadata and the digital resource upload). With that method, the mobile application is able to implement a one-step registration, sending the object metadata and digital resource at the same time, using an asynchronous connection. The servlet receives the request and makes the needed steps to register the object metadata, obtaining a unique id and uploading the resource. After that, WM sends the response message to the mobile application, which shows the user the registration result.

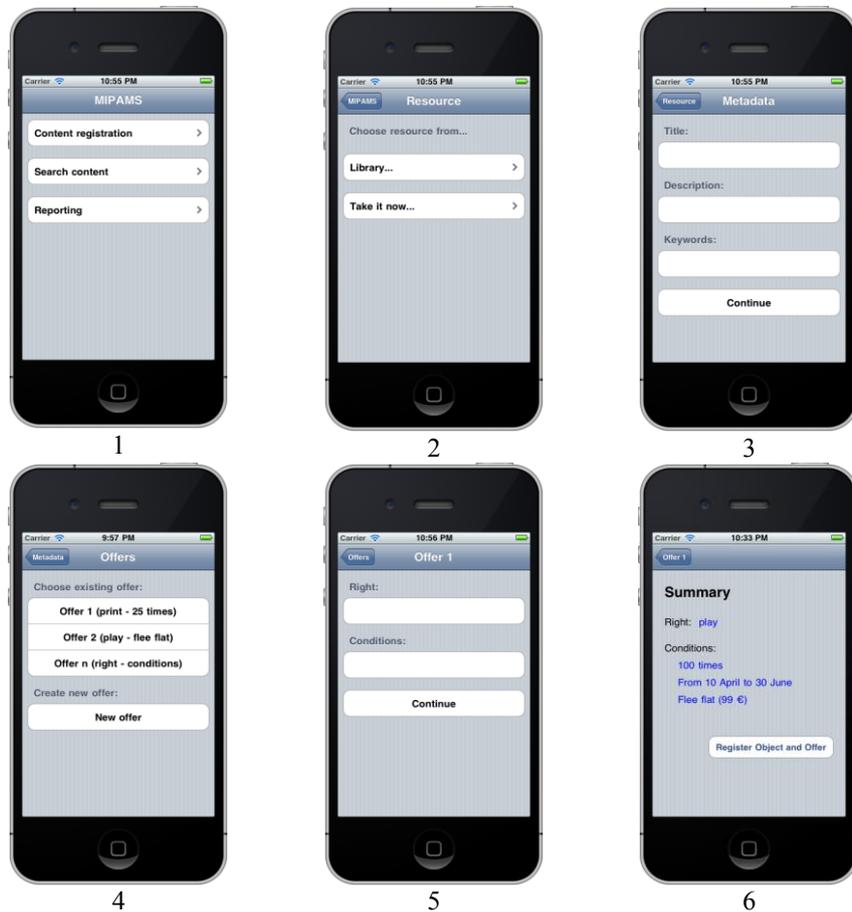


Fig. 6. MIPAMS mobile client screenshots.

Distribution. One of the most restrictive things in Apple mobile devices is the business model used on the applications distribution. When an application is uploaded to the App Store, Apple retains the 30% of the benefits, and does not let the application use any other payment system. That constraint reduces the possibilities of creating a parallel business model inside this kind of applications. But, for the specific case of MIPAMS, the client application distributed using the App Store would be free of charge, allowing users to register and upload content inside MIPAMS. It would not be possible to buy any content directly through it, avoiding Apple's distribution charges. So, for the business model presented, MIPAMS application is not breaking Apple's rules but complementing them. The user will just use the application to upload or query the content, metadata and offers, but the real transactions for acquiring the licenses will be done through the MIPAMS web, possibly using a PC or a tablet.

Furthermore, MIPAMS mobile client would not allow the user to buy or download content. The content download and use would imply a much higher complexity in the application development and the possibility that Apple refuses it. It would be interesting to have some preview of the content as the user searches through them, but this possibility will be studied for future versions of software.

It is worth noting that Android Market also retains a 30% of benefits when selling an Android App. Thus, the business model for Android would be very similar to the one for Apple. Also in this case, content will be purchased through MIPAMS portal instead of the mobile application.

5 Conclusions and Future Work

This paper has presented a business model for the development of mobile applications for the secure management and distribution of content captured with the mobile device itself. This model lies in the MIPAMS Architecture, a standards-based architecture where different modules provide the protection, rights definition and object registration functionality needed for secure content distribution. We have also presented a use case in section 3 to clarify how a mobile application connecting with MIPAMS should be implemented, describing the different steps involved in object registration, metadata introduction and how to associate offers to the content registered using the mobile platform. Section 4 describes the implementation of a mobile application to connect with MIPAMS over iOS operating system. Some screenshots and some possibilities for implementation and distribution of the mobile application have been described.

