Xavier Maroñas, Eva Rodríguez, Jaime Delgado

Distributed Multimedia Applications Group (DMAG)
Departament d'Arquitectura de Computadors
Universitat Politècnica de Catalunya
C/ Jordi Girona, 1-3, E-08034 Barcelona, Spain

{xmaronas, evar, jaime.delgado}@ac.upc.edu

AN ARCHITECTURE FOR THE INTEROPERABILITY BETWEEN RIGHTS EXPRESSION LANGUAGES BASED ON XACML

Abstract: None of the current initiatives to provide interoperability between Digital Rights Management (DRM) systems, such as CORAL [3] or Marlin [13], provide a global solution to enable different devices to render content governed by different DRM systems. Interoperability applies to different aspects of DRM, such as languages for rights expression, digital objects declaration and protection information declaration. We focus on the interoperability between Rights Expressions Languages (RELs). In this area, we have already done some work, including the definition of profiles for the MPEG-21 REL [14] standard. Based on those profiles for specific domains (such as the mobile or broadcasting ones), a couple of approaches have been developed for interoperability between licenses in different RELs, such as the use of extensible style sheets transformations or the use of high level modelling schemas. In order not to be restricted to specific profiles, this paper presents a solution based on the use of a XACML system based architecture to achieve interoperability between RELs. We demonstrate the feasibility of the proposed solution showing that the main elements of rights expressions can be expressed in XACML policies without loss of information. Furthermore, instead of translating between different RELs through an intermediate one, we directly map to XACML and perform the authorisation with it. In this way, we do not lose any information conveyed in the input REL.

Keywords: DRM, REL, interoperability, XACML.

1. Introduction

The aim of this paper is to present an architecture for the interoperability between different Rights Expression Languages (RELs), based on XACML [5]. This implies the development of a DRM system based on XACML, so we need to formalise mappings between different RELs and XACML, but not the other way around, since all the operations, such as the authorisation process, are done inside the XACML-based DRM system. The paper presents a particular case: the translation between the ODRL Rights Expression Language (REL) and the XACML policy language.

The first sections of this paper, section 2 and 3, briefly presents rights expression language focussing on ODRL and MPEG-21 REL, since they were the RELs used in the proposed architecture, as well as eXtensible Access Control Markup Language (XACML).

Section 4 describes the previous attempts that our research group has made in the field of the interoperation between different DRM systems and, more specifically, between the RELs used by these systems.

The architecture, presented in section 5, shows a main system based on the XAMCL standard, with the addition of translators between the other DRM REL standard languages and the XACML. Then, the example of a specific translator between ODRL and XACML and MPEG-21 and XACML, in section 6, describes the equivalences between both languages.

Finally, section 0 presents the conclusions of this novel approach as well as the future work that has to be done.

2. Rights Expression Languages

Rights expressions are defined to be the terms that govern the usage of digital assets through the complete digital value chain. The rights expressions are normally presented to the different actors of the digital value chain as an Extensible Markup Language (XML) [4] file, which contains the permissions and constraints that govern the digital asset. This XML file, called license or rights object by most of the DRM systems, is expressed according to a Rights Expression Language (REL). Licenses are generated according to a specific REL with the content usage rules associated to a specific entity and digital asset. They are usually associated to a digital object and could be interpreted and enforced by authorisation tools of DRM systems. In some DRM systems, licenses also contain information related to the protection of digital objects, for example the key needed to decipher the digital object or asset. Next, we introduce the ODRL REL [17]. However, we also consider in our work other RELs, such as MPEG-21 [14].

2.1. ODRL

The Open Digital Rights Language (ODRL) [17] is a proposed language for the DRM community for the standardisation of expressing rights information over content. The ODRL is intended to provide flexible and interoperable mechanisms to support transparent use and consumption of governed digital assets. It is an XML-based usage grammar, as the MPEG-21 REL [14].

ODRL is based on an extensible model for rights expressions, which involves three core entities and their relationships. The party element identifies entities (e.g., humans, roles, etc.). It includes end users and rights holders. The right element defines the permissions, which can then contain constraints, requirements, and conditions. The asset element includes any physical or digital content. Assets must be uniquely identified and may consist of many subparts and be in many different formats. In ODRL version 2, these three elements (asset, party and action), are all included under the permission element.

