Using ODRL to express rights for different content usage scenarios

Carlos Serrão, Miguel Dias and Jaime Delgado

***Abstract*— The expression of rights over generic content is one of the most important functions in any DRM system [1][2]. It is impossible to conceive such a system without the possibility to define how and under which conditions content can be used by the end-user and any other user in the content lifecycle chain. ODRL [14] represents an opportunity to have rights expression richness, flexibility and at the same time openness.**

This paper addresses those characteristics in the ODRL language by providing examples on how ODRL is currently being used in several content usage scenarios, such as music download and streaming, video-surveillance data streaming and storage and remote sensing of JPEG2000 images.

This paper also makes a short reference to the OpenSDRM architecture [3][4], an open DRM system that uses ODRL as its rights expression language, providing an interoperable rights enforcing layer. This layer acts as middleware to enforce the expressed rights over the content, through the provision of the digital Wallet concept [3]. The module which implements this concept is capable of accessing the rights locally or over the network, interpret and enforce them to the requesting content applications.

***Index Terms*— ODRL, OpenSDRM, REL, Wallet, XML**

I. Introduction

The Rights Expression definition is one of the most relevant functionalities of any DRM system [1][2]. It allows the expression of rights which are associated with a particular content and with a specific user and usage. Although this is important, rights expression is only effective if it is associated with technology that can enforce such rights on the content [1][2]. This paper describes and discusses a system, based on a client-side digital Wallet that works as an intermediate layer between the final user content rendering applications and the rights expression language. This technology is also associated with the description of license templates by different License servers. This paper also provides three different scenarios where this system is being applied together with ODRL [14]. These scenarios include the electronic commerce of digital music on a portal, the streaming of video-surveillance data and the controlled access to remote sensing images in JPEG2000 format.

The first scenario refers to one of the most attractive types of content exchanged over the Internet – digital music. Although this represents an opportunity for music producers that can use a larger massive channel to reach new consumers with radically different business models, it also represents a menace due to increasing copyright infringements [4][15]. Most of these infringements to copyright are performed while exchanging music over P2P networks. This scenario has already been developed and tested, and has been deployed on a service, which is referred to as Music-4You [26]. On this scenario, ODRL was used to express the licenses that described the rights of a certain user to access the content. Although this is an interesting scenario, it is not a new one.

The second scenario focused in this paper is the storage and streaming of video-surveillance data using JPEG2000 [18][19] (in particular Motion JPEG2000). This scenario, currently under development, uses ODRL licenses to express the rights of a particular user to access to the video-surveillance data. This scenario was recently demonstrated in the WCAM European Project, under the FP6 IST framework [21].

The final scenario which this paper describes relates to the usage of ODRL to express the rights to access JPEG2000-based Earth Observation products. This scenario has also been developed and demonstrated in the HICOD2000 European Space Agency project [25].

The paper is structured as follows: in section 2, a short description of OpenSDRM, an open DRM platform, will be presented with a specific focus on how the platform generates and manages licenses [3]. In section 3, we present the technique used by a middleware layer to manage the licenses and the rights at the client-side. Section 4, tackles the Usage Scenarios: music download and streaming; video-surveillance streaming and storage; and remote sensing of JPEG2000 images. In section 5, we extract some conclusions.

II. The OpenSDRM solution

OpenSDRM is a service-oriented DRM platform [3][4], independent from the type of content, the content protection system and the implemented business model. It can be used

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 Universitat Pompeu Fabra, Departament de Tecnologia, Pg. Circumval·lació 8, E-08003 Barcelona, Spain (e-mail: jaime.delgado@upf.edu).

with multiple communication protocols and is based on the emerging service-oriented paradigm (SOAP [13], WSDL and UDDI) approach [3], called Service Oriented Architecture (SoA). OpenSDRM (Figure 1) covers most of the content lifecycle phases: from content authoring, distribution and management of the related rights up to the final user.

The OpenSDRM platform (Figure 1) was designed having in mind concepts such as content adaptation and a wide range of business models applicability (download, super-distribution, streaming or even broadcasting). In a more technical approach, OpenSDRM is composed by a set of external actors (red circles) or systems (orange square) and a set of internal components (inside the yellow center square) [3]. The internal components are oriented towards the service they supply, and are described in more detail in the next section. From a more technical point of view, 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 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 this service will be provided by an UDDI server. OpenSDRM makes an extensive usage of ODRL to specify and manage the rights associated to content in each of the presented scenarios [3][4].

