How Open DRM platforms can shape the future of DRM

[1]Víctor Torres, [2]Carlos Serrão, [3]Jaime Delgado and [2]Miguel Sales Dias

[1]victor.torres@upf.edu, [2]carlos.serrao@iscte.pt, [3]jaime.delgado@ac.upc.edu, [4]miguel.dias@iscte.pt

[1]UPF – Universitat Pompeu Fabra, Departament de Tecnologia, DMAG, Pg. Circumval·lació 8, E-08003 Barcelona, Spain

[2]ISCTE/DCTI/ADETTI, Ed. ISCTE, Instituto Superior de Ciências do Trabalho e da Empresa, Dep. Ciências e Tecnologias de Informação, ADETTI, Av. Das Forças Armadas, 1600-082 Lisboa, Portugal

[3]UPC – Universitat Politècnica de Catalunya, Dept. d'Arquitectura de Computadors, DMAG, Campus Nord Mòdul D6, E-08034 Barcelona, Spain

Abstract

Looking at modern Information and Communication Technologies and at the variety of devices and gadgets with the potential to exchange information and share functionalities, we consider that interoperability is in fact a killer functionality. However, interoperability is not always easy to achieve. A clear example of such complexity is the lack of interoperability between Digital Rights Management (DRM) systems.

The authors strongly believe that the road to DRM interoperability has to pass through the definition of open platforms, that is, those containing open and defined interfaces to allow the exchange of the necessary data to interoperate, as opposed to other more closed and obscure solutions. In this paper, we analyze some of the currently available DRM architectures, focusing on those who provide open-source, open-specifications or open-interfaces, and identifying their functional modules. We also provide some hints on how those platforms could interoperate.

Keywords

K.4.4.d Intellectual property, H.5.1 Multimedia Information Systems

1 Introduction

One of the most perfect examples of interoperability in the IT world is the Internet. The Internet is a hardware, architectures and systems heterogeneous environment; however, every hardware device or application can exchange information in a common established way. This is only possible because there is a single standard communication protocol on the network bounding everything together (IP - Internet Protocol). Any system or application willing to use the Internet has to implement the mechanisms to comply with the IP specifications.

This is a straightforward way of providing interoperability; i.e., everyone agrees to follow a single standard. This approach works perfectly in the Internet case but it cannot be applied in other situations. Multimedia content is one of these situations. The multimedia World presents a panorama where almost everything is proprietary – content formats, media players, multimedia content protection mechanisms and multimedia rights management – and where no single standard exists that is strictly followed and implemented by everyone. Therefore, interoperability in multimedia Digital Rights Management (DRM) is far more complex to handle than in the Internet scenario.

Some authors [1] have suggested a set of approaches to DRM interoperability, based on International Standards: full-format interoperability, connected interoperability and configuration driven interoperability. In the full-format interoperability case, all protected content conforms to some unique globally standardized format. This is hard to accomplish, since all of the content providers and DRM software manufacturers would have to agree on the same file format to use. This is nevertheless the strategy that is being followed by Open Mobile Alliance (OMA). In OMA, the DRM Content Format (DCF) is a format that all the devices need to know and implement. In a second alternative approach, translation third parties can be used to translate operations from one DRM regime to another. This seems to have a more solid background and a set of translation entities may actually exist in the future, allowing the translation between different DRM functionalities to accomplish the same objective. In this approach a peer-to-peer architecture may be needed to be established in which each node allows an interface to its peers, and if it can not satisfy a direct request then it redirects the search to other peers. Another option for this approach is the “intermediated digital rights management” [2] that identifies four tasks to be carried out by the intermediary in transferring content in the format used by the content provider to the format required by the end-user. Rights management tasks are executed by a third party server (the intermediary) on behalf of the content scripts and end-users. The third and final approach for DRM interoperability upholds that by downloading adequate tools any DRM system can get the ability to process protected content on end users devices. This is also a valid alternative for the DRM interoperability problem, allowing devices and digital content rendering applications to “grow” its own capabilities and functionalities to enable different DRM regimes according to the ones governing the protected content. For instance, this is the DRM interoperability model that is uphold by MPEG-4 IPMP-Extensions [3].

