# Using Web-Services to Manage and Control Access to Multimedia Content

Carlos Serrão, Miguel Dias and Jaime Delgado

***Abstract*— In a largely interconnected World, the Web-Services (WS) computing paradigm is gaining momentum. Most Web Services applications existing today are being developed in the E-Business or E-Commerce context, mainly for Enterprise Application Integration (EAI) [12]. This paper describes a distributed architecture that largely uses WS technology to control and manage the access to multimedia content and that represents the new and emerging market of Digital Rights Management (DRM). This architecture deploys some critical DRM elements, in a service-oriented architecture, such as device and user identification and authentication, content registration and protection, license representation and production and payment.**

**This paper presents the conceptual architecture, referred to as OpenSDRM [3], and provides some technical details about its development and deployment.**

***Index Terms*— SOA, DRM, SOAP, Security Layers, XML Security, n-Services, Service Enabling Platform**

## I. Introduction

The demand for digital content is growing, especially in the audio-visual and entertainment sectors. The roll-out of broadband network capabilities, the improvement of media compression technologies and the overwhelming availability of P2P network sharing applications, are factors that togheter, make content available for everyone [7]. This is not, in essence, a negative point but the real socio-economic and other problems arise when the content exchanged is subject to copyright infringements [8]. Content rights holders (authors) and more specifically, content owners (distributors, editors), are placing a huge pressure on both technology providers and legislators to put an end to this copyright circunventiom situation, responsible for causing substantial economic and financial loss. Examples of this are, for example, the RIAA process against the illegal MP3 distribution phenomena [5].

Large Information Technologies (IT) companies such as Microsoft, InterTrust, Apple and Real, are researching and developing technological measures that enforce private license contracts to enable the distribution of digital content, while preserving its copyrights. However, they are not alone in their efforts. The European Commission through its Research Frameworks programs is sponsoring a set of R&D projects to tackle this Intellectual Property and Copyrights circumvention issues. Digital Rights Management is an issue that is one of Europe's top priorities and is addressed in many projects and programs. MediaNet, a 6th Framework Programme Integrated Project is one of such European funded projects, in the area of Networked Audiovisual Systems and Home Platforms, that is dealing with these kind of issues [1].

MediaNet addresses the domain of digital multimedia personal communication and content distribution, as well as cooperation schemes between content owners, service providers, network access operators, and telecommunication, computer, components and consumer electronics industries [1]. The objective is to remove the obstacles to end-to-end digital communications and content exchange, from content/service providers to customers and between persons, over shared broadband access and home network infrastructures at the same time [2].

Assuming an open system reference architecture model, MediaNet partners jointly study a number of critical constituents of the on-line delivery chain (the e-media chain), made of various technologies, equipment or services, that are considered as pre-requisite elements for the creation of a myriad of new media services, supplied by multiple – and sometimes cooperating sometimes competing - providers and vendors in Europe [2].

MediaNet is integrating a DRM architecture, referred to as OpenSDRM, in the MediaNet Reference Architecture, in order to provide DRM-features for some selected MediaNet Use-Cases, which require DRM support [1]. OpenSDRM was primarily developed for a previous EU funded project (IST MOSES), which has provided valuable contributions to the specification, implementation (Reference Software) and validation (Conformance Testing) of the first ISO standard dealing with DRM issues, the MPEG-2/4 IPMPX

OpenSDRM is a platform that integrates some critical DRM elements that were developed using a completely distributed architecture. These critical elements include: content protection, rights expression, license management, content and metadata registration and payment [4]. This integration is being performed specially with another

Manuscript received April 10, 2005.

Carlos Serrão and Miguel Dias are with Adetti/ISCTE - Ed. ISCTE – Av. das Forças Armadas, 1600-082, Lisboa, Portugal; (e-mail: Carlos.Serrao@iscte.pt , Miguel.Dias@iscte.pt.)

Jaime Delgado is with UPF, Departament de Tecnologia, Pg. Circumvallació 8, E-08003 Barcelona, Spain (e-mail: jaime.delgado@upf.edu).

platform called Service Enabling Platform (SEP) [2], a middleware architecture which provides the necessary services to home networks applications provided by network operators. One of such services is DRM, provided by OpenSDRM as a SEP external functional capability.

