Universitat Politècnica de Catalunya

Department of Computer Architecture

Ph.D. Dissertation

Rights and Services Interoperability for Multimedia Content Management

Author: Xavier Maroñas Borras

Advisor: Professor Jaime Delgado Mercé

Co-advisor: Silvia Llorente Viejo

Co-advisor: Eva Rodríguez Luna

A dissertation submitted to the Department of Computer Architecture and the Committee on Graduate Studies of Universitat Politècnica de Catalunya in partial fulfilment of the requirements for the degree of Doctor.

Author: Xavier Maroñas Borras
Advisor: Professor Jaime Delgado Mercé
Co-advisor: Silvia Llorente Viejo
Co-advisor: Eva Rodríguez Luna

Rights and Services Interoperability for Multimedia Content Management

Abstract

The main goal of this research project is to describe the definition of interoperability mechanisms between rights expression languages and policy languages. Starting from languages interoperability, the intention is to go a step further and define how services for multimedia content management can interoperate by means of service-oriented generic and standardised architectures.

In order to achieve this goal, several standards and existing initiatives will be analysed and taken into account. Regarding rights expression languages and policy languages, standards like MPEG-21 Rights Expression Language (REL), Open Digital Rights Language (ODRL) and eXtensible Access Control Markup Language (XACML) are considered. Regarding services for content management, the Multimedia Information Protection And Management System (MIPAMS), a standards-based architecture and the Multimedia service platform technologies (MSPT), also known as MPEG-M standard are considered.

The first part of the contribution describes how MPEG-21 REL, ODRL and XACML can interoperate, defining the mapping mechanisms to translate expressions from language to language. Moreover, the work done in the VISNET-II Network of Excellence and the AXMEDIS project to prove the validity of the interoperability methods described is presented.

The second part of the contribution describes how to describe standards based building blocks to provide interoperable services for multimedia content management. This definition is based on the analysis of existing content management use cases, from the ones involving less security over multimedia content managed to the ones providing full-featured digital rights management (DRM) (including access control and ciphering techniques) to support secure content management. In this section it is also presented the work done in the research projects AXMEDIS, Musiteca and Culturalive and in the standardisation of MPEG-M, particularly on elementary services and service aggregation.

A la meva familia.

Table of Contents

Part I. Introduction 9

1 Introduction 11

1.1 Context 11

1.2 Objectives 11

1.3 Structure of the document 12

Part II. State of Art 15

2 Rights Expression Languages 17

2.1 MPEG-21 Rights Expression Language (REL) 17

2.1.1 Principals 19

2.1.2 Rights 19

2.1.3 Resources 20

2.1.4 Conditions 20

2.2 Open Digital Rights Language (ODRL) 24

2.2.1 Open Mobile Alliance Digital Rights Management Rights Expression Language (OMA DRM REL) 27

2.3 eXtensible Access Control Markup Language (XACML) 28

2.3.1 XACML Policy Language 29

2.3.2 Data-flow model 30

2.3.3 XACML Context 31

3 Multimedia middleware architectures 33

3.1 Multimedia Information Protection And Management System (MIPAMS) 33

3.1.1 Description 33

3.1.2 History and evolution 33

3.1.3 MIPAMS Services 37

3.2 MPEG-M: Multimedia service platform technologies (MSPT) 39

3.2.1 History 39

3.2.2 Overview of MPEG eXtensible Middleware (MXM) 39

3.2.3 Overview of Multimedia service platform technologies (MSPT) 41

Part III. Contribution 47

4 Architectures for RELs Interoperability 49

4.1 Introduction 49

4.2 UML- and ER-based rights expression languages interoperability 49

4.2.1 Expressing XML Rights Expressions using UML 50

4.2.2 Expressing XML rights expressions using entity-relationship model 53

4.2.3 Translating rights expressions 55

4.3 XACML-based architecture for REL interoperability 56

4.3.1 Use of XACML to provide interoperability between RELs 58

4.4 Broker-based architecture for DRM systems interoperability 67

4.4.1 Application scenarios 69
5 Standards-Based Building Blocks for Modelling Content Management Scenarios 79
5.1 Introduction 79
5.2 Scenarios description 80
5.2.1 Content Licensing 80
5.2.2 Content Licensing Authorisation-Based Content Access Control 81
5.2.3 DRM-Enabled Content Access Control 81
5.2.4 DRM-Enabled Content Access for Mobile Devices 82
5.3 Standards-based Building Blocks (SB3) 82
5.3.1 SB3 Operations 83
5.4 Mapping SB3 to existing initiatives 84
5.4.1 Mapping SB3 to MIPAMS modules 85
5.4.2 Mapping SB3s to MPEG-M part 4 Elementary Services 87
5.4.3 Relationship between MIPAMS and MPEG-M 87
5.5 Business use cases 88
5.5.1 Seller use case 89
5.5.2 Buyer use case 90
5.6 Modelling business use cases applying MPEG-M Part 5 methodology 91
5.6.1 Application of the methodology to the seller use case 92
5.6.2 Application of the methodology to the buyer use case 96
5.7 Implementing MPEG-M part 5 scenarios using MIPAMS 99
5.7.1 Web application 99
5.7.2 Mobile application 102
5.7.3 Other scenarios implemented 105
Part IV. Conclusions 121
6 Conclusions and Future Work 123
6.1 Conclusions 123
6.2 Publications 124
6.3 Future Work 129
Part V. References and Abbreviations 131
7 References 133
8 Glossary of Abbreviations 139

List of Figures

Figure 1. REL License Structure. 18
Figure 2. MPEG-21 REL License Example. 22
Figure 3. MPEG-21 Authorisation Request. 23
Figure 4. MPEG-21 REL Authorisation Story. 24
Figure 5. ODRL model (extracted from http://www.w3.org/TR/odrl/). 25
Figure 6. ODRL License. 26
Figure 7. ODRL agreement element. 26
Figure 8. ODRL permission element. 27
Figure 9. OMA DRM REL license structure. 28
Figure 10. Simplified Policy language model. 30
Figure 11. XACML Data-flow diagram (from http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf). 31
Figure 12. XACML context (from http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf). 32
Figure 13. XACML Request. 32
Figure 14. XACML Response. 32
Figure 15. DMAG MIPAMS Original Platform. 34
Figure 16. DMAG MIPAMS First Evolution. 34
Figure 17. DMAG MIPAMS Refinement. 35
Figure 18. DMAG MIPAMS Architecture. 37
Figure 19. MXM Components model. 40
Figure 20. MPEG-M device architecture; the middleware is populated by Technology Engines (TEs), Protocol Engines (PEs) and one or more Orchestration Engines (ORCH). 43
Figure 21. MPEG-21 REL UML Class Diagram. 51
Figure 22. OMA DRM REL UML Class Diagram. 52
Figure 23. MPEG-21 REL ER Diagram. 54
Figure 24. OMA DRM REL ER Diagram. 55
Figure 25. Proposed XACML architecture. 58
Figure 26. ODRL Rights vs XACML Policy. 59
Figure 27. MPEG-21 REL License vs XACML Policy. 63
Figure 28. iRMBroker architecture. 68
Figure 29. DRM Systems interoperability – Video Surveillance scenario. 69
Figure 30. Video surveillance use case – positive authorisation. 71
Figure 31. DRM and AC systems interoperability – Virtual Collaboration scenario. 72
Figure 32. Use Case: Alice edits document. 73
Figure 33. Use Case: Alice stores document. 74
Figure 34. Use Case: Charlie edits document. 75
Figure 35. Use Case: Charlie stores document. 75
Figure 36. Use Case: Bob reads document. 76
Figure 37. Virtual Collaboration Policy example – Aa designer. 77
Figure 38. MPEG-21 REL license example– Aa designer. 78
Figure 39. Object registration and offer creation on the web application. 90
Figure 40. Object search and purchase on the web application. Authorisation and rendering of content on the applet. 91
Figure 41. Main options in the portal. 99
Figure 42. Object registration. 100
Figure 43. Resource registration. 100
Figure 44. Offer creation. 101
Figure 45. Search between objects purchased by the user. 101

Figure 46. MIPAMS Player applet. Positive and Negative authorisation. 102
Figure 47. MIPAMS connection with MPEG-M. 102
Figure 48. Some application captures. 103
Figure 49. Object and license information screen. 104
Figure 50. Authorising content rendering and rendering content in Android player. 105
Figure 51. AXMEDIS simplified DRM architecture. 106
Figure 52. Different mobile implemented solutions. 107
Figure 53. Registration of image or video captured with the mobile device, including offers. 109
Figure 54. Search of object reports. 111
Figure 55. MIPAMS mobile client screenshots. 114
Figure 56. CulturaLive project architecture. 115
Figure 57. Licensing, Authorisation and Reporting in CulturaLive. 116
Figure 58. MPEG-21 REL license for giving access permission to live event. 118
Figure 59. Licensing link from Freebase to Musiteca's trading portal. 119

List of Tables

Table 1. Namespace prefixes. 19
Table 2. Elementary Services classified by Operations and Entities 45
Table 3. Syntax for the rights and policy elements. 59
Table 4. Syntax for the permission and rule elements. 60
Table 5. Syntax for the party and subject elements. 60
Table 6. Syntax for the action-related elements. 60
Table 7. Syntax for the asset and resource elements. 61
Table 8. Syntax for the constraint and condition elements. 62
Table 9. Syntax for the license and policy elements. 64
Table 10. Syntax for the grant and rule elements. 64
Table 11. Syntax for the principal and subject elements. 65
Table 12. Syntax for the right and action elements. 65
Table 13. Syntax for the right and action elements. 66
Table 14. Syntax for the condition elements. 67
Table 15. Scenarios functionality summary 80
Table 16. Standards-based building blocks 82
Table 17. SB3 operations. 83
Table 18. Correspondence between SB3 and MIPAMS Services 85
Table 19. Correspondence between SB3 and MIPAMS Services operations 85
Table 20. Correspondence between SB3 and Elementary Services 87
Table 21. Correspondence between SB3 Operations, MIPAMS Operations and Elementary Services. 88

Acknowledgements

This research work was carried out under the framework of the Distributed Multimedia Applications Group (DMAG), and also in the context of the PhD program on Computer Architecture and Technology held in the Department of Computer Architecture of the Universitat Politècnica de Catalunya.

The work would not have been possible without the cooperation and advice of my directors Jaime Delgado, Silvia Llorente and Eva Rodríguez. Also, I cannot forget to thank the rest of DMAG members, whose support has been very encouraging.

And finally, I want to thank Silvia, Nico, my family and all my friends for their understanding, endless patience and encouragement.

Part I. Introduction

1 Introduction

This introductory chapter serves to two purposes. The main one is to describe the objectives of the research work. Then, it facilitates the reading of the document by giving and overview of its structure and providing the necessary elements to position the work in the framework of other research projects.

1.1 Context

For the past few years, many content managers have emerged. From the simplest players for specific formats, such as VHS players, to the most complex media centres capable of playing multiple formats on the same device. Furthermore, with the evolution of these content managers has also evolved the way the people use them, stop being only consumers, to be also the content authors. For that reason, many of these managers have created their own access control mechanisms to control the life cycle of the content.

In order to properly express these roles of creator and consumer, the content managers had defined a whole series of rights languages and policies that would allow either authenticate the authorship, protect the content, control its consumption or even track all the movements. In addition, it was necessary to define a service architecture to be able to offer all these functionalities in each of the platforms.

At the end, all these have result in a very wide variety of multimedia content managers, each with its own rights language and its own services architecture. And that is the main problem of all this platforms, even most of them have become a “standard”, they have not made any effort on trying to define a common language or architecture that enables the communication between them. And so, it is becoming more necessary to define something that makes possible the interoperability between them, for the user to be able to act over a resource using any of these platforms interchangeably.

This “interoperability” is the premise on which this research work is based, and on which have been researching and proposing several ways to afford the problem and the corresponding solutions and conclusions reached.

1.2 Objectives

The main aim of this research work is interoperability.

In this thesis, interoperability is considered with two different purposes. First, to control the access to multimedia content by using different rights expression languages and policy languages. Second, interoperability applied to the provision of complex services coming from different organisations and platforms. Therefore, the title of the PhD Thesis work, “Rights and Services Interoperability for Multimedia Content Management”, identifies these two interoperability purposes (“Rights” and “Services”) in the specific context of “Multimedia Content Management”.

This research has been carried out inside the Distributed Multimedia Applications Group [DMA13a].

In order to achieve the goals, several standards and initiatives have been analysed and taken into account. From the standards side, the most relevant ones that have been considered come from the Moving Pictures Experts Group (MPEG) [MPE13a], a working group of ISO/IEC whose mission is to develop standards for coded representation of digital audio and video and related data. In this context, the standards considered are MPEG-21 and MPEG-M. Other rights expression languages and policy languages have been considered for the sake of the completeness of the interoperability analysis.

MPEG-21 specifies a multimedia framework with the objective of providing interoperability among systems that deliver multimedia content. Nevertheless, MPEG-21 describes "separated" components or elements (Digital Items, Rights Expressions, Protection Mechanisms, etc.), but does not describe the way they should interconnect to offer complex services covering the secure management and distribution of multimedia content. In this sense, the MIPAMS [DEL11a] platform, a standards-based service oriented modular platform developed by the DMAG, allows the implementation of complex multimedia services by properly combining its modules. The modules comprising this platform implement different parts of the MPEG-21 standard, such as Rights Expression Language (REL) [ISO04a], Event Reporting (ER) [ISO06b], Digital Item Declaration (DID) [ISO05a] and Intellectual Property Management and Protection Components (IPMP) [ISO06a] and provide the mechanisms to combine them as user applications require. For some of the modules implemented, different versions are provided to support different standards. MIPAMS also facilitates interconnection to external systems by using Simple Object Access Protocol (SOAP) [W3C07a] based web services. From the work done in MIPAMS [TOR04a], several DMAG members contributed to the other MPEG standard relevant for this research work: MPEG-M [ISO13a]. This standard defines a suite of standards that are being developed for enabling the easy design and implementation of media-handling value chains.

Rights Expression Languages (and not only the MPEG-21 one) and Multimedia Middleware Architectures (MIPAMS and MPEG-M) are therefore the key background of this PhD Thesis. However, work was already started in different previous research projects that helped the DMAG to develop MIPAMS and contribute to the MPEG-21 and MPEG-M standards.

Starting from this, the objectives could be summarised in the following steps:

- First: Analyse how to provide mappings between rights expression languages and policy languages. The first approach will be to use UML, while XACML will be used in a subsequent step.
- Second: The next step for rights information mapping will consist in specifying a translation architecture, where different scenarios will be implemented.
- Third: Analyse use cases and scenarios for secure management and distribution of multimedia content and describe features needed. The necessary building blocks for the development of those scenarios will be specified.
- Fourth: Analyse how to implement the identified scenarios with systems developed in existing standards-based initiatives.

1.3 Structure of the document

This document is mainly composed of two parts, i.e. the *State of the Art* and the *Contribution*, completed by other smaller sections such as the introduction or the conclusions, which includes publications, and future work.

The State of the Art part of the document is split into two chapters (chapters 2 and 3) and compiles all the results of the exploratory phase. Chapter 2 presents the information related to rights expression languages and policy languages. In particular: MPEG-21 Rights Expression Language [ISO04a], Open Digital Rights Language (ODRL) [W3C02a] and eXtensible Access Control Markup Language (XACML) [OAS05a]. For chapter 3, in order to be able to define standards-based building blocks to facilitate the provision of interoperable services for the secure multimedia content management and distribution, two different initiatives in this area have been selected. In particular, the choices have been a standards-based service oriented architecture, the Multimedia Information Protection And Management System (MIPAMS) [DEL11a], and the Multimedia Service Platform Technologies (MSPT) standard, also known as MPEG-M [ISO13a].

The Contribution part of the document describes, as stated in the Objectives sub-section, the research that has been carried out in the fields of, first, rights expression and policy languages interoperability and, second, in the definition of standards based building blocks to provide interoperability between content management and distribution services.

Part II. State of Art

2 Rights Expression Languages

The different parties involved in the online distribution and consumption of multimedia resources need to exchange information about the rights, terms, and conditions associated with each resource at each step in the multimedia resource lifecycle. For example in distribution and super distribution business models, the information related to the rights and the terms and conditions under which the rights may be exercised needs to be communicated to each participant in the distribution chain.

In an end-to-end system, other considerations such as authenticity and integrity of Rights Expressions become important. For example, any content provider or distributor who issues rights to use or distribute resources must be identified and authorised. In addition, a Rights Expression may be accessed by different participants who require mechanisms and semantics for validating the authenticity and integrity of the Rights Expression. A common Rights Expression Language that can be shared among all participants in this digital workflow is required.

Right expression languages (RELs) are languages devised to express conditions of use of digital content. They have been proposed to describe licenses governing the terms and conditions of content access. Right expression languages can be used for example to describe an agreement between a content provider and a distributor, or between a distributor and an end user or can be used to express the copyright associated to a given digital content such as video, an e-book or a piece of music, by specifying under which conditions the user is allowed to exercise a right such as play, print or copy.

The most relevant right expression languages are MPEG-21 REL based on the eXtensible rights Markup Language (XrML) [CON02a] proposed by ContentGuard, Inc. [CON13a] and the Open Digital Rights Language (ODRL) [W3C02a] proposed by Renato Ianella from IPR Systems [IPR13a]. XrML and ODRL syntactically are based on XML while structurally they both conform to the axiomatic principles of rights modelling first laid down by among others Dr. Mark Stefik of Xerox PARC, the designer of the Digital Property Rights Language (DPRL) [XER98a]. Main differences between both languages are that ODRL has some media specific constructs that XrML does not specify, as the inheritance model and the ability of specifying attributes of digital objects, as file formats or encoding rates among others. On the other hand, ODRL has the advantage that is more concise, then resultant licenses are more compact that their equivalents in XrML. This is important for example in mobile environments and this is one of the reasons why OMA chose ODRL instead of XrML.

License Script [CHO03a] is a logic-based rights expression language that tries to avoid some intrinsic disadvantages of XML-based RELs such as the complicated syntax of them when the conditions of use become complex and the lack of formal semantics. License Script has a declarative as well a procedural reading and this makes it possible to capture a multitude of sophisticated usage patterns precisely and unambiguously.

2.1 MPEG-21 Rights Expression Language (REL)

Part 5 of the MPEG-21 standard specifies the syntax and semantics of a Rights Expression Language. MPEG chose XrML as the basis for the development of the MPEG-21 Rights expression language. MPEG-21 Rights Expression Language (REL) [ISO04a] specifies the syntax and semantics of a language for issuing rights for users to act on Digital Items and elements within them.

The most important concept in REL is the license that conceptually is a container of grants, each one of which conveys to a principal the sanction to exercise a right against a resource. A license if formed by the elements title, inventory, grant or grantGroup and otherInfo. Title element provides a descriptive phrase about the License that is intended for human consumption in user interfaces. Inventory element is used for defining variables within a License. The Grants and GrantGroups contained in a license are the means by which authorisation policies are conveyed in the REL architecture. In the other information element additional information relevant for the license can be placed. It uses the wildcard construct from XML Schema. It is important to take into account that not all processors of REL licenses will understand the semantics of the fields within the elements. Figure 1 shows the structure of a REL License.

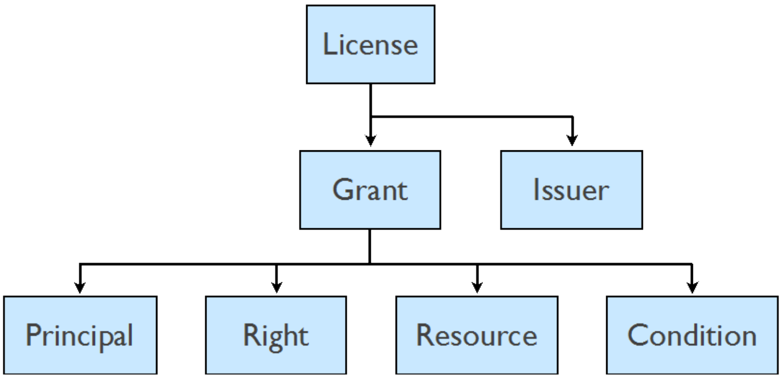


Figure 1. REL License Structure.

The most important concept within a license is the grant that 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. A Grant is an XML structure that is at the heart of the rights management and authorisation policy semantics that REL is designed to express.

A grant is formed by four elements, a Principal that represents the unique identification of an entity involved in the granting of rights. A Right that specifies an action or activity that a Principal may perform on, or using, some associated target Resource. A Resource that represents the digital object against which the Principal of a Grant has can exercise a Right. The Condition element represents grammatical terms, conditions and obligations that a Principal must satisfy before it may take advantage of an authorisation conveyed to it in a Grant. The issuer element may contain two pieces of information, an identification of the issuer, possibly coupled with a digital signature for the license and a set of issuer-specific details about the circumstances under which the license has been issued. The optional issuer-specific element may include any of the following information: the specific date and time at which this issuer claims to have effected the issuance of the license; and the mechanism or mechanisms by which the Issuer of the license will, if he later revokes it, post notice of such revocation.

The structure of a REL license is the described if it is in clear text, but it can contain an encryptedLicense element if the license is encrypted. The encryptedLicense element provides a mechanism by which the contents of a License may be protected and then not accessed by unauthorised parties. This mechanism is based on the XML Encryption Syntax and Processing (XML Encryption).

The principals, rights, resources and conditions of the REL are organised in three main groups. The first one, the Core specifies structural elements and types and how are they related. The standard extension and the multimedia extension specify standard or multimedia principals, rights, resources and conditions. Each one of the parts is related to a namespace. Table 1 gives the prefix and the corresponding namespace.

Table 1. Namespace prefixes.

Part	Namespace prefix	Namespace
Core	r	urn:mpeg:mpeg21:2003:01-REL-R-NS
Standard	sx	urn:mpeg:mpeg21:2003:01-REL-SX-NS
Multimedia	mx	urn:mpeg:mpeg21:2003:01-REL-MX-NS

At the heart of REL is the REL Core Schema whose elements and types define the core structural and validation semantics that comprises the essence of the specification. The REL Core Schema includes different elements and types organised in four main groups:

2.1.1 Principals

Within REL, principals represent the unique identification of an entity involved in the granting or exercising of rights. They identify the entity that is permitted to exercise granted rights. The principal element and its type are both conceptually abstracts. Then, principal elements do not indicate how a particular principal identified and authenticated. For this purpose, types that are derivations of the principal element have been defined. These types have been defined in extensions to REL. However, there are derivations that are important and central enough to be defined within the REL core itself:

- allPrincipals: This element is a simple container of Principals. Semantically, it represents the conjunction of all the principals represented by all of its children.
- keyHolder: Instances of the KeyHolder element represent entities which are identified by their possession of a certain cryptographic key.

2.1.2 Rights

Within REL, right represent a verb that a principal may be authorised to carry out. Typically, a right specifies an action or activity that a principal may exercise over a digital resource.

The element right and its type are conceptually abstract. Therefore, the type right itself does not indicate any action or activity to be exercised. These actions or activities are defined in types that are derivations of the right element. Such derived types also have been defined in extensions to REL. However, the following rights pertain to the REL core itself:

- issue: When the right of a license is to issue, then the resource against which the right is applied shall be a grant or grantGroup that conveys the authorisation for the principal to issue the resource.
- obtain: This right can be conceptualised as an offer or advertisement for the sale of the contained grant. When the right of a license is to obtain, then the resource shall be a grant or a grantGroup.
- possessProperty: It represents the right for the associated principal to claim ownership of a particular characteristic, for example that this principal is member of a video club, which is listed as the resource associated to this right.
- revoke: This right represents the authorised act of exercising the revoke right by a principal.

2.1.3 Resources

An instance of type resource represents the object against which a principal of a grant can some right. The element resource and its type are conceptually abstract. Therefore, the type resource does not indicate any digital object. The digital objects have been defined in types that are derivations of the resource element in extensions to REL. The relevant resources defined within the REL core are:

- digitalResource: This element provides the means by which an arbitrary sequence of digital bits can be identified as being the target object of a grant within a license
- propertyAbstract: An instance of type propertyAbstract represents some kind of property that can be possessed by principals via possessProperty right.

2.1.4 Conditions

Within REL, instances of the type condition represent restrictions and constraints that a Principal must satisfy before it can exercise the granted rights. The semantic specification of each condition indicates the details of the obligations and constraints that use of the condition imposes. Then, when these requirements are fulfilled, the condition is satisfied.

The condition element and its type are conceptually abstracts. Therefore, the type Condition does not indicate the any restriction or constraint. The conditions have been defined in types that are derivations of the condition element in extensions to REL. The conditions defined within the REL core that can be considered as relevant to detail:

- AllConditions: This element is a simple container of conditions
- validityInterval: A ValidityInterval condition indicates a contiguous, unbroken interval of time in which rights can be exercised. The start and end of this interval are specified by the child elements of the validityInterval element:
    - notBefore element indicates the instant in time at which the interval begins
    - notAfter element indicates the instant in time at which the interval ends

The Standard Extension schema defines terms to extend the usability of the Core Schema, some of them are:

- Right Extensions: Right Uri.

- Resource Extensions: Property Extensions and Revocable.
- Condition Extensions: Stateful Condition, State Reference Value Pattern, Exercise Limit Condition, Transfer Control Condition, Seek Approval Condition, Track Report Condition, Track Query Condition, Validity Interval Floating Condition, Validity Time Metered Condition, Validity Time Periodic Condition, Fee Condition and Territory Condition.
- Payment Abstract and its Extensions: Payment Abstract, Rate, Payment Flat, Payment Metered, Payment per Interval, Payment per Use, Best Price Under, Call for Price and Markup.
- Service Description: WSDL and UDDI
- Country, Region and Currency Qualified Names: Namespace URI Structure, Country Qualified Names, Region Qualified Names and Currency Qualified Names.
- Matches XPath Function: Regular Expression Syntax and Flags.
- The REL Multimedia Extension expands the Core Schema by specifying terms that relate to digital works. Specifically describes rights, conditions and metadata for digital works, that includes:
- Rights: Modify, Enlarge, Reduce, Move, Adapt, Extract, Embed, Play, Print, Execute, Install, Uninstall and Delete.
- Resources: Digital Item Resources.
- Conditions: Resource Attribute Conditions, Digital Item Conditions, Marking Conditions, Security Conditions and Transactional Conditions.
- Resource Attribute Set Definitions: Complement, Intersection, Set and Union.

A typical example of a REL license issued to an end-user. In this case a distributor, MusicDist, issues to a user, Alice, a license that permits her the right of play a song, TheEnd.mp3, during this year. The license is sketched in Figure 2.

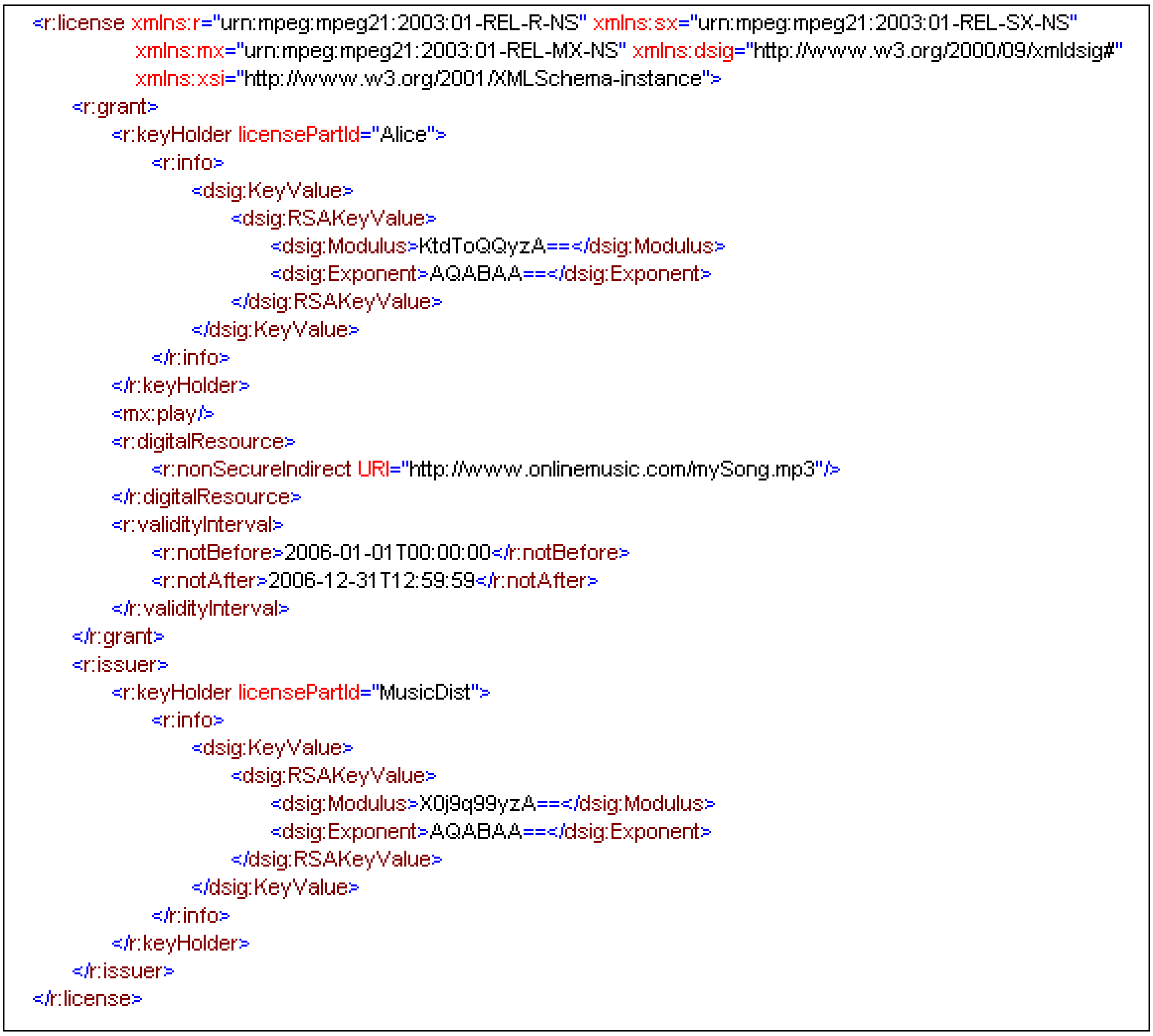
The main elements of the license are the grant and the issuer. The grant element is formed by four elements. The keyHolder that represents the user, Alice, which is identified by her possession of a certain cryptographic key. Then, she is identified as the Principal that possess the private key that corresponds to this-here public key. The play element that represents the right. The definition of Play in the Rights Data Dictionary is to derive a transient and directly perceivable representation of a resource. The digitalResource element that provides a means by which an arbitrary sequence of digital bits can be identified as being the target object of relevance within the Grant. Conceptually, an instance of DigitalResource defines an algorithm by which a sequence of bits is to be located. If the bits are to be physically located at some external location, for example in this example they are located on a Web site, the nonSecureIndirect element child is used where it is indicated the algorithm used to allocate the bits. In this example it is indicated that the song is in the URI http://www.webmusic.com/TheEnd.mp3. And the fourth one, the ValidityInterval element that represents the condition. It indicates a contiguous, unbroken interval of time. The semantics of this Condition is that the interval of the exercise of a Right to which a ValidityInterval is applied must lie wholly within this interval. The delineation of the interval is expressed by the presence, as children of the Condition, of up to two specific fixed time instants, notBefore of type xsd:dateTime, indicates the inclusive instant in time at which the interval begins, 1 January 2006. And the notAfter element of type xsd:dateTime, indicates the inclusive instant in time at which the interval ends, 31 December 2006. Therefore, with this license the user can play the song during this year.

The issuer element indicates the entity that issues the license. In this example, it represents the music distributor that has the right to issue this kind of licenses to end-users.

Other important concept of the REL is the authorisation model that may be used by any implementation of software which makes an authorisation decision using REL licenses. The central question that lies in this decision making process "is a principal authorised to exercise a right against a resource?"

The REL Authorisation Model makes use of an authorisation request (see Figure 3), an authorisation context, an authorisation story, and an authoriser.

An authorisation request can be conceptualised as representing the question if is it permitted for a given Principal to perform a given Right upon a given Resource during a given time interval based on a given authorisation context, a given set of Licenses, and a given trust root.



```
<r:license xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS"
           xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <r:grant>
        <r:keyHolder licensePartId="Alice">
            <r:info>
                <dsig:KeyValue>
                    <dsig:RSAKeyValue>
                        <dsig:Modulus>KtdToQQyzA==</dsig:Modulus>
                        <dsig:Exponent>AQABAA==</dsig:Exponent>
                    </dsig:RSAKeyValue>
                </dsig:KeyValue>
            </r:info>
        </r:keyHolder>
        <mx:play/>
        <r:digitalResource>
            <r:nonSecureIndirect URI="http://www.onlinemusic.com/mySong.mp3"/>
        </r:digitalResource>
        <r:validityInterval>
            <r:notBefore>2006-01-01T00:00:00</r:notBefore>
            <r:notAfter>2006-12-31T12:59:59</r:notAfter>
        </r:validityInterval>
    </r:grant>
    <r:issuer>
        <r:keyHolder licensePartId="MusicDist">
            <r:info>
                <dsig:KeyValue>
                    <dsig:RSAKeyValue>
                        <dsig:Modulus>X0j9q99yzA==</dsig:Modulus>
                        <dsig:Exponent>AQABAA==</dsig:Exponent>
                    </dsig:RSAKeyValue>
                </dsig:KeyValue>
            </r:info>
        </r:keyHolder>
    </r:issuer>
</r:license>
```

Figure 2. MPEG-21 REL License Example.

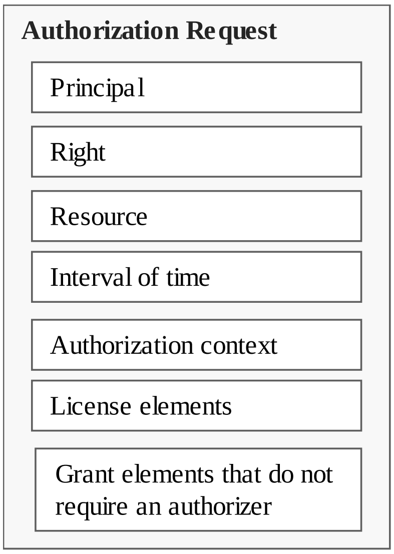


Figure 3. MPEG-21 Authorisation Request.

The authorisation request contains the following members:

- the principal element, which is the identity of the entity for which permission is requested
- the right element, which embodies the semantics of the action which is requested to be permitted
- the resource element identifying the Resource upon which permission is requested
- the interval of time during which the requested performance of the right by the principal upon the resource is considered to take place. This may be either an instantaneous point in time or an unbroken interval of time
- the authorisation context containing properties representing statements that are to be considered true for the purposes of establishing the requested permission
- the set of license elements that may be consulted to establish the requested permission. The algorithm will attempt to find authorised grants or grantGroups within this licenses that it can use to establish a basis for an affirmative authorisation decision
- the set of grant elements that do not require an authoriser for the purposes of establishing the requested permission
- The authorisation story (see Figure 4) contains the following elements:
- a primitive grant, it is used to demonstrate to which authorisation requests the authorisation story applies
- either a grant or a grantGroup, it represents the actual grant or grant group that is authorised by the authoriser of the authorisation story
- an authoriser, it contains the following members:
    - the license in which the principal is authorised
    - the principal that authorised the license above
    - the time instant in which the license was issued
    - the authorisation context that contains the properties representing statements that were considered true for the purposes of establishing the permission
    - an authorisation story

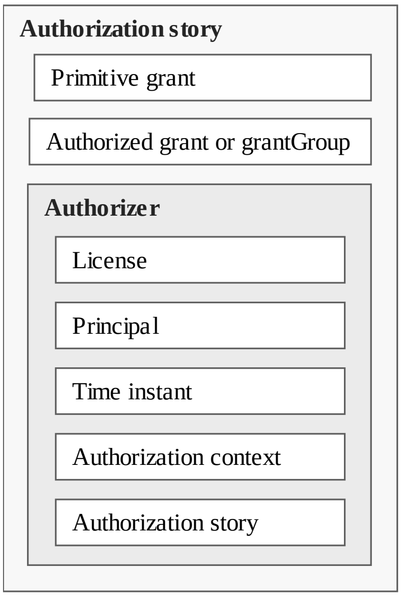


Figure 4. MPEG-21 REL Authorisation Story.

2.2 Open Digital Rights Language (ODRL)

The Open Digital Rights Language (ODRL) [W3C02a] 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 and innovative use of digital resources in publishing, distributing and consuming of electronic publications, digital images, audio and movies, learning objects, computer software and other creations in digital form. This is an XML-based usage grammar.

ODRL is focused on the semantics of expressing rights languages and definitions of elements in the data dictionary. ODRL can be used within trusted or untrusted systems for both digital and physical assets (resources).

ODRL is based on an extensible model for rights expressions. It is presented in Figure 5.

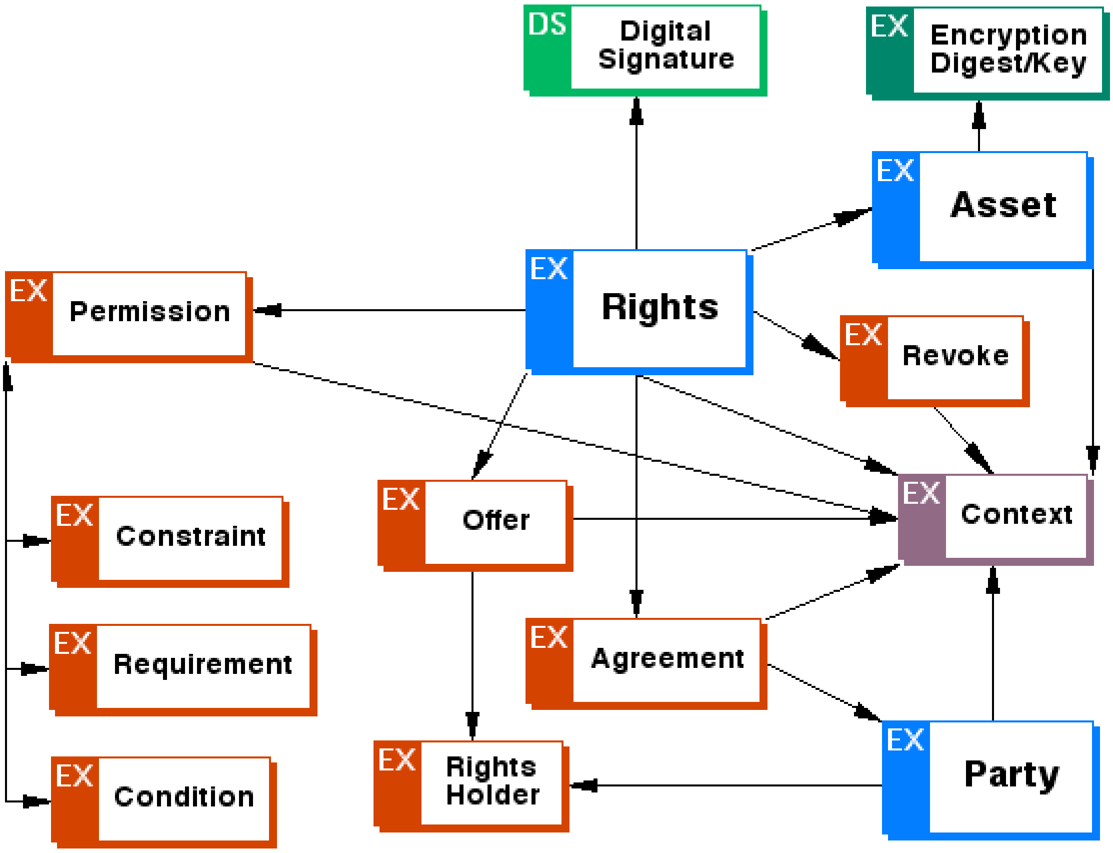


Figure 5. ODRL model (extracted from http://www.w3.org/TR/odrl/).