But, is there any other reason why the Internet has became such a success in terms of interoperability? We all have to conclude that the major reason for this is openness – specifications openness, platform openness and even (in some cases) source code openness. This is why the authors uphold the same approach to address the DRM interoperability issues. From the three approaches identified previously to achieve DRM interoperability, two of them require open specifications availability to be really effective.

Therefore, in the following sections of this article we will introduce and describe three different open DRM platforms to further explore and clarify this open DRM approach for interoperability.

2 Openness and Interoperability

Interoperability is a serious and hot topic in the DRM community. The authors of this article are convinced, based on their past experience in the DRM field, that the DRM future will have to evolve through that way.

Interoperability can be seen from three different perspectives. If we consider the end-user side perspective, one of the most important interoperability aspects is the possibility of using the same legally owned content in multiple devices. However, this very simple and basic usage scenario is not yet possible today.

From the content provider perspective, interoperability is also a desirable aspect. First, by using interoperable content a wider range of users can be reached. Second, in the production side it is much cheaper to produce content in a common format than to produce specific versions for each available end-user device and to apply different specific protection mechanisms.

On the retailer side, the present model will give place to a more generic one capable of trading content for all the available platforms and increasing competition.

Currently, two major commercial DRM systems are used for content protection and rights management: Windows Media DRM (WMDRM) and Apple Fairplay. They are incompatible, since they have different file formats, protection mechanisms, rights management systems and support different devices. Content acquired on one platform will not work on the other. However, they share something in common: they are both closed DRM systems. They don’t have public specifications available, don't provide connectivity interfaces for other DRM systems and their source-code is not available.

WMDRM is the Microsoft DRM for Windows Media content. It is a closed DRM system; however, some few details about the platform are provided. Moreover, it is possible to get a Software Development Kit (SDK) from Microsoft, after signing a rigid contract agreement, which will allow to set-up and operate a Windows Media store, capable of protecting content and issuing licenses.

Fairplay, from Apple, is even more restrictive than WMDRM. No information is provided about the internal details of the system and there is no SDK available for developing a compatible store or device. It is a complete vertical model that supports the dominant position of the Apple iTunes store.

At the current state, there is little chance for WMDRM and Fairplay to interoperate (or any other solution that follows the same approach). Also, no single one will overcome all the others. The lack of interoperability between different DRM systems can lead to disappointing user experiences as well as jeopardizing the digital media concept. For example, it may happen that someone who purchases a new protected DVD is unable to reproduce it on their player due to the lack of interoperability between the DRM systems that have been used. Therefore, the only viable alternative is the open model uphold in this article.

3 Open DRM platforms

In our previous analysis of the DRM interoperability problem (see 1. Introduction), we have made references to three different approaches to address this problem. Nonetheless, the most viable ones to deal with this problem is to have intermediary brokers between the different DRM components and functions and also the possibility to extend these components through the download of new functionalities. In order for any of the two approaches to be possible, one of these two requirements must be met: 1) DRM components and functions should be openly specified, so that the present functionalities could be extended and new ones could be easily integrated or 2) the open and available implementation of such DRM components should be in an open source-code regime. The ideal scenario is the one where

both requirements are satisfied – however, such scenario is not common in the DRM field. Most of the currently available DRM platforms are neither open-specification neither open-source.

This section will focus both on open-specification and open-source platforms to identify a translation/brokering mechanism between the different functionalities provided by the different components of DRM platforms. In this study, one open-source implementation (OpenSDRM) and two open-specification DRM platforms (DMAG/MIPAMS, OMA-DRM) have been selected for analysis.

3.1 Open-source DRM platform

OpenSDRM (Open and Secure DRM) is an open specification and open-source implementation of a DRM platform. It originated from the European FP5 IST project MOSES [4], but since then its initial functionalities have been extended [5][6][7]. OpenSDRM relies on a distributed functionality paradigm where each of the DRM components are web-services, and all messages between the components, are SOAP-based, over SSL channels.

