Experiencing Digital Rights Management in Mobile Environments

Silvia Llorente, Jaime Delgado, Xavier Maroñas, Rubén Barrio
Universitat Politècnica de Catalunya, Jordi Girona, 1-3, 080034, Barcelona, Spain
{silviall, jaime.delgado, xmaronas, rubenb}@ac.upc.edu

Abstract

Content consumption in mobile devices is no longer limited to ringtones, games or images. The increasing capabilities of new mobile devices include now music and video clips. The content sent to the mobile device and even content created from devices has an increasing value. The application of security mechanisms and the introduction of Digital Rights Management (DRM) can give companies and final users a new way of interchanging content. Nevertheless, it is not easy to provide applications that work on different mobile devices. The specific features of these devices (manufacturers, operating systems, programming environments, capabilities, etc.) make difficult to implement innovative applications for them, especially those related to DRM, which usually involve several security aspects. Different approaches have been taken to overcome these problems, which are presented here.

1. Introduction

Current Digital Rights Management (DRM) systems have sometimes been criticised by consumers of digital content. The lack of interoperability between DRM systems and the increasing number of systems proposed do not cover the expectations of consumers nor companies. On the one hand, consumers complain about the fact of not being able to consume the legally purchased content on any device they own as they were able to do with traditional distribution mechanisms (for instance, CD or DVD). For this reason, users download non protected content from peer to peer networks, which can be stored and then consumed on any device supporting the corresponding format. On the other hand, distribution companies lose an increasing amount of money through illegal downloads and piracy that have not been solved by current DRM systems.

In this scenario, mobile devices are entering into the content consumption arena, as they offer to the consumers more capable devices with bigger screens and higher processing power. Moreover, new devices like digital cameras offer mobile services and capabilities in a world which tend to an always connected environment.

In this paper we present several solutions implemented for offering DRM capabilities to mobile devices based on current standards and initiatives. Due to the specific features of these kind of devices (different operating system, diverse programming environments, diverging capabilities), different solutions have to be applied to the systems implemented. The work presented here wants to give an idea of the difficulty of development and some of the solutions adopted to overcome the problems found. The work presented has been developed by the Distributed Multimedia Applications Group (DMAG) [1] in several research projects. DMAG is a leading research group in the DRM systems specification, standardisation and development area.

The paper is organised as follows. First, we briefly describe the DRM standards and initiatives that can be applied to the mobile environment. Then, we explain the different DRM solutions we have implemented for the mobile environment. After this, we show some of the problems found and approaches adopted during development of the different solutions described. Finally, we draw some conclusions and future work in the area of DRM systems applied to mobile devices.

2. DRM for Mobile Devices

In this section we describe the different standards for DRM that apply to the mobile environment. Some of them are specific for mobiles, like the one defined by the Open Mobile Alliance (OMA) [2], but others are more general for the protection and distribution of digital content on any environment, like MPEG-21 [3].

2.1. Open Mobile Alliance (OMA) DRM

This is the DRM system defined by the Open Mobile Alliance (OMA), a forum of more than 200 industry partners leading activities on the mobile communications environment. The aim of the DRM system defined by OMA is that the different features defined are implemented by mobile devices provided by the different manufacturers and has to be interoperable among them.

The scope of OMA DRM (Open Mobile Alliance DRM) [4] is to enable the controlled consumption of

978-0-7695-3406-0/08 $25.00 © 2008 IEEE
DOI 10.1109/AXMEDIS.2008.9

digital media objects by allowing content providers the ability, for example, to manage previews of DRM Content, to enable superdistribution of DRM Content and to enable transfer of content between DRM Agents. The OMA DRM specifications provide mechanisms for secure authentication of trusted DRM Agents, and for secure packaging and transfer of usage rights and DRM Content to trusted DRM Agents.

Regarding the description of rights associated to content, OMA defines a Rights Expression Language (REL) [5], based on Open Digital Rights Language (ODRL) [6]. In OMA, rights are self contained objects (Rights Objects, RO), separated from content and independent from the content type. In this way, one can distribute content in the OMA DRM Content Format (DCF) [7] separately from rights objects. The user will only be able to access the content if he owns the corresponding rights objects. This feature allows superdistribution of content, that is, one user can send protected content to another user who will be able to consume it only if he purchases the corresponding rights object.