The three core entities of the ODRL foundational model and their relationships are sketched in Figure 6 and detailed below:

- Party includes end users and Rights Holders. Parties can be humans, organisations, and defined roles. In the previous example, Alice is the party.
- Right includes permissions, which can then contain constraints, requirements, and conditions. Permissions are the actual usages or activities allowed over the assets (e.g. play, print, etc.). Constraints are limits to these permissions (e.g. print an e-book for a maximum of 3 times). Requirements are the obligations needed to exercise the permission. Conditions specify exceptions that, if they become true, expire the permissions and re-negotiation may be required. In the previous example, print is the right that includes the constraint of “3 times”.
- Asset includes any physical or digital content. They must be uniquely identified and may consist of many subparts and be in many different formats. Assets can also be nontangible expressions of works and/or manifested in particular renditions. In the previous example, the book is the asset.

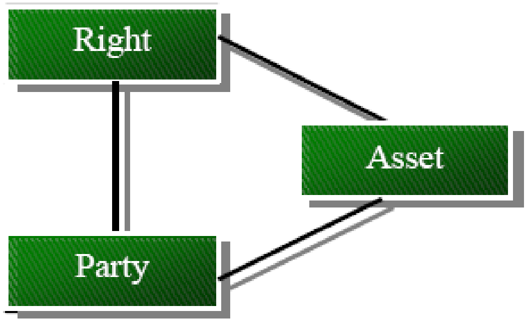


Figure 6. ODRL License.

The ODRL model provides mechanisms for defining the agreements made between parties for specific rights over assets, as well as the offers done by right holders. Both consists of an asset, rights holder, permission and context elements. Figure 7 sketches the syntactic structure for the agreement element.

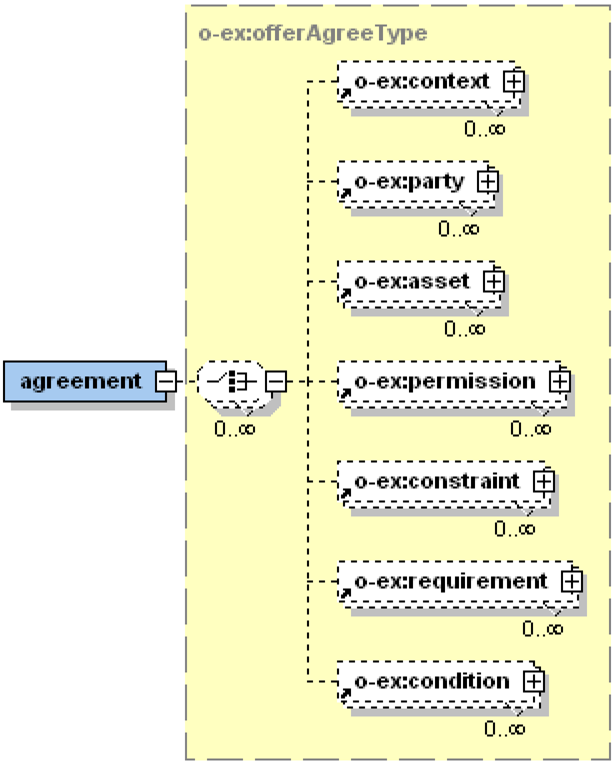


Figure 7. ODRL agreement element.

The ODRL model also defines permissions for offers and agreements, which specify the set of activities allowed over the asset, they are classified in usage, reuse, transfer and asset management. Additionally, permissions can support exclusivity in form of an attribute indicating that it is confined to nominated parties. Restriction on permissions can be defined using the constraint element. Constraints consist on aggregations of abstract elements which include usage limits, device constraints, temporal limitations, aspect constraints, and purpose of usage. The permission model is depicted in Figure 8.

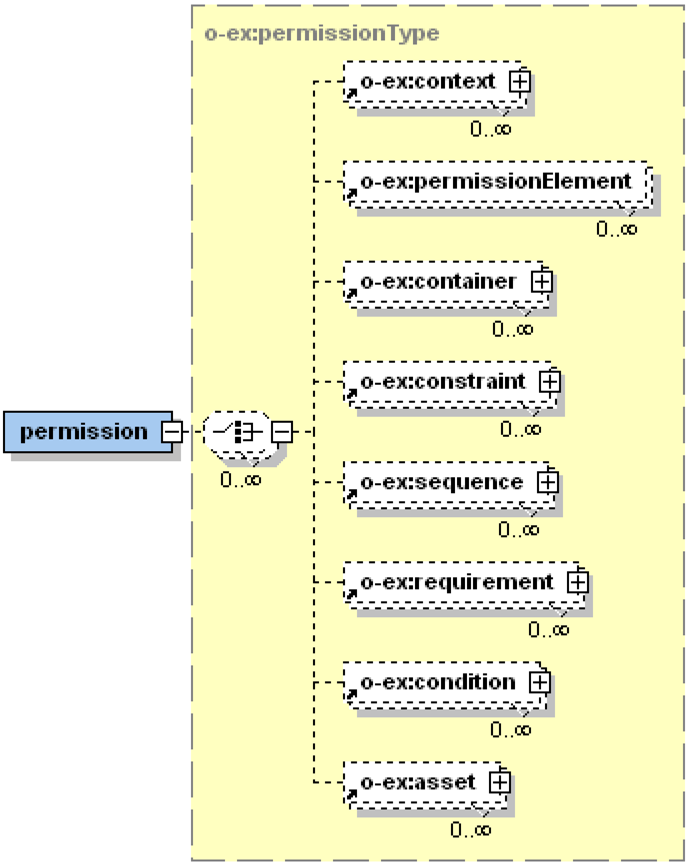


Figure 8. ODRL permission element.

ODRL supports the expression of preconditions by means of the Requirement element. Three types of preconditions have been defined: fees, user interactions and asset usage requirements (for example attribution). Preconditions in an ODRL license must be met to obtain the requested permissions.

Exceptions for permissions can be defined in ODRL. They are expressed as rights conditions that if occur, that is, become true, associated permissions are no longer valid.

Rights holders also can be represented in ODRL. They can be used to express the payment to be done to the indicated party for each transaction over the asset as percentage of the value of the transaction or as a fixed value.

2.2.1 Open Mobile Alliance Digital Rights Management Rights Expression Language (OMA DRM REL)

Open Mobile Alliance (OMA) [OMA13a] is the leading industry forum for the mobile environment. It was formed in 2002 by nearly 200 companies, including mobile operators, device and network suppliers, information technology companies and content and service providers. The requirements of the whole value chain actors' can be considered, as all of them are part of the forum.

The main aim is to provide specifications for supporting the creation of interoperable end-to-end mobile services, independent from networks and platforms.

OMA has developed OMA DRM [OMA13a], its digital rights management architecture to provide protection of content inside the mobile environment. They have also defined a rights expression language, OMA DRM REL [OMA06a], based on ODRL [W3C02a]. Using OMA DRM REL it is possible to express rights over an asset (content) defining permissions and constraints (conditions) on its usage. OMA defines its own rights expression language and data dictionary, both based on ODRL ones.

OMA DRM REL defines a subset of ODRL, being their licenses more limited. Figure 9 shows the basic structure of an OMA DRM REL license. OMA licenses do not define the party element, as it is implicit to the user of the mobile device and the rights are a subset of the ones in ODRL. The constraints can be defined at permission and right levels. Constraints at permission level affect to all rights defined inside the permission.

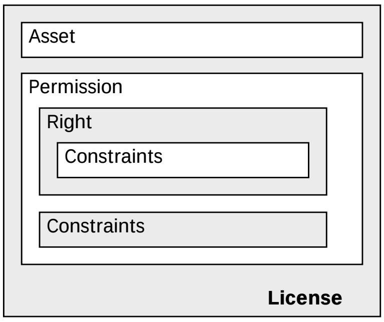


Figure 9. OMA DRM REL license structure.

2.3 eXtensible Access Control Markup Language (XACML)

The Extensible Access Control Markup Language (XACML) [OAS05a] standard was specified by OASIS [OAS13a], which is a not-for-profit organisation that drives the development and adoption of open standards. OASIS has more than 500 corporate and individual members in 100 countries around the world. OASIS produces worldwide standards for security, Web services, XML conformance, electronic publishing, etc., and for interoperability within and between marketplaces.

The XACML standard was devised for expressing authorisation policies in XML against objects that can be identified in XML. One of the reasons for defining the XACML was the amount of proprietary and application specific access control policy languages used to define policies that could not be shared across different applications. Moreover, many of the existing languages were not extensible, didn’t support policies and were not expressive enough to meet new requirements.

The XACML specification enables the use of different types of policies (time and date-based, indexable, deny policies and dynamic), of arbitrary attributes in policies, Role Based Access Control (RBAC), and security labels, without requiring changes to the applications that use XACML.

The XACML 2.0 specification consists of eleven normative documents, including four XML Schemas and seven prose specifications. The main features of XACML are defined in the core Extensible Access Control Markup Language (XACML) Version 2.0 specification, supported by the Core Policy Schema and Core Context Schema. The XACML version 2 specification provides the model descriptions for data-flow, XACML context (canonical representation of a decision request and an authorisation decision), and policy language (rule, policy, policy set).

A SAML 2.0 Profile [OAS05b] of XACML defines a profile for the use of the OASIS Security Assertion Markup Language (SAML) Version 2.0 to carry XACML 2.0 policies, policy queries and responses, authorisation decisions, and authorisation decision queries and responses.

The XML Digital Signature Profile of XACML uses the W3C XML-Signature Syntax and Processing Standard for providing authentication and integrity protection for XACML schema instances.

2.3.1 XACML Policy Language

The XACML standard specifies a policy language for expressing access control policies implemented in XML. The requirements for the XACML policy language are listed below:

- Provide methods to deal with a distributed set of policy components.
- Provide methods for combining rules and policies that apply to a particular decision request.
- Provide methods for dealing with multiple subjects.
- Provide methods for basing authorisation decision on attributes of the subject and resource; or on the contents of a resource.
- Provide operators on attributes of subjects, resources and environments.
- Provide a method to specify the actions to be performed jointly with policy enforcement.
- Provide methods for rapidly identify the policy that applies to a given, subject, resource and action.

From these requirements the XACML policy language model was defined. It is depicted is in Figure 10. The three top-level policy elements defined for this model are: *rule*, *policy* and *policySet*. The rule element contains a boolean expression that is intended to be the basic unit of management within an XACML policy administration point, and can be re-used in multiple policies. The main components of a *rule* are a *target*, an *effect* and a *condition*. 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 intended consequence of a true evaluation for the rule as stated by the writer of the rule. The *condition* element represents a boolean expression that refines the applicability of the rule.

The *policy* element consists of *rule* elements and mechanisms for combining the results of their evaluation. Specifically, it consists of a target element, a rule-combining algorithm identifier, a set of rules and obligations. The *obligations* element specifies the actions that must be performed in conjunction with policy evaluation.

Finally, the *policySet* element is defined for combining separate policies into a single combined policy. It contains a set of *policy* or other *policySet* elements and the procedure for combining the results of their evaluation.

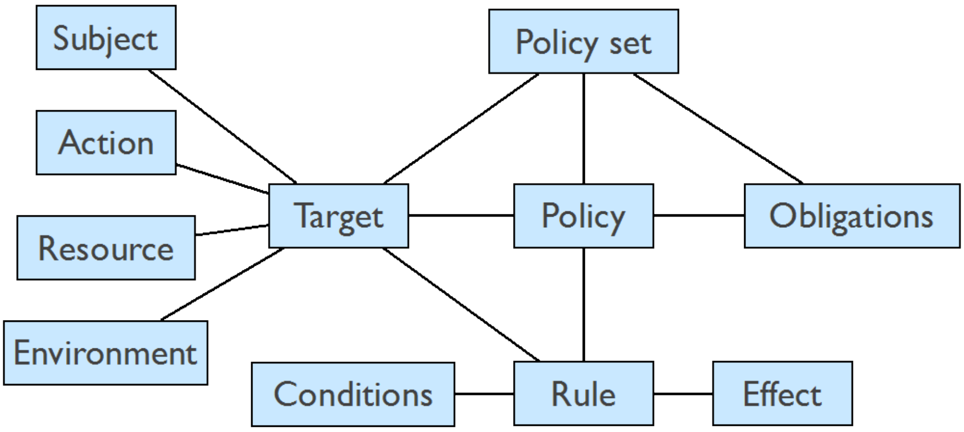


Figure 10. Simplified Policy language model.

2.3.2 Data-flow model

The XACML standard also defines the data-flow model (see Figure 11), which operates according to the following steps:

- The Policy Administration Point (PAP), which manages policies, writes a complete policy or policy set making them available to the Policy Decision Point (PDP), which will be the responsible to evaluate and issue authorisation decisions.
- The requester sends an access request to the Policy Enforcement Point (PEP), which in turn sends it to the context handler (in its native format), which is the responsible to construct the XACML request and send it to the PDP.
- The PDP requests any additional attributes to the context handler, which requests them to the Policy Information Point (PIP), since it is the system source of attribute values
- The PIP obtains the requested attributes and returns them, jointly with the resource if required, to the context handler, which sends them to the PDP.
- The PDP evaluates the policy and returns the authorisation decision within the response context to the context handler, which translates and returns it to the PEP.
- The PEP fulfils the obligations and permits or denies access to the resource.

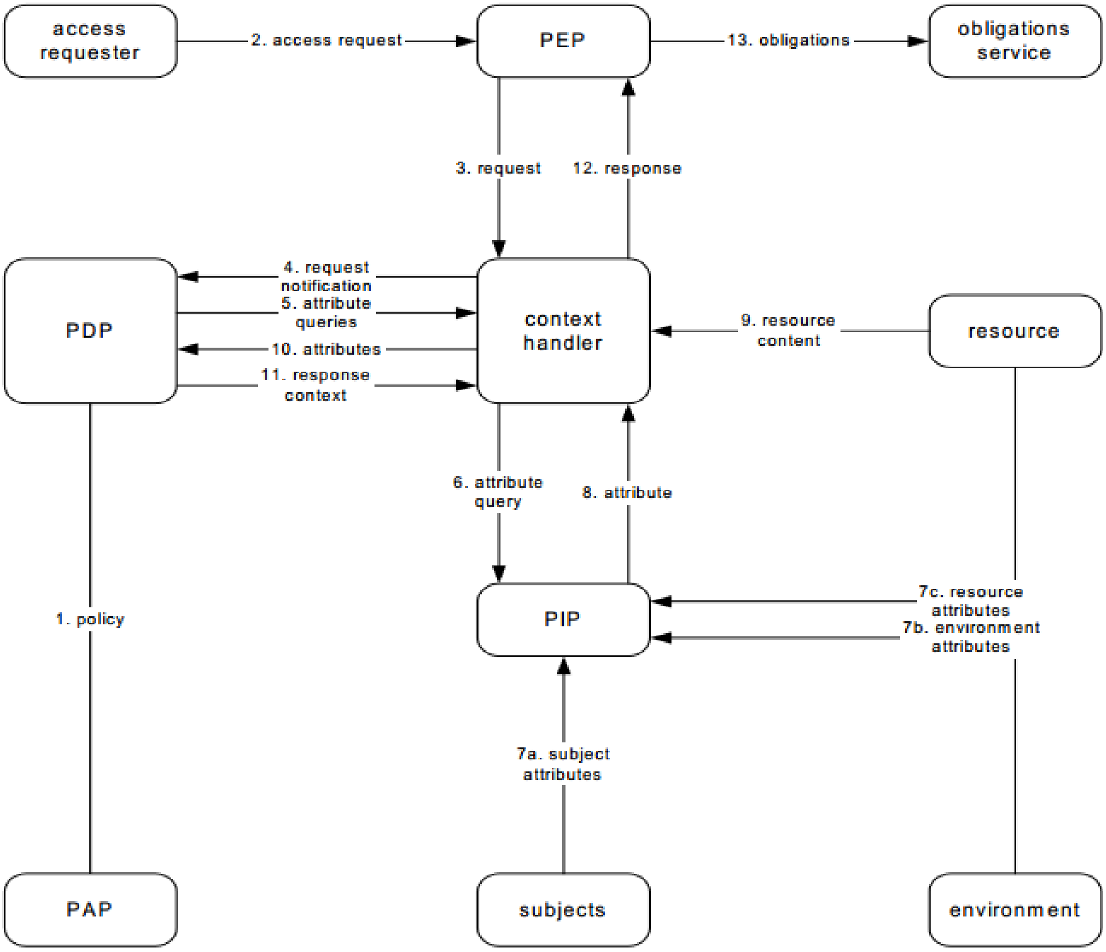


Figure 11. XACML Data-flow diagram (from http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf).

2.3.3 XACML Context

The XACML standard defines the representation for the inputs and outputs of the PDP, so-called XACML Context. It is normatively defined, as the XACML policy language, in an XML Schema. It has as input a Request element (see Figure 12), which consists of subject, resource, action and environment attributes referenced by an instance of a policy for example in form of XPath expressions over the context or attribute designators. The response consists of the decision, status and obligations.

The model for the XACML context, as well as the request and response elements, are depicted in Figures 12 to 14.

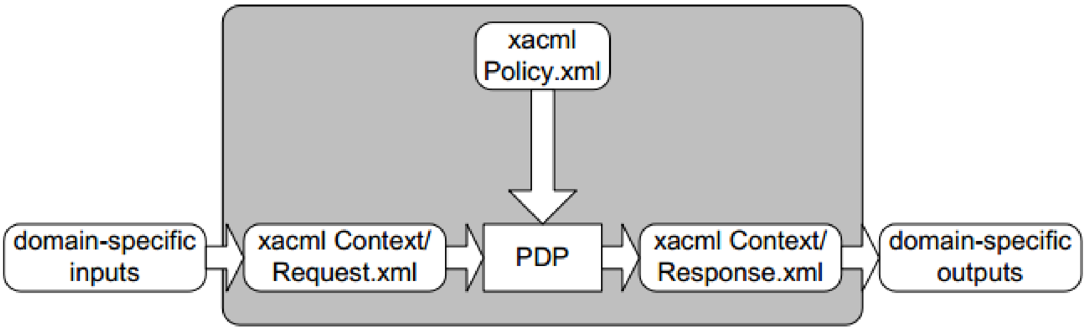


Figure 12. XACML context (from http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf).

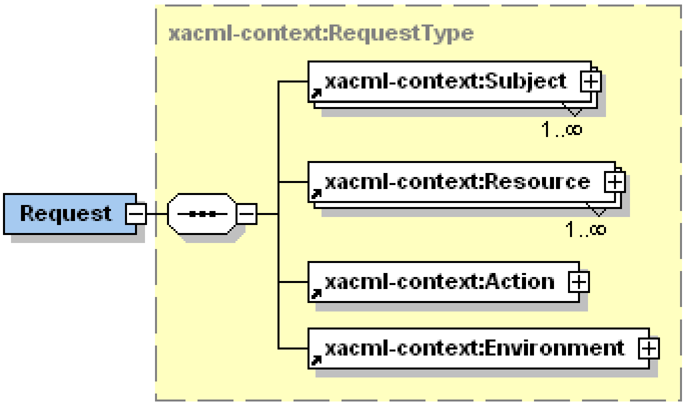


Figure 13. XACML Request.

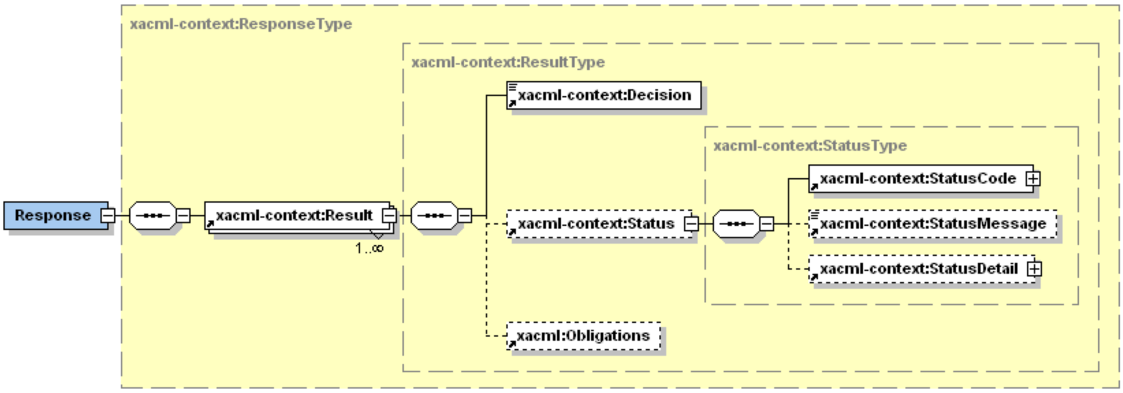


Figure 14. XACML Response.

3 Multimedia middleware architectures

3.1 Multimedia Information Protection And Management System (MIPAMS)

3.1.1 Description

MIPAMS (Multimedia Information Protection And Management System) [DEL11a] is a service-oriented content management platform, developed by the DMAG-UPC (Distributed Multimedia Applications Group, Universitat Politècnica de Catalunya) [DMA13a]. It is mainly intended to applications where management of rights is needed.

MIPAMS platform is based on the flexible web services approach, as it consists of several components and services, which provide the functionality needed for governing and protecting multimedia content. The operations provided by each service are Simple Object Access Protocol (SOAP)-based web services [W3C07a].

One of the advantages of having service-oriented content management functionality relies on the possibility of decoupling it into different subsystems depending on the needs of the application that is going to be implemented, while being able to share the same common services between different applications with different requirements, thus reducing costs.

Some examples of service functionality group are:

- Content Management, which includes functionality for registering and searching content.
- Security, which includes functionality for licensing, authorisation, protection and tracking.
- Distribution, which includes functionality for content transfer.

MIPAMS potential users can select the group of functionalities they need to implement their business scenarios, like just content management, content management and security or content management and distribution. Many combinations are possible.

MIPAMS encompasses an important part of the content value chain, from content creation and distribution to its consumption by final users, including adaptation of content [CAR11a]. Moreover, MIPAMS has proved to be useful in other application scenarios different from those related to management and distribution of multimedia content, like Social Networks [DEL10a] [LLO10a] and medical information management [ROD11a].

3.1.2 History and evolution

MIPAMS platform has evolved from its first description in [TOR04a] to its current state thanks to the work done in different research projects [AXM04a] [XAC05a] [MAC06a]. This evolution is briefly described next, as it has followed several steps. The original platform modules are shown in Figure 15.

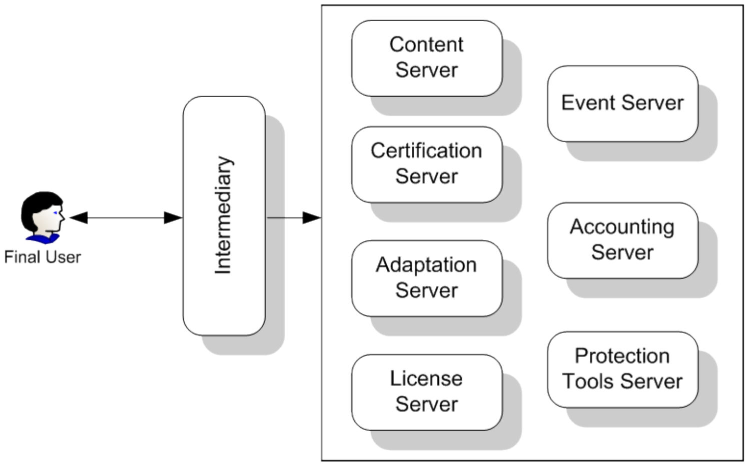


Figure 15. DMAG MIPAMS Original Platform.

The functionality of each module represented in the architecture is the following:

- Accounting server: Keeps track of what happens in the system, including statistics and traces.
- Adaptation server: It performs the adaptation of the content depending on the characteristics of the final user terminal.
- Certification server: It certifies the entities present in the system, including other modules and final users. It includes registration, authentication or key delivery.
- Content server: It provides the content that final users may request. It can be internally decomposed into several modules, for instance, if one wants to separate digital items describing resources from the resource itself or if the content is stored in an external system.
- Event server: Receives events information associated to content usage in order to advise the author or distributor of the content, if needed.
- License server: It provides licensing functionality needed to access the content. It includes license creation and license validation.
- Protection tools server: It stores the tools needed for the protection of content.

The second version of the architecture was published in [TOR05a]. In it, the servers and its functionality were refined and it was established the relationship between the different servers present in the architecture, as shown in Figure 16.

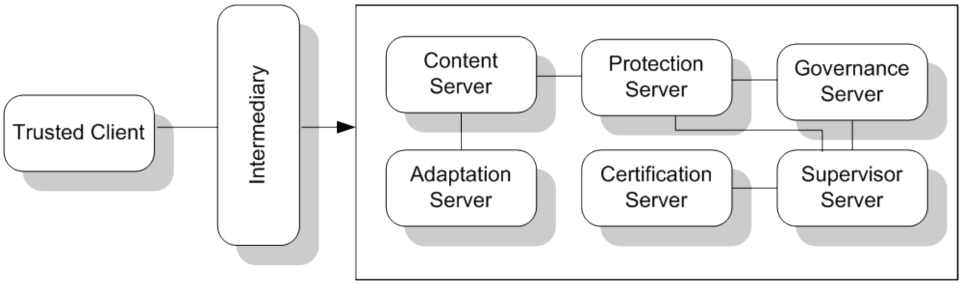


Figure 16. DMAG MIPAMS First Evolution.

The functionality of each module represented in the architecture is the following:

- Adaptation server. It performs the adaptation of content and its associated metadata, depending on transmission, storage and consumption constraints. The adaptation of metadata can also involve the adaptation of the related licenses (in fact, the creation of new ones), as derived objects or content can be seen as new creations with regard to original ones.
- Certification server. It includes registration, authentication and verification of the actions done by the user over trusted tools, like viewer, browser or editor.
- Content server. Same as the original one.
- Governance server. The governance server includes the following functionality: license generation, license storage, authorisation and translation support. It corresponds to the original License server.
- Protection server. It is responsible for protecting the content or digital objects, which become protected objects, mainly using encryption techniques and managing the encryption keys. It can also provide other protection mechanisms, like scrambling.
- Supervisor server. It receives event reports related to content usage or user blocking requests from different modules of the architecture. When a blocking request is received, the Supervisor immediately blocks a user, preventing him the access to the system. It corresponds to the original Accounting and Event servers.

The work done in several research projects helped in the refinement of the architecture and the services provided. In [TOR08a], MIPAMS architecture was completely described almost in its current shape. This version of the architecture was already validated in the research projects AXMEDIS [AXM04a], VISNET-II [VIS06a] and GILDDA [GIL07a]. Figure 17 shows the servers present at that precise moment and the relationship between them. These relationships are represented as arrows, labelled with the operations used.

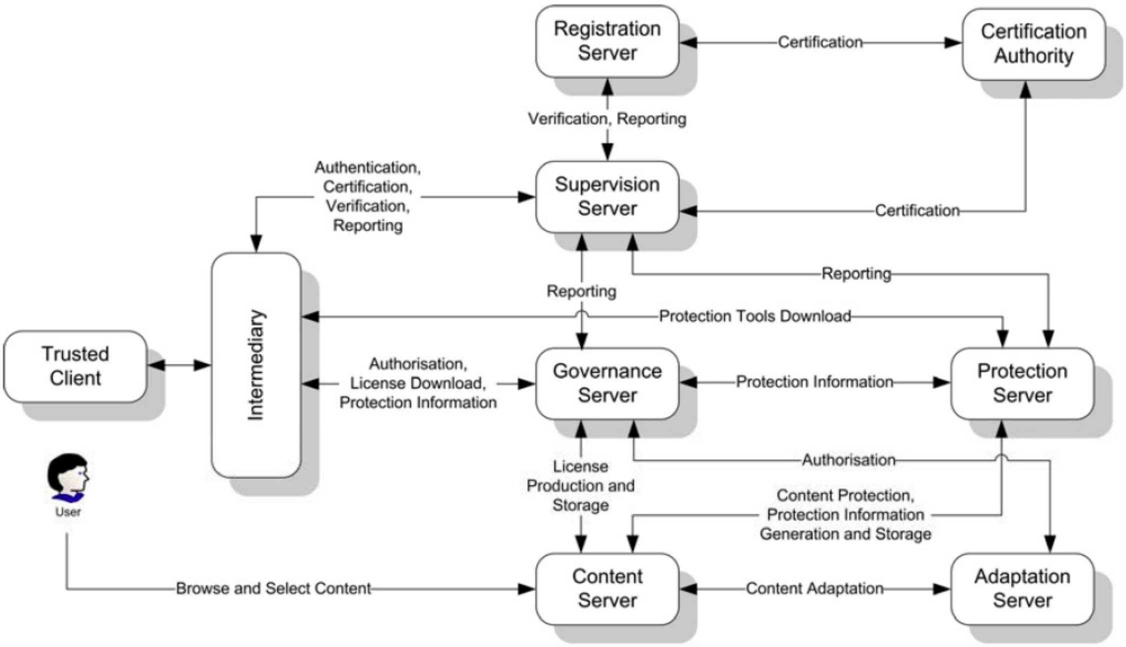
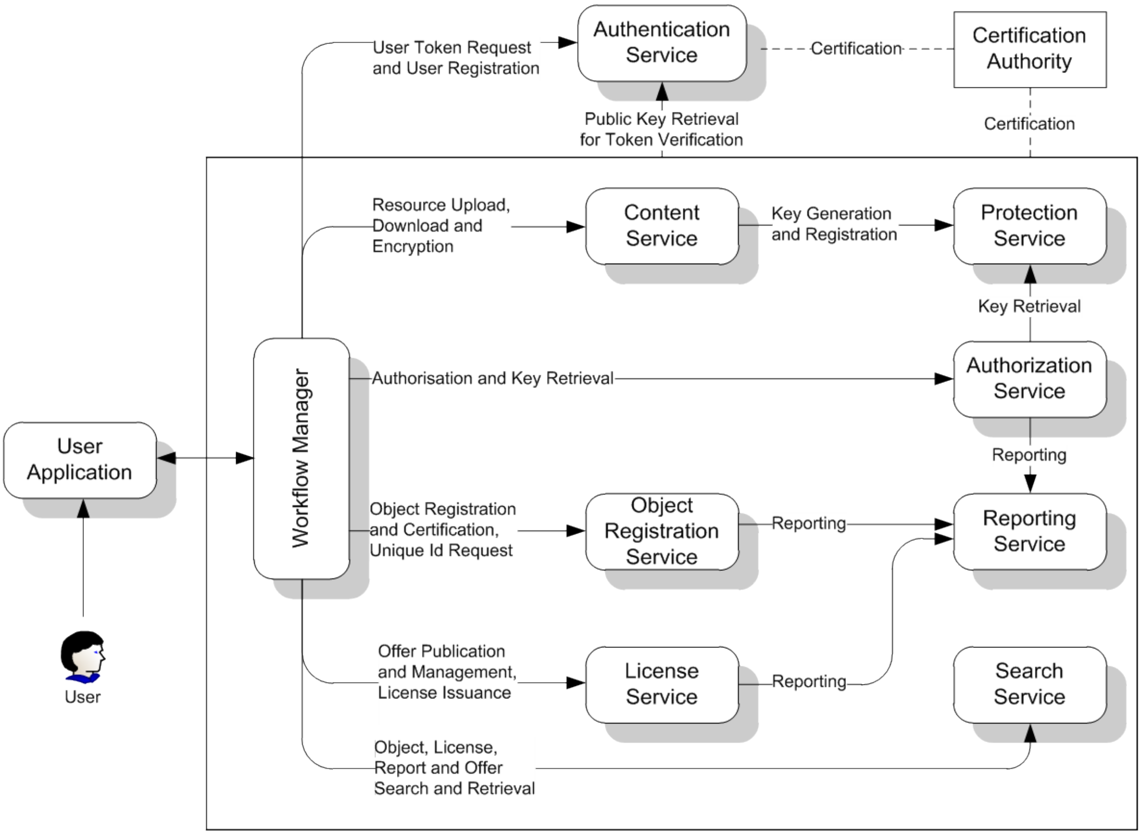


Figure 17. DMAG MIPAMS Refinement.

Some servers were revised in this version. In particular, the Certification Server was decomposed into the Registration Server, the Supervision Server (formerly Supervisor Server) and Certification Authority. The functionality of these new servers was as follows:

- Certification Authority: It issues X.509 [IET02a] for the different components and actors (users) in the system.
- Registration Server: It registers actors and tools. The registration of a user has a certificate as result. The registration of a tool allows this tool to be verified when installed on user's devices.
- Supervision Server: It authenticates and supervises actors and system components. Moreover, it is responsible for extracting and registering a fingerprint for installed tools so that they can be verified during their whole life operation and requesting the tool certificate to the Certification Authority. It also verifies the user tool integrity during its operation by checking its fingerprint, registered during certification. Moreover it receives the action reports regarding content consumption or other relevant issues in the system as e.g. license generation.

Finally, the servers evolved to become the current services, which have maintained the underlying core functionality whilst removing the operations which were not needed or were too complex to be implemented in a real scenario. To mention a few, common functions maintained from the original MIPAMS architecture are object registration using MPEG-21 DID, license formalisation and authorisation using MPEG-21 REL or event reporting using MPEG-21 ER, which have just been reorganised. For instance, the License server changed its name to Governance server and it corresponds now to License and Authorization services. Another example is Content Server, which has been decomposed into Object Registration and Content services, separating object metadata and structure from resources. This version of the architecture has also been validated in several research projects like DRM-MM [DRM05a], MCM-LC [MCM09a], IPOS-DS [IPO07a] [TOR09a] [TOR09b], Musiteca [MUS08a] and Culturalive [CUL09a]. Figure 18 depicts the MIPAMS architecture. A general overview of its components and the different services being offered (in alphabetical order) is presented in the next subsection.



User Token Request and User Registration
Authentication Service
Certification
Certification Authority
Public Key Retrieval for Token Verification
Certification
Resource Upload, Download and Encryption
Content Service
Key Generation and Registration
Protection Service
Key Retrieval
Authorisation and Key Retrieval
Authorization Service
Reporting
User Application
Workflow Manager
Object Registration and Certification, Unique Id Request
Object Registration Service
Reporting
Reporting Service
User
Offer Publication and Management, License Issuance
License Service
Reporting
Search Service
Object, License, Report and Offer Search and Retrieval

Figure 18. DMAG MIPAMS Architecture.

3.1.3 MIPAMS Services

The Authentication Service (ATS) is needed to authenticate the identity of users. It generates SAML (Security Assertion Markup Language [OAS05b])-based tokens that identify MIPAMS users. Any service in the MIPAMS architecture will require a token argument to be provided in order to authenticate users. Tokens are digitally signed by the ATS, so that they can be checked for authenticity and integrity by the receiving service. Moreover, the ATS deals with user registration and management (i.e. personal data modification, user account deactivation, etc.).

The Authorization Service (AS) checks whether a user owns any appropriate license that grants him the right to perform a requested action (e.g., play) over a digital object. The authorisation is based on the mechanism defined in [ISO04a]. The AS shares the access to the license repository with the LS. If the user is able to perform the action and the requested content is encrypted, the AS will retrieve the encryption keys from the Protection Service and return them to the requesting application. This is the only means for accessing encryption keys, which is performed as an atomic operation.

The Certification Authority (CA), which issues credentials for the different Components and Actors in the system, as X.509 certificates and private keys for the different architectural components.

The Content Service (CS) enables applications to upload and download digital resources such as audio or video files, text documents, etc. Those resources can be optionally encrypted under request, according to the available encryption mechanisms it provides. If encryption is selected, the protection keys will be first requested to the Protection Service (PS) and then registered through the same service, once encryption is performed. Content upload requires content to be uniquely identified. Since MIPAMS deals with single resource objects, the identifier being associated to content will be the same used for the object that contains it, and must be passed as input argument. This identifier can be requested to the Object Registration Service prior to the content upload or obtained from an external application using MIPAMS (it depends on the scenario).

The License Service (LS) deals with rights offers and the issuance of licenses. Rights offers are set up by content creators or rights holders after registering content. They include the rights being offered for acquisition by other users and the conditions being applicable to those rights. License issuance refers to the process by which a license is generated as the result of a rights purchase, acquisition or because a rights holder directly grants some user a set of rights. Licenses are expressed using MPEG-21 Rights Expression Language [ISO04a].

The Object Registration Service (ORS) enables applications to request a digital representation of content and metadata (i.e. digital objects) to be generated and registered in the system. Content and metadata are packaged together following the MPEG-21 Digital Item [ISO05a] approach. Once registered, objects are digitally signed by the ORS so that they can be checked for authenticity and integrity. The ORS also provides unique identifiers for those applications that need to upload content to the CS, as already explained.

The Protection Service (PS) generates encryption keys upon request, registers encryption keys associated to uniquely identified content and provides the encryption keys for protected content to the AS. When using MPEG-21 Intellectual Property Management and Protection [ISO06a] scheme and descriptors, the PS also offers the possibility to download the protection tools being used by those applications that may be out-of-date.

The Reporting Service (RS) collects usage reports regarding the registration of objects, the issuance of licenses and the authorisations being performed. It is also capable of building standards-based representations of those reports, such as MPEG-21 Event Reports [ISO06b]. Those reports may be used for computing statistics as well as for billing or tracking purposes.