OpenSDRM (Figure 1) uses two important concepts largely referenced afterwards: Actors and Components. An **Actor** is a person, an entity or an organization that uses a **Component**. A **Component** is a tool co-operating to offer a set of DRM-related functionalities. Its architectural elements are described hereafter.

Figure 1 – OpenSDRM generic architecture

The **Authentication Server** (AUS) is used for Actors and Components registration and authentication. Its main functions are: (a) components registration in the system, including functionalities to update and to delete/revoke Components; (b) users registration that will interact with the Components. It is also used to update and delete/revoke users; (c) verify if a user has installed a valid wallet on their client; and (d) available payment gateways verification and validation.

The **Configuration Server** (CFS) registers the location of all the Components, so that they can be accessed from other components. The main functionalities of this server are: (a) component location registration and its details; and (b) specific component information provision to other components.

The **License Server** (LIS) creates, manages and delivers licenses. The main functionalities of this server are: (a) content protection keys secure storage; (b) user available licenses reporting; (c) specific user and content identifier license creation; (d) update and delete/revoke licenses on the system; and (e) license retrieval by end users systems. This mechanism bounds the license together with the content key(s) in a protected manner and sends them back to the end-user.

The **Protection Tools Server** (ITS) registers the new protection tools and receives authenticated requests for specific protection tool downloading. The server makes protection tools available to the CPS to allow the content protection. Major functionalities include: (a) new protection tools introduction; (b) available protection tools listing; (c) check the provided protection functionalities; and (d) available protection tools download.

The **Payment Gateway Server** (PGW) verifies and validates the payment methods provided by a User. Its major functions are: (a) request payment clearance from the billing system, clearing the user's payment instrument; (b) payment capture from a previously cleared transaction; and (c) subscription of new content stores to the billing system.

The **Content Preparation Server** (CPS) receives from a source and encodes raw content to a specific format, adds metadata and protects it according to a protection tool(s). The encoding format is out of the scope of OpenSDRM. The **Registration Server** (RGS) assigns unique identifiers to content. The main

functionalities are: (a) new content registration; (b) metadata registration; (c) list the available content and metadata on the system that matches with specific criteria.

The **Media Delivery Server** (MDS) registers the storage and delivery of content to the client. Main functions include: (a) store content location on the system; (b) notify the system of content user requests; and (c) notify the system of content downloads.

The **Commerce Server** (COS) is responsible for presenting and trading content with the users.

The **Wallet** is a client-side component that interacts with the OpenSDRM platform to enforce the DRM functions on owned DRM-governed content. Its main functions include: (a) user registration and certification; (b) wallet validation and certification; (c) sensitive information secure storage; (d) download of licenses whenever necessary; and (e) download of protection tools.

The **Content Rendering Application** (Media Player) component renders protected and governed content. The Media Player is connected to the Wallet to access information needed to obtain the necessary license(s) and key(s) to render the content. Its implementation is dependent on the type of content that is protected, while the DRM mechanisms are not.

3.2 Open-specification DRM platforms

3.2.1 MIPAMS (Multimedia Information Protection and Management System)

MIPAMS (Multimedia Information Protection and Management System) [8][9][10] manages multimedia information using DRM and content protection. The architecture consists of several modules and services, which provide a subset of the whole system functionality needed for managing and protecting multimedia content. MIPAMS is a service-oriented DRM platform and all its modules use the web services flexible approach.

MIPAMS encompasses an important part of the content value chain, from content creation and distribution to its consumption by final users (Figure 2).

Figure 2 – MIPAMS generic architecture

The **Certification entity** involves different functionalities: Certification Authority, Registration server and Supervision server. The **Certification Authority** issues X.509 credentials for the different Components and Actors in the system. Its main functions are: (a) to issue installed tool certificate (first time a tool is used), (b) issue user certificates, when users are registered and (c) issue component certificates for the different architectural elements. The **Registration Server** registers actors and tools. The actor's registration results in a user certificate, whereas tools registration is performed to verify them once installed on client devices. The main functions of the component are: (a) to register tools and (b) to register actors. The **Supervision Server** authenticates actors and system components. Moreover, it extracts and registers a fingerprint of the installed tools for future verification and for tool certification requests to Certification Server. It also verifies the client integrity during its operation by checking its fingerprint, registered during certification. Moreover, it receives the action reports regarding content consumption or other relevant issues in the system. The main functions of the component are: (a) to verify installed tools against registered tools, (b) register new installed client tool and fingerprint, (c) request installed tool certificate and (d) receive and store action reports.

