OPEN-SOURCE SOFTWARE AS A DRIVER FOR DIGITAL CONTENT E-COMMERCE AND
DRM INTEROPERABILITY

1Carlos Serrão, 2Joaquim Marques, 1Miguel Dias and 3Jaime Delgado

1ADETTI/ISCTE, Ed. ISCTE, Av. Das Forças Armadas, 1600-082 Lisboa, Portugal
2IPCB, Av. Pedro Alvares Cabral N°12, 6000-084 Castelo Branco, Portugal
3UPC, Dept. AC, DMAG, Campus Nord Mòdul D6, E-08034 Barcelona, Spain

1{carlos.serrao, miguel.dias}@iscte.pt, 2marques@ipcb.pt, 3jaime.delgado@ac.upc.edu
Abstract

Open-source software (OSS) is one of the hottest trends in terms of the software development in the current days. It is highly discussed if this is the best model for software creation when compared to non-open-source alternatives, and most of the times this discussion is mostly driven by passion than by facts. The main rational behind this paper is to discuss the opportunities and challenges that exist behind the usage or not of OSS in the development of DRM technology, and how it can affect the existence of different interoperable DRM technologies. The authors believe that open-source is a valuable strategy to deal with the current DRM interoperability problems. Furthermore we will establish a bridge between the open-source software developments in terms of DRM (mainly the OpenSDRM project) and how it can act as a driver for DRM interoperability.

1. INTRODUCTION

Electronic commerce hype started several years ago with an established idea that every company willing to embrace this new way of doing business would grow extremely fast and would earn huge profits [7]. The initial e-commerce businesses started by selling physical goods through electronic channels and focused primarily on the growth of the number of customers rather than on profit earning – with some exceptions most of these early e-commerce adopters went bankrupt. Meanwhile, new e-commerce companies with more solid business models have emerged and some of them based their primary business on the distribution and selling of digital content. This has been possible mainly due to two factors – the bandwidth increase and the digital compression technologies. In addition, the desire to unveil the new potential offered by digital content has worked as an important push factor to drive business [8].

These two important digital content facilitators have also contributed to a new phenomenon in this field – the “perfect” digital piracy. Why do we call it perfect? Because it is perfect in the sense that it enables the possibility to make “perfect” unlimited copies and to have a “perfect” and free distribution channel – the Internet – to make it reach millions of end-users. However, it is interesting to notice that the above mentioned factors also provide a growth potential for legitimate digital content businesses. These types of businesses are currently flourishing all over the Internet and some particular music stores have already sold more than 1 million digital music tracks over the Internet. One of the most successful digital content businesses on the Internet is Apple iTunes – the Apple iTunes digital content store is associated with the success of Apple iPod, a digital content portable device.

Both the Apple iTunes and any other digital content business rely on a set of technological measures to protect and sustain the continuity of their business, through the employment of copy-protection and rights management mechanisms (typically known as Digital Rights Management - DRM). These mechanisms employ techniques and rules that try to prevent the non-authorized usage of digital content and the digital copyright circumvention enabling the user to use digital content in a controlled way. It is relevant to notice that although there are alternatives to the Apple iTunes digital content store, any buyer that acquires digital content from Apple iTunes store, any buyer that acquires digital content from Apple iTunes is automatically confined to iTunes software or to Apple iPod portable player – this means that any other user with any other digital content rendering application or portable content device is unable to use the legally acquired content from Apple iTunes store. This is a problem for modern digital content users, but at the same time it’s also a big issue for content providers, because they either need to make their content available for a different set of platforms.