The Search Service (SS) enables applications to perform accurate searches amongst metadata in the MIPAMS system. That is, it is the front-end for requesting any information present in MIPAMS services databases. Thus, it can be used for searching content, licenses, offers or reports or a combination of them.

The User Application (UA) is the player, edition tool, browser or any other means that is managed by the user to deal with the DRM functionality, such as registering and accessing protected contents. The UA may have an internal trusted module or intermediary to enforce DRM, which consists of a secure local repository for licenses, protection information, offline operation reports and other critical data. In those cases, it may be responsible for estimating tool fingerprints, require offline authorisations, unprotect content, track offline operations and manage content protection information.

The Workflow Manager (WM) may be an integral part of the UA or otherwise be located in the server part (e.g. web portal, brokerage service) to reduce the UA complexity. It can be seen as a broker to whom the UA requests different operations to be performed, as object registration, content upload, rights offer management, license acquisition, authorisation, etc.

3.2 MPEG-M: Multimedia service platform technologies (MSPT)

3.2.1 History

MPEG-M (ISO/IEC 23006) [ISO13a] [ROD10a] is an initiative of the MPEG standardisation group (ISO/IEC JTC1 SC29/WG11) [MPE13a]. It defines a suite of standards that are being developed for enabling the easy design and implementation of media-handling value chains.

It currently has two editions, the first one is called MPEG eXtensible Middleware (MXM), which is completely standardised, and the second one is known as Multimedia service platform technologies (MSPT). Some parts of MSPT are currently under development. It is expected that all its parts will be standardised during 2013.

The next subsections briefly describe both versions of MPEG-M, highlighting their features.

3.2.2 Overview of MPEG eXtensible Middleware (MXM)

MXM supports different business models and aims at satisfying the many parties which in one way or another may interact with media. To describe this standard, the focus is on the definition of the media value chain and the architecture elements.

3.2.2.1 Media value chain

The media value chain represents the life cycle of multimedia content since it is conceived until it is consumed. Inside this value chain different actors may be involved, representing different business models and steps into the evolution of content. In summary, it is a matter of transferring some content from a source to its destination, using a distribution channel and defining the different features applicable to the content. To describe the different participants and elements, MXM defines some abstract elements that represent the transaction, which include:

- ***Content sources.*** MXM provides content creators (content sources) with the mechanisms to represent audio, still image video or graphics in the form of structured digital items. Once content is structured, it is possible to register content thanks to the protocols defined to communicate with content registration agencies or to store content in remote repositories using existing delivery methods. It is also possible for creators to define terms and conditions of access, trade and manipulation of its content. Inside the content creation phase, the following roles are considered: creators of original works (authors), adapters who derive new works from existing ones and interpreters who perform different versions of a work.
- ***Content handlers.*** Between content sources and content receivers, several intermediaries can be found, like content distributors, retailers, publishers, Collective Management Societies, etc. MXM provides them with the means to exchange, adapt, share, trade, etc. digital goods among themselves or with the end user in a controlled way. Both licensing and adaptation mechanisms can be expressed in a standardised manner.
- ***Content receivers.*** MXM provides the mechanisms to implement application and services on top of its middleware to facilitate content consumption to end users.
- ***The message.*** The format used in the MXM value chain is the Digital Item, which was defined in MPEG-21, the multimedia framework.

- ***The channel.*** Several transmission channels are supported in MXM, including the way to secure them. Communication between two devices is established after mutual authentication.

3.2.2.2 MXM architecture elements

The MXM standard is organised in four public documents, describing the architecture, the APIs, the reference software and the protocols:

– MXM architecture and technologies [ISO11a], which describes the whole design, including the MXM Device that is the platform able to run MXM applications. Applications may run on top of the operating system or use MXM Engines providing MXM technologies and protocols.

– MXM application programming interface [ISO11b], which describes the API provided by each MXM Engine. This API is defined in Java, C++ or both and does not impose any implementation of API operations.

– MXM conformance and reference software [ISO11c], which provides a sample model implementation and the way for

– MXM protocols [ISO10a], which are the XML messages exchanged between MXM Engines and delivered through Web Services.

The MXM standard is mainly concerned with the interfaces of the MXM Engines and the interface to a master Engine called Orchestrator Engine. The relationship between the different components is shown in Figure 19.

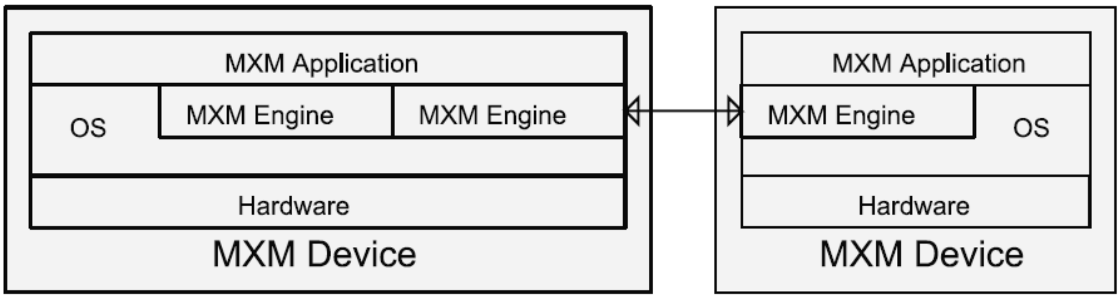


Figure 19. MXM Components model.

The MXM standard defines 18 engines, which are briefly described below:

- Content Protocol Engine: Provides procedures to identify a content item from a content identification device, to access it and store it from/to a content provider device and to authenticate it completely or in parts.
- Content Search Engine: Implements the MPEG Query Format [ISO08a] methods to access content repositories (the content provider device).
- Digital Item Adaptation (DIA) Engine: Provides the methods to parse, access and create information contained in usage environment description elements (context information needed to perform content adaptation) [ISO07a].
- Digital Item Engine: Provides the functionality to create, access and edit digital items [ISO05a].
- Digital Item Streaming (DIS) Engine: Provides operations for the serialisation and access of digital item as streams, as described in [ISO07b].

- Domain Engine: Implements the methods to exchange information with the domain management device.
- Event Reporting (ER) Engine: Tracks every relevant action in the system, as well as to create Event Report Request and Event Reports, as defined in [ISO06b].
- Intellectual Property Management and Protection (IPMP) Engine: Defines classes to create and access IPMP data structures [ISO06a], which describes the mechanisms and tools for the protection of digital items.
- IPMP Tool Protocol Engine: Provides the methods to access an IPMP Tool Body, i.e., how to access an implementation of protection algorithms.
- License Protocol Engine: Declares the methods to access store and revoke licenses remotely in a license provider device.
- Media Framework Engine: Supports video, audio, image and 3D graphics handling. It provides methods to create (encode) and access (decode) the elementary streams.
- Metadata Engine: Provides functionality to create and access audio, still image, 3D graphics and video metadata.
- MPEG-21 File Engine: Provides the functionality to manage MPEG-21 File format files [ISO05b], including creation and access.
- Media Value Chain Ontology (MVCO) Engine: Provides the methods to access the value chain ontology and manage Digital Items and users in conformance with the IP value chain model.
- Orchestrator Engine: It is a special engine capable of invoking predefined sequences of calls in the others engines and capable of providing a simpler, unified interface to a priori known application domains.
- Rights Expression Language (REL) Engine: Defines the methods for creating and accessing licenses and authorising users to perform operations over content based on the licenses they own [ISO04a].
- Security Engine: Provides cryptographic algorithms, digital signature and the tools to achieve trust in devices. Note that devices interoperate with each other after having mutually authenticated themselves, by exchanging certificates provided by the Certification Authority.
- Rendering Engine: Provides the access to hardware graphics acceleration, input device handling and its abstraction.

3.2.3 Overview of Multimedia service platform technologies (MSPT)

As already mentioned, MPEG-M defines a suite of standards that are being developed for enabling the easy design and implementation of media-handling value chains. The main aim behind the different parts of this standard is to provide a set of middleware Application Programming Interfaces (API), elementary service protocols and formats and service aggregation mechanisms to facilitate the creation of content management and distribution systems that can be interoperable between them. The definition of common API's, protocols and interfaces will permit the integration and use of services implemented by different organisations.

This edition of the MPEG-M standard comprises five public documents, which include Architecture, API, Reference Software, Elementary Services and Service Aggregation. These documents are close to the final steps of the standardisation process.

In its second edition, MPEG-M is referred as Multimedia Service Platform Technologies (MSPT), and it maintains the architecture and design philosophy of the first edition, but stressing its Service Oriented Architecture (SOA) character. SOA has been specially applied to parts 4 [ISO13b] and 5 [ISO13c] of the standard. Part 4 defines elementary services providing basic functionality for implementing any content distribution scenario and part 5 describes how to aggregate services. The services to aggregate could be those present in part 4. New elementary services, that is, those not present in part 4, should be described following the guidelines in part 4 and registered with the mechanism defined in part 5.

More specifically, the second edition of MPEG-M is subdivided into the following five parts, most of them close to the final steps of the standardisation process:

- Part 1 - Architecture: Specifies the architecture that is part of an MPEG-M implementation;
- Part 2 - MPEG Extensible Middleware (MXM) Application Programming Interface (API): specifies the middleware APIs;
- Part 3 - Conformance and Reference Software: specifies conformance tests and the software implementation of the standard;
- Part 4 - Elementary Services: specifies elementary service protocols between MPEG-M applications;
- Part 5 - Service Aggregation: specifies mechanisms enabling the combination of elementary services and other services to build aggregated services.

3.2.3.1 MPEG-M Part 1

Part 1 describes the MPEG-M architecture, its elements and Application Programming Interfaces (APIs) that enable MPEG-M compliant devices to be interoperable even if different manufacturers implement them. An MPEG-M device is a device equipped with MPEG-M engines. It can have several MPEG-M applications running on it such as an audiovisual player or a content creator combining audio-visual resources with metadata and rights information.

The elements of the MPEG-M architecture are MPEG-M engines, MPEG-M engine APIs, MPEG-M orchestrator engine, MPEG-M orchestrator engine API, MPEG-M device, and MPEG-M application. *MPEG-M engines* are collections of specific technologies that can be meaningfully bundled together; the *MPEG-M engine APIs* can be used to access functionalities of MPEG-M engines; an *MPEG-M orchestrator engine* is a special MPEG-M engine capable of creating chains of MPEG-M engines to execute a high-level application call; the *MPEG-M orchestrator engine API* can be used to access the MPEG-M orchestrator engine; an *MPEG-M device* is a device equipped with MPEG-M engines; and an *MPEG-M application* is an application that runs on an MPEG-M device and makes calls to the MPEG-M engine APIs and the MPEG-M orchestrator engine API.

Figure 20 shows the general architecture of a MPEG-M device, where MPEG-M applications running on an MPEG-M device could call, via an application-middleware API, the Technology

Engines (TEs) in the middleware to access local functionality modules, and the Protocol Engines (PEs) to communicate with applications running on other devices by executing elementary or aggregated service protocols among them. The role of the orchestrator engine is to set up a more complex chain of TEs and PEs.

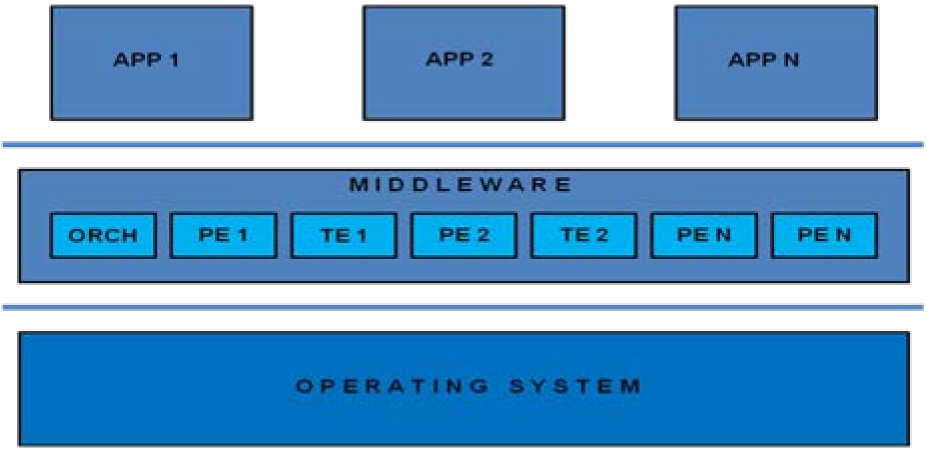


Figure 20. MPEG-M device architecture; the middleware is populated by Technology Engines (TEs), Protocol Engines (PEs) and one or more Orchestration Engines (ORCH).

3.2.3.2 MPEG-M Part 2

Part 2 specifies a set of APIs, which are the gateway to the MPEG-M middleware – providing access to its technology engines as specified in Part 1 – for any application running on an MPEG-M device.

Conceptually, these APIs are divided in four categories, namely creation APIs, editing APIs, access APIs and engine-specific APIs. *Creation APIs* are used to create data structures, files and elementary streams conforming to the respective standards; *Editing APIs* are used to modify an existing data structure, file, elementary stream in order to obtain a derived object still conforming to the respective standard; *Access APIs* are used to parse data structures, files, decode elementary streams in order to retrieve the information contained within; and *Engine-specific APIs* are those that do not fall into the above categories, such as APIs for license authorisation and content rendering.

Furthermore, Part 2 of the standard contains the description and the API specification of MPEG-M engines, which are classified into three types: a) Protocol Engines (PEs), b) Technology Engines (TEs), and c) Orchestrator Engines.

Protocol Engines instantiate the communication protocol of the elementary services, so they have a one-to-one relationship. PE APIs have been designed in a unified way, providing interfaces for creating and parsing protocol requests and responses, as specified in MPEG-M part 4, as well as for performing the requests and receiving the responses.

Technology Engines are responsible for carrying out the actual operation of an elementary service and are organised in terms of schema and technology handlers. The schema handler is mainly used for managing the schemata dictated by the standards used for implementing the corresponding technology. The technology handler is responsible for exposing the API that allows handling of the underlying technology.

TE’s supported by the standard are briefly described next:

- Contract Expression Language (CEL) Engine: Defines the methods for handling CEL [ISO13d] expressions and providing functionalities, such as: creation of contract expressions and accessing data contained in them.
- Digital Item Engine: Defines the APIs for handling digital item data structures and providing functionalities, such as: creation of digital items and data retrieval from them, as well as management of elements Item, Statement, Descriptor, Component, Resource, License, Metadata and Event Report Requests inside digital items.
- Event Reporting Engine: Defines the methods for operating over event reporting data structures.
- Intellectual Property Management and Protection (IPMP) Engine: Defines the methods for operating over IPMP data structures and providing functionalities, such as: creation of IPMP data structures and accessing data contained in them.
- Media Framework Engine: It is a high level MXM Engine, grouping together several media specific engines, such as: Video, Image, Audio and Graphics Engines. It implements common functionalities (independent on the media type) such as resource loading and saving.
- Metadata Engine: Defines the methods for handling creation and management of metadata structures.
- MPEG-21 File Format Engine: Defines the methods for operating over MPEG-21 file format files and providing functionalities, such as: creation of MPEG-21 files and accessing data contained in them.
- Overlay Engine: Specifies a minimum set of interfaces that should be implemented by any device participating in a content delivery network (CDN). Emphasis is given in peer-to-peer networks that are quite popular with users for content discovery.
- Rights Expression Language (REL) Engine: Defines the methods for handling rights expressions and providing functionalities, such as: creation of rights expressions, accessing data contained in them and users' authorisation to exercise rights.
- Security Engine: Defines security-related methods providing functionalities, such as: creation of new credentials and management of public-key based certificates; generation of symmetric keys and encryption/decryption of data; storing of confidential information such as licenses and keys in the secure repository; certification of tools integrity; and, enabling complex authentication protocols.
- Search Engine: Defines the methods for operating over metadata structures and providing functionalities, such as: creation and parsing of MPQF query structures.

Finally, orchestrator engines manage the execution flow of technology engines in the context of an elementary service, an aggregated service or an application.

3.2.3.3 MPEG-M Part 3

Part 3 is about the conformance and reference software. The APIs and the Elementary Services are given in MPEG-M Part 2 and MPEG-M Part 4, respectively.

This part of the standard mainly describes the software repository structure, from where the conformance and reference software can be obtained to implement new services on top of this structure.

3.2.3.4 MPEG-M Part 4

Part 4 specifies Elementary Services (ES) and their protocols. They are the key elements in achieving services interoperability in the MPEG-M ecosystem.

The standard defines different types of ES: regular ESs, when specific operations are performed over a specific kind of entity; generic ESs, with specified entity but generic operations and abstract ESs with specified operation but generic entities. Table 2 shows the Elementary Services defined in the standard, together with the entities to which they apply.

Table 2. Elementary Services classified by Operations and Entities

	Content	Contract	Device	Event	License	Service	User
Authenticate	X	X					X
Authorize							X
Check With		X			X		
Create	X	X			X		
Deliver	X	X					
Describe	X		X			X	X
Identify	X	X	X		X		X
Negotiate		X			X		
Package	X						
Post	X						
Present		X			X		
Process	X				X		
Request	X	X	X	X	X		
Revoke	X	X			X		
Search	X	X	X		X	X	X
Store	X	X		X	X		
Transact	X				X		
Verify		X	X		X		

3.2.3.5 MPEG-M Part 5

Part 5 specifies how ESs and perhaps other existing Aggregated Services (ASs) should be combined to build new ASs. To do so, it provides a methodology which defines the basic steps for the definition of ASs.

The methodology for the ASs definition comprises the following steps:

1. Provision of a narrative description of the actions that a new AS would perform; in other words, the use case or scenario to be implemented as AS.

2. Identification of ESs and ASs that are needed by a new AS in order to be implemented. These ESs and ASs could be classified as those: a) described in Part 4 and Part 5, respectively; b) registered with the corresponding RA; and, c) external ESs specifically required by the new AS. The external ESs could also be registered with the RA for further use by third parties.
3. Provision of a textual description of the AS workflow describing the interactions between the Client and Service Provider.
4. Resulting AS service workflow formal description provision. It should describe both protocol and service by including the service workflow graphical representation and optional its XML serialisation.
5. Optional registration of the resulting AS with the RA. The RA syntactically validates each registered AS.

As the nature of ASs can be very varied, the standard gives several examples of ASs definition, showing how each of them follows the steps of the methodology to describe service aggregation.

3.2.3.6 MPEG-M summary

The combination of the different parts of MPEG-M standard, especially parts 4 and 5, covers the content management and distribution scenarios that will be described in the Contribution section. For instance, DRM-enabled content access control scenario would make use of the Authenticate User, Authorize User, Create License, Store License, Search Content and Store Event Elementary Services. With the aggregation of these services it is possible to define the buyer point of view of this scenario, as described in part 5 of the standard [ISO13c]. It is worth noting that any of the described scenarios have a buyer (content consumer) and seller (content producer or distributor) use cases. In this sense, the Elementary Services needed to implement each of them will be different although the aggregation will be always done in a similar way. For a complete description of this standard see [KUD13a].

Part III. Contribution

4 Architectures for RELs Interoperability

4.1 Introduction

One of the main limitations of existing DRM systems is the lack of interoperability, which force users to be tied to specific rendering devices, those of the system which provided the content.

Different initiatives have been conducted to provide interoperability to DRM systems, as CORAL [KAL07a] or Marlin [MAR13a]. However, none of them provide a global solution to enable different devices to render content governed by different DRM systems. The DMAG research group [DMA13a] has 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. In this PhD Thesis work the focus is on the interoperability between Rights Expressions Languages (RELs). Before, some work had already been done on this topic at the DMAG. Initially, the tasks were concentrated in the syntactic interoperability between rights expression languages, focusing on two standard initiatives, MPEG-21 REL and ODRL, which were competing to have a place in the market. In [POL04a] 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. In this approach, Extensible Style Sheets Transformations (XSLT) [W3C13a] were used to translate from licenses expressed in one rights expression language to another.

Meanwhile, the Open Mobile Alliance (OMA) [OMA13a] 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 [DEL05a].

This contribution chapter describes several attempts to provide interoperability between RELs using different techniques. The first one deals with two translation mechanisms based on both programmatic and database approaches. In it, high-level models in Unified Modelling Language (UML) [OMG12a] and Entity Relationship (ER) [CHE76a] are defined to perform the translation. The second attempt tries to overcome the limitations of the previous ones, proposing a solution based on the use of a XACML system based architecture to achieve interoperability between RELs. The feasibility of the proposed solution is demonstrated 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, a direct map to XACML is done in order to perform the authorisation with it. In this way, information conveyed in the input REL is not lost. Finally, a third attempt based on a rights management broker solution is presented, together with some application scenarios.

4.2 UML- and ER-based rights expression languages interoperability

The work described in this sub-section shows how to implement rights expression languages (RELs) interoperability using a programmatic approach based on two well-known high-level modelling schemas, Unified Modelling Language (UML) [OMG12a] and Entity-Relationship (ER) [CHE76a].

The objective was to translate between them rights expressions formalised using MPEG-21 Rights Expression Language (MPEG-21 REL) [ISO04a] and Open Mobile Alliance [OMA13a] Digital Rights Management Rights Expression Language (OMA DRM REL) [OMA06a]. This modelling and implementation was done inside the AXMEDIS project [AXM04a] as it was needed to transform rights expressions described in MPEG-21 REL to use them in mobile devices supporting OMA DRM REL [DEL06a].

To describe the translation process, the different modelling implemented for these RELs by means of UML and ER diagrams are presented. Then, two different approaches for translating licenses are shown.

The main limitation of this attempt to provide interoperability between different RELs is that only the common rights and conditions defined in both RELs can be taken into account. Even though, it is not worth adding the new terms, since the license based authorisers compliant with a REL do not understand them, and thus when they find any of them in a license they authorise negatively as they usually work in the most restrictive way.

4.2.1 Expressing XML Rights Expressions using UML

To be able to translate rights expressions from language to language, the first work done was to express both languages in an abstract UML class diagram. The reason for creating these diagrams was that it was needed to manipulate rights expressions in a software development project. The XML format is very readable and human comprehensive but is not very functional when you need to interact with it. So, the solution was to create different class diagrams for each REL.

To represent each language, different approaches were selected. For MPEG-21 REL, a subset of the whole language is represented, as it was almost unmanageable with the programming tools available at that time. For OMA DRM REL, it was simpler, as it is a profile of ODRL 1.0 [W3C02a]. The details for each representation are presented next.

4.2.1.1 Expressing MPEG-21 REL in UML

To express MPEG-21 REL into a UML class diagram, a simple but scalable solution is used. The main aim was to simplify the translation from one REL to another, expressing as many conditions as required and also adding new conditions, if they appeared.

The model defined for MPEG-21 REL Licenses allowed the representation of any license expressed with it, just using an appropriate parser. With this model, the syntactic structure of the licenses is not restricted in any way. Nevertheless, it gives a clear advantage when searching licenses, as a common format for all of them is defined, focusing on the most important fields for searching purposes.

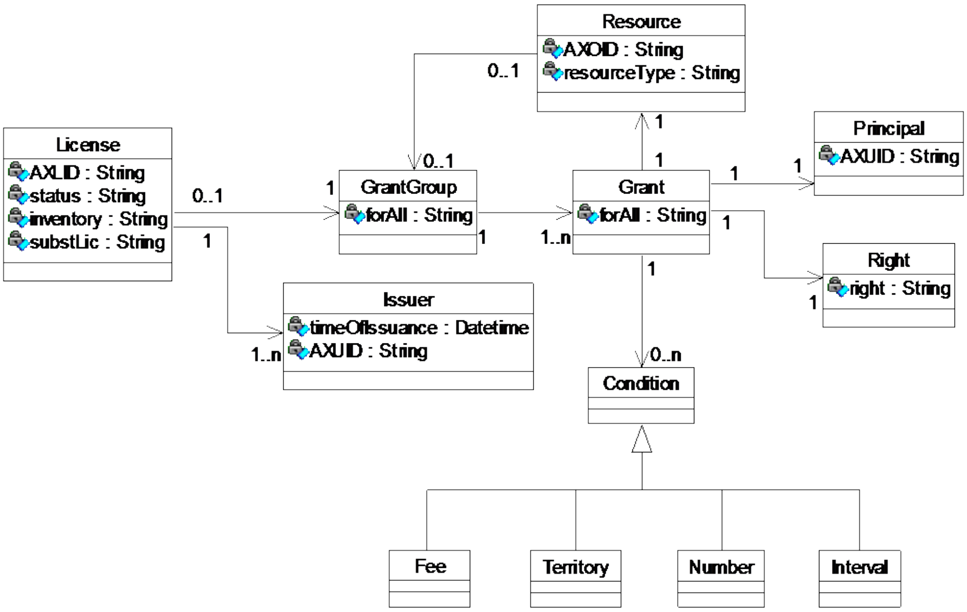


Figure 21 shows the UML class diagram proposed for representing the MPEG-21 REL licenses.

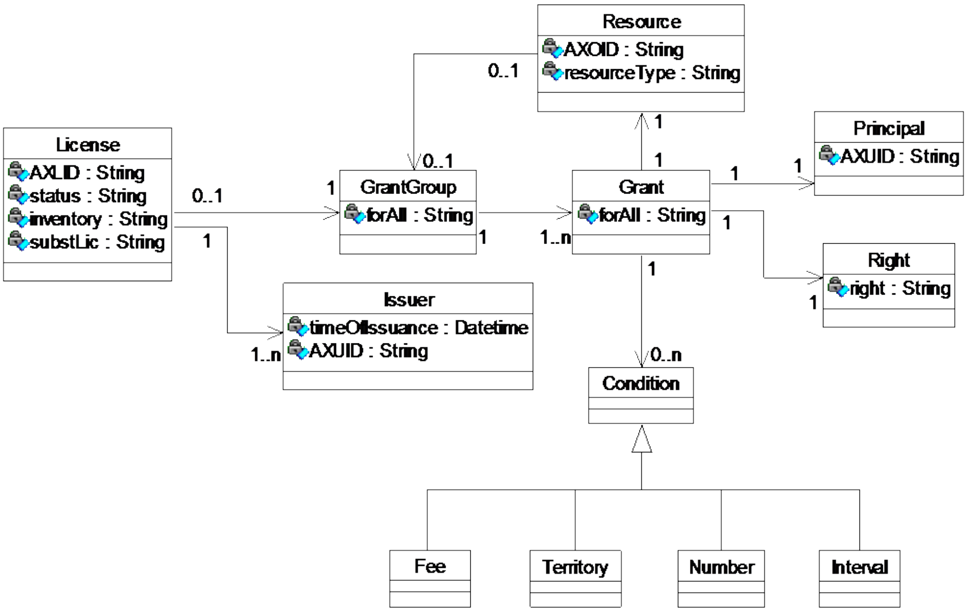


Figure 21. MPEG-21 REL UML Class Diagram.

Each license can have a LicID element that contains a unique identifier for the license, there is also an issuer element and a GrantGroup. A status element contains the status of the license, e.g. revoked, a substLic element that contains the license that replaces the revoked one and an inventory element (that contains the variables defined in the license that can be referenced within it).

Each GrantGroup contains a set of Grants and the forAll element where the variables or patterns are defined within this GrantGroup are placed.

Each Grant contains the information of the right granted, the resource, the principal and an optional set of conditions related to that right, and the forAll element where the variables or patterns defined within this grant are placed.

It is worth noting that a resource can be also a GrantGroup, to support Distribution Licenses.

4.2.1.2 OMA DRM REL

For OMA DRM REL licenses, a solution following the same approach as for MPEG-21 REL is defined. In this way, it is possible to cover all the elements that can be expressed with the OMA language, apart from the digital signature of the license present in the digest element, which is not supported.

Figure 22 describes the UML class diagram representing the OMA DRM REL licenses.

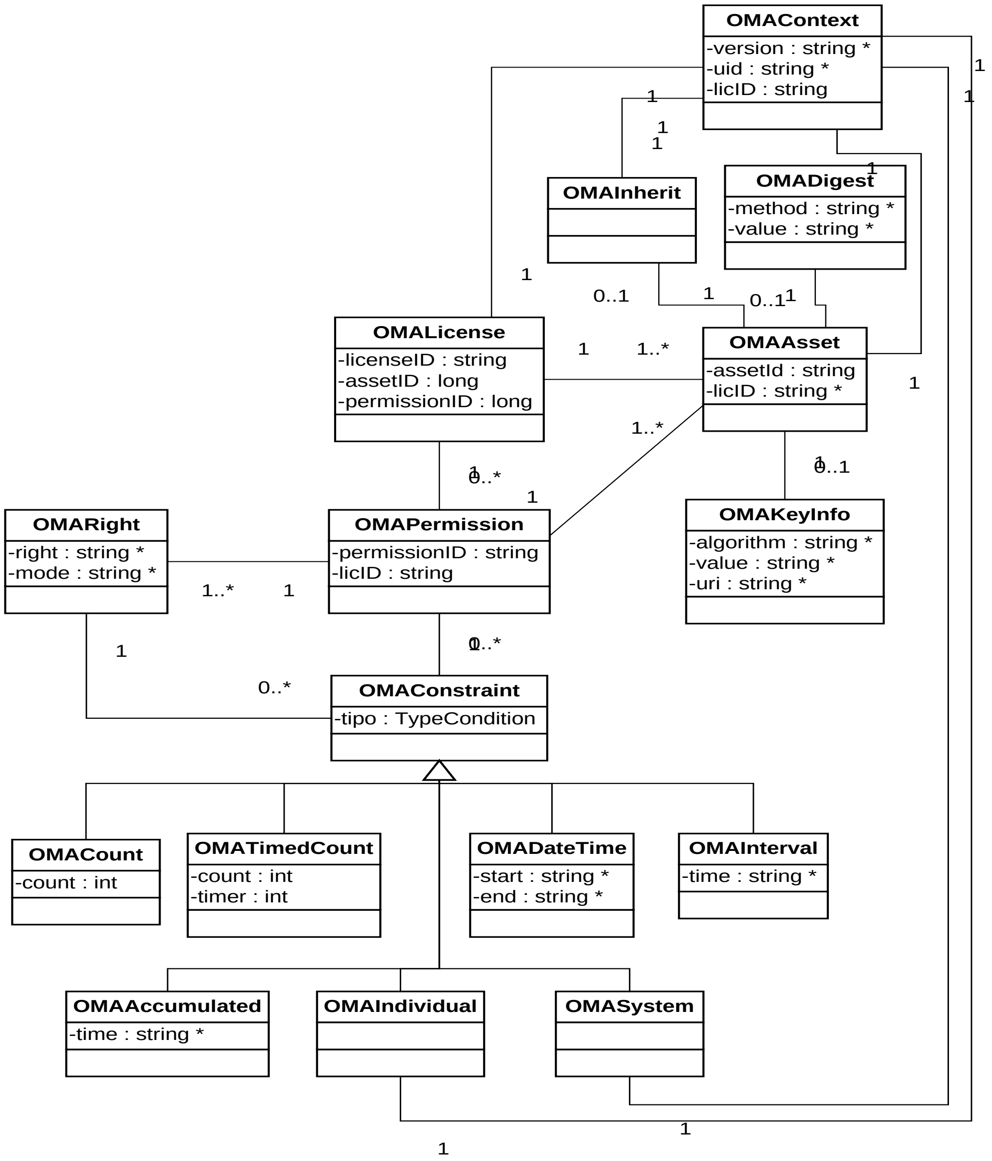


Figure 22. OMA DRM REL UML Class Diagram.

Each OMALicense is defined by a licenseID, that is an unique identifier for it, a OMAContext element, that expresses the uid and the version of the standard that have been used to create the license, and two more vectors containing the OMAAssets and OMAPermissions. Note that, opposite to MPEG-21, the OMAAssets (resources) and the OMAPermissions (constraints and rights) are stored separately because this is the way the license will represent it.

The OMAAsset stores all the information about the OMAContext (the real name and its version), the OMADigest information, the OMAInherit information, if the license derives from another one, and the OMAKey Information, which is represented by an algorithm and a value.

The OMAPermission is the relation of one or more assets with one or more rights. The OMAPermission can have a set of constraints that applies to all the rights, and each right can also have a set of constraints that only affects itself.

All the OMAConstraints that can be represented are defined. They are Count, Timed Count, Date Time, Interval, Accumulated, Individual and System.

The licenses represented follow version 2.0 of the OMA standard [OMA06a].

4.2.2 Expressing XML rights expressions using entity-relationship model

Persistency is not a problem in any of the right expression languages used, as they can be expressed using XML language and stored into the file system. Transforming from XML to the equivalent object model based on the class diagram, or vice versa, is not an expensive work. So, licenses could always be serialised in XML format.

The problem appears when you have to authorise (verify that someone can exercise a right over a digital resource) and you have a very large number of licenses. The authorisation process must perform non-trivial searches over the licenses, and it is not always possible to load all the available licenses in memory, like an object model. This is the main reason to create also an Entity Relationship Model to store licenses in a relational database. Then, the ER model designed can be transferred to a database, which can be set up for performing searches quickly and efficiently.

4.2.2.1 MPEG-21 REL ER model

The design of the ER model for MPEG-21 REL follows the principle of providing a simple but scalable solution for expressing licenses into ER diagrams.

In order to facilitate license representation into ER structure, the use of a common structure for licenses in order to simplify the parsing from the XML-based license into the ER structure has been imposed.

To represent the content of a license in an Entity-Relationship diagram, the focus is on the relations with a multiplicity 0..n. These relations show the number of different tables that are needed to store the represented information. The relations with a multiplicity of 1 – 1 can be always stored in the same table.

Figure 23 shows how to create the different tables to store the license information. This solution provides the model for storing both End-user and Distributor Licenses.

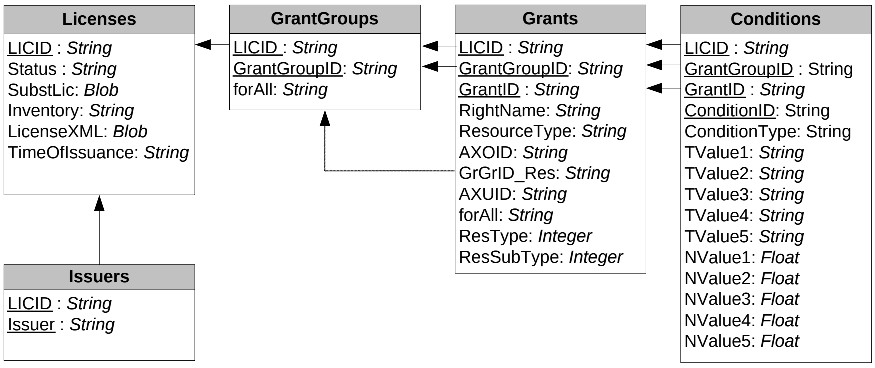


Figure 23. MPEG-21 REL ER Diagram.

For expressing the different types of conditions available, the following fields in the corresponding table have been defined:

- ConditionType: It indicates which kind of condition is being expressed.
- Five Tvalue fields and two NValue fields (more can be added if desired), whose values depend on the conditionType. TValue fields represent textual information and NValue fields represent numerical information.

Using this structure it is very easy to define new conditions and implement support classes in the corresponding programming language depending on the condition.

Complex queries over the defined ER diagram can also be made, only asking for different ConditionType, TvalueX and NvalueX.

Other possible approach for expressing conditions could have been to define a different table for each condition. This approach was discarded, as conditions expressed in MPEG-21 REL have a lot of different possible and optional values, and making queries over such a structure could be even more complex than using only one table or the original XML file.

4.2.2.2 OMA DRM REL ER model

The definition of the ER structure of the OMA DRM REL was very similar to the one proposed for MPEG-21 REL. The approach considered the multiplicity of the elements involving this kind of licenses, specially the assets and the permissions.

Figure 24 represents the structure that defines the solution for storing OMA DRM licenses in a scalable and efficient way.

As in the static model representation, the assets and the permissions have been separated into two different tables, as they do not have a 1 – 1 multiplicity. For this reason an extra table was added to store the relation between both elements.

For each asset only the values that are useful to do the authorisation process are stored. They involve its context and the Key Information.

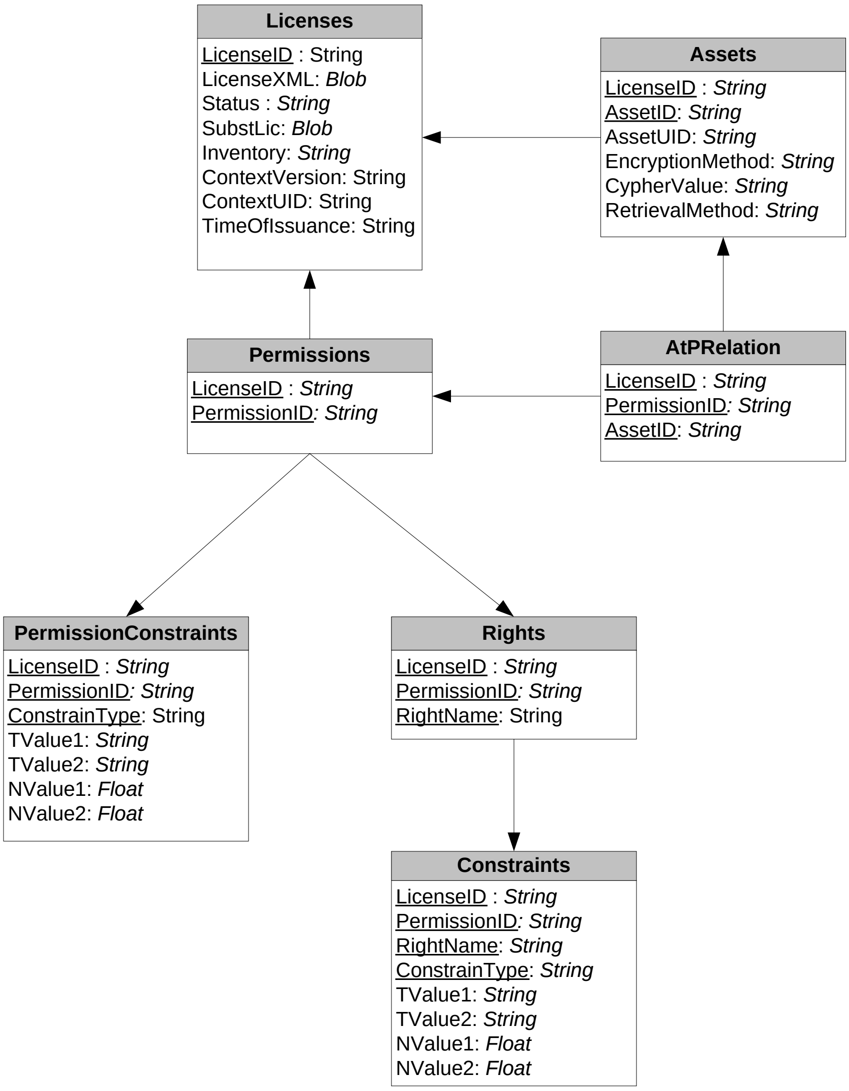


Figure 24. OMA DRM REL ER Diagram.