For complementing the use case a first version of the application has been presented. Nevertheless, we are thinking of other kind of mobile devices, tablets, which have a bigger screen and more rendering capabilities than smaller mobile devices, like iPod or iPhone. A more complex application may be developed and distributed for tablets.

Moreover, we are also evaluating the use of Android operating system for the secure management and distribution of multimedia content with MIPAMS. The use of web services and the lessons learnt from the iOS implementation will facilitate this task.

Acknowledgements. This work has been partially supported by the Spanish government through the project MCM-LC (TEC 2008-06692-C02-01).

References

1. Delgado, J., Torres, V., Llorente, S., Rodríguez, E.: Rights management in architectures for distributed multimedia content applications. Trustworthy Internet, Heidelberg: Springer, (May 2011)
2. Distributed Multimedia Applications Group (DMAG), <http://dmag.ac.upc.edu>

3. Delgado, J., Llorente, S., Rodríguez, E., Torres-Padrosa, V.: A Mobile Scenario for Electronic Publishing based on the MIPAMS Architecture. 15th International Conference on Electronic Publishing “Digital Publication and Mobile Technologies”, (June 2011)
4. iOS technologies, <http://developer.apple.com/technologies/ios/>
5. Delgado, J., Rodríguez, E., Llorente, S.: User’s Privacy in Applications provided through Social Networks. 2nd ACM Workshop on Social Media (WSM 2010). ISBN: 978-1-60558-933-6 (2010)
6. Llorente, S., Rodríguez, E., Delgado, J.: Secure Management of Social Networks Applications Data. Proceedings of the 8th International Workshop for Technical, Economic and Legal Aspects of Business Models for Virtual Goods (2010)
7. Rodríguez, E., Delgado, J., Alcalde, G.: Protection of patients’ privacy by means of standard technologies. Proceedings of the 9th International Workshop for Technical, Economic and Legal Aspects of Business Models for Virtual Goods (2011)
8. ISO/IEC. (2005). ISO/IEC IS 21000:2 – Part 2: Digital Item Declaration
9. ISO/IEC. (2004). ISO/IEC IS 21000:5 – Part 5: Rights Expression Language
10. ISO/IEC. (2006). ISO/IEC IS 21000:15 – Part 15: Event Reporting
11. OASIS. (2005). Security Assertion Markup Language (SAML). <http://saml.xml.org/>
12. Internet Engineering Task Force (IETF): RFC 3280 Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, April 2002.
13. Digital Media Project. Chillout. <http://chillout.dmpf.org/>
14. Digital Media Project. <http://www.dmpf.org/>
15. European FP7 Project CONVERGENCE. <http://www.ict-convergence.eu/partners/>
16. ISO/IEC. (2011). ISO/IEC DIS 23006-4 Multimedia service platform technologies – Part 4: Elementary Services
17. Android Open Source Project (AOSP), <http://source.android.com/about/philosophy.html>
18. Blackberry overview, <http://us.blackberry.com/ata glance/>
19. Windows phone, <http://www.microsoft.com/windowsphone/es-es/default.aspx>
20. Simple Object Access Protocol (SOAP), <http://www.w3.org/TR/soap/>
21. Fielding, R. T., Representational State Transfer (REST), 5th chapter of Thesis dissertation “Architectural Styles and the Design of Network-based Software Architectures”, <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
22. Android Developers, <http://developer.android.com>
23. Android SDK, <http://developer.android.com/sdk/index.html>
24. Chang, G., Tan, C., Li, G., Zhu, C.: Developing Mobile Applications on the Android Platform, WMMP 2008, LNCS 5960, pp. 264–286, Springer-Verlag Berlin Heidelberg (2010)
25. Android 3.1 Highlights, <http://developer.android.com/sdk/android-3.1-highlights.html>
26. Apple, <http://www.apple.com/>
27. XCode tools, <http://developer.apple.com/xcode/>
28. iOS developer center, <http://developer.apple.com/devcenter/ios/index.action>
29. Objective C programming language overview, <http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html>
30. Llorente, S., Delgado, J., Maroñas, X.: Implementing mobile DRM with MPEG-21 and OMA. In WOSIS 2007 proceedings. INSTICC Press (2007)
31. Llorente, S., Delgado, J., Maroñas, X., Barrio, R.: Experiencing Digital Rights Management in Mobile Environments. In AXMEDIS 2008 proceedings. IEEE Computer Society (2008)
32. ASIHTTP library, <http://allseeing-i.com/ASIHTTPRequest/>