The **Governance Server** creates, delivers, stores, and translates licenses to other languages when needed. It also checks if users have the appropriate rights to perform the requested actions over objects

according to the license terms in an online mode. Their main functions are: (a) license generation, (b) license storage, (c) online authorization and (d) translation support.

The **Protection Server** deals with the procedures used for digital objects protection. Main functionalities include: (a) protection tools description and download, (b) protection of objects and (c) protection keys generation and storage.

The **Content Server** main functionalities are: (a) users enabling to browse/select and purchase content, (b) provision of content that final users may request to user applications, (c) encoding and metadata inclusion to raw contents received from providers and (d) registration of the digital objects metadata, which can be stored independently from the resource itself. This functionality should be available for the Content Server itself as well as for client production tools.

The **Adaptation Server** adapts contents, their associated metadata or even licenses, depending on transmission, storage and consumption constraints. It could be included in the Content Server as an integral part of it.

The **Trusted Client** interacts with the client application to enforce DRM. It consists of a trusted software module and a secure local repository for licenses, protection information, offline operation reports and other critical data. Main functionalities include: (a) estimate trusted client and tool fingerprint, (b) require offline authorization, (c) unprotect content, (d) track offline operations, (d) store installed tool certificate and (e) store content protection information.

The **Intermediary** is usually an integral part of the trusted client but could also be located in the server part to reduce its complexity. It can be seen as a broker to whom the trusted client requires the authorization and keys needed to unprotect the content. Main functionalities include: (a) require verification to Supervision Server, (b) require online authorization to Governance Server, (c) send offline operations to Supervision Server and (d) download protection tools from Protection Server.

The **Application** is the player or edition tool which needs to deal with the protected contents. In the case of the player, it needs to get the unprotected content to be reproduced. In the case of a production tool, it needs to request authorisation before allowing the user act upon the objects as e.g. to modify an image or embed it in another object.

3.2.2 OMA DRM (Open Mobile Alliance DRM)

OMA DRM (Open Mobile Alliance DRM) version 2.0 [11] (Figure 3) has been developed to enable the controlled consumption of digital media objects by allowing content providers the ability, for example, to manage previews of DRM Content, to enable super distribution of DRM Content, and to enable transfer of content between DRM Agents.

Similarly to the previous DRM architectures, OMA DRM defines a set of Actors and Components in its reference architecture. The most relevant are the DRM Agent, Content Issuer, Rights Issuer, User and Off-Device Storage.

The **DRM Agent** (DRM-A) represents a trusted entity in a device. This entity enforces permissions and constraints associated with DRM content, controlling the access and usage of DRM content.

The **Content Issuer** (CI) delivers DRM content. OMA DRM defines the DRM content format to be delivered to DRM-A, and also defines the way the content can be transported from a CI to a DRM-A using alternative transport mechanisms. The DRM content packaging may be handled directly by the CI or it may receive it from an external source.

The **Rights Issuer** (RI) is the OMA DRM entity that assigns permissions and constraints to DRM content, and generates Rights Objects (RO). The RO is represented in XML and expresses permissions and constraints associated with DRM content.

A **User** is a human user of DRM content, which can only access DRM content through a trusted DRM Agent.

The **Off-Device Storage** allows DRM content to be stored off-device, for backup purposes, to free memory on the device. RO with stateless permissions can also be off-device stored.

The OMA DRM system enables CI to distribute Protected Content and RI to issue Rights Objects for the Protected Content. The DRM system is independent of media object formats, operating systems, and runtime environments. For User consumption of the Content, Users acquire Permissions to Protected Content by contacting RI. RI grants appropriate Permissions for the Protected Content to User Devices. The Content is cryptographically protected when distributed; hence, Protected Content will not be usable without an associated Rights Object issued and cryptographically bound to the User's Device.