4.2.3 Translating rights expressions

The first approach to the transformation of licenses form MPEG-21 REL to OMA and the other way around was to use XSLT (Style Sheets) [W3C13a]. Using this technique the structure of the licenses being transformed is very limited and restricted [ROD05a]. To improve the quality of the translation, an approach based on profiling [DEL05a] is also proposed. In this case, the rights expressions are more or less equivalent. From the lessons learnt in both approaches, a different way was tried, taking advantage of the UML and the ER model to define two different alternatives to solve the problem. They are described in detail next.

4.2.3.1 Based on UML

OMA DRM REL and MPEG-21 REL express a license format; nevertheless the structure of these two kinds of licenses is quite different. So it is not possible to use a syntactic approach to perform the transformation but a semantic one based on the UML model.

The *License* class in the MPEG-21 diagram is equivalent to *OMALicense* class, and it is the object that holds the relationship of the model.

Another similarity is found at the *Condition* and *OMAConstraint* classes, which can be translated directly.

The critical point is that in MPEG-21 each *Grant* stores information about one right, one resource, one right holder and the conditions of this relationship. On the other hand, OMA DRM stores all that information in different classes with different relationships (*OMAAsset*, *OMAPermission* and *OMARight*).

In order to translate the OMA DRM REL to MPEG-21 REL model, a *Grant* for each relation that involves one *OMAAsset* with one *OMAPermission* and one *OMARight* has to be generated. Moreover, it has to be taken into account that a set of *OMAConstraints* can affect one specific *OMARight* or the *OMAPermission* (that involves all the rights related with it).

Note that exist some elements in both languages that cannot be exactly translated and which have to be transformed not in the syntactic way but semantic. For example, the TimedCount Constraint from OMA, or the adapt Right from MPEG-21.

If you want to translate from MPEG-21 REL to OMA DRM, you have to take care about the number of *OMAPermissions* that would be generated. To do that you have to group the *Grants* by the applied set of Conditions, in order to generate the minimum number of *OMAPermissions* and *OMARights*.

4.2.3.2 Based on a relational model

The other way to translate the licenses from one language to the other is to use the Relational model. In this case, SQL sentences like selects and inserts to perform the transformations can be used.

To transform from OMA DRM ER model to the MPEG-21 one the use two intermediate tables is required. They contain both the grants and the conditions table. These two intermediate tables can be fulfilled with a complex select that involves the join of Permissions, Assets, and Rights tables on OMA DRM ER model. And these two intermediate tables can be directly transformed to the Grants and Conditions tables of the MPEG-21 ER model.

To transform the information from the MPEG-21 ER model to the OMA DRM one, it cannot be granted that the resulting OMA license has the optimal structure. This is because every Grant in the MPEG-21 will be translated to a Permission with only one asset and one right.

In the case the obtaining of the optimal OMA license from the MPEG-21 one was required, it must be found which Grants could belong to the same Permission. These ones would have the same set of rights with the same set of conditions for every right for a set of resources. This cannot be done using only SQL sentences, and requires a quite complex algorithm. Because of that it is prefered to use the UML model to perform license transformations from license to license.

4.3 XACML-based architecture for REL interoperability

One of the objectives of this thesis is to demonstrate the feasibility to express license terms and conditions using XACML. With this goal in view, existing initiatives to use XACML for the representation of authorisation and entitlement policies have been analysed. More specifically, the OASIS eXtensible Access Control Markup Language (XACML) was carefully considered. This framework defines schemas for authorisation policies and for authorisation decision requests and responses. Moreover, it enables the description of complex access control rules. Another advantage of XACML is its extensibility, as new rights and conditions and implement their authorisation can be added. In this way, the solution is not limited to a subset of the RELs, but adding new rights is allowed, whose semantics will be taken into account, since its namespace will be considered when authorising, and regarding the new conditions its validation will be implemented in the XACML authoriser as they appear.

This section proposes the use of XACML 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. So, an architecture for the interoperability between different Rights Expression Languages (RELs), based on XACML [OAS05a] is presented. This implies the development of a DRM system based on XACML, so it is needed 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. In the chapter two particular cases are developed: the translation between the ODRL Rights Expression Language (REL) and the XACML policy language and the translation between the MPEG-21 Rights Expression Language (REL) and the XACML policy language.

The XACML standard, presented in detail in section 2.3, specifies a policy language model. The model defined in this standard consists of three main elements: rule, policy and policySet. The rule element contains a boolean expression that can be re-used in multiple policies. The policy element consists of rule elements, which in turn consists of a target element that defines the set of resources, subjects, actions and environments to which the rule is intended to apply; an effect that element indicates the intended consequence of a true evaluation for the rule; and a condition element that represents a Boolean expression that refines the applicability of the rule. The policySet element is defined for combining separate policies into a single combined policy.

On the other hand, licenses consist of two main elements: grants or permissions and issuers. The grant or permission element of a license conveys to an entity some rights subject certain conditions. A grant is formed by four elements: 1) a principal, that represents the entity involved in the granting or exercising of rights, 2) a right, that specifies the action or activity that a principal may perform, 3) a resource, that represents the object against which the principal has the rights, and 4) a condition element, that represents grammatical terms, conditions and obligations that a principal must satisfy. The issuer element contains two pieces of information, the identification of the issuer and a set of specific details about the issuance of the license.

Next, the solution proposed to enable licenses and policy interoperability is presented and it is demonstrated that licenses expressed according to a REL, such as the MPEG-21 REL or OMA DRM REL, can be also represented in an XACML policy.

Figure 25 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.

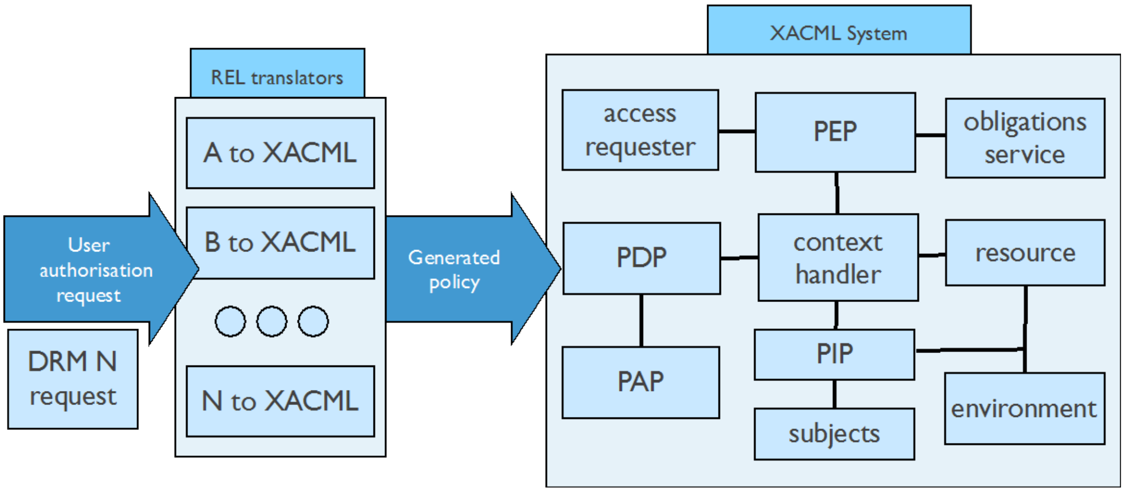


Figure 25. Proposed XACML architecture.

On the 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.

4.3.1 Use of XACML to provide interoperability between RELs

Once presented the desired architecture of the whole system, the focus is 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.

4.3.1.1 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 the second version is more similar. Figure 26 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.

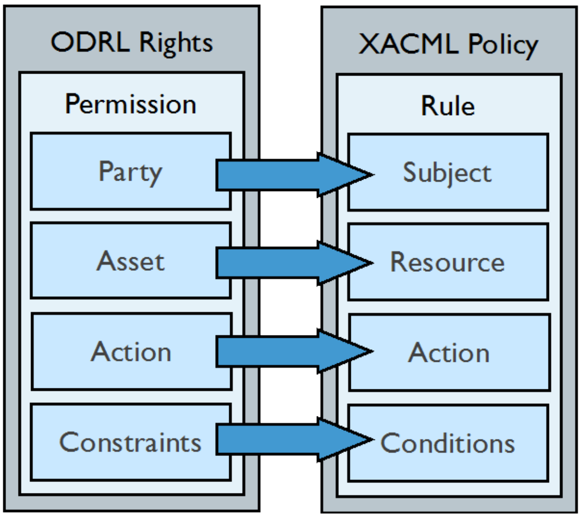


Figure 26. ODRL Rights vs XACML Policy.

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

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 3.

Table 3. Syntax for the rights and policy elements.

ODRL - rights	XACML - Policy

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

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 4.

Table 4. Syntax for the permission and rule elements.

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

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 5.

Table 5. Syntax for the party and subject elements.

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

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 6.

Table 6. Syntax for the action-related elements.

ODRL - play	XACML - Policy
<o-dd:play/>	<xacml:Policy> <xacml:Actions> <xacml:Action> <xacml:ActionMatch MatchId = "urn:oasis:names:tc:xacml:1.0:function:strin

	g-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>

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 7 shows an example of how the ODRL asset element can be translated into the corresponding XACML elements**.**

Table 7. Syntax for the asset and resource elements.

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

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 8 shows an example of the translation of the spatial constraint of ODRL to the corresponding elements in the XACML policy language.

Table 8. Syntax for the constraint and condition elements.

ODRL - constraint	XACML - Condition
<o-ex:constraint> <o-dd:spatial o-ex:type = ”prism:vocabs/ISO3166/ES”> </o-ex:constraint>	<xacml:Condition> <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>

4.3.1.2 MPEG-21 REL and the XACML Policy Language

The same work has been done for the MPEG-21 REL [ISO04a] in order to allow the expression of the syntax and semantics of MPEG-21 REL licenses within XACML policies. First, as for ODRL, the correspondence between the main elements of MPEG-21 REL licenses and XACML policies has been defined (see Figure 27).

The structure of a MPEG-21 license and a XACML policy is quite different but their elements can be mapped directly as shown in Figure 27. One example is the MPEG-21 REL Grant element, which has its equivalent XACML element, named Rule. What MPEG-21 REL defines as the Principal, the entity to which the rights described on the license are granted, can be expressed using the Subject element on a XACML policy. In both schemas, the object is described with the Resource element. The Right element of a license can be expressed in a policy using the Action element. Finally, both MPEG-21 REL and XACML policy schemas have the Condition element that describes the constraints that the entity that holds the rights has to accomplish.

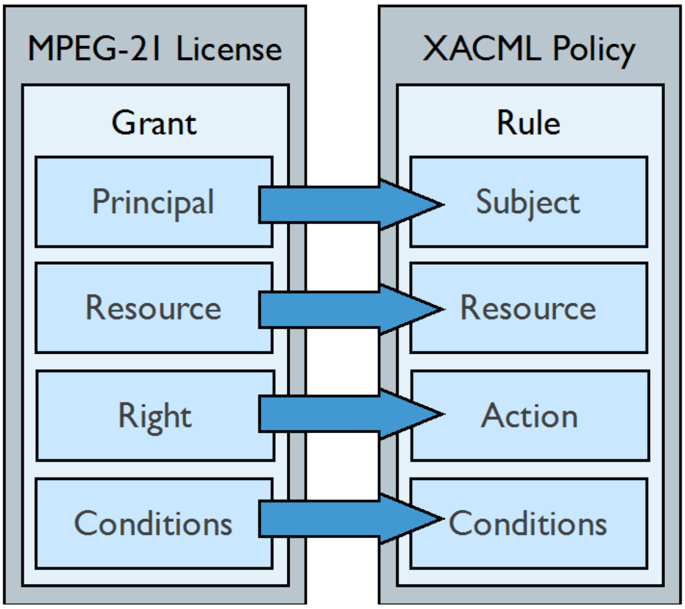


Figure 27. MPEG-21 REL License vs XACML Policy.

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). The XML elements are equivalent or, in some cases, the same in both languages, but this does not mean that the internal structures and elements are built on the same way.

The main difference between the Principal and the Subject elements is the way in which their contents are described in both schemas. In the MPEG-21 case, the content of the Principal element is a specific element, while in XACML it is an attribute of the Subject element which indicates the system the type of content that will describe the entity.

The same happens with the Resource element, for which has been defined some specific elements in the multimedia and standard extensions of the MPEG-21 REL, that are used in licenses. While XACML just defines an xsd:any type restricted by an attribute description of the data type.

For the Right element, MPEG-21 REL schema just defines a set of verbs with its namespace as an XML element itself, while in XACML it follows the same structure as the previous elements.

Finally, the Condition elements defined in both languages are the most different ones. On the one hand, MPEG-21 defines some XML elements that represent the different conditions supported by this standard (for example: the ValidityInterval, ExerciseLimit, etc.). Each of these elements has its own attributes that are different in number and type, and all of them are defined in the corresponding MPEG-21 REL schema (Core, Multimedia Extension, Standard Extension, etc.). The conditions are one of the most complex parts of an MPEG-21 REL license and, for that reason, any translator from or to MPEG-21 REL needs to know the meaning of each condition to be able to translate it and put it in the correct element. On the other hand, XACML has not defined any specific element for representing conditions. They are just represented by simple operations describing boolean functions. An illustrative example is the representation of the MPEG-21 REL ExerciseLimit condition in a XACML policy. The ExerciseLimit condition can be represented in a XACML using a specific XML element with one parameter that specifies an integer comparison between the number of times that the entity has exercised the right; and an integer number defined in the policy. This would generate a function like 'integer A = integer B'.

Tables from 9 to 14 contain relevant examples that illustrate the equivalences between MPEG-21 REL license parts and the corresponding XAMCL policy sections.

General definition

The first difference between both types of documents is the root element, which describes the digital contract. For MPEG-21 REL documents, the root element is the license element; while in XACML documents, the root element is the Policy element. Although the name, both XML elements describe the same concept and have the same attributes like a unique identifier for the document.

Table 9. Syntax for the license and policy elements.

MPEG-21 - license	XACML – Policy

`<r:license licenseId="1" […] >`
`[…]`
`</r:license>`
	`<xacml:Policy PolicyId="urn:oasis:names:tc:xacml:2.0:example:policyid:1" [..]>`
	`[…]`
	`</xacml:Policy>`

Grant

In both languages there is an element 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 element is called grant in the MPEG-21 REL language, and Rule in the XACML one. The elements contained by this one are described in the next subsections.

Table 10. Syntax for the grant and rule elements.

MPEG-21 - grant	XACML – Rule

`<r:grant>`
`[…]`
`</r:grant>`
	`<xacml:Rule>`
	`[…]`
	`</xacml:Rule>`

Subject

In both documents 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. In both languages, this element can represent either one entity or a group of entities. In MPEG-21, this element is the keyHolder, which can be represented by a generic identifier in the keyName field, but also by a digital certificate, or using any other XML representation of its personal information. In the XACML case, the schema defines the element Subjects that contains all the Subject instances. Every Subject element describes the same as the keyHolder in the MPEG-21 REL language. The main difference between both languages is that while MPEG-21 defines an XML element for every type of subject, XACML makes a match between two elements, one for describing the operation for the matching function, and the other one with the subject value itself.

Table 11 presents an example of a subject defined by a String. In the MPEG-21 side, the identifier is in the keyHolder element, while in the XACML side the SubjectMatch element defines a ‘string-equal’ function for the subject attribute.

Table 11. Syntax for the principal and subject elements.

MPEG-21 - principal	XACML – Subject

`<r:keyHolder>`
`<r:info>`
`<dsig:KeyName>subjectId</dsig:KeyName>`
`</r:info>`
`</r:keyHolder>`
	`<xacml:Subjects>`
	`<xacml:Subject>`
	`<xacml:SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">`
	`<xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">subjectId</xacml:AttributeValue>`
	`<xacml:SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id" DataType="http://www.w3.org/2001/XMLSchema#string"/>`
	`</xacml:SubjectMatch>`
	`</xacml:Subject>`
	`</xacml:Subjects>`

Action

MPEG-21 REL defines a list of rights to be used in any license. Each one of these rights has a specific meaning defined in the MPEG-21 RDD. Otherwise, XACML defines the ActionMatch element, which is a placeholder for any right defined by the rights holder (creator, distributor, etc.).

In the example of Table 12, the MPEG-21 specific right play (defined in the Multimedia Extension of the standard), is mapped as a simple ‘string-equal’ matching function in the XACML side.

Table 12. Syntax for the right and action elements.

MPEG-21 - right	XACML – Action

`<mx:play/>`	`<xacml:Actions>`
	`<xacml:Action>`
	`<xacml:ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">`
	`<xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">play</xacml:AttributeValue>`
	`<xacml:ActionAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="urn:oasis:names:tc:xacml:1.0:resource:xpath"/>`

	</xacml:ActionMatch> </xacml:Action> </xacml:Actions>

Resource

The resource element is defined in both languages and it follows the same structure of the subject element. On one hand, the MPEG-21 standard defines a set of elements for representing different types of resources. In the example of Table 13, the license has the diReference element that contains an identifier to a MPEG-21 Digital Item declaration. Om the other hand, XACML does not define specific types of resources, it only defines the ResourceMatch element in which any type of resource can be placed, as well as its required matching function.

Table 13. Syntax for the right and action elements.

MPEG-21 - resource	XACML – Resource
<mx:diReference> <mx:identifier>resourceId</mx:identifier> </mx:diReference>	<xacml:Resources> <xacml:Resource> <xacml:ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal"> <xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">resourceId</xacml:AttributeValue> <xacml:ResourceAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="urn:oasis:names:tc:xacml:1.0:resource:xpath"/> </xacml:ResourceMatch> </xacml:Resource> </xacml:Resources>

Conditions

The last mandatory element in a license grant is the Condition element. As happens with the other elements, the main difference between the MPEG-21 and the XACML version is that the first one has a limited set of predefined elements specified in the standard documentation, while in the second one it is just open. On one hand, MPEG-21 defines some elements like: territory, exerciseLimit, fee or validityInterval. These elements have its specific meaning and syntax. On the other hand, XACML defines a simple Condition element with the typical matching functions in it.

Table 14 presents an example that illustrates how to express the territory condition of a MPEG-21 license in an equivalent XACML policy. In this case, MPEG-21 defines two more XML elements, the country and the region, with its specific meaning. In the corresponding XACML policy, it has been transformed to a simple string comparison between both attribute values.

Table 14. Syntax for the condition elements.

MPEG-21 - allConditions	XACML – Condition
<r:allConditions> <sx:territory> <sx:location> <sx:country xmlns:iso="urn:mpeg:mpeg21:2003:01-REL-SX-NS:country">Country</sx:country> <sx:region xmlns:iso="urn:mpeg:mpeg21:2003:01-REL-SX-NS:region">Region</sx:region> </sx:location> </sx:territory> </r:allConditions>	<xacml:Condition> <xacml:Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and"> <xacml:Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"> <xacml:Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"> <xacml:AttributeSelector DataType="http://www.w3.org/2001/XMLSchema#string" RequestContextPath="//xacml-context:Resource/xacml-context:ResourceContent/location/country"/> </xacml:Apply> <xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Country</xacml:AttributeValue> </xacml:Apply> <xacml:Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"> <xacml:Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"> <xacml:AttributeSelector DataType="http://www.w3.org/2001/XMLSchema#string" RequestContextPath="//xacml-context:Resource/xacml-context:ResourceContent/location/region"/> </xacml:Apply> <xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Region</xacml:AttributeValue> </xacml:Apply> </xacml:Apply> </xacml:Condition>

4.4 Broker-based architecture for DRM systems interoperability

The work presented in this Thesis has been done in the context of the definition of architectures for DRM systems and Access Control Frameworks interoperability.

Digital objects are managed in a controlled way by both, DRM systems and Access Control Frameworks. Then, a global interoperability solution, that enables users' transparent access and usage of digital media, requires interoperability between them. This solution will imply interoperability between the different elements of these systems, mainly for digital objects formats, rights expressions, policies and protection information and tools. To this end, an Interoperability Broker (iRMBroker) has been defined. The Broker consists of different modules that provide interoperability between digital rights and access control rules, which is one of the objectives of this thesis, between digital objects and between protection information. Figure 28 shows the architecture of the iRMBroker.

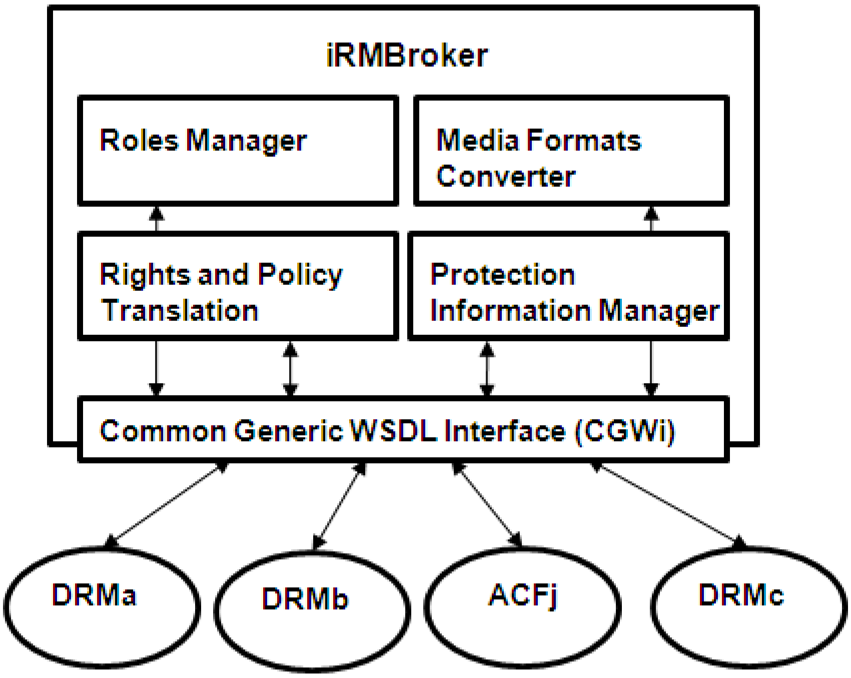

iRMBroker
Roles Manager
Media Formats Converter
Rights and Policy Translation
Protection Information Manager
Common Generic WSDL Interface (CGWi)
DRMa
DRMb
ACFj
DRMc

Figure 28. iRMBroker architecture.

Rights management and Access control Frameworks use different formats of digital objects. For example, OMA DRM uses DCF [OMA04a], while MIPAMS uses MPEG-21 DIDL [ISO05a]. The role of the iRMBroker Digital Media Formats Converter is to convert from one specific format to another. When a DRM systems or Access Control Framework does not know how to handle a specific format it contacts the iRMBroker and requests to the iRMBroker to convert that specific content to a friendly format. Then, the iRMBroker verifies the system credentials and validates them, and package the raw content in a DRM friendly format, returning the converted digital object.

The Protection Information Manager module is in charge to provide interoperability between protection tools for the different DRM systems and AC Frameworks. It retrieves the protection information associated to the protected contents and converts this specific information to the format of the requesting system. In this way it enables to handle protected contents at any level of granularity from a complete media object to specific resources or regions within digital contents.

The Roles Manager module manages the roles of users of the DRM and AC architectures. In all these architectures, authentication of users involves the identification of users and assignment of roles to them. The iRMBroker validates the provided tokens and assign roles to users according the content they are trying to access or use.

Finally, the Digital Rights and Policy translator converts rights expressions and policies to the format required by the requesting system. In the next subsections two application scenarios that validate the approach are presented.

4.4.1 Application scenarios

The proposed solution has been validated in two different scenarios, video surveillance and virtual collaboration, in the context of the European Network of Excellence, VISNET II [VIS06a].

4.4.1.1 Video Surveillance

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 with iRMBroker 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 [VIS06a], 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 [AXM04a] 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 [OMA13a]. 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 29 sketches the architecture for the proposed scenario.

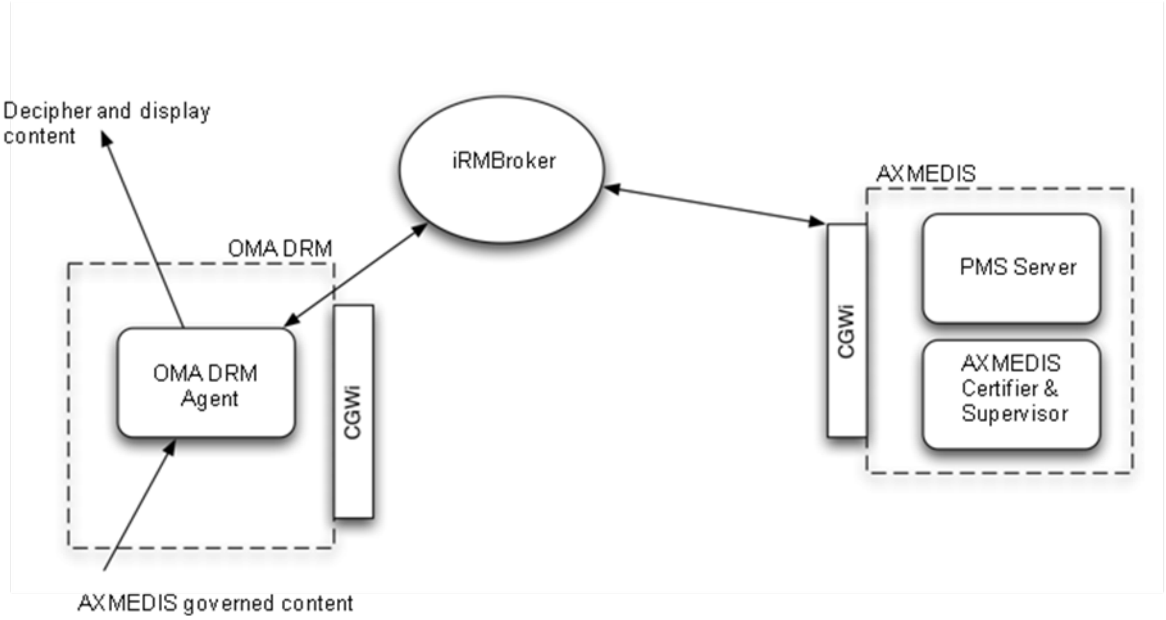


Figure 29. DRM Systems interoperability – Video Surveillance scenario.

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 authorisation 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 30):

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.

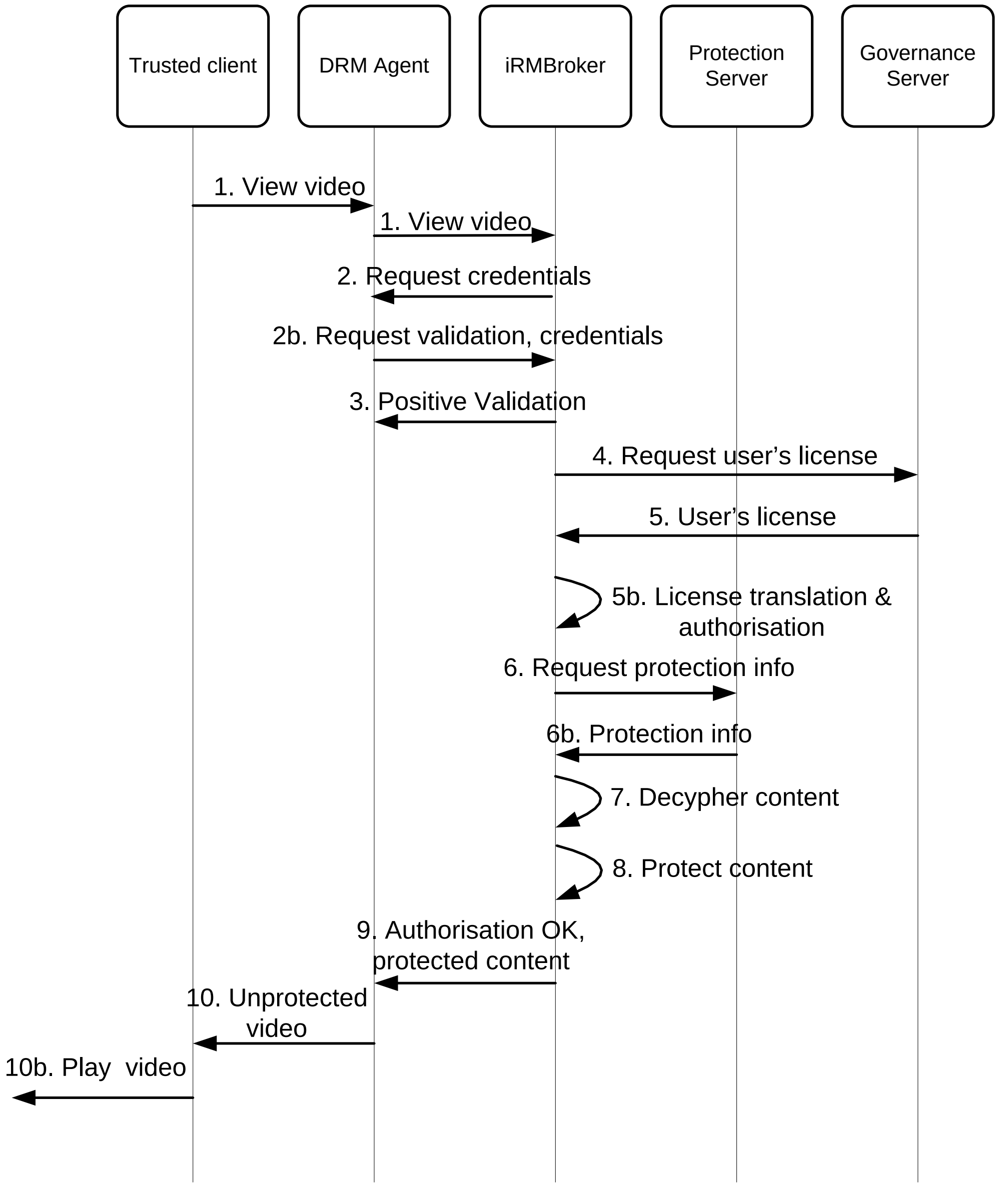


Figure 30. Video surveillance use case – positive authorisation.

4.4.1.2 Virtual Collaboration

The second application scenario in which was proved the applicability of the DRM systems and AC Frameworks interoperability solution was Virtual Collaboration. In this scenario, three different organisations, Aa, Bb and Cc, have set up a collaborative working environment using an Access Control and DRM architectures to allow them to work together on a project to design and produce a new widget. This collaborative working environment includes a shared data repository situated on Bb's network, which can store protected data, which can only be used according to the associated policies and licenses. The development of the widget is done in three stages. During the design stage a document is used to capture the specifications. This document is worked on jointly by designers in Aa and Bb over a number of weeks. At the end of this period, a final version of the document is ready for review by the managers of the project from all the organisations; this stage is called widget design review phase. After review, the document is finally released to allow the engineers in Bb and Cc to start work on producing a prototype widget. During the production stage, since it is considered highly commercially sensitive by the organisations involved, it has been decided that the engineers only will have permissions to read the document.

A walk-through of how the access control and DRM architectures are used in this scenario is given below. It is assumed that a small number of employees are assigned the project-specific role of Designer in organisations Aa and Bb. Only these employees are able to work on the joint design document. Each organisation has one nominated Project Manager, who is able to read the design document during its production and the review phase. Finally, organisations Bb and Cc have a small number of employees assigned the project-specific role of Engineer, which are responsible for producing the prototype. These Engineers should not be able to read the joint design document until it has been produced and reviewed.

In the scenario under consideration, several organisations, including ‘Aa’ and ‘Bb’, have set up a VO using the ACF and DRM architectures to allow them to work together on a project to design and produce a new widget. This VO includes a shared repository, which will be assumed is situated on Bb’s network, which stores protected data which is downloadable by anybody, but can only be used according to the associated policies or licenses. Aa and Bb architectures make use of different content protection technologies, therefore accessing data across the networks leads to interoperability issues. Example use cases are given below for the widget design phase, with associated sequence diagrams. These consider Alice (an Aa Designer) who wishes to retrieve, edit and then store her changes to the design document. For clarity, the interactions between Alice and the Network Access Servers on Aa and Bb are not shown.

The collaborative widget development scenario specified makes use of a combined access control and DRM architecture, as sketched in Figure 31 (“Obscure” is the name of one of the communicating platforms (the Access Control one); the other one (the DRM-based one) is based on MIPAMS). The following of this section provide definitions of use cases based on the 'collaborative widget development' scenario.

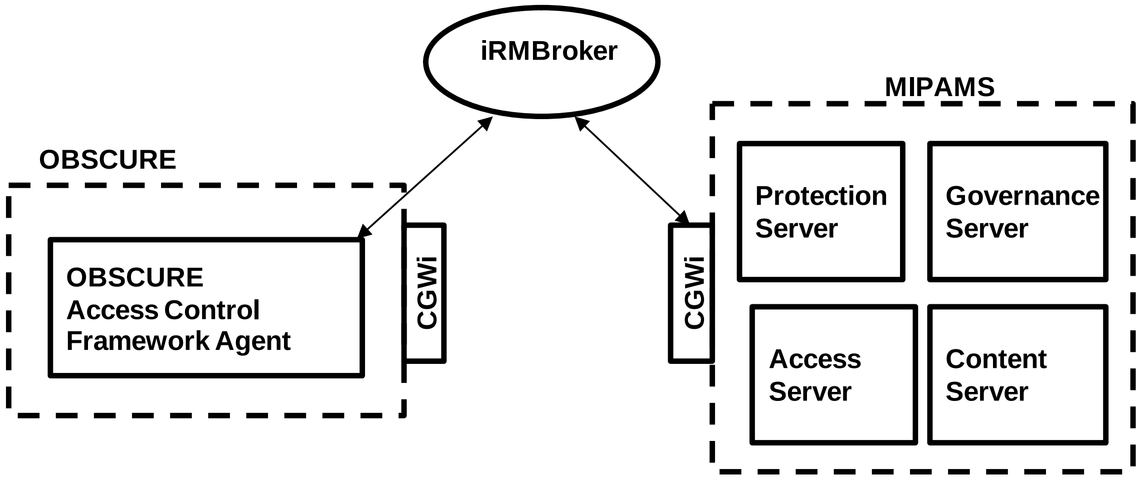


Figure 31. DRM and AC systems interoperability – Virtual Collaboration scenario.

Widget Design Phase

Use Case: Alice edits document

1. Alice authenticates herself to the Network Access Server on Aa and requests to activate her Designer role. The Network Access Server verifies she is allowed to do so, and returns a token containing this role.
2. Alice authenticates herself as a Designer to the Data Access Server on Aa using this token, and requests access to the design document. The Data Access Server retrieves the relevant license for the document and the Designer role, and checks that access is allowed. It then generates the decryption key and returns it to Alice.
3. Alice decrypts the document and edits it.

Figure 32 shows the sequence diagram associated with the 'Alice edits document' use case.

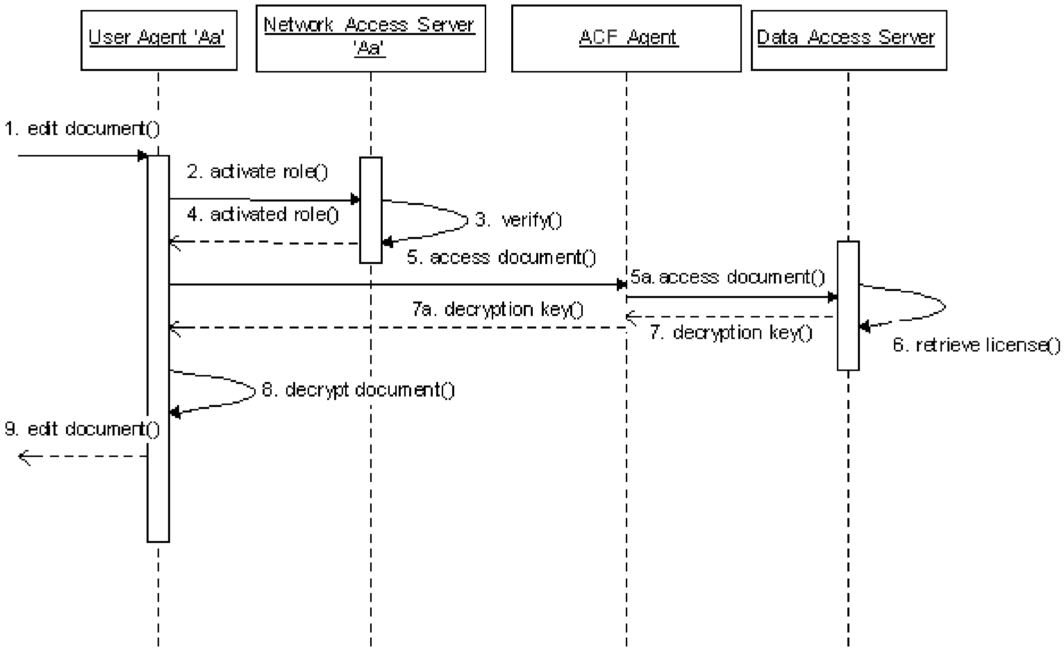


Figure 32. Use Case: Alice edits document.

Use Case: Alice stores document

1. Alice contacts the Protection Server to apply protection to a document.
2. Alice contacts the Governance Server to create the licenses for this document. Two role-based licenses, for Designers and Project Managers respectively, are created by the Governance Server in order to restrict access to these roles only. The Governance Server stores these licenses in a repository accessible by the Data Access Server.
3. Alice formats the protected data into a new Digital Item using Content Server software, and stores it back on the shared data repository.

Figure 33 shows the sequence diagram associated with the 'Alice stores document' use case.