2.2. MPEG-21 REL

Part 5 of the MPEG-21 standard [14] specifies the syntax and semantics of a rights expression language. The most important concept of the MPEG-21 REL are licenses that can be conceptualised as containers of grant elements each one of which conveys to a particular principal the sanction to exercise some identified right against some identified resource, possibly subject to the need for some condition to be first fulfilled. An MPEG-21 REL license is made up essentially of a grant element, whose semantics will be used by the authorisation tools to determine if users have the appropriate rights to exercise the actions that they request. The second important element of a license is the issuer element that contains the identification of the issuer and a set of specific details about the circumstances under which the license has been issued.

3. XACML

The Extensible Access Control Markup Language (XACML) [5] standard was specified by OASIS [16], which is a not-for-profit organisation that develops open standards for security, Web services, XML conformance, etc., and for marketplaces interoperability. The XACML standard was devised for expressing authorisation policies in XML, which will apply to objects that can be identified in XML. The main motivation for the definition of the XACML was the amount of proprietary and application specific access control policy languages used to define policies, which once defined cannot be shared across different systems. The XACML specification enables the use of different types of policies without requiring changes to the applications that use XACML. It uses the W3C XML-Signature Syntax and Processing Standard [27] for providing authentication and integrity protection for XACML policies. The XACML version 2.0 specification provides the model descriptions for dataflow, XACML context (canonical representation of a decision request and an authorisation decision), and policy language (rule, policy, policy set).

The XACML standard specifies a policy language model. The three top-level policy elements defined for this model are: rule, policy and policySet. The rule element is the basic unit of management within an XACML policy administration point. The main components of the rule element are: the target, effect and condition elements. The target element defines the set of resources, subjects, actions and environments to which the rule is intended to apply. The effect element indicates the consequence of a true evaluation for the rule. The condition element refines the applicability of the rule. The policy element consists of rule elements and mechanisms for combining the results of their evaluation. The obligations element specifies the actions that shall be performed in conjunction with the policy evaluation. Finally, the policySet element enables the combination of separate policies into a single policy.

4. Previous attempts on the Rights Expression Languages Interoperability

Authors of this paper, jointly with other members of the DMAG research group [10], have been working since several years on the DRM systems interoperability issue. Interoperability may apply to different aspects of DRM, such as rights expression languages, digital objects formats and protection information declaration languages.

Initially, we concentrated in the interoperability between rights expression languages. We focus on two standard initiatives, MPEG-21 REL and ODRL, which are competing to have a place in the market. In [22], a first approach to achieve interoperability between these two rights expression languages was proposed. In this first study, the authors concluded that a syntactic approach to map licenses expressed in two different languages would only be feasible for a subset of both languages, which could be identified as profiles. Meanwhile, the Open Mobile Alliance (OMA) [18] defined a DRM system to enable the consumption of digital content in a controlled manner for the mobile domain. Further work of the authors was the definition of a subset of the MPEG-21 REL to just provide the same features as the OMA DRM REL [2] [18]. This work was presented to MPEG and it was chosen as one of the starting points for the definition of the MPEG-21 REL Mobile and Optical Media profile [15]. The same approach was taken to provide interoperability between a subset of the MPEG-21 REL and other rights information standards defined for governing digital broadcast content, such as TV-Anytime RMPI [11], B-cast Rights Object of OMA DRM v2.0 [19], DVB-CPCM USI [9] and CCI [20]. Then, the MPEG-21 REL Dissemination And Capture (DAC) profile was defined to provide interoperable mechanisms to represent the usage rules of digital broadcast programs.

Besides the presented initial work, where Extensible Style Sheets Transformations (XSLT) [6] were used to translate from licenses expressed in one rights expression language to another, other different approaches were considered. One of them, presented in [12], was based on the use of high level modelling schemas, like Unified Modeling Language (UML) [25] and Entity-Relationship [2]. However, this solution has the same limitations as the previous one, since interoperability is only achieved for specific subsets of rights expression languages.