The Protected Content can be delivered to the Device by any means. But the Rights Objects are tightly controlled and distributed by the RI in a controlled manner. The Protected Content and Rights Objects can be delivered to the Device together, or separately. The system does not imply any order or “bundling” of these two objects. It is not within the scope of the DRM system to address the specific payment methods employed by the RI.

The OMA DRM specifications define the format and the protection mechanism for DRM Content, the format and the protection mechanism for the Rights Object, and the security model for management of encryption keys. The OMA DRM specifications also define how DRM Content and Rights Objects may be transported to devices using a range of transport mechanisms. Any interaction between network entities is out of scope.

Figure 3 – OMA-DRM generic architecture

Before content is delivered, it is packaged to protect it from unauthorized access. A content issuer delivers DRM Content, and a RI generates a Rights Object. The CI and RI embody roles in the system. Depending on deployment they may be provided by the same or different actors, and implemented by the same or different network nodes.

DRM Content cannot be used without an associated Rights Object, and may only be used according to the permissions and constraints specified in a Rights Object.

Rights Objects associated with DRM Content have to be enforced at the point of consumption, which is modeled in the OMA DRM specifications by the DRM Agent. The DRM Agent embodies a trusted component of a device, responsible for enforcing permissions and constraints for DRM Content on the device, and controlling access to DRM Content on the device.

DRM Content can only be accessed with a valid Rights Object, and so can be freely distributed, as Rights Objects are cryptographically bound to DRM Agents. This enables super distribution, as users can freely pass DRM Content between them. To access DRM Content on the new device, a new Rights Object has to be requested and delivered to a DRM Agent on that device.

4 Open DRM platforms compared

Three DRM platforms have been described in the previous sections of this paper (OpenSDRM, MIPAMS and OMA-DRM). They all share some commonalities: either the specifications are public, or they provide public documented interfaces, or their source-code is publicly available.

OMA-DRM development has been supported by a wide community of companies and it is widely implemented and deployed in the marked by the major mobile phone suppliers (Nokia, Sony-Ericsson and others). OMA has already launched two versions of its DRM recommendations – the later one (version 2.0) reached its maturity in middle 2006 – and it is not yet widely implemented in mobile terminals. Both MIPAMS and OpenSDRM were born in the heart of the scientific research community, supported by public financing (National and European), and are mostly used by academic communities. MIPAMS and OpenSDRM continue to evolve with the aid of the community. OpenSDRM has recently been converted to an open-source project, and its source-code can be obtained at Sourceforge.

Although OMA DRM has been developed to address a very specific vertical sector – the mobile phone industry – it can be used, up to a certain extend, in other different usage scenarios as well. OMA DRM mandates a specific file format container – the DRM Content Format (DCF) – to hold the DRM-governed items. This is a key-point in the interoperability of the different devices developed by different manufactures and operated by different mobile Telecom providers. On the other hand, content, rights, business models and platform independence were the three major design goals of OpenSDRM. Therefore, OpenSDRM can be easily adapted to several types of content and to different content business models – in fact this has already been done since OpenSDRM has been used in different European research projects, dealing, for example with MPEG-4 [3] and JPEG2000 [13]. MIPAMS content format is based on MPEG-21 Digital Items, thus providing enough flexibility to handle many different content formats. Licenses are expressed in MPEG-21 REL, although MIPAMS provides mechanisms for the conversion from and to OMA DRM version 2.0 REL.