2.2. MPEG-21

This standard defines a framework for dealing with different aspects of multimedia information management. Different parts of MPEG-21 standard normatively specify different pieces and formats needed by a complete DRM system. MPEG-21 Rights Expression Language (REL, Part 5) [8] defines a machine-readable language to declare rights and conditions for the distribution of multimedia content. MPEG-21 Intellectual Property Management and Protection Components (IPMP, Part 4) [9] deals with the standardisation of a general solution for the management and protection of Intellectual Property. Multimedia content can be protected in order to ensure that the access to it is done according to the license terms. The solution lies in the use of digital signatures and encryption techniques over the digital content, which makes it possible to deploy a business model that ensures the accomplishment of the license terms in a controlled way. MPEG-21 Event Reporting (ER, Part 15) [10] provides a standardised means for sharing information about events amongst peers and users. Such events are related to the usage and interaction with the multimedia content done by users and/or peers.

Although MPEG-21 is a general framework, MPEG-21 REL has defined several profiles to facilitate interoperability with other standards and initiatives for different environments. The most relevant for the work presented here is the first profile, so-called Mobile And optical Media (MAM) profile [11]. It addresses the needs of mobile and optical media domains. In particular, it facilitates the interoperability with OMA DRM REL v2 [5] and defines the new rights and conditions needed for the pre-recorded optical media and mobile domains.

3. Implementation of DRM solutions for Mobile Devices

In [12], we presented how to implement a DRM system for mobiles in the context of a research project. The solution proposed was based on the two most relevant initiatives in the field, MPEG-21 and Open Mobile Alliance (OMA), as we took elements from both of them: OMA License format, MPEG-21 REL authorisation model and notifications using MPEG-21 Event reporting. With them, we implemented a light DRM solution, as the mobiles capabilities in terms of processing, Java language libraries support and storage were limited. Nevertheless, we were able to test the implementation in mobile phones (Nokia 6680) and PDAs (Qtek 2020i), with slight modifications for both programming environments.

This is not the only experience we have in this field, as we have also implemented a mobile DRM solution in the AXMEDIS European project [13]. Moreover, we are implementing a DRM solution for mobiles for a Spanish company, called Futurlink [14]. These two implementations are described in more detail in the next sections.

3.1 AXMEDIS

One of the major aims of AXMEDIS European Project [13] is to create and exploit an innovative technological framework for automatic production and distribution of cross-media contents over a number of different distribution channels (e.g., networked PC, PDA, kiosk, mobile phone, i-TV, etc) with DRM (Digital Rights Management). In the next subsections we briefly describe AXMEDIS DRM architecture and the solution implemented for the mobile environment.

3.1.1. AXMEDIS DRM Architecture. Inside the AXMEDIS project, it has been designed and implemented a complete DRM architecture. The general description of the AXMEDIS DRM architecture main modules is as follows:

- *Protection Processor*: This client tool module is responsible for estimating the client tool fingerprint, enabling or disabling the tool, verifying the tool integrity and unprotecting protected multimedia objects.

- *Protection Manager Support Client (PMS Client)*: This client tool module manages and stores protection information, licenses, reports offline performed actions and other secured information in a local secure storage system. It is responsible for authorising users to perform actions over objects with respect to digital licenses during offline operation. It also delivers protection information to the protection processor or requests it to the AXCS after a positive authorisation. It acts also as the intermediary module used by Protection Processor to contact AXCS to certify and verify tools.
- *Protection Manager Support Server (PMS Server)*: The functionality provided by this module is the following: 1) creation and storage of rights expressions, 2) adaptation of rights expressions, including translation, 3) authorisation of content usage based on the licenses owned by the user and 4) requesting protection information to the AXCS if needed. MPEG-21 REL and its authorisation model is used for creation of rights expressions and authorisation of actions over content, but also adaptation of rights expressions is implemented, including translation to OMA DRM REL.
- *AXMEDIS Certifier and Supervisor (AXCS)*: AXCS is composed by several modules that provide user registration, tool certification and verification, user and tool management, generation of unique identifiers and object metadata collection. AXCS is also responsible for saving the Protection Information related to protected multimedia objects as well as the actions performed on them, the so called Action Logs. They are the particular implementation of MPEG-21 Event Reports in the AXMEDIS context.