Figure 1 - URGE and Apple iTunes: a perfect example of lack of interoperability
Interoperability is currently one of the most discussed and debated problems in the DRM – more particularly the current lack of interoperability of nowadays DRM solutions. The actual trend in DRM systems is one-to-one, in which there is a store that sells digital content that can be used in a particular content rendering application and on a specific brand of portable media player – this is too restrictive. An example of this is one of the latest digital content music stores that are currently being promoted over the Internet, its part of a joint venture between Microsoft and MTV and its called URGE. URGE is associated with the launch of Windows Media Player 11 and Windows Vista. If we take a quick look at the URGE web-site FAQ it is possible to notice that URGE is incompatible with Apple iPod. Therefore any iPod user will not be able to buy digital content from URGE, unless they buy another portable content player that is compatible with URGE’s DRM (Windows Media Rights Manager). Additionally there are some DRM providers that deliberately use incompatibility strategies to lock users into buying particular hardware content rendering devices (portable or not). This old fashioned business strategy tries to raise entry barriers to new players in the same field and at the same time are quite penalizing for end-users. Therefore, this is clearly a situation that is not very comfortable for them and the perfect situation is to have a vision with n-to-n DRM. N-to-n DRM is a vision of DRM in which each of the digital content providers could abstract from whatever DRM technology it will be used to deliver and distribute their digital content and with the warranty that it will reach maximum audience with a larger base of potential consumers. For final users, n-to-n DRM means that whatever content rendering software, whatever portable device content player, and whatever content store the user will always be able to access its legally obtained and owned digital content. This paper tries to provide some clues about where and how open-source software can help solve this problem, or at least work as facilitator for this. OSS can help and promote DRM interoperability enabling an attractive experience to end users. Many authors have already recognized that DRM interoperability is a difficult task, and Koeman et al [11] suggests three approaches, based on International Standards to solve this issue:

• Full-format interoperability – all protected content conforms to some globally standardised format. This is a hard thing to achieve, since all of the content providers and all the DRM software manufacturers would have to have an agreement of the same file format to use. This is, nevertheless the strategy that’s being followed by OMA DRM – in OMA, the DRM Content Format (DCF) is a format that each of the devices need to know and implement.

• Connected interoperability – translation third parties are 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 exist on the future, for instance implementing web-services that will allow the translation between different DRM systems to accomplish the same objective – to enable DRM interoperability between different DRM providers. In this approach a peer-to-peer architecture is established in which each node allow an interface to its peers, and if it can’t satisfy a direct request them redirects the search to other peers. Another approach is the “intermediated digital rights management” [4] where are identified four tasks to be carried 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.

• Configuration driven interoperability – by downloading adequate tools any DRM system can get the ability to process protected content on end users devices. This is also a more valid and viable alternative for the DRM interoperability problem allowing each device and each digital content rendering application 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 [15].

Full-format interoperability is clearly the most advantageous for final users, affording a convenient way to enjoy and share digital content. However, it is highly improbable that a universal format emerges that is suited to all the present and future needs of digital content, applications and devices. Security also plays an important role in this issue because once a breach is detected on a standard, usually, standard bodies do not move with the required speed to patch it. OSS has an excellent opportunity to circumvent this issue.