Both OMA-DRM and OpenSDRM define DRM architectures that deal directly with the distribution of rights to the final consumer. This means that they have mechanisms (the DRM Agent in OMA and the Wallet in OpenSDRM) that handle all the authorisation clearance and rights parsing at the user-side, having specific protocols for getting the rights from rights issuance entities (Rights Issuer in OMA and the License Server in OpenSDRM). OMA DRM implements a protocol called ROAP (Remote Object Access Protocol), while OpenSDRM uses a specific web-services call entry in the license issuer. MIPAMS extends this model by adding the possibility that the licenses issued are not final user licenses. MIPAMS supports final user as well as distribution licenses. This supposes that a content distributor will need to own a distribution license in order to be able to derive final user licenses from it, something that will be controlled when issuing the end-user license. On the other hand, MIPAMS also supports the usage of license-like documents that can be defined by content creators or content owners, which are useful to state the rights that a distributor could exploit over the content and the related conditions. Any distribution license, when created, must fit the rights and conditions stated in those license-like documents. The authorisation algorithm performed for final users against end-user licenses can optionally involve the verification of the distribution license from which they derive. In this way, the whole content value chain can be controlled.

From an architectural point of view, both MIPAMS and OpenSDRM platform are richer than OMA. From a functional point of view, there are functionalities that are shared between all of the three platforms. However, some extended functionality offered by MIPAMS and not considered in OMA is the following:

- Rights enforcement: MIPAMS enables local or remote enforcement of rights, whereas OMA performs it at the point of consumption.

- Supervision: MIPAMS Supervision Server enables post-usage payment and provides statistical, tracking and control functionalities, which are not considered in OMA.
- Certification and trust on clients: In OMA, the trust on DRM Agents is based on the possession of digital certificates. However, the periodic verification of the client modules integrity is not considered.

In the same direction, OpenSDRM is quite comparable with the MIPAMS DRM platform. In fact, OpenSDRM offers also some extensions when compared to the OMA-DRM platform, such as:

- The license template mechanism present in OpenSDRM allows an easy extension of the Rights Expression Language (REL) supported and OpenSDRM is therefore not particularly tied to any specific REL;
- The Wallet, which has a similar behaviour as OMA's DRM Agent, is capable of not only handling with registration and authentication processes, license download, enforcement and authorization, but also with the download of new protection tools;
- OpenSDRM offers the connection to payment functionalities, whereas in OMA no payment functionalities are mentioned;
- OpenSDRM DRM mechanisms are independent of the type of governed content, while OMA specifies its own OMA DCF format.

OMA also presents some advantages over the OpenSDRM and MIPAMS platforms:

- It is widely tested, supported and implemented in most mobile phones (in particular OMA-DRM version 1);
- It uses a common format, which in part simplifies the overall DRM mechanisms.

A key aspect of DRM is security. In this particular aspect, MIPAMS, OpenSDRM and OMA-DRM provide a secure environment for the DRM operations, making extensive usage of symmetric and asymmetric cryptography, digital credentials and secure protocols – such mechanisms are detailed on the platforms specifications and are out of the scope of this paper.

5 Interoperability Solution for Open Systems

The open model, as we referred to in section 2, is a model in which any DRM solution willing to provide interoperability (that is, several levels of interoperability) would have at least a set of public and documented interfaces, implemented in such a way that they would allow any other application or DRM solution to invoke such interfaces. This public interfaces would expose internal DRM functions, that would produce some type of result when invoked with the appropriate parameters. Consider a typical example: a user has acquired a music piece on Windows Media store and wants to play it on its iPod. The iTunes software would invoke the public interface of the Windows Media DRM License Server to download a license for content. Once the license is obtained the music can be played on iPod. This is a very simple scenario that can only be achieved if an open DRM model is adopted.

The availability of the source-code for this model is not a must, but it is desirable. If the source-code is available, the integration of elements is facilitated by several reasons. One is the possibility of checking how an interface works internally and make the invoking functions act in a more reliable way. A second is that anyone could define public interfaces for DRM solutions, providing wider choices for interoperability.

While considering open systems and open-source, usually a concern emerges – security. This is a very complex question that is out of the scope of this paper. But, if by one side, open-source systems are more

prone to scrutiny and therefore can be shortcut, on the other so can closed systems. In fact, one of the major motivations of people cracking these systems is just the simple fact and challenge that they are closed.

As in any other computing system that wishes to exchange information with others, there are two major approaches for this: either a specific private interface is defined to allow the communication between two single entities (1-to-1 relationships) or, in alternative, a generic open interface is defined to allow the communication with a generic third party that will route the information for the target system (n-to-n relationships). The second approach, more generic, can be seen as a kind of DRM Middleware, composed by a supra-set of DRM functionalities, capable of brokering and mapping each of the specific individual DRM functions into other specific DRM similar functions.