The main limitation of these first attempts to provide interoperability between different RELs is that only can be taken into account the common rights and conditions defined in both RELs. Even though, it's not worth adding the new terms, since the license based authorisers compliant with a REL do no understand them, and thus when they find any of them in a license they authorise negatively as they work in the most restrictive way. On the other hand, if we consider as superset of all the RELs, XACML due to its extensibility, we can add new rights and conditions and implement their authorisation. In this way, we are not limiting our solution to a subset of the RELS, but we allow adding new rights, whose semantics we will take into account, since we will consider its namespace when authorising, and regarding the new conditions its validation will be implemented in the XACML authoriser as they appear. The rest of this paper proposes the use of an XACML profile to provide interoperability between RELs. This approach overcomes the limitations of the previously discussed solutions thanks to the flexible syntax and semantics of the XACML policy elements (specifically of the subject, resource, action and conditions elements). A mapping to XACML of every REL to consider must be specified. Due to XACML characteristics, no information in any REL will be lost. Hence, the best approach is to implement a XACML authorisation system that directly executes the authorisation algorithm from the mapping initiated by the license in any REL. Therefore, there is no need to translate between RELs.

5. Proposed architecture for DRM systems interoperability

Figure 1 shows the solution proposed taking the XACML as the main system for the interaction with any other rights expression language. The main idea is to have a system that will accept any request from any DRM REL format, will process the request converting that document in the corresponding

XACML policy and will pass it to the system for this to complete the authorisation process.

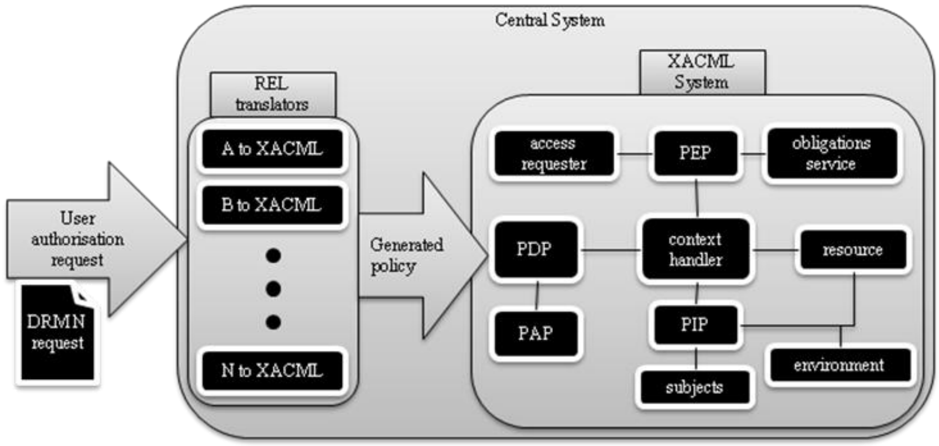

Figure 1. Proposed XACML based architecture

On one hand, when the system receives a request expressed in its own DRM format, it processes it as usual using the modules specifically designed for it. On the other hand, when it receives a request in another DRM format, it is redirected to the translator module. First of all, the request is processed in order to determine the DRM format to which it is compliant. Once recognised it, the request is redirected to the corresponding translator in order to obtain a valid XACML policy and the corresponding authorisation request. Finally, this policy and the context information are used to perform the authorisation and generate the appropriate response.

In this solution, the XACML system is not used to translate from one REL to another. Doing that, it should lose some information depending on the final REL capabilities. The solution proposed uses a complete XACML system, adding support for translating from any REL to XACML policy system which can be seen as a superset that covers all the possibilities of the other languages. Using XACML as the main language assures that this translation will not lose any data from the original format.

6. Use of XACML to provide interoperability between RELs

Once presented the desired architecture of the whole system, the rest of the paper will focus on the translation module. This section presents the research work conducted on the feasibility to express ODRL rights terms and conditions using XACML policy elements without losing data. The benefits that can be obtained include the definition of mechanisms for expressing license elements in XACML documents and the definition of verification algorithms for authorisation purposes.

6.1. ODRL and the XACML Policy Language

ODRL rights are expressed in XML. The first step to translate from these ones to XACML policies is to recognise the equivalent elements in both schemas. The first version of the ODRL standard presents a structure quite different from the XACML one, but with the second version they are more similar. Figure 2 presents a visual representation of the correspondence between the elements of both languages.

ODRL has the permission element that involves all the basic information as the rule element does in the XACML policy. The party entity contains the information about the user that holds the right, as the subject element does. For representing the resource, ODRL defines the asset element. The action entity, which defines the right that may be exercised by the user, is represented by the same element but its structure is very different, as it is shown in the next subsections. Finally, ODRL represent the condition with the constraint element that is similar to the corresponding MPEG-21 REL one, more than with the XACML element.