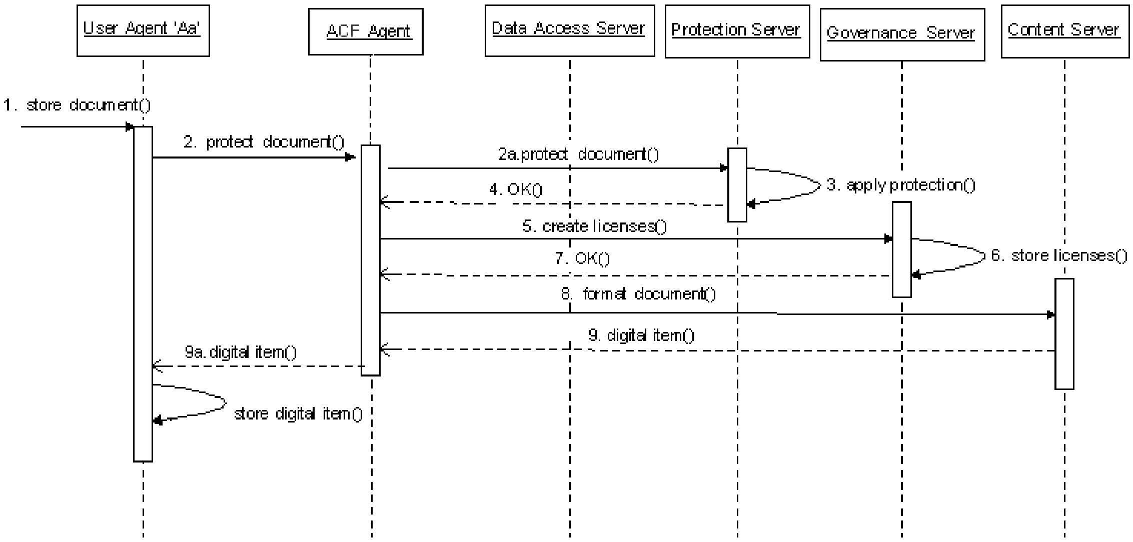


Figure 33. Use Case: Alice stores document.

Widget Design Review Phase

Use Case: Charlie edits document

1. Charlie authenticates himself to the Network Access Server on Bb and requests to activate his Project Manager role. The Network Access Server verifies he is allowed to do so, and returns a token containing this role.
2. Charlie authenticates himself as a Project Manager in Bb to the Network Access Server on Aa using this token. The Network Access Server maps the contained role to a Project Manager in Aa, and returns a token containing this role.
3. Charlie authenticates himself as a Project Manager to the Data Access Server on Aa using this token, and requests access to the design document. The Data Access Server retrieves the relevant license for the document and the Project Manager role, and checks that access is allowed. It then generates the decryption key and returns it to Charlie.
4. Charlie decrypts the document and edits it.

Figure 34 shows the sequence diagram associated with the 'Charlie edits document' use case.

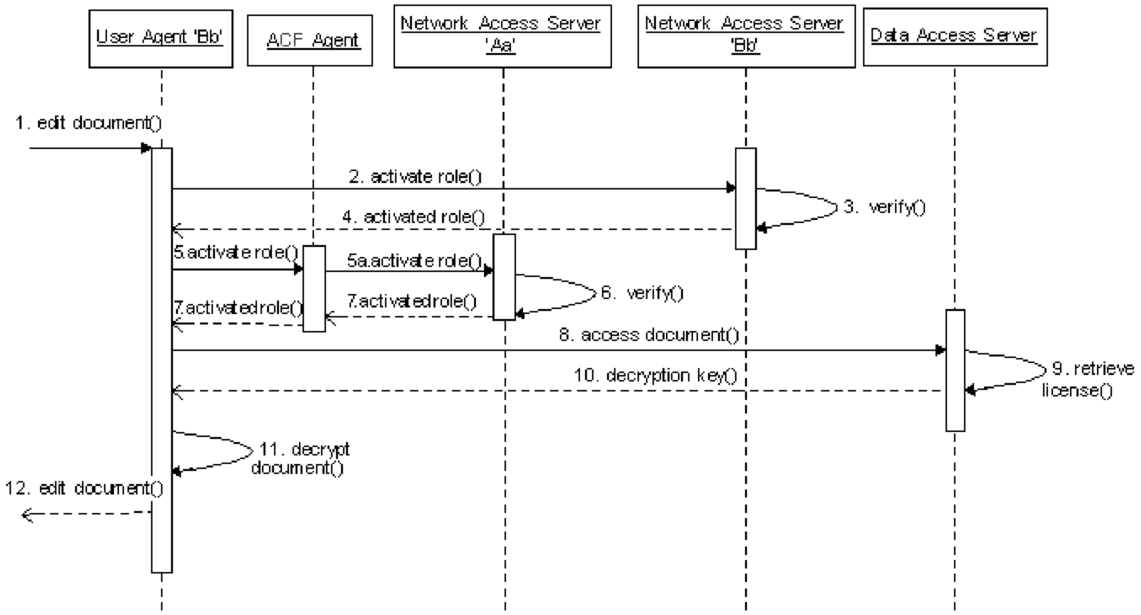


Figure 34. Use Case: Charlie edits document.

Use Case: Charlie stores document

1. Charlie contacts the Protection Server to apply protection to a document.
2. Charlie contacts the Governance Server to create the licenses for this document. Two role-based licenses, for Designers and Project Managers respectively, are created by the Governance Server in order to restrict access to these roles only. The Governance Server stores these licenses in a repository accessible by the Data Access Server.
3. Charlie formats the protected data into a new Digital Item using Content Server software, and stores it back on the shared data repository.

Figure 35 shows the sequence diagram associated with the 'Charlie stores document' use case.

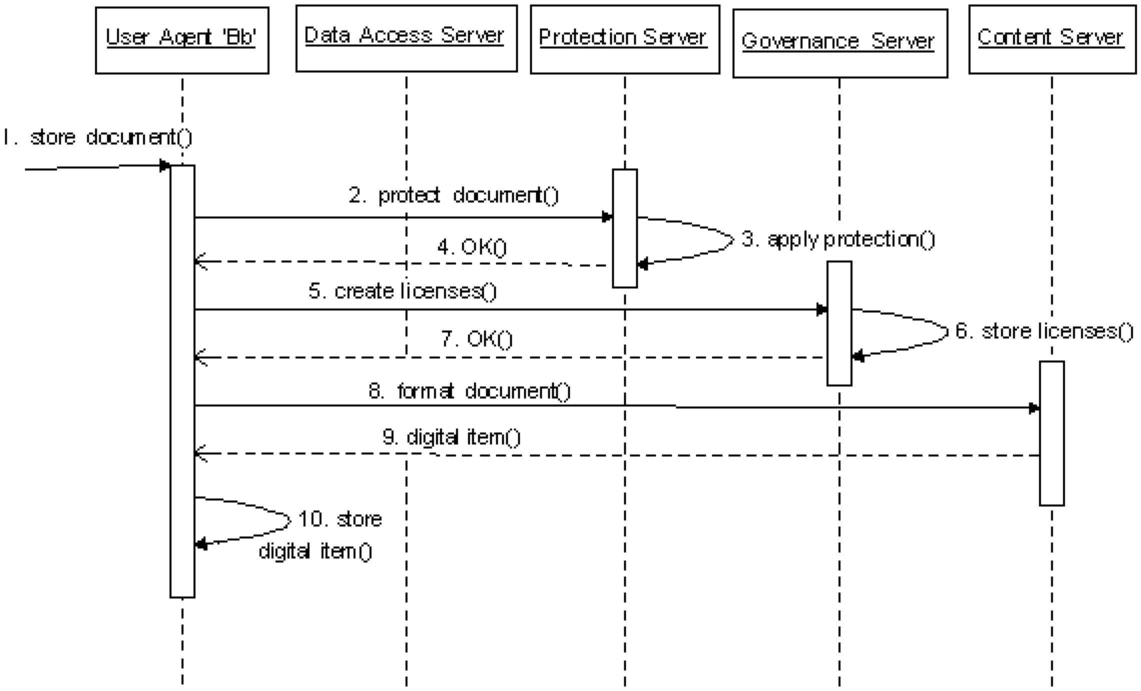


Figure 35. Use Case: Charlie stores document.

Widget Production Phase

Use Case: Bob reads document

1. Bob authenticates himself to the Network Access Server on Bb and requests to activate his Engineer role. The Network Access Server verifies he is allowed to do so, and returns a token containing this role.
2. Bob authenticates himself as an Engineer in Bb to the Network Access Server on Aa using this token. The Network Access Server maps the contained role to a Engineer in Aa, and returns a token containing this role.
3. Bob authenticates himself as an Engineer to the Data Access Server on Aa using this token, and requests access to the design document. The Data Access Server retrieves the relevant license for the document and the Engineer role, and checks that access is allowed. It then generates the decryption key and returns it to Bob.
4. Bob decrypts the document and reads it.

Figure 36 shows the sequence diagram associated with the 'Bob reads document' use case.

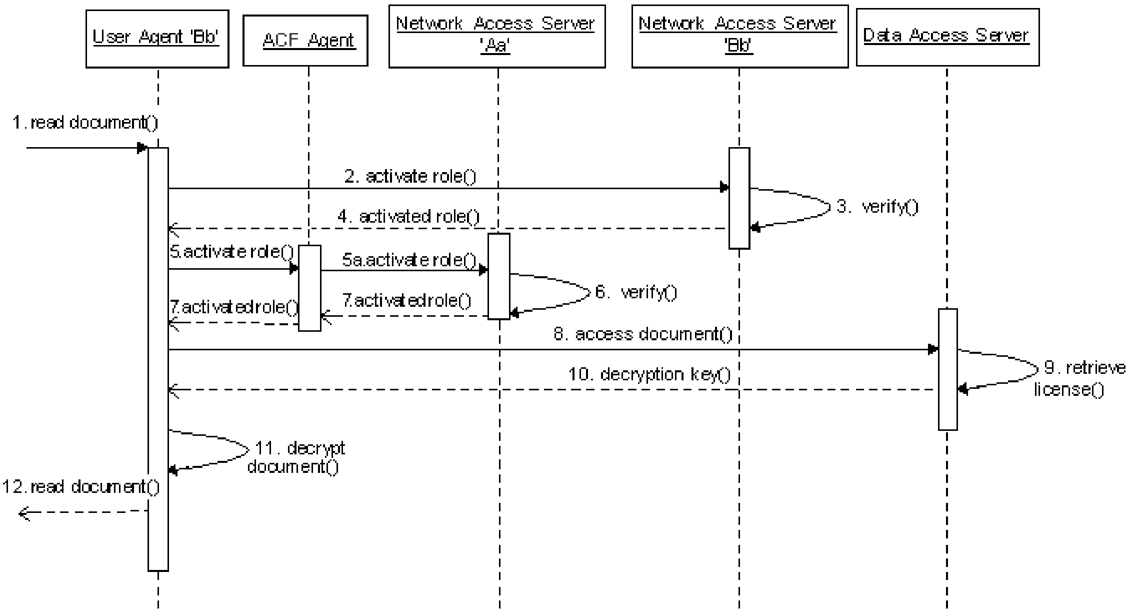


Figure 36. Use Case: Bob reads document.

In the Virtual Collaboration scenario, content usage rules can be defined using MPEG-21 or OMA DRM REL licenses, as well as access to resources and/or applications can be defined using XACML policies. Figure 37 sketches content usage rules expressed by means of an XACML policy. These rules also can be expressed using an MPEG-21 REL license that grants to designers of organisation Aa the right to edit the widget specification document (see Figure 38).

```
<Policy xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
PolicyId="urn:oasis:names:tc:example:SimplePolicy1" RuleCombiningAlgId="identifier:rule-combining-algorithm:deny-
overrides">
        <Description> Virtual Collaboration access control policy </Description>
```

```
    <Target/>
    <Rule RuleId="urn:oasis:names:tc:xacml:2.0:example:SimpleRule1" Effect="Permit">
        <Description> Any subject in the organisationAAA.com domain can view any document in the
        http://organisationAAA.com/schemas/doc.xsd namespace</Description>
        <Target>
            <Subjects>
                <Subject>
                    <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match">
                        <AttributeValue              DataType="http://www.w3.org/2001/XMLSchema#string">
                        DesignerOrganisationAa.com </AttributeValue>
                        <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
                        DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"/>
                    </SubjectMatch>
                </Subject>
            </Subjects>
            <Resources>
                <Resource>
                    <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                        <AttributeValue
                        DataType="http://www.w3.org/2001/XMLSchema#string">urn:example:vc:schemas:</Attri
                        buteValue>
                        <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:2.0:resource:target-
                        namespace" DataType="http://www.w3.org/2001/XMLSchema#string"/>
                    </ResourceMatch>
                    <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:xpath-node-match">
                        <AttributeValue
                        DataType="http://www.w3.org/2001/XMLSchema#string">/vc:doc</AttributeValue>
                        <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:xpath"
                        DataType="http://www.w3.org/2001/XMLSchema#string"/>
                    </ResourceMatch>
                </Resource>
            </Resources>
            <Actions>
                <Action>
                    <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                        <AttributeValue
                        DataType="http://www.w3.org/2001/XMLSchema#string">edit</AttributeValue>
                        <ActionAttributeDesignator      AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                        DataType="http://www.w3.org/2001/XMLSchema#string"/>
                    </ActionMatch>
                </Action>
            </Actions>
        </Target>
    </Rule>
</Policy>
```

Figure 37. Virtual Collaboration Policy example – Aa designer.

```
<r:license      xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS"      xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS"
xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS"                    xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/XML/1998/namespace" xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-REL-R-NS rel-
r.xsd urn:mpeg:mpeg21:2003:01-REL-SX-NS rel-sx.xsd urn:mpeg:mpeg21:2003:01-REL-MX-NS rel-mx.xsd">
  <r:grant>
    <r:keyHolder licensePartID="AaDesigner">
      <r:info>
        <dsig:KeyValue>
          <dsig:RSAKeyValue>
            <dsig:Modulus>KtdToQQyzA==</dsig:Modulus>
            <dsig:Exponent>AQABAA==</dsig:Exponent>
          </dsig:RSAKeyValue>
        </dsig:KeyValue>
      </r:info>
    </r:keyHolder>
    <mx:adapt/>
    <mx:diReference>
      <mx:identifier>urn:example:vc:doc</mx:identifier>
    </mx:diReference>
    </r:grant>
    <r:issuer>
        <r:keyHolder>
            <r:info>
                <dsig:KeyValue>
                    <dsig:RSAKeyValue>
                        <dsig:Modulus>X0j9q99yzA==</dsig:Modulus>
                        <dsig:Exponent>AQABAA==</dsig:Exponent>
                    </dsig:RSAKeyValue>
                </dsig:KeyValue>
            </r:info>
        </r:keyHolder>
    </r:issuer>
</r:license>
```

Figure 38. MPEG-21 REL license example– Aa designer.

5 Standards-Based Building Blocks for Modelling Content Management Scenarios

5.1 Introduction

Content distribution may cover different scenarios depending on aspects like if access to content has to be controlled, if content usage has to be reported or if licensing for content consumption is required. In this section, the analysis done over different content management and distribution scenarios and the identification of the high level functionalities needed by them, as described in [LLO12a] is presented. From this analysis, several standards-based building blocks are proposed in order to provide the functionality required by secure content management and distribution.

From the analysis of existing content management and distribution scenarios, the following ones have been identified, ordered by their level of complexity:

- ***Content licensing***: There is a license associated to content, but neither protection nor access control is required. This is the case for Creative Commons [CRE12a] licenses over content.
- ***Content licensing and authorisation-based content access control***: The access to content is authorised based on permissions owned by the user.
- ***DRM-enabled content access control***: The content is encrypted and decryption is authorised only if user owns an appropriate permission.
- ***DRM-enabled content access for mobile devices***: The same as the previous one, but oriented to mobile devices.

Once scenarios have been identified, they have been decomposed into the different features they provide, that is, their main high level functionality has been defined, which includes the following:

- ***Authentication***: Users and software components need to be authenticated when accessing a system, since permissions need to be bundled to them.
- ***Authorisation-based content access control***: Content access and usage is controlled by checking that users own a suitable license. If content access is for free, authorisation can be used to check if reporting is required. Otherwise, authorisation is needed for billing users according to content usage.
- ***Object management***: The representation of content generally uses a specific format to facilitate packaging and registration so as to be uniquely identified in the system. Registered content can be easily referenced from permissions.
- ***Licensing***: Permissions applicable to content usage need to be formalised through licenses when content is acquired. Even if licenses are used, content could be provided for free, while licensing conditions would serve for requiring, e.g., reporting of content usage.
- ***Protection***: Different mechanisms can be used to protect content when access to it is not open (e.g. encryption, watermarking, compression or a combination of mechanisms).
- ***Search***: Users need a way to find content in order to acquire it or to prove its registration. Filling content metadata is vital to offer such functionality.

- ***Storage and retrieval***: Users should be able to retrieve content from wherever it is stored if they own the appropriate usage rights.
- ***Tracking***: Content owners often want to be informed about content usage, and sometimes bill according to the real usage. Therefore, content usage needs to be registered. Other operations, like content creation may be also reported for informative purposes.

As stated in the beginning of this section, the work presented comes from an analysis of how content management systems cover different content management and distribution scenarios, including those that require digital rights management. Based on this analysis, a series of common building blocks have been identified, providing each part of the functionality required to model and implement different scenarios [DEL11a] [ISO13a].

The rest of this section provides a complete description of the content management and distribution scenarios presented [LLO12a] and the identification of standards-based building blocks (SB3) based on these scenarios, together with their operations. Later on, the mapping between MIPAMS [DEL11a] middleware services and operations and SB3s and operations is provided. Next, the mapping between MPEG-M Part 4: Elementary Services [ISO13b] and SB3s operations is described. Finally, some implementation issues of the scenarios based on MIPAMS middleware are presented.

5.2 Scenarios description

In this section, the content management and distribution scenarios presented in section 5.1 is described, illustrating their functioning by means of an example.

According to the high-level functionalities identified in section 5.1, Table 15 presents a summary of the specific needs of the scenarios presented in the rest of this section.

Table 15. Scenarios functionality summary

	Object Management	*Protection*	*Search*	*Licensing*	*Authorisation*	*Storage*	*Tracking*
Licensing	X		X	X			X
Licensing and Authorisation	X		X	X	X	Opt.*	X
Content Access Control (incl. mobile)	X	X	X	X	X	X	X

*Optional

5.2.1 Content Licensing

This scenario is applicable to some sites with specialised content (e.g. professional medical images) willing to offer users the possibility to trade with it. Although content can be directly accessed from those sites, it may be subject to some restrictions that do not enable users to use it for free. This is the case when content is distributed under copyright (all rights reserved) for example, or one of the different Creative Commons Non-Commercial models. The web portal would define some license templates to be chosen by users if desired when uploading their images. Content would be automatically registered through external services and a link would be provided from each image towards the trading portal for those users interested in licensing them (e.g. for being published in medical publications or used for comparing them with other medical images results).

5.2.2 Content Licensing Authorisation-Based Content Access Control

This scenario is useful for applications where users need to handle or modify content without restriction or when users do not want to be limited to use specific applications. Although the access to content is authorisation-based, content is given unprotected to the purchasing user, so they can enjoy it without restrictions.

For instance, it would be suitable for a professional trusted environment, where a content distributor may want a solution for trading and distributing unprotected audiovisual content. Content trade may support different licensing options, such as different pricing schemes, time frames, territories, etc. Content could be delivered unprotected, since it will be transformed by its recipient in order to adapt it to different online and offline publishing formats. However, the content distributor may want to be sure that only those clients that own a license can download content. That is, content access needs to be controlled and reported.

Another possible scenario would be that devised for content providers or distributors that want to use their own protection mechanisms and content management systems, consequently content is never stored outside their systems. In such scenario, when registering content, proprietary identifiers supplied by content provider need to be used for identifying external content provided by content owner. Once objects are registered, rights offers can be bundled to them. Licenses will be issued to formalise the acquisition by interested parties. Content providers or distributors will have to design their own applications to manage the access to encryption keys and content from their systems integrating them in access content applications (e.g. players and editors) or otherwise provide an API so that their content can be accessed from third-party applications. An example could be a TV broadcaster willing to license its own productions, but without storing them outside their systems, thus preserving complete control over its storage and distribution.

5.2.3 DRM-Enabled Content Access Control

This is the most common scenario, since it is covered by almost all DRM architectures. In this case, there is a need for an interface so that content creators can register and publish content together with their corresponding offers. This functionality can be provided by means of specific editing user applications or otherwise integrated in a web portal. Once content is registered, it can subsequently be linked from external sites. Therefore, content promoted in external sites can include links towards the licensing portal. Moreover, the licensing portal itself would also be useful for promotion. In this business scenario, content is accessed by using DRM-adapted tools such as players and other rendering applications.

An example of a business scenario in the music industry could be a music producer looking for a solution for publishing, trading and distributing protected audio files. As in the previous scenario, content trade needs to support different licensing options, such as different pricing schemes, time frames, territories, etc. However, here content access needs to be protected, controlled and optionally reported. Content registration and publication can be provided to the music producer through a customised publishing and trading portal that makes use of external services. Content access will be done through a DRM-enabled application that downloads or receives the streamed content. It is important to note that the previous example does not cover the case where content has to be further processed, in which a different set of rights (e.g. adapt, embed) would be needed.

5.2.4 DRM-Enabled Content Access for Mobile Devices

Inside this scenario the use of mobile devices not only for rendering but also for registering content being captured with the mobile device is considered. The goal for being able to register content from mobile devices would be related to author attribution or business opportunity, since content may have a relevant monetary value. This would be the case of images or videos corresponding to natural disasters, accidents or celebrities found in unexpected or funny situations. In such situations, the most important feature is the "opportunity" of the multimedia content being captured; that is, the fact they were taken in the right place at the right moment without the presence of official mass media when the event occurred. These videos or images may be used on TV shows or electronic newspapers or magazines, which could become a source of income for recognised authors.

5.3 Standards-based Building Blocks (SB3)

Based on the functionalities needed by the different scenarios identified in section 5.2, the definition of several building blocks is proposed. The main requirement for the building blocks defined is that they should be as generic as possible, in order to facilitate interoperability and provide flexibility. This will also allow its application to other business models working with different kinds of multimedia content not directly related to distribution of content for leisure purposes [LLO10a] [DEL10a] [DEL12a]. To do so, well-known standards in the area of application of each building block has been used to define them, so they will be referred as Standards-based Building Blocks (SB3) from now on.

The SB3 identified for the definition of content management scenarios are listed in Table 16, together with a brief description and the standard each one refers to. Using these SB3, any of the scenarios described in section 5.1 can be defined.

Table 16. Standards-based building blocks

Building Block	Description	Related Standard(s)
Access Control	Defines content usage rights and conditions through permissions in order to formalise content purchase. It is worth noting that, although a license can be applied to content usage, content may be provided for free. It also controls that user can access to content based on the permissions she owns.	MPEG-21 REL ODRL XACML MPEG-21 CEL
Authentication	Checks that users and software components are able to access the rest of components of the system. This SB3 has to be invoked before any other SB3.	SAML 2.0
Object Management	Represents and identifies content in a specific format to facilitate packaging and registration.	MPEG-21 DID MPEG-21 DII MPEG-21 IPMP Dublin Core

		Application-dependent metadata
Protection	Protects content using encryption mechanisms when content access is not open.	Encryption Algorithms MPEG-21 IPMP
Search	Provides searching functionality based on different metadata associated to content and licenses.	MPQF SQL
Storage and Retrieval	Provides access to content after positive authorisation, that is, user owns the appropriate rights.	Sockets-based HTTP FTP
Tracking	Stores information about different events occurred regarding content registration, licensing and content usage.	MPEG-21 ER

Legend:

MPEG-21 REL: ISO/IEC IS 21000-5 Rights Expression Language [ISO04a]

ODRL: Open Digital Rights Language [W3C02a]

XACML: eXtensible Access Control Markup Language [OAS05a]

MPEG-21 CEL: ISO/IEC IS 21000-20 Contract Expression Language [ISO13d]

SAML v2.0: Security Assertion Markup Language [OAS05b]

MPEG-21 DID: ISO/IEC IS 21000-2 Digital Item Declaration [ISO05a]

MPEG-21 DII: ISO/IEC IS 21000-3 Digital Item Identification [ISO03a]

MPEG-21 IPMP: ISO/IEC IS 21000-3 Intellectual Property Management and Protection [ISO06a]

MPQF: MPEG Query Format [ISO08a]

SQL: Structured Query Language [ISO11d]

HTTP: HyperText Transfer Protocol [W3C99a]

FTP: File Transfer Protocol [POS85a]

MPEG-21 ER: ISO/IEC IS 21000-15 Event Reporting [ISO06b]

5.3.1 SB3 Operations

Each SB3 provides several operations to make use of the functionality they provide. They are listed in Table 17, including a brief description of its operation and the standards used for its implementation.

Table 17. SB3 operations

Building Block	Operation	Description
Authentication	Login	Allows user to enter the system with her credentials that can be username and password, a SAML authentication request or other method required. In response, a SAML authentication response is generated, which proves user

		identity in front of the rest of services.
Access Control	Permission Creation	It is used to create permissions over digital objects assigned to users inside the system. MPEG-21 REL and XACML are the standards supported to express permissions.
	Permission Revocation	It is used to revoke an existing permission, which will not authorise user actions any more.
	Authorisation of User Actions	Checks if user has permission to perform a specific action over a digital resource. This authorisation depends on the language used for expressing permissions. Both MPEG-21 REL and XACML authorisation mechanisms are supported.
Object Management	Object Creation	Stores the information (metadata, structure, relationship) associated to one or more digital resources. MPEG-21 DID is used for the formalisation of digital objects. MPEG-21 IPMP is also used when security of the digital object structure is required.
	Object Deletion	Removes the information associated to a digital object. Associated permissions are removed or revoked.
Protection	Key Storage	Generates and stores an encryption key associated to a digital resource / object.
	Key Retrieval	Retrieves an encryption key associated to a digital resource / object. This operation can only be done after positive authorisation.
Search	Search	Performs searches over different types of information in the system, digital objects, offers, licenses or reports, according to the parameters passed. SQL queries have been implemented.
Storage and Retrieval	Resource Storage	Uploads a digital resource (image, video, audio, document, etc.) associated to a digital object. Sockets, FTP and HTTP solutions for content storage and retrieval have been implemented.
	Resource Retrieval	Downloads a digital resource (image, video, audio, document, etc.) associated to a digital object.
Tracking	Activity Report Creation	Stores the information associated to the operations occurred in the system, like object creation, permission creation or authorisation of actions. MPEG-21 ER is used to describe the activities done in the system.

5.4 Mapping SB3 to existing initiatives

In order to prove the appropriateness / correctness / completeness of the SB3 defined, in the following subsections it is provided the mapping of the SB3 defined to existing middleware platforms and standards that implement content management and distribution scenarios presented in section 5.2.

Specifically, two mappings are provided. The first one maps MIPAMS platform, described in section 3.1, services and operations to SB3 operations. The second one maps MPEG-M elementary services to SB3 operations. Finally, a complete comparison of the operations provided by MIPAMS, MPEG-M elementary services and SB3 operations proposed is done.

5.4.1 Mapping SB3 to MIPAMS modules

This section describes the mapping of the building blocks defined in the previous section to the services conforming MIPAMS platform. The correspondence between each SB3 and the corresponding MIPAMS Service is listed in Table 18.

Table 18. Correspondence between SB3 and MIPAMS Services

SB3	MIPAMS Service
Access Control	Authorization Service
Access Control	Licensing Service
Authentication	Authentication Service
Object Management	Object Registration Service
Protection	Protection Service
Search	Search Service
Storage and Retrieval	Content Service
Tracking	Reporting Service

As it can be seen, there is almost a one-to-one correspondence between building blocks and MIPAMS services. Therefore, it would be easy to convert MIPAMS services into the proposed SB3.

5.4.1.1 Mapping SB3 Operations to MIPAMS operations

In this section it is defined how SB3 operations map to MIPAMS Services operations. For each operation, it is provided the MIPAMS service it corresponds, the name of the operation in MIPAMS, the corresponding SB3 together with the operation and a brief description of its functionality. Operations provided by MIPAMS services are implemented using existing standards. The specific standard used to implement each operation is indicated in the corresponding operation description. The complete mapping is listed in Table 19.

Table 19. Correspondence between SB3 and MIPAMS Services operations

MIPAMS Service and operation	SB3 and operation	Description
Authentication - login	Authentication - Login	Allows user to enter the system with her credentials. In response, a SAML v2.0 token is generated, which has to be passed to the rest of operations, which check its validity before execution.
Authorization - authorize	Access Control - Authorisation of User Actions	Checks if user has an appropriate license to perform an action over a digital resource. The authorisation algorithm used is the one defined

		by MPEG-21 REL.
Licensing - createLicense	Access Control - Permission Creation	It is used to create offers or licenses inside the system. If license creation is requested, it is checked that the corresponding offer exists. The format used for the serialisation of the license is MPEG-21 REL.
Licensing - revokeLicense	Access Control - Permission Revocation	It is used to revoke an existing license, which will not authorise user actions any more.
Protection - storeKey	Protection - Key Storage	Generates and stores an encryption key associated to a digital resource / object.
Protection - retrieveKey	Protection - Key Retrieval	Retrieves an encryption key associated to a digital resource / object. This operation can only be done after positive authorisation.
Registration - sendObject	Object Management - Object Creation	Stores the information associated to one or more digital resources using the MPEG-21 DI format. For the definition of specific metadata, different standards depending on scope of the application implemented. The most basic one used is Dublin Core.
Registration - deleteObject	Object Management - Object Deletion	Removes the information associated to a digital object. Associated offers and licenses are also removed.
Reporting - saveER	Tracking - Activity Report Creation	Stores the information associated to the operations occurred in the system, like object creation, license creation or authorisation of actions.
Search - executeSearch	Search - Search	Performs searches over different types of information in the system, digital objects, offers, event reports, according to the parameters passed.
Content - storeResource	Storage and Retrieval - Resource Storage	Uploads a digital resource (image, video, audio, document, etc.) associated to a digital object.
Content - retrieveResource	Storage and Retrieval - Resource Retrieval	Downloads a digital resource (image, video, audio, document, etc.) associated to a digital object.

MIPAMS platform has also been successfully used in the implementation of eHealth applications [ROD11a], just substituting the licensing and authorisation services by ones based on XACML policies, where it was taken advantage from the rest of services in the platform.

5.4.2 Mapping SB3s to MPEG-M part 4 Elementary Services

MPEG-M Part 4 Elementary Services specifies Elementary Services (ES) [ISO13b] and their protocols, which are key elements in achieving services interoperability in the MPEG-M ecosystem. Nevertheless, ESs are mainly based on MPEG standards. With the definition of SB3s the intention is to go a step further, considering not only MPEG standards, but also other standards like ODRL or XACML for permission definition and authorisation.

This section describes the mapping of the SB3s defined in section 5.3 to the Elementary Services defined in MPEG-M Part 4 Elementary Services (ES), briefly described in section 3.2.3.4. The correspondence between each SB3 operation and the corresponding Elementary Service is listed in Table 20. In this case, several ESs are needed to provide the complete functionality supported by a SB3. So, the mapping between both becomes more complex, although still possible.

Table 20. Correspondence between SB3 and Elementary Services

SB3	Operation	Elementary Service
Authentication	Login	Authenticate User
Access Control	Permission Creation	Create License, Store License, Identify License (optional)
	Permission Revocation	Revoke License
	Authorisation of User Actions	Authorize User
Object Management	Object Creation	Create Content, Store Content, Identify Content (optional), Process Content (optional, needs to be specialised to use it)
	Object Deletion	Revoke Content
Protection	Key Storage	Associated to Create Content
	Key Retrieval	Associated to Authorize User
Search	Search	Search Content, Search License
Storage and Retrieval	Resource Storage	Store Content
	Resource Retrieval	Post Content, Deliver Content
Tracking	Activity Report Creation	Store Event

5.4.3 Relationship between MIPAMS and MPEG-M

MIPAMS (Multimedia Information Protection And Management System) implementation and set up was previous [TOR04a] to the description of the Multimedia Service Platform Technologies (MSPT), also known as MPEG-M (ISO/IEC 23006), standardisation initiative. MPEG-M is formed by several parts (5 at the present moment), whose aim is to define how devices, services and users interact in order to manage digital content.

Regarding MPEG-M, part 4 defines elementary services that can be combined to provide complex services. Some of them are already developed in MIPAMS, as shown in Table 21, where the equivalence between MIPAMS modules operations, SB3 operations and MPEG-M elementary services is shown.

MIPAMS has also been the basis for the implementation of several content management scenarios, presented in section 5.2. These scenarios combine operations from different MIPAMS modules by means of the Workflow Manager module. This approach is closely related to another part of MPEG-M, part 5 (ISO/IEC 23006-5 Aggregated Services), whose aim is the definition of guidelines for the composition of complex services from elementary and external services.

Table 21. Correspondence between SB3 Operations, MIPAMS Operations and Elementary Services

SB3 and operation	MIPAMS Service and operation	Elementary Service (s)
Authentication - Login	Authentication - login	Authenticate User
Access Control - Authorisation of User Actions	Authorization - authorize	Authorize User
Access Control - Permission Creation	Licensing - createLicense	Create License, Store License, Identify License
Access Control - Permission Revocation	Licensing - revokeLicense	Revoke License
Protection - Key Storage	Protection - storeKey	Associated to Create Content
Protection - Key Retrieval	Protection - retrieveKey	Associated to Authorize User
Object Management - Object Creation	Registration - sendObject	Create Content, Store Content, Identify Content, Process Content
Object Management - Object Deletion	Registration - deleteObject	Revoke Content
Tracking - Activity Report Creation	Reporting - saveER	Store Event
Search - Search	Search - executeSearch	Search Content, Search License
Storage and Retrieval - Resource Storage	Content - storeResource	Store Content
Storage and Retrieval - Resource Retrieval	Content - retrieveResource	Post Content, Deliver Content

5.5 Business use cases

This section describes two business use cases contributed to MPEG (ISO/IEC JTC1/SC29/WG11) [LLO12b] that finally became part of MPEG-M Part 5 standard.

They are based on the implementation of part of the DRM-enabled content access control scenario. In the use case presented it is implemented by means of a web application that accesses MIPAMS platform. They describe the operations needed to sell some multimedia content from the seller's point of view and to purchase some multimedia content from the buyer's point of view.

5.5.1 Seller use case

Figure 39 shows a sequence diagram to indicate the order of the operations that can be done on the web application from the seller point of view. It is worth noting that WM module is integrated into the web, so the services are directly invoked from the web, although other solutions are possible. Operation order is as follows:

1. User selects Register Object menu option to start the registration of a new object.
2. User has to fill information related to the complete object.
3. User clicks on Add Resource button, to add a new resource and fill its metadata.
4. User uploads resource file. It is temporally stored into the web application.
5. Object information is completely filled and user requests its registration.
6. WM requests ORS identifiers for the object and for the different resources being part of the object.
7. WM requests registration of resources one by one.
8. CS asks for encryption keys if resource file has to be protected.
9. CS encrypts and stores the file.
10. CS registers key into PS. The key is associated to the resource and object identifiers.
11. CS informs WM that resource has been properly registered. Steps 7 to 11 are repeated for each resource.
12. WM asks for object registration when all resources have been uploaded into the CS.
13. ORS sends a report to RS informing of the registration of the object.
14. WM informs user that object has been properly registered.
15. User creates offers over the registered object.
16. WM sends LS the offer created by the user.
17. LS sends a report to RS informing of the creation of a new license. Steps 15 to 17 are repeated for each offer created over the object.

The login operation is omitted but it should be done before step 1.

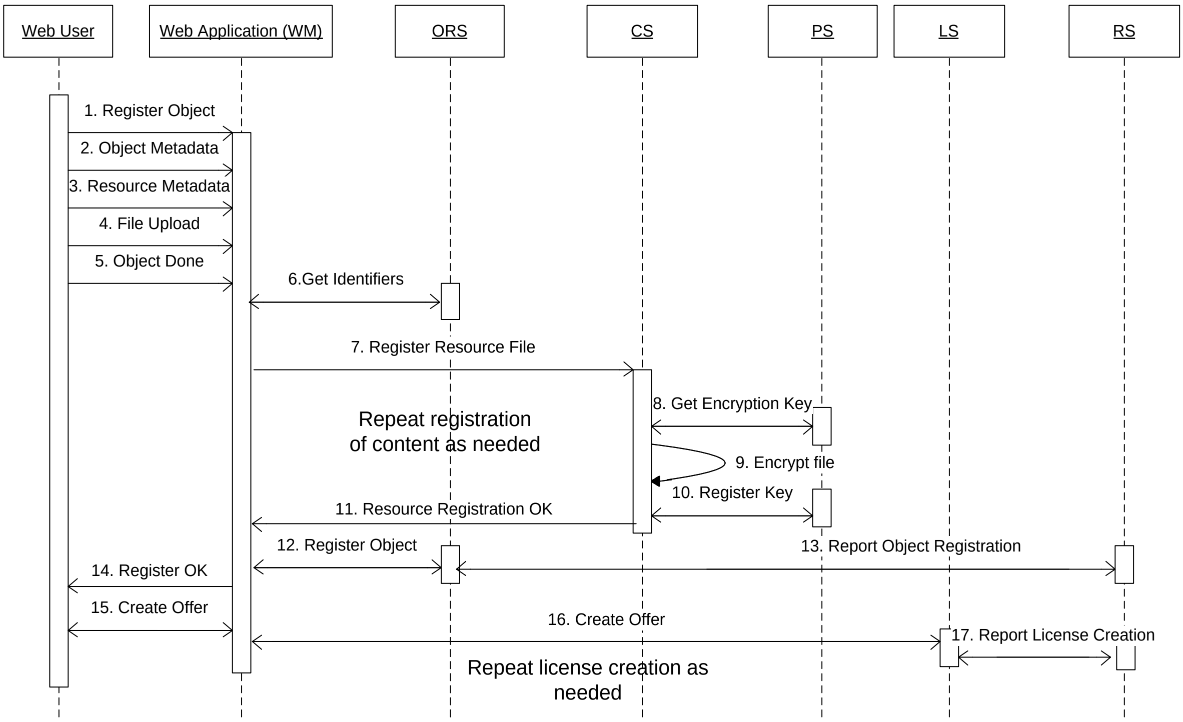


Figure 39. Object registration and offer creation on the web application.

Legend:

WM: Workflow Manager

ORS: Object Registration Service

CS: Content Service

PS: Protection Service

LS: License Service

RS: Reporting Service

5.5.2 Buyer use case

Figure 40 shows a sequence diagram to indicate the order of the operations that can be done from the web application from the buyer point of view. Operation order is as follows:

1. User selects Search and Buy menu option to look for the objects for sale. They can be filtered by Title and Creator.
2. WM sends object search request to the SS. Then, SS sends the search result to the WM.
3. Objects for sale are presented to the user.
4. User selects an object from the ones presented. She has to select an offer from the ones provided by seller in order to purchase the object.
5. WM asks for license creation to LS.
6. LS sends a report to RS informing of the license creation.
7. User downloads object she has purchased to render its content.

8. WM requests DI to SS.
9. User receives object.
10. User wants to render the object. Applet player is started.
11. Before rendering resource, applet asks AS for user authorisation. If authorisation is positive, keys for rendering content are provided.
12. User requests a resource, applets gets it from CS.
13. AS sends a report informing of positive authorisation to the user.
14. The received resource is decrypted and shown to the user. If user is not authorised, then the resource is not shown and user is informed of this fact.