2. DRM AND OPEN-SOURCE SOFTWARE

The introduction of different models of OSS development has revolutionized the way software is designed, produced, distributed and managed. The main idea behind most of the OSS development is to share knowledge, in which the developed knowledge and software is shared in the hope that another developer could add more value to the previous developments and continue this share process – there are exceptions and
variations to this, but in essence they share the same goal. This model enhances win-win relationships because it makes available software and source-code for free and at the same time it incorporates new developments made by third parties. This development model is completely open in the sense that anyone can make changes to the source-code and make them available for others. There are some purists that defend that open-source can only be considered like that if the full stack of software is also open-source. For instance, for these purists only software developed to target open-source operating systems, or the Apache Web server can be considered as open-source. The authors of this paper do not uphold this model since it is to limiting – we defend the logic that open-source is all means of software that releases the source of the binary result. As an alternative to this software development model, there is a well established traditional closed source approach. In this model, we include not only the commercial off the shelf (COTS) software but also any binary-only form of software distribution, such as freeware. It is not the objective of this paper to discuss whether open-source is better or not than closed-source software, but the authors rather discuss if the applicability of the open-source model to DRM systems development can benefit or not the interoperability aspects of the DRM systems themselves. One of the most important misconceptions in open-source software that is particularly important for DRM systems is about open-source software insecurity. In special this misconception is related to the fact that the software is less secure because everyone can look at the source-code to look for flaws and vulnerabilities. The fact is that the availability of source code has several security advantages. First, it facilitates the scrutiny by many people to find and flush out weaknesses in the design and code processes. Second, when a vulnerability is found, fixes or workarounds can be made without waiting for the code authors if the vulnerability is a critical one [16, 17]. Third, the fixes made also can be studied and scrutinised to ensure that they are made correctly. Finally, independent code checks and audits can only be made with source code availability. Another important advantage of source-code availability is that in the event that the platform on which one uses the application has been made obsolete by the vendor, it is still possible to continue using the application safely on the obsolete platform even if vulnerabilities have been discovered for the application in question since one can download the updated source and re-compile it. For closed-source applications one has no choice but to upgrade to the newer supported platforms.

An important issue that needs to be brought also to discussion is the fact that the unavailability of source code does not mean vulnerabilities cannot be discovered. Modern debugging and software development tools can be used to examine, disassemble and reverse-engineer the close-source executables to look for bugs and vulnerabilities. The most glaring illustration of this is the never-ending vulnerabilities being discovered for Microsoft products like IIS, IE and Outlook – just to mention a few. Another important concept and analogy was long time ago proposed by Kerckhoffs [2] which defended that a cryptosystem should be secure even if everything about the system, except the key, is public knowledge. Later on this concept was extended by Eric Raymond [3] to open-source as well, saying that any security software design that doesn’t assume the enemy possesses the source-code is already untrustworthy. It seems clear that security through obscurity will never work and that open-source software can be as secure as closed-source software. This is not and will never be an impediment for the development of DRM systems.

3. DRM AS OPEN-SOURCE

There are currently a set of OSS DRM initiatives that make available a set of software and the correspondent source-code. The following initiatives that are listed on the following sections reflect the number of OSS projects that are currently addressing the DRM issue and some of them are at the same time addressing the

![Diagram](attachment:media-s drm architecture.png)
DRM interoperability problems. There are perhaps more, but the authors consider Media-S, OpenIPMP, DReaM and OpenSDRM some of the most important ones.

3.1. Media-S

Media-S is an open-source DRM solution owned by SideSpace Solutions. Although Media-S claims to be format-independent, the available release only supports the Ogg Vorbis open-source audio codec. In terms of architecture, Media-S is divided into three components – a packager, a client and a license server (Figure 2). Media-S is based on XML (for the creation of licenses and data in licenses), and an open-source version of SSL (OpenSSL). For security, the Media-S DRM architecture separates licenses from content, and uses a combination of user and device authentication. The lack of a globally unique identity mechanism means that it could be easy to fake the license from one DRM protected to another; and it has also the potential to confuse the DRM client; if two different DRM had the same identifier it could contribute to a difficult penetration on mass market. Another problem with Media-S is the use of non-standard license Rights Expression Language (REL). Media-S uses their own REL specifications, even with the availability of two well documented REL standards (such as ODRL and XrML). These two facts contribute to major difficulties for other DRM systems to interoperate with Media-S [6].

Media-S DRM was released later on 2003 and it seems to have had a low impact on the DRM industry. As an open-source project, Media-S has released their last version of the source-code in 2003 – so it seems to be stagnated and with low activity.