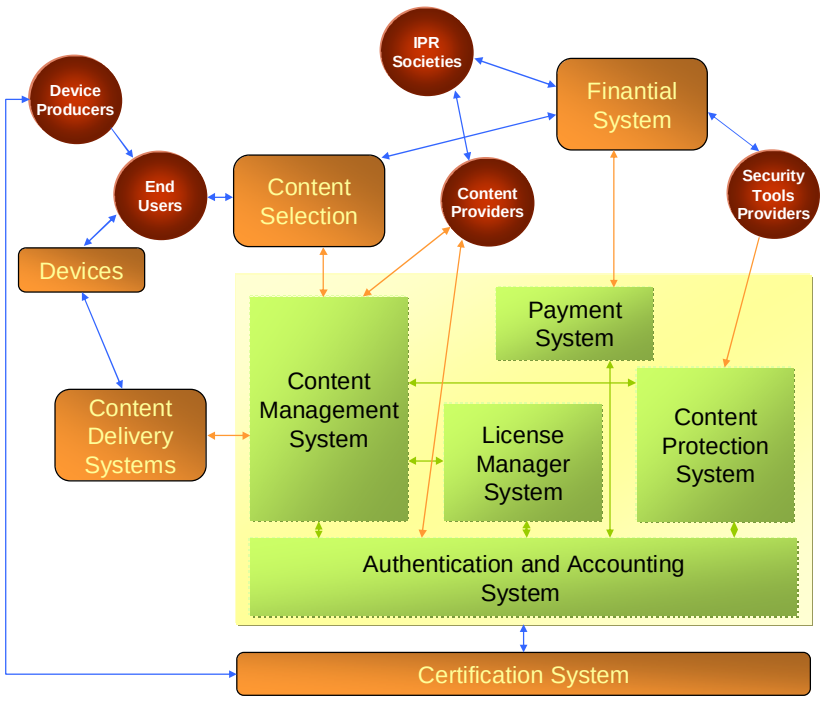

Figure 1 – OpenSDRM service-oriented architecture

A. *External Actors and Systems*

The main external actors and systems that interact with the OpenSDRM architecture (Figure 1) are: the End-Users, the Security Tools Providers, the Content Providers, the Financial System, the Content Selection System, the Content Delivery System, the Devices and the Certification System [3].

The **End-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 **Security 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 are registered and made available to OpenSDRM for use in content rights protection. These tools will need to comply with some guidelines, defined by the platform manager. These guidelines bound together with a subscription, are translated into a business relation that must exist between a given Content Provider and the Security/Protection Tools Provider. 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. This information and content will be made available to End-users.

The **Financial System** 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 **Content Selection System** is the module on which the End-Users can select the content that they want to enjoy. This can take the form of an Electronic Commerce site or an Electronic Program Guide.

The **Content Delivery System** is the system which is responsible for delivering the content to the End-Users or to the End-Users devices. This system is a generic entity that can be instantiated with any kind of content delivery system (download, broadcast, etc.) that is independent from the rights management system itself.

The **Device** is client-side system that represents the software or hardware that will be used to render the content. This is a generic system 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).

The **Certification System** 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 [3]. This Certification System 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 [3][4].

B. Internal Components & Interfaces

The main internal components of the OpenSDRM platform are: Content Management System, License Manager System, Payment System, Content Protection System and the Authentication and Accounting System [3].

The **Content Management System** is a system responsible for performing several functions. This system is responsible for content preparation and protection, content registration, content selection and trading and content delivery.

- Content preparation and protection: 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.
- Content registration: a function which 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 [6] directives about Digital Item Identification (DII) [7], using a reduced version of the MPEG-21 DII Digital Object Identifiers [6][7][16].
- Content selection and trading: is an integration function 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.
- Content delivery: is a function 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 **License manager System** is a system 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).

The **Payment System** is a system responsible for verifying and validating the payment methods provided by the User to the Content Management System while acquiring content.

The **Content Protection System** is the system 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 **Authentication and Accounting System** is a key-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].

All the above systems are interconnected and they were developed using a web-services paradigm: SOAP (Simple Object Access Protocol) and WSDL (Web Services Description Language). Each of these services is self-explanatory in terms of describing its external interfaces which allow the entrance of new components in a simple and seamless way. On the other end, each of this identified components exchange their messages, recurring to the SOAP protocol.

III. LICENSE MANAGEMENT ON THE SYSTEM

One of the more interesting mechanisms that are described on this paper relates to the fact that licenses are handled at the client-side by a middleware layer, called OpenSDRM Wallet [3][4]. This Wallet (Figure 2) is capable of managing

the access to protected content by different content handling applications. Every time an application wishes to perform an operation over the content, it contacts the Wallet that authorizes or not such operation according to what is specified on the license. This layer allows the coexistence of many DRM-protected files and DRM-enabled applications on a single client system, presenting a horizontal approach to DRM.