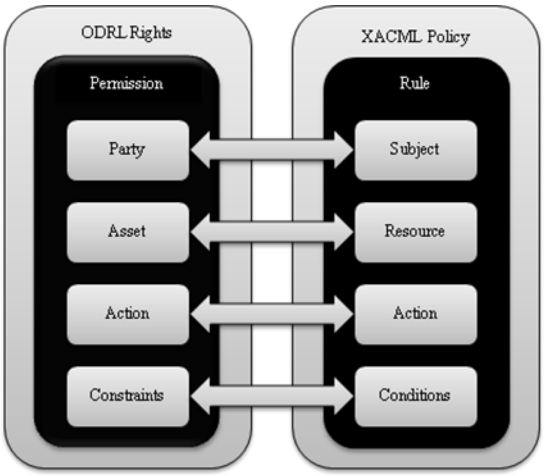

Figure 2. ODRL Rights vs XACML Policy

The next subsection identifies specific possible solution for the translation between all the basic elements of an ODRL rights to the XACML policy ones.

6.1.1. Rights

The first difference between both types of documents is the root element, which describes the digital contract. For ODRL documents, the root element is the *rights* element; while in XACML documents, the root element is the *policy* element. See Table 1.

Table 1. Syntax for the rights and policy elements

ODRL	XACML
<o-ex:rights […]>	<xacml:Policy>
[…]	[…]
</o-ex:rights>	</xacml:Policy>

6.1.2. Permission

Permission is the element used in ODRL for grouping the basic information, like the user that will hold the rights, the object involved in the contract or the conditions that must be accomplished. This is presented in Table 2.

Table 2. Syntax for the rights and policy elements

ODRL	XACML
<o-ex:permission>	<xacml:Rule>
[…]	[…]
</o-ex:permission>	</xacml:Rule>

6.1.3. Party

In ODRL there is an element for describing the subject who will hold the rights. This subject can be a physical person, a company, a specific domain, etc. This element is the *party*, and it is recommended to use a standard representation for describing it. An example of a *party* element content is another ODRL element, context, that includes an *uid* for identifying the user. This example is presented in Table 3.

Table 3. Syntax for the rights and policy elements

ODRL	XACML
<o-ex:party>	<xacml:Subjects>

<o-ex:context>	<xacml:Subject>
<o-dd:uid>subjectId</o-dd:uid>	<xacml:SubjectMatch MatchId =
</o-ex:context >	"urn:oasis:names:tc:xacml:1.0:function:string-equal">
</o-ex:party>	<xacml:AttributeValue DataType="[…]string">
	Subjected </xacml:AttributeValue>
	<xacml:SubjectAttributeDesignator AttributeId="…" DataType="[…]string"/>
	</xacml:SubjectMatch>
	</xacml:Subject>
	</xacml:Subjects>

6.1.4. Action

ODRL, similar to MPEG-21 REL, defines a list of specific actions that will apply to the rights owner. Each action has a specific meaning described in the standard, and has to be mapped taking it in account. An example of one of these ODRL actions can be found in Table 4.

Table 4. Syntax for the rights and policy elements

ODRL	XACML
<o-dd:play/>	<xacml:Policy>
	<xacml:Actions>
	<xacml:Action>
	<xacml:ActionMatch MatchId =
	"urn:oasis:names:tc:xacml:1.0:function:string-equal">
	<xacml:AttributeValue DataType="[…]string">play</xacml:AttributeValue>
	<xacml:ActionAttributeDesignator DataType="[…]#string" AttributeId="urn:oasis:names:tc:xacml:1.0:resource:xpath"/>
	</xacml:ActionMatch>
	</xacml:Action>
	</xacml:Actions>
	</xacml:Policy>

6.1.5. Asset

The resource in ODRL is represented by an *asset* element that contains a context element in it. This context typically contains at least an *uid* element with the unique identifier of the resource, and may contain other elements like the type or any other descriptor. Table 5 shows an example of how the ODRL asset element can be translated into the corresponding XACML elements.