The login operation is omitted but it should be done before step 1.

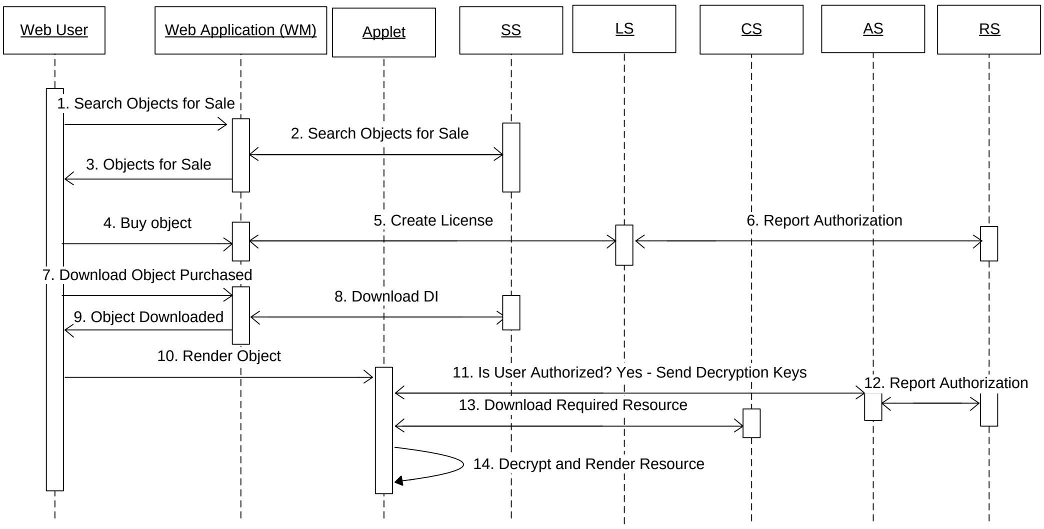

Web User
Web Application (WM)
Applet
SS
LS
CS
AS
RS
1. Search Objects for Sale
2. Search Objects for Sale
3. Objects for Sale
4. Buy object
5. Create License
6. Report Authorization
7. Download Object Purchased
8. Download DI
9. Object Downloaded
10. Render Object
11. Is User Authorized? Yes - Send Decryption Keys
12. Report Authorization
13. Download Required Resource
14. Decrypt and Render Resource

Figure 40. Object search and purchase on the web application. Authorisation and rendering of content on the applet.

Legend:

WM: Workflow Manager

SS: Search Service

LS: License Service

CS: Content Service

AS: Authorization Service

RS: Reporting Service

5.6 Modelling business use cases applying MPEG-M Part 5 methodology

In this section it is described how the MPEG-M Part 5: Service Aggregation methodology can be applied to define the business use cases presented in section 5.5, by means of two Aggregated Services (AS) built using ES defined in MPEG-M Part 4. The first AS presented corresponds to the buyer use case and the second AS corresponds to the seller use case. These use cases have been adapted to make use of ESs, applying the methodology steps as described next.

The MPEG-M Part 5 methodology defines five steps, some of them optional, to describe an aggregated service. The detailed description of these steps is listed next:

1) Narrative description of the service to be offered, by means of a textual description of the use case or scenario that is to be described as AS.
2) Identification of Elementary and Aggregated Services that are part of the AS to be defined. These ESs and ASs could be:

   2.1) ESs described in ISO/IEC 23006-4:2013 and/or ASs described in ISO/IEC 23006-5:2013.

   2.2) MPEG-M compliant ESs and/or ASs registered in the MPEG-M Registration Authority (MPEGMRA).

   2.3) External ESs specifically defined for the AS being defined. In this case, the messages must be specified following the rules provided in ISO/IEC 23006-4:2013. This new ESs may also be registered in the MPEGMRA.
3) Provision of a textual description of the AS service workflow describing interaction between Client and Service Provider making use of the following table:

Steps	Service invoked	Client + Message	Service Provider + Message	Description
Order of the operations involved.	Elementary (ES), External (EES) or Aggregated service (AS) involved. ES, EES or AS is added after service's name.	Who is taking the Client's role and which message is sent. Message is only indicated when Client is making a request.	Who is taking the Service Provider role and which message is sent. Message is only indicated when Service Provider is giving a response.	Textual description of what is done in this step.

4) Provision of a formal description of the service workflow of the resulting AS. It should describe both protocol and service by including the following elements:
   - Graphical service workflow diagram. BPMN 2.0 [OMG11a] can be used as a graphical representation for workflows.
   - (Optional) Service workflow XML serialisation. BPMN 2.0 can be used as a XML serialisation for workflows.
5) Optional registration of the resulting AS in the MPEGMRA. The MPEGMRA will syntactically validate the registered AS.

Later on, the AS can be implemented using different techniques. Some of them will be presented in section 5.7.

5.6.1 Application of the methodology to the seller use case

Step 1. Narrative description

In this use case, a content creator wants to register some multimedia content she has created (music score, video, photo or a combination of them) for putting it on sale. To do so, she connects to a web-based application which is the front-end to a MPEG-M compliant system which provides her with the functionality requested, that is, content registration, content storage, licensing capabilities and event reporting, among others.

Step 2. Identification of ESs and ASs present in the AS

The ESs to be used are: Authenticate User, Identify Content, Create Content, Store Content, Create License, Store License and Store Event.

Step 3. Workflow textual description

The service workflow associated to the Register and Sell Content aggregated service (RAS) is as follows:

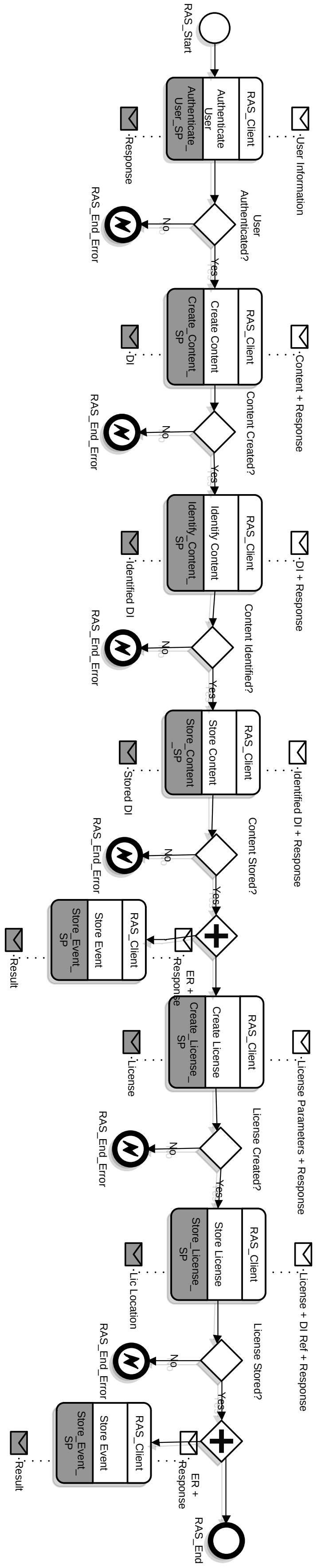
Steps	**Service invoked**	**Client + Message**	**Service Provider + Message**	**Description**
1.	Authenticate User (ES)	RAS Client `AuthenticateUserRequest`	Authenticate User SP	*RAS Client* tries to authenticate in the system.
2.	Authenticate User (ES)	RAS Client	Authenticate User SP `AuthenticateUserResponse`	*Authenticate User SP* tries to authenticate RAS Client with the information provided. If response is negative, AS stops here.
3.	Create Content (ES)	RAS Client `CreateContentRequest`	Create Content SP	*RAS Client* provides the required information to create some content. This information may include content metadata, resources and licenses (offers at this step).
4.	Create Content (ES)	RAS Client	Create Content SP `CreateContentResponse`	*Create Content SP* creates the content with the information sent from client. If this operation is not successful, AS stops here.
5.	Identify Content (ES)	RAS Client `IdentifyContentRequest`	Identify Content SP	*RAS Client* requests an identifier for the newly created content.
6.	Identify Content (ES)	RAS Client	Identify Content SP `IdentifyContentResponse`	*Identify Content SP* generates an identifier for the content. If this operation is not successful, AS stops here.
7.	Store Content (ES)	RAS Client `StoreContentRequest`	Store Content SP	*RAS Client* requests content to be stored.
8.	Store Content (ES)	RAS Client	Store Content SP `StoreContent`	*Store Content SP* stores the identified content sent by the

			Response	client. If this operation is not successful, AS stops here.
9.	Store Event (ES)	RAS Client StoreEventRequest	Store Event SP	*RAS Client* requests event to be stored. (This step is done in parallel with step 11).
10.	Store Event (ES)	RAS Client	Store Event SP StoreEventResponse	*Store Event SP* stores an event informing of the content stored.
11.	Create License (ES)	RAS Client CreateLicenseRequest	Create License SP	*RAS Client* provides the required information to create some licenses over content created in previous steps.
12.	Create License (ES)	RAS Client	Create License SP CreateLicenseResponse	*Create License SP* creates the license(s) with the information sent from client. If this operation is not successful, AS stops here.
13.	Store License (ES)	RAS Client StoreLicenseRequest	Store License SP	*RAS Client* requests license storage.
14.	Store License (ES)	RAS Client	Store License SP StoreLicenseResponse	*Store License SP* stores the licenses sent by RAS Client. If this operation is not successful, AS stops here.
15.	Store Event (ES)	RAS Client StoreEventRequest	Store Event SP	*RAS Client* requests event to be stored.
16.	Store Event (ES)	RAS Client	Store Event SP StoreEventResponse	*Store Event SP* stores an event informing of the licenses stored.

Step 4. Formal description of the service workflow

In this example, BPMN 2.0 is used for the formal description of the proposed AS. This description is as follows:

BPMN 2.0 Diagram



5.6.2 Application of the methodology to the buyer use case

Step 1. Narrative description

In this use case, a user wants to purchase some multimedia content on sale. To do so, she connects to a web-based application which is the front-end to a MPEG-M compliant system which provides her with the functionality requested, that is, content search, licensing capabilities and event reporting, among others.

Step 2. Identification of ESs and ASs present in the AS

The ESs to be used are: Authenticate User, Search Content, Create License, Store License, Store Event and Authorize User.

Step 3. Workflow textual description

The service workflow associated to the Buy and Consume Content aggregated service (BAC) is as follows:

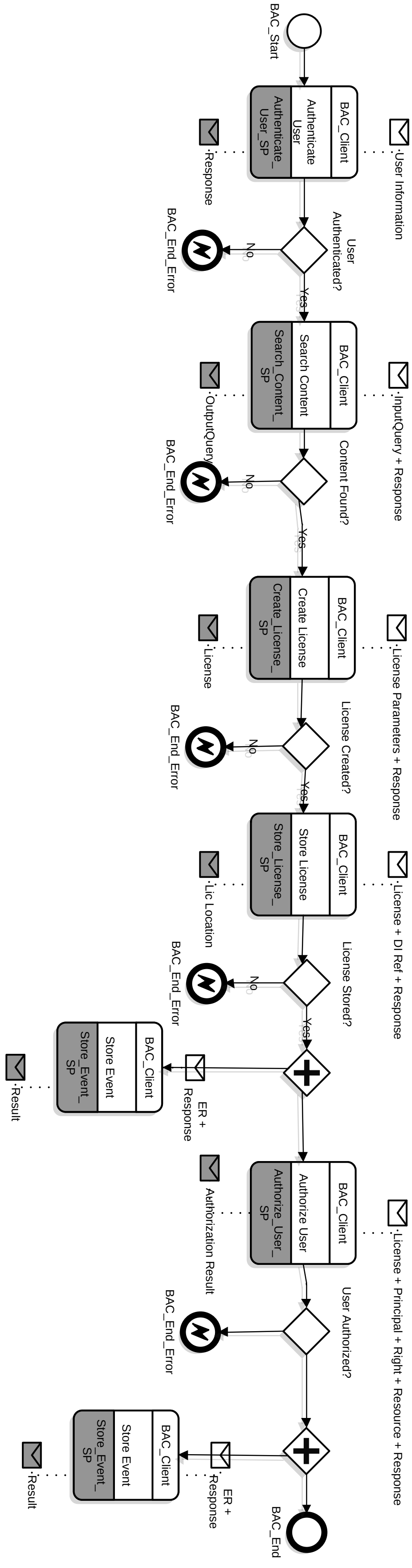
Steps	Service invoked	Client + Message	Service Provider + Message	Description
1.	Authenticate User (ES)	BAC Client `AuthenticateUserRequest`	Authenticate User SP	*BAC Client* tries to authenticate in the system.
2.	Authenticate User (ES)	BAC Client	Authenticate User SP `AuthenticateUserResponse`	*Authenticate User SP* tries to authenticate BAC Client with the information provided. If response is negative, AS stops here.
3.	Search Content (ES)	BAC Client `Search ContentRequest`	Search Content SP	*BAC Client* makes a search over offered content.
4.	Search Content (ES)	BAC Client	Search Content SP `Search ContentResponse`	*Search Content SP* makes a search with the requirements given by the BAC Client. If this operation is not successful, AS stops here.
5.	Create License (ES)	BAC Client `CreateLicenseRequest`	Create License SP	*BAC Client* provides the required information to create some licenses over content found in step 4.
6.	Create License (ES)	BAC Client	Create License SP `CreateLicenseResponse`	*Create License SP* creates the license(s) with the information sent from client. If this operation is not successful, AS stops here.
7.	Store License (ES)	BAC Client `StoreLicenseRequest`	Store License SP	*BAC Client* requests license storage.

8.	Store License (ES)	BAC Client	Store License SP StoreLicenseResponse	*Store License SP* stores the licenses sent by BAC Client. If this operation is not successful, AS stops here.
9.	Store Event (ES)	BAC Client StoreEventRequest	Store Event SP	*BAC Client* requests event to be stored. (This step is done in parallel with step 11).
10.	Store Event (ES)	BAC Client	Store Event SP StoreEventResponse	*Store Event SP* stores an event informing of the licenses stored.
11.	Authorize User (ES)	BAC Client AuthorizeUserRequest	Authorize User SP	*BAC Client* requests authorisation for content consumption.
12.	Authorize User (ES)	BAC Client	Authorize User SP AuthorizeUserResponse	*Authorize User SP* if user is authorised. If this operation is not successful, AS stops here.
13.	Store Event (ES)	BAC Client StoreEventRequest	Store Event SP	*BAC Client* requests event to be stored.
14.	Store Event (ES)	BAC Client	Store Event SP StoreEventResponse	*Store Event SP* stores an event informing of the content consumption.

Step 4. Formal description of the service workflow

In this example, BPMN 2.0 is used for the formal description of the proposed AS. The description is as follows:

BPMN 2.0 Diagram



5.7 *Implementing MPEG-M part 5 scenarios using MIPAMS*

This section describes some implementation issues related to the scenarios and use cases presented in sections 5.5 and 5.6. It is worth noting that MIPAMS has been successfully used in several research projects [MUS08a] [CUL09a] [MAR11a] that implement part of these scenarios.

Nevertheless, the applications presented here, a web and a mobile application, have a special relevance, as they have inspired the business use cases proposed to MPEG-M and have also been demonstrated to MPEG as examples of implementation of MPEG-M [DEL13a].

The web application implements a preliminary version of the buyer and seller use cases described in section 5.5.

The mobile application implements buyer and seller use cases using Android Operating System over MIPAMS. It has been implemented inside two final degree projects [FLO13a] [BER12a] and it also implements an MPEG-M wrapper to permit MIPAMS modules' operations to work as MPEG-M elementary services. Some of the underlying ideas of this application where presented in [DEL11b] [MAR12a].

5.7.1 Web application

In this application MIPAMS modules are used to provide the user with a complete electronic content registry and trading platform [MIP13a]. It is worth noting that the WM module is integrated inside the web application itself, as it acts as user interface and controls the order of the operations invoked depending on the menu options selected by the user. In this case, there is not a predefined order of the operations, as after login into the web application the user can select different options between the ones provided. Initial options for the user are shown in Figure 41.

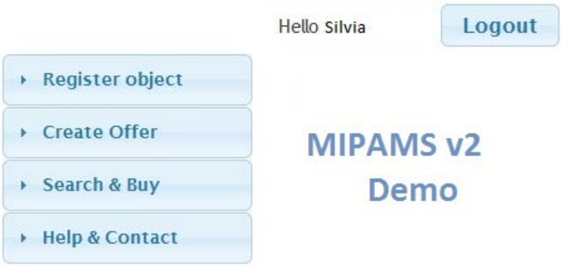
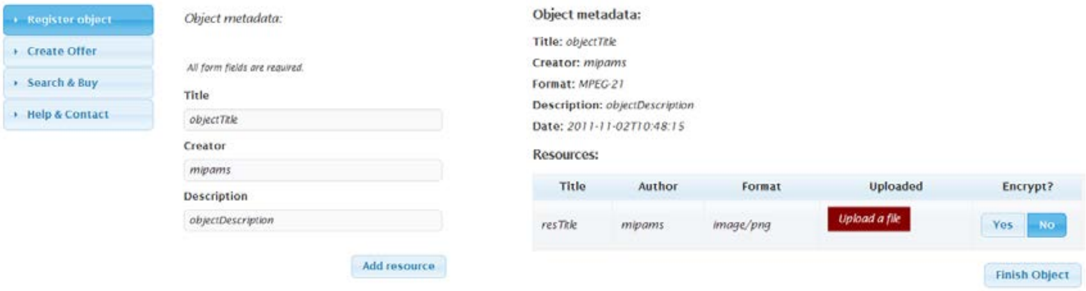


Figure 41. Main options in the portal.

The access to this registry is done through a dedicated web portal that enables the registration of content (Register Object option), the definition of different licensing options (Create Offer option), the search and purchase of content (licensing using option Search & Buy), as shown in Figure 41. Finally, a Help & Contact option is provided.

Content access is done through a DRM-enabled applet completely integrated into the web application, while any action in the system is registered through the RS. In this way it can be controlled that the content being accessed has been previously purchased by the user.

Figure 42 shows the web application screens where the user can register a new object. Some information is required from the user and other information, like the creator, is automatically filled. In order to register the object, the metadata needed is the title, the creator and the general description of the object being registered. From this screen it is also possible to add resources. Second part of the image shows the final screen for object registration, where a summary of the object is presented.



Object metadata:

All form fields are required.

Title: objectTitle

Creator: mipams

Description: objectDescription

Object metadata:

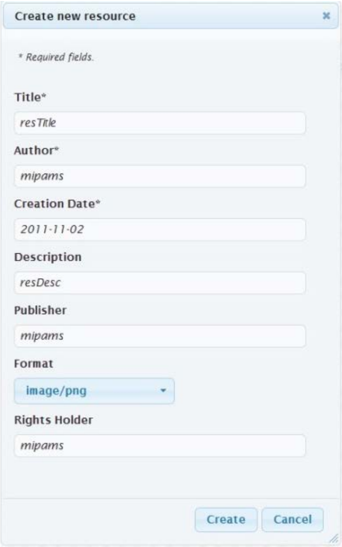
Title: objectTitle
Creator: mipams
Format: MPEG-21
Description: objectDescription
Date: 2011-11-02T10:48:15

Resources:

Title	Author	Format	Uploaded	Encrypt?
resTitle	mipams	image/png	Upload a file	Yes No

Figure 42. Object registration.

Figure 43 shows the information required for adding a resource. In this case it is an image that is part of the object being registered. Again, some fields are prefilled to help user when registering content.



* Required fields.

Title*: resTitle

Author*: mipams

Creation Date*: 2011-11-02

Description: resDesc

Publisher: mipams

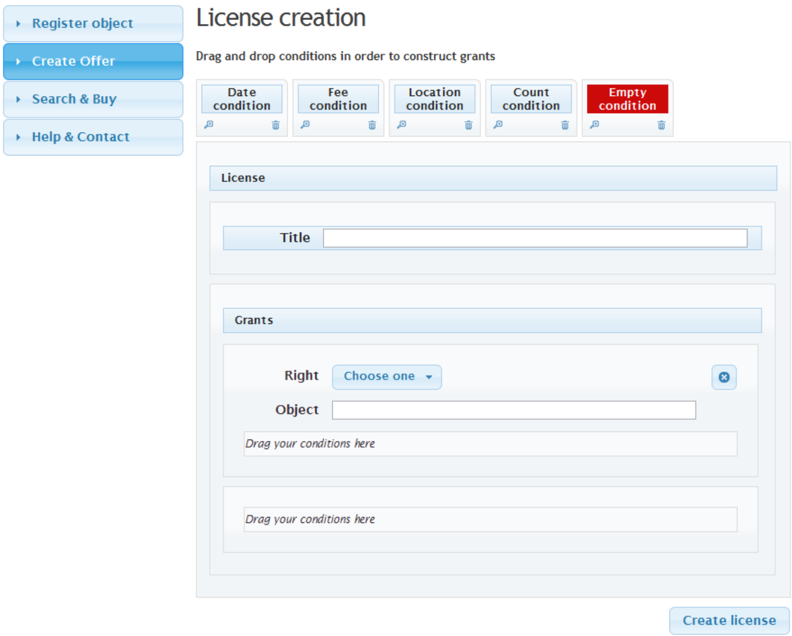
Format: image/png

Rights Holder: mipams

Figure 43. Resource registration.

Figure 44 shows part of the offer creation screen. In the web application it is possible to add new conditions by using drag and drop mechanism. The conditions implemented are Date Condition (to indicate a time interval), Fee Condition (to indicate how the user has to pay for consuming content), Location Condition (for defining specific territories) and Count Condition (for specifying the number of times that a user can consume some content). Right is the only element compulsory in the offer, conditions are optional.

Offers created will be internally stored in MPEG-21 REL as an XML file and it represents which conditions can select a user for consuming the digital object purchased. When a user buys an object, the offer is converted into a license by associating the offer to the user, filling a field called principal inside the license with the corresponding user identifier. Afterwards, when user wants to consume this content, AS will check if conditions in at least one of the licenses owned by user over this content are met. In this case, AS will send a positive authorisation and the keys from unprotecting content, if required.

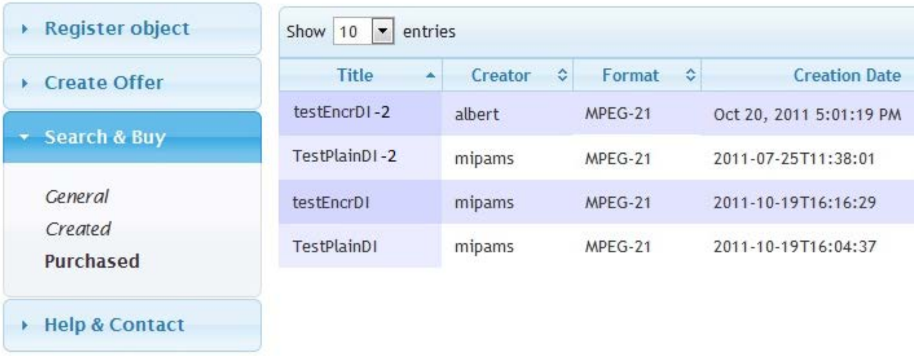


License creation

Drag and drop conditions in order to construct grants

Figure 44. Offer creation.

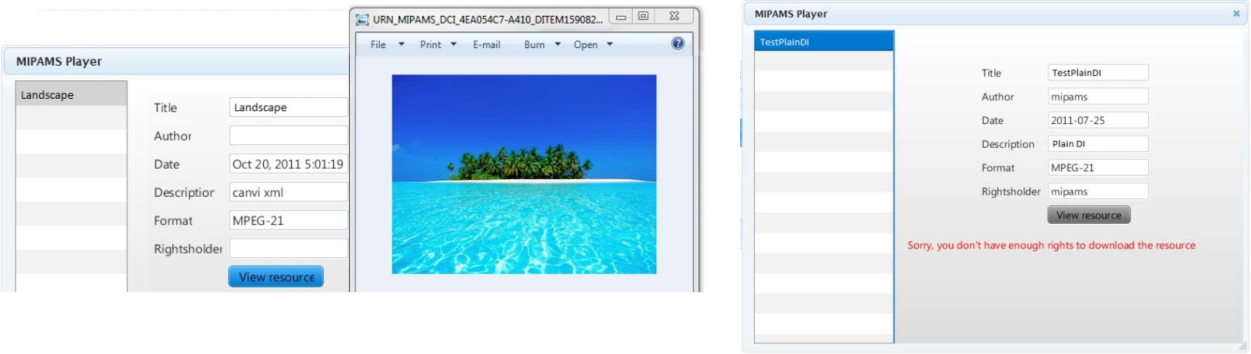
Figure 45 shows the objects purchased by a user of the web application. This screen shows the information of the digital object. When an object is selected, the different resources and its associated information are presented to the user and, if user downloads the object and she is authorised, then the MIPAMS applet player will show metadata associated to resources. It will also give the user the possibility of viewing the resource.



Title	Creator	Format	Creation Date
testEncrDI-2	albert	MPEG-21	Oct 20, 2011 5:01:19 PM
TestPlainDI-2	mipams	MPEG-21	2011-07-25T11:38:01
testEncrDI	mipams	MPEG-21	2011-10-19T16:16:29
TestPlainDI	mipams	MPEG-21	2011-10-19T16:04:37

Figure 45. Search between objects purchased by the user.

Figure 46 shows the MIPAMS player, a DRM-enabled applet, which gets digital item representations in XML language (following MPEG-21 DI format) together with its associated resources (images, text, word processing files, etc.). It prompts user for login and password, authorises access against licenses owned by user and, if user is authorised, recovers resource encryption keys and shows resources to the user. The applet uses the system application associated to each kind of file for presenting it to the user. Figure 46 also shows the applet screen when user is not authorised to access a resource.



MIPAMS Player

Landscape

Title: Landscape
Author:
Date: Oct 20, 2011 5:01:19
Descriptior: canvi xml
Format: MPEG-21
Rightsholdeı:

MIPAMS Player

TestPlainDI

Title: TestPlainDI
Author: mipams
Date: 2011-07-25
Description: Plain DI
Format: MPEG-21
Rightsholder: mipams

Sorry, you don't have enough rights to download the resource

Figure 46. MIPAMS Player applet. Positive and Negative authorisation.

5.7.2 Mobile application

The aggregated service described in subclause 6.4.3 of ISO/IEC 23006-5 (buyer scenario) can be implemented with MIPAMS by means of a mobile application. This scenario defines how a user can search for some content in the platform, purchase it and render into the mobile device after positive authorisation (based on the license she has purchased).

The developed application makes use of some Elementary Services in order to build an Aggregated Service that represents the buyer scenario described in the standard. The Elementary Services used are:

- Authenticate user in front of MIPAMS
- Search license between the ones available in the system
- Present license, showing the information it contains
- Create license for a specific user and object
- Authorize user, which checks if user has an appropriate license to perform a right
- Render content after positive authorisation

In order to implement the Aggregated Service, a MPEG-M interface for MIPAMS is provided. In other words, applications will call MPEG-M Elementary Services that will map to the MIPAMS services to be executed. The application architecture is shown in Figure 47.

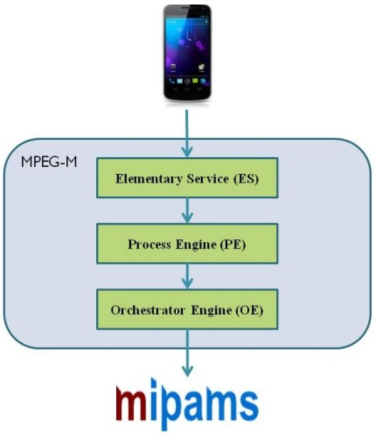
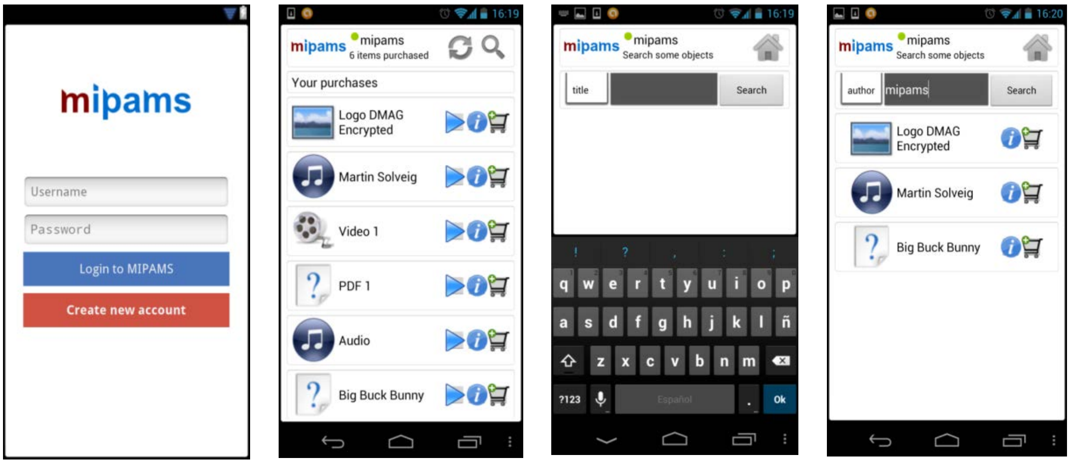


Figure 47. MIPAMS connection with MPEG-M.

The functionalities of the application implemented are presented below. Some screenshots and an explanation are provided. The login screen is depicted in Figure 48.a. It allows to login into MIPAMS by providing a username and a password. It also allows creating a new user.



a.Login screen b.User menu c.Search menu d.Search example

Figure 48. Some application captures.

When the user is logged in, the user menu is shown. The menu contains an action bar indicating the username, the number of items the user has purchased in MIPAMS, a search icon and a refresh icon. The user menu is depicted in Figure 48.b.

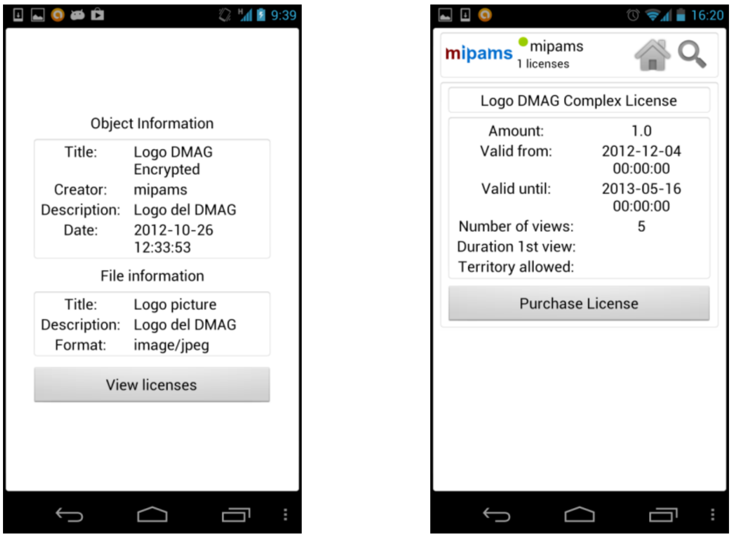
The screen displays the content the user has already purchased in a scrollable list. The user can render the objects (with a previous license based authorisation) and get the metadata of both the purchased Digital Item and its licenses. The refresh icon allows getting again the purchased items.

By pressing the search icon, a new screen is displayed. A search bar allows the user to search by author or title of the content. When the user presses the search button, the results are shown in a scrollable list like the user menu. For each result of the search, the user can display both the information about the content or its purchasable licenses. The search menu and an example of search result are depicted in Figures 48.c and 48.d.

The ⓘ icon displays the metadata of a Digital Item. It is accessible from both the user menu and the search menu. The information, which can be seen in Figure 49.a, is presented as follows:

- The next information of the Digital Item is presented at the top: title, creator, description and creation date.
- The information of all the object resources is presented down: title, description and file format.

The licenses of the object can also be displayed if the "View licenses" button is pressed. The screen is Figure 49.b.



Object Information

Title: Logo DMAG Encrypted
Creator: mipams
Description: Logo del DMAG
Date: 2012-10-26 12:33:53

File information

Title: Logo picture
Description: Logo del DMAG
Format: image/jpeg

View licenses

mipams
mipams
1 licenses

Logo DMAG Complex License

Amount: 1.0
Valid from: 2012-12-04 00:00:00
Valid until: 2013-05-16 00:00:00
Number of views: 5
Duration 1st view:
Territory allowed:

Purchase License

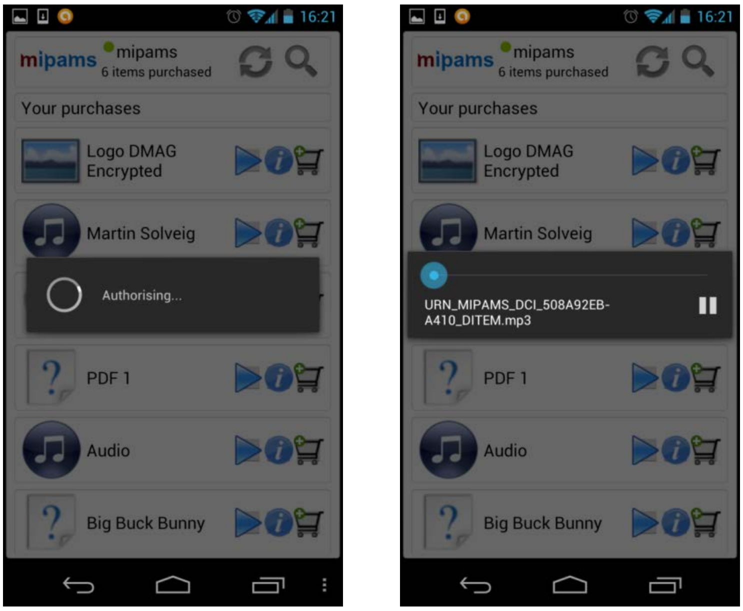
a. Object Information b. License Information

Figure 49. Object and license information screen.

The 🛒 icon displays information about the licenses of the object. If the icon is pressed in the user menu, it displays the information about the licenses the user has purchased for this item. Otherwise, if the user presses the button from the search menu, it shows all the purchasable licenses of this object. The licenses are presented in a scrollable list with the following information:

- License title.
- The cost of purchasing the license. Different currency could be defined.
- A time interval in which the license is valid (i.e. the interval in which the user is able to render the content).
- The maximum number of times a user can view the content.
- The territory where rendering the content is allowed.
- The time interval in which the render is available from the first time the user started to render it (i.e. two days since the first view).

When the user presses the ▶ icon from the user menu, an authorisation request is performed. The authorisation grants the access to the content if there is at least one license that allows the user to render this object. If the authorisation is positive, the content is downloaded and the default Android player opens the file to render it. An example of authorisation and content rendering is depicted in Figures 50.a and 50.b.



Authorising...

URN_MIPAMS_DCI_508A92EB-A410_DITEM.mp3

a. Authorising rendering b. Rendering a song

Figure 50. Authorising content rendering and rendering content in Android player.

5.7.3 Other scenarios implemented

This section describes other scenarios implemented that follow the service aggregation approach. They are previous to the work and publication of the MPEG-M standard series and are based on the MIPAMS content management platform.

5.7.3.1 First experiences with mobile applications

The work in the definition and implementation of mobile applications for the secure management and distribution of multimedia content started in the AXMEDIS project. This project and the tasks done in this area are briefly described next.

5.7.3.1.1 AXMEDIS project

AXMEDIS European Project [AXM04a] created an innovative technological framework for automatic production and distribution of cross-media contents over a number of different distribution channels (e.g., PC, PDA, kiosk, mobile phone, i-TV, etc.) with Digital Rights Management (DRM). To do so, the AXMEDIS DRM architecture was defined. DMAG-UPC was responsible of the implementation of this architecture.

The focus is on the solution implemented for the mobile environment which became one of the starting points, although not the only one, for the mobile scenarios based on the MIPAMS platform defined afterwards.

AXMEDIS DRM Architecture. The general description of the AXMEDIS DRM architecture main modules is as follows:

- *Protection Processor*: This client tool module was responsible for estimating the client tool fingerprint, enabling or disabling the tool, verifying the tool integrity and unprotecting protected multimedia objects.
- *Protection Manager Support Client (PMS Client)*: This client tool module managed and stored protection information, licenses, reports offline performed actions and other

secured information in a local secure storage system. It was responsible for authorising users to perform actions over objects with respect to digital licenses during offline operation. It also delivered protection information to the protection processor or requested it to the AXCS after a positive authorisation. It acted also as the intermediary module used by Protection Processor to contact AXCS to certify and verify tools.

- *Protection Manager Support Server (PMS Server)*: The functionality provided by this module was the following: 1) creation and storage of rights expressions, 2) adaptation of rights expressions, including translation, 3) authorisation of content usage based on the licenses owned by the user and 4) requesting protection information to the AXCS if needed. MPEG-21 REL and its authorisation model was used for creation of rights expressions and authorisation of actions over content, but also adaptation of rights expressions was implemented, including translation to OMA DRM REL.
- *AXMEDIS Certifier and Supervisor (AXCS)*: AXCS was composed by several modules that provided user registration, tool certification and verification, user and tool management, generation of unique identifiers and object metadata collection. AXCS was also responsible for saving the Protection Information related to protected multimedia objects as well as the actions performed on them, the so called Action Logs. They were the particular implementation of MPEG-21 Event Reports in the AXMEDIS context.

Figure 51 shows the relationship between these modules in the AXMEDIS DRM architecture.

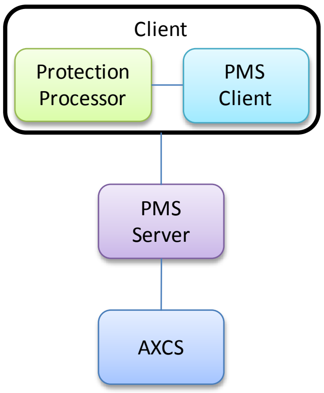


Figure 51. AXMEDIS simplified DRM architecture.

AXMEDIS Solution for Mobiles. The DRM solution for mobile devices implemented in the AXMEDIS project was mostly based on the MPEG-21 standard (license format, authorisation algorithm, event reporting) with some elements taken from OMA DRM (license format).

The licensing in this project was mostly done with MPEG-21 licenses. OMA licenses were created from the MPEG-21 ones using a translation tool, which worked in both directions (from OMA to MPEG-21 and from MPEG-21 to OMA). OMA licenses were used on devices supporting natively OMA DRM. Not all MPEG-21 REL licenses can be translated to the OMA ones, as MPEG-21 considers both distribution and final user licenses whilst OMA only allows final user licenses. Moreover, several elements are not able to be translated between the two languages, as they may not exist on the counterpart. This situation was partly solved by the MPEG-21 REL MAM profile.