The OpenSDRM Wallet (Figure 2) is at the same time a Windows component which is responsible for holding down some of the user private information, such as some authentication credentials which allow the user to perform electronic payments to support the acquired content. The Wallet can store information in a secure way, either locally in the final user's computer (on an encrypted file-system or on the registry) or remotely on a server (on an encrypted database).

Figure 2 – OpenSDRM Wallet running

Nowadays, most of the existing DRM approaches are essentially vertical: examples of these include Microsoft Windows Media Rights Management (WMRM) [5][8] or Apple iTunes [9]. While a solution like Microsoft WMRM is a Microsoft end-to-end system-dependent (even at the client-side) relying on Windows Media Player to obtain the licenses and enforce them on the content [5][8][12], OpenSDRM follows a more horizontal approach in which several content applications can share the access to content, mediated by the OpenSDRM Wallet. This fact provides an important client-side interoperability layer. At the same time this approach also provides server-side interoperability since clients are independent from the server where they obtain the licenses.

A previous and important step is executed between the content application and the OpenSDRM Wallet in order to authenticate the application so that it can request content operations to the Wallet (this may include receiving content deciphering keys provided in the licenses). This means that any of the applications that wish to use this system will need to know how to execute an enrolment process composed by the following two steps:

- Enroll and request authentication to the OpenSDRM Wallet, exchanging a set of credentials with it, to enable application authentication and the establishment of a secure channel between the application and the Wallet – this secure channel will be unique by for the application and the Wallet;
- Request authorization to the OpenSDRM Wallet to perform operations over the content. This process includes the extraction of content unique identifier and requesting the Wallet the permission to use the content. The Wallet is responsible for getting the license from the server, parsing it; analyzing the rights that are associated to it before giving permission or rejecting the operation over the content (this may include passing the decryption key to the application or the appropriate protection tool).

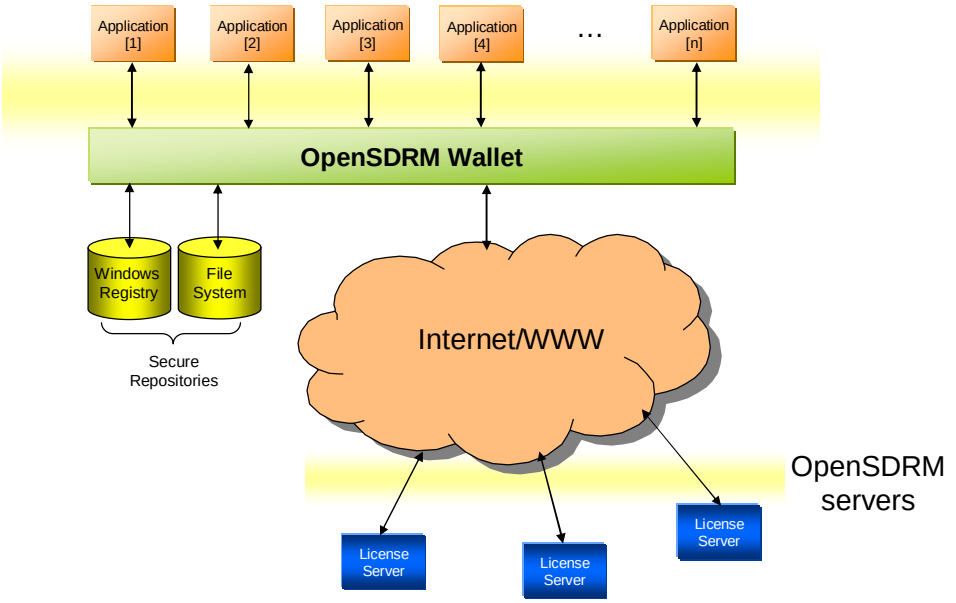

Figure 3 - OpenSDRM Wallet mediating Access to licenses

In order for this to work, the client content application does not need to know anything about the Rights Expression Language (REL) that is being used to express the rights. Therefore, this simplifies these applications design, and more, provides a layer of interoperability of the different RELs that may be used to express the rights.

Nonetheless, the content rendering application will need to be able to perform the following operations (already sketched before):

- Establish a trust relation with the underlying OpenSDRM Wallet, during a specific enrolment mechanism through the exchange of cryptographic credentials;
- Define and use a simple transaction protocol with the Wallet to request access to content operations. This transaction protocol is based on requests from the application and answers from the Wallet. An example could be a music player application asking permission to the Wallet to play a protected music track once. In this case the music player sends a message to the Wallet: “RENDER CID1234”. The Wallet receives this message and verifies that the User has a valid license. If the evaluation process is positive then the Wallet returns the key to render the content: KEY;
- Implement the necessary mechanism to establish a link with the content protection technology to be able to render it.