Table 5. Syntax for the rights and policy elements

ODRL	XACML
<o-ex:asset>	<xacml:Resources>
<o-ex:context>	<xacml:Resource>
<o-dd:uid>	<xacml:ResourceMatch MatchId =
resourceId	"urn:oasis:names:tc:xacml:1.0:function:string-equal">
</o-dd:uid>	<xacml:AttributeValue DataType="[…]integer">resourceId</xacml:AttributeValue>
</o-ex:context>	
</o-ex:asset>	<xacml:ResourceAttributeDesignator DataType="[…]#string" AttributeId="urn:oasis:names:tc:xacml:1.0:resource:xpath"/>
	</xacml:ResourceMatch>
	</xacml:Resource>
	</xacml:Resources>

6.1.6. Constraints

In ODRL the restrictions over the permissions are in the *constraint* elements. These elements, as happens in the MPEG-21 REL, are delimited by the standard. Some of the available constraints in ODRL are: *count* (the number of times the corresponding permission may be exercised), *spatial* (for delimiting the geographic area), *datetime* (the interval within the action may be done), watermark (the watermarking value of the asset), *cpu* (used as unique identifier of the user computer), etc. Table 6 shows an example of the translation of the spatial constraint of ODRL to the corresponding elements in the XACML policy language.

Table 6. Syntax for the rights and policy elements

ODRL	XACML
<o-ex:constraint>	<xacml:Condition>

<o-dd:spatial o-ex:type = ”prism:vocabs/ISO3166/ES”> </o-ex:constraint>	<xacml:Apply FunctionId = "urn:oasis:names:tc:xacml:1.0:function:string-equal"> <xacml:AttributeSelector DataType="[…]#string" RequestContextPath = "//xacml-context:Resource/xacml-context:ResourceContent/location/country"/> <xacml:AttributeValue DataType="[…]#string"> ES</xacml:AttributeValue> </xacml:Apply> </xacml:Condition>

6.2. MPEG-21 REL and the XACML Policy Language

The same work has been done for the MPEG-21 REL [23]. First, as for ODRL, the correspondence between the main elements of MPEG-21 REL licenses and XACML policies has been defined (see Figure 3). Then, syntax and semantics for the equivalent XACML policy elements has been defined. In this first activity, this has been done for each one of the main elements of an MPEG-21 REL license grant (i. e. principal, right, resource and conditions). Table 7 contains relevant examples that illustrate the equivalences between MPEG-21 REL license elements and its corresponding syntax in XAMCL policies.

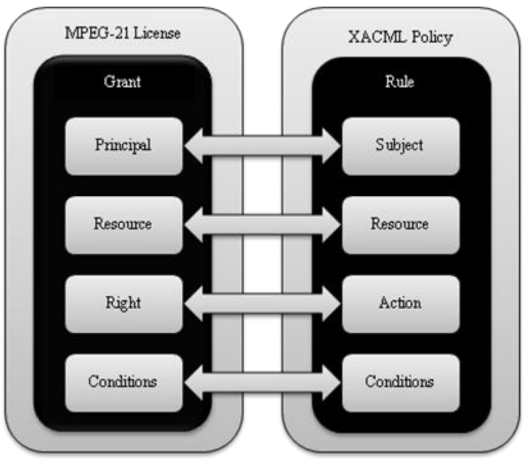

Figure 3. MPEG-21 REL licenses vs XACML Policies

Table 7. MPEG-21 REL and XACML policy

MPEG-21 REL	XACML
<r:license licenseId="1">[…]	<xacml:Policy PolicyId="urn:oasis:names:tc:xacml:2.0:ex:policyid:1">
</r:license>	[…]
	</xacml:Policy>
<r:grant> […] </r:grant>	<xacml:Rule> […] </xacml:Rule>
<r:keyHolder>	<xacml:Subjects>
<r:info>	<xacml:Subject>
<dsig:KeyName>subjectId </dsig:KeyName>	<xacml:SubjectMatch MatchId="[…]string-equal">
</r:info>	<xacml:AttributeValue DataType="[...]string">
</r:keyHolder>	subjectId</xacml:AttributeValue>
	<xacml:SubjectAttributeDesignator AttributeId="[…]subject-id" DataType="[…]string"/>
	</xacml:SubjectMatch>
	</xacml:Subject>
	</xacml:Subjects>
<mx:play/>	<xacml:Actions>
	<xacml:Action>
	<xacml:ActionMatch MatchId="[…]string-equal">
	<xacml:AttributeValue DataType="[…]string">play</xacml:AttributeValue>
	<xacml:ActionAttributeDesignator DataType="[…]string" AttributeId="[…]xpath"/>
	</xacml:ActionMatch>
	</xacml:Action>
	</xacml:Actions>
<mx:diReference>	<xacml:Resources>
<mx:identifier>resourceId	<xacml:Resource>
</mx:identifier>	<xacml:ResourceMatch MatchId="[…]string-equal">
</mx:diReference>	<xacml:AttributeValue DataType="[…]integer">resourceId </xacml:AttributeValue>
	<xacml:ResourceAttributeDesignator DataType="[…]string" AttributeId="[…]xpath"/>