Figure 1 shows the relationship between these modules in the AXMEDIS DRM architecture.

3.1.2. AXMEDIS Solution for Mobiles. The DRM solution for mobile devices implemented in the AXMEDIS project is mostly based on the MPEG-21 standard (license format, authorisation algorithm, event reporting) with some elements taken from OMA DRM (license format).

The licensing in this project is mostly done with MPEG-21 licenses. OMA licenses are created from the MPEG-21 ones using a translation tool, which works in both directions (from OMA to MPEG-21 and from MPEG-21 to OMA). OMA licenses are used on devices supporting natively OMA DRM. Not all MPEG-21 REL licenses can be translated to the OMA ones, as MPEG-21 considers both distribution and final user licenses whilst OMA only allows final user licenses. Moreover, several elements cannot be translated between the two languages, as they may not exist on the counterpart. This situation has been partly solved by the MPEG-21 REL MAM profile, but current implementation does not cover it completely.

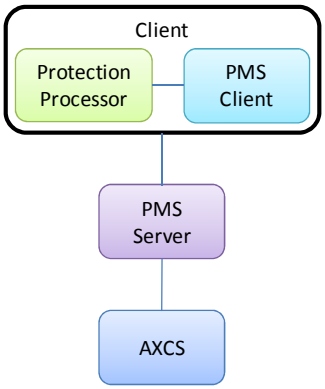

Figure 1: AXMEDIS simplified DRM architecture

So, for the implementation of the DRM solution we use distribution and final user MPEG-21 licenses, MPEG-21 authorisation model and OMA licenses (translated from the MPEG-21 ones). The event reporting mechanism implemented is an implementation of MPEG-21 ER and it is used by all DRM tools implemented in the project (not only for the mobile environment).

We have implemented two different solutions for different kind of mobile devices. The first one is a Java language midlet that implements a servlet client that sends the minimum information from the device to a servlet. The mechanism is similar to the one implemented by browsers in order to send form information to a web application. The servlet is the intermediary between the mobile device and the PMS Client, a client module implemented for PC that connects with the trusted servers through a secure channel. This solution can be used by devices supporting the Java midlet libraries needed to establish a secure http connection. In particular, we have tested it with PDAs and mobile phones. The second solution is a version for Pocket PC of the C++ library that implements the PMS Client, a client module described below. In both cases, the connection established between mobile device and the different servers is secure and each element of the chain has its corresponding X.509 certificate for authenticating it.

Figures 2 and 3 show the architecture of the DRM solution implemented for mobile devices and PDA's in the AXMEDIS project.

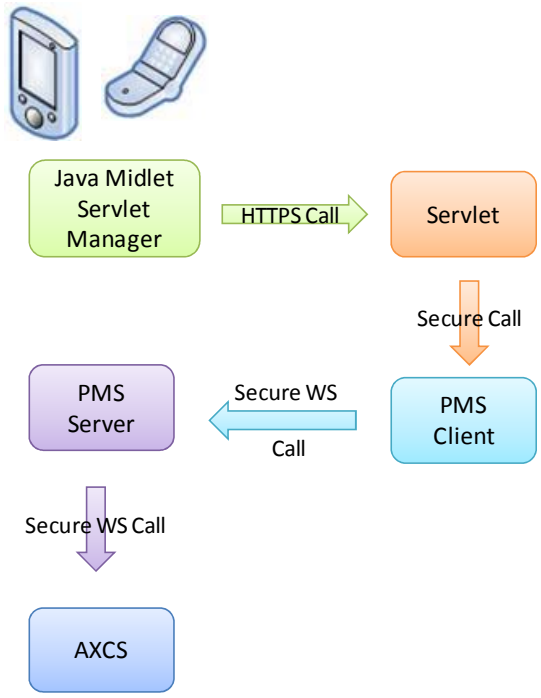

Figure 2: Mobile device implemented solution

The modules present in the different figures have the following functionality:

- *Java Midlet Servlet client:* Sends the basic information to be able to authorise the user to perform an action over content on the mobile device.
- *Servlet:* Receives the information sent from the mobile device and transforms it into a call to the PMS Client.
- *Protection Manager Support Client (PMS Client):* This client tool module manages secure connection with PMS Server. In the PC version, it is able to store information regarding licenses and content consumption to perform local authorisations when working off-line (not connected to PMS Server).
- *Protection Manager Support Server (PMS Server):* This server side module is responsible for authorising users to perform actions over objects and requesting protection information to AXCS if needed. It acts also as an intermediary module to contact AXCS from PMS Client.
- *AXMEDIS Certifier and Supervisor (AXCS):* AXCS receives the Action Logs corresponding to the actions over content done in the mobile device.

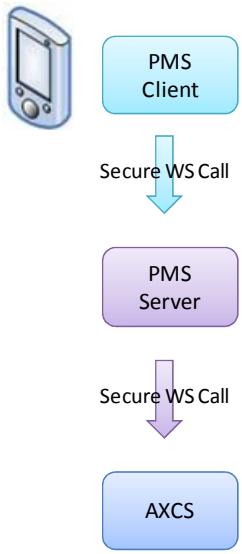

Figure 3: PDA implemented solution

For more details on AXMEDIS architecture regarding DRM modules and its functionality, see [15].

3.2 DRM System for Final Mobile Users

In this project, the tools implemented for providing DRM capabilities to mobiles are based on libraries implemented by us or the company we are working with in Java and C++ programming languages.

The tools we are implementing provide: License creation in OMA DRM REL, authorisation of user actions based on MPEG-21 REL authorisation algorithm, reporting of user actions, content packaging, protection/unprotection of content and context information storage. The so-called context information is the information needed to control that a user can perform an action over protected and governed content. This may include the number of times the user has consumed the content, first time he consumed it, licenses used to permit content consumption, etc. The content is protected and packaged together with the license that allows its consumption. When sent to the user, content protection keys are personalised in order to not permit illegal content superdistribution. In this way, a user can send content to another user's mobile device, but the receiver will not be able to render the protected and governed content received. Other adaptation operations (changes in image dimension, music codec, changes in video dimension or codec) could be done, depending on the user's device.

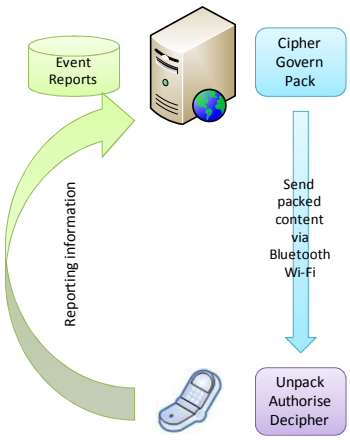

Figure 4: Overview of modules implemented

In this solution, distribution licenses are not used, as each license is directly sent to the mobile device together with the content it governs when a final user purchases it. Authorisation of actions is done locally (on the mobile device) and the user can only render content if he is authorised. Later on, when the user purchases new content, the reporting of actions performed is sent to the server for billing purposes. Figure 4 shows the main modules involved in the DRM system implemented. The server is in charge of preparing the protected and governed content in a package that will be sent to the mobile via Bluetooth or Wi-Fi, depending on the mobile capabilities. The mobile has to unpack, authorise and decipher the content locally in order to render it. When user connects for searching new content, reporting information is sent to the server.

4. Implementation Issues in the Mobile Environment

In this section we would like to describe problems, solutions and lessons learnt from the implementation of DRM systems for the mobile environment. Some of the problems presented are inherent to the development of applications for mobile devices whilst others are specific of protection mechanisms needed to provide a secure and trusted DRM system. The different issues found while implementing mobile applications are described in the following sections.