So, for the implementation of the DRM solution it was used distribution and final user MPEG-21 licenses, MPEG-21 authorisation model and OMA licenses (translated from the

MPEG-21 ones). The event reporting mechanism implemented was based on MPEG-21 ER and it was used by all DRM tools implemented in the project.

Two different solutions for different kind of mobile devices were implemented. The first one was a Java language midlet that implemented a servlet client that sent the minimum information from the device to a servlet. The mechanism is similar to the one implemented by browsers in order to send form information to a web application. The servlet is the intermediary between the mobile device and the PMS Client, a client module implemented for PC that connects with the trusted servers through a secure channel. This solution could be used by devices supporting the Java midlet libraries needed to establish a secure http connection. The second solution was a version for Pocket PC of the C++ library that implemented the PMS Client, a client module described below. In both cases, the connection established between mobile device and the different servers was secure and each element of the chain had its corresponding X.509 certificate for authenticating it.

Figure 52 show the architecture of the DRM solution implemented for mobile devices and PDA's in the AXMEDIS project.

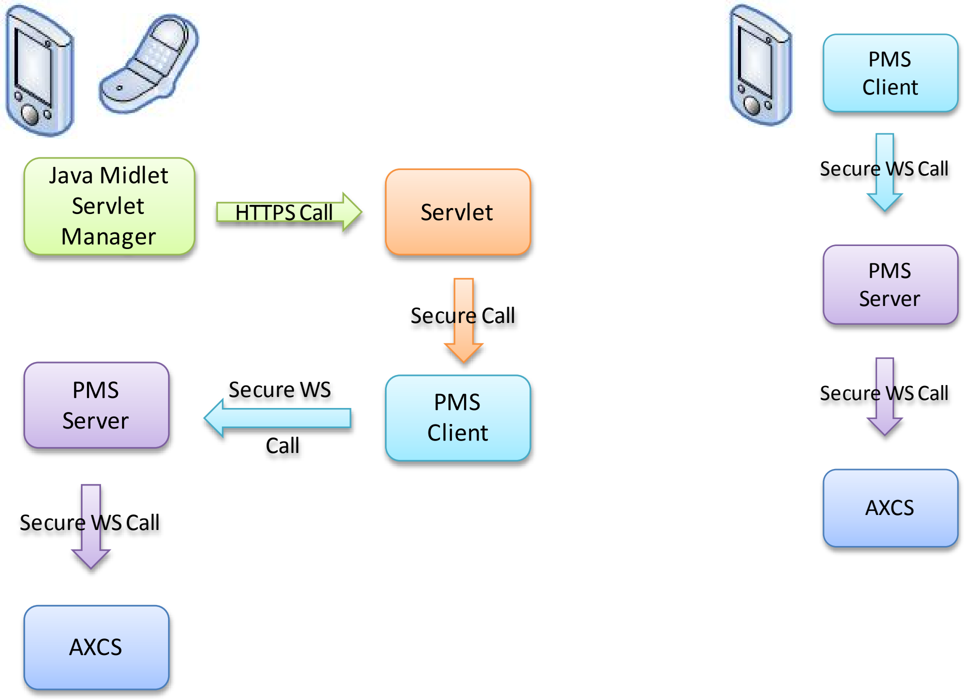


Figure 52. Different mobile implemented solutions.

The modules present in Figures 51 and 52 have the following functionality:

- *Java Midlet Servlet client:* Sent the basic information to be able to authorise the user to perform an action over content on the mobile device.
- *Servlet:* Received the information sent from the mobile device and transformed it into a call to the PMS Client.
- *Protection Manager Support Client (PMS Client):* This client tool module managed secure connection with PMS Server. In the PC version, it was able to store information regarding licenses and content consumption to perform local authorisations when working off-line (not connected to PMS Server).

- *Protection Manager Support Server (PMS Server):* This server side module was responsible for authorising users to perform actions over objects and requesting protection information to AXCS if needed. It acted also as an intermediary module to contact AXCS from PMS Client.
- *AXMEDIS Certifier and Supervisor (AXCS):* AXCS received the Action Logs corresponding to the actions over content done in the mobile device.

These modules provided the needed functionality to implement DRM-based rendering of multimedia content in mobile devices. See [LLO08a] for details.

5.7.3.2 Mobile application for an electronic publishing scenario

This scenario was firstly described in [DEL11b] and its implementation presented in [MAR12a]. It corresponds to DRM-enabled content access for mobile devices scenario presented in section 5.2.4.

Particularly, this scenario describes why mobile devices are especially relevant for electronic publishing, providing new ways of generating “publishable” content which may have an added value. The original scenario considered the principle that mobile devices are not the most suitable devices for capturing and distributing high quality images or videos. This is not true anymore, as existing smartphones and tablets have an image and video quality as good as portable digital cameras and can obtain almost professional results [SAM13a]. However, the foremost principle considered, is still valid. This principle is that those devices are always available and, possibly the most important feature for this scenario is the “opportunity” of the multimedia content being captured; that is, it was taken at the right place at the right moment. This is particularly important for unforeseen events such as natural disasters, accidents or celebrities found in an unexpected or funny situation, as this kind of events cannot be later reproduced and official mass media are not present at the instant when the event happens. In such cases, the multimedia content could even have monetary value for the author, who may try to sell or at least ask for attribution of the images or videos taken.

Starting from that scenario, a specific use case using a mobile scenario with Apple devices integrated with the MIPAMS platform is detailed.

5.7.3.2.1 Use Case

The use case presented in this section defines how MIPAMS modules interact with the mobile application. In such cases, the multimedia content captured by the mobile device may even have economical relevance, as they could be published in online newspapers, the gossip news or even broadcasted on television. Therefore, the author may register the content for different purposes: to try to get some revenues or just for later attribution.

Figure 53 depicts MIPAMS modules involved in the electronic publishing scenario and their interaction with the Mobile User (the person using the mobile device) and the Mobile Application (a specific application for registering content). It shows the registration of the content, image or video into the registration portal, including how the author can create some offers over the content to get some revenues. It is worth noting that one could create an offer that provides the content for free, but the authorship and some limiting conditions may still apply.

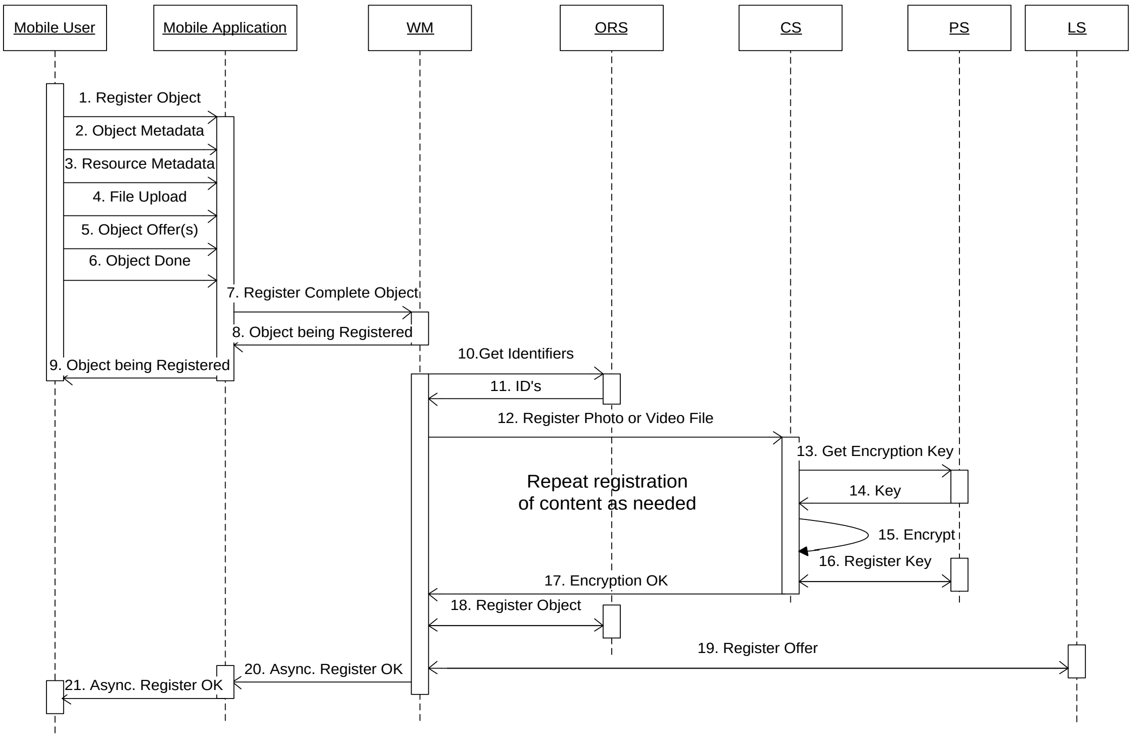


Figure 53. Registration of image or video captured with the mobile device, including offers.

Legend:

WM: Workflow Manager

ORS: Object Registration Service

CS: Content Service

PS: Protection Service

LS: License Service

Figure 53 illustrates the content registration process, including also offer creation, to facilitate the registration in a one-step-process from a Mobile Application. The specific steps involved in this process are the following:

1. User starts registration of content locally in an application for mobile devices (Mobile Application, MA). Some information is predefined to facilitate the registration process.
2. User fills a form with all metadata associated to the complete digital object. Some fields, like author, can be automatically filled by the MA.
3. User fills several forms (one for each image or video, that is, each resource) with metadata associated to each resource.
4. User selects the file containing each resource. This file is accessible by the mobile device (inside any local storage).
5. User defines the offers applying to the object. User has to insert the different sale conditions they offer for the registered object. These conditions include what can be done with the content (play, print, etc.) together with some conditions (territory, number of times one can perform the action or payment conditions). Again, license templates or predefined forms can be used to facilitate the task.

6. User indicates that all object information has been inserted and the registration information has to be sent to the server.
7. MA sends all information to the Workflow Manager (WM) module.
8. WM sends an immediate response to the MA. Later on, if the registration is successful, the Mobile User will be informed through the MA.
9. MA shows an indication that registration is in progress to the Mobile User.
10. WM requests identifiers to the Object Registration Service (ORS).
11. ORS sends the identifiers requested to the WM, one for the object and one for each resource file (even if they have not been already uploaded).
12. WM sends resource to the Content Service (CS).
13. Since the user has requested encryption of the resource, CS asks for encryption keys to the Protection Service (PS).
14. PS returns the keys for encryption algorithm and key length specified by CS.
15. CS encrypts and stores the file with the given key.
16. CS registers the encryption key in the PS for permitting later decryption.
17. CS sends WM notification of correct content storage and encryption. Steps 10 to 17 are repeated for each resource uploaded by the user. If no resources are uploaded, these steps can be done later.
18. When all resources are properly uploaded and encrypted, WM requests ORS the registration of the complete object, which is digitally signed to guarantee digital object integrity. The format used for storing the object is the MPEG-21 Digital Item.
19. WM registers the offer provided by the Mobile User. If there is more than one offer, this step will be repeated as many times as needed.
20. WM sends asynchronous notification of object registration to the MA.
21. MA informs the user that the object has been properly registered. In any case, Mobile User always can search her registered objects to check if everything is correct. This is especially useful when the mobile device connection is lost for some reason.

Once there are offers over a registered object, other users are able to purchase it. At that moment, a license is created based on the selected offer and the purchasing user can access the purchased content. For the moment, content download is only possible from an application installed in a laptop or PC, but with the major introduction of tablets between users, other possible applications and business scenarios could apply. Tablets are devices between 7 and 10 inches of screen that are not as powerful as PCs but they are more than a simple mobile phone. So, for tablets one could use the mobile application for registering content due to its quickness but also content download and consumption could be done combining both functionalities, the ones for mobile phones and the ones for PC.

Nevertheless, there are some other tasks that could be done from the mobile device. After registering and creating offers over some content, the author may look for her registered objects or check if any of her objects has been purchased, requesting the event reports associated to them. Figure 54 illustrates the interaction between the Mobile User and the different MIPAMS modules. In fact, there is one service dedicated to searches inside MIPAMS, which is the Search Service. This service accesses the necessary databases (Objects, Reports, etc.) to get the requested information.

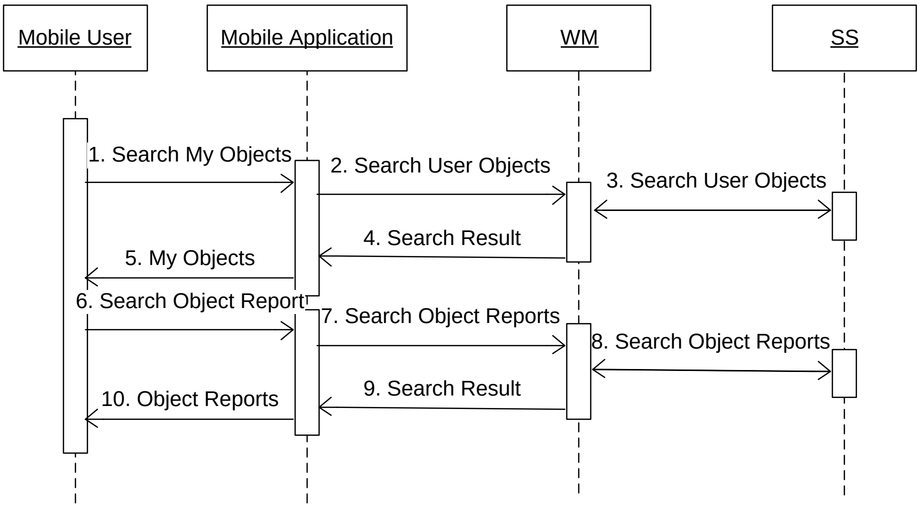


Figure 54. Search of object reports.

Figure 54 illustrates how a user can look for her registered objects and request the activity reports (mainly purchases) over them. The following steps are involved:

1. Mobile User searches her objects through the Mobile Application.
2. MA sends the request to the WM.
3. WM contacts the SS, which searches the objects registered by the specified author. SS sends the search result to the WM.
4. WM returns results to the MA.
5. MA presents objects information to the Mobile User.
6. The Mobile User selects an object from the ones returned by the previous search in the MA and requests its reports.
7. MA sends a report request to WM indicating which user's object wants to query.
8. WM sends a report request to SS indicating which user's object wants to query. SS returns Event Reports accomplishing the criteria to the WM.
9. WM returns Event Reports to the MA.
10. MA shows results to the Mobile User.

There are other possibilities in the use cases presented. For instance, the registration process may be implemented as a simple one-step process using predefined offers, where the right and some basic conditions are already defined and the user only has to select them. Once content is registered, users may always create other offers from a web portal. For instance, a user gives play permission to a low quality version of the content and later on offers a high quality version for broadcasting at a higher price.

Another improvement for mobile devices may be the minimisation of metadata associated to the different resources. This will facilitate registration as the messages sent will be smaller and thus the connection time will be less, too.

Apart from the mobile application, users will always be able to access content from a PC, and update offers and metadata as required.

Finally, a major concern for the mobile user may be the connection bandwidth and speed. So, if she does not have a quick connection, uploading a photo or a video may not be possible. In that case, she may just register basic metadata and offers and upload the digital resource later, by means of Wi-Fi connection, providing better connection capabilities and usually supported by current mobile devices.

5.7.3.2.2 iPhone implementation

This section describes the details of the development and the distribution of a MIPAMS client in a mobile environment. In the development subsection there are also some screenshots of a preliminary interface design of the application.

Development. Something to be considered when implementing a mobile application are the target device capabilities. Although there are many different devices that use the same OS, all of them have some common features: the reduced size of the screen and the limited computational capacity. Even the way in which user interacts with the device is very different from a PC environment. For all these reasons the application has to be as simple as possible in terms of screen interfaces and steps. A one-step content registration is proposed and the use of predefined offers to make the registration process easier and faster. Those tasks can be achieved by sending just one compacted message, reducing both the time and the amount of data sent.

Although both the content metadata and license introduced by the user are restricted, the current implementation of MIPAMS includes a web portal, which can be accessed from any web browser. Using this portal, a user may change object's information afterwards. That makes sense in a scenario where the user wants to upload the minimum object information to make it accessible as quickly as possible.

Another way to reduce the time waiting for the registration response from the server is the possibility of uploading the digital object separated from the metadata. This would reduce the global upload time and also the amount of data sent which is something to be taken into account when the user's connection is slow or inexistent. In addition, the registration process could be done asynchronously, letting the user interact with the other options of the application or even close it just a moment after the registration message is sent. In this case, the user would receive a message from the server with the registration result when the process is finished.

A first version of the described mobile application using iOS technologies has been developed. The first conceptual interface screenshots are the ones shown in Figure 55. In this version, only the content registration (metadata and licenses) and content upload is available. The search of other's objects and offers and the option of searching reports will be implemented later. As it is a mobile application, it would not include all the features that the MIPAMS portal has, but there are some other operations that could be considered like the possibility of including a simple license editor in the mobile device.

All these points make sense when talking about devices like phones or PDAs, but there are some other mobile devices that have to be taken in account. Both iOS and Android have a version of the operating system designed for tablets. It is not clear if it has to be a native application for this kind of devices or it is better to use an adapted web application.

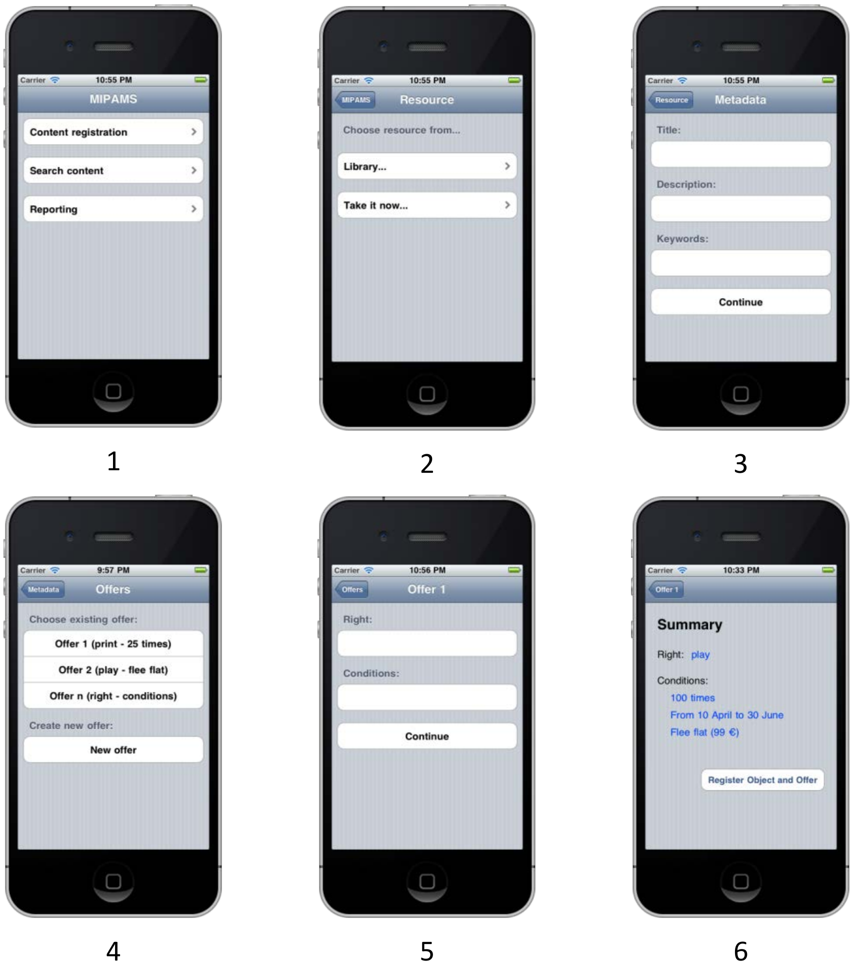
Image 1 in Figure 55 shows the different options provided to the user, content registration and search and reporting. Image 2 refers to the location of the content to be uploaded, followed by Image 3 where some basic metadata can be filled. Images 4 and 5 refer to the offers part of the application. Image 4 shows the offers menu, where some predefined offers can be selected. Image 5 shows the information the user has to fill in case she decides to create a new offer. Finally, Image 6 shows the object registration summary, showing the basic information that will be sent to the server for registering the object, which includes object information together with offers created.

At the time of writing, some of the steps mentioned above have been already developed. Some of the Workflow Manager (WM) functions in a servlet have been implemented. Doing so, the information sent and received by the mobile device is reduced to the minimum, and the time waiting for the response.

The first service that has been connected is the Authentication one. As this application is supposed to work in a trusted and secure environment, the user will not be able to use it without being authenticated in the MIPAMS system.

This first service has been connected using an OpenSource library called, ASIHTTP [ASI11a]. This framework lets us establish an HTTP connection in a very simple way. The application just has to know the destination URL and the parameters to send; then the library will manage the connection, the request and the response, including the possible errors. Also, it can work both synchronous and asynchronously, returning control to the mobile application while it is sending a request and waiting for the response. The library implements some protocol methods to alert the application whether it receives the response or identifies connection errors.

After this connection, the application is ready to continue with the registration process. To do so, the connection architecture used for login is maintained. A new method has been developed in the WM servlet, responsible of the Object Creation process (both the metadata and the digital resource upload). With that method, the mobile application is able to implement a one-step registration, sending the object metadata and digital resource at the same time, using an asynchronous connection. The servlet receives the request and makes the needed steps to register the object metadata, obtaining a unique id and uploading the resource. After that, WM sends the response message to the mobile application, which shows the user the registration result.



Summary

Right: play

Conditions:
100 times
From 10 April to 30 June
Flee flat (99 €)

Figure 55. MIPAMS mobile client screenshots.

Distribution. One of the most restrictive things in Apple mobile devices is the business model used on the applications distribution. When an application is uploaded to the App Store, Apple retains the 30% of the benefits, and does not let the application use any other payment system. That constraint reduces the possibilities of creating a parallel business model inside this kind of applications. But, for the specific case of MIPAMS, the client application distributed using the App Store would be free of charge, allowing users to register and upload content inside MIPAMS. It would not be possible to buy any content directly through it, avoiding Apple's distribution charges. So, for the business model presented, MIPAMS application is not breaking Apple's rules but complementing them. The user will just use the application to upload or query the content, metadata and offers, but the real transactions for acquiring the licenses will be done through the MIPAMS web, possibly using a PC or a tablet.

Furthermore, MIPAMS mobile client would not allow the user to buy or download content. The content download and use would imply a much higher complexity in the application development and the possibility that Apple refuses it. It would be interesting to have some preview of the content as the user searches through them, but this possibility will be studied for future versions of software.

It is worth noting that Android Market also retains a 30% of benefits when selling an Android App. Thus, the business model for Android would be very similar to the one for Apple. Also in this case, content will be purchased through MIPAMS portal instead of the mobile application.

5.7.3.3 Broadcasting of live cultural events: CulturaLive

This section describes CulturaLive [CUL09a] [MAR11a], a project for broadcasting live cultural events from local televisions in a low cost way, respecting rights governing broadcasted content. The main objective of this project was to develop a system to generate and broadcast in real time video streams with professional quality from any source to multiple destinations. It provided to professional users of the cultural environment an appropriate solution for live audiovisual content distribution and reception. This implementation corresponds to the DRM-enabled content access control scenario described in subsection 5.2.3.

Specifically, a platform for content syndication and exploitation providing professional quality of local scope live content was offered to local televisions. This platform also included licenses for governing live content being transmitted. Using these licenses, it was possible to control access to live events and also manage content purchase and later billing. These services were provided by DMAG [DMA13a] through its MIPAMS platform, already described in section 3.1. In this way, the project provided local television stations in Catalonia with a system for the acquisition and transmission of cultural content such as live operas, concerts, conferences, major events or even festivals, including access control and rights management.

This system was an inexpensive alternative to existing live retransmission models, which are unachievable for broadcasters with a low budget because they are done via satellite connections. More than that, using the architecture presented in Figure 56 based on Peer-to-Peer (P2P) nodes, the retransmission could be reached far from its source with minimum delay. Using such an architecture, Digital Rights Management (DRM) services provided by MIPAMS did not interfere with live retransmissions.

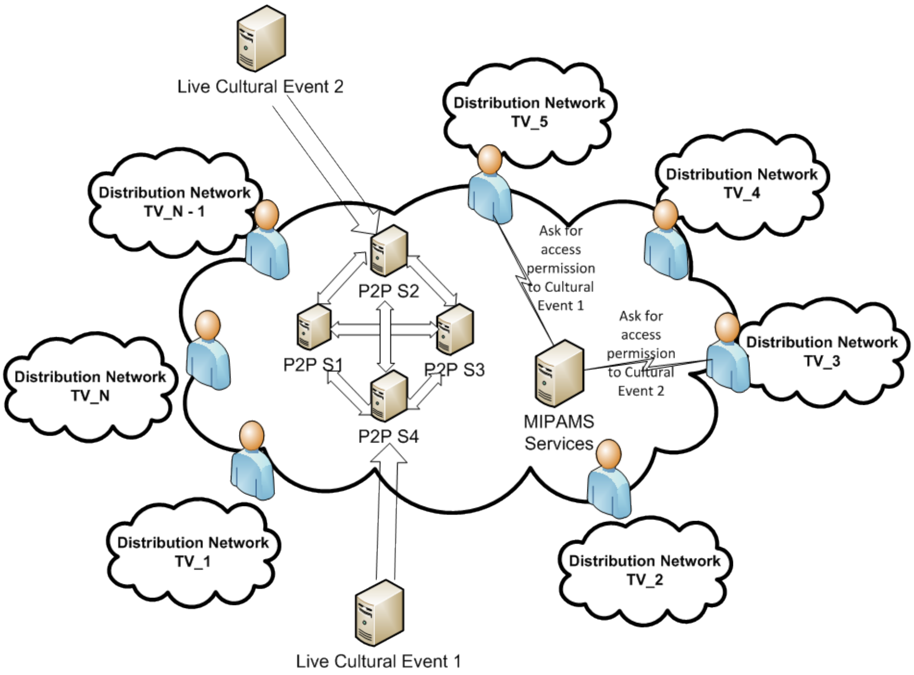

Live Cultural Event 2
Distribution Network TV_5
Distribution Network TV_4
Distribution Network TV_N - 1
Ask for access permission to Cultural Event 1
P2P S2
Ask for access permission to Cultural Event 2
Distribution Network TV_3
P2P S1
P2P S3
Distribution Network TV_N
P2P S4
MIPAMS Services
Distribution Network TV_1
Distribution Network TV_2
Live Cultural Event 1

Figure 56. CulturaLive project architecture.

Figure 56 shows the CulturaLive project architecture, where different local television stations participated in the distribution network. This network was composed by a P2P network where live events were transmitted. The local TV stations interested in broadcasting these events had to purchase the corresponding license and, when the live event was occurring, ask for authorisation to MIPAMS services and, if authorisation was positive, connect to the P2P network to receive it in professional quality and broadcast it through its TV channel in real time.

In Figure 56, two live cultural events are being transmitted via the P2P network. TV station 5 asks for access permission for cultural event 1 to MIPAMS services. TV station 3 asks for access permission for cultural event 2. If these operations are authorised (they had already purchased a license for accessing live event), TV stations are able to access the corresponding live content from P2P network and broadcast it through their stations.

5.7.3.3.1 Integrating MIPAMS

In this section the integration between MIPAMS and Culturalive client application for the creation and management of licenses governing the live multimedia contents broadcasted is described. Figure 57 shows the steps involving offer creation, content purchase and usage and reporting.

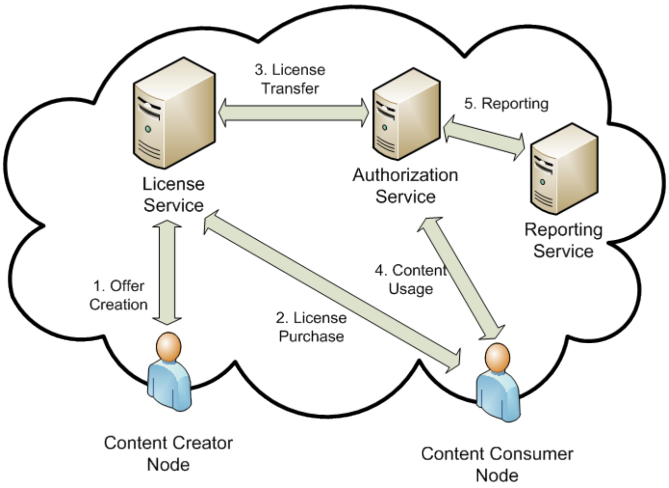


Figure 57. Licensing, Authorisation and Reporting in CulturaLive.

Figure 57 shows the typical use case involving content creator and content consumer. In this use case, content creator wants to broadcast live digital multimedia content. In order to permit the consumption of this content by other users, content creator has to create some offers over the event. The sequence of actions is as follows:

1. Content Creator (CCr) asks for an offer creation over the live content to License Service (LS). CCr decides which conditions apply (time or territory conditions, pricing, etc.). From that moment on, other users can purchase this live event. It is worth noting that offer creation and license purchase have to be done before the live content is to be broadcasted.
2. Content Consumer (CCo) can query for offers available for a specific event. From the ones presented, CCo can purchase the most suitable one for the live event to LS.
3. LS creates the license for CCo and transfers it to Authorization Service (AS) for later authorisation of access to the event.

4. CCo wants to consume the live event when it is to be broadcasted. To do so, she has to ask for authorisation to AS, which checks that a license exist for permitting access to the event and that all conditions apply. In case of positive authorisation, CCo can access to the live content.
5. Finally, AS sends a report to Reporting Service (RS) informing of the authorisation decision (positive or negative) that will be later used for billing purposes. It is worth noting that reports can also be used for tracking user actions or claims resolution.

A specific interface through the Workflow Manager (WM) in order to connect the client application to the MIPAMS platform used in the project had to be implemented. The reason for doing so was that the client application used a specific proprietary application protocol, which did not follow web service paradigm. The main operations of the protocol involved were the following:

- Create offer: This operation generates an offer for the live content. It has to be called for each offer the CCr wants to associate to the content.
- Search: It was used to search offers or licenses with the following particularities for each case.
    - Offer: This operation searches offers associated to a specific live event for presenting them to a user interested in purchasing it.
    - License: This operation searches licenses already purchased by a user for a specific content.
- Rights Issuance: It was used to purchase content or to authorise its usage with the following particularities for each case.
    - Purchase content: This operation creates a license that gives permission to a user to consume some content.
    - Authorise access to content: This operation checks if a user is authorised to consume some content.

As it can be seen, the operations Search and Rights Issuance were used for two different purposes. This was done to minimise the number of operations involved in the integration of digital rights management into the client application. To distinguish between them, different parameters to Search and Rights Issuance operations were passed.

It is worth noting that as broadcasting of the cultural event was done in real time, the offers creation and search and the license purchase concerning live event had to take place before the event was broadcasted. Only the authorisation process had to be done at the beginning of the transmission. This assures that the client received the content without delay coming from the use of MIPAMS services.

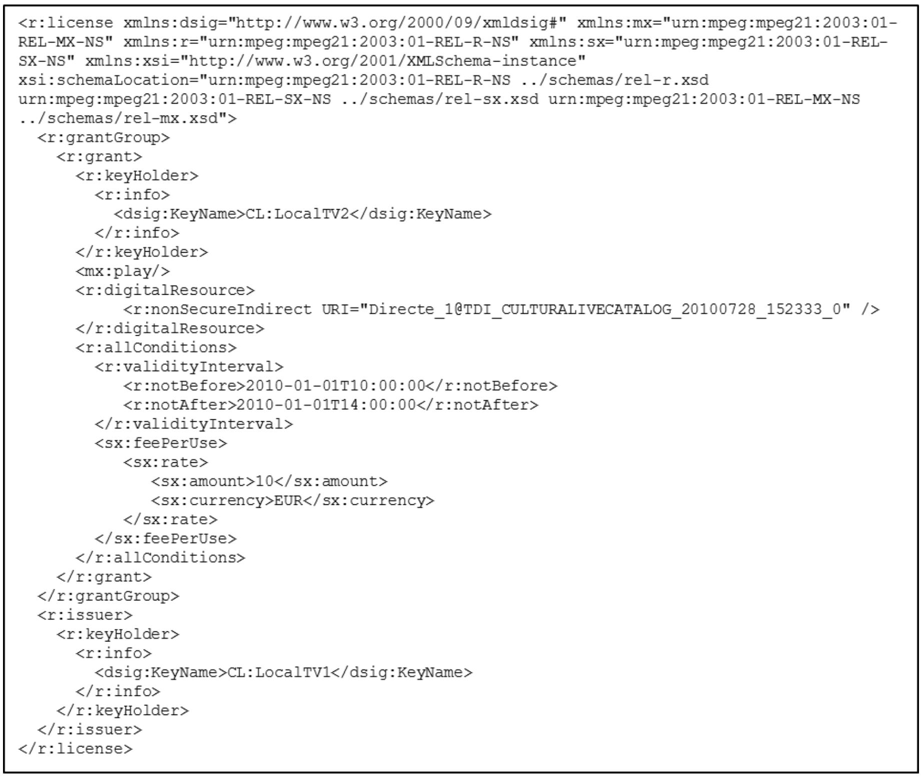
5.7.3.3.2 Management of Rights in CulturaLive

In CulturaLive project, MPEG-21 REL licenses were used, because this is the rights expression language currently implemented in MIPAMS.

For managing access to the live cultural events broadcasted in CulturaLive, two different kinds of rights expressions, offers and licenses were used. Offers defined the right and conditions that the broadcaster of live event was willing to issue to any other TV participating in the project. Licenses defined the rights and conditions (coming from the original offer) that broadcaster issued to the license purchaser TV. Offer creation and license purchase had to be done prior to the live event broadcast.

Afterwards, when a TV requested authorisation for accessing a live event, AS module checked if there was a license issued by the event broadcaster for this TV and this specific event. If so, the authorisation was positive and a report was generated and sent to RS. If there was no license, then the authorisation response was negative and TV was not able to access live event transmission.

Figure 58 shows the license between local TV station 1 and local TV station 2 where TV 1 gives play permission for live event Directe_1 to TV 2.



```
<r:license xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" xmlns:mx="urn:mpeg:mpeg21:2003:01-
REL-MX-NS" xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-
SX-NS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-REL-R-NS ../schemas/rel-r.xsd
urn:mpeg:mpeg21:2003:01-REL-SX-NS ../schemas/rel-sx.xsd urn:mpeg:mpeg21:2003:01-REL-MX-NS
../schemas/rel-mx.xsd">
  <r:grantGroup>
    <r:grant>
      <r:keyHolder>
        <r:info>
          <dsig:KeyName>CL:LocalTV2</dsig:KeyName>
        </r:info>
      </r:keyHolder>
      <mx:play/>
      <r:digitalResource>
           <r:nonSecureIndirect URI="Directe_1@TDI_CULTURALIVECATALOG_20100728_152333_0" />
      </r:digitalResource>
      <r:allConditions>
        <r:validityInterval>
           <r:notBefore>2010-01-01T10:00:00</r:notBefore>
           <r:notAfter>2010-01-01T14:00:00</r:notAfter>
        </r:validityInterval>
        <sx:feePerUse>
           <sx:rate>
              <sx:amount>10</sx:amount>
              <sx:currency>EUR</sx:currency>
           </sx:rate>
        </sx:feePerUse>
      </r:allConditions>
    </r:grant>
  </r:grantGroup>
  <r:issuer>
    <r:keyHolder>
      <r:info>
        <dsig:KeyName>CL:LocalTV1</dsig:KeyName>
      </r:info>
    </r:keyHolder>
  </r:issuer>
</r:license>
```

Figure 58. MPEG-21 REL license for giving access permission to live event.

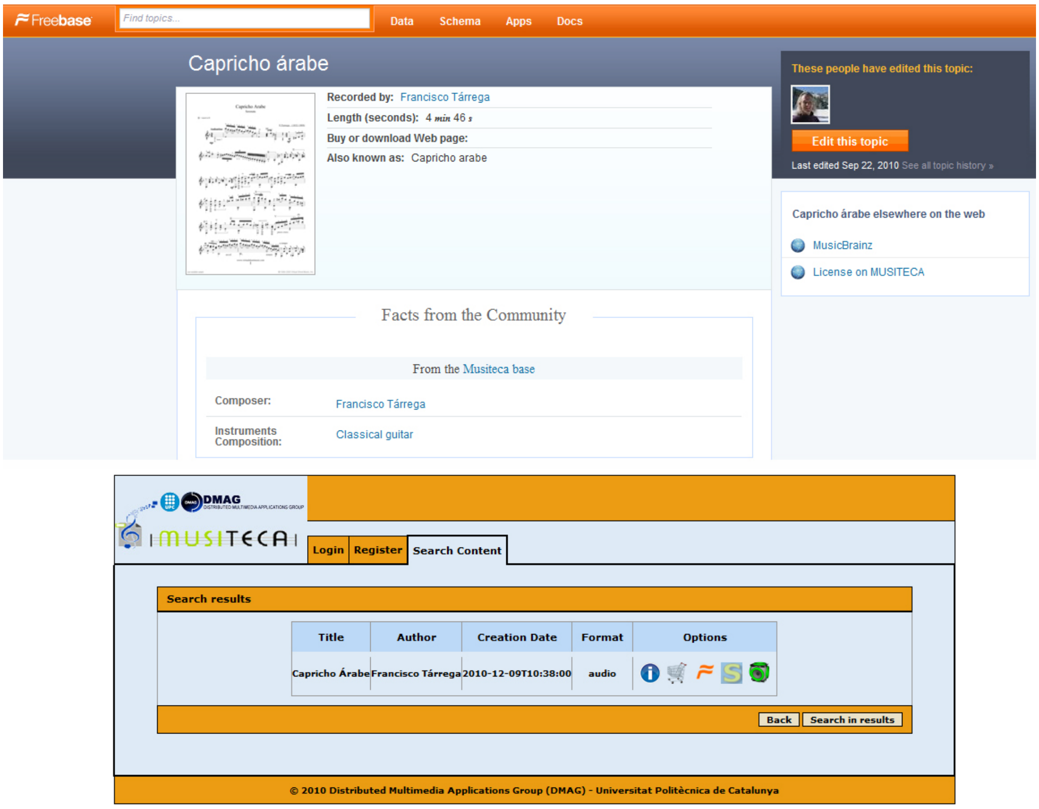
So, the license shown in Figure 58 consists of the issuer (Local TV 1), the principal (Local TV 2), the right (play) and the object to be accessed (Directe_1). In this license there are two conditions, one indicating the time interval where this license is valid (from 1/1/2010 at 10 AM to 14 AM) and another indicating pricing scheme, which is to pay 10 euro per use of the license. According to this license, TV 2 will be able to access live event Directe_1 between 10 and 14 AM the first of January 2010. If it requests more than once during this time interval, TV 2 will have to pay 10 euro each time they do so (a report will be generated each time the authorisation is requested). After 14 AM, if TV 2 requests authorisation of this live event, they will receive a negative authorisation, as conditions will not apply. This as a lot of sense, as, being a live event broadcasted in real time, if TV 2 tries to access it after its finalisation, they will find no transmission at all.