## II. The OpenSDRM platform

OpenSDRM is a service-oriented DRM platform, independent from the type of content, the protection applied to a specific content and the business model under which the content is consumed. It can be used with multiple communication protocols and is based on the emerging service-oriented paradigm (SOAP, WSDL and UDDI) approach [3]. OpenSDRM, which is depicted in Figure 1, covers most of the content lifecycle phases: from content authoring, distribution and management of the related rights up to the final user.

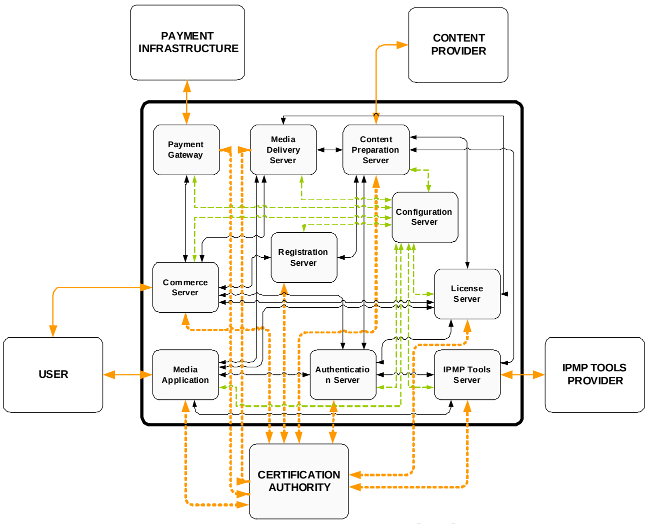


Fig. 1. OpenSDRM service-oriented architecture

The OpenSDRM infrastructure was designed having in mind concepts such as content adaptation and business models wide range applicability (download, super-distribution, streaming or even broadcasting). In a more technical approach, OpenSDRM is composed by a set of external actors or elements and a set of internal components [3]. The internal components are oriented towards the service they supply, and are described in more detail in the next section. These internal components are self-descriptive, in the sense that they expose an open WSDL description of the services they provide, and any authenticated component or actor can connect to it and use its services – DRM services. These components communicate with each other using SOAP messages [13]. The discovery and identification of services is currently being provided by a configuration server, but on a near future this service will be provided by an UDDI server.

### A. *External Components & Actors*

The components and actors that may interact externally with the OpenSDRM architecture are, the User, the Protection Tools Provider, the Content Provider, the Payment Infrastructure and the Certification Authority [3].

The User represents an entity who wishes to use some content. This content may or may not be protected. However the way to access and display such content may require the use of protected devices, software and licenses. The User will make requests to OpenSDRM in order to: provide identification information, perform authentication, download licenses and use the content.

The Protection Tools Provider is any organization that produces tools and technologies for encryption, scrambling, watermarking and others that can be applied to content protection. These tools will be made available to OpenSDRM for use in content rights protection. These tools will need to comply with some guidelines. These guidelines and a subscription, are translated into a business relation that must exist between a given Content Provider and the Protection Tools Provider, since mostly, a given producer and/or distributor of content, may want to choose which type of protection the content will have and, respectively, which tools can be applied to the content and from which supplier.

The Content Provider is any multimedia content supplier that feeds a Commerce Platform or a Content Management System, connected to the OpenSDRM with content and optional metadata.

The Payment Infrastructure facilitates the commercialization of content. OpenSDRM plays an important role since it provides the services for handling electronic payments. The interface between OpenSDRM and the Payment Infrastructure is generic and independent from the payment method, allowing therefore a multiplicity of payment systems.

The Certification Authority is responsible for receiving requests for and issuing credentials to entities. These credentials will be used by entities to authenticate themselves to each other, allowing the establishment of secure and authenticated communication channels between them (this is part of the establishment of one of the two OpenSDRM's security layers). All the components in the OpenSDRM architecture communicate using the channel security provided by the SSL/TLS protocol. This Certification Authority may be internal to OpenSDRM, and therefore entirely managed by some entity, or it may be an external commercial entity, such as Verisign or Thawte [4].

### B. *Internal Components & Interfaces*

The internal components of the OpenSDRM platform include: Media Application, Media Delivery Server, Commerce Server, Authentication Server, License Server, Protection Tools Server, Registration Server, Content Preparation Server and the Payment Gateway [3].

The Content Preparation (CPS) is a component responsible for the content preparation. It receives raw content from a specified source or sources and encodes it on a specified format, adds metadata and protects it. It is not implemented using the WS approach, although it uses some components that provide such approach.

The Payment Gateway (PGW) is a server component responsible for verifying and validating the payment methods provided by the User to a Commerce Server.

The Commerce service (COS) is an integration component responsible for establishing the liaison between the platform that actually supplies the content and the DRM platform. Normally, content is chosen via web browser, some very generic metadata might be consulted, information about the price is also available, and especially the content usage conditions might be established.

The Media Delivery service (MDS) is a server component responsible for notifying the appropriate content servers that a given content has been requested and that needs to be feed to the final user.

The content Registration service (RGS) is a server component whose role is to assign unique identifiers to content and to register metadata information for that specific content. The service assigns unique identifiers to content using the MPEG-21 directives about Digital Item Identification (DII), using a reduced version of the MPEG-21 DII Digital Object Identifiers (DOI) [6].

The Authentication service (AUS) is a key-component of the system. It is responsible for authenticating all the internal services and components as well as some external actors to the DRM system. It validates the access rights of all of them working as a single sign-on point, registering and managing components and users on the system. It uses cryptographic XML credentials to authenticate both components and users in order to authenticate the transactions exchanged between them (XML Encryption and XML Signature) [10, 11].

The License manager service (LIS) is a server component responsible for house-keeping the rules associating a user, the content and his/her corresponding access rights. This component will accept connections from authenticated content rendering application clients for downloading licenses, which will be applied to the protected content through an appropriate protection tool. The licenses are XML formatted using Open Digital Rights Language (ODRL/OMA profile) or the Rights Expression Language (REL), developed by MPEG-21 [6].

The Content Protection tools service (ITS) is the server component responsible for registering new protection tools and for receiving authenticated client content rendering application requests for the downloading of a specific protection tool. It is also responsible for making protection tools available to the Content Preparation service to allow the protection of content.

The Content Rendering Application (MPL) is client-side component that represents the software that will be used to render the content. This is a generic component with the particularity of being able to display/playback the appropriate content for which the necessary audio/video codec should be available (if this codec is not available it must be downloaded from a remote secure server).

## III. OpenSDRM Security

OpenSDRM is in its essence a service-oriented and distributed architecture. The communication that takes place between the single components, using usually insecure networks. This introduces special needs regarding the security of this communication. As it was already introduced before, OpenSDRM uses two different security layers as depicted in Figure 2. The first security layer is established at the communication level, which will provide the necessary secure and authenticated communication medium to components to communicate with each other. This first layer is using SSL and X.509 certificates that are installed on the servers allowing this first authentication and ciphering layer. The second security layer is established at the application level, ensuring the security, integrity, authentication and non-repudiation mechanisms needed by the different components which provide the OpenSDRM services.

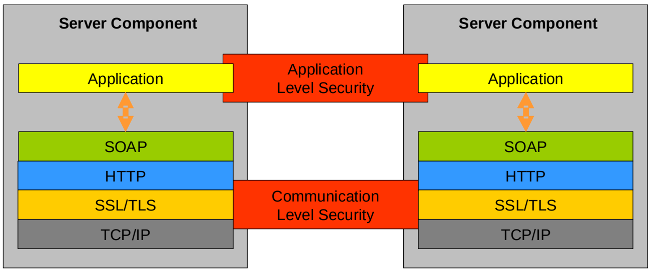


Fig. 2. OpenSDRM security layers

This second layer, which corresponds to the exchange of SOAP messages internally between the different OpenSDRM services, and between external actors and those same services, uses the possibilities offered by the two W3C XML security-related recommendations: SOAP extensions to use XML Signature and XML Encryption [13]. This is part of the new XML security-related architecture (WS-Security). The authentication between components is provided using a specific developed mapping of X.509 certificates in XML.

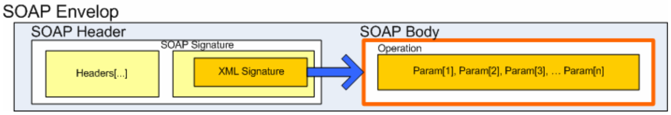


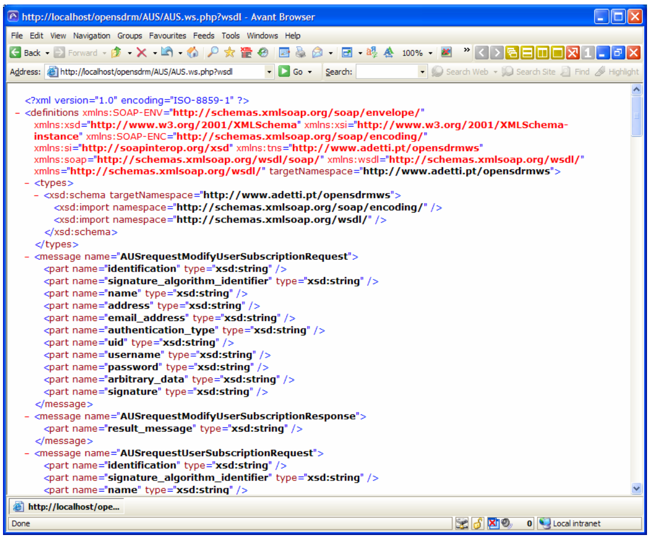
Fig. 3. Format of SOAP messages

Each of the services that are part of the OpenSDRM platform uses this two layered security model. One might think that this is an overlap, or even an overuse of security techniques, however they are not. While the underlying layer provides the authenticity of the web-servers (where the services are deployed) and the confidentiality of the communication channel between these servers, the upper layer provides only authentication, integrity and non-repudiation for the different services and actors that interact over the platform, which may be independent of the web-server itself. The other main reason for using SSL is because the OpenSDRM communications are mostly point-to-point and without big scalability requirements.

### A. *OpenSDRM implementation*

The OpenSDRM deployment focused primarily on the selection of open-source tools available for both Windows and Linux platforms to allow a wider range of content protection and user management scenarios. Another concern in the development, was the fact that it had to be based on openly available specifications and standards, such as ODRL [14]. The selected architecture was based on a LAMP approach (LAMP – Linux, Apache, MySQL and PHP), although the final solution was required also to run on MS Windows operating systems. The selected development environment was open-source based and also available for these two platforms (Apache 2.0, MySQL 4.0 and PHP5).

The core part of the service platform was developed entirely in PHP, using the NuSoap library, which provides PHP with both SOAP and WSDL capabilities. This allowed the possibility that the services could be almost auto-descriptive, generating automatically the WSDL, whenever invoked with the appropriate parameters (using for instance http://localhost/opensdrm/AUS/AUS.ws.php?wsdl). On the other hand, some other components of the OpenSDRM platform, responsible for the integration between some client tools and the rest of the platform, were developed using other technologies such as C# and Java. This highlights the truly interoperable nature of OpenSDRM and its service oriented architecture.


http://localhost/opensdrm/AUS/AUS.ws.php?wsdl - Avant Browser
File Edit View Navigation Groups Favourites Feeds Tools Windows Help
Back Forward 100%
Address: http://localhost/opensdrm/AUS/AUS.ws.php?wsdl Go Search:
Search Web Search Site Find Highlight
http://localhost/ope...
Done 0 Local intranet

```
  <?xml version="1.0" encoding="ISO-8859-1" ?>
- <definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
   instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
   xmlns:si="http://soapinterop.org/xsd" xmlns:tns="http://www.adetti.pt/opensdrmws"
   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
   xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://www.adetti.pt/opensdrmws">
 - <types>
  - <xsd:schema targetNamespace="http://www.adetti.pt/opensdrmws">
      <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
    </xsd:schema>
   </types>
 - <message name="AUSrequestModifyUserSubscriptionRequest">
    <part name="identification" type="xsd:string" />
    <part name="signature_algorithm_identifier" type="xsd:string" />
    <part name="name" type="xsd:string" />
    <part name="address" type="xsd:string" />
    <part name="email_address" type="xsd:string" />
    <part name="authentication_type" type="xsd:string" />
    <part name="uid" type="xsd:string" />
    <part name="username" type="xsd:string" />
    <part name="password" type="xsd:string" />
    <part name="arbitrary_data" type="xsd:string" />
    <part name="signature" type="xsd:string" />
   </message>
 - <message name="AUSrequestModifyUserSubscriptionResponse">
    <part name="result_message" type="xsd:string" />
   </message>
 - <message name="AUSrequestUserSubscriptionRequest">
    <part name="identification" type="xsd:string" />
    <part name="signature_algorithm_identifier" type="xsd:string" />
    <part name="name" type="xsd:string" />
```

Fig. 4. Sample WSDL generation for OpenSDRM services

Our approach makes possible OpenSDRM to be able to interface with external content management systems, content e-commerce platforms or even client-side content rendering applications. The latest are completely independent and unrelated to the OpenSDRM platform, up to the moment where they received protected content and need to render it. This process starts a specific client-side OpenSDRM middleware, which connects the client rendering application with the OpenSDRM license manager service, so that the client can obtain the necessary license and the corresponding keys to access the content. This is only one interaction example among many others. This is a much reasonable process, rather than to have proprietary license download protocols, which depend on the content rendering application.

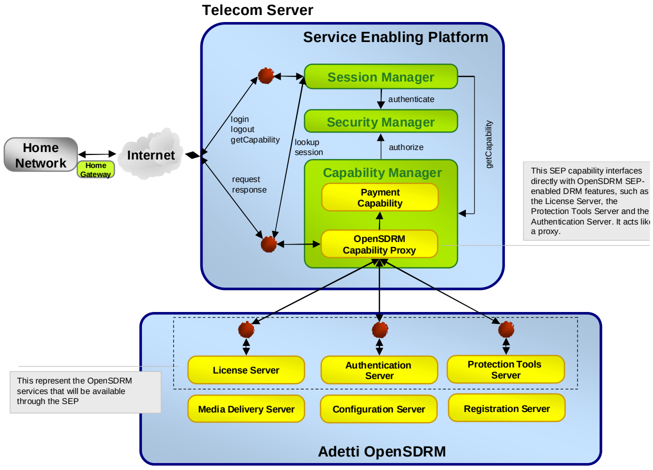
## IV. Integration between OpenSDRM and the MediaNet SEP

The DRM platform that it is being described on this paper is currently being used and tested in different usage scenarios: music download, surveillance video-streaming and interactive access to JPEG2000 images. However, in the scope of the MediaNet project, the DRM platform will be used for the first time in the context of Home Networking linked with the Telecom Application Service platform. providing DRM capabilities to several identified use cases [1].

As mentioned, the MediaNet project aims at enabling a myriad of multimedia applications over a common reference architecture. To define this common reference infrastructure as well as the interfaces between the stakeholders, one or more use cases are being proposed by the project partners. These use cases will be employed to define and validate the MediaNet Systems Reference Architecture [2]. Within this Systems Reference Architecture, the n-Services Platform is a software platform aiming at accelerating the introduction of innovative, value-added services by application service providers [2]. This platform, which is – in the scope of MediaNet - located in the network access provider domain, exposes generic capabilities, the so-called n-Services, through standardized interfaces and technology. These generic n-Services can be (re-)used by multiple application service providers [1]. The availability of an n-Services platform offers advantages to the network access providers as well as the application service providers. On the one hand, it enables the network access provider to resell capabilities to multiple application service providers and to charge these providers for usage of the capabilities. On the other hand, the availability of generic capabilities enables the application service providers to outsource certain generic aspects of their application to the network access provider, while focusing their own developments on the functionality that is specific for their applications. The Service Enabling Platform (SEP) is the instantiation within MediaNet of an n-Services Platform [2].

The SEP is composed by a different set of components, but in terms of integration three of them are the most relevant: the Session Manager, the Security Manager and the

Capability Manager. The Session Manager is responsible for establishing and maintaining sessions for clients (e.g. application service providers). The session manager is always active and its reference is registered in a Naming Service. This reference is the entry point for external applications/clients to obtain access towards the services provided by SEP. The Security Manager authenticates users and authorizes operations and/or access towards resources based on the user's security profile, managed by external applications. Security profiles can be defined in a static "off-line" way or through a security administration client. The Capability Manager consists of a capability repository, an interface to browse or discover supported capabilities and a register for capability addition/removal/change notification subscription. One of the capabilities that will be referenced by this Capability Manager will be the DRM support of the OpenSDRM platform.



This SEP capability interfaces directly with OpenSDRM SEP-enabled DRM features, such as the License Server, the Protection Tools Server and the Authentication Server. It acts like a proxy.

This represent the OpenSDRM services that will be available through the SEP

Fig. 5. Integration between SEP and OpenSDRM

Any client application and user on the Home Network is required first to establish a session with the SEP at the Telecom Application Server. This is the first entry point of any SEP-enabled application. The SEP authenticates the user or application through the Security Manager and assigns a unique session identifier to it. After this process the client can request capabilities from the SEP.