4.1 Device Incompatibility

The first and most important problem one finds when developing mobile applications is the incompatibility between the different models of mobile phones. This is also true for other kind of mobile devices, like PDA's, that have a completely different set of features compared with phones. Although OMA has made an important effort in this sense defining specifications for the different components that mobiles can include, the fact is that, when implementing a new application, it is almost impossible that it works for different manufacturers or even for different models of the same manufacturer.

One important reason for this incompatibility comes from the Java language libraries included into the mobiles and the restriction for the addition of new ones. Mobiles use the Java Micro Edition (Java ME) version of Java [16], which is hardcoded inside the device. There are two different versions of it, which provide functionalities that depend on the features of the mobile device where it is going to be included. The basic version is the one called Connected Limited Device Configuration (CLDC) [17], which works for small devices. The other version is called Connected Device Configuration (CDC) [18], which can be used in more capable devices like smartphones and PDA's.

When a manufacturer includes Java ME inside a mobile device, the inclusion of newer or more powerful Java ME libraries is not possible. So, if a newly developed application needs some functionality from a library not natively included in the mobile device, it will not work as the library cannot be installed.

This fact is especially critical when new libraries appear providing security or other kind of relevant functionality for DRM systems and applications cannot take profit from them, although the mobile device could hypothetically support them.

To overcome this problem, the solution we have applied is the implementation of different versions of the same application, addressed to different kinds of devices. The most usual solution adopted is the inclusion of an intermediate server that supplies the functionality provided by the not available Java library.

As we are implementing applications mainly based on secure web services, with the solution adopted, apart from the duplication of implementation effort, we also have a security problem as we need to establish a secure channel between mobile device and the intermediate server and between intermediate server and the secure web service, instead of directly connecting from the device to the secure web service.

4.2 Use of Secure Channels

Another problem found comes from the need of establishing secure channels with servers as many of the applications developed are not stand-alone applications installed into the mobile device. In our

specific case, client certification is also needed (not only server certification, that is the usual situation). This means that we need to import certificates in the device.

This is not an easy task, as, again, root CA certificates are hardcoded inside the device and the inclusion of new ones is difficult and they are not always accepted by the contracted servers, which can cause an application failure that has nothing to do with the application. This is not a specific problem of mobile devices, as the same applies to PC browsers and applications, which have some root CA certificates already installed and, if a user wants to trust in a new root CA, he has to install its certificate into the device's certificate repository. The need of installation of a new root CA certificate can cause a lack of trust from the user on the application that needs it. Nevertheless, the limited capabilities of mobile devices make this task a little more difficult in terms of installation facilities and error recovery.

4.3 Application Performance

Regarding application performance, we have found that the applications we have developed (basically for security, external connectivity and XML files processing) need a lot of resources from the mobile device in terms of memory, persistent storage and CPU processing capacity.

This is a handicap when trying to execute an application in a device with not enough processing resources, for instance when one wants to decipher an image or find an XML document inside the device persistent local storage.

Again, to overcome the problem an intermediate server is needed to perform the most expensive operations in terms of processing.

An example of functionality that can be done out of the mobile device is the authorisation of actions a user can perform over a multimedia content he has purchased. If we extract this module from the device we may gain in terms of persistent memory usage (licenses are not stored inside the mobile) and document processing time (licenses are described in XML language and its processing is quite expensive). On the other hand, we are forcing the user to be connected each time he wants to perform any operation over purchased content and this probably implies some connection cost that could be avoided if the module was inside.

An example of functionality that is not so easy to extract from the device is the deciphering capability, because in this case we can lose system integrity using an intermediate server. In this case, if a ciphered image is deciphered in the intermediate server and then sent to the mobile device, we have two security flaws (at least): user has the content available in clear in his device, so he can pass it to another user, and, unless the communication is secure, someone could capture the content in clear during its transmission from the server to the device.

4.4 Testing Applications

Another important issue found when developing mobile applications is that the destination device cannot be used for performing application development. This is not the case for other software developments, as the developing platform is usually very similar to the production one and the application can be tested during development. So, the development of the mobile applications is done in a computer using different mobile platform simulators for testing the functionality implemented. Then, after the application is working in the simulator, the acid test is to try to install and use it in a real device.