On the server-side of the OpenSDRM solution, one or more License servers can issue licenses (ODRL formatted, for example) [14] that are bound to the user and content. These licenses specify how content can be used by the user, according to a set of pre-established parameters.

The system contemplates the existence of one or more License Servers on the system, and also the possibility that each of the License Servers can issue more than one license type. In many current real cases, the License Server is strongly linked with the place where the content is obtained and with the implemented business model. This situation creates most of the times an unnecessary complexity in licenses issuance and management, and also trends to work as an interoperability blocking force. However, our approach tries to minimize this problem.

The foreseen OpenSDRM model is the one in which many content supply services can exist with business relationships with multiple License servers. These License servers can issue multiple licenses to many users – it is a many to many relationship.

With this idea in mind, OpenSDRM uses a template system for license creation [3]. This system allows the definition of the business model (or models) for each content business by the definition of the specific parameters that can be modified on a pre-created ODRL license template (Listing 1).

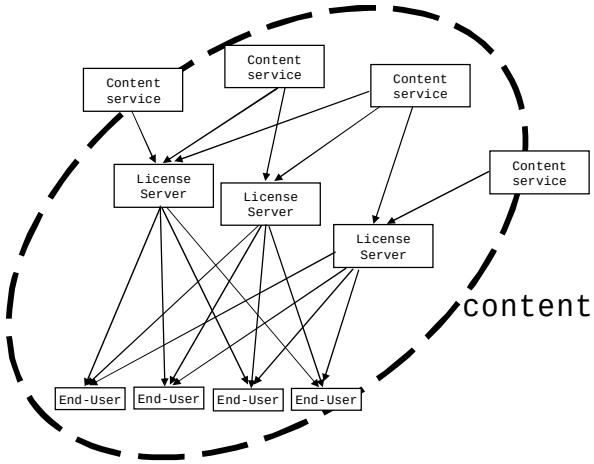

Figure 4 - License distribution schema

There can be as many templates as there are business models (or variations) on the system, and the final license is an instantiation of the business model for an end-user concerning a specific content. The following example represents an ODRL simplified license template adapted to a specific content and business model (for simplification, the presented license template does not have the Content Encryption Key (CEK) ciphered nor is digitally signed).

```
<?xml version="1.0" encoding="UTF-8" ?>
<o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
         xmlns:ds="http://odrl.net/1.1/ODRL-DD"
         xsi:schemaLocation="http://odrl.net/1.1/ODRL-EX
../schemas/ODRL-EX-11.xsd
         http://odrl.net/1.1/ODRL-DD ../schemas/ODRL-DD-11.xsd">
   <o-ex:agreement>
      <o-ex:asset>
         <ds:keyInfo>
            <ds:keyValue>%KEY%</ds:keyValue>
         </ds:keyInfo>
         <o-ex:context>
            <o-dd:uid>%CID%</o-dd:uid>
            <o-dd:name>%PARAM_1%</o-dd:name>
         </o-ex:context>
      </o-ex:asset>
      <o-ex:permission>
         <o-dd:play>
            <o-ex:constraint>
               <o-dd:individual>%UID%</o-dd:individual>
               <o-dd:count>%PARAM_2%</o-dd:count>
               <o-dd:datetime>
                <o-dd:start>%SDATE%</o-dd:start>
                <o-dd:end>%EDATE%</o-dd:end>
               </o-dd:datetime>
            </o-ex:constraint>
         </o-dd:play>
      </o-ex:permission>
   </o-ex:agreement>
</o-ex:rights>
```

Listing 1 - Example ODRL License template specific for a business model

On the license template all the parameters that can be replaced are represented using a specific notation (%KEY%, %CID%, %UID%, %SDATE, %EDATE, %PARAM%). The license production process works in the following way (Figure 5): (1) each of the content suppliers defines their own ODRL license templates, specifying business rules and conditions for each of the templates. When an end-user obtains protected content from some content supplier, a license is produced (3) using the specific license template defined previously, the content unique identifier and the user identifier (2). Afterwards, the license can be downloaded (4) by the end user

– not directly by the end-user but by the Wallet.