Our proposed approach for interoperability between the three open DRM solutions presented, which may be extended to any other open DRM solution available, can be summarized in the following:

- Identify the open DRM solutions that need to interoperate: this is the starting point of the process defined here because it will be necessary to identify if the interoperability requirements are met, to enable the identification of the core DRM functionalities;
- Identify which are the major DRM functions that each open DRM solution offers and the corresponding interfaces: this is an important step because it will enable the identification of the specific functionalities, the data formats, the input and output parameters and their behaviour. If possible these interfaces should be described using the Web Services Description Language (WSDL) to be self descriptive;
- Define a super-set of functionalities that any open DRM solution may require: this super-set of functionalities will be internally implemented in the DRM as a routing mechanism to the real open DRM solution;
- Implement a set of public and generic interfaces that map to the previously identified functionalities: the complete set of generic interfaces will be part of the DRM Middleware that can be implemented using a Service Oriented Architecture (SoA) approach;
- Create a location mechanism to allow different DRM open systems to interoperate with each other through the middleware.

For the development of the DRM Middleware solution our aim is to have, at the end, a global super-set of DRM functionalities that could be accessed through generic interfaces so that any DRM implementation could make use of any of them to interact and interoperate with other DRM systems.

As an example of the approach uphold here, let's consider the OMA and OpenSDRM case. Imagine that a user has acquired content that is governed by OMA DRM and is trying to use it on a OpenSDRM based player. OpenSDRM knows that the content is governed by another DRM solution and would have to contact the DRM Middleware to get the appropriate authorization to render the content. The DRM Middleware knows how to map the generic request into the specific open interface provided by OMA DRM, routing the authorisation through the appropriate OMA DRM rights issuer (this request may include the necessary credentials for user or device authentication required by OMA DRM). OMA DRM returns an answer containing the Rights Object to the DRM Middleware. This answer is parsed into the generic format used by the DRM Middleware and sent back to the OpenSDRM. If the OpenSDRM does not know how to handle the rights contained in the Rights Object, the DRM Middleware can also provide generic services for conversion between different rights expression languages, or handle the authorisations at the server-side.

From the interoperability and integration perspective, the openness is a “sine qua non” condition for the achievement of real DRM interoperability. This type of interoperability will bring to the digital content consumers a better user experience and increase the potential revenue of content authors and distributors.

6 Conclusions

There is a shift on the way users are using legal digital content. Naturally, they want to use their digital content as the same way they have always used analogue content. Consumers associations and Governments are starting to request and even impose to DRM vendors, interoperability between their products. Therefore, interoperability has become DRM's hottest topic.

In this paper, we have presented the reasons why existing closed DRM solutions prevent interoperability and the alternative to these solutions: open DRM solutions. We have presented and described three DRM platforms that follow this model, two of them emerging from the academic world and one defined and extensively used in the mobile industry. We have also highlighted the basic commonalities and differences between them.

Although we had focused our work in these three particular solutions, other open DRM approaches do exist and we would like to emphasize the excellent work being done by other open DRM related initiatives, such as the Digital Media Project (DMP) [14], MPEG-21 [15], Sun's DReaM [16], Coral [17] and OpenIPMP [18].

The current digital content market has been dominated by both Apple iTunes Fairplay and Microsoft Windows Media Rights Management. The Apple iTunes Fairplay represents the implementation of a market strategy vertically integrated and strongly dominated by Apple, which controls the full digital content value-chain, from the content producer side, until the end-user side, where governed content can only be played on Apple certified content rendering devices, the iPods. In the case of Windows Media Rights Management, the model is not so restrictive. The WMRM can be licensed to third parties so that content producers can control their own governed content production, licensing and distribution. This governed digital content can be rendered on any device containing the Windows Media Player software or in any software/device that is compatible with the PlaysForSure logo. These are the top dominant players in the digital governed-content market. However these two systems do not provide interoperability between each other.