Depending on the technology used, the installation can be more or less easy. For instance, installing a Java midlet on a smartphone using Windows Mobile is quite easy, as the midlet can be copied into the corresponding folder using the connection with a computer (via ActiveSync application). Then, the Midlet Manager allows the installation of the midlet into the smartphone.

Nokia mobile phones also offer an application for synchronising mobile with a computer that also allows copying midlets into the device. Then, the midlet can be used from the mobile. Other manufacturers possibly offer similar approaches for midlet installation.

At this point, we have the midlet installed into the mobile device. It should be taken into account that we are describing applications that make use of external connection or security capabilities. These features usually require more complex resources than other kind of applications that can be usually running on the device.

So, what may happen when testing the midlet into the real device? First, we can find that the midlet runs, but that is not able to establish the connection with the server due to a number of reasons (libraries not present, certificates installed but not found, etc.). Moreover, it may occur that some application that works well in the simulator is very slow on a particular device. We have experienced this problem with the execution of symmetric ciphering algorithms like DES or 3DES on certain kind of devices. As an example, the ciphering and deciphering of an image of 152 KB in print network graphics (.png) format can take almost 4

times more depending on the mobile phone model and up to 10 times more than the execution in the WTK (Java Wireless Toolkit) simulator.

The conclusion we can reach is that the test phase in real devices may show that a mobile application is useless due to resource consumption or to the lack of other mobile features needed, that were available for the simulator.

5. Future Trends in Mobile Applications

From the above sections we might give the impression that working in DRM aspects with mobile devices is a loss of time and effort, but, taking into account the number of GSM and 3GSM connections in the world (around 3 billion [19]) maybe it is worth doing this effort.

We should also consider that many artists sell their songs or video clips inside mobile devices before selling them on traditional stores, giving an added value to the purchasers of the mobile device. In this way artists avoid final users downloading their new songs in an illegal way, passing them later to the device.

Moreover, mobile devices are no more consumption devices but also creation ones. Mobile phones and PDA's integrated video and photo cameras as well as digital cameras incorporating Wi-Fi connection facilitate creation and sharing of new content. The quality of the content obviously depends on the device, but, in a world where immediacy is a key value, the possibility of taking and sending a photo or a video of any event throughout the world in seconds should not be rejected when considering the implementation of new applications for mobile devices. We should also take into account that this content could be part of digital newspapers, where an increasing number of content sent from readers is being incorporated, or even television news, where accidents or tragic incidents occur and neighbours are the first in recording what is happening.

So, the implementation of security techniques, DRM and other applications like metadata annotation associated to content created by mobile devices has to be seriously considered as this content will have an increasing value in the near future. Not only final users can take profit from their devices, also professional photographers and reporters may be interested in the identification of their photos for its later selling and distribution. The use of intermediate servers that fill the missing information associated to content (for instance using MPEG-21 Digital Item [20] or the Digital Media Project (DMP) [21] Content Information (DCI) [22]) and the existence of registries of digital content will be very important to achieve the objective of giving added value to content created using mobile devices.

6. Conclusions and Future Work

In this paper we have presented how Digital Rights Management (DRM) applications based on different standards and international initiatives have been implemented to cover mobile devices. These applications have been implemented by the Distributed Multimedia Applications Group (DMAG) in the context of different research projects in the last years.

During the implementation of these applications, we have faced several issues regarding programming environments, device capabilities and application testing and installation. Different solutions have been taken to solve these issues, which include the development of several modules for implementing the same functionality in different kind of devices, the use of intermediate servers or the need of library re-implementation for improving its performance on specific mobile phones. The lesson learnt and different approaches taken to solve the different issues found have been described throughout this paper.

However, we believe that applying DRM and other security techniques to content addressed to mobile devices will become a key issue in the near future as they can render more valuable content thanks to their increasing processing capabilities and screen size. The work done by the Open Mobile Alliance in the definition of generic specifications for mobile environments is particularly relevant in the DRM topic.