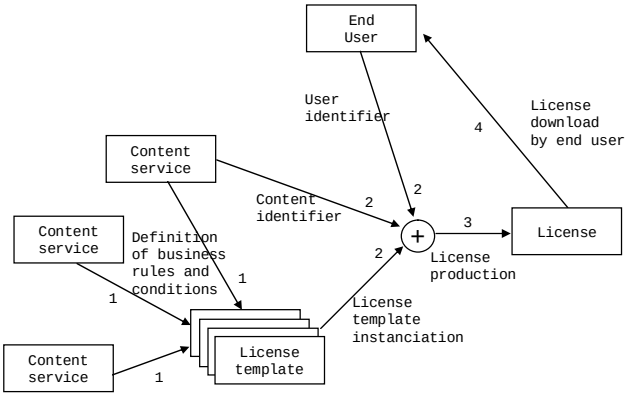

Figure 5 - Process to define a License

The described system is being used in several scenarios. These scenarios, described on the following sections, share the same License Server and the same digital Wallet at the client-side, but they have a different license template for each scenario: music download and streaming, video-surveillance and remote sensing JPEG2000 images.

For each of the scenarios, the most representative business model conditions have been identified and established on the License Server – in some cases more than one license template can be established for the same business scenario. In the case of the system presented here, the license templates are defined manually, but the production of web-based license template definition software is predicted that will allow content service providers (or content authors) to express their own rules on content usage in a very simple and natural way.

IV. USAGE SCENARIOS

All the scenarios that are presented in this paper use DRM and ODRL to control the access and conditions, of a given user or device to a particular content. Although these scenarios are quite different in nature, the used licenses share some commonalities (and at the same time some specific differences). In what concerns the commonalities among all the three proposed scenarios, they can be summarized in the following:

- Content identification (%CID%): each license contains the unique identifier which specifies that the license refers to a specific content [16][17], or content part;
- User identification (%UID%): all the licenses contain a way of specifying which user, group or domain is bound to the license;
- Expiry date (%SDATE%, %EDATE%): this parameter indicates the license validity period. This parameter supersedes the render content count (in case it exists), meaning that if the validity period expires before the counter reaches 0, the license is considered invalid;
- Content Encryption Key(s) (%KEY%): each of the licenses have one or multiple Content Encryption Key (CEK) that can be used by the appropriate end-user applications to access the protected content;
- License confidentiality and integrity: all the licenses (although not specified on the examples given on this paper) have the CEK ciphered in such a manner that can only be deciphered by the user's Wallet and all the licenses are digitally signed by the License Server to prevent their modification.

On the other hand, each of the proposed scenarios has specific conditions that are imposed on their license templates which were defined by the content providers. These conditions are specified and exemplified on the following sections.

A. *Music download and streaming*

This scenario, similar to others, represents a typical music portal, where an end-user can go and select some tracks of music to download/stream to listen [4]. OpenSDRM is used to control the access to the music and a specific license template was established for this particular scenario. This license template allows the specification of the following conditions:

- Play count: this parameter allows to setup how many times the content can be rendered by the end-user application;
- Operations: this parameter allows the definition of a set of possible operations that might be conducted over the content – in the case of the presented music business model the possible operations are: lend, save and play.

The following example (Listing 2) provides a sample license for the music download business model, with some of the generic and specific license parameters instantiated.

```
<?xml version="1.0" encoding="UTF-8" ?>
<o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
         xmlns:ds="http://odrl.net/1.1/ODRL-DD"
         xsi:schemaLocation="http://odrl.net/1.1/ODRL-EX
../schemas/ODRL-EX-11.xsd
         http://odrl.net/1.1/ODRL-DD ../schemas/ODRL-DD-11.xsd">
   <o-ex:agreement>
      <o-ex:asset>
         <ds:keyInfo>
            <ds:keyValue>%KEY%</ds:keyValue>
         </ds:keyInfo>
         <o-ex:context>
            <o-dd:uid>%CID%</o-dd:uid>
            <o-dd:name>Call On Me</o-dd:name>
         </o-ex:context>
      </o-ex:asset>
      <o-ex:permission>
         <o-dd:lend/>
         <o-dd:play>
            <o-ex:constraint>
               <o-dd:individual>%UID%</o-dd:individual>
               <o-dd:count>%PARAM_1%</o-dd:count>
               <o-dd:datetime>
                <o-dd:start>%SDATE%</o-dd:start>
                <o-dd:end>%EDATE%</o-dd:end>
               </o-dd:datetime>
            </o-ex:constraint>
         </o-dd:play>
      </o-ex:permission>
   </o-ex:agreement>
</o-ex:rights>
```

Listing 2 - Example of ODRL license for the music download scenario

This scenario was developed during an IST RTD project called MOSES [22], in a specific trial which targeted the

electronic commerce of digital music. This trial exploited a service called Music-4You (Figure 6), which allowed the users to obtain music and acquire the respective licenses. It used a dynamic price adjustment mechanism that established the final price according to the usage conditions selected by the final user.