![Figure 3 - The openIPMP DRM architecture](image)

3.2. openIPMP

The openIPMP DRM was developed by Objectlab, and is based on open standards, including MPEG-4 and implements one of the leading MPEG-21 proposed content identification schemes to solve the challenge of uniquely identifying digital assets. The openIPMP DRM includes an implementation of the Intellectual Property Management and Protection (IPMP) specification for MPEG-4. openIPMP allows for the expression of licenses using a choice of the leading rights expression languages, such as MPEG-REL and ODRL. Within the openIPMP framework, each MPEG-REL [19] or ODRL XML license [13] is signed by the openIPMP Certificate Authority for authenticity and is cryptographically specific to its intended recipient.

openIPMP is a framework (Figure 3) composed by a collection of tools and services capable of delivering a robust, scalable, and adaptive infrastructure to support the management and secure delivery of multimedia content. The openIPMP DRM is conformant to industry standards enabling interaction between various hardware and software products not locking the user to proprietary solutions. To manage rights and enforce role based rules and permissions, the framework needs to be able to uniquely identify every user of the system. The openIPMP DRM issues each user a Digital Certificate (or Digital Id) when they register with the system and enables secure, confidential communications with the openIPMP server components ensuring that sensitive data is not compromised during transit.

This open-source DRM is an active project that has recently released the second version of the openIPMP platform adding support for OMA, ISMAcrypt and MPEG-2. openIPMP due to its extensive support and to its distributed architecture has the potential to interoperate with other DRM solutions.

3.3. DReaM

DReaM is a Sun initiative to develop a DRM solution focusing on open-standards. According to DReaM own information, whenever the market
requires proprietary solutions DReaM will be capable of integrating with these solutions providing openness and interoperability that meets customer requirements. DReaM is an initiative to leverage the methodology of Service Oriented Architectures (SOA) and introduce rights management services that leverage open standards and support cross-service capabilities [5].

The DReaM architecture supports the separation between the rights management components (through the decoupling of authentication, licensing, rights management and protection systems). This disintermediation enables the choice and selection of these technologies independent of each other without any compromise for the overall solution. There are two key elements for disintermediation in DReaM:

- Separation of rights management from the content protection systems;
- Separation of identity and authentication services from individual hardware devices.

DReaM has a central objective towards the creation of an interoperable DRM, offering the capability to inter-operate directly with other content protection technologies and supporting services that enable both Conditional Access System (CAS) and DRM. A key-concept in the DReaM platform is the disintermediation concept – this enables multiple instances of these components to exist in a DRM/CAS system. The DReaM disintermediation system (Figure 4) enables the coexistence of multiple instances of content protection specific components (player, licensor and packager) and components that are not content protection specific (disintermediation agent, conductor, catcher, licensing conductor, contracts manager, authentication service, shop and transaction system and content delivery system).

### 3.4. OpenSDRM

OpenSDRM was the selected name for the implementation of a set of server-side software components that implemented the supported functionalities needed by the MPEG-4 IPMP-X implementation on the European-sponsored MOSES project. OpenSDRM, in its genesis has two words that seem contradictory by nature – open and secure – but that are not [9]. Meanwhile, OpenSDRM has evolved and is now an open-source project and the source code is available under a LGPL license.

The OpenSDRM infrastructure (Figure 5) was designed with the concern to be adaptable and applicable to all types of content, business models and distribution channels (download, super-distribution, streaming or even broadcasting) [10]. This way is possible to enable media involving multimedia to function, by enforcing licensing agreements for content use and offering business opportunities to the content rights owner and content provider. OpenSDRM implementation has followed the horizontal approach where the implementation has followed the guidelines and specifications of open standards. This way is possible to address DRM interoperability by defining common interfaces, tools and mechanisms that different DRM solutions should comply. OpenSDRM is also based on the emerging Service Oriented Architecture (SoA) of the W3C consortium [12], providing a distributed nature to each of the major DRM functions and also a clear interface defined in WSDL for external services to integrate and a communication mechanism based on SOAP. OpenSDRM uses ODRL [13, 14] to express rights, a language for rights expression derived from an international effort aimed at developing and promoting an open standard for the DRM expression language which has been adopted by OMA. Nevertheless the
platform can be easily expanded to use other RELs due to a license template mechanism.