We would also like to emphasise the new trends foreseen in the area of content management from mobile devices. We believe that content created from these devices will increase in value and importance as new business models could appear when final users may become photographers or reporters due to its proximity to events.

Finally, regarding the future work in the development of DRM related applications for the mobile environment, we will follow the evolution of the different standards and initiatives for this area, considering which parts of them best fit into new solutions that could be implemented, based on the requirements of the application to develop.

Acknowledgements. This work has been partly supported by the Spanish administration (DRM-MM project, TSI 2005-05277) and has been partly developed within AXMEDIS [13], a European Integrated Project funded under the European Commission IST FP6 program.

7. References

[1] Distributed Multimedia Applications Group (DMAG). In http://research.ac.upc.edu/dmag.

[2] Open Mobile Alliance (OMA). In http://www.openmobilealliance.com.

[3] ISO/IEC, ISO/IEC 21000-1, Information technology – Multimedia framework (MPEG-21) – Part 1: Vision, Technologies and Strategy.

[4] OMA, Digital Rights Management, Version 2.1, OMA-DRM-DRM-v2_1. In http://www.openmobilealliance.org/Technical/release_program/docs/DRM/V2_1-20070724-C/OMA-TS-DRM_DRM-V2_1-20070724-C.pdf.

[5] OMA, DRM Rights Expression Language, Version 2.1, OMA-DRM-DRMREL-v2_1. In http://www.openmobilealliance.org/Technical/release_program/docs/DRM/V2_1-20070724-C/OMA-TS-DRM_REL-V2_1-20070724-C.pdf.

[6] Open Digital Rights Language Initiative, Open Digital Rights Language (ODRL). In http://odrl.net.

[7] OMA, DRM Content Format, Version 2.1, OMA-DRM-DCF-V2_1. In http://www.openmobilealliance.org/Technical/release_program/docs/DRM/V2_1-20070724-C/OMA-TS-DRM_DCF-V2_1-20070724-C.pdf.

[8] ISO/IEC, ISO/IEC 21000-5, Information technology – Multimedia framework (MPEG-21) – Part 5: Rights Expression Language. In ISO/IEC Standards.

[9] ISO/IEC, ISO/IEC 21000-4, Information technology – Multimedia framework (MPEG-21) – Part 4: Intellectual Property Management and Protection Components. In ISO/IEC Standards.

[10] ISO/IEC, ISO/IEC 21000-15, Information technology – Multimedia framework (MPEG-21) – Part 15: Event Reporting. In ISO/IEC Standards.

[11] ISO/IEC, ISO/IEC 21000-5:2004/Amd.1:2007 Rights Expression Language, Amendment 1: MAM (Mobile And optical Media) Profile. In ISO/IEC Standards.

[12] Llorente, S., Delgado, J., Maroñas, X., Implementing Mobile DRM with MPEG-21 and OMA. In WOSIS 2007 proceedings. INSTICC Press.

[13] FP6 AXMEDIS project. In http://www.axmedis.org.

[14] Futurlink. In http://www.futurlink.com.

[15] Torres, V., Delgado, J., Llorente, S. Trusting software tools in a secure DRM architecture. In AXMEDIS 2007 proceedings. IEEE Computer Society.

[16] Sun, Java Technology for Mobile. In http://developers.sun.com/mobility/index.jsp.

[17] Sun, Sun Java Wireless Toolkit for CLDC. In http://java.sun.com/products/sjwtoolkit/overview.html.

[18] Sun, Sun Java Toolkit for CDC. In http://java.sun.com/products/cdctoolkit/overview.html.

[19] GSM World, Number of GSM and 3GSM connections. In http://www.gsmworld.com/news/press_2008/press08_31.shtml.

[20] ISO/IEC, ISO/IEE 21000-2, Information technology – Multimedia framework (MPEG-21) – Part 2: Digital Item Declaration. In ISO/IEEE Standards.

[21] The Digital Media Project (DMP). In http://www.dmpf.org.

[22] The Digital Media Project. 2008. Approved Document No. 3 – Technical Specification: Interoperable DRM Platform, Version 3.1. No.1103/GA17. In http://www.dmpf.org/open/dmp1103.zip.