Figure 6 – The Music-4You web-site

B. *Video-surveillance streaming and storage*

This scenario aims at the development of an integrated system for secure delivery of video surveillance data over a wireless network, while remaining scalable and robust to transmission errors. To achieve these goals, the content is encoded in Motion-JPEG2000 [21] and streamed with a specific RTP [27] protocol encapsulation to prevent the loss of packets containing the most essential data. Protection of the video data is performed at content level using the standardized JPSEC syntax [20], along with flexible encryption of quality layers or resolution levels. OpenSDRM is used to manage all authenticated peers on the WLAN (from end-users to cameras), as well as to manage the rights to access and display conditionally the video data. The OpenSDRM License Server produces licenses for this scenario based on the following parameters:

- Resolution level: the video-surveillance data maybe streamed with different quality resolution layer. The license defined in these scenarios allows the definition of different access levels concerning the resolution layer;
- Operations: this parameter allows the specification of the possible operations that can be conducted over the content by a given user or group of users: save, display or play.

The following example (Listing 3) provides a sample license for the video-surveillance streaming business model, with some of the generic and specific license parameters instantiated.

```
<?xml version="1.0" encoding="UTF-8" ?>
<o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
        xmlns:ds="http://odrl.net/1.1/ODRL-DD"
        xsi:schemaLocation="http://odrl.net/1.1/ODRL-EX
../schemas/ODRL-EX-11.xsd
        http://odrl.net/1.1/ODRL-DD ../schemas/ODRL-DD-11.xsd">
   <o-ex:agreement>
      <o-ex:asset>
         <ds:keyInfo>
            <ds:keyValue>%KEY_1%</ds:keyValue>
         </ds:keyInfo>
         <o-ex:context>
            <o-dd:uid>%CID_1%</o-dd:uid>
         </o-ex:context>
      </o-ex:asset>
      <o-ex:permission>
         <o-dd:save/>
         <o-dd:display>
            <o-ex:constraint>
               <o-dd:datetime>
                <o-dd:start>%SDATE%</o-dd:start>
                <o-dd:end>%EDATE%</o-dd:end>
               </o-dd:datetime>
            </o-ex:constraint>
         </o-dd:display>
         <o-dd:play>
            <o-ex:constraint>
               <o-dd:datetime>
                <o-dd:start>%SDATE%</o-dd:start>
                <o-dd:end>%EDATE%</o-dd:end>
               </o-dd:datetime>
            </o-ex:constraint>
         </o-dd:play>
      </o-ex:permission>
   </o-ex:agreement>
   <o-ex:agreement>
      <o-ex:asset>
         <ds:keyInfo>
            <ds:keyValue>%KEY_2%</ds:keyValue>
         </ds:keyInfo>
         <o-ex:context>
            <o-dd:uid>%CID_2%</o-dd:uid>
         </o-ex:context>
      </o-ex:asset>
      <o-ex:permission>
         <o-dd:save/>
         <o-dd:display>
            <o-ex:constraint>
               <o-dd:group>%UID%</o-dd:group>
               <o-dd:datetime>
                <o-dd:start>%SDATE%</o-dd:start>
                <o-dd:end>%EDATE%</o-dd:end>
               </o-dd:datetime>
            </o-ex:constraint>
         </o-dd:display>
         <o-dd:play>
            <o-ex:constraint>
               <o-dd:group>%UID%</o-dd:group>
               <o-dd:datetime>
                <o-dd:start>%SDATE%</o-dd:start>
                <o-dd:end>%EDATE%</o-dd:end>
               </o-dd:datetime>
            </o-ex:constraint>
         </o-dd:play>
      </o-ex:permission>
   </o-ex:agreement>
</o-ex:rights>
```

Listing 3 - Example of ODRL license for the video-surveillance streaming

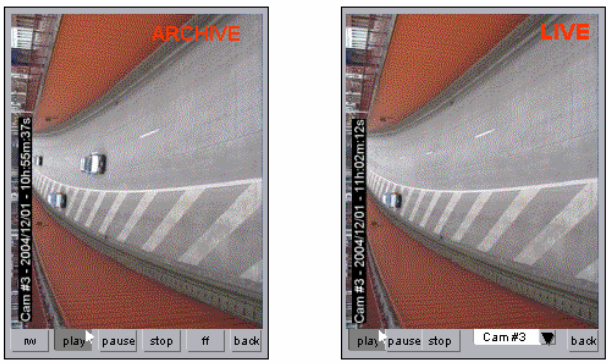

Figure 7 – WCAM prototype application

This scenario is currently under development by the FP6 IST RTD project called WCAM [23]. The system prototype has already been presented at the end of the first year of the project (Figure 7), and will be continuously improved towards its testing during the next Annecy 2005 International Animated Festival [24], where a live trial will be conducted to

the system.