The OpenSDRM development is targeted primarily for WAMP and LAMP environments and makes use of open-source tools such as PHP and MySQL. Due to its distributed and open nature, OpenSDRM is a completely scalable platform that has potential for integration with other DRMs.

4. OPEN-SOURCE DRM AND INTEROPERABILITY

In the previous section we have briefly reviewed and analyse some OSS that address DRM. This revision has revealed that by opposition to the many commercial existing DRM platforms now have some serious open alternatives. This analysis has just covered some selected projects although there are some others, like PachyDRM, Marlin and Digital Media Project (DMP) that deserve attention as well. PachyDRM is a DRM platform that is primarily targeted for the mobile phones market with a lightweight implementation of the OMA-DRM standard. DMP is currently working on its reference software. Both and in particular DMP put a strong emphasis on the interoperable aspects of DRM.

We have also introduced in section 1 different approach to DRM interoperability based on standards. On the other hand, we have pointed out the use of OSS as an approach to DRM interoperability, and we have shown different specific OSS solutions for DRM.

However, not all of them really address interoperability from a global point of view, and when they do so, they have not yet fully achieved it, since DRM implies many different issues and it is not easy to handle all of them.

So, there are two dimensions to the interoperability problem. One directly deals with the DRM complexity, and means that we need to handle protection (encryption, decryption, watermarking, key distribution, etc.), authorization based on licenses (rights expressions, verification, license distribution, etc.), metadata, enforcement, governance, authorities and others.

A second dimension could be related to how we try to get interoperability, and we could define different DRM interoperability levels. Then, the interoperability of DRM systems could be classified in the following levels:

- Proprietary systems;
- Standards and architectures;
- Software framework based;
- Open Source.

The first level, the “old” solution of trying to achieve interoperability by forcing everybody to use a specific closed system is, as we have previously stated, clearly not a feasible solution. The “Standards and architectures” level could be a solution, to be approached, as indicated in section 1, from three different points of view, i.e. full format, connected and
configuration-driven interoperability. Different standardization organizations and industry and use fora are working in this direction, trying to specify standards at different levels, from REL to full interoperability architectures. One problem with this approach is the selection of the “neutral” format/protocol/architecture that should be used as a gateway to interoperate with the “others”. Since the number of these “standards” is growing, a solution in this direction is getting more difficult to reach. However, this does not mean that standards are not useful; on the contrary, standards for expressing rights expressions, for expressing the semantics of rights and conditions or for specifying an algorithm for authorizing access to a protected content, as simple examples, are really necessary.

There is an intermediate level before we go to the OSS level (the one we have been discussing in the paper), what we call “software framework based”. The idea is the development of a set of tools, i.e. a “software framework” that is publicly available and that DRM systems implementers could use to build their required system. Although this framework could be also open source code, there are a series of reasons to avoid a fully open source model. The idea is that these tools offered in the framework are guaranteed by someone, i.e., a trusted organization that should certify all implementations of the tools (this would make a complete OSS rather complex to manage, although not impossible). The AXMEDIS European Project [21] is developing one of these frameworks that will contain tools not only for DRM, but also for multimedia content creation, distribution, consumption and management. The final business model to use is still under discussion, but the specifications of the tools are already available. It is finally worth mentioning that standards are followed when available and interoperability between standards, mainly through mappings, is also foreseen.

The openness of OSS with the availability of source is an additional level to facilitate DRM interoperability. However this may be not enough. It is also important to have a good distribution/separation of DRM functionalities and a clear interface between those functions and possible external systems.