	</xacml:ResourceMatch>
	</xacml:Resource>
	</xacml:Resources>
<r:allConditions>	<xacml:Condition>
<sx:territory>	<xacml:Apply FunctionId="[…]and">
<sx:location>	<xacml:Apply FunctionId="[…]string-equal">
<sx:country xmlns:iso="[…]country">	<xacml:Apply FunctionId="[…]string-equal">
Country	<xacml:AttributeSelector DataType="[…]string" RequestContextPath="[…]country"/>
</sx:country>	</xacml:Apply>
</sx:location>	<xacml:AttributeValue DataType="[…]string">Country</xacml:AttributeValue>
</sx:territory>	</xacml:Apply>
</r:allConditions>	</xacml:Condition>

7. Application scenarios

Nowadays, content is governed in different ways depending on the DRM solution chosen. Each DRM system uses a specific rights expression language for governing digital assets, a certain language for the expression of digital objects, etc. Therefore, it is important to define solutions that enable users to exchange digital objects between different DRM systems. The solution proposed in this paper can be used by any DRM system that tries to interoperate with other DRM system. Authors have developed a prototype in the context of the VISNET II project [26], which enables different rights management systems interoperate in Virtual Collaboration and Video Surveillance applications.

In the video surveillance scenario under consideration, a judge wants to review the video of a robbery in the cash point of a bank branch. The images have been recorded by the video surveillance system of the bank, and a protected version of the video is stored in a central database. The DRM system chosen by the bank to manage video surveillance content is the AXMEDIS [1] system. The judge, who is reviewing the case in his office, wants to view the images in her PDA. She is a registered user of the OMA system [18]. Therefore, in this scenario, the judge tries to access to AXMEDIS content from the OMA system. This is possible through the VISNET-II broker. Figure 4 sketches the architecture for the proposed scenario.

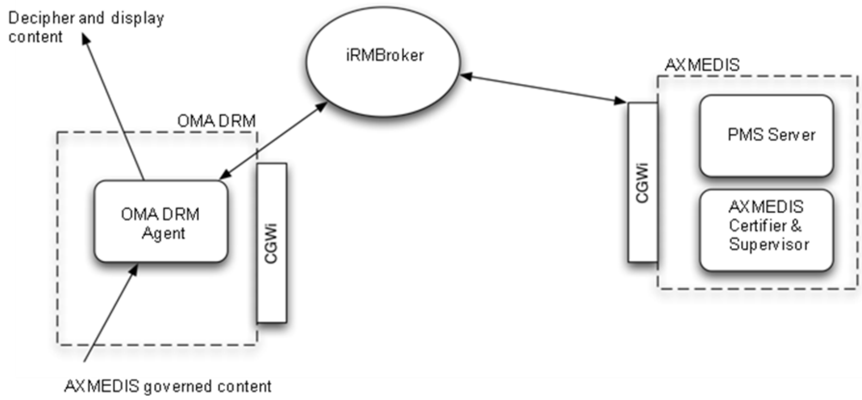

Figure 4. DRM Systems interoperability – Video Surveillance scenario

The OMA Client player has obtained an AXMEDIS formatted and governed content. This content formatted as an MPEG-21 Digital item, is basically composed of relevant metadata and the ciphered resource. To obtain the necessary protection information, and to perform the authorization process and obtain the content to render, the OMA client player should implement a full MIPAMS client, but this is a heavy process and the client will grow with each interoperation. This is a poor way to achieve the interoperability between the different solutions and therefore not the solution that has been followed on this work. To avoid the OMA client to perform this process, the iRMBroker will enable the OMA to obtain the content in a format that is completely transparent for OMA, and will abstract OMA from the required operations to perform.