5.7.3.4 Other implementations of scenarios using MIPAMS

This subsection describes other scenarios described in 5.2, implemented with the MIPAMS platform.

5.7.3.4.1 Content licensing scenario using MIPAMS

This scenario was implemented in Musiteca [MUS08a], a research project funded by the Spanish Administration. Figure 59 shows how content is linked from an external site, the Musiteca knowledge base on Freebase (http://musiteca.freebase.com/), which holds information about musical content, to Musiteca's trading portal, where content can be licensed by means of MIPAMS License Service. In this scenario, content access was not a concern, since it was devised for content that was already available or already distributed online.



Capricho árabe

Recorded by: Francisco Tárrega
Length (seconds): 4 min 46 s
Buy or download Web page:
Also known as: Capricho arabe

These people have edited this topic:

Last edited Sep 22, 2010 See all topic history »

Capricho árabe elsewhere on the web
- MusicBrainz
- License on MUSITECA

Facts from the Community

From the Musiteca base

Composer:	Francisco Tárrega
Instruments Composition:	Classical guitar

Search results

Title	Author	Creation Date	Format	Options
Capricho Árabe	Francisco Tárrega	2010-12-09T10:38:00	audio	

© 2010 Distributed Multimedia Applications Group (DMAG) - Universitat Politècnica de Catalunya

Figure 59. Licensing link from Freebase to Musiteca's trading portal.

5.7.3.4.2 DRM-enabled content access control scenario using MIPAMS

This scenario was also implemented in Musiteca. In this project, some of the services conforming MIPAMS (License Service, Reporting Service, Authorization Service, Content Service, Object Registration Service, Search Service and Certification Authority) to implement an electronic content intellectual property registry and trading platform were used. The access to the Musiteca repository was done through a web portal that enabled the registration of content, the definition of different licensing options (i.e. offers), the content acquisition (licensing) and the access to the acquired content for authorised users. Content access was done through a DRM-enabled application, while any action in the system is reported and registered through the Reporting Service.

Part IV. Conclusions

6 Conclusions and Future Work

6.1 Conclusions

The work presented in this thesis has been carried out in the context of different research activities of the DMAG research group of the UPC, including several co-funded projects and the development of international standards. Major European projects, such as AXMEDIS and VISNET-II, have been important in setting up the basis for this work and achieving the first results. MPEG standards development, mainly MPEG-21 and MPEG-M, have also been an unsurpassable environment to contribute and validate the advances in rights and content management services interoperability.

Due to this collaborative working environment, both at the DMAG and at multi-partner projects, it is worth making clear the original contributions done by the candidate, which are detailed in the next two paragraphs.

Regarding the Architectures for RELs Interoperability part, there are several original contributions. The first one is the description of XML rights expressions using UML diagrams and the specification of the translation between MPEG-21 REL and OMA DRM REL. This contribution was completely described in the publication *Translation between XML-Based Rights Expressions Using UML and Relational Models*. The second one is the definition of the mappings between XACML, ODRL and MPEG-21 REL. This major contribution was presented in the publication *An architecture for the interoperability between Rights Expression Languages based on XACML*. The third one is the definition of the connection between the iRMBroker and AXMEDIS in the Video Surveillance scenario, which was contributed to the VISNET-II NoE.

With respect to the Standards-Based Building Blocks for Modelling Content Management Scenarios part, there are also several original contributions done by the candidate. The first one is the definition of the Standards-Based Building Blocks (SB3) and the identification of existing standards for each of them. Also related to this contribution, the mapping between SB3 and existing initiatives was done. This contribution has been completely described in the publication submitted to Multimedia Tools and Applications Journal, *Definition of Standards-Based Building Blocks for Multimedia Content Management*. The second contribution is the definition of the management of rights in the CulturaLive research project, as described in the publication *Management and Distribution of Rights Governed Live Cultural Events*. Finally, the analysis and specification of the different mobile scenarios and applications was also done by the candidate, and the results published into the paper *Implementing Mobile Applications with the MIPAMS Content Management Platform*.

Going into specific conclusions themselves, we should mention that services providing access to, and governed consumption of, multimedia content have increased and improved in the last years. Services definition and rights expression and management are key to their success and, for that reason, a lot of research has been done in these aspects. But, despite the innovations appeared in these areas, this thesis has identified a problem that has to be analysed and solved. This is the lack of interoperability, not only between the rights expression languages but also between services. In this work, both analysis and proposed solutions are presented, divided into two sections that deal with the interoperability between rights expression languages and services in the context of the multimedia content management.

On the first part of the contribution, several alternatives to solve the problem of different ways of representing rights and policies have been developed. With the proposed solutions, it is possible to interoperate between systems using different specifications and formats, using different kind of translations, a centralised system or even an intermediate broker. One solution presented uses an existing extensible standard solution, XACML, as the central DRM system. With the addition of translation modules and the possibility to add any needed condition that does not exist by default in the XACML policy language, it has been virtually able to translate any request coming from a different DRM standard and process it using the existing service architecture. This solution was fully operative but does not have the ability to make two external systems to interact between them. For that reason, this research goes a step forward and defines an architecture called iRMBroker. It represents the idea of having an independent architecture connected to many different DRM systems. When one of these platforms wants to manage a content that it does not understand, asks the broker for a system capable of managing the specific format and rights language. If a connection exists, the broker takes the request and manages it. This solution has been tested in very different application scenarios such as video surveillance or collaborative environments.

On the second part of the contribution, the analysis and implementation of different secure multimedia management systems has been useful to specify high level services and operations that simplify the development of new multimedia content services. To reach the proposed solution, the Standards-Based Building Blocks, the possible scenarios where a content manager could be used were first analysed. With that information, the basic functionalities have been extracted identifying the needs in any system of that kind. Finally, the Blocks that form the complete service architecture have been defined. This is a first step in the way to define interoperable service architecture, with the main modules necessary to design a content management system to support different scenarios of application.

Nevertheless, the solutions proposed may fall on deaf ears as the different standardisation efforts and commercial platforms try to separate as much as possible instead of providing interoperability between them, as all of them like to fight their own battles. This situation leads to market segmentation in the field of digital rights management, very difficult to solve due to competing commercial interests. Content rendering and creation in smartphones, connected cameras or tablets and content rendering in smart televisions increase the complexity of systems trying to provide access control, governance or tracking of user actions done inside these platforms. In this sense, the emergence of new standards like MPEG-M, where also services and its aggregation is defined, should help in solving interoperability issues. In any case, industry has to be involved in order to provide solutions conforming the standard. The final conclusion from this work is that, although several standards and initiatives both for rights expression languages and services are proposed, it is possible to find a solution that provides interoperability to them. The hardest part is to get companies involved and adopt the solution proposed.

The several refereed publications in a variety of research and scientific environments confirm the validity of the results and the future lines of work.

6.2 Publications

The work in the PhD Thesis has resulted in several refereed publications. They are briefly described next.

Definition of Standards-Based Building Blocks for Multimedia Content Management

Llorente, S., Delgado, J., Maroñas, X., Florido, J., Submitted to the Multimedia Tools and Applications Journal. ISSN: 1380-7501.

This paper describes in detail the building blocks presented in this Thesis work, including the relationship with the multimedia middleware architectures analysed, MPEG-M and MIPAMS. It also provides some tests performed over the MIPAMS platform, focusing on the mobile application done using MPEG-M and MIPAMS, already presented in previous sections.

Multimedia Tools and Applications journal is intended for academics, practitioners, scientists and engineers who are involved in multimedia system research, design and applications. Specific areas of interest include: Multimedia Tools and Applications, Prototype multimedia systems and platforms, Home and Education and Training, among others.

Implementing Mobile Applications with the MIPAMS Content Management Platform

Maroñas, X., Llorente, S., Rodríguez, E., Delgado, J., Mobile Multimedia Communications, 7th International ICST Conference, MOBIMEDIA 2011, Cagliari, Italy, September 5-7, 2011, Revised Selected Papers, Volume 79, Springer Berlin Heidelberg, 2012. ISBN: 978-3-642-30418-7.

This paper is a revised version of the one presented to the 7th International ICST Mobile Multimedia Communications Conference (Mobimedia 2011), published by Springer. It presents how the electronic publishing mobile scenario described in [DEL11b] can be implemented using an iPhone mobile application connected to the MIPAMS platform.

The International ICST Mobile Multimedia Communications Conference is focused on the description of innovative multimedia services and applications in mobile environments.

Management and Distribution of Rights Governed Live Cultural Events

Maroñas, X., Llorente, s., Delgado, J. 9th International Workshop for Technical, Economic and Legal Aspects of Business Models for Virtual Goods (Virtual Goods 2011), 28-30 September, Barcelona (Spain). Sponsored by IFIP TC 6, Gesellschaft fuer Informatik e. V., ODRL Initiative, PrestoPrime European Project and W3C.

This paper describes how the MIPAMS platform was used to manage and govern live cultural events in the context of the Culturalive research project. In particular, it describes new conditions to support the governance with live events using MPEG-21 REL licenses and the implementation done to test it. Moreover, it is presented how the MIPAMS platform was connected with the external system providing the live cultural event. Search and authorisation features were supported in this connection.

The VIRTUAL GOODS (International Workshop for Technical, Economic and Legal Aspects of Business Models for Virtual Goods) workshop addresses legal, ethical and security aspects around virtual goods and services, including the ownership rights governing them. This workshop is held with the Open Digital Rights Language (ODRL) workshop and community group meeting, where ODRL description, evolution and usage are discussed.

Access control issues in Social Networks

Carreras, A., Rodríguez, E., Delgado, J., Maroñas, X., 11th International Workshop of the Multimedia Metadata Community (WISMA 2010), 19-20 May, Barcelona (Spain). Sponsored by

Multimedia Metadata Community, Universitat Politècnica de Catalunya · UPC BarcelonaTECH and Segur@ Project.

This paper addressed two main issues on access control in Social Networks: interoperability among policy languages used by Social Networks and the lack of elements in existing standards to describe specific access control policies for Social Networks. The proposed approach was to use access control policies and rights expression languages to solve interoperability issues and to use XACML to negotiate access control policies.

The Workshop on Interoperable Social Multimedia Applications (WISMA) workshop organised by the Multimedia Metadata Community addresses metadata issues surrounding multimedia content, especially those coming from MPEG-7 and MPEG-21. The 2010 edition, which was the 11th one, was devoted to Web 2.0 and Social Networks.

Enhancing Rights Management Systems Through the Development of Trusted Value Networks

Torres, V., Delgado, J., Maroñas, X., Llorente, S., Gauvin, M. 7th International Workshop on Security in Information Systems (WOSIS 2009) , May 2009, Milan (Italy). INSTICC Press 2009, ISBN 978-989-8111-91-3.

This paper describes the work done in the IPOS-DS project, focusing on its real application and commercialisation to support multimedia content life cycle with the definition of new conditions over MPEG-21 REL to describe Rights Over Derivatives (ROD). ROD defines how a content creator can get revenues from the works (adaptation, interpretation, etc.) derived from his/her original works.

The WOSIS workshop is currently preparing its tenth edition. This workshop is focused in the research in Security in Information Systems, addressing security issues of different nature. They have specially focused on the mobile security issues along its history, although other aspects, like cloud computing security or e-Health have also been considered. This workshop is co-located with the International Conference on Enterprise Information Systems (ICEIS), which covers wider research areas, including databases and information systems integration, human computer interaction or artificial intelligence, among others.

A web-based rights management system for developing trusted value networks

Torres, V., Delgado, J., Maroñas, X., Llorente, S., Gauvin, M. 18th International World Wide Web Conference (WWW 2009). April 2009, Madrid (Spain).

This paper presents some implementation details of the IPOS-DS project, describing the portal implemented to support ROD over content registered in the platform.

The World Wide Conference is the one of the most important conferences in the engineering and computer science areas. This year has been held the 22nd edition of this conference, focused, among others, in the areas of web search, semantic web, security and many other related web technologies.

An architecture for the interoperability between Rights Expression Languages based on XACML

Maroñas, X., Rodríguez, E., Delgado, J., 7th International Workshop for technical, economic and legal aspects of business models for virtual goods (Virtual Goods 2009), September 2009, Nancy (France). Sponsored by IFIP TC 6, Gesellschaft fuer Informatik e. V. and ODRL Initiative.

This paper presents a novel solution to achieve interoperability between different rights expression languages (RELs). This solution is based on the use of a profile of XACML to provide interoperability between Rights Expression Languages. As explained in the paper, a XACML policy can be used to express end user licenses terms defined into a REL, such as ODRL or MPEG-21 REL, which is described in detail in the contribution section of this thesis work. The VIRTUAL GOODS (International Workshop for Technical, Economic and Legal Aspects of Business Models for Virtual Goods) workshop addresses legal, ethical and security aspects around virtual goods and services, including the ownership rights governing them. This workshop is held with the Open Digital Rights Language (ODRL) workshop and community group meeting, where ODRL description, evolution and usage are discussed.

Experiencing Digital Rights Management in Mobile Environments

Llorente, S., Delgado, J., Maroñas, X., Barrio, R. 4th International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS 2008). November 2008. Florence (Italy). Firenze University Press. ISBN: 978-88-8453-811-6.

This paper presents the work done inside the AXMEDIS research project in the implementation of mobile applications to support the AXMEDIS DRM environment. In particular, it describes how the mobile applications were implemented and how they interacted with the different running services, which provided authorisation, tracking and security features to the system.

The AXMEDIS conference is focused on the research, developments and applications in the cross media domain, exploring new and innovative technologies to meet the challenges of the sector. It has brought together the experiences and communities coming from the WEDELMUSIC conference series, the MUSICNETWORK and other co-located workshops.

A standard-based approach on the use of contextual information for the adaptation authorisation

Carreras, A., Maroñas, X., Delgado, J. 4th International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS 2008). November 2008. Florence (Italy). Firenze University Press. ISBN: 978-88-8453-811-6.

This paper presents how new conditions can be added MPEG-21 Rights Expression Language licenses to make possible the authorisation of content adaptations. To do so, MPEG-21 Digital Item Adaptation (DIA) Usage Environment Descriptors (UED) have been used together with MPEG-21 REL rights expressions, to describe which are the restrictions to adapt some content (screen width, resolution, etc.). It is also described the authorisation mechanism needed to support the new conditions.

The AXMEDIS conference is focused on the research, developments and applications in the cross media domain, exploring new and innovative technologies to meet the challenges of the sector. It has brought together the experiences and communities coming from the WEDELMUSIC conference series, the MUSICNETWORK and other co-located workshops.

Implementing Mobile DRM with MPEG-21 and OMA

Llorente, S., Delgado, J., Maroñas, X. 5th International Workshop on Security in Information Systems (WOSIS 2007). June 2007, Madeira (Portugal). INSTICC Press 2007 ISBN 978-972-8865-96-2.

This paper presents the work done inside the AXMEDIS, DRM-MM and i2cat Machine research projects in the implementation of mobile applications supporting OMA DRM REL and MPEG-21 related standards. More specifically, it describes how a mobile device supporting OMA DRM REL could be authorised using the authorisation mechanism defined in MPEG-21 REL and report user actions by means of MPEG-21 ER.

The WOSIS workshop is currently preparing its tenth edition. This workshop is focused in the research in Security in Information Systems, addressing security issues of different nature. They have specially focused on the mobile security issues along its history, although other aspects, like cloud computing security or e-Health have also been considered. This workshop is co-located with the International Conference on Enterprise Information Systems (ICEIS), which covers wider research areas, including databases and information systems integration, human computer interaction or artificial intelligence, among others.

Translation between XML-Based Rights Expressions Using UML and Relational Models

Delgado, J., Llorente, S., Barrio, R., Maroñas. X., 2nd International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS 2006). December 2006. Leeds (England). IEEE Press, ISBN: 0-7695-2625-X.

This paper describes how to convert rights expressions from different rights expression languages formalised with XML using Unified Modelling Language and Entity-Relationship models. From these top-level models, it is possible to implement a program that performs the transformation with a high degree of accuracy.

The AXMEDIS conference is focused on the research, developments and applications in the cross media domain, exploring new and innovative technologies to meet the challenges of the sector. It has brought together the experiences and communities coming from the WEDELMUSIC conference series, the MUSICNETWORK and other co-located workshops.

6.3 Future Work

The research work presented in this document could continue into different research lines. Both for updating the proposed solutions and verify its validity with the latest versions of the standards analysed, and for opening new research lines focused on finding new ways to achieve the interoperability and expanding the proposed solutions.

A first possibility is related to the rights expression languages interoperability. In this context, the work to be done would be to verify that the interoperability mechanisms presented are valid for the newly developed rights expression languages, like ODRL version 2.0 [W3C12a] and XACML version 3.0 [OAS13b]. Moreover, other alternatives should be taken into account, such as the recently approved MPEG-21 part 20, which defines a Contract Expression Language (CEL) [ISO13d], for the expression of multimedia contracts.

Regarding MPEG-21 CEL, it could also be considered as a possible Standards Based Building Block (SB3), together with part 21 of MPEG-21 standard, Media Contract Ontology (MCO) [ISO13e]. Therefore, another future line is the identification of standards coming from MPEG and other organisations, to include them into the SB3's. The utilisation of standards is a key issue in the definition of those SB3's, so emerging standards should be continuously evaluated in order to add them to the current ones.

Another possibility is to analyse different standards that are not directly related with the DRM systems. The idea could be to add the necessary service architecture and functionalities to them, and use it to expand an existing standard. Even if this solution may not make interoperable the other DRM systems, to include these features in an open and widely used standard, like HTML, could help to create a common point used by the other solutions to be compatible between them.

Finally, other research line arising from the work done is the application of rights expression and policy languages to privacy protection in different environments like social networks, smart cities, e-health scenarios or specific mobile applications dealing with personal information. Additionally, the research results obtained may be also contributed to the definition of new SB3's to deal with privacy aspects, presenting the solutions proposed to the appropriate forums.

Part V. References and Abbreviations

7 References

[ASI11a] ASIHTTP library. (2011). http://allseeing-i.com/ASIHTTPRequest/

[AXM04a] AXMEDIS (IST-2004-511299). (2004-2008). Automating Production of Cross Media Content for Multi-channel Distribution, http://www.axmedis.org, European Commission.

[BER12a] Berbiela, A. (2012). Desenvolupament d'una aplicació per distribuir continguts digitals generats a dispositius mòbils amb la plataforma DMAG-MIPAMS. Computer Engineering Final Degree Project. Facultat d'Informàtica de Barcelona (FIB). Universitat Politècnica de Catalunya(UPC) – BarcelonaTECH.

[CAR11a] Carreras, et al. (2011). Architectures and Technologies for Adapting Secured Content in Governed Multimedia Applications. IEEE Multimedia, vol. 18 no. 4, 2011.

[CHE76a] Chen, P. (1976). The Entity-Relationship Model--Toward a Unified View of Data. ACM Transactions on Database Systems, Vol. 1, No. 1, March 1976, Pages 9 – 36.

[CHO03a] Chong, C. et al. (2003). LicenseScript: A Novel Digital Rights Language and its Semantics. Proceedings of the Third International Conference WEB Delivering of Music (WEDELMUSIC'03). IEEE Computer Society.

[CON02a] ContentGuard, eXtensible rights Markup Language (XrML). (2002). http://en.wikipedia.org/wiki/XrML.

[CON13a] ContentGuard. (2013). http://www.contentguard.com/.

[CRE12a] Creative Commons. (2013). http://creativecommons.org/licenses/.

[CUL09a] CulturaLive research Project (2009REGIÓ 00024). (2009). Generalitat de Catalunya. http://dmag.ac.upc.edu/proyecto_detalle.php?id_proyecto=41.

[DEL05a] Delgado, J., Prados, J., Rodríguez, E. (2005). Profiles for interoperability between MPEG-21 REL and OMA DRM. 7th International IEEE Conference on E-Commerce Technology 2005 (IEEE CEC 2005).

[DEL06a] Delgado, J., Llorente, S., Barrio, R., Maroñas, X. (2006). Translation between XML-Based Rights Expressions Using UML and Relational Models. Second International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS 2006).

[DEL10a] Delgado, J., Rodríguez, E., Llorente, S. (2010). User's Privacy in Applications provided through Social Networks. Second ACM Workshop on Social Media (WSM 2010).

[DEL11a] Delgado, J., Torres, V., Llorente, S., Rodríguez, E. (2011). Rights management in architectures for distributed multimedia content applications. In: Trustworthy Internet, Springer, 2011, ISBN 978-88-470-1817-4.

[DEL11b] Delgado, J., Llorente, S., Rodríguez, E., Torres-Padrosa, V. (2011) A Mobile Scenario for Electronic Publishing based on the MIPAMS Architecture. In: 15th International Conference on Electronic Publishing - Digital Publication and Mobile Technologies.

[DEL12a] Delgado, J., Llorente, S., Rodríguez, E. (2012). Digital Rights and Privacy Policies Management as a Service. Consumer Communications and Networking Conference (CCNC), IEEE, Digital Object Identifier: 10.1109/CCNC.2012.6181035, Page(s): 527 – 531.

[DEL13a] Delgado, J., Florido, J., Llorente, S. (2013). M28138: MPEG-M demo based on the MIPAMS platform, Input contribution to the 103th MPEG meeting held in Geneva in January 2013.

[DMA13a] Distributed Multimedia Applications Group (DMAG). (2013). http://dmag.ac.upc.edu/.

[DRM05a] Gestión de derechos digitales en e-servicios multimedia (DRM-MM) (2005-2008). http://dmag.ac.upc.edu/proyecto_detalle.php?id_proyecto=23.

[DVB05a] Digital Video Broadcasting (DVB). (2005). Digital Video Broadcasting (DVB) Content Protection and Copy Management (CPCM) System, Usage State Information (USI), SB1497.

[EAR05a] Earnshaw, N. (2005). Consideration of the TV-Anytime RMPI Specification for Inclusion in the Stationary Audio Video Device and Supporting Rights Managed Environment, DMP/dmp0394.

[FLO13a] Florido, J. (2013). Desenvolupament d'una aplicació Android compatible amb el sistema MIPAMS de gestió de drets. Computer Engineering Final Degree Project. Facultat d'Informàtica de Barcelona (FIB). Universitat Politècnica de Catalunya(UPC) – BarcelonaTECH.

[GIL07a] Gestión Integral para el Libro Digital: Derechos de Autor, contenidos y negocio (GILDDA) (2007-2008). http://dmag.ac.upc.edu/proyecto_detalle.php?id_proyecto=27.

[IET02a] Internet Engineering Task Force (IETF) (2002) RFC 3280 Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.

[IPO07a] Intellectual Property Operations System - Digital Shadow (IPOS-DS). (2007-2008). http://dmag.ac.upc.edu/proyecto_detalle.php?id_proyecto=34.

[IPR13a] IPR Systems. (2013). http://www.iprsystems.com/.

[ISO03a] ISO/IEC. (2003). ISO/IEC IS 21000-3 – Part 3: Digital Item Identification.

[ISO04a] ISO/IEC. (2004). ISO/IEC IS 21000:5 – Part 5: Rights Expression Language.

[ISO05a] ISO/IEC. (2005). ISO/IEC IS 21000:2 – Part 2: Digital Item Declaration.

[ISO05b] ISO/IEC. (2005). ISO/IEC 21000:9 – Part 9: File Format.

[ISO06a] ISO/IEC (2006). ISO/IEC IS 21000:4 – Part 4: Intellectual Property Management and Protection Components.

[ISO06b] ISO/IEC. (2006). ISO/IEC IS 21000:15 – Part 15: Event Reporting.

[ISO07a] ISO/IEC. (2007). ISO/IEC IS 21000:7 – Part 7: Digital Item Adaptation.

[ISO07b] ISO/IEC. (2007). ISO/IEC IS 21000:18 – Digital Item Streaming.

[ISO07c] ISO/IEC. (2007). ISO/IEC IS 21000:5 AMD1 - MPEG-21 Rights Expression Language Amendment 1 MAM (Mobile And Optical Media) Profile.

[ISO08a] ISO/IEC. (2008). ISO/IEC IS 15938:12 – Part 12: Query format.

[ISO10a] ISO/IEC (2010). ISO/IEC IS 23006-4:2010 – MPEG-M (MPEG Extensible Middleware) - Part 4: MXM protocols.

[ISO11a] ISO/IEC (2011). ISO/IEC IS 23006-1:2011 – MPEG-M (MPEG Extensible Middleware) - Part 1: MXM architecture and technologies.

[ISO11b] ISO/IEC (2011). ISO/IEC IS 23006-2:2011 – MPEG-M (MPEG Extensible Middleware) - Part 2: MXM API.

[ISO11c] ISO/IEC (2011). ISO/IEC IS 23006-3:2011 – MPEG-M (MPEG Extensible Middleware) - Part 3: MXM Reference software.

[ISO11d] ISO/IEC. (2011). ISO/IEC 9075-1:2011 – Part 1: Framework (SQL/Framework).

[ISO13a] ISO/IEC. (2013). ISO/IEC 23006, Information Technology – Multimedia Service Platform Technologies – MPEG-M.

[ISO13b] ISO/IEC. (2013). ISO/IEC IS 23006:4 – Part 4: Elementary Services.

[ISO13c] ISO/IEC. (2013). ISO/IEC IS 23006:5 – Part 5: Service Aggregation.

[ISO13d] ISO/IEC. (2013). ISO/IEC IS 21000:20 – Contract Expression Language (CEL). Publication pending.

[ISO13e] ISO/IEC. (2013). ISO/IEC IS 21000:21 – Media Contract Ontology (MCO). Publication pending.

[KAL07a] Kalker, T. et al. (2007). The Coral DRM Interoperability Framework. (2007). http://www.coral-interop.org/.

[KUD13a] Kudumakis, P., Sandler, M., Anadiotis, A.-C. G., Venieris, I. S., Difino, A., Wang, X., Tropea, G., Grafl, M., Rodríguez-Doncel, V., Llorente, S., Delgado, J. (2013). N13952 MPEG-M: A Digital Media Ecosystem for Interoperable Applications, MPEG-M Whitepaper. http://mpeg.chiariglione.org/white-papers

[LLO07a] Llorente S., Delgado J., Maroñas X. (2007). Implementing Mobile DRM with MPEG-21 and OMA. 5th International Workshop on Security in Information Systems (WOSIS 2007). Funchal, Madeira (Portugal).

[LLO08a] Llorente, S., Delgado, J., Maroñas, X., Barrio, R. (2008). Experiencing Digital Rights Management in Mobile Environments. AXMEDIS 2008 proceedings. IEEE Computer Society.

[LLO10a] Llorente, S., Rodríguez, E., Delgado, J. (2010). Secure Management of Social Networks Applications Data. Proceedings of the 8th International Workshop for Technical, Economic and Legal Aspects of Business Models for Virtual Goods

[LLO12a] Llorente, S., Delgado, J., Rodríguez, R., Torres-Padrosa, V. (2012). Standards-based Architectures for Content Management, PrePrint, ISSN: 1070-986X, DOI Bookmark: http://doi.ieeecomputersociety.org/10.1109/MMUL.2012.58.

[LLO12b] Llorente, S., Delgado, J., Allasia, W., Gallo, F. (2012). ISO/IEC JTC1/SC29/WG11/M24880. Working document for the preparation of ISO/IEC 2nd DIS 23006-5 (MPEG-M Part 5: Service Aggregation).

[MAC06a] Projecte Machine - i2cat. (2006-2007). Fundació i2cat. http://dmag.ac.upc.edu/proyecto_detalle.php?id_proyecto=24.

[MAR11a] Maroñas, X., Llorente, S., Delgado, J. (2011). Management and distribution of rights governed live cultural events. Proceedings of the 9th International Workshop for Technical, Economic and Legal Aspects of Business Models for Virtual Goods.

[MAR12a] Maroñas, X., Llorente, S., Rodríguez, E., Delgado, J. (2012). Implementing mobile applications with the MIPAMS content management platform, Mobile multimedia communications: 7th International ICST Conference MOBIMEDIA 2011: Cagliari, Italy, September 5–7, 2011: Revised selected papers, Pages 266-280. ISBN: 978-3-642-30418-7.

[MAR13a] Marlin. (2013). http://www.marlin-community.com/.

[MCM09a] Multimedia Content Life Cycle Management (MCM-LC) (2009-2011). http://dmag.ac.upc.edu/proyecto_detalle.php?id_proyecto=40.

[MIP13a] MIPAMS Web application demonstration. (2013). http://dmag1.ac.upc.edu/MIPDemoWeb/.

[MPE13a] ISO/IEC JTC1 SC29/WG11, MPEG. (2013). The Moving Picture Experts Group, http://mpeg.chiariglione.org/.

[MUS08a] Musiteca research Project (TSI-020501-2008-117). (2008). Ministerio de Industria, Turismo y Comercio (Subprograma Avanza I+D). http://dmag.ac.upc.edu/proyecto_detalle.php?id_proyecto=39.

[OAS05a] OASIS. (2005). eXtensible Access Control Markup Language (XACML) v2.0, http://www.oasis-open.org/specs/index.php#xacmlv2.0.

[OAS05b] OASIS. (2005). Security Assertion Markup Language (SAML) version 2.0. http://saml.xml.org/saml-specifications.

[OAS13a] OASIS. (2013). http://www.oasis-open.org/home/index.php.

[OAS13b] OASIS. (2013). eXtensible Access Control Markup Language (XACML) version 3.0. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf.

[OMA04a] Open Mobile Alliance (OMA). (2004). DRM Content Format, Version 1.0, OMA-Download-DRMCF-V1_0-20040615-A.

[OMA05a] Open Mobile Alliance (OMA). (2005). OMA DRM V2.0 Extensions For Broadcast Support, OMA-TS-DRM-XBSV1_0-20051209-D.

[OMA06a] Open Mobile Alliance (OMA). (2006). DRM Rights Expression Language Version 2.0. http://technical.openmobilealliance.org/Technical/release_program/docs/DRM/V2_0-20060303-A/OMA-TS-DRM-REL-V2_0-20060303-A.pdf.

[OMA13a] Open Mobile Alliance (OMA). (2013). http://www.openmobilealliance.org/.

[OMG11a] OMG BPMN 2.0. (2011). Business Process Model and Notation (BPMN) Version 2.0, Object Management Group, January 2011, http://www.omg.org/spec/BPMN/2.0/.

[OMG12a] OMG UML 2.0 (2012). Unified Modeling Language (UML). http://www.uml.org.

[POL04a] Polo J., Prados J. and Delgado J. (2004). Interoperability between ODRL and MPEG-21 REL, First International ODRL Workshop, Vienna (Austria), April 2004, ISBN 1-74064-500-6

[POS85a] Postel, J., Reinolds, J. (1985). File Transfer Protocol (FTP). http://www.ietf.org/rfc/rfc959.txt.

[ROD05a] Rodríguez, E., Prados, J., Delgado, J. (2005). Interoperability between different Rights Expression Languages and Protection Mechanisms. 1st International Conference on Automated Production of Cross Media Content for Multichannel Distribution (AXMEDIS 2005).

[ROD10a] Rodríguez V, Delgado J, Chiariglione F et al (2010). Interoperable Digital Rights Management based on the MPEG Extensible Middleware, Multimedia Tools and Applications. Ed. Springer Netherlands.

[ROD11a] Rodríguez, E., Delgado, J., Alcalde, G. (2011). Protection of patients' privacy by means of standard technologies. Proceedings of the 9th International Workshop for Technical, Economic and Legal Aspects of Business Models for Virtual Goods.

[SAM13a] Samsung (2013). Smart cameras. http://www.samsung.com/es/consumer/cameras-camcorders/smart-cameras/.

[TOR04a] Torres, V., Rodríguez, E, Llorente, S., and Delgado, J. (2004). Architecture and Protocols for the Protection and Management of Multimedia Information. Second International Workshop on Multimedia Interactive Protocols and Systems. MIPS 2004. November 16-19. Grenoble (France). Lecture Notes in Computer Science (LNCS), Vol. 3311, pp. 252–263. Springer Berlin Heidelberg New York. ISBN-10: 3-540-23928-6. ISSN: 0302-9743.

[TOR05a] Torres, V., Rodríguez, E., Llorente, S., Delgado, J. (2005). Trust and Rights in Multimedia Content Management Systems. Proceedings of the IASTED International Conference on Web Technologies, Applications, and Services (WTAS 2005). ACTA Press, Anaheim Calgary Zurich, Pages 89-94.

[TOR08a] Torres, V. (2008). PhD. Dissertation. Contribution to an Architecture for Multimedia Information Management and Protection Based on Open Standards.

[TOR09a] Torres, S., Delgado, J., Maroñas, X., Llorente, S., Gauvin, M. (2009) A web-based rights management system for developing trusted value networks. In: Proceedings of the 18th International World Wide Web Conference – WWW2009

[TOR09b] Torres, S., Delgado, J., Maroñas, X., Llorente, S., Gauvin, M. (2009) Enhancing rights management systems through the development of trusted value networks. In: WOSIS 2009 proceedings. INSTICC Press (2009)

[VIS06a] Networked Audiovisual Media Technologies (VISNET-II) (2006-2009). http://dmag.ac.upc.edu/proyecto_detalle.php?id_proyecto=25.

[W3C99a] World Wide Web Consortium (W3C). (1999). Hypertext Transfer Protocol - HTTP/1.1. http://www.w3.org/Protocols/rfc2616/rfc2616.html.

[W3C02a] World Wide Web Consortium (W3C) Community and Business Groups. (2002). Open Digital Rights Language (ODRL) version 1.1. http://www.w3.org/TR/odrl/.

[W3C07a] World Wide Web Consortium (W3C). (2007). SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). http://www.w3.org/TR/soap12-part1/.

[W3C12a] World Wide Web Consortium (W3C) Community and Business Groups. (2012). Open Digital Rights Language (ODRL) version 2.0. http://www.w3.org/community/odrl/two/model/.

[W3C13a] World Wide Web Consortium (W3C). (2013) eXtensible Stylesheet Language web site, http://www.w3.org/Style/XSL/.

[XAC05a] Xarxa IP Audiovisual de Catalunya (XAC) (2005-2007). Fundació i2cat - Generalitat de Catalunya. http://dmag.ac.upc.edu/proyecto_detalle.php?id_proyecto=22.

[XER98a] Xerox PARC. (1998). Digital Property Rights Language (DPRL). http://xml.coverpages.org/dprl.html

8 Glossary of Abbreviations

A	
AC	Access Control
ACF	Access Control Frameworks
API	Application Programming Interfaces
ATS	Authentication Service
AS	Authorization Service
ASs	Aggregated Services
AXCS	AXMEDIS Certifier and Supervisor
AXMEDIS	Automating Production of Cross Media Content for Multi-channel Distribution
AXOID	AXMEDIS Object Identifier
AXUID	AXMEDIS User Identifier
B	
BAC	Buy and Consume Content aggregated service
BPMN	Business Process Model and Notation
C	
CA	Certification Authority
CCo	Content Consumer
CCr	Content Creator
CDN	Content Delivery Network
CEL	Contract Expression Language
CGWi	Common Generic WSDL Interface
CS	Content Service
D	
DCF	DRM Content Format
DI	Digital Item
DIA	Digital Item Adaptation
DID	Digital Item Declaration
DIS	Digital Item Streaming
DMAG	Distributed Multimedia Applications Group
DPRL	Digital Property Rights Language
DRM	Digital Rights Management
DRM-MM	Digital Rights Management in MultiMedia e-services
E	
EES	Elementary External Services
ER	Event Reporting

ESs	Elementary Services
F	
FTP	File Transfer Protocol
G	
GILDDA	Gestión Integral para el Libro Digital. Derechos de Autor, Contenidos y Negocio
GrGrID	Grant Group Identifier
H	
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
I	
ICEIS	International Conference on Enterprise Information Systems
IEC	International Electrotechnical Commission
IPMP	Intellectual Property Management and Protection
IPOS-DS	Intellectual Property Operations System- Digital Shadow
IPR Systems	Intellectual Property Rights Systems
iRM Broker	iRights Management Broker
ISO	International Organisation for Standardisation
L	
LS	License Service
M	
MA	Mobile Application
MAM	Mobile And optical Media
MCM-LC	Multimedia Content Management-Life Cycle
MIPAMS	Multimedia Information Protection And Management System
MPEG-21	Moving Pictures Experts Group
MPEG-21 CEL	MPEG-21 Contract Expression Language
MPEG-21 MAM	MPEG-21 Mobile And optical Media
MPEG-21 MCO	MPEG-21 Media Contract Ontology
MPEG-M	MPEG Extensible Middleware
MPEGMRA	MPEG-M Registration Authority
MPQF	MPEG Query Format
MSPT	Multimedia Service Platform Technologies
MXM	MPEG eXtensible Middleware
MVCO	Media Value Chain Ontology
O	
OASIS	Organization for the Advancement of Structured Information Standards

ODRL	Open Digital Rights Language
OMA	Open Mobile Alliance
ORCH	Orchestration Engines
ORS	Object Registration Service
P	
P2P	Peer-to-Peer
PAP	Policy Administration Point
PC	Personal Computer
PDA	Personal Digital Assistant
PDP	Policy Decision Point
PEs	Protocol Engines
PEP	Policy Enforcement Point
PIP	Policy Information Point
PMS	Protection Manager Support
PS	Protection Service
R	
RAS	Register and Sell Content Aggregated Service
RBAC	Role Based Access Control
REL	Rights Expression Language
ROD	Rights Over Derivative
RS	Reporting Service
S	
SAML	Security Assertion Markup Language
SB3	Standards-Based Building Blocks
SOAP	Simple Object Access Protocol
SP	Service Provider
SQL	Structured Query Language
SS	Search Service
T	
TEs	Technology Engines
U	
UA	User Application
UED	Usage Environment Descriptors
UDDI	Universal Description, Discovery and Integration
UML	Unified Modelling Language
UPC	Universitat Politècnica de Catalunya
URI	Uniform Resource Identifier

V	
VHS	Video Home System
VISNET-II	Networked Audiovisual Media Technologies
VO	Virtual Organization
W	
W3C	World Wide Web Consortium
WISMA	Workshop on Interoperable Social Multimedia Applications
WM	Workflow Manager
WOSIS	Workshop on Security In Information Systems
WSDL	Web Service Definition Language
X	
XACML	eXtensible Access Control Markup Language
XAC	Xarxa Audiovisual de Catalunya
XML	eXtensible Markup Language
XrML	eXtensible rights Markup Language
XSLT	eXtensible Style Sheets Transformations