There is a problem in modern days with open-source projects – open-source may not entirely mean that it is free. In particular, in the DRM case, most of the technologies and standards applicable are encumbered by patents. This in fact means that the source-code and the resulting implementation may be free, but the usage of such software on a commercial exploitation may be not. One good example of such is the case of ODRL and OMA. ODRL is a REL that is open; however, there are registered patents that claim to cover the possibility of expressing rights using XML, so in fact all the REL implementation that use XML to express rights would need to pay royalties to the patent owner if they would request so and a Court would not declare the patent as not applicable. Several researchers are trying to demonstrate that the concepts covered by those patents are older than them [20]. This is an important obstacle to the OSS DRM deployment. Directly, this has an impact on DRM interoperability as well – this would almost eliminate OSS DRM from the interoperability processes because they will not most likely pay any fee to any patent holder.

On the other hand there is also the issue of GPL version 3. GNU Public License (GPL) is the license that governs most of the OSS projects in the World. Its principles are quite simple – someone who uses GPL-ed source-code in its software development must also release the resulting source-code under a GPL license. GPL ensures that no one could actually take a piece of source-code (or even an entire software) developed by someone else and close it, brand it and sell it as if it was his own. So, why is GPLv3 such a problem for OSS DRM? Because, in essence, the new GPL license will prevent the use of GPL governed source-code in the development of DRM applications. As a free software license, GPLv3 intrinsically disfavours technical attempts to restrict users' freedom to copy, modify, and share copyrighted works. Each of its provisions shall be interpreted in light of this specific declaration of the licensor's intent. Regardless of any other provision of GPLv3, no permission is given to distribute covered works that illegally invade users' privacy, nor for modes of distribution that deny users that run covered works the full exercise of the legal rights granted by GPLv3 [18].

These two last factors may have a deep impact in the DRM OSS projects, and consequently on DRM interoperability as well. The first is not only directly applicable to OSS but the second one is very restrictive for OSS DRM implementations. Therefore DRM interoperability, although very desirable and achievable in OSS DRM gets seriously compromised by this. However the doors to DRM interoperability are always open due to the availability of either the source-code or by very well and documented interfaces.

On the other hand the lack of interoperable open-source DRM solutions will always create opportunities for closed non-interoperable solutions to get circumvented – it is curious to notice that most of the attacks that iTunes and the FairPlay system have suffered in the past have to do precisely with the unavailability of the iTunes platform on Linux.

5. CONCLUSIONS

Unfortunately DRM has a bad image on end users. Most of the measures and mechanisms implemented by content providers to enforce some control mechanism
over digital content clash directly with end-users requirements – a classical example: a user acquires a copy-protected CD on a traditional music store and is unable to play it on its PC or on its car player. This is completely unacceptable by end-users and motivates the strongest “anti-DRM” feeling and invades the user’s right to use the legally acquired content. Another example is the case where a user acquires content that is governed by a DRM regime and is unable to use the content on devices that do not implement any (or implement a different) DRM regime. This interoperability problem is still an issue that needs to be solved for DRM to mature and become less obtrusive.

Most of the commercial DRM solutions existing today follow a completely vertical strategy, in which they are incompatible between them. These solutions are closed and do not offer any opportunity for interoperability – in fact, as part of their business strategy, interoperability is seen as a bad thing and as a menace to their own sustainability. On the other hand there are the content producers that wish to increase their digital content distribution and controlled usage and the end-users that want to use digital content as similar as possible to their analogue counterparts. To achieve these two goals, DRM Interoperability is crucial.

We and other authors have already introduced some of the strategies needed to achieve DRM interoperability; however these strategies can only be accomplished if the current and any future DRM solutions either provide open-specifications and/or source-code to allow this interoperability. At least, the most crucial DRM functions of each of the DRM systems must be publicly specified and source-code libraries (free of charge) should be made available to allow others to build the translation or bridge mechanisms between the different DRM solutions. Therefore, it is clear that OSS can play an important role in DRM interoperability and that traditional proprietary solutions will have to adapt to a World where governed digital content should have no barriers and its usage should be done without any technical restrictions.

6. REFERENCES