The process can be described in the following steps (see Figure 5):

1. Once authenticated in the system, Alice tries to load an AXMEDIS formatted and governed object (containing the video of the robbery);
2. The OMA Client, invokes the iRMBroker, passing the content URI and a set of credentials;
3. The iRMBroker validates the OMA platform, identifies the MIPAMS content and the AXMEDIS platform location, and validates the credentials presented;
4. The iRMBroker, invokes the appropriate AXMEDIS instance, through the publicly available CGWi operations, in order to obtain the user's license, which grants her permissions to exercise the requested operation;
5. The iRMBroker obtains the license, translates it to XACML and performs the authorisation;

6. Since the user has the appropriate permissions (Alice is the judge of the case), the iRMBroker requests the protection information (about the content) to the AXMEDIS system, through the publicly available CGWi operations;
7. The iRMBroker extracts the raw content, deciphering it using the protection information obtained;
8. The obtained raw unprotected content will be ciphered by AXMEDIS with some credentials, and with an algorithm supported by the OMA client – this credentials will be made available to OMA in a protected format.
9. The OMA Client obtains the content ciphered and the credentials to obtain it.
10. The OMA client deciphers the content and renders it.

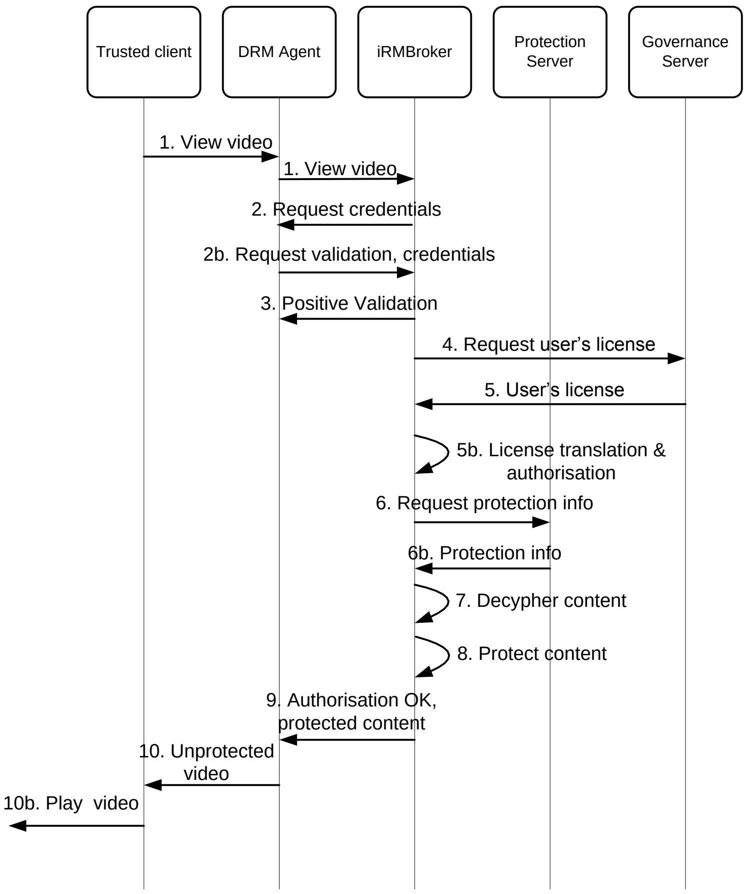

Figure 5. Video surveillance use case – positive authorisation

8. Conclusions and Future Work

This paper presents a novel solution to achieve interoperability between different rights expression languages. This solution is based on the use of a profile of XACML to provide interoperability between RELs. After analysing XACML, we state that the terms in end user licenses expressed according to a REL, such as the ODRL or MPEG-21 REL, can be also represented in a XACML policy. Although this paper describes in detail the work done for the ODRL, it also presents a summary of the work done for the MPEG-21 REL.

The benefits of this solution regarding to the previous ones proposed in [22][24][8] and [12], mainly lie in the flexibility of the XACML policy language, that provides interoperability for the complete syntax and semantics of the Rights Expression Languages (i.e., principals, rights, resources and conditions), not only for specific subsets or profiles. This is achieved due to the way in which the syntax and semantics of the XACML policy elements (specifically of the subject, resource, action and conditions elements) have been defined.

Finally, this paper proposes an architecture for DRM systems interoperability in terms of RELs. It proposes the XACML system to be the central one, which will solve license based authorisation requests for users trying to use (e.g., render, adapt, etc.) a given content governed by a different DRM system. Although we have already implemented several authorisation modules[7][23], we are now in the process of implementing a XACML-based authorisation server that will be integrated in a complete system able to enforce licenses expressed in different RELs.

References

[1].AXMEDIS, *http://www.axmedis.org/*

[2].Chen P., *The Entity-Relationship Model--Toward a Unified View of Data*, ACM Transactions on Database System, Vol. 1, No. 1, March 1976, pp 9-36.