On an unprecedented move, digital music providers and even authors are starting to refute DRM as solution for upholding their rights. Partly this is because these two market dominators are still back to back, and they didn’t find the way to make their solutions interoperate.

It is clear that closed DRM systems are not suitable for the interoperability problem and that a new open model should emerge to tackle this issue. Only this model offers the necessary conditions to achieve full interoperability between the different DRM solutions. In our vision, this interoperability will be achieved through the definition of a DRM Middleware that would map each of the specific DRM solutions functionalities into a super-set of generic functionalities, which could be provided by a Service Oriented Architecture.

At this point it impossible to predict what will happen in the following years in terms of DRM technology and which will be the market dominator. However, at least in the music market, the DRM copy-protection aspects seem to become progressively abandoned, and even some bands are turning their backs to the major record companies. It looks music content authors are starting to realize that the Internet and the Web is creating the opportunities for them to cut the middleman, and for them to provide directly their music through alternative channels. Some questions remain however: will they be capable of controlling their own content value-chain and survive despite the digital piracy that will continue to exist?

Nevertheless, the authors believe that the management of rights aspect of DRM is still valid and will continue to manage the way governed content can be used.

We need better rights management solutions, that are open and interoperable, and that take into account not only the content providers requirements but also user's requirements.

References

[1]. Koenen, R.H., et al, "The long march to interoperable digital rights management", Proceedings of the IEEE, 92:883-897, 2004

[2]. Schmidt, A, U,.et al "Interoperability challenges for DRM systems", International Workshop for technology, Economy, Social and Legal aspects of virtual Goods, Ilmenau, Germany, 2004

[3]. Serrao C., Dias M., Kudumakis P., From OPIMA to MPEG IPMP-X - A standard's history across R&D projects, In Special Issue on European Projects in Visual Representation Systems and Services, Image Communications, Elsevier, 2005

[4]. MOSES web-site, http://atlantis.tilab.com/projects/moses/, as visited in 30.11.2006

[5]. HICOD2000 web-site, http://www.hicod2000.org, as visited in 30.11.2006

[6]. WCAM web-site, http://www.ist-wcam.org, as visited in 30.11.2006

[7]. MediaNet web-site, http://www.ist-medianet.org, as visited in 30.11.2006

[8]. Torres, V., Rodríguez, E., et al. Use of standards for implementing a Multimedia Information Protection and Management System. In Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS 2005). IEEE Computer Society, 2005, 197-204.

[9]. Torres, V., Delgado, J., et al. An implementation of a trusted and secure DRM architecture. In On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops (IS'06). Lecture Notes in Computer Science, vol. 4277. Springer-Verlag, 2006, 312-321.

[10]. Delgado, J., Torres, V., et al. Rights and Trust in Multimedia Information Management. In 9th IFIP TC-6 TC-11 Conference on Communications and Multimedia Security (CMS 2005). Lecture Notes in Computer Science, vol. 3677. Springer-Verlag, 2005, 55-64.

[11]. Distributed Multimedia Applications Group (DMAG), http://dmag.upf.edu, as visited in 30.11.2006

[12]. Open Mobile Alliance (OMA), Digital Rights Management V2.0, http://www.openmobilealliance.org, as visited in 30.11.2006

[13]. Serrao, C., Dias, L., Serra, A., and Dias, M. (2006a). Jpeg2000 image compression and visualization for desktop and mobile clients. In Proceedings of the Atlantic Europe Conference on Remote Imaging and Spectroscopy, pages 8–14. AECRIS2006, Preston, UK, InderScience

[14]. Digital Media Project (DMP), http://chillout.dmpf.org, as visited in 30.11.2006

[15]. ISO/IEC, ISO/IEC IS 21000-4 – Intellectual Property Management and Protection

[16]. DReaM, http://www.openmediacommons.org, as visited in 30.11.2006

[17]. Coral Consortium, http://www.coral-interop.org/, as visited in 15.02.2007

[18]. OpenIPMP, http://sourceforge.net/projects/openipmp, as visited in 30.11.2006