C. *Remote sensing of JPEG2000 images*

The third and final scenario that will be presented in this paper refers to a content business situation in which an end-user can access an Earth Observation (EO) portal on the WWW and order some visible EO products which are then converted to JPEG2000 images [18][19]. These JPEG2000 EO products are protected by the EO portal supplier and sent in an encrypted format (using the JPSEC format) to the end-user. OpenSDRM is used to protect the access to the multiple resolutions of the EO product and to control which operations can be conducted over the content. OpenSDRM produces licenses for the EO products based on a template that allows the specification of the following parameters:

- Resolution level: the JPEG2000 EO products have different resolutions (to a maximum number of six). Each of the resolutions is protected with a different key and the access to each level can be conditioned to a particular user or user group;
- Operations: this parameter allows the specification of which are the operations that can be conducted on the content. In this particular business model the save operation is the one that is possible to specify. This operation allows the end-user to recover the original EO product format.

The following example (Listing 4) provides a sample license for the remote sensing images business model, with some of the generic and specific license parameters instantiated.

```
<?xml version="1.0" encoding="UTF-8" ?>
<o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
         xmlns:ds="http://odrl.net/1.1/ODRL-DD"
         xsi:schemaLocation="http://odrl.net/1.1/ODRL-EX
../schemas/ODRL-EX-11.xsd
         http://odrl.net/1.1/ODRL-DD ../schemas/ODRL-DD-11.xsd">
   <o-ex:agreement>
      <o-ex:asset>
         <ds:keyInfo>
            <ds:keyValue>%KEY_1%</ds:keyValue>
         </ds:keyInfo>
         <o-ex:context>
            <o-dd:uid>%CID_1%</o-dd:uid>
         </o-ex:context>
      </o-ex:asset>
      .
      .
      .
      <o-ex:asset>
         <ds:keyInfo>
            <ds:keyValue>%KEY_6%</ds:keyValue>
         </ds:keyInfo>
         <o-ex:context>
            <o-dd:uid>%CID_6%</o-dd:uid>
         </o-ex:context>
      </o-ex:asset>
      <o-ex:permission>
         <o-dd:display>
            <o-ex:constraint>
               <o-dd:individual>%UID%</o-dd:individual>
               <o-dd:datetime>
                <o-dd:start>%SDATE%</o-dd:start>
                <o-dd:end>%EDATE%</o-dd:end>
               </o-dd:datetime>
            </o-ex:constraint>
         </o-dd:display>
         <o-dd:display>
            <o-ex:constraint>
               <o-dd:individual>%UID%</o-dd:individual>
               <o-dd:datetime>
                <o-dd:start>%SDATE%</o-dd:start>
                <o-dd:end>%EDATE</o-dd:end>
               </o-dd:datetime>
            </o-ex:constraint>
         </o-dd:display>
      </o-ex:permission>
   </o-ex:agreement>
</o-ex:rights>
```

Listing 4 - Example of ODRL license for the remote sensing scenario

This scenario was developed during a European Space Agency (ESA) project, called HICOD2000 [25]. HICOD2000 implemented this scenario that allowed the service provider to protect the EO products and to define at the same time licenses which controlled the end-user access to such products. This system was implemented and was integrated within ESA EO products portal.

The users can browse EO products from the ESA portal, select the products and the corresponding resolution level, and perform its payment. The ESA portal connects the EO product service provider that produces the JPEG2000 version of the EO product and protects it.

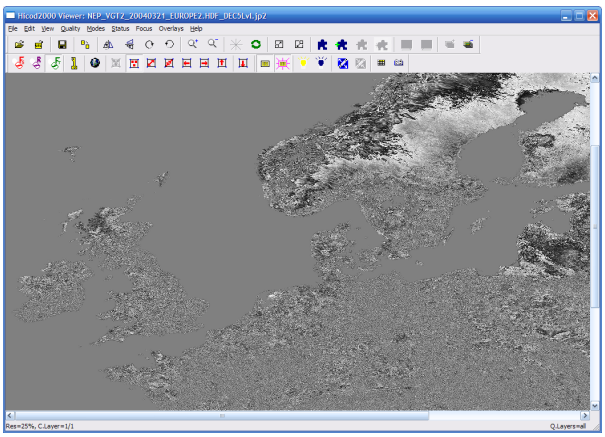

Figure 8 – The HICOD2000 Viewer

The EO service provider uses the OpenSDRM platform to specify the licenses of each of the EO products. When the user receives the EO product he can open it on a specific viewer (Figure 8) that enforces the license over the content (which is protected).