[3].CORAL CONSORTIUM CORE ARCHITECTURE OVERVIEW. CCAWC EDITORS. JUNE 2006. CCAWC CORE ARCHITECTURE. CCAWC EDITORS. JUNE 2006. ECOSYSTEM-A SPECIFICATION. CCAWC EDITORS. JUNE 2006, *http://www.coral-interop.org/*

[4] EXTENSIBLE MARKUP LANGUAGE (XML), *http://www.w3.org/XML/*

[5].EXTENSIBLE ACCESS CONTROL MARKUP LANGUAGE (XACML) V2.0, *http://www.oasis-open.org/specs/index.php#xacmlv2.0*

[6] EXTENSIBLE STYLESHEET LANGUAGE WEB SITE, *http://www.w3.org/Style/XSL/*

[7] Delgado, J., Gallego, I., Rodríguez, E., *Use of a Rights Expression Language for Secure Music Distribution*. Journal of New Music Research. Volume 34, Number 2. June 2005.

[8] Delgado J., Prados J. and Rodríguez E., *Profiles for interoperability between MPEG-21 REL and OMA DRM*, 7th International IEEE Conference on Ecommerce Technology 2005 (IEEE CEC 2005). Munich (Germany). July 2005.

[9] DIGITAL VIDEO BROADCASTING (DVB) CONTENT PROTECTION AND COPY MANAGEMENT (CPCM) SYSTEM, USAGE STATE INFORMATION (USI), SB1497, NOVEMBER 2005.

[10] DISTRIBUTED MULTIMEDIA APPLICATIONS GROUP (DMAG), *http://dmag.ac.upc.edu/*

[11] Earnshaw, Consideration of the TV-Anytime RMPI Specification for Inclusion in the Stationary Audio Video Device and Supporting Rights Managed Environment, DMP/dmp0394, April 2005.

[12] Llorente S., Delgado J., Maroñas X., *Implementing Mobile DRM with MPEG-21 and OMA*, 5th International Workshop on Security in Information Systems (WOSIS 2007). Funchal, Madeira (Portugal). June 2007.

[13] MARLIN ARCHITECTURE OVERVIEW. MARLIN DEVELOPER COMMUNITY. *http://www.marlin-community.com/*

[14] ISO/IEC. ISO/IEC IS 21000-5 - MPEG-21 – RIGHTS EXPRESSION LANGUAGE.

[15] ISO/IEC. ISO/IEC IS 21000-5 AMD1 - MPEG-21 RIGHTS EXPRESSION LANGUAGE AMENDMENT 1 MAM (MOBILE AND OPTICAL MEDIA) PROFILE.

[16] OASIS, *http://www.oasis-open.org/home/index.php*

[17] OPEN DIGITAL RIGHTS LANGUAGE (ODRL), *http://odrl.net*

[18] OPEN MOBILE ALLIANCE (OMA), *http://www.openmobilealliance.org/*

[19] OPENSDRM, *http://sourceforge.net/projects/opensdrm/*

[20] OMA DRM V2.0 EXTENSIONS FOR BROADCAST SUPPORT, OMA-TS-DRM-XBS-V1_0-20051209-D, DECEMBER 2005

[21] OC-SP-CCCP-IF-C01-050331, OPENCABLE™ INTERFACE SPECIFICATIONS CABLECARD™ COPY PROTECTION SYSTEM INTERFACE SPECIFICATION; PART 6. COPY CONTROL INFORMATION, MARCH, 2005

[22] Polo J., Prados J. and Delgado J., *Interoperability between ODRL and MPEG-21 REL*, First International ODRL Workshop, Vienna (Austria), April 2004, ISBN 1-74064-500-6

[23] Rodríguez E., Delgado J., *Verification algorithms for governed use of multimedia content*. Online Information Review Journal. Vol. 31, pages 38-58. 2007.

[24] Rodríguez E., Prados J. and Delgado J., *Interoperability between different Rights Expression Languages and Protection Mechanisms*, 1st International Conference on Automated Production of Cross Media Content for Multichannel Distribution (AXMEDIS 2005), Florence (Italy), December 2005

[25] UNIFIED MODELLING LANGUAGE WEB SITE, *http://www.uml.org*

[26] VISNET II NOE, *http://www.visnet-noe.org/*

[27] W3C XML-SIGNATURE SYNTAX AND PROCESSING STANDARD, *http://www.w3.org/TR/xmldsig-core/*