One of the capabilities that may be requested by clients and application is DRM, provided by OpenSDRM, which is considered an external capability of SEP. The integration of both is made through the usage of an OpenSDRM Capability Proxy (OCP) which redirects the requests made by SEP clients to the corresponding service endpoint in the OpenSDRM platform. The OpenSDRM services which are directly connected to SEP and using this OCP, are the License Manager service, the Authentication service and the Protection Tools service. Whenever the client or application is trying to access to DRM-protected content, it connects to the SEP platform, authenticates and requests the DRM capability, to be able to access to the corresponding license to obtain the clearance to access to content. The capability reference is then returned to the client application, which in turn can connect to the services provided by OpenSDRM, through the SEP OCP. All the messages exchanged between the client application and the SEP, as well as the ones exchanged between the SEP and the OpenSDRM platform, are SOAP-based.

## V. Conclusions

Web-services have been initially considered as "yet just another EAI technology" that companies could use to facilitate the integration of their own internal business applications. However the service-oriented nature of web-services can be used for much more than EAI use cases [12]. One of such applications is the one described on this paper that reports its usage on the Digital Rights Management field.

OpenSDRM is an open service-oriented platform that is used to control and manage the IPR associated with multimedia content, and we have shown that this is an area of application where web-services can play a major role. The distributed nature of this platform allows the seamless provision of DRM critical functions [7] over the network access of an end-to-end media delivery distribution reference architecture. This is especially important for the interoperability between the DRM functions provided by the OpenSDRM platform [3]. Due to this interoperability it is possible to have multiple content management applications or content commerce sites to use the same standard services provided by the platform.

The presented DRM platform takes advantage of the new XML/WS [13] security model to provide an application-level security layer to uphold the identification, authentication and non-repudiation of data exchanged by the different actors and components. At the same time, this level upgrades the security of the underneath SSL communication channel.

This DRM platform is used to manage the IPR of contents of several business models. One of the most innovative uses of this framework, within the MediaNet project, is in use cases that deal with Home Networking, linked with the Telecom Application Service platform. The DRM functions required by content providers and/or by the end-user applications can be provided by the SEP as an external capability of the platform [2]. Therefore, any application or any service provider with DRM requirements can fulfil such requirements requesting them form the OpenSDRM platform via the SEP. On the Home Networking side, any user or application willing to access DRM-protected content can request access clearance to that content through the SEP [2]. These services may include functions such as user authentication, content protection tools download, license and content keys download and, in the future, billing and payment.

## REFERENCES

[1] Travert S., Lemonier M., “The MediaNet Project”, WIAMIS2004, April 2004, Lisbon

[2] Eisink J., “DA.3.1 - Digital Home Network Reference Architecture Definition”, MediaNet project internal deliverable, December 2004

[3] Serrão C., “Open Secure Infrastructure to control User Access to multimedia content”, WEDELMUSIC2004, September 2004, Barcelona

[4] Serrão C., Neves D., Kudumakis P., Barker T., Balestri M., "OPEN SDRM – AN OPEN AND SECURE DIGITAL RIGHTS MANAGEMENT SOLUTION", IADIS 2003, Lisboa

[5] Siegert G., Serrão C., "An Open-Source Approach to Content Protection and Digital Rights Management in Media Distribution Systems", ICT Conference 2003, Copenhagen December 2003

[6] Bormans J., Hill K., "MPEG-21 Overview v.4", ISO/IEC JTC1/SC29/WG11/N4801, 2002

[7] Duhl J., Keroskian S., "Understanding DRM Systems", IDC Whitepaper 2001

[8] Rosenblatt B., "Solving the Dilemma of Copyright Protection Online", JEP Journal December, 1997 vol3, ISSN 1080-2711

[9] Multimedia Description Schemes (MDS) Group, "MPEG-21 Rights Expression Language WD V3", ISO/IEC JTC1/SC29/WG11/N4816, 2002

[10] “XML Signature Syntax and Processing”, W3C Recommendation, February 2002, http://www.w3.org/TR/xmldsig-core/

[11] “XML Encryption Syntax and Processing”, W3C Recommendation, December 2002, http://www.w3.org/TR/xmlenc-core/

[12] Race S., “Business Value of Web Services”, CommerceNet, March 2003, http://www.commerce.net/docs/bvowsr.pdf

[13] “SOAP Security Extensions: Digital Signature”, W3C Note, February 2001, http://www.w3.org/TR/2001/NOTE-SOAP-dsig-20010206/

[14] “Open Digital Rights Language”, ODRL, August 2002, http://odrl.net