V. CONCLUSIONS

In this paper we have described a system that uses ODRL to express rights over protected content [14]. This system, referred to as OpenSDRM [3][4], uses a mechanism that enables interoperability at the client-side of the different protected content types and different content applications – the OpenSDRM digital Wallet. This mechanism enables DRM-supported applications to request, to the digital Wallet middleware, authorization to perform operations over the protected content. The required clearance of these operations, mediated by the Wallet, is expressed in ODRL-formatted licenses [14]. However, the system is REL-independent.

The system is based on the notion of license templates, which are defined taking into account the content business rules expressed by the content supplier. The presented system enables a multiplicity of different license conditions for different content suppliers, since, whenever a license is issued

to a given user, the license server instantiates a license template with the appropriate parameters.

The paper has also provided three different usage scenarios in which the system is being used, demonstrating its applicability and usefulness in mediating the access to digital music, remote sensing images and video-surveillance streams. These three different scenarios share the same License server with three different ODRL license templates. The License server, according to the content service, issues a specific license (instantiating the template) that can subsequently be downloaded after by the license system middleware, present at the end user client. This client-side middleware, the OpenSDRM digital Wallet, receives requests from the applications to be granted access to operations with the content.

REFERENCES

[1] Chiariglione, L., "Intellectual Property in the Multimedia Framework", Management of Digital Rights, Berlin (2000)

[2] Duhl J., Keroskian S., "Understanding DRM Systems", IDC White Paper, 2003

[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., "OpenSDRM - An Open and Secure Digital Rights Management Solution", IADIS 2003, Lisboa,

[5] Prunela A., "Windows Media Technologies: Using Windows Media Rights Manager to Protect and Distribute Digital Media", MSDN Magazine, December (2001), http://msdn.microsoft.com/msdnmag/issues/01/12/DRM/default.aspx

[6] Bormans J., Hill K., "MPEG-21 Overview v.5", ISO/IEC JTC1/SC29/WG11/N5231, 2002

[7] ISO/IEC 21000-3 Information technology -- Multimedia framework (MPEG-21) -- Part 3: Digital Item Identification

[8] Microsoft, "Architecture of Windows Media Rights Manager", Microsoft Corporation, http://www.microsoft.com/windows/windowsmedia/howto/articles/drmarchitecture.aspx, May 2004

[9] Lenzi, R. et al, "Apple iTunes Music Store", technical report, The Interactive-Music Network, June 2003

[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] Microsoft, "Scenarios for Windows Media DRM", Microsoft Corporation, 2004, http://www.microsoft.com/windows/windowsmedia/drm/scenarios.aspx

[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) Version 1.1", W3C Note, September 2002, http://www.w3.org/TR/odrl/

[15] Iannella R., "Digital Rights Management (DRM) Architectures", D-Lib Magazine, Volume 7 Number 6 ISSN 1082-9873, June 2001 http://www.dlib.org/dlib/june01/iannella/06iannella.html

[16] Dalziel, J., "DOI in a DRM environment", White Paper, Copyright Agency Limited, 2004

[17] Rosenblatt, B. "Enterprise Content Integration with the Digital Object Identifier: A Business Case for Information Publishers", June 2002

[18] ISO/IEC 15444-1/ IUT-T T.800, JPEG2000 Image Coding System - Part 1: Core Coding System, 2000.

[19] D. Taubman and M. Marcellin, JPEG 2000: Image Compression Fundamentals, Standards and Practice, Kluwer Academic Publishers, 2002.

[20] JPSEC Final Committee Draft 1.0, ISO/IEC JTC1/SC29 WG1 N3480, November 2004

[21] Sadourny, Y., Conan, V., Serrão, C. Fonseca, P., "WCAM: secured video surveillance with Digital Rights Management", WCAM project, SPIE, 2004

[22] MOSES web-site, http://www.ist-moses.org

[23] WCAM web-site, http://www.ist-wcam.org

[24] Annecy 2005 International Animated Festival web-site, http://www.annecy.org

[25] HICOD2000 web-site, http://www.hicod2000.org

[26] Music-4You web-site, http://www.music-4you.com

[27] Schulzrinne H., Casner S., Frederick R., Jacobson V., „RTP: A Transport Protocol for Real-Time Applications", RFC